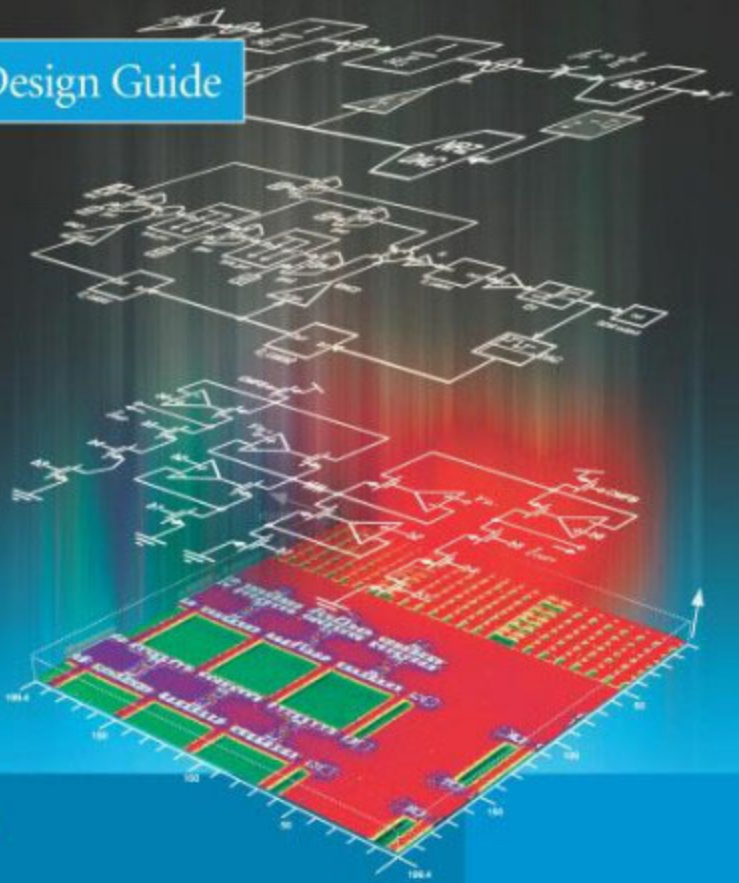
José M. de la Rosa | Rocío del Río

CMOS Sigma-Delta Converters

Practical Design Guide



WILEY

IEEE
IEEE PRESS

CMOS SIGMA-DELTA CONVERTERS

CMOS SIGMA-DELTA CONVERTERS

PRACTICAL DESIGN GUIDE

José M. de la Rosa and Rocío del Río
University of Seville, Spain

WILEY
A John Wiley & Sons, Ltd., Publication

IEEE
IEEE PRESS

This edition first published 2013

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

Library of Congress Cataloging-in-Publication Data

Rosa, Jose M. de la.
CMOS sigma-delta converters : practical design guide / Jose M. de la Rosa and Rocio del Rio.
pages cm
Includes bibliographical references and index.
ISBN 978-1-119-97925-8 (hardback : alk. paper) — ISBN 978-1-118-56922-1 (ebook/epdf) — ISBN 978-1-118-56843-9 (epub) — ISBN 978-1-118-56842-2 (mobi)— ISBN 978-1-118-56923-8 1. Metal oxide semiconductors, Complementary—Design and construction. I. Title.
TK7871.99.M44R668 2013
621.3815′9—dc23

2012041956

A catalogue record for this book is available from the British Library.

ISBN: 978-1-119-97925-8

Set in 10/12pt Times by Laserwords Private Limited, Chennai, India

This book is dedicated to the memory of my son José Manuel
José M. de la Rosa

To my wife Visi, my daughter María, my son Jaime, my parents Carmen and Juan, and my parents-in-law María Luisa and José Antonio

José M. de la Rosa

To my husband Juanan and my son Mario

Rocío del Río

"If you love what you do and are willing to do what it takes, it's within your reach. And it'll be worth every minute you spend alone at night, thinking and thinking about what it is you want to design or build. It'll be worth it, I promise."

Steve Wozniak (*iWoz*, 2006)

Contents

List of Abbreviations

ΣΔ	Sigma-delta
ΣΔM	Sigma-delta modulator
AAF	Antialiasing filter
AC	Alternate current
A/D	Analog-to-digital
ADC	Analog-to-digital converter
ADSL	Asymmetric digital subscriber line
AMPS	Advanced mobile phone system
ASIC	Application-specific integrated circuit
BB	Baseband
BE	Backward-Euler
BGA	Ball grid array
BP	Band-pass
BP-ΣΔM	Band-pass sigma-delta modulator
BPF	Band-pass filter
CAD	Circuit aided design
CDMA	Code division multiple access
CDS	Correlated double sampling
CLA	Clocked averaging
CMFB	Common-mode feedback
CMOS	Complementary MOSFET
CPU	Central processing unit
CS	Current-steering
CT	Continuous-time
CT-ΣΔM	Continuous-time sigma-delta modulator
D/A	Digital-to-analog
DAC	Digital-to-analog converter
DC	Direct current
DCL	Digital cancelation logic
DEM	Dynamic element matching
DMT	Discrete multitone

DNL	Differential nonlinearity
DOR	Digital output rate
DR	Dynamic range
DRC	Design rule checker
DSP	Digital signal processor
DT	Discrete-time
DT-ΣΔM	Discrete-time sigma-delta modulator
DVB	Digital video broadcasting
DVB-H	Digital video broadcasting - handheld
DWA	Data weighted averaging
EDGE	Enhanced data-rates for global evolution
ELD	Excess loop delay
ENOB	Effective number of bits
ESD	Electrostatic discharge
FE	Forward-Euler
FFT	Fast Fourier transform
FIR	Finite impulse response
FOM	Figure of merit
FS	Full scale
GB	Gain-bandwidth product
GPS	Global positioning system
GSM	Global system for mobile communications
GUI	Graphic user interface
H-ΣΔM	Hybrid sigma-delta modulator
HD	Harmonic distortion
HDL	Hardware description language
HRZ	Half-delay return-to-zero
IBN	In-band noise power
IC	Integrated circuit
IF	Intermediate frequency
IIP	Input-referred intercept point
IIR	Infinite impulse response
IIT	Impulse-invariant transformation
ILA	Individual level averaging
IM	Intermodulation distortion
INL	Integral nonlinearity
I/O	Input–output
I/Q	Inphase/quadrature
ISI	Intersymbol interference
ITF	Integrator transfer function
LDI	Lossless discrete integrator
LNA	Low-noise amplifier
LP	Low-pass

LP-ΣΔM	Low-pass sigma-delta modulator
LPE	Layout parasitic extractor
LPF	Low-pass filter
LSB	Least significant bit
LTE	Long term evolution
LTCC	Low-temperature co-fired ceramic
LTI	Linear time-invariant
LVS	Layout versus schematic
MASH	Multistage noise shaping
MEX	MATLAB executable
MiM	Metal-insulator-metal
MoM	Metal-oxide-metal
MOS	Metal-oxide-semiconductor
MOSFET	MOS field-effect transistor
MOST	MOS transistor
MR	Multirate
MR-ΣΔM	Multirate sigma-delta modulator
MTPR	Multitone power ratio
nMOS	n-channel MOSFET
NRZ	Nonreturn-to-zero
NTF	Noise transfer function
OL	Overload level
OS	Output swing
OSR	Oversampling ratio
OTA	Operational transconductance amplifier
PCB	Printed circuit board
PDF	Probability density function
PDM	Pulse-density modulation
PLL	Phase-locked loop
pMOS	p-channel MOSFET
PSD	Power spectral density
PWL	Piece-wise linear
PWM	Pulse-width modulation
QFP	Quad flat package
RF	Radio frequency
RTF	Resonator transfer function
RZ	Return-to-zero
SAR	Successive approximation register
SC	Switched-capacitor
SDR	Software-defined radio
SFDR	Spurious-free dynamic range
S/H, S&H	Sample and hold

SI	Switched-current
SMASH	Sturdy multistage noise shaping
SMD	Surface-mount device
SNR	Signal-to-noise ratio
SNDR	Signal-to-noise-plus-distortion ratio
SQNR	Signal-to-quantization-noise ratio
SoC	System on chip
SR	Slew rate
STF	Signal transfer function
TDC	Time-to-digital converter
TEQ	Time-encoding quantizer
THD	Total harmonic distortion
UGBW	Unity gain bandwidth
UMTS	Universal mobile telecommunications system
USMR	Upsampling multirate
USTF	Unity signal transfer function
VCO	Voltage-controlled oscillator
VCRO	Voltage-controlled ring oscillator
V/I	Voltage-to-current
WCDMA	Wideband code division multiple access
WiMAX	Worldwide interoperability for microwave access
WLAN	Wireless local area network

Preface

Sigma-Delta modulators ($\Sigma\Delta$Ms) have become one of the best choices for the implementation of analog/digital interfaces integrated in CMOS technologies. Compared to other kinds of analog-to-digital converters (ADCs), $\Sigma\Delta$Ms cover the widest conversion region of the resolution-versus-bandwidth plane, being the most efficient solution to digitize very diverse types of signals in an increasing number of application scenarios, which span from high-resolution low-bandwidth data conversion (like digital audio, sensor interfaces, and instrumentation) to ultra-low power biomedical systems and medium-resolution broadband wireless communications. This versatility, together with their robustness and their simplicity in many practical situations, has motivated that more and more engineers today consider $\Sigma\Delta$Ms as a first choice for their research projects and their industrial products.

The first idea underlying the operation of $\Sigma\Delta$Ms was patented by Cutler in 1960 [1], although its application to the construction of data converters was first reported in the published literature by Inose *et al* in 1962 [2]. The operation of $\Sigma\Delta$Ms is relatively simple to describe, although sometimes difficult to analyze. Essentially, the fundamental principle behind $\Sigma\Delta$Ms is based on the combination of two signal processing techniques, namely *oversampling* and quantization *noise shaping*. The former consists of taking the signal samples at a higher rate than the one dictated by the Nyquist sampling theorem. These samples are commonly quantized with a large error by using a low-resolution quantizer. The resulting oversampled quantization error is filtered in the modulator feedback loop, so that its frequency spectrum is *shaped* in such a way that a large portion of its power is pushed out of the signal band, where it is removed by a digital filter. The outcome of the combined action of oversampling and noise shaping allows $\Sigma\Delta$Ms to achieve a high-precision digitization by using a low-resolution coarse quantizer. Therefore, unlike other kinds of ADC architectures that require high-precision analog circuits, $\Sigma\Delta$Ms trade the accuracy of their analog circuitry by the speed of digital signal processing, thus achieving a higher degree of insensitivity to circuit error mechanisms and potentially benefiting from CMOS technology evolution towards the nanometer scale.

Prompted by the mentioned benefits and fueled by technology downscaling and industry trends in consumer digital electronics, the original concept of noise shaping described above has evolved over the last five decades through many $\Sigma\Delta$M generations, giving rise to a pleiad of architectures, circuit and system design techniques, and a number of Integrated Circuits (ICs), which have pushed the state-of-the-art on $\Sigma\Delta$Ms forward, yielding to innovative research results and successful industry products. All these advances and research works have lead (and continue doing so) to a vast amount of technical

literature. Indeed, since the publication of pioneer works like the widely cited papers written by Candy [3, 4] and Boser and Wooley [5], the number of publications has increased significantly including hundreds of patents, thousands of research papers, some tutorial papers [6–8], as well as tens of introductory and specialized monographs [9–29]. However, with so much material and abundance of technical information published, many designers—particularly novel designers and also some experienced designers focused on a specific subtopic of ΣΔMs—may become disoriented and lost. This has motivated some authors to put all these pieces of information together in a comprehensive and systematic way.

Apart from the earlier books aiming to catalogue the existing publications on ΣΔMs [9], one of the first attempts to present a guide for ΣΔM designers is the book edited by Norsworthy *et al* in 1997 [10], also known as "the yellow book" by the ΣΔM community. This book deals with a number of important subjects in ΣΔMs and it was contributed by a number of experts in the field, thus making it more difficult to present its contents in a coherent and consistent way. With this objective in mind, some authors have put their effort on writing tutorial monographs dealing with the systematic design of ΣΔMs.

Among others, the book written by Schreier and Temes, published in 2005 [21], often named "the green book", has become one of the most popular books on ΣΔ converters. This book provides an excellent and comprehensive treatment of ΣΔMs, their operating principles, and main architectures, presenting several design examples constructed using the well-known Schreier's MATLAB toolbox [30]. Although some examples of continuous-time (CT) circuit implementations are given, the book mainly focuses on system-level description, considering a switched-capacitor (SC) implementation. Some other remarkable examples are the book written by Medeiro *et al* in 1999 [13]—focused on the systematic design of SC ΣΔMs—and the book of Ortmanns and Gerfers [22] published in 2006, which is still one of the most complete monographs on CT ΣΔMs reported to date. All these books, as well as other monographs reported in the technical literature, give a partial view of ΣΔMs, paying more attention to some particular aspects of the design of ΣΔMs, and/or a type of architecture, circuit technique, or application.

In this scenario, this book attempts to cover some of these knowledge gaps, by providing a broader and systematic description of the *universe* of ΣΔMs, their diverse types of architectures, circuit techniques, analysis and synthesis methods and CAD tools, as well as their practical design considerations. From this perspective, the book has a twofold purpose. First, it constitutes a unique monograph that results from compiling the enormous number of technical and research works reported to date on the topic of ΣΔMs, and presents the results of such a compilation in a didactical, pedagogical, and intuitive style. The second main objective and a key feature of this book is to serve as a *practical guide for designers*, putting emphasis on explaining practical design issues involved in the whole design flow of ΣΔMs: from specifications to chip implementation and characterization. To this end, a *top-down* approach is followed, presenting the contents in a hierarchical way; that is, going from theoretical fundamentals, system-level design equations, and behavioral models to circuit, transistor-level, and physical implementation, in order to provide readers the necessary understanding and insight into the recent advances, trends, and challenges involved in the design of state-of-the-art ICs.

Indeed, it is the top-down approach adopted in this book that inspires the hierarchical way in which the contents are organized. Thus, following this introduction, Chapter 1

begins from top, giving an introductory survey of ΣΔMs, their principles of operation, fundamental architectures, analysis and synthesis methods, as well as a taxonomical description of the diverse variety of practical ΣΔM topologies, the nature of signal (low-pass and band-pass), as well as the dynamics involved (either discrete-time or continuous-time). In this chapter ΣΔMs are considered ideal systems, except for their inherent quantization error. Chapter 2 descends one level in the modulator hierarchy to analyze the effect of main circuit error mechanisms as well as architectural and timing nonidealities, considered in both SC and CT circuit implementations. The mathematical models, analytical procedures, and design guidelines described in this chapter provide sufficient understanding of the main practical problems affecting the performance of ΣΔMs in practice.

The knowledge derived from the first two chapters is presented in this book as an essential part of the systematic *top-down/bottom-up* synthesis methodology of ΣΔMs, that is described in Chapter 3. This chapter analyzes different strategies for the high-level modeling and simulation of ΣΔMs, focusing on the so-called *behavioral modeling and simulation* techniques. A step-by-step procedure to develop efficient behavioral models in the MATLAB/SIMULINK environment is described and illustrated with a number of examples of the main ΣΔM building-block models. As an application, a time-domain behavioral simulator named SIMSIDES, is described and applied to the high-level sizing and verification of some case studies. The contents of this chapter are extended and complemented in Appendixes A and B. Appendix A gives a more complete user guide of SIMSIDES and Appendix B provides an overview of all behavioral models and libraries included in this simulator.

Chapter 4 moves farther down from the system-level description given in previous chapters to the circuit and physical level. This chapter provides a number of necessary design recommendations and practical recipes to complete the design flow of a ΣΔM, showing the step-by-step methodology to transform a behavioral-model description given in Chapter 3 into an electrical schematic initially based on macromodels, and then implemented with transistors, and finally concluding the design cycle with the layout and chip implementation. Plenty of examples, case studies, and simulation test benches are given to illustrate the practical issues and design considerations addressed in the chapter, that cover from electrical analysis and simulation using SPICE-like simulators to layout design considerations, chip prototyping, and experimental measurements of ΣΔMs in the laboratory.

To conclude the book, Chapter 5 gives an overview of the state-of-the-art ΣΔM ICs, comparing their performance with Nyquist-rate ADCs. Overall, more than 300 state-of-the-art IC references have been studied in detail and considered in this review, including papers published from 1990 to June 2012. Therefore, following the practical philosophy that inspires this book, the diverse families of state-of-the-art ΣΔM architectures and circuit techniques are exhaustively analyzed and compared to extract practical and empirical design guidelines from the statistical data, trying to identify the incoming trends, design challenges, as well as the solutions proposed by cutting-edge ICs that are in the frontiers of ΣΔMs.

The book contents are addressed and structured for a large audience: from senior designers who want to acquire a deeper and updated insight into ΣΔMs, to nonexperienced undergraduate students who are looking for a comprehensive, uniform, and self-contained

reference into this hot topic. Bearing this in mind, the style and main purpose of the book is to serve also as an educational and reference textbook for undergraduate and graduate students. Indeed, the book is based on a number of graduate courses given by the authors, including master and doctorate degree programs, invited lectures, and IEEE conference tutorials. All these materials have been adapted and updated so that a large portion of the book can be also used (and indeed it has been used) in both undergraduate and graduate courses.

However, in spite of the *encyclopedic* nature of the book, it is impossible to give an exhaustive description of all the topics contained in the thousands of publications dealing with ΣΔMs. Instead, the book tries to cover the main subtopics, providing sufficient insight to understand the other ones, that are just overviewed and sometimes even omitted. In order to try to palliate these unavoidable deficiencies, an exhaustive list of specific references is included at the end of each chapter. Overall, the book contains around 500 selected references in order to guide readers to increase their understanding of the diverse research topics dealing with the ΣΔ world.

The huge amount of information contained in the book is complemented and updated with a number of electronic resources, that have been prepared by the authors and are freely available on the Web. To this purpose, all the data analyzed in the state-of-the-art survey presented in Chapter 5 have been collected in a spreadsheet, which is available at http://www.imse-cnm.csic.es/∼jrosa/CMOS-SDMs-Survey-IMSE-JMdelaRosa.xlsx. This database is periodically kept up to date and aims to be a complement to the popular Murmann's ADC survey data collection [31]. In addition, a fully functional version of the time-domain behavioral simulator SIMSIDES is freely available on demand at http://www.imse-cnm.csic.es/simsides. The simulator includes a number of examples, containing the case studies presented in the book and many more examples and demos. Apart from the SIMSIDES software, the majority of examples and test benches of different CAD tools used throughout the book are also available on the Web at www.wiley.com/go/delarosa_converters.

Last but not least, it is important to mention that the aforementioned web sites will be regularly updated with new pieces of information and updated material related to the state-of-the-art ΣΔM ICs, SIMSIDES examples and demos, as well as new inputs provided by us and hopefully by our readers. Therefore, your feedback is very important and very welcomed!

We hope that you enjoy reading this book as much as we have enjoyed writing it.

José M. de la Rosa and Rocío del Río
Sevilla, October 2012

References

[1] C. C. Cutler, "Transmission System Employing Quantization," US Patent No. 2,927,962, 1960.
[2] H. Inose, Y. Yasuda, and J. Murakami, "A Telemetering System by Code Modulation—Δ-Σ Modulation," *IRE Transactions on Space Electronics and Telemetry*, vol. **8**, pp. 204–209, September 1962.
[3] J. Candy and O. J. Benjamin, "The Structure of Quantization Noise from Sigma-Delta Modulation," *IEEE Transactions on Communications*, pp. 1316–1323, 1981.
[4] J. Candy, "A Use of Double Integration in Sigma-Delta Modulation," *IEEE Transactions on Communications*, vol. **33**, pp. 249–258, March 1985.

[5] B. E. Boser and B. A. Wooley, "The Design of Sigma-Delta Modulation Analog-to-Digital Converters," *IEEE Journal of Solid-State Circuits*, vol. **23**, pp. 1298–1308, December 1988.
[6] P. M. Aziz et al., "An Overview of Sigma-Delta Converters," *IEEE Signal Processing Magazine*, vol. **13**, pp. 61–84, January 1996.
[7] I. Galton, "Delta-Sigma Data Conversion in Wireless Transceivers," *IEEE Transactions on Microwave Theory and Techniques*, vol. **50**, pp. 302–315, January 2002.
[8] J. M. de la Rosa, "Sigma-Delta Modulators: Tutorial Overview, Design Guide, and State-of-the-Art Survey," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. **58**, pp. 1–21, January 2011.
[9] J. Candy and G. Temes, *Oversampling Delta-Sigma Data Converters: Theory, Design and Simulation*, IEEE Press, 1991.
[10] S. R. Norsworthy, R. Schreier, and G. C. Temes, *Delta-Sigma Data Converters: Theory, Design and Simulation*, IEEE Press, 1997.
[11] J. Cherry and W. Snelgrove, *Continuous-Time Delta-Sigma Modulators for High-Speed A/D Conversion*, Kluwer Academic Publishers, 1999.
[12] J. V. Engelen and R. van de Plassche, *BandPass Sigma-Delta Modulators: Stability Analysis, Performance and Design Aspects*, Kluwer Academic Publishers, 1999.
[13] F. Medeiro, B. Pérez-Verdù, and A. Rodríguez-Vázquez, *Top-Down Design of High-Performance Sigma-Delta Modulators*, Kluwer Academic Publishers, 1999.
[14] V. Peluso, M. Steyaert, and W. Sansen, *Design of Low-Voltage Low-Power CMOS Delta-Sigma A/D Converters*, Kluwer Academic Publishers, 1999.
[15] S. Rabii and B. A. Wooley, *The Design of Low-Voltage, Low-Power Sigma-Delta Modulators*, Kluwer Academic Publishers, 1999.
[16] L. Breems and J. H. Huijsing, *Continuous-Time Sigma-Delta Modulation for A/D Conversion in Radio Receivers*, Kluwer Academic Publishers, 2001.
[17] Y. Geerts, M. Steyaert, and W. Sansen, *Design of Multi-bit Delta-Sigma A/D Converters*, Kluwer Academic Publishers, 2002.
[18] J. M. de la Rosa, B. Pérez-Verdú, and A. Rodríguez-Vázquez, *Systematic Design of CMOS Switched-Current Bandpass Sigma-Delta Modulators for Digital Communication Chips*, Kluwer Academic Publishers, 2002.
[19] M. Kozak and I. Kale, *Oversampling Delta-Sigma Modulators*, Springer, 2003.
[20] O. Bajdechi and J. Huising, *Systematic Design of Sigma-Delta Analog-to-Digital Converters*, Kluwer Academic Publishers, 2004.
[21] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*, Wiley-IEEE Press, 2005.
[22] M. Ortmanns and F. Gerfers, *Continuous-Time Sigma-Delta A/D Conversion: Fundamentals, Performance Limits and Robust Implementations*, Springer, 2006.
[23] K. Philips and A. H. M. van Roermund, *Sigma Delta A/D Conversion for Signal Conditioning*, Springer, 2006.
[24] R. del Río, F. Medeiro, B. Pérez-Verdú, J. M. de la Rosa, and A. Rodríguez-Vázquez, *CMOS Cascade $\Sigma\Delta$ Modulators for Sensors and Telecom: Error Analysis and Practical Design*, Springer, 2006.
[25] L. Yao, M. Steyaert, and W. Sansen, *Low-Power Low-Voltage Sigma-Delta Modulators in Nanometer CMOS*, Springer, 2006.
[26] P. G. R. Silva and J. H. Huijsing, *High Resolution IF-to-Baseband $\Sigma\Delta$ ADC for Car Radios*, Springer, 2008.
[27] R. H. van Veldhoven and A. H. M. van Roermund, *Robust Sigma Delta Converters*, Springer, 2011.
[28] A. Morgado, R. del Río, and J. M. de la Rosa, *Nanometer CMOS Sigma-Delta Modulators for Software Defined Radio*, Springer, 2011.
[29] E. Janssens and A. van Roermund, *Look-Ahead Based Sigma-Delta Modulation*, Springer, 2011.
[30] R. Schreier, The Delta-Sigma Toolbox v. 7.3. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/19, 2009.
[31] B. Murmann, ADC Performance Survey 1997–2012. [Online]. Available: http://www.stanford.edu/~murmann/adcsurvey.html, 2012.

Acknowledgements

We would like to express our deepest gratitude to Prof. Belén Pérez-Verdú and Prof. Angel Rodríguez-Vázquez from the University of Sevilla. We are indebted to them for so many things we learnt, not only related to the contents of this book, but also many life values and skills we have put in practice throughout our professional careers, first as their PhD students, then as senior researchers, and later as university professors. Our warm and special acknowledgement and dedication of this book goes to them.

We are also particularly grateful to our colleagues at the Institute of Microelectronics of Sevilla and the Department of Electronics and Electromagnetism of the University of Sevilla, especially to Prof. Francisco Fernández, Dr. Manuel Delgado, Dr. Rafael Castro, Dr. Rafael Domínguez, Dr. Oscar Guerra, Mr. Joaquín Ceballos, and Mr. Miguel Angel Lagos. Among them, we are especially grateful to our friend Dr. Fernando Medeiro, with whom we took our first steps in the fascinating world of sigma-delta modulators, working side by side in a number of research and industrial projects. Thank you very much Fernando!

We would also like to acknowledge our past and present undergraduate and graduate students, especially Dr. Jesús Ruíz-Amaya, Mr. Rafael Romay, Dr. Ramón Tortosa, Dr. Alonso Morgado, Dr. Edwin Becerra, Mr. Gerardo García, Mr. Luis Guerrero-Linares, and Mr. Sohail Asghar. Some pieces of material used for the preparation of this book have been adapted from their PhD works, including a number of examples and case studies. Specifically, the first version of SIMSIDES—the time-domain behavioral simulator described in this book— was developed by Dr. Ruíz-Amaya as part of his graduate thesis.

We would like to give our thanks to the staff at Wiley, in particular to the editors Liz Wingett, Laura Bell, Richard Davies, and Peter Mitchell, as well as to the anonymous reviewers of the book proposal for their constructive and valuable comments and suggestions, which became very useful for us to make this manuscript a reality, trying to keep a good trade-off between accuracy (resolution) and time to publication (speed). This is indeed a well-known compromise for us as analog designers. We also appreciate our discussions over all these years with so many experts in the field of sigma-delta converters, with whom we had the opportunity to meet at diverse conferences and technical meetings. Many of them were sponsored by the IEEE Circuits and Systems Society and the IEEE Solid-State Circuits Society.

In addition to all these great people, our work has been always supported by a number of National and International Research and Industrial Projects. More recently, our research

was funded by the Spanish Ministry of Economy and Competitiveness (with support from the European Regional Development Fund) under contract TEC2010-14825/MIC.

Finally, our most grateful thanks go for our families. None of this would have been possible without their love, patience, confidence, and support. Thank you so much for everything!

1

Introduction to ΣΔ Modulators: Basic Concepts and Fundamentals

This chapter is conceived as an introduction to ΣΔ analog-to-digital converters (ADCs). Their operation principle consists in combining oversampling, quantization error processing, and negative feedback for improving the effective resolution of a coarse quantizer. These basic concepts are presented in Section 1.1 and their effects on the performance of ΣΔ converters are compared with Nyquist-rate converters. Section 1.2 shows the basic architecture, ideal behavior, and performance metrics of ΣΔ converters, and sketches the architectural alternatives to enhance their resolution.

Before presenting practical topologies for the implementation of ΣΔ modulation, the large variety of the existing ΣΔ realizations is briefly classified in Section 1.3 according to the type of modulator architecture (single loops or cascades), the circuit techniques employed (discrete time (DT) or continuous time (CT)), and the nature of the signals being converted (low pass (LP) or band pass (BP)). Starting from the case of DT, LP, single-bit ΣΔ modulators, the implications of these different alternatives are then presented in an incremental way.

Section 1.4 is dedicated to single-loop ΣΔ architectures. Second- and higher-order single-loops are considered, taking into account issues related to their practical implementation and problems not addressed by linear models, such as instabilities. Cascade ΣΔ topologies are covered in Section 1.5. In Section 1.6 the topological study is extended to ΣΔ modulators using multibit embedded quantizers, analyzing their pros and cons. Techniques to circumvent the disadvantages, such as dynamic element matching (DEM) or dual quantization, are revised.

The conversion of BP signals is covered in Section 1.7, taking into account its typical application in digital radio receivers. The basic techniques for synthesizing DT, BP ΣΔ modulators are presented, together with practical aspects for their implementation. Finally, Section 1.8 addresses the realization of CT ΣΔ modulators, discussing their advantages compared to DT ones and the existing alternatives for the loop filter and the feedback implementation.

CMOS Sigma-Delta Converters: Practical Design Guide, First Edition. José M. de la Rosa and Rocío del Río.

1.1 Basics of A/D Conversion

ADCs are electronic systems that perform the transformation of analog signals—which are continuous in time and in amplitude—into digital signals—which are discrete in both time and amplitude. Figure 1.1 illustrates the general block diagram of an ADC intended for the conversion of LP signals, which essentially consists of an *antialiasing filter* (AAF), a *sampler*, and a *quantizer*. First, the analog input signal $x_a(t)$ of the ADC passes through the AAF, an LP analog filter than prevents out-of-band components from folding over the signal bandwidth B_w during the subsequent sampling, what would corrupt the signal information. The resulting band-limited signal $x(t)$ is sampled at a rate f_s by the S/H circuit, thus yielding a DT signal $x_s(n) = x(nT_s)$, where $T_s = 1/f_s$. Finally, the values of $x_s(n)$ are quantized using N bits, so that each continuous-valued input sample is mapped onto the closer discrete-valued level out of the 2^N that cover the input range, yielding the converter digital output $y_d(n)$.

As shown in Figure 1.1, the fundamental processes involved in the A/D conversion are sampling and quantization, whose implications are discussed in the following text.

1.1.1 Sampling

The sampling process performs the continuous-to-discrete transformation of the input signal in time and imposes a limit on the bandwidth of the analog input signal. According to the Nyquist theorem, to prevent information loss, $x(t)$ must be sampled at a minimum rate of $f_N = 2B_w$, often referred to as the *Nyquist frequency*. On the basis of this criterion, ADCs in which analog input signal is sampled at the minimum rate ($f_s = f_N$) are called *Nyquist-rate ADCs*. Conversely, ADCs in which $f_s > f_N$ are called *oversampling ADCs*. How much faster than required the input signal is sampled is expressed in terms of the *oversampling ratio* (OSR), defined as

$$\mathrm{OSR} = \frac{f_s}{2B_w} \tag{1.1}$$

Whether oversampling is used or not in an ADC has a noticeable influence on the requirements of its AAF. As in Nyquist-rate ADCs the input signal bandwidth B_w coincides with $f_s/2$, aliasing will occur if $x_a(t)$ in Figure 1.1 contains frequency components above $f_s/2$. High-order analog AAFs are thus required to implement sharp transition bands capable of removing out-of-band components with no significant attenuation of the signal band, as illustrated in Figure 1.2a for the LP case. Conversely, as $f_s/2 > B_w$ in oversampling ADCs, the replicas of the input signal spectrum that are created by the sampling process are farther apart than in Nyquist-rate ADCs. As illustrated in Figure 1.2b,

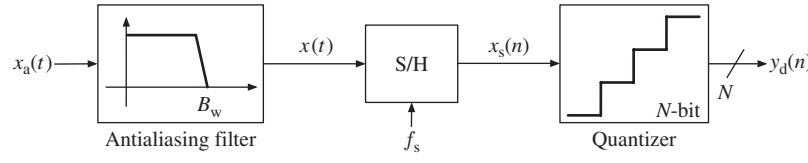


Figure 1.1 General block diagram of an A/D converter. A Nyquist-rate ADC is assumed.

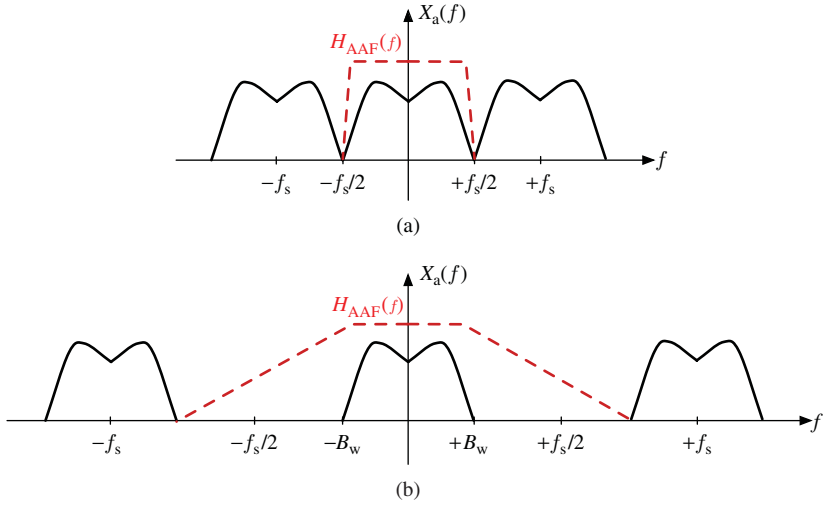


Figure 1.2 Antialiasing filter for (a) Nyquist-rate ADCs and (b) oversampling ADCs.

frequency components of the input signal in the range $[B_w, f_s - B_w]$ do not alias within the signal band, so that the filter transition band can be smoother, what greatly reduces the order required for the AAF and simplifies its design.

1.1.2 Quantization

The quantization process also introduces a limitation on the performance of an ideal ADC, because an error is generated while performing the continuous-to-discrete transformation of the input signal in amplitude, commonly referred to as *quantization error*. The operation of quantizers is illustrated in Figure 1.3. As a matter of example, Figure 1.3c depicts the I/O characteristic of a quantizer with $N = 2$, although results apply to a generic N-bit quantizer. Input amplitudes within the full-scale input range $[-X_{FS}/2, +X_{FS}/2]$ are *rounded* to 1 out of the 2^N different output levels, which are usually encoded into a binary digital representation. If these levels are equally spaced, the quantizer is said to be *uniform* and the separation between adjacent output levels is defined as the *quantization step*

$$\Delta = \frac{Y_{FS}}{2^N - 1} \tag{1.2}$$

where Y_{FS} stands for the full-scale output range. As X_{FS} and Y_{FS} are not necessarily equal, the quantizer may exhibit a gain different from unity, as indicated in Figure 1.3c by the slope k_q. As shown in Figure 1.3e, the quantizer operation thus inherently generates a *rounding* error that is a nonlinear function of the input. Note that, if $q(n)$ is kept within the range $[-X_{FS}/2, +X_{FS}/2]$, the quantization error $e(n)$ is bounded within $[-\Delta/2, +\Delta/2]$. The former input range is known as the *nonoverload region* of the quantizer, as opposed to ranges with $|q(n)| > \Delta/2$, for which the magnitude of $e(n)$ grows monotonously.

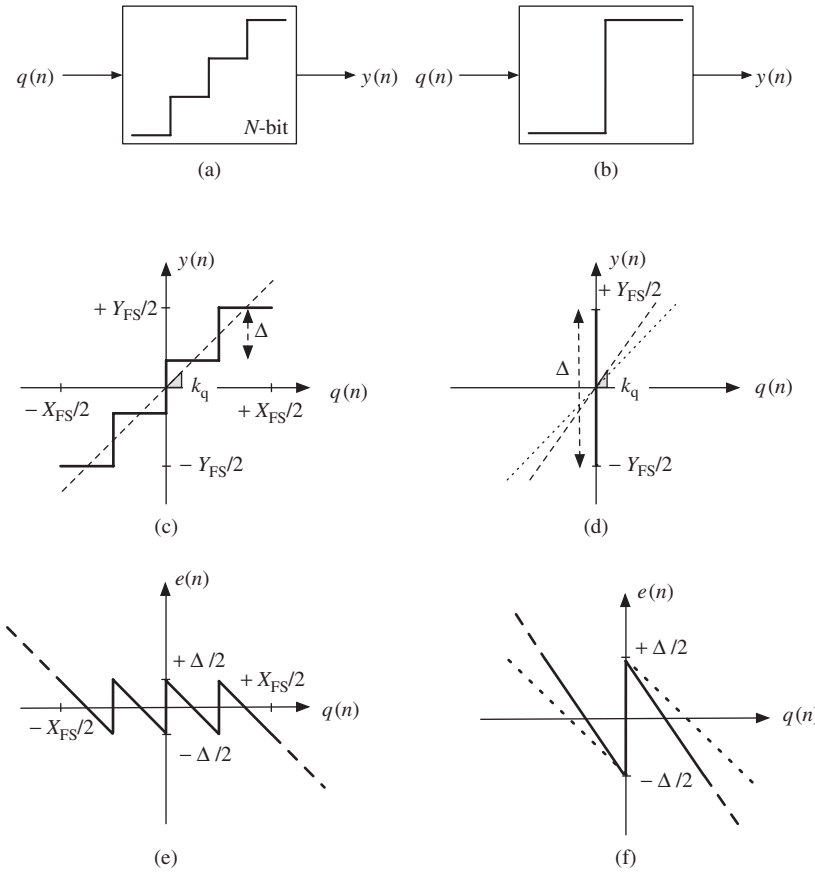


Figure 1.3 Illustration of the quantization process: (a) multibit quantizer block, (b) single-bit quantizer block, (c) I/O characteristic of a multibit quantizer, (d) I/O characteristic of a single-bit quantizer, (e) multibit quantization error, and (f) single-bit quantization error.

Figure 1.3 also shows the operation of a single-bit quantizer ($N = 1$). Note from Figure 1.3d that, compared to the multibit case, the output of a single-bit quantizer is determined by the input sign only, regardless of its magnitude. Therefore, the gain k_q is undefined and can be arbitrarily chosen.

1.1.3 Quantization White Noise Model

In practice, an ideal quantizer as that shown in Figure 1.4a is often modeled using the linear scheme in Figure 1.4b if several assumptions are made on the statistical properties of the quantization error [1–3]. As already shown in Figure 1.3e, the quantization error $e(n)$ is systematically determined by the quantizer input signal $q(n)$. Nevertheless, if $q(n)$ is assumed to change randomly from sample to sample within the range $[-\Delta/2, +\Delta/2]$, $e(n)$ will also be uncorrelated from sample to sample. Under these requirements, the quantization error can be viewed as a *random process with a uniform probability distribution*

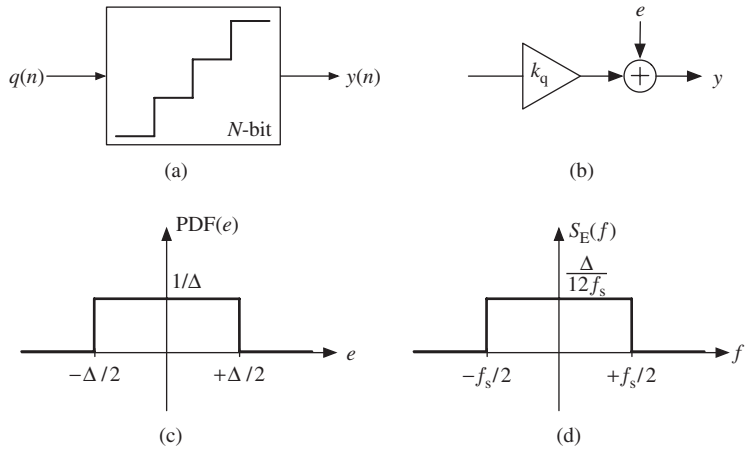


Figure 1.4 Quantization noise: (a) multibit quantizer block, (b) equivalent linear model with additive white noise, (c) probability density function (PDF), and (d) power spectral density.

in the range $[-\Delta/2, +\Delta/2]$, as illustrated in Figure 1.4c. The power associated to the quantization error can thus be computed as

$$\overline{e^2} = \sigma_e^2 = \int_{-\infty}^{+\infty} e^2 \mathrm{PDF}(e)\, \mathrm{d}e = \frac{1}{\Delta} \int_{-\Delta/2}^{+\Delta/2} e^2\, \mathrm{d}e = \frac{\Delta^2}{12} \tag{1.3}$$

The former assumption implies that, as illustrated in Figure 1.4d, the power of the quantization error will also be *uniformly* distributed in the range $[-f_s/2, +f_s/2]$, yielding

$$\overline{e^2} = \int_{-\infty}^{+\infty} S_E(f)\mathrm{d}f = S_E \int_{-f_s/2}^{+f_s/2} \mathrm{d}f = \frac{\Delta^2}{12} \tag{1.4}$$

so that the power spectral density (PSD) of the quantization error in that range is

$$S_E = \frac{\overline{e^2}}{f_s} = \frac{\Delta^2}{12 f_s} \tag{1.5}$$

These assumptions are collectively known as the *additive white noise approximation* of the quantization error and allow the representation of a quantizer, which is deterministic and nonlinear, with the random linear model in Figure 1.4b, in which $y(n) = k_q q(n) + e(n)$ with $e(n)$ being a *quantization noise*.[1]

On the basis of this approximation of the quantization error to a white noise, the performance of ideal ADCs can be easily evaluated. For a Nyquist ADC, in which

[1] Although the assumptions underlying the additive white noise approximation are hardly met in practice and are not strictly valid, it is commonly used in ADC design and usually yields good results—the larger the number of bits in the quantizer, the better. Even though strictly speaking, it is not valid for stand-alone single-bit quantizers, it is also employed in the design of single-bit ΣΔ modulators [4].

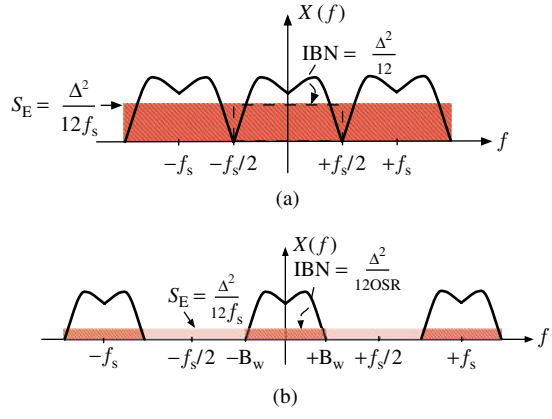


Figure 1.5 Quantization noise in (a) Nyquist-rate ADCs and (b) oversampling ADCs.

$f_s = 2B_w$, all the quantization noise power falls inside the signal band and passes to the ADC output as a part of the input signal itself, as illustrated in Figure 1.5a. Conversely, if an oversampled signal is quantized, because $f_s > 2B_w$, only a fraction of the total quantization noise power lies within the signal band, as illustrated in Figure 1.5b. The *in-band noise power* (IBN) caused by the quantization process in an ideal oversampling ADC is thus,

$$\text{IBN} = \int_{-B_w}^{+B_w} S_E(f)\,\mathrm{d}f = \int_{-B_w}^{+B_w} \frac{\Delta^2}{12f_s}\,\mathrm{d}f = \frac{\Delta^2}{12\text{OSR}} \tag{1.6}$$

so that the larger the OSR, the smaller the IBN.[2]

The *dynamic range* (DR) of an ideal ADC can be determined as the ratio of the output power at the frequency of an input sinusoid with maximum amplitude to the in-band quantization noise power:

$$\text{DR(dB)} = 10\log_{10}\left(\frac{P_{\text{sig,out,max}}}{\text{IBN}}\right) \tag{1.7}$$

From Figure 1.3c, the maximum input amplitude in the nonoverload region of an N-bit quantizer is $X_{FS}/2$ and its corresponding output power can be approximated to [5],

$$P_{\text{sig,out,max}} \approx \frac{(2^N\Delta/2)^2}{2} = 2^{2N-3}\Delta^2 \tag{1.8}$$

so that, using Equations 1.6 and 1.8, the DR of an ideal oversampling ADC yields

$$\text{DR(dB)} \approx 6.02N + 1.76 + 10\log_{10}(\text{OSR}) \tag{1.9}$$

[2] Note that Equation 1.6 for the IBN of oversampling ADCs also holds true for Nyquist ADCs, just considering OSR = 1. The same applies for subsequent expressions derived from Equation 1.6.

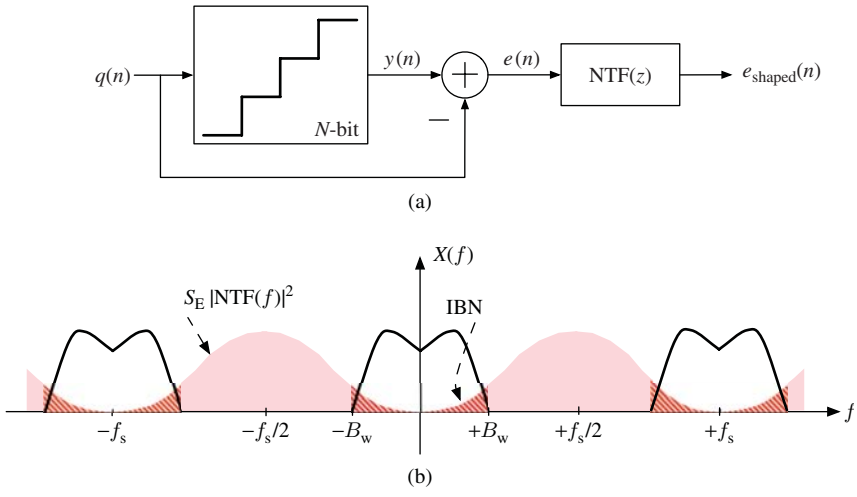


Figure 1.6 Quantization noise shaping: (a) conceptual block diagram and (b) effect on the in-band noise of an oversampling noise-shaping ADC.

Note that, for a Nyquist ADC—that is, OSR = 1 in Equation 1.9—each additional bit in the quantizer results in a DR increase of approximately 6 dB. For an oversampling ADC, the DR further increases with the OSR by approximately 3 dB/octave, so that using for instance an OSR of 4 is similar to having one extra bit in the N-bit quantizer.

1.1.4 Noise Shaping

An approach to further increase the accuracy of an oversampling ADC is *shaping* the quantization white noise in the frequency domain—that is, filtering it—in such a way that most of its power lies outside the signal band. This is illustrated in Figure 1.6a, where the quantization noise is conceptually obtained by subtracting the quantizer input signal $q(n)$ from its output $y(n)$ and then passes through a filter transfer function, usually called *noise transfer function* (NTF).

For quantizers working on LP signals, the NTF is of high-pass type and can be easily obtained from a differentiator filter, with a Z-domain transfer function given by

$$\mathrm{NTF}(z) = (1 - z^{-1})^L \tag{1.10}$$

where L stands for the filter *order*. Taking into account that $z = e^{sT_s} = e^{j2\pi f/f_s}$, the magnitude of the pure-differentiator NTF in Equation 1.10 can be approximated for low frequencies to

$$|\mathrm{NTF}(f)| = |1 - e^{-j2\pi f/f_s}|^L = \left[2\sin\left(\frac{\pi f}{f_s}\right)\right]^L \approx \left(\frac{2\pi f}{f_s}\right)^L, \quad \text{for } f \ll f_s \tag{1.11}$$

so that the power due to the shaped quantization noise that lies within the signal band (Figure 1.6b) yields

$$\mathrm{IBN} = \int_{-B_\mathrm{w}}^{+B_\mathrm{w}} S_\mathrm{E}(f)|\mathrm{NTF}(f)|^2 \,\mathrm{d}f \approx \frac{\Delta^2}{12}\frac{\pi^{2L}}{(2L+1)\mathrm{OSR}^{(2L+1)}} \tag{1.12}$$

Using Equations 1.8 and 1.12, the DR of an ideal oversampling noise-shaping ADC can be obtained as

$$\mathrm{DR(dB)} \approx 6.02N + 1.76 + 10\log_{10}\left(\frac{2L+1}{\pi^{2L}}\right) + (2L+1)10\log_{10}(\mathrm{OSR}) \tag{1.13}$$

Note that, in comparison with Equation 1.9, if oversampling is used in combination with noise shaping, the DR increases with OSR by approximately $3(2L+1)$ dB/octave.

1.2 Basics of Sigma-Delta Modulators

Contrary to the ADCs discussed so far, which are open loop systems from a control perspective, Sigma-Delta ($\Sigma\Delta$) ADCs rely on a feedback path to achieve a closed-loop control of the quantization error. The fundamentals on how the shaping of quantization noise is implemented in practice, as well as the basic architecture, performance metrics, and ideal behavior of oversampling noise-shaping ADCs is presented in the following sections.

1.2.1 Topology of $\Sigma\Delta$ ADCs

Figure 1.7 illustrates the basic block diagram of a $\Sigma\Delta$ ADC intended for the conversion of LP signals, which consists of the following:

- *Antialiasing filter (AAF)*, which band limits the analog input signal to avoid aliasing during its subsequent sampling. As discussed in Section 1.1.1, oversampling considerably relaxes the attenuation requirements of the AAF, so that smooth transition bands are usually sufficient compared to Nyquist-rate ADCs.
- *Sigma-Delta modulator* ($\Sigma\Delta M$), in which the oversampling and quantization of the band-limited analog signal take place. The quantization noise of the embedded B-bit quantizer is shaped in the frequency domain by placing an appropriate loop filter $H(z)$ before it and closing a negative feedback loop around them. Low-resolution quantizers, with B typically in the range 1–5 bit, are sufficient for obtaining small IBN and high accuracy in the A/D conversion.
- *Decimation filter*, in which a high-selectivity digital filter sharply removes the out-of-band spectral content of the $\Sigma\Delta$M output and thus most of the shaped quantization noise. The decimator also reduces the data rate from f_s down to the Nyquist frequency, while increasing the word length from B to N bits to preserve resolution.

The $\Sigma\Delta$ modulator is the block that has most influence on the performance of the ADC, basically because it is responsible for the sampling and quantization processes that

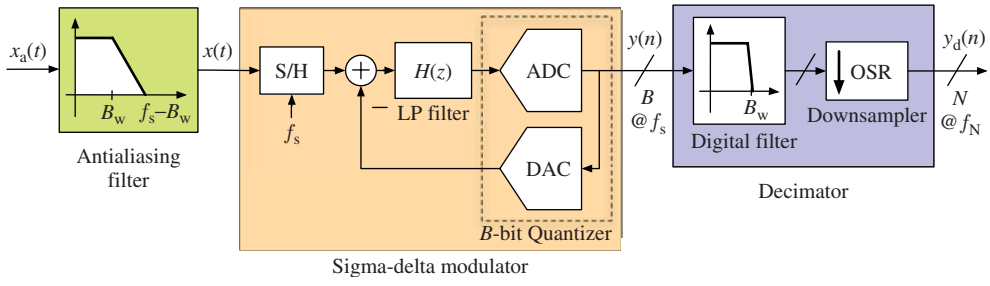


Figure 1.7 General block diagram of a ΣΔ ADC. A low-pass discrete-time ΣΔM is assumed.

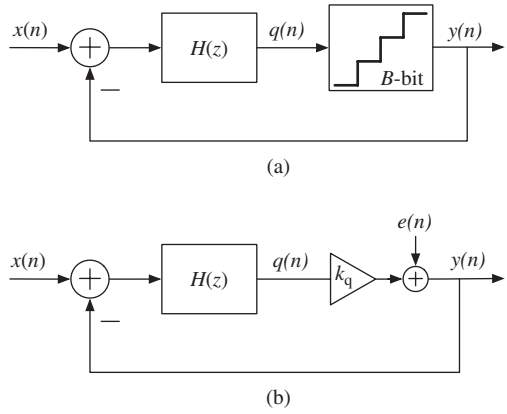


Figure 1.8 ΣΔ modulator: (a) block diagram and (b) ideal linear model.

ultimately limit the accuracy of the A/D conversion. We will focus on this block from now on, although it must be kept in mind that a ΣΔ ADC is more than a ΣΔM!

1.2.2 Signal Processing in ΣΔMs

The basic scheme of a ΣΔ modulator consists of a loop filter $H(z)$ and a B-bit quantizer in a feedback loop, as shown in Figure 1.8a [6]. Let us consider that the gain of the loop filter is large within the signal band and small outside it. Owing to the action of negative feedback, the analog input signal x and the analog version of the ΣΔM output y will practically coincide within the signal band, so that the error signal $x - y$ in this closed-loop system is very small within the signal band. As the B-bit quantizer is uniform, most of the differences between the input and the output of the ΣΔM will be placed at higher frequencies, so that the quantization noise is shaped in the frequency domain and most of its power is pushed outside the signal band.

Using the linear additive white noise model in Figure 1.4b for the embedded quantizer, the ΣΔM in Figure 1.8a can be modeled as the two-input (x and e) one-output (y) linear system in Figure 1.8b, which is described in the Z-domain as

$$Y(z) = \mathrm{STF}(z)X(z) + \mathrm{NTF}(z)E(z) \tag{1.14}$$

where STF and NTF stand for the signal and noise transfer functions, respectively given by

$$\mathrm{STF}(z) = \frac{k_q H(z)}{1 + k_q H(z)}, \quad \mathrm{NTF}(z) = \frac{1}{1 + k_q H(z)} \tag{1.15}$$

Note that, if the loop filter is designed such that $|H(f)| \gg 1$ within the signal band, then $|\mathrm{STF}(f)| \approx 1$ and $|\mathrm{NTF}(f)| \ll 1$; that is, the quantization noise is ideally canceled while the input signal is perfectly transferred to the output.

For the conversion of LP signals, the simplest loop filter $H(z)$ that exhibits the desired frequency performance is an integrator,

$$\mathrm{ITF}(z) = \frac{z^{-1}}{1 - z^{-1}} \tag{1.16}$$

that, in combination with an embedded quantizer with $k_q = 1$, leads to a ΣΔM whose output is given by

$$Y(z) = z^{-1} X(z) + (1 - z^{-1}) E(z) \tag{1.17}$$

and builds up a first-order, high-pass shaping of the quantization noise—see Equation 1.10. For the sake of illustration, Figure 1.9 shows the output signal of a first-order ΣΔM with a embedded 3-bit (8-level) quantizer for a sinusoidal input signal. Note that, due to the combined action of oversampling and negative feedback, the modulator output is a pulse-density modulated (PDM) signal whose local average tracks the input signal value within adjacent code transitions.

1.2.3 Performance Metrics of ΣΔMs

Contrary to Nyquist-rate ADCs, whose performance is mainly characterized by static performance metrics—that is, monotonicity, gain and offset errors, differential nonlinearity (DNL), and integral nonlinearity (INL) [5]—ΣΔ ADCs' characteristics are typically

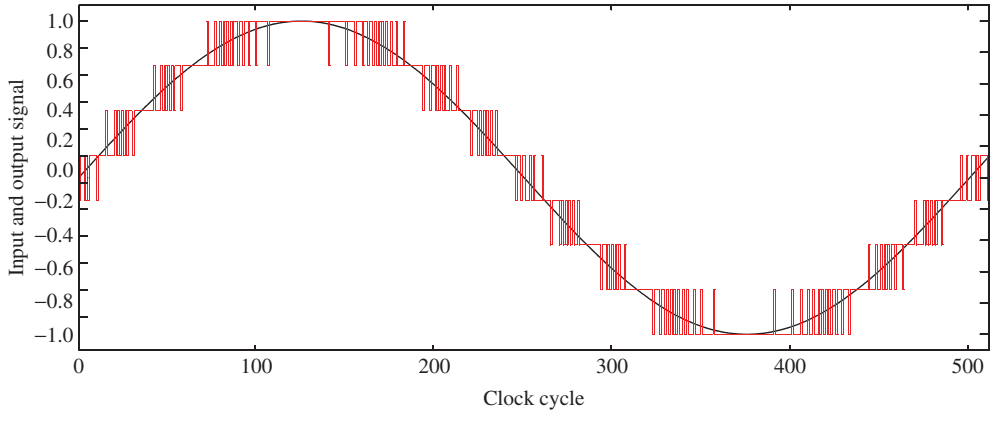


Figure 1.9 PDM output signal of a first-order, ΣΔ modulator with an embedded 3-bit quantizer for an input sinusoid.

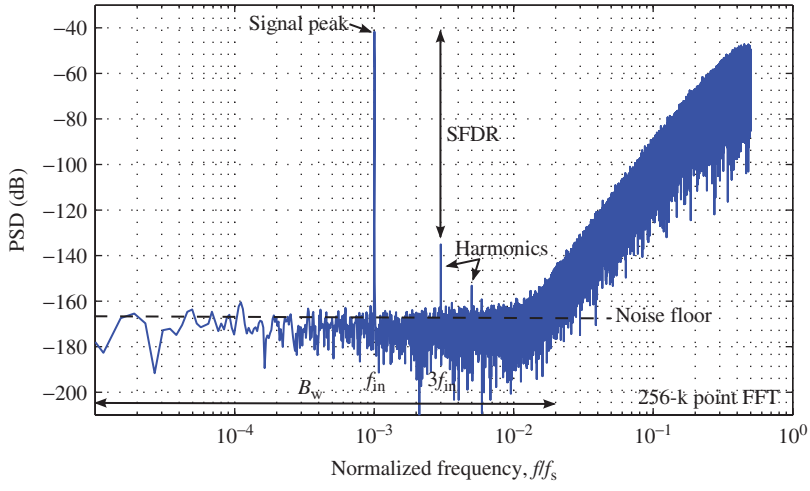


Figure 1.10 Illustration of a typical output spectrum of a ΣΔ modulator and its main characteristics. A low-pass ΣΔM is assumed.

measured using dynamic performance metrics, which are obtained from the frequency-domain representation of the time-domain digital output sequence. The latter thus requires the computation of the fast Fourier transform (FFT) of a finite-length output sequence with a specific *windowing function*, as will be discussed in Chapter 4. From that power spectrum representation of a ΣΔM output sequence, some spectral metrics are directly measured and other noise and power metrics are derived.

Figure 1.10 illustrates an exemplary spectrum of a ΣΔM output sequence when a sinusoidal signal with frequency f_{in} is applied at its input. The main characteristics of the spectrum are highlighted; for example, the length of the digital sequence from which the FFT has been computed, the output signal peak at the frequency f_{in} corresponding to the converted signal, etc. As will be discussed in Chapter 2, nonidealities of the circuitry used for implementing the ΣΔM deviate in practice the output spectrum from a *purely* shaped quantization noise. On the one hand, linear errors give rise to a noise floor, as well as to a degradation of the shaping order. On the other, nonlinear errors generate distortion, which is typically noticeable for large input amplitudes, but *submerged* under the noise floor for small input signal amplitudes. Spectral metrics such as the *spurious-free dynamic range* (SFDR)—that is, the ratio of the signal power to the strongest spectral tone [5]—can be directly measured from the output modulator spectrum, as shown in Figure 1.10.

Noise and power metrics are derived from the ΣΔM output spectra by integration over the signal bandwidth and are typically collected in a single plot as shown in Figure 1.11. These metrics are usually the most important measures and comprise the following:

- *Signal-to-noise ratio* (SNR), which is the ratio of the output power at the frequency of an input sinusoid to the *uncorrelated* IBN:

$$\mathrm{SNR(dB)} = 10\log_{10}\left(\frac{P_{\mathrm{sig,out}}}{\mathrm{IBN}}\right) \tag{1.18}$$

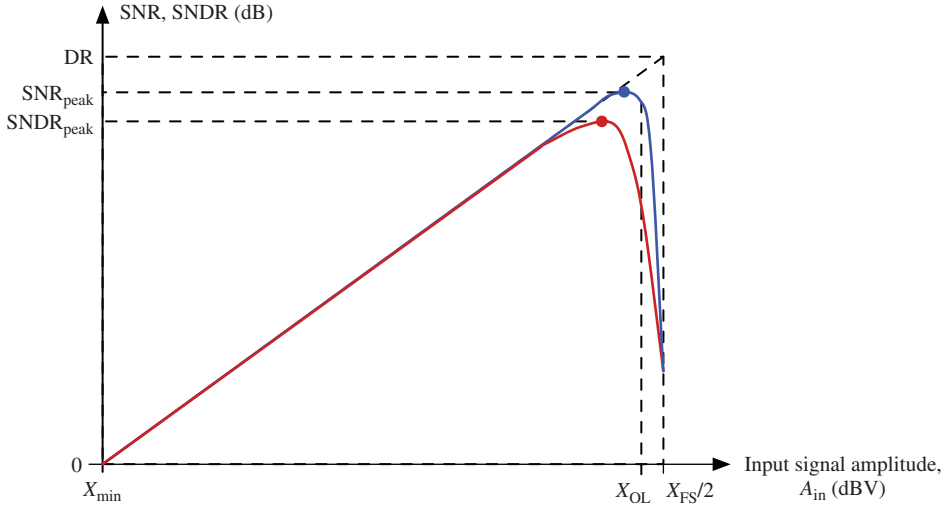


Figure 1.11 Illustration of the performance metrics of a ΣΔ modulator on a typical SNR curve.

It accounts for the modulator linear performance only, so that the in-band power associated to harmonics of the input signal is not considered as part of the IBN for SNR computation.

If an ideal ΣΔM is considered and only the in-band quantization noise is accounted for in the IBN computation, the term *signal-to-quantization-noise ratio* (SQNR) is often employed.

- *Signal-to-noise-plus-distortion ratio* (SNDR), which is defined as the ratio of the output power at the frequency of an input sinusoid to the total IBN power, also accounting for possible harmonics at the ΣΔM output. As illustrated in Figure 1.11, this makes a typical SNDR curve to deviate from the SNR curve only for large input amplitudes, for which the generated distortion is noticeable. Therefore, the output spectra from which the SNDR curve is computed are typically obtained by applying an input signal at $f_{in} \leq B_w/3$ (for LP ΣΔMs), so that at least the second and third harmonics lie within the signal band.
- *Dynamic range* (DR), which can be defined as the ratio of the output power at the frequency of an input sinusoid with maximum amplitude to the output power for a small input amplitude for which SNR = 0 dB; that is, so it cannot be distinguished from the error. Ideally, a sinusoid with maximum amplitude at the modulator input will provide an output sinusoid sweeping the full-scale range Y_{FS} of the embedded quantizer, so that

$$\mathrm{DR(dB)} = 10\log_{10}\left(\frac{P_{\mathrm{sig,out,max}}}{\mathrm{IBN}}\right) = 10\log_{10}\left[\frac{(Y_{\mathrm{FS}}/2)^2}{2\mathrm{IBN}}\right] \tag{1.19}$$

- *Effective number of bits* (ENOB): as the DR of an ideal N-bit Nyquist-rate converter is given by Equation 1.9 with OSR = 1, a similar expression can be established for ΣΔMs

$$\mathrm{ENOB(bit)} = \frac{\mathrm{DR(dB)} - 1.76}{6.02} \tag{1.20}$$

where ENOB can be defined as the number of bits needed for an ideal Nyquist-rate ADC to achieve the same DR as the ΣΔ ADC. The performance of oversampled ΣΔ converters and Nyquist-rate ADCs can thus be compared in a simple way [7].

Instead of the DR, the peak SNDR is also often used in Equation 1.20 to express the accuracy of the A/D in a ΣΔ modulator in bits.

- *Overload Level* (OL): as illustrated in Figure 1.11, the SNR of a ΣΔ modulator increases monotonously with the input signal amplitude (A_{in}), but sharply drops for input amplitudes close to half of the full-scale input range of the embedded quantizer ($X_{FS}/2$) due to its overload and the associated IBN increase. The overload level is considered to define the maximum input amplitude for which the ΣΔM still operates correctly and can almost be arbitrarily defined, but it is typically chosen as the amplitude for which the SNR drops 6 dB below the peak SNR [8].

1.2.4 Performance Enhancement of ΣΔMs

The output of an ideal LP Lth-order ΣΔ modulator in the Z-domain can be considered to be

$$Y(z) = \mathrm{STF}(z)X(z) + \mathrm{NTF}(z)E(z) = z^{-L}X(z) + (1 - z^{-1})^{L}E(z) \tag{1.21}$$

where $|\mathrm{STF}(f)| = 1$ and the NTF builds up an Lth-order high-pass shaping of the quantization noise of the embedded quantizer. If a B-bit quantizer is employed, the DR of the ΣΔM can be obtained from Equations 1.12 and 1.19 to ideally yield

$$\mathrm{DR(dB)} = 10\log_{10}\left(\frac{P_{\mathrm{sig,out,max}}}{\mathrm{IBN}}\right) \approx 10\log_{10}\left[\frac{3}{2}(2^{B}-1)^{2}\frac{(2L+1)\mathrm{OSR}^{(2L+1)}}{\pi^{2L}}\right] \tag{1.22}$$

taking into account that $Y_{FS} = (2^{B} - 1)\Delta$—see Equation 1.2 —and considering quantization noise as the only contribution to the IBN.

Note from Equation 1.22 that the DR of a ΣΔ modulator is ideally determined by the values of L, OSR, and B, which can thus be considered as the *three key parameters* that define the ΣΔM at the high level. The pros and cons of increasing the DR of a ΣΔ modulator by increasing each of these parameters are briefly discussed in the following:

- *High-order ΣΔ modulators*. The accuracy of the A/D conversion can be considerably improved by increasing the noise-shaping order, because a larger fraction of the total quantization noise power will be pushed out of the signal band. Figure 1.12 illustrates the ideal noise-shaping functions of orders ranging from 1 to 5. The case $L = 0$—that is, no shaping—is also included for comparison purposes. The DR enhancement if L is increased in one for a given OSR can be obtained from Equation 1.22 to be

$$\Delta\mathrm{DR(dB)} \approx 10\log_{10}\left[\frac{2L+3}{2L+1}\left(\frac{\mathrm{OSR}}{\pi}\right)^{2}\right] \tag{1.23}$$

This means that, for instance, the DR of a fourth-order ΣΔM with OSR = 32 is ideally 21.3 dB (3.5 bit) larger than that of a third-order ΣΔM.

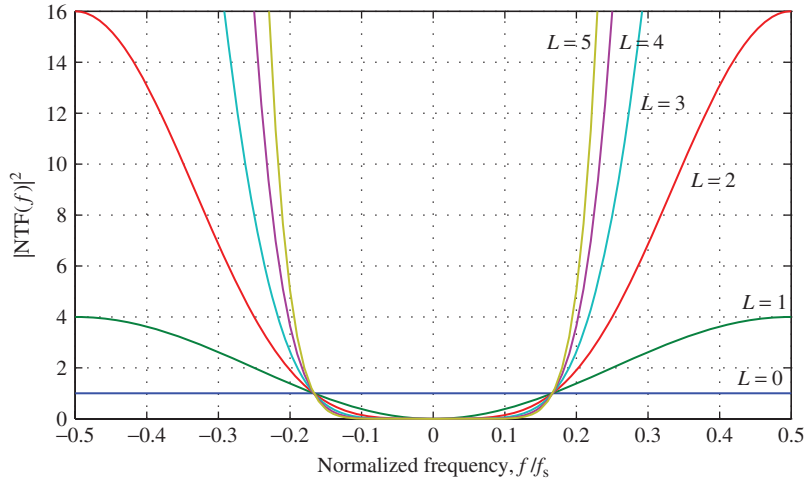


Figure 1.12 Illustration of the shaping of quantization noise as a function of frequency in a ΣΔM. NTF is given by Equation 1.10 and L stands for the noise-shaping order.

However, as will be discussed in Section 1.4.2, the use of high-order ($L > 2$) loop filters gives rise to stability problems in a ΣΔM. Although these problems can be circumvented, the DR of a high-order ΣΔM will in practice be smaller than predicted in Equation 1.22.

- *High OSR ΣΔ modulators*. Figure 1.13 shows the ideal DR as a function of OSR for noise-shaping orders ranging from 0 (no shaping) to 5 and assuming a single-bit embedded quantizer ($B = 1$). As illustrated, the combination of oversampling and noise-shaping considerably enhances the ΣΔM performance for OSR > 4. Note from Equation 1.22 that the DR of an ideal Lth-order ΣΔ modulator increases with OSR in $3(2L + 1)$ dB/octave.

  However, for a given conversion bandwidth B_w, the OSR cannot be arbitrarily increased, because it leads to a higher sampling frequency f_s for the operation of the ΣΔ circuitry. The latter, if achievable in practice for a given technological process, leads to larger power consumption.
- *Multibit ΣΔ modulators*. An increase in B leads to a decrease of the quantization step Δ and thus to a reduction of the quantization noise power. Each additional bit in the embedded quantizer of a ΣΔM is considered to typically yield a 6 dB (1 bit) improvement on the DR [9].

  However, a multibit embedded quantizer requires a multilevel DAC to close the negative feedback loop in the ΣΔM. Contrary to a two-level feedback DAC ($B = 1$), which is inherently linear, a multilevel DAC will in practice be nonlinear to some extent. As noticeable from Figure 1.17, the DAC nonlinearity will be directly added to the ΣΔM input and will thus appear at the output, as $|\mathrm{STF}(f)| \approx 1$ within the signal band. Therefore, the linearity required in a multibit DAC equals in practice that wanted for the ΣΔ modulator. This point will be further discussed in Section 1.6.

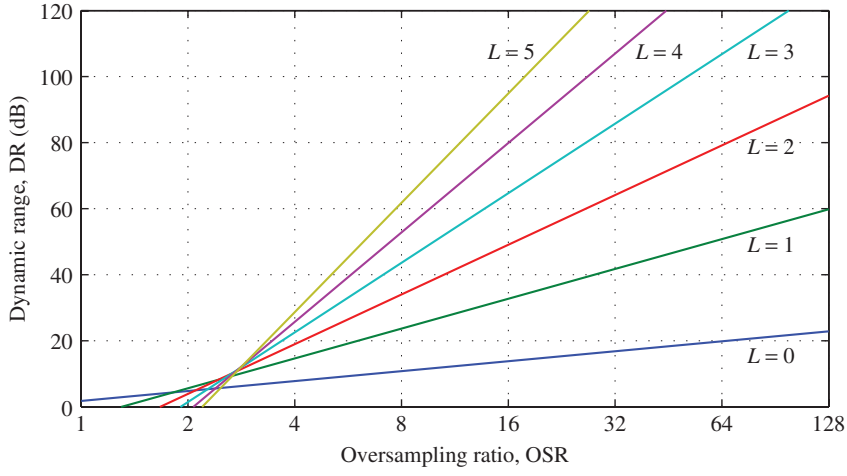


Figure 1.13 Ideal dynamic range of a ΣΔM as a function of the oversampling ratio for different noise-shaping orders (L). A single-bit internal quantizer ($B = 1$) is assumed.

1.3 Classification of ΣΔ Modulators

The strategies discussed in Section 1.2.4 for improving the DR of a ΣΔM may be combined in many different ways, giving rise to the huge number of ΣΔM topologies reported in literature, which can be grouped attending to different classification criteria [10]:

- *Single-Loop versus Cascade ΣΔMs (attending to the number of quantizers employed).* ΣΔMs employing only one quantizer are called *single-loop* topologies, whereas those employing several quantizers are often named *cascade* or MASH ΣΔMs. These topological alternatives will be discussed in Sections 1.4 and 1.5.
- *Single-Bit versus Multibit ΣΔMs (attending to the number of bits in the embedded quantizer).* Their pros and cons will be discussed in Section 1.6.
- *Low-Pass versus Band-Pass ΣΔMs (attending to the nature of the signals being converted).* The A/D conversion of LP signals has been assumed in previous sections, but BP ΣΔMs can also be built, as will be discussed in Section 1.7.
- *Discrete-Time versus Continuous-Time ΣΔMs (attending to the nature of loop filter dynamics).* The use of a DT loop filter in the ΣΔM has been assumed in previous sections. However, CT ΣΔMs can be also implemented in practice. According to this classification criteria, *hybrid* CT/DT ΣΔMs take advantage of the benefits of both DT and CT implementations, which will be discussed in Section 1.8.

Describing all possible ΣΔM architectures derived from previous classification criteria goes beyond the scope of this book. A detailed analysis of them can be found in the many papers and books devoted to the topic in the ΣΔ literature [4, 9, 11–23]. Instead, we will hereafter focus on the most representative families of ΣΔ modulators.

1.4 Single-Loop ΣΔ Modulators

ΣΔ modulators that make use of only one embedded quantizer are usually referred to as *single-loop topologies*. To get familiar with these architectures, their performance, their circuit-level implementation, as well as some other practical aspects are first addressed considering the case of second-order ΣΔMs. Afterward, the problematics of stability in higher-order ΣΔMs is presented, together with architectural alternatives to circumvent it.

1.4.1 Second-Order ΣΔM

Figure 1.14a shows a second-order ΣΔ modulator built up by cascading two DT integrators [24], with each integrator receiving a weighted feedback path from the DAC. Coefficients a_i are usually called integrator *scalings* or *weights*. Under linear analysis, the modulator output in the Z-domain yields

$$Y(z) = \frac{k_q a_1 a_2 \frac{z^{-2}}{(1-z^{-1})^2} X(z) + E(z)}{1 + k_q a_1 a_2 \frac{z^{-2}}{(1-z^{-1})^2} + k_q a_2 \frac{z^{-1}}{(1-z^{-1})}} \tag{1.24}$$

where k_q stands for the gain of the quantizer. For a pure second-order shaping, Equation 1.24 needs to be simplified to

$$Y(z) = z^{-2} X(z) + (1 - z^{-1})^2 E(z) \tag{1.25}$$

so that the following expressions for the integrator coefficients need to be fulfilled:

$$\begin{aligned} k_q a_1 a_2 &= 1 \\ k_q a_2 &= 2 \end{aligned} \tag{1.26}$$

Figure 1.14b shows an alternative representation of the second-order ΣΔM according to the coefficients notation in [12, 19], which allows to allocate different weights in the forward and feedback paths of each integrator, using coefficients g_i and g'_i, respectively. As exemplary illustrated in Figure 1.15, the notations in Figure 1.14a and b can be easily connected with the equalities:

$$\begin{aligned} a_1 &= \frac{g'_1 g_2}{g'_2} \\ a_2 &= g'_2 \end{aligned} \tag{1.27}$$

Both nomenclatures for the integrator scaling coefficients of ΣΔ modulators will be used throughout this book. The notation in [8, 9] is closer to the modulator architectural level, whereas the notation in [12, 19] is closer to the actual circuit-level implementation, in which integrators with more than one SC input branch are usually employed. The latter is thus useful to accurately account for some nonidealities of the practical ΣΔM implementation, which will be covered in Chapter 2.

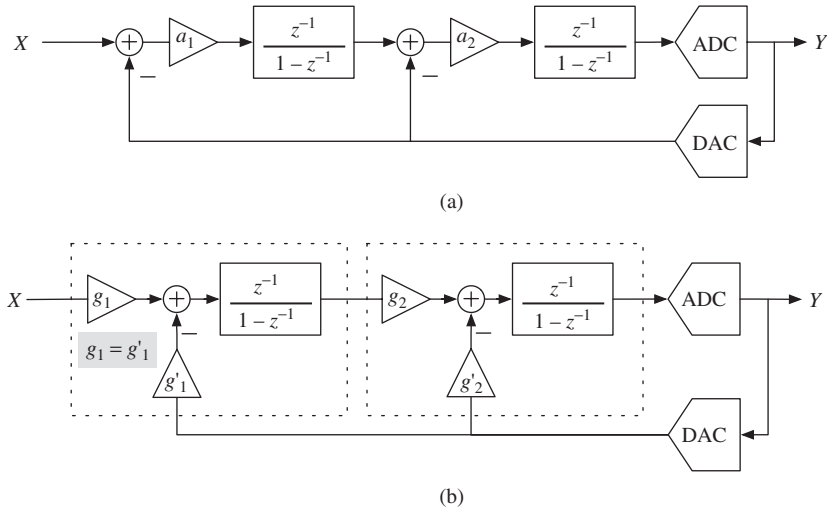


Figure 1.14 Block diagram of second-order ΣΔMs and different notations: (a) general representation of the DT-ΣΔM using the notation in [8, 9] and (b) alternative representation of the DT-ΣΔM using the notation in [12, 19].

For the sake of illustration, Figure 1.16 shows a possible implementation of the second-order ΣΔM in Figure 1.14 using fully-differential SC circuitry and assuming single-bit quantization. The modulator differential input signal is denoted by X and the modulator digital output Y, after the comparator, controls the feedback connection of reference voltages $V_{\text{ref+}}$ and $V_{\text{ref–}}$ to the integrators. The modulator full scale range Δ thus equals $2V_{\text{ref}}$, with $V_{\text{ref}} = V_{\text{ref+}} - V_{\text{ref–}}$. Note from the first SC integrator in Figure 1.16 that both the modulator input signal and the DAC feedback signal are processed through the sampling capacitor C_S. For the second integrator, the output of the first integrator is processed through both C_{S1} and C_{S2}, whereas the DAC feedback signal is processed only through C_{S2}. The modulator scaling coefficients are thus implemented as the following capacitor ratios (Figure 1.14b)

$$\begin{aligned} g_1 &= g'_1 = \frac{C_S}{C_{I1}} \\ g_2 &= \frac{C_{S1} + C_{S2}}{C_{I2}}, \quad g'_2 = \frac{C_{S2}}{C_{I2}} \end{aligned} \tag{1.28}$$

In practice, the value of the integrator weights are selected to fulfill the relations in Equation 1.26, also taking into account their implication in some aspects of the modulator performance, such as the following:

- Keeping the state variables (integrator outputs) bounded to ensure the modulator stability. The second-order ΣΔM is stable for inputs in the range $[-0.9\Delta/2, +0.9\Delta/2]$ if $g'_2 > 1.25 g'_1 g_2$, regardless the quantizer gains k_q [24]. This condition is already met if $g'_2 = 2g'_1 g_2$—see Equations 1.26 and 1.27.

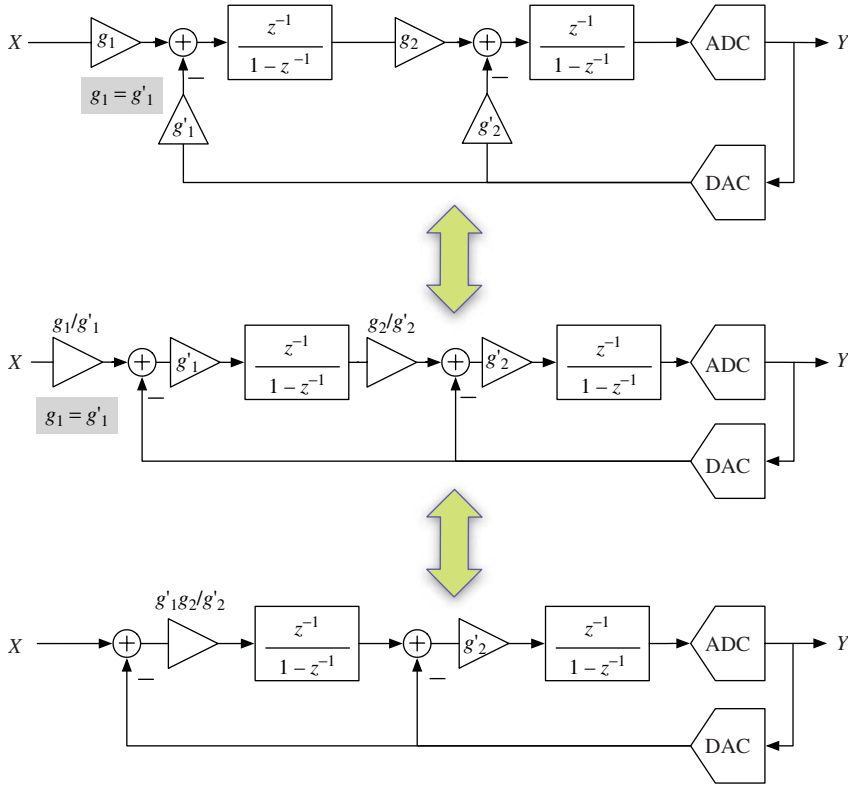


Figure 1.15 Illustration of the equivalence between the DT representations in Figure 1.14a and 1.14b.

- Keeping the modulator overload level as close as possible to the full scale to ensure a high-peak SNR—see Figure (1.11).
- Minimizing the required signal range at the integrators outputs; that is, the integrator output swing demands must be attainable with the intended voltage supply and as low as possible to reduce power consumption and to facilitate circuit design.
- Simplifying the practical implementation of the integrator weights as ratios of unit capacitors.

Generally speaking, the selection of the scaling coefficients of a ΣΔ modulator involves solving several trade-offs between architectural-, circuit-, and technological-level aspects of the practical implementation, so that the optimum selection for a given application may not apply in a different scenario. For the sake of illustration, Table 1.1 shows several sets of weights reported for the second-order, single-bit ΣΔM in Figure 1.14. All sets exhibit an overload level $X_{\mathrm{OL}} \approx -4$ dBFS (i.e., -4 dB below the full-scale amplitude $V_{\mathrm{ref}} = \Delta/2$). The required integrator output swing and the minimum number of unit capacitors are also included. Capacitor sharing between weights in the same integrator has been considered.

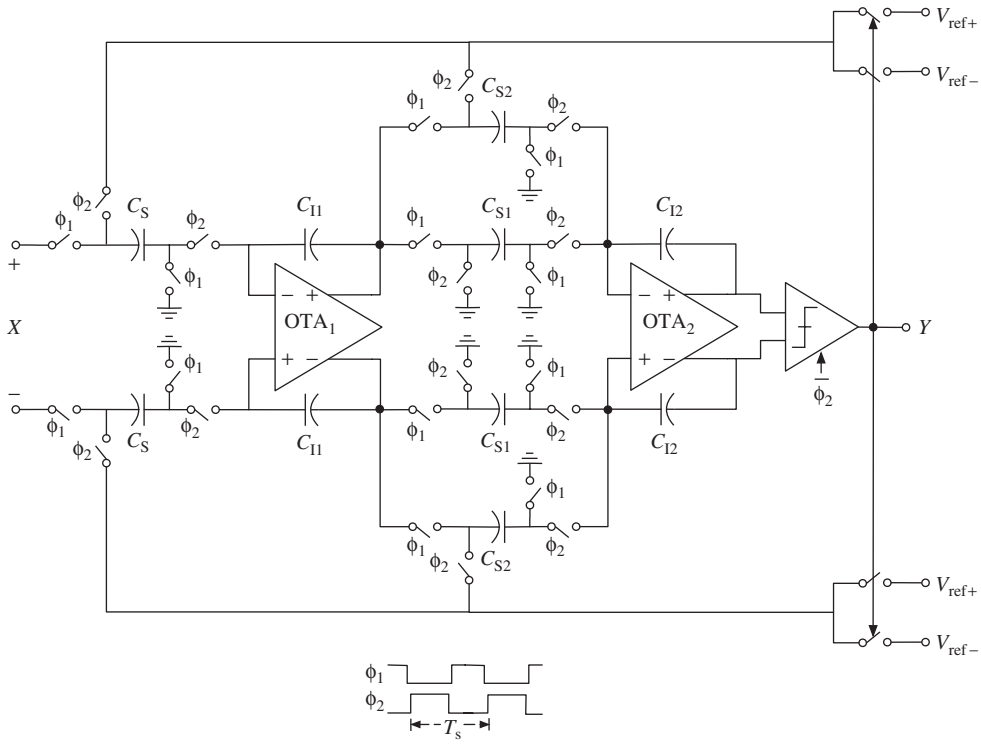


Figure 1.16 Fully-differential SC implementation of a second-order ΣΔ modulator.

Table 1.1 Comparison of some sets of coefficients reported for the second-order single-bit ΣΔM

	[7]	[25]	[26]	[27]
g_1, g_1'	1/2, 1/2	1/4, 1/4	1/2, 1/2	1/3, 1/3
g_2, g_2'	1/2, 1/2	1/2, 1/4	1, 1/2	3/5, 2/5
a_1, a_2	0.5, 0.5	0.5, 0.25	0.5, 0.5	0.5, 0.4
Overload level	−4 dBFS	−4 dBFS	−4 dBFS	−4 dBFS
Integrator output swing	$\pm 1.5V_{ref}$	$\pm 0.75V_{ref}$	$\pm 1.25V_{ref}$	$\pm 1.0V_{ref}$
Unit capacitors (2× in fully diff)	$6(=3+3)$	$11(=5+6)$	$9(=5+4)$	$12(=4+8)$

Second-Order ΣΔM with Unity STF

Figure 1.17 shows an alternative second-order ΣΔM topology that makes use of feed-forward paths to implement a so-called *unity* STF [28, 29]. Under linear analysis, the modulator output in the Z-domain yields

$$Y(z) = X(z) + (1 - z^{-1})^2 E(z) \tag{1.29}$$

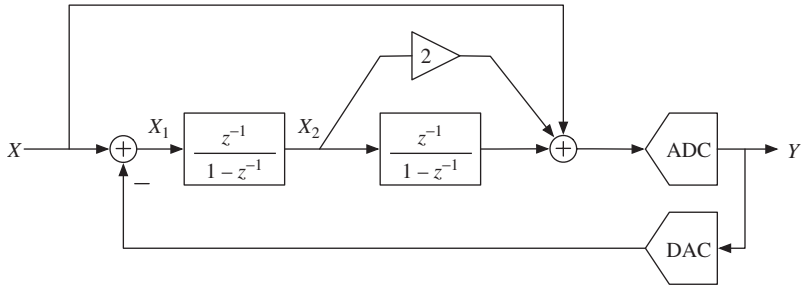


Figure 1.17 Second-order ΣΔ modulator with unity STF.

so that $\mathrm{STF}(z) = 1$—that is, the signal transfer function equals 1 at all frequencies—whereas NTF is unaffected.

One of the most appealing features of the unity-STF ΣΔM in Figure 1.17 is that ideally there is no input signal trace processed by the integrators. Indeed, the integrator inputs in Z-domain can be obtained as

$$\begin{aligned} X_1(z) &= -(1 - z^{-1})^2 E(z) \\ X_2(z) &= z^{-1}(1 - z^{-1})^2 E(z) \end{aligned} \tag{1.30}$$

showing that they depend only on the quantization error. In practice, there will be some residual component of the modulator input signal at the integrator inputs, but it is normally negligible. This means that, if nonlinearities of the circuit implementation are accounted for, the generated distortion will be considerably lower for the unity-STF ΣΔM in Figure 1.17 compared to the one traditional ΣΔM in Figure 1.14. Moreover, the technique is effective for any OSR, which makes unity-STF ΣΔMs especially suited for lowering the sensitivity to circuit imperfections in wideband applications, in which low OSR values are required.

The described concept of unity STF can be extended to a noise shaping of any order. The only requirement is to ideally make $\mathrm{STF}(z) = 1$, without changing the modulator NTF. In recent years, it has been often applied in ΣΔMs for wideband and multimode applications [30–34].

1.4.2 High-Order ΣΔMs

The simplest way to extend a ΣΔM to an arbitrary Lth-order shaping consists of including L integrators before the quantizer. Extending the second-order ΣΔM in Figure 1.14a, the topology in Figure 1.18 is obtained, which is known as an Lth-order single-loop ΣΔM with *distributed feedback* [35].[3] Ideally, its NTF can be derived from linear analysis

[3] The discrete-time filters $H(z)$ in Figure 1.18 are assumed to be integrators with the transfer function in Equation 1.16.

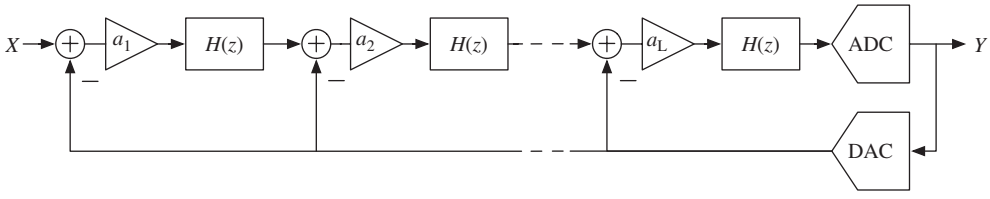


Figure 1.18 High-order single-loop ΣΔM with distributed feedback.

and equated to Equation 1.10 to derive a set of relations between the integrator scaling coefficients to be fulfilled for obtaining a pure-differentiator noise shaping—similarly as done in Equation 1.26 for the second-order ΣΔM. The in-band quantization noise and the modulator DR would thus be ideally given by Equations 1.12 and 1.13, respectively.

However, this modulator performance cannot be met in practice because ΣΔMs with pure-differentiator FIR NTFs are prone to instability if $L > 2$, exhibiting unbounded states and poor SNR compared to that predicted by linear analysis. In general, instability appears at the modulator output as a large-amplitude, low-frequency oscillation, leading to long bitstreams of alternating +1s and −1s. This tendency to instability can be explained as follows [36]. For a ΣΔM to be stable, the quantizer input must not be allowed to become too large. As the quantizer input is obtained as (Figure 1.18)

$$Q(z) = \mathrm{STF}(z)X(z) + [\mathrm{NTF}(z) - 1]E(z) \tag{1.31}$$

the gain of $\mathrm{NTF}(z) - 1$, or simply $\mathrm{NTF}(z)$, must not be too large. However, as clearly visible from Figure 1.12, the out-of-band gain of FIR NTFs of the form $(1 - z^{-1})^L$ rapidly increases for $L > 2$, yielding $\max[|\mathrm{NTF}(f)|^2] = 2^{2L}$ at $f = f_s/2$. Consequently, it starts to overload the quantizer, which yields a significant decrease of the modulator SNR.

This problem can be circumvented by resorting to single-loop ΣΔMs with IIR NTFs of the form $\mathrm{NTF}(z) = (z - 1)^L/D(z)$, with $D(z)$ being a polynomial determined by the modulator scaling coefficients that helps to limit the out-of-band gain of NTF.[4] However, unlike second-order ΣΔMs, for which a stability condition has been extracted [24], determining exact conditions that guarantee the stability of higher-order, single-loop ΣΔMs is still an open question. In [8, 37], it is shown, using behavioral simulations, that high-order ΣΔMs are *conditionally* stable; that is, with proper selection of the scaling coefficients, a stable operation can be obtained for inputs restricted within a certain range and for certain initial conditions of the state variables. However, despite the absence of general stability conditions, high-order ΣΔMs have been successfully designed since the late 1980s [38]. Indeed, the topology in Figure 1.18 has been widely used and optimal coefficients for it have been presented in [8].

[4] Note that the scaling coefficients can be thus designed to build a high-pass Butterworth or Chebyshev filter for NTF.

Its STF and NTF can be calculated under linear analysis as[5]

$$\mathrm{STF}(z) = \frac{k_\mathrm{q} \prod_{i=1}^{L} a_\mathrm{i} H^L(z)}{1 + k_\mathrm{q} \sum_{i=1}^{L} \left[\left(\prod_{j=i}^{L} a_\mathrm{j} \right) H^{L-i+1}(z) \right]} \tag{1.33}$$

$$\mathrm{NTF}(z) = \frac{1}{1 + k_\mathrm{q} \sum_{i=1}^{L} \left[\left(\prod_{j=i}^{L} a_\mathrm{j} \right) H^{L-i+1}(z) \right]} \tag{1.34}$$

If integrators described by Equation 1.16 are used as filters—$H(z) = \mathrm{ITF}(z)$—the NTF can be approximated for low frequencies to [8]

$$|\mathrm{NTF}| \approx \frac{|1 - z^{-1}|^L}{k_\mathrm{q} \prod_{i=1}^{L} a_\mathrm{i}} \tag{1.35}$$

in which the complete scaling of the outermost feedback branch of the ΣΔM dominates the noise-shaping behavior. Similarly as done in Equations 1.11 and 1.12, the IBN of an Lth-order, single-loop ΣΔM with distributed feedback thus yields

$$\mathrm{IBN_L} \approx \frac{\Delta^2}{12} \frac{1}{(k_\mathrm{q} \prod_{i=1}^{L} a_\mathrm{i})^2} \frac{\pi^{2L}}{(2L+1)\mathrm{OSR}^{2L+1}} \tag{1.36}$$

so that the IBN increases by a factor $1/(k_\mathrm{q} \prod_{i=1}^{L} a_\mathrm{i})^2$ compared to an ideal ΣΔM.

A major drawback of this loop filter implementation in Figure 1.18 is that the integrator outputs contain a significant amount of the input signal [41], so that the integrators require significant swing capabilities and/or the scaling coefficients need to be low. This can be circumvented using the loop filter topology in Figure 1.19, which is a chain of integrators with *feed-forward summation* [42]. Under linear analysis, the corresponding STF and NTF can be calculated as

$$\mathrm{STF}(z) = \frac{k_\mathrm{q} \sum_{i=1}^{L} a_{\mathrm{L-i+1}} H^{L-i+1}(z)}{1 + k_\mathrm{q} \sum_{i=1}^{L} a_{\mathrm{L-i+1}} H^{L-i+1}(z)} \tag{1.37}$$

$$\mathrm{NTF}(z) = \frac{1}{1 + k_\mathrm{q} \sum_{i=1}^{L} a_{\mathrm{L-i+1}} H^{L-i+1}(z)} \tag{1.38}$$

[5] The quantizer gain k_q of the linear quantizer model is explicitly considered hereinafter. As discussed in Section 1.1.2, the gain of a multibit quantizer is clearly defined—for example, $k_\mathrm{q} = 1$ if its input and output full-scale ranges are the same (Figure 1.4)—but that of a single-bit quantizer can be arbitrarily chosen. Nevertheless, the effective value of k_q needs to be estimated to quantitatively analyze the performance of ΣΔMs. Many different approaches exist to find a good approximation [8, 39, 40]. Among them, the one adopted here for the case of ΣΔMs with distributed feedback corresponds to that in [20]

$$k_\mathrm{q} = \begin{cases} 1/a_1, & \text{single-bit first-order } \Sigma\Delta\mathrm{M} \\ 2/a_\mathrm{L}, & \text{single-bit } L\text{th-order } \Sigma\Delta\mathrm{M} \\ 1, & \text{multibit } \Sigma\Delta\mathrm{M} \end{cases} \tag{1.32}$$

because of its simplicity and its good agreement with simulation results.

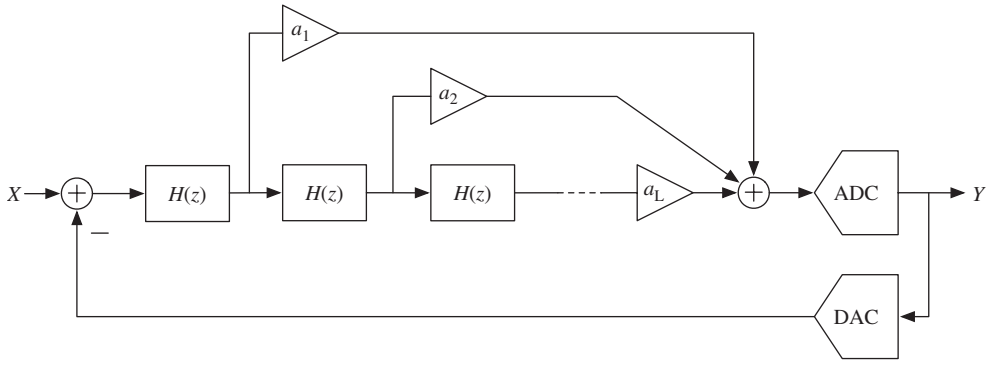


Figure 1.19 High-order single-loop ΣΔM with feed-forward summation.

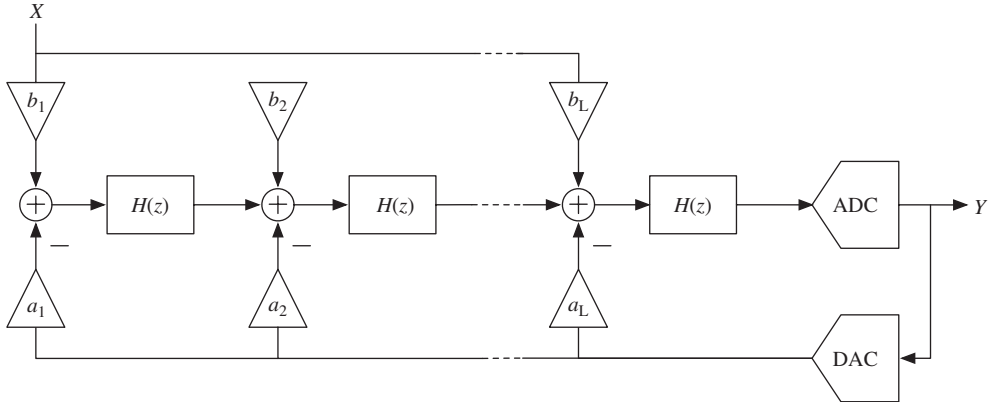


Figure 1.20 High-order single-loop ΣΔM with distributed feedback and distributed feed-forward input paths.

Note that NTF structure is the same as in Equation 1.34; therefore, the IBN of this ΣΔM topology yields an expression similar to that in Equation 1.36.

However, for the case of the distributed feedback topology in Figure 1.18 and the feed-forward summation topology in Figure 1.19, the loop filter for the STF and the NTF are essentially identical—compare Equation 1.33 with 1.34 and Equation 1.37 with 1.38. This means that, if the NTF is designed for the desired noise-shaping behavior, both topologies also fix the STF function, as $\mathrm{STF}(z) = 1 - \mathrm{NTF}(z)$ [41].[6]

If a certain degree of freedom is desired in designing both the modulator NTF and STF, the topology in Figure 1.20 can be used, which is a chain of integrators with *distributed feedback and distributed feed-forward input paths* [43]. In this architecture, the zeros of STF can be fixed with coefficients b_i without affecting

[6] In addition, the STF of the feed-forward summation architecture contains some peaking at high frequencies, what may jeopardize the modulator stability if precautions are not taken [41].

the pole placement, so both STF and NTF can be separately optimized to some extent [41].[7]

1.5 Cascade ΣΔ Modulators

As discussed in Section 1.4.2, stability problems arising from implementing a high-order NTF with a single-loop ΣΔM can be circumvented with adequate scaling coefficients, but they result in a significant decrease of the DR compared with an ideal ΣΔM. An alternative approach to obtain a high-order noise shaping while avoiding instabilities is found in the so-called *cascade* ΣΔMs, also known as *multiloop* ΣΔMs or *multistage noise shaping* (MASH) ΣΔMs [45–48]. Their architecture is illustrated in Figure 1.21 and consists of N stages of ΣΔ modulators, in which each stage remodulates a scaled version

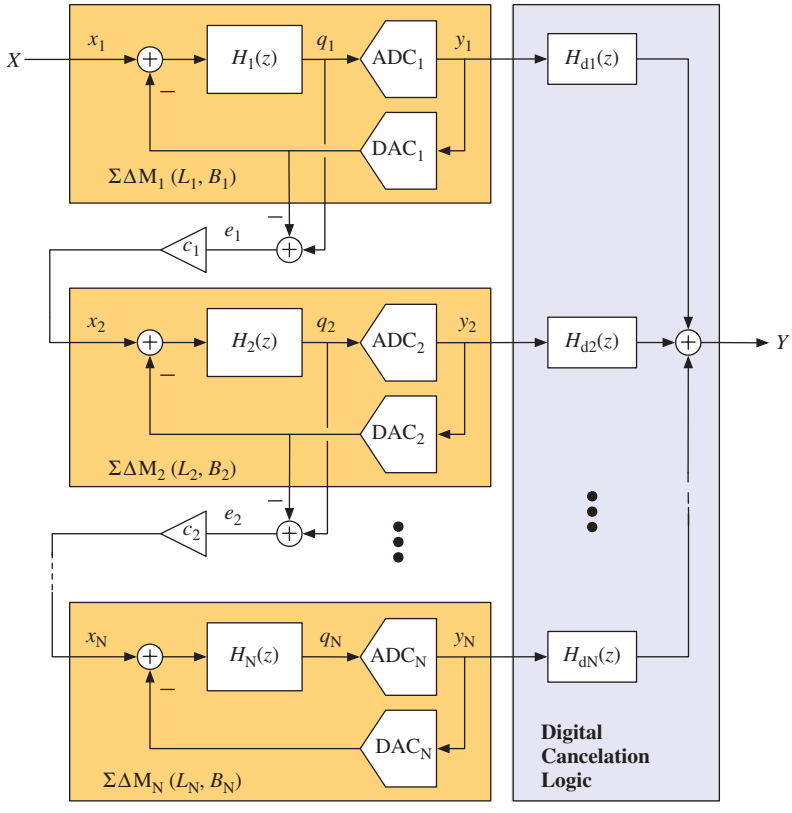


Figure 1.21 General topology of an N-stage cascade ΣΔ modulator.

[7] Local feedback loops—similar to those described in Section 1.7.2 for creating the notches in the NTF of BP ΣΔMs—can also be included in LP ΣΔM topologies above to move the NTF zeros away from DC and optimally spread them over the signal band to minimize the IBN [44]. NTFs with inverse Chebyshev filtering characteristics can also be designed [41].

of the quantization error generated in the preceding one. The outputs of the cascaded stages y_i are conveniently processed in digital domain to cancel out at the overall ΣΔM output y all the quantization errors, but that of the back-end stage. In addition, the latter quantization error appears at the cascade output shaped with an order L equal to the summation of those of the cascaded stages ($L = L_1 + L_2 + \cdots + L_N$). Unconditionally, stable high-order shaping can thus be obtained if only first- and second-order ΣΔMs are cascaded ($L_i \leq 2$), because all feedback loops are local to the low-order ΣΔ stages and there is no interstage feedback. Therefore, the performance of a multistage ΣΔM is similar to that of an ideal high-order, single-loop with no stability issues.[8]

The operation principle of cascade ΣΔMs can be easily understood considering an exemplary two-stage cascade. Using linear analysis, the stages output can be expressed in the Z-domain as

$$\begin{aligned} Y_1(z) &= \mathrm{STF}_1(z)X_1(z) + \mathrm{NTF}_1(z)E_1(z) \\ Y_2(z) &= \mathrm{STF}_2(z)X_2(z) + \mathrm{NTF}_2(z)E_2(z) \end{aligned} \tag{1.39}$$

where the input of the ΣΔ stages are given by $X_1(z) = X(z)$ and $X_2(z) = -c_1E_1(z)$, and $E_1(z)$ and $E_2(z)$ stand for the quantization error of the respective stage. The overall modulator output after the digital cancelation logic (DCL) can thus be obtained from the expressions above as

$$\begin{aligned} Y(z) &= H_{d1}(z)Y_1(z) + H_{d2}(z)Y_2(z) \\ &= \mathrm{STF}_{casc}(z)X(z) + \mathrm{NTF}_{1,casc}(z)E_1(z) + \mathrm{NTF}_{2,casc}(z)E_2(z) \end{aligned} \tag{1.40}$$

where $\mathrm{STF}_{casc}(z)$, $\mathrm{NTF}_{1,casc}(z)$, and $\mathrm{NTF}_{2,casc}(z)$ stand for the overall cascade transfer functions for the input signal and the quantization errors, respectively

$$\begin{aligned} \mathrm{STF}_{casc}(z) &= H_{d1}(z)\mathrm{STF}_1(z) \\ \mathrm{NTF}_{1,casc}(z) &= H_{d1}(z)\mathrm{NTF}_1(z) - c_1H_{d2}(z)\mathrm{STF}_2(z) \\ \mathrm{NTF}_{2,casc}(z) &= H_{d2}(z)\mathrm{NTF}_2(z) \end{aligned} \tag{1.41}$$

Note that, if the signal processing in the DCL matches part of the signal processing in the analog side, Equation 1.41 yields

$$\left.\begin{aligned} H_{d1}(z) &= \mathrm{STF}_2(z) \\ H_{d2}(z) &= \tfrac{1}{c_1}\mathrm{NTF}_1(z) \end{aligned}\right\} \Rightarrow \left\{\begin{aligned} \mathrm{STF}_{casc}(z) &= \mathrm{STF}_1(z)\mathrm{STF}_2(z) \\ \mathrm{NTF}_{1,casc}(z) &= 0 \\ \mathrm{NTF}_{2,casc}(z) &= \tfrac{1}{c_1}\mathrm{NTF}_1(z)\mathrm{NTF}_2(z) \end{aligned}\right. \tag{1.42}$$

so that the first-stage quantization error is canceled at the overall output and that of the second stage is attenuated with an order equal to the summation of both stages' orders ($L = L_1 + L_2$).

[8] Cascade ΣΔMs can be *ideally* extended to an arbitrary number of stages. However, as will be shown in detail in Chapter 2, the effectiveness of cascading a large number of ΣΔ stages to achieve an arbitrary high-order noise shaping is limited in practice by circuit nonidealities, which preclude the complete cancelation of low-order-shaped quantization errors of the front-end ΣΔ stages at the modulator output. This effect is known as *noise leakage*.

For the generic N-stage cascade ΣΔM in Figure 1.21, assuming $\mathrm{STF_i}(z) = z^{-L_\mathrm{i}}$ and $\mathrm{NTF_i}(z) = (1 - z^{-1})^{L_\mathrm{i}}$ in the individual stages, the overall modulator output thus yields

$$Y(z) = z^{-L}X(z) + \frac{1}{\prod_{i=1}^{N-1} c_\mathrm{i}}(1 - z^{-1})^{L}E_\mathrm{N}(z), \text{ with } L = \sum_{i=1}^{N} L_\mathrm{i} \tag{1.43}$$

under the required matching between the analog processing in the ΣΔ stages and the DCL. The in-band quantization noise of the cascade is then given by

$$\mathrm{IBN_{casc}} = \frac{\Delta_\mathrm{N}^2}{12}\frac{1}{\prod_{i=1}^{N-1} {c_\mathrm{i}}^2}\frac{\pi^{2L}}{(2L+1)\mathrm{OSR}^{2L+1}} \tag{1.44}$$

where Δ_N stands for the quantization step employed in the last ΣΔ stage. Note from Equation 1.44 that only the interstage scaling coefficients c_i, which prevent a premature overload of the subsequent stages, decrease the performance below that of an ideal Lth order ΣΔM. Typical values of $1/\prod_{i=1}^{N-1} c_\mathrm{i}$ are between 2 and 4 if single-bit quantization is employed, which lead to a reduction in the ideally attainable DR of only 6–12 dB (1–2 bit). These performance losses are inherent to cascade ΣΔMs, but they are significantly smaller than those resulting from optimized high-order single loops. In addition, they are independent of OSR.

The aforementioned benefits of cascade ΣΔMs have favored the development of a large number of different topologies:

- 2-1 ΣΔM, which stands for a third-order two-stage ΣΔM built up with a second-order stage followed by a first-order one [46]. It is also known as a *SOFO cascade*—for second-order first-order.
- 2-2 ΣΔM, which represents a fourth-order cascade built up with two second-order stages [49, 50].
- 2-1-1 ΣΔM, which stands for a fourth-order, third-stage cascade [25].
- 2-2-1 ΣΔM [51].
- 2-1-1-1 ΣΔM [52].
- 2-2-2 ΣΔM [53].
- etc.

Note from the cascade topologies above that the first stage is usually a second-order ΣΔM and that first-order ΣΔMs are avoided at the front end [48]. The reason for that is twofold. First, the quantization error from the first stage would be only first-order shaped and noise leakages would be larger. Second, the tonal behavior of a first-order, first stage would menace the performance of the cascade. In addition, although low-order ΣΔ stages are most frequently cascaded, 3-1 and 3-2 ΣΔMs have also been reported [54, 55].

Figure 1.22 illustrates the block diagram of a 2-1-1 cascade, in which coefficients a_i represent the in-loop integrator scaling factors, whereas coefficients b_i and c_i determine the interstage scaling factors. The first stage performs as an ideal second-order ΣΔM and

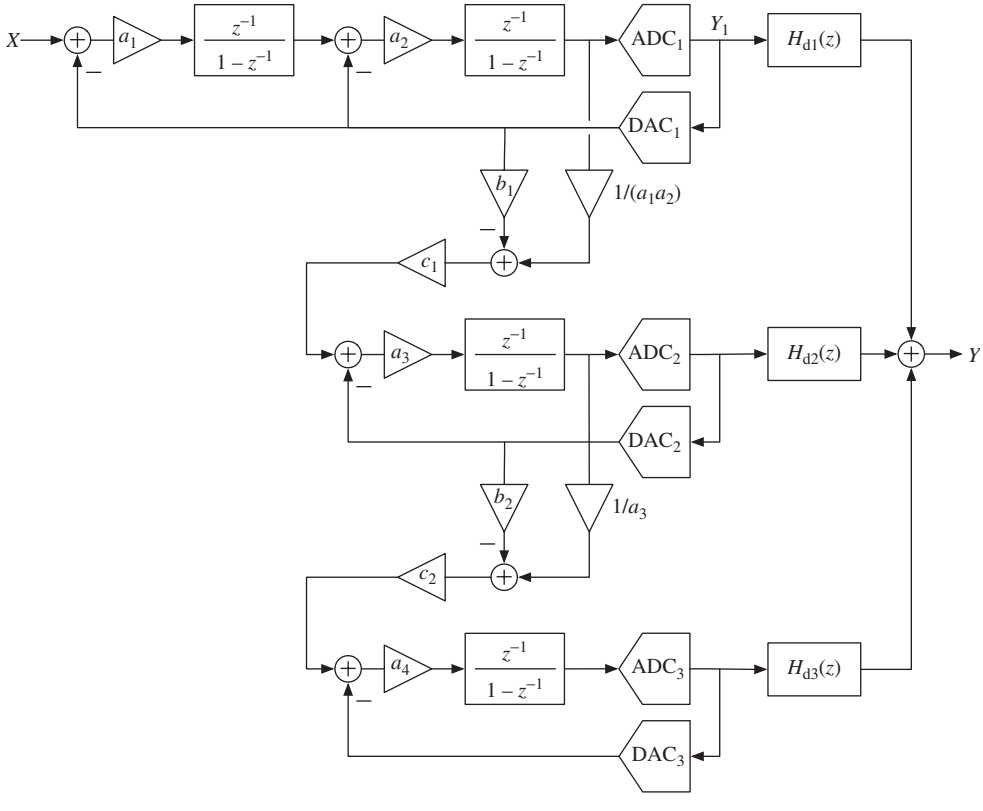


Figure 1.22 Block diagram of a 2-1-1 cascade ΣΔ modulator using the notation in [8, 9]. The relations in Equations 1.45 and 1.46 must be fulfilled for the correct operation of the cascade.

the second and third stages as an ideal first-order ΣΔM under the assumptions

$$\begin{aligned} &k_{q1}a_1a_2 = 1,\ k_{q1}a_2 = 2 \\ &k_{q2}a_3 = 1 \\ &k_{q3}a_4 = 1 \end{aligned} \tag{1.45}$$

where k_{qi} stands for the gain of the quantizer in the ith ΣΔ stage. The matching required between the signal processing in the ΣΔ stages and the digital processing in the DCL leads to

$$\begin{aligned} H_{d1}(z) &= z^{-2}[1 + (b_1 - 1)(1 - z^{-1})^2][1 + (b_2 - 1)(1 - z^{-1})^3] \\ H_{d2}(z) &= \frac{1}{c_1}z^{-1}(1 - z^{-1})^2[1 + (b_2 - 1)(1 - z^{-1})^3] \\ H_{d3}(z) &= \frac{1}{c_1c_2}(1 - z^{-1})^3 \end{aligned} \tag{1.46}$$

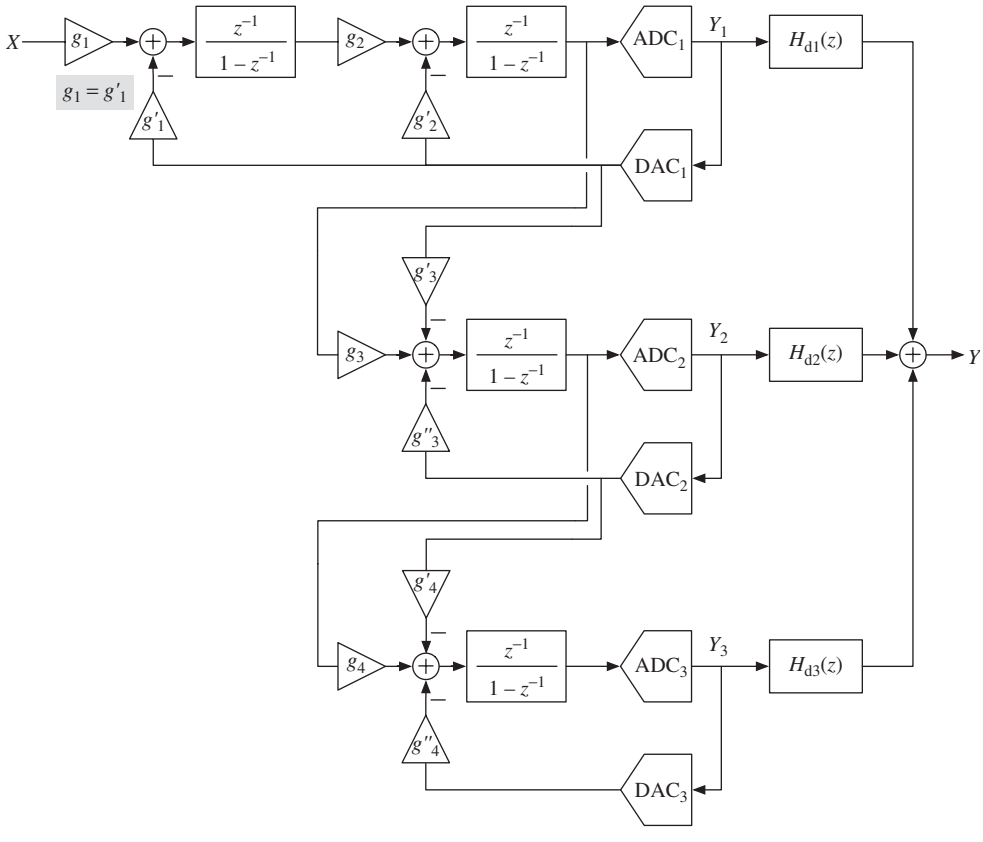


Figure 1.23 Alternative representation of the 2-1-1 ΣΔM in Figure 1.22 using the notation in [12, 19]. Modulator coefficients (a_i, b_i, c_i) are mapped onto integrator input coefficients (g_i, $g_i^{'}$, $g_i^{''}$), which are closer to the circuit-level implementation.

for a complete cancelation of the first- and second-stage quantization errors at the cascade output. For the sake of completeness, Figure 1.23 shows an alternative representation of the 2-1-1 ΣΔM according to the notation in [12, 19]. Proceeding in a similar way as done in Figure 1.15 for the second-order ΣΔM, both notations can be easily connected with the equalities:

$$
\begin{aligned}
a_1 &= \frac{g_1' g_2}{g_2'},\ a_2 = g_2',\ a_3 = g_3'',\ a_4 = g_4'' \\
b_1 &= \frac{g_3'}{g_1' g_2 g_3},\ b_2 = \frac{g_4'}{g_3'' g_4} \\
c_1 &= \frac{g_1' g_2 g_3}{g_3''},\ c_2 = \frac{g_3'' g_4}{g_4''}
\end{aligned}
\tag{1.47}
$$

Table 1.2 Comparison of some sets of coefficients reported for the 2-1-1 single-bit ΣΔM

	[56]	[25]	[27]
g_1, g_1'	1/4, 1/4	1/4, 1/4	1/3, 1/3
g_2, g_2'	1, 1/2	1/2, 1/4	3/5, 2/5
g_3, g_3', g_3''	1, 1/2, 1/2	1, 3/8, 2/8	5/6, 3/6, 2/6
g_4, g_4', g_4''	1, 1/2, 1/2	1, 1/4, 1/4	1, 1/3, 1/3
a_1, a_2, a_3, a_4	0.5, 0.5, 0.5, 0.5	0.5, 0.25, 0.25, 0.25	0.5, 0.4, 0.33, 0.33
b_1, c_1, b_2, c_2	2, 0.5, 1, 1	3, 0.5, 1, 1	3, 0.5, 1, 1
$1/(c_1 c_2)$	2	2	2
Loss of DR	−6 dB	−6 dB	−6 dB
Overload level	−3 dBFS	−2.5 dBFS	−2 dBFS
Integrator output swing	$\pm(0.75, 1, 1, 1)V_{\text{ref}}$	$\pm(0.75, 0.7, 0.6, 0.6)V_{\text{ref}}$	$\pm(1, 1, 0.9, 0.8)V_{\text{ref}}$
Unit capacitors (2×)	$17(= 5 + 4 + 4 + 4)$	$35(= 5 + 6 + 16 + 8)$	$29(= 4 + 8 + 11 + 6)$

The value of the analog coefficients in the 2-1-1 ΣΔM need to be selected to fulfill the relations in Equation 1.45, but many different sets of values can do the work. On top of that, extra considerations are in practice taken into account, such as the following:

- Minimizing the resulting loss of performance in comparison to an ideal ΣΔM.
- Maximizing the modulator overload level to achieve a high-peak SNR.
- Simplifying the circuit-level implementation of the set of analog coefficients. For SC-ΣΔMs, this means taking into account practical capacitor ratios using unit elements, enabling capacitor sharing within the integrators, reducing the total number of unit capacitors for area saving, etc.
- Minimizing the integrator output swing requirements, especially in case of a low-voltage supply.
- Simplifying the implementation of the DCL. To that purpose, power of two coefficients are often preferred for b_1, b_2, $1/c_1$, and $1/c_2$—see Equation 1.46 —in order to use shift registers only.

Table 1.2 illustrates some sets of analog coefficients reported for the 2-1-1 single-bit ΣΔM, together with their main resulting features. Optimized coefficients for some other cascade topologies can be found in [8].

1.6 Multibit ΣΔ Modulators

As discussed in Section 1.2.4, the DR of a ΣΔM can be enhanced if the resolution of the embedded quantizer is increased. The main advantages of resorting to multibit ΣΔ modulators are as follows:

- The in-band quantization noise power is roughly reduced 6 dB per additional bit in the embedded quantizer, thanks to the smaller quantization step Δ.

- Internal nonlinearities are weaker in multibit ΣΔMs than in their single-bit counterparts. The quantizer operation better fits the additive white noise model in Section 1.1.3 and the phenomena caused by nonlinear dynamics are less evident.
- For a given order in the loop filter, the stability properties of multibit ΣΔMs are better than for single-bit ΣΔMs [57].

These benefits suggest that, for a targeted performance of the ΣΔM, multibit quantization can be traded for noise shaping and/or oversampling. Indeed, multibit ΣΔMs are often employed in broadband applications to compensate for the limited OSR. Nevertheless, multibit quantizers also have important drawbacks that may counter the former advantages:

- They require more analog circuitry and are more difficult to design than single-bit ones.
- Contrary to 1-bit quantizers, which are *intrinsically* linear because only two levels are used for quantization, multibit quantizers exhibit in practice some nonlinearities in their transfer characteristic, mostly due to device *mismatching*, which significantly influence the ΣΔM performance.

1.6.1 Influence of Multibit DAC Errors

Figure 1.24 illustrates an enhanced version of the linear model of a multibit ΣΔM in Figure 1.8b. Errors related to the multibit conversion are added to the quantization error e that has been considered so far; namely, an error e_{ADC} associated to the A/D conversion and an error e_{DAC} in the subsequent D/A conversion required to reconstruct the analog feedback signal. Note that e_{ADC} is injected in the same path as the quantization error e and, therefore, it is also attenuated within the signal band by noise shaping. However, DAC errors are injected in the feedback path and, therefore, they directly add to the ΣΔM input signal and pass to the ΣΔM output as part of the input signal itself. Consequently, the linearity of a multibit ΣΔM will be *no better* than that of the multibit embedded DAC and the latter must be designed to achieve the linearity targeted for the whole ΣΔ ADC, what may be challenging under the influence of component mismatching.

Figure 1.25 conceptually illustrates the parallel architecture that is typically used for the multibit quantizer in ΣΔMs, in which the resolution is usually low ($B \leq 5$). The B-bit ADC consists of a bank of $2^B - 1$ comparators that digitizes the loop filter output into thermometer code, which will be subsequently coded into binary. The DAC employs $2^B - 1$ unit elements (capacitors, resistors, current sources, etc., depending on the circuit

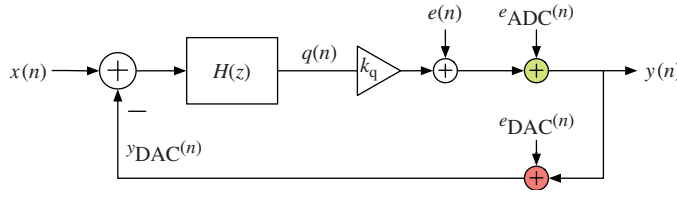


Figure 1.24 Linear model of a multibit ΣΔM including errors in the embedded ADC and DAC.

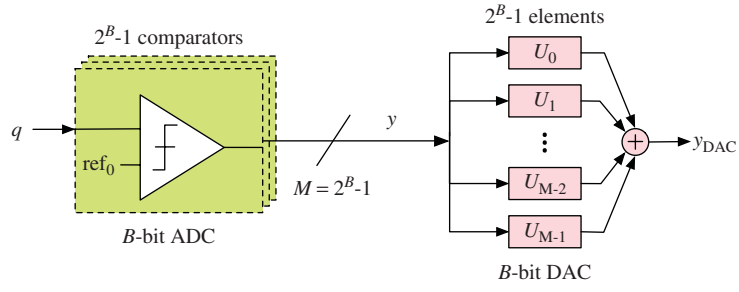


Figure 1.25 Parallel topology of a typical multibit quantizer embedded in a ΣΔM.

implementation) to reconstruct the analog feedback signal using 2^B levels (numbered from 0 to $M = 2^B - 1$). The ith analog output level is generated by activating i unit elements and adding their outputs (charges or currents). DAC errors are caused by the mismatching between its unit elements, which makes the DAC output levels deviate from their nominal values. Assuming that the actual value of each unit element follows a Gaussian distribution, the worst-case relative error in the DAC output y_{DAC} can be estimated as

$$\sigma\left(\frac{\Delta y_{DAC}}{y_{DAC}}\right) \approx \frac{1}{2\sqrt{2^B}}\,\sigma\left(\frac{\Delta U}{U}\right) \tag{1.48}$$

where $\sigma(\Delta U/U)$ stands for the relative error in the value of the unit element. Obviously, the DAC accuracy increases with the number of unit elements, thanks to the parallel topology. However, for a ΣΔM with 4-bit embedded quantization to achieve 16-bit linearity, the required matching for the DAC unit elements should be better than 0.01% (13 bits). Device matching achieved in present-day CMOS processes is nevertheless in the range of 0.1% (10 bits) and the required accuracy in the elements can only be obtained through the parallel connection of many more of them (~ 64). This means that achieving linearities better than 12 or 13 bits in multibit ΣΔMs by means of relying only on *standard* device matching usually leads to prohibitive area occupation.

A direct method to improve the standard device matching is laser trimming, what can sometimes be done at the foundry, but at the expense of additional fabrication and/or measurement steps and increased cost. Calibration and correction schemes have also been proposed, either in analog or digital domain [58], but they are often expensive to implement in terms of system design complexity, hardware requirements, and power consumption.

Among the different alternatives that have been developed through the years for achieving high-linear multibit ΣΔMs, two of them clearly prevail because of the modest component matching required and the reduced circuit complexity involved. These approaches are discussed in the following text.

1.6.2 DEM Techniques

As previously discussed, mismatches among the unit elements cause DAC nonlinearities that generate harmonic distortion in the ΣΔM. For a multibit DAC with the topology in

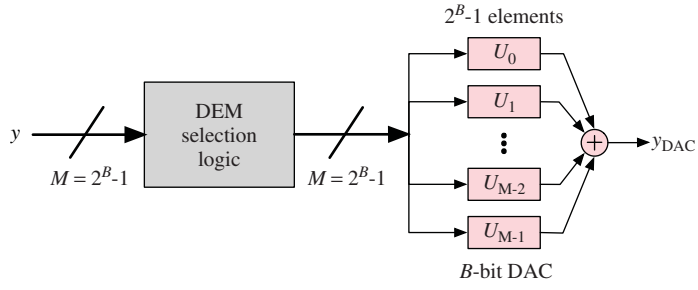


Figure 1.26 Incorporating an element selection logic for applying DEM to a multibit DAC.

Figure 1.25, there is a univocal correspondence between the thermometric input code (y) and the respective error of the DAC output (y_{DAC}), because the *same* unit elements are always used for generating a given DAC output level. The operation principle of DEM consists in breaking this direct correspondence by varying over time the set of elements that are employed for generating a given DAC level, thus transforming its fixed error into a *time-varying* one. To that purpose, as conceptually illustrated in Figure 1.26, a digital block is incorporated, which controls the selection of unit elements at each clock cycle according to an algorithm that tries to null the average error in each DAC level over time. This way, part of the DAC error power that lies in the low-frequency range will thus be pushed to higher frequencies and removed by the decimator.

The incorporation of DEM techniques to multibit ΣΔMs is facilitated by digitally oriented CAD tools and can represent a little area overhead in the current CMOS process, what explains the large number of algorithms that have been developed though the years. A detailed overview of many of them can be found in [9, 58, 59]. They can be categorized as follows:

- *Randomization Algorithms*, in which the DAC unit elements are selected according to pseudo-randomly configured networks (e.g., butterfly structures) [60]. Harmonic distortion induced by the DAC is transformed into white noise, whose out-of-band power will be removed by the decimation filter. The DAC error power lying within the signal band will nevertheless increase the noise floor of the ΣΔM.
- *Rotation Algorithms*, in which the DAC unit elements are selected in a periodic manner for shifting harmonic distortion out of the signal band. The ΣΔM noise floor is not increased, but the signal processing can originate mixed frequency components that fold over the modulator passband. *Clocked averaging* (CLA) [61] is an example of this kind of DEM techniques.
- *Mismatch-Shaping Algorithms*, in which the DAC unit elements are selected according to algorithms that conform the mismatching error to push most of its power to higher frequencies. The order of the mismatch shaping is normally limited to one or two. *Individual level averaging* (ILA) [62] and *data weighted averaging* (DWA) [63]—and its many modifications—pertain to this kind of algorithms.
- *Vector-Quantizer Structures*, in which a digital ΣΔ converter in error-feedback configuration is incorporated to achieve high-order shaping [64].

DWA and Pseudo-DWA Algorithms

Among the many different DEM alternatives, DWA [63]—or a modified version of DWA—dominates in high-speed, high-resolution ΣΔ modulators with multibit quantization. In such high-speed ΣΔMs, the complexity of the DEM algorithm becomes a concern because the delay introduced by the DEM selection logic in the ΣΔ feedback loop can limit the maximum achievable clock frequency. Regarding this, DWA results in highly practical implementations, especially when the number of DAC elements is large [59].

Consider an M-element DAC, as that shown in Figure 1.26, with input code $y(n)$. In conventional DWA [63], the DAC unit elements are selected sequentially from the DAC array beginning with the next available unused element. An index pointer *ptr* stores the address of that element in a digital register and governs the rotational element selection process, so that the DAC elements selected at time n are those from $ptr(n)$ to $[ptr(n) + y(n) - 1]\ mod\ M$ by increasing order. Every clock cycle, the index pointer is incremented modulus M by the DAC input code $y(n)$ according to

$$ptr(n+1) = [ptr(n) + y(n)]\ mod\ M, \quad 0 \le ptr(n) \le M - 1 \tag{1.49}$$

Figure 1.27 illustrates how the elements of a 4-bit DAC would be selected following the DWA algorithm given earlier, which achieves a *first-order shaping* of the DAC mismatch errors.

However, if the ΣΔM input is DC or varies slowly, and there is a rational relation between its value and the number of DAC elements, out-of-band tones are generated that may fold back to the signal band due to the modulation by the ΣΔM's output waveform [63]. Several modifications of the conventional DWA algorithm have been proposed to reduce its *tonal behavior*, such as *rotated* data weighted averaging (R-DWA) [65], *randomized* data weighted averaging (Rn-DWA) [66], *bidirectional* data weighted

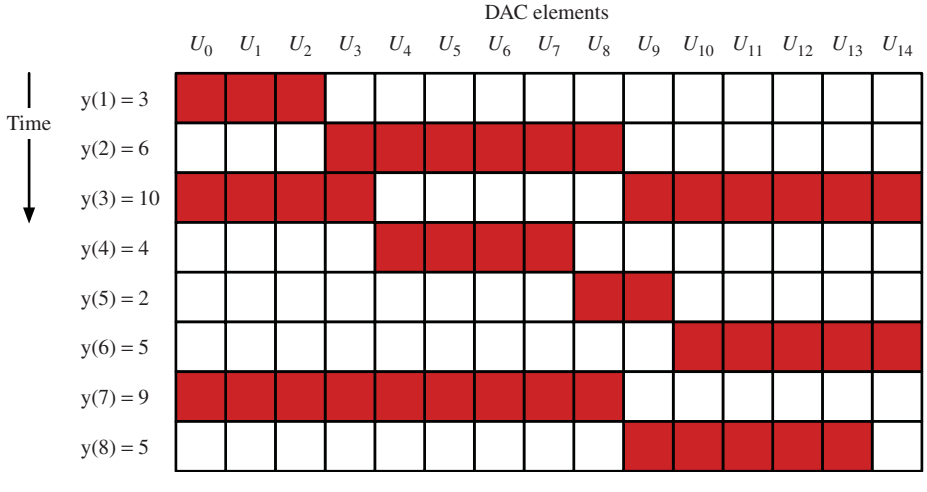


Figure 1.27 Selection of unit elements in a 15-element DAC according to the DWA algorithm in Equation 1.49. The shaded boxes indicate the elements that contribute to generate the DAC output level $y_{DAC}(n)$ for the corresponding input code $y(n)$.

averaging (Bi-DWA) [67], *partitioned* data weighted averaging (P-DWA) [51], and *pseudo* DWA [68].

Among them, pseudo DWA significantly decreases the tonal power with a minor modification of the DWA algorithm. Pseudo DWA modifies the DWA scheme by inverting with periodicity n_{inv} the least-significant bit (LSB) of the DAC input code $y(n)$ used to update the index pointer in Equation 1.49. The element-selection process is thus essentially the same as in conventional DWA except that, in every n_{inv} clock cycle, a DAC element is either reselected or skipped depending on whether the previous DAC input code was odd or even, respectively. This modification of the DWA algorithm breaks the cyclic nature of the element-selection process and, hence, reduces the tonal behavior [68]. The choice of n_{inv} is, however, a compromise between linearity and resolution. If n_{inv} is too large—note that $n_{\text{inv}} = \infty$ corresponds to conventional DWA—the signal-dependent tones will not be eliminated. If n_{inv} is too small, different DAC elements will be used at significantly different rates, what results in an increase of the in-band noise. Deriving an analytical expression for the optimum value of n_{inv} is rather complex, but simple behavioral simulations can be used to find an appropriate value for a given multibit ΣΔM [68].

1.6.3 Dual Quantization

Rather than reducing the DAC mismatch errors that are injected in the feedback of a multibit ΣΔM—as DEM techniques do—dual quantization basically consists in injecting them at a *different point*, where their influence on the overall ΣΔM linearity is not severe. The operation principle of dual quantization is to employ both single- and multibit quantization at a time in a ΣΔM: two-level quantization for its intrinsic linearity and multibit quantization for its reduced error power. The concept is applicable to both single-loop and cascade ΣΔMs.

Dual-Quantization Single-Loop ΣΔMs

Figure 1.28 shows a third-order, single-loop ΣΔM with dual quantization [69], in which the first two integrators are fed by a two-level DAC, whereas the third one is fed by a

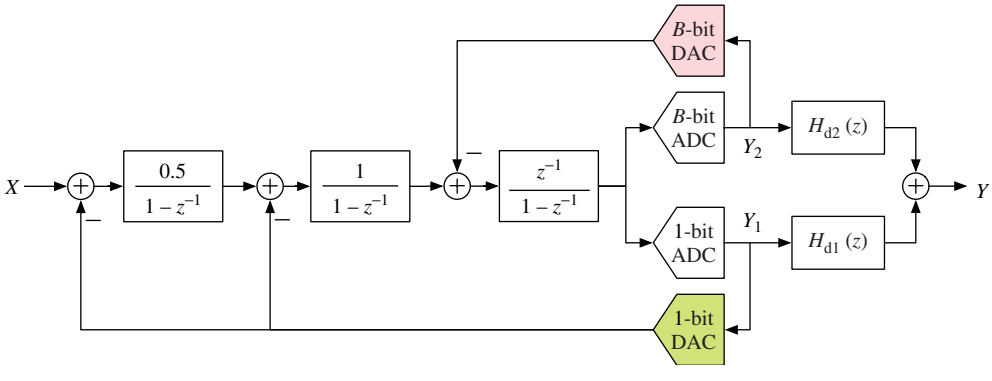


Figure 1.28 A third-order single-loop ΣΔM employing dual quantization [69].

multibit DAC. The outputs of the corresponding ADCs are digitally processed to cancel the *coarse* 1-bit quantization error at the overall output, in which only the *fine* multibit quantization error ideally remains. The modulator linearity is not menaced, as DAC mismatch errors are suppressed by the gain of the first two integrators and are thus second-order, high-pass shaped. The third-order ΣΔM also benefits from improved stability thanks to the multibit feedback in the back-end integrator. Note that the topology, however, requires a DCL and thus suffers from noise leakages.

The concept in Figure 1.28 can be generalized to higher-order ΣΔMs. As the order of the loop filter increases, the number of back-end integrators with multibit feedback is a trade-off between aggressive noise shaping (to improve stability) and linearity requirements of the multibit DAC.

Dual-Quantization Cascade ΣΔMs

Dual-quantization schemes are *naturally* incorporated to cascade ΣΔMs [70]. As shown in Section 1.5, the output of a cascade ΣΔM ideally contains only the input signal and the last-stage quantization error, whereas the quantization errors from the remaining stages are removed by the DCL. The DR of the ΣΔM can be thus easily increased using multibit quantization only in the back-end stage. The remaining quantizers can be single-bit ones to retain linear feedback in the front-end stages. Therefore, the resulting topology is that illustrated in Figure 1.21 with $B_{\mathrm{i}} = 1$ ($i = 1, \ldots, N-1$) and $B_{\mathrm{N}} = B$. This way, nonlinearities in the multibit DAC will appear at the overall cascade output with a shaping of order equal to the summation of the order of the preceding ΣΔ stages, so that Equation 1.43 yields

$$\begin{aligned} Y(z) = z^{-L}X(z) + \frac{1}{\prod_{i=1}^{N-1} c_{\mathrm{i}}}(1-z^{-1})^{L}E_{\mathrm{N}}(z) \\ + \frac{1}{\prod_{i=1}^{N-1} c_{\mathrm{i}}}(1-z^{-1})^{(L-L_{\mathrm{N}})}E_{\mathrm{DAC}}(z), \text{ with } L = \sum_{i=1}^{N} L_{\mathrm{i}} \end{aligned} \tag{1.50}$$

if the error in the back-end multibit DAC $E_{\mathrm{DAC}}(z)$ is taken into account in the linear analysis. Many cascade ΣΔM integrated circuits using this dual-quantization scheme can be found in the literature [4, 10, 18].

Many cascade ΣΔMs employing multibit quantization in all stages have also been reported [51, 67]. Note that, under ideal conditions, the quantization errors of all but the last stage are canceled at the overall cascade output by the DCL. Multibit quantization can be used in these stages with a twofold purpose as follows:

- For reducing the power of the respective quantization errors that will in practice leak to the output. DEM techniques can be employed in the front-end ΣΔ stage for achieving the targeted modulator linearity, whereas the remaining multibit stages often rely just on the shaping provided by the preceding integrators [51].
- For increasing the value of the interstage scaling coefficients c_{i} in the ΣΔ cascade—see Figure 1.21—in comparison with a 1-bit approach, while avoiding overloading [67]. This way, the factor $1/\prod_{i=1}^{N-1} c_{\mathrm{i}}$ that amplifies the last-stage quantization error in

Equation 1.50 can be made smaller than unity, what increases the DR over that ideally attainable with an Lth-order B-bit ΣΔM.

Cascade ΣΔMs using trilevel (1.5-bit) quantizers [53] are also often employed, as they yield a 3-dB reduction of the quantization error power compared to 1-bit quantization. Although trilevel coding is not inherently linear, it is often used in fully-differential SC ΣΔMs, because highly linear trilevel DACs can be easily implemented using just one extra switch [71].

1.7 Band-Pass ΣΔ Modulators

The operation principle of low-pass ΣΔ modulators (LP-ΣΔMs)—in which the quantization noise is high-pass shaped for its suppression around DC—can be extended to the more general case of stop-band filtering of the quantization noise for its suppression around a *nonzero* frequency in a so-called band-pass ΣΔ modulator (BP-ΣΔM) [72, 73]. A direct application of this band-pass approach can be found in the A/D conversion of many wireless receiver systems, in which a BP-ΣΔM is used for the digitization of *intermediate frequency* (IF) signals [18, 74].[9] Although a broadband Nyquist-rate ADC could a priori be an alternative to BP-ΣΔMs for the digitization in IF-conversion receivers, the bandwidth of IF signals is typically much smaller than the carrier frequency and the reduction of the quantization noise over the entire Nyquist band thus becomes very inefficient. Instead, if a BP-ΣΔM is used, the quantization noise is attenuated only in a narrowband around the IF location, thus taking advantage of a high OSR to achieve high DR requirements, even in the presence of strong interfering signals.

Figure 1.29 illustrates a typical block diagram of the IF-to-baseband section in a digital receiver based on a BP-ΣΔM. As shown, the BP-ΣΔM contains a BP loop filter $H(z)$ for obtaining a stop-band NTF with zeros placed at a nonzero frequency, often referred to as the *notch frequency* f_n. The digital output of the modulator is mixed to DC by a digital quadrature mixer and then LP filtered and decimated by a quadrature decimation filter to remove out-of-band spectral components and quantization noise. The resulting baseband digital data are finally processed in a DSP.[10]

Note that an LP-ΣΔM has its NTF zeros at or near $z = 1$ (which corresponds to DC), whereas those of a BP-ΣΔM are placed elsewhere on the unit circle; that is, $z = e^{j2\pi f_n/f_s}$ (which corresponds to f_n, with $0 < f_n < f_s/2$). As the NTF zeros occur in complex conjugate pairs, realizing L zeros of the NTF in the passband of a BP-ΣΔM thus requires a $2L$th-order BP loop filter. In other words, the shaping performed on the quantization noise by a $2L$th-order BP-ΣΔM is equivalent to that performed by an Lth-order LP-ΣΔM.

[9] The application of BP-ΣΔMs has also been extended toward the direct digitization of *radio frequency* (RF) signals. Although RF ΣΔ ADCs were initially implemented in SiGe processes [75, 76], they are already a reality in CMOS technologies too [77, 78]. Details on this can be found in Section 5.3.7.

[10] The IF-conversion of digital radios illustrated in Figure 1.29 can also be modified in such a way that the BP ΣΔ ADC directly digitizes quadrature input signals, using a so-called *quadrature* BP-ΣΔM [79, 80]. In that case, the quantization noise needs to be stop-band filtered only for positive (or negative) frequencies. Quadrature BP-ΣΔMs can thus be more efficient than conventional BP-ΣΔMs, as no power is dedicated to digitize the negative-frequency content of the input [18].

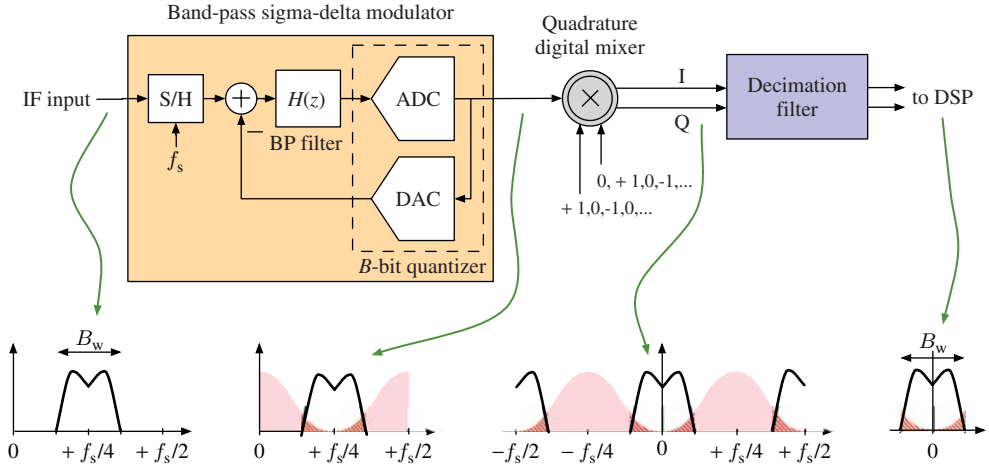


Figure 1.29 Typical section of an IF-conversion receiver based on a BP ΣΔ ADC. For the sake of illustration, the input IF frequency f_{IF} is assumed to be $f_s/4$ and a notch frequency $f_n = f_s/4$ is assumed in the BP-ΣΔM and in the quadrature digital mixer.

1.7.1 The $z \rightarrow -z^2$ LP–BP Transformation

A common design choice for the notch frequency of a BP-ΣΔM is $f_n = f_s/4$, because this location optimizes the trade-off between antialiasing filtering and image-rejection filtering in digital radio receivers [74]. It also helps to simplify the quadrature digital mixer to baseband, as the general digital cosine and sine sequences ($e^{j2\pi n f_n/f_s}$) reduce to a quadrature data series of +1s, 0s, and −1s, as illustrated in Figure 1.29. More importantly, the NTF zeros of a so-called $f_s/4$ BP-ΣΔM are placed at $z = \pm j$, so that the synthesis of the BP ΣΔ loop filter can be easily derived from an initial LP prototype by applying the transformation

$$z \xrightarrow[0-f_s/4]{\text{LP-BP}} -z^2 \tag{1.51}$$

which is often referred to as the $z \rightarrow -z^2$ LP–BP transformation of ΣΔ modulators.

This transformation is exemplary applied in Figure 1.30 to a second-order LP-ΣΔM to obtain a fourth-order $f_s/4$ BP-ΣΔM, in which the integrators are replaced by *resonators*. The resulting BP architecture preserves all properties of its LP original concerning dynamics, stability, resolution, etc. Indeed, it can be shown that the main performance figures (IBN, SNR, DR, etc.) of $f_s/4$ BP-ΣΔMs have the *same* expressions as the ones derived in previous sections for the LP case. For the sake of illustration, if the $z \rightarrow -z^2$ transformation is applied to the NTF of an ideal Lth-order LP ΣΔM in Equation 1.21, that resulting for an ideal $2L$th-order BP ΣΔM yields

$$\text{NTF}(z) = (1 + z^{-2})^L \tag{1.52}$$

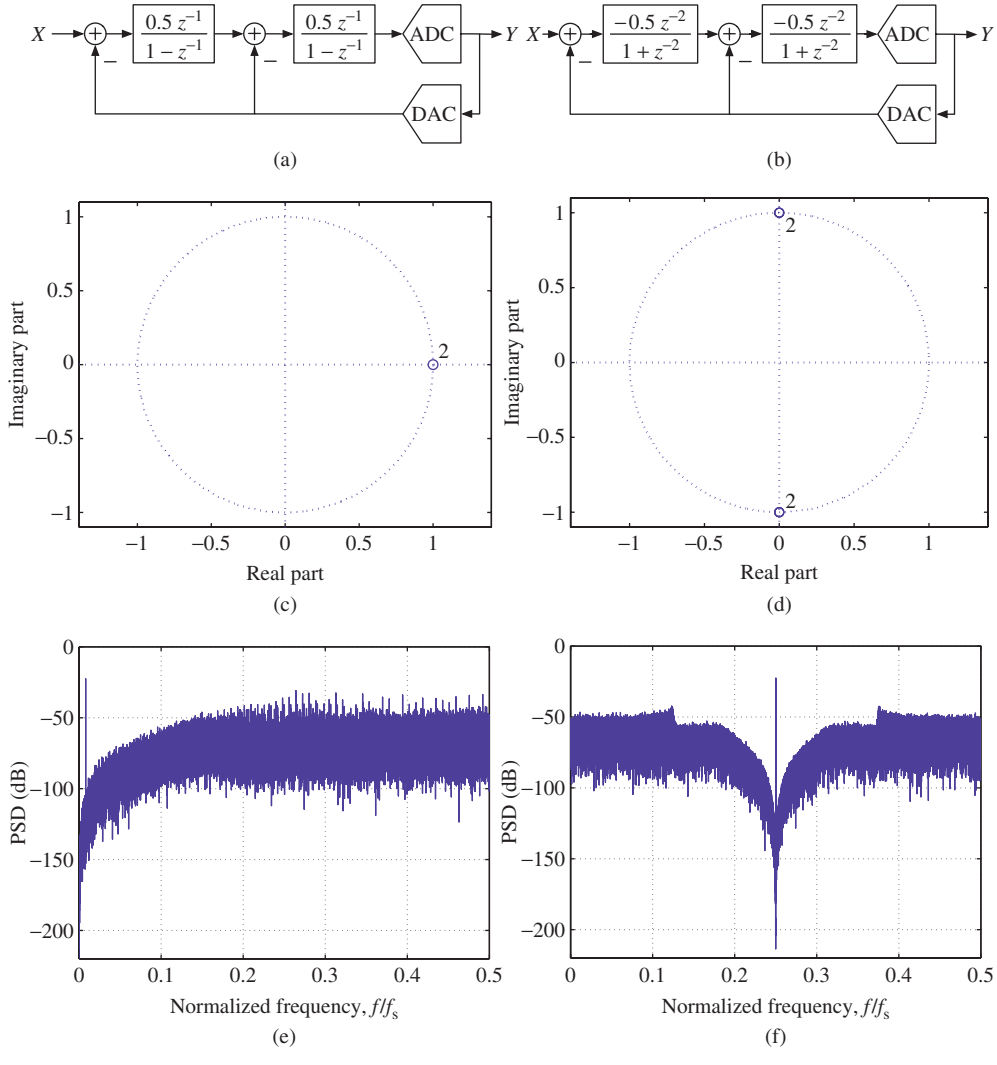


Figure 1.30 Illustration of the $z \rightarrow -z^2$ LP–BP transformation of ΣΔMs: (a) block diagram of a second-order LP-ΣΔM, (b) block diagram of the resulting fourth-order $f_s/4$ BP-ΣΔM, (c) zero-pole plot of the NTF of the LP-ΣΔM, (d) zero-pole plot of the NTF of the BP-ΣΔM, (e) output spectrum of the LP-ΣΔM ($f_{in} \approx f_s/128$), and (f) output spectrum of the BP-ΣΔM ($f_{in} \approx f_s/4$).

By integrating the shaped quantization noise over the signal band, the resulting IBN is obtained to be

$$\mathrm{IBN} = \int_{f_n - \frac{B_W}{2}}^{f_n + \frac{B_W}{2}} S_E(f)|\mathrm{NTF}(f)|^2 \mathrm{d}f \approx \frac{\Delta^2}{12} \frac{\pi^{2L}}{(2L+1)\mathrm{OSR}^{(2L+1)}} \tag{1.53}$$

so that the expression equals that of the LP case in Equation 1.12.

However, centering the signal passband at $f_s/4$ has some disadvantages. On the one hand, in the presence of nonlinearities in the analog circuitry of the ΣΔM, any intermodulation distortion products resulting from the mixing of tones at $f_s/2$ with the input signal will fall inside the modulator passband and will thus corrupt the signal information. On the other, for a given input IF, the clock rate demands are more restrictive than placing f_n between $f_s/4$ and $f_s/2$.[11]

1.7.2 BP-ΣΔMs with Optimized NTF

In practice, the $z \rightarrow -z^2$ transformation was commonly used, but many more custom-designed loop filters are nowadays reported, in which the synthesis of the optimized STF and NTF is directly addressed through the proper placement of their poles and zeros [82–84]—this is especially the case if the loop filter has to show special characteristics, such as adjacent-channel suppression or partial mixing inside the ΣΔM [16, 22, 23]. In this respect, BP-ΣΔMs exhibit the same architectural variety as LP-ΣΔMs and the trade-offs between the different structures are also essentially the same [18]. BP modulators can be implemented using either single-loop or cascade topologies, with a similar trade-off between improved stability and increased sensitivity to noise leakages due to nonidealities in the analog circuitry. Similarly, the loop filter of BP-ΣΔMs can be implemented using any of the topologies commonly employed for LP-ΣΔMs, such as the feedback, feed-forward, and hybrid topologies shown in Figures 1.18, 1.19, and 1.20, respectively.

Figure 1.31 shows two architectural alternatives for the implementation of a fourth-order BP-ΣΔM, which allow the optimization of both STF and NTF. The BP architecture in Figure 1.31a is based on the LP-ΣΔM with distributed feedback in Figure 1.18, whereas that in Figure 1.31b is based on the LP-ΣΔM with feed-forward summation in Figure1.19. Both band-pass topologies include input feed forward for eliminating the input-signal content from the loop filter and thus relaxing the requirements of the analog circuitry [29]. In addition, if designed properly, input feed forward flattens the STF to a constant without any peaking, what prevents undesirable amplification of out-of-band interferers [18].

Note from Figure 1.31 that the BP loop filters are obtained by shifting the poles from DC to nonzero frequencies by adding internal feedback paths (coefficients g_i) in the initial LP topologies. Figure 1.32a depicts the *lossless discrete integrator* (LDI) loop that builds the resonator,[12] which may be implemented using SC techniques as shown in Figure 1.32b.

[11] The $z \rightarrow -z^2$ transformation also offers the possibility of centering the IF input at $3f_s/4$ [81] instead of $f_s/4$, given that the spectrum is symmetrical with respect to $f_s/2$. Making $f_{IF} = 3f_s/4$ preserves the requirements of the AAF compared to the $f_{IF} = f_s/4$ case, but the image-rejection filter specifications can be relaxed. In addition, it allows for either the clock rate to be reduced to 1/3 or the signal processing to be three times faster. The only disadvantage is that OSR is reduced by a factor of 3.

[12] Note that one of the DT integrators has a z^{-1} delay in the numerator, whereas the other does not. A slightly less effective resonator would be implemented if both integrators had a z^{-1} delay, as the poles would be moved on a vertical line from the $(1, j0)$ point away from the real axis and would not exhibit infinite gain at the resonance frequency [4].

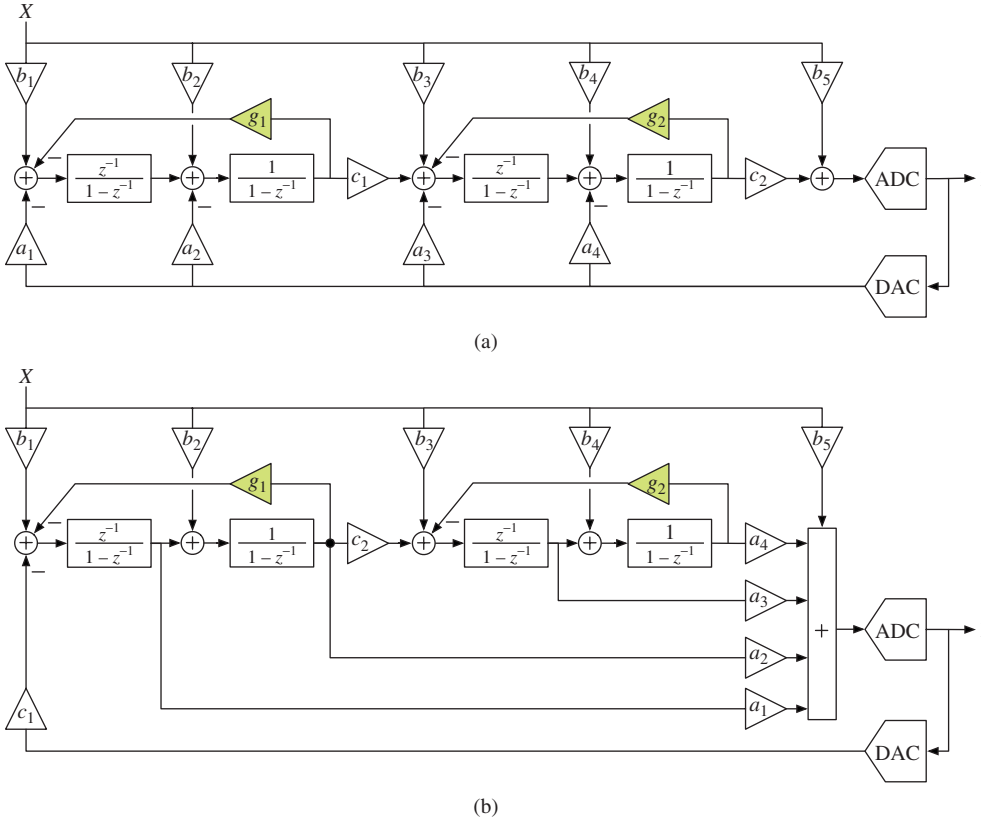


Figure 1.31 Illustration of fourth-order BP ΣΔ topologies that allow the NTF optimization: (a) cascade of resonators with feedback and (b) cascade of resonators with feed-forward summation.

The transfer function of the LDI resonator is given by

$$\mathrm{RTF}(z) = \frac{z^{-1}}{1-(2-g)z^{-1}+z^{-2}} \tag{1.54}$$

so that the poles are $z_p = (1-g/2) \pm j\sqrt{1-(1-g/2)^2}$. Note that, for $0 < g < 4$, the poles of RTF lie on the unit circle, and thus the quality factor Q of the resonator is ideally infinite.[13] The resonant frequency of a resonator corresponds to a notch in the NTF of the BP-ΣΔM at a frequency

$$\frac{2\pi f_n}{f_s} = \cos^{-1}\left(1-\frac{g}{2}\right) \tag{1.55}$$

[13] The resonator Q is limited in practice by the finite DC gain of the amplifiers. However, finite resonator Q is not usually a limiting nonideality, as $Q > 100$ is easily achieved.

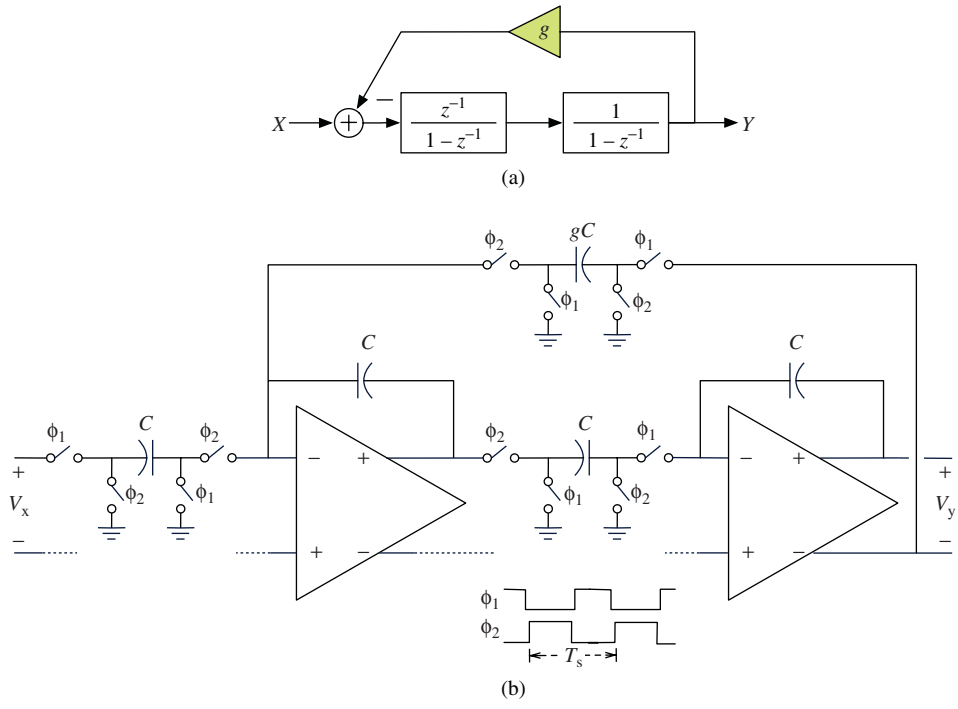


Figure 1.32 Illustration of an LDI resonator: (a) block diagram and (b) SC implementation. Only half of the differential circuit is explicitly shown for the sake of simplicity.

which can thus be placed at any arbitrary position from 0 to $f_s/2$ depending on the value of g—which is determined in an SC implementation by a capacitor ratio (Figure 1.32b).

Therefore, the center frequency of a BP-ΣΔM can be made *programmable* by changing the resonant frequency of its resonators [85]—for example, by using programmable banks of capacitors for the SC implementation of coefficients g_i. This feature is commonly exploited in digital radios, as the *same* programmable BP-ΣΔM can be used for the digitization of different IF frequencies.[14]

1.8 Continuous-Time ΣΔ Modulators

The majority of ΣΔMs reported up to recent years were implemented using DT circuit techniques, mostly based on SC circuits. However, the increasing demand for ever faster ADCs in broadband communication systems has raised the interest in CT-ΣΔMs over the past years, as they are able to operate at higher sampling rates with lower power consumption than their DT counterparts [16, 20, 75].

[14] However, this approach is only suitable when varying the modulator passband over a narrow frequency range. To maintain stability and SNR over a wide programmability range, other parameters such as the NTF poles, the STF, and the resonator signal swings need to be controlled in addition to the resonant frequencies by programming more modulator coefficients. In [86], this is done using SC techniques.

Figure 1.33 illustrates the general block diagram of a CT-ΣΔM for the case of LP input signals. To facilitate the direct comparison with DT implementations, the same partitioning as in Figure 1.7 has been adopted.

Looking at both figures, several differences are clearly visible between DT- and CT-ΣΔMs. The most significant one is related to the location of the sampling operation, which moves from the modulator input in DT-ΣΔMs to the point before the quantizer in CT-ΣΔMs. The loop filter can thus be realized using CT circuit techniques, but, given that the modulator output is a DT signal and the modulator input is a CT signal, a discrete-to-continuous time (DT–CT) transformation is required in CT-ΣΔMs to create the feedback signal. This can be better visualized comparing the signal processing involved in the block diagrams in Figures 1.34a and 1.34b.

The reconstruction of the modulator output signal is very critical in CT-ΣΔMs and has a significant impact on the overall modulator behavior [87]. There are a number of DAC waveforms that can be used in CT-ΣΔMs. Figure 1.35 shows a summary of the most representative possibilities, including the nomenclature used for the feedback waveforms

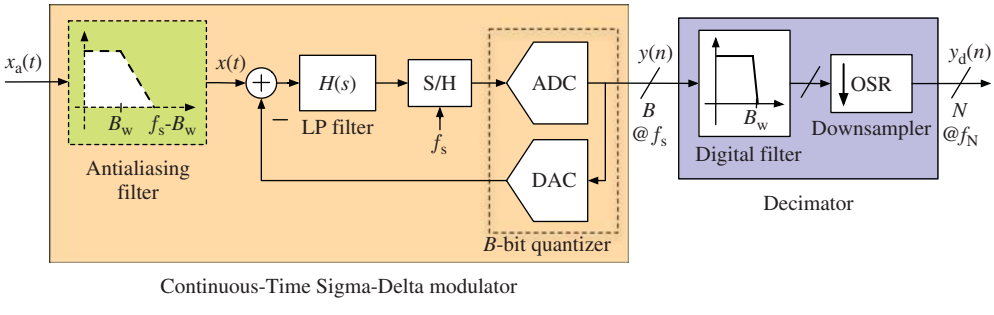


Figure 1.33 General block diagram of a continuous-time ΣΔ ADC. A low-pass ΣΔM is assumed.

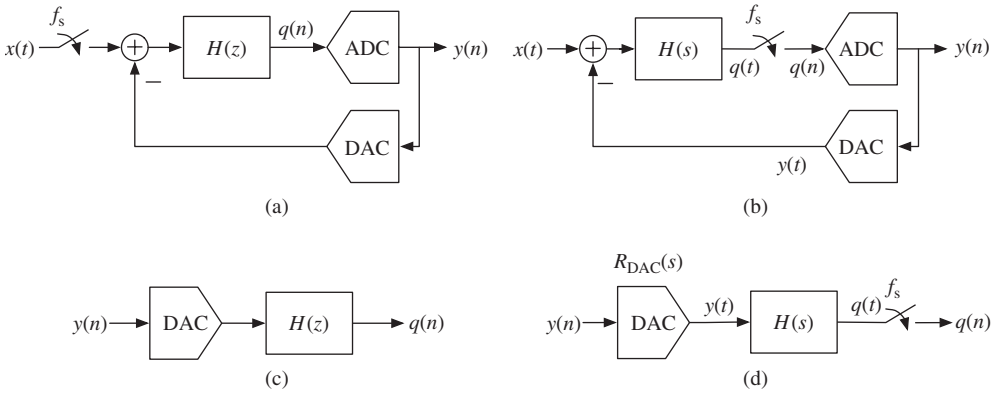


Figure 1.34 Conceptual block diagrams of ΣΔMs: (a) DT modulator, (b) CT modulator, (c) open-loop representation of a DT-ΣΔM, and (d) open-loop representation of a CT-ΣΔM.

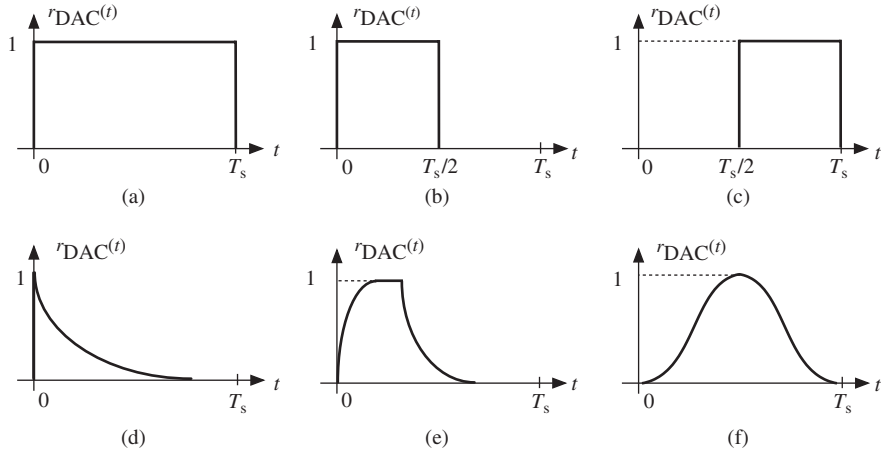


Figure 1.35 Most common DAC impulse responses: (a) nonreturn-to-zero (NRZ), (b) return-to-zero (RZ), (c) half-delay return-to-zero (HRZ), (d) switched capacitor (SC), (e) exponential slope, and (f) cosine.

extracted from [20]. Among others, the most commonly used DACs incorporate rectangular feedback pulses of basically three types: nonreturn-to-zero (NRZ) (Figure 1.35a), return-to-zero (RZ) (Figure 1.35b), and half-delay return-to-zero (HRZ) (Figure 1.35c). The DAC impulse response of these rectangular pulses can thus be globally expressed as

$$r_{\mathrm{DAC}}(t) = r_{(\alpha,\beta)}(t) = \begin{cases} 1, & \alpha T_s \le t < \beta T_s \\ 0, & \text{otherwise} \end{cases} \tag{1.56}$$

where (α, β) equals $(0, 1)$, $(0, 1/2)$, and $(1/2, 1)$ for NRZ, RZ, and HRZ DACs, respectively. Their Laplace S-transforms can also be generally written as

$$R_{\mathrm{DAC}}(s) = \frac{e^{-s\alpha T_s} - e^{-s\beta T_s}}{s} \tag{1.57}$$

Although they combine DT and CT signals, modulators fitting into the architecture in Figure 1.34b are generically considered as CT-ΣΔMs. Owing to the combination of both types of dynamics, together with the intrinsic nonlinearity associated to quantization, the mathematical analysis of CT-ΣΔMs becomes more difficult than in the case of DT-ΣΔMs [16, 20, 75, 84, 87]. However, they present several pros compared to DT-ΣΔMs, such as[15]

- An explicit AAF can be avoided (Figure 1.33). As the sampling operation takes place before the quantizer, the resulting STF is affected by a $\mathrm{sinc}(\pi f/f_s)$ function [24]. This sinc characteristic attenuates the signal spectrum exactly at multiples of the sampling frequency and thus leads to an *implicit AAF* in CT-ΣΔMs.

[15] CT-ΣΔMs also present several cons compared to DT-ΣΔMs, such as larger errors in the practical realization of the modulator coefficients, larger sensitivity to jitter, to loop delay, etc., which will be discussed in Chapter 2.

- Errors associated to the sampling process have less impact on the modulator performance. As the sampling operation takes place before the quantizer, the resulting errors attenuate in a similar way as the quantization error does.
- There is no *settling error* associated to the loop filter circuitry. As will be shown in Chapter 2, signals in DT circuits must settle to their steady-state values within a given accuracy, because complete settling would require infinite time.
- The operation speed is larger. This is inherent to the operation of CT circuits, in which the circuit dynamics is not a parasitic such as in DT circuits, but a design primitive.
- They are not affected by kT/C noise as SC-ΣΔMs do.

1.8.1 DT–CT Transformation of ΣΔMs

In the past, most work on ΣΔMs was focused on DT implementations, and great collective effort was made to develop innovative architectures, accurate models, and design and simulation CAD tools for DT-ΣΔMs. Therefore, a straightforward procedure for designing CT-ΣΔMs comes from considering an equivalent DT loop filter as a starting point, designing the DT-ΣΔM to meet the required performance, and then applying a DT–CT transformation.

The Impulse-Invariant Transformation

Let us consider again the block diagram of DT- and CT-ΣΔMs in Figures 1.34a and b, respectively, and the signal processing involved. The underlying principle of the DT–CT transformation consists in achieving the equivalence of both modulators by imposing that the input of the quantizers are the same at the sampling instants; that is,

$$q(n) = q(t)|_{t=nT_s} \tag{1.58}$$

If this condition is met, the output bitstreams of both modulators and the noise performance would thus be identical. As illustrated in Figures 1.34c and d, the condition in Equation 1.58 can be translated into an equivalence of the input–output signal processing that is performed in both ΣΔMs in open-loop configuration, which yields

$$\mathcal{Z}^{-1}\{H(z)\} = \mathcal{L}^{-1}\{R_{DAC}(s)H(s)\}|_{t=nT_s} \tag{1.59}$$

where $\mathcal{Z}^{-1}\{\cdot\}$ and $\mathcal{L}^{-1}\{\cdot\}$ stand for the inverse Z- and L-transform operators, respectively, and R_{DAC} stands for the CT transfer function of the DAC (see Equation 1.57 for the case of rectangular waveforms). In the time domain, this leads to the condition

$$h(n) = [r_{DAC}(t) * h(t)]|_{t=nT_s} = \int_{-\infty}^{+\infty} r_{DAC}(\tau)h(t-\tau)d\tau|_{t=nT_s} \tag{1.60}$$

where $*$ stands for the convolution operator and $r_{DAC}(t)$ stands for the impulse response of the specific DAC (Equation 1.56 for rectangular waveforms). This transformation between DT and CT domain is known as the *invariant-impulse transformation* (IIT), because it makes the open-loop impulse responses of both ΣΔMs equal at the sampling instants.

Table 1.3 CT equivalents of first-order to fourth-order DT low-pass loop filter poles for the rectangular feedback DAC pulses defined in equation 1.56 [20]

Z-Domain	S-Domain Equivalents with $f_s(\text{Hz}) = 1/T_s$
$\frac{1}{z-1}$	$\frac{\omega_0}{s},\ \omega_0 = \frac{f_s}{\beta-\alpha}$
$\frac{1}{(z-1)^2}$	$\frac{\omega_1 s+\omega_0}{s^2},\ \omega_0 = \frac{f_s^2}{\beta-\alpha},\ \omega_1 = \frac{f_s(\alpha+\beta-2)}{2(\beta-\alpha)}$
$\frac{1}{(z-1)^3}$	$\frac{\omega_2 s^2+\omega_1 s+\omega_0}{s^3},\ \omega_0 = \frac{f_s^3}{\beta-\alpha},\ \omega_1 = \frac{f_s^2(\alpha+\beta-3)}{2(\beta-\alpha)},$ $\omega_2 = \frac{f_s[\beta(\beta-9)+\alpha(\alpha-9)+4\alpha\beta+12]}{12(\beta-\alpha)}$
$\frac{1}{(z-1)^4}$	$\frac{\omega_3 s^3+\omega_2 s^2+\omega_1 s+\omega_0}{s^4},\ \omega_0 = \frac{f_s^4}{\beta-\alpha},\ \omega_1 = \frac{f_s^3(\alpha+\beta-4)}{2(\beta-\alpha)},$ $\omega_2 = \frac{f_s^2[(\beta-\alpha)^2+2\alpha\beta-12(\alpha+\beta)+22]}{12(\beta-\alpha)},$ $\omega_3 = \frac{f_s[\beta^2(\alpha-2)+\alpha^2(\beta-2)-8\alpha\beta+11(\alpha+\beta)-12]}{12(\beta-\alpha)}$

On the basis of the IIT in Equation 1.59, the most usual procedure to design CT-ΣΔMs consists of: first, matching the equivalent filter $H(z)$ with a reference DT loop filter chosen to fulfill the specifications; then, solving for the coefficients of the CT loop filter $H(s)$ with the specific DAC response considered; and, finally, implementing $H(s)$ with CT techniques, usually based on Gm-C or active-RC techniques.[16] In theory, any arbitrary ΣΔM topology, either single-loop or cascade, can be synthesized this way.

The information compiled in Tables 1.3 and 1.4 for the equivalence of LP loop filter poles at DC ($z = 1$, $s = 0$) in DT and CT domains is useful in such a methodology [20]. For general poles, the original table in [88] can be employed, which can also be useful for the design of BP CT-ΣΔMs. Finally, the use of these tables of equivalent poles is easy and can also be automated and, moreover, extended to other DAC impulse responses different from the rectangular waveforms, as done in [20].

DT–CT Transformation of a Second-Order ΣΔM

For the sake of illustration, Figure 1.36 shows a second-order CT-ΣΔM obtained using such a method from its DT equivalent in Figure 1.14a. Note from the former figure that the DT integrators—with transfer function given by Equation 1.16—are transformed into

[16] Careful choice of the CT filter structure is needed to have sufficient degrees of freedom to implement the reference DT loop filter [84].

Table 1.4 DT Equivalents of first-order to fourth-order CT low-pass loop filter poles for the rectangular feedback DAC pulses defined in equation 1.56 [20]

S-Domain	Z-Domain Equivalents with $f_s(\text{Hz}) = 1/T_s$
$\frac{f_s}{s}$	$\frac{\omega_0}{z-1},\ \omega_0 = \beta - \alpha$
$\frac{f_s^2}{s^2}$	$\frac{\omega_1 z + \omega_0}{(z-1)^2},\ \omega_0 = \frac{(\beta^2 - \alpha^2)}{2},\ \omega_1 = \frac{[\beta(1-\beta) - \alpha(1-\alpha)]}{2}$
$\frac{f_s^3}{s^3}$	$\frac{\omega_2 z^2 + \omega_1 z + \omega_0}{(z-1)^3},\ \omega_0 = \frac{(\beta^3 - \alpha^3)}{6},$
	$\omega_1 = -\frac{(\beta^3 - \alpha^3)}{3} + \frac{(\beta^2 - \alpha^2)}{2} + \frac{(\beta - \alpha)}{2},$
	$\omega_2 = -\frac{(\beta^3 - \alpha^3)}{6} - \frac{(\beta^2 - \alpha^2)}{2} + \frac{(\beta - \alpha)}{2}$
$\frac{f_s^4}{s^4}$	$\frac{\omega_3 z^3 + \omega_2 z^2 + \omega_1 z + \omega_0}{(z-1)^4},\ \omega_0 = \frac{(\beta^4 - \alpha^4)}{24},$
	$\omega_1 = -\frac{(\beta^4 - \alpha^4)}{8} + \frac{(\beta^3 - \alpha^3)}{6} + \frac{(\beta^2 - \alpha^2)}{4} + \frac{(\beta - \alpha)}{6},$
	$\omega_2 = \frac{(\beta^4 - \alpha^4)}{8} - \frac{(\beta^3 - \alpha^3)}{3} + \frac{2(\beta - \alpha)}{3},$
	$\omega_3 = -\frac{(\beta^4 - \alpha^4)}{24} + \frac{(\beta^3 - \alpha^3)}{6} - \frac{(\beta^2 - \alpha^2)}{4} + \frac{(\beta - \alpha)}{6}$

CT integrators with *unscaled* transfer functions given by

$$\text{ITF}(s) = \frac{1}{sT_s} = \frac{f_s}{s} \tag{1.61}$$

and that the scaling coefficients k_i of the CT-ΣΔM are associated to the feedback paths [20]. This notation has been adopted because it can easily account for some nonidealities in the loop filter, as will be shown in Chapter 2. Note from Figure 1.36 that the discrimination of the signal scaling coefficient k_{sig} and the first feedback scaling coefficient k_1 accounts for changing feedback coefficients when adopting DAC feedback pulses different from the rectangular NRZ [20]. Therefore, the input scaling coefficient remains equally constant to $k_{1,NRZ}$ regardless of the particular feedback waveform that is used.

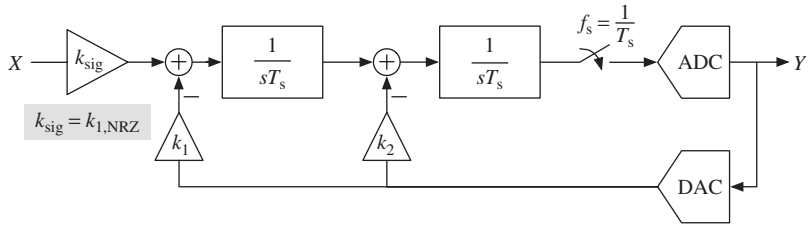


Figure 1.36 Block diagram of a second-order CT-ΣΔM using the notation in [20].

The CT equivalent of the DT-ΣΔM in Figure 1.14a can be easily calculated by, on the one hand, applying the IIT in Equation 1.59 to the DT loop filter

$$H(z) = -\frac{a_1 a_2}{(z-1)^2} - \frac{a_2}{z-1} \xrightarrow[\text{NRZ}]{\text{DT-CT}} H(s) = -a_1 a_2 \frac{f_s^2}{s^2} - \left(a_2 - \frac{a_1 a_2}{2}\right)\frac{f_s}{s} \tag{1.62}$$

where an NRZ rectangular waveform has been exemplary assumed in the feedback DAC of the CT equivalent. To that purpose, rows 1 and 2 of Table 1.3 have been adopted considering $(\alpha, \beta) = (0, 1)$. On the other hand, the CT loop filter in Figure 1.36 is obtained as

$$H(s) = -k_1 \frac{f_s^2}{s^2} - k_2 \frac{f_s}{s} \tag{1.63}$$

Equating the CT coefficients in Equations 1.62 and 1.63 results in the following relations between the coefficients of the second-order DT-ΣΔM and the CT coefficients for the case of an NRZ DAC:

$$\begin{aligned} k_{1,\text{NRZ}} &= a_1 a_2 \\ k_{2,\text{NRZ}} &= a_2 - \frac{a_1 a_2}{2} \end{aligned} \tag{1.64}$$

The DT scaling coefficients $(a_1, a_2) = (0.5, 0.5)$ typically used—see Table 1.1 and optimal coefficients reported in [8]—thus result in $k_{1,\text{NRZ}} = 0.25$ and $k_{2,\text{NRZ}} = 0.375$.

For the sake of illustration, Figure 1.37 shows a possible implementation of the resulting CT-ΣΔM using active-RC integrators. Note that the corner frequency ω_I of an active-RC integrator is determined by the value of $1/(RC)$, so that the scaling coefficients of the CT-ΣΔM yield [20]:

$$\text{ITF}_i(s) = \frac{\omega_{Ii}}{s} = k_i \frac{f_s}{s} = \frac{1}{sRC} \quad \Rightarrow \quad k_i f_s = \omega_{Ii} = \frac{1}{RC} \tag{1.65}$$

For the second-order CT-ΣΔM in Figures 1.36 and 1.37, the relations to be fulfilled for the correct implementation of the modulator coefficients are thus

$$\begin{aligned} R_{in1} C_{I1} &= \tfrac{1}{k_{sig} f_s}, & R_{DAC1} C_{I1} &= \tfrac{1}{k_1 f_s} \\ R_{in2} C_{I2} &= \tfrac{1}{f_s}, & R_{DAC2} C_{I2} &= \tfrac{1}{k_2 f_s} \end{aligned} \tag{1.66}$$

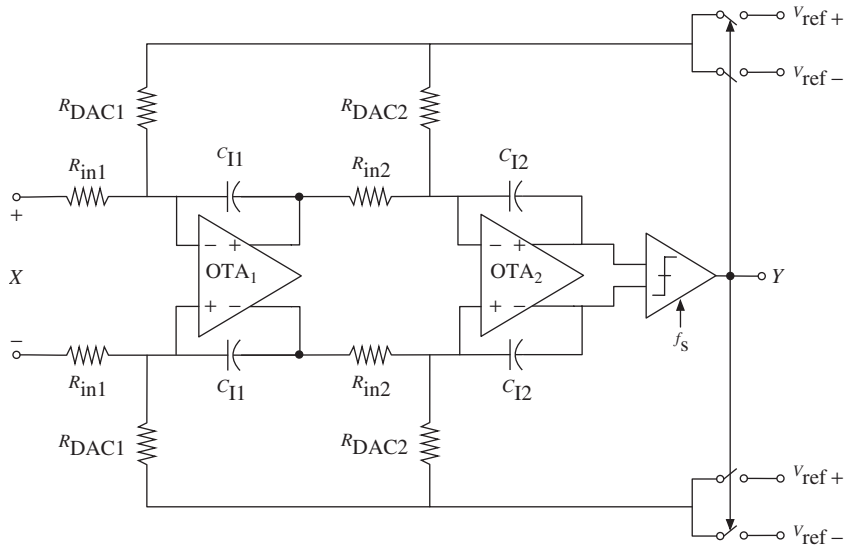


Figure 1.37 Active-RC implementation of a second-order CT-ΣΔM with a single-bit NRZ DAC.

which, for the particular case of NRZ DAC considered, yield

$$\begin{aligned} R_{\text{in1}} &= R_{\text{DAC1}}, & R_{\text{DAC1}}C_{\text{I1}} &= \frac{1}{0.25 f_s} \\ R_{\text{in2}} &= 0.375 R_{\text{DAC2}}, & R_{\text{DAC2}}C_{\text{I2}} &= \frac{1}{0.375 f_s} \end{aligned} \tag{1.67}$$

1.8.2 Direct Synthesis of CT-ΣΔMs

The main problem of using the DT–CT transformation method given earlier is that it may yield an increase of complexity in the analog circuitry, with the subsequent penalty in sensitivity to variations on the technological process parameters. This is particularly critical in the case of CT cascades, where, to get a functional modulator while keeping the digital cancelation logic of the original DT-ΣΔM, every integrator and DAC output must be connected to the integrator input of later stages [20].

An alternative method for designing the CT loop filter directly uses the desired NTF(f) as a starting point, just in the same way as described in Section 1.4.2 for synthesizing optimized NTFs in the DT case. This synthesis method is often referred to as *direct CT synthesis method*. Usually, an inverse Chebyshev distribution of the NTF(f) zeros is considered because it has advantages in terms of SNR and stability. Once the desired NTF(f) has been chosen, the loop filter can be derived from the linearized model [16].

It can be shown that applying the direct synthesis method to cascade CT-ΣΔMs, more efficient architectures are obtained in terms of loop filter optimization, analog circuitry complexity, power consumption, and robustness to circuit element tolerance errors [89]. However, there are two major drawbacks to this method: first, previous knowledge of

DT-ΣΔMs is not reused, which raises questions about stability; second, simulations are harder due to the fact that every simulation has to be done in CT.

1.9 Summary

This chapter has presented an introduction to ΣΔ ADCs. The benefits of employing oversampling and quantization noise shaping in the digitization of signals have been analyzed and compared to the performance of Nyquist-rate ADCs. Among the several blocks that build up an ΣΔ ADC, the chapter has focused on the ΣΔ modulator. Its general topology, ideal operation, and performance metrics have been presented.

The existing methods for increasing the effective resolution of an ΣΔ modulator have been discussed, presenting practical ways to implement stable high-order noise shaping topologies and to resort to multibit quantization. The DT and CT implementation of ΣΔ modulators have been addressed, as well as their application to convert LP or BP signals. The implications of these different alternatives at the architectural and circuit levels have been presented in an incremental way, starting from the initial case of DT, LP single-bit modulators.

These basic concepts have been discussed mainly considering the quantization noise as the only source of error limiting the resolution of an ΣΔ modulator. The effect of nonidealities associated with the practical circuit implementation of the modulator blocks will be analyzed in Chapter 2.

References

[1] W. Bennett, "Spectra of Quantized Signals," *Bell System Technical Journal*, vol. **27**, pp. 446–472, July 1948.

[2] B. Widrow, "Statistical Analysis of Amplitude-Quantized Sampled-Data Systems," *Transactions of the AIEE - Part II: Applications and Industry*, vol. **79**, pp. 555–568, January 1960.

[3] A. B. Sripad and D. L. Snyder, "A Necessary and Sufficient Condition for Quantization Errors to be Uniform and White," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. **25**, pp. 442–448, October 1977.

[4] S. R. Norsworthy, R. Schreier, and G. C. Temes, *Delta-Sigma Data Converters: Theory, Design and Simulation*, IEEE Press, 1997.

[5] R. van de Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*, Springer, 2003.

[6] H. Inose, Y. Yasuda, and J. Murakami, "A Telemetering System by Code Modulation $-\Delta-\Sigma$ Modulation," *IRE Transactions on Space Electronics and Telemetry*, vol. **8**, pp. 204–209, September 1962.

[7] B. E. Boser and B. A. Wooley, "The Design of Sigma-Delta Modulation Analog-to-Digital Converters," *IEEE Journal of Solid-State Circuits*, vol. **23**, pp. 1298–1308, December 1988.

[8] A. Marques, V. Peluso, M. S. Steyaert, and W. M. Sansen, "Optimal Parameters for ΔΣ Modulator Topologies," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. **45**, pp. 1232–1241, September 1998.

[9] Y. Geerts, M. Steyaert, and W. Sansen, *Design of Multi-bit Delta-Sigma A/D Converters*, Kluwer Academic Publishers, 2002.

[10] A. Rodríguez-Vázquez, F. Medeiro, J. M. de la Rosa, R. del Río, R. Tortosa, and B. Pérez-Verdú, "Sigma-Delta CMOS ADCs: An Overview of the State-of-the-Art," Chapter 2 in *CMOS Telecom Data Converters* (A. Rodríguez-Vázquez, F. Medeiro, and E. Janssens, Editors), Kluwer Academic Publishers, 2003.

[11] J. Candy and G. Temes, *Oversampling Delta-Sigma Data Converters: Theory, Design and Simulation*, IEEE Press, 1991.

[12] F. Medeiro, B. Pérez-Verdú, and A. Rodríguez-Vázquez, *Top-Down Design of High-Performance Sigma-Delta Modulators*, Kluwer Academic Publishers, 1999.

[13] J. Cherry and W. Snelgrove, *Continuous-Time Delta-Sigma Modulators for High-Speed A/D Conversion*, Kluwer Academic Publishers, 1999.
[14] V. Peluso, M. Steyaert, and W. Sansen, *Design of Low-Voltage Low-Power CMOS Delta-Sigma A/D Converters*, Kluwer Academic Publishers, 1999.
[15] S. Rabii and B. A. Wooley, *The Design of Low-Voltage, Low-Power Sigma-Delta Modulators*, Kluwer Academic Publishers, 1999.
[16] L. Breems and J. H. Huijsing, *Continuous-Time Sigma-Delta Modulation for A/D Conversion in Radio Receivers*, Kluwer Academic Publishers, 2001.
[17] J. M. de la Rosa, B. Pérez-Verdú, and A. Rodríguez-Vázquez, *Systematic Design of CMOS Switched-Current Bandpass Sigma-Delta Modulators for Digital Communication Chips*, Kluwer Academic Publishers, 2002.
[18] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*, Wiley-IEEE Press, 2005.
[19] R. del Río, F. Medeiro, B. Pérez-Verdú, J. M. de la Rosa, and A. Rodríguez-Vázquez, *CMOS Cascade ΣΔ Modulators for Sensors and Telecom: Error Analysis and Practical Design*, Springer, 2006.
[20] M. Ortmanns and F. Gerfers, *Continuous-Time Sigma-Delta A/D Conversion: Fundamentals, Performance Limits and Robust Implementations*, Springer, 2006.
[21] L. Yao, M. Steyaert, and W. Sansen, *Low-Power Low-Voltage Sigma-Delta Modulators in Nanometer CMOS*, Springer, 2006.
[22] P. G. R. Silva and J. H. Huijsing, *High Resolution IF-to-Baseband ΣΔ ADC for Car Radios*, Springer, 2008.
[23] R. H. van Veldhoven and A. H. M. van Roermund, *Robust Sigma Delta Converters*, Springer, 2011.
[24] J. Candy, "A Use of Double Integration in Sigma-Delta Modulation," *IEEE Transactions on Communications*, vol. **33**, pp. 249–258, March 1985.
[25] G. Yin and W. Sansen, "A High-Frequency and High-Resolution Fourth-Order ΣΔ A/D Converter in BiCMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. **29**, pp. 857–865, August 1994.
[26] F. Medeiro, B. Pérez-Verdú, J. M. de la Rosa, and A. Rodríguez-Vázquez, "Multi-Bit Cascade ΣΔ Modulator for High-Speed A/D Conversion with Reduced Sensitivity to DAC Errors," *IET Electronics Letters*, vol. **34**, pp. 422–424, March 1998.
[27] A. M. Marques, V. Peluso, M. S. J. Steyaert, and W. M. Sansen, "A 15-b Resolution 2-MHz Nyquist Rate ΔΣ ADC in a 1-μm CMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. **33**, pp. 1065–1075, July 1998.
[28] P. Benabes, A. Gauthier, and D. Billet, "New Wideband Sigma-Delta Convertor," *IET Electronics Letters*, vol. **27**, pp. 1575–1577, August 1993.
[29] J. Silva, U. Moon, J. Steensgaard, and G. C. Temes, "Wideband Low-Distortion Delta-Sigma ADC Topology," *IET Electronics Letters*, vol. **37**, pp. 737–738, June 2001.
[30] R. Gaggl, M. Inversi, and A. Wiesbauer, "A Power Optimized 14-Bit SC ΔΣ Modulator for ADSL CO Applications," *IEEE ISSCC Digest of Technical Papers*, pp. 82–83, February 2004.
[31] K. Y. Nam, S. M. Lee, D. K. Su, and B. A. Wooley, "A Low-Voltage Low-Power Sigma-Delta Modulator for Broadband Analog-to-Digital Conversion," *IEEE Journal of Solid-State Circuits*, vol. **40**, pp. 1855–1864, September 2005.
[32] T. Christen, T. Burger, and Q. Huang, "A 0.13 μm CMOS EDGE/UMTS/WLAN Tri-Mode ΣΔ ADC with -92dB THD," *IEEE ISSCC Digest of Technical Papers*, pp. 240–241, February 2007.
[33] T. Christen and Q. Huang, "A 0.13 μm CMOS 0.1-20 MHz Bandwidth 86-70 dB DR Multi-Mode DT ΔΣ ADC for IMT-Advanced," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 414–417, September 2010.
[34] A. Morgado, R. del Río, J. M. de la Rosa, L. Bos, J. Ryckaert, and G. van der Plas, "A 100 kHz-10 MHz BW, 78-to-52 dB DR,4.6-to-11 mW Flexible SC ΣΔ Modulator in 1.2-V 90-nm CMOS," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 418–421, September 2010.
[35] R. W. Adams, P. F. Ferguson, A. Ganesan, S. Vincelette, A. Volpe, and R. Libert, "Theory and Practical Implementation of a Fifth-Order Sigma-Delta A/D Converter," *Journal of the Audio Engineering Society*, vol. **39**, pp. 515–528, July 1991.
[36] R. W. Adams and R. Schreier, "Stability Theory in ΣΔ Modulators," Chapter 4 in *Delta-Sigma Data Converters: Theory, Design and Simulation* (S. R. Norsworthy, R. Schreier, and G. C. Temes, Editors), IEEE Press, 1997.
[37] F. O. Eynde, "High-Performance Analog Interfaces for Digital Signal Processors," PhD Thesis, Katholieke Universiteit Leuven, 1990.

[38] W. L. Lee and C. G. Sodini, "A Topology for Higher Order Interpolative Coders," Proc. of the IEEE Intl. Symp. on Circuits and Systems, pp. 459–462, May 1987.

[39] S. H. Ardalan and J. J. Paulos, "An Analysis of Nonlinear Behavior in Delta-Sigma Modulators," *IEEE Transactions on Circuits and Systems*, vol. **34**, pp. 593–603, June 1987.

[40] L. A. Williams and B. A. Wooley, "Third-Order Cascaded Sigma-Delta Modulators," *IEEE Transactions on Circuits and Systems*, vol. **38**, pp. 489–498, May 1991.

[41] R. W. Adams, "The Design of High-Order Single-Bit ΣΔ ADCs," Chapter 5 in *Delta-Sigma Data Converters: Theory, Design and Simulation* (S. R. Norsworthy, R. Schreier, and G. C. Temes, Editors), IEEE Press, 1997.

[42] D. R. Welland, B. P. del Signore, E. J. Swanson, T. Tanaka, K. Hamashita, S. Hara, and K. Takasuka, "A Stereo 16-Bit Delta-Sigma A/D Converter for Digital Audio," *Journal of the Audio Engineering Society*, vol. **37**, pp. 476–486, July 1989.

[43] P. F. Ferguson, A. Ganesan, and R. W. Adams, "One Bit Higher Order Sigma-Delta A/D Converters," *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. **2**, pp. 890–893, May 1990.

[44] R. Schreier, "An Empirical Study of Higher Order Single Bit Sigma Delta Modulators," *IEEE Transactyions on Circuits and Systems –II: Analog and Digital Signal Processing*, vol. **40**, pp. 461–466, August 1993.

[45] Y. Matsuya, K. Uchimura, A. Iwata, T. Kobayashi, M. Ishikawa, and T. Yoshitome, "A 16-bit Oversampling A-to-D Conversion Technology Using Triple-Integration Noise Shaping," *IEEE Journal of Solid-State Circuits*, vol. **22**, pp. 921–929, December 1987.

[46] L. Longo and M. Copeland, "A 13 bit ISDN-Band Oversampled ADC Using Two-Stage Third-Order Noise Shaping," *Proc. of the IEEE Custom Integrated Circuit Conf.*, pp. 21.2.1–4, 1988.

[47] W. Chou, P. Wong, and R. Gray, "Multi-Stage Sigma-Delta Modulation," *IEEE Transactions on Information Theory*, vol. **35**, pp. 784–796, July 1989.

[48] M. Rebeschini, N. R. van Bavel, P. Rakers, R. Greene, J. Caldwell, and J. R. Haug, "A 16-b 160 kHz CMOS A/D Converter Using Sigma-Delta Modulation," *IEEE Journal of Solid-State Circuits*, vol. **25**, pp. 431–440, April 1990.

[49] T. Karema, T. Ritoniemi, and H. Tenhunen, "An Oversampled Sigma-Delta A/D Converter Circuit Using Two-Stage Fourth Order Modulator," *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. **4**, pp. 3279–3282, 1990.

[50] H. Baher and E. Afifi, "Novel Fourth-Order Sigma-Delta Convertor," *IET Electronics Letters*, vol. **28**, pp. 1437–1438, July 1992.

[51] K. Vleugels, S. Rabii, and B. Wooley, "A 2.5-V Sigma-Delta Modulator for Broadband Communications Applications," *IEEE Journal of Solid-State Circuits*, vol. **36**, pp. 1887–1899, December 2001.

[52] R. del Río, F. Medeiro, B. Pérez-Verdú, and A. Rodríguez-Vázquez, "High-Order Cascade Multibit ΣΔ Modulators for xDSL Applications," *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. **2**, pp. 37–40, 2000.

[53] I. Dedic, "A Sixth-Order Triple-Loop ΣΔ CMOS ADC with 90 dB SNR and 100 kHz Bandwidth," *IEEE ISSCC Digest of Technical Papers*, pp. 188–189, 1994.

[54] K. Cornelissens and M. Steyaert, "A 1-V 84-dB DR 1-MHz Bandwidth Cascade 3-1 Delta-Sigma ADC in 65-nm CMOS," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 332–335, 2009.

[55] R. Tortosa, A. Aceituno, J. M. de la Rosa, A. Rodríguez-Vázquez, and F. V. Fernández, "A 12-bit, 40MS/s Gm-C Cascade 3-2 Continuous-Time Sigma-Delta Modulator," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 1–4, 2007.

[56] R. del Río, F. Medeiro, J. M. de la Rosa, B. Pérez-Verdú, and A. Rodríguez-Vázquez, "Reliable Analysis of Settling Errors in SC Integrators: Application to ΣΔ Modulators," *IET Electronics Letters*, vol. **36**, pp. 503–504, March 2000.

[57] T. Brooks, "Architecture Considerations for Multi-Bit ΣΔ ADCs," Chapter in *Analog Circuit Design –Structured Mixed-Mode Design, Multi-Bit Sigma-Delta Converters, Short Range RF Circuits* (M. Steyaert, A. H. M. van Roermund, and J. H. Huijsing, Editors), Kluwer Academic Publishers, 2002.

[58] L. R. Carley, R. Schreier, and G. C. Temes, "Delta-Sigma ADCs with Multibit Internal Converters," Chapter 8 in *Delta-Sigma Data Converters: Theory, Design and Simulation* (S. R. Norsworthy, R. Schreier, and G. C. Temes, Editors), IEEE Press, 1997.

[59] A. A. Hamoui and K. W. Martin, "High-Order Multibit Modulators and Pseudo Data-Weighted-Averaging in Low-Oversampling $\Delta\Sigma$ ADCs for Broad-Band Applications," *IEEE Trans. on Circuits and Systems –I: Regular Papers*, pp. 72–85, January 2004.

[60] L. R. Carley and J. Kenney, "A 16-bit 4'th Order Noise-Shaping D/A Converter," Proc. of the IEEE Custom Integrated Circuits Conf., pp. 21.7.1–4, 1988.

[61] K. B. Klaasen, "Digitally Controlled Absolute Voltage Division," *IEEE Transactions on Instrumentation and Measurement*, vol. **24**, pp. 106–112, June 1975.

[62] B. Leung and S. Sutarja, "Multibit $\Sigma-\Delta$ A/D Converter Incorporating a Novel Class of Dynamic Element Matching Techniques," *IEEE Transactions on Circuits and Systems –II: Analog and Digital Signal Processing*, vol. **39**, pp. 35–51, January 1992.

[63] R. T. Baird and T. Fiez, "Linearity Enhancement of Multibit $\Delta\Sigma$ A/D and D/A Converters using Data Weighted Averaging," *IEEE Transactions on Circuits and Systems –II: Analog and Digital Signal Processing*, vol. **42**, pp. 753–762, December 1995.

[64] R. Schreier and B. Zhang, "Noise-Shaped Multibit D/A Converter Employing Unit Elements," *IET Electronics Letters*, vol. **31**, pp. 1712–1713, September 1995.

[65] R. E. Radke, A. Eshraghi, and T. S. Fiez, "A 14-bit Current-Mode $\Sigma\Delta$ DAC Based Upon Rotated Data Weighted Averaging," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 1074–1084, August 2000.

[66] M. Vadipour, "Techniques for Preventing Tonal Behavior of Data Weighted Averaging Algorithm in $\Sigma-\Delta$ Modulators," *IEEE Transactions on Circuits and Systems –II: Analog and Digital Signal Processing*, vol. **47**, pp. 1137–1144, November 2000.

[67] I. Fujimori, L. Longo, A. Hairapetian, K. Seiyama, S. Kosic, J. Cao, and S.-L. Chan, "A 90-dB SNR 2.5-MHz Output-Rate ADC Using Cascaded Multibit Delta-Sigma Modulation at $8\times$ Oversampling Ratio," *IEEE Journal Solid-State Circuits*, vol. **35**, pp. 1820–1828, December 2000.

[68] A. A. Hamoui and K. Martin, "Linearity Enhancement of Multibit $\Delta\Sigma$ Modulators Using Pseudo Data-Weighted Averaging," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. III.285–288, 2002.

[69] A. Hairapetian, G. C. Temes, and Z. X. Zhang, "Multibit Sigma-Delta Modulator with Reduced Sensitivity to DAC Nonlinearity," *IET Electronics Letters*, vol. **27**, pp. 990–991, May 1991.

[70] B. P. Brandt and B. A. Wooley, "A 50-MHz Multibit Sigma-Delta Modulator for 12-b 2-MHz A/D Conversion," *IEEE Journal of Solid-State Circuits*, vol. **26**, pp. 1746–1756, December 1991.

[71] R. Reutemann, P. Balmelli, and Q. Huang, "A 33mW 14b 2.5MSample/s $\Sigma\Delta$ A/D Converter in 0.25 µm Digital CMOS," *IEEE ISSCC Digest of Technical Papers*, vol. **1**, p. 316, 2002.

[72] R. Schreier and M. Snelgrove, "Bandpass Sigma-Delta Modulation," *IET Electronics Letters*, vol. **25**, pp. 1560–1561, November 1989.

[73] P. H. Gailus, "Method and Arrangement for a Sigma Delta Converter for Bandpass Signals," US Patent 4,857,828, Aug. 1988, filed Jan. 28 1988, 1989.

[74] J. M. de la Rosa, B. Pérez-Verdú, R. del Río, F. Medeiro, and A. Rodríguez-Vázquez, "Bandpass Sigma-Delta A/D Converters: Fundamentals, Architectures and Circuits," Chapter 11 in *CMOS Telecom Data Converters* (A. Rodríguez-Vázquez, F. Medeiro, and E. Janssens, Editors), Kluwer Academic Publishers, 2003.

[75] J. Cherry, W. Snelgrove, and W. Gao, "On the Design of a Fourth-Order Continuous-Time LC Delta-Sigma Modulator for UHF A/D Conversion," *IEEE Transactions on Circuits and Systems –II: Analog and Digital Signal Processing*, vol. **47**, pp. 518–530, June 2000.

[76] B. Thandri and J. Silva-Martinez, "A 63 dB 75-mW Bandpass RF ADC at 950 MHz Using 3.8-GHz Clock in 0.25-µm SiGe BiCMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. **42**, pp. 269–279, February 2007.

[77] J. Ryckaert, J. Borremans, B. Verbruggen, L. Bos, C. Armiento, J. Craninckx, and G. van der Plas, "A 2.4 GHz Low-Power Sixth-Order RF Bandpass $\Delta\Sigma$ Converter in CMOS," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 2873–2880, November 2009.

[78] N. Beilleau, H. Aboushady, F. Montaudon, and A. Cathelin, "A 1.3 V 26 mW 3.2 GS/s Undersampled LC Bandpass $\Sigma\Delta$ ADC for a SDR ISM-band Receiver in 130 nm CMOS," *Proc. of the IEEE Radio Frequency Integrated Circuits Symp.*, 2009.

[79] S. A. Jantzi *et al*., "Quadrature Bandpass $\Delta\Sigma$ Modulation for Digital Radio," *IEEE Journal of Solid-State Circuits*, vol. **32**, pp. 1935–1950, December 1997.

[80] T. Paulus, S. S. Somayajula, T. A. Miller, B. Trotter, C. Kyong, and D. A. Kerth, "A CMOS IF Transceiver with Reduced Analog Complexity," *IEEE Journal of Solid-State Circuits*, vol. **33**, pp. 2154–2159, December 1998.

[81] L. Louis, J. Abcarius, and G. W. Roberts, "An Eight-Order Bandpass ΔΣ Modulator for A/D Conversion in Digital Radio," *IEEE Journal of Solid-State Circuits*, vol. **34**, pp. 423–431, April 1999.

[82] S. A. Jantzi and W. M. Snelgrove, "Bandpass Sigma-Delta Analog-to-Digital Conversion," *IEEE Transactions on Circuits and Systems*, vol. **38**, pp. 1406–1409, November 1991.

[83] S. A. Jantzi, W. M. Snelgrove, and P. F. Ferguson, "A Fourth-Order Bandpass Sigma-Delta Modulator," *IEEE Journal of Solid-State Circuits*, vol. **28**, pp. 282–291, March 1993.

[84] J. V. Engelen and R. van de Plassche, *BandPass Sigma-Delta Modulators: Stability Analysis, Performance and Design Aspects*, Kluwer Academic Publishers, 1999.

[85] R. F. Cormier, T. L. Sculley, and R. H. Bamberger, "A Fourth Order Bandpass Delta-Sigma Modulator with Digitally Programmable Pass-band Frequency," *Analog Integrated Circuits and Signal Processing*, vol. **12**, pp. 217–229, 1997.

[86] K. Yamamoto, A. C. Carusone, and F. P. Dawson, "A Delta-Sigma Modulator With a Widely Programmable Center Frequency and 82-dB Peak SNDR," *IEEE Journal of Solid-State Circuits*, vol. **43**, pp. 1772–1782, August 2008.

[87] O. Shoaei, "Continuous-Time Delta-Sigma A/D Converters for High Speed Applications," PhD Dissertation, Carleton University, 1995.

[88] J. Cherry and W. Snelgrove, "Excess Loop Delay in Continuous-Time Delta–Sigma Modulators," *IEEE Transactions on Circuits and Systems –II: Analog and Digital Signal Processing*, vol. **46**, pp. 376–389, April 1999.

[89] R. Tortosa, J. M. de la Rosa, F. V. Fernández, and A. Rodríguez-Vázquez, "A New High-Level Synthesis Methodology of Cascaded Continuous-Time ΣΔ Modulators," *IEEE Transactions on Circuits and Systems–II: Express Briefs*, vol. **53**, pp. 739–743, August 2006.

2

Circuits and Errors: Systematic Analysis and Practical Design Issues

As discussed in Chapter 1, ADCs based on ΣΔ modulation offer key advantages for their practical implementation in present-day CMOS processes compared with other data-conversion techniques. Unlike Nyquist-rate converters, which require high precision in their building blocks to achieve overall high accuracy, oversampling and quantization noise shaping allow to trade speed for accuracy. In this way, an operation that can be made relatively insensitive to imperfections on the analog circuit can be obtained at the cost of increased complexity and speed in the associated digital circuitry [1].

The principles of ΣΔ modulation were presented in the previous chapter and alternative ΣΔM topologies (single-loop and cascades) and implementation techniques (DT and CT) were presented. However, the achievable performance of different alternatives was mainly addressed taking only quantization error into account. Besides this error—which is inherent to any analog-to-digital conversion technique—only the effect of DAC errors was considered to compare the performance of single-bit and multibit ΣΔMs at the architectural level.

This chapter analyzes the main nonideal mechanisms affecting the performance of both SC and CT ΣΔMs. Although it is commonly accepted that ΣΔ ADCs are less sensitive to nonidealities in the analog circuitry than other conversion techniques, their impact will be larger the more demanding the ADC specifications. Therefore, the influence of these errors on the modulator performance must be carefully considered during early design phases. However, this chapter is not intended to be an exhaustive description of all nonideal effects, but a practical description of the main ones. The aim is to provide sufficient insight on the problem and to present analytical procedures that can be applied to other error mechanisms.

The first part of the chapter is devoted to circuit errors with large influence on the behavior of SC-ΣΔMs, such as integrator leakage, capacitor mismatch, settling errors, and kT/C noise. The second part of this chapter covers the dominant circuit errors in CT-ΣΔMs, especially clock jitter, excess loop delay, and time-constant errors. The main sources of distortion in both types of ΣΔMs are also discussed. System-level considerations,

CMOS Sigma-Delta Converters: Practical Design Guide, First Edition. José M. de la Rosa and Rocío del Río.

behavioral models, and closed-form expressions are obtained for the influence of each nonideality. From them, estimable guidelines for the design of ΣΔMs can be extracted. These are put into practice in a case study at the end of this chapter.

2.1 Nonidealities in Switched-Capacitor ΣΔ Modulators

There are a number of circuit nonidealities and nonlinearities that degrade the performance of the analog modulator blocks. The way in which these nonidealities affect the performance of ΣΔMs depends on many different factors, among others: the nature of the error itself, the influence of the specific circuit, its effect on the modulator noise transfer function, etc.

In the case of SC implementations, the main nonideal effects can be grouped as illustrated in Figure 2.1 according to the ΣΔM circuit they affect as:

- Amplifiers: output swing, finite gain, dynamic limitations, and circuit noise.
- Switches: on-resistance, thermal noise, charge injection, and clock feedthrough.
- Capacitors: mismatch and nonlinearity.
- Multibit ADCs and DACs: offset, gain error, and nonlinearity.
- Clock: jitter.

The influence of nonidealities on the performance of ΣΔMs strongly depends on the location of the corresponding noise source in the modulator. According to these criteria, the above errors can be classified into two main families:

- **Errors that modify the modulator NTF** such as the finite amplifier gain and gain-bandwidth product and capacitor mismatch. Their effect strongly depends on the modulator topology. For instance, cascade ΣΔMs are more sensitive to capacitor mismatch and finite amplifier gain than single-loop architectures. The same applies for low-pass ΣΔMs with optimized zeros and band-pass ΣΔMs with local feedback.

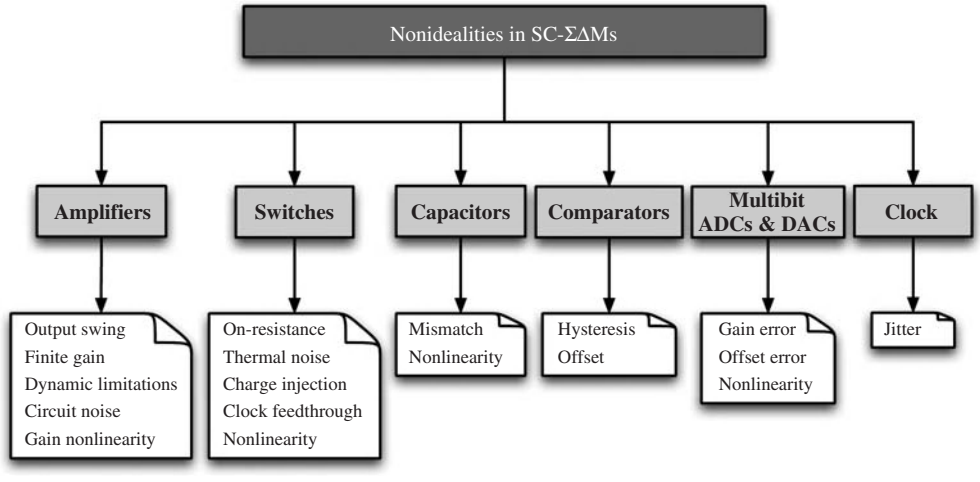


Figure 2.1 Main nonidealities affecting the performance of switched-capacitor ΣΔMs.

- **Errors that can be modeled as additive noise sources at the modulator input** and, hence, are not in-band attenuated by the noise shaping. Their effect is thus independent of the modulator topology. Among other, some errors belonging to this family are clock jitter, circuit noise, and distortion caused by circuit nonlinearities.

In the case of those nonideal effects affecting the modulator NTF, the procedure that is commonly used to analyze their impact on the modulator performance is [2, 3]:

- Obtaining an integrator equivalent circuit taking into account the nonideal effect under study.
- Analyzing the impact of the nonideality on the integrator transfer function ITF(z), such that ITF(z) $\rightarrow$ ITF$_\epsilon$(z), with ϵ being the error vector including all nonideal parameters involved in the integrator circuit equivalent obtained in the previous step.
- To compute the effect of ϵ on a $\Sigma\Delta$M, the integrator transfer functions are replaced with ITF$_\epsilon$(z) and a linear quantizer model is considered to obtain the nonideal NTF$_\epsilon$(z).
- The nonideal NTF is integrated within the signal band to obtain the degraded in-band noise power IBN(ϵ). Usually, some approximations are required to obtain closed-form expressions for IBN(ϵ) as a function of ϵ.

This procedure is applied to low-pass SC-$\Sigma\Delta$Ms throughout this section, although it can be easily generalized to band-pass SC-$\Sigma\Delta$Ms—working on the resonator transfer functions RTF(z)—and to CT-$\Sigma\Delta$Ms as well—working on S-domain.

For the sake of exemplification, single-loop $\Sigma\Delta$Ms with distributed feedback (see Section 1.4.2) and $\Sigma\Delta$ cascades are considered (see Section 1.5). For these modulator topologies, optimal coefficients for second-, third-, and fourth-order $\Sigma\Delta$Ms can be found in [4] for the single-bit case and in [5] for the multibit case.

Wherever behavioral simulation results are presented throughout this section corresponding to single-bit single-loop implementations, the following modulator coefficients have been used for the distributed feedback topologies (Figure 1.18):

- $(a_1, a_2) = (0.5, 0.5)$ for the second-order $\Sigma\Delta$M (SL2 for short)
- $(a_1, a_2, a_3) = (0.2, 0.5, 0.5)$ for the third-order $\Sigma\Delta$M (SL3 for short)
- $(a_1, a_2, a_3, a_4) = (0.2, 0.2, 0.5, 0.5)$ for the fourth-order $\Sigma\Delta$M (SL4 for short)

For single-bit cascade $\Sigma\Delta$Ms, their performance is often exemplified throughout this section with behavioral simulation results on a 2-1-1 $\Sigma\Delta$M (Figure 1.22), with the following coefficients:

- $(a_1, a_2, a_3, a_4, b_1, c_1, b_2, c_2) = (0.5, 0.5, 0.5, 0.5, 2, 0.5, 1, 1)$

2.2 Finite Amplifier Gain in SC-$\Sigma\Delta$Ms

In Chapter 1, the ideal performance of the different low-pass SC-$\Sigma\Delta$Ms presented was derived considering the ideal transfer function of an ideal SC FE integrator:

$$\mathrm{ITF}(z) = \frac{z^{-1}}{1 - z^{-1}} \tag{2.1}$$

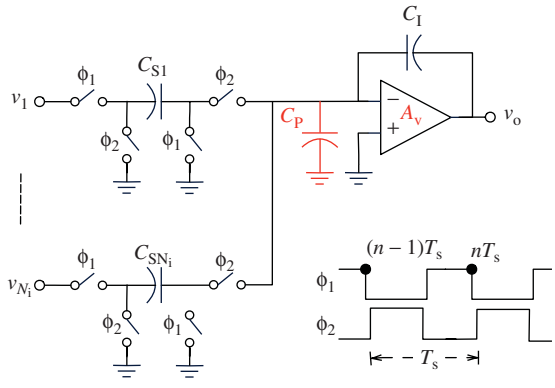


Figure 2.2 SC FE integrator with N_i input paths and finite amplifier gain A_v.

If the finite amplifier gain A_v and the parasitic capacitor C_P at the amplifier summing node is accounted for in the charge transfer of an SC FE integrator, as shown in Figure 2.2, its difference equation can be written as [3]:

$$\begin{aligned} v_o(nT_s) = & \frac{1+\left(1+\frac{C_P}{C_I}\right)\frac{1}{A_v}}{1+\left(1+\frac{C_P}{C_I}+\sum_{i=1}^{N_i}\frac{C_{Si}}{C_I}\right)\frac{1}{A_v}} v_o[(n-1)T_s] \\ & + \frac{\sum_{i=1}^{N_i}\frac{C_{Si}}{C_I} v_i[(n-1)T_s]}{1+\left(1+\frac{C_P}{C_I}+\sum_{i=1}^{N_i}\frac{C_{Si}}{C_I}\right)\frac{1}{A_v}} \end{aligned} \tag{2.2}$$

Transforming Equation 2.2 to the Z-domain and identifying terms as in the following expression,

$$v_o(z) = \mathrm{ITF}_{A_v}(z)\left[\sum_{i=1}^{N_i}\frac{C_{Si}}{C_I} v_i(z)\right] \tag{2.3}$$

the transfer function of the integrator when affected by finite amplifier gain—that is, the transfer function of a *leaky* integrator—yields:

$$\mathrm{ITF}_{A_v}(z) = \frac{1}{1+\left(1+\frac{C_P}{C_I}+\sum_{i=1}^{N_i}\frac{C_{Si}}{C_I}\right)\frac{1}{A_v}} \frac{z^{-1}}{1-z^{-1}\left[\frac{1+\left(1+\frac{C_P}{C_I}\right)\frac{1}{A_v}}{1+\left(1+\frac{C_P}{C_I}+\sum_{i=1}^{N_i}\frac{C_{Si}}{C_I}\right)\frac{1}{A_v}}\right]} \tag{2.4}$$

Therefore, when compared with the ideal case in Equation 2.1, amplifier finite gain introduces a gain error in the ITF and a shift of its pole from its ideal position at DC ($z = 1$). Neglecting the gain error and noting that $\sum_{i=1}^{N_i}\frac{C_{Si}}{C_I} = \sum_{i=1}^{N_i} g_i$, Equation 2.4 can be approximated to a more *handy* expression as:

$$\mathrm{ITF}_{A_v}(z) \approx \frac{z^{-1}}{1-z^{-1}\left(1-\frac{\sum_{i=1}^{N_i} g_i}{A_v}\right)} \tag{2.5}$$

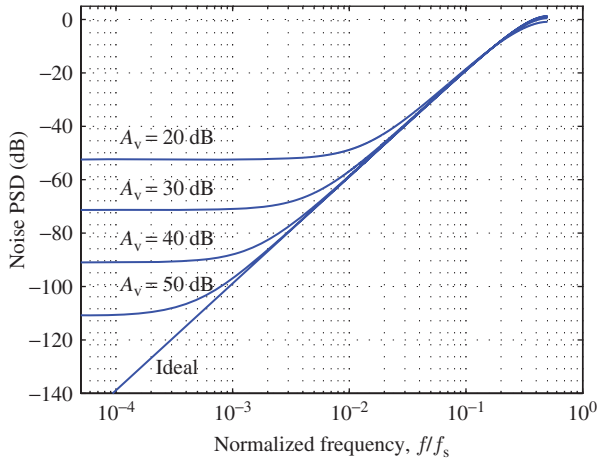


Figure 2.3 Degradation of the noise shaping of a second-order SC-ΣΔM with finite amplifier gain.

Let us exemplarily consider the effect of the modified ITF on the second-order DT-ΣΔM in Figure 1.15b. Assuming that $\sum_{i=1}^{N_i} g_i \sim 1$ for the two integrators in the modulator, the NTF affected by finite amplifier gain can be easily derived [3]. Given that the poles of the ITFs become the zeros of the NTF, both zeros of NTF move away from DC as the amplifier gain decreases. Figure 2.3 depicts the PSD of quantization error that is obtained for several values of A_v and illustrates the degradation of the modulator noise shaping. The following approximated expression can be derived for the in-band noise when integrator leakage is accounted for [3]:

$$\mathrm{IBN}_2(A_v) \approx \frac{\Delta^2}{12}\frac{1}{(k_q a_1 a_2)^2}\left(\frac{1}{{A_v}^4 \mathrm{OSR}} + \frac{2\pi^2}{3{A_v}^2 \mathrm{OSR}^3} + \frac{\pi^4}{5\mathrm{OSR}^5}\right) \tag{2.6}$$

Note that the DC gain of the amplifiers should be in the range of the oversampling ratio ($A_v \approx \mathrm{OSR}$) in order to keep every term in Equation 2.6 proportional to OSR^{-5} and retain the ideal noise shaping. For usual values of the oversampling ratio and the amplifier DC gain, this equation can be further simplified to:

$$\mathrm{IBN}_2(A_v) \approx \frac{\Delta^2}{12}\frac{1}{(k_q a_1 a_2)^2}\left(\frac{2\pi^2}{3{A_v}^2 \mathrm{OSR}^3} + \frac{\pi^4}{5\mathrm{OSR}^5}\right) \tag{2.7}$$

A similar procedure can be followed to derive the modified NTF of Lth-order loops affected by integrator leakage and thus calculate the increased in-band noise. For an Lth-order SC-ΣΔM with distributed feedback, the in-band noise can be obtained as follows [3],

$$\mathrm{IBN}_L(A_v) \approx \frac{\Delta^2}{12}\frac{1}{(k_q \prod_{i=1}^{L} a_i)^2}\left[\frac{1}{{A_v}^{2L}\mathrm{OSR}} + \sum_{i=1}^{L-1}\frac{L(L-1)\cdots(L-i+1)\pi^{2i}}{i!(2i+1){A_v}^{2(L-i)}\mathrm{OSR}^{(2i+1)}} + \frac{\pi^{2L}}{(2L+1)\mathrm{OSR}^{2L+1}}\right] \tag{2.8}$$

where again the amplifier DC gains must be in the range of the oversampling ($A_v \approx \text{OSR}$) to keep every term in Equation 2.8 proportional to $\text{OSR}^{-(2L+1)}$ and retain the ideal Lth-order noise shaping. The expression above can usually be further simplified to [3]:

$$\text{IBN}_L(A_v) \approx \frac{\Delta^2}{12}\frac{1}{(k_q\prod_{i=1}^{L}a_i)^2}\left[\frac{L\pi^{2(L-1)}}{(2L-1)A_v{}^2\text{OSR}^{(2L-1)}}+\frac{\pi^{2L}}{(2L+1)\text{OSR}^{2L+1}}\right] \quad (2.9)$$

Integrator leakages can be foreseen to have a stronger impact on cascade ΣΔMs, as the degradation of the ITF filtering leads to a modification of the cascaded loop filters (in the analog side) that is not compensated for by the cancelation logic (DCL in the digital side)—see Figure 1.21. This imbalance will cause quantization errors of all the cascaded stages to appear at the modulator output.

For the particular case of a 2-1-1 DT cascade as that illustrated in Figure 1.22, the IBN when integrator leakage is accounted for can be obtained as [3]:

$$\begin{aligned}\text{IBN}_{211}(A_v) \approx{}& \frac{\Delta_1^2}{12}\frac{1}{(k_q a_1 a_2)^2}\frac{4\pi^2}{3A_v^2\text{OSR}^3}+\frac{\Delta_2^2}{12}\frac{1}{c_1^2}\frac{\pi^4}{5A_v^2\text{OSR}^5}\\ &+\frac{\Delta_3^2}{12}\frac{1}{(c_1c_2)^2}\left(\frac{\pi^6}{7A_v^2\text{OSR}^7}+\frac{\pi^8}{9\text{OSR}^9}\right)\end{aligned} \quad (2.10)$$

Note that the amplifier DC gain required to retain the ideal noise shaping increases while moving from the back-end to the front-end stages of the cascade. Therefore, for the third-stage amplifier $A_v \approx \text{OSR}$ is sufficient, but for the second-stage amplifier $A_v \approx \text{OSR}^2$ and for the first-stage amplifiers $A_v \approx \text{OSR}^3$. Note also that, in case multibit quantization of B bits is employed in the last stage of the cascade, these requirements further increase by a factor $(2^B-1)^2$.

A detailed analysis of the effect of integrator leakage on the IBN of generic cascade SC-ΣΔMs can be found in [3], as well as of particular cascade configurations. For an Lth-order N-stage cascade DT-ΣΔM as that illustrated in Figure 1.21, the IBN considering integrator leakages yields [3],

$$\begin{aligned}\text{IBN}_{\text{casc}}(A_v) \approx{}& \frac{\Delta_1^2}{12}\frac{1}{(k_q\prod_{i=1}^{L_1}a_i)^2}\frac{\alpha_1\pi^{2(L_1-1)}}{(2L_1-1)A_v^2\text{OSR}^{2L_1-1}}\\ &+\frac{\Delta_2^2}{12}\frac{1}{c_1^2}\frac{\alpha_2\pi^{2(L_1+L_2-1)}}{[2(L_1+L_2)-1]A_v^2\text{OSR}^{2(L_1+L_2)-1}}\\ &+\cdots+\frac{\Delta_N^2}{12}\frac{1}{\prod_{i=1}^{N-1}c_i{}^2}\left[\frac{\alpha_N\pi^{2(L-1)}}{(2L-1)A_v^2\text{OSR}^{2L-1}}+\frac{\pi^{2L}}{(2L+1)\text{OSR}^{2L+1}}\right]\end{aligned} \quad (2.11)$$

where L_i corresponds to the order of the ith stage in the cascade and the value of coefficients α_i depends on L_i.

Figure 2.4 shows the effect of the finite DC gain of amplifiers on single-loop and cascade SC-ΣΔMs, exemplarily illustrated for second-, third-, and fourth-order single-loops and for a 2-1-1 cascade. In all cases, the in-band noise is computed from the modulator NTF using the nonapproximated ITF in Equation 2.4, as well as from the approximated closed-form expressions in Equations 2.9 and 2.10. Note that both results are in good

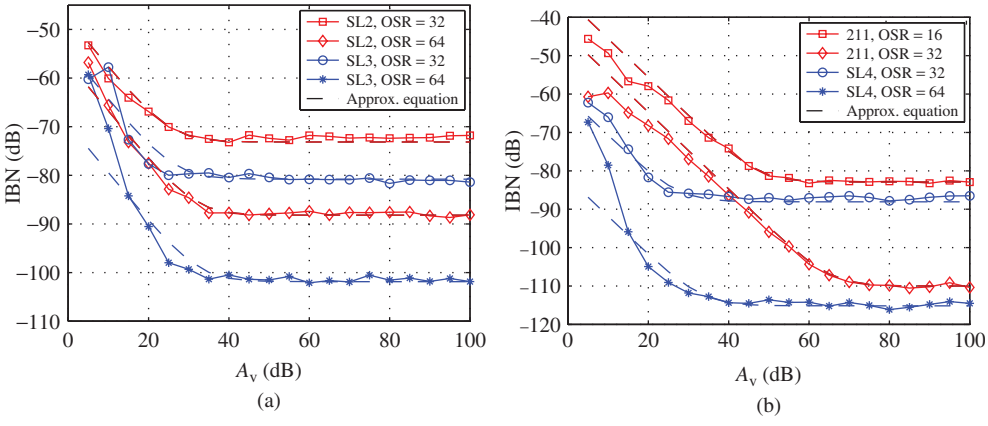


Figure 2.4 Influence of finite amplifier gain on the in-band noise of SC-ΣΔMs: (a) second- and third-order loops and (b) 2-1-1 cascade and fourth-order loop. Approximated results have been obtained from Equations 2.9 and 2.10.

accordance. Note also that the larger sensitivity of cascade ΣΔMs to integrator leakages is evident from Figure 2.4b.

2.3 Capacitor Mismatch in SC-ΣΔMs

As illustrated in Figure 1.16, in SC-ΣΔMs integrator gain coefficients g_i are implemented as capacitor ratios C_{Si}/C_I and the implemented values will thus deviate from the nominal ones due to variations in technological process parameters. For the case of a particular gain coefficient g that is implemented as the ratio of m to n unit capacitors C_u, the actual implemented coefficient g_ϵ will exhibit an error ϵ_g that can be estimated as [3],

$$\left.\begin{aligned} g_\epsilon &= g(1 \pm \epsilon_g) \\ g &= \frac{C_S}{C_I} = \frac{mC_u}{nC_u} \end{aligned}\right\} \quad \Rightarrow \quad \epsilon_g = 3\frac{\sigma_g}{g} = 3\sqrt{\frac{1}{m} + \frac{1}{n}}\frac{\sigma_{C_u}}{C_u} \;\leq\; 3\sqrt{2}\sigma_C \qquad (2.12)$$

where the integrator gain error is estimated in the worst case as three times its relative standard deviation, which can itself be related to the relative standard deviation of the unit capacitor used—or σ_C for short. Note that the estimation for ϵ_g in Equation 2.12 should be divided by $\sqrt{2}$ in fully-differential SC integrators.

Nowadays, SC-ΣΔMs are mostly implemented in mixed-mode technological processes that include precise capacitor primitives, such as MiM or MoM capacitors, with a mismatch typically lower than 0.1%. This means that integrator gain errors in SC-ΣΔMs will normally be lower than 0.3%—or even less in case a large number of unit capacitors and common-centroid layout techniques are used for implementing the coefficients.

These small deviations of the integrator coefficients due to capacitor mismatch can be foreseen to have little impact on the in-band noise of single-loop ΣΔMs, as the filtering provided by the integrators remains unchanged. Indeed, if integrator gain errors are accounted for, the IBN of an Lth-order SC-ΣΔM with distributed feedback can be

estimated as,

$$\mathrm{IBN_L}(\epsilon_g) \approx \frac{\Delta^2}{12}\frac{1}{(k_q\prod_{i=1}^{L}a_i)^2}\frac{\pi^{2L}}{(2L+1)\mathrm{OSR}^{2L+1}}\prod_{i=1}^{L}(1+\epsilon_{gi})^2 \tag{2.13}$$

where ϵ_{gi} refers to the gain error of the ith integrator, which can be estimated in the worst case as $3\sigma_C$. Note from Equation 2.13 that, for the IBN of second-order ΣΔM to increase in 3 dB, integrator gain errors should be as large as 20%—too large indeed to be considered as an actual mismatch!

Conversely, capacitor mismatch has a strong impact on cascade ΣΔMs, as the deviation of the integrator gains is not compensated for by the digital coefficients of the DCL. Therefore, quantization errors of the cascaded stages will leak to the modulator output with low-order shapings, considerably increasing the modulator IBN.

For the general case of an Lth-order N-stage cascade SC-ΣΔM, the IBN if integrator gain errors are taken into account can be approximated to [3],

$$\begin{aligned}\mathrm{IBN_{casc}}(\epsilon_g) \approx{}& \frac{\Delta_1^2}{12}\frac{1}{(k_q\prod_{i=1}^{L_1}a_i)^2}\frac{\pi^{2L_1}}{(2L_1+1)\mathrm{OSR}^{2L_1+1}}\left(\sum_{i=1}^{L_1}\epsilon_{gi}\right)^2\\ &+\frac{\Delta_2^2}{12}\frac{1}{c_1^2}\frac{\pi^{2(L_1+L_2)}}{[2(L_1+L_2)+1]\mathrm{OSR}^{2(L_1+L_2)+1}}\left(\sum_{i=L_1+1}^{L_1+L_2}\epsilon_{gi}\right)^2\\ &+\cdots+\frac{\Delta_N^2}{12}\frac{1}{\prod_{i=1}^{N-1}c_i^2}\frac{\pi^{2L}}{(2L+1)\mathrm{OSR}^{2L+1}}\left(1+\sum_{i=L-L_N}^{L}\epsilon_{gi}\right)^2\end{aligned} \tag{2.14}$$

whereas, for the particular case of a 2-1-1 cascade, the former equation yields [3]:

$$\begin{aligned}\mathrm{IBN_{211}}(\epsilon_g) \approx{}& \frac{\Delta_1^2}{12}\frac{1}{(k_q a_1 a_2)^2}\frac{\pi^4}{5\mathrm{OSR}^5}(\epsilon_{g1}+\epsilon_{g2})^2+\frac{\Delta_2^2}{12}\frac{1}{c_1^2}\frac{\pi^6}{7\mathrm{OSR}^7}\epsilon_{g3}^2\\ &+\frac{\Delta_3^2}{12}\frac{1}{(c_1c_2)^2}\frac{\pi^8}{9\mathrm{OSR}^9}(1+\epsilon_{g4})^2\end{aligned} \tag{2.15}$$

Note that similarly to what happened in Section 2.2 for the case of amplifier DC gain—the requirements on the integrator gain error for retaining the ideal noise shaping get more stringent as we move from the back-end to the front-end stages. Thus, for the integrator in the second stage of the cascade $\epsilon_{g3} \approx \mathrm{OSR}^{-1}$ is sufficient, but for the first-stage integrators $\epsilon_{g1}, \epsilon_{g2} \approx \mathrm{OSR}^{-2}$. Note also that, in case multibit quantization of B bits is employed in the last stage of the cascade, these requirements again increase by a factor $(2^B-1)^2$.

Figure 2.5 illustrates this increasing impact of capacitor mismatch with OSR and with the stage of the cascade being considered. Note that results displayed correspond to worst-case estimations of the IBN based on Equation 2.15 with $\epsilon_{gi} = 3\sigma_C$. More accurate estimations would require Monte Carlo simulation of the modulator behavioral model considering the particular implementation of each integrator gain coefficient in terms of unit capacitors.

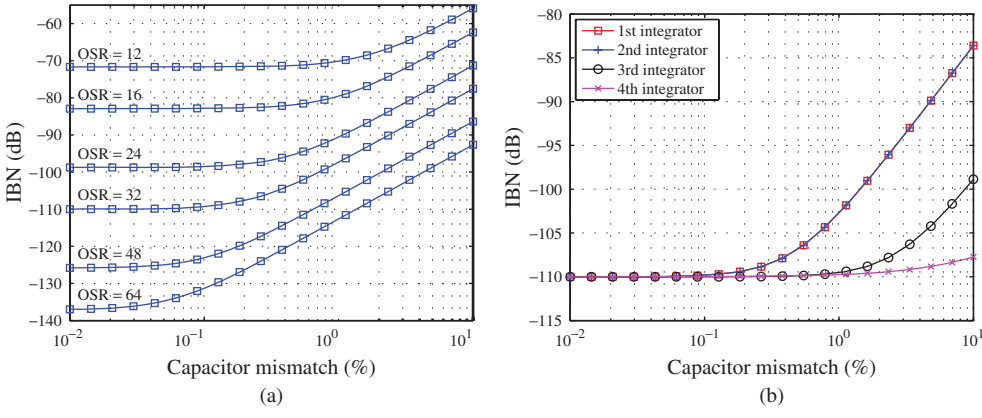


Figure 2.5 Influence of capacitor mismatch on the in-band noise of a 2-1-1 SC-ΣΔM: (a) considering the same mismatch error in all integrators and (b) individual impact of the mismatch error in each integrator for OSR = 32. Worst-case estimations of IBN considering $\epsilon_{gi} = 3\sigma_C$ in Equation 2.15.

2.4 Integrator Settling Error in SC-ΣΔMs

Speed limitations in SC integrators due to the limited dynamic response of amplifiers cause errors in the charge transfer. The impact of the resulting error in the integrator output voltage settling error on the modulator performance will be higher, the higher the sampling frequency. As the clock frequency increases in SC-ΣΔMs to cope with larger conversion bandwidths, integrator settling error becomes one of the bottlenecks for their practical implementation. On the one hand, the time slot for the integrator operation gets reduced; on the other hand, the amplifier dynamic requirements must be minimized to optimize the modulator power consumption. Therefore, an adequate understanding of the mechanisms degrading the settling of SC integrators and an accurate quantification of the generated errors become mandatory to obtain efficient ΣΔ designs.

2.4.1 Behavioral Model for the Integrator Settling

The behavioral model of the transient response for SC FE integrators included in this section is based on the analysis presented in [6]. The model includes the effect of the amplifier dynamic limitations, such as finite gain-bandwidth product (GB) and slew rate (SR), on the charge transfer during both the integration and sampling phases. Also, parasitic capacitors associated with amplifiers and switches, as well as the capacitor load at the integrator output—which changes from integration to sampling—are taken into account. To accurately describe the dynamic performance and determine the integrator output voltage, the equivalent circuit shown in Figure 2.6 is solved in the behavioral model. In the circuit scheme, the SC integrator under study is considered to have N_i input branches and another SC integrator acting as a load—that is, the N_o input branches of the latter are connected to the output of the former. On the other hand, the amplifier is modeled

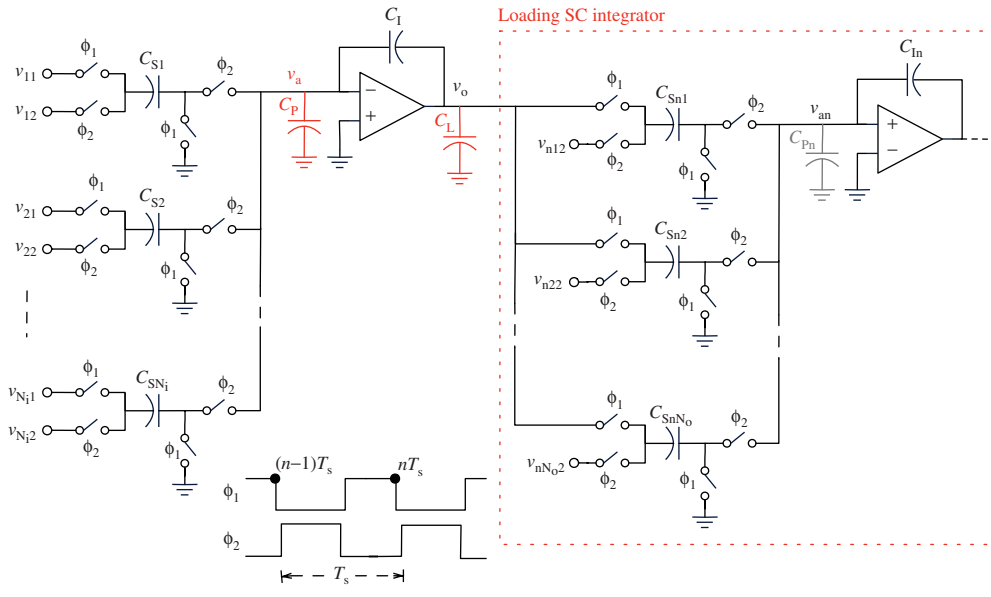


Figure 2.6 SC FE integrator under consideration followed by a loading SC integrator.

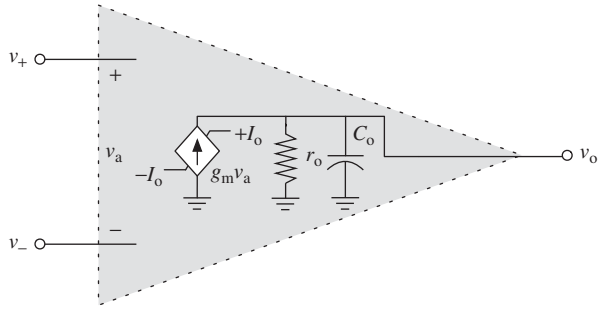


Figure 2.7 Amplifier single-pole model with limited output current.

as shown in Figure 2.7, with a single-pole dynamic (to account for the finite bandwidth) and a nonlinear characteristic with maximum output current I_o (to account for the limited SR).[1]

The analysis of this model for the incomplete settling error begins with the computation of the equivalent capacitive load at the amplifier output node during both the sampling

[1] Note that C_o in Figure 2.7 is merged with C_L in Figure 2.6. Note also that r_o is included in the amplifier model in Figure 2.7 for completeness, but its effect is actually neglected in the analysis of the integrator dynamics in Figure 2.6 under the assumption $g_o \ll g_m$.

(ϕ_1) and integration (ϕ_2) phases, respectively given by,

$$\begin{aligned} C_{\text{eq},\phi_1} &= C_\text{P} + \left(C_\text{L} + \sum_{i=1}^{N_\text{o}} C_{\text{Sni}}\right)\left(1 + \frac{C_\text{P}}{C_\text{I}}\right) \\ C_{\text{eq},\phi_2} &= C_\text{P} + \sum_{i=1}^{N_\text{i}} C_{\text{Si}} + C_\text{L}\left(1 + \frac{C_\text{P} + \sum_{i=1}^{N_\text{i}} C_{\text{Si}}}{C_\text{I}}\right) \end{aligned} \tag{2.16}$$

where C_{Si} represents the sampling capacitor of the ith SC input branch of the integrator under consideration, C_{Sni} is the sampling capacitor of the ith SC input branch of the load integrator, C_P is the parasitic capacitor at the summation node of the input SC branches, and C_L is the amplifier load capacitor.

The settling model is analyzed during a complete clock cycle (during both clock phases) considering the different possibilities for the amplifier dynamic operation—that is, linearly or in slew—and keeping track of the voltage at both the integrator output v_o and the amplifier summation node v_a. Therefore, the error in the integrator output voltage at the end of one sampling-integration process can be accurately obtained.

Let $v_\text{a}[(n-1/2)T_\text{s}]$ and $v_\text{o}[(n-1/2)T_\text{s}]$ be the respective amplifier input and output voltages at the end of a preceding integration phase, which will serve as initial conditions to derive of the integrator evolution during a complete clock cycle. The voltage at the amplifier summation node at the end of the next sampling phase—that is, at $t = nT_\text{s}$—, can be accurately obtained as [6],

$$v_\text{a}(nT_\text{s}) = \begin{cases} v_{\text{a0},\phi_1} \text{e}^{-\frac{g_\text{m}}{C_{\text{eq},\phi_1}}\frac{T_\text{s}}{2}} & |v_{\text{a0},\phi_1}| \le \frac{I_\text{o}}{g_\text{m}} \\ \frac{I_\text{o}}{g_\text{m}}\text{sgn}(v_{\text{a0},\phi_1})\text{e}^{-\frac{g_\text{m}}{C_{\text{eq},\phi_1}}\left(\frac{T_\text{s}}{2} - t_{\text{o},\phi_1}\right)} & |v_{\text{a0},\phi_1}| > \frac{I_\text{o}}{g_\text{m}}, t_{\text{o},\phi_1} \le \frac{T_\text{s}}{2} \\ v_{\text{a0},\phi_1} - \frac{I_\text{o}}{C_{\text{eq},\phi_1}}\text{sgn}(v_{\text{a0},\phi_1})\frac{T_\text{s}}{2} & |v_{\text{a0},\phi_1}| > \frac{I_\text{o}}{g_\text{m}}, t_{\text{o},\phi_1} > \frac{T_\text{s}}{2} \end{cases} \tag{2.17}$$

where t_{o,ϕ_1} is the duration of the SR-limited integrator settling (relative to $T_\text{s}/2$) given by,

$$t_{\text{o},\phi_1} = \frac{C_{\text{eq},\phi_1}}{g_\text{m}}\left(\frac{g_\text{m}|v_{\text{a0},\phi_1}|}{I_\text{o}} - 1\right) \tag{2.18}$$

sgn() is the sign function, and v_{a0,ϕ_1} represents the value of v_a at the beginning of the sampling phase, which can be computed as,

$$v_{\text{a0},\phi_1} = v_\text{a}[(n-1/2)T_\text{s}] - \sum_{i=1}^{N_\text{o}} \frac{C_{\text{Sni}}}{C_{\text{eq},\phi_1}}\{v_\text{o}[(n-1/2)T_\text{s}] - v_{C_{\text{Sni}}}[(n-1/2)T_\text{s}]\} \tag{2.19}$$

where $v_{C_{\text{Sni}}}$ is the voltage across capacitor C_{Sni}.[2]

The integrator output voltage at the end of the sampling phase can be obtained as,

$$v_\text{o}(nT_\text{s}) = v_\text{o}[(n-1/2)T_\text{s}] + \left(1 + \frac{C_\text{P}}{C_\text{I}}\right)\{v_\text{a}(nT_\text{s}) - v_\text{a}[(n-1/2)T_\text{s}]\} \tag{2.20}$$

as opposed to the ideal situation in which $v_\text{o}(nT_\text{s}) = v_\text{o}[(n-1/2)T_\text{s}]$.

[2] Note that the behavioral model requires keeping track of the summation node voltage of the loading integrator v_{an} also, as $v_{C_{\text{Sni}}}[(n-1/2)T_\text{s}] = v_{\text{ni2}}[(n-1/2)T_\text{s}] - v_{\text{an}}[(n-1/2)T_\text{s}]$.

Note from Equations 2.17 and 2.20 that, for the integrator model in Figure 2.6, the amplifier gain-bandwidth product and output SR during sampling are obtained as:

$$\begin{aligned} \mathrm{GB}_{\phi_1}(\mathrm{rad/s}) &= \frac{g_\mathrm{m}}{C_{\mathrm{eq},\phi_1}} \\ \mathrm{SR}_{\phi_1}(\mathrm{V/s}) &= \left(1+\frac{C_\mathrm{P}}{C_\mathrm{I}}\right)\frac{I_\mathrm{o}}{C_{\mathrm{eq},\phi_1}} \end{aligned} \tag{2.21}$$

During the integration phase, the incomplete settling model is evaluated proceeding in a similar way as done for the sampling phase. Thus, at the end of the subsequent integration phase—that is, at $t = (n+1/2)T_\mathrm{s}$—, the value of v_a is given by [3],

$$v_\mathrm{a}[(n+1/2)T_\mathrm{s}] = \begin{cases} v_{\mathrm{a0},\phi_2}\mathrm{e}^{-\frac{g_\mathrm{m}}{C_{\mathrm{eq},\phi_2}}\frac{T_\mathrm{s}}{2}} & |v_{\mathrm{a0},\phi_2}| \le \frac{I_\mathrm{o}}{g_\mathrm{m}} \\ \frac{I_\mathrm{o}}{g_\mathrm{m}}\mathrm{sgn}(v_{\mathrm{a0},\phi_2})\mathrm{e}^{-\frac{g_\mathrm{m}}{C_{\mathrm{eq},\phi_2}}\left(\frac{T_\mathrm{s}}{2}-t_{\mathrm{o},\phi_2}\right)} & |v_{\mathrm{a0},\phi_2}| > \frac{I_\mathrm{o}}{g_\mathrm{m}},\ t_{\mathrm{o},\phi_2} \le \frac{T_\mathrm{s}}{2} \\ v_{\mathrm{a0},\phi_2} - \frac{I_\mathrm{o}}{C_{\mathrm{eq},\phi_2}}\mathrm{sgn}(v_{\mathrm{a0},\phi_2})\frac{T_\mathrm{s}}{2} & |v_{\mathrm{a0},\phi_2}| > \frac{I_\mathrm{o}}{g_\mathrm{m}},\ t_{\mathrm{o},\phi_2} > \frac{T_\mathrm{s}}{2} \end{cases} \tag{2.22}$$

where t_{o,ϕ_2} is the duration of the SR-limited integrator settling (relative to $T_\mathrm{s}/2$), given by

$$t_{\mathrm{o},\phi_2} = \frac{C_{\mathrm{eq},\phi_2}}{g_\mathrm{m}}\left(\frac{g_\mathrm{m}|v_{\mathrm{a0},\phi_2}|}{I_\mathrm{o}} - 1\right) \tag{2.23}$$

and v_{a0,ϕ_2} represents the value of v_a at the beginning of the integration phase. The latter can be computed as

$$\begin{aligned} v_{\mathrm{a0},\phi_2} = {} & \frac{1}{C_{\mathrm{eq},\phi_2}}\left(1+\frac{C_\mathrm{L}}{C_\mathrm{I}}\right)\sum_{i=1}^{N_\mathrm{i}} C_{\mathrm{Si}}\{v_{\mathrm{i2}}(nT_\mathrm{s}) - v_{\mathrm{i1}}[(n-1/2)T_\mathrm{s}]\} \\ & + \frac{C'}{C_{\mathrm{eq},\phi_2}}v_\mathrm{a}(nT_\mathrm{s}) \end{aligned} \tag{2.24}$$

where v_{i1}, v_{i2} are the voltages connected to the input of the ith SC branch during ϕ_1, ϕ_2, respectively, and C' represents:

$$C' = C_\mathrm{P} + C_\mathrm{L}\left(1+\frac{C_\mathrm{P}}{C_\mathrm{I}}\right) \tag{2.25}$$

The integrator output voltage at the end of the integration phase can be obtained as,

$$\begin{aligned} v_\mathrm{o}[(n+1/2)T_\mathrm{s}] = {} & v_\mathrm{o}(nT_\mathrm{s}) + \sum_{i=1}^{N_\mathrm{i}}\frac{C_{\mathrm{Si}}}{C_\mathrm{I}}\{v_{\mathrm{i1}}[(n-1/2)T_\mathrm{s}] - v_{\mathrm{i2}}(nT_\mathrm{s})\} \\ & - \left(1+\frac{C_\mathrm{P}}{C_\mathrm{I}}\right)v_\mathrm{a}(nT_\mathrm{s}) \\ & + \left(1+\frac{C_\mathrm{P}+\sum_{i=1}^{N_\mathrm{i}} C_{\mathrm{Si}}}{C_\mathrm{I}}\right)v_\mathrm{a}[(n+1/2)T_\mathrm{s}] \end{aligned} \tag{2.26}$$

as opposed to the ideal integration process with no dynamic limitations, in which the last two terms in Equation 2.26 are null.

The amplifier gain-bandwidth product and output SR during this phase can be obtained similarly as for the sampling phase to be:

$$\begin{aligned} \mathrm{GB}_{\phi_2}(\mathrm{rad/s}) &= \frac{g_\mathrm{m}}{C_{\mathrm{eq},\phi_2}} \\ \mathrm{SR}_{\phi_2}(\mathrm{V/s}) &= \left(1 + \frac{C_\mathrm{P} + \sum_{i=1}^{N_\mathrm{i}} C_{\mathrm{Si}}}{C_\mathrm{I}}\right) \frac{I_\mathrm{o}}{C_{\mathrm{eq},\phi_2}} \end{aligned} \tag{2.27}$$

Figure 2.8 illustrates how the equations above can be concatenated to accurately keep track of the summation and output voltages of an SC integrator over the clock periods. They can be easily incorporated into CAD tools for the behavioral simulation of SC-ΣΔMs—or SC circuits in general. Moreover, the previous model can be easily

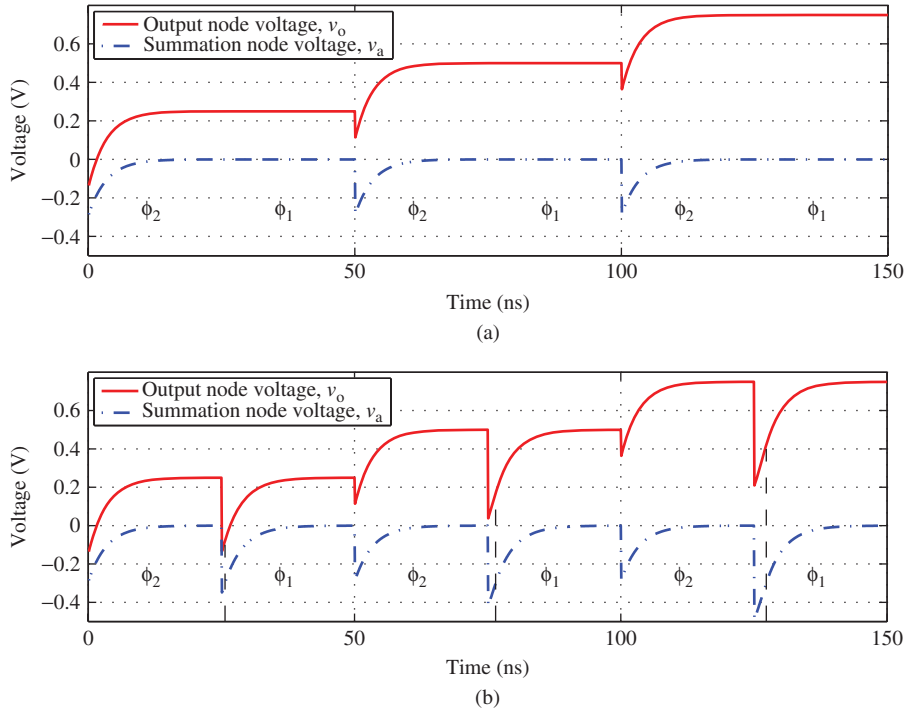


Figure 2.8 Illustration of the influence of switching load conditions on the transient response of an SC integrator: (a) loading SC branches are not considered and (b) one loading SC branch with a 0.5-pF capacitor is considered. (Vertical dashed lines indicate time positions $t_{\mathrm{o},\phi 1}$ where the integrator ends an SR-limited response and starts evolving linearly). Parameters used are (Figures 2.6 and 2.7): $v_{11} = 0$, $v_{12} = -1$V, $C_{\mathrm{S1}} = 0.25$ pF, $C_\mathrm{I} = 1$ pF, $C_\mathrm{P} = 0.1$ pF, $C_\mathrm{L} = 1$ pF, $g_\mathrm{m} = 0.5$ mA/V, and $I_\mathrm{o} = 0.15$ mA for the SC integrator under consideration, $v_{\mathrm{n}12} = -1$V and $C_{\mathrm{Sn1}} = 0.5$ pF for the loading SC integrator, and $T_\mathrm{s} = 50$ ns.

extrapolated to other operating conditions: integration and sampling phases with different durations, different switching loading conditions at the integrator output, to include the parasitic capacitance of the switches, etc.

2.4.2 *Linear Effect of Finite Amplifier Gain-Bandwidth Product*

The model for the transient response of SC integrators described earlier can be easily incorporated into behavioral simulators for SC-ΣΔMs to accurately quantify the influence of settling errors on the modulator performance—in terms of both the increased in-band noise and the generated distortion. Besides this behavioral model, it is often useful to work at the early design stages (high-level design) with closed-form expressions which, although being coarse approximations of the behavioral model, can help to gain insight of the influence of settling parameters on different modulator topologies. For this purpose, a linear transient response will be assumed for SC integrators in this section, as if settling error was determined only by the finite amplifier GB with no limitation on the SR.

With these considerations in mind, the finite difference equation of an SC FE integrator can be obtained from Equations 2.17, 2.20, 2.22, and 2.26 to be,

$$v_o[(n+1/2)T_s] \approx v_o[(n-1)T_s] + \frac{C_S}{C_I}(1-\epsilon_{st})\{v_1[(n-1)T_s] - v_2[(n-1/2)T_s]\} \quad (2.28)$$

where only one input branch is considered for simplicity. The settling error associated with the linearly limited transient response is represented by ϵ_{st}, which thus contains terms in $e^{-GB_{\phi_1}T_s/2}$ and in $e^{-GB_{\phi_2}T_s/2}$—with GB in rad s^{-1}. If settling errors associated with integration dominate on the overall defective settling over those originated during sampling, the linear settling error can be simply reduced to:

$$\epsilon_{st} \approx e^{-GB_{\phi_2}\frac{T_s}{2}} = e^{-\pi\frac{GB_{\phi_2}(Hz)}{f_s}} \quad (2.29)$$

Transforming Equation 2.28 to the Z-domain, the integrator output results in,

$$v_o(z) \approx \frac{C_S}{C_I}(1-\epsilon_{st})\frac{z^{-1}v_1(z) - z^{-1/2}v_2(z)}{1-z^{-1}} \quad (2.30)$$

so that, under the assumptions earlier, settling error translates into a gain error in the ideal ITF, whose effect on the IBN of SC-ΣΔMs can be computed in a similar way as formerly done for capacitor mismatch in Section 2.3. Therefore, Equations 2.13–2.14 still hold for quantifying the effect of linear defective settling to first order, just by replacing ϵ_g with ϵ_{st}.

Figure 2.9 illustrates the effect of the finite amplifier GB on single-loop and cascade SC-ΣΔMs. The in-band noise of second- and third-order single loops and of a 2-1-1 cascade is computed considering both the behavioral model for the integrator settling in Section 2.4.1 and the approximated closed-form expressions of the generated gain error. Large amplifier output currents have been used in behavioral simulations to make the influence of SR limitation negligible and thus consider linear errors only. Note that, as expected, cascades ΣΔMs are more sensitive to GB limitations that single loops. Usually, an amplifier GB of $1–2\,f_s$ is sufficient for single-loop modulators to achieve

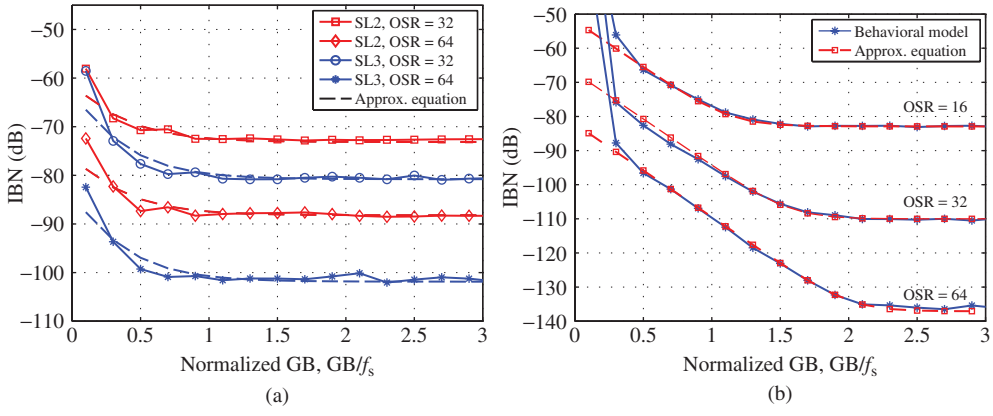


Figure 2.9 Simulation results for the influence of amplifier GB on the in-band noise of SC-ΣΔMs: (a) second- and third-order loops and (b) 2-1-1 cascade. Approximated results have been obtained from Equations 2.13 and 2.15 with $\epsilon_{st} = e^{-\pi GB(Hz)/f_s}$.

full performance, whereas the requirement increases to $3-10\,f_s$ for cascade ΣΔMs as the oversampling ratio increases.

2.4.3 *Nonlinear Effect of Finite Amplifier Slew Rate*

Contrary to errors arising from finite amplifier GB, finite amplifier SR caused by limited output current capability has a purely nonlinear effect on the performance of ΣΔMs, generating distortion and an increase in the noise floor.

For the case of single-loop SC-ΣΔMs, SR-limited integrator dynamics basically translate into distortion. Figure 2.10 illustrates the impact of amplifier SR on a single-bit

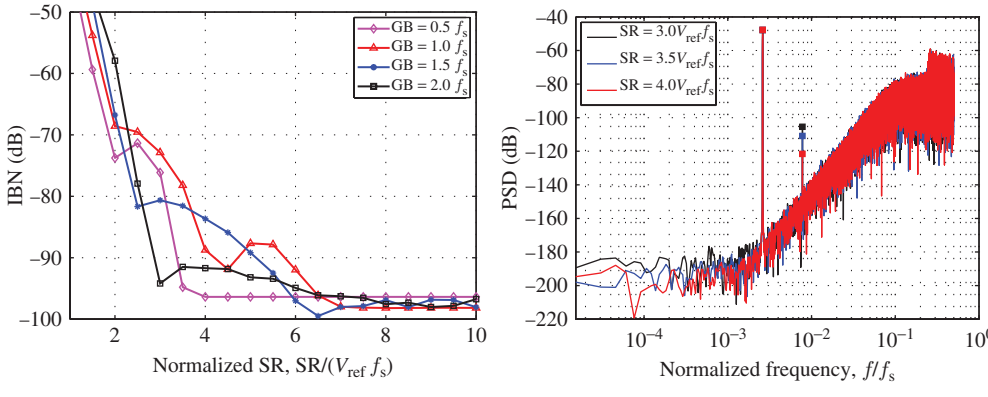


Figure 2.10 Simulation results for a third-order SC-ΣΔM with OSR = 64 under the influence of finite amplifier slew rate: (a) effect on the in-band noise and (b) effect on the output spectrum for GB = $1\,f_s$. Input signal with $P_{in} = -6$ dBFS and $f_{in} = B_w/3$. Generated distortion is included in the IBN computation.

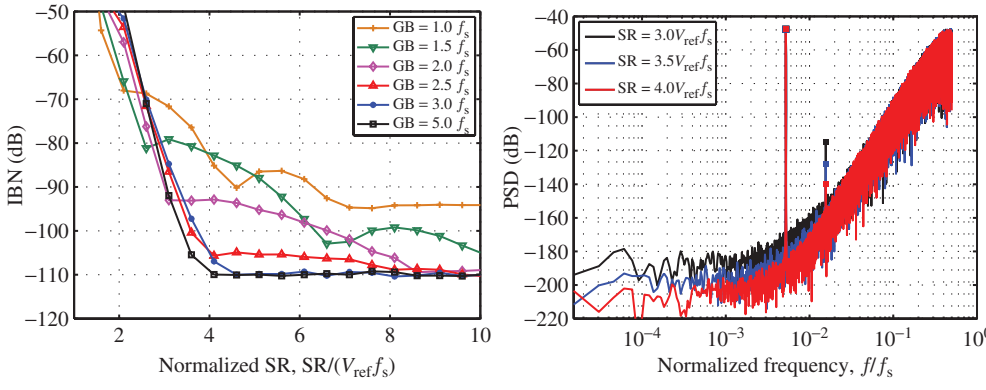


Figure 2.11 Simulation results for a 2-1-1 SC-ΣΔM with OSR = 32 under the influence of finite amplifier slew rate: (a) effect on the in-band noise and (b) effect on the output spectrum for GB $= 3f_s$. Input signal with $P_{in} = -6$ dBFS and $f_{in} = B_w/3$. Generated distortion is included in the IBN computation.

third-order ΣΔM operating with an oversampling ratio of 64. An input tone with −6 dBFS ($0.5V_{ref}$ amplitude) and frequency equal to $B_w/3$ is applied to the modulator in behavioral simulations. Note from the presented results that, depending on the amplifier GB, an SR of $4-8V_{ref}f_s$ is enough to reduce the power of generated distortion to a level that does not affect IBN.

For the case of cascade SC-ΣΔMs, finite amplifier SR generates distortion as well as an increase in the noise floor due to noise leakages, as shown in Figure 2.11. For that reason, SR requirements are larger than for single loops and usually range from 4 to $10V_{ref}f_s$ depending on the amplifier GB—the larger the GB, the lower the required SR.

Finally, note that the SR-limited integrator dynamic is a nonlinear signal-dependent phenomenon whose occurrence frequency during the modulator operation is directly determined by the signal level at the integrators inputs. Therefore, the ultimate way to reduce SR requirements on an SC-ΣΔM is to resort to multibit internal quantization.

2.4.4 *Effect of Finite Switch On-Resistance*

Switches in the SC branches of ΣΔMs are implemented with MOSFETs—using either single nMOS or pMOS transistors, or CMOS transmission gates—that operate in the triode region when on and thus exhibit in practice a nonzero on-resistance.

If the on-resistance of the switches is the only nonideality accounted for in the operation of an SC integrator, it clearly leads to an incomplete charge transfer due to the *RC* time constant that is created in the SC branch. Considering for instance the scheme in Figure 2.12, the integrator output voltage can be obtained as [3],

$$v_o(z) = \frac{C_S}{C_I}(1-\epsilon_{R_{on,\phi_2}})\left[\frac{(1-\epsilon_{R_{on,\phi_1}})z^{-1}v_1(z) - z^{-1/2}v_2(z)}{1-z^{-1}}\right] \tag{2.31}$$

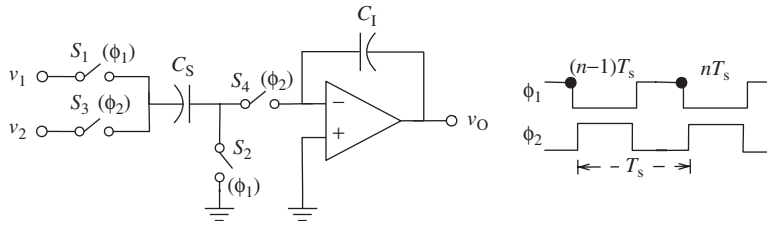


Figure 2.12 SC FE integrator with a single input branch.

where $\epsilon_{R_{on,\phi_1}}$ represents the charging error in C_S during ϕ_1 related to the on-resistance of switches S_1 and S_2 and $\epsilon_{R_{on,\phi_2}}$ represents the error in the charge transfer from C_S to C_I during ϕ_2 related to the on-resistance of switches S_3 and S_4. If R_{on} is the on-resistance of a single switch, assuming that all switches are of the same size and that both clock phases have the same duration leads to:

$$\epsilon_{R_{on,\phi_1}} = \epsilon_{R_{on,\phi_2}} = e^{-\frac{1}{2R_{on}C_S}\frac{T_S}{2}} = e^{-\pi\frac{f_{R_{on}}(\text{Hz})}{f_S}} \tag{2.32}$$

Therefore, charge transfer error due to the on-resistance of the switches translates into a gain error in the ideal ITF, whose effect on the IBN of SC-ΣΔMs can be computed in a similar way as formerly done in Sections 2.3 or 2.4.2. Therefore, finite switch on-resistance will have considerably lower impact on ΣΔ single loops than on cascades.

Besides the former discussion on how the switch on-resistance, as a stand-alone nonideality, affects an SC integrator, its effect is better accounted for in practice in combination with the limited amplifier dynamics. Figure 2.13 shows electrical simulation results to illustrate the influence of R_{on} on the transient response of the same SC integrator formerly considered in Figure 2.8. Only one clock cycle is shown here to gain in visibility of the R_{on} effect. Note that the linear amplifier response is slowed down as the on-resistance increases, affecting the integrator settling during both the sampling and the integration

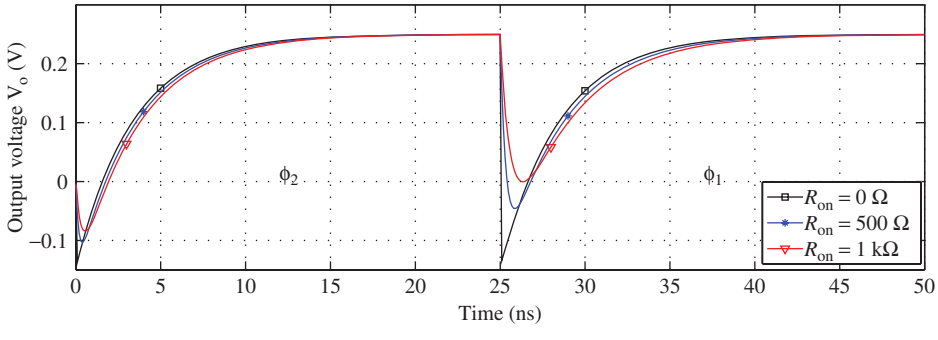


Figure 2.13 Illustration of the influence of the switch on-resistance on the transient response of an SC integrator with a loading SC branch. Simulation parameters used are same as those for Figure 2.8 for comparison purposes.

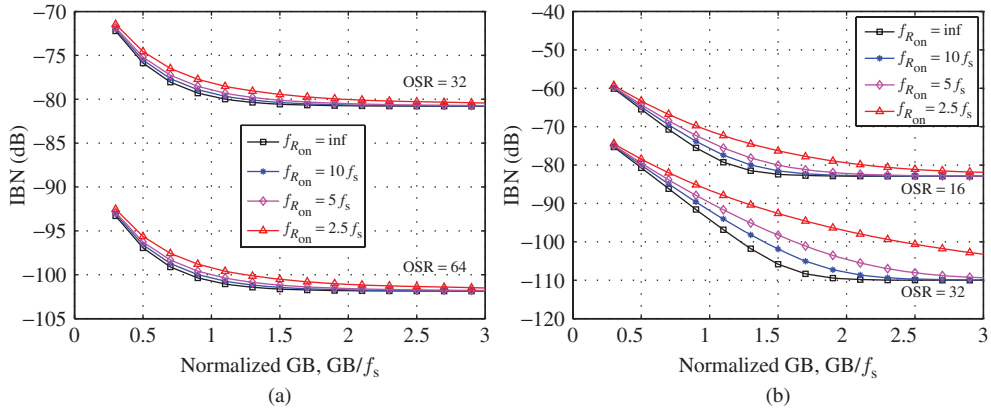


Figure 2.14 Influence of switch on-resistance on the in-band noise of SC-ΣΔMs: (a) third-order loop and (b) 2-1-1 cascade. Approximated results have been obtained from Equations 2.13 and 2.15 with $\epsilon_{st,R_{on}} = e^{-\pi GB(Hz)/[1+GB(Hz)/f_{R_{on}}]}$.

phase. To incorporate this effect into behavioral simulations, the effective amplifier GB during both clock phases can be approximated to [3],

$$\begin{aligned} GB_{R_{on},\phi_1}(Hz) &\approx \frac{GB_{\phi_1}(Hz)}{1+GB_{\phi_1}(Hz)/f_{R_{on},\phi_1}}, \quad \text{with} \quad f_{R_{on},\phi_1} = \frac{1}{2\pi\cdot 2R_{on}C_{Sn}} \\ GB_{R_{on},\phi_2}(Hz) &\approx \frac{GB_{\phi_2}(Hz)}{1+GB_{\phi_2}(Hz)/f_{R_{on},\phi_2}}, \quad \text{with} \quad f_{R_{on},\phi_2} = \frac{1}{2\pi\cdot 2R_{on}C_{S}} \end{aligned} \tag{2.33}$$

where f_{R_{on},ϕ_1} and f_{R_{on},ϕ_2} represent the RC poles introduced by loading SC branch and by the input SC branch, respectively, and GB_{ϕ_1} and GB_{ϕ_2} are respectively given by Equations 2.21 and 2.27.

In addition, results presented in Section 2.4.2 for the linear effect of finite amplifier GB on the IBN of SC-ΣΔMs can be easily refined to include the slow-down effect of the switches R_{on}. To that purpose, the following gain error

$$\epsilon_{st,R_{on}} = e^{-\pi \frac{GB(Hz)}{1+GB(Hz)/f_{R_{on}}}} \tag{2.34}$$

can be considered in Equations 2.13 and 2.14. Figure 2.14 illustrates the combined linear effect of the finite amplifier GB and finite switch on-resistance on a third-order single loop and on a 2-1-1 cascade SC-ΣΔM. The lower sensitivity of single loops to these errors is clear and a switch on-resistance such that $f_{R_{on}}$ is $4-5f_s$ is sufficient, in combination with the limited amplifier GB, to achieve full performance. This requirement for the R_{on} usually increases to $10-20f_s$ for cascade ΣΔMs as the oversampling increases.

2.5 Circuit Noise in SC-ΣΔMs

Electronic noise generated in transistors and resistors is present in any circuit implementation and imposes an ultimate limit to the resolution of ADCs. However, its impact is more

severe in DT-ΣΔMs that employ SC-techniques due to the white spectrum of the main circuit noise sources. In an SC-ΣΔM, these broadband noise components are sampled together with the input signal at the clock frequency, so that they fold over the modulator passband and may cause a considerable increase of the modulator in-band noise due to aliasing.

As stated in Section 2.1, the influence of nonidealities on the IBN of ΣΔMs is mainly determined by the location of the corresponding noise source in the modulator. With respect to circuit noise, all SC integrators in a ΣΔM add noise in the modulator passband, but the role of the front-end integrator is indeed dominant. When referred to the modulator input, noise power contributed by the remaining integrators is divided by the gain of preceding integrators within the modulator passband, so their influence strongly diminishes while moving from front-end to back-end integrators. Conversely, no shaping takes place at the modulator input and the first integrator has thus to fulfill the noise and linearity requirements of the complete ΣΔM.

Let us consider the SC integrator in Figure 2.15a to be the front-end integrator of an SC-ΣΔM. Two input SC branches are considered: the one including capacitor C_{S1} is assumed to sample the modulator input signal ($v_1 = v_{in}$), whereas the one including capacitor C_{S2} samples the DAC feedback signal ($v_2 = v_{fb}$). The main sources of circuit noise in SC integrators have been incorporated into the equivalent models in Figures 2.15b and 2.15c during each of the clock phases, namely, thermal noise generated in the switches and noise generated in the amplifier—both thermal and flicker components to be considered.[3]

Figure 2.15b shows the model for the thermal noise introduced by switches controlled by clock phase ϕ_1. For both SC branches, the two active switches are assumed to have the same on-resistance (R_{on}) and they are in series with a noise voltage source v_{sw}. The PSD of each of these noise sources in a single-sided frequency representation is thus $S_{sw} = 4kT(2R_{on})$, where k is Boltzmann's constant and T is the absolute temperature. Each of the noise sources generates a sample-and-held noise component in the corresponding capacitor voltage given by the well-known kT/C expression due to the foldover effect [7-9]:

$$S_{sw,C_{Si}}(f) \approx \frac{2kT}{C_{Si} f_s} \operatorname{sinc}^2(\pi f / f_s) \tag{2.35}$$

Figure 2.15c shows the model for the thermal noise introduced by switches controlled by clock phase ϕ_2 and for the noise in the amplifier. These switches originate an additional noise component in the capacitor voltage of the corresponding SC branch, similarly given by Equation 2.35. On the other hand, a single-pole model is assumed for the amplifier and its equivalent input noise is modeled by a voltage source v_{amp} at the positive input terminal. As illustrated in Figure 2.16, the amplifier noise is basically determined by a broadband thermal component and a narrowband flicker component, so that

$$S_{amp}(f) \approx S_{amp,th}(f) + S_{amp,1/f}(f) \approx S_{amp,th}\left(1 + \frac{f_{cr}}{f}\right) \tag{2.36}$$

where $S_{amp,th}$ represents the amplifier thermal noise PSD referred to its input and f_{cr} represents the amplifier corner frequency—that is, the frequency at which $1/f$ noise is

[3] The noise associated with the DAC reference voltage is not considered here for simplicity, but it can be incorporated in a similar way into the amplifier noise [3].

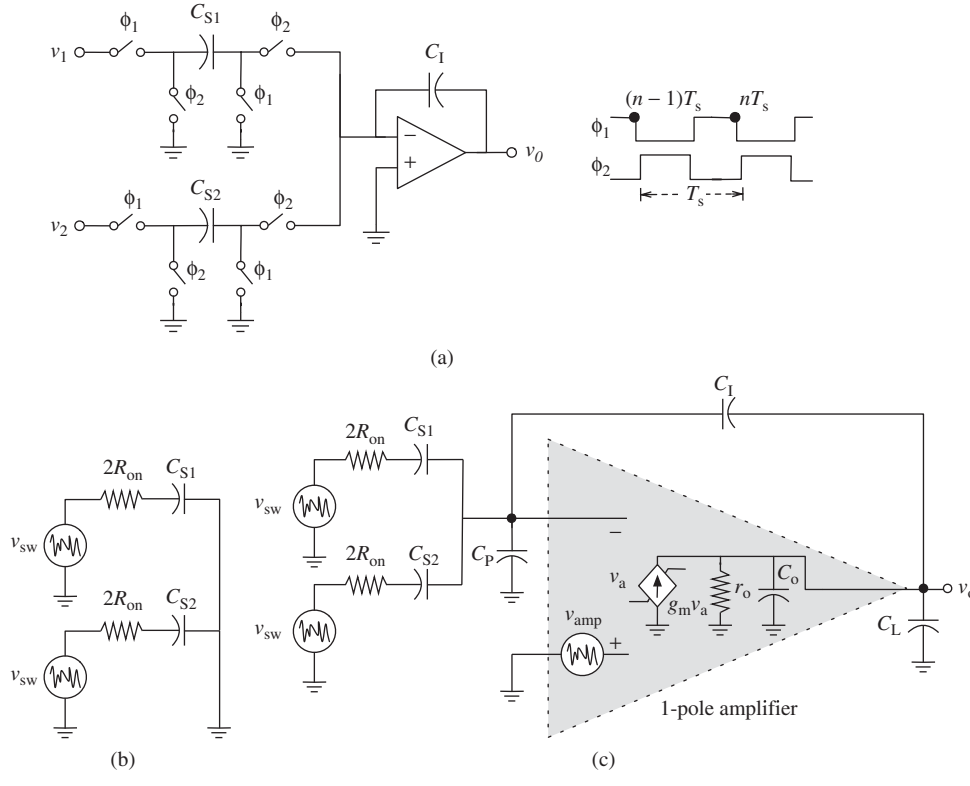


Figure 2.15 Circuit noise analysis in an SC integrator: (a) SC FE integrator with two input paths (single-ended version), (b) equivalent circuit model for sampling, with noise sources due to switches active during ϕ_1 and (c) equivalent circuit model for integration, with noise sources due to switches active during ϕ_2 and due to the amplifier.

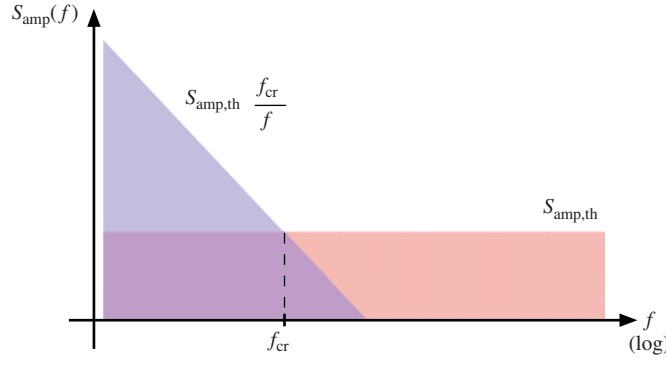


Figure 2.16 Illustration of the PSD of the amplifier noise showing the contributions of $1/f$ and thermal noise.

equal to the thermal noise.[4] The amplifier noise generates correlated sample-and-held noise components in the integrator sampling capacitors that can be obtained as

$$S_{\text{amp},C_{\text{Si}}}(f) \approx S_{\text{amp,th}}\left(\frac{2B_{\text{w,noise}}}{f_{\text{s}}} + \frac{f_{\text{cr}}}{f}\right)\text{sinc}^2(\pi f/f_{\text{s}}) \tag{2.37}$$

where the amplifier noise bandwidth $B_{\text{w,noise}}$ is required to account for the foldover effect of thermal noise. As shown in [3], it can be estimated as $B_{\text{w,noise}}(\text{Hz}) \approx \text{GB}_{\phi_2}(\text{rad/s})/4$, with GB_{ϕ_2} given by Equation 2.27.

Adding up the former circuit noise components in the SC integrator, the total input-referred[5] noise PSD yields [3]

$$S_{\text{noise,in}}(f) \approx 2\left[2S_{\text{sw},C_{\text{S1}}}(f) + 2S_{\text{sw},C_{\text{S2}}}(f)\frac{C_{\text{S2}}^2}{C_{\text{S1}}^2}\right] + S_{\text{amp},C_{\text{Si}}}(f)\left(1 + \frac{C_{\text{S2}}}{C_{\text{S1}}}\right)^2 \tag{2.38}$$

where the factor 2 multiplying $S_{\text{sw},C_{\text{Si}}}(f)$ accounts for the contributions of switches controlled by ϕ_1 and ϕ_2, whereas the factor 2 before the brackets accounts for the actual fully-differential implementation of the SC integrator, in which the number of SC branches and thus of switches doubles in comparison to the single-ended scheme in Figure 2.15. Replacing Equations 2.35 and 2.37 in 2.38, the total input-referred noise PSD of the front-end integrator of an SC-ΣΔM can thus be approximated to be

$$S_{\text{noise,in}}(f) \approx \frac{8kT}{C_{\text{S1}}f_{\text{s}}}\left(1 + \frac{C_{\text{S2}}}{C_{\text{S1}}}\right) + S_{\text{amp,th}}\left(\frac{\text{GB}_{\phi_2}}{2f_{\text{s}}} + \frac{f_{\text{cr}}}{f}\right)\left(1 + \frac{C_{\text{S2}}}{C_{\text{S1}}}\right)^2 \tag{2.39}$$

in which the approximation $\text{sinc}^2(\pi f/f_{\text{s}}) \approx 1$ for $f \ll f_{\text{s}}$ has been used for simplicity.

The input-referred IBN of an SC-ΣΔM due to circuit noise can be easily obtained by integrating the former expression over the input signal bandwidth, so that

$$\begin{aligned}\text{IBN}_{\text{noise}} = \int_0^{B_{\text{w}}} S_{\text{noise,in}}(f)\text{d}f &\approx \frac{4kT}{C_{\text{S1}}\text{OSR}}\left(1 + \frac{C_{\text{S2}}}{C_{\text{S1}}}\right) \\ &+ S_{\text{amp,th}}\left[\frac{\text{GB}_{\phi_2}}{4\text{OSR}} + f_{\text{cr}}\ln\left(\frac{B_{\text{w}}}{f_0}\right)\right]\left(1 + \frac{C_{\text{S2}}}{C_{\text{S1}}}\right)^2\end{aligned} \tag{2.40}$$

in which the $1/f$ noise component has been integrated from a frequency $f_0 > 0$ to exclude DC due to its logarithmic nature.

For an SC-ΣΔM to achieve a given noise performance, the sum of all three components in Equation 2.40 have to meet the demanded noise floor. Note that:

[4] For a single-stage amplifier, the thermal noise component can be approximated to $S_{\text{amp,th}} \approx \frac{8kT}{3g_{\text{m}}}2(1 + n_{\text{th}})$, where g_{m} is the transconductance of the amplifier input transistor, n_{th} represents the noise factor contributed by the remaining transistors in the amplifier, and a factor 2 accounts for the fully-differential amplifier implementation. The amplifier corner frequency strongly depends on the size of the input transistors, their type, and the technological process, and is thus not easy to accurately estimate it with a closed-form expression. However, both noise parameters can be easily characterized by means of electrical simulation.

[5] That is, referred to the voltage across capacitor C_{S1} that samples the modulator input signal.

- For a given OSR, to reduce the contribution of the switches' thermal noise, the size of the sampling capacitors at the modulator input must be increased, which results in larger speed requirements for the amplifier and thus in larger power consumption.
- For a given OSR, to reduce the contribution of the amplifier thermal noise, its GB must be reduced as much as the integrator settling requirements allow.
- To reduce the flicker contribution the amplifier corner frequency must be kept low. In low-bandwidth applications, cancelation techniques such as correlated double sampling (CDS) or chopper are often required for further reduction of the $1/f$ component [10].

2.6 Clock Jitter in SC-ΣΔMs

Discrete-time ΣΔMs are affected in practice by timing uncertainties[6] in the clock phases that control the SC operation. However, they exhibit larger tolerance to clock jitter than Nyquist converters, because jitter sensitivity is reduced by the modulator OSR [12].

The effect of clock jitter in SC-ΣΔMs is mainly limited to a sampling uncertainty of the modulator input signal. Timing uncertainties during the integration phase only cause an extra error to be added to the integrator settling error and their influence can be neglected in practice, whereas the contributions of other integrators than the front-end one will be reduced by the noise shaping. Therefore, different SC-ΣΔMs exhibit similar sensitivity to clock jitter [13].

Sampling time uncertainty causes a nonuniform sampling of the modulator input signal that results in an increase of the in-band error power. The magnitude of this increase is usually estimated for SC-ΣΔMs assuming random statistical properties for the clock jitter [12]. For a modulator input sinewave $v_{\text{in}}(t) = A_{\text{in}} \sin(2\pi f_{\text{in}} t)$ as shown in Figure 2.17, an uncertainty of Δt in the sampling instant causes an error in the sampled signal given by:

$$\epsilon_{\text{j}} = v_{\text{in}}(nT_{\text{s}} + \Delta t) - v_{\text{in}}(nT_{\text{s}}) \approx \left.\frac{\text{d}v_{\text{in}}(t)}{\text{d}t}\right|_{nT_{\text{s}}} \Delta t = (2\pi f_{\text{in}}) A_{\text{in}} \Delta t \cos(2\pi f_{\text{in}} nT_{\text{s}}) \quad (2.41)$$

Under the assumption of white jitter, the power of this modulated error distributes uniformly, so that only a fraction of it is located within the ΣΔM passband. The in-band noise due to clock jitter can thus be easily obtained as

$$\text{IBN}_{\text{j}} = \int_{-B_{\text{w}}}^{+B_{\text{w}}} \frac{A_{\text{in}}^2}{2} \frac{(2\pi f_{\text{in}} \sigma_{\text{j}})^2}{f_{\text{s}}} \text{d}f = \frac{A_{\text{in}}^2}{2} \frac{(2\pi f_{\text{in}} \sigma_{\text{j}})^2}{\text{OSR}} \quad (2.42)$$

where σ_{j} represents the standard deviation of the timing uncertainty. Taking into account that $A_{\text{in}} \leq \Delta/2$ and $f_{\text{in}} \leq f_{\text{s}}/(2\text{OSR})$, an upper bound can be calculated for Equation 2.42

$$\text{IBN}_{\text{j}} \leq \frac{\Delta^2}{8} \frac{(\pi f_{\text{s}} \sigma_{\text{j}})^2}{\text{OSR}^3} \quad (2.43)$$

showing that the sensitivity of SC-ΣΔMs to clock jitter is reduced by OSR^{-3}.

[6] This effect is inherent to every clock generation circuitry—crystal oscillators, PLL-based oscillators, etc.—and is mostly caused by thermal noise, phase noise, and spurious tones that degrade the spectral purity of clock signal [11].

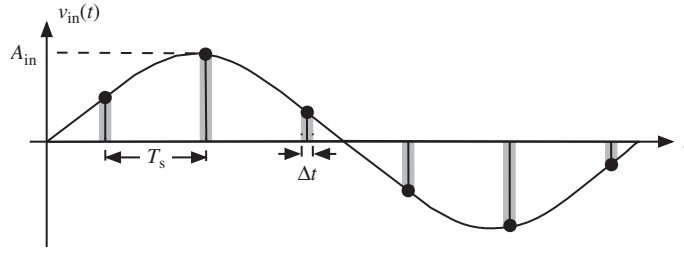


Figure 2.17 Illustration of nonuniform sampling of a signal due to clock jitter. The gray shaded areas represent the timing uncertainties.

2.7 Sources of Distortion in SC-ΣΔMs

Analog devices used for the implementation of ΣΔMs exhibit in practice a certain nonlinearity. These nonlinearities generate distortion and thus limit the peak SNDR attainable for high input amplitudes. Nevertheless, deriving closed-form expressions for the distortion generated in a ΣΔM is in general much more awkward than analyzing the effect of linear errors. Therefore, several simplifications are often made to handle nonlinearities. First, only the sources of distortion associated with the front-end integrator in the ΣΔM are considered, as they directly add to the modulator signal with no attenuation and thus dominate the overall modulator nonlinearity. Distortion generated in subsequent integrators is suppressed by the increasing noise shaping when referred to the modulator input, so that their contributions can be considered negligible in practice. Second, each source of nonlinearity is conceived as a small deviation from the ideal linear behavior—that is, as a weak nonlinearity—that affects the modulator performance in an additive way.

Figure 2.18 illustrates the main sources of distortion in an SC integrator, in which a fully-differential topology is assumed for the suppression of even-order harmonics. In SC-ΣΔMs, linearity is basically limited by the voltage dependency of capacitors, of the switches on-resistance, and of the amplifier gain, as well as by the SR-limited integrator dynamics (already discussed in Section 2.4.3). Distortion arising from charge injection in

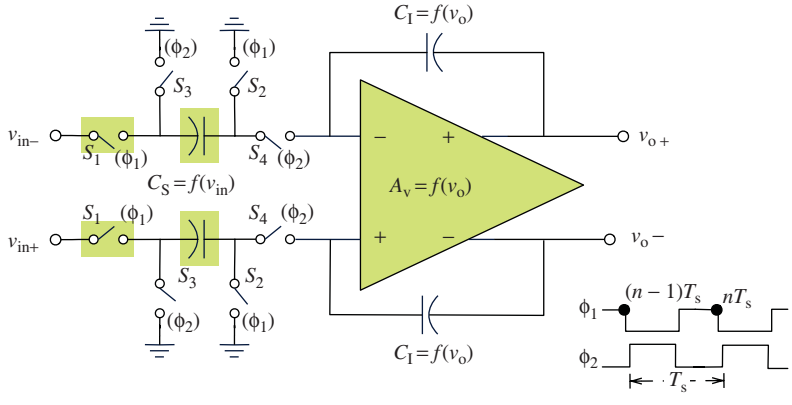


Figure 2.18 Main sources of distortion in a fully-differential SC integrator.

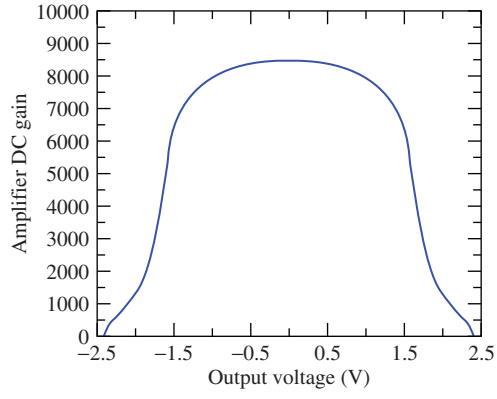


Figure 2.19 Illustration of the dependency of the amplifier gain on the output voltage level.

the switches can be neglected if clock phases with delayed falling edges are employed [14]. Besides, given the highly linear capacitors that present mixed-mode technological processes offer—such as MiM and MoM capacitors—their effect will not be further considered here.[7] The influence of the nonlinearity of the switches and of the amplifier gain will be discussed later.

2.7.1 Nonlinear Amplifier Gain

The DC gain of an amplifier exhibits in practice a voltage-dependent characteristic, because the output resistance of the amplifier output transistors decreases as the amplifier output voltage deviates from the quiescent point. Figure 2.19 illustrates this effect with electrical simulations results of a folded-cascode amplifier designed in a 2.5-V 0.25-μm CMOS process. Note that the amplifier DC gain is about 8500 (78.5 dB) at the common-mode output voltage, but it decreases for increasing output levels and drops abruptly near the amplifier saturation region.

The influence of the amplifier gain nonlinearity can be easily incorporated into the leaky integrator model in Section 2.2, so that the difference equation in Equation 2.2 can be rewritten as [3]

$$\begin{aligned} v_{\mathrm{o}}(nT_{\mathrm{s}}) = & \frac{1+\left(1+\frac{C_{\mathrm{P}}}{C_{\mathrm{I}}}\right)\frac{1}{A_{\mathrm{v}}(v_{\mathrm{o,n-1}})}}{1+\left(1+\frac{C_{\mathrm{P}}}{C_{\mathrm{I}}}+\sum_{i=1}^{N_{\mathrm{i}}}\frac{C_{\mathrm{Si}}}{C_{\mathrm{I}}}\right)\frac{1}{A_{\mathrm{v}}(v_{\mathrm{o,n}})}} v_{\mathrm{o}}[(n-1)T_{\mathrm{s}}] \\ & + \frac{\sum_{i=1}^{N_{\mathrm{i}}}\frac{C_{\mathrm{Si}}}{C_{\mathrm{I}}} v_{\mathrm{i}}[(n-1)T_{\mathrm{s}}]}{1+\left(1+\frac{C_{\mathrm{P}}}{C_{\mathrm{I}}}+\sum_{i=1}^{N_{\mathrm{i}}}\frac{C_{\mathrm{Si}}}{C_{\mathrm{I}}}\right)\frac{1}{A_{\mathrm{v}}(v_{\mathrm{o,n}})}} \end{aligned} \tag{2.44}$$

where $A_{\mathrm{v}}(v_{\mathrm{o,n-1}})$ represents the effective amplifier gain at the output voltage corresponding to clock cycle $(n-1)T_{\mathrm{s}}$ and $A_{\mathrm{v}}(v_{\mathrm{o,n}})$ corresponds to that of clock cycle nT_{s}. As will

[7] If different capacitor primitives are to be considered for the SC implementation, the interested reader can find details on the generated distortion in [3, 13].

be shown in Section 3.3, solving this difference equation in an iterative way, together with a table look-up for the amplifier gain, allows for accurately accounting for the voltage dependency of the amplifier gain in transient behavioral simulations—in spite of the nonlinearity being weak or strong!

For weak nonlinearities, a polynomial approximation can be used for modeling the voltage dependency near the quiescent point and for obtaining rough estimations of the generated distortion. Let us assume that the amplifier gain of the front-end integrator in an SC-ΣΔM, such as that shown in Figure 2.18, is expressed as

$$A_{\mathrm{v}}(v_{\mathrm{o}}) = A_{\mathrm{v}}(1 + c_{\mathrm{nl1}} v_{\mathrm{o}} + c_{\mathrm{nl2}} v_{\mathrm{o}}^2 + ...) \tag{2.45}$$

where c_{nli} represents the ith-order voltage-gain coefficient of the amplifier DC gain. If a sinewave with amplitude A_{in} is applied at the modulator input, the input-referred distortion of the third-order harmonic can be estimated as [3, 15]

$$\mathrm{HD}_3 \approx \frac{c_{\mathrm{nl2}}}{4} \frac{\left(1 + \frac{C_{\mathrm{P}}}{C_{\mathrm{I}}} + \frac{C_{\mathrm{S}}}{C_{\mathrm{I}}}\right)}{A_{\mathrm{v}}} \left(\frac{C_{\mathrm{S}}}{C_{\mathrm{I}}}\right)^2 A_{\mathrm{in}}^2 \tag{2.46}$$

where C_{P} is the parasitic capacitor at the amplifier input nodes. Note that decreasing the integrator gain coefficient clearly helps to reduce distortion. However, the most direct way to reduce the effect of the amplifier gain nonlinearity is increasing the value of the amplifier gain itself [3, 13].

2.7.2 *Nonlinear Switch On-Resistance*

Switches in SC-ΣΔMs are usually implemented as CMOS transmission gates, so that, at least, either the nMOS or the pMOS transistors are on for a given voltage level to be transmitted. Figure 2.20a sketches the on-conductance of nMOS and pMOS switches,

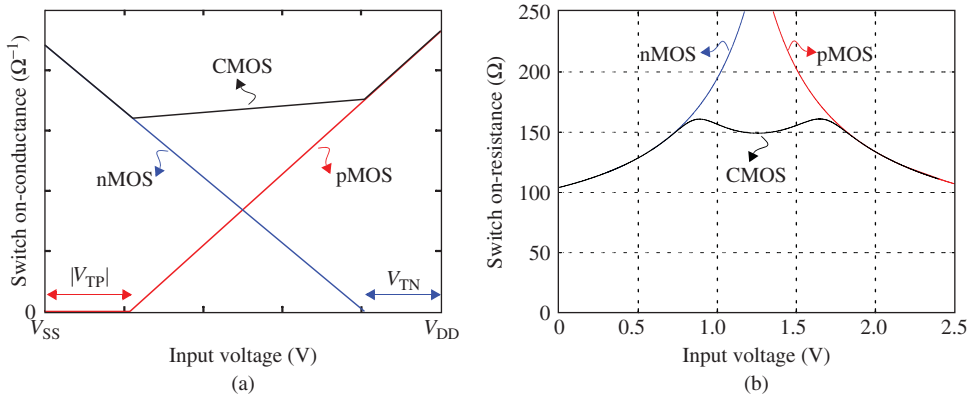


Figure 2.20 Illustration of a switch on-state performance: (a) sketch of the on-conductance versus input voltage and (b) simulation results of the on-resistance versus input voltage in a 2.5-V 0.25-µm CMOS process.

assuming that they exhibit a resistance in the triode region that can be approximated to [3, 16]

$$\begin{aligned} R_{\text{onN}} &\approx \frac{1}{K_{\text{N}}\frac{W_{\text{N}}}{L_{\text{N}}}(V_{\text{GS}} - V_{\text{TN}} - \frac{V_{\text{DS}}}{2})} \approx \frac{1}{K_{\text{N}}\frac{W_{\text{N}}}{L_{\text{N}}}(V_{\text{DD}} - V_{\text{TN}} - V_{\text{in}})} \\ R_{\text{onP}} &\approx \frac{1}{K_{\text{P}}\frac{W_{\text{P}}}{L_{\text{P}}}(V_{\text{SG}} - |V_{\text{TP}}| - \frac{V_{\text{SD}}}{2})} \approx \frac{1}{K_{\text{P}}\frac{W_{\text{P}}}{L_{\text{P}}}(V_{\text{in}} - V_{\text{SS}} - |V_{\text{TP}}|)} \end{aligned} \tag{2.47}$$

where $V_{\text{in}} = (V_{\text{D}} + V_{\text{S}})/2$ represents the switch input voltage—that is, the common-mode voltage of the drain and source terminals. The on-resistance of the CMOS transmission gate is thus obtained as $R_{\text{on}} = R_{\text{onN}} \| R_{\text{onP}}$, warranting a rail-to-rail operation of the switch as long as $V_{\text{DD}} - V_{\text{SS}} > V_{\text{TN}} + |V_{\text{TP}}|$.[8] Figure 2.20b shows electrical simulation results of a transmission gate in a 2.5-V 0.25-μm CMOS process, in which the voltage dependency of the switch on-resistance is clearly visible.

To analyze the relative influence of the different switches in the front-end integrator of an SC-ΣΔM on the generated distortion, let us consider the schematic in Figure 2.18. The modulator input signal is sampled on capacitors C_{S} through switches S_1 and S_2 during ϕ_1. As switches S_1 are connected to the modulator input, their on-resistances directly depend on the modulator input level and are the dominant source of distortion. However, switches S_2 have one of their terminals connected to the common-mode voltage—that is, to a voltage that remains approximately constant over time—so that the voltage level of these switches is not expected to change much over the clock periods [5]. As a result, the distortion introduced by switches S_2 will be considerably lower than that of switches S_1. The same reasoning can be applied to switches S_3 and S_4 during ϕ_2: switches S_3 have one terminal connected to a fixed voltage—the common-mode voltage, as depicted in Figure 2.18, or the DAC feedback voltage—and switches S_4 are connected to the virtual ground of the amplifier. Their influence on the generated distortion can thus be neglected in practice.

Chapter 4 will demonstrate the distortion generated by nonlinear sampling in an SC-ΣΔM can be accurately evaluated through transistor-level electrical simulations of the equivalent circuit in Figure 2.21. A tone with large amplitude can be applied at the differential input and the differential voltage stored in capacitors C_{S} can be collected at the clock rate to compute the FFT and measure the THD.

Finally, Section 4.4.1 will demonstrate that the generated distortion can be reduced not only by keeping the switch as linear as possible, but also by reducing the value of the on-resistance itself. Figure 2.22 illustrates the switch on-resistance for different alternatives for the relative sizing of the switch transistors. If the sizes compensate the difference in the transconductance parameter of the nMOS and pMOS transistors—that is, $K_{\text{N}}W_{\text{N}} = K_{\text{P}}W_{\text{P}}$, as used for instance in [3]—the nonlinearity of the on-resistance is low, but its average value is larger than for $W_{\text{N}} = W_{\text{P}}$—as used in [5]. In the latter case, the switch area and its parasitic capacitors increase, but the slow-down effect of the switch R_{on} on the integrator settling will decrease, as discussed in Section 2.4.4. Note that the

[8] Note that, if $V_{\text{DD}} - V_{\text{SS}} < V_{\text{TN}} + |V_{\text{TP}}|$, a gap will appear at the mid-range in Figure 2.20a, because neither the nMOS switch nor the pMOS will conduct. In a low-voltage environment, this problem is often overcome using clock-boosting techniques [17, 18] or low-V_{T} transistors.

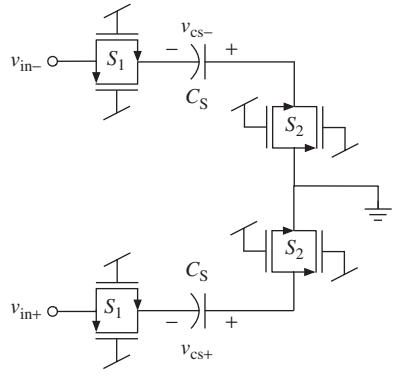


Figure 2.21 Equivalent circuit for evaluating distortion during sampling due to the switch nonlinearity.

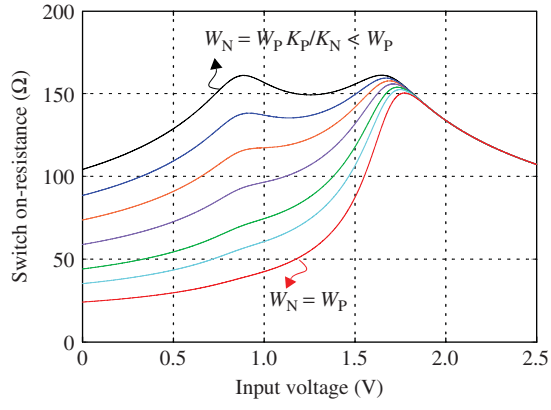


Figure 2.22 Illustration of the switch on-resistance nonlinearity for different transistor sizings in a 2.5-V 0.25-μm CMOS process.

design trade-offs above can be solved in opposite directions depending on the particular linearity and speed requirements of an SC-ΣΔM, as well as the modulator input range relative to the supply voltage.

2.8 Nonidealities in Continuous-Time ΣΔ Modulators

As illustrated in Figure 2.23, circuit errors in CT-ΣΔMs can be divided into two main categories:

- Building-block errors, which are the nonideal effects derived from the modulator loop filter implementation—similar to the SC case—, such as finite amplifier gain (for active-RC implementations), integrator time-constant error, circuit noise, nonlinearities, etc.

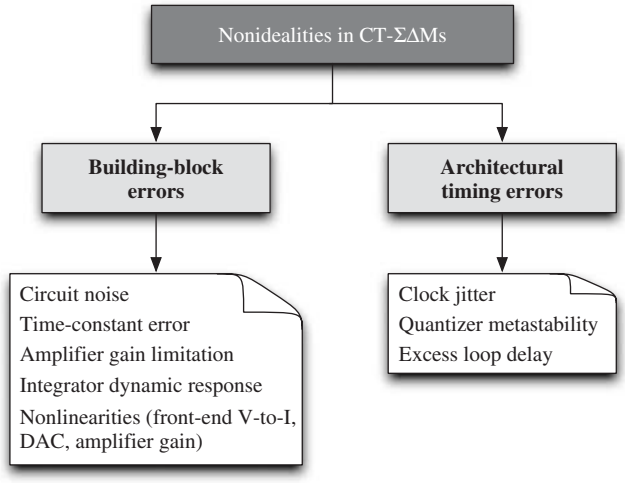


Figure 2.23 Main nonidealities affecting the performance of continuous-time ΣΔMs.

- Architectural timing errors, namely: clock jitter, excess loop delay, and quantizer metastability.

The former group of errors causes similar effects on the performance of CT-ΣΔMs as in the case of their DT counterparts. Therefore, they can also be classified according to their degradation on the modulator performance: either causing a deviation in NTF or an additive noise component at the modulator input.

As stated in Section 1.8, CT implementations are potentially faster than SC ones, leading to much more relaxed designs (in terms of power consumption) when high-speed operation is required. In addition, they do not suffer from kT/C noise. However, SC implementations have intrinsically lower parameter variations, as most circuit parameters are defined by capacitor ratios, instead of by absolute parameter values as in the case of CT-ΣΔMs. In the following sections, main nonideal effects of CT-ΣΔMs are described, putting emphasis on the most critical design issues.

2.9 Clock Jitter in CT-ΣΔMs

Continuous-time ΣΔMs are more severely affected by timing uncertainties than their DT counterparts. Conversely to DT-ΣΔMs, sampling time uncertainties occur at the quantizer input, where the jitter-induced error is strongly suppressed by the noise shaping and can thus be neglected in practice. However, errors resulting from timing uncertainties in the DAC feedback signal add directly to the modulator input with no suppression, hence being the dominant jitter effect and limiting the overall modulator performance.

The magnitude of jitter-induced errors heavily depends on the pulse shape of the DAC feedback signal as well as on the statistical properties of clock jitter. Both aspects have been extensively studied in published literature [2, 19-25]. In the following sections the effect of DAC pulse shapes that are most commonly used—that is, NRZ, RZ, and SC

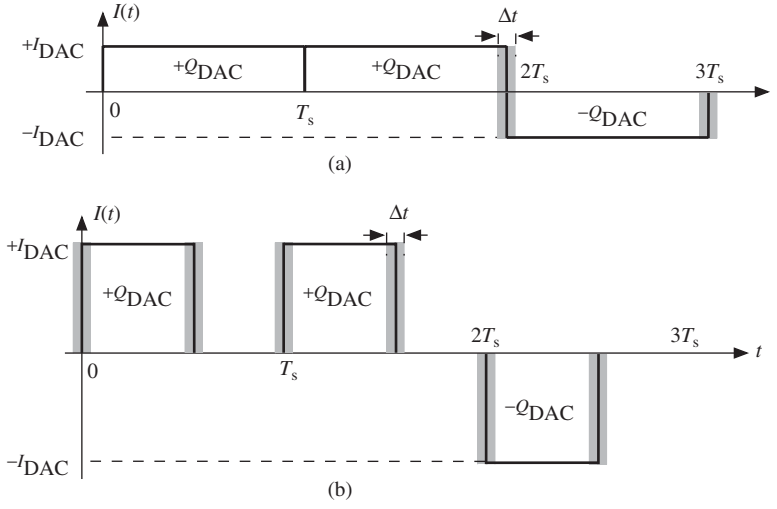


Figure 2.24 Illustration of the jitter effect on the feedback signal of a single-bit CT-ΣΔM: (a) NRZ DAC and (b) RZ DAC. The gray shaded areas represent the timing uncertainties. A pulse sequence of $(+1, +1, -1)$ is considered.

pulses—is revised under a random jitter assumption to provide a general overview of its effect.

2.9.1 Jitter in Return-to-Zero DACs

Figure 2.24 illustrates the effect of clock jitter on rectangular-shaped DAC feedback signals of a single-bit CT-ΣΔM. Both RZ and NRZ schemes, typically employed in SI DACs, are depicted for comparison purposes.

If the timing uncertainty in one side of an RZ pulse is considered, jitter will induce a charge error on the feedback signal $\epsilon_{Q_{DAC}}$ within one clock period that is given by

$$\epsilon_{Q_{DAC}} = I_{DAC}\Delta t \tag{2.48}$$

where I_{DAC} represents the amplitude of the RZ DAC pulses and Δt for the timing uncertainty. If a random jitter is assumed and both sides of the RZ pulse are accounted for (Figure 2.24b), the variance of the jitter-induced charge error per clock cycle yields

$$\sigma^2_{Q,RZ} = 2I^2_{DAC}\sigma^2_j \tag{2.49}$$

where σ^2_j represents the variance of the timing uncertainty and the factor 2 results from assuming that the instants of both jittered sides are statistically independent. Relating Equation 2.49 to time, the former variance of charge error can be translated into a current error variance:

$$\sigma^2_{I,RZ} = \frac{\sigma^2_{Q,RZ}}{T^2_s} = 2I^2_{DAC}(f_s\sigma_j)^2 \tag{2.50}$$

The amplitude of the RZ DAC pulse I_{DAC} is determined by the feedback scaling coefficient and the DAC step width Δ. For the first integrator of a single-bit CT-ΣΔM, it is given by [2]

$$I_{DAC} = k_{1,RZ}\,\Delta/2 \tag{2.51}$$

so that:

$$\sigma^2_{I,RZ} = \left(\frac{\Delta}{2}\right)^2 2k^2_{1,RZ}(f_s\sigma_j)^2 \tag{2.52}$$

To obtain the resulting in-band noise, this noise component can be easily related to the modulator input as [2]

$$IBN_{j,RZ} \approx \frac{\sigma^2_{I,RZ}}{k^2_{1,NRZ}OSR} = \left(\frac{\Delta}{2}\right)^2 \frac{k^2_{1,RZ}}{k^2_{1,NRZ}} \frac{2(f_s\sigma_j)^2}{OSR} \tag{2.53}$$

showing that the sensitivity of a CT-ΣΔM with RZ feedback to clock jitter is reduced by only OSR^{-1}. It thus compares unfavorably with DT-ΣΔMs, in which a suppression proportional to OSR^{-3} is achieved—see Equation 2.43.

2.9.2 *Jitter in NonReturn-to-Zero DACs*

The effect of clock jitter in NRZ feedback DACs can be obtained in a similar way as previously done for RZ DACs. However, contrary to RZ feedback pulses, jitter affects the feedback charge only if the feedback signal changes its state, as shown in Figure 2.24a. Thus, the variance of the jitter-induced charge error per clock cycle for an NRZ DAC is in practice less than half of the RZ counterpart in Equation 2.49, because a state transition does not necessarily occur in every cycle. For large input signals and single-bit quantization, it can be approximated to [2, 19]:

$$\sigma^2_{Q,NRZ} \approx 0.7 I^2_{DAC}\sigma^2_j \tag{2.54}$$

On the other hand, if a transition state takes place in an RZ DAC, the amplitude of the step is Δ instead of $\Delta/2$ in Equation 2.51, so that:

$$I_{DAC} = k_{1,NRZ}\,\Delta \tag{2.55}$$

Therefore, the resulting in-band noise for a single-bit CT-ΣΔM employing NRZ feedback DAC can be estimated as:

$$IBN_{j,NRZ} \approx \frac{\sigma^2_{I,NRZ}}{k^2_{1,NRZ}OSR} = \Delta^2 \frac{0.7(f_s\sigma_j)^2}{OSR} \tag{2.56}$$

Taking into account that $k_{1,RZ}$ is typically twice the value of $k_{1,\,NRZ}$—that is, the duration of the RZ pulse is typically $T_s/2$—the sensitivity to clock jitter of both rectangular-shape feedback DACs can be easily compared from Equations 2.53 and 2.56

$$\frac{IBN_{j,NRZ}}{IBN_{j,RZ}} \approx \frac{0.7}{2} \approx -4.5\ \text{dB} \tag{2.57}$$

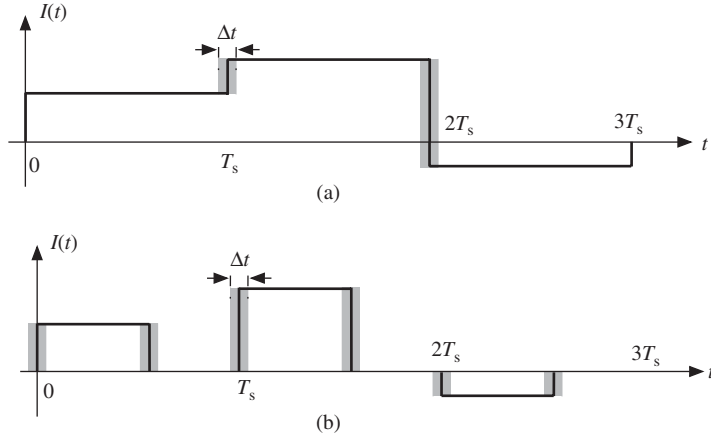


Figure 2.25 Illustration of the jitter effect on the feedback signal of a multibit CT-ΣΔM: (a) NRZ DAC and (b) RZ DAC. The gray shaded areas represent the timing uncertainties.

showing that NRZ feedback leads to lower sensitivity to clock jitter than the RZ counterpart [2, 25].

Resorting to multibit quantization is a common approach for reducing the sensitivity to clock jitter of CT-ΣΔMs. Figure 2.25a illustrates a DAC feedback signal employing multibit NRZ. It intuitively comes out that the number of state transitions per clock cycle increases from 0.7 (for the single-bit case) to a value close to 1 (for the multibit case), but two adjacent NRZ pulses will mostly differ by only 1LSB, thus reducing the jitter-induced charge error per clock cycle compared with Equation 2.54. Therefore, the in-band noise due to clock jitter of an NRZ CT-ΣΔM is reduced approximately 6 dB per additional bit [2]. From Figure 2.25b it can also be intuitively derived that the reduction of jitter noise by resorting to multibit RZ is only minor compared with multibit NRZ, because the feedback signal has mostly to change by more than 1LSB at each clock cycle.

2.9.3 *Jitter in Switched-Capacitor DACs*

The use of shaped pulses in the feedback DAC is an alternative approach commonly employed for reducing the sensitivity of CT-ΣΔMs to clock jitter. Figure 2.26 illustrates a typical feedback signal in an SC DAC, in which the RZ scheme uses an exponentially decaying waveform. If the timing uncertainty in one side of the SC-RZ pulse is considered, the charge error on the feedback signal $\epsilon_{Q_{\mathrm{DAC}}}$ induced by jitter within one clock period yields

$$\epsilon_{Q_{\mathrm{DAC}}} \approx I_{\mathrm{DAC}}\, \mathrm{e}^{-\frac{T_s}{2\tau_{\mathrm{DAC}}}}\, \Delta t \tag{2.58}$$

where $\tau_{\mathrm{DAC}} = R_{\mathrm{DAC}} C_{\mathrm{DAC}}$ is the time constant[9] of the DAC exponential current. A duration of $T_s/2$ is assumed for the RZ interval in Equation 2.58. If both jittered sides are considered

[9] The resistance R_{DAC}, via which the DAC capacitor C_{DAC} is discharged, is generally determined by the resistance of the DAC switches and the integrator input impedance in feedback configuration [26].

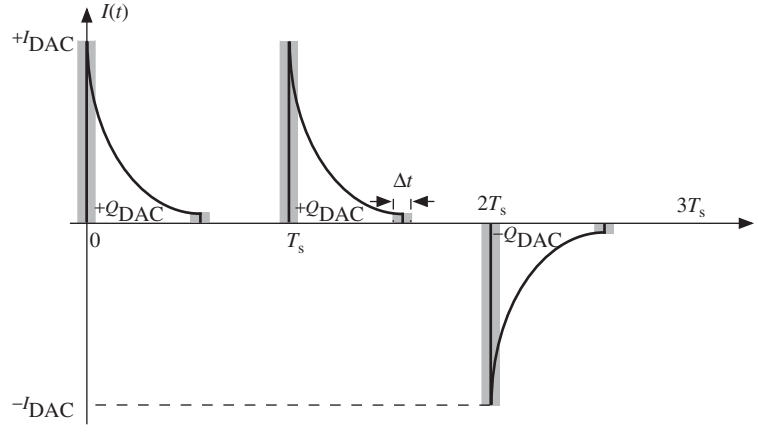


Figure 2.26 Illustration of the jitter effect on the feedback signal of a CT-ΣΔM with an SC DAC. The gray shaded areas represent the timing uncertainties. A pulse sequence of $(+1, +1, -1)$ is considered.

(Figure 2.26) and their instants are assumed uncorrelated, the variance of the jitter-induced charge error per clock cycle for an SC-RZ DAC can be approximated to:

$$\sigma^2_{\mathrm{Q,SC}} = 2\left(I_{\mathrm{DAC}}\, \mathrm{e}^{-\frac{T_\mathrm{s}}{2\tau_{\mathrm{DAC}}}}\right)^2 \sigma^2_{\mathrm{j}} \tag{2.59}$$

Following a similar procedure to that in Section 2.9.1 for an SI-RZ feedback, the resulting in-band noise can be obtained as [2]:

$$\mathrm{IBN}_{\mathrm{j,SC}} \approx \frac{\sigma^2_{\mathrm{I,SC}}}{k^2_{\mathrm{1,NRZ}}\mathrm{OSR}} = \left(\frac{\Delta}{2}\right)^2 \frac{k^2_{\mathrm{1,SC}}}{k^2_{\mathrm{1,NRZ}}} \left(\mathrm{e}^{-\frac{T_\mathrm{s}}{2\tau_{\mathrm{DAC}}}}\right)^2 \frac{2(f_\mathrm{s}\sigma_\mathrm{j})^2}{\mathrm{OSR}} \tag{2.60}$$

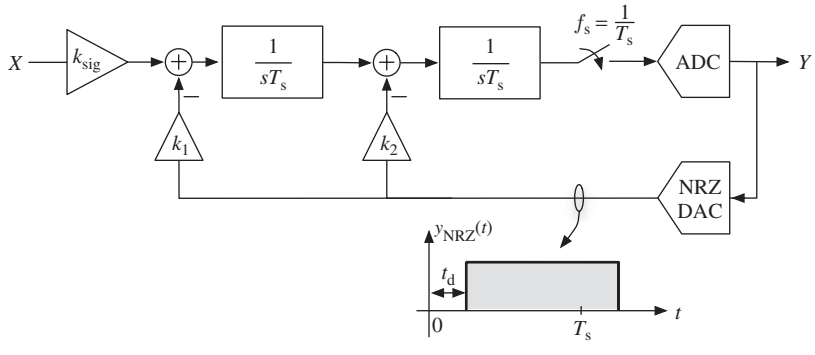
Note that the jitter-induced noise is exponentially reduced compared with an SI-RZ feedback—see Equation 2.53.

2.10 Excess Loop Delay in CT-ΣΔMs

Excess loop delay (ELD) is a timing error that largely impacts the behavior of CT-ΣΔMs. It can be defined as the *constant* delay between the quantizer sampling edge and the corresponding edge in the feedback DAC pulse and ultimately arises from the nonzero switching time of transistors in the quantizer and DAC circuitry [27].[10] ELD can be expressed as a fraction of the sampling period

$$t_\mathrm{d} = \tau_\mathrm{d} T_\mathrm{s} \tag{2.61}$$

[10] As will be discussed in Section 2.11, deliberate delays can also be introduced between the quantizer clocking edge and the subsequent latch feeding the DAC to increase the available time slot for the quantizer decision and mitigate the influence of quantizer metastability [19, 28-30].



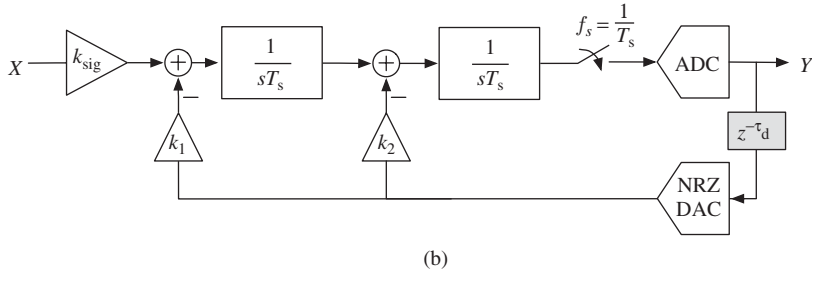


Figure 2.27 Illustration of excess loop delay in a second-order CT-ΣΔM: (a) effect on an NRZ DAC pulse and (b) equivalent diagram with an explicit delay block between the ADC and the DAC.

and is often represented as an explicit delay that is inserted prior to the DAC, as shown in Figure 2.27.

Note from Figure 2.27a that, if an NRZ pulse waveform is used in the feedback DAC, ELD shifts part of the feedback pulse into the next clock cycle. This actually results in a change of the DAC impulse response from its ideal expression

$$r_{\mathrm{DAC}}(t) = r_{(0,1)}(t) = \begin{cases} 1, & 0 \le t < T_s \\ 0, & \text{otherwise} \end{cases} \tag{2.62}$$

to,

$$r^*_{\mathrm{DAC}}(t) = r_{(\tau_d, 1+\tau_d)}(t) = \begin{cases} 1, & t_d \le t < T_s + t_d \\ 0, & \text{otherwise} \end{cases} \tag{2.63}$$

which can be expressed as a linear combination of a DAC pulse from τ_d to 1 and a one-sample-delayed DAC pulse from 0 to τ_d [29]:

$$r^*_{\mathrm{DAC}}(t) = r_{(\tau_d, 1+\tau_d)}(t) = r_{(\tau_d, 1)}(t) + r_{(0,\tau_d)}(t-1) \tag{2.64}$$

Taking into account these two feedback pulses, Equation 1.59 and Table 1.4 can be adopted to derive the DT-equivalent of the actual CT-ΣΔM affect by ELD,

$$H(z, \tau_d) = \mathcal{Z}\{\mathcal{L}^{-1}[R^*_{\mathrm{DAC}}(s)H(s)]|_{t=nT_s}\} \tag{2.65}$$

to finally derive that the actual order of the equivalent DT-ΣΔM increases, which decreases the maximum stable input amplitude and degrades the noise shaping performance [29].

This derivation is exemplarily completed here for the second-order CT-ΣΔM shown in Figure 2.27a [2, 29]. The equivalent DT loop filter of the first and second branch of $H(s)$ yields (with $f_s = 1$),

$$\begin{aligned} \frac{-k_1}{s^2} &\xrightarrow{\text{CT–DT}} \frac{-k_1(1-\tau_d)^2 z - k_1(1-\tau_d^2)}{2(z-1)^2} + z^{-1}\frac{-k_1\tau_d(2-\tau_d)z - k_1\tau_d^2}{2(z-1)^2} \\ \frac{-k_2}{s} &\xrightarrow{\text{CT–DT}} \frac{-k_2(1-\tau_d)}{z-1} + z^{-1}\frac{-k_2\tau_d}{z-1} \end{aligned} \tag{2.66}$$

where the z^{-1} in the second terms indicates the shift by one sample due ELD. Adding the equivalent DT terms of both modulator branches in Equation 2.66 gives:

$$H(z,\tau_d) = \frac{\alpha_2 z^2 + \alpha_1 z + \alpha_0}{2z(z-1)^2}, \text{ with } \begin{cases} \alpha_2 = -k_1(1-\tau_d)^2 - 2k_2(1-\tau_d) \\ \alpha_1 = -k_1(1+2\tau_d-2\tau_d^2) + 2k_2(1-2\tau_d) \\ \alpha_0 = -k_1\tau_d^2 + 2k_2\tau_d \end{cases} \tag{2.67}$$

Note that for $\tau_d = 0$, Equation 2.67 becomes the ideal DT loop filter $H(z)$ of a second-order modulator as shown in Figure 1.14a—see Equation 1.64. However, for $\tau_d \neq 0$, the order of the DT transfer function increases by 1, so that the equivalence in Equation 1.59 cannot be fulfilled with the original number of scaling coefficients (k_1 and k_2).

The detrimental effect of ELD on the modulator stability and dynamic range can be somewhat overcome by tuning the loop filter coefficients [29]. However, the restoration of the actual equivalence to the ideal DT loop filter under the influence of ELD requires introducing one more degree of freedom in the CT diagram—that is, one more coefficient.[11] The simplest approach to do this is the insertion of an additional feedback path scaled by k_c^* [31], as exemplary shown in Figure 2.28 for a second-order CT-ΣΔM. Note that the CT loop filter is modified to,

$$H_m(s) = -\frac{k_1^*}{s^2} - \frac{k_2^*}{s} - k_c^* \tag{2.68}$$

so that the $\mathcal{Z}$-transform of the modified loop filter

$$H_m(z,\tau_d) = \mathcal{Z}\{\mathcal{L}^{-1}[R_{DAC}^*(s)H_m(s)]|_{t=nT_s}\} \tag{2.69}$$

can then be mapped on the $\mathcal{Z}$-transform of the ideal loop filter

$$H(z) = \mathcal{Z}\{\mathcal{L}^{-1}[R_{DAC}(s)H(s)]|_{t=nT_s}\} \tag{2.70}$$

[11] If an RZ-DAC is used in the CT-ΣΔM, ELD shifts part of the feedback pulse into the next clock cycle only if the delay t_d exceeds the time slot between the end of the pulse and the end of the sample; for example, $t_d > T_s/2$. If that is not the case, the order of equivalent DT loop filter does not increase, but ELD leads to a deviation of the actual modulator coefficients from the desired original values; i.e., to a scaling mismatch that increases the in-band noise. However, in high-speed CT-ΣΔMs, ELD can become so significant that even RZ pulses get shifted to the next clock cycle, thus requiring an extra degree of freedom to compensate the effect [2].

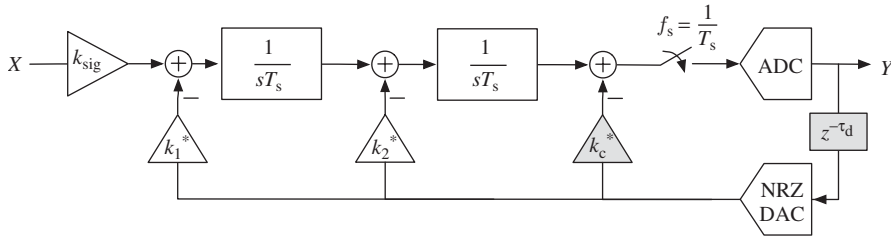


Figure 2.28 Classical approach for compensating excess loop delay in a second-order CT-ΣΔM.

to derive the value of coefficients k_i^*. For the second-order CT-ΣΔM in Figure 2.28, the scaling coefficients for ELD compensation can be found to be [2]:

$$k_1^* = k_1, \quad k_2^* = k_1\tau_d + k_2, \quad k_c^* = \frac{k_1\tau_d^2}{2} + k_2\tau_d \tag{2.71}$$

Note that the approach for ELD compensation illustrated in Figure 2.28 requires an additional DAC and a summing amplifier prior to the ADC. Other architectural alternatives have been proposed [29, 30, 32-34], although they share the same underlying principle of providing one extra degree of freedom. The interested reader can refer to [35] for a comparison of ELD compensation techniques and their extension to the case of cascaded CT-ΣΔMs.

2.11 Quantizer Metastability in CT-ΣΔMs

In Section 2.9, the sensitivity of CT-ΣΔMs to random clock jitter has been discussed considering its impact on the variation in the feedback charge. In spite of an ideally perfect sampling clock with no timing uncertainties, quantizer metastability can also introduce a statistical variation in the charge that is fed back. As real quantizers contain a regenerative stage with a finite regeneration gain, quantizer inputs with a magnitude near 0 will take longer to resolve than inputs with a large magnitude [19]. This nonideal effect is often referred to as *signal-dependent* delay, as opposed to *constant* excess delay due to signal propagation already discussed in Section 2.10. Figure 2.29 illustrates these different delay components for a real quantizer as a function of its input signal magnitude v_q.[12]

The variation of the comparison time with the signal level at the quantizer input affects the performance of a CT-ΣΔM in a way similar to clock jitter. As the quantizer input signal in a ΣΔM is ideally decorrelated from the modulator input, the times when the quantizer input is near 0 or has a large magnitude appear randomly over time [19]. This results in a random component in the feedback charge from one clock period to another, and thus to an increase of the modulator in-band noise.[13]

[12] Hysteresis component in practice means that the quantizer sometimes does not make a decision to change the output bit when it should. This nonideality is not further considered here because its influence on the performance of ΣΔ modulators is almost negligible [12].

[13] As shown in [19], quantizer metastability can also result in a considerable decrease of the modulator SNDR for small input levels.

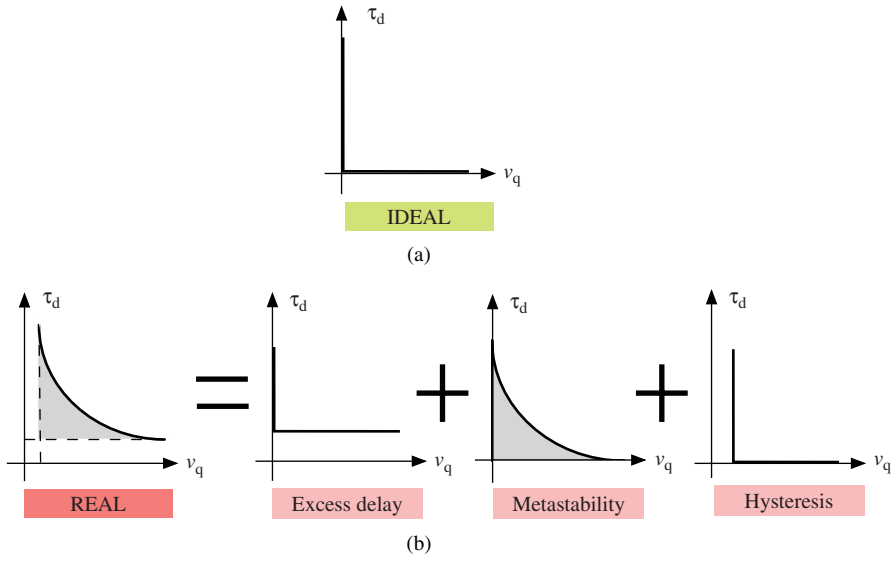


Figure 2.29 Quantizer delay versus input level: (a) ideal characteristic and (b) real characteristic as the addition of components due to constant excess delay, metastability, and hysteresis [19].

Although the effect of quantizer metastability on the modulator performance resembles that of random clock jitter, the architectural approaches to circumvent it are in practice closer to the analysis of ELD in Section 2.10. The easiest alternative consists of inserting a latching stage between the quantizer and the feedback DAC, which is clocked differently from the quantizer to provide the quantizer a constant time to resolve [19, 28]. Figure 2.30b illustrates the approach proposed in [19], in which signal-dependent delay is palliated by introducing a constant ELD of $\tau_d = 1/2$. The loop filter coefficients can thus be tuned for reducing the unfavorable effect of this fixed delay on the modulator stability and resolution. Figure 2.30c shows the architectural solution proposed in [30], in which a full delay is introduced before the DAC of the main loop whereas a half delay and an additional DAC establish a secondary loop to accommodate ELD—in an approach similar to that illustrated in Figure 2.28. Note that the proposed architecture provides enough degrees of freedom for modifying the CT loop filter and restoring the actual equivalence to the ideal DT-version of Figure 2.30a, so that both nonzero excess loop delay and signal-dependent delay are compensated in practice.

2.12 Finite Amplifier Gain in CT-ΣΔMs

Finite DC gain of amplifiers affects CT-ΣΔMs exactly in the same way as shown in Section 2.2 for DT-ΣΔMs. Let us consider an ideal active-RC integrator, in which the transfer function of the ith input path to the output is:

$$\mathrm{ITF}_i(s) = \frac{1}{sR_iC_I} = \frac{k_i f_s}{s} \tag{2.72}$$

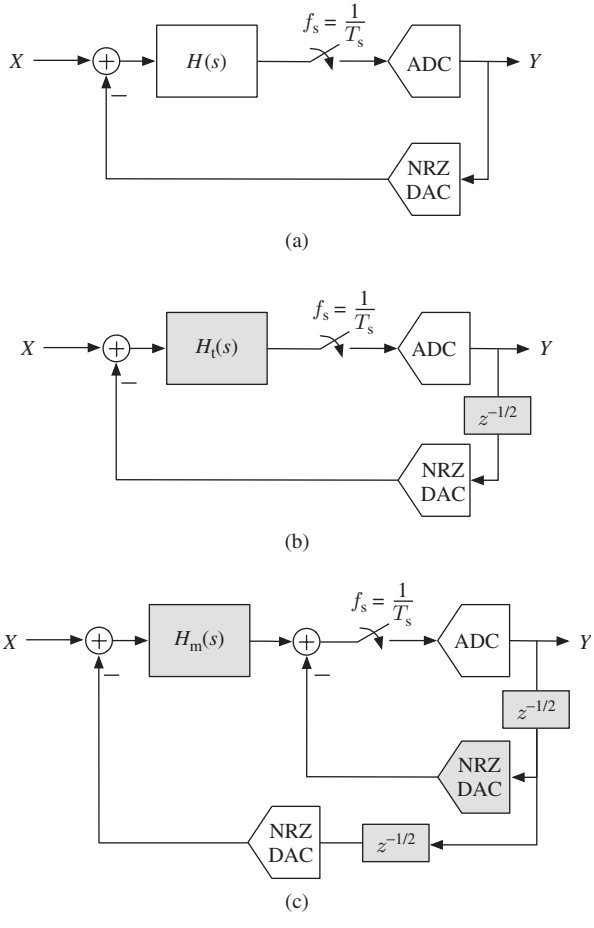


Figure 2.30 Towards the architectural compensation of timing errors in CT-ΣΔMs: (a) conventional architecture suffering from signal-dependent loop delay, (b) alternative architecture with extra half-clock delay and loop coefficients tuning [19] and (c) alternative architecture for compensating both excess loop delay and quantizer metastability [30].

If the finite amplifier gain A_v is accounted for in the CT integrator, as shown in Figure 2.31, the transfer function of the ith input path to the output can be derived as [2],

$$\mathrm{ITF}_{i,A_v}(s) = \frac{k_i f_s}{s\left(1+\frac{1}{A_v}\right)+\frac{1}{A_v}\sum_{i=1}^{N_i} k_i f_s} \approx \frac{k_i f_s}{s+\frac{1}{A_v}\sum_{i=1}^{N_i} k_i f_s} \tag{2.73}$$

i.e., as a single-pole transfer function with a DC gain of $A_v k_i / \sum_{i=1}^{N_i} k_i$ and a pole at $f_s \sum_{i=1}^{N_i} k_i / A_v$, which is displaced away from its ideal DC position.

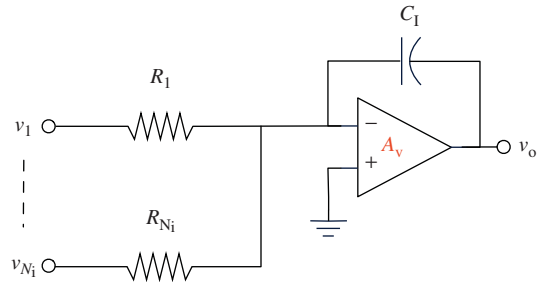


Figure 2.31 Active-RC integrator with N_i input paths and finite amplifier gain A_v.

Assuming that $\sum_{i=1}^{N_i} k_i \sim 1$ for all integrators in the CT-ΣΔM, the NTF degraded by finite amplifier gain can be derived following a procedure similar to that described in Section 2.2 for SC-ΣΔMs. The in-band noise of a second-order CT-ΣΔM as that shown in Figure 1.36 can be thus obtained as [2],

$$\mathrm{IBN}_2(A_v) \approx \frac{\Delta^2}{12}\frac{1}{(k_q k_{1,\mathrm{NRZ}})^2}\left(\frac{2\pi^2}{3{A_v}^2\mathrm{OSR}^3}+\frac{\pi^4}{5\mathrm{OSR}^5}\right) \tag{2.74}$$

which matches Equation 2.7 for DT implementations taking into account that the DT–CT equivalence yields $k_{1,\mathrm{NRZ}} = a_1 a_2$–see Equation 1.64.[14]

Similarly, the in-band noise of an Lth-order CT-ΣΔM with distributed feedback under the influence of leaky integrators yields [2],

$$\mathrm{IBN}_L(A_v) \approx \frac{\Delta^2}{12}\frac{1}{(k_q k_{1,\mathrm{NRZ}})^2}\left[\frac{L\pi^{2(L-1)}}{(2L-1){A_v}^2\mathrm{OSR}^{(2L-1)}}+\frac{\pi^{2L}}{(2L+1)\mathrm{OSR}^{2L+1}}\right] \tag{2.75}$$

which again matches Equation 2.9 if $k_{1,\mathrm{NRZ}} = \prod_{i=1}^{L} a_i$ is accounted for in the DT–CT equivalence.

Finally, if a 2-1-1 cascade CT-ΣΔM is considered, the in-band noise under the influence of finite amplifier gain yields [2],

$$\begin{aligned}\mathrm{IBN}_{211}(A_v) \approx{}& \frac{\Delta_1^2}{12}\frac{1}{(k_q k_{1,\mathrm{NRZ}})^2}\frac{4\pi^2}{3A_v^2\mathrm{OSR}^3}+\frac{\Delta_2^2}{12}\frac{1}{c_1^2}\frac{\pi^4}{5A_v^2\mathrm{OSR}^5}\\ &+\frac{\Delta_3^2}{12}\frac{1}{(c_1c_2)^2}\left(\frac{\pi^6}{7A_v^2\mathrm{OSR}^7}+\frac{\pi^8}{9\mathrm{OSR}^9}\right)\end{aligned} \tag{2.76}$$

which again matches Equation 2.10 of the equivalent SC-ΣΔM.

Note that the conclusions derived in Section 2.2 for the influence of finite amplifier gain on the performance of DT-ΣΔMs with either single-loop or cascade topologies are thus directly applicable to the CT case.

[14] Note that the notation adopted in Figure 1.36 [2] in terms of distributed feedback coefficients k_i facilitates the equivalence of expressions for the effect of loop filter nonidealities in both DT and CT domains.

2.13 Time-Constant Error in CT-ΣΔMs

As discussed in Section 2.3, deviations of the integrator gain coefficients from their nominal values due to variations in technological process parameters affect the performance of ΣΔMs differently depending on the modulator architecture. Single-loop topologies exhibit a quiet robust performance against integrator gain variations, whereas cascades suffer from the leakage of low-order-shaped quantization noise. However, the impact of this nonideality in SC-ΣΔMs is not critical, as integrator gains are implemented as capacitor *ratios* and thus make use of components with a *matching* better than 0.1% in present CMOS processes.

Conversely, as shown in Section 1.8, integrator gains in CT-ΣΔMs are mapped into $1/(RC)$ or g_m/C values, involving thus *absolute* component values. *Tolerances* of 10—20% are not unusual in present CMOS technologies under process and temperature variations, which increases the possible variation of integrator gains to more than 30% [2]. Although the nature of the nonideality is essentially the same as for DT-ΣΔMs—that is, an integrator gain error—the magnitude of its impact can be clearly foreseen to be considerably larger on CT-ΣΔMs.

If a tolerance ϵ_{RC} is accounted for in the time constant of an active-RC integrator, the transfer function of the ith input path to the output modifies to [2],

$$\mathrm{ITF}_{i,RC}(s) = \frac{1}{sR_iC_I(1+\epsilon_{RC})} = \frac{k_i}{(1+\epsilon_{RC})}\frac{f_s}{s} \tag{2.77}$$

which can be used for recomputing the NTF and the IBN of a particular CT-ΣΔM architecture under the influence of this nonideality—assuming a modulator linear model.

For an Lth-order CT-ΣΔM with distributed feedback, the IBN can be thus estimated as [2],

$$\mathrm{IBN}_L(\epsilon_{RC}) \approx \frac{\Delta^2}{12}\frac{1}{(k_q k_{1,NRZ})^2}\frac{\pi^{2L}}{(2L+1)\mathrm{OSR}^{2L+1}}\prod_{i=1}^{L}(1+\epsilon_{RCi})^2 \tag{2.78}$$

where ϵ_{RCi} refers to the time-constant error of the ith integrator. Note that the equation above is similar to Equation 2.13 for the influence of integrator gain errors in DT-ΣΔMs, taking into account that $k_{1,NRZ} = \prod_{i=1}^{L} a_i$ in the DT–CT equivalence.

For a 2-1-1 CT cascade, the IBN under the influence of time-constant errors yields [2],

$$\begin{aligned}\mathrm{IBN}_{211}(\epsilon_{RC}) \approx{}& \frac{\Delta_1^2}{12}\frac{1}{(k_q k_{1,NRZ})^2}\frac{\pi^4}{5\mathrm{OSR}^5}(\epsilon_{RC1}+\epsilon_{RC2})^2 + \frac{\Delta_2^2}{12}\frac{1}{c_1^2}\frac{\pi^6}{7\mathrm{OSR}^7}\epsilon_{RC3}^2 \\ &+ \frac{\Delta_3^2}{12}\frac{1}{(c_1c_2)^2}\frac{\pi^8}{9\mathrm{OSR}^9}(1+\epsilon_{RC4})^2\end{aligned} \tag{2.79}$$

which is again similar to Equation 2.79 for the equivalent SC-ΣΔM.

Figure 2.32 shows simulation results for the impact of time-constant errors on the performance of single-loop and cascade CT-ΣΔMs, and compares them with the approximated

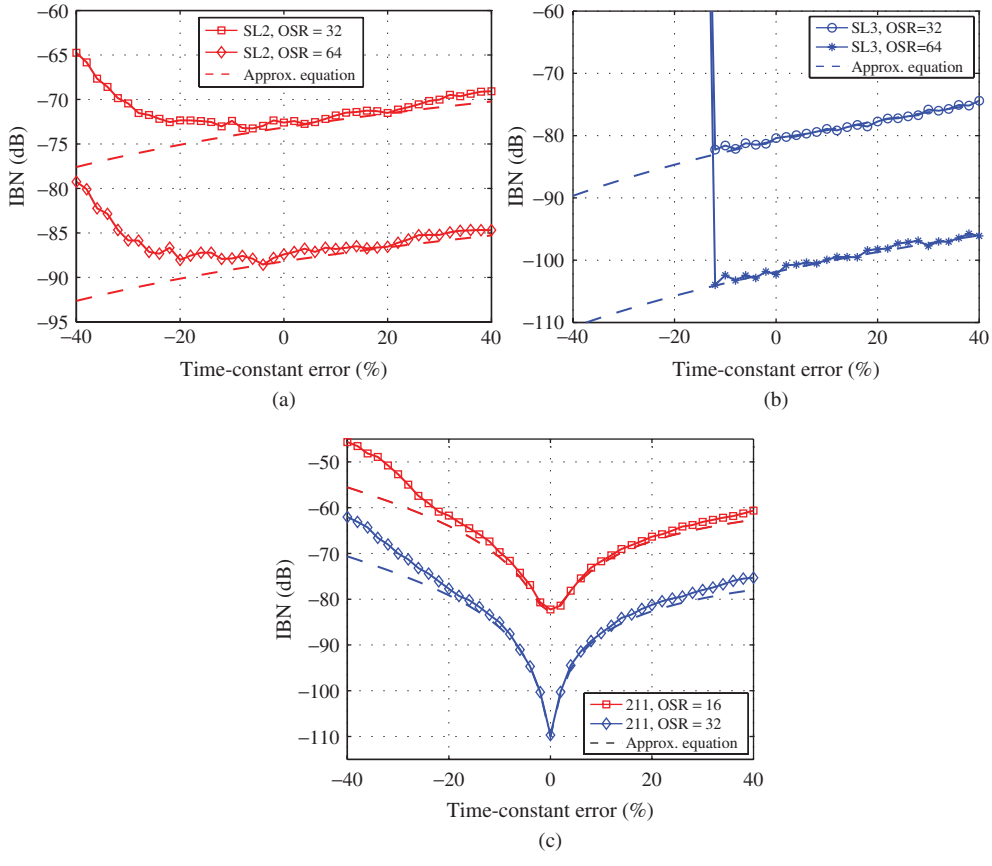


Figure 2.32 Simulation results for the in-band noise of CT-ΣΔMs under the influence of time-constant errors: (a) second-order single-loop, (b) third-order single-loop and (c) 2-1-1 cascade. Simulation results have been obtained for an input signal with $P_{\text{in}} = -20$ dBFS and $f_{\text{in}} = B_{\text{w}}/3$ and for an NRZ rectangular feedback. Approximated results have been obtained from Equations 2.78 and 2.79.

expressions in Equations 2.78 and 2.79.[15] Note from the simulation results that single-loop modulators are affected by time-constant errors in two different ways:

- For positive tolerances ($\epsilon_{\text{RC}} > 0$), the integrator gain coefficients k_{i} in Equation 2.77 are reduced–or, similarly, coefficients a_{i} in the equivalent DT-ΣΔM. This leads only to less aggressive noise shapings and thus to an increase of the IBN.

[15] To achieve accurate matching between analytical and simulation results for large error gains, the effective quantizer gain of single-bit ΣΔMs in Equation 1.32 has to be modified to [2],

$$k_{\text{q}} = \begin{cases} (1+\epsilon_1)/a_1, & \text{single-bit first-order } \Sigma\Delta\text{M} \\ 2(1+\epsilon_{\text{L}})/a_{\text{L}}, & \text{single-bit } L\text{th-order } \Sigma\Delta\text{M} \end{cases} \tag{2.80}$$

where ϵ_{L} is the error in the last integrator gain.

- For negative tolerances ($\epsilon_{RC} < 0$), integrator gain coefficients are increased, leading to more aggressive noise shapings. Therefore, after a short decrease of the IBN, negative time-constant errors reduce the modulator overload level and jeopardize its stability, as clearly visible for the third-order loop in Figure 2.32b.

The larger sensitivity of CT-ΣΔMs to component tolerances is also noticeable from Figure 2.32c. Note that the IBN of the 2-1-1 cascade with OSR = 32 increases nearly 30 dB for a time-constant error of 20%.

Concluding, the design of high-order single-loop CT-ΣΔMs is in practice more limited by their tendency to instability (for $\epsilon_{RC} < 0$) than by the degradation of the IBN (for $\epsilon_{RC} > 0$). Therefore, if no countermeasures are taken against the influence of gain errors, less aggressive noise shapings must be implemented using a set of scaling coefficients that are nonoptimal in terms of the modulator performance, but for which stability can be guaranteed over all the ϵ_{RC} possible variations [36, 37].

Alternatively, time-constant errors can be reduced in CT-ΣΔMs by tuning the absolute value of passive components with programmable banks of capacitors [30] or resistors [38].[16]

Conversely to tuning the analog filter, cascade architectures also offer the possibility of digitally correcting the error cancelation logic to reduce the power of low-order-shaped quantization noise that leaks to the modulator output [39]. Digital correction techniques have been successfully used for reducing the impact of time-constant errors in CT-ΣΔMs [40, 41].

2.14 Finite Integrator Dynamics in CT-ΣΔMs

The influence of finite amplifier GB can be easily incorporated into CT modulators by replacing the amplifier in Figure 2.31 by a single-pole model,

$$A(s) = \frac{A_v}{1 + \frac{s}{w_A}}, \quad \text{with} \quad GB = A_v \omega_A \tag{2.81}$$

where ω_A is the dominant pole of the amplifier inside the integrator. The integrator transfer function of the ith input path to the output is thus modified to [2],

$$ITF_{i,GB}(s) \approx \frac{k_i f_s}{s} \frac{\frac{GB}{GB + \sum_{i=1}^{N_i} |k_i f_s|}}{1 + \frac{s}{GB + \sum_{i=1}^{N_i} |k_i f_s|}} \tag{2.82}$$

where an integrator gain error arises due to the finite amplifier GB, as well as a second pole. For the sake of clarity, the modified ITF can thus be rewritten as,

$$ITF_{i,GB}(s) \approx \frac{k_i}{(1 + \epsilon_{GB})} \frac{f_s}{s} \frac{1}{\left(1 + \frac{s}{\omega_p}\right)}, \quad \text{with} \quad \begin{cases} \epsilon_{GB} = \sum_{i=1}^{N_i} \omega_{Ii} / GB \\ \omega_p = GB + \sum_{i=1}^{N_i} \omega_{Ii} \end{cases} \tag{2.83}$$

where $\omega_{Ii} = |k_i f_s|$.

[16] Transconductance tuning in gm-C filters and MOSFET tuning in MOSFET-C implementations are also widely used in CT analog filter design. However, they are not usually used in CT-ΣΔMs, because the nonlinearity associated with active components considerably limits the modulator performance [30].

Regarding Equation 2.83, the influence of the second integrator pole is often neglected to derive closed-form expressions for the influence of finite amplifier GB on CT-ΣΔMs. In that case, the expressions for the IBN due to finite GB can be approximated from the corresponding expressions for the gain error introduced by time-constant variations—Equations 2.78 and 2.79—just by replacing ϵ_{RC} with the expression for ϵ_{GB} in Equation 2.83. The observations already made in Section 2.13 on the different sensitivity of single-loop and cascade architectures and on possible correction techniques also apply here.

The former approximation can a priori be considered too coarse, but it provides results that are in accordance to the simulation of NRZ CT-ΣΔMs if the amplifier GB is not very low (GB $\geq f_s$) [2].

However, finite amplifier GB is a typical nonideality concerning the dynamics of a ΣΔM and, as such, by changing the shape and the dynamic behavior of the feedback pulses, its influence on the modulator performance is expected to change [2]. Conversely, not much published work exists that extends the basic single-pole model in Equation 2.81 and/or applies it to different feedback pulse forms [42]. Even less literature exists for the influence of the finite amplifier SR on the performance of CT-ΣΔMs and most designs strongly rely on simulations. The reason for that may be found in the fact that, generally speaking, CT-ΣΔMs can work with a finite amplifier GB and SR in the integrators lower than SC implementations (see Section 2.4.2 and 2.4.3), due to the lack of high-current peaks of the latter.

2.15 Circuit Noise in CT-ΣΔMs

As discussed in Section 2.5, circuit noise strongly impacts the in-band noise of SC-ΣΔMs due to the aliasing of broadband noise components that are sampled together with the modulator input signal. Conversely, sampling capacitors are not used in purely CT-ΣΔMs, so that kT/C noise is not present, which is a clear advantage. Broadband noise components are thus filtered before they get to the sampler, so that the foldover effect is attenuated by the modulator noise shaping [2].

Let us consider the active-RC integrator in Figure 2.33 to be the front-end integrator of a CT-ΣΔM. Two input branches are considered: one for the modulator input signal (v_{in}) and

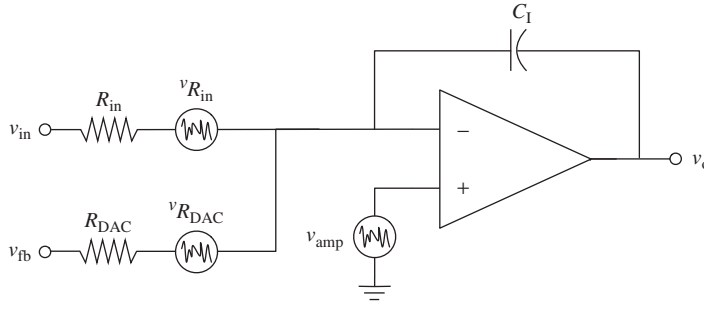


Figure 2.33 Circuit noise sources in a CT active-RC integrator with two input paths. Single-ended version.

the other for the DAC feedback signal (v_{fb}). Resistors thermal noise and amplifier noise have been incorporated to the scheme in Figure 2.33.[17] On the one hand, each resistor has a noise voltage source v_{R_i} in series, whose PSD in a single-sided representation is thus

$$S_{R_i} = 4kTR_i \tag{2.84}$$

where k is Boltzmann's constant and T is the absolute temperature. On the other hand, the voltage source v_{amp} represents the amplifier input noise. As illustrated in Figure 2.16, it is basically determined by a broadband thermal component and a narrowband $1/f$ component, so that $S_{amp}(f)$ is given by Equation 2.36.

Adding up the former circuit noise components in the CT integrator, the total input-referred noise PSD yields [25, 26]

$$S_{noise,in}(f) \approx 2\left(S_{R_{in}} + S_{R_{DAC}}\frac{R_{in}^2}{R_{DAC}^2}\right) + S_{amp}(f)\left[(2\pi f R_{in}C_I)^2 + \left(1+\frac{R_{in}}{R_{DAC}}\right)^2\right] \tag{2.85}$$

where the factor 2 before the brackets accounts for the actual fully-differential implementation of the CT integrator, in which the number of resistors doubles compared with the single-ended scheme in Figure 2.33. Replacing Equations 2.84 and 2.36 in 2.85, the total input-referred noise PSD of the front-end integrator of the CT-ΣΔM can thus be approximated to:

$$\begin{aligned} S_{noise,in}(f) \approx\ & 8kT\left(R_{in} + R_{DAC}\frac{R_{in}^2}{R_{DAC}^2}\right) \\ & + S_{amp,th}\left(1+\frac{f_{cr}}{f}\right)\left[(2\pi f R_{in}C_I)^2 + \left(1+\frac{R_{in}}{R_{DAC}}\right)^2\right] \end{aligned} \tag{2.86}$$

The input-referred IBN of a CT-ΣΔM due to circuit noise can be easily obtained by integrating the former equation over the input signal bandwidth, yielding

$$\begin{aligned} IBN_{noise} =\ & \int_0^{B_w} S_{noise,in}(f)\,df \approx 8kTR_{in}B_w\left(1+\frac{R_{in}}{R_{DAC}}\right) \\ & + S_{amp,th}\left[B_w + f_{cr}\ln\left(\frac{B_w}{f_0}\right)\right]\left(1+\frac{R_{in}}{R_{DAC}}\right)^2 \end{aligned} \tag{2.87}$$

in which the $1/f$ noise component has been integrated from a frequency $f_0 > 0$ to exclude DC. Taking into account that $R_{in}C_I = 1/f_s = 1/(2B_w\text{OSR})$, resulting terms that are proportional to OSR^{-2} have been neglected in the final calculation [2].

For a CT-ΣΔM to achieve a given noise performance, the sum of all three components in Equation 2.87 have to meet the demanded noise floor. Note that, to reduce

[17] The noise associated with the DAC reference voltage is not considered here for simplicity, but it can be incorporated in a similar way to the amplifier noise [25, 26].

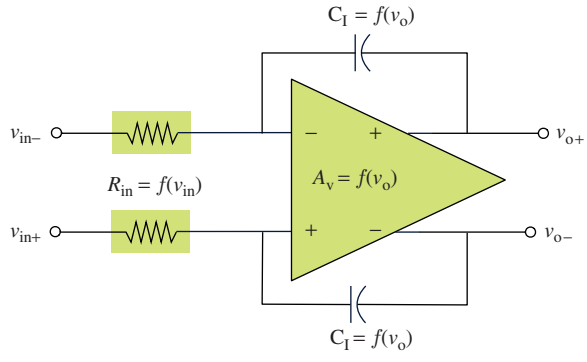


Figure 2.34 Main sources of distortion in a fully-differential active-RC integrator.

the contribution of the resistors' thermal noise, the integrator resistance at the modulator input must be decreased, which results in a larger power consumption, given that $f_s = 1/(R_{in}C_I)$. To reduce the amplifier noise, its transconductance can be increased (for the white noise component) and the size of the input transistors can be increased (for the flicker component).

2.16 Sources of Distortion in CT-ΣΔMs

The linearity of a CT-ΣΔM is ultimately limited by that of the input stage and by signal-dependent errors arising from the feedback DAC. Both sources of distortion are briefly discussed next.

2.16.1 Nonlinearities in the Front-End Integrator

As stated in Section 2.7, the linearity of a ΣΔ modulator is ultimately limited by the nonlinearities associated with the first integrator. Therefore, for medium- and high-resolution CT-ΣΔMs, active-RC integrators are usually employed at the modulator front end rather than Gm-C integrators [2, 25, 43], because a more linear voltage-to-current conversion can be obtained in practice with resistors than with active devices.

Similarly as done in Figure 2.18 for SC implementations, Figure 2.34 illustrates the main sources of distortion in an active-RC integrator, in which the amplifier usually plays the dominant role, due to the high linearity achieved by passive devices in present mixed-mode technological processes. On the contrary, the DC gain of an amplifier exhibits in practice a high dependency on the common-mode output voltage due to degradation of its output impedance, as illustrated in Figure 2.19. Moreover, nonlinearities in the differential input pair of the amplifier are accentuated by the finite DC gain. In [43] it is shown how a small residue differential voltage at the input pair due to finite gain is modulated and the third-order harmonic distortion is derived as

$$HD_3 \approx \frac{1}{64 g_m R_{in}^3 I_B^2}\left(1 + \frac{R_{in}}{R_{DAC}}\right) A_{in}^2 \tag{2.88}$$

where R_{in} is the input resistor, R_{DAC} is the feedback resistor, and g_m and I_B are, respectively, the transconductance and the bias current of the input transistors of the amplifier.

Note from the equation earlier that the modulator linearity can be considerably improved by increasing R_{in} as much as the thermal noise limit allows–see Equation 2.87. Although with lower efficiency, a larger amplifier input transconductance also benefits linearity and at the same time helps to reduce thermal noise.

2.16.2 *Intersymbol Interference in the Feedback DAC*

In practical realizations of CT-ΣΔMs, besides timing errors due to emerging excess loop delay (see Section 2.10) and nonlinearities arising from mismatches between the output levels of multibit DACs (see Section 1.6), the implemented feedback DAC also exhibits SR limitations as well as different rise and fall times. These nonidealities cause additional errors that directly add to the modulator input and tend to degrade the performance, unless they are made small enough.

For the case of CT-ΣΔMs employing an NRZ-DAC, these nonidealities introduce intersymbol interference (ISI) [44], as illustrated in Figure 2.35a. Note that any limitation on the DAC SR or mismatch between its rise and fall times makes the area of one 1-symbol differ from half of the area of two consecutive 1-symbols. Therefore, the resulting errors in the feedback charge depend on the modulator output sequence and thus create a signal-dependent distortion that limits the modulator performance.

The most common solution to avoid ISI is to resort to RZ feedback pulses, so that, in spite of different rise and fall times in the DAC response, the error remains constant in every clock period and independent of the output sequence, as shown in Figure 2.35b. This way, the resulting error can be compensated by tuning the modulator scaling coefficients [2], similarly to the compensation of pure ELD already discussed in Section 2.10.

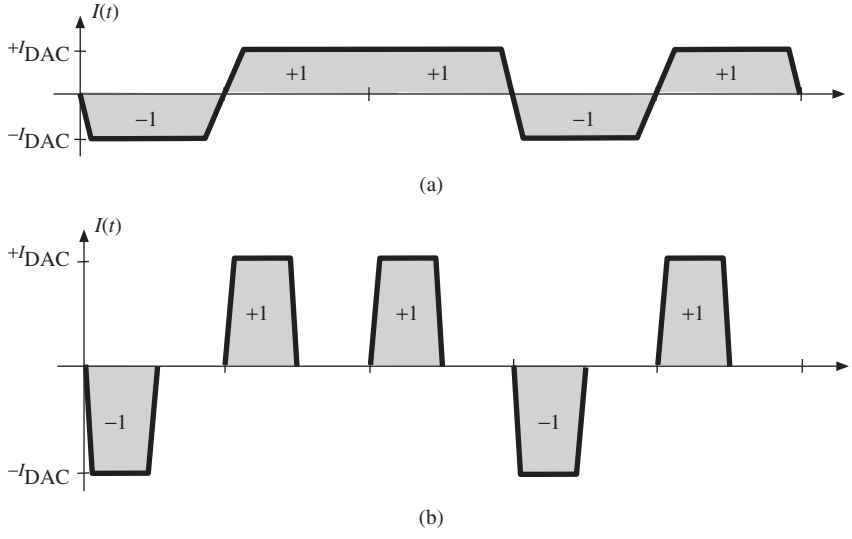


Figure 2.35 Influence of the DAC slew rate limitation on the feedback charge for: (a) NRZ DAC, causing a signal-dependent charge error (ISI) and (b) RZ DAC, causing a constant charge error.

2.17 Case Study: High-Level Sizing of a ΣΔM

For the sake of illustration, this section provides guidelines on how to face the design of a ΣΔ modulator to fulfill a given set of specifications by means of the closed-form expressions and the behavioral models previously derived in the chapter. The impact of the different circuit nonidealities on the ΣΔM performance will be considered in a cumulative way, trying to map the modulator specifications onto the electrical requirements of its main building blocks in a *top-down design methodology*–often referred to as the *high-level sizing* of a ΣΔM.

This approach will be exemplified on a 2-1-1 cascade SC-ΣΔM operating at 100-MHz clock frequency intended to achieve an effective resolution of around 12bit in the A/D conversion of low-pass signals with 4-MHz bandwidth. To that purpose, let us consider the Z-domain block diagram of a 2-1-1 DT cascade illustrated in Figure 1.23, together with the modulator scaling coefficients in the first column of Table 1.2 [6]. Let us also assume a modulator full-scale range of 2V.

Ideal Modulator Performance

The ideal in-band quantization error of the 2-1-1 cascade ΣΔM can be particularized from Equation 1.44 to:

$$\mathrm{IBN_Q} = \frac{\Delta_3^2}{12}\frac{1}{(c_1c_2)^2}\frac{\pi^8}{9\mathrm{OSR}^9} \tag{2.89}$$

If single-bit quantization is used in the three cascaded stages ($B_1 = B_2 = B_3 = 1$), Equation 2.89 yields an $\mathrm{IBN_Q}$ of −67.2 dB for $\Delta_3 = 2$ (as $B_3 = 1$) and OSR = 12.5 (as $f_s = 100$ MHz and $B_w = 4$ MHz). Therefore, an oversampling ratio of 12.5 is clearly insufficient only in combination with a fourth-order shaping for the cascade to achieve the required IBN; e.g., −83 dB for a 13-bit ENOB, as obtained from Equations 1.19 and 1.20.

Taking advantage of the easiness to employ dual-quantization schemes in cascade ΣΔMs (see Section 1.6.3), $\mathrm{IBN_Q}$ can be easily reduced by resorting to a multibit quantizer in the last stage. For $B_3 = 3$, the last-stage quantization step reduces to $\Delta_3 = 2/(2^3 - 1)$ and $\mathrm{IBN_Q}$ thus yields −84.1 dB. Figure 2.36 illustrates the spectra of the ΣΔ cascade for an input sinusoid of −6 dBFS at 1.33 MHz ($f_{in} = B_w/3$). The figure compares the spectrum of the overall output of the fourth-order multibit cascade (labeled as 211mb) with those of its partial second- and third-order single-bit outputs (labelled as SL2 and 21, respectively). The increase in the shaping order is evident from the slopes of the spectra, whereas the reduction of the quantization error power for the multibit case is noticeable at the high-frequency region.

Figure 2.37 illustrates a possible implementation of the 2-1-1 multibit ΣΔM using fully-differential SC techniques. Note that only the mixed-signal section of the cascade is shown, whereas the required digital filtering to combine the stage outputs (Y_1, Y_2, and Y_3) and to ideally cancel the lower-order quantization errors is omitted–see Figure 1.23. Note also that the stages would operate with a differential reference voltage $V_{ref} = V_{ref+} - V_{ref-} = 1\mathrm{V}$ to obtain the 2-V differential full-scale range. The values of the sampling and integrating capacitors in Figure 2.37 are expressed as multiples of a unit capacitor C_u, whose ratios implement the modulator scaling coefficients in the first column of Table 1.2 [6]. To

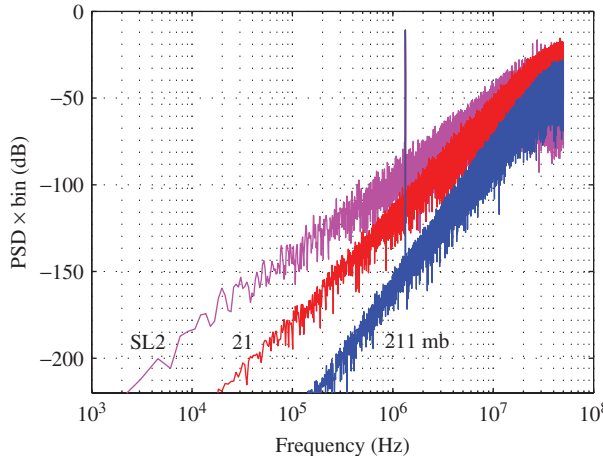


Figure 2.36 Illustration of the second-, third-, and fourth-order shaping in the 2-1-1 ΣΔM. Quantization noise in the three ΣΔ stages is considered as the only source of error. Input signal with $P_{in} = -6$ dBFS and $f_{in} = B_w/3$.

adapt them to the multibit quantization in the last stage, the corresponding coefficients are doubled ($g_4 = 2$, $g_4^{'} = g_4^{''} = 1$ in the multibit case).

Noise Leakages

As discussed in Section 1.6.1, errors in the multibit DAC will be mostly determined by the mismatching between its unit elements, which would be resistors for the SC-ΣΔM in Figure 2.37. From Equation 1.48 worst-case errors in the 3-bit DAC output can be estimated as,

$$\epsilon_{DAC} \approx \frac{1}{2\sqrt{2^{B_3}}}(3\sigma_R) \tag{2.90}$$

where σ_R represents the relative error in the value of the unit resistor and three sigmas are considered. Thanks to the dual-quantization scheme these errors are injected at the last-stage input and will be third-order shaped at the modulator output, so that their contribution to the total IBN can be estimated from Equation 1.50 as:

$$IBN_{DAC} \approx \epsilon_{DAC}^2 \frac{1}{(c_1 c_2)^2} \frac{\pi^6}{7OSR^7} \tag{2.91}$$

Assuming a standard deviation of 0.1% in the unit resistors–which is a reasonable value in present-day mixed-signal CMOS processes–IBN_{DAC} equals −113.6 dB; that is, it is negligible compared with $IBN_Q = -84.1$ dB.

As discussed throughout this chapter, cascade ΣΔMs are especially sensitive to leakages of low-order quantization errors due to circuit-level nonidealities. For the case of SC ΣΔ cascades, the most critical error mechanisms are the finite gain of the amplifiers and the mismatching between the unit capacitors used for implementing the modulator coefficients. Figure 2.38 illustrates the impact of noise leakages on the output spectrum of the 2-1-1

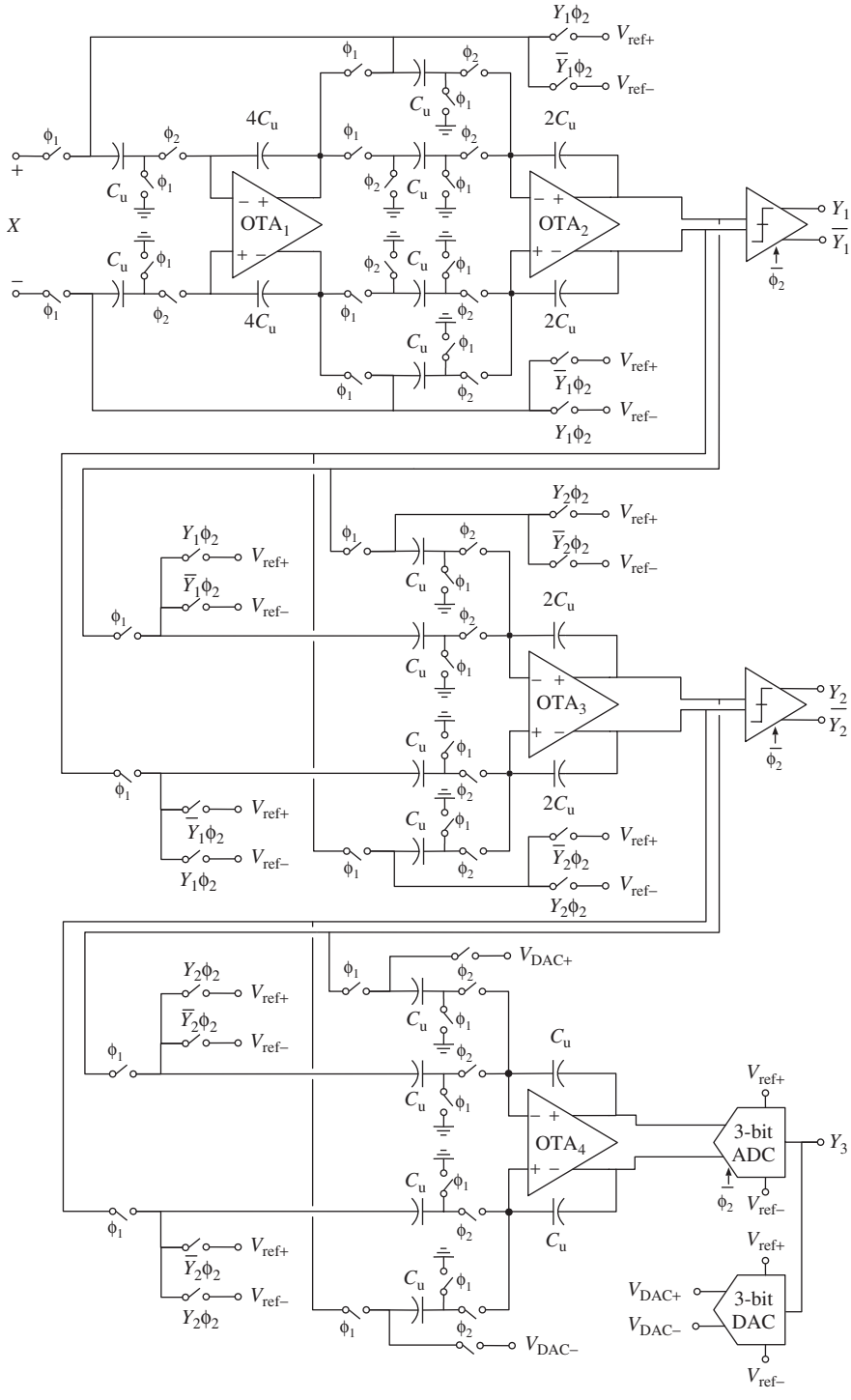


Figure 2.37 Fully-differential SC implementation of the 2-1-1 multibit ΣΔ modulator.

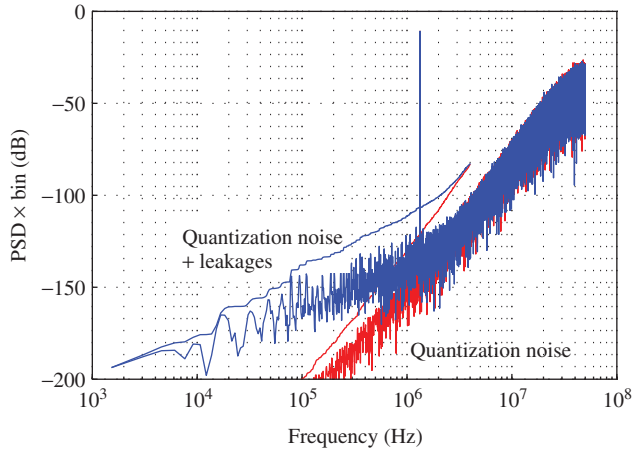


Figure 2.38 Effect of noise leakages on the modulator output spectrum ($A_v = 65$ dB, $\sigma_C = 0.1\%$, $\sigma_R = 0.1\%$). Input signal with $P_{in} = -6$ dBFS and $f_{in} = B_w/3$.

multibit ΣΔM for a DC gain of 65 dB in the amplifiers and a capacitor mismatch of 0.1%, in addition to the multibit DAC errors. Results are compared with a purely ideal behavioral simulation only considering quantization errors (as done in Figure 2.36). The integrated error powers versus frequency are also included for comparison purposes. Note that, in spite of the degradation of the ideal shaping performance due to noise leakages, the resulting error power integrated in the 4-MHz bandwidth (−82.6 dB) is only slightly larger than the simulated IBN_Q (−84.1 dB). The influence of both error mechanisms can be separately estimated from Equations 1.19 and 1.20 as:

$$IBN_{A_v} \approx \frac{\Delta_1^2}{12}\frac{1}{(k_q a_1 a_2)^2}\frac{4\pi^2}{3A_v^2 OSR^3} + \frac{\Delta_2^2}{12}\frac{1}{c_1^2}\frac{\pi^4}{5A_v^2 OSR^5} + \frac{\Delta_3^2}{12}\frac{1}{(c_1 c_2)^2}\frac{\pi^6}{7A_v^2 OSR^7} \quad (2.92)$$

$$IBN_{\sigma_C} \approx \frac{\Delta_1^2}{12}\frac{1}{(k_q a_1 a_2)^2}\frac{\pi^4}{5OSR^5}(2\cdot 3\sigma_C)^2 + \frac{\Delta_2^2}{12}\frac{1}{c_1^2}\frac{\pi^6}{7OSR^7}(3\sigma_C)^2 \quad (2.93)$$

The increase in the modulator IBN due to integrator leakages can thus be estimated as −91.3 dB for $A_v = 65$ dB, whereas that due to capacitor mismatch equals −91.0 dB for $\sigma_C = 0.1\%$.[18]

The former equations can also be employed to evaluate the influence of noise leakages on the performance of the 2-1-1 ΣΔM under different clock rates and multibit quantizations. This is illustrated in Figure 2.39, which shows the effective resolution of the ΣΔ cascade against B_3 for varying OSR. Note that the ENOB curves tend to saturate in the presence of nonidealities, leading to a practical useful limit of multibit quantization. However, for the values assumed for the nonideal parameters A_v, σ_C, and σ_R, the initial

[18] Note that the closed-form expressions in Equations 2.91, 2.92, and 2.93 provide an accurate estimation of the increase in the modulator IBN, because the result obtained from $IBN_Q + IBN_{DAC} + IBN_{A_v} + IBN_{\sigma_C}$ (−84.1 dB − 113.6 dB − 91.3 dB−91.0 dB = −82.7 dB) is in great accordance with behavioral simulation results in Figure 2.38 (−82.6 dB).

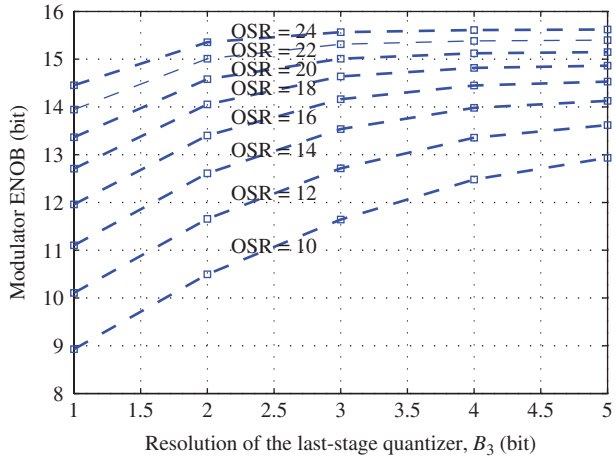


Figure 2.39 Modulator effective resolution versus the number of bits in the last-stage quantizer for varying OSR. All noise leakages are considered ($A_v = 65$ dB, $\sigma_C = 0.1\%$, $\sigma_R = 0.1\%$).

selection of $B_3 = 3$ for OSR $= 12.5$ provides a good trade-off for achieving the targeted modulator performance.[19]

Circuit Noise

As discussed in Section 2.5, the contribution of circuit noise to the overall IBN is dominated by the front-end integrator. Only kT/C noise and amplifier thermal noise will be considered in this high-level sizing of the 2-1-1 SC-ΣΔM.

On the one hand, the kT/C noise budget determines the minimum value of the sampling capacitor (C_S) to be used. A small value will be selected to reduce the capacitive load of the integrators and, hence, their settling requirements. This noise contribution can be estimated from Equation 2.40 as:

$$\mathrm{IBN}_{\mathrm{kT/C}} \approx \frac{4kT}{C_S \mathrm{OSR}} \tag{2.94}$$

For $C_S = C_u = 0.4$ pF in the front-end integrator (Figure 2.37), $\mathrm{IBN}_{\mathrm{kT/C}}$ equals -84.8 dB; that is, it yields an in-band contribution similar to that of the ideal quantization noise.[20]

On the other hand, the contribution of the amplifier thermal noise can be estimated from Equation 2.40 as,

$$\mathrm{IBN}_{\mathrm{amp,th}} \approx S_{\mathrm{amp,th}} \frac{\mathrm{GB}_{\phi_2}}{4\mathrm{OSR}} \tag{2.95}$$

which requires knowing a priori the amplifier gain-bandwidth product to quantify the foldover effect. The required GB_{ϕ_2} can be estimated by means of considering the integrator

[19] Note that a selection of $B_3 = 2$ for OSR $= 14$ ($f_s = 112$ MHz) leads to a similar modulator ENOB under the influence of the same nonidealities.

[20] A similar budget is often allocated for IBN_Q and $\mathrm{IBN}_{\mathrm{kT/C}}$ to optimize the power consumption of SC-ΣΔMs.

settling error as a linear gain error, as discussed in Section 2.4.2. Therefore, the in-band noise due to limited linear dynamics[21] can be obtained similarly to Equation 2.96 as,

$$\mathrm{IBN}_{\mathrm{st}} \approx \frac{\Delta_1^2}{12}\frac{1}{(k_q a_1 a_2)^2}\frac{\pi^4}{5\mathrm{OSR}^5}(2\epsilon_{\mathrm{st}})^2 + \frac{\Delta_2^2}{12}\frac{1}{c_1^2}\frac{\pi^6}{7\mathrm{OSR}^7}(\epsilon_{\mathrm{st}})^2 \tag{2.96}$$

where ϵ_{st} represents the gain error, given by:

$$\epsilon_{\mathrm{st}} \approx \mathrm{e}^{-\pi\frac{\mathrm{GB}_{\phi_2}(\mathrm{Hz})}{f_s}} \tag{2.97}$$

For $\mathrm{GB}_{\phi_2} \approx 2.1 f_s$ (around 200 MHz), $\mathrm{IBN}_{\mathrm{st}}$ equals -96.8 dB. This value is considerably smaller than $\mathrm{IBN}_{\mathrm{Q}}$ and $\mathrm{IBN}_{\mathrm{kT/C}}$ to have some margin in the noise budget for allocating more accurate settling error estimations in the next steps.

Based on this value of GB_{ϕ_2} for settling considerations, the input-referred thermal noise of the amplifier can be estimated for a given noise budget. For $S_{\mathrm{amp,th}} = (6\ \mathrm{nV}/\sqrt{\mathrm{Hz}})^2$, $\mathrm{IBN}_{\mathrm{amp,th}}$ equals -90.3 dB. The total in-band white noise will be thus dominated by kT/C noise and can be obtained to equal -83.7 dB.

Settling Error

Assuming that the parasitics at the amplifier input and output nodes are $C_{\mathrm{P}} = 0.1$ pF and $C_{\mathrm{L}} = 0.5$ pF respectively, the equivalent load at the amplifier output during integration can be obtained from Equation 2.16

$$C_{\mathrm{eq},\phi_2} = C_{\mathrm{P}} + C_{\mathrm{S}} + C_{\mathrm{L}}\left(1 + \frac{C_{\mathrm{P}} + C_{\mathrm{S}}}{C_{\mathrm{I}}}\right) \tag{2.98}$$

to be around 1.2pF. Formerly, the required amplifier GB was estimated considering only a linear settling error to be $\mathrm{GB}_{\phi_2} = 2.1 f_s$. According to Equation 2.27, the amplifier transconductance for a dominant-pole model yields around 1.5mA/V.

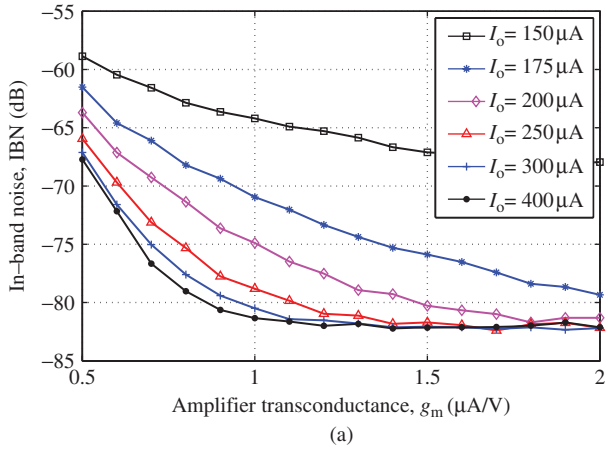
Figure 2.40 shows simulation results for the considered 2-1-1 ΣΔM using the behavioral model for the transient response of SC integrators presented in Section 2.4.1, which accurately accounts for both linear and nonlinear integrator dynamics. The simulated modulator IBN against the amplifier transconductance is shown in Figure 2.40a for different values of the amplifier output current. Note that, for $g_{\mathrm{m}} = 1.5$ mA/V, the modulator performance is not limited by nonlinear integrator dynamics as long as $I_{\mathrm{o}} \geq 250$ μA. According to Equation 2.27, the required amplifier SR can be estimated as

$$\mathrm{SR}_{\phi_2}(\mathrm{V/s}) = \left(1 + \frac{C_{\mathrm{P}} + C_{\mathrm{S}}}{C_{\mathrm{I}}}\right)\frac{\mathrm{S_o}}{C_{\mathrm{eq},\phi_2}} \tag{2.99}$$

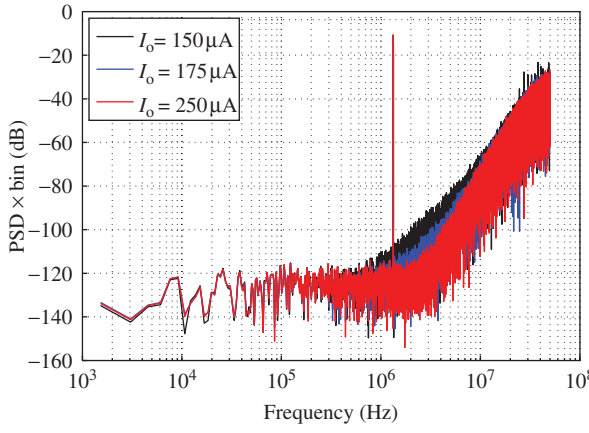
to be around 275V/μs.

Figure 2.40b illustrates the effect of the amplifier limited output current on the modulator output spectra for a -6 dBFS@1.33 MHz input tone. Note that, for $I_{\mathrm{o}} \leq 250$ μA, the limited amplifier SR clearly degrades the shaping performance of the modulator.

[21] Assuming thus nonlimited nonlinear settling; i.e., infinite SR.



(a)



(b)

Figure 2.40 Influence of the amplifier dynamics on the modulator performance: (a) IBN against the amplifier transconductance for different values of the amplifier output current and (b) output spectrum for $g_m = 1.5$ mA/V and different amplifier output currents. Input signal with $P_{in} = -6$ dBFS and $f_{in} = B_w/3$. Noise leakages and thermal noise are also accounted for in the behavioral simulations.

Overall High-Level Sizing and Noise Budget

Table 2.1 summarizes the electrical specifications of the main modulator blocks derived in the presented high-level sizing of the 2-1-1 SC-ΣΔM. Accordingly, Table 2.2 shows the selected noise budget for achieving the targeted resolution of 12 bit at 4-MHz conversion bandwidth. According to the closed-form expressions used, the total in-band noise is −80.1 dB, leading to a DR of 77.1 dB (12.5 bit).

Note from Table 2.1 that the derived requirements apply for the front-end integrator, whereas those of the remaining integrators can be usually scaled down to some extent. Extensive behavioral simulations are often performed for this purpose.

Table 2.1 High-level sizing of the 2-1-1 SC-ΣΔM

Modulator	Topology	2-1-1
	Dual quantization, B_1, B_2, B_3	1 bit, 1 bit, 3 bit
	Signal bandwidth, B_w	4 MHz
	Clock frequency, f_s	100 MHz
	Oversampling ratio, OSR	12.5
	Reference voltage, V_{ref}	1 V
Front-end integrator	Unit capacitor, C_u	0.4 pF
	Capacitor standard deviation, σ_C	0.1%
	Sampling capacitor, C_S	0.4 pF
	Integration capacitor, C_I	1.6 pF
Front-end amplifier	DC gain, A_v	65 dB
	Input capacitor, C_P	0.1 pF
	Output capacitor, C_L	0.5 pF
	Equivalent output load, C_{eq,ϕ_2}	1.2 pF
	Transconductance, g_m	1.5 mA/V
	Gain-bandwidth product, GB_{ϕ_2}	206.5 MHz ($= 2.1 f_s$)
	Slew rate, SR_{ϕ_2}	275 V/μs ($= 2.7 V_{ref} f_s$)
	Input-referred thermal noise, $S_{amp,th}$	$(6\ nV/\sqrt{Hz})^2$
	Output swing	$\pm V_{ref}$
Last-stage quantizer	Resistor standard deviation, σ_R	0.1%

Table 2.2 Noise budget for the high-level sizing of the 2-1-1 SC-ΣΔM proposed in Table 2.1

Quantization noise			**−84.1 dB**
	Multibit DAC error	−113.6 dB	
	Integrator leakage	−91.3 dB	
	Capacitor mismatch	−91.0 dB	
Total noise leakages			**−88.1 dB**
Settling error (linear)			**−96.8 dB**
	kT/C noise	−84.8 dB	
	Amplifier thermal noise	−90.3 dB	
Total thermal noise			**−83.7 dB**
Total in-band noise			**−80.1 dB**

Note also that the generated distortion has not been strictly evaluated, so that the derived values for the gain and SR of the front-end amplifier may increase depending on the involved nonlinearities not to degrade the modulator SNDR.

However, the requirements obtained for the modulator blocks can be used as the starting point for their electrical design. A *bottom-up* approach is then often followed to ensure the fulfillment of the targeted modulator specifications.

2.18 Summary

In this chapter, the main error mechanisms degrading the performance of SC- and CT-ΣΔMs have been analyzed. These errors are caused by nonidealities affecting the circuit implementation of the modulator analog blocks and produce extra error components that add to the in-band quantization error noise that can severely limit the modulator performance. Therefore, nonidealities have been studied in detail and behavioral models and, if possible, practical closed-form expressions have been presented to estimate their influence on the ΣΔM behavior.

Errors modifying the noise transfer function of the ΣΔM have been analyzed, such as the finite amplifier gain, capacitor mismatch, and integrator settling for SC-ΣΔMs, and excess loop delay and time-constant error for CT-ΣΔMs. As shown, single-loop modulator topologies are less sensitive to these errors than cascades, which suffer from noise leakage.

Nonidealities whose effect can be modeled as an additive noise source at the modulator input have been studied, such as circuit noise and clock jitter in both SC- and CT-ΣΔMs. The main sources of distortion in both types of modulator implementation have also been addressed.

References

[1] S. R. Norsworthy, R. Schreier, and G. C. Temes, *Delta-Sigma Data Converters: Theory, Design and Simulation*, IEEE Press, 1997.

[2] M. Ortmanns and F. Gerfers, *Continuous-Time Sigma-Delta A/D Conversion: Fundamentals, Performance Limits and Robust Implementations*, Springer, 2006.

[3] R. del Río, F. Medeiro, B. Pérez-Verdú, J. M. de la Rosa, and A. Rodríguez-Vázquez, *CMOS Cascade ΣΔ Modulators for Sensors and Telecom: Error Analysis and Practical Design*, Springer, 2006.

[4] A. Marques, V. Peluso, M. S. Steyaert, and W. M. Sansen, "Optimal Parameters for ΔΣ Modulator Topologies," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. **45**, pp. 1232–1241, September 1998.

[5] Y. Geerts, M. Steyaert, and W. Sansen, *Design of Multi-bit Delta-Sigma A/D Converters*, Kluwer Academic Publishers, 2002.

[6] R. del Río, F. Medeiro, J. M. de la Rosa, B. Pérez-Verdú, and A. Rodríguez-Vázquez, "Reliable Analysis of Settling Errors in SC Integrators: Application to ΣΔ Modulators," *IET Electronics Letters*, vol. **36**, pp. 503–504, March 2000.

[7] J. Fischer, "Noise Sources and Calculation Techniques for Switched Capacitor Filters," *IEEE Journal of Solid-State Circuits*, vol. **17**, pp. 742–752, August 1982.

[8] C.-A. Gobet and A. Knob, "Noise Analysis of Switched Capacitor Networks," *IEEE Transactions on Circuits and Systems*, vol. **CAS-30**, pp. 37–43, January 1983.

[9] R. Gregorian and G. C. Temes, *Analog MOS Integrated Circuits for Signal Processing*, John Wiley & Sons, 1986.

[10] C. C. Enz and G. C. Temes, "Circuit Techniques for Reducing the Effects of Op-Amp Imperfections: Autozeroing, Correlated Double Sampling, and Chopper Stabilization," *Proceedings of the IEEE*, vol. **84**, pp. 1584–1614, November 1996.

[11] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, Cambridge University Press, 2004.

[12] B. E. Boser and B. A. Wooley, "The Design of Sigma-Delta Modulation Analog-to-Digital Converters," *IEEE Journal of Solid-State Circuits*, vol. **23**, pp. 1298–1308, December 1988.

[13] B. Brandt, P. Ferguson, and M. Rebeschini, "Analog Circuit Design of ΣΔ ADCs," *Chapter 11 in Delta-Sigma Data Converters: Theory, Design and Simulation* (S. R. Norsworthy, R. Schreier, and G. C. Temes, Editors), IEEE Press, 1997.

[14] D. Haigh and B. Singh, "A Switching Scheme for Switched Capacitor Filters which Reduces the Effect of Parasitic Capacitances Associated with Switch Control Terminals," *Proc. of the IEEE Intl. Symp. on Circuits and Systems,* pp. 586–589, 1983.

[15] G. Yin and W. Sansen, "A High-Frequency and High-Resolution Fourth-Order ΣΔ A/D Converter in BiCMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. **29**, pp. 857–865, August 1994.
[16] K. R. Laker and W. M. C. Sansen, *Design of Analog Integrated Circuits and Systems*, McGraw-Hill, 1994.
[17] J.-T. Wu and K.-L. Chang, "MOS Charge Pumps for Low-Voltage Operation," *IEEE Journal of Solid-State Circuits*, vol. **33**, pp. 592–597, April 1998.
[18] K. Bult, "Analog Design in Deep Sub-Micron CMOS," *Proc. of the IEEE European Solid-State Circuits Conf.,* pp. 11–17, 2000.
[19] J. Cherry and W. Snelgrove, "Clock Jitter and Quantizer Metastability in Continuous-Time Delta–Sigma Modulators," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. **46**, pp. 661–676, June 1999.
[20] S. Luschas and H. S. Lee, "High-Speed ΣΔ Modulators with Reduced Timing Jitter Sensitivity," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. **49**, pp. 712–720, November 2002.
[21] O. Oliaei, "Design of Continuous-Time Sigma-Delta Modulators with Arbitrary Feedback Waveform," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. **50**, pp. 437–444, August 2003.
[22] O. Oliaei, "Sigma-Delta Modulator with Spectrally Shaped Feedback," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. **50**, pp. 518–530, September 2003.
[23] L. Hernández, A. Wiesbauer, S. Patón, and A. DiGiandomenico, "Modelling and Optimization of Low Pass Continuous-Time Sigma-Delta Modulators for Clock Jitter Noise Reduction," Proc. of the IEEE Intl. Symp. on Circuits and Systems, pp. 1072–1075, 2004.
[24] L. Hernández, P. Rombouts, E. Prefasi, S. Patón, and C. L. M. García, "A Jitter Insensitive Continuous-Time ΣΔ Modulator Using Transmission Lines," Proc. of the IEEE Intl. Conf. on Electronics, Circuits and Systems, pp. 109–112, 2004.
[25] R. H. van Veldhoven and A. H. M. van Roermund, *Robust Sigma Delta Converters*, Springer, 2011.
[26] P. G. R. Silva and J. H. Huijsing, *High Resolution IF-to-Baseband ΣΔ ADC for Car Radios*, Springer, 2008.
[27] J. Cherry and W. Snelgrove, "Excess Loop Delay in Continuous-Time Delta–Sigma Modulators," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. **46**, pp. 376–389, April 1999.
[28] J. F. Jensen, G. Raghavan, A. E. Cosand, and R. H. Walden, "A 3.2-GHz Second-Order Delta-Sigma Modulator Implemented in InP HBT Technology," *IEEE Journal of Solid-State Circuits*, vol. **30**, pp. 1119–1127, October 1995.
[29] J. van Engelen, R. J. van de Plassche, E. Stikvoort, and A. G. Venes, "A Sixth-Order Continuous-Time Bandpass Sigma-Delta Modulator for Digital IF Radio," *IEEE Journal of Solid-State Circuits*, vol. **34**, pp. 1753–1764, December 1999.
[30] S. Yan and E. Sánchez-Sinencio, "A Continuous-Time ΣΔ Modulator With 88-dB Dynamic Range and 1.1-MHz Signal Bandwidth," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 75–86, January 2004.
[31] P. Benabes, M. Keramat, and R. Kielbasa, "A Methodology for Designing Continuous-Time Sigma-Delta Modulators," Proc. of the IEEE European Design and Test Conf., pp. 46–50, 1997.
[32] P. Fontaine, A. N. Mohieldin, and A. Bellaouar, "A Low-Noise Low-Voltage CT ΔΣ Modulator with Digital Compensation of Excess Loop Delay," IEEE ISSCC Digest of Technical Papers, pp. 498–499, 2005.
[33] G. Mitteregger, C. Ebner, S. Mechnig, T. Blon, C. Holuigue, and E. Romani, "A 20-mW 640-MHz CMOS Continuous-Time ΣΔ ADC With 20-MHz Signal Bandwidth, 80-dB Dynamic Range and 12-bit ENOB," *IEEE Journal of Solid-State Circuits*, vol. **41**, pp. 2641–2649, December 2006.
[34] S. Pavan, N. Krishnapura, R. Pandarinathan, and P. Sankar, "A Power Optimized Continuous-Time ΔΣ ADC for Audio Applications," *IEEE Journal of Solid-State Circuits*, vol. **43**, pp. 351–360, February 2008.
[35] M. Keller, A. Buhmann, J. Sauerbrey, M. Ortmanns, and Y. Manoli, "A Comparative Study on Excess-Loop-Delay Compensation Techniques for Continuous-Time Sigma–Delta Modulators," *IEEE Trans. on Circuits and Systems—I: Regular Papers*, vol. **55**, pp. 3480–3487, December 2008.
[36] F. Gerfers, M. Ortmanns, and Y. Manoli, "A 1.5-V 12-bit Power-Efficient Continuous-Time Third-Order ΣΔ Modulator," *IEEE Journal of Solid-State Circuits*, vol. **38**, pp. 1343–1352, August 2003.
[37] S. Patón, A. di Giandomenico, L. Hernández, A. Wiesbauer, T. Poetscher, and M. Clara, "A 70-mW 300-MHz CMOS Continuous-Time ΣΔ ADC With 15-MHz Bandwidth and 11 Bits of Resolution," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 1056–1063, July 2004.

[38] L. Breems, R. Rutten, R. van Veldhoven, and G. van der Weide, "A 56 mW Continuous-Time Quadrature Cascaded ΣΔ Modulator With 77 dB DR in a Near Zero-IF 20 MHz Band," *IEEE Journal of Solid-State Circuits*, vol. **42**, pp. 2696–2705, December 2007.

[39] P. Kiss, J. Silva, A. Wiesbauer, T. Sun, U.-K. Moon, J. T. Stonick, and G. C. Temes, "Adaptive Digital Correction of Analog Errors in MASH ADC's–Part II: Correction Using Test-Signal Injection," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. **47**, pp. 629–638, July 2000.

[40] L. Breems, R. Rutten, and G. Wetzker, "A Cascaded Continuous-Time ΣΔ Modulator with 67-dB Dynamic Range in 10-MHz Bandwidth," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 2152–2160, December 2004.

[41] Y.-S. Shu, J. Kamiishi, K. Tomioka, K. Hamashita, and B.-S. Song, "LMS-Based Noise Leakage Calibration of Cascaded Continuous-Time ΣΔ Modulators," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 368–379, February 2010.

[42] M. Ortmanns, F. Gerfers, and Y. Manoli, "Compensation of Finite Gain-Bandwidth Induced Errors in Continuous-Time Sigma–Delta Modulators," *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. **51**, pp. 1088–1099, June 2004.

[43] L. Breems and J. H. Huijsing, *Continuous-Time Sigma-Delta Modulation for A/D Conversion in Radio Receivers*, Kluwer Academic Publishers, 2001.

[44] J. Cherry and W. Snelgrove, *Continuous-Time Delta-Sigma Modulators for High-Speed A/D Conversion*, Kluwer Academic Publishers, 1999.

3

Behavioral Modeling and High-Level Simulation

The analysis of nonideal effects described in Chapter 2 allows us to derive precise equivalent circuits and models for the different ΣΔM building blocks. This chapter shows how these models can be used for improving the accuracy and computational efficiency of system-level simulations. This constitutes an essential tool for the systematic design of ΣΔMs as will be described in Chapter 4.

This chapter is organized as follows. Section 3.1 describes the systematic *top-down/bottom-up* synthesis methodology commonly followed to design ΣΔMs. Section 3.2 compares the different simulation approaches that can be used for the evaluation of ΣΔMs, emphasizing on the benefits of behavioral simulation techniques. Section 3.3 focuses on the behavioral modeling technique and Section 3.4 shows an efficient way of implementing ΣΔM block models in SIMULINK. Finally, Section 3.5 presents SIMSIDES, a time-domain behavioral simulator implemented in the MATLAB/SIMULINK environment that is based on the behavioral modeling techniques described in this chapter. Finally, two case studies are described in Section 3.6 to illustrate the use of SIMSIDES for the high-level sizing and simulation of ΣΔMs.

3.1 Systematic Design Methodology of ΣΔ Modulators

One of the most common approaches used for the systematic design of high-performance ΣΔMs is based on the well-known *top-down/bottom-up* hierarchical synthesis methodology, conceptually illustrated in Figure 3.1a [1]. In this approach, a given system is divided into several *hierarchical levels* so that at each *abstraction* level of the system hierarchy, a design (or *sizing*) process takes place, thus transmitting (or *mapping*) the system specifications in a hierarchical way, from the top level to the bottom level. The reverse path in Figure 3.1a corresponds to the hierarchical bottom-up verification process of the system performance [2].

3.1.1 System Partitioning and Abstraction Levels

Figure 3.1b depicts the hierarchical synthesis methodology that is usually adopted in the design of ΣΔMs. The system is partitioned in the following hierarchical levels [1–5]:

CMOS Sigma-Delta Converters: Practical Design Guide, First Edition. José M. de la Rosa and Rocío del Río.

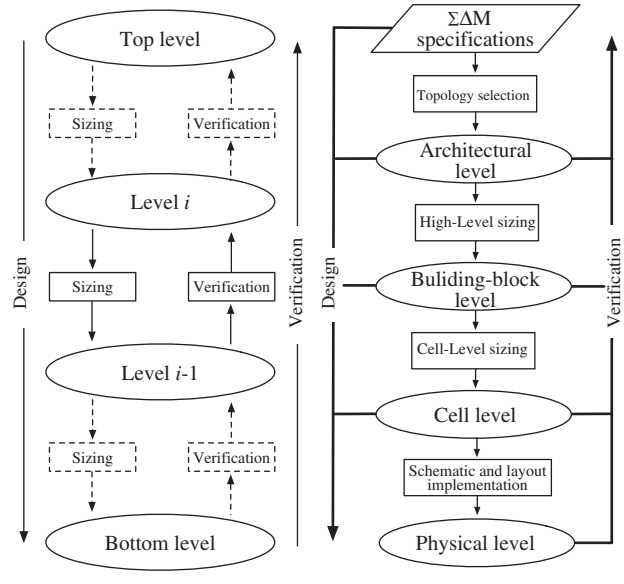


Figure 3.1 Hierarchical synthesis methodology: (a) Conceptual block diagram and (b) system partitioning commonly used in ΣΔMs.

- *Architecture or Topology Level*: that is, single-loop or cascade ΣΔM, single-bit or multibit quantization, low-pass or band-pass, DT or CT implementation, etc.
- *Subcircuit or Building-Block Level*: that is, amplifiers, transconductors, comparators, capacitors, resistors, switches, etc.
- *Cell Level*: that is, the circuit topology of a given building block, for instance folded-cascode or telescopic cascode OTA, SC or current-steering DAC, nMOS or CMOS switches, etc.
- *Physical Level*: which covers from transistor-level schematics to the layout and chip implementation.

The design process of a ΣΔM starts from the system-level *specifications*, that is, the effective resolution (ENOB) and the signal bandwidth (B_w). The first goal is to find out the best modulator topology that fulfills these specifications with minimum power dissipation. To this purpose, initial ideal design equations of NTF and IBN—which are based on a linear model of the embedded quantizers as described in Chapter 1—are used for calculating approximate values for the main ΣΔM parameters, that is, OSR, L, and B. Once these parameters are known, the architecture topology can be synthesized using more accurate nonlinear model equations. To this purpose, Schreier's MATLAB Delta-Sigma toolbox [6, 7] is widely used in the ΣΔ community. Usually, there are several topologies which are a priori good candidates to meet a given set of modulator specifications.

In order to determine the best ΣΔM architecture, some analytical procedures are normally used for estimating the power consumption of the different ΣΔM topologies [8]. These procedures are based on compact expressions of the in-band noise power, similar to those derived in Chapter 2, which contemplate both architectural and technological

features, and include the impact of some critical circuit errors such as thermal noise, finite OTA DC gain (or equivalent), and incomplete settling (GB and SR). Recent works demonstrate that nonlinearities can be efficiently incorporated in the Schreier's toolbox in order to fine tune this architectural exploration procedure [9].

On the basis of the system-level estimation of the power consumption, a ranking of different candidate ΣΔM architectures can be determined in order to decide the most suitable topology. Other important criteria at this step are the sensitivity to circuit element tolerances and mismatch (especially important in CT cascade topologies), the modulator loop-filter stability (particularly critical in high-order single-loop topologies), and/or the impact of the feedback DAC nonlinearity in multibit ΣΔMs.

3.1.2 Sizing Process

Once the modulator architecture has been selected, the next step consists of mapping the modulator-level specifications (ENOB and B_w) onto building-block specifications, that is, amplifier finite DC gain, output swing, GB, SR, etc. In this system-level design process—commonly referred to as *high-level sizing*—the design parameters are the electrical specifications of the different ΣΔM subcircuits, namely, amplifiers, transconductors (in CT-ΣΔMs), comparators, switches, and passive elements, that is, capacitors, resistors, and even inductors in the case of some BP-ΣΔMs [10]. The result of this multidimensional design space exploration constitutes the start point of the *electrical* or *cell-level sizing* process where, after selecting the appropriate topology for every ΣΔM subcircuit, the corresponding transistor sizes and bias currents are obtained.

Therefore, as shown in Figure 3.1b, the design methodology of ΣΔMs can be essentially divided into two main sizing processes: high-level sizing and cell-level sizing. Both of them are tasks that require a multitude of *redesign* iterations until the specifications at each level are met [11]. This procedure is conceptually depicted in Figure 3.2 [5]. At each iteration the performance of the circuit is evaluated at a given point of the design space,

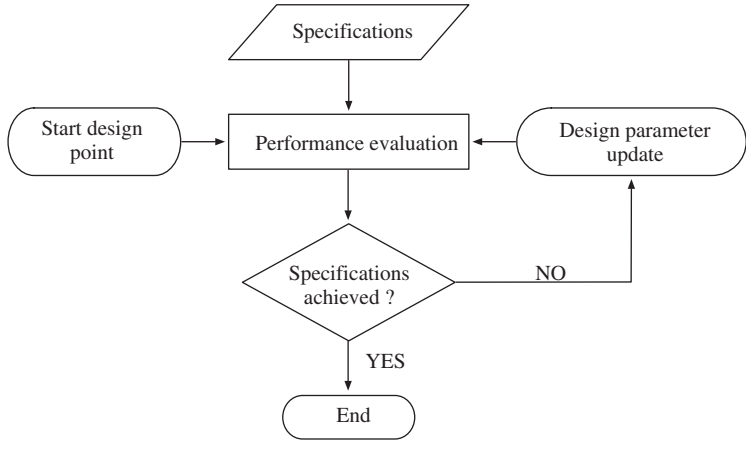


Figure 3.2 Iterative procedure usually followed for high-level sizing and cell-level sizing.

the design parameters are modified according to such an evaluation, and the procedure is repeated. Note that, although this procedure is the same in both high-level sizing and cell-level sizing, the design problem and its formulation is different in both cases. High-level sizing is a system-level synthesis process where the inputs variables are the *system-level (modulator) specifications* and the output variables are the building-block electrical specifications (*design parameters*). The latter specifications constitute the input variables for the cell-level sizing process.

The design parameter selection in Figure 3.2 can be carried out "manually," that is, the entire design space is searched by considering all possible combinations of design parameters. In this case, once the whole design space has been completely explored and checked, the best design can be selected among those that meet the required specifications with the lowest (estimated) power dissipation and area. However, this "vast force" search approach is quite inefficient—and even unfeasible in many cases—in terms of computational resources and CPU time.[1] Instead, an *optimization engine* is normally used for guiding the exploration of the design space. To this purpose, diverse algorithms such as *simulated annealing* [12] or *genetic* algorithms [4] may be used for the optimization. Usually, the same optimization methodology—normally based on the formulation and minimization of a design-oriented *cost function*—is used in both sizing tasks, that is, high-level sizing and cell-level sizing [3].

Different performance-evaluation strategies can be followed in Figure 3.2. Essentially, the cost function can be evaluated by means of equations or simulations [1]. Although the former is much faster than the latter, the accuracy of the results strongly depends on the topology, that is, the ΣΔM architecture (for high-level sizing) and the circuit schematic (for cell-level sizing). For that reason, the most common approach followed by the ΣΔM community has been based on using simulation as performance evaluation [3–5, 13–19]. However, in contrast to the optimization process—which is used in both sizing tasks—a different simulation approach is considered to evaluate the performance of either the entire (system-level) modulator or just a single building block, such as an amplifier or a comparator (cell-level). The latter can be analyzed using an electrical (SPICE-like) simulator with a high degree of accuracy and computational efficiency. On the contrary, the evaluation of the system-level (modulator) performance can be carried out following different simulation approaches discussed in the next section.

3.2 Simulation Approaches for the High-Level Evaluation of ΣΔMs

The high-level synthesis described above requires performing a large number of simulations. Depending on the design and required specifications, hundreds or even thousands of iterations may be needed. Therefore, in order to make this part of the design feasible, simulations should consume short CPU times. This is particularly critical in the case of ΣΔMs because of their nonlinear *oversampled*-data nature. For that reason, in order to compute the IBN power at the output of a ΣΔM with enough numerical accuracy,

[1] At the end of this chapter, a simulation-based high-level synthesis methodology is explained and applied to some case studies. As will be shown, this synthesis method does not use any optimizer, and it is based on parametric simulations for checking both isolated and cumulative effects of different building-block errors in order to evaluate the performance of a ΣΔM. The resulting synthesis is reasonably efficient in terms of CPU time and it can be fine-tuned using optimization.

a transient simulation of at least 2^{16} clock cycles is usually required.[2] Therefore, a transient analysis of a ΣΔM using common transistor-level (SPICE-like) circuit simulation is too slow for allowing an efficient space exploration, regardless of whether it is performed by a manual parameter sweeps or guided by an optimizer. As an example, a 2^{16}-point transient analysis in HSPICE of a cascade 2-1 SC-ΣΔM including the clock-phase generator and other auxiliary subcircuits takes over 85 h CPU time in a 2.2-GHz core with 4-GB RAM. This means that a synthesis process based on this performance-evaluation approach would take months or even years! Hence, transistor-level simulation is obviously computationally unfeasible for synthesis purposes, and it is normally used only for the final design verification, as will be discussed in Chapter 4 [19].

In addition to the long CPU times required to simulate a ΣΔM at transistor level, another practical issue associated with this simulation approach is related to the convergence problems that typically arise in electrical simulators such as HSPICE [21] or Spectre [22]. Moreover, transistor-level simulation is not a suited simulation method for the high-level sizing process because the design parameters—that is, device sizes and biasing—are hierarchically far from system-level specifications. Hence, it is quite difficult for a designer to get an insight into what is happening at the system level by running simulations based on transistor-level models, as the impact of a given transistor width or length has a direct impact on many different performance metrics of ΣΔM building blocks. For instance, the size of the input differential-pair transistor of an OTA affects the values of finite DC gain, nonlinearity, input parasitic capacitance, GB, etc.

3.2.1 Alternatives to Transistor-Level Simulation

The above-mentioned reasons suggest that increasing the abstraction level of the simulation approach is mandatory to both speed up the simulations and work with design parameters that are closer to the system-level specifications. However, the price to pay for increasing the abstraction level (and consequently the simulation speed) is the loss of accuracy. This problem has motivated the exploration of different simulation approaches in order to optimize the trade-off between CPU time and precision—conceptually illustrated in Figure 3.3 [4].

An alternative to the transistor-level simulation approach that also runs on electrical simulators consists of using circuit *macromodels*. These macromodels are usually implemented as equivalent circuits based on ideal voltage- and current-controlled sources that allow to model the main nonideal effects. This approach has the advantage that it can be combined with transistor-level schematics using the same (electrical) simulator. Thus, critical parts of the system can be modeled at transistor level, whereas the remaining ones can be modeled less accurately with macromodels, leading to what is commonly known as a *multilevel simulation* technique [2]. Obviously, the simulation CPU time will increase with the number of parts being modeled at transistor level. Therefore, in most practical

[2] The FFT algorithms that are traditionally used for obtaining the output spectra of many ADCs in general, and of ΣΔMs in particular, are more efficiently computed if the number of points of the data sequence is a power of 2 [20]. Therefore, the number of points is often expressed as 32k point or 64k point, if 2^{15} or 2^{16} clock cycles are considered, respectively.

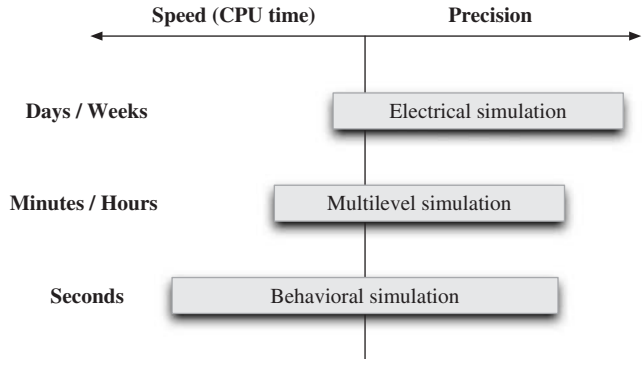


Figure 3.3 Comparison of simulation approaches in terms of CPU time and accuracy.

cases, the use of this approach does not imply a significant improvement with respect to the full transistor-level simulation approach.[3]

As an example, Table 3.1 compares the CPU times required to simulate a fourth-order single-loop ΣΔM in Cadence Spectre circuit simulator, considering 2^{15}-point transient analysis with `moderate` setting [22]. The modulator topology consists of a cascade of resonators with feed-forward summation (Figure 1.31b). This architecture requires five amplifiers to implement the four loop-filter integrators and the active adder. Simulations were carried out considering ideal macromodels for the switches and the digital blocks (including the clock-phase generator). It can be noted from Table 3.1 that the CPU time increases by approximately 30–40 min with every additional amplifier that is simulated considering a transistor-level implementation. The overall CPU time goes from 45 min required to simulate the entire system with macromodels to 4 h when all amplifiers are implemented at transistor level. These CPU times are more than doubled if `conservative` mode option is considered in the simulation; over 8 h are required to complete the simulation.

Table 3.1 CPU time required to simulate a fourth-order single-loop ΣΔM considering different situations in a multilevel approach

Simulation Approach	CPU Time (`moderate` mode)	CPU Time (`conservative` mode)
All opamps macromodeled	45 min	2 h
One transistor-level opamp	1 h, 20 min	4 h
Two transistor-level opamps	2 h	5 h, 30 min
Five transistor-level opamps	4 h	8 h, 20 min

Simulations were carried out in a SUN Fire X2200 M2 server with 4-GB RAM and a 2.2-GHz Dual Core AMD Opteron CPU, running a 64-bit Linux operating system.

[3] The use of macromodels may also be a good strategy to implement the required *bridge* between the system level and the electrical level, as will be described in Chapter 4.

As stated above, it is important to mention that the data in Table 3.1 does not account for the time consumed to simulate the digital part of the system—that is, feedback DAC, DEM, clock-phase generator, digital output buffers, etc.—that was considered ideal in these simulations. If transistor-level implementation is considered for those circuits, the overall CPU time for a conservative simulation mode is increased up to one day or even more. Under these conditions, a 2^{16}-point transient simulation would take over three days CPU time!

3.2.2 Event-Driven Behavioral Simulation Technique

As illustrated in Figure 3.3, the best trade-off between accuracy and speed is achieved by the so-called *event-driven behavioral simulation* technique [14]. In this approach—conceptually illustrated in Figure 3.4—the modulator is partitioned into a set of subcircuits—often called *basic* or *building blocks*—with independent functionality [1, 12]. In ΣΔMs, the most important building blocks are integrators and resonators,[4] and the embedded quantizers, made up of an ADC and a DAC. The behavioral modeling

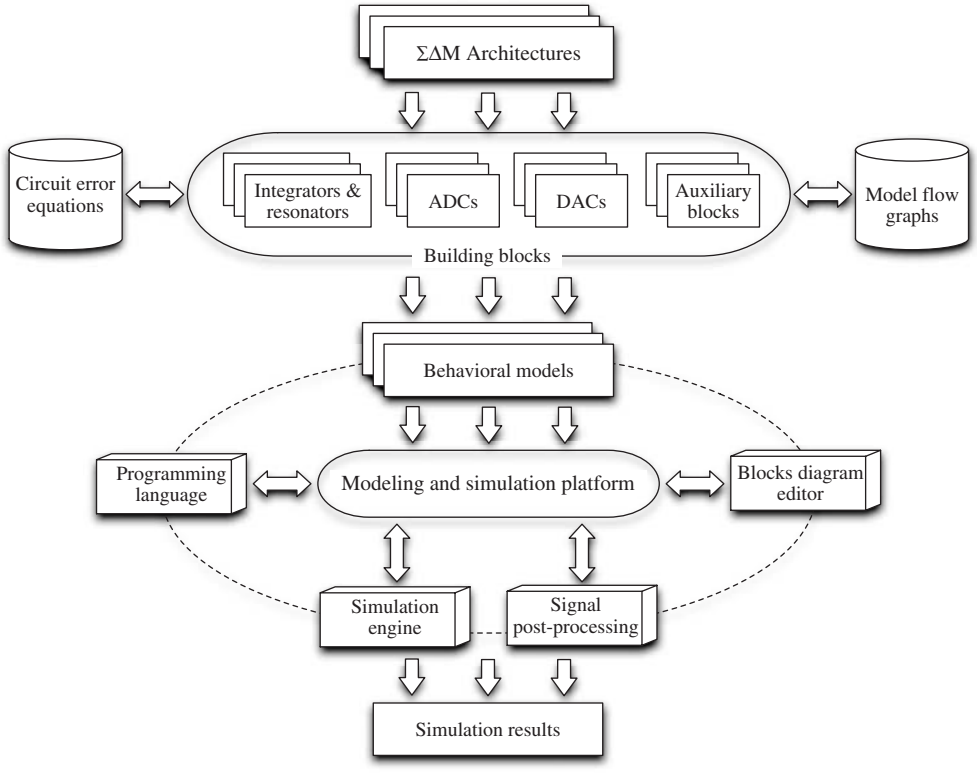


Figure 3.4 Conceptual block diagram of the behavioral modeling and simulation process.

[4] Integrators are the basic elements of the loop filter in LP-ΣΔMs. These circuits can be combined to implement resonators in order to implement BP-ΣΔMs, as described in Chapter 1.

technique thus consists in describing each of these building blocks by a model, often referred to as a *behavioral model* or *behavioral law*, that emulates their actual operation and takes into account the effect of the main nonideal circuit error mechanisms described in Chapter 2.

Note that, strictly speaking, the definition given above for a behavioral model does not necessarily imply that the model is implemented by equations. Indeed, there is an alternative approach that consists of using the so-called *table look-up* models [23, 24]. The idea behind these models is based on extracting the input–output characteristics of a given building block from electrical simulations and then mapping the extracted information in the form of tables which are used for modeling the functionality of that building block. This way, those tables are used as an alternative to the original transistor-level circuits to accelerate the simulations with a high degree of accuracy. However, the price to pay is the loss of generality and of reusability of the models because they strongly depend on the circuit topology. Indeed, new tables have to be generated whenever the architecture itself or any circuit parameter is modified.

For the aforementioned reasons, table look-up methodologies are more suited for bottom-up verification than for high-level synthesis. Indeed, the most commonly used behavioral modeling approach is based on finite-difference equations. These equations describe the functionality of building blocks by expressing their output signals in terms of their internal state variables and their input signals. Therefore, the accuracy of the behavioral simulation strongly depends on how precisely those equations describe the actual behavior of the corresponding ΣΔM subcircuit [5, 15, 19].

3.2.3 Programming Languages and Behavioral Modeling Platforms

Behavioral models and the corresponding simulation engine (Figure 3.4) can be codified and implemented in a number of platforms using different programming/modeling languages such as C. The latter is a general purpose and universal language that presents high flexibility in describing behavioral models and allowing implementation of behavioral simulation tools in many different operating systems and platforms. Early approaches for the behavioral-level simulation of ΣΔMs were completely compiled in C language, demonstrating to be very appropriate for fast simulation and high-level synthesis of ΣΔMs [12].

The main disadvantage of C-coded behavioral modeling and simulation is that it is restricted to a limited number of building-block models—that is, those included in the corresponding (previously programmed) libraries—thus reducing the kind of ΣΔM architectures that can be simulated. From this point of view, this approach is not flexible because building-block models cannot be easily modified and the extension to new building blocks and architectures is constrained by the capabilities of the simulation engine, as well as by the designer skills on C programming language. This may explain why the majority of reported C-based behavioral simulators are intended to simulate SC-ΣΔMs [4, 12].

In order to overcome all these problems, several alternative approaches have been followed to implement behavioral models. One approach that has gained popularity with ΣΔM designers is based on the use of standard hardware description languages (HDL), such as either VHDL [25] and its analog extensions [26] or Verilog and Verilog AMS [19]. As will be shown in Chapter 4, VHDL-based behavioral model descriptions can be combined with HDL models of other analog, digital, and mixed-signal circuits, and they

can be integrated in the design flow of commercial design environments such as Cadence Design FrameWork II [27].

An alternative and widely used approach for the behavioral modeling and simulation of ΣΔMs consists of using the MATLAB/SIMULINK platform [28, 29]. This well-known mathematical software—which constitutes a standard CAD platform today in science and engineering—presents a number of advantages in terms of friendliness of the user interface: flexibility for the extension of building blocks library and for the simulation of either DT or CT systems, good trade-off between accuracy and simulation speed, and a direct access to very powerful tools for signal postprocessing [5, 15].

The rest of this chapter is devoted to the behavioral modeling and simulation of ΣΔMs in the MATLAB/SIMULINK environment, explaining the different approaches to model ΣΔM building blocks and how to use these models for the efficient simulation of ΣΔMs.

3.3 Implementing ΣΔM Behavioral Models

The analysis of error mechanisms in ΣΔMs allows designers to obtain a set of closed-form expressions that shows the degradation caused by circuit-level electrical parameters at different levels of the modulator hierarchy. On the one hand, the analytical procedure described in Chapter 2 is used for propagating the effect of errors from the building-block (either integrator or resonator) transfer function to the modulator NTF, in order to obtain the ΣΔM performance metrics, that is, IBN and SNDR. As stated in Section 3.1, simplified versions of these equations relating architectural parameters (L, OSR, and B) to circuit-level errors are very appropriate for initial system-level estimations of the power consumption and preliminary architecture selection. On the other hand, precise equations describing the functionality of building blocks as a function of error parameters constitute the basis for building accurate behavioral models. To this end, those equations must be transformed into computational flowcharts that can be implemented by programming languages. This procedure is illustrated in the next section for two basic building blocks: an SC FE integrator and a Gm-C integrator. These two circuits are basic building blocks of SC- and CT-ΣΔMs, respectively.[5]

3.3.1 From Circuit Analysis to Computational Algorithms

Figure 3.5 shows the conceptual (single-ended) schematic of an SC FE integrator (Figure 3.5a) and a Gm-C integrator (Figure 3.5b). Considering ideal circuit elements, the output voltage v_o of Figure 3.5a is given by the following finite-difference equation:

$$v_o(nT_s) = \frac{C_s}{C_i} v_i[(n-1)T_s] + v_o[(n-1)T_s] \tag{3.1}$$

where v_i is the input voltage, C_s and C_i are, respectively, the sampling and integration capacitors, and nT_s stands for the nth sampling time instant, with T_s being the sampling period.

[5] The procedure described in this section focuses on a behavioral model considering firstly ideal behavior and then adding only the effect of the nonlinear finite amplifier gain in SC integrators and of the finite OTA gain in Gm-C integrators. A more complete description of the behavioral model, as well as its implementation in the MATLAB/SIMULINK environment, will be detailed in the following sections.

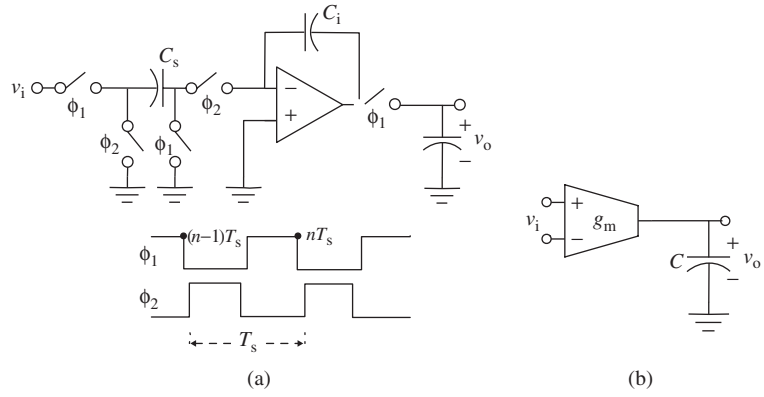


Figure 3.5 Conceptual schematic of: (a) an SC FE integrator and (b) a Gm-C integrator.

Note that Equation 3.1 is easy to solve numerically because the value of the output voltage at a given time instant nT_s can be computed by adding the value of the output at the previous sampling period $(n-1)T_s$ and the value of the input voltage at $(n-1)T_s$ multiplied by C_s/C_i. This operation is conceptually represented by the flowchart shown in Figure 3.6 and can be expressed in a computational model as follows:

```
vinit=0;
Cs=1e-12;
Ci=1e-12;
n=1;
nfinal=10;
vi(1:10)=1;
vo(1)=vinit;
while n<=nfinal
    n=n+1;
    vo(n)=vo(n-1)+Cs/Ci*vi(n-1);
end;
```

where `vinit` stands for the initial condition, `nfinal` stands for the final sampling instant, and $C_s = C_i = 1\text{pF}$ has been assumed.[6] A `while` statement has been considered, although other iterative statements such as `if` or `for` can be used as well.

The ideal functionality of Figure 3.5b can be mathematically expressed by the following differential equation:

$$g_m v_i(t) = C\frac{dv_o(t)}{dt} \tag{3.2}$$

where g_m is the transconductance and C is the integration capacitance. The above expression can be integrated by MATLAB/SIMULINK solvers very efficiently [28].

[6] The MATLAB code used in this example can be easily translated to other programming languages such as C, Verilog-A, or VHDL-AMS [19].

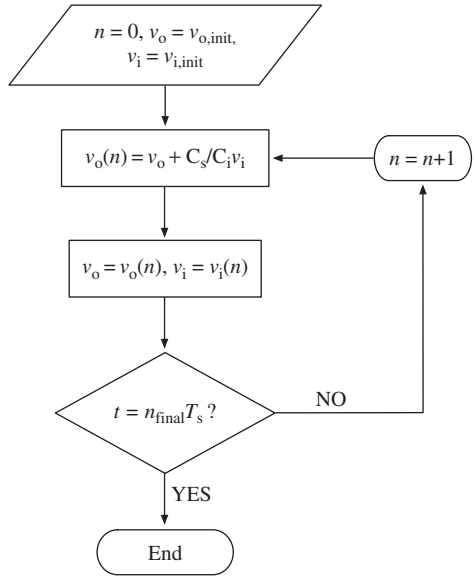


Figure 3.6 Flowchart used for computing behavioral Equation 3.1.

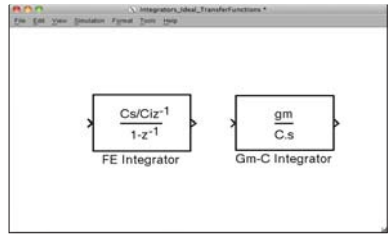


Figure 3.7 Ideal SIMULINK models of the integrators shown in Figure 3.5.

The ideal models described for SC and CT integrators in the time domain can be implemented very easily in the frequency domain using SIMULINK elementary library blocks [29] as illustrated in Figure 3.7. The corresponding transfer functions can be obtained by applying a Z-transform and an S-transform to Equations 3.1 and 3.2, respectively, giving

$$\begin{aligned} v_o(z) &= \frac{C_s/C_i \cdot z^{-1}}{(1 - z^{-1})} \cdot v_i(z) \\ v_o(s) &= \frac{g_m}{sC} \cdot v_i(s) \end{aligned} \tag{3.3}$$

where $v_i(z)$ and $v_o(z)$ stand for the Z-transforms of $v_i(nT_s)$ and $v_o(nT_s)$ in Figure 3.5a and $v_i(s)$ and $v_o(s)$ are the S-transforms of $v_i(t)$ and $v_o(t)$ in Figure 3.5b. Both models can be computed very efficiently using the discrete-time and continuous-time solvers provided in the SIMULINK software [29].

3.3.2 Time-Domain versus Frequency-Domain Behavioral Models

The ideal behavioral models described can be either represented in the time domain by Equations 3.1 and 3.2 or in the frequency domain by Equation 3.3. There is an exact correspondence between both of them, giving rise to identical results. Indeed, provided that the ΣΔM building blocks can be treated as LTI systems, the most convenient (and simplest) way of modeling their behavior is using frequency-domain transfer functions. However, this is not very useful in most practical situations because, as stated in Chapter 2, the performance of ΣΔMs is degraded by the effect of circuit-level errors. As shown in this chapter, the majority of circuit errors are modeled in a more accurate way if behavioral models are described in the time domain instead of in the frequency domain.

As an example, let us consider the effect of OTA finite DC gain on the integrators in Figure 3.5. This effect can be modeled as shown in Figure 3.8, where A_v stands for the finite DC voltage gain of the opamp in the SC FE integrator and g_o is the finite output conductance of the transconductor in the Gm-C integrator, such that the finite DC gain is given by g_m/g_o. The time-domain equations describing the behavior of the integrators in Figure 3.8 are given by

$$\begin{aligned} v_o(nT_s) &= \frac{1+\mu}{1+(1+g)\mu} v_o[(n-1)T_s] + \frac{g}{1+(1+g)\mu} v_i[(n-1)T_s] \\ g_m v_i(t) &= C\frac{dv_o(t)}{dt} + g_o v_o(t) \end{aligned} \tag{3.4}$$

where $\mu \equiv 1/A_v$ and $g = C_s/C_i$. Note that the behavioral model of the SC integrator with finite DC gain can be computed using a flowchart similar to the ideal one shown in Figure 3.6 by simply modifying the corresponding multiplication factors of v_i and v_o according to Equation 3.4. In the case of CT integrators, the corresponding differential equation can be numerically solved using either C or MATLAB code as programming language.[7]

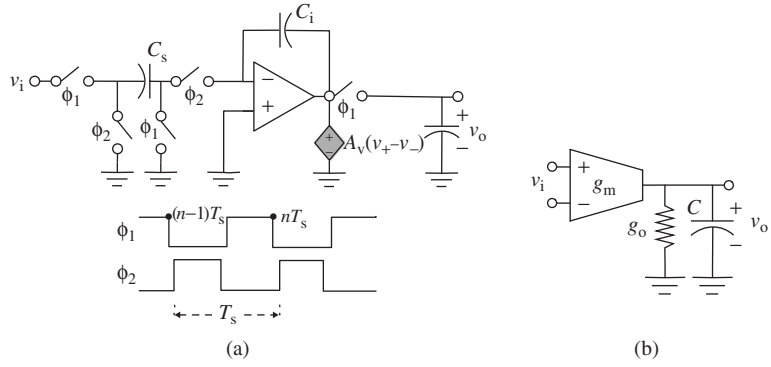


Figure 3.8 Modeling OTA finite DC gain in: (a) SC FE integrators and (b) Gm-C integrators.

[7] The interested reader can find a detailed documentation related to the implementation of the continuous-time and discrete-time *stateflow* charts available in MATLAB [30].

Operating in a similar way as in the previous section, the (frequency-domain) transfer functions of both circuits in Figure 3.8 can be obtained yielding

$$\begin{aligned} v_o(z) &= \frac{gz^{-1}}{1+(1+g)\mu-(1+\mu)z^{-1}} \cdot v_i(z) \\ v_o(s) &= \frac{g_m/g_o}{1+sC/g_o} \cdot v_i(s) \end{aligned} \tag{3.5}$$

The above-mentioned transfer functions can be easily modeled using elementary blocks of the SIMULINK library as shown in Figure 3.9. However, as stated above, frequency-domain models are not useful when several nonideal and nonlinear effects are considered.[8] Instead, a time-domain model considering the operation during different clock phases is more accurate to include different circuit-level phenomena. In order to illustrate this, let us analyze the circuit in Figure 3.8a in both clock phases, ϕ_1 and ϕ_2.

At the end of clock phase ϕ_1 (sampling phase) of period $(n-1)T_s$ the sampling capacitor C_s and the integration capacitor C_i are charged at the following voltages, respectively,

$$\begin{aligned} v_{C_s,\phi_1} &= -v_i[(n-1)T_s] \\ v_{C_i,\phi_1} &= -(1+\mu) \cdot v_{o1}[(n-1)T_s] = -(1+\mu) \cdot v_{o1}[(n-3/2)T_s] \end{aligned} \tag{3.6}$$

where $v_{o1}(nT_s)$ denotes the voltage at the output of the opamp in Figure 3.8a.

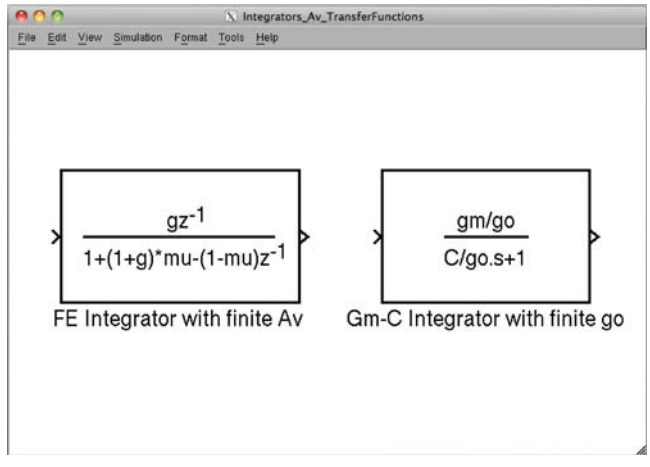


Figure 3.9 Modeling the effect of finite DC gain on the integrators transfer function in SIMULINK.

[8] Since the mathematical treatment of nonlinear errors becomes very complicated when the analysis is carried out in the time domain, some authors propose alternative ways to model the effect of *weak nonlinearities* by combining MATLAB functions with linear SIMULINK models [15] in SC-ΣΔMs, or by properly modifying the Schreier's Delta-Sigma Toolbox to simulate CT-ΣΔMs in the discrete-time [9].

At the end of clock phase ϕ_2 (integration phase) of period $(n-1/2)T_s$ the voltages across C_s and C_i are, respectively, given by

$$\begin{aligned} v_{C_s,\phi_2} &= -\mu \cdot v_{o1}[(n-1/2)T_s] \\ v_{C_i,\phi_2} &= -(1+\mu) \cdot v_{o1}[(n-1/2)T_s] \end{aligned} \tag{3.7}$$

The charge-balance equation at the negative input node of the opamp in Figure 3.8a can be written as

$$C_s \cdot (v_{C_s,\phi_2} - v_{C_s,\phi_1}) + C_i \cdot (v_{C_i,\phi_2} - v_{C_i,\phi_1}) = 0 \tag{3.8}$$

Replacing Equations 3.6 and 3.7 into Equation 3.8 gives

$$[1+(1+g)\mu] \cdot v_{o1}[(n-1/2)T_s] = g \cdot v_i[(n-1)T_s] + (1+\mu) \cdot v_{o1}[(n-3/2)T_s] \tag{3.9}$$

Taking into account that $v_o(nT_s) = v_{o1}[(n-1/2)T_s]$, the above expression transforms into the SC part of Equation 3.4, which after taking the Z-transform results in Equation 3.5.

Note that the charge-balance analysis described in Equations 3.6–3.8 is more complex than using Z-domain transfer functions to model simple phenomena, such as the ideal behavior of simple building blocks and/or the effect of linear errors, such as finite DC gain. However, as the model becomes more accurate including more nonideal circuit errors and/or nonlinearities, the charge-balance model is the most suitable approach to be implemented in a behavioral simulator. As an illustration, let us assume the voltage dependence of the finite DC gain in SC FE integrators.[9] This nonlinear effect can be modeled by replacing the voltage linear gain A_v parameter in Figure 3.8a by a nonlinear function of the output voltage $A_v(v_o) \simeq A_v \cdot (1 + \text{avnl}_1 \cdot v_o + \text{avnl}_2 \cdot v_o^2 + \cdots)$, where avnl_i represents the ith-order nonlinear voltage-gain coefficient [31]. Taking these nonlinear effects into account, the charge-balance equations transform into the following ones:

$$\begin{aligned} v_{C_s,\phi_1} &= -v_i[(n-1)T_s] \\ v_{C_i,\phi_1} &= -[1+1/A_{v(n-3/2)}] \cdot v_{o1}[(n-1)T_s] = -(1+\mu) \cdot v_{o1}[(n-3/2)T_s] \\ v_{C_s,\phi_2} &= -[1/A_{v(n-1/2)}] \cdot v_{o1}[(n-1/2)T_s] \\ v_{C_i,\phi_2} &= -[1+A_{v(n-1/2)}] \cdot v_{o1}[(n-1/2)T_s] \end{aligned} \tag{3.10}$$

where $A_{v(n-1/2)} = A_v(v_{o1}[(n-1/2)T_s])$ and $A_{v(n-3/2)} = A_v(v_{o1}[(n-3/2)T_s])$.

From Equations 3.9 and 3.10, it can be shown that the nonlinear finite-difference equation describing the behavior of the integrator can be written as

$$\begin{aligned} \left(1+\frac{1+g}{A_{v(n-1/2)}}\right) \cdot v_{o1}[(n-1/2)T_s] &= g \cdot v_i[(n-1)T_s] \\ &\quad + \left(1+\frac{1}{A_{v(n-3/2)}}\right) v_{o1}[(n-3/2)T_s] \end{aligned} \tag{3.11}$$

Note that an iterative procedure is needed to compute the behavioral model in Equation 3.11, because the output voltage of the opamp $v_{o1}(nT_s)$ depends on the amplifier nonlinear gain, which in turns changes with $v_{o1}(nT_s)$.

[9] Precise time-domain behavioral models of CT-ΣΔM building blocks including the main circuit effects will be described in Section 3.4.

3.3.3 Implementing Time-Domain Behavioral Models in MATLAB

The behavioral model described by Equations 3.10 and 3.11 can be programmed using a so-called M-file [28] with the MATLAB code shown in Figure 3.10, where `C1` and `C2` are, respectively, the sampling capacitance (C_s) and integration capacitance (C_i), and `count` is a parameter that determines the clock phase, with `count=0` corresponding to the sampling phase and `count=1` to the integration phase. This way, as illustrated in Figure 3.11, the clock-phase scheme can be selected to be either the one shown in Figure 3.8a or the complementary one, that is, with the input switch being clocked at ϕ_2. The input signal is modeled by a matrix made up of two vectors, `u(1)` and `u(2)`, while the output is stored in a variable named `y`, with `yold` modeling the output sample stored in the previous clock phase and `ytemp` being a temporal state variable used until convergence is achieved.

```
function y = intfeavnl(u,AV,AVNL1,AVNL2,AVNL3,AVNL4,C1,C2,PHI)
persistent VC1 VC2 VC1OLD count yold
if (isempty(VC1))
    VC1=0;
    VC2=0;
    VC1OLD=0;
    count=0;
    yold=0;
end
     if (PHI==1)
           if (count==0)
               VC1 =u(1);
               VC2 =yold;
               y=yold;
               count=count+1;
           else
               VC1OLD = VC1;
               VC1 = u(2);
               AVNEW = AV;
               ITER = 0;
               AVVAR=1;
               while (AVVAR>0.01&ITER<50),
                   ytemp = ((C1/C2)*(VC1OLD-VC1)*(AVNEW/(1+AVNEW+(C1/C2)))) + (VC2*((AVNEW+1)/(1+AVNEW+(C1/C2))));
                   AVOLD = AVNEW;
                   VO2 = ytemp^2;
                   VO3 = ytemp^3;
                   VO4 = ytemp^4;
                   AVNEW = AV*(1+(AVNL1*ytemp)+(AVNL2*VO2)+(AVNL3*VO3)+(AVNL4*VO4));
                   AVVAR = abs((AVOLD-AVNEW)/AVNEW);
                   ITER = ITER + 1;
               end
               y = ytemp;
               yold=y;
               count=0;
           end
    elseif (count==0)
               VC1OLD = VC1;
               VC1 = u(2);
               AVNEW = AV;
               ITER = 0;
               AVVAR=1;
               while (AVVAR>0.01&ITER<50),
                   ytemp = ((C1/C2)*(VC1OLD-VC1)*(AVNEW/(1+AVNEW+(C1/C2)))) + (VC2*((AVNEW+1)/(1+AVNEW+(C1/C2))));
                   AVOLD = AVNEW;
                   VO2 = ytemp^2;
                   VO3 = ytemp^3;
                   VO4 = ytemp^4;
                   AVNEW = AV*(1+(AVNL1*ytemp)+(AVNL2*VO2)+(AVNL3*VO3)+(AVNL4*VO4));
                   AVVAR = abs((AVOLD-AVNEW)/AVNEW);
                   ITER = ITER + 1;
               end
               y = ytemp;
               yold=y;
               count=count+1;
           else
               VC1 = u(1);
               VC2 = yold;
               y=yold;
               count=0;
           end
```

Figure 3.10 MATLAB code for the behavioral model of the SC FE integrator in Figure 3.8a including the effect of the amplifier gain nonlinearity.

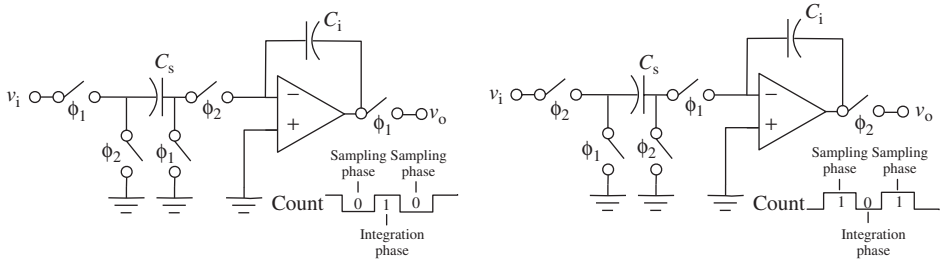


Figure 3.11 Meaning of count in the behavioral model of an SC FE integrator, considering that the input switch clock phase is: (a) ϕ_1 and (b) ϕ_2.

Note that the convergence criterion used in the iterative procedure is `AVVAR=abs[(AVOLD =AVNEW) /AVNEW] < thrs`, where `thrs` models the threshold value selected for convergence (usually `thrs=0.01`), `abs(x)` stands for the absolute value of `x`, and `AVOLD` and `AVNEW` are, respectively, the old and new values of the parameter to be solved—A_v in this example. Following this criterion, convergence is normally reached in three or four iterations, which does not result in excessive CPU time [5].

Figure 3.12 shows a SIMULINK model of the SC FE integrator in Figure 3.8a with nonlinear DC gain based on the MATLAB code described in Figure 3.10. Note that a MATLAB function block—named `MATLAB Fcn` block in SIMULINK [29]—is used for this purpose. Figure 3.12b shows the MATLAB Fcn used in this example, called `intfescavnl`, that applies the M-file shown in Figure 3.10 to the input of the block. The input arguments of the MATLAB functions (e.g., `AV,AVNL1,AVNL2...C1,C2,PHI`) are included in the `Function Block Parameters` dialogue window [29] illustrated in Figure 3.12c, which can be configured by the user. Note that two additional blocks in Figure 3.12b are used for properly sampling the output signal at the correct sampling instant. One of these blocks is a `Unit Delay block` that adds an extra delay of $T_s/2$. The other block is a so-called SIMULINK `S-function` block, which will be explained in Section 3.4.

One of the problems of using M-files is that the MATLAB interpreter is called at each time step, and this slows down the simulation [28]. This problem is aggravated as the model complexity increases. Let us consider, for instance, a cascade 2-1-1 SC-ΣΔM in which all building blocks are ideal except for the nonlinear DC gain—modeled by the M-file shown in Figure 3.10. Figure 3.13 shows the SIMULINK block diagram of the modulator, highlighting its main parts. A simulation of 2^{16} clock periods of this modulator takes 148 s in a 2.4-GHz core with 4-GB RAM.[10]

An alternative approach to the simulation of ΣΔMs based on MATLAB functions was proposed in [15, 32]. The models included in this toolbox [33] are based on the interconnection of SIMULINK standard library blocks, being very intuitive and useful for

[10] This simulation was carried out using the normal mode in SIMULINK. The use of the accelerator mode available in SIMULINK speeds up the simulation regardless of the modeling strategy used. However, the accelerator mode cannot be used for some types of M-files, for instance when these files use parameters that have more than one dimension—very usual in the modeling of ΣΔM building blocks [29].

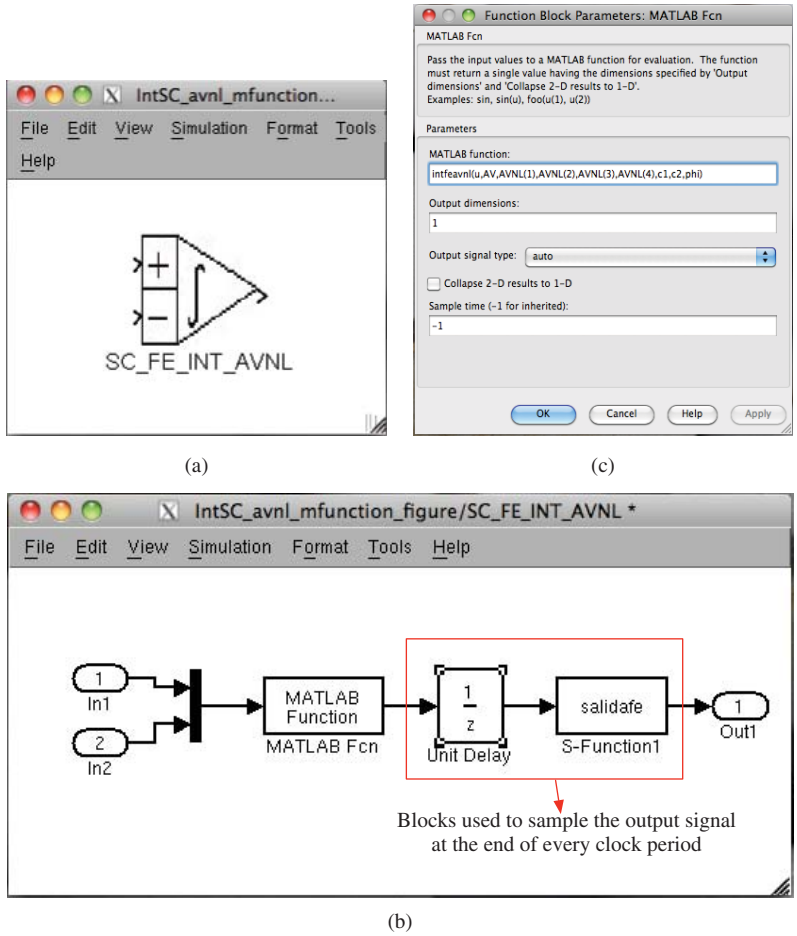


Figure 3.12 SIMULINK model of the SC FE integrator shown in Figure 3.8a: (a) SIMULINK mask, (b) SIMULINK block diagram including a MATLAB function, and (c) associated dialogue window.

system-level evaluation. However, the block library is limited to SC circuits and uses relatively simple models which do not take into account some limitations such as, for instance, the nonlinearities associated with the OTA DC gain and with capacitors. The models are very intuitive, although the implementation of each building block requires several sets of elementary SIMULINK blocks using MATLAB functions, with the subsequent penalty in computation time. This issue is very critical for an optimization-based synthesis process, as stated in previous sections. Moreover, most parts of the models are implemented in the Z-domain, and hence, the circuit behavior during different clock phases is not taken into account. This may lead to a not very precise modeling of some errors associated with the integrator transient response, as discussed in Section 2.4.

Figure 3.14 shows an exemplary SIMULINK block diagram of a cascade 2-1-1 SC-$\Sigma\Delta$M modeled with the SIMULINK toolbox developed by Brigatti [33]. The integrator

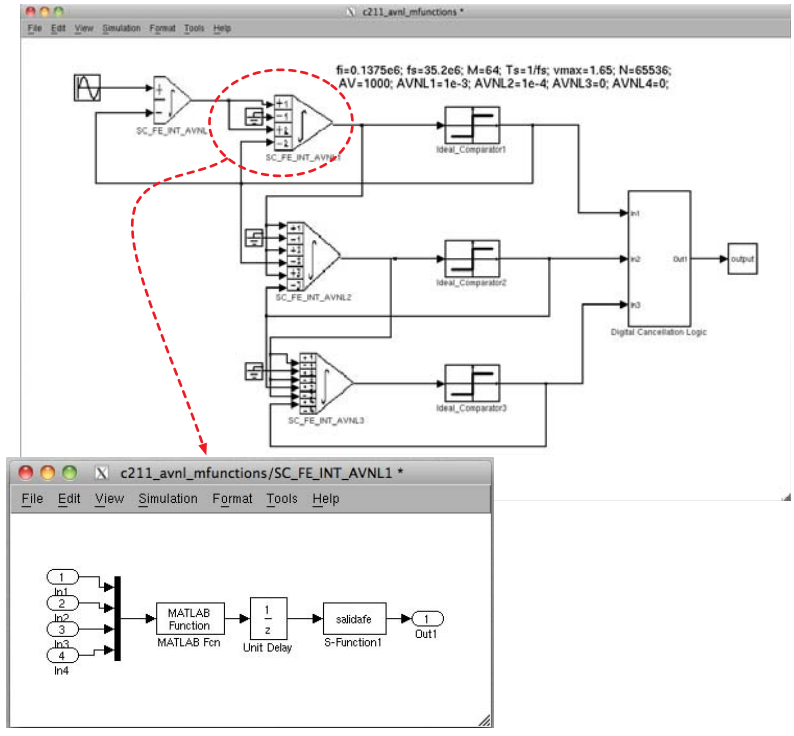


Figure 3.13 SIMULINK model of a cascade 2-1-1 SC-ΣΔM with nonlinear DC gain modeled using the M-file shown in Figure 3.10.

behavioral models already include finite open-loop opamp DC gain, incomplete settling error, slew rate, mismatch capacitor ratio error, and thermal noise. In addition, the main nonlinear effects were added to the original model, namely nonlinear sampling switch on-resistance, nonlinear capacitors, and nonlinear open-loop opamp DC gain, the latter being modeled using the M-function given in Figure 3.10. The main parts of this block diagram, as well of those parameters required to simulate the modulator, are shown in Figure 3.14. A 2^{16}-point simulation in a 2.4-GHz core with 4-GB RAM takes 80 s. This CPU time can be reduced to 8 s if the ΣΔM building-block behavioral modes are implemented in C-code using the so-called SIMULINK *S-functions* [34].

A *System-function* or *S-function* is a computer language description of a SIMULINK block that can be written in MATLAB code, Ada, Fortran, or C-code [34]. The latter are special-purpose source files that allow inclusion of computation algorithms written in C to SIMULINK models. This approach speeds up the simulations—up to 50 times faster in some cases—compared to the use of MATLAB functions or M-files to code the behavioral models, even when the accelerator mode is used [5]. In addition to the benefits in terms of CPU time, the use of S-functions for the behavioral simulation of ΣΔMs allows designers to model circuit-level error mechanisms in a more accurate way, as will be described in the rest of this chapter.

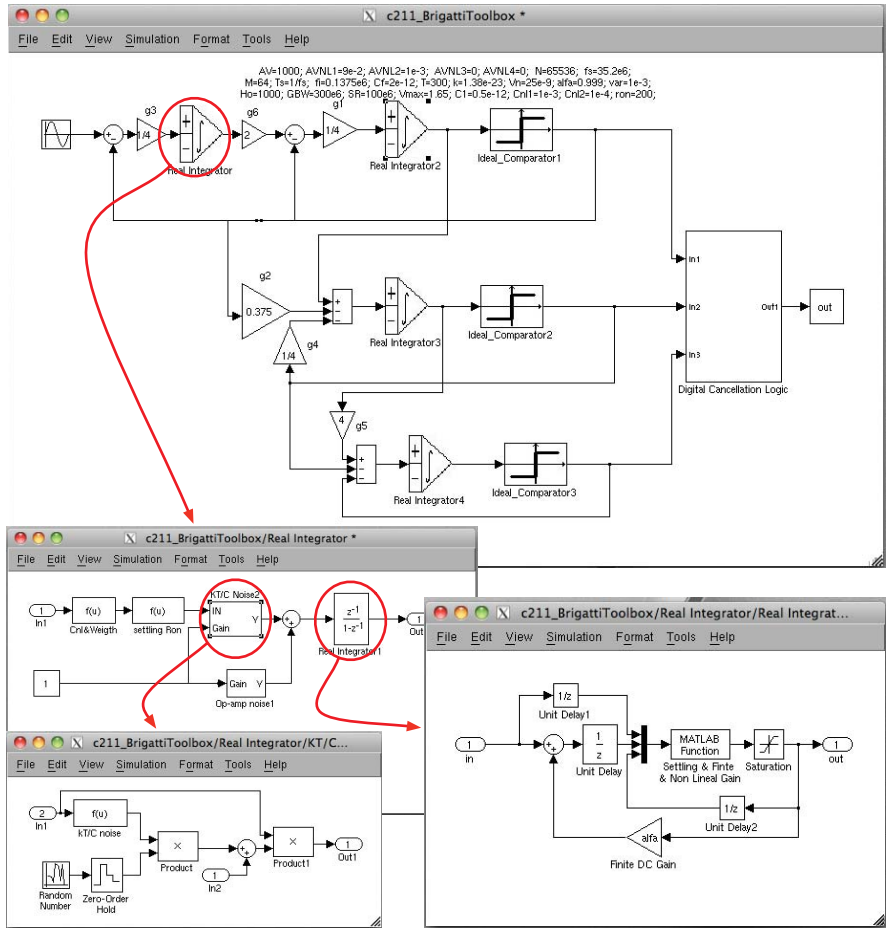


Figure 3.14 SIMULINK model of a cascade 2-1-1 SC-ΣΔM using Brigatti's toolbox [33].

3.3.4 Building Time-Domain Behavioral Models as SIMULINK C-MEX S-Functions

Figure 3.15a illustrates a step-by-step procedure to implement the behavioral model of a ΣΔM building block in the SIMULINK environment using C-coded S-functions. The main steps are the following [5]:

- *Definition of a Computational Model*. Given a ΣΔM building block that includes a set of nonidealities, a computational model that allows to calculate the output as a function of the input(s) and the internal states (if any) defined.
- *Generation of the C Code* corresponding to the computational model defined in the previous step.
- *Implementation of the Computational Model into a C-MEX S-Function*. To this purpose, SIMULINK provides different S-function template files that can accommodate

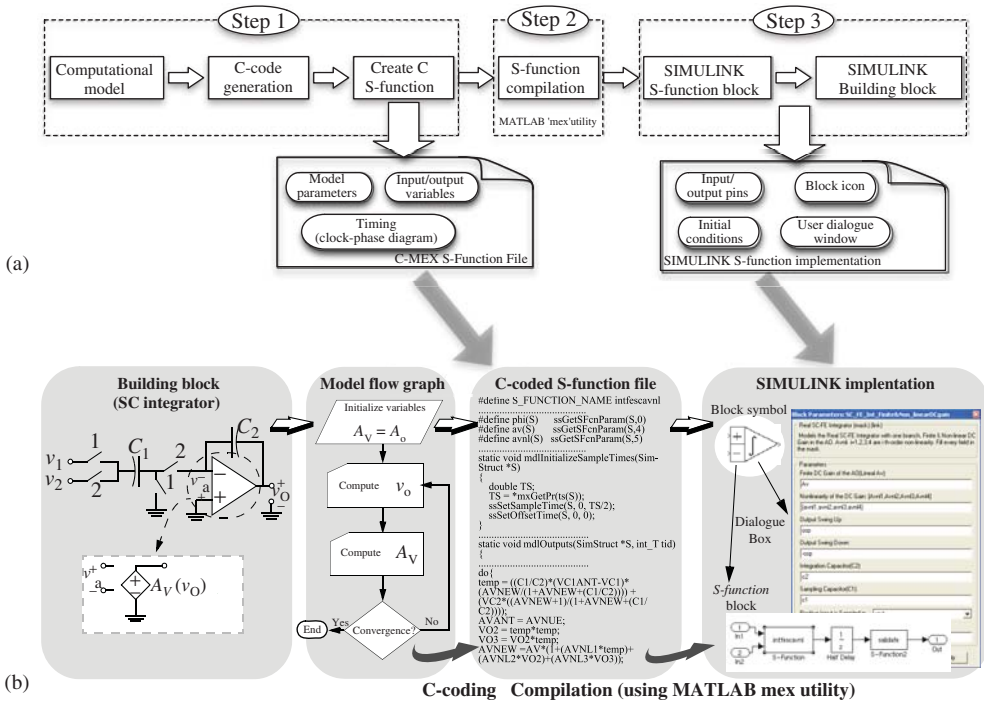


Figure 3.15 Procedure to incorporate a behavioral model into the MATLAB/SIMULINK environment using C-MEX S-functions: (a) conceptual step-by-step flow and (b) illustration of the process for an SC FE integrator with finite nonlinear DC gain.

the C-coded computational model generated in the previous step. Both DT and CT building blocks can be modeled, as will be shown in next sections. S-function template files are composed of several subsections, named *callback methods*, which are routines that perform different tasks required at each simulation stage. Among others, these tasks include variable initialization, computation of output variables, update of state variables, etc [34].

- *Compilation of the S-Function*. This is done using the `mex` utility provided by MATLAB [34]. MEX utility can be run from the MATLAB command prompt window, by typing `mex filename.c`, where `filename.c` can be either a single C-coded S-function file or a combination of different source C files.[11] The resulting object files are dynamically compiled and linked in SIMULINK when needed in a given simulation.
- *Incorporation of the Model into the SIMULINK Environment*. This can be done using the `S-function block` of the SIMULINK libraries [34]. A block diagram containing the S-function block is created including the input/output pins. The dialogue box is used for specifying the name of the underlying S-function. In addition, model parameters are also included in this box which can be used for modifying the parameter values.

[11] MATLAB MEX utility can be used for compiling one or more C/C++ or Fortran source files. This book focuses on C/C++ source files.

As an example, Figure 3.15b illustrates how to apply the main steps listed to create the S-function of an SC FE integrator with finite and nonlinear DC gain such as that shown in Figure 3.8. Note that the computational model flow graph in Figure 3.15b shows only the iterative procedure used for computing the nonlinear DC gain, whose entire MATLAB code is listed in Figure 3.10. When more nonidealities are to be considered, a more complex computation model—that appropriately takes all nonidealities into account in the right sequence—is needed, as will be detailed in next sections. For the sake of simplicity, the example in Figure 3.15b only shows some significant sections of the S-function file associated with the SC integrator model and how this S-function is incorporated into the SIMULINK environment.

Figure 3.16 shows the entire C-coded S-function file of an SC integrator with finite nonlinear DC gain, highlighting its main parts. Note that, in addition to the computational model itself (shown in Figure 3.17), an S-function includes other important functions that are required to simulate the behavioral model using SIMULINK. The majority of these functions and routines are included in the S-function template files provided by MATLAB. Therefore, the easiest way to model ΣΔM building blocks is to modify those template files by including the corresponding model parameters, state variables, input/output signals, clock-phase diagram scheme, etc.[12] MATLAB provides detailed documentation and a number of examples which are very useful to do this task [34].

For instance, the S-function file shown in Figure 3.16 is made up of the following parts and functions:

- `S-function name` and definitions. This section of the S-function is used to define the name of the block model and its corresponding parameters. In the example of Figure 3.16, the model parameters are `phi`, `ts`, `c1`, `c2`, `av`, `avnl`, `cp`, `opspos`, and `opsneg`, the latter being the positive and negative limits of the amplifier output swing.
- `mdlInitializeSizes`, where the number of inputs, outputs, and internal states are specified. Note that building blocks used in SIMULINK may have a vector of inputs, a vector of outputs, and a vector of states. The dimensions of these vectors are also specified in this part of the S-function. In this example, two input ports, zero state variables, and one output are defined, the size of all of them being one. The number of sampling rates (or sampling times)—one in this case—is also specified in the `mdlInitializeSizes` function.
- `mdlInitializeSampleTimes`, where the sampling time and the clock-phase scheme is defined. This function is used to define the building block at single-rate, at multirate, or if it is a CT circuit. The number of clock phases, the on-time period of each of them as well as the delays—called *offset* in the S-function—among them are also specified in this routine. For instance, in this example, a nonoverlapping two-phase clock with a 50% duty cycle is considered.
- `mdlStart`. This routine performs the initialization of the S-function, setting up the required model parameters and the initial values of the internal variable states.

[12] Although the procedure described in this section can be easily implemented using the S-function template files provided by MATLAB, note also that there is a SIMULINK block, named S-function Builder, that builds an S-function from specifications and C-code provided by the user [29].

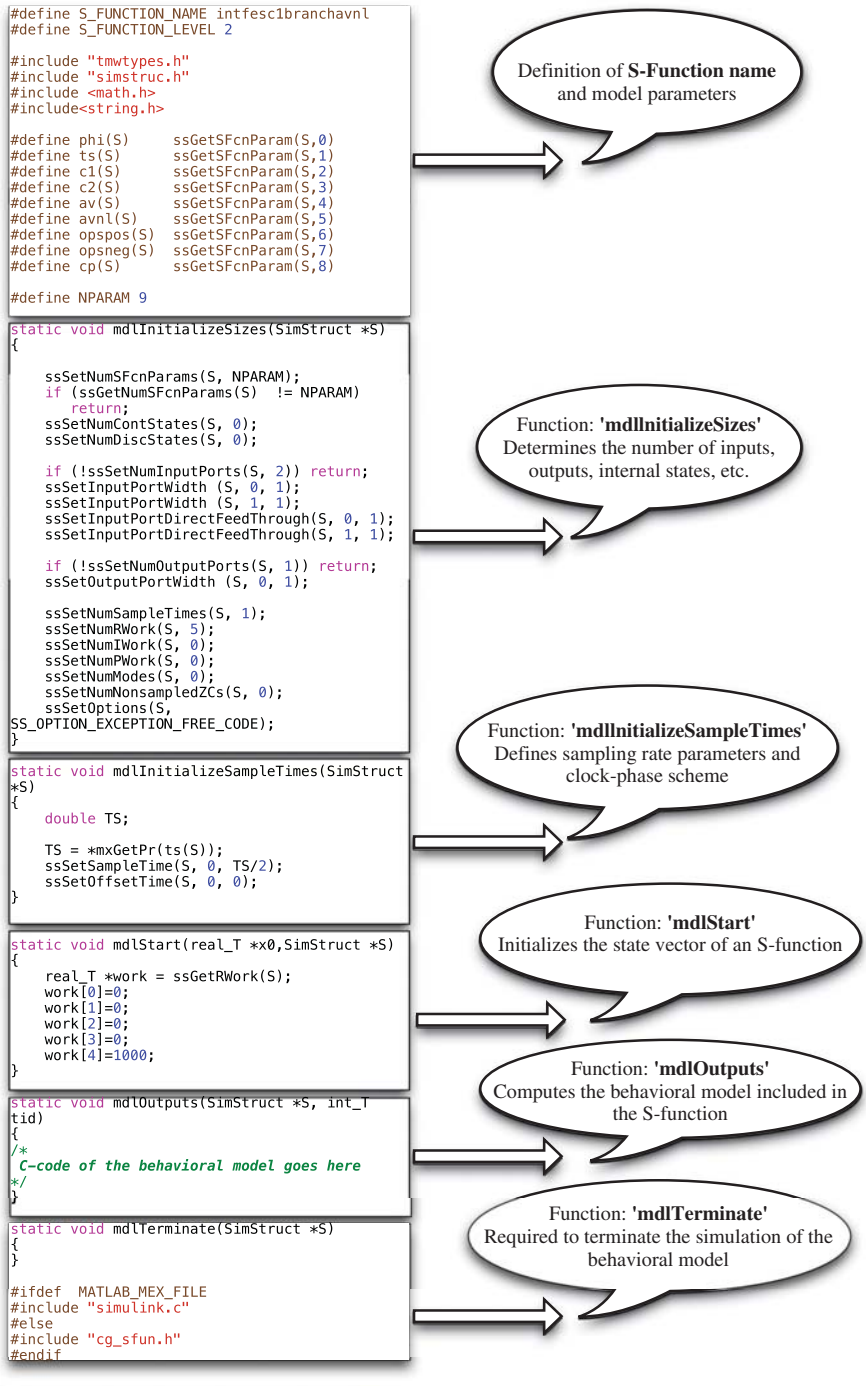
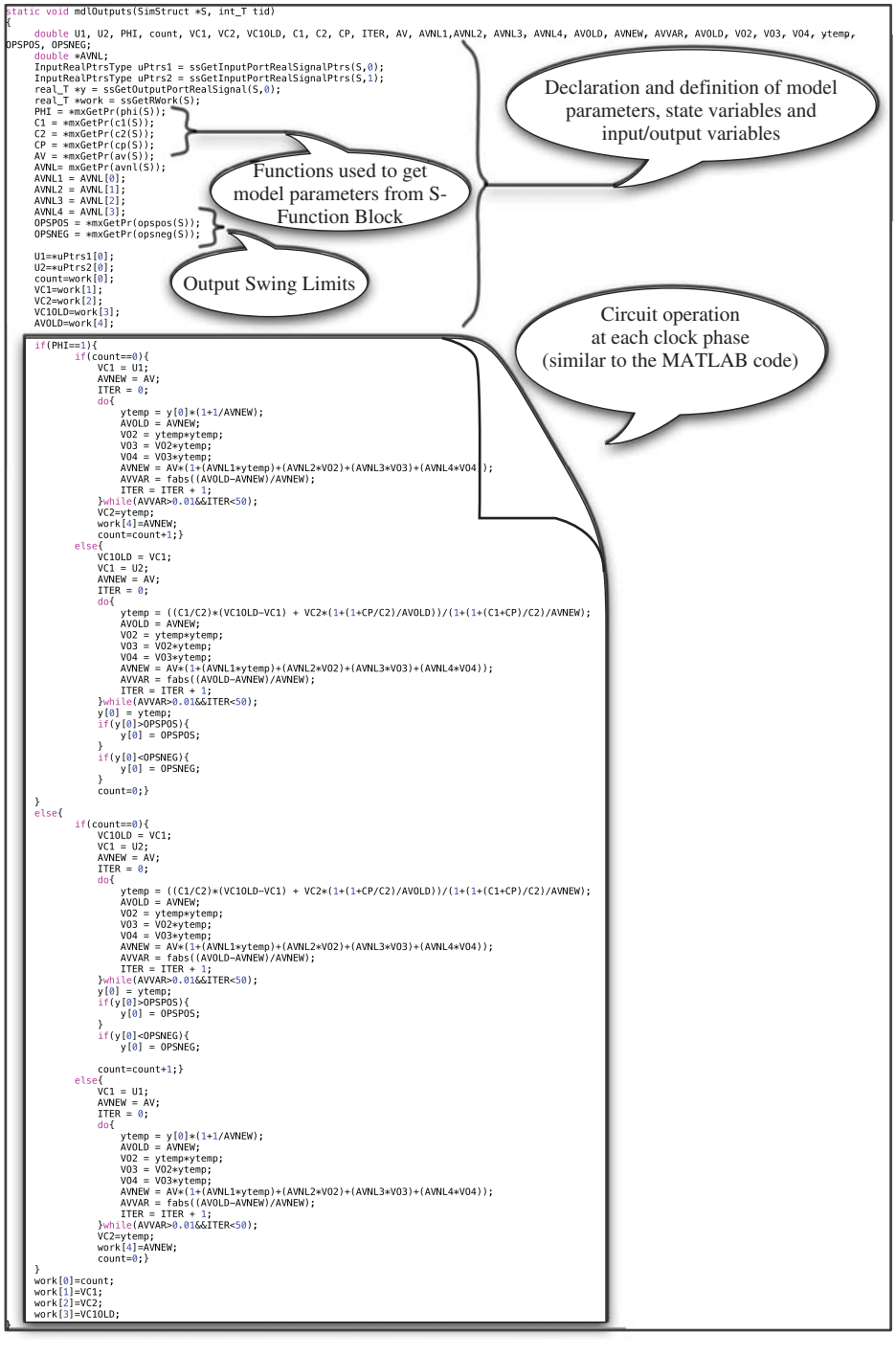


Figure 3.16 MATLAB C-coded S-function file of the integrator in Figure 3.10.




```
static void mdlOutputs(SimStruct *S, int_T tid)
{
    double U1, U2, PHI, count, VC1, VC2, VC1OLD, C1, C2, CP, ITER, AV, AVNL1,AVNL2, AVNL3, AVNL4, AVOLD, AVNEW, AVVAR, AVOLD, VO2, VO3, VO4, ytemp,
OPSPOS, OPSNEG;
    double *AVNL;
    InputRealPtrsType uPtrs1 = ssGetInputPortRealSignalPtrs(S,0);
    InputRealPtrsType uPtrs2 = ssGetInputPortRealSignalPtrs(S,1);
    real_T *y = ssGetOutputPortRealSignal(S,0);
    real_T *work = ssGetRWork(S);
    PHI = *mxGetPr(phi(S));
    C1 = *mxGetPr(c1(S));
    C2 = *mxGetPr(c2(S));
    CP = *mxGetPr(cp(S));
    AV = *mxGetPr(av(S));
    AVNL= mxGetPr(avnl(S));
    AVNL1 = AVNL[0];
    AVNL2 = AVNL[1];
    AVNL3 = AVNL[2];
    AVNL4 = AVNL[3];
    OPSPOS = *mxGetPr(opspos(S));
    OPSNEG = *mxGetPr(opsneg(S));

    U1=*uPtrs1[0];
    U2=*uPtrs2[0];
    count=work[0];
    VC1=work[1];
    VC2=work[2];
    VC1OLD=work[3];
    AVOLD=work[4];

    if(PHI==1){
        if(count==0){
            VC1 = U1;
            AVNEW = AV;
            ITER = 0;
            do{
                ytemp = y[0]*(1+1/AVNEW);
                AVOLD = AVNEW;
                VO2 = ytemp*ytemp;
                VO3 = VO2*ytemp;
                VO4 = VO3*ytemp;
                AVNEW = AV*(1+(AVNL1*ytemp)+(AVNL2*VO2)+(AVNL3*VO3)+(AVNL4*VO4));
                AVVAR = fabs((AVOLD-AVNEW)/AVNEW);
                ITER = ITER + 1;
            }while(AVVAR>0.01&&ITER<50);
            VC2=ytemp;
            work[4]=AVNEW;
            count=count+1;}
        else{
            VC1OLD = VC1;
            VC1 = U2;
            AVNEW = AV;
            ITER = 0;
            do{
                ytemp = ((C1/C2)*(VC1OLD-VC1) + VC2*(1+(1+CP/C2)/AVOLD))/(1+(1+(C1+CP)/C2)/AVNEW);
                AVOLD = AVNEW;
                VO2 = ytemp*ytemp;
                VO3 = VO2*ytemp;
                VO4 = VO3*ytemp;
                AVNEW = AV*(1+(AVNL1*ytemp)+(AVNL2*VO2)+(AVNL3*VO3)+(AVNL4*VO4));
                AVVAR = fabs((AVOLD-AVNEW)/AVNEW);
                ITER = ITER + 1;
            }while(AVVAR>0.01&&ITER<50);
            y[0] = ytemp;
            if(y[0]>OPSPOS){
                y[0] = OPSPOS;
            }
            if(y[0]<OPSNEG){
                y[0] = OPSNEG;
            }
            count=0;}
    }
    else{
        if(count==0){
            VC1OLD = VC1;
            VC1 = U2;
            AVNEW = AV;
            ITER = 0;
            do{
                ytemp = ((C1/C2)*(VC1OLD-VC1) + VC2*(1+(1+CP/C2)/AVOLD))/(1+(1+(C1+CP)/C2)/AVNEW);
                AVOLD = AVNEW;
                VO2 = ytemp*ytemp;
                VO3 = VO2*ytemp;
                VO4 = VO3*ytemp;
                AVNEW = AV*(1+(AVNL1*ytemp)+(AVNL2*VO2)+(AVNL3*VO3)+(AVNL4*VO4));
                AVVAR = fabs((AVOLD-AVNEW)/AVNEW);
                ITER = ITER + 1;
            }while(AVVAR>0.01&&ITER<50);
            y[0] = ytemp;
            if(y[0]>OPSPOS){
                y[0] = OPSPOS;
            }
            if(y[0]<OPSNEG){
                y[0] = OPSNEG;

            count=count+1;}
        else{
            VC1 = U1;
            AVNEW = AV;
            ITER = 0;
            do{
                ytemp = y[0]*(1+1/AVNEW);
                AVOLD = AVNEW;
                VO2 = ytemp*ytemp;
                VO3 = VO2*ytemp;
                VO4 = VO3*ytemp;
                AVNEW = AV*(1+(AVNL1*ytemp)+(AVNL2*VO2)+(AVNL3*VO3)+(AVNL4*VO4));
                AVVAR = fabs((AVOLD-AVNEW)/AVNEW);
                ITER = ITER + 1;
            }while(AVVAR>0.01&&ITER<50);
            VC2=ytemp;
            work[4]=AVNEW;
            count=0;}
    }
    work[0]=count;
    work[1]=VC1;
    work[2]=VC2;
    work[3]=VC1OLD;
}
```

Figure 3.17 C-code included in the *mdlOutputs* section of the S-function file in Figure 3.16.

- `mdlOutputs`. This is the main part of the S-function because the C-code of the behavioral model is introduced in this section. The routine computes the model and stores the corresponding results in an output array.
- `mdlTerminate` is a routine that must be included in the S-function file in order to keep the required template structure because SIMULINK calls it at the end of the simulation. In the more general case, this function is used for performing any action required at the end of the simulation, such as freeing memory. In this example, this function is not used and hence it is empty.

As stated previously, once the C-coded S-function source file (`intfesc1-branchavnl.c`) has been generated, it must be compiled to make it executable in MATLAB. In this example, the compilation is done by running the following sentence `mex intfesc1branchavnl.c` in the MATLAB command window. As a result, a compiled file—named MEX file—is generated. Indeed, the term MEX comes from MATLAB EXecutable [28].

The compiled MEX S-function is incorporated into a SIMULINK model using the `S-function block` available in the `User-Defined Functions` SIMULINK library. Figure 3.18 illustrates the SIMULINK block associated with the S-function of

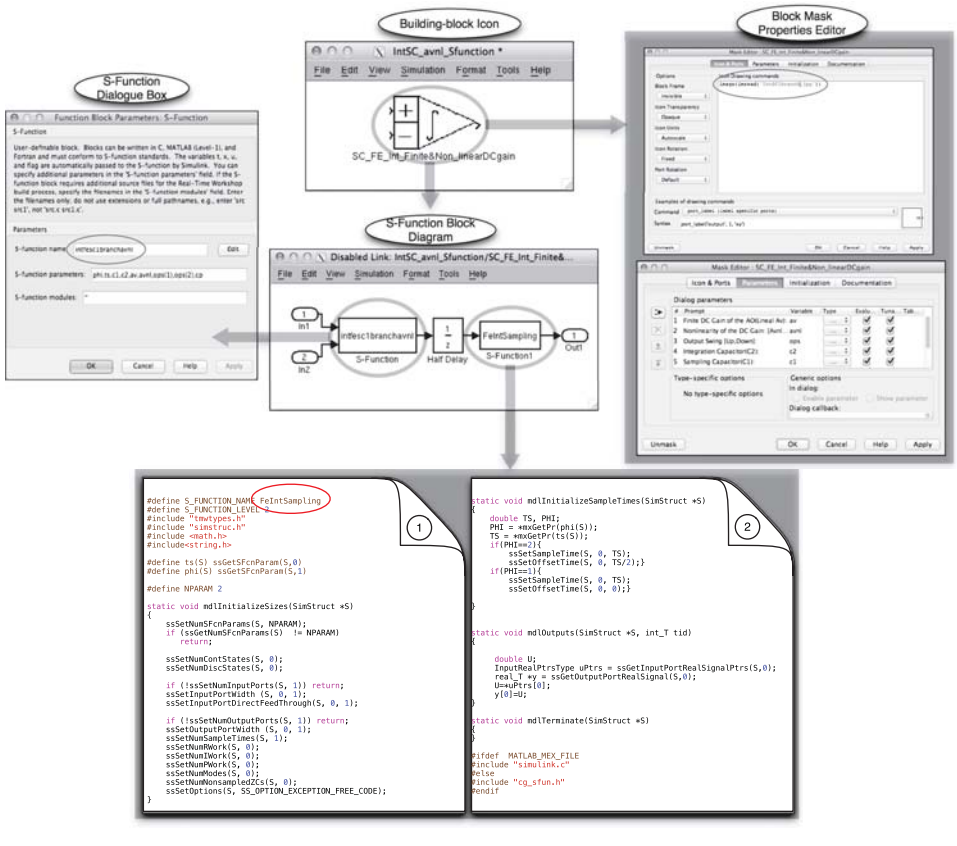


Figure 3.18 Illustrating the implementation of the S-function of Figure 3.16 in SIMULINK.

Figure 3.16 showing the main dialogue windows. The S-function name and the model parameters are entered into the `Block Parameter` dialogue box. It is very important that the block name, as well as the model parameters, are the same as those included in the source file of Figure 3.16. The input/output terminal ports are connected to the S-function block in the SIMULINK block diagram.

Note that an additional S-function block named `FeIntSampling` is included in the block diagram of Figure 3.18. This block—whose C-coded S-function source file is shown in Figure 3.16—is used for properly sampling the integrator output at the correct time instant. This additional S-function is combined with a `Half Delay` SIMULINK block to have a more precise control of the clock phase at which the integrator output is taken and to collect only one output sample per clock period.

In order to make the use of S-functions easy to use in more complex systems, a block mask can be created. A specific mask icon can be used instead of the SIMULINK subsystem's standard icon. This is very useful to identify different building blocks in a given system such as a ΣΔM and this will be shown in the next sections. The mask also has a dialogue box where the designer can change the model parameters in a simple way. These model parameters are passed to the S-function and dynamically linked into SIMULINK when they are needed during a simulation.

The procedure described in this section must be followed in order to create the behavioral models of the different ΣΔM building blocks. Next section explains in detail some of the most important building-block models, as well as their implementation in the MATLAB/SIMULINK environment using C-coded S-functions.

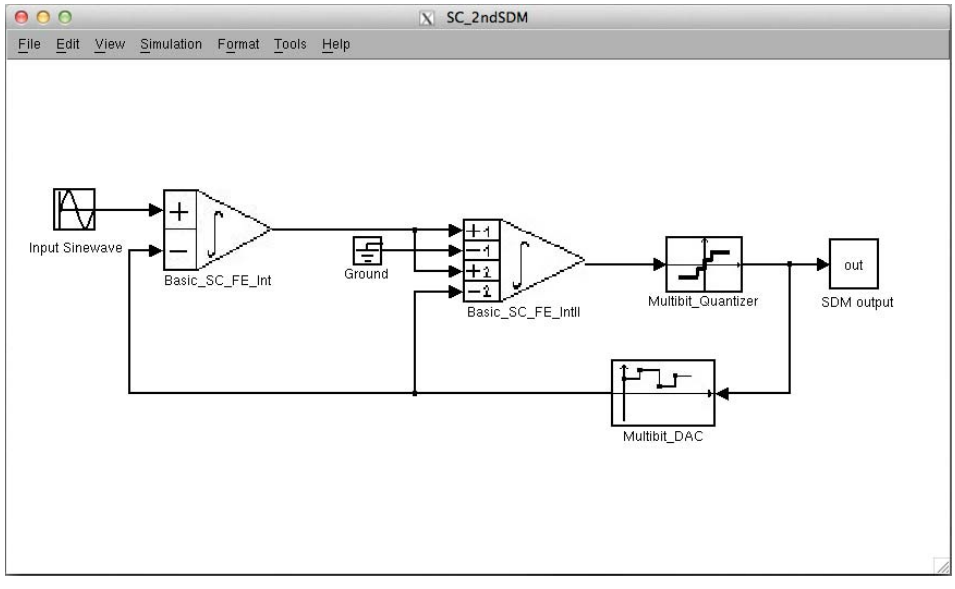
3.4 Efficient Behavioral Modeling of ΣΔM Building Blocks using C-MEX S-Functions

All ΣΔM building blocks can be modeled by C-MEX S-functions using the methodology described in the previous section. This way, any ΣΔM architecture can be simulated at system level in an efficient way in terms of accuracy and CPU time. As an illustration, Figure 3.19 shows the SIMULINK block diagram of a second-order ΣΔM implemented using SC integrators (Figure 3.19a) and Gm-C integrators (Figure 3.19b). These block diagrams contain S-function blocks that model integrators, quantizers, and feedback DACs. Other auxiliary subcircuits, such as the output impedances of the Gm-C integrators (made up of R_o and C) and the digital latches are also included in the models.

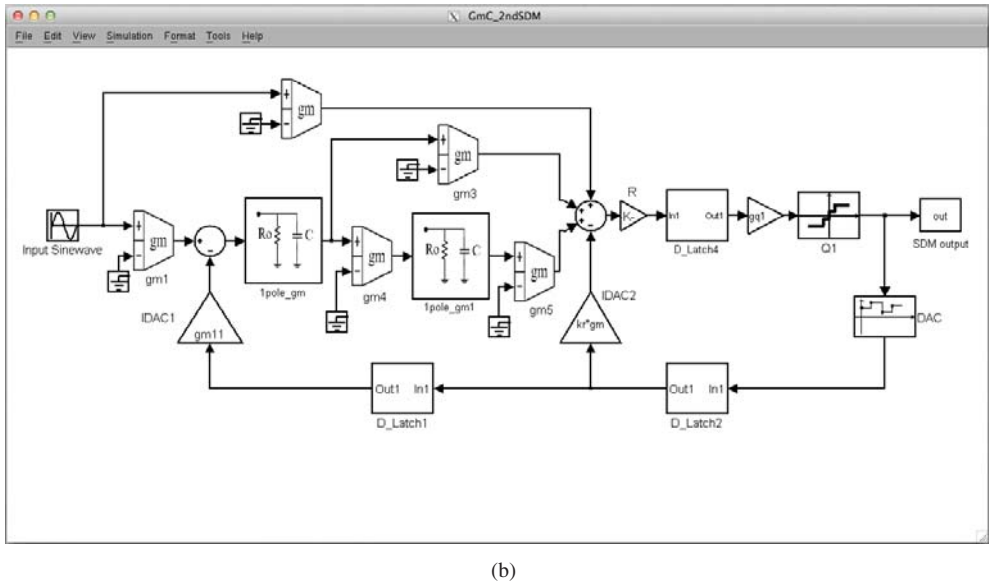
Each block in Figure 3.19 can be implemented using different circuit topologies which are affected by a number of circuit error mechanisms. A detailed explanation of all different ΣΔM building-block behavioral models and their corresponding S-functions is beyond the scope of this book. Instead, this section describes the model of the most essential elements of ΣΔMs—that is, integrators, quantizers, and DACs—taking into account the most important nonideal effects discussed in Chapter 2 and specially emphasizing on those aspects related to their implementation using C-MEX S-functions.

3.4.1 Modeling of SC Integrators using S-Functions

Let us consider again the conceptual schematic of an SC FE integrator shown in Figure 3.8a. Note that a single-ended schematic is shown for the sake of simplicity.



(a)



(b)

Figure 3.19 Illustrating the behavioral model of a second-order ΣΔM using S-functions: (a) SC implementation and (b) Gm-C implementation.

However, the behavioral model described below takes into account a fully-differential topology.

Its ideal behavior is described by the finite-difference equation given in Equation 3.1. The impact of finite OTA DC gain was considered in Section 3.3, taking into account both linear and nonlinear effects. However, in practice, the performance of SC integrators is degraded by a number of error mechanisms, as described in Chapter 2. An accurate behavioral model must take into account the contribution of the main circuit errors, as well as the clock phases in which they affect the performance of the circuit to be modeled—an SC FE integrator in this case. To this purpose, the effect of all SC circuit nonidealities is computed by following the iterative procedure shown in the flow graph of Figure 3.20 [5].

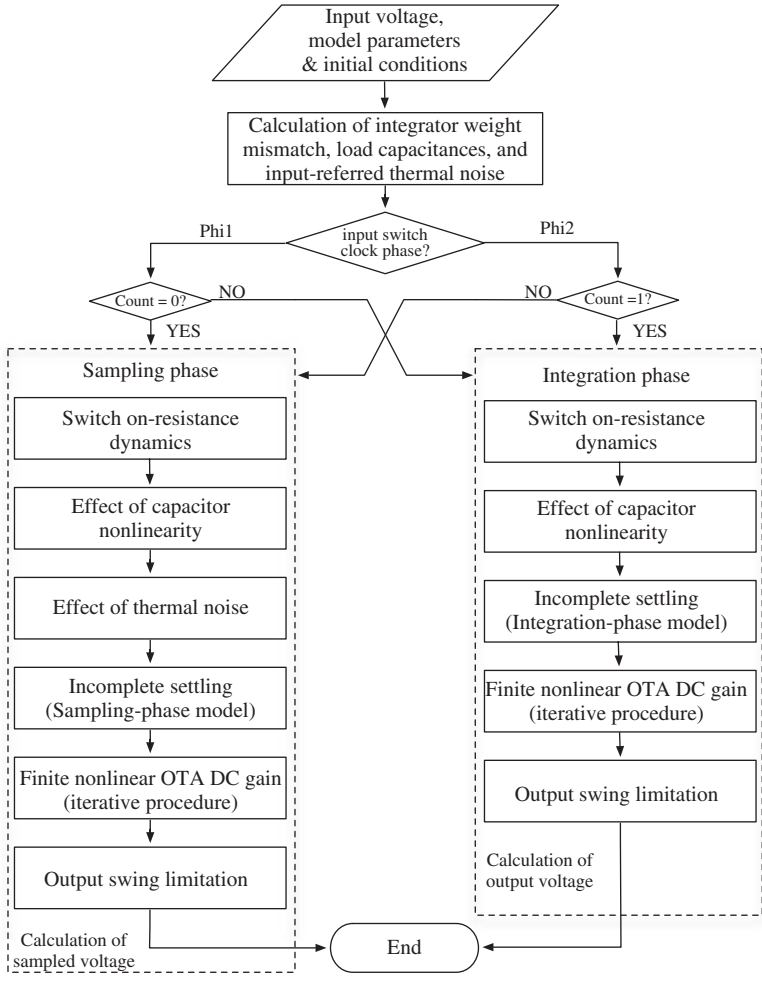


Figure 3.20 Flowchart of the SC FE integrator computational model.

The model starts by loading the values of the required model parameters, the input signal, and the initial conditions—that is, the voltages at internal nodes of the integrator (including the values stored on the sampling and integration capacitor) stored in the previous clock period and which are usually set to zero at the beginning of a simulation. Before starting to compute the behavioral model, some initial calculations are done, namely, the equivalent input-referred noise, the actual value of the integrators weight due to capacitor mismatch, and the parasitic and load capacitances at different nodes of the integrator during sampling and integration clock phases.

The different integrator clock phases that correspond to the two main branches in Figure 3.20 are selected according to the value of the clock-phase counter `count` and the input switch phase—either ϕ_1 or ϕ_2, as illustrated in Figure 3.11. At each clock phase, the effect of main SC circuit nonidealities is taken into account, namely, finite (linear and nonlinear) switch on-resistance, capacitor nonlinearity, thermal noise, incomplete settling, finite (linear and nonlinear) OTA DC gain (modeled as described in Section 3.3), and output swing limitation. The behavioral model of these errors is based on the nonideal equations explained in Chapter 2. These equations can be codified in C and incorporated in an S-function as illustrated in Figures 3.21–3.23.

Note that the S-function file follows the same structure as that shown in Figure 3.16, but adding more detailed pieces of information to incorporate the mentioned circuit errors into the behavioral model. For the sake of simplicity, Figures 3.21–3.23 show only the most important parts of the S-function file. Although additional comments have been included in the C-code, the following sections explain the main parts of the S-function by linking the C-code with the design equations described in Chapter 2 for the different SC circuit nonidealities.

Capacitor Mismatch and Nonlinearity

As discussed in Section 2.3, capacitor mismatch is modeled as a random deviation of the integrators weight from its nominal value g. This random variation is included in the C-code [35] as a normal (or Gaussian) distribution with a mean value of g (denoted as `MEAN` in Figure 3.21) and a variance that is provided as a model parameter named `VARIANCE`.

Another nonideal effect associated with the integrator capacitors is the nonlinear dependence of their capacitance on the voltage drop across them (v), which is modeled as a polynomial function given by

$$C_{s,i}(v) = C_{s,i_{(nominal)}} \cdot (1 + CNL_1 \cdot v + CNL_2 \cdot v^2) \tag{3.12}$$

where $C_{s,i_{(nominal)}}$ are the nominal values of the sampling (C_s) or the integration capacitors (C_i) and $CNL_{1,2}$ are, respectively, the first- and second-order nonlinear voltage coefficients represented in the model by `CNL1,2` (Figure 3.22). This way, the charge stored in these capacitors is given by [36]

$$Q_{s,i}(v) = \int_0^v C_{s,i}(v)dv = C_{s,i_{(nominal)}} \cdot \left(v + \frac{CNL_1}{2} \cdot v^2 + \frac{CNL_2}{3} \cdot v^3\right) \tag{3.13}$$

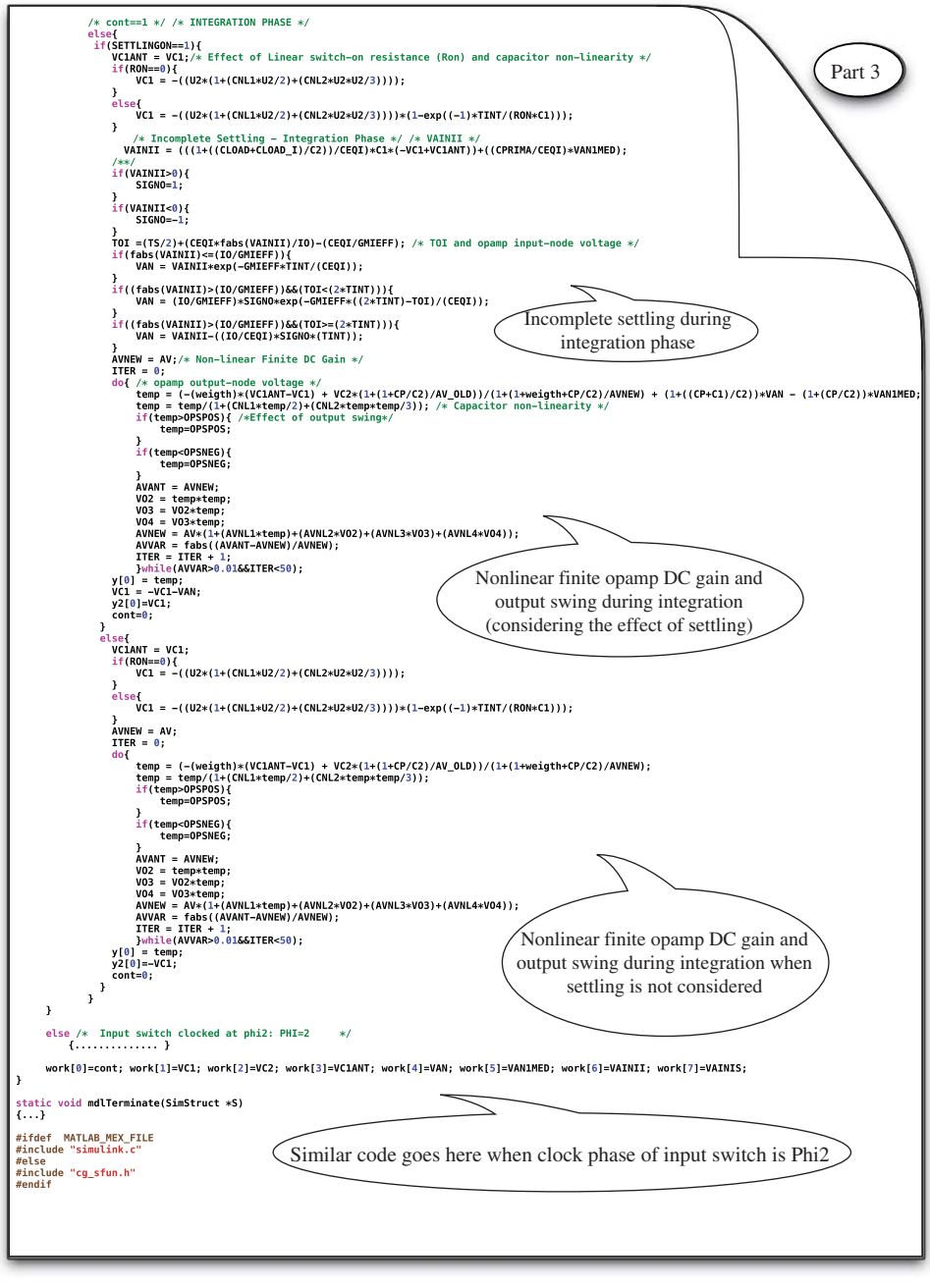
Part 1

```
#define S_FUNCTION_NAME intfesc1branch_allerrors
#define S_FUNCTION_LEVEL 2
...
#define phi(S)          ssGetSFcnParam(S,0)   /* Input switch clock phase */
#define ts(S)           ssGetSFcnParam(S,1)   /* Sampling time */
#define c(S)            ssGetSFcnParam(S,2)   /* Capacitor values and non-linear coefficients */
#define var(S)          ssGetSFcnParam(S,3)   /* Integrator weight's variance, eq. vrms input and temperature */
#define ops(S)          ssGetSFcnParam(S,4)   /* Output swing limits */
#define ao(S)           ssGetSFcnParam(S,5)   /* OTA Finite DC gain, transconductance and maximum output current */
#define setnoise(S)     ssGetSFcnParam(S,6)   /* Flag that activates the effect of thermal noise   */
#define avnl(S)         ssGetSFcnParam(S,7)   /* Non-linear coefficients of the finite DC gain */
#define cp(S)           ssGetSFcnParam(S,8)   /* OTA parasitic capacitances and load capacitances */
#define settlingon(S)   ssGetSFcnParam(S,9)   /* Flag that activates incomplete settling model */
#define setin(S)        ssGetSFcnParam(S,10)  /* Integrator identifier used in the incomplete settling model */
#define idown(S)        ssGetSFcnParam(S,11)  /* Integrator identifier */
#define idnext(S)       ssGetSFcnParam(S,12)  /* Identifier of the Integrator connected at the Integrator output */
#define setloads(S)     ssGetSFcnParam(S,13)  /* Flag that activates additional load capacitances  */
#define CLOAD_S(S)      ssGetSFcnParam(S,14)  /* Additional load capacitance in the sampling phase */
#define CLOAD_I(S)      ssGetSFcnParam(S,15)  /* Additional load capacitance in the integration phase */
#define setphases(S)    ssGetSFcnParam(S,16) /* Flag that activates the reduction of the clock phase switch-on time */
#define phases_reduction(S)    ssGetSFcnParam(S,17) /* Duty cycle, reduction of the integration and sampling time */
#define BW(S)     ssGetSFcnParam(S,18)          /*  SIGNAL BANDWIDTH */
#define numid(S)    ssGetSFcnParam(S,19)       /* Integrator identification number, i.e. int1, int2, int3... */
#define NPARAM 20

static void mdlInitializeSizes(SimStruct *S)
{...
     ssSetInputPortWidth (S, 3, DYNAMICALLY_SIZED); /* Input port where the "From" block is connected */
...}

static void mdlInitializeSampleTimes(SimStruct *S) /* Clock-phase definition (the same as for ideal SC Integ. */
{...}

static void mdlStart(real_T *x0,SimStruct *S)  /*Initial conditions (similar to ideal SC integrator */
{...}

static void mdlOutputs(SimStruct *S, int_T tid)
{    /* Declaration of model variables and parameters */
     double U1, U2, PHI, cont, VC1, VC2, VC1ANT
...
       InputRealPtrsType uPtrs1 = ssGetInputPortRealSignalPtrs(S,0);
...
       InputRealPtrsType uPtrs4 = ssGetInputPortRealSignalPtrs(S,3);
     real_T *y = ssGetOutputPortRealSignal(S,0);
...
      /* DEFINITION OF MODEL PARAMETERS */
     PHI = *mxGetPr(phi(S)); /* Input switch clock phase */
     setloads = *mxGetPr(setloads(S)); /* Additional Load capacitances during the sampling (S) and integration phase (I) */
     if (setloads) { CLOAD_S = *mxGetPr(CLOAD_S(S)); CLOAD_I = *mxGetPr(CLOAD_I(S));}
     else {CLOAD_S = 0; CLOAD_I = 0;}

     setphases = *mxGetPr(setphases(S));  /* Duration of Sampling/Integration clock-phase */
     if (setphases==1){
       phases_reduction = mxGetPr(phases_reduction(S)); duty_cycle=phases_reduction[0];
       int_red=phases_reduction[1];samp_red=phases_reduction[2];}
     else{duty_cycle=50; int_red=0; samp_red=0;}

     p = mxGetPr(c(S)); /* Capacitor values and non-linear coefficients */
     C1 = p[0]; C2 = p[1]; CNL1 = p[2]; CNL2 = p[3];
     p = mxGetPr(ao(S)); /* OTA Finite DC gain (AV), transconductance (GM) and maximum output current (IO) */
     AV = p[0]; GM = p[1]; IO = p[2];
     p = mxGetPr(avnl(S)); /* Non-linear coefficients of the finite DC gain */
     AVNL1 = p[0]; AVNL2 = p[1]; AVNL3 = p[2]; AVNL4 = p[3];
     p = mxGetPr(var(S)); /* Integrator weight's variance, eq. vrms input and temperature */
     VARIANCE = p[0]; INPSD = p[1]; TEMP = p[2]; RON = p[3];
     p = mxGetPr(cp(S)); /* OTA Input parasitic capacitances and load capacitance */
        CP1 = p[0]; CP2 = p[1]; CLOAD = p[2];
     TS = *mxGetPr(ts(S)); /* Sampling time */
     FS = 1/TS; /* Sampling frequency */
     p = mxGetPr(ops(S)); /* Output swing limits */
     OPSPOS = p[0]; OPSNEG = p[1];
     SETTLINGON = *mxGetPr(settlingon(S)); /* Settling flag that activates incomplete settling model */
     SETIN = *mxGetPr(setin(S)); /* Integrator identifier to be used in the incomplete settling error computation */
     U1=*uPtrs1[0]; /* Value of signal connected to input branch #1 */
     U2=*uPtrs2[0]; /* Value of signal connected to input branch #2 */
     UNOISE=*uPtrs3[0]; /* Value of noise source connected at the input */
     N=*uPtrs4[0]; /* Number of input branches */
     cont=work[0]; /* Counter used to determine clock phase: sampling phase / integration phase */

/*      Initial values of VC1(n), VC1(n-1)... */
     VC1=work[1]; VC2=work[2]; VC1ANT=work[3];VAN=work[4]; VAN1MED=work[5];VAINII=work[6]; VAINIS=work[7]; AV_OLD=work[8];

/*   Reduction of clock-phase duration */
     TINT=TS*(duty_cycle/100)-int_red; /* Integration phase duration */
     TSAMP=TS*(1-duty_cycle/100)-samp_red; /*Sampling phase duration */

/*   Mismatch Parameters and calculation of integrator's weight      */
     MEAN = C1/C2; /* Nominal value of the integrator weight */
     SEMILLA = 1; /*Gaussian random number generator */
     a=16807; m=2147483647; seed=SEMILLA; zno=seed; zn=(a*zno)%m;noise_int=zn/m;
     zno=zn; zn=(a*zno)%m; seed=zn;noise_int2=zn/m;
     weigth=MEAN+(VARIANCE*sqrt(-2*log(noise_int))*cos(2*3.1416*noise_int2));
```

Figure 3.21 Excerpt of the MATLAB C-coded S-function file of an SC FE integrator considering all circuit errors: model parameters, definition of clock-phase timing, and calculation of integrator weight including mismatch (**Part 1 of 3**).


```
/*   Calculation of the equivalent load capacitance in the integration (CEQI) and sampling phase (CEQS)*/
       CP=CP2; CSUM=0;
       if (setloads==1)
       {
           CEQI = C1+CP+((CLOAD+CLOAD_I)*(1+((C1+CP)/C2))); CPRIMA = CP + ((CLOAD+CLOAD_I)*(1+(CP/C2)));
           if (SETIN==1)
           {
               for(i=N+1;i<=2*N;i++){CSUM=CSUM + (*uPtrs4[i]);}
               CEQS = CP+(CLOAD+CLOAD_S+CSUM)*(1+CP/C2);
           }
           else {CEQS= CP + ((CLOAD+CLOAD_S)*(1+(CP/C2)));}
       }
       else
       {
           CEQI = C1+CP+(CLOAD*(1+((C1+CP)/C2))); CPRIMA = CP + (CLOAD*(1+(CP/C2)));
           if (SETIN==1)
           {
               for(i=N+1;i<=2*N;i++){ CSUM=CSUM + (*uPtrs4[i]);}
               CEQS = CP+(CLOAD+CSUM)*(1+CP/C2);
           }
           else
           {
               CEQS=CPRIMA;
           }
       }

/*   Thermal Noise Parameters and Calculation of the input referred noise voltage */

       setnoise = *mxGetPr(setnoise(S)); K = 1.380658*pow(10,(-23));
       SIN_PHI1 = K*TEMP/(C1*FS); /*   Switches' contribution (PHI1) */
       z1I=GM/CPRIMA; p1I=GM/CEQI; p2=CEQI/CPRIMA/(2*RON*C1);
           BWn_SPHI2=p1I*p2*(p1I*p2+(z1I*z1I))/(4*z1I*z1I*(p1I+p2));
       SS=2*K*TEMP*2*RON; SIN_PHI2 = 2*BWn_SPHI2*SS/FS; /*   Switches' contribution (PHI2) */
       SOP = INPSD*INPSD; BWn_OP= p1I*p2/(4*(p1I+p2));SIN_OP=2*BWn_OP*SOP/FS; /*   Amplifier contribution */

       if(RON==0){DELTANOISE=sqrt(12*SIN_OP*FS);}
       else {DELTANOISE = sqrt(12*(2*SIN_PHI1+2*SIN_PHI2+SIN_OP)*FS);} /* SIN_PHI1,2 ARE DOUBLED TO TAKE INTO ACCOUNT THE DIFFERENTIAL IMPLEMENTATION */

       if (setnoise!=1) {DELTANOISE=0;}

       VNOISE = UNOISE*DELTANOISE;

       /* Code that accounts for the effect of the RON on the dynamics */
           ...
           GMSEFF=GM*CEQS/(CEQS+(2*RNON*CSUM*GM)); /* RNON = switch-on resistance of the switches connected at the output of the integrator */
           GMIEFF=GM*CEQI/(CEQI+(2*RON*C1*GM));

          /*  Input switch clocked at phi1: PHI=1     */

       if(PHI==1){
            if(cont==0){ /* SAMPLING PHASE */
              if(SETTLINGON==1){ /* Effect of Linear switch-on resistance (Ron), capacitor non-linearity and thermal noise */
                if(RON==0){VC1 = -((U1*(1+(CNL1*U1/2)+(CNL2*U1*U1/3)))+VNOISE);}
                else{VC1 = -((U1*(1+(CNL1*U1/2)+(CNL2*U1*U1/3)))+VNOISE)*(1-exp((-1)*TSAMP/(RON*C1)));}
                if (SETIN==1) /* Incomplete Settling - Sampling Phase */
                    {
                        VAINIS = VAN;
                        for(i=1;i<=N;i++){VAINIS = VAINIS - (*uPtrs4[N+i]/CEQS)*(y[0]- (*uPtrs4[i]));}
                    }
                else {VAINIS = VAN;}
                if(VAINIS>=0){SIGNO=1;}
                if(VAINIS<0){SIGNO=-1;}
                            TOS = (CEQS*fabs(VAINIS)/IO)-(CEQS/GMSEFF); /* TOS and opamp input-node voltage */
                            if(fabs(VAINIS)<=(IO/GMSEFF)){VAN1MED = VAINIS*exp(-GMSEFF*TSAMP/(CEQS));}
                if((fabs(VAINIS)>(IO/GMSEFF))&&(TOS<(TSAMP))){VAN1MED = (IO/GMSEFF)*SIGNO*exp(-GMSEFF*((TSAMP)-TOS)/(CEQS));}
                if((fabs(VAINIS)>(IO/GMSEFF))&&(TOS>=(TSAMP))){VAN1MED = VAINIS-((IO/CEQS)*SIGNO*(TSAMP));}
                AVNEW = AV; /* Non-linear Finite DC Gain - Iterative procedure */
                ITER = 0;
                do{
                    temp = y[0]*(1+1/AVNEW)+((1+(CP/C2))*(VAN1MED-VAN)); /* opamp output-node voltage */
                    temp = temp*(1+(CNL1*temp/2)+(CNL2*temp*temp/3)); /* Capacitor non-linearity */
                    AVANT = AVNEW; VO2 = temp*temp; VO3 = VO2*temp; VO4 = VO3*temp;
                    AVNEW = AV*(1+(AVNL1*temp)+(AVNL2*VO2)+(AVNL3*VO3)+(AVNL4*VO4));
                                    AVVAR = fabs((AVANT-AVNEW)/AVNEW); ITER = ITER + 1;
                }while(AVVAR>0.01&&ITER<50);
                VC2=temp;
                work[8]=AVNEW;
                if(VC2>OPSPOS){ /* Effect of output swing */
                    VC2 = OPSPOS; /* MAXIMUM VALUE OF THE SATURATION VOLTAGE */
                }
                if(VC2<OPSNEG){
                    VC2 = OPSNEG; /* MINIMUM VALUE OF THE SATURATION VOLTAGE */
                }
                cont=cont+1;
              }
              else{
                if(RON==0){VC1 = -((U1*(1+(CNL1*U1/2)+(CNL2*U1*U1/3)))+VNOISE);}
                else{VC1 = -((U1*(1+(CNL1*U1/2)+(CNL2*U1*U1/3)))+VNOISE)*(1-exp((-1)*TSAMP/(RON*C1)));}
                AVNEW = AV;
                ITER = 0;
                do{
                    temp = y[0]*(1+1/AVNEW);
                    temp = temp*(1+(CNL1*temp/2)+(CNL2*temp*temp/3));
                    AVANT = AVNEW; VO2 = temp*temp; VO3 = VO2*temp; VO4 = VO3*temp;
                    AVNEW = AV*(1+(AVNL1*temp)+(AVNL2*VO2)+(AVNL3*VO3)+(AVNL4*VO4));
                    AVVAR = fabs((AVANT-AVNEW)/AVNEW); ITER = ITER + 1;
                }while(AVVAR>0.01&&ITER<50);
                VC2=temp;
                work[8]=AVNEW;
                if(VC2>OPSPOS){ /*Effect of output swing*/
                    VC2 = OPSPOS;
                }
                if(VC2<OPSNEG){
                    VC2 = OPSNEG;
                }
                cont=cont+1;
              }
```

Figure 3.22 Excerpt of MATLAB C-coded S-function file of an SC FE integrator considering all circuit errors: calculation of equivalent load capacitances, input-referred thermal noise, capacitor nonlinearity, incomplete settling during sampling phase, nonlinear finite opamp DC gain, and output swing **(Part 2 of 3)**.




```
            /* cont==1 */ /* INTEGRATION PHASE */
            else{
             if(SETTLINGON==1){
                VC1ANT = VC1;/* Effect of Linear switch-on resistance (Ron) and capacitor non-linearity */
                if(RON==0){
                    VC1 = -((U2*(1+(CNL1*U2/2)+(CNL2*U2*U2/3))));
                }
                else{
                    VC1 = -((U2*(1+(CNL1*U2/2)+(CNL2*U2*U2/3))))*(1-exp((-1)*TINT/(RON*C1)));
                }
                  /* Incomplete Settling - Integration Phase */ /* VAINII */
                  VAINII = (((1+((CLOAD+CLOAD_I)/C2))/CEQI)*C1*(-VC1+VC1ANT))+((CPRIMA/CEQI)*VAN1MED);
                /**/
                if(VAINII>0){
                    SIGNO=1;
                }
                if(VAINII<0){
                    SIGNO=-1;
                }
                TOI =(TS/2)+(CEQI*fabs(VAINII)/IO)-(CEQI/GMIEFF); /* TOI and opamp input-node voltage */
                if(fabs(VAINII)<=(IO/GMIEFF)){
                    VAN = VAINII*exp(-GMIEFF*TINT/(CEQI));
                }
                if((fabs(VAINII)>(IO/GMIEFF))&&(TOI<(2*TINT))){
                    VAN = (IO/GMIEFF)*SIGNO*exp(-GMIEFF*((2*TINT)-TOI)/(CEQI));
                }
                if((fabs(VAINII)>(IO/GMIEFF))&&(TOI>=(2*TINT))){
                    VAN = VAINII-((IO/CEQI)*SIGNO*(TINT));
                }
                AVNEW = AV;/* Non-linear Finite DC Gain */
                ITER = 0;
                do{ /* opamp output-node voltage */
                    temp = (-(weigth)*(VC1ANT-VC1) + VC2*(1+(1+CP/C2)/AV_OLD))/(1+(1+weigth+CP/C2)/AVNEW) + (1+((CP+C1)/C2))*VAN - (1+(CP/C2))*VAN1MED;
                    temp = temp/(1+(CNL1*temp/2)+(CNL2*temp*temp/3)); /* Capacitor non-linearity */
                    if(temp>OPSPOS){ /*Effect of output swing*/
                        temp=OPSPOS;
                    }
                    if(temp<OPSNEG){
                        temp=OPSNEG;
                    }
                    AVANT = AVNEW;
                    VO2 = temp*temp;
                    VO3 = VO2*temp;
                    VO4 = VO3*temp;
                    AVNEW = AV*(1+(AVNL1*temp)+(AVNL2*VO2)+(AVNL3*VO3)+(AVNL4*VO4));
                    AVVAR = fabs((AVANT-AVNEW)/AVNEW);
                    ITER = ITER + 1;
                    }while(AVVAR>0.01&&ITER<50);
                y[0] = temp;
                VC1 = -VC1-VAN;
                y2[0]=VC1;
                cont=0;
             }
             else{
                VC1ANT = VC1;
                if(RON==0){
                    VC1 = -((U2*(1+(CNL1*U2/2)+(CNL2*U2*U2/3))));
                }
                else{
                    VC1 = -((U2*(1+(CNL1*U2/2)+(CNL2*U2*U2/3))))*(1-exp((-1)*TINT/(RON*C1)));
                }
                AVNEW = AV;
                ITER = 0;
                do{
                    temp = (-(weigth)*(VC1ANT-VC1) + VC2*(1+(1+CP/C2)/AV_OLD))/(1+(1+weigth+CP/C2)/AVNEW);
                    temp = temp/(1+(CNL1*temp/2)+(CNL2*temp*temp/3));
                    if(temp>OPSPOS){
                        temp=OPSPOS;
                    }
                    if(temp<OPSNEG){
                        temp=OPSNEG;
                    }
                    AVANT = AVNEW;
                    VO2 = temp*temp;
                    VO3 = VO2*temp;
                    VO4 = VO3*temp;
                    AVNEW = AV*(1+(AVNL1*temp)+(AVNL2*VO2)+(AVNL3*VO3)+(AVNL4*VO4));
                    AVVAR = fabs((AVANT-AVNEW)/AVNEW);
                    ITER = ITER + 1;
                    }while(AVVAR>0.01&&ITER<50);
                y[0] = temp;
                y2[0]=-VC1;
                cont=0;
             }
            }
        }

        else /*  Input switch clocked at phi2: PHI=2     */
            {.............. }

        work[0]=cont; work[1]=VC1; work[2]=VC2; work[3]=VC1ANT; work[4]=VAN; work[5]=VAN1MED; work[6]=VAINII; work[7]=VAINIS;
    }

    static void mdlTerminate(SimStruct *S)
    {...}

    #ifdef  MATLAB_MEX_FILE
    #include "simulink.c"
    #else
    #include "cg_sfun.h"
    #endif
```

Figure 3.23 Excerpt of MATLAB C-coded S-function file of an SC FE integrator considering all circuit errors: calculation of incomplete settling during integration phase, capacitor nonlinearity, nonlinear finite opamp DC gain, and output swing **(Part 3 of 3)**.

Note that the resulting expression can be viewed as the charge stored in a nominal linear capacitor $C_{s,i_{(nominal)}}$ with a voltage across it given by $(v + \mathrm{CNL}_1/2 \cdot v^2 + \mathrm{CNL}_2/3 \cdot v^3)$.

Input-Referred Thermal Noise

The circuit noise model takes into account the thermal noise generated in the switches and the opamp [31]. Flicker (or $1/f$) noise is not considered in the system-level behavioral model because it is assumed that it will be canceled out by a proper electrical design at transistor level. Other noise sources contributions, such as thermal and flicker noise sources associated with the $\Sigma\Delta$M reference voltages, are usually considered during the electrical (transistor-level) design. This way, based on the analysis described in Section 2.5, the rms value of the input-referred thermal noise included in the model (`VNOISE` in Figure 3.22) is approximated by

$$v_{\mathrm{th}} = \mathrm{UNOISE}\sqrt{12 f_s (2S_{\mathrm{in}\phi_1} + 2S_{\mathrm{in}\phi_2} + S_{\mathrm{op}})} \tag{3.14}$$

where UNOISE is a random number in the range of $(-0.5, +0.5)$ generated by the `Uniform Random Number` building block available in SIMULINK, S_{op} represents the PSD of the thermal noise generated by the opamp, and $S_{\mathrm{in}\phi_{1,2}}$ stands for the PSD of the thermal noise generated in the switches during clock phases $\phi_{1,2}$. These PSD functions, respectively, denoted as `SIN_OP, SIN_PHI1, SIN_PHI2` in the C-code of Figure 3.22 are calculated as a function of different electrical parameters involved in the thermal noise model analyzed in Section 2.5. These parameters include, among others, the switch on-resistance R_{on} (denoted as `RON` in the C-code of Fig. 3.22), the equivalent noise bandwidth of noise sources associated with the switches during clock phase ϕ_2 (denoted as `BWn_SPHI2` in Figure 3.22), and the equivalent bandwidth of the amplifier thermal noise (`BWn_OP` in Figure 3.22). Note that if ideal switches are assumed—that is, $R_{\mathrm{on}} = 0$—only the amplifier contributes to the thermal noise model.

Switch On-Resistance Dynamics

The effect of finite switch on-resistance R_{on} is considered in both clock phases. For instance, as shown in the C-code of Figures 3.21–3.23, the value of the voltage sampled on C_s taking into account the effect of capacitor nonlinearity, thermal noise, and the linear sampling process due to R_{on} is given by

$$v_{C_s} = -\left[(v_i + \frac{\mathrm{CNL}_1}{2} \cdot v_i^2 + \frac{\mathrm{CNL}_2}{3} \cdot v_i^3) + v_{\mathrm{th}}\right]\left(1 - e^{\frac{-T_S}{R_{\mathrm{on}} C_s}}\right) \tag{3.15}$$

where v_i is the value of the input signal at the end of the sampling phase—denoted as `U1` in the C-code.

As stated in Section 2.7.2, switches are usually implemented as CMOS transmission gates and the value of R_{on} strongly depends on the value of the voltage drop across the nodes of the switch. This nonlinear phenomenon causes harmonic distortion which, as will be shown in Chapter 4, increases with the ratio between the input frequency and the sampling frequency [37], thus being especially critical in broadband applications.

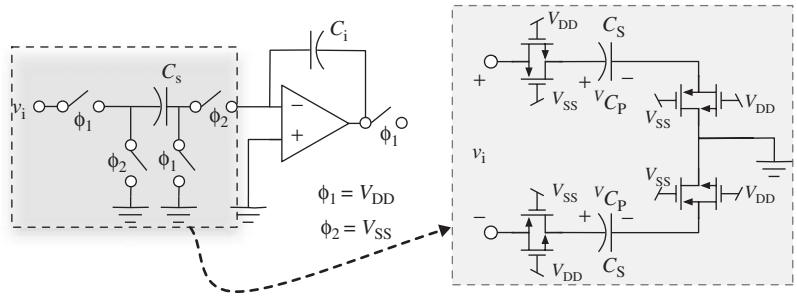


Figure 3.24 Front-end part of a fully-differential SC integrator during the sampling phase.

In order to model the effect of nonlinear R_{on} in CMOS switches, let us consider again the circuit in Figure 2.21. This circuit—depicted in Figure 3.24 for the sake of clarity—models the sampling operation of a fully-differential SC FE integrator during the sampling phase. Note that only the input CMOS switches connected to C_s on ϕ_1 (sampling phase in Figure 3.24) need to be considered. The behavioral model assumes that the MOS transistors implementing the switches operate in the linear (ohmic) region and that their drain currents $I_{D(N,P)}$ can be expressed as

$$\begin{aligned} I_{DN} &= \beta_N \left[V_{DS}(V_{GS} - V_{TN}) - \frac{V_{DS}^2}{2} \right] \\ I_{DP} &= \beta_P \left[V_{SD}(V_{SG} - |V_{TP}|) - \frac{V_{SD}^2}{2} \right] \end{aligned} \tag{3.16}$$

where $V_{T(N,P)}$ denotes the threshold voltage of the nMOS and pMOS transistors, V_{DS} and V_{GS} stand for the drain-to-source and gate-to-source voltages of the nMOS transistor, and V_{SD} and V_{SG} are the source-to-drain and source-to-gate voltages of the pMOS transistor.

From Equation 3.16, it can be shown that the currents flowing through the sampling capacitor in the positive and negative branches of Figure 3.24 can be, respectively, written as

$$\begin{aligned} I_{Cp} &= C_s \frac{dv_{Cp}}{dt} = g_{on}\left(+\frac{v_i}{2} - v_{Cp}\right) + \Delta\beta\left(\frac{v_{Cp}^2}{2} - \frac{v_i^2}{8}\right) \\ I_{Cn} &= C_s \frac{dv_{Cn}}{dt} = g_{on}\left(-\frac{v_i}{2} - v_{Cn}\right) + \Delta\beta\left(\frac{v_{Cn}^2}{2} - \frac{v_i^2}{8}\right) \end{aligned} \tag{3.17}$$

where $\Delta\beta = (\beta_N - \beta_P)$, $g_{on} \equiv 1/R_{on} = [\beta_N(V_{DD} - V_{TN}) + \beta_P(-V_{SS} - |V_{TP}|)]$ is the operating-point switch on-conductance—see Equation 2.47—and $v_{C_s} = v_{Cp} - v_{Cn}$ represents the differential voltage sampled on C_s.

In order to incorporate the effect of nonlinear sampling caused by CMOS switches in a DT behavioral model, the former nonlinear differential equations can be *discretized* and solved numerically. For instance, if the sampling phase is divided into a number

$(j = 1, \ldots, N_s)$ of time intervals, δt and a finite-difference approximation is used for integrating Equation 3.17, the sampled voltages of Figure 3.24 can be expressed as

$$v_{Cp}(t) \simeq \frac{C_s + g_{on}\delta t - \sqrt{(C_s + g_{on}\delta t)^2 + \beta\delta t\{\delta t v_i(t)[\beta\frac{v_i(t)}{4} - g_{on}] - 2C_s v_{Cp}(t-\delta t)\}}}{\beta\delta t}$$

$$v_{Cn}(t) \simeq \frac{C_s + g_{on}\delta t - \sqrt{(C_s + g_{on}\delta t)^2 + \beta\delta t\{\delta t v_i(t)[\beta\frac{v_i(t)}{4} + g_{on}] - 2C_s v_{Cn}(t-\delta t)\}}}{\beta\delta t} \tag{3.18}$$

where $t = j\delta t$, $\delta t = T_s/(2N_s)$ and $v_{C_s}(t-\delta t) \equiv v_{Cp}(t-\delta t) - v_{Cn}(t-\delta t)$ is the differential sampled voltage on C_s at time instant $t - \delta t$.

Figure 3.25 shows a C-code that can be used in a C-MEX S-function to model the effect of nonlinear R_{on} based on the solution given in Equation 3.18, where `VC1(p,n)` and

```
switch (model)
                {
                case 1:     /*Finite and Lineal Ron*/
                   {
                       VC1=(VC1OLD+(*uPtrs1[0]-VC1OLD)*(1-exp(-1*TS/(2*RON*C1))));
                       break;
                   }
                case 2:     /*Non-linear Ron (analytic model)*/
                   {
                       g= p[0];
                       B= p[1];
                       for(i=0;i<k/2;i++) /* Division of the sampling period in "k" dt time intervals */
                       {
                           if (signal==1){ /* Sinewave input */
                               Vi=A*sin(2*pi*fi*(i*dt+n*TS/2)+phase);}
                           else{
                               Vi=sig[k/2*n+i]; /*Generic waveform input (g=gON, B=Beta */
                           }
                           VC1p=(C1+dt*g-sqrt(pow((C1+g*dt),2)+B*dt*(dt*Vi*(B*Vi/4-g)-2*C1*VC1OLDp)))/(B*dt);
                           VC1n=(C1+dt*g-sqrt(pow((C1+g*dt),2)+B*dt*(dt*Vi*(B*Vi/4+g)-2*C1*VC1OLDn)))/(B*dt);
                           VC1OLDp=VC1p;
                           VC1OLDn=VC1n;
                       }
                       VC1=VC1p-VC1n;
                       break;
                   }
                case 3:     /*Non-linear Ron (lookup table model)*/
                   {
                   for(i=0;i<k/2;i++)
                       {
                       if (signal==1){
                           Vi=A*sin(2*pi*fi*(i*dt+n*TS/2)+phase);}
                       else{
                           Vi=sig[k/2*n+i];
                       }
                       Ronp=0;
                       Ronn=0;
                       for(j=0;j<m;j++)
                       {
                           Ronp = Ronp + (p[j])*pow(Vi/2,m-j-1);
                           Ronn = Ronn + (p[j])*pow(-Vi/2,m-j-1);
                       }
                       VC1p= (VC1OLDp+(Vi/2-VC1OLDp)*(1-exp(-1*dt/(Ronp*C1))));
                       VC1n= (VC1OLDn+(-Vi/2-VC1OLDn)*(1-exp(-1*dt/(Ronn*C1))));
                       VC1OLDp=VC1p;
                       VC1OLDn=VC1n;
                       }
                   VC1= VC1p-VC1n;
                   break;
                   }
}
```

Linear Ron

Nonlinear Ron (analytic model)

Nonlinear Ron (table look-up approach)

Figure 3.25 Excerpt of the C-coded S-function file including the model of finite (nonlinear) switch on-resistance.

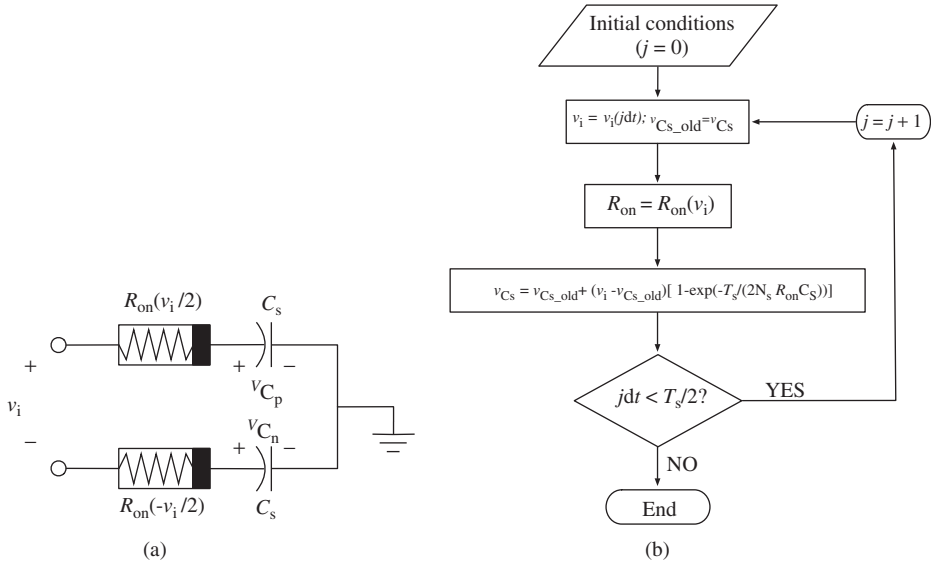


Figure 3.26 Modeling the nonlinear switch on-resistance as a polynomial function of the input signal: (a) equivalent circuit and (b) flowchart.

`VC1OLD(p,n)` denote, respectively, $v_{C(p,n)}(t)$ and $v_{C(p,n)}(t - \delta t)$. Note that two different cases are considered for the input signal waveform, namely, a sinusoidal waveform and a generic waveform. In both cases, the input waveform data can be provided by the user through the SIMULINK S-function. For the sake of completeness, the code in Figure 3.25 includes also a linear version sampling process due to finite R_{on}, in case nonlinearities are not considered.

The C-code in Figure 3.25 shows also an alternative approach to model the effect of the nonlinear switch on-resistance that is based on the equivalent circuit shown in Figure 3.26a. In this model, the effect of switches connected to the common voltage node are neglected (see Section 2.7.2) and R_{on} is modeled as a polynomial function of the input signal, given by [5]

$$R_{on}(v_i) \simeq \sum_{j=1}^{N_s} p_j \cdot v_i^j(j\delta t) \tag{3.19}$$

Taking into account the above-mentioned nonlinear characteristic for R_{on}, the voltages sampled at the positive and negative branches of Figure 3.26a can be, respectively, written as

$$\begin{aligned} v_{Cp}(t) &\simeq v_{Cp}(t - \delta t) + \left[+\frac{v_i(t - \delta t)}{2} - v_{Cp}(t - \delta t) \right] \left(1 - e^{\frac{-T_S}{R_{on}[+v_i(t)]C_S}} \right) \\ v_{Cn}(t) &\simeq v_{Cn}(t - \delta t) + \left[-\frac{v_i(t - \delta t)}{2} - v_{Cn}(t - \delta t) \right] \left(1 - e^{\frac{-T_S}{R_{on}[-v_i(t)]C_S}} \right) \end{aligned} \tag{3.20}$$

which can be computed following the flowchart depicted in Figure 3.26b. As shown in Figure 3.25, the corresponding C-code has been incorporated in the S-function. Note that the coefficients of the polynomial function in Equation 3.19 can be used either in a synthesis process to evaluate the maximum nonlinearity tolerated for a given set of specifications or to verify a given design using a table look-up approach. In the latter case, the polynomial function coefficients p_j can be obtained from a curve-fitting process applied to the nonlinear R_{on} characteristic obtained in an electrical (transistor-level) simulation of the CMOS switch. The curve-fitting process can be easily performed using the `polyfit` routine provided by MATLAB [28].

Incomplete Settling Error

The behavioral model of the transient response for SC FE Integrators included in Figures 3.21–3.23 is based on the analysis presented in [38] which was discussed in Section 2.4. The model includes the effect of the opamp dynamic performance limitations, such as GB and SR, on the charge transfer during both sampling and integration phases. Parasitic capacitors associated to both opamp and CMOS switches, as well as the capacitor load at the integrator output—which changes from sampling to integration phase—are also taken into account. To this end, the equivalent circuit scheme shown in Figure 2.6 is solved in the behavioral model. As shown in Figure 2.6, the model assumes a number of N_i input SC branches—made up of its corresponding sampling capacitor and switches—and that another SC integrator with N_o input SC branches is connected (as a load) to the output node of the integrator. The model used for the amplifier, shown in Figure 2.7 includes a single-pole dynamic and a nonlinear characteristic with maximum output current I_o (`I0` in the C-code).[13]

As stated in Section 2.4, the evaluation of the incomplete settling error model begins with the computation of the equivalent capacitive load at the opamp output node during both the integration and sampling phases, given by Equation 2.16. This equation is included in the model of Figure 3.22, where the sampling capacitance of the ith input SC branch of the load integrator is denoted as `uPtrs4[N+i]`, the parasitic capacitor associated with the summation node of the input SC branches is denoted as `CP`, and the load capacitance is `CLOAD`. Note that the model in Figure 3.22 has the possibility of including a different value of C_L during the integration and the sampling phase. To this purpose, two different model parameters are used, respectively, named `CLOAD_I`, `CLOAD_S`. This way, the load capacitance during integration and sampling phases are, respectively, given by `CLOAD+CLOAD_I` and `CLOAD+CLOAD_S`.

After computing the equivalent load capacitances, the settling model is evaluated during both clock phases considering the different possibilities of the opamp operation, that is, linearly or in slew. This way, at the end of the sampling phase of period nT_s the voltage

[13] Note that the model described in the S-function of Figures 3.21–3.23 corresponds to an SC FE integrator with one input SC branch. However, the model presented here can be extended to SC integrators with N_i input SC branches.

at the input node of the opamp (represented by VAN in Figure 3.22) is computed as [38]

$$v_a(nT_s) = \begin{cases} v_{ai,s} e^{-\frac{g_{msef} T_s}{2C_{eq,s}}} & |v_{ai,s}| \le \frac{I_o}{g_{msef}} \\ \frac{I_o}{g_{msef}} \mathrm{sgn}(v_{ai,s}) e^{-\frac{g_{msef}(T_s/2 - t_{o,s})}{C_{eq,s}}} & |v_{ai,s}| > \frac{I_o}{g_{msef}} \text{ and } t_{o,s} < \frac{T_s}{2} \\ v_{ai,s} - \frac{I_o}{C_{eq,s}} \mathrm{sgn}(v_{ai,s}) \frac{T_s}{2} & |v_{ai,s}| > \frac{I_o}{g_{msef}} \text{ and } t_{o,s} \ge \frac{T_s}{2} \end{cases} \tag{3.21}$$

where

$$t_{o,s} = \frac{C_{eq,s}}{I_o} |v_{ai,s}| - \frac{C_{eq,s}}{g_{msef}} \tag{3.22}$$

and $\mathrm{sgn}(v_{ai,s})$ is the sign function of $v_{ai,s}$ (denoted as VAINIS in Figure 3.22), which represents the value of v_a at the beginning of the sampling phase given by

$$v_{ai,s} = v_a[(n-1/2)T_s] - \sum_{i=1}^{N_o} \frac{C_{Sni}}{C_{eq,s}} \{v_o[(n-1/2)T_s] - v_{C_{Sni}}[(n-1/2)T_s]\} \tag{3.23}$$

where $v_{C_{Sni}}$ is the voltage across capacitor C_{Sni}, respectively, represented by the parameters uPtrs4[N+i] and uPtrs4[i] in Figure 3.22.

Note that Equations 3.21 and 3.22 are essentially the same as Equations 2.17 and 2.18, but replacing the OTA transconductance g_m by a parameter denoted as g_{msef} (GMSEFF in Figure 3.22). This parameter represents the effective transconductance of the opamp that is used for modeling the GB degradation because of the switch on-resistance during sampling phase, as described in Section 2.4.4 [31].

Therefore, once $v_a(nT_s)$ has been calculated, the voltage at the output node of the opamp is computed as

$$\begin{aligned} v_o(nT_s) = {} & v_o[(n-1/2)T_s] \\ & + \left\{1 + \frac{1}{A_v[v_o(nT_s)]}\right\} \left(1 + \frac{C_P}{C_I}\right) \{v_a(nT_s) - v_a[(n-1/2)T_s]\} \end{aligned} \tag{3.24}$$

in which the effect of the nonlinear finite OTA DC gain is also accounted for. The former equation is solved by following the iterative procedure described in Section 3.3. Note from Figure 3.22 that the nonlinearity of the capacitor—not shown in Equation 3.24 for the sake of simplicity—is also taken into account in the behavioral model to get a more accurate value of $v_o(nT_s)$.

During the integration phase the incomplete settling model is evaluated proceeding in a similar way as during the sampling phase (Section 2.4). The value of v_a at the end of the integration phase of period $(n+1/2)T_s$ is thus given by [38]

$$v_a[(n+1/2)T_s] = \begin{cases} v_{ai,i} e^{-\frac{g_{mief} T_s}{2C_{eq,i}}} & |v_{ai,i}| \le \frac{I_o}{g_{mief}} \\ \frac{I_o}{g_{mief}} \mathrm{sgn}(v_{ai,i}) e^{-\frac{g_{mief}(T_s - t_{o,i})}{C_{eq,i}}} & |v_{ai,i}| > \frac{I_o}{g_{mief}} \text{ and } t_{o,i} < T_s \\ v_{ai,i} - \frac{I_o}{C_{eq,i}} \mathrm{sgn}(v_{ai,i}) \frac{T_s}{2} & |v_{ai,i}| > \frac{I_o}{g_{mief}} \text{and } t_{o,i} \ge T_s \end{cases} \tag{3.25}$$

where g_{mief} (GMIEFF in Figure 3.23) stands for the effective OTA transconductance during the integration phase and

$$\begin{aligned} t_{\text{o,i}} &= \frac{T_{\text{s}}}{2} + \frac{C_{\text{eq,i}}}{I_{\text{o}}}|v_{\text{ai,i}}| - \frac{C_{\text{eq,i}}}{g_{\text{mief}}} \\ v_{\text{ai,i}} &= \frac{1}{C_{\text{eq,i}}}\left(1 + \frac{C_{\text{L}}}{C_{\text{I}}}\right)\sum_{i=1}^{N_{\text{i}}} C_{\text{S}_{\text{i}}}\{v_{\text{i2}}(nT_{\text{s}}) - v_{\text{i1}}[(n-1/2)T_{\text{s}}]\} + \frac{C^{'}}{C_{\text{eq,i}}}v_{\text{a}}(nT_{\text{s}}) \end{aligned} \tag{3.26}$$

with v_{ij} being the voltage at the jth input node of the ith input SC branch, $C_{S_{\text{i}}}$ is the sampling capacitor of the ith input SC branch, and $C^{'}$ (CPRIMA in the C-code) is given by

$$C^{'} = C_{\text{P}} + C_{\text{L}}\left(1 + \frac{C_{\text{P}}}{C_{\text{I}}}\right) \tag{3.27}$$

After computing $v_{\text{a}}[(n+1/2)T_{\text{s}}]$, the value of the opamp output node is solved in an iterative way (Figure 3.23) taking into account the effect of nonlinear OTA DC gain and capacitor nonlinearity. This iterative procedure converges typically in a few iterations and provides very accurate results in close agreement with transistor-level simulations [5].

Before concluding this section, it is very important to pay attention to a practical issue related to the implementation of the incomplete settling error model in a programming platform in general, and in the MATLAB/SIMULINK environment in particular. It can be noted that in order to implement this error, it is not sufficient to provide information about the building block itself (i.e., the SC integrator to be modeled), but also about those building blocks connected to its output. For instance, the values of $v_{C_{\text{Sni}}}$ at $(n-1/2)T_{\text{s}}$ are needed to compute v_{a} at nT_{s}—see Equation 3.23. Therefore, an accurate behavioral model must incorporate the required data computed in previous clock cycles that correspond to those building blocks connected at the output of the integrator. This can be implemented in the SIMULINK environment by using the `From` and `Goto` blocks provided by the `Signal Routing` SIMULINK library [29].

Figure 3.27 illustrates the use of `From` and `Goto` blocks for the behavioral model of SC FE integrators in SIMULINK. In this example, two SC FE integrators are connected to build a second-order ΣΔM. In order to distinguish both building blocks in their corresponding behavioral models, two different identifiers are used, named int3 and int4 in this case. As shown in the corresponding S-function diagram, `From` and `Goto` blocks allow passing of the required information between both SC integrators without actually connecting them, what notably simplifies the implementation of arbitrary ΣΔMs without modifying the model code.

In the example of Figure 3.27, the information provided by the back-end integrator (identified as int4) is included in the model of the front-end integrator (int3) by connecting the `From` block to an input port of the front-end integrator. Similarly, the front-end integrator uses the `Goto` block to provide its stored data to another building block. The information stored essentially consists of a data array that contains the values of the sampling capacitors and the sampled voltages of the input SC branches of a given integrator. This data is read by the model by using dynamically sized inputs, for instance `uPtrs4` array in the C-code of Figure 3.21.

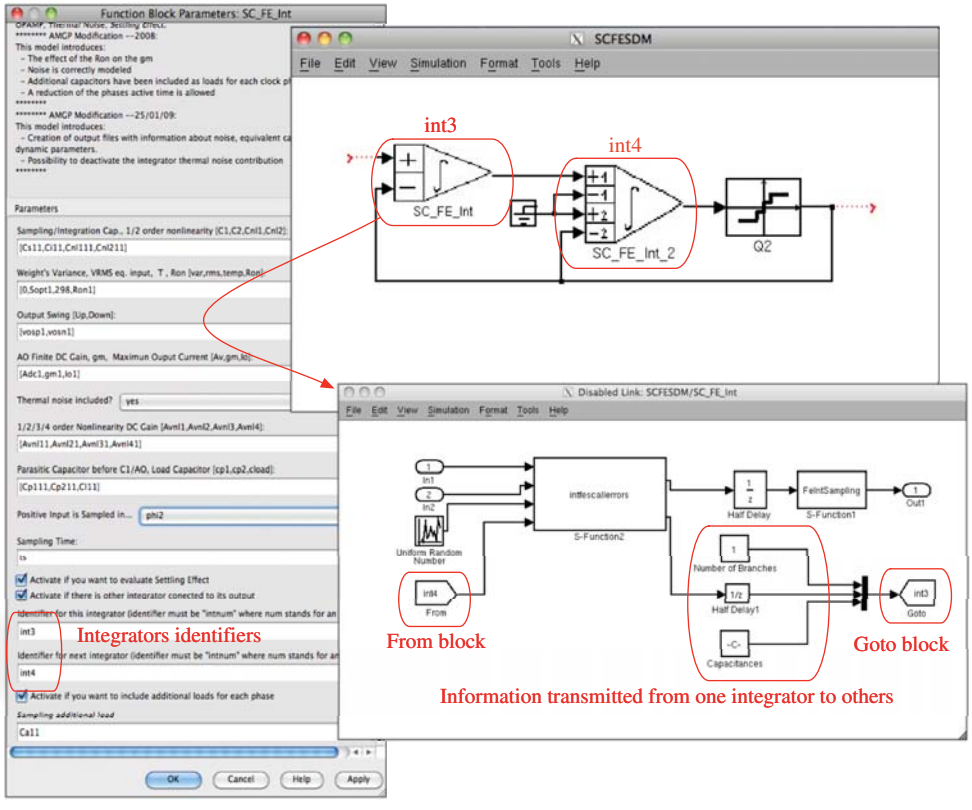


Figure 3.27 Illustrating the use of From and Goto SIMULINK blocks to transmit information between the models of two SC FE integrators in a second-order ΣΔM.

3.4.2 Modeling of CT Integrators using S-Functions

CT integrators constitute the most critical building blocks of CT-ΣΔMs. As discussed in Chapter 2, these blocks can be implemented using different circuit topologies, namely active-RC, Gm-C, MOSFET-RC, Gm-MC, etc. All of them may have the same ideal characteristics, although involving a number of design trade-offs and error limitations that can be modeled using C-MEX S-functions. This section focuses on the behavioral modeling of S-functions for Gm-C circuits, considering their most important limiting factors. The behavioral model of other CT integrator topologies is considered in Appendices A and B.

Let us consider the Gm-C integrator conceptually depicted in Figure 3.5b, whose ideal behavior is governed by Equation 3.2. In practice, this ideal behavior is degraded by the action of a number of circuit errors, namely, input-referred thermal noise, circuit–element tolerances and mismatch, input and output voltage saturation, transconductance nonlinearity, finite OTA DC gain, transient response (including either a single-pole or a two-pole model for the OTA), etc. [39]. There are several ways of modeling these effects using S-functions, as discussed below.

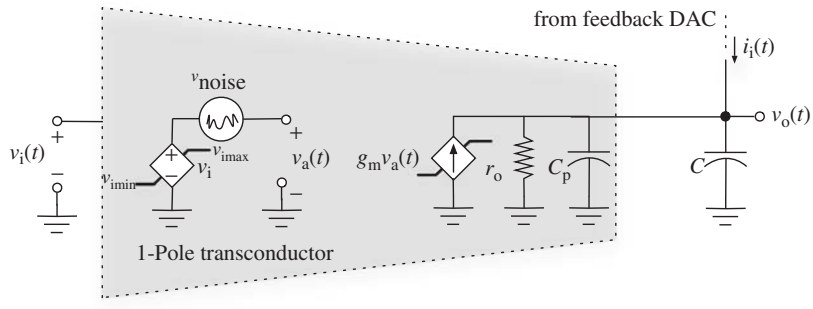


Figure 3.28 Equivalent circuit of a Gm-C integrator considering a single-pole model.

Single-Pole Gm-C Model

Figure 3.28 shows the equivalent circuit of a Gm-C integrator considering a single-pole dynamic. The model includes also the following nonideal effects: input-referred thermal noise, input/output voltage saturation, time-constant error, finite DC gain (modeled as a finite output conductance), and a nonlinear transconductance that depends on the input voltage v_i as

$$g_m \simeq g_{mo} \cdot (1 + g_{m1} \cdot v_i + g_{m2} \cdot v_i^2) \tag{3.28}$$

where g_{mo} is the nominal value of the transconductance and $g_{m(1,2)}$ are the nonlinear transconductance coefficients.

The circuit errors of Figure 3.28 are computed by following the iterative procedure depicted in the flowchart of Figure 3.29. This computational model can be included in a C-MEX S-function as shown in Figure 3.30 and implemented in an S-function block as illustrated in Figure 3.31. Note that the C-MEX S-function file is made up of the same parts as those S-functions described in previous sections, beginning with a model parameter and state variable initialization and computing the different errors according to the flow of Figure 3.29. The main difference is related to those model aspects related to the timing. Thus, the CT nature of the circuit is specified in the `mdInitializeSampleTimes` structure, where the sample time is defined as `CONTINUOUS_SAMPLE_TIME` to point out that this S-function corresponds to a CT building block [34].

The output voltage is computed using the `mdlDerivatives` routine [34] that solves the following differential equation:

$$\frac{dv_o(t)}{dt} = \frac{1}{C + C_p} \cdot [g_m v_i(t) - g_o v_o(t) + i_i(t)] \tag{3.29}$$

where g_o is the finite output conductance (denoted as `go` in Figure 3.30), C_p is the parasitic capacitance that models the integration time-constant error (`Cp` in Figure 3.30), and $i_i(t)$ is the current-mode input that can be used for modeling the current provided by a feedback current-steering DAC in a CT-ΣΔM (Figure 3.31).

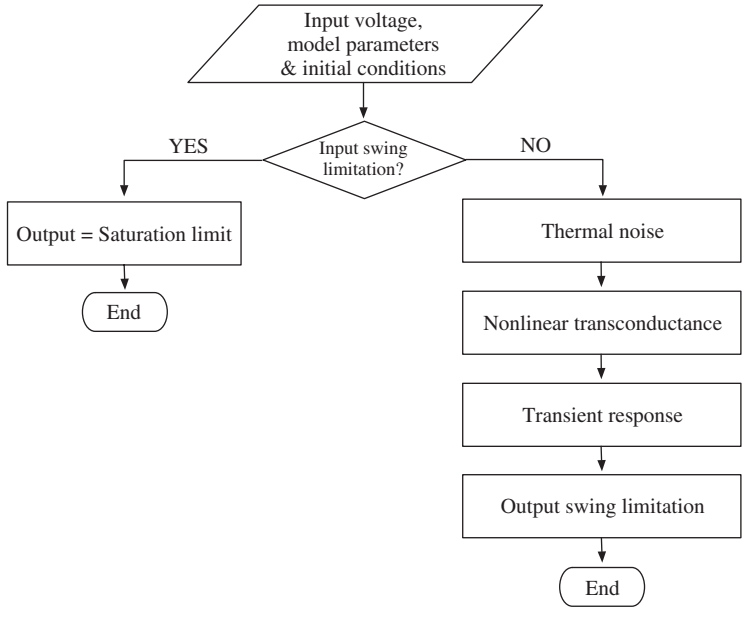


Figure 3.29 Flowchart of the Gm-C integrator computational model.

Two-Pole Dynamics Model

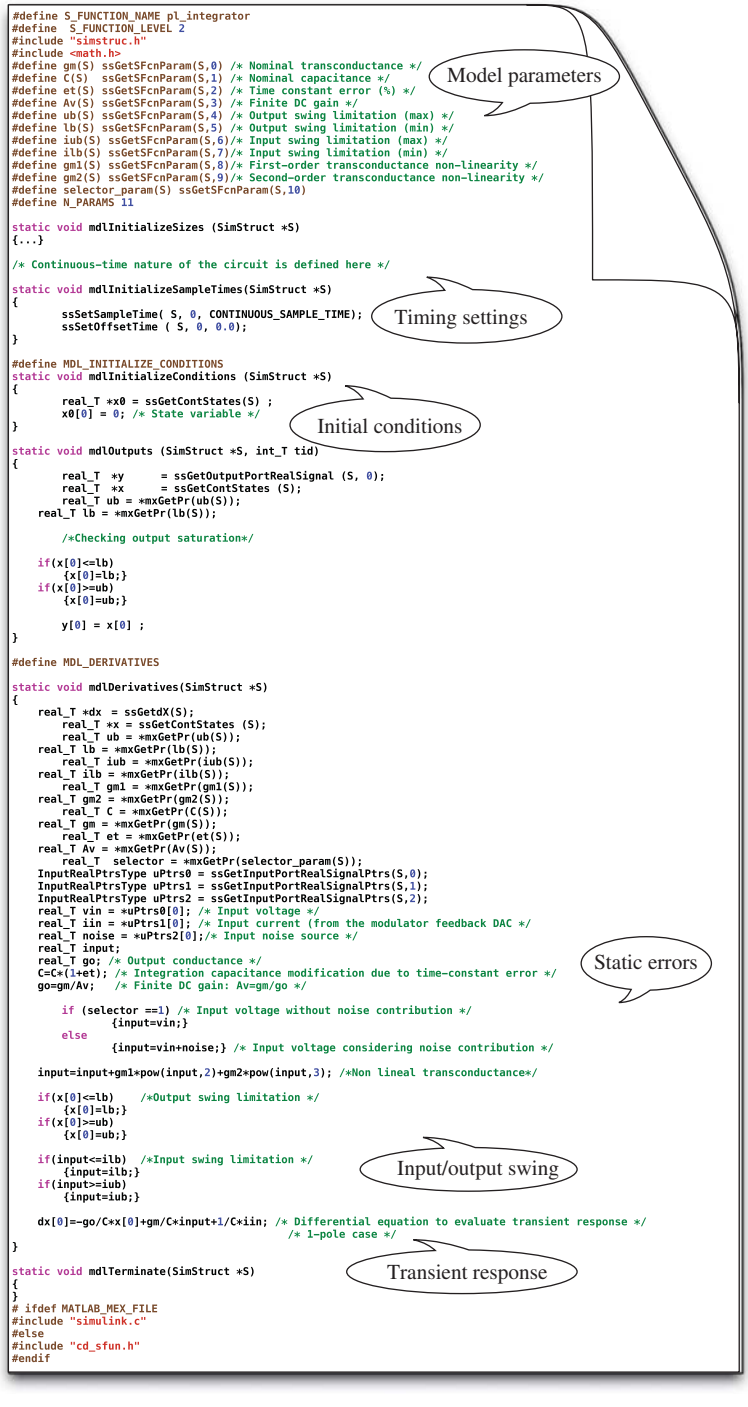
A two-pole transient response can be included in the model of a Gm-C integrator using the equivalent circuit shown in Figure 3.32. This model can be incorporated in an S-function as illustrated in Figure 3.33, which shows only the excerpt of the C-MEX file corresponding to the `mdlDerivatives` routine. In this case, the output voltage is computed by solving the following set of differential equations:

$$\begin{aligned}\frac{dv_a(t)}{dt} &= \frac{1}{C_a}\cdot\left[\frac{v_a(t)}{r_a} - g_{ma}v_i(t)\right]\\ \frac{dv_o(t)}{dt} &= \frac{1}{C+C_p}\cdot\left[g_m v_a(t) - g_o v_o(t) + i_i(t)\right]\end{aligned} \tag{3.30}$$

where v_a and v_o are represented in Figure 3.33 by two state variables named `x[0]` and `x[1]`, respectively.

Modeling Transconductors as S-Functions

The model of a Gm-C integrator can be also implemented as the cascade of two building blocks, as illustrated in Figure 3.34a: a transconductor and an output impedance circuit made up of the parallel connection of an output resistance ($r_o \equiv 1/g_o$), the integration capacitor C, and its parasitic capacitance C_p. Figure 3.34b shows the implementation of Figure 3.34a in SIMULINK. The transconductor S-function block includes the




```
#define S_FUNCTION_NAME pl_integrator
#define  S_FUNCTION_LEVEL 2
#include "simstruc.h"
#include <math.h>
#define gm(S) ssGetSFcnParam(S,0) /* Nominal transconductance */
#define C(S)  ssGetSFcnParam(S,1) /* Nominal capacitance */
#define et(S) ssGetSFcnParam(S,2) /* Time constant error (%) */
#define Av(S) ssGetSFcnParam(S,3) /* Finite DC gain */
#define ub(S) ssGetSFcnParam(S,4) /* Output swing limitation (max) */
#define lb(S) ssGetSFcnParam(S,5) /* Output swing limitation (min) */
#define iub(S) ssGetSFcnParam(S,6)/* Input swing limitation (max) */
#define ilb(S) ssGetSFcnParam(S,7)/* Input swing limitation (min) */
#define gm1(S) ssGetSFcnParam(S,8)/* First-order transconductance non-linearity */
#define gm2(S) ssGetSFcnParam(S,9)/* Second-order transconductance non-linearity */
#define selector_param(S) ssGetSFcnParam(S,10)
#define N_PARAMS 11

static void mdlInitializeSizes (SimStruct *S)
{...}

/* Continuous-time nature of the circuit is defined here */

static void mdlInitializeSampleTimes(SimStruct *S)
{
        ssSetSampleTime( S, 0, CONTINUOUS_SAMPLE_TIME);
        ssSetOffsetTime ( S, 0, 0.0);
}

#define MDL_INITIALIZE_CONDITIONS
static void mdlInitializeConditions (SimStruct *S)
{
        real_T *x0 = ssGetContStates(S) ;
        x0[0] = 0; /* State variable */
}

static void mdlOutputs (SimStruct *S, int_T tid)
{
        real_T  *y      = ssGetOutputPortRealSignal (S, 0);
        real_T  *x      = ssGetContStates (S);
        real_T ub = *mxGetPr(ub(S));
    real_T lb = *mxGetPr(lb(S));

        /*Checking output saturation*/

    if(x[0]<=lb)
        {x[0]=lb;}
    if(x[0]>=ub)
        {x[0]=ub;}

        y[0] = x[0] ;
}

#define MDL_DERIVATIVES

static void mdlDerivatives(SimStruct *S)
{
    real_T *dx  = ssGetdX(S);
        real_T *x = ssGetContStates (S);
        real_T ub = *mxGetPr(ub(S));
    real_T lb = *mxGetPr(lb(S));
        real_T iub = *mxGetPr(iub(S));
    real_T ilb = *mxGetPr(ilb(S));
        real_T gm1 = *mxGetPr(gm1(S));
    real_T gm2 = *mxGetPr(gm2(S));
        real_T C = *mxGetPr(C(S));
    real_T gm = *mxGetPr(gm(S));
        real_T et = *mxGetPr(et(S));
    real_T Av = *mxGetPr(Av(S));
        real_T  selector = *mxGetPr(selector_param(S));
    InputRealPtrsType uPtrs0 = ssGetInputPortRealSignalPtrs(S,0);
    InputRealPtrsType uPtrs1 = ssGetInputPortRealSignalPtrs(S,1);
    InputRealPtrsType uPtrs2 = ssGetInputPortRealSignalPtrs(S,2);
    real_T vin = *uPtrs0[0]; /* Input voltage */
    real_T iin = *uPtrs1[0]; /* Input current (from the modulator feedback DAC */
    real_T noise = *uPtrs2[0];/* Input noise source */
    real_T input;
    real_T go; /* Output conductance */
    C=C*(1+et); /* Integration capacitance modification due to time-constant error */
    go=gm/Av;   /* Finite DC gain: Av=gm/go */

        if (selector ==1) /* Input voltage without noise contribution */
                {input=vin;}
        else
                {input=vin+noise;} /* Input voltage considering noise contribution */

    input=input+gm1*pow(input,2)+gm2*pow(input,3); /*Non lineal transconductance*/

    if(x[0]<=lb)    /*Output swing limitation */
        {x[0]=lb;}
    if(x[0]>=ub)
        {x[0]=ub;}

    if(input<=ilb)  /*Input swing limitation */
        {input=ilb;}
    if(input>=iub)
        {input=iub;}

    dx[0]=-go/C*x[0]+gm/C*input+1/C*iin; /* Differential equation to evaluate transient response */
                                         /* 1-pole case */
}

static void mdlTerminate(SimStruct *S)
{
}
# ifdef MATLAB_MEX_FILE
#include "simulink.c"
#else
#include "cd_sfun.h"
#endif
```

Figure 3.30 C-coded S-function file of a Gm-C integrator with a single-pole dynamic.

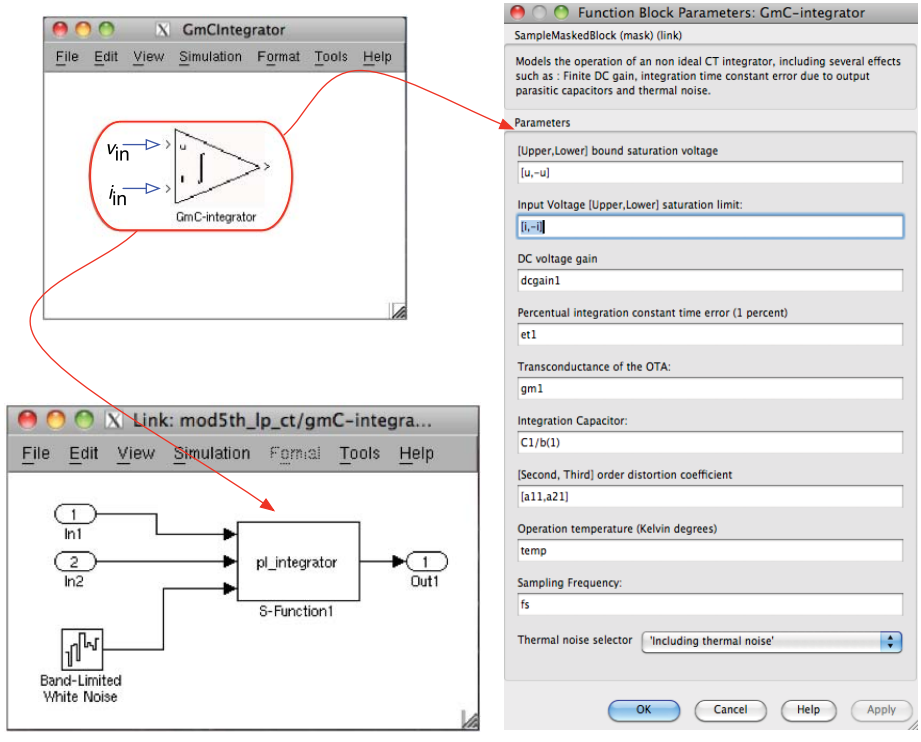


Figure 3.31 Illustrating the incorporation of the Gm-C integrator S-function in SIMULINK.

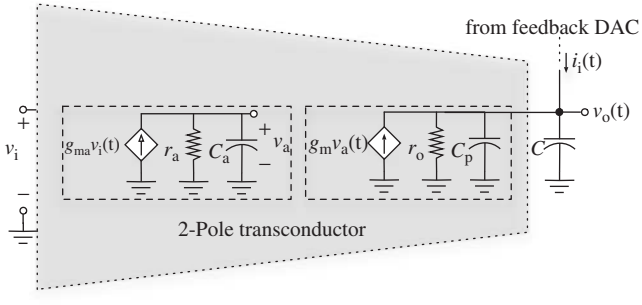


Figure 3.32 Equivalent circuit of a Gm-C integrator considering a two-pole model [5].

input-referred noise, saturation voltage at both the input and the output nodes, and a transconductance that is a nonlinear function of the input voltage given by

$$g_m \simeq g_{mo} \cdot \left(1 - \frac{4 \cdot g_{mo}}{3 \cdot 10^{\frac{IIP3}{10}}} \cdot v_i^2\right) \tag{3.31}$$

where IIP3 is the input-referred third-order intercept point, used as a model parameter.

```
static void mdlOutputs (SimStruct *S, int_T tid)
{
        real_T  *y      = ssGetOutputPortRealSignal (S, 0);
        real_T  *x      = ssGetContStates (S);
        real_T ub = *mxGetPr(ub(S));
    real_T lb = *mxGetPr(lb(S));

    if(x[1]<=lb)
        {x[1]=lb;}
    if(x[1]>=ub)
        {x[1]=ub;}

        y[0] = x[1] ;
}

#define MDL_DERIVATIVES

static void mdlDerivatives(SimStruct *S)
{
    real_T *dx  = ssGetdX(S);
        real_T *x = ssGetContStates (S);
        real_T ub = *mxGetPr(ub(S));
    real_T lb = *mxGetPr(lb(S));
        real_T iub = *mxGetPr(iub(S));
    real_T ilb = *mxGetPr(ilb(S));
        real_T gm1 = *mxGetPr(gm1(S));
    real_T gm2 = *mxGetPr(gm2(S));
    real_T fp = *mxGetPr(fp(S));       /* fp= 1/(ra Ca) */
        real_T C = *mxGetPr(C(S));
    real_T gm = *mxGetPr(gm(S));
        real_T et = *mxGetPr(et(S));
    real_T Av = *mxGetPr(Av(S));
        real_T  selector = *mxGetPr(selector_param(S));
    InputRealPtrsType uPtrs0 = ssGetInputPortRealSignalPtrs(S,0);
    InputRealPtrsType uPtrs1 = ssGetInputPortRealSignalPtrs(S,1);
    InputRealPtrsType uPtrs2 = ssGetInputPortRealSignalPtrs(S,2);
    real_T vin = *uPtrs0[0];
    real_T iin = *uPtrs1[0];
    real_T noise = *uPtrs2[0];
    real_T input;
    real_T go;
    C=C*(1+et);
    go=gm/Av;

        if (selector ==1)
                {input=vin;}
        else
                {input=vin+noise;}

    /*Non lineal transconductance*/
    input=input+gm1*pow(entrada,2)+gm2*pow(entrada,3);

    /*Output Saturation*/
    if(x[1]<=lb)
        {x[1]=lb;}
    if(x[1]>=ub)
        {x[1]=ub;}

    /*Input Saturation*/
    if(input<=ilb)
        {input=ilb;}
    if(input>=iub)
        {input=iub;}

    dx[0]=fp*(input-gm*x[0]);
    dx[1]=-go/C*x[1]+gm/C*x[0]+1/C*iin;
}
```

Differential equations
($x[0] = v_a$; $x[1] = v_o$)

Figure 3.33 Excerpt of the S-function of a Gm-C integrator considering a two-pole model.

3.4.3 Behavioral Modeling of Quantizers using S-Functions

Quantizers are also essential building blocks of ΣΔMs. These blocks are made up of an ADC and a DAC that are embedded in the ΣΔM loop, the former in the feed-forward path and the latter in the feedback path.[14] As it also happens with integrators, there are

[14] As stated in Chapter 1, strictly speaking, a quantizer is an analog block with both input and output being analog quantities, except that the output amplitude is discretized in a number of analog levels. For that reason, a quantizer is made up of the cascade of an ADC and a DAC. However, the ADC embedded in a ΣΔM is usually represented by a quantizer symbol, although it is connected in the feedback loop to a DAC to complete the quantization

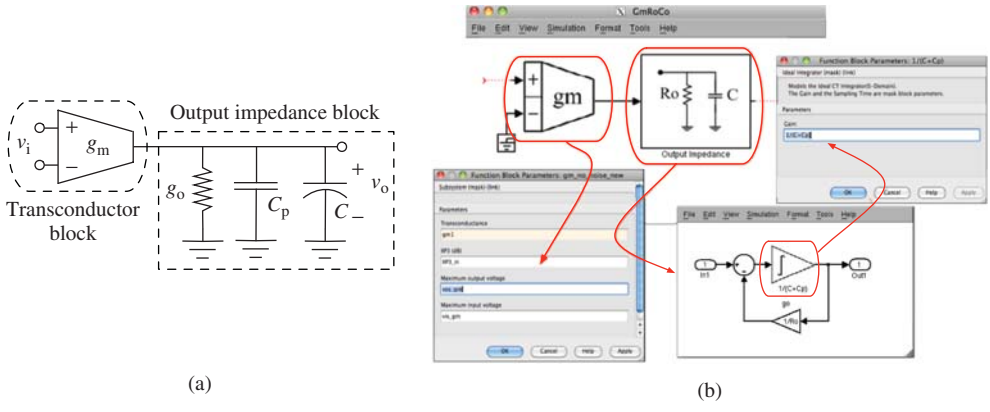


Figure 3.34 Gm-C integrator implemented as a transconductor and an output impedance circuit: (a) conceptual schematic and (b) implementation in SIMULINK.

plenty of different ADC and DAC topologies that are usually embedded in ΣΔMs. Their selection depends on many factors, such as the circuit nature of the loop filter, the number of bits (or levels) of the quantization process, the circuit topology, etc. All of them can be modeled with S-functions, following the same philosophy used for integrators. In this section, two commonly used exemplary blocks are presented as a matter of illustration. Other models of embedded ADCs and DACs can be found in Appendix B.

Modeling Multilevel ADCs as S-Functions

As stated in previous chapters, the quantizers embedded in ΣΔMs are usually implemented by multibit flash ADCs. These circuits—made up of a bank of comparators (or single-bit quantizers) and a resistive ladder—are subject to a number of circuit errors whose effects on the ΣΔM performance are greatly attenuated by the action of the modulator loop filter. However, they must be taken into account in the behavioral models in order to get efficient and accurate designs. Some of these errors are comparator offset and hysteresis, integral nonlinearity (INL), gain error, etc., which can be easily modeled using S-functions following the methodology described in this section.

Figure 3.35a shows the main parts of the S-function block used for modeling a multilevel ADC. This block can be used in either DT- or CT-ΣΔMs for implementing quantizers with an arbitrary number of quantization levels. This way, if an odd number of levels is set up, a *midtreat* quantization characteristic is implemented. Otherwise, a *midrise* quantizer is computed. The model includes the following circuit nonideal effects: INL, gain error, and offset error—all of them expressed in terms of the least significant bit (LSB). Other model parameters are the input and output full-scale (FS) range, the clock phase at which the input is sampled, and the sampling time.

process. This kind of graphic and conceptual representations can be found in many a papers and books, although it is not strictly correct.

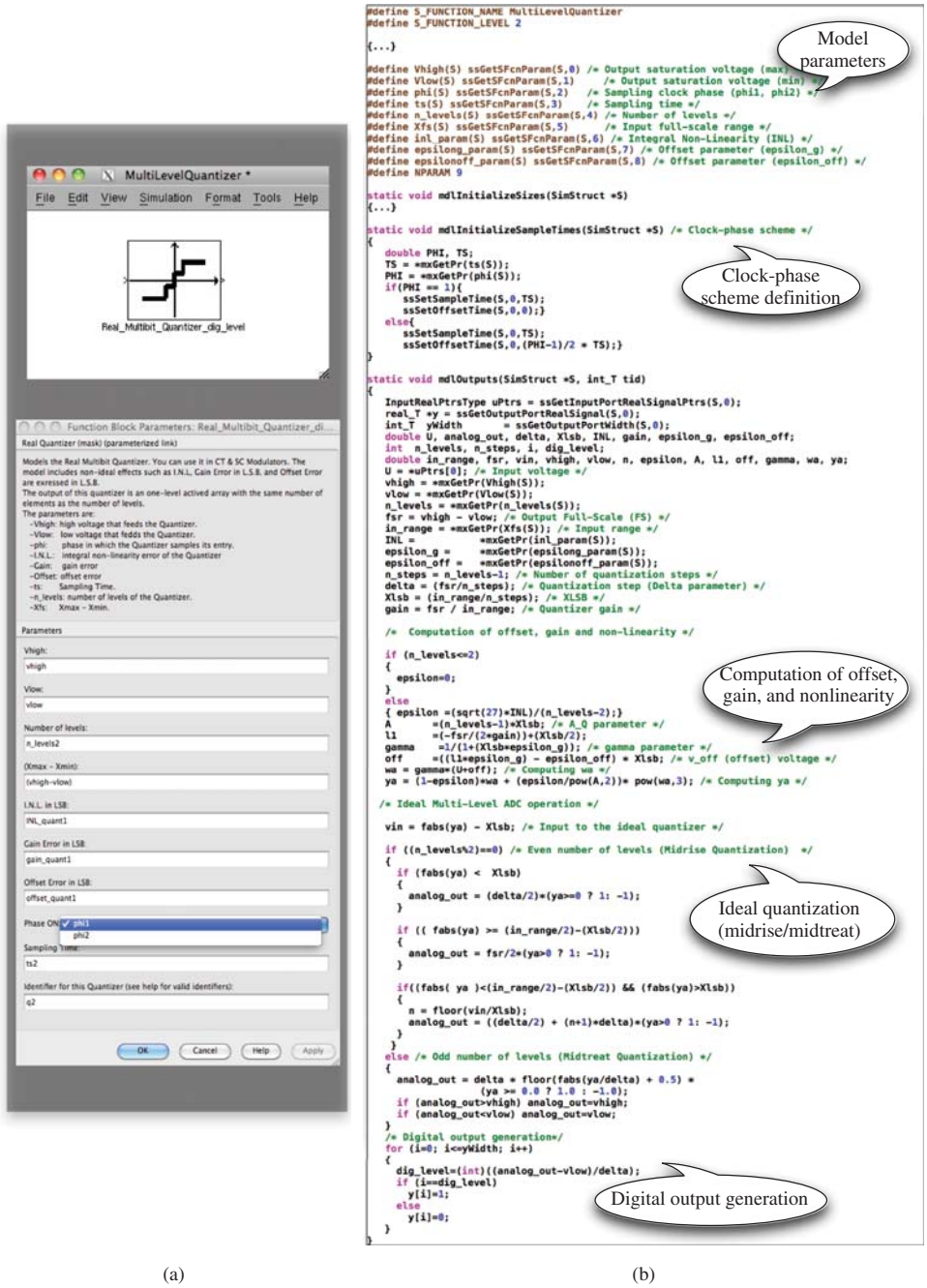


Figure 3.35 Illustrating the behavioral model of multilevel ADCs: (a) S-function block and (b) excerpt of the C-code.

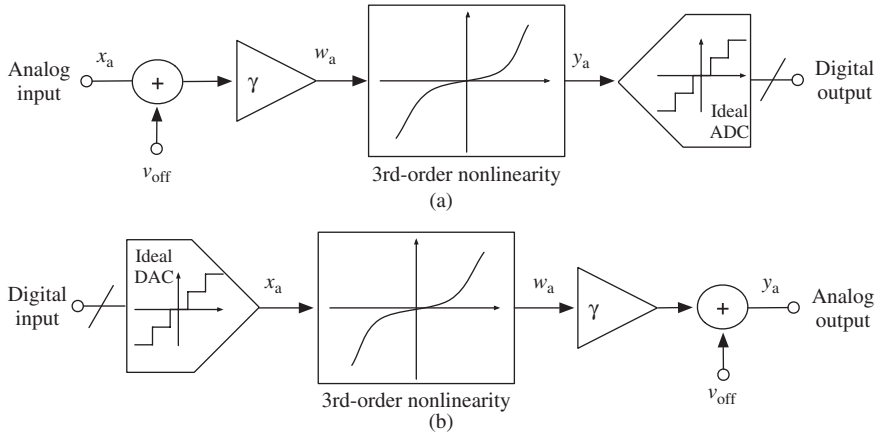


Figure 3.36 Conceptual block diagram of the behavioral model used for the quantizer embedded in ΣΔMs [12]: (a) ADC and (b) DAC.

Figure 3.35b shows the main parts of the S-function file implementing the model of the multilevel ADC, whose conceptual block diagram is shown in Figure 3.36a. This model is based on the one presented in [12], which has been adapted to multilevel quantizers. Essentially, the model consists of the cascade connection of an adder block (that includes the offset error), a linear gain block, a nonlinear transfer function, and an ideal multilevel ADC. The operations of these blocks are sequentially computed in Figure 3.35b that basically codifies the following equations:

$$y_a = (1-\epsilon_0)w_a + \frac{\epsilon_0}{A^2}w_a^3, \quad \text{with} \quad \begin{cases} w_a = \gamma(x_a + v_{off}) \\ \epsilon_0 = \frac{\sqrt{27}}{N_L - 2} \cdot \text{INL} \\ A = (N_L - 1)\Delta \end{cases} \tag{3.32}$$

where N_L is the number of levels (named `n_levels` in Figure 3.35b), Δ is the quantization step (`Xlsb` in Figure 3.35b), and v_{off} is the quantizer offset (`off` in Figure 3.35b).

Modeling Multilevel DACs as S-Functions

Embedded feedback DACs must be accurately modeled at the system level in order to take into account, from the very beginning of the design, some circuit errors that can severely degrade the performance of ΣΔMs. This is the case of multibit (or multilevel) DACs, where the mismatches among the unit circuit elements (capacitors, resistors, current sources, etc.) used for reconstructing the analog feedback signal give rise to a nonlinear input–output characteristics, and consequently to harmonic distortion. This problem is aggravated in CT-ΣΔMs where the feedback DAC transforms the modulator output signal from DT domain to CT domain. This signal reconstruction is very critical and subject to some limiting errors, such as clock jitter error and transient response delay, that have a significant impact on the overall behavior of ΣΔMs, as discussed in Chapter 2. Therefore,

all these effects must be considered in the behavioral models and can be also implemented in MATLAB S-functions.

As an example, Figure 3.37a shows the S-function of a multibit DAC used in CT-ΣΔMs. This model allows implementation of three of the most used DAC waveforms—NRZ, RZ, and HRZ—and includes most critical errors, such as gain error, offset, INL, transient response delay, and clock jitter.

Figure 3.37b shows the C-code of the S-function, highlighting their main parts. Basically, the behavioral model is based on the same concept used for multilevel ADCs, but implemented in a dual way as depicted in Figure 3.36b; that is, starting from an ideal D/A conversion and then applying the effects of the nonlinearity, gain error, and offset. Note that the DAC model uses the number of bits as a model parameter instead of number of levels, although the latter can be also included in the models as shown in Appendix B.

Two different cases have been considered for the DAC delay (named `delay` in Figure 3.37b): a fixed delay, which is independent of the input voltage, and a signal-dependent delay, which is modeled as [40]

$$\text{delay}(v_i) = d_0 + \frac{d_1}{x_1|v_i|} \tag{3.33}$$

where d_0 stands for the fixed delay (`d0` in Figure 3.37b) and d_1 and x_1 (`d1,x1` in Figure 3.37b are curve-fitting parameters that can either be extracted from electrical (transistor-level) simulations or be used for high-level synthesis purposes.

Another important error considered in Figure 3.37 is the clock jitter. This error is modeled as a sampling time instant uncertainty given by

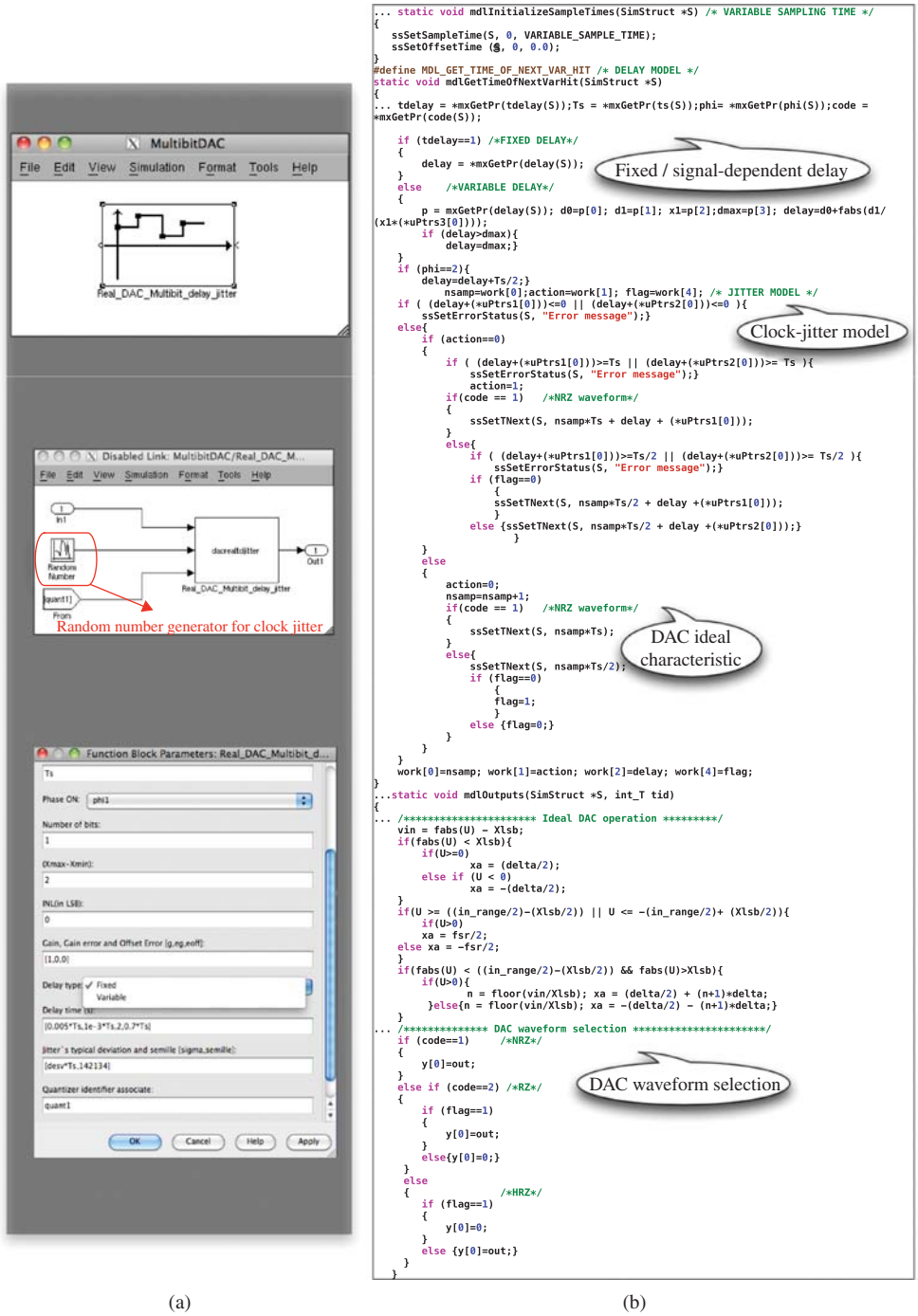
$$t_n = nT_s + \beta_n \tag{3.34}$$

where $n = 1, 2, \ldots$; t_n is the time instant; β_n stands for the time uncertainty implemented in the model as random Gaussian noise source with zero mean and standard deviation provided as a model parameter in Figure 3.37a. This noise source is incorporated as an input port to the S-function block and included in the C-code using `*uPtrs2[0]` as shown in Figure 3.37b.

Note that the C-code modeling the clock jitter strongly depends on the DAC waveform as a different number of clock-signal edges may occur. For that reason, Equation 3.34 is valid for NRZ DAC, while the time uncertainty in the clock edges in RZ or HRZ DACs is modeled as

$$\begin{aligned} t_{n1} &= nT_s + \beta_{n1} \\ t_{n2} &= nT_s + \frac{T_s}{2} + \beta_{n2} \end{aligned} \tag{3.35}$$

where t_{n1} and t_{n2} refer to the time instants of the first and second clock edge, respectively, and β_{n1} and β_{n2} stand for the corresponding clock-edge time uncertainties. Note that a variable sampled time is used in Figure 3.37b (modeled by `VARIABLE_SAMPLE_TIME` parameter) to take into account the effect of clock jitter on the sampling time instant.



(a) (b)

Figure 3.37 Illustrating the behavioral model of multibit DACs: (a) S-function block and (b) excerpts of the C-code including the model of DAC transient response delay, clock jitter, and waveform selection.

3.5 SIMSIDES: A SIMULINK-Based Behavioral Simulator for ΣΔMs

All ΣΔM building blocks and their associated error mechanisms can be modeled as C-MEX S-functions following the methodology described in previous sections. On the basis of this philosophy, a complete toolbox can be developed in the MATLAB/SIMULINK environment for the time-domain behavioral simulation of ΣΔMs. This is the case of SIMSIDES, a SIMulink-based SIgma-DElta simulator that takes advantage of the benefits provided by MATLAB, namely, a friendly user interface, high flexibility for the extension of new models and building blocks, and a powerful set of signal processing routines [5].

SIMSIDES can be used for simulating any arbitrary ΣΔM topology, considering a circuit implementation with either DT circuits, CT circuits, or a mix of them, that is, the so-called hybrid CT/DT-ΣΔMs. In the case of DT-ΣΔMs, behavioral models included in SIMSIDES consider either SC or SI [41] circuit techniques, although the majority of models in SIMSIDES deal with SC circuits, because is the most commonly used DT circuit technique. As far as CT-ΣΔMs is concerned, the main integrator circuit topologies are included in SIMSIDES S-functions, namely: Gm-C, active-RC, Gm-MC, etc. Overall, more than 150 S-functions and 250 behavioral models are available in the toolbox. A list of the most important ones is included in Appendix B, where a brief description of the different models, their main functionality, and parameters is given. This section summarizes the most important features of SIMSIDES, overviewing its model libraries, its general structure, and the user interface.

3.5.1 Model Libraries Included in SIMSIDES

The building blocks modeled in SIMSIDES are grouped into a number of SIMULINK libraries and sublibraries which can be classified attending to different criteria, as conceptually depicted in Figure 3.38. The first classification criterion is related to the modulator system hierarchy level in which a given building block is placed. Attending to this criterion, ΣΔM building blocks are organized in the following libraries: integrators and resonators (basic blocks in ΣΔMs), quantizers, DACs, and auxiliary blocks. The latter include analog and digital mixers, DEM algorithm S-functions, latches models, etc, which are required to simulate some specific ΣΔM architectures. A second classification criterion deals with the circuit technique used for implementing ΣΔM building blocks. Thus, there are sublibraries including FE and LD integrators, Gm-C and active-RC integrators, etc.

All SIMSIDES model libraries are in turn classified into two main categories (not shown in Figure 3.38 for the sake of simplicity): ideal libraries and real libraries. The former ones include only S-functions that contain ideal models of different building blocks—classified attending to the aforementioned criteria. By contrast, real libraries contain S-function behavioral models that include the most critical error mechanisms that degrade the performance of ΣΔMs. Table 3.2 summarizes all building blocks modeled in SIMSIDES as well as the error mechanisms included in their S-function behavioral models.

As an illustration, Figure 3.39 shows some of the most representative libraries and sublibraries including integrators, resonators, quantizers, and DACs. As illustrated in

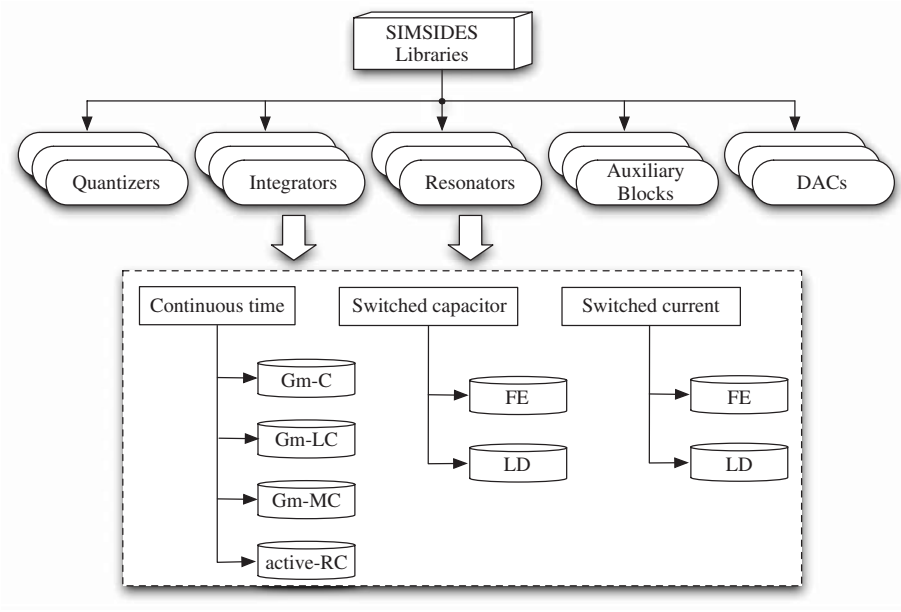


Figure 3.38 Classification of SIMSIDES model libraries.

Table 3.2 Circuits and error mechanisms modeled in SIMSIDES

Circuit Technique	Building Block	Error Mechanism
Switched-Capacitor	Amplifiers	Finite and Nonlinear DC gain, incomplete settling error, output swing limitation, thermal noise
	Switches	Thermal noise, finite and nonlinear switch on-resistance
	Capacitors	Mismatch, nonlinearities, parasitic capacitances
Switched-Current	Memory cells and Integrators	Linear and nonlinear gain error, thermal noise, finite output–input conductance ratio error, charge injection error, incomplete settling error
Continuous-Time	Integrators	Finite and nonlinear DC gain, nonlinear transconductance, thermal noise, output swing limitation, transient response
All circuit techniques	Clock generator	Clock jitter
	Comparators	Hysteresis and offset
	Quantizers/DACs	Nonlinearity (INL), gain error, excess loop delay, offset

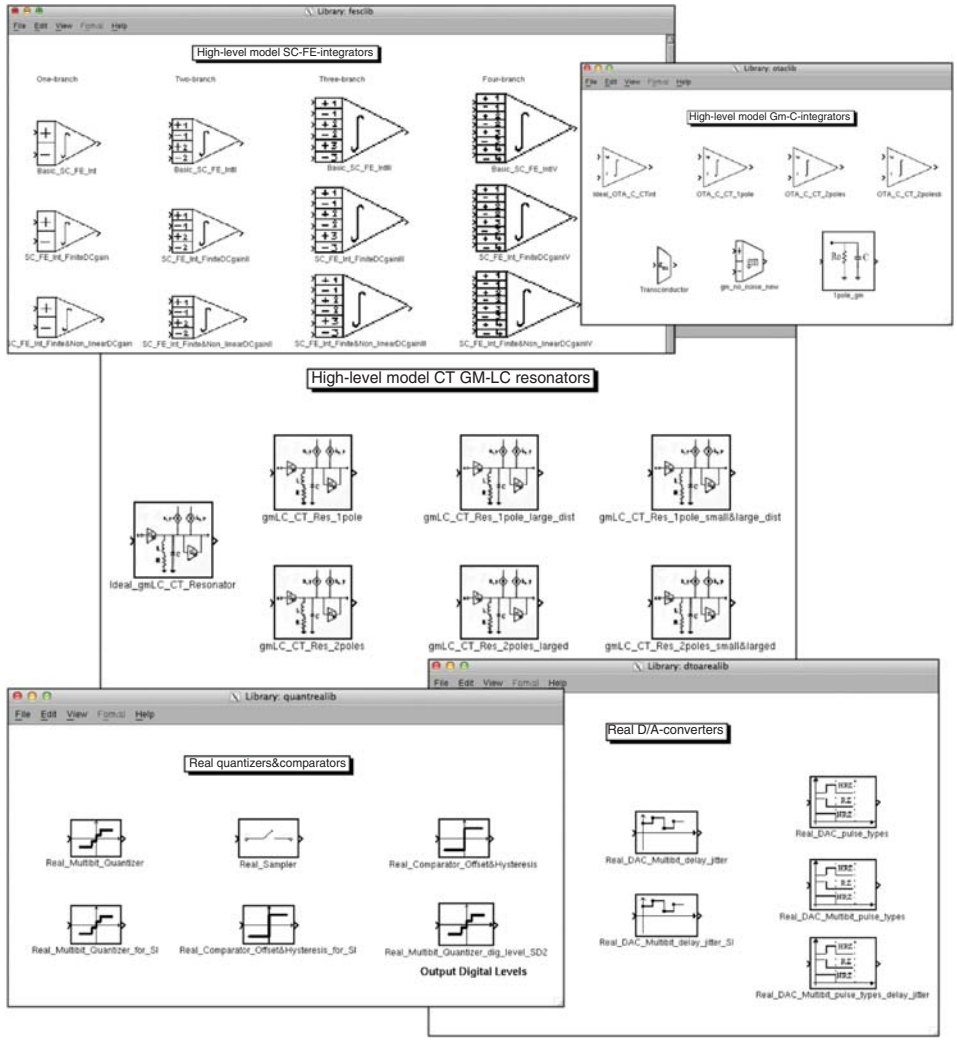


Figure 3.39 Illustrating some SIMSIDES model libraries.

Figure 3.40, there is a number of different SC integrator S-function blocks with different number of input branches and nonideal effects included in their models, going from ideal models to the most precise ones that include all circuit nonidealities. The benefits of this approach are twofold: On the one hand, it allows to evaluate the impact of isolated circuit error mechanisms in a very simple way, without dealing with model parameters. This is particularly appropriate for the high-level sizing process where different error parameters are taken as design variables. On the other hand, the use of different building-block model approaches, which evolve from the most ideal and simplest approximations to the most accurate and complex ones, may be particularly useful for novel designers, who are not very familiar with some circuit-level parameters used in the most precise behavioral models.

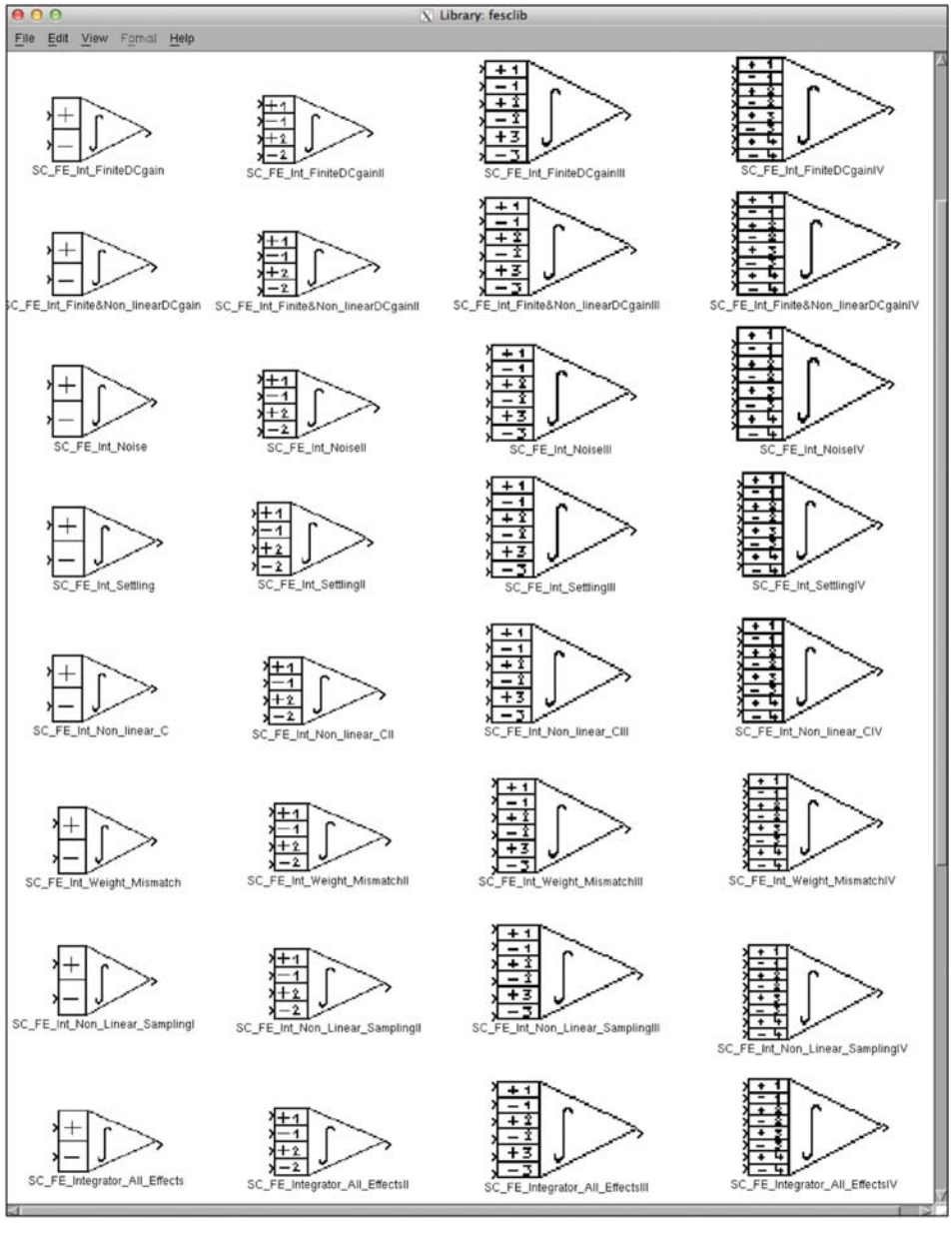


Figure 3.40 SIMSIDES library including different SC FE integrator model approaches.

Apart from the mentioned libraries containing isolated ΣΔM building blocks, SIMSIDES includes also some additional libraries with examples of the most commonly used ΣΔM architectures, considering both LP and BP topologies, single-loop and cascade, single-bit and multibit embedded quantization, different circuit techniques (DT, CT, and hybrid CT/DT), etc.

3.5.2 Structure of SIMSIDES and User Interface

Figure 3.41 shows the general structure of SIMSIDES. First, the modulator architecture is defined by properly interconnecting the building blocks included in the SIMSIDES libraries discussed in the previous section. After the modulator block diagram has been created, the designer can set the model parameters and the simulation options required by the toolbox to do the simulation. Most commonly used analyses can be carried out, namely, time-domain simulations to obtain output spectra and SNR/SNDR-versus-input curves, parametric simulations considering the variation of a given model parameter, Monte Carlo simulations, etc. Output data generated by the simulator consists of time-domain series which can be further processed to get typical performance figures. This way, both integrator input/output histograms and/or output spectra are computed using the routines provided by the signal processing toolbox in MATLAB [28]. Other typical performance metrics can be evaluated, such as SNR/SNDR, harmonic and intermodulation distortion, etc. These figures are computed using a collection of internal MATLAB routines, specifically developed for SIMSIDES [5].

SIMSIDES includes a graphical user interface (GUI) that allows designers to browse through all steps of the simulation and to postprocess the simulation results. As an illustration, Figure 3.42 shows some of the most important parts of the SIMSIDES GUI, highlighting some of its menus. Although a more detailed explanation about SIMSIDES

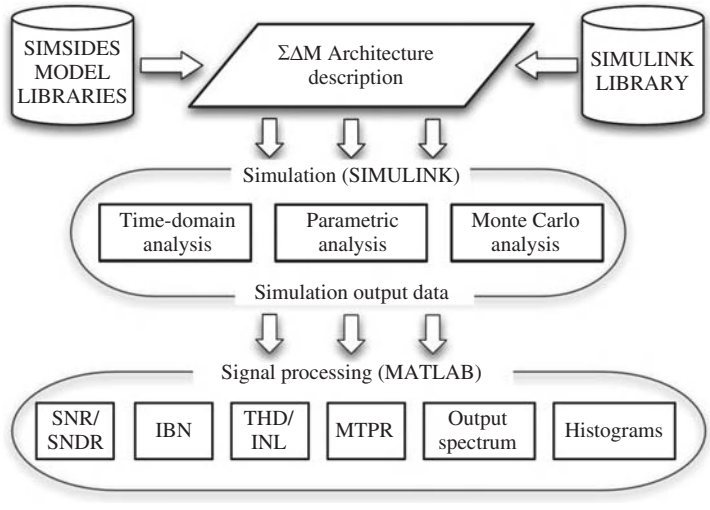


Figure 3.41 General structure of SIMSIDES.

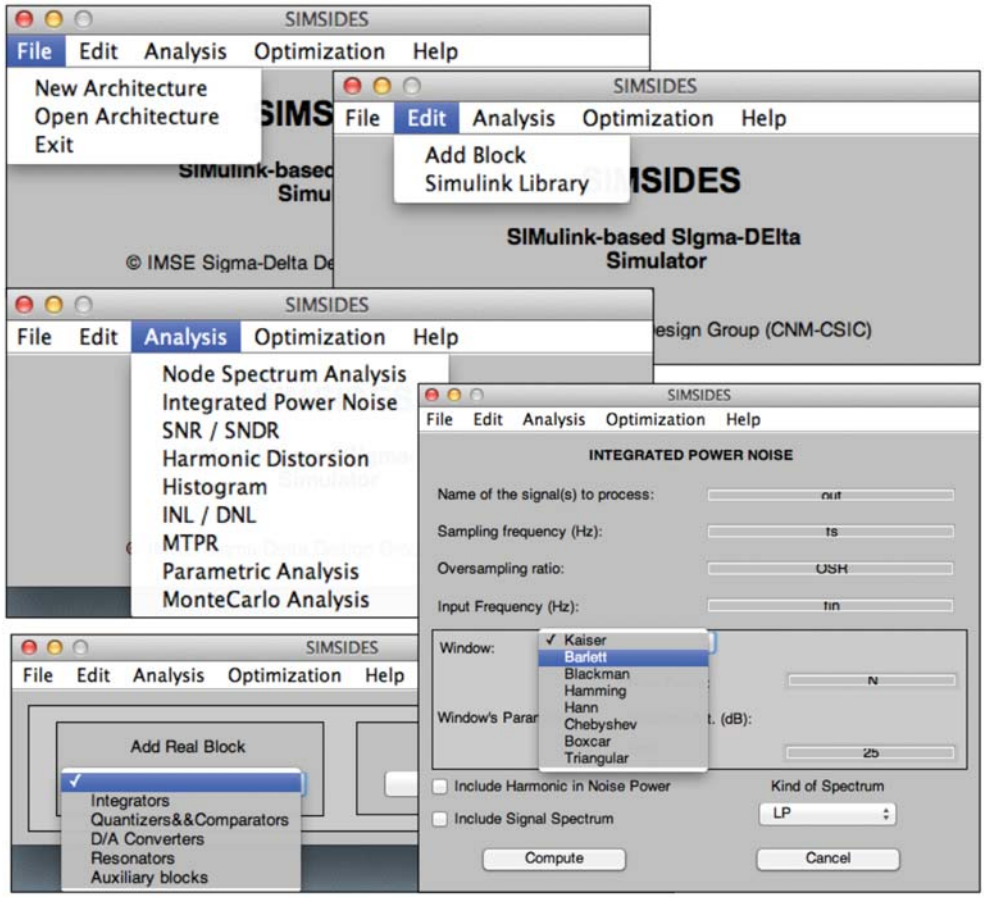


Figure 3.42 Illustrating some parts of the SIMSIDES GUI.

GUI is given in Appendix A, the use of the toolbox is illustrated in this section through a simple example based on the behavioral model of a cascade 2-1 SC-ΣΔM, where several types of analyses are carried out.

Creating a New ΣΔM Block Diagram

SIMSIDES is started by typing `simsides` in the MATLAB command window and the main window is displayed. Selecting the pop-up menu named `File` and then `New Architecture` (Figure 3.42), a new (empty) SIMULINK model window is displayed. This is illustrated in Figure 3.43 for a cascade 2-1 SC topology that can be created by adding building blocks from `Edit->Add Block` pop-up menu and then choosing the circuit technique to implement the integrator blocks—SC FE integrators in this case. The corresponding model library—named `fesclib` in this example (illustrated

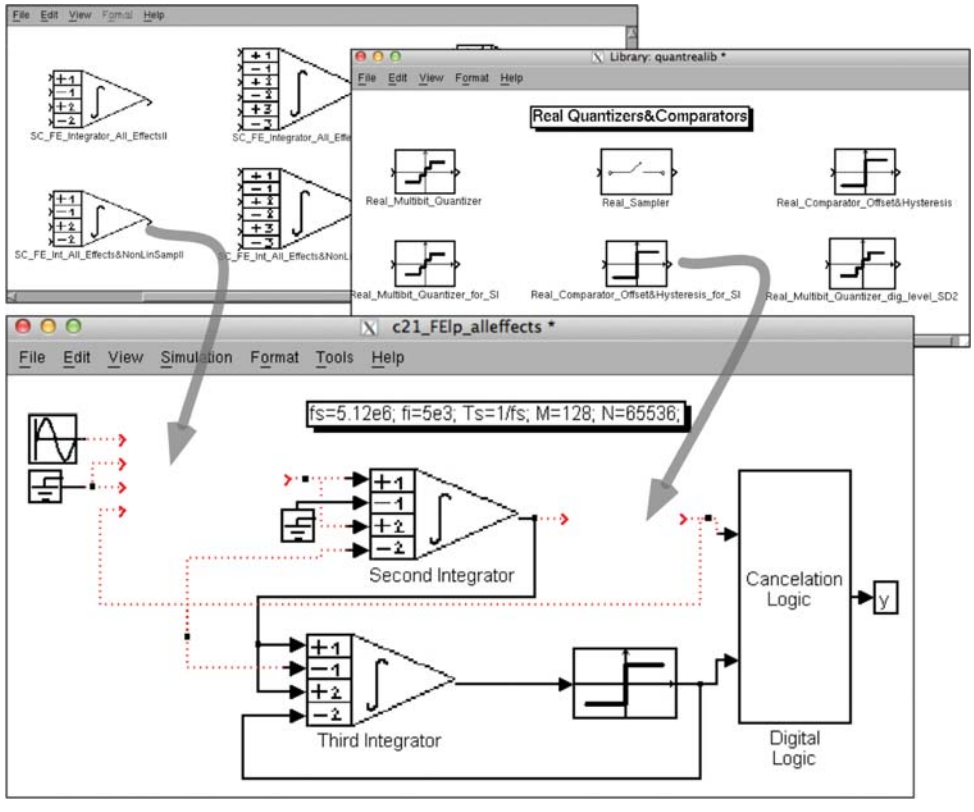


Figure 3.43 Creating the block diagram of a cascade 2-1 SC-ΣΔM in SIMSIDES.

in Figure 3.40)—is displayed and the appropriate integrator blocks can be included in the new model by dragging and dropping the corresponding blocks. A similar procedure can be followed to incorporate the comparator model and, once all blocks are properly connected, the block diagram in Figure 3.43 is finished.

Setting Model Parameters

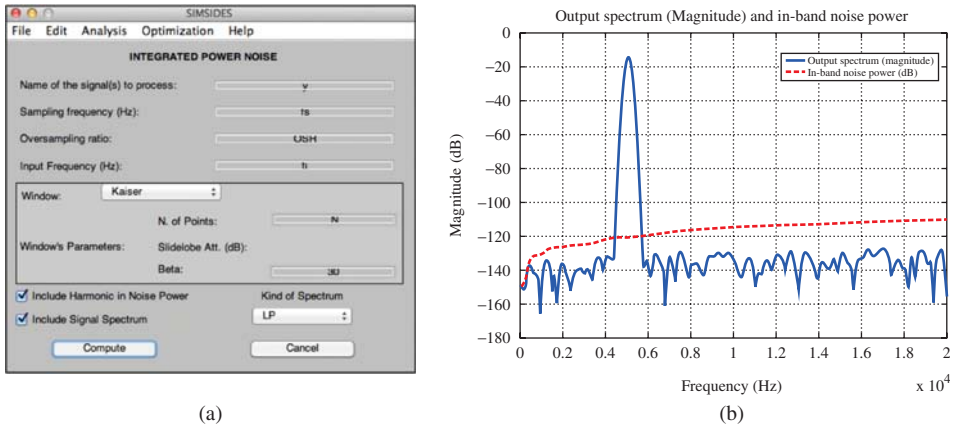
Once the block diagram has been defined, the next step consists of setting the model parameters. Although this is something that can be done in the MATLAB command window, as the number of model parameters increases, it becomes more practical to save their values in an M-file. In addition to the model parameters themselves, some global parameters must be defined in order to run the simulation. These parameters are input signal frequency (`fi` in Figure 3.43), sampling time and frequency (`Ts,fs`), and the number of simulation clock cycles (`N=65536`). The latter must be included in the `Configuration Parameters` menu of the SIMULINK model window where the `solver options` is also configured (`Fixed-step` and `discrete` in this example).

Simulation Analyses

After setting the model parameters, a simulation can be started from the `Simulation->Start` pop-up menu as is usually done in SIMULINK. This is the usual procedure followed in SIMSIDES to run a one-sample simulation that is used for computing an output spectrum, for instance. Thus, once the simulation has finished, the output spectrum can be computed and plotted from the `Analysis->Node Spectrum Analysis` pop-up menu (Figure 3.42). Selecting this option, a new window is displayed to set some parameters required to compute the output spectrum, such as the sampling frequency, the signal to be processed, as well as some parameters related to the window used for FFT computation. In the example shown in Figure 3.44a, a Kaiser window with `N` points and a beta of 30 has been used. As an illustration, Figure 3.44b shows the in-band output spectrum of the modulator in Figure 3.43. This figure plots also the IBN corresponding to OSR = 128. This data has been obtained by choosing `Analysis->Integrated Power Noise` pop-up menu. In a similar way, the SNR/SNDR can be computed by selecting `Analysis->Integrated Power Noise` in the SIMSIDES main window.

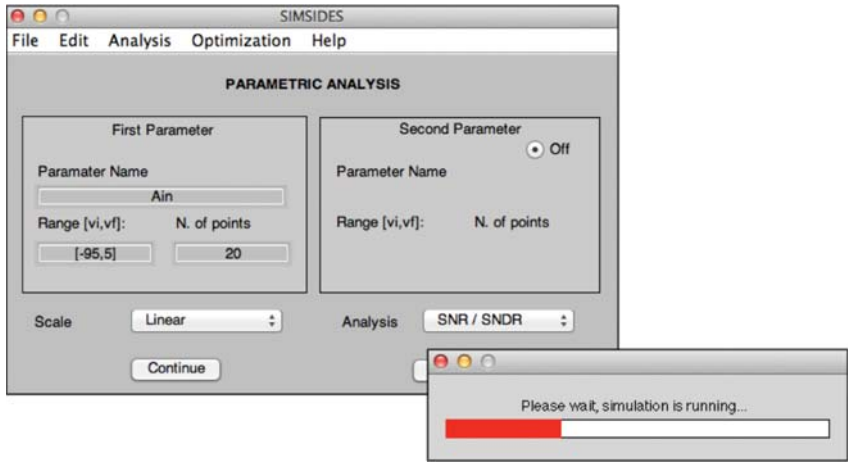
Apart from the mentioned simulation analyses, other useful performance metrics can also be evaluated in SIMSIDES, such as linearity—considering either static figures such as INL or dynamic figures such as THD or MTPR. All these analyses can be combined with a parametric analysis in order to evaluate the impact of a given model parameter. This kind of analysis is especially useful for high-level sizing, as will be illustrated in the next section. Here, an example of this analysis is shown to obtain a typical SNR-versus-input figure.

To this purpose, `Analysis->Parametric Analysis` pop-up menu is chosen and a new window is displayed as shown in Figure 3.45a. There is the possibility to select between one-parameter analysis or two-parameter analysis. In this example, the former case is chosen and the input signal amplitude range is defined. After that, the next steps consist of defining several parameters required to compute the SNDR. Once all parameters have been properly defined, the parametric simulation is started. A simulation-progress window is graphically displayed and finally the results are plotted in Figure 3.45b.

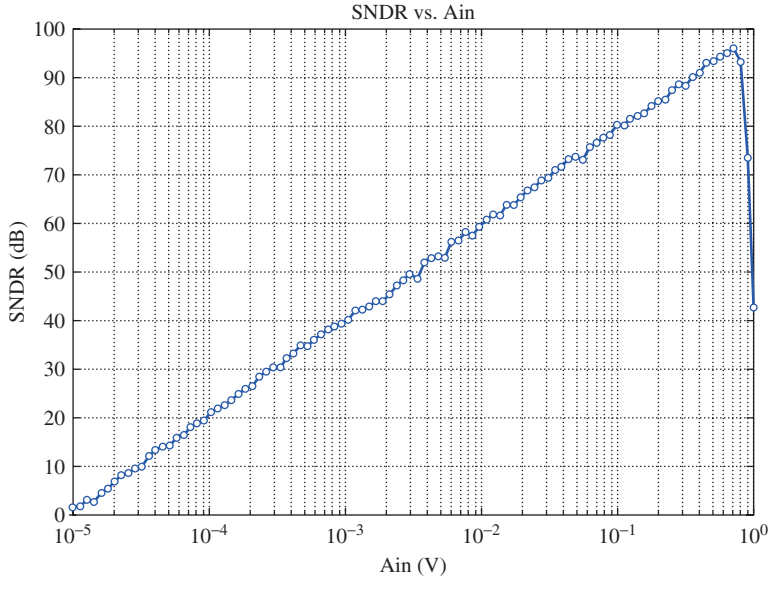


(a) (b)

Figure 3.44 Computing IBN in SIMSIDES: (a) user window and (b) in-band output spectrum.



(a)



(b)

Figure 3.45 Computing SNDR versus input amplitude in SIMSIDES: (a) user window and (b) SNDR versus input amplitude.

3.6 Using SIMSIDES for the High-Level Sizing and Verification of ΣΔMs

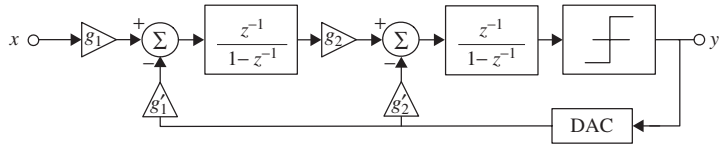
To conclude this chapter, this section illustrates the use of SIMSIDES for the high-level sizing and verification of ΣΔMs. To this purpose, two different ΣΔM architectures and circuit techniques are considered as case studies:

- An SC second-order single-loop single-bit ΣΔM
- A CT fifth-order cascade 3-2 multibit ΣΔM

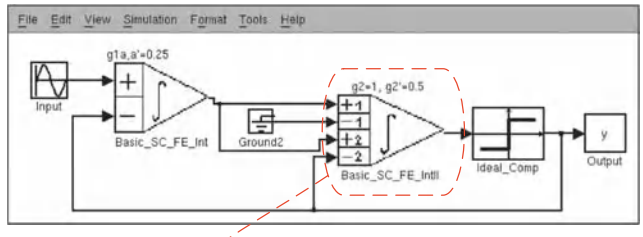
3.6.1 SC Second-Order Single-Bit ΣΔM

Let us consider the Z-domain block diagram of a second-order ΣΔM shown in Figure 3.46a. In this example, an ideal feedback DAC and a one-bit quantization with 1-V FS range will be assumed. This block diagram can be implemented in SIMSIDES as shown in Figure 3.46b, where Z-domain transfer functions have been replaced by SC FE integrator S-function blocks as illustrated in Figure 3.46c.

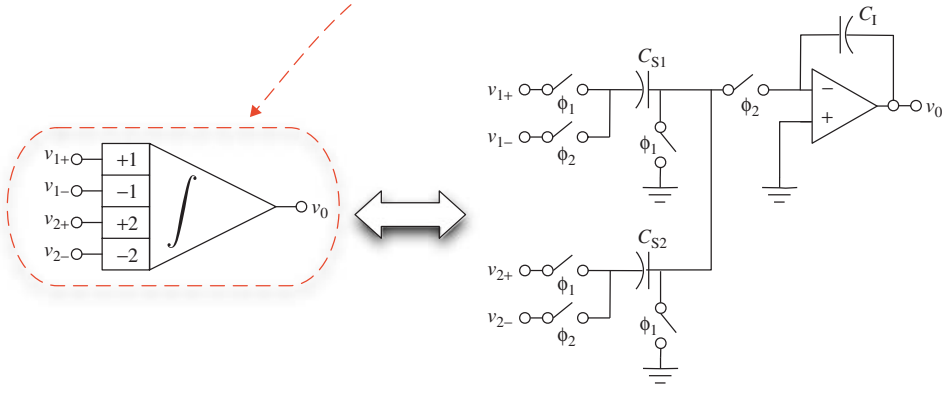
Figure 3.47 represents the ideal modulator output spectrum and the IBN for OSR = 128, considering a sampling frequency of $f_s = 2.56$ MHz and an input tone



(a)



(b)



(c)

Figure 3.46 Second-order SC-ΣΔM case study: (a) Z-domain block diagram, (b) SIMSIDES implementation, and (c) symbol of a two-branch SC integrator in SIMSIDES and its equivalent circuit.

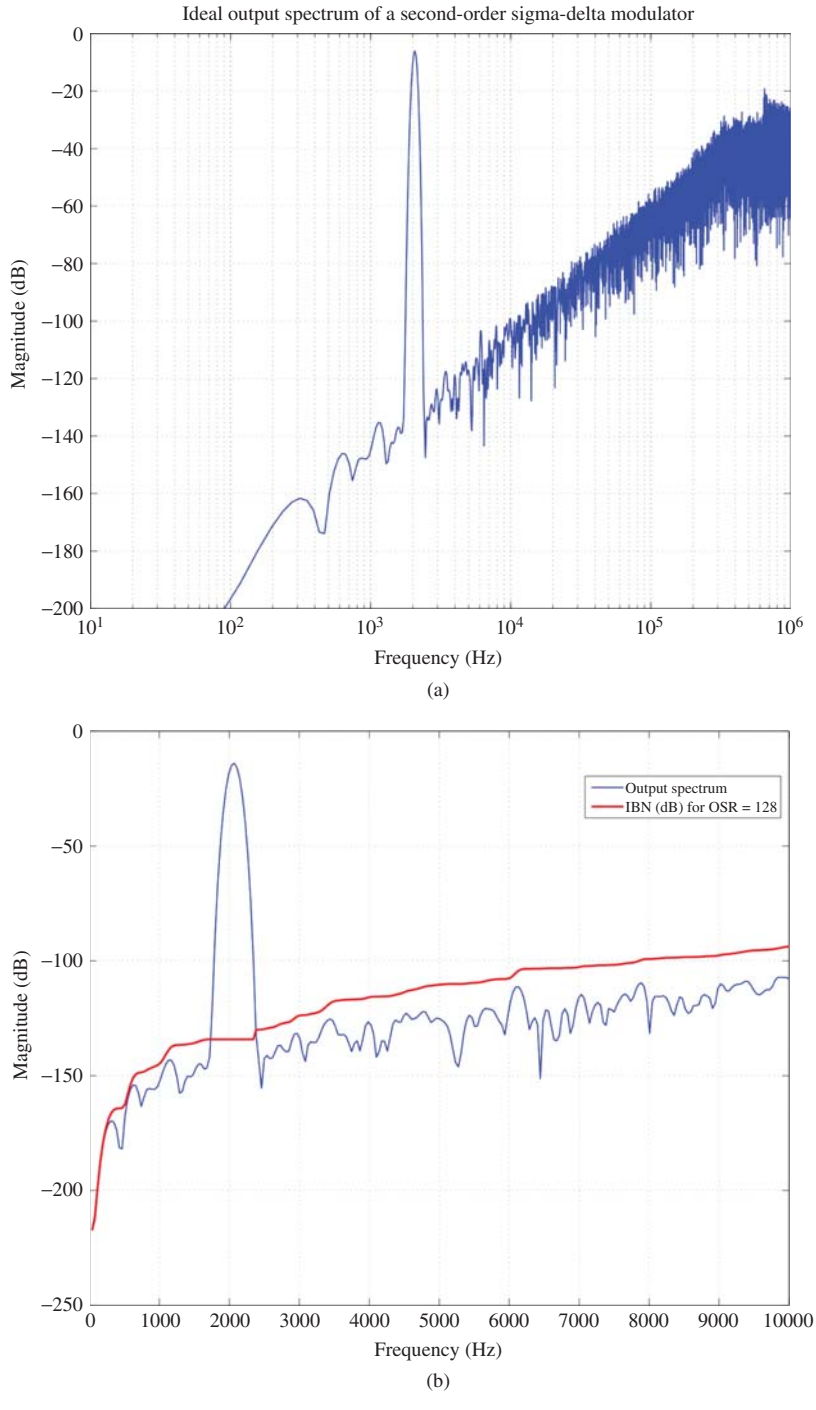


Figure 3.47 (a) Ideal output spectrum of the modulator in Figure 3.46 and (b) IBN for OSR = 128.

with half-scale amplitude (0.5 V). The shaped quantization noise increases at a rate of 15 dB/octave—according to theoretical predictions. The ideal SNR computed by SIMSIDES is 87 dB ($\simeq$ 14 bit), 103 dB ($\simeq$ 16.8 bit), and 119 dB ($\simeq$ 19.4 bit), respectively, for OSR = 128, 256, and 512, corresponding to a signal bandwidth of B_w =10, 5, and 2.5 kHz, respectively. Alternatively, the same ideal effective resolutions can be obtained for a signal bandwidth of $B_w = 10$ kHz if $f_s = 2.56$, 5.12, and 10.24 MHz. In practice, this ideal performance is degraded by the action of circuit errors. In this example, the impact of the following nonidealities will be evaluated: opamp finite DC gain, thermal noise, and incomplete transient response. For each nonideality, the error bound that allows achieving the ideal effective resolution will be found in order to map modulator-level specifications onto building-block specifications.

Effect of Amplifier Finite DC Gain

There are two possibilities to analyze the effect of a given error in SIMSIDES. One consists of using the behavioral model that contains only that isolated effect, whereas the other consists of using a complete behavioral model in which all error parameters are set ideal except for those related to the nonideality which is going to be evaluated. In this example, the first approach will be followed. Thus, in order to simulate the effect of finite DC gain A_v the corresponding S-function models of SC FE integrators available in SIMSIDES are used.

A parametric analysis can be carried out in SIMSIDES in order to obtain the minimum (or critical) value of A_v—represented by $A_{v_{crit}}$—required to achieve the ideal resolution. This is illustrated in Figure 3.48 where the SNDR is plotted versus A_v for OSR = 128, 256, and 512, resulting in $A_{v_{crit}} > 100$, 200, and 400, respectively.[15]

Effect of Thermal Noise

Following the same procedure as in the previous section, the effect of circuit (thermal) noise can be simulated. Two main thermal noise sources are considered, namely, the input-referred noise of the opamp (denoted as V_n) and the kT/C noise—evaluated by varying the value of the sampling capacitance C_s. In this example, only the contribution of the front-end integrator will be taken into account.[16]

Figure 3.49 depicts SNR versus V_n and C_s for OSR =128, 256, and 512. According to Equation 3.14, ideal values for other model parameters except V_n and C_s are considered. Note from Figure 3.49 that a critical value of $V_{n_{crit}} < 1\text{nV}/\sqrt{\text{Hz}}$ can be tolerated in all cases while $C_{s_{crit}} > 2$, 7, and 10 pF is approximately required for OSR =128, 256, and 512, respectively. Note that such large values required for C_s are a direct consequence of the high effective resolutions with low modulator filter order and single-bit embedded quantizer. These capacitor values may force reducing the digitized signal bandwidth B_w in order to reduce the impact of incomplete settling with a feasible power consumption.

[15] The parametric analysis varying A_v was carried out considering bandwidths of 10 and 20 kHz (and their corresponding values of f_s for OSR = 128, 256, and 512). As this nonideality is static, note from Figure 3.48 that the absolute value of B_w does not have any influence on the results obtained.

[16] Note that the noise sources of the second integrator are attenuated in the signal band by the gain of the front-end integrator.

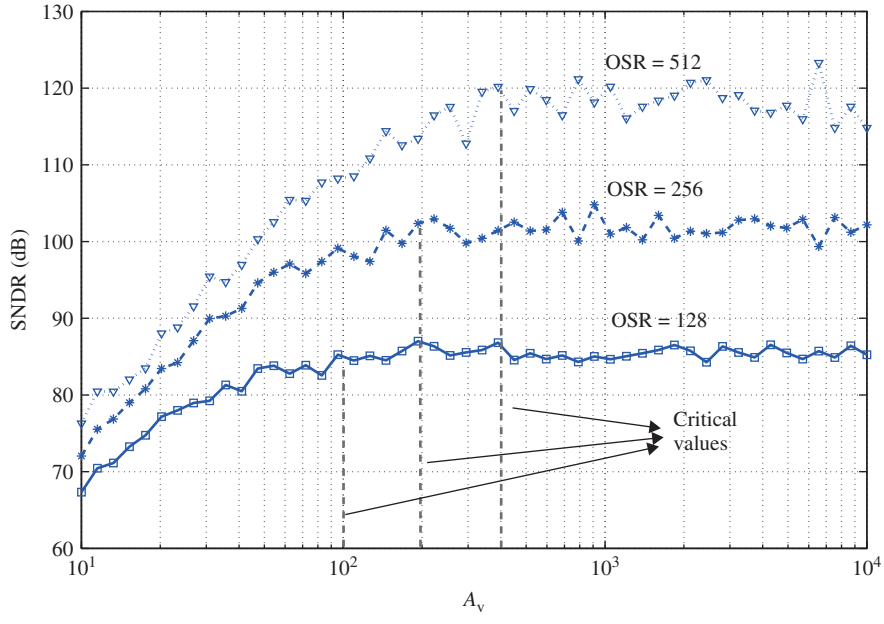


Figure 3.48 SNDR versus A_v of the ΣΔM in Figure 3.46 for different values of OSR.

Effect of the Incomplete Settling Error

In order to evaluate the impact of the incomplete settling error, two model parameters are considered, namely, the OTA transconductance g_m and the maximum OTA output current I_o. The former determines the minimum requirements in terms of GB, whereas the latter limits the maximum SR that can be achieved for a given value of the equivalent load capacitance—automatically computed by the SIMSIDES model as discussed in Section 3.4. As in previous errors, only the impact of the front-end integrator is considered here for illustration purposes.

Figure 3.50 shows the effect of g_m and I_o on the performance of the modulator for different values of OSR. Note that in this case the absolute value of f_{in} and B_w are important, because the incomplete setting affects the dynamic response of the integrators. In this example, an input tone at $f_{in} = B_w/5$ is considered, for $B_w =$ 10 kHz. Figure 3.50a depicts the modulator SNDR versus g_m, for C_s =10 pF and I_o = 5 mA. Considering these simulation conditions, the effect of SR can be neglected. Alternatively, Figure 3.50b depicts the SNDR versus I_o considering $g_m = 5\,\text{mA V}^{-1}$, so that only the effect of SR is evaluated and the impact of GB is nulled. The critical values of g_m and I_o for the different cases are highlighted in the figure.

Cumulative Effect of All Errors

Simulations carried out in previous sections analyzed the isolated impact of the most important circuit errors limiting the performance of the ΣΔM shown in Figure 3.46. However, that performance can be further degraded as a consequence of the cumulative effect

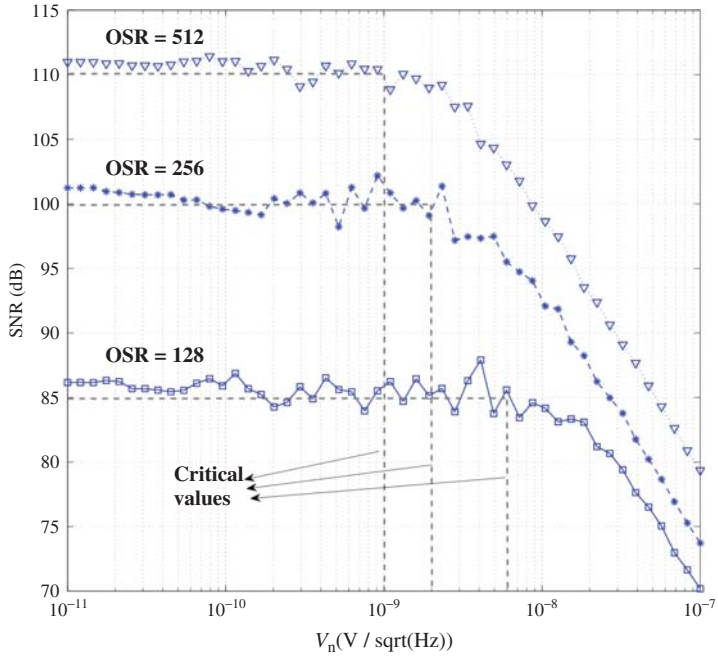


Figure 3.49 Effect of thermal noise sources on the ΣΔM in Figure 3.46: (a) SNR versus V_n and (b) SNR versus C_s.

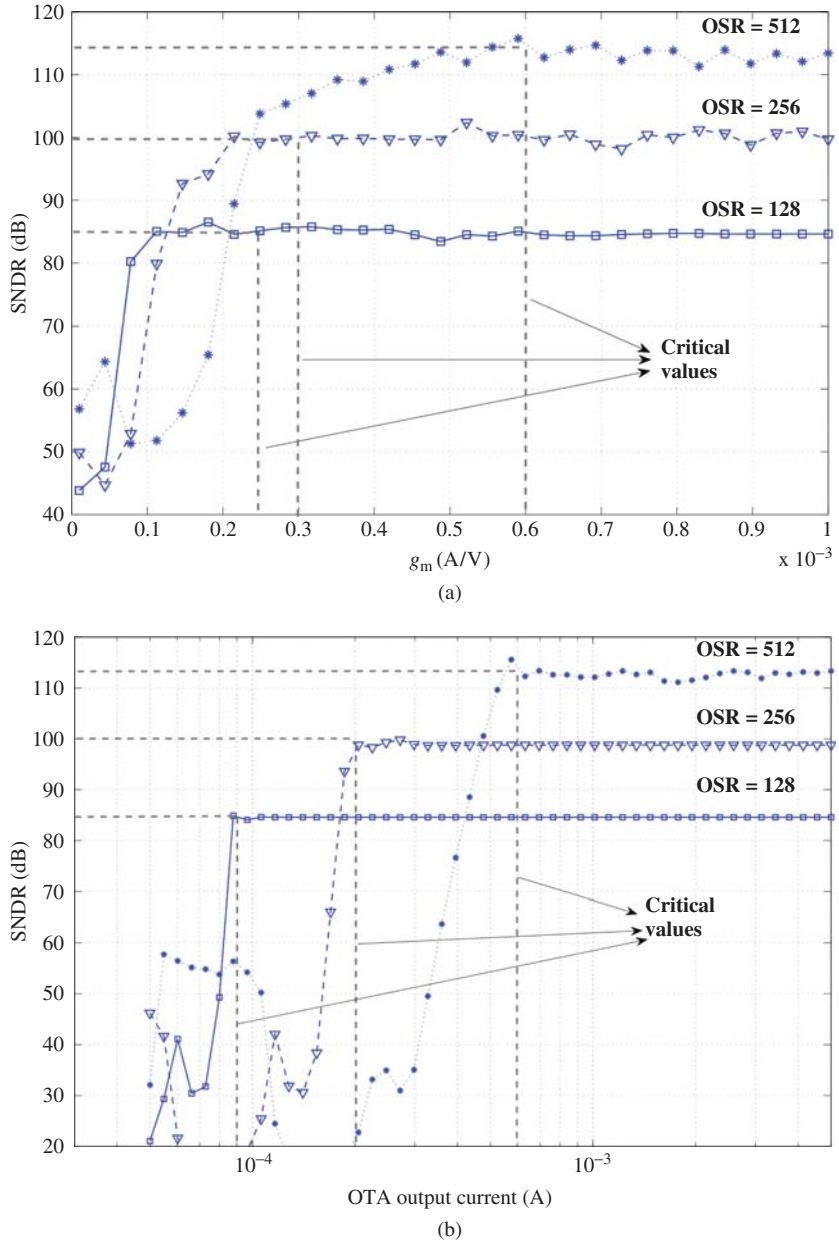


Figure 3.50 Effect of incomplete settling on the ΣΔM in Figure 3.46 for $B_w = 10$ kHz: (a) SNDR versus g_m for $I_o = 5$ mA and (b) SNR versus I_o for $g_m = 5\,\text{mA V}^{-1}$.

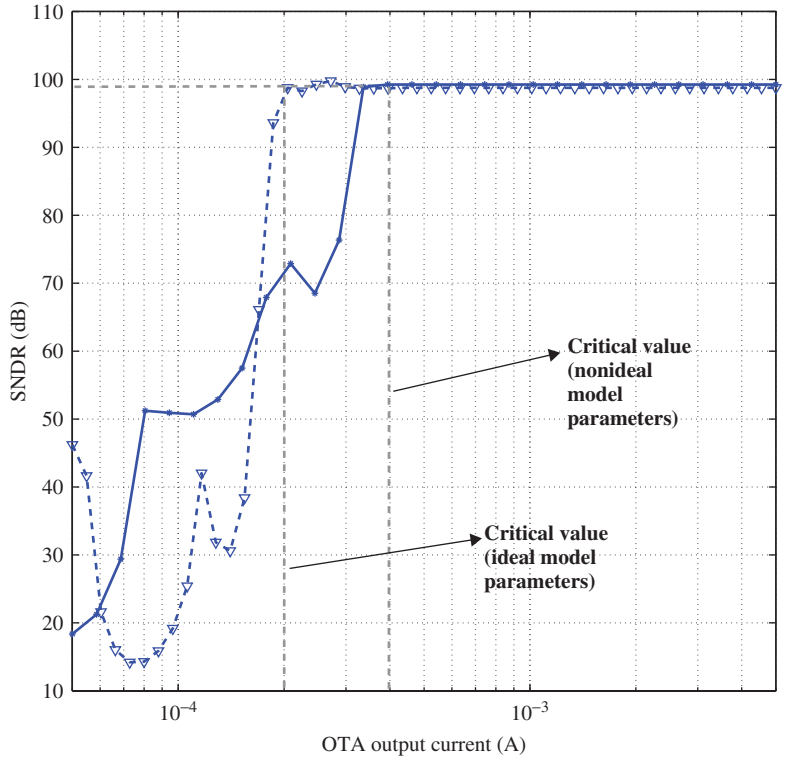


Figure 3.51 Illustrating the cumulative and isolated effect of the amplifier output current I_o.

of the different errors acting together—which is actually the case in practical situations! To illustrate this effect, Figure 3.51 depicts the modulator SNDR versus I_o considering both its isolated effect and the cumulative effect of this parameter, that is, considering other nonideal model parameters. Note that in the latter case, a more demanding value of I_o is required to achieve the specified resolution.

Figure 3.52 plots the output spectrum of the modulator considering all circuit errors, taking into account the critical values of error model parameters obtained for OSR = 512. Figure 3.53 depicts the SNDR versus input signal level for different values of OSR and B_w. Note that the achieved effective resolution is according to the specifications.

The results of the high-level sizing process are summarized in Table 3.3, where system-level (modulator) specifications are mapped into building-block (integrator) specifications. For the sake of completeness, the required values for GB and SR are also included in the table. Note that this data takes into account the interaction of different circuit errors considered in the example and hence may be more restrictive than that obtained from the isolated analysis described. In this case, the effect of switch on-resistance on the OTA transconductance and GB described in Section 3.4.1 has not been taken into account for the sake of simplicity. However, this effect should be considered in practice, resulting in more demanding integrator dynamic specifications. As an illustration, Figure 3.54 depicts the information provided by the corresponding SIMSIDES integrator model if the mentioned

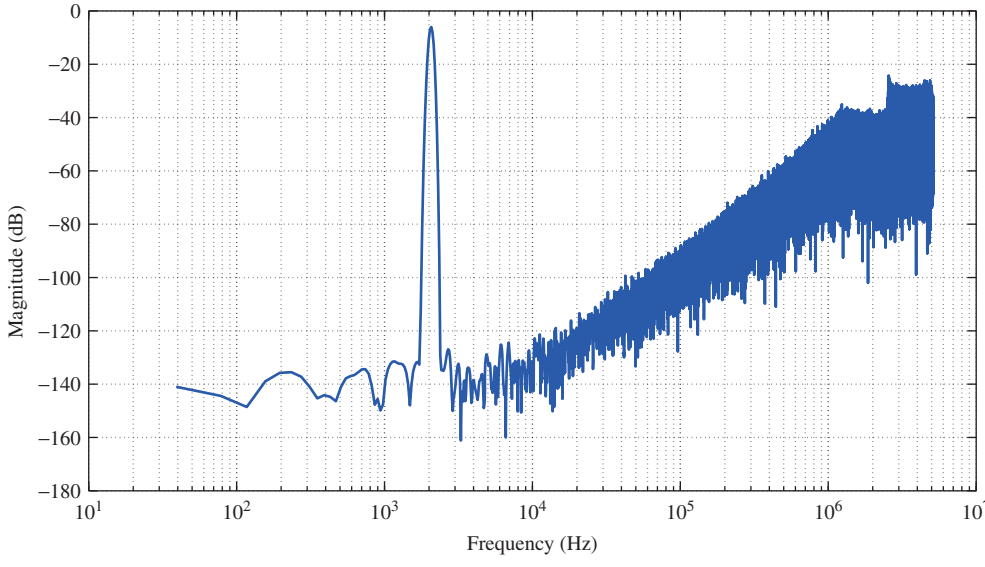


Figure 3.52 Output spectrum of the ΣΔM in Figure 3.46 considering the effect of all circuit errors.

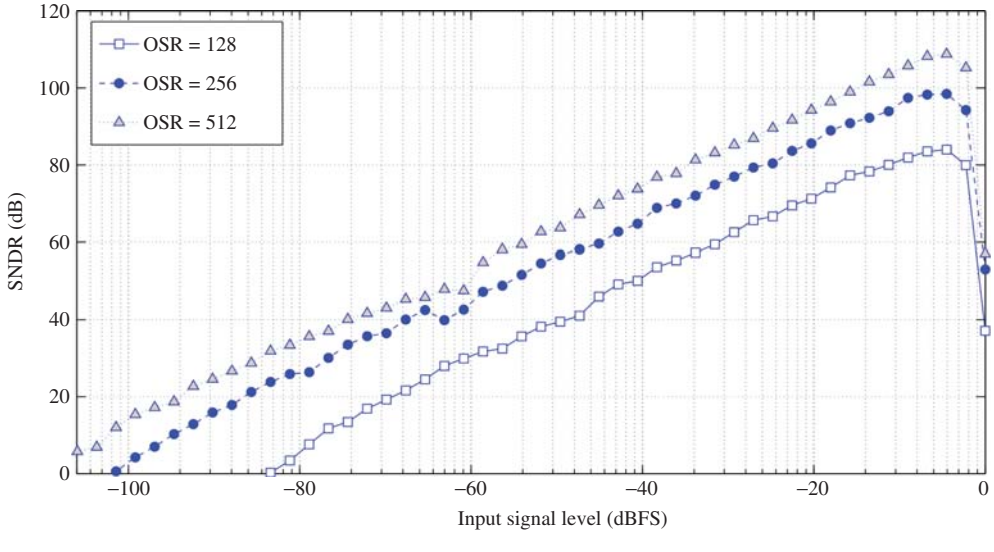


Figure 3.53 SNDR versus input signal level for $B_w = 10$ kHz considering all circuit errors.

Table 3.3 High-level sizing of the second-order SC-ΣΔM shown in Figure 3.46 ($C_{eq} = 2.6\,pF$)

Model Parameter	OSR = 128	OSR = 256	OSR = 512
A_v	100	400	400
V_n (nV/$\sqrt{Hz}$)	6	2	1
C_s (pF)	2	7	10
g_m (mA V^{-1})	0.25	0.3	0.6
I_o (mA)	0.09	0.5	1
GB (MHz)	100	120	240
SR (Vμs)	46	200	400

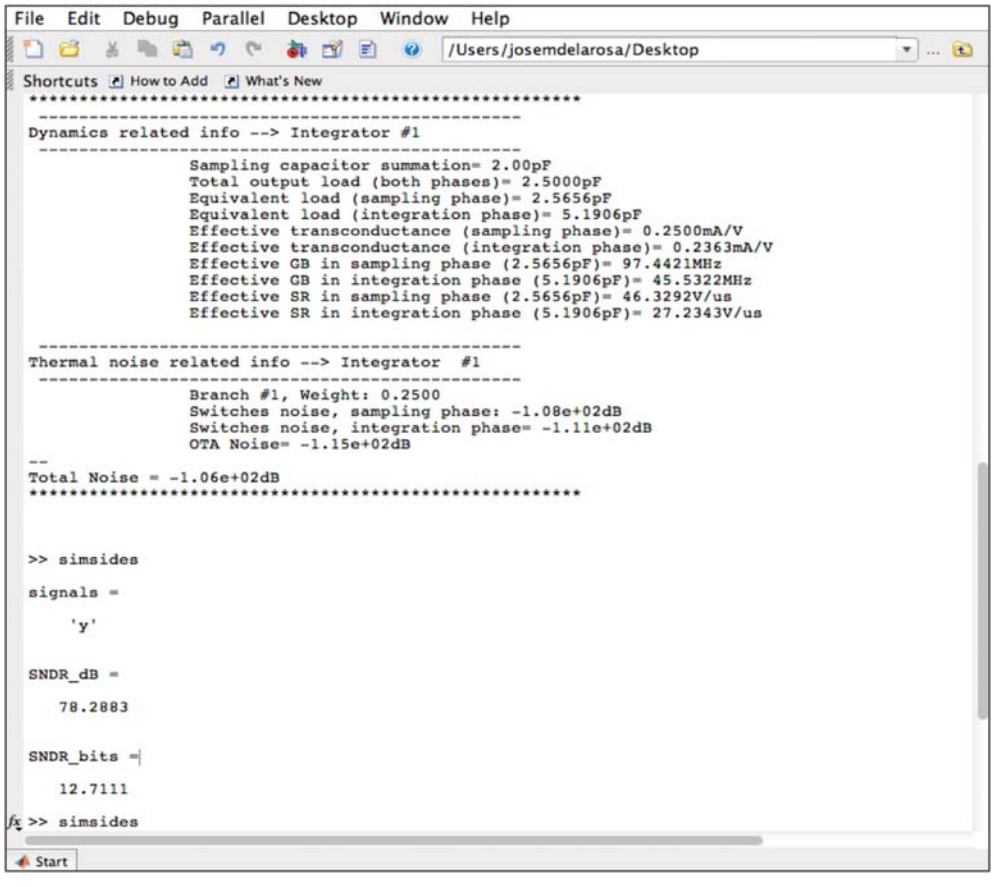


Figure 3.54 Information provided by the SIMSIDES SC FE integrator model when the effect of switch on-resistance is taken into account in the transient response.

degradation is considered. Among other data, the model provides the effective values of the equivalent load capacitance, GB, and SR obtained for each clock phase. Taking into account these model parameters, the half-scale SNDR for OSR = 128 is 78.3 dB, that is, approximately 9 dB less than the ideal one.

3.6.2 CT Fifth-Order Cascade 3-2 Multibit ΣΔM

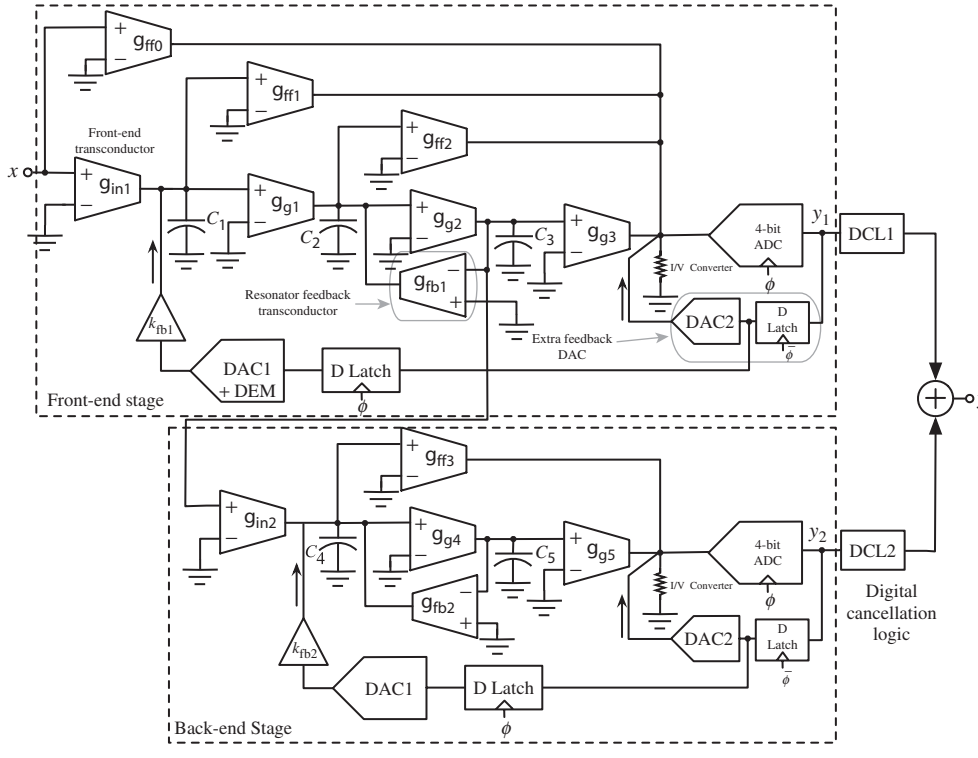
The second case study is a CT two-stage cascade ΣΔM consisting of a third-order front-end stage and a second-order back-end stage. Figure 3.55a shows the conceptual block schematic of the modulator and its corresponding implementation in SIMSIDES is depicted in Figure 3.55b [42]. The front-end stage is made up of an integrator and a resonator, whereas the second-stage loop filter is essentially a resonator. Multibit (4-bit) quantization and NRZ feedback DAC are used in both stages and DEM techniques are also included to reduce the impact of DAC mismatch on the modulator linearity. In both stages an extra feedback branch—made up of a DAC and a D-latch—is connected from the output to the input of the quantizer to compensate for the effect of excess loop delay [43], as described in Section 2.10.

The loop filter of both stages is implemented using Gm-C integrators, whereas current-mode DACs are used in the feedback loop. The modulator has been synthesized in the CT domain following the methodology described in [44] and resonator poles are placed at an optimum position to minimize the magnitude of NTF within the signal bandwidth. Similar to any other cascade ΣΔM, the DCL functions of each stage (i.e., $DCL_{1,2}$, not explicitly shown in Figure 3.55 for the sake of simplicity) can be derived from the STF and NTF of both modulator stages, by annulling the first-stage quantization error at the overall modulator output [44] (Section 1.5).

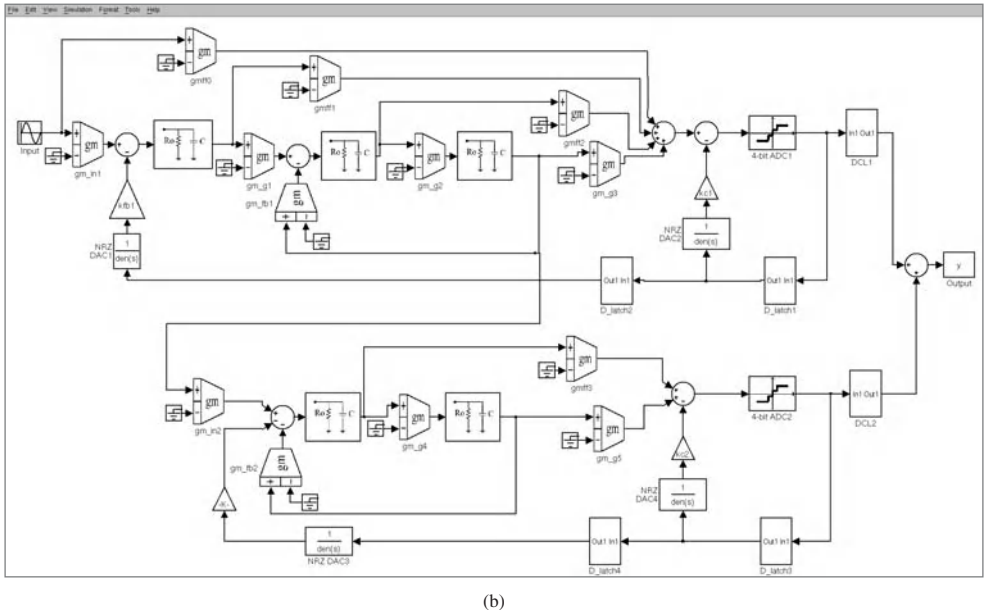
Note from Figure 3.55b that Gm-C integrators are implemented in SIMSIDES using the model described in Section 3.4.2 (Figure 3.34), that is, as the cascade of a transconductor and an output impedance circuit (output resistance and capacitance in parallel). Alternatively, Gm-C integrators can be also modeled in SIMSIDES as a single building block as shown in Figure 3.31. This approach is more suited for those ΣΔMs with a reduced number of loop-filter coefficients—commonly implemented as transconductors. In contrast, the modulator in this example has a number of feed-forward coefficients which are commonly implemented as transconductors.

Table 3.4 sums up the values of loop-filter transconductances g_i, as well as the capacitances C_i used in the modulator. These values have been found through an iterative simulation-based process that, starting from the nominal values required to place the NTF zeros, optimizes the modulator performance in terms of DR and stability within the FS range. Unit transconductors are used in the majority of loop-filter transconductors which can be tuned to keep the time constant C/g_m unchanged over C variations.

Figure 3.56a shows the ideal output spectrum of the modulator considering an −20 dBFS input tone at 1.49 MHz when clocked at $f_s = 240$ MHz. Note that shaped quantization noise presents two notches at 11.5 and 18.5 MHz which minimize the IBN in the target signal bandwidth ($B_w = 20$ MHz). The ideal effective resolution achieved by the modulator is illustrated in Figure 3.56b that depicts the SNDR versus input signal level for OSR = 6, corresponding to $f_s = 240$ MHz and $B_w = 20$ MHz. Under these conditions, the maximum effective resolution that can be achieved is approximately



(a)



(b)

Figure 3.55 Gm-C fifth-order cascade 3-2 ΣΔM with 4-bit quantization in both stages: (a) conceptual schematic and (b) SIMSIDES implementation.

Table 3.4 Loop-filter coefficients of the CT cascade 3-2 ΣΔM in Figure 3.55

Unit circuit elements
$C_u = 3.65$ pF, $g_u = 190\ \mu A\ V^{-1}$
Capacitors
$C_1 = C_2 = C_3 = C_u$, $C_4 = C_5 = 2C_u$
Feed-forward transconductances
$g_{in1} = 852\ \mu A\ V^{-1}$, $g_{ff0} = g_{ff2} = 2g_u$, $g_{ff1} = 4g_u$, $g_{in2} = g_{ff3} = 5g_u$
$g_{g1} = g_{g5} = 3g_u$, $g_{g2} = 5g_u$, $g_{g2} = g_u$, $g_{g4} = 7g_u$
Feedback transconductances
$g_{fb1} = g_{fb2} = g_u$, $k_{fb1} = 730\ \mu A\ V^{-1}$, $k_{fb2} = 6g_u$

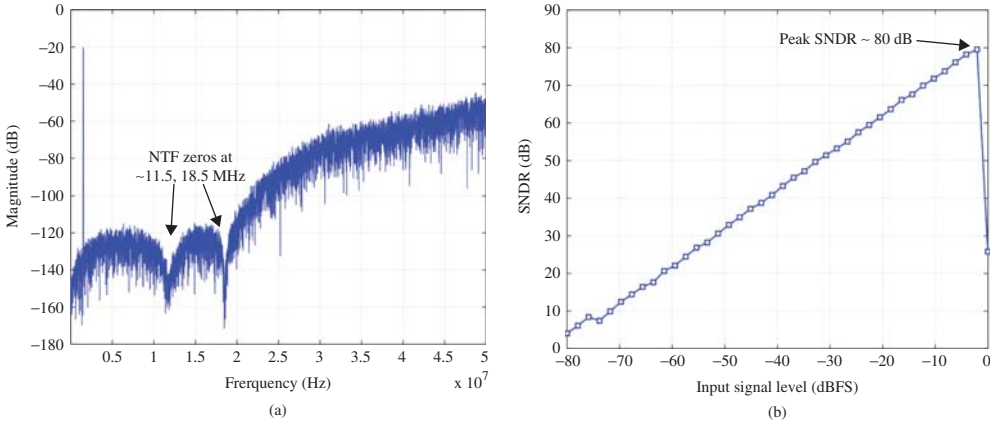


Figure 3.56 Ideal simulation results of the modulator in Figure 3.55: (a) output spectrum and (b) SNDR versus input signal level for OSR = 6 ($B_w = 20$ MHz, $f_s = 240$ MHz), considering a FS reference of 0.5 V.

13 bit. However, this performance is degraded in practice by the action of circuit nonidealities, as analyzed subsequently.

Impact of Nonideal Effects

The behavioral model used for the transconductors in SIMSIDES takes into account several nonideal circuit effects,[17] including finite DC gain, output saturation voltage, and the input-referred third-order intercept point IIP3. As an illustration, Figure 3.57 shows the effect of the finite DC gain of loop-filter transconductors on the SNDR for OSR = 6, 12, 24, when clocked at $f_s = 240$ MHz. It can be noted that as the OSR increases, the effect of this error is attenuated, as predicted by the theoretical analysis described in Section 2.2. For a signal bandwidth of $B_w = 20$ MHz (OSR = 6), a finite DC gain larger than 50 dB is required.

[17] A more detailed explanation of the nonideal circuit effects and electrical parameters included in SIMSIDES block models can be found in Appendix B.

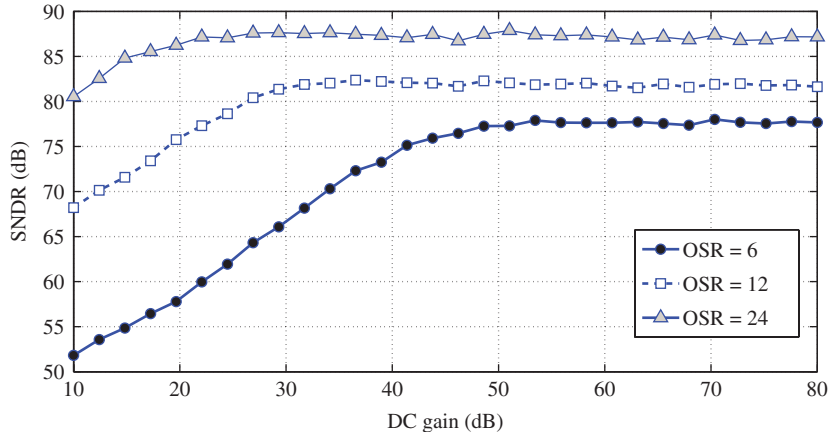


Figure 3.57 Effect of finite DC gain of loop-filter transconductors on the SNDR of Figure 3.55 for different values of OSR.

Figure 3.58 illustrates the effect of IIP3 on the modulator performance, comparing the degradation caused by different transconductors, namely, front-end transconductor (g_{in1} in Figure 3.55), input feed-forward transconductor (g_{ff0}), and the rest of loop-filter transconductors. Figure 3.58a and b illustrates the output spectrum of the modulator considering two −10 dBFS input tones located at 1.49 and 2.02 MHz. The output spectrum in Figure 3.58a assumes an IIP3 = 20 dBm for all loop-filter transconductors and that g_{in1} and g_{ff0} are ideal. In contrast, Figure 3.58b assumes that all transconductors are ideal, except for the front-end transconductor that has an IIP3 = 20 dBm. As expected, the front-end transconductor severely degrades the linearity of the modulator, causing a number of intermodulation products to appear in the signal band. The effect of the nonlinearity of the different transcondutors on the modulator resolution is better illustrated in Figure 3.58c that represents the SNDR versus IIP3 of the different types of transconductors in the modulator. Note that the effective resolution is not degraded if IIP3 > 28 dBm in the front-end transconductor. This specification is relaxed for the rest of transconductors, in which IIP3 > 5 dBm is enough to achieve the ideal SNDR.

Apart from the impact of the aforementioned nonidealities, the effect of circuit element tolerances and component mismatch are particularly critical in the design of cascade CT-ΣΔMs. The absolute tolerances can be controlled by using time-constant tuning, as is the case for this design example. However, the impact of mismatch error still remains and must be taken into account at the early design stages. Behavioral simulation using SIMSIDES can be used for this purpose. Thus, in order to evaluate the impact of this error on the performance of the modulator in Figure 3.55, maximum values of mismatch were estimated for the 130 nm CMOS process of the final implementation. The results of this analysis are shown in Figure 3.59, where the SNR is represented as a function of the standard deviation of transconductances and capacitances (σ_{gm} and σ_C, respectively). Note that for each point of this surface, a Monte Carlo analysis consisting of 150 SIMSIDES simulations was carried out. The value of SNR that is represented in this figure corresponds to that obtained by more than 90% of the simulations for each value of σ_{gm} and σ_C. Note

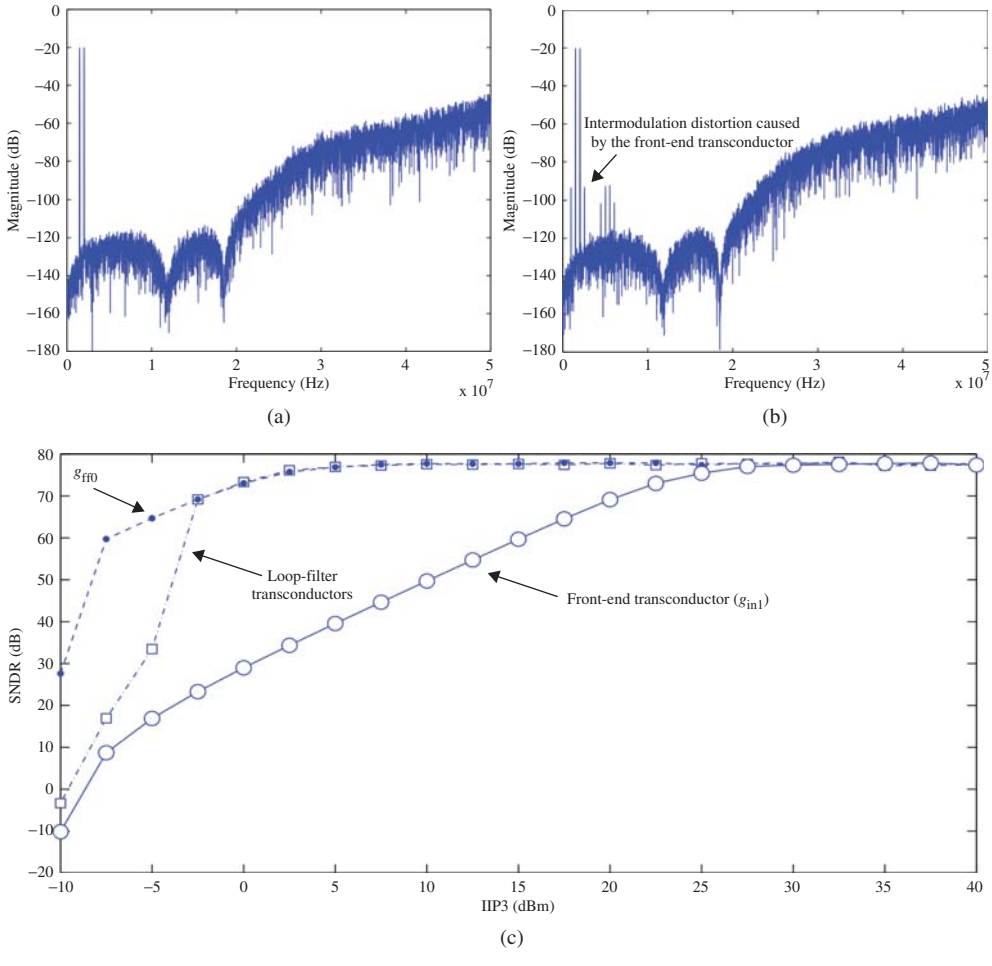


Figure 3.58 Effect of transconductance nonlinearity on the performance of the CT-ΣΔM in Figure 3.55: (a) output spectrum assuming that g_{in1} and g_{ff0} are ideal and that all loop-filter transconductors have an IIP3 = 20 dBm, (b) output spectrum considering that all transconductors are ideal, except for g_{in1} that has an IIP3 = 20 dBm, and (c) SNDR versus IIP3 for the different transconductors.

that even for the worst-case mismatch, an effective resolution higher than 12 bits can be achieved.

High-Level Synthesis and Verification

As a case study, let us consider that the modulator in Figure 3.55 is designed to fulfill the following specifications: 12 bit effective resolution within 20-MHz signal bandwidth [44]. These specifications can be mapped onto the modulator building-block specifications by following a parametric-based analysis, as described in the previous section. Alternatively,

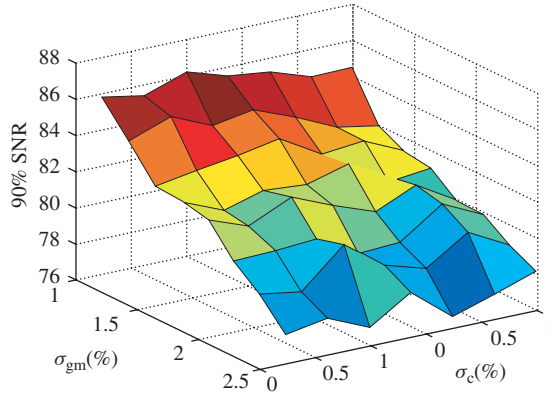


Figure 3.59 Illustrating the effect of mismatch on the SNR of the ΣΔM in Figure 3.55.

an optimization-based procedure can be followed in which an optimization engine is used for design parameter selection and a behavioral simulator (SIMSIDES in this case) is used for performance evaluation [5]. As an illustration of the latter approach, a statistical optimizer was combined with SIMSIDES for the high-level sizing of the ΣΔM in Figure 3.55.

Table 3.5 sums up the results of the sizing process, showing the critical (maximum/minimum) values of the circuit electrical performance parameters that can be tolerated to meet the required modulator performance. As stated earlier, the specifications of the front-end transconductor—specifically the finite DC gain and the third-order nonlinearity—are more demanding than the rest of transconductors. For this reason, different circuit topologies are, in practice, used for implementing the front-end transconductor and the loop-filter transconductors in Gm-C ΣΔMs, as will be discussed in Chapter 4.

Apart from the transconductor specifications, the electrical parameters of other building blocks—such as the comparators used in embedded multibit flash ADCs or the current-mode feedback DACs—are also given in Table 3.5. A description of the behavioral models of these blocks and their associated parameters can be found in Appendix B. In addition to the requirements listed in Table 3.5, the modulator building blocks must be designed so that their thermal noise contribution does not limit the modulator resolution. This is particularly critical for the front-end transconductor and feedback DAC1 in the first stage, which are connected to the modulator input. Therefore, the impact of noise sources in these blocks should be evaluated in SIMSIDES following a parametric analysis similar to that shown in Section 3.6.1.

Figure 3.60 shows the SNDR curve of the modulator in Figure 3.55b when all nonideal effects are taken into account. It can be noted that the peak SNDR is 76.8 dB (12.5 bit), which meets target specifications. Therefore, once the performance of the modulator has been validated by behavioral simulation considering the impact of the main circuit error mechanisms, the resulting building-block electrical parameters in Table 3.5 can be used as initial design specifications for the modulator subcircuits at transistor level. Their implications in the transistor-level design and their corresponding design trade-offs will be discussed in Chapter 4.

Table 3.5 High-level sizing of the CT cascade ΣΔM in Figure 3.55

Front-End Transconductor	
Finite DC gain	$\geq$ 70 dB
Linear input swing	$\leq$ 0.3 V
Linear output swing	$\leq$ 0.3 V
Third-order nonlinear coefficient	≤ -86 dB
Loop-Filter Transconductors	
Finite DC gain	$\geq$ 50 dB
Linear input swing	$\leq$ 0.3 V
Linear output swing	$\leq$ 0.3 V
Third-order nonlinear coefficient	≤ -56 dB
Multibit Embedded ADCs	
Comparator offset	$\leq$ 20 mV
Comparator hysteresis	$\leq$ 20 mV
Comparator resolution time	$\leq$ 1 ns
Feedback DACs	
Unit current standard deviation	$\leq$ 0.15%LSB
Finite output resistance	$\geq$ 12MΩ
Settling time	$\leq$ 0.5 ns

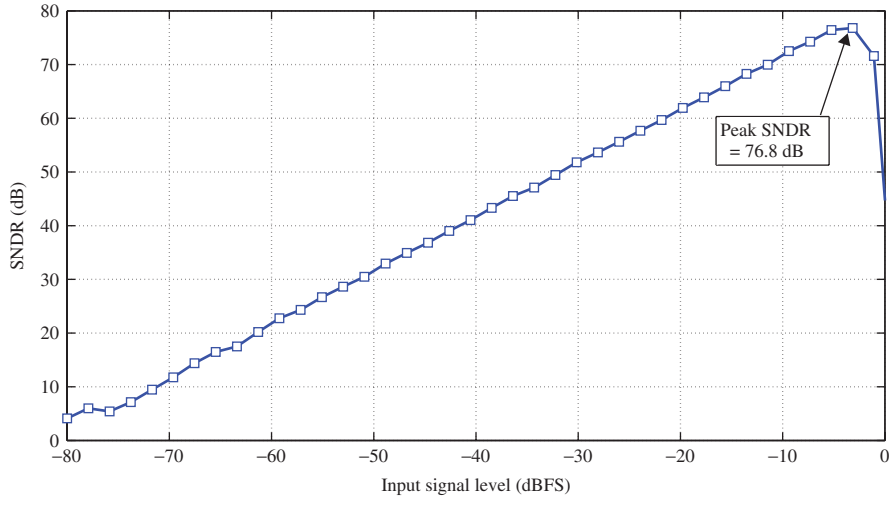


Figure 3.60 SNDR versus input signal level considering all nonideal effects in Figure 3.55.

3.7 Summary

This chapter discussed the use of behavioral modeling and simulation techniques for the high-level analysis and synthesis of ΣΔMs. After examining different approaches and alternatives to the transistor-level electrical simulation of ΣΔMs, it has been demonstrated that using time-domain behavioral models implemented with MATLAB C-MEX

S-functions is a very efficient technique in terms of accuracy, CPU simulation time, and flexibility to incorporate new circuit effects and building-block models. A step-by-step procedure to implement precise ΣΔM behavioral models using this technique has been described in detail and applied to the fundamental ΣΔM building blocks, namely, integrators, quantizers, and embedded DACs. On the basis of this modeling approach, a SIMULINK-based time-domain behavioral simulator for ΣΔMs named SIMSIDES has been described and some case studies have been presented to illustrate the use of this simulator in the high-level sizing and verification of ΣΔMs. The results obtained from this process constitute the starting point for the electrical (transistor-level) design and verification of ΣΔMs described in Chapter 4.

References

[1] G. Gielen and J. Franca, "CAD Tools for Data Converter Design: An Overview," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. **43**, pp. 77–89, February 1996.

[2] V. F. Dias *et al*., "Design Tools for Oversampling Data Converters: Needs and Solutions," *Microelectronics Journal*, vol. **23**, pp. 641–650, 1992.

[3] F. Medeiro *et al*., "A Vertically Integrated Tool for Automated Design of ΣΔ Modulators," *IEEE Journal of Solid-State Circuits*, vol. **30**, pp. 762–772, July 1995.

[4] K. Francken *et al*., "A High-Level Simulation and Synthesis Environment for Delta-Sigma Modulators," *IEEE Transactions Computer-Aided Design of Integrated Circuits Systems*, vol. **22**, pp. 1049–1061, August 2003.

[5] J. Ruiz-Amaya *et al*., "High-Level Synthesis of Switched-Capacitor, Switched-Current and Continuous-Time ΣΔ Modulators Using SIMULINK-based Time-Domain Behavioral Models," *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. **51**, pp. 1795–1810, September 2005.

[6] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*, Wiley-IEEE Press, 2005.

[7] R. Schreier, The Delta-Sigma Toolbox v. 7.3. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/19., 2009.

[8] J. M. de la Rosa *et al*., "A CMOS 110-dB@40-kS/s Programmable-Gain Chopper-Stabilized Third-Order 2-1 Cascade Sigma-Delta Modulator for Low-Power High-Linearity Automotive Sensor ASICs," *IEEE Journal of Solid-State Circuits*, vol. **40**, pp. 2246–2264, November 2005.

[9] S. Pavan, "Efficient Simulation of Weak Nonlinearities in Continuous Time Oversampling Converters," *IEEE Transactions on Circuits and Systems I—Regular Papers*, vol. **57**, pp. 1925–1934, August 2010.

[10] J. Ryckaert, J. Borremans, B. Verbruggen, L. Bos, C. Armiento, J. Craninckx, and G. van der Plas, "A 2.4 GHz Low-Power Sixth-Order RF Bandpass ΔΣ Converter in CMOS," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 2873–2880, November 2009.

[11] F. V. Fernandez *et al*., "Design Methodologies for Sigma-Delta Converters," *Chapter 15 in CMOS Telecom Data Converters* (A. Rodríguez-Vázquez, F. Medeiro, and E. Janssens, Editors), Kluwer Academic Publishers, 2003.

[12] F. Medeiro, B. Pérez-Verdú, and A. Rodríguez-Vázquez, *Top-Down Design of High-Performance Sigma-Delta Modulators*, Kluwer Academic Publishers, 1999.

[13] C. H. Wolff and L. Carley, "Simulation of Δ−Σ Modulators Using Behavioral Models," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 376–379, May 1990.

[14] V. Liberali *et al*., "TOSCA: A Simulator for Switched-Capacitor Noise-Shaping A/D Converters," *IEEE Transactions Computer-Aided Design of Integrated Circuits Systems*, vol. **12**, pp. 1376–1386, September 1993.

[15] P. Malcovati *et al*., "Behavioral Modeling of Switched-Capacitor Sigma-Delta Modulators," *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. **50**, pp. 352–364, March 2003.

[16] G. Gielen *et al*., "An Analytical Integration Method for the Simulation of Continuous-Time ΔΣ Modulators," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. **23**, pp. 389–399, March 2004.

[17] H. Zare-Hoseini, I. Kale, and O. Shoaei, "Modeling of Switched-Capacitor Delta-Sigma Modulators in SIMULINK," *IEEE Transactions on Instrumentation and Measurement*, vol. **54**, pp. 1646–1654, August 2006.

[18] M. Keller *et al*., "A Method for the Discrete-Time Simulation of Continuous-Time Sigma-Delta Modulators," *Proc. of the IEEE Intl. Symposium on Circuits and Systems*, pp. 241–244, May 2007.
[19] G. Suárez, M. Jiménez, and F. O. Fernández, "Behavioral Modeling Methods for Switched-Capacitor ΣΔ Modulators," *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. **54**, pp. 1236–1244, June 2007.
[20] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd edn, Prentice Hall, 2009.
[21] Synopsys, *HSPICE Simulation and Analysis User Guide*, Synopsys Inc., 2006.
[22] Cadence, *Spectre Circuit Simulator User Guide*, Cadence Design Systems Inc., 2002.
[23] R. Bishop *et al*., "Table-based Simulation of Delta-Sigma Modulators," *IEEE Transactions on Circuits and Systems—I*, vol. **37**, pp. 447–451, March 1990.
[24] G. Brauns *et al*., "Table-based Modeling of Delta-Sigma Modulators using ZSIM," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. **9**, pp. 142–150, February 1990.
[25] IEEE-Standards, "IEEE VHDL Language Reference Manual," IEEE Std 1076-2002, 2002.
[26] IEEE-Standards, "IEEE VHDL 1076.1 Language Reference Manual," IEEE Std 1076.1-1999, 2002.
[27] Cadence, *Cadence Design Framework II*, Cadence Design Systems Inc., Available at: http://www.cadence.com., 2010.
[28] Mathworks, *Using MATLAB Version 6,* The Mathworks Inc., 2002.
[29] Mathworks, *Using SIMULINK Version 5,* The Mathworks Inc., 2002.
[30] Mathworks, *Stateflow 7.7,* The Mathworks Inc., 2002.
[31] R. del Río, F. Medeiro, B. Pérez-Verdú, J. M. de la Rosa, and A. Rodríguez-Vázquez, *CMOS Cascade ΣΔ Modulators for Sensors and Telecom: Error Analysis and Practical Design*. Springer, 2006.
[32] S. Brigati *et al*., "Modeling Sigma-Delta Modulator Nonidealities in SIMULINK," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 2384–2387, May 1999.
[33] S. Brigati, SD Toolbox. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/2460., 2002.
[34] Mathworks, *Writing S-Functions Version 5,* The Mathworks Inc., 2002.
[35] W. Press *et al*., *Numerical Recipes in C. The Art of Scientific Computing*, 2nd edn, Cambridge University Press, 1992.
[36] B. Razavi, *Principles of Data Conversion System Design*, IEEE Press, 1995.
[37] W. Yu, S. Sen, and B. Leung, "Distortion analysis of MOS track-and-hold sampling mixers using time-varying Volterra series," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. **46**, pp. 101–113, February 1999.
[38] R. del Río, F. Medeiro, J. M. de la Rosa, B. Pérez-Verdú, and A. Rodríguez-Vázquez, "Reliable analysis of settling errors in SC integrators: application to ΣΔ modulators," *IET Electronics Letters*, vol. **36**, pp. 503–504, March 2000.
[39] M. Ortmanns and F. Gerfers, *Continuous-Time Sigma-Delta A/D Conversion: Fundamentals, Performance Limits and Robust Implementations*, Springer, 2006.
[40] J. Cherry and W. Snelgrove, "Excess Loop Delay in Continuous-Time Delta–Sigma Modulators," *IEEE Transaction on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. **46**, pp. 376–389, April 1999.
[41] J. M. de la Rosa, B. Pérez-Verdú, and A. Rodríguez-Vázquez, *Systematic Design of CMOS Switched-Current Bandpass Sigma-Delta Modulators for Digital Communication Chips*, Kluwer Academic Publishers, 2002.
[42] R. Tortosa, A. Aceituno, J. M. de la Rosa, A. Rodríguez-Vázquez, and F. V. Fernández, "A 12-bit, 40MS/s Gm-C Cascade 3-2 Continuous-Time Sigma-Delta Modulator," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 1–4, 2007.
[43] S. Yan and E. Sánchez-Sinencio, "A Continuous-Time ΣΔ Modulator With 88-dB Dynamic Range and 1.1-MHz Signal Bandwidth," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 75–86, January 2004.
[44] R. Tortosa, J. M. de la Rosa, F. V. Fernández, and A. Rodríguez-Vázquez, "A New High-Level Synthesis Methodology of Cascaded Continuous-Time ΣΔ Modulators," *IEEE Transactions on Circuits and Systems—II: Express Briefs*, vol. **53**, pp. 739–743, August 2006.

4

Circuit-Level Design, Implementation, and Verification

The behavioral modeling and simulation techniques described in Chapter 3 can be used for the high-level synthesis and verification of ΣΔMs so that the modulator-level specifications are efficiently mapped onto building-block (circuit-level) specifications. Thus, at this stage of the design cycle, the modulator is still modeled at system level, but the electrical performance parameters of all ΣΔM circuit elements (switches, capacitors, amplifiers, transconductors, comparators, etc.) have been already derived from the high-level sizing process. Those parameters are in turn the circuit-level specifications, which constitute the start point for the electrical (transistor-level) and physical design process of the modulator. This process—conceptually illustrated in Figure 4.1—comprises a number of successive steps in which the initial behavioral-model diagram of the modulator is transformed into a circuit schematic—initially implemented with macromodels and finally with transistors—afterward into a layout, and finally into a chip implementation for experimental verification in a laboratory.

This chapter gives some design issues and practical recipes to complete the design flow illustrated in Figure 4.1. Section 4.1 deals with macromodel implementation of ΣΔMs as an essential design stage to relate behavioral-level models with circuit-level description. Section 4.2 describes how to include circuit noise in electrical-level simulations of ΣΔMs and Section 4.3 shows how to process the modulator output data extracted from electrical simulations in SPICE-like simulators, in order to characterize the performance of ΣΔMs. Section 4.4 moves down to the transistor-level implementation, giving a number of practical design guidelines and describing diverse simulation test benches to properly design and characterize the performance of basic ΣΔM building blocks. Other auxiliary circuits needed to implement ΣΔMs are discussed in Section 4.5. Finally, Sections 4.6 and 4.7 deal with some of the most important design issues related to the layout, prototyping, and testing of high-performance ΣΔMs.

4.1 Macromodeling ΣΔMs

The transformation from a behavioral-level description into a circuit schematic as conceptually depicted in Figure 4.1 is carried out in several steps. Thus, at the early stages of the design cycle, hardware description languages (HDL), such as Verilog-A [1], and

CMOS Sigma-Delta Converters: Practical Design Guide, First Edition. José M. de la Rosa and Rocío del Río.

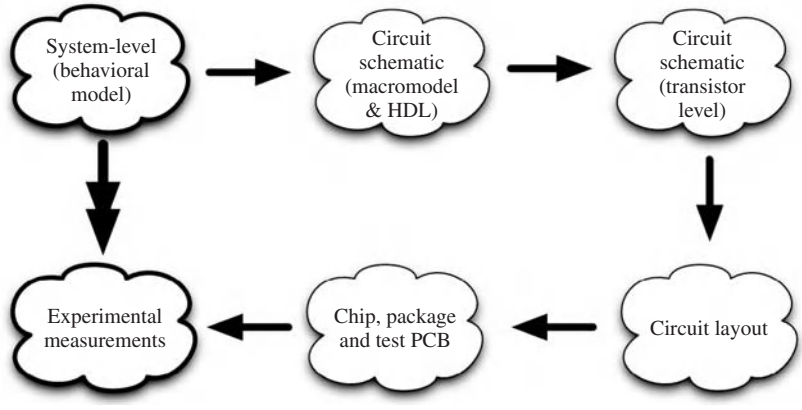


Figure 4.1 Conceptual step-by-step design flow of ΣΔMs.

macromodels are used for representing different modulator building blocks. These models include main circuit error limitations derived from system-level behavioral models, for instance implemented in SIMSIDES. These models are progressively replaced by transistor-level implementations as the different ΣΔM blocks are designed. This way, the performance of the modulator is analyzed and checked at different stages of the design cycle by combining the impact of those subcircuits which have been designed at the transistor level with those ones which have not been sized yet.

This section explains how to use macromodels to implement ΣΔMs at the circuit level in electrical simulators. Most important building-block equivalent circuits are derived and some examples are shown to illustrate their use.

4.1.1 SC Integrator Macromodel

Figure 4.2 shows a single-ended equivalent circuit frequently used for simulating FE SC integrators with macromodels. The corresponding macromodel for a fully-differential implementation is shown in Figure 4.3 [2]. Both equivalent circuits use ideal capacitors, whereas the switches and the OTA include nonideal circuit effects as discussed subsequently.

Switch Macromodel

As illustrated in Figure 4.2, switches are usually modeled as a linear switch on-resistance R_{on} in series with an ideal switch, which is controlled by the corresponding clock phase, either ϕ_1 or ϕ_2. Note that the switch model in SPICE-like simulators consists of a highly nonlinear resistor whose resistance depends on the control clock-phase voltage v_ϕ as follows:

$$R_{switch} = \begin{cases} R_{on} & v_\phi \geq v_{TS} \\ R_{off} & v_\phi < v_{TS} \end{cases} \tag{4.1}$$

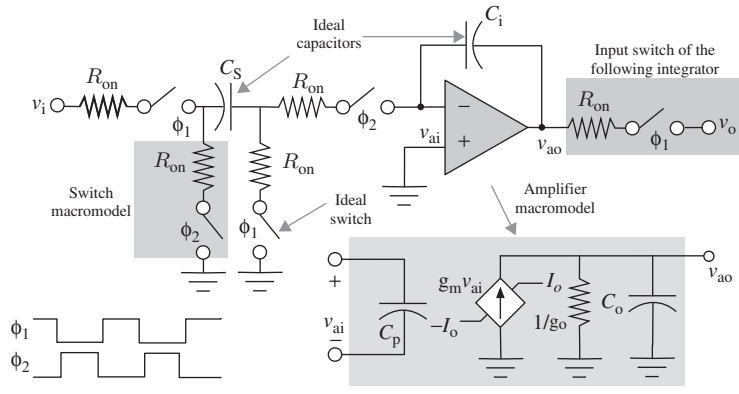


Figure 4.2 Macromodel of a single-ended FE SC integrator.

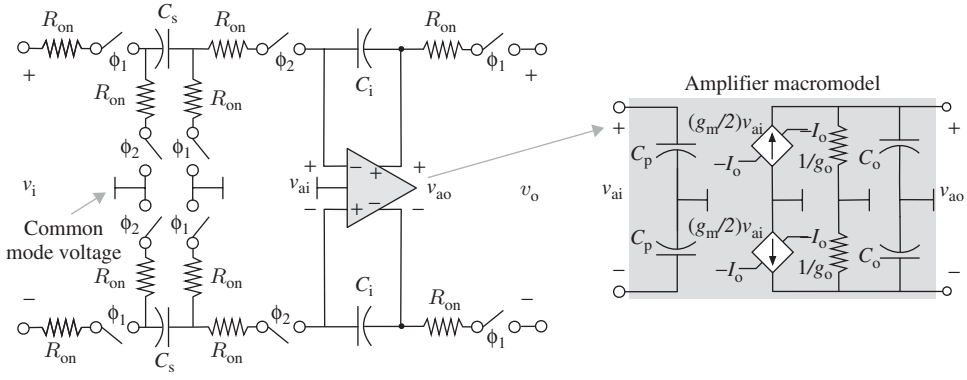


Figure 4.3 Macromodel of a fully-differential FE SC integrator.

where R_{off} stands for the switch off-resistance that is ideally $R_{off} \to \infty$, and v_{TS} stands for a threshold voltage that determines if the switch is either closed (on) or open (off). In practice, an almost ideal switch can be modeled in electrical simulators by setting R_{off} and R_{on} just high/low enough to be negligible with respect to other circuit elements [3]. For instance, typical values of R_{on} are chosen to be in the order of μΩ, while a very high value of R_{off} is considered, typically in the order of GΩ.[1]

Instead of using this simple model, a macromodel that is closer to the transistor-level topology implementation can be used. For instance, let us consider a CMOS switch similar to those shown in the sampling circuit of Figure 2.21. This switch is made up of a pMOS switch and an nMOS switch connected in parallel. Figure 4.4 shows an equivalent circuit that takes into account the on-resistance of both MOST switches as well as their associated parasitic capacitances, denoted as C_{ps}. Note that this macromodel keeps the symmetry of the original MOST-based circuit.

[1] Note that quasi ideal values of R_{off} and R_{on} can cause tolerance and convergence problems in electrical simulations. This can be controlled by properly setting the corresponding numerical tolerance parameters in the simulation options [4].

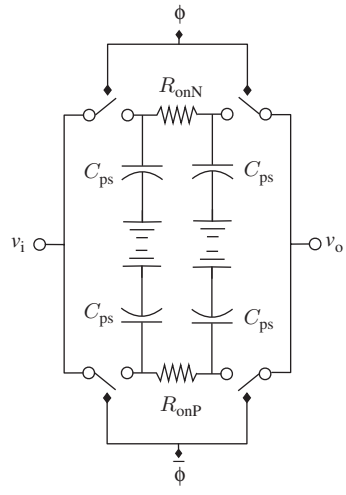


Figure 4.4 Macromodel of a CMOS switch.

OTA Macromodel

The OTA circuit is modeled by a well-known single-pole amplifier where g_m, g_o, and C_o denote, respectively, the transconductance, output conductance, and output capacitance. In addition, the voltage-controlled current source used for modeling the OTA transconductance has two saturation limits, $(-I_o, +I_o)$. These limits model the minimum and maximum output currents provided by the OTA. This way, this equivalent circuit takes into account the finite DC gain, GB, and SR limitations of the OTA, respectively given by

$$A_v \equiv \frac{g_m}{g_o} \quad , \quad \text{GB} \equiv \frac{g_m}{C_o} \quad , \quad \text{SR} \equiv \frac{I_o}{C_o} \tag{4.2}$$

Note that real values of GB and SR deviate in practice from the above expressions because of the effect caused by bottom-plate parasitic capacitances, denoted as C_p in Figure 4.3, as well as the capacitive load due to the SC network connected at the output of the integrator.

4.1.2 CT Integrator Macromodel

The equivalent circuit normally used for the macromodel of CT integrators is based on a one-pole (or two-pole) OTA model together with the additional circuit elements that are required to implement the integrator topology, that is, Gm-C, active-RC, etc. [5].

Active-RC Integrators

Figure 4.5 shows a simple macromodel circuit commonly used for active-RC implementations. In this example, a one-pole OTA model similar to that shown in Figures 4.2 and 4.3

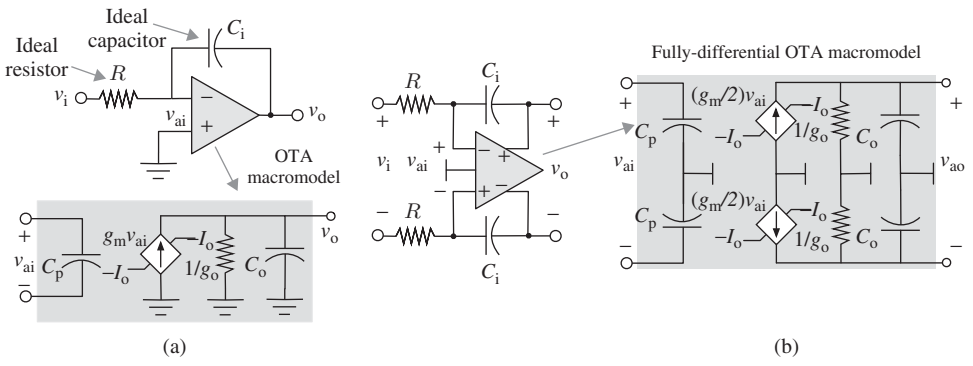


Figure 4.5 Active-RC integrator macromodel: (a) single-ended schematic and (b) fully-differential schematic.

is used, although higher dynamics can be implemented if necessary. In a similar way to SC integrators, parasitic and load capacitances of the OTA can also be considered.

Gm-C Integrators

On the basis of the same OTA macromodel circuit, Gm-C integrators can be modeled by the equivalent circuit shown in Figure 4.6a, which includes a second pole modeled by the time constant R_pC_p. In this case, the S-domain transfer function of the integrator is given by

$$\frac{v_o(s)}{v_i(s)} = \frac{-g_m/g_o}{(1 + sC/g_o)(1 + sR_pC_p)} \tag{4.3}$$

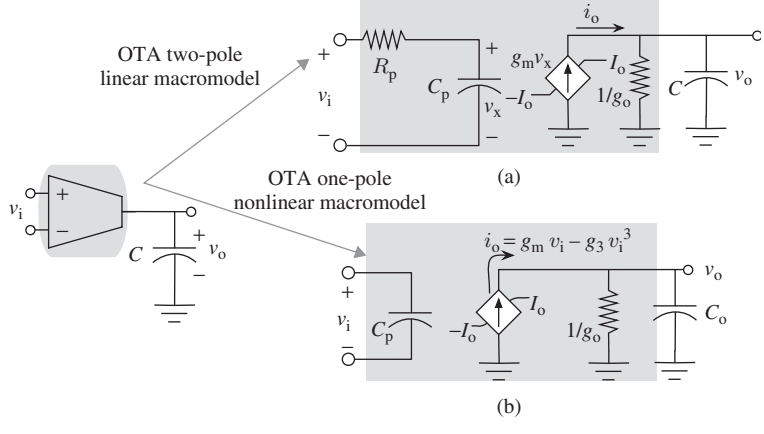


Figure 4.6 Gm-C integrator macromodel: (a) two-pole linear model and (b) one-pole weakly nonlinear model.

4.1.3 Nonlinear OTA Transconductor

The OTA macromodel considered in previous sections assumed a linear model for the transconductance g_m and an output current saturation I_o. In practice, the transconductor of the OTA is assumed to be weakly nonlinear, such that its static output current i_o is related to the OTA input voltage v_i as follows [6]:

$$i_o = \begin{cases} g_m v_i - g_3 v_i^3 & |v_i| \leq \frac{IIP3}{2} \\ \text{sgn}(v_i) I_o & |v_i| > \frac{IIP3}{2} \end{cases} \tag{4.4}$$

where $\text{sgn}(x)$ denotes the sign function of x, g_3 is the third-order nonlinear coefficient of the transconductor, IIP3 stands for the input-referred third-order intercept point, and I_o is the maximum output current provided by the transconductor. It can be shown from Equation 4.4 that IIP3 and I_o are related to g_m and g_3 as follows:

$$\text{IIP3} = 2\sqrt{\frac{g_m}{3g_3}} \tag{4.5}$$

$$I_o = \frac{2g_m}{3}\sqrt{\frac{g_m}{3g_3}} \tag{4.6}$$

Figure 4.6b shows an equivalent circuit for the OTA macromodel that includes the weakly nonlinear transconductance element. Note that this equivalent circuit can be easily modeled in SPICE-like simulators using nonlinear voltage-controlled current sources [3]. Moreover, other sources of nonlinearities associated with the voltage dependency of integrated resistors and capacitors can also be included in the integrator macromodels using signal-dependent voltage/current sources. However, this kind of nonlinear effects are more accurately considered in the simulation when the circuit is modeled at device level, that is, taking into account the device models provided by the foundry, such as unsalicided polysilicon resistors or MiM capacitors.

4.1.4 Embedded Flash ADC Macromodel

Multibit/multilevel ADCs required to implement the quantizers embedded in ΣΔMs are typically implemented using a well-known flash architecture. This kind of ADCs is made up of a bank of comparators that compare the quantizer input signal (i.e., the output of the integrator connected to the quantizer) with a set of reference voltages generated in a resistor ladder [7]. These reference voltages correspond to the transition points between the different adjacent quantization intervals [8].

As an illustration, Figure 4.7a shows the fully-differential schematic of a trilevel flash ADC, which is made up of two comparators and a resistive divider. The latter is made up of four unit resistors R_u, which are connected in series between the negative reference voltage $V_{ref-} = -v_{ref}$ and the positive reference voltage $V_{ref+} = v_{ref}$. In this example, there are two transition points, $-v_{ref}$ and $+v_{ref}$, in the quantizer differential characteristic illustrated in Figure 4.7b, which correspond to the differential tabs of the resistor ladder in Figure 4.7a. Thus, the quantizer input $v_i \equiv v_{i+} - v_{i-}$ is compared with $V_{R+} - V_{R-}$ and $V_{R-} - V_{R+}$, in the first and the second comparator, respectively. The thermometer code

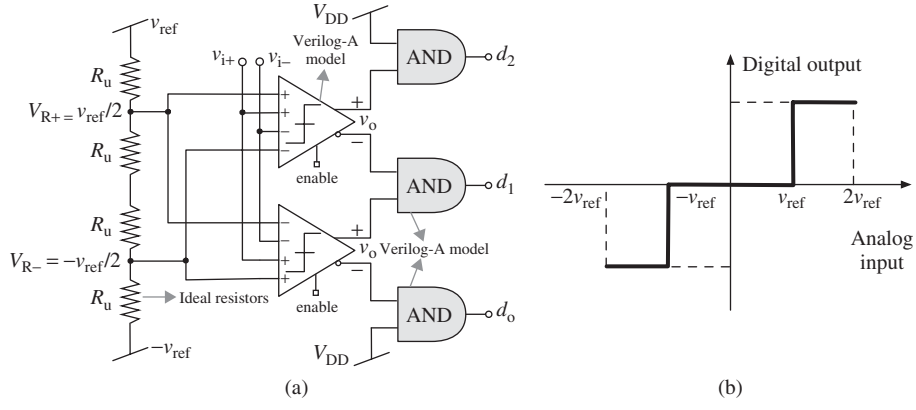


Figure 4.7 Macromodeling multilevel flash ADCs: (a) conceptual schematic of a trilevel flash ADC and (b) static differential transfer characteristic.

generated at the comparators outputs is converted into a 1-of-3 code (d_{0-2}) using three AND gates.

The circuit shown in Figure 4.7a can be used for simulating ΣΔMs with macromodels. To this end, resistors are considered ideal circuit elements,[2] while comparators and AND gates are modeled using Verilog-A. As an illustration, Figure 4.8 shows the Verilog-A code used for modeling the comparators in Figure 4.7. The model—based on the Open Verilog International (OVI) language reference manual [10, 11]—is quite simple and includes the input offset. The comparison only takes place when the clock signal named `enable` is in the high state. The static input–output transfer characteristic is computed by using an hyperbolic tangent function (`tanh`), which is scaled by a parameter named `comp_slope`. The latter determines the static resolution of the comparator by modifying the voltage gain around the input offset voltage, modeled by `sigin_offset` parameter.

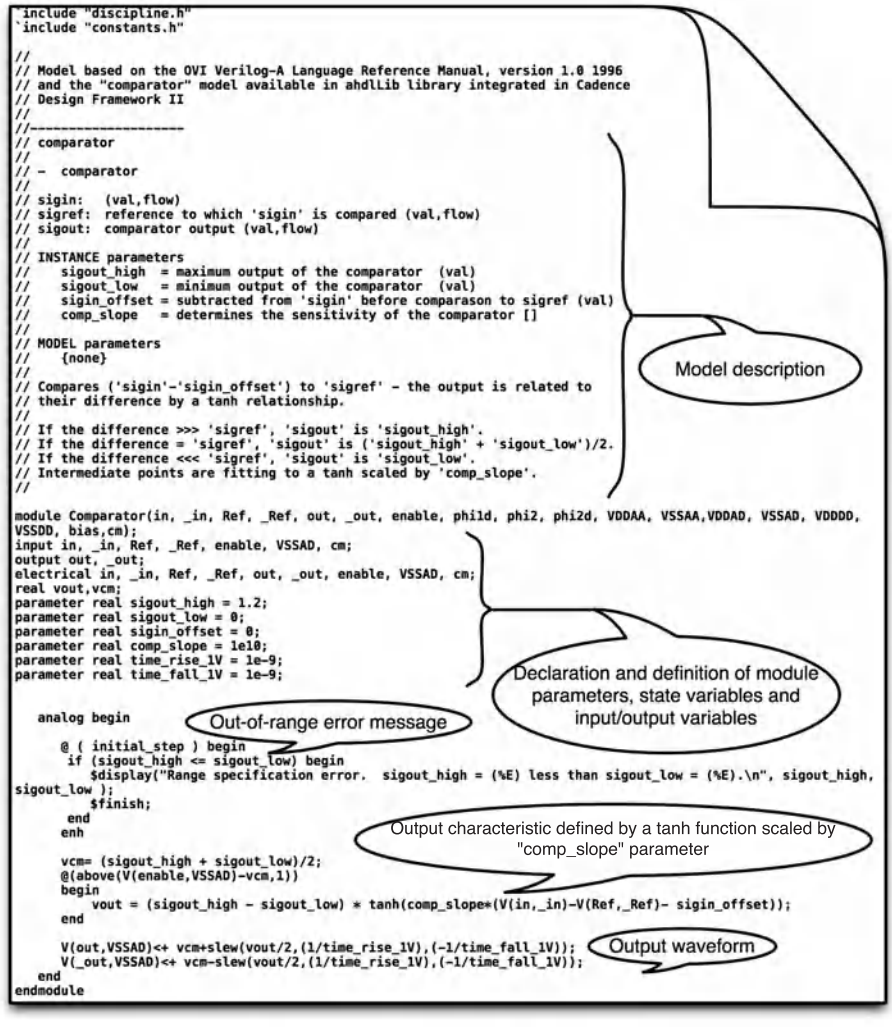
4.1.5 Feedback DAC Macromodel

There are essentially two types of feedback DAC circuits used in ΣΔMs, namely SC DACs and current mode, also named *current-steering* (CS) DACs [12]. The former are mainly used in SC-ΣΔMs, although they are used also in some CT-ΣΔMs, specifically those implemented with active-RC integrators intended for low-frequency applications [13]. In contrast, *switched-current*[3] or current-steering feedback DACs are commonly used in wideband CT-ΣΔMs, particularly—but not only—those based on Gm-C loop-filter implementations.

As an example of SC DACs, let us consider again the trilevel ADC shown in Figure 4.7a. The 1-of-3 coded digital output (d_0, d_1, d_2) is fed back to the ΣΔM loop-filter SC integrators by using three AND gates as illustrated in Figure 4.9. In this

[2] These resistors are commonly implemented in practice using unsalicided $p+$ poly layers [9].

[3] Current-mode feedback DACs are also used in SI-ΣΔMs, which are a particular case of DT-ΣΔMs where the loop filter processes current-mode signals, and it is implemented with SI integrators instead of SC integrators [14]. The difference between switched-current cells and current-steering DACs will be discussed later on in this chapter.




```
`include "discipline.h"
`include "constants.h"

//
// Model based on the OVI Verilog-A Language Reference Manual, version 1.0 1996
// and the "comparator" model available in ahdlLib library integrated in Cadence
// Design Framework II
//
//--------------------
// comparator
//
// -  comparator
//
// sigin:   (val,flow)
// sigref:  reference to which 'sigin' is compared (val,flow)
// sigout:  comparator output (val,flow)
//
// INSTANCE parameters
//    sigout_high  = maximum output of the comparator  (val)
//    sigout_low   = minimum output of the comparator  (val)
//    sigin_offset = subtracted from 'sigin' before comparason to sigref (val)
//    comp_slope   = determines the sensitivity of the comparator []
//
// MODEL parameters
//    {none}
//
// Compares ('sigin'-'sigin_offset') to 'sigref' - the output is related to
// their difference by a tanh relationship.
//
// If the difference >>> 'sigref', 'sigout' is 'sigout_high'.
// If the difference = 'sigref', 'sigout' is ('sigout_high' + 'sigout_low')/2.
// If the difference <<< 'sigref', 'sigout' is 'sigout_low'.
// Intermediate points are fitting to a tanh scaled by 'comp_slope'.
//

module Comparator(in, _in, Ref, _Ref, out, _out, enable, phi1d, phi2, phi2d, VDDAA, VSSAA,VDDAD, VSSAD, VDDDD,
VSSDD, bias,cm);
input in, _in, Ref, _Ref, enable, VSSAD, cm;
output out, _out;
electrical in, _in, Ref, _Ref, out, _out, enable, VSSAD, cm;
real vout,vcm;
parameter real sigout_high = 1.2;
parameter real sigout_low = 0;
parameter real sigin_offset = 0;
parameter real comp_slope = 1e10;
parameter real time_rise_1V = 1e-9;
parameter real time_fall_1V = 1e-9;


   analog begin

      @ ( initial_step ) begin
       if (sigout_high <= sigout_low) begin
          $display("Range specification error.  sigout_high = (%E) less than sigout_low = (%E).\n", sigout_high,
sigout_low );
          $finish;
       end
      enh

      vcm= (sigout_high + sigout_low)/2;
      @(above(V(enable,VSSAD)-vcm,1))
      begin
          vout = (sigout_high - sigout_low) * tanh(comp_slope*(V(in,_in)-V(Ref,_Ref)- sigin_offset));
      end

      V(out,VSSAD)<+ vcm+slew(vout/2,(1/time_rise_1V),(-1/time_fall_1V));
      V(_out,VSSAD)<+ vcm-slew(vout/2,(1/time_rise_1V),(-1/time_fall_1V));
   end
endmodule
```

Figure 4.8 Verilog-A code used for simulating comparators in quantizer macromodels [10, 11].

case, the macromodel of the DAC is simply based on the Verilog-A models of the AND gates as well as the macromodel circuits used for the switches described in Section 4.1.1.

Following the same philosophy, a macromodel circuit that can be used for current-steering DACs is based on simple macromodels of current sources and switches.[4] As an illustration, Figure 4.10 shows the macromodel of a fully-differential trilevel current-steering NRZ DAC. It consists of a set of current sources controlled, through switches, by a 1-of-3 coded digital data (d_i). As will be discussed later in this chapter, the ideal

[4] Alternatively, a more ideal macromodel can be simply based on voltage-controlled current sources as will be illustrated later in this chapter.

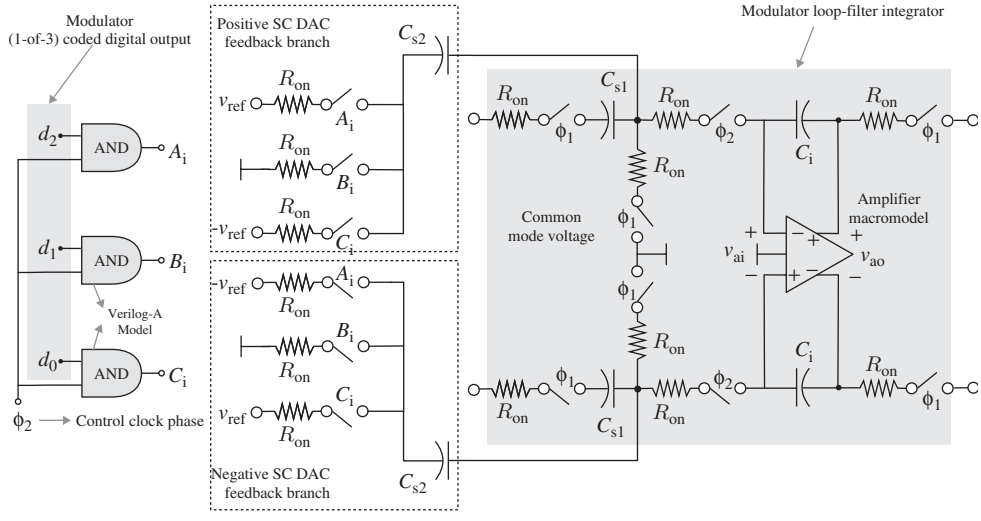


Figure 4.9 Macromodel of a trilevel SC DAC connected to an SC FE integrator within a ΣΔM loop filter. Note that different SC branches, using two different sampling capacitors C_{s1} and C_{s2}, are used in this model. However, there are many situations in practice where a single SC branch is shared by both the input signal and the feedback DAC as will be shown later in this chapter.

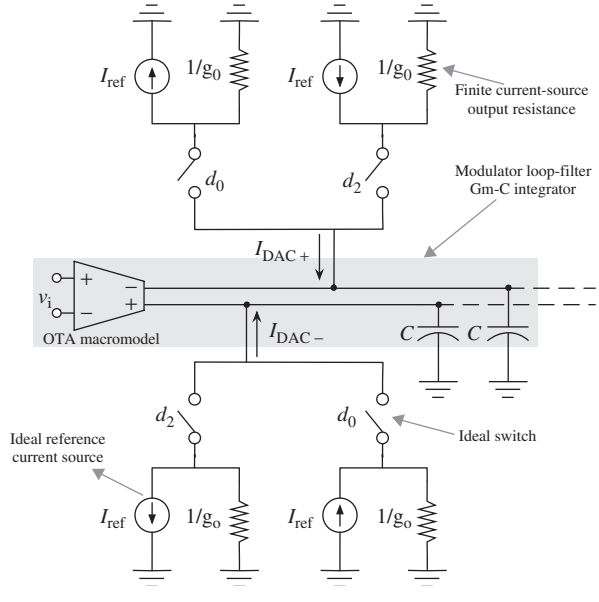


Figure 4.10 Illustrating the macromodel of a trilevel current-steering DAC connected to a Gm-C integrator within a ΣΔM loop filter.

operation of the current sources, and consequently the current-steering DAC, is degraded in practice by mismatch errors, finite output impedance (of the current sources), thermal gradients, etc. The majority of these errors, such as mismatch and technology-related errors, require doing a large number of simulations in order to evaluate their impact on the performance of the modulator. For that reason, these errors are usually considered at system-level behavioral models, for instance implemented in MATLAB/SIMULINK as described in Chapter 3. Some other nonidealities, such as output impedance of current sources, which also affect the dynamic operation of the modulator loop filter, can be easily macromodeled by simply including an output resistance in parallel with each unit (or reference) current source, as illustrated in Figure 4.10.

4.1.6 Examples of ΣΔM Macromodels

To conclude this section, a couple of examples are described to illustrate the use of the macromodels for the implementation of ΣΔMs. The first one is based on SC circuits while the second one is implemented with active-RC circuits. In both cases, the well-known, single-loop, second-order ΣΔM is used as a demonstration vehicle, considering an embedded trilevel quantizer.

SC Second-Order Example

Figure 4.11 shows the conceptual schematic of the second-order SC-ΣΔM under study, which includes a trilevel embedded quantizer. The values of capacitors used are highlighted in the figure. The circuit elements forming this modulator—that is switches, amplifiers, a flash (trilevel) ADC, and an SC (trilevel) DAC—can be implemented using the macromodels described in previous section.

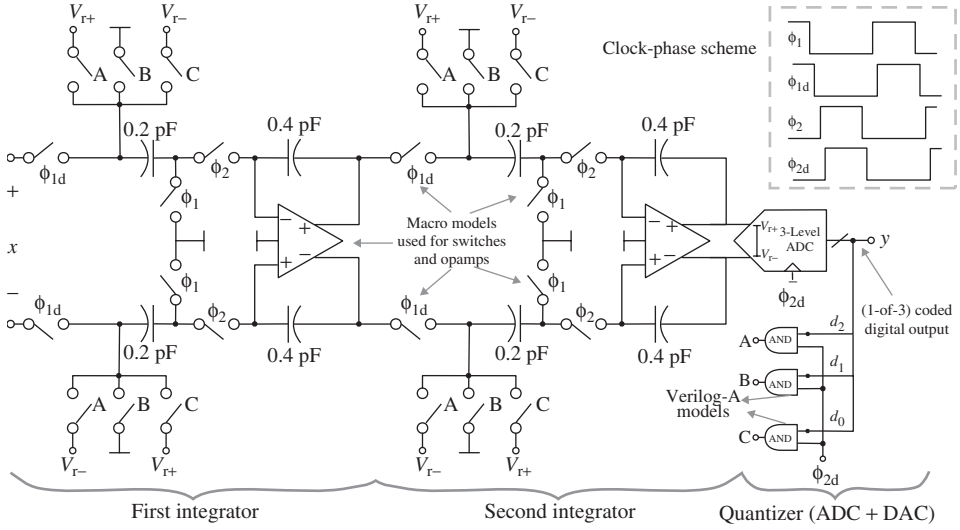


Figure 4.11 Conceptual schematic of a second-order SC-ΣΔM with a trilevel embedded quantizer.

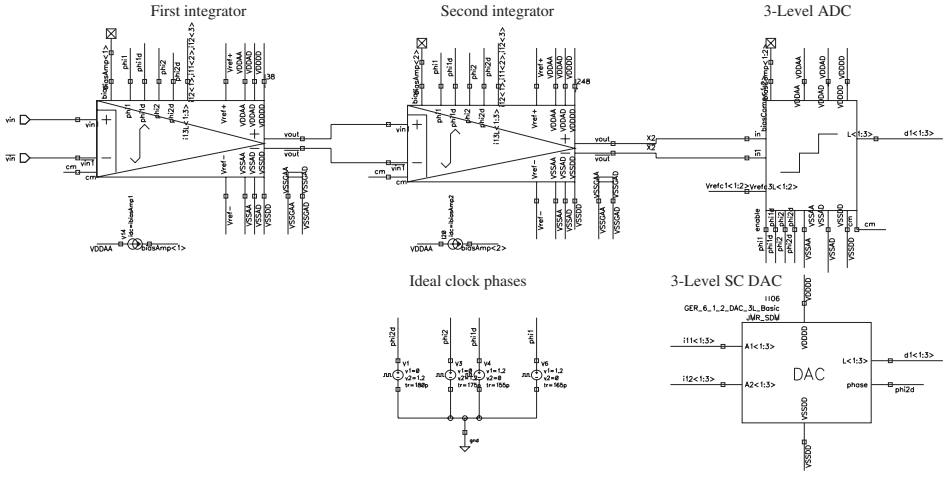


Figure 4.12 Implementation of the modulator in Figure 4.11 using Cadence Virtuoso Schematics.

Figure 4.12 shows the schematic of the modulator in Figure 4.11 implemented in Cadence Design FrameWork, using Cadence Virtuoso Schematic. In this example, ideal clock phases (implemented with ideal voltage sources) are used for the sake of simplicity. This is a common practice at the very beginning of the design phase as the use of a clock-phase generator circuit—discussed later in this chapter—slows down the simulations unnecessarily.

Note that each modulator building block in Figure 4.12 uses a suitable schematic symbol. This is very useful in practice to clearly identify the different parts of the modulator and to establish an appropriate hierarchic partitioning that allows us to systematize the design from a top-down/bottom-up approach. This way, designers can move through the modulator hierarchy, using the most convenient schematic representation according to the circuit part being analyzed. As an illustration, Figure 4.13 shows the macromodel of the first integrator in Figure 4.12, where the symbols for the switches, capacitors, and opamps are clearly identified.

As discussed earlier in this chapter, the use of even ideal macromodels allows designers to clearly define the electrical representation of a $\Sigma\Delta$M circuit, including all their nodes and branches. These macromodels can be progressively replaced by their transistor-level implementations as the different building blocks are being sized. This is something that can be done very easily in some circuit design environments, such as in Cadence Design Framework. Figure 4.14 shows how to change the type of implementation, often referred to as *view*. Note that four different cell view names are available in this example: symbol (which is the highest abstraction level), macromodel, schematic (transistor-level), and layout.

As an illustration, Figure 4.15 shows the macromodel of different parts of the SC integrator in Figure 4.13. Figure 4.15a shows the macromodel of the fully-differential amplifier and Figure 4.15b depicts the macromodel of a CMOS switch. In both cases, the different model parameters can be set according to the building-block specifications derived from behavioral simulations, for instance using SIMSIDES as described in Chapter 3.

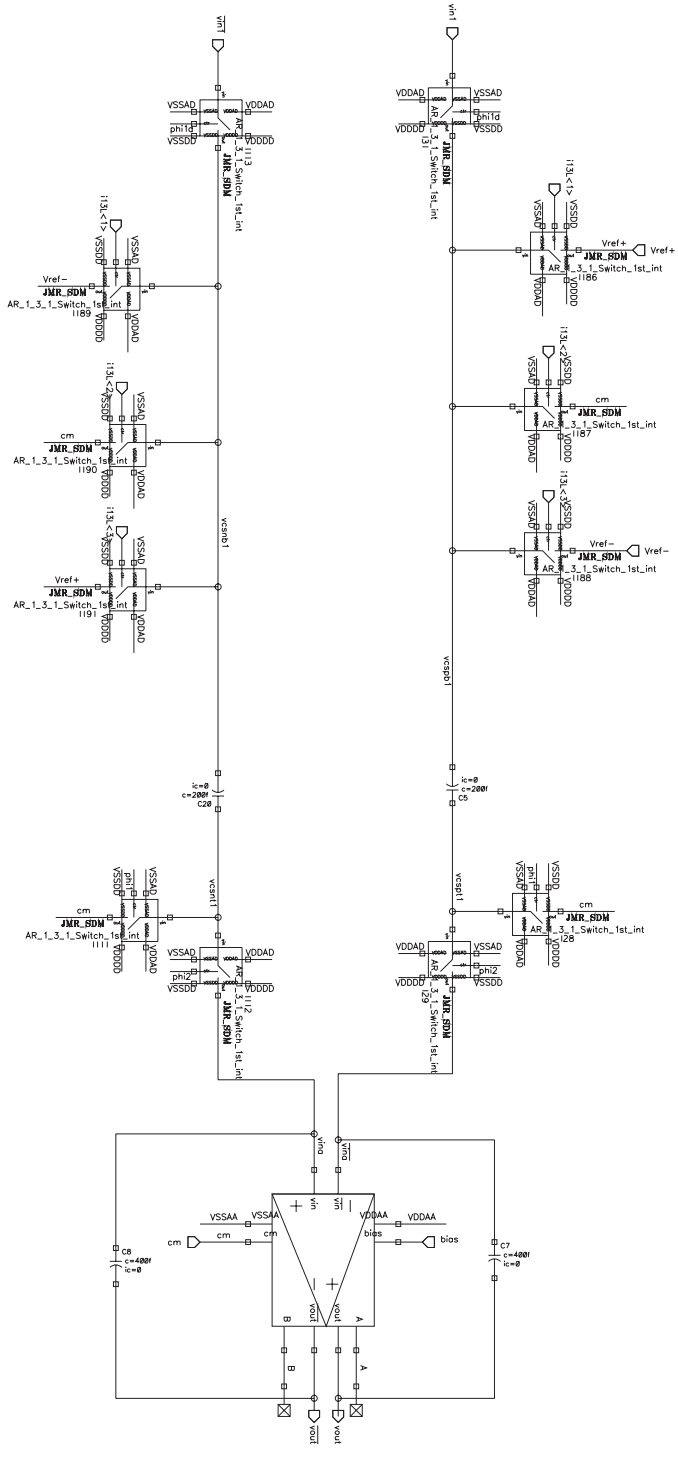


Figure 4.13 Schematic of the SC first integrator of Figure 4.12.

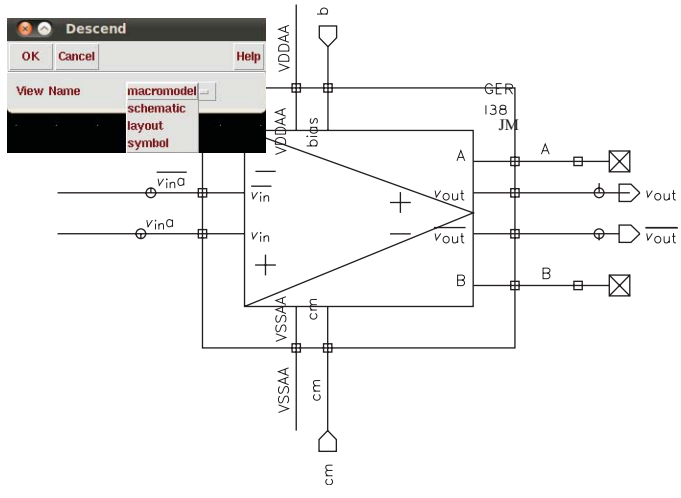


Figure 4.14 Illustrating the selection of a circuit view name in Cadence Design FrameWork.

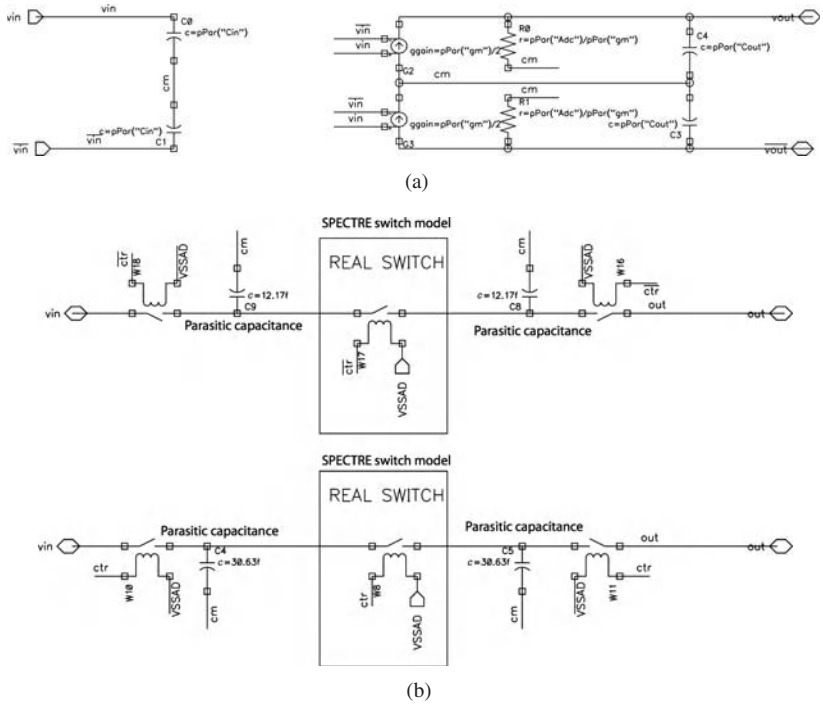


Figure 4.15 Illustrating the macromodel implementation of different circuit elements of SC integrators in Cadence Design FrameWork: (a) fully-differential amplifier and (b) switch.

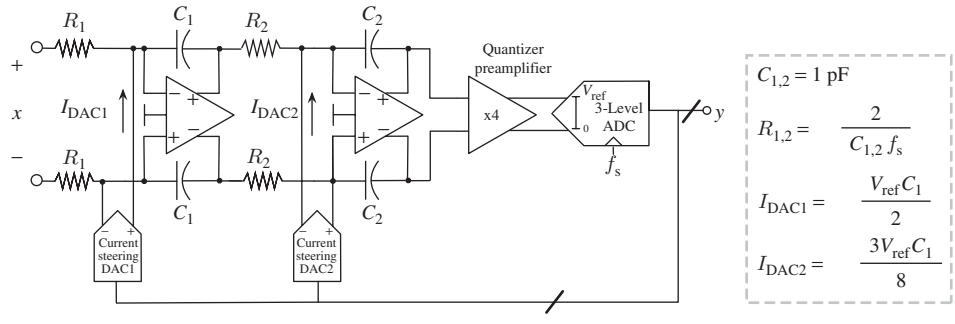


Figure 4.16 Conceptual schematic of a second-order active-RC ΣΔM with a trilevel embedded quantizer and CS feedback DACs.

Second-Order Active-RC ΣΔM

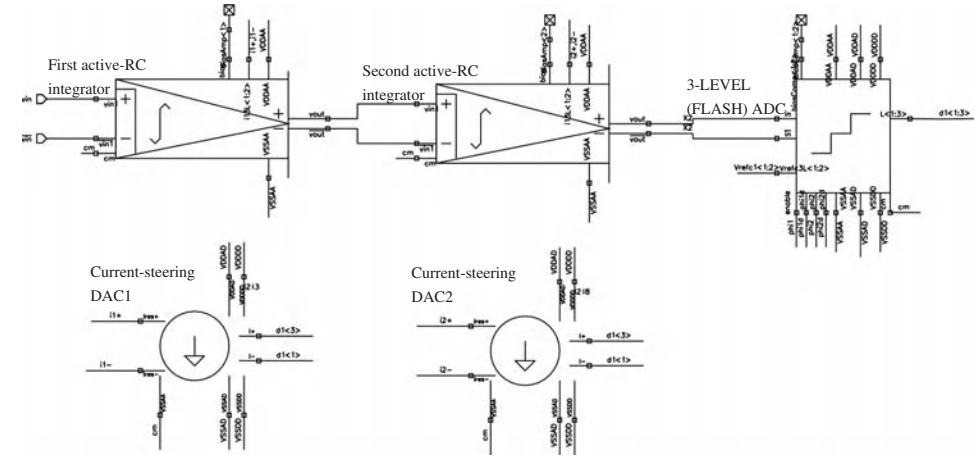
Figure 4.16 shows the conceptual schematic of a second-order active-RC ΣΔM with trilevel embedded quantizer. For the sake of simplicity, the modulator does not include any excess-loop delay cancellation technique. The values and expressions of the resistances, capacitances, and feedback DAC currents are shown in the figure, where V_{ref} denotes the FS reference voltage of the modulator.

The circuit shown in Figure 4.16 can be modeled very easily using the macromodel circuits described in previous sections. Figure 4.17 shows an example implemented in Cadence Virtuoso schematic editor, highlighting the main parts of the circuit. The opamp included in the active-RC model (Figure 4.17b) uses the macromodel described in Figure 4.5b. In this example, the trilevel quantization is implemented as illustrated in Figure 4.18. The ADC, depicted in Figure 4.18a, is modeled as shown in Figure 4.7b. However, for the sake of simplicity, the current-steering DAC, shown in Figure 4.18b, is modeled by two ideal voltage-controlled current sources, which emulate the ideal switches connected in series with the reference currents (Figure 4.10).

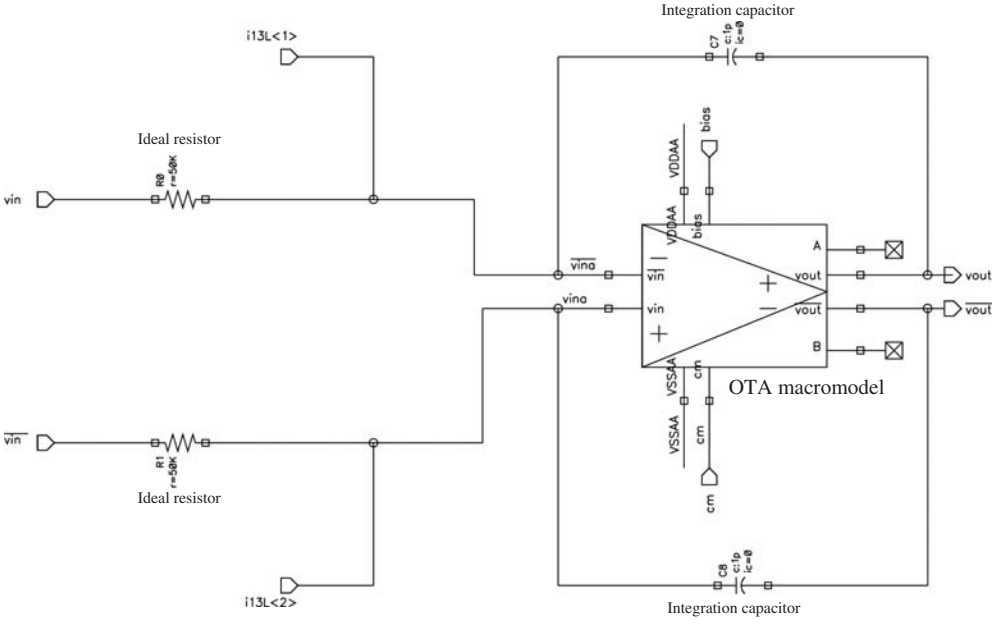
4.2 Including Noise in Transient Electrical Simulations of ΣΔMs

As stated in Chapter 2, circuit noise is an ultimate limiting factor degrading the performance of ΣΔMs. Therefore, it is essential to take into account this effect in all steps of the design flow. At system level, accurate models described in Chapter 2 can be incorporated in behavioral simulations, for instance, using SIMSIDES as detailed in Chapter 3. At electrical level, however, the majority of SPICE-like simulators do not include noise sources in the transient analysis,[5] which makes their analysis at transistor level more complicated.

[5] The majority of examples included in this book used Synopsis® HSPICE®. However, recent versions of some SPICE-like simulators, such as Cadence-Spectre [15] incorporates the possibility to include noise sources in transient analysis. However, the method described in this section allows to isolate the effect of a given noise source instead of considering all noise sources acting together, as done in a regular noise analysis in CADENCE-SPECTRE. Moreover, the methodology explained in this book is well known and can be applied to the majority of SPICE-like simulators, including SPECTRE as well.



(a)



(b)

Figure 4.17 Schematic of the modulator in Figure 4.16 implemented in Cadence Virtuoso schematic editor: (a) modulator and (b) active-RC integrator.

This section describes a methodology that allows designers to do electrical (transistor-level) simulations of ΣΔMs including noise sources. The method—based on generating a noisy data sequence in MATLAB and then injecting this data sequence in the electrical simulation—can be used in most SPICE-like simulators such as PSPICE and HSPICE [4, 16].

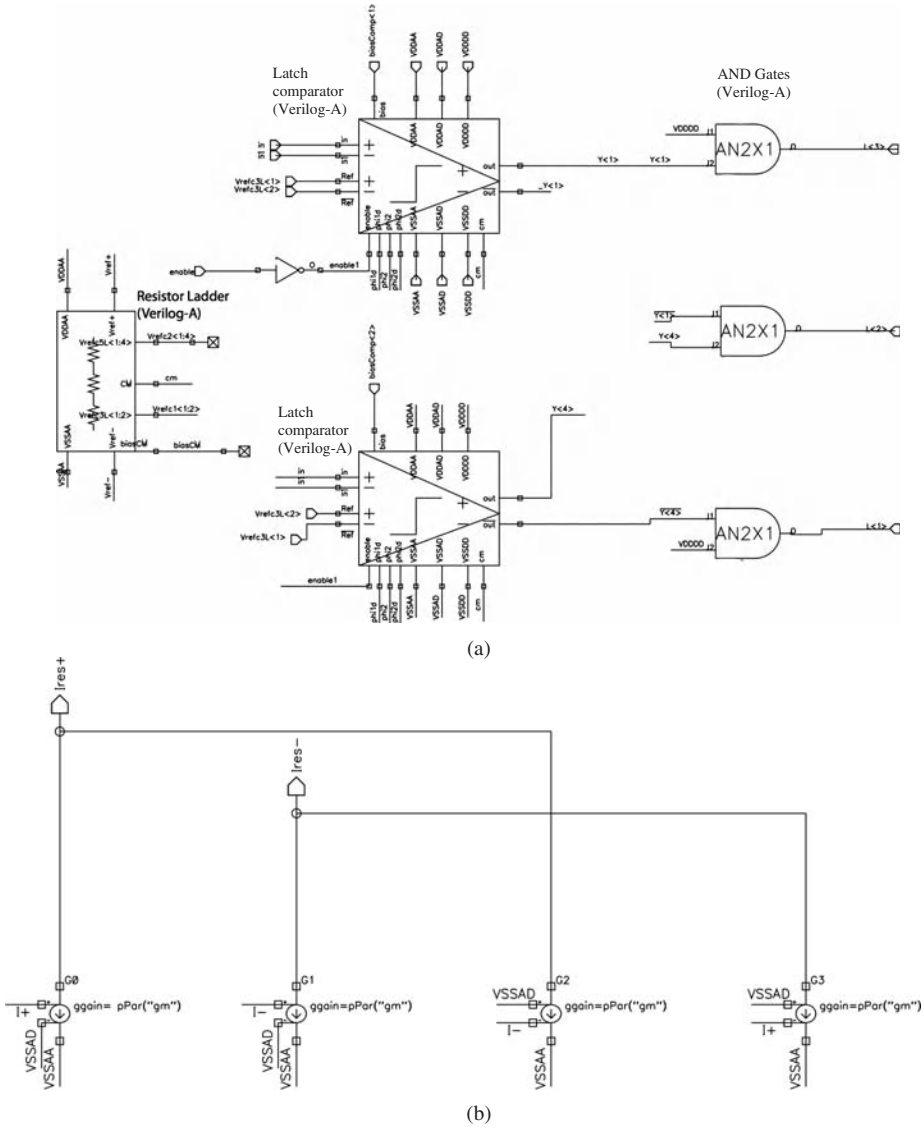


Figure 4.18 Illustrating the macromodel implementation of a trilevel quantizer in Cadence Virtuoso schematic editor: (a) flash ADC and (b) current-steering DAC.

4.2.1 Generating and Injecting Noise Data Sequences in HSPICE

Let us consider the simple SC circuit shown in Figure 4.19a, in which a noise voltage source v_{ni} is sampled by an ideal switch S_1 and stored on capacitor C_S. This circuit can be simulated using the HSPICE netlist shown in Figure 4.19b. In this example $C_S = 1\,\text{pF}$, the switch off/on-resistances used in the ideal switch model are, respectively,

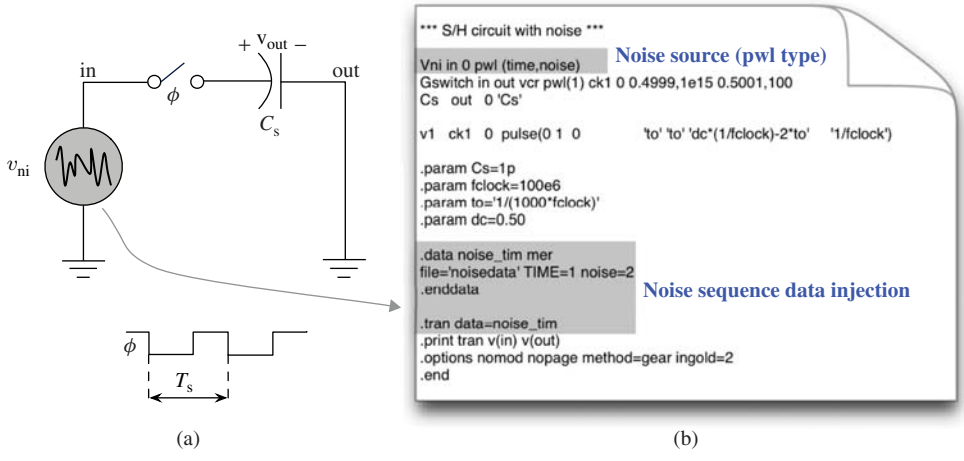


Figure 4.19 Injecting noise data sequences in transient HSPICE simulations: (a) sampling circuit example and (b) SPICE netlist.

$R_{\text{off}} = 10^{15}\Omega$ and $R_{\text{on}} = 100\Omega$ and the clock frequency is $f_s = 100$ MHz. Note that the noise sequence data is *injected* in the HSPICE transient simulation through the use of the `.DATA` statement [16]. This command allows inclusion of data that has been externally generated. In this example, a two-column (time, voltage) format file, named `noise-data`, is loaded, and the transient analysis uses the time data provided in column 1 of file `noisedata` as the sweep input parameter.

In order to compute the noise data used in the electrical simulations, it should be taken into account that $v_{\text{ni}}(t)$ is a random signal, and hence, its instantaneous value is not known. Instead, it can be described as a random process of zero mean and an amplitude uniformly distributed in the range $(-V_{\text{ni}}/2, +V_{\text{ni}}/2)$, where V_{ni} denotes the rms value of $v_{\text{ni}}(t)$. In this case, the mean-square value of v_{ni} is given by [17]

$$\overline{v_{\text{ni}}^2} = \frac{1}{V_{\text{ni}}} \int_{-V_{\text{ni}}/2}^{+V_{\text{ni}}/2} v_{\text{ni}}^2 \mathrm{d}v_{\text{ni}} = \frac{V_{\text{ni}}^2}{12} \tag{4.7}$$

Assuming a band-limited noise source, the PSD of v_{ni} can be computed as

$$S_{v_{\text{ni}}} \simeq \frac{\overline{v_{\text{ni}}^2}}{B_{\text{wni}}} = \frac{V_{\text{ni}}^2}{12 B_{\text{wni}}} \tag{4.8}$$

where B_{wni} stands for the equivalent noise bandwidth of v_{ni}. Hence, if $v_{\text{ni}}(t)$ is sampled at f_{sn}, the value of v_{ni} at instant nT_{sn} can be computed as

$$v_{\text{ni}}(nT_s) = \text{rnd}(\sqrt{12 f_{\text{sn}} S_{v_{\text{ni}}}}) \tag{4.9}$$

where $\text{rnd}(x)$ represents a random number in the range $(-x/2, +x/2)$ and $T_{\text{sn}} \equiv 1/f_{\text{sn}}$ [18].

```
%% GENERATION OF NOISE SEQUENCE OF GIVEN Vrms AMPLITUDE IN A GIVEN BANDWIDTH
%
% noise_Vrms:   Intended amplitude (in Vrms) of the noise integrated
%               in the BW_noise bandwidth
% BW_noise:     Bandwidth in which the noise has been integrated to extract
%               its amplitude
% N:            Numer of samples of the noise time sequence
% fs:           Sampling frequency of the noise time sequence
%
%% SETTING NOISE-SOURCE PARAMETERS
%
noise_Vrms=1e-5;
BW_noise=1e9;
N=65536;
fs=2*BW_noise;
%
%% GENERATION OF NOISE SEQUENCE
%
noise_psd=noise_Vrms^2;
x=sqrt(12*fs/(2*BW_noise)*noise_psd);
y=-x/2+x*rand(N,1);
tstep=1/BW_noise;
%
%% GENERATION OF TIME VECTOR AND SAVING DATA TO THE FILE NAMED noisedata.dat
%
t=0:tstep:(N-1)*tstep;
noisedata=[t' y];
save noisedata.dat noisedata -ascii
%
```

Noise source parameters

Noise data generation

Save data to a file

Figure 4.20 MATLAB code used for generating an N-point data sequence derived from Equation 4.9.

Figure 4.20 shows the MATLAB code used for generating an N-point data sequence derived from Equation 4.9. Note that the data sequence is saved as a two-column format file, in which the first column represents the time series (i.e., $0, T_s, 2T_s, \ldots$) and the second column is the noise data sequence generated using Equation 4.9.

As an illustration, Figure 4.21 shows the PSD of the noise sampled and stored in the capacitor of Figure 4.19a, considering different values of f_s. In this example, a noise source with $V_{ni} = 10\ \mu$Vrms, $B_{wni} = 1$ GHz, and $f_{sn} = 2B_{wni}$ was considered in order to *emulate* an *unlimited-band* noise source. This way, the time interval between two consecutive samples is low enough (1 ns in this example), thus enabling to model the noise source as a CT source [19], which is filtered by the circuit made up of R_{on} and C_S, with R_{on} being the switch on-resistance. This results in an equivalent bandwidth of the sampled noise given by $1/(4R_{on}C_S)$. As shown in Figure 4.21, aliasing occurs in this case because $f_s < 1/(4R_{on}C_S)$. As a consequence the noise power increases within the Nyquist band, that is, from DC to f_s.

4.2.2 Analyzing the Impact of Main Noise Sources in SC Integrators

The simulation technique formerly described can be used for verifying the impact of most critical noise sources on the performance of main $\Sigma\Delta$ building blocks through transient SPICE simulations. This is particularly critical in SC circuits because of the action of their sampled-data circuit nature, with the subsequent effect of the sampling process on noise sources.

As an illustration, let us consider an SC FE integrator similar to that shown in Figure 3.5a. As stated in Chapter 2, their main sources of circuit noise are generated

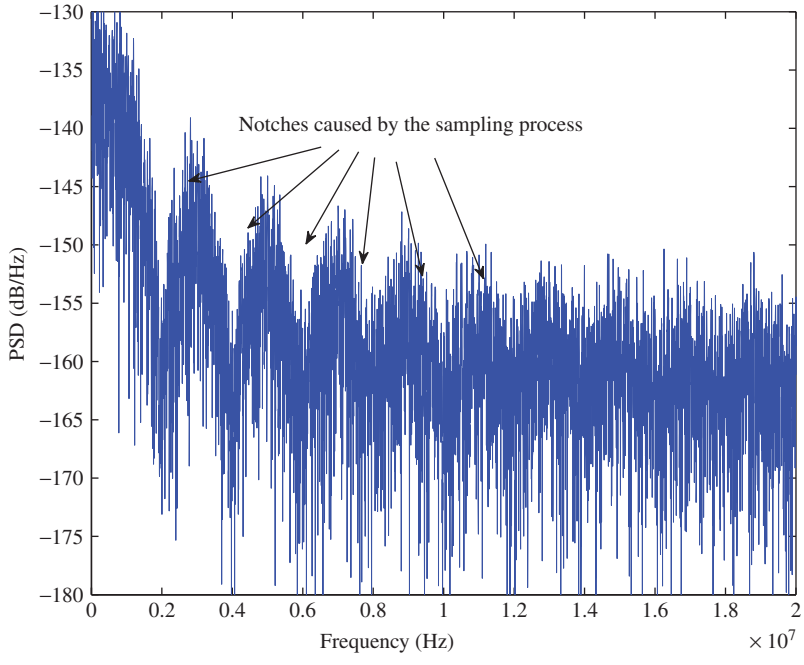


Figure 4.21 Illustrating the effect of sampling noise in HSPICE transient simulations. Output spectrum of the voltage stored in C_S in Figure 4.19a, considering the following simulation data: $R_{on} = 50\Omega$, $C = 0.5$ pF, $f_s = 1$ MHz.

in the switches and in the amplifier. Figure 4.22 shows the circuit schematics that can be used for evaluating the influence of these noise sources on SC integrators and Figure 4.23 shows the corresponding HSPICE netlist. Note that all circuits are built through simple macromodels for the switches and the amplifier in order to isolate the effect of the different error sources to be considered. Figure 4.22a shows the schematic used for simulating the thermal noise introduced by switches controlled by clock phase ϕ_1, and Figure 4.22b shows the corresponding test-bench schematic for the evaluation of thermal noise generated in the switches controlled by ϕ_2. The test-bench schematic used for the noise sources generated in the amplifier is shown in Figure 4.22c. In the latter case, both thermal and flicker noise components need to be generated. This can be done as detailed in the next section.

4.2.3 Generating and Injecting Flicker Noise Sources in Electrical Simulations

The procedure described in Section 4.2.1 assumed white noise sources. However, some noise sources of noise in ΣΔMs also include flicker $(1/f)$ noise components that might be critical in low-bandwidth applications such as sensors, instrumentation, and biomedical applications.

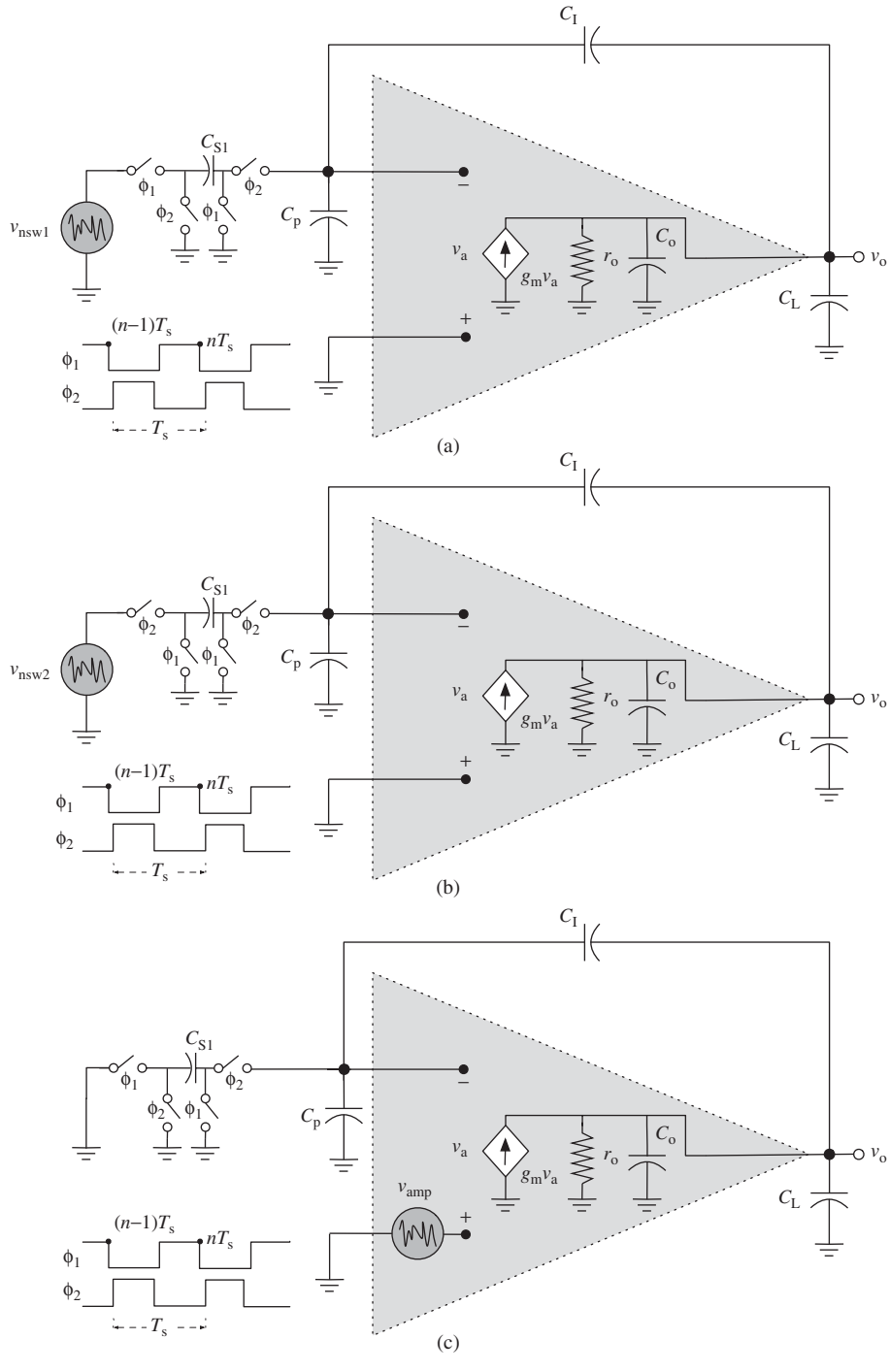


Figure 4.22 Equivalent single-ended circuits used for simulating the effect of main noise sources in SC integrators: (a) noise contribution of switches controlled by ϕ_1 and (b) noise contribution of switches controlled by ϕ_2, and (c) noise contribution of the amplifier.

```
** Transient simulation of SC FE Integrators with noise sources

* Noise sources introduced by switches controlled by phi1
* Comment these two lines for the analysis of vnsw2/vamp noise

V11 11 0 pwl (time,noise)
V12 12 0 0

* Noise sources introduced by switches controlled by phi2
* Comment these two lines for the analysis of vnsw1/vamp noise

V11 11 0 0
V12 12 0 pwl (time,noise)

* Comment these five lines for the analysis of vamp noise

Gswitch11 11    c1in   vcr pwl(1) ck1 0 2.4999,1e15 2.5001,150
Gswitch21 c1in  12     vcr pwl(1) ck2 0 2.4999,1e15 2.5001,150
C1        c1in  c1out 'C1'
Gswitch31 c1out in-    vcr pwl(1) ck2  0 2.4999,1e15 2.5001,150
Gswitch41 c1out 0      vcr pwl(1) ck1  0 2.4999,1e15 2.5001,150

* Noise sources introduced by the opamp
* Comment these eight lines for the analysis of vnsw1/vnsw2 noise
V11 11 0 0
V12 12 0 0
Gswitch11 11    c1in   vcr pwl(1) ck1 0 2.4999,1e15 2.5001,150
Gswitch21 c1in  12     vcr pwl(1) ck2 0 2.4999,1e15 2.5001,150
C1        c1in  c1out 'C1'
vnoise in- in2- 0 pwl (time,noise)
Gswitch31 c1out in2-   vcr pwl(1) ck2  0 2.4999,1e15 2.5001,150
Gswitch41 c1out 0      vcr pwl(1) ck1  0 2.4999,1e15 2.5001,150

Cp  in- 0   'Cp'
Co  in- out 'Co'
Cl1 out 0   'Cl1'

Rin in- 0 1000e10
Gm out 0 in- 0 'Gm'
Ro out 0  1e15

v2   ck2   0  pulse(0 5 'dc*(1/fclock)'  'to' 'to' '(1-dc)*(1/fclock)-2*to' '1/fclock')
v1   ck1   0  pulse(0 5  0              'to' 'to' 'dc*(1/fclock)-2*to'     '1/fclock')

ec1 c1 0 c1in c1out 1

.param C1=0.5p
.param Cp=0.1p
.param Co=2p
.param Cl1=0.1p
.param Gm=1e-3
.param fclock=70e6
.param to='1/(1000*fclock)'
.param dc=0.90

.data noise_tim mer
file='noise20480.dat' TIME=1 noise=2
.enddata

.tran data=noise_tim
.print tran v(out)
.options nomod nopage method=gear post ingold=2
.end
```

Noise contribution of switches

Noise contribution of opamp

Common section

Noise sequence injection

Figure 4.23 HSPICE netlist of the circuits shown in Figure 4.22.

Flicker noise can be generated as a colored noise sequence in MATLAB by filtering a white noise source through a filter with the following transfer function [20]

$$H_{\mathrm{f}}(z) = \frac{1}{(1 - z^{-1})^{\alpha}} \tag{4.10}$$

where α is a real number between 0 and 2. This way, the corresponding noise sequence data can be injected in HSPICE transient simulations using the same method as that described in previous sections.

Alternatively, colored noise data sequence, including both $1/f$ and white noise components, can be generated by extracting the PSD data of the corresponding noise source using a `.NOISE` analysis in SPICE and then, generating a noise time sequence in MATLAB which is equivalent to the captured PSD, and that can be injected in transient simulations using the `.DATA` statement.

Figure 4.24 shows a MATLAB code used for generating a time data sequence according to the electrical data captured from a `.NOISE` simulation in HSPICE. In this example, simulation output data resulting from the noise analysis in HSPICE is stored in a file named `psd_eq_preamp_1f_white_RRF.dat`, which is assumed to be expressed in V^2/Hz, that is, corresponding to a PSD curve. Both PSD and rms values of the $1/f$ and white noise components are identified and computed, and the corresponding time data series are generated using Equation 4.9.

As an illustration, Figure 4.25 shows the output spectra generated by the MATLAB routine in Figure 4.24. Note that a huge number of points ($N = 2^{25}$ in this example) are required to generate FFTs at low frequencies and see the flicker noise corner frequency.

4.2.4 Test Bench to Include Noise in the Simulation of ΣΔMs

At the end of the design phase, transistor-level simulations of the whole ΣΔM are mandatory in order to check that the electrical performance agrees with system-level behavioral simulations, and consequently with target specifications. In this situation, injecting noise sources in SPICE transient simulations is required. To this end, the total input-referred noise source, including the effect of sampling, can be generated and injected at the input node of the modulator following the methodology described in previous sections.

As an illustration, Figure 4.26a shows how to inject noise in transient simulations carried out using Cadence Virtuoso Schematic Editor [21] and Cadence-Spectre simulator [15]. The input-referred noise source is injected in the modulator (a fourth-order cascade 2-2 SC architecture in this example) by including a piece-wise linear (PWL) voltage source at the input node. The noise data sequence is loaded from a file as shown in Figure 4.26b. In this case, the noise data sequence is generated using the MATLAB code shown in Figure 4.27, where the standard deviation of the noise source is computed as

$$v_{\mathrm{ni}} = \mathrm{randn}\left(\sqrt{f_{\mathrm{s}}\frac{P_{\mathrm{ni}}}{2B_{\mathrm{w}}}}\right) \tag{4.11}$$

where f_{s} is the sampling frequency of the modulator, B_{w} is the signal bandwidth, and P_{ni} (`power_IBN` in Figure 4.27) is the IBN due to the noise source—derived from the behavioral simulation data provided by SIMSIDES. Note that in this example, a Gaussian

```
%% GENERATION OF COLORED NOISE SEQUENCE EXTRACTING DATA FROM HSPICE
%
% The PSD curve must contain a 1/f and a white component
% The PSD curve must have been obtained from 0 to fs/2
%
clear; clf;
rand('state',0) % Return RAND to its default initial state
%                 Helps to add repeatability to the results
%N=2^18;         % Length of noise time sequences to be generated
N=2^25;          % N must be >> 1 to plot noise PSD at low frequencies
%
%% Loads and plots PSD curve obtained by electrical SPICE simulation
%
data_in=load('psd_eq_ref_1f_white_RRF.dat'); % Input data is assumed to be expressed in V^2 (PSD)
f_p=data_in(:,1);
noise_psd_p=data_in(:,2);

%
%% Estimation of 1/f and white components

% Flicker (1/f) noise
Sf=noise_psd_p(1)*f_p(1);
noise_psd_1f_p=Sf./f_p;
% Thermal (white) noise
Sw=noise_psd_p(length(f_p));
noise_psd_white_p=Sw*ones(length(f_p),1);

%
%% Generation of white noise sequence
fs=2*max(f_p);
noisew_power=Sw*fs/2;
xw=sqrt(12*noisew_power);
yw=-xw/2+xw*rand(N,1);

%% Generation of 1/f noise sequence
% [1] Generation of an initial white noise sequence
fs=2*max(f_p);
noisewi_power=fs/2;
xwi=sqrt(12*noisewi_power);
ywi=-xwi/2+xwi*rand(N,1);
% [2] Computation of its FFT
ywi_freq=fft(ywi); % MATLAB FFT (double-sided spectrum)
% [3] Filtering of the white noise power with the desired H(f)
% Equivalently, the noise % amplitude is filtered with sqrt(H(f)
freq=(fs/N:fs/N:fs)'; % Double-sided frequency vector
H2_freq=freq./freq; % To validate with H(f)=1
H2_freq=Sf./freq;
yf_freq=sqrt(H2_freq).*(ywi_freq);
yf=real(ifft(yf_freq)); % MATLAB IFFT is used for simplicity

%% Addition of white and 1/f noise sequences
y=yw+yf; % Generated noise sequence including both white and 1/f components
```

Estimation of noise components from electrical data

White-noise sequence generation

1/f noise sequence generation

Figure 4.24 MATLAB code used for generating colored noise data sequence extracted from HSPICE .NOISE simulation.

noise is generated using the `randn` function provided by MATLAB. Several cases of IBN are considered, which correspond to a reconfigurable ΣΔM for multistandard applications. As an illustration, Figure 4.28 shows the output spectrum of a fourth-order cascade 2-2 SC-ΣΔM, in which the effect of injecting thermal noise is highlighted. In this example, macromodels have been considered in all ΣΔ building blocks in order to speed up the simulation, although the method is obviously valid for transistor-level simulations as well.

4.3 Processing ΣΔM Output Results of Electrical Simulations

The ΣΔM output bitstreams obtained by electrical simulations need to be properly processed in order to characterize main figures of merit, namely output spectrum, IBN,

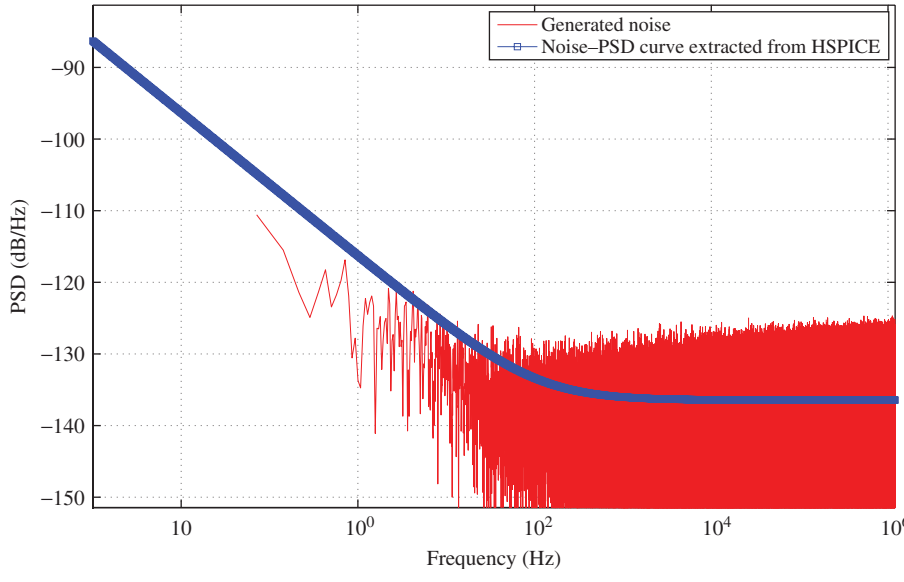
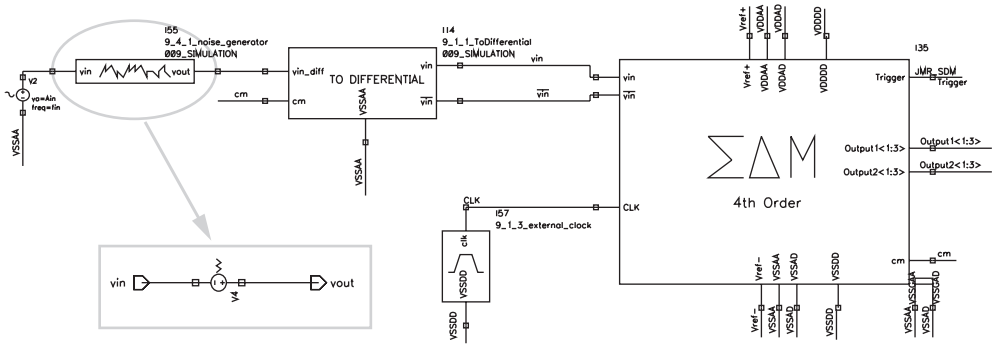


Figure 4.25 Output spectrum generated by the MATLAB routine shown in Figure 4.24.

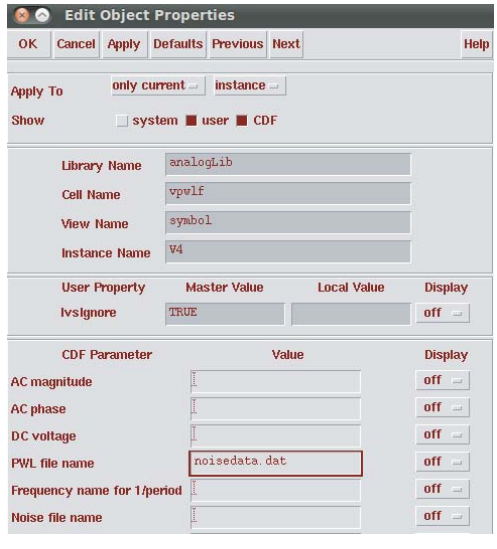
SNR/SNDR, THD, etc. To this purpose, the following step-by-step procedure can be followed:[6]

1. Collect ΣΔM output bitstreams by taking only one sample per clock period during the clock phase in which the comparator outputs are settled.
2. Save the output data in an adequate file format, so that it can be properly loaded and post-processed using a signal processing software such as MATLAB. The most convenient and usual file format is a multiple column text file in which each column corresponds to an output bitstream of the ΣΔM. In cascade topologies, the bitstreams of all the modulator stages must be saved.
3. Load the data in MATLAB and compose the digital outputs from the corresponding bitstreams. For instance, in a ΣΔM with a 3-bit quantizer, the digital output (made up of three binary outputs) is transformed into its equivalent 0-to-7 code and the associated analog level is scaled by the full-scale reference voltage.
4. If the ΣΔM is a cascade topology, the different stage outputs need to be processed by the DCL in order to compose the overall modulator output. Thus, if the DCL is not implemented by hardware, this process can be easily implemented by properly modeling the DCL block diagram in SIMULINK.
5. Compute the different figures of merit (FFT, IBN, SNR/SNDR, THD, etc.) using the corresponding routines, for instance, using MATLAB signal processing toolbox or SIMSIDES post-processing facilities.

[6] The procedure described in this section can be applied to the experimental output results measured in the laboratory, discussed later in this chapter.



(a)



(b)

Figure 4.26 Test-bench example to inject noise in transient simulations of ΣΔMs using Cadence-Spectre simulator: (a) schematic in the Virtuoso editor environment and (b) object properties windows highlighting how to load the noise data sequence file.

Figure 4.29 shows a conceptual diagram of the step-by-step procedure formerly described, considering that the electrical simulations are carried out with Cadence-Spectre, and the results are processed using SIMSIDES. As an example, let us consider a fourth-order cascade 2-2 SC-ΣΔM with trilevel quantization in both stages. Figure 4.30a shows the test-bench schematic used in Cadence to simulate the modulator. In this example, the output of the front-end and the back-end stages of the modulator are, respectively, named `Output1<1:3>` and `Output2<1:3>`, where `Outputj<i>` corresponds to the *i*th output bit of the *j*th stage quantizer, considering a 1-of-3 digital code as illustrated in Figure 4.7. Overall, six bitstreams are collected and stored in a text file. This task is implemented by the blocks named `WRITE OUTPUT`, which implement

```
%% GENERATION OF A GAUSSIAN NOISE SEQUENCE WITH GIVEN IBN POWER
clear; close all;
%% Case 1
% power_IBN_dB=-84.2;
% BW=1e5;
% fs=40e6;
% num_cycles=65786;
%% Case 2
% power_IBN_dB=-77.1;
% BW=5e5;
% fs=80e6;
% num_cycles=33269;
%% Case 3
% power_IBN_dB=-73.9;
% BW=1e6;
% fs=120e6;
% num_cycles=16885;
%% Case 4
power_IBN_dB=-65.2;
BW=10e6;
fs=320e6;
num_cycles=16640;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
power_IBN=10^(power_IBN_dB/10);
std_deviation=sqrt(fs/2*1/BW*power_IBN);

mean=0;
noise=mean+std_deviation*randn(num_cycles,1);
t=0:1/fs:(num_cycles-1)/fs;

data=[t' noise];
save noisedata.dat data -ascii
```

Noise source parameters (four different cases are considered)

Noise data generation

Save data to a file

Figure 4.27 MATLAB code used for generating an N-point data sequence derived from Equation 4.11.

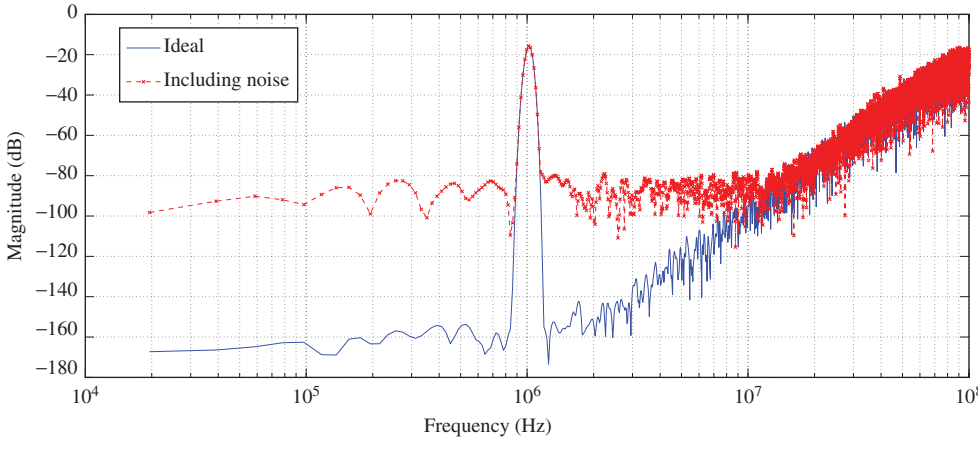


Figure 4.28 Output spectrum of a fourth-order cascade 2-2 SC-ΣΔM. The simulation was carried out in Cadence-Spectre, considering macromodels for all building blocks and injecting the input-referred noise source as illustrated in Figure 4.26.

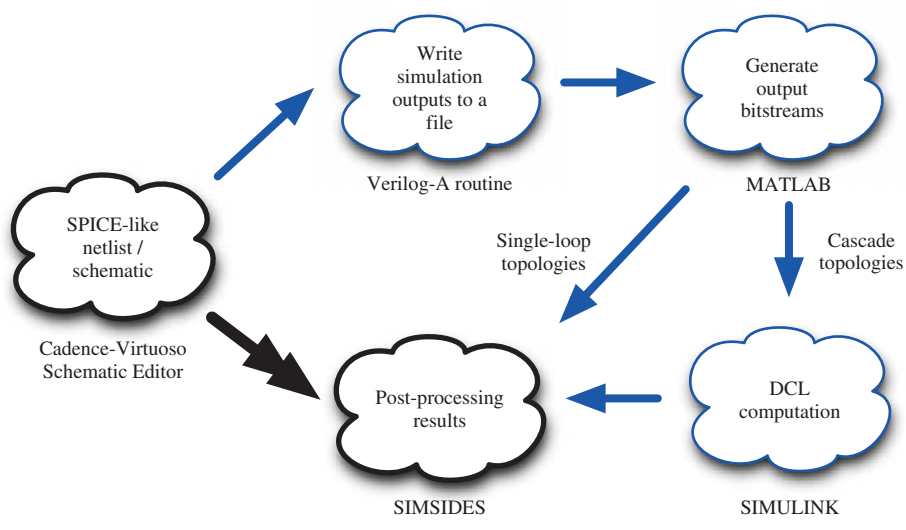
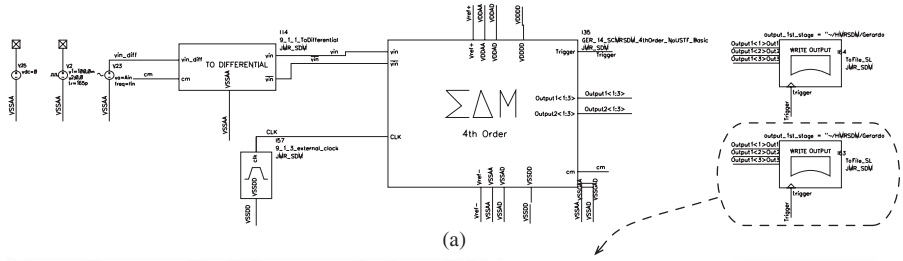


Figure 4.29 Step-by-step procedure to process electrical simulation outputs.



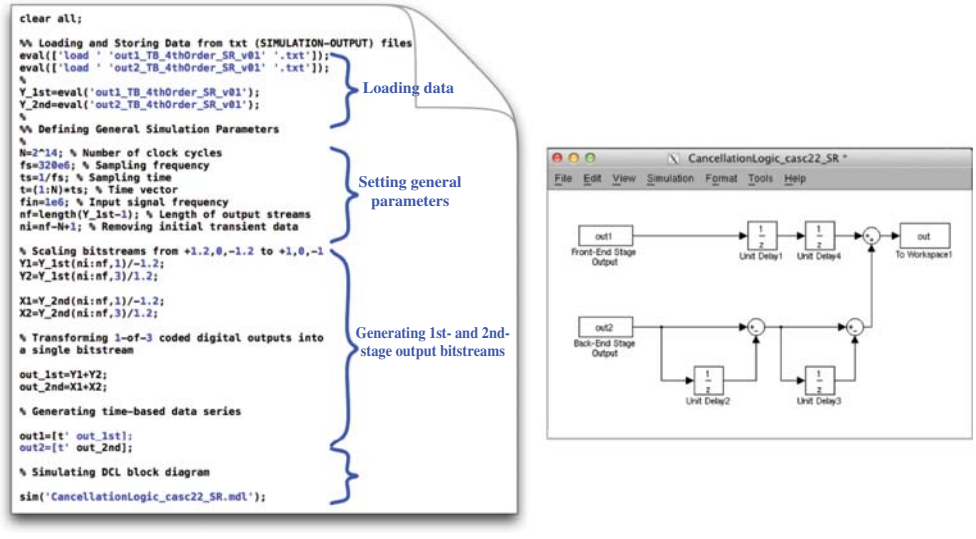

```
// VerilogA code for WRITE OUTPUT block

`include "constants.vams"
`include "disciplines.vams"
module ToFile_SL(trigger,Out1, Out2, Out3);
input Out1, Out2, Out3;
input trigger;
electrical trigger, Out1, Out2, Out3;
integer wfile1;
real vth;
parameter output_1st_stage = "~/DFWII/out1.txt";
// string constant, must be on one line

      analog begin
               vth=0.6;
            @(initial_step("dc","ac","tran","xf"))begin
                  wfile1=$fopen(output_1st_stage);
            end

      @( cross(V(trigger)-vth,-1)  ) begin
            $fstrobe(wfile1,"%1.1f %1.1f
%1.1f",V(Out1),V(Out2),V(Out3));
      end

      $fclose(wfile1);

      end
endmodule
```

(b)

```
0.0 1.2 0.0
0.0 1.2 0.0
0.0 1.2 0.0
0.0 1.2 0.0
0.0 1.2 0.0
0.0 1.2 0.0
0.0 1.2 0.0
0.0 1.2 0.0
0.0 0.0 1.2
1.2 0.0 0.0
1.2 0.0 0.0
0.0 0.0 1.2
0.0 0.0 1.2
1.2 0.0 0.0
0.0 0.0 1.2
1.2 0.0 0.0
0.0 0.0 1.2
1.2 0.0 0.0
0.0 0.0 1.2
1.2 0.0 0.0
0.0 0.0 1.2
1.2 0.0 0.0

……..

0.0 0.0 1.2
1.2 0.0 0.0
```

(c)

Figure 4.30 Collecting and storing the output bitstreams of a ΣΔM in an electrical simulation: (a) test-bench schematic in Cadence Design FrameWork, (b) Verilog-A code to capture simulation output results, and (c) excerpt of the generated output file (text format).



(a) (b)

Figure 4.31 MATLAB routine used for processing ΣΔM outputs from electrical simulations: (a) MATLAB code and (b) DCL block diagram implemented in SIMULINK.

the Verilog-A code shown in Figure 4.30b. Note that the output data—consisting of the three bitstreams named `Out1, Out2, Out3`—are collected and stored in the text file at a rate of one sample per clock cycle as illustrated in Figure 4.30c. The event at which the data are collected is governed by a trigger signal, named `trigger` in this example. This trigger signal corresponds to the clock phase in which the comparator outputs are settled, which in this example is clock phase ϕ_2.

Once the simulation output data has been stored in a multicolumn text file, this data can be loaded and processed using the MATLAB routine shown in Figure 4.31a. Note that the bitstreams of both stages are scaled from the values of the supply voltage (1.2V) to 1V and the 1-of-3 codification is transformed into a single-bit output format. Both output bitstreams are processed by a DCL implemented in SIMULINK as shown in Figure 4.31b. Once this DCL diagram is simulated, the overall modulator output is saved into the MATLAB workspace and can be processed using SIMSIDES post-processing facilities. As an illustration, the output spectrum shown in Figure 4.28 was obtained using the procedure and routines described in this section.

4.4 Design Considerations and Simulation Test Benches of ΣΔM Basic Building Blocks

Once the modulator has been verified using macromodels and the performance has been evaluated considering main nonideal circuit and physical effects (including circuit noise), the next step is the electrical transistor-level design of ΣΔM building blocks and circuit elements. In this book, we will distinguish between two different categories of ΣΔM

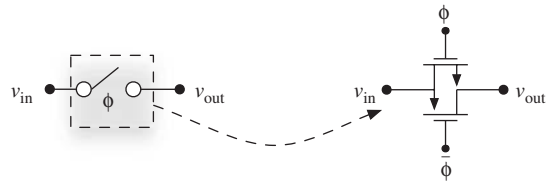


Figure 4.32 Switch symbol and its equivalent CMOS circuit.

building blocks or subcircuits. The first category, referred to as *basic* building blocks, includes the loop filter (essentially based on integrators and resonators) and the embedded quantizer, made up of an ADC (usually a flash ADC made of a bank of comparators) and a DAC. The second category includes a number of so-called *auxiliary* building blocks which are also needed to implement a ΣΔM IC. Among others, the most important auxiliary blocks are the clock-phase generator, the master bias generator, the reference voltage generator, and the digital circuits required for buffering and signal processing.

This section deals with the design of basic ΣΔM building blocks, focusing on their essential circuit elements, namely switches, comparators, OTAs, and DACs. The main design considerations concerning these circuits are described together with practical simulation test benches frequently used for characterizing their main electrical performance metrics, which have been derived from system-level behavioral simulations.

4.4.1 Design Considerations of CMOS Switches

Practically, all switches used in SC-ΣΔMs are of CMOS type, that is, based on a pMOS and an nMOS transistor connected in parallel, as illustrated in Figure 4.32.[7] As stated in Chapters 2 and 3, the most important design specification of CMOS switches is the switch on-resistance R_{on}. The value of R_{on} is mainly constricted by dynamic considerations that affect the integrator transient response and consequently the effective resolution of the modulator [22].

Let us consider that the CMOS switch of Figure 4.32 is switched on, that is, $\phi = V_{DD}$ and $\bar{\phi} = V_{SS}$. Assuming that the nMOS and pMOS transistors operate in the ohmic region, their on-resistances can be approximated by Equation 2.47, and the overall CMOS switch on-resistance is the result of the parallel connection of resistors R_{onN} and R_{onP}.

Trade-Off Between R_{on} and the CMOS Switch Drain/Source Parasitic Capacitances

As discussed in Section 2.7.2, the value of R_{on} can be reduced by increasing the aspect ratio (W/L) of both transistors in the CMOS switch. However, this increases the transistors area, and consequently their associated drain/source parasitic capacitances, with the

[7] Some subcircuits, such as some types of SC common-mode feedback (CMFB) circuits or the latches used in some comparators, may use either nMOS or pMOS switches. However, the vast majority of switches in SC-ΣΔMs are implemented as CMOS transmission gates.

subsequent penalty in the transient response and integrators dynamics degradation. Therefore, there is a trade-off between the maximum value of R_{on} that can be tolerated—which can be determined by behavioral simulation as shown in Chapter 3—and the drain/source parasitic capacitances associated with the CMOS switch that are in turn conditioned by the value of capacitors used in the SC branches. This way, switch transistor sizes can be scaled down across the modulator chain, using higher sizes in the front-end switches—where larger capacitors are chosen according to thermal noise considerations—while lower sizes are tolerated in back-end integrators, where smaller capacitors are normally used and hence the influence of switch parasitic capacitances is diminished.

Characterizing the Nonlinear Behavior of R_{on}

According to Equation 2.47, the values of R_{onN} and R_{onP} depend on the switch common-mode voltage, $v_{CM} \equiv (v_{in} + v_{out})/2$, and hence on the drain and source voltages (denoted as v_{in} and v_{out} in Figure 4.32) of the nMOS and pMOS transistors. As a consequence, the value of R_{on} becomes a nonlinear function of the voltage being transmitted, thus generating harmonic distortion as discussed in Chapters 2 and 3.

In order to evaluate the nonlinear characteristic of R_{on} in CMOS switches, the circuit shown in Figure 4.33a can be used. The gate of nMOS transistor is connected to V_{DD} and the gate of pMOS transistor is connected to V_{SS} so that both transistors are switched on. A small voltage imbalance, typically in the order of 10–20 mV, is applied across the CMOS switch to guarantee that both nMOS and pMOS are properly biased and operate in the linear region. A DC common-mode voltage source is connected to the input node

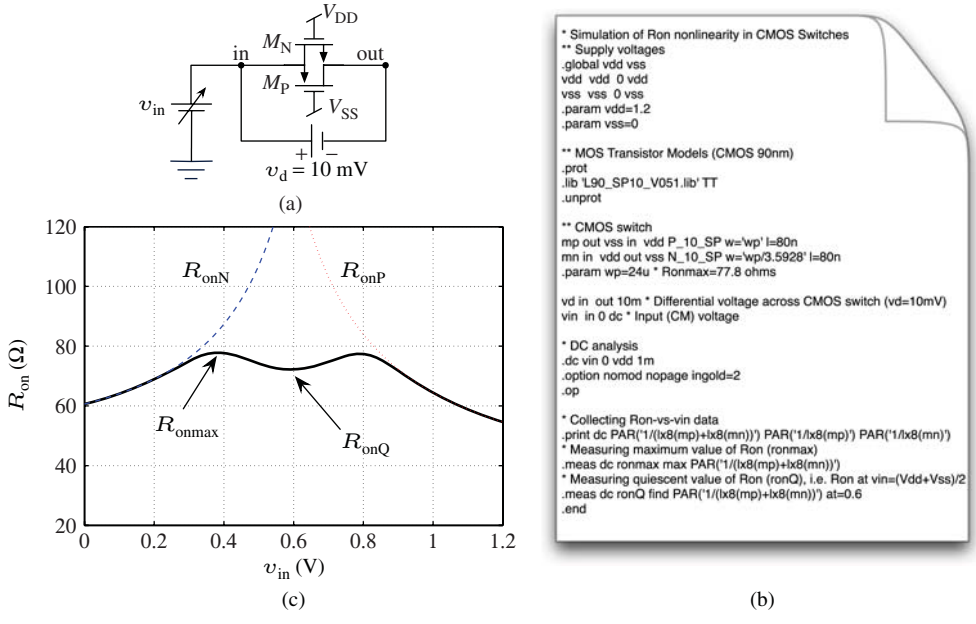


Figure 4.33 Characterizing nonlinear switch on-resistance: (a) test-bench circuit, (b) HSPICE netlist, and (c) R_{on} versus v_{in} considering a 90-nm CMOS technology with 1.2-V supply voltage.

of the switch. This voltage is swept in a DC analysis in order to evaluate its impact on the variation of R_{on}.

Figure 4.33b shows the SPICE netlist used for simulating the circuit in Figure 4.33a. In this example, a 10-mV voltage, named `vd`, is applied across the switch and the common-mode voltage (`vin` in Figure 4.33a) is swept using a `.DC` analysis in SPICE. The value of R_{on} can be extracted from each operating point in the `.DC` analysis by means of a parameter defined as `PAR(1/(lx8(mp)+lx8(mn))` where `lx8(mp,n)` is an alias parameter used in HSPICE to represent the DC drain-source conductance of MOS transistors (M_N and M_P in Figure 4.33a). Thus, the values of R_{onN} and R_{onP} are extracted from `1/lx8(mn)` and `1/lx8(mp)`, respectively [16].

Figure 4.33c represents R_{on} as a function of v_{in}, giving rise to a function similar to that shown in Figure 2.20b. The curves corresponding to both R_{onN} and R_{onP} are also depicted to illustrate the separate contribution of each transistor to the overall switch on-resistance. The maximum value of R_{on}, denoted as `ronmax`, and the quiescent value of R_{on}, denoted as `ronQ`, can also be extracted from HSPICE simulations using the `.meas` command [16] as detailed in the netlist shown in Figure 4.33b.

The R_{on}-versus-v_{in} characteristic shown in Figure 4.33c has been obtained for $W_P K_P = W_N K_N$, which according to Equation 2.47, gives an almost symmetrical function. An alternative approach consists of increasing W_N to equal W_P, as illustrated in Figure 2.22. For the sake of completeness a similar figure is depicted in Figure 4.34, considering a 90-nm CMOS technology with a 1.2-V supply voltage. It can be noted how as W_N

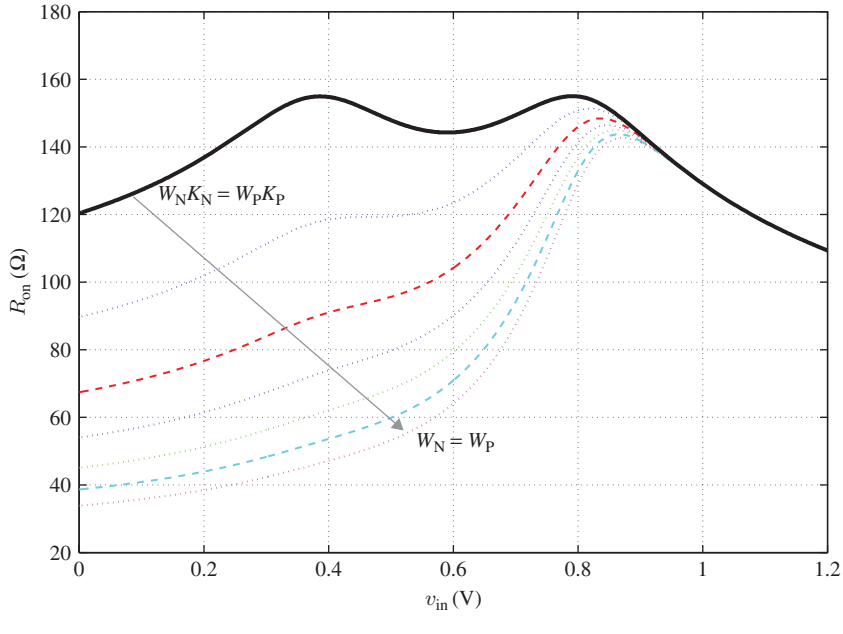


Figure 4.34 R_{on} versus v_{in} for different values of W_N, considering a 90-nm CMOS technology with 1.2-V supply voltage ($L_{N,P} = 80$ nm).

increases, the nonlinearity of the switch on-resistance increases, although its average value decreases. Hence, as discussed in Section 2.7.2, if the sizing of the nMOS and pMOS transistors in a CMOS switch compensates the difference in their transconductance parameters, the switch on-resistance nonlinearity is reduced, but the average on-resistance is larger than using the same sizes, that is, $W_N = W_P$. In the latter case, the area, and consequently the values of parasitic drain/source capacitances, increases as well although the overall effect of the finite switch on-resistance on the settling performance decreases [23]. Therefore, there is a design trade-off involving the switch on-resistance nonlinearity and its average value, which at the modulator level translates in the well-known analog design trade-off between speed (limited by the incomplete settling) and linearity (limited by nonlinear switch on-resistance) [22]. Nevertheless, in the majority of state-of-the-art SC-ΣΔMs, CMOS switches are designed to keep a low enough average value of R_{on}, while keeping a symmetrical R_{on}-versus-v_{in} characteristic similar to that shown in Figure 4.33c.

Influence of Technology Downscaling

According to Equation 2.47, the reduction of the supply voltage caused by CMOS technology downscaling causes an increase of R_{on}. However, this effect can be compensated by the lower channel lengths ($L_{N,P}$) used in smaller technologies. This is illustrated in Figure 4.35, where R_{on} is represented versus W_P for $W_P K_P = W_N K_N$, considering different CMOS processes from 250 to 90 nm. Note that, generally speaking, the design of switches

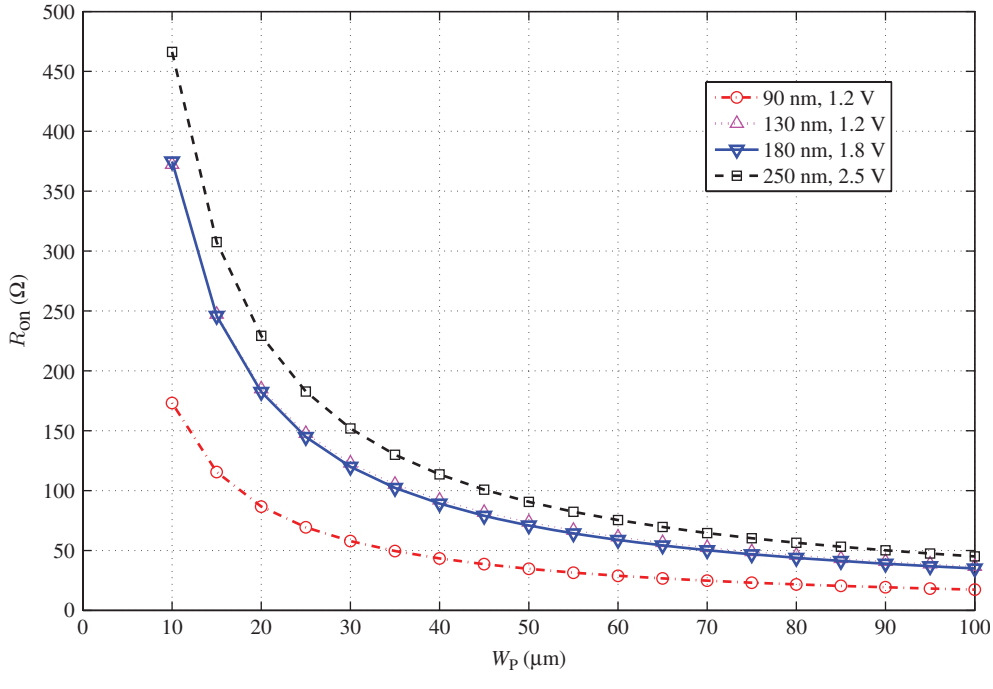


Figure 4.35 Illustrating the effect of technology downscaling on R_{on}.

benefits from technology downscaling as lower values of R_{on} can be obtained for the same (or even smaller) switch sizing, with the subsequent advantages in terms of silicon area and robustness against parasitic capacitances. It can be noticed how migrating from 180 to 130 nm has no effect on R_{on} because the reduction in MOS transistor sizes is compensated by the voltage supply downscaling. However, comparing 130 nm with 90 nm, both using 1.2-V supply voltage, the downscaling process becomes beneficial.

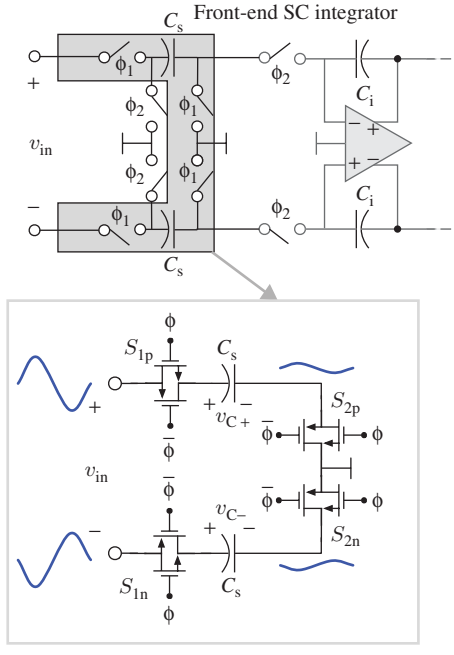
Evaluating Harmonic Distortion

Figure 4.36a shows the equivalent circuit that can be used for the evaluation of the harmonic distortion caused by the nonlinear sampling process because of the finite switch on-resistance. This test-bench circuit—which contains essentially the equivalent circuit shown in Figure 2.22—corresponds to the fully-differential implementation of the input SC branch in a typical front-end integrator used in SC-ΣΔMs—conceptually illustrated in Figure 4.36a. As discussed in Section 2.7.2, as switch $S_{1p,n}$ is directly connected to the input node, its nonlinear on-resistance may vary a lot during the sampling period, thus generating considerable harmonic distortion. In contrast, switch $S_{2p,n}$ has one of its terminals connected to a fixed voltage—the analog common-mode voltage—so that the voltage across this switch remains approximately constant over clock periods. As illustrated in Figure 4.36a, the voltage variations at the input node of S_2 are considerably lower than that in S_1. As a consequence, the effect of S_2 can be neglected in practice. Indeed, the test-bench circuit shown in Figure 4.36b will give essentially the same results as the circuit in Figure 4.36a in the majority of practical situations.

The corresponding SPICE netlist is shown in Figure 4.36c. A transient analysis is carried out considering different situations for the input signal, that is, a single tone signal, a two-tone signal, etc. To this end, a single-ended source is converted to a differential input signal using voltage-controlled voltage sources, while some subcircuits are used for representing the switches and sampling branches included in the test bench. A `.TRAN` analysis is carried out with a printing time step defined as the sampling period—10 ns in this example. The stop time is given by `N/fs`, where `N` is the number of simulation clock cycles (4096 in this example) and `fs` is the sampling frequency (100 MHz in this example).

As stated above, harmonic distortion is mainly caused by the switches that are directly connected to the input ($S_{1p,n}$) given that the voltages at the input/output nodes of the remaining ones remain approximately constant over clock periods, and consequently R_{on} is approximately constant during the sampling phase time. The same reasoning applies to those switches that are in any of the following situations:

- Switches that are connected to nodes where the voltage keeps constant over the sampling period.
- Switches that are connected to the output of an SC circuit. For instance all switches of the back-end SC integrators, that is, all integrators except the front-end one. In all these switches, the voltages in their terminals remain constant over clock periods.
- Sampling switches that are connected to the input node of SC-ΣΔMs whenever the ratio between the input signal frequency and the sampling frequency is small, typically less than one-tenth.




```
*Simulation of the HD due to nonlinear sampling

.global vdd vss
vdd  vdd  0  vdd
vss  vss  0  vss

**** MOSFET MODELS (90nm CMOS)
.prot
.lib 'L90_SP10_V051.lib' TT
.unprot

*vin in 0 sin(0 'Vin' 'fi') * Single-tone input signal
vin1 in in2 sin(0 'Vin' 'fi1') * Two-tone input signal
vin2 in2 0 sin(0 'Vin' 'fi2')
**********************************************

ep inp 0 in 0 0.5
en inn 0 in 0 -0.5
xsamp inp op phi1 phi1n samp
xsamn inn on phi1 phi1n samp

** CMOS switch subcircuit
.subckt swt in out phi phin
mp out vss in  vdd P_10_SP w='wp' l=80n
mn in  vdd out vss N_10_SP w='wp/3.5928' l=80n
.ends

** Single-ended branch circuit
.subckt samp in out phi phin
xsw1 in out phi phin swt
c out 0 cs
.ends

** Clock-phase generator

v1   phi1  0      pulse('vss' 'vdd'  0'trf'      'trf'  'teff'  'Ts')
v1n  phi1n 0      pulse('vdd' 'vss'  0'trf'      'trf'  'teff'  'Ts')

.param fs='100e6' * Sampling frequency
.param Ts= '1/fs' * Sampling period
.param td= '0.025n'
.param tov='0.025n'
.param trf='0.02n'
.param teff='Ts/2-td-tov-2*trf'

**********************************************
.param wp='12u' * Ronmax=155
.param BW='10e6' * Signal bandwidth (LP-SDM)
.param fi='BW/3*1.012345' * Input frequency
.param fi1='24.5012345e6' * Input frequency (1) (BP-SDM)
.param fi2='25.5012345e6' * Input frequency (2) (BP-SDM)
.param N=4096 * Number of points
.param tfinal='N/fs'
.param cs=1p
.param vdd=1.2
.param vss=0
.param Vin='vdd/(2*sqrt(2))'

.tran 10n 'tfinal'
.print tran v(in) v(op,on)
.options nomod nopage method=gear ingold=2
+ absi=1e-12 absv=1e-4 delmax=0.25n
+reli=1e-5 trtol=3
.end
```

(c)

Figure 4.36 Characterizing harmonic distortion caused by nonlinear sampling: (a) test-bench equivalent circuit, (b) practical (simplified) version of the test-bench circuit, and (c) HSPICE netlist. (A test-bench circuit with $\phi = V_{DD}$ and $\bar{\phi} = V_{SS}$ can be used as well.)

The latter situation is illustrated in Figure 4.37, where a sinewave input signal with different values of the input frequency f_{in} is represented within the sampling time period, from 0 to $T_s/2$, with $T_s = 1/f_s$ and $f_s = 100$ MHz in this example. Note that as the ratio between f_{in} and f_s increases, there are larger variations of v_{in} over the sampling period, which translates into a higher variation of R_{on}, thus increasing the harmonic distortion.

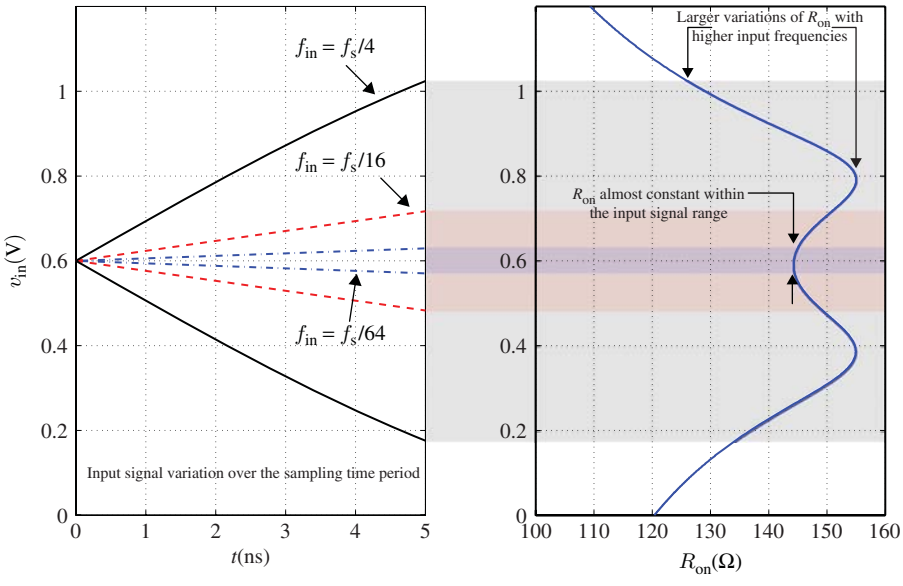


Figure 4.37 Impact of increasing the input signal frequency on the variation of R_{on} over the sampling period. A fully-differential sinewave input signal of frequency f_{in} is considered with $f_s = 100$ MHz. This figure plots both positive and negative single-ended inputs.

The analysis of the harmonic distortion caused by the nonlinear sampling process using the Volterra series method [24, 25] demonstrates that the third-order harmonic distortion caused by this dynamic nonlinearity is approximately given by [26]

$$HD_3 \simeq \frac{\pi f_{in} C_S R_{on}}{2(V_{ON} - V_T)^2} A_{in}^2 \tag{4.12}$$

where A_{in} is the input signal amplitude, V_{ON} denotes the switch-on voltage (either V_{DD} or $|V_{SS}|$), and V_T is the maximum (worst-case) value of V_{TN} and $|V_{TP}|$.

The expression given in Equation 4.12 is consistent with the results highlighted in Figure 4.37, showing a direct dependency of HD_3 on f_{in}. This is illustrated in Figure 4.38, where several output spectra of the circuit in Figure 4.36b are depicted for different values of f_{in}, with $A_{in} = V_{DD}/2$. These output spectra have been processed with SIMSIDES based on the use of a Kaiser FFT window.

In practical situations, f_{in} is chosen to be $f_{in} = B_w/3$, with B_w being the signal bandwidth. This way, the third-order harmonic components fall into the signal bandwidth. This is not the case of BP-ΣΔMs where the notch frequency (i.e., the center signal frequency) is typically placed at $f_s/4$ as stated in Chapter 1. In this case, other distortion metrics, such us the third-order intermodulation distortion IM_3 is used. Thus, in this case a two-tone input signal is used in the test-bench circuit of Figure 4.36b. As an illustration, Figure 4.39 shows the output spectrum of Figure 4.36b considering two tones with

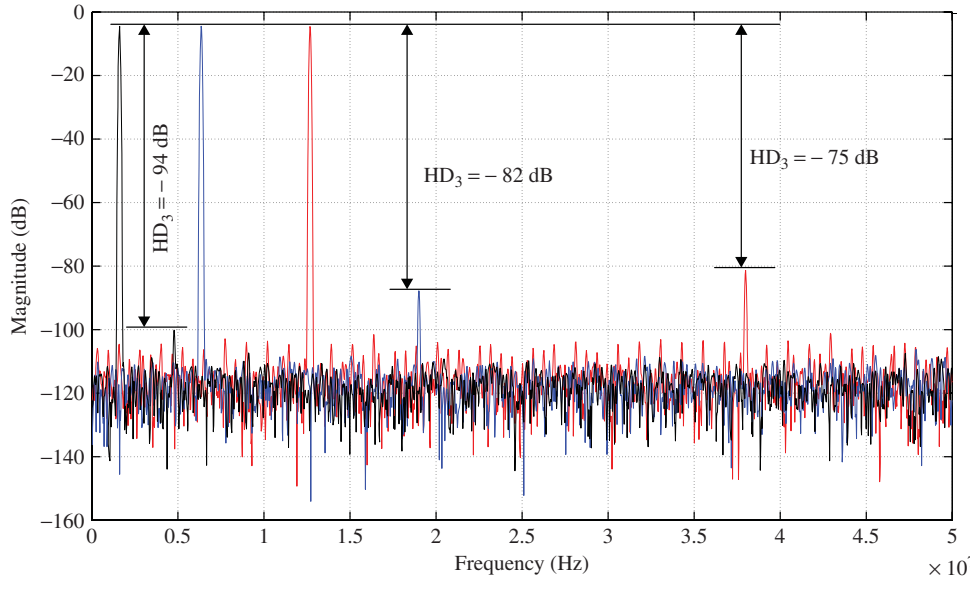


Figure 4.38 Illustrating the impact of increasing the input signal frequency on the harmonic distortion caused by nonlinear sampling process. Three cases of f_{in} are considered: $f_{in} = f_s/64$, $f_{in} = f_s/16$, and $f_{in} = f_s/8$, where $f_s = 100$ MHz.

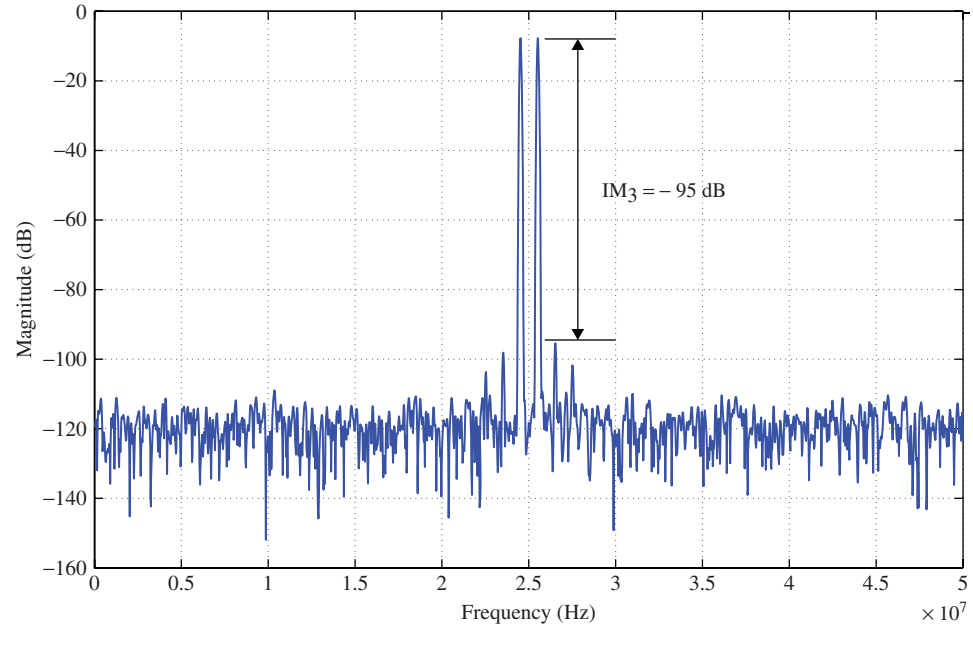


Figure 4.39 Intermodulation distortion caused by the nonlinear sampling operation. Data used in the simulation: 90-nm CMOS technology, $V_{DD} = 1.2$ V, $W_P = 12$ µm, $W_N = 3.34$ µm, $L_P = L_N = 80$ nm, $R_{onQ} = 155\Omega$.

amplitude $V_{DD}/(2\sqrt{2})$ located at[8] $f_{in1} = 24.5012345$ MHz and $f_{in2} = 25.5012345$ MHz with $f_s = 100$ MHz.

4.4.2 Design Considerations of Operational Amplifiers

Voltage amplifiers are basic circuits of SC-ΣΔMs used for building SC integrators and resonators. They are also used for implementing active-RC integrators in CT-ΣΔMs. As already discussed in Chapters 2 and 3, the main electrical requirements of amplifiers can be determined from closed-form expressions and behavioral simulations, and usually comprise specifications for the DC gain, output swing, dynamic behavior, and input-referred noise.

Typical Amplifier Topologies

Many different topologies can be considered in order to fulfill the derived amplifier specifications at the transistor level. Figure 4.40 depicts some of the most representative amplifier topologies that are widely used in ΣΔM design, namely:

- *Telescopic Amplifier* (Figure 4.40a). This single-stage topology is capable of providing a moderate DC gain and an excellent dynamic behavior while being very power efficient, as it employs a single current branch. However, the topology requires five stacked transistors, what results in a reduced output swing and complicates its design (or even prevents its use) in low-voltage ΣΔ implementations. Nevertheless, the telescopic amplifier should be considered the best option if high DC gain and high output swing are not required.
- *Folded Cascode Amplifier* (Figure 4.40b). This single-stage topology exhibits an output swing larger than that of the telescopic amplifier—only four transistors are stacked—but doubles the power consumption because of the two current branches required. It provides a very good settling behavior, although its first nondominant pole—and thus its phase margin—is somewhat lower compared to the telescopic amplifier. Folded cascode amplifiers are often used if moderate DC gain is required in ΣΔMs with medium- and low-voltage supply.
- *Folded Cascode Amplifier with Gain Boosting* (Figure 4.40c). This topology provides larger DC gain than the conventional folded cascode amplifier in Figure 4.40b by means of increasing its output resistance through the regulation of the cascode transistors. The auxiliary amplifiers are usually designed as simple as possible for their additional power consumption not to penalize that of the overall amplifier. Gain-boosting techniques are often employed in single-stage amplifiers, although especial attention must be paid so

[8] The input sinewave frequency f_{in} should be located precisely in an FFT bin in order to avoid signal power to spread to adjacent bins [27]. To this end, the number of periods N_p of the input signal within the simulated time (N/f_s, where N stands for the number of points in the FFT and f_s for the sampling frequency) must be an integer. To meet this constraint, the value of the input frequency can thus be adjusted according to

$$f_{in} \leftarrow N_p \frac{f_s}{N} \quad , \text{ where } N_p = N\lfloor \frac{f_{in}}{f_s} \rfloor \tag{4.13}$$

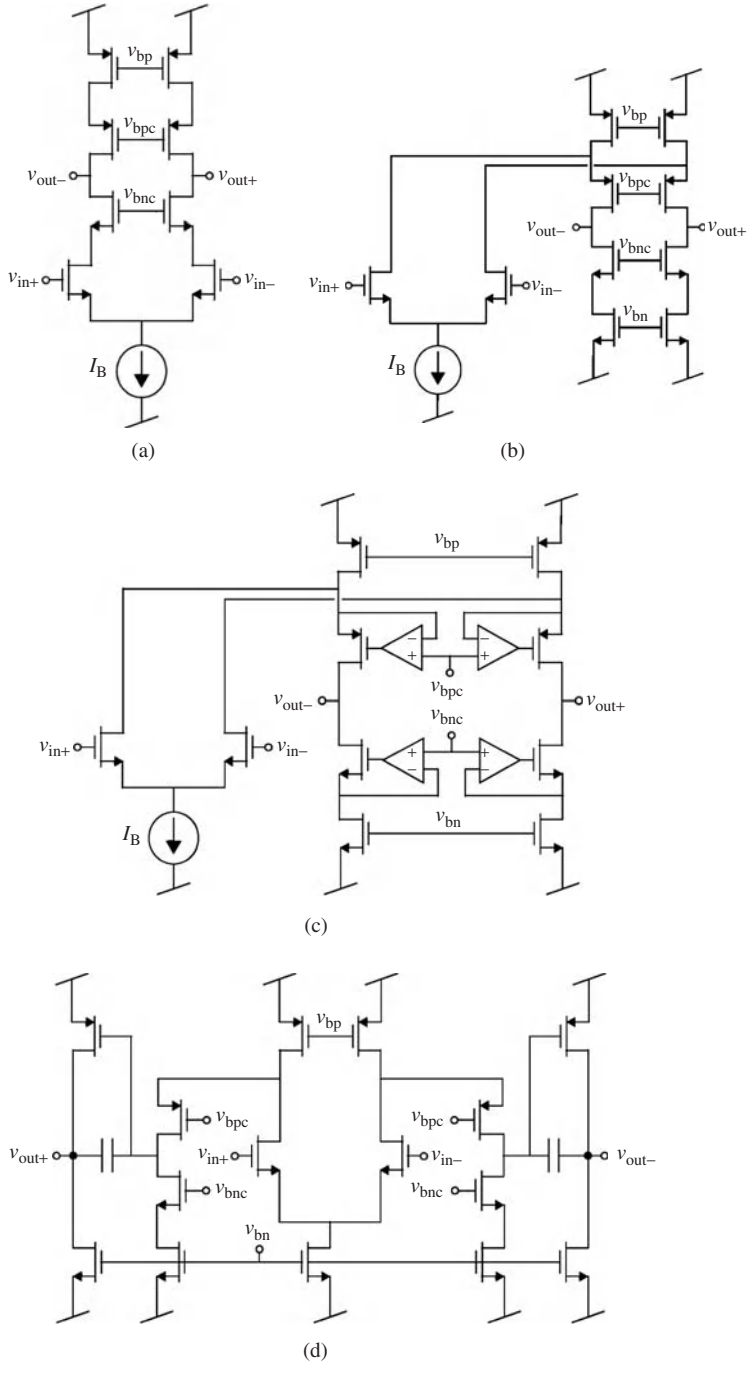


Figure 4.40 Amplifier topologies commonly employed in ΣΔ modulators: (a) telescopic amplifier, (b) folded cascode amplifier, (c) folded cascode amplifier with gain boosting, and (d) two-stage amplifier with Miller compensation.

that the inner feedback loop does not degrade the amplifier frequency response, or even make it unstable in closed-loop form.

- *Two-Stage Amplifier with Miller Compensation* (Figure 4.40d). Two-stage amplifiers are capable of providing a high DC gain as well as a large output swing, as the voltage gain is obtained with two amplification stages rather than cascoding. However, their settling behavior becomes more complex than that of single-stage amplifiers—because of the additional poles and zeros that the internal compensation introduces—and they usually result in higher power consumption.

Common-Mode Feedback Networks

Single-ended implementations of ΣΔMs are only rarely seen and the vast majority of reported ΣΔMs—similar to most analog circuits—employ fully-differential circuits because of their reduced sensitivity to even order harmonics and their better power supply rejection ratio. In the case of SC-ΣΔMs, the differential implementation also helps to reduce the power consumption, as, for the same influence of the kT/C noise, the sampling capacitors can be halved compared to the single-ended case thanks to the doubled input signal range. Also, the clock feedthrough and charge injection of the switches cancel to a common-mode signal.

These reasons lead to the use of fully-differential amplifier topologies—as those previously depicted in Figure 4.40—which thus require an additional circuit to set the common-mode component of the output voltage to the appropriate level, that is, a CMFB network. These nets operate by means of *sensing* the common-mode level of the output voltages, *comparing* it with the desired common-mode level V_{CM}, and *adapting* the bias conditions of the amplifier through negative feedback so that $(v_{out+} + v_{out+})/2 \approx V_{CM}$. These tasks can be done either continuously or in a discrete-time manner, leading to CT or SC CMFB nets, respectively.

For the sake of illustration, Figure 4.41 shows an alternative version of the folded cascode amplifier in Figure 4.40b using a pMOS differential input pair, together with

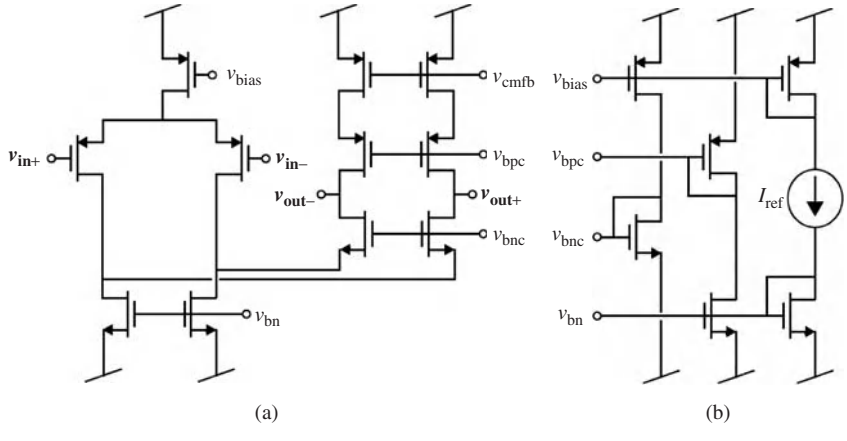


Figure 4.41 Folded cascode amplifier: (a) core circuit and (b) bias circuit.

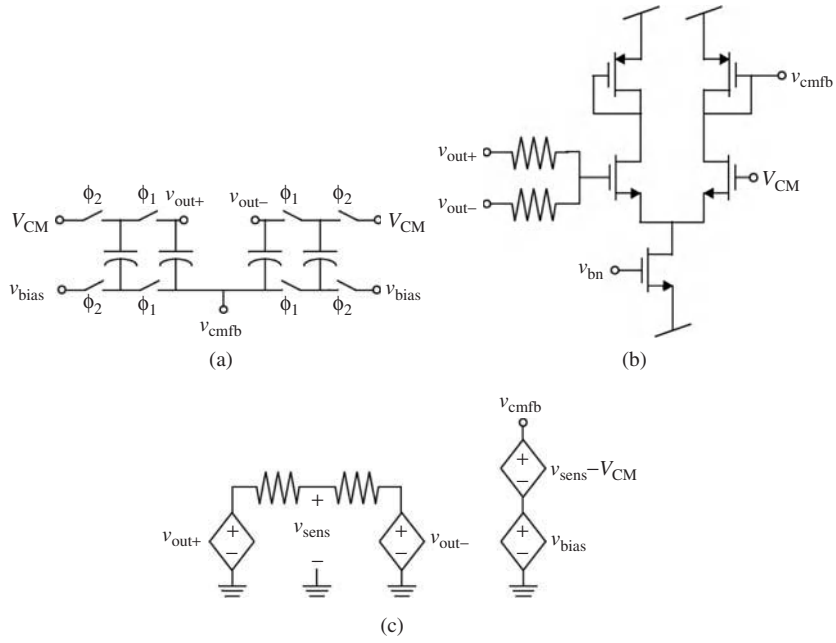


Figure 4.42 Alternative implementations of the common-mode feedback: (a) SC circuit, (b) CT circuit, and (c) ideal circuit for simulation purposes.

its bias circuit. Figure 4.42a and b, respectively, depict the DT or CT alternatives for implementing the corresponding CMFB net. Note from the SC CMFB net in Figure 4.42a that the common-mode voltage of amplifier nodes v_{out+} and v_{out-} is sensed and compared with V_{CM} through capacitors that switch according to the nonoverlapping clock phases ϕ_1 and ϕ_2 of the SC-$\Sigma\Delta$M. In the case of CT CMFB nets, as shown in Figure 4.42b, the output common-mode voltage is sensed through resistors and compared with V_{CM} using a differential pair.

SC CMFB nets are often preferred in SC-$\Sigma\Delta$Ms to CT counterparts, as their design is straightforward, they lead to very small area overhead, and have no static power consumption. Conversely, although designing a CT CMFB net is usually not difficult, it leads to additional power consumption and involves a static inner feedback loop with sufficient gain that affects the dynamic response of the overall amplifier.

For the sake of completeness, Figure 4.42c shows an ideal CMFB net that can be employed together with the amplifier in Figure 4.41 for simulation purposes. Note that operation principle of the CMFB net is just macromodeled using voltage-controlled sources.

Characterization of the Amplifier in AC

Figure 4.43 shows the HSPICE netlist of the folded cascode amplifier in Figure 4.41. The included transistor-level sizing corresponds to an amplifier designed for an SC-$\Sigma\Delta$M

```
** NETLIST EXAMPLE FOR THE CHARACTERIZATION OF AN AMPLIFIER

.prot
.LIB  './umc_13/fdk/VB04_PB/Models/Hspice/L130E_HG_RVT33_V131.lib' TT
.unprot

* AMPLIFIER DESCRIPTION
.subckt folded_cascode vin+ vin- vout+ vout- vdd vss
* Amplifier core
M1   vc1   vin+   vc  vc   P_HG_33_L130E w=17.73u  l=0.65u  m=6
M2   vc2   vin-   vc  vc   P_HG_33_L130E w=17.73u  l=0.65u  m=6
M3   vc    vbias  vdd vdd  P_HG_33_L130E w=49.42u  l=1.14u  m=10
M4   vc1   vbn    vss vss  N_HG_33_L130E w=15.00u  l=2.33u  m=11
M5   vc2   vbn    vss vss  N_HG_33_L130E w=15.00u  l=2.33u  m=11
M6   vout- vbnc   vc1 vss  N_HG_33_L130E w= 4.75u  l=0.34u  m=3
M7   vout+ vbnc   vc2 vss  N_HG_33_L130E w= 4.75u  l=0.34u  m=3
M8   vout- vbpc   vc3 vdd  P_HG_33_L130E w= 3.21u  l=0.44u  m=6
M9   vout+ vbpc   vc4 vdd  P_HG_33_L130E w= 3.21u  l=0.44u  m=6
M10  vc3   vcmfb  vdd vdd  P_HG_33_L130E w=49.42u  l=1.14u  m=6
M11  vc4   vcmfb  vdd vdd  P_HG_33_L130E w=49.42u  l=1.14u  m=6
* Amplifier biasing
M3B  vbias vbias  vdd vdd  P_HG_33_L130E w=49.42u  l=1.14u  m=1
M4B  vbn   vbn    vss vss  N_HG_33_L130E w=15.00u  l=2.33u  m=1
M6bp vbnc  vbias  vdd vdd  P_HG_33_L130E w=49.42u  l=1.14u  m=1
M6b  vbnc  vbnc   vss vss  N_HG_33_L130E w= 2.56u  l=3.83u  m=1
M8bn vbpc  vbn    vss vss  N_HG_33_L130E w=15.00u  l=2.33u  m=1
M8b  vbpc  vbpc   vdd vdd  P_HG_33_L130E w= 3.12u  l=1.35u  m=1
Iref vbias vbn 48u
* Ideal common-mode feedback net
eop eout+ 0 vout+ 0 1
eon eout- 0 vout- 0 1
rp  eout+ vsens 1
rn  eout- vsens 1
vcm vcm 0 1.65
e1 vcmfb aux vsens vcm 1
e2 aux   0   vbias 0   1
.ends

* OPEN-LOOP CONFIGURATION
vdd  vdd 0 3.3
vss  vss 0 0
vcm  vcm 0 1.65
evin+ vin+ vcm d 0 0.5
evin- vin- vcm d 0 -0.5
vd  d  0 ac
xfolded  vin+ vin- vout+ vout- vdd vss folded_cascode
clp vout+ 0 1.4p
cln vout- 0 1.4p
eout out 0 vout+ vout- 1
Rout out 0 100x

.options post nomod nopage ingold=2 method=gear
* PERFORMED ANALYSIS
.ac dec 50 10 1g
.noise v(out) vd
.print ac v(out) inoise onoise
.tf v(out) vd
.pz v(out) vd
* PERFORMED MEASUREMENTS
.meas ac  Av     find v(out) at=10
.meas ac  f3dB   when v(out)='Av*0.707' cross=1
.meas ac  GB     param='Av*f3dB'
.meas ac  UGBW   when vdb(out)=0 cross=1
.meas ac  phase  find vp(out) when vdb(out)=0
.meas ac  PM     param='180+phase'

.end
```

Figure 4.43 HSPICE netlist for the AC characterization of the folded cascode amplifier in Figure 4.41.

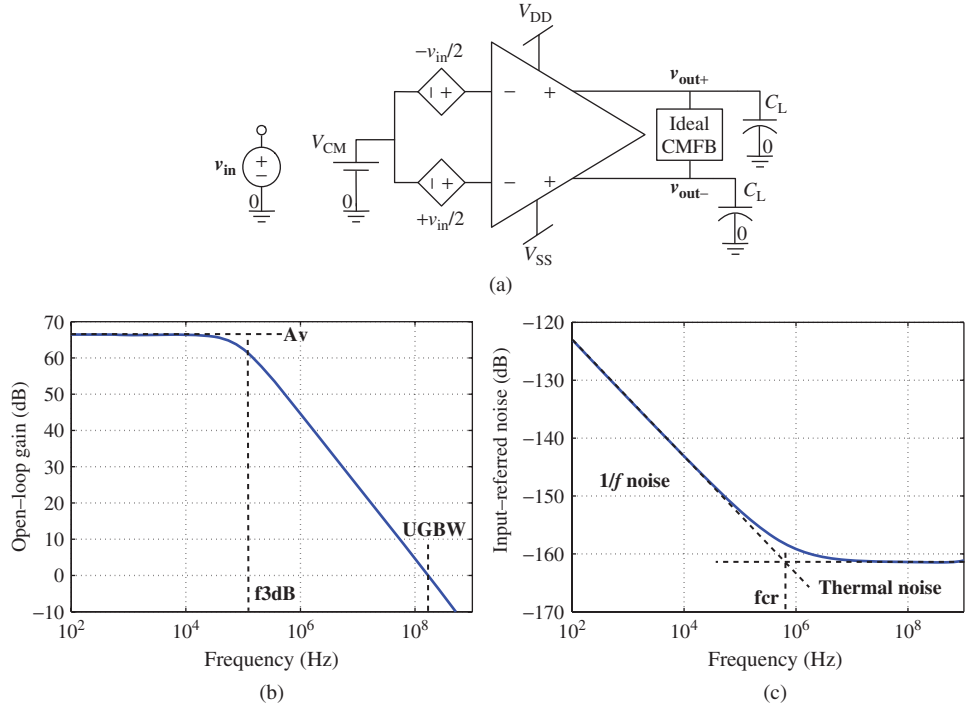


Figure 4.44 Amplifier simulation results obtained from an AC analysis in HSPICE: (a) test-bench circuit, (b) open-loop amplifier frequency response, and (c) input-referred noise PSD.

implemented in a 0.13-µm CMOS process with 3.3-V supply [28]. The netlist includes the simulation test bench depicted in Figure 4.44a for the electrical characterization of the amplifier performance in an open-loop AC analysis, namely:

1. The amplifier frequency response, from which the following features can be directly determined:
   - DC gain (denoted as `Av` in the measurements section at the end of the netlist)
   - Three decibel loss frequency (denoted as `f3dB`)
   - Gain-bandwidth product (denoted as `GB`)
   - Unity-gain bandwidth (denoted as `UGBW`)
   - Phase margin (denoted as `PM`)
2. The PSD of the input-referred amplifier noise, from which the following features can be extracted:
   - $1/f$ noise component
   - Thermal noise component
   - Noise corner frequency

For that purpose an AC differential input signal is applied to the amplifier and its differential output voltage ($v_{out+} - v_{out-}$) is computed. The test bench includes an ideal CMFB

net as that shown in Figure 4.42c, as well as loading capacitors C_L at the amplifier output nodes. The value of C_L (1.4 pF in the netlist in Figure 4.43) corresponds to an estimation of the equivalent amplifier load during the integration phase—C_{eq,ϕ_2} in Equation 2.16.

Figure 4.44b shows the simulated frequency response of the amplifier gain, in which A_v, $f_{3\,dB}$, and UGBW are identified. Their measured values are $A_v = 2100 = 66.4$ dB, $f_{3\,dB} = 81.5$ kHz, and UGBW = 170 MHz. The amplifier GB can be obtained from the $A_v f_{3\,dB}$ product and equals 171 MHz, that is, GB $\approx$ UGBW, which is characteristic of a single-stage amplifier with a clearly dominant pole and results in a large phase margin (PM $= 82.6^{\circ}$ in this case).

Figure 4.44c shows the simulated PSD of the input-referred amplifier noise, in which the flicker and white components can be clearly identified, as well as the corner frequency (around 600 kHz in this case). Note that, as discussed in Section 4.2, the obtained PSD curve can also be used for generating a colored noise data sequence that captures the amplifier noise frequency response and injecting it in transient electrical simulations of the ΣΔM.

Characterization of the Amplifier in DC

Figure 4.45a illustrates a simulation test bench that allows to obtain the amplifier I–V transfer characteristic and easily extract the amplifier transconductance g_m and maximum

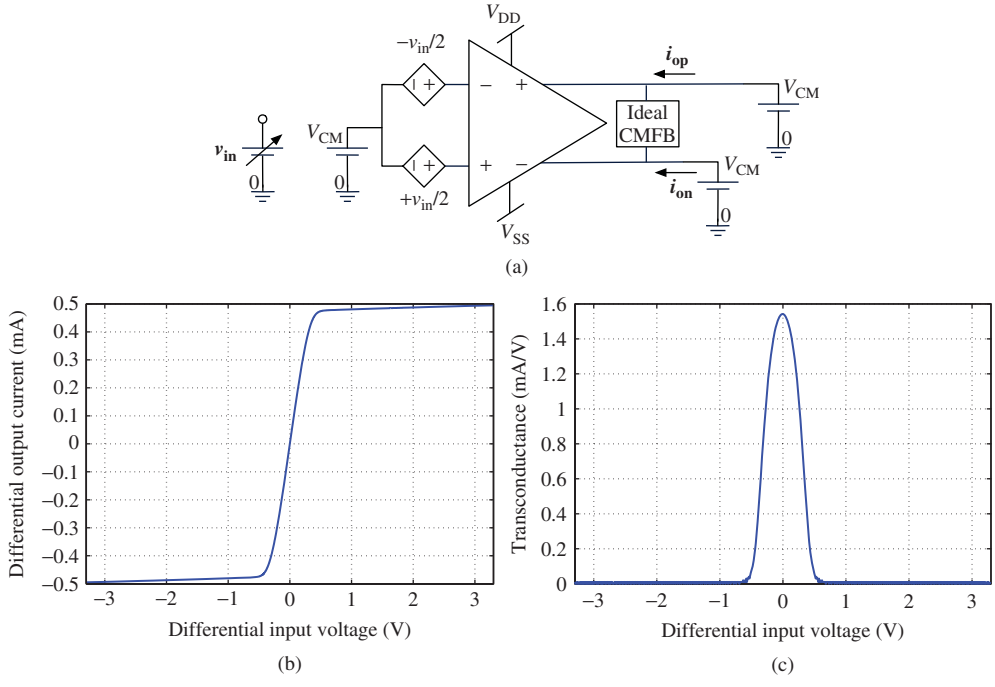


Figure 4.45 Amplifier simulation results obtained from a DC analysis in HSPICE: (a) test-bench circuit and (b) I–V transfer characteristic, and (c) transconductance versus input voltage.

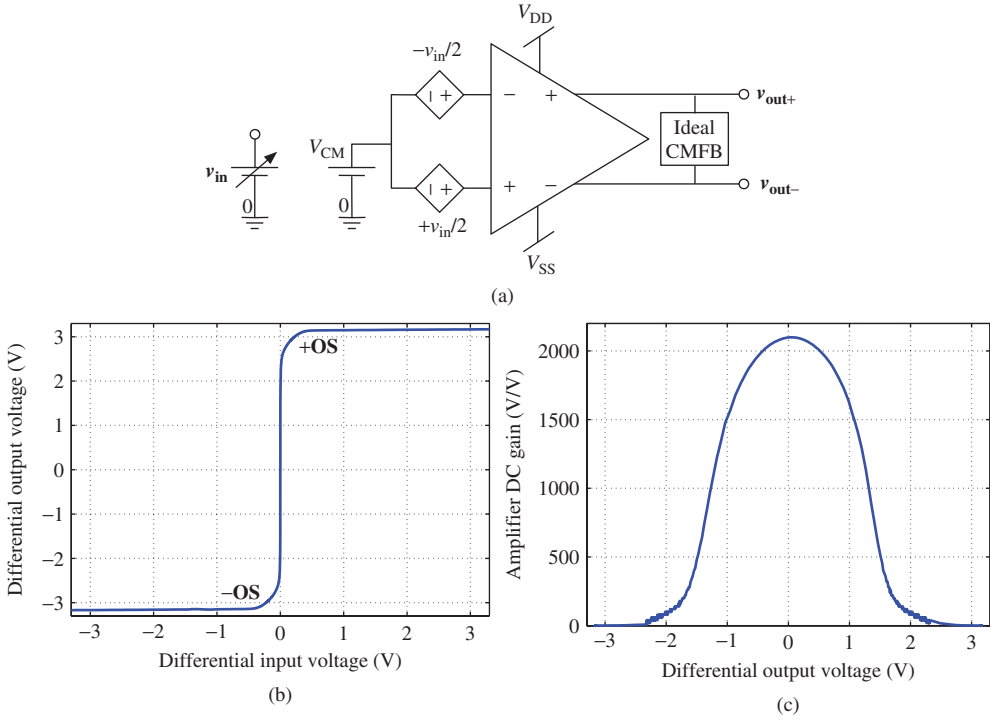


Figure 4.46 Amplifier simulation results obtained from a DC analysis in HSPICE: (a) test-bench circuit, (b) open-loop voltage transfer characteristic, and (c) amplifier gain versus output voltage.

output current I_o. For that purpose a DC differential signal ranging from $-2(V_{DD} - V_{SS})$ to $+2(V_{DD} - V_{SS})$ is applied at the amplifier input, while its output nodes are fixed to the common-mode level through voltage sources whose current is measured.

Figure 4.45b shows the simulated differential output current ($i_{out} = i_{op} - i_{on}$) of the folded cascode amplifier in Figure 4.43 versus the differential input voltage v_{in}. From it the maximum amplifier output current can be measured ($\pm I_o = \pm 0.49$ mA in this case).

The amplifier transconductance DC characteristic can be easily obtained from the slope of the I–V curve in the Figure 4.45b by simply computing di_{out}/dv_{in}. The resulting curve is depicted in Figure 4.45c, from which the amplifier transconductance at the quiescent point g_m can be derived ($g_m = 1.54$ mA V^{-1} in this example).[9]

Characterization of the Amplifier Gain Nonlinearity

Figure 4.46a illustrates a different DC simulation test bench for extracting in this case the amplifier voltage transfer characteristic. From it the amplifier output swing OS and

[9] Note that the measured values for g_m and GB, and the loading capacitor C_L, can be combined to provide an estimation of the amplifier output parasitic capacitor C_o. Given that $GB = g_m/(C_L + C_o)$ in a single-stage amplifier, $C_o = 0.3$ pF for the folded cascode amplifier under consideration.

amplifier DC gain nonlinearity can be easily derived. For that purpose a DC differential signal ranging from $-2(V_{DD} - V_{SS})$ to $+2(V_{DD} - V_{SS})$ is again applied at the amplifier input, but with no loading conditions at its output nodes.

Figure 4.46b shows the simulated differential output voltage ($v_{out} = v_{out+} - v_{out-}$) versus the differential input voltage v_{in} for the folded cascode amplifier in Figure 4.43.

The dependency of the amplifier gain on the output voltage level can be easily obtained from the data contained in the Figure 4.46b, just by computing dv_{out}/dv_{in} from the curve and depicting the results against v_{out} (instead of v_{in}). Figure 4.46c shows the resulting gain curve for the folded cascode amplifier in the example, from which the DC gain at the quiescent point A_v can also be derived ($A_v = 2100$, in accordance to low-frequency AC results in Figure 4.44b).

Finally, note that the amplifier voltage characteristic in Figure 4.46b can be obtained from a relatively coarse DC sweep of the input voltage, whereas a very fine sweep is required to accurately obtain the amplifier gain curve in Figure 4.46c. For the sake of illustration, the DC input voltage was varied from -3.3 to $+3.3$ V in 10-mV steps to obtain the graphical representation in Figure 4.46b, whereas the DC simulation was run again with steps of only 50 μV (but only from -0.3 to $+0.3$ V) to have enough granularity in the v_{out} axis in the results shown in Figure 4.46c.

4.4.3 Design Considerations of Transconductors

As already stated in this book, transconductors are essential building blocks of CT-ΣΔMs. They are used for building Gm-C integrators and also to implement ΣΔM loop-filter coefficients. As an illustration, Figure 4.47 shows two different common examples of CT-ΣΔM implementations. The first one, shown in Figure 4.47a, uses a front-end active-RC integrator, whereas the rest of loop-filter integrators are implemented using Gm-C topologies. This approach is normally chosen because of the better linearity performance achieved by active-RC integrators as compared with Gm-C integrators. If the linearity specification is not very restrictive, purely Gm-C implementations similar to that shown in Figure 4.47b are usually preferred because of their potential faster operation. Note that in both examples depicted in Figure 4.47, loop-filter coefficients are implemented using transconductors.

The main electrical characteristics affecting the performance of transconductors used in ΣΔMs are the following: input/output swing, transconductance value and tuning range, finite DC gain, finite GB, mismatch error, and nonlinearity (usually characterized by the IIP3 parameter). The limit values of these design specifications are derived from behavioral system-level simulations that are further fine-tuned using macromodel electrical simulations as described in Section 4.1. Once these specifications have been clearly identified, a proper circuit topology is chosen and designed at the transistor level in order to meet them.

Obviously, there are plenty of different transconductor circuits that can be used for implementing the loop filter of CT-ΣΔMs. The detailed explanation of all of them is beyond the scope of this book. Instead, the following sections focus on two circuit examples. One of them is suited for the implementation of front-end, Gm-C integrators, while the other one is more appropriate to build Gm-C integrators embedded in the loop filter.

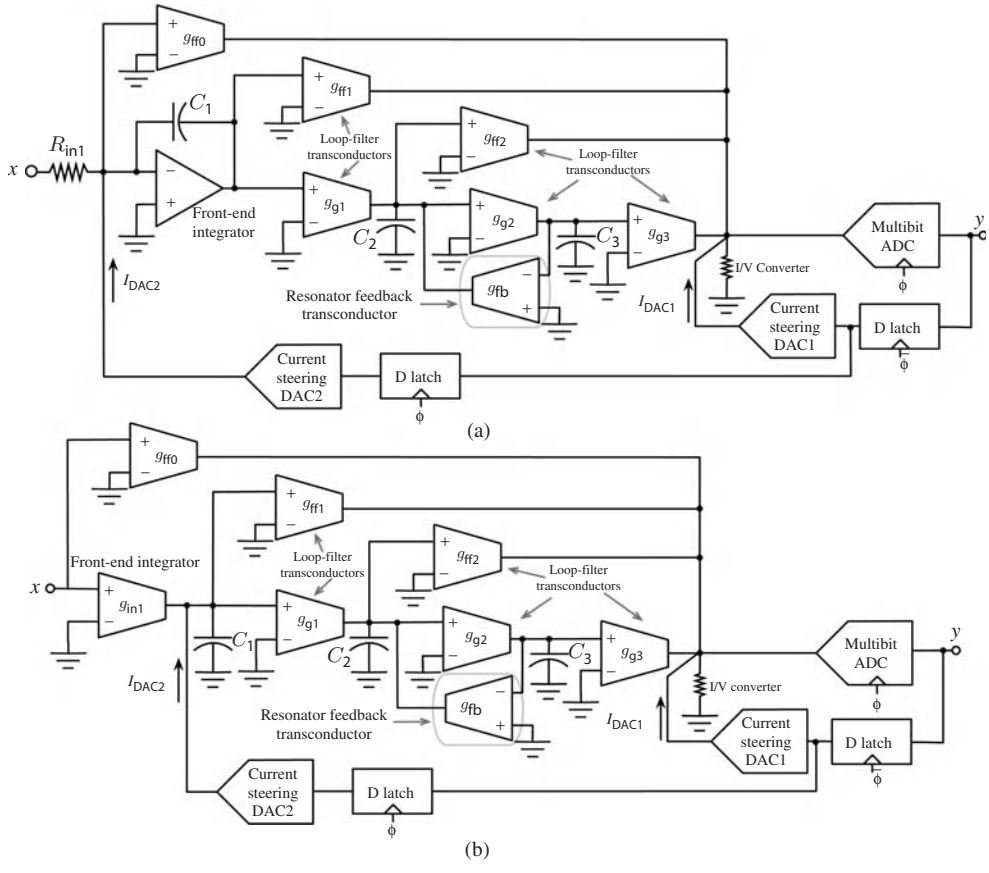


Figure 4.47 Illustrating the use of transconductors in a third-order, single-loop CT-ΣΔM, considering: (a) an active-RC front-end integrator and (b) a Gm-C front-end integrator.

Example of Front-End Transconductor

One of the main limiting factors in open-loop Gm-C integrators is their poor linearity. Indeed, this is especially critical at the input node of ΣΔMs because harmonic distortion caused by the front-end V/I transconductor is directly translated to the digital domain without any attenuation. Thus, special emphasis should be put to design the front-end transconductor with high enough linearity, because it can severely degrade the performance of the modulator.

Figure 4.48 shows an example of front-end transconductor [29]. The circuit combines gain boosting techniques with resistive source degeneration in order to increase the linearity of the transconductance. Note that in this example, nMOS transistors are used in the input differential pair because a triple-well option is available in the technology where the transconductor is going to be integrated. This way, the MOS body effect can be avoided by simply connecting the source terminal to the substrate terminal. Otherwise, if

the triple-well option is not available, pMOS transistors should be used for the differential input pair.

As an example, let us consider that the circuit is placed at the front-end of a cascade 3-2 CT-ΣΔM with a front-end stage similar to that shown in Figure 4.47b, and that this modulator is intended to digitize 20-MHz signals with 12-bit effective resolution. According to behavioral simulations carried out in SIMSIDES, the following specifications are obtained for the front-end transconductor:

- Finite DC gain: >70 dB
- Differential input/output swing: 0.3 V
- Third-order nonlinearity: > -86 dB.

In addition to the above specifications, input-referred thermal noise must be kept low enough not to degrade the effective resolution (12 bit) required by the modulator. Moreover, according to the synthesis of the ΣΔM loop-filter coefficients, in this example the nominal input V/I transconductance was chosen to be 850 μA and the integration capacitance is 3.65 pF.

In order to design the circuit in Figure 4.48 to cope with the above specifications, basic OTA design equations for the nominal transconductance, finite DC gain, SR, and GB are usually derived by hand [30] and used as initial point for the electrical design. Transistors are biased and sized in the saturation region using these by-hand equations and considering the voltage limits specified by the input/output swing. This initial design is fine tuned by electrical simulation in order to satisfy the required specifications with minimum power consumption. In addition, as in other essential ΣΔM building blocks, the circuit must be simulated considering the effect of technology corners and mismatch deviations. The circuit in Figure 4.48 was designed in a 0.13-μm CMOS technology to cope with the aforementioned specifications, achieving a DC gain of 73.8 dB, a maximum

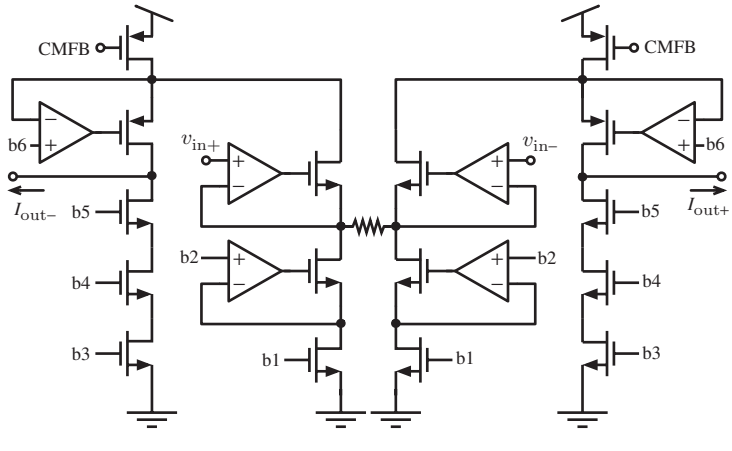


Figure 4.48 Example of ΣΔM front-end transconductor with enhanced linearity [29].

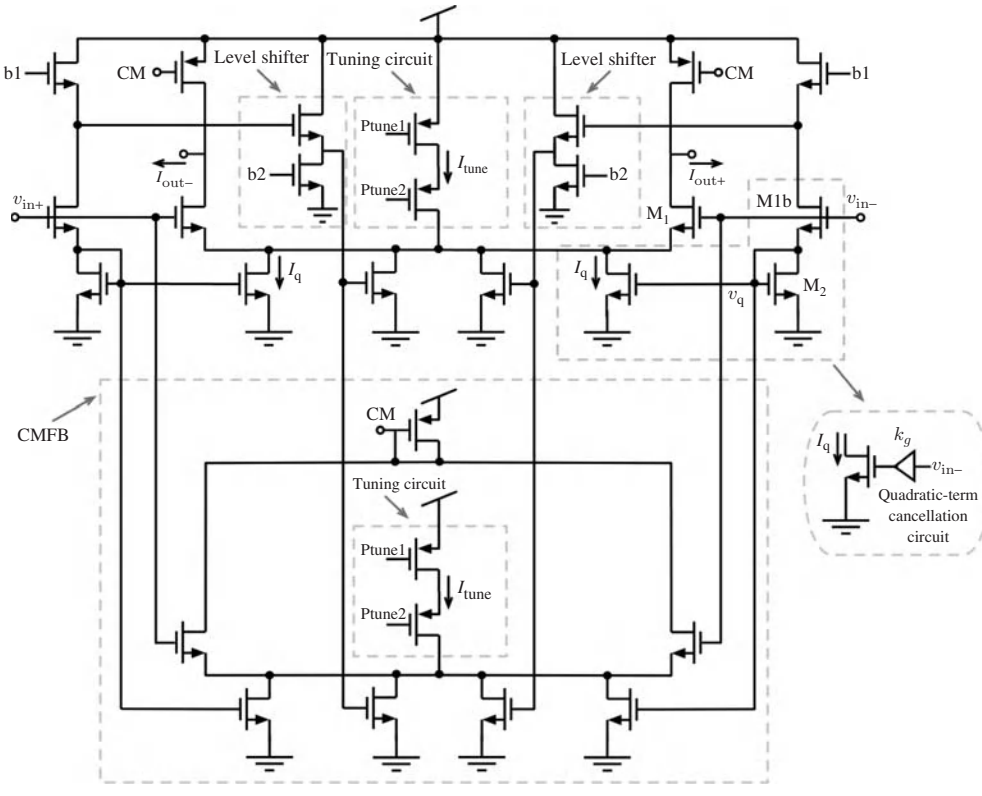


Figure 4.49 Example of ΣΔM loop-filter transconductor with quadratic-term cancellation technique [29].

differential input/output swing of 0.3 V and $HD_3 = -89$ dB, considering a single 1.2-V supply voltage [29].

Example of Loop-Filter Transconductor

Figure 4.49 shows an example of a transconductor (with its corresponding CMFB circuit) that can be used for building the loop-filter coefficients of CT-ΣΔMs [29]. High-speed operation is achieved by using only feed-forward paths. These paths introduce a high frequency zero that extends the frequency range. The circuit uses a quadratic-term cancellation technique in order to improve the linearity of the transconductance. The basic idea behind this technique is to include an additional tail current source having its current in a quadratic dependence on the input signal, as conceptually illustrated in Figure 4.49. It can be shown that the quadratic term of the drain current of the differential-pair transistors can be cancellated if $k_2 = k/k_g^2$, where k_g is a gain factor that takes into account any signal scaling that may be applied to the input signal, and k and k_2 stand for the large-signal transconductances of the input differential-pair transistor (M_1) and the extra MOS transistor added (M_2), respectively.

A very critical design consideration that must be taken into account in the design of CT-ΣΔM loop-filter circuit elements is their tunability. Thus, transconductors must be designed so that they can be tuned in order to keep the loop-filter time constants (C/g_m) unchanged over technology parameter variations. To this purpose, a circuit tuning strategy must be incorporated in the loop-filter transconductors in order to make the design of CT-ΣΔMs feasible and robust against circuit element tolerances. In the example shown in Figure 4.49, the transconductance can be tuned through the tail bias current I_{tune}. Note that, for voltage headroom reasons, this current source is connected between the positive supply voltage and the common source node. Thus, a variation of I_{tune}—through bias voltages named $P_{tune(1,2)}$—changes the value of the nominal transconductance, without significantly affecting the linearity of the transconductor.

Note that for the tuning to be effective, each transconductance of the ΣΔM is usually implemented as the parallel connection of unit transconductors and a number of Monte Carlo simulations are required in order to guarantee that the performance of the transconductors—particularly the linearity—is not affected by the mismatch. In this example, a unit transconductance of 25 μA V^{-1} was used, while the values of I_{tune} ranged from 5 μA to 25 μA.

As stated in the previous section, the linearity and gain specifications are not so demanding as in this case of front-end integrators. For instance, considering the same modulator specifications described in the previous section, the electrical specifications of the loop-filter transconductors will vary depending on their position in the modulator chain. The most restrictive specifications are the following:

- Finite DC gain: >50 dB
- Differential input/output swing: 0.3 V
- Third-order nonlinearity: > −56 dB.

As an illustration, Figure 4.50 shows some typical characteristics obtained by electrical simulation of transconductors using the test-bench schematic shown in Figure 4.51.

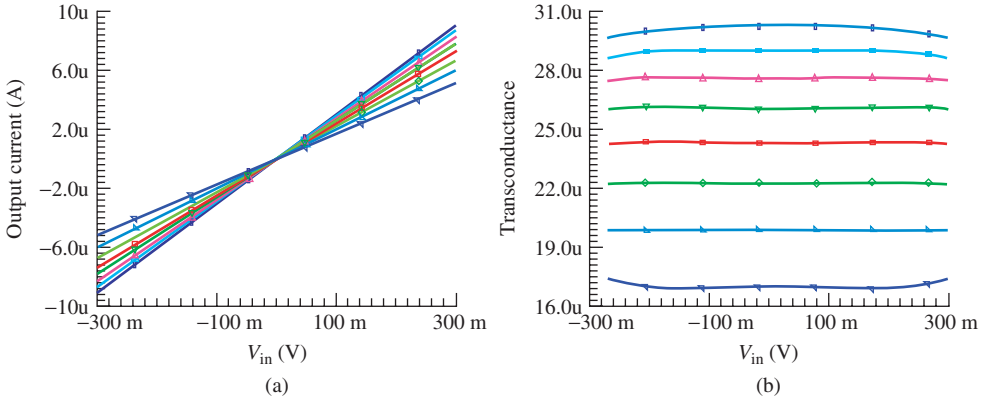


Figure 4.50 Illustrating the electrical performance of ΣΔM loop-filter transconductors: (a) output current versus input voltage for different values of the tuning current I_{tune} and (b) transconductance versus input voltage. I_{tune} ranges from 5 μA to 23 μA.

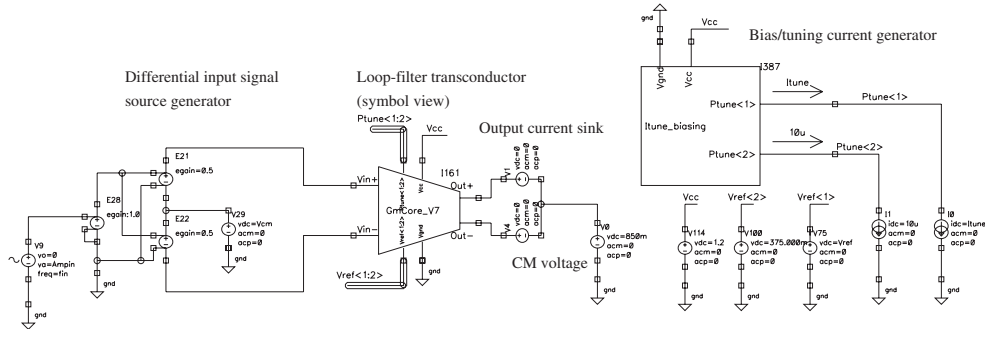


Figure 4.51 Test-bench schematic used for characterizing the ΣΔM loop-filter transconductors. Note that this test bench can be used for DC, AC, and transient analysis. Alternatively, a load circuit can be connected at the output of the transconductor in order to emulate a circuit environment closer to the actual situation of loop-filter transconductors embedded in a CT-ΣΔM.

Figure 4.50a shows the output current as a function of the input current for different values of I_{tune} ranging from 5 μA to 23 μA in this example. The linearity of the transconductance can be calculated by computing the derivative of this I−V characteristic. This is represented in Figure 4.50b. Following this procedure, the maximum transconductance variation for each of the values of I_{tune} can be calculated. In this example, a deviation below 2% is obtained in all cases within the specified input voltage range (0.3 V).

Figure 4.52 shows the frequency response of a Gm-C integrator made up of a unit transconductor (with a nominal transconductance of 25 μA V^{-1}) and the unit integration capacitor (3.65 pF in this example). Note that the phase error at unity gain of the integrator is below 1°, while the high-frequency poles are placed at a much higher frequency than the GB of the integrator. This is due to the fact that the transconductor in Figure 4.49 is essentially composed of feed-forward paths, which facilitates a higher speed of operation.

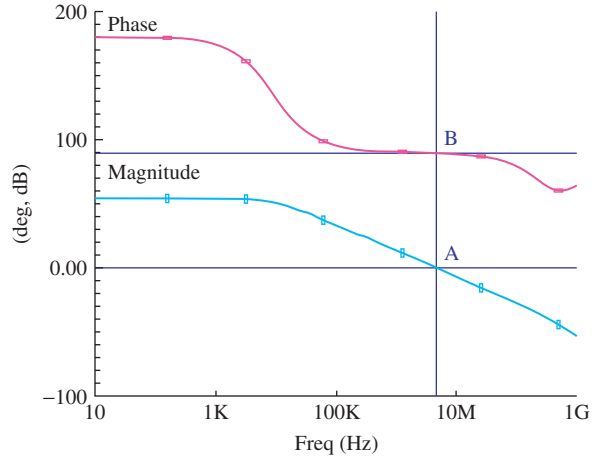


Figure 4.52 Phase and gain response of the loop-filter (unit) transconductor in Figure 4.49.

4.4.4 Design Considerations of Comparators

Comparators are essential building blocks of ΣΔ ADCs. These circuits are used for building the quantizer embedded in the modulator. Because of their position in the modulator loop, the comparator specifications are not very demanding in most practical cases, as circuit errors are attenuated by NTF in the same way as the quantization noise. However, the design of main performance limitations (i.e., offset, hysteresis, and comparison time) must be carefully taken into account in order to optimize the performance of ΣΔMs. Typical static specifications require an offset and a hysteresis in the order of tens of millivolts, and a maximum comparison time of 1/4 of the clock period, that is, around one half of the time interval corresponding to the strobe phase.[10] These specifications can be achieved by using the so-called regenerative latch topologies described in the following subsection.

Regenerative Latch-Based Comparators

The majority of comparator circuits used in ΣΔMs are based on a DT positive-feedback regenerative latch that is built by cross-coupling a pair of inverters as conceptually depicted in Figure 4.53a [31]. The inverters amplify the differential input voltage $v_{i+} - v_{i-}$ to obtain the saturated differential output $v_{o+} - v_{o-}$, according to the characteristics drawn in Figure 4.53b. During the so-called reset phase (ϕ_r high), the differential input is stored in the input sampling capacitors C and the circuit is driven to the central state Q_0 (Figure 4.53b). During the comparison or *strobe* phase (ϕ_a high), the differential input is retrieved, forcing an initial state either on the right ($v_{i+} > v_{i-}$) or on the left ($v_{i+} < v_{i-}$) of Q_0. From this initial state, the action of positive feedback forces the output to evolve either toward Q_H (for $v_{i+} > v_{i-}$) or toward Q_L (for $v_{i+} < v_{i-}$). In both cases, the dynamic evolution around the central point is realized at a high speed due to the action of positive feedback.

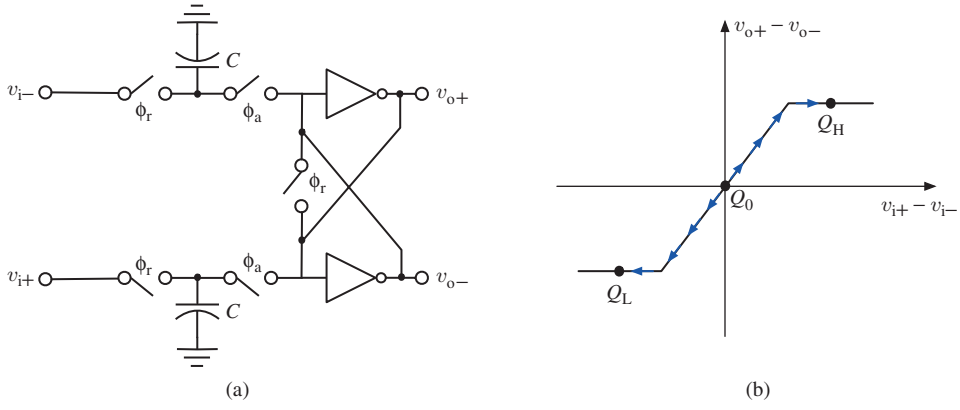


Figure 4.53 Positive-feedback regenerative latch: (a) conceptual schematic and (b) illustration of dynamic trajectories in the input–output characteristic.

[10] Strobe phase is the clock phase in which the comparator is active, that is, it is comparing the input signals. This phase is sometimes referred to as amplification phase or simply comparison phase.

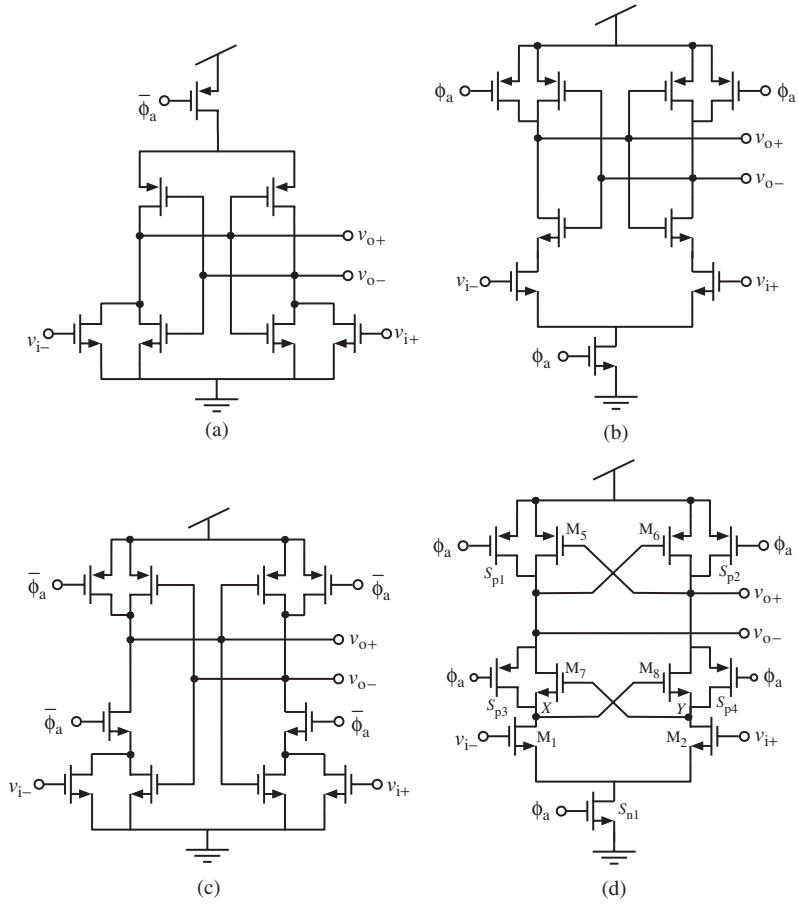


Figure 4.54 Examples of CMOS latches frequently used in ΣΔMs: (a) [32], (b) [33], (c) [34], and (d) [35].

Figure 4.54 shows some regenerative latch CMOS circuits [32–35] that are commonly used for implementing comparators in state-of-the-art ΣΔMs, all of them based on the conceptual model shown in Figure 4.53a. In practice, the static resolution of all these circuits is limited by dissymmetries between the positive and the negative branches of the fully-differential circuit, as well as other second-order circuit phenomena such as the kick-back noise on the integrator connected to the comparator. Thus, a preamplifier is usually placed at the input of the regenerative latch in order to improve the static resolution of these kinds of comparators [36].

Figure 4.55 shows a typical example of a CMOS regenerative latch comparator with preamplifier.[11] It consists of a pMOS differential input pair ($M_{1,2}$), a CMOS regenerative

[11] Some comparators used in SC-ΣΔMs with feed-forward paths include an SC network at the input of the preamplifier to merge with SC adders. These SC networks may be used also for improving the preamplifier performance and for implementing reconfiguration of the number of bits of the embedded quantizer [9].

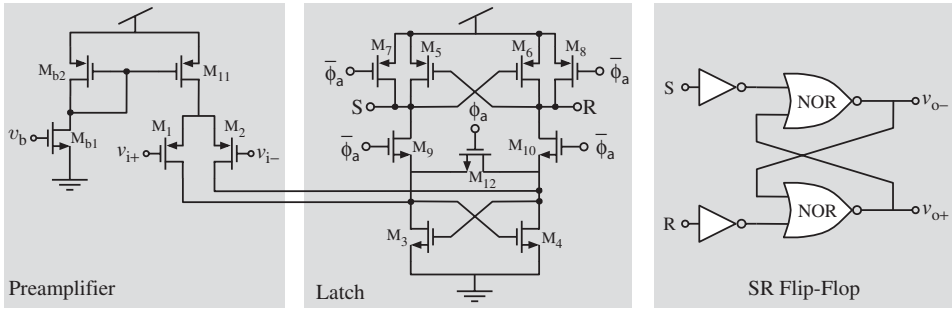


Figure 4.55 Example of a typical CMOS regenerative latch comparator with preamplifier.

latch circuit, and a set-reset (SR) flip-flop to store the voltage provided by the latch. The latch circuit is composed of an nMOS flip-flop ($M_{3,4}$) with a pair of nMOS switches ($M_{9,10}$) for strobing and an nMOS switch (M_{12}) for resetting, and a pMOS flip-flop ($M_{5,6}$) with a pair of pMOS precharge switches ($M_{7,8}$).

The operation of the circuit in Figure 4.55 is as follows: During clock phase $\bar{\phi}_a$ (reset phase), the latch is in reset mode and the input differential pair injects a current proportional to the differential input voltage ($v_{i+} - v_{i-}$), which generates an initial voltage imbalance across the on-resistance of switch M_{12}. The voltage difference across M_{12} is amplified when the latch becomes enabled during the comparison (amplification) phase ϕ_a, achieving a very fast comparison time due to the action of positive feedback.

Design Guidelines

The design of the comparators used in ΣΔMs is carried out according to the high-level specifications extracted from behavioral simulations as described in Chapter 3. The main design parameters included in the behavioral models, which degrade the performance of ΣΔMs, are essentially two static parameters, namely, offset and hysteresis. In addition, the transient response of the comparator must be fast enough to complete their operation within the comparison clock phase. Therefore, the analog sections[12] of the comparator (i.e., the preamplifier and the latch) must be carefully sized accordingly to these specifications.

The preamplifier aims to fulfill several goals, namely, to obtain a high DC gain in order to reduce the comparator input-referred offset, a low kick-back noise, and high speed, while keeping low parasitic input capacitances. Therefore, the increase of the preamplifier DC gain should not be done exclusively by increasing the size of the input differential-pair transistor—in order to increase g_m—for a given bias current, because this would also increase the input parasitic capacitance. This design trade-off can be solved using a preamplifier schematic with a high output resistance similar to that shown in Figure 4.56. In this circuit, transistors M_4 and M_5 are biased in the ohmic region and hence used as resistors to increase the output resistance of the preamplifier.

Another important design trade-off that takes place in the latch involves the required mismatch specifications (which have a direct effect on the static resolution) and the

[12] The SR flip-flop is a digital circuit that in the majority of cases can be designed using minimum sizes.

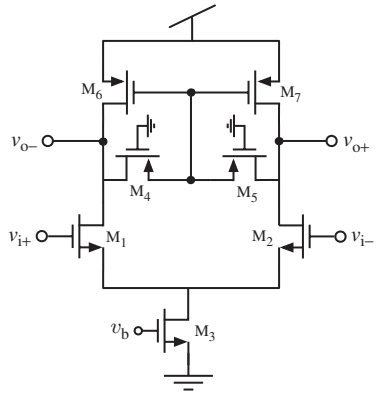


Figure 4.56 Single-stage preamplifier with ohmic-biased transistors (M_4 and M_5) to increase the output resistance.

comparator speed. This way, using minimum sizes for the inverter transistors (M_{3-6} in Figure 4.55) minimizes the MOS parasitic capacitances, thus benefiting the transient response. However, minimum-sized transistors increase the sensitivity to technology process variations.

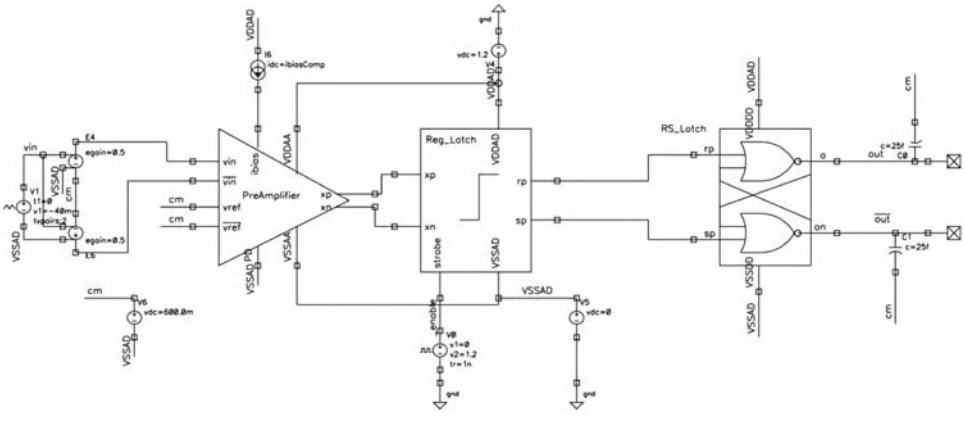
A minimum length should be used for the switches involved (M_{7-10} in Figure 4.55) in order to reduce the switch on-resistance as well as the effect of charge injection. Note that the input voltage difference is sensed across switch M_{12}, and hence its on-resistance becomes critical for a correct operation of the comparator. Indeed, the performance of the latch is very sensitive to the size of M_{12}, that has motivated some designers to use more robust topologies.

This is the case of the latch shown in Figure 4.54d [35]. The operation of this circuit is governed by one clock phase, named ϕ_a in the figure. Thus, when strobe or amplification clock signal (ϕ_a) triggers to a logic one, switch S_{n1} turns on and transistors $M_{1,2}$ will process the differential output of the preamplifier. After a short time, one of the input differential-pair transistors will turn off depending on the sign of the input voltage imbalance, thus creating a differential voltage between nodes X and Y. Meanwhile, M_7 and M_8 will turn on and, later, the cross-coupled scheme will start working, leading to a fast regeneration of the initial voltage imbalance [9].

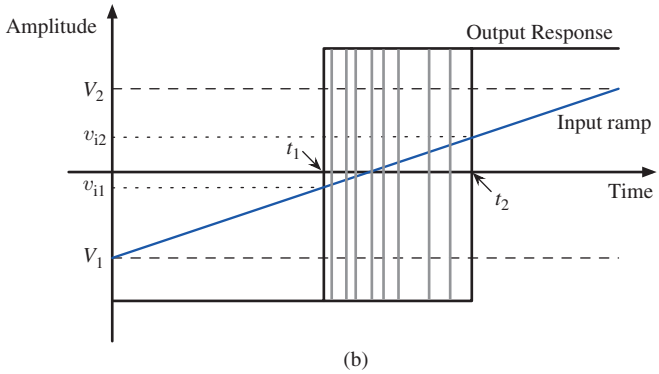
Although regenerative latch comparators are very fast, the comparison time and the dynamic resolution must be properly characterized at transistor level. The following sections give some practical ways to do this task.

Characterization of Offset and Hysteresis Based on the Input-Ramp Method

Comparator offset and hysteresis can be characterized in electrical simulations using the test-bench circuit shown in Figure 4.57a. A slow ramp-waveform input signal is applied to the comparator, so that the offset and hysteresis parameters can be extracted from the output voltage waveform as conceptually depicted in Figure 4.57b. Note that in order to obtain a more precise value of both performance metrics (offset and hysteresis), the



(a)



(b)

Offset computation

$$v_{of} = \frac{v_{i1} + v_{i2}}{2}$$

Hysteresis computation

$$v_h = \frac{v_{i1} - v_{i2}}{2}$$

Figure 4.57 Characterization of offset and hysteresis based on the input-ramp method: (a) Cadence Virtuoso test-bench schematic and (b) conceptual output response and computation of offset and hysteresis parameters.

voltage limits of the input ramp should be approximately of the same order as the specified offset and hysteresis.

Let us consider an example in which the target specifications for the offset and the hysteresis are ± 10 and ± 20 mV, respectively. In order to take into account the impact of mismatch and technology parameter variations, a large number of simulations considering all technology corners as well as a Monte Carlo analysis should be carried out. As an illustration, Figure 4.58 shows some typical output waveforms obtained in HSPICE for the characterization of the offset and hysteresis.

The method based on the input ramp does not allow for memory or hysteresis issues in the comparator operation because the voltage value of the input signal is always increasing (or decreasing) in amplitude, which is not common in practice. A more realistic test bench consists of using a sinusoidal or triangular input signal. This way, the input signal is forced to take alternating signs in order to allow for memory effects [9].

Another disadvantage of the ramp-waveform, test-bench method is that a slower ramp is needed to obtain the required accuracy in the characterization of the offset and the

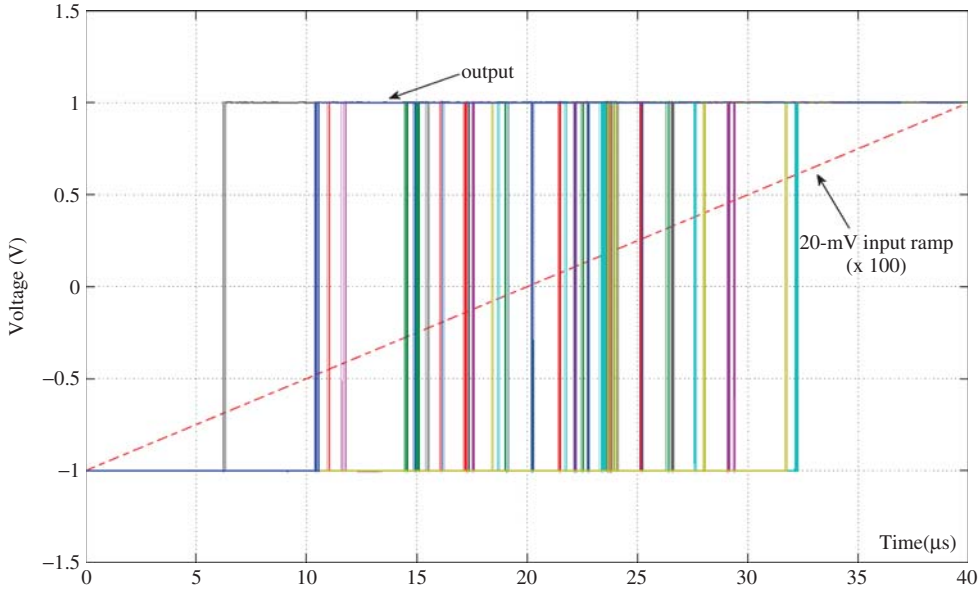


Figure 4.58 Illustrating the electrical characterization of hysteresis in regenerative latch comparators. Monte Carlo simulation in HSPICE.

hysteresis. The slower the input ramp is, the longer the transient simulation, which might lead to long CPU times, as typically 100–200 simulations are needed for an accurate Monte Carlo simulation.

Characterization of Offset and Hysteresis Based on the Bisectional Method

A more efficient way of characterizing the input offset of the comparator is based on the so-called *bisectional* method. In this method, graphically illustrated in Figure 4.59,

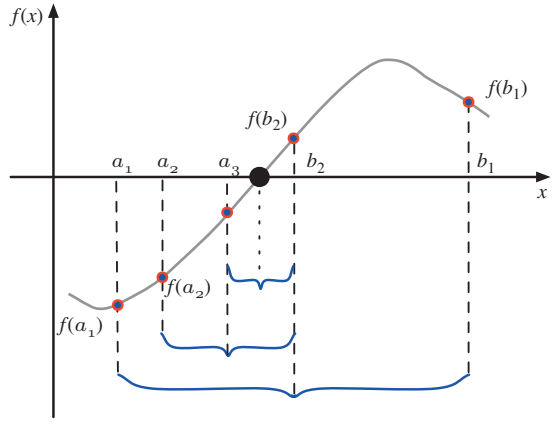


Figure 4.59 Graphical illustration of the bisectional method algorithm.

a root-finding algorithm works by repeatedly halving an interval and then selecting the subinterval in which the root exists [37]. Given two points, a and b, such that $f(a)$ and $f(b)$ have opposite signs, the intermediate value theorem says that $f(x)$ must have at least one root in the interval $[a, b]$ as long as $f(x)$ is continuous. Thus, the bisection algorithm is then applied to the subinterval where the sign change occurs until a solution under a defined tolerance is obtained. It can be shown that the number of iterations required to reach convergence with a tolerance error ϵ is $n = \log_2[(b-a)/\epsilon]$ [37].

Figure 4.60a shows the test bench used in Cadence Virtuoso Schematic editor for the comparator offset computation. The preamplifier and the regenerative latch are placed inside a feedback loop, in which the preamplifier input signal is controlled by a block that implements the bisectional algorithm described earlier. A Verilog-A code [1], shown in Figure 4.60b, is used for this purpose. The input of this block is the differential output of the regenerative latch, so the algorithm basically selects the next input voltage based on the comparator output, following the procedure depicted in Figure 4.59. Note that a delayed version of the strobe clock signal triggers the Verilog-A block corresponding to the bisectional algorithm. This time delay must be longer than the response time of the regenerative latch, but shorter than half of the cycle of the comparator strobe.

Characterizing the Comparison Time

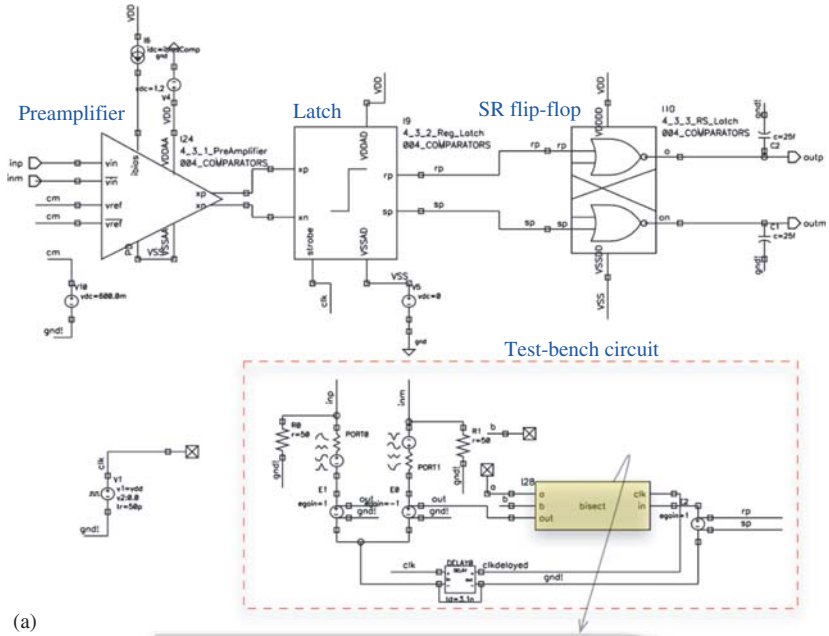
Figure 4.61 illustrates the electrical simulation method that is usually followed to characterize the resolution speed and the comparison time of comparators. An input step of ΔV is applied, where ΔV is the specified static resolution of the comparator (10 mV in this example). The comparison time, also referred to as *resolution time*, is defined in this test bench as the time interval between the time instant in which the strobe phase becomes low at the end of the comparison phase and the time instant when the output reaches the corresponding logic level, that is, V_{DD} for a positive input step or V_{SS} otherwise. Monte Carlo and technology corner variations need to be checked and the worst-case values are usually taken as the design specification.

4.4.5 Design Considerations of Current-Steering DACs

As already discussed earlier in this book, the DACs used in the feedback path of ΣΔMs are mainly implemented using SC and SI or current-steering circuit techniques.[13] The design of SC feedback DACs involves taking into account considerations for the design of switches and capacitors according to the design criteria previously stated in this chapter and in Chapter 2. Indeed, the switches and capacitors used for building SC DACs are embedded in the SC integrators and quantizers (Figure 4.9). Thus, SC DACs do not require special attention apart from the design issues related to switches already discussed.

In contrast, CS DACs—mostly used in CT-ΣΔMs—are essential building blocks affecting the performance of the modulator because of several nonideal circuit phenomena already discussed in Chapters 2 and 3, namely clock jitter error, transient response (and its effect on the excess loop delay error), and linearity (due to device mismatch of unit current

[13] The difference between SI and CS techniques is discussed later.



(a)

(b)

```
// VerilogA for GVdP_comparator, bisect, veriloga

`include "constants.vams"
`include "disciplines.vams"
module bisect(in, clk, a, b, out);

   input in; voltage in;
   input clk; voltage clk;
   output a; voltage a;
   output b; voltage b;
   output out; voltage out;

   parameter real va = -0.1;
   parameter real vb = 0.1;
   parameter real vth = 0.6;
   parameter integer dir = +1 from [-1:+1] exclude 0;
                    // if dir=+1, rising clock edge triggers
                    // if dir=-1, falling clock edge triggers
   parameter real td = 0 from [0:inf);  // output delay
   parameter real tt = 0 from [0:inf);  // output transition time

   real      val;
   real      vbl;
   real      vout;
   real      level;
analog begin
    @(cross(V(clk) - vth, dir)) begin
       val=V(a);
       vbl=V(b);
       vout=V(out);
       level = (V(in)-0);
       if (level<0)
        val=vout;
       else
        vbl=vout;
    end
    @(initial_step) begin
       val=va;
       vbl=vb;
    end
   vout=(val+vbl)/2;
   V(a) <+ transition( val, td, tt );
   V(b) <+ transition( vbl, td, tt );
   V(out) <+ transition( vout, td, tt );
end
endmodule
```

Figure 4.60 Test bench used for the characterization of the comparator offset based on the bisectional algorithm method: (a) Cadence Virtuoso Schematic and (b) Verilog-A code for the bisectional algorithm.

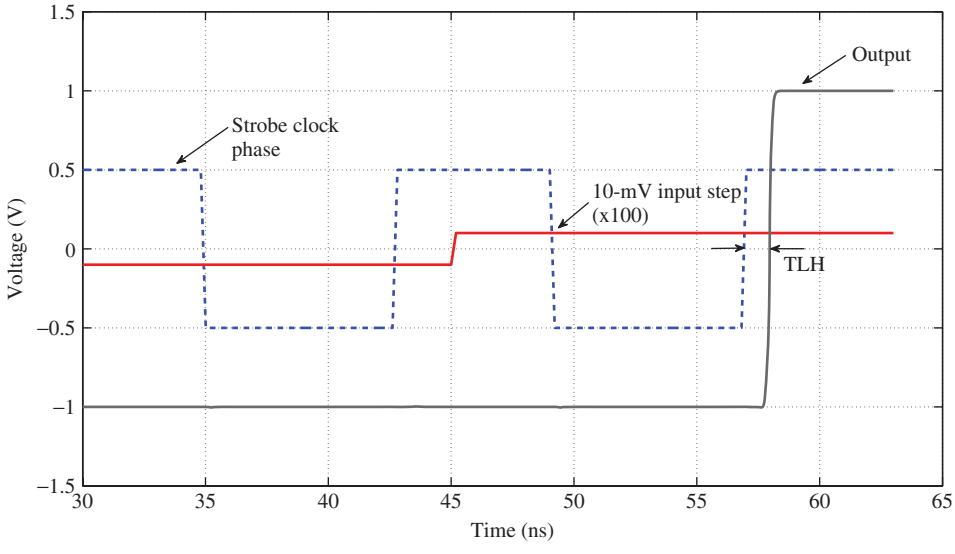


Figure 4.61 Illustrating the characterization of comparison time in regenerative latch comparators.

sources). Indeed, CS DACs are specifically suited for wideband CT-ΣΔMs because of their potential benefits in terms of high-speed operation and the convenience to interface with both Gm-C and active-RC CT-ΣΔMs [38]. This is illustrated in Figure 4.62 that shows a conceptual schematic of the input summing node of a CT-ΣΔM. Note in Figure 4.62a that the output current of the feedback CS DAC is *naturally*[14] added with the output current of the loop-filter Gm-C transconductor. In the case of active-RC implementations, the current-mode adding operation takes place at the *virtual ground* input node of the amplifier as illustrated in Figure 4.62b.

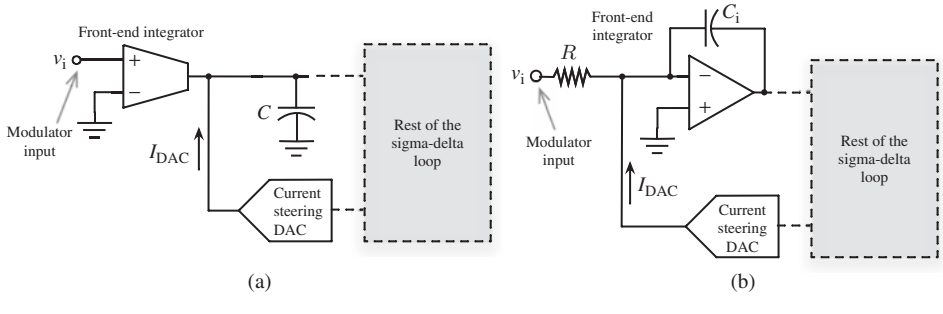


Figure 4.62 Illustrating the connection of CS DACs at the input summing node of (a) Gm-C ΣΔMs and (b) active-RC ΣΔMs.

[14] As will be discussed later, the high impedance output node of a Gm-C transconductor may be not suited to *inject* the feedback current provided by a CS DAC because of two main reasons. On the one hand, large signal swings reduce the voltage headroom required to keep the current cell transistors in the saturation region. On the other hand, the high impedance node causes that small current errors result in large voltage errors on the capacitors.

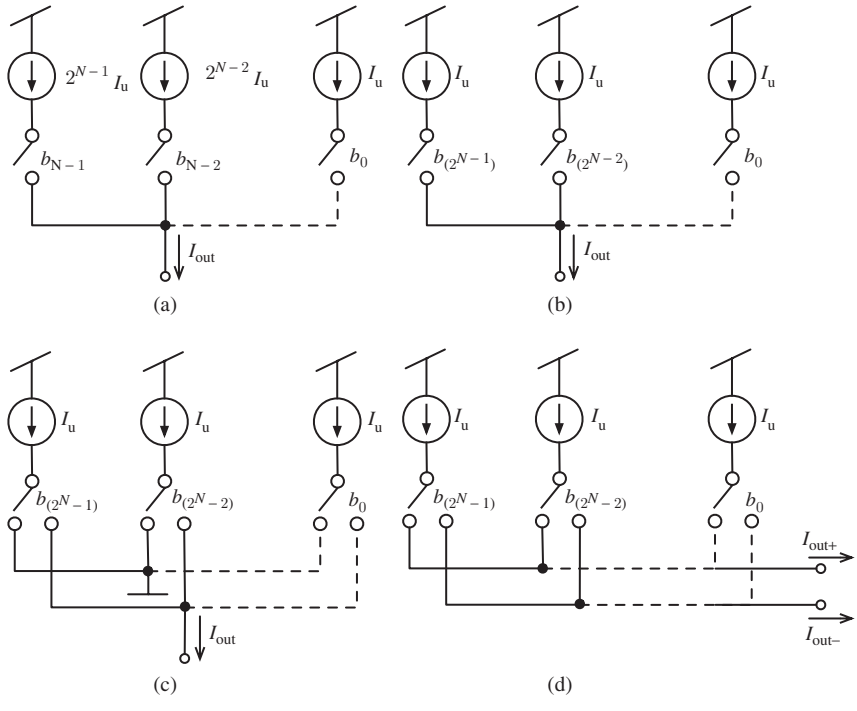


Figure 4.63 Conceptual schemes of switched-current DACs: (a) based on binary-weighted current cells, (b) based on thermometer-coded current cells, (c) single-ended current-steering DAC, and (d) fully-differential current-steering DAC.

This section pays attention to the main design criteria usually taken into account in the design of CS DACs—from the circuit-level perspective—focusing on those issues that must be considered during the electrical (transistor-level) design of ΣΔMs. Special emphasis is put on their main performance metrics of CS DACs, showing how to characterize these metrics with practical test benches implemented in SPICE-like simulators.

Fundamentals and Basic Concepts of CS DACs

Figure 4.63a and b show two conceptual schemes of a CS DAC [12], where the output current is obtained by adding a number of switched unit current cells together. The switches are controlled by the DAC input bits, which are in turn the outputs of the quantizer embedded in the ΣΔM loop filter. Thus, if a binary code is used, N binary-weighted scaled current cells are required, as shown in Figure 4.63a. Note that, although this solution requires the minimum number of elements, they are more sensitive to device element mismatch, mainly because of the very different values of the cell currents. For that reason, a thermometer-coded CS DAC similar to that shown in Figure 4.63b is usually chosen to implement the feedback DACs of CT-ΣΔMs. This approach *relaxes* the matching requirements at the price of exponentially increasing the number of unit elements, that is, the number of current cells and switches.

Note that the circuit schemes shown in Figure 4.63a and b cannot be implemented in practice because of to the current glitches appearing when current cells are left open. For that reason, the scheme shown in Figure 4.63c—commonly referred to as *current-steering*—is usually preferred. In this approach, the currents provided by the cells are *steered* or *redirected* either to the DAC output summing node or to a dummy low-impedance node, by means of complementary switches.

The scheme in Figure 4.63c is particularly useful in fully-differential implementations, which is one of the most common situations in practice. In this case, conceptually depicted in Figure 4.63d, unit current cells are steered either to the positive or to the negative output current, depending on the corresponding input digital code.

Practical Implementation of CS DACs in ΣΔMs

Figure 4.64a shows an alternative implementation to Figure 4.63d, which provides a fully-balanced (complementary) differential output current with maximized power efficiency. As conceptually highlighted in the figure, two different kinds of current cells, p-type and

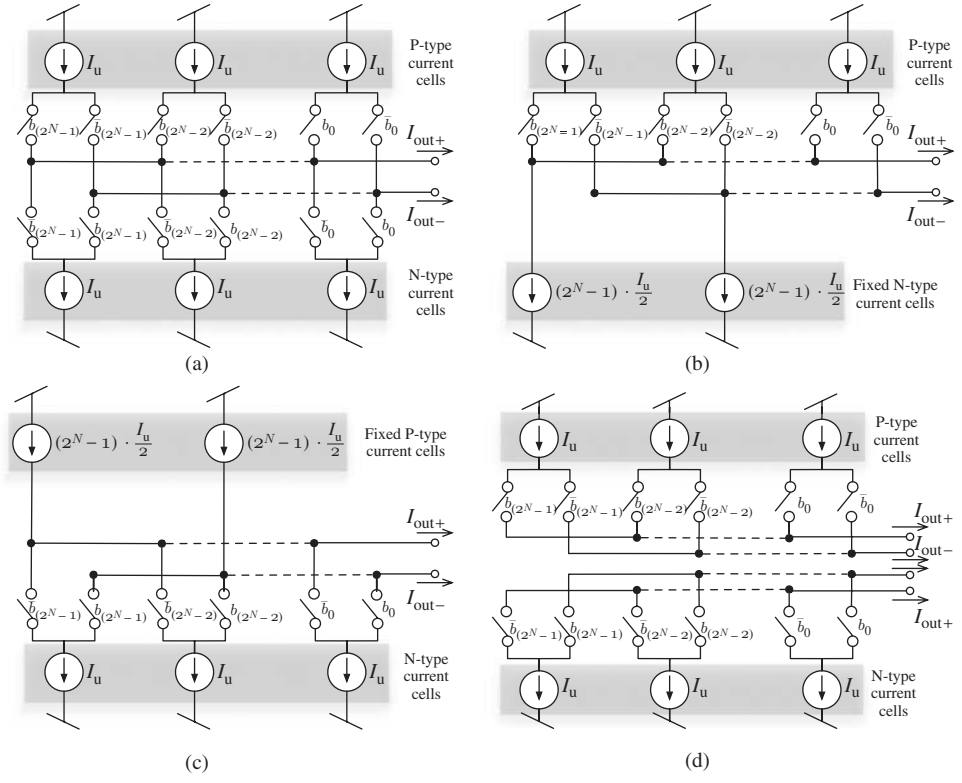


Figure 4.64 Conceptual schemes of fully-balanced (complementary) current-steering DACs considering: (a) switchable p-type and n-type current cells, (b) fixed n-type current cells, (c) fixed p-type current cells, and (d) different output nodes for p-type and n-type current cells. P-type and n-type current cells are sometimes called current sources and current sinks, respectively.

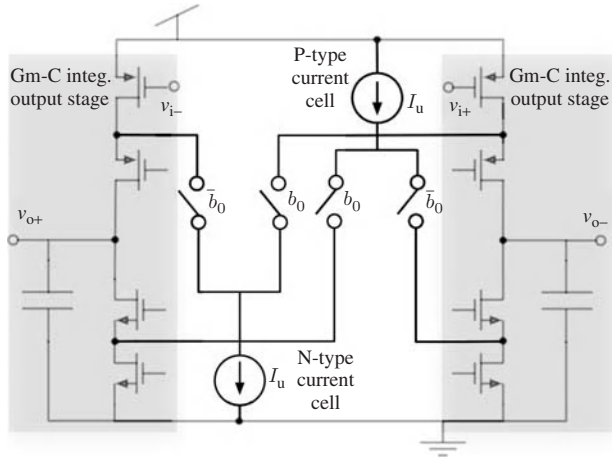


Figure 4.65 Illustrating the use of CS DACs with p-type and n-type current cells connected at different nodes of Gm-C integrators as proposed in [39].

n-type, are used for providing the current sources and sinks needed. The main problem of this DAC topology is in achieving the required voltage headroom for both types of current cells.

In order to palliate this limitation, diverse DAC topologies have been proposed to relax the voltage headroom specification. The most common approach consists of keeping one of the current cell types fixed and the other switchable. This is conceptually illustrated in the CS DAC architectures depicted in Figure 4.64b and c. The former, which uses a fixed n-type current cell and switchable p-type current cells, allows a larger headroom for the (switchable) p-type current cells. The opposite situation is given in Figure 4.64c. In both cases, the power efficiency is reduced to 50%. The choice of one of these two topologies will be conditioned by the voltage headroom requirement for each type of cells in a given design.

Figure 4.64d shows a CS DAC architecture that keeps a power efficiency of 100% with relaxed specifications for the voltage headroom of both types of current cells. In this approach, p-type and n-type current cells are connected to different nodes, which are strategically chosen to provide the highest headroom voltage for each type of cell. This approach was successfully proposed and implemented by the authors in [39]. In this case, illustrated in Figure 4.65, the p-type current cells are connected to the source node of the n-type cascode transistors of the Gm-C integrator output stage, while the p-type current cells are connected to the corresponding source terminal of the nMOS cascode transistors, maximizing thus the headroom for both types of cells. Note also that the source terminal of the cascode transistors provides a low-impedance node to the CS DAC output, thus reducing its output swing and making its design more robust [39].

Current Cell Circuits, Error Limitations, and Design Criteria

The operation of CS DACs described above is degraded in practice by the effect of circuit implementation of switches and current cells. Among others, the main limiting

factors[15] affecting the performance of CS DACs in ΣΔMs are caused by random errors because of to device mismatch and systematic errors due to output impedance of the current cell, thermal gradients, layout edge effects, and incomplete settling error.

Therefore, current cells are designed to satisfy a number of design specifications that are derived from behavioral simulations. Some of these specifications involve the value of the output resistance, the transient response, mismatch, output swing, etc. Among others, there is usually a strong trade-off between the required cell mismatch and the settling time. Thus, in order to relax the mismatch requirements without penalizing the linearity of the modulator, linearization techniques—such as DEM or digital calibration—are used as described in Section 1.6.

Essentially, the following requirements should be taken into consideration for designing the current cells of CS DACs in CT-ΣΔMs:

- Reduce the mismatch among unit current cells.
- Increase the current cell output impedance, making use of cascode stages.
- Ensure that transistors operate in the saturation region for the specified integrator output swing. This is easier to achieve when ΣΔM loop-filter integrators are active-RC, as the signal swing is limited at the virtual ground node of the OTA.
- Reduce the clock feedthrough error, which can be controlled by keeping parasitic capacitances as low as possible. An additional strategy to reduce this error consists of minimizing the voltage difference between the switch on-state and the switch off-state. This technique—often referred to as *soft-driving*[16]—reduces also the overshoot currents due to the clock-signal transitions and allows switches to operate also in the saturation region, thus acting as cascode transistors for further boosting the current cell output impedance.
- Reduce the CS DAC noise contribution to the required IBN specification of the modulator. Note that both thermal and flicker noise sources associated with current cells are summed at the ΣΔM input node, thus constituting an ultimate limiting factor.

In order to cope with the aforementioned design criteria with optimized power consumption, a huge number of different current cell topologies have been proposed in the open literature for the implementation of CS DACs used in CT-ΣΔMs. The circuit topologies span from basic cells—such as a single transistor—to simple cascode, regulated cascode, etc. The detailed description of all the different CS DAC current cell approaches goes beyond the scope of this book.

CS 4-bit DAC Example

As a case study, let us consider a CS 4-bit DAC based on the circuit topology shown in Figure 4.64c. It consists of two fixed p-type current sources and 15 $(2^4 - 1)$ n-type switchable current cells that are controlled through simple nMOS switches by a

[15] Design considerations of switches described in Section 4.4.1 must also be taken into account. In most practical cases, simple nMOS/pMOS switches satisfy the required specifications of CS DACs.

[16] One of the main disadvantages of using soft-driving techniques is that an additional level-shifter circuit is required to generate the switch control voltages.

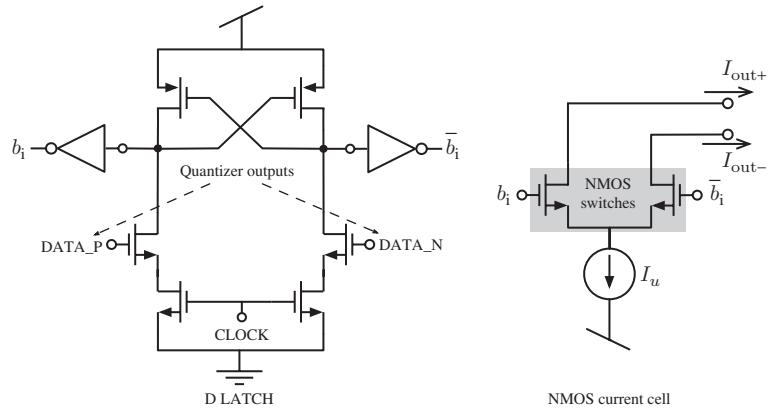


Figure 4.66 Illustrating the use of D-latches to connect the ΣΔM quantizer output and the nMOS switches of CS DACs.

thermometer-coded input data. These data are entered into the DAC through a set of D-latches driving the switch gates as illustrated in Figure 4.66.

The 4-bit CS DAC was used for implementing the NRZ feedback DAC of a fifth-order cascade 3-2 Gm-C CT-ΣΔM intended to digitize 20-MHz signals with 12-bit effective resolution, with a sampling frequency of 240 MHz [29]. These modulator specifications were mapped onto building-block specifications using SIMSIDES as described in Section 3.6.2. As a result, the following specifications were defined for the CS DAC:

- Output resistance: > 12 MΩ
- Settling time: < 0.5 ns
- Mismatch error: $< 0.15\%$ LSB.

Among others, there is a strong design trade-off between the required mismatch error and the settling time error. This trade-off was relaxed in this design example by using DEM linearization techniques that reduced the mismatch requirement to $< 0.6\%$ LSB. The unit current cell was chosen to be $I_u = 48$ μA, which leads to 360 μA (i.e., $7.5I_u$) pMOS current sources.

Another important limitation of the CS DAC in this design example was caused by the headroom voltage required for the pMOS current cells. As the common-mode voltage of the modulator was set to 0.75 V (forced by a design requirement of the loop-filter transconductors) and the FS voltage range was 0.3 V, the headroom voltage for the pMOS and the nMOS current sources resulted in 0.3 and 0.6 V, respectively. In order to satisfy all these specifications, the gain-boosted pMOS current source similar to that shown in Figure 4.67a is considered, while regulated cascode topologies, depicted in Figure 4.67b, can be used for the nMOS current cells. These cells can be designed to satisfy the aforementioned specifications, while dissipating a reasonable power consumption of 0.49 and 0.1 mW, respectively, for the pMOS and the nMOS cell [29].

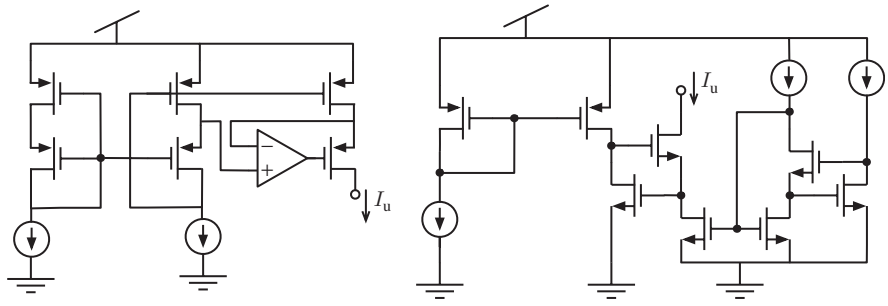


Figure 4.67 Examples of CS DAC current cells: (a) pMOS gain-boosted current cell and (b) nMOS regulated cascode current cell.

4.5 Auxiliary ΣΔM Building Blocks

As stated earlier in this chapter, in addition to the basic building blocks used for implementing the loop filter and the quantizer of a ΣΔM, there are other subcircuits required to make an IC operative. This section overviews the most important blocks, showing their fundamental schematics as well as some practical considerations to take into account in their design.

4.5.1 Clock-Phase Generators

As any sampled-data system, the operation of ΣΔMs is governed by a clock signal. Usually, CT-ΣΔMs require one clock signal and its complementary version. In contrast, SC-ΣΔMs need to divide the clock period into several time intervals or *clock phases* that are generated by a digital circuit, commonly referred to as *clock-phase generator*.

Phase Generation

Figure 4.68a and b show two well-known digital circuits frequently used for generating the clock phases needed in SC-ΣΔMs. Essentially, the operation of both circuits is based on the use of bistable flip-flops to generate several periodic signals from an input signal, also named master clock. The scheme in Figure 4.68a consists of a feedback loop made up of two NAND gates, each one connected in series with a cascade of inverters. The number and sizes of these inverters provide the required clock-phase delays and nonoverlapping intervals.

Figure 4.68b shows an alternative implementation of the clock-phase generator, which is also based on the use of flip-flops [22]. In this case, there is no *global* feedback loop and the different clock phases are generated by connecting D-type latches in cascade to provide the required delays and inversions.

In some high-frequency applications that demand a very high-precision, high-speed, and low-jitter, clock-phase scheme, the master clock signal used in Figure 4.68a and b

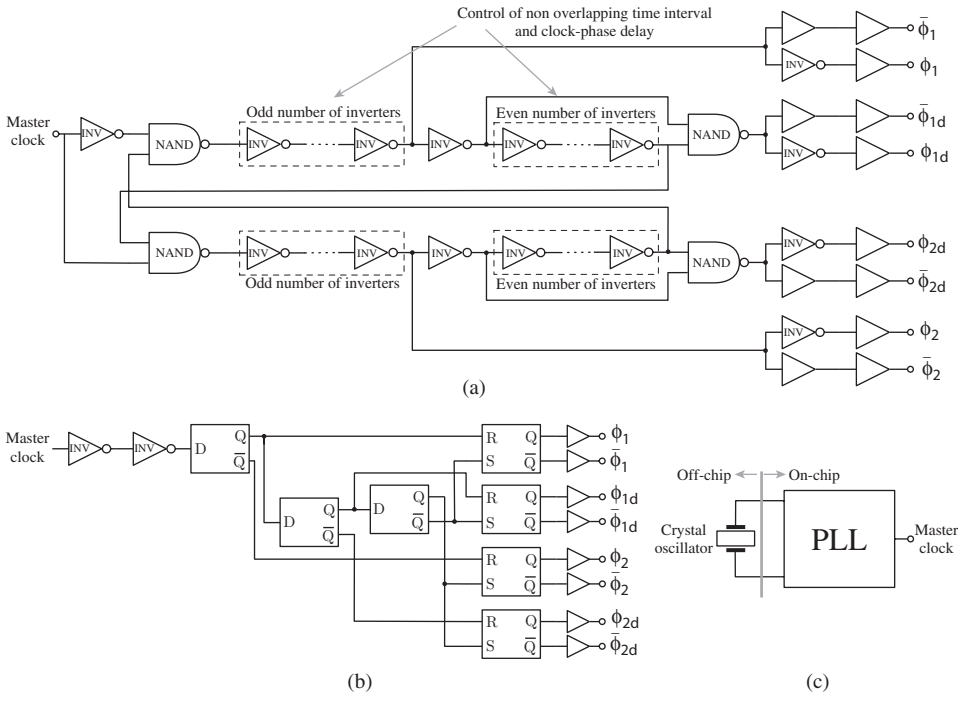


Figure 4.68 Conceptual schemes of clock-phase generators frequently used in SC-ΣΔMs: (a) feedback loop made up of two NANDs and a cascade of inverters, (b) D-latch-based generator, and (c) master clock generated by an (on-chip) PLL frequency synthesizer and an (off-chip) crystal oscillator. The use of on-chip PLLs allows generation of high-frequency, high-precision (low-jitter) clock signals, usually at the price of increasing the power consumption.

is synthesized by an on-chip phase-locked loop (PLL) and an off-chip well-controlled crystal oscillator [40, 41], as conceptually depicted in Figure 4.68c.[17]

Note that both clock drivers in Figure 4.68a and b generate two nonoverlapped clock phases—ϕ_1, ϕ_2—that control the sampling and integration operations of SC integrators. Delayed versions of the clock phases—ϕ_{1d}, ϕ_{2d}—are also generated in order to attenuate the error caused by signal-dependent charge injection produced during the turn-off process of input switches in SC integrators [43]. Complementary versions of the four clock phases—that is, $\bar{\phi}_1$, $\bar{\phi}_2$, $\bar{\phi}_{1d}$, *and* $\bar{\phi}_{2d}$ needed to control CMOS switches and some other ΣΔM subcircuits (such as latch-based comparators)—are also synthesized by properly combining inverters and digital buffers as illustrated in Figure 4.68a and b.

Phase Buffering

All generated clock-phase signals need to be properly driven by a buffer tree similar to that conceptually shown in Figure 4.69. This way, the differences in capacitive load

[17] Some CT-ΣΔMs using clock signals in the gigahertz range use off-chip ultra-low-jitter signal sources and 180°-hybrid circuits to generate the master clock and its complementary signal [42].

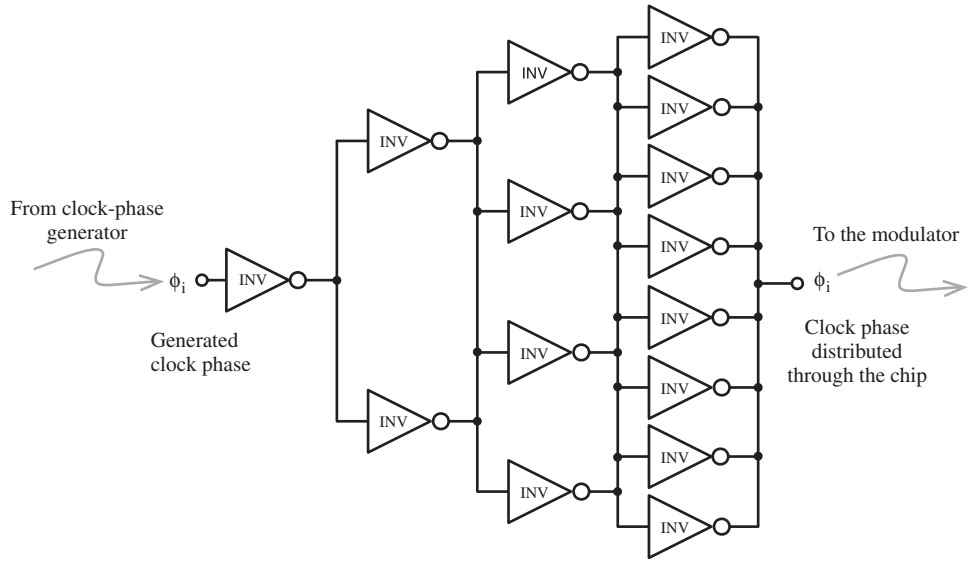


Figure 4.69 Example of buffer tree used for driving clock phases. The number and dimensions of inverters depends on the fan-out requirements of each clock phase.

among all phases become equalized. This is very important in practice because, if clock phases are not properly equalized, the different load capacitances connected at each clock phase will have a direct influence on the delays (and nonoverlapping intervals) among the phases, and may thus destroy the clock scheme generated by the circuits in Figure 4.68a and b, and consequently the operation of the SC-ΣΔM.

In order to design the clock-phase buffer tree, the parasitic capacitance loading each clock phase has to be accurately calculated. This piece of information can be extracted from electrical simulations by summing up the input parasitic capacitances of all sub-circuits (essentially CMOS switches and digital gates) to which the clock phases are connected.

Phase Distribution

As clock phases are used in many different parts of SC-ΣΔMs, these signals need to be routed through the entire chip. To this purpose, a U-shaped bus—conceptually depicted in Figure 4.70a—is used, where each clock phase is isolated by implementing a Faraday cage with two ground walls at each side of the routed signals, as illustrated in Figure 4.70b. Note that each clock phase (ϕ_i) is closely routed with its complementary phase ($\bar{\phi}_i$). Both clock phases are surrounded by ground (GND in Figure 4.70b) strips of the same metal as that used for the clock phases. The whole bus is covered by the same ground above and below the routed phases with plates implemented at the upper and lower metal layers, respectively.

Figure 4.70c shows a lumped LCR equivalent circuit that takes into account the circuit parasitics of the actual transmission line in Figure 4.70b, in which the values of L_T, C_T,

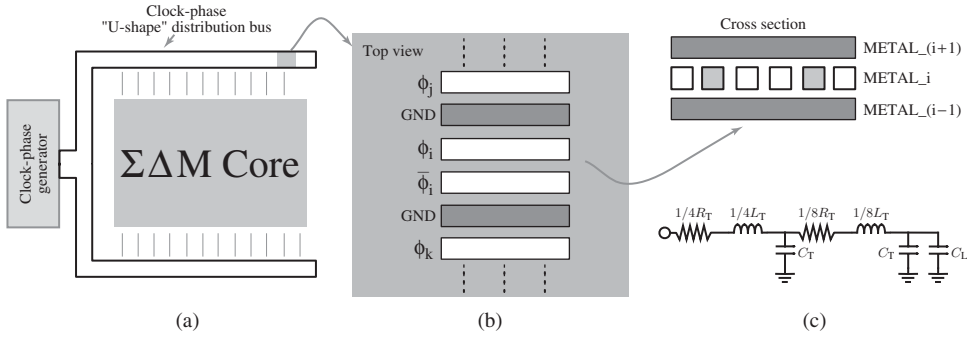


Figure 4.70 Clock-phase distribution and routing along a ΣΔM chip: (a) conceptual U-Shape bus distribution, (b) clock-phase signal isolation using a Faraday cage, and (c) equivalent circuit model extracted from electromagnetic simulations. In some practical situations, the inverted versions of clock signals are not globally routed, but locally generated by an inverter. Similarly, clock phases can be either globally or locally buffered depending on the fan-out requirements in each case [9].

and R_T can be extracted using electromagnetic simulations that take into account the technological data of the different metal layers and dielectrics involved [9]. Note that post-layout simulations of the circuit in Figure 4.70c, together with the buffer tree described above, have to be carried out in order to optimize the design of clock-phase generators in terms of silicon area and power consumption.

4.5.2 Generation of Common-Mode Voltage, Reference Voltage, and Bias Currents

The following reference and bias voltages are needed for a correct operation of ΣΔMs:

- *Reference voltage*, used in the embedded quantizer, that is, the reference ladder of the flash ADC and the feedback DAC.
- *Common-mode voltage*, extensively used by all ΣΔM subcircuits.
- *Bias currents*, required to bias all ΣΔM building blocks.

These reference and bias voltages and currents must have reduced dependence on the temperature, supply voltage, and technology process parameters in order to design robust ΣΔMs. To this purpose, dedicated circuits have to be incorporated in ΣΔM ICs in order to generate the aforementioned DC quantities.

Bandgap Circuit

As in other analog ICs, the majority of ΣΔMs generate their internal reference voltages and bias currents from a DC temperature-independent voltage. This voltage is generated using the well-known *bandgap*-reference generator circuits, often referred to as *bandgap circuits*.

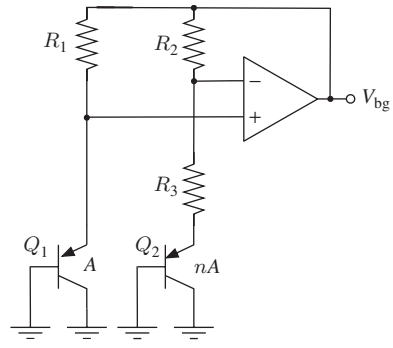


Figure 4.71 Example of bandgap circuit [44] used in some ΣΔMs [9].

Figure 4.71 shows an example of bandgap circuit [44] used in ΣΔMs [9]. The circuit makes use of lateral bipolar transistors usually available in standard CMOS technologies to generate a DC voltage given by [44]:

$$V_{bg} = V_{EB1} + U_T \cdot \ln\ n \cdot (1 + \frac{R_2}{R_3}) \tag{4.14}$$

where V_{EB1} is the forward-biased emitter-base voltage of transistor Q_1, $U_T \equiv kT/q$, and $U_T \cdot \ln\ n$ stands for the difference between the emitter-base voltages of Q_1 and Q_2.

Considering that $\frac{\partial V_{EB}}{\partial T} \simeq -1.5\ \text{mV K}^{-1}$ and $\frac{\partial U_T}{\partial T} \simeq 0.087\ \text{mV K}^{-1}$, a bandgap voltage ($V_{bg}$) with zero temperature coefficient can be achieved if $\ln\ n \cdot (1 + R_2/R_3) \simeq 17.2$. Taking this condition into account and assuming that $V_{EB1} \simeq 0.8$ V in Equation 4.14, a bandgap voltage of $V_{bg} \simeq 1.25$ V can be generated [44].

Reference Voltage Generator

The reference voltage V_{ref} required for the modulator operation can be obtained as a linear function of the bandgap voltage

$$V_{ref} = V_{r+} - V_{r-} = \alpha V_{bg} \tag{4.15}$$

where α is a proportionality factor. For instance, if $\alpha = 4/5$, then a reference voltage of $V_{ref} = 1$ V is obtained. This can be easily implemented[18] using a fully-differential amplifier in inverting configuration, as that shown in Figure 4.72. This way, by simply choosing $R_2 = 4/5R_1$, a reference voltage of $V_{ref} = 1$ V is obtained. Note that a buffer—that can be implemented by using a simple (asymmetric) OTA circuit—is used for driving the bandgap voltage V_{bg}.

The main design considerations that must be taken into account for the generation of reference voltages in ΣΔMs are a fast dynamic response (settling) as well as a low output impedance between the V_{r+} and V_{r-} lines, so that no dynamic distortion is introduced in

[18] In a design prototype, ΣΔM reference voltages can be optionally provided by an off-chip circuit included in a test PCB as will be discussed later. However, this solution is neither practical nor robust if the ΣΔM is embedded in a chip together with other circuit components which form a given electronic system.

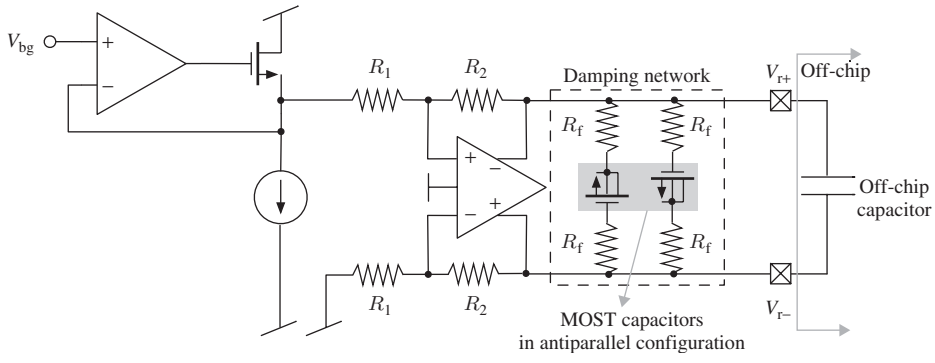


Figure 4.72 Example of reference voltage generator circuit [9].

the loop-filter integrators. To this end, the example shown in Figure 4.72 uses an off-chip capacitor which is connected between V_{r+} and V_{r-}. The value of this capacitor must be chosen according to the parasitic capacitances connected at these nodes (because of the bonding pads, bonding wires, lead frame, and package pin), so that the spurious components around half the sampling frequency are removed from the differential reference voltage. In addition, a damping network that is made up of an RC circuit based on MOS capacitors connected in antiparallel configuration, is used for removing ringing voltages added to the reference voltages [9].

Master Bias Current Generator

All current sources and sinks required to bias the ΣΔM subcircuits (opamps, comparators, etc.) need to be generated internally (on-chip) from a master bias current by a single circuit, commonly referred to as *master bias current generator*. Figure 4.73 shows an example

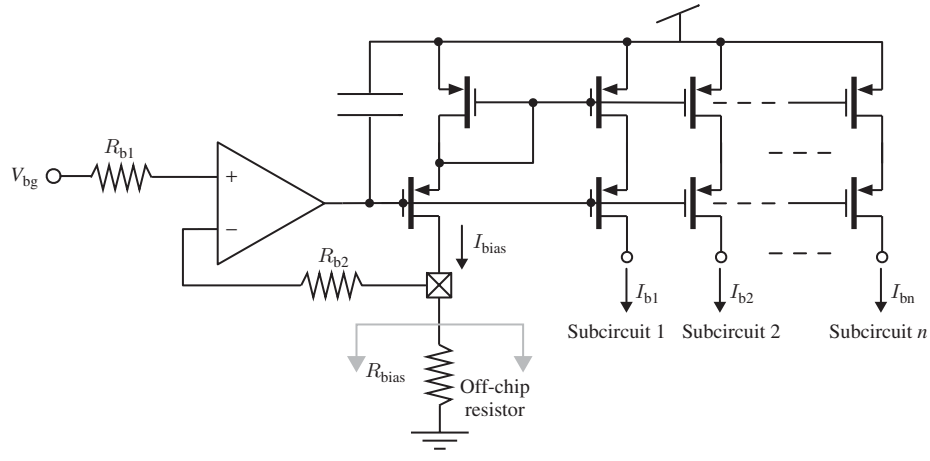


Figure 4.73 Example of master bias current generator. The resistor used for generating the master bias current R_{bias} can alternatively be implemented on-chip using unsalicided poly resistors.

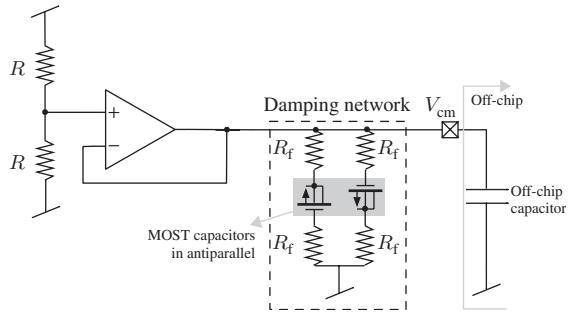


Figure 4.74 Example of common-mode voltage generator circuit [9].

of master bias current generator where a single master current is generated from the bandgap voltage and an external (off-chip) resistor. This resistor can be also implemented on-chip using unsalicided poly resistors. The generated master bias current is mirrored and properly scaled to bias all amplifiers (used in integrators), the preamplifying stages of comparators, as well as other auxiliary (analog) ΣΔM building blocks such as the reference voltage generator and the common-mode voltage generator (described in the next section).

In some applications, adaptive bias currents are implemented by means of programmable current mirrors based on the combination of switchable transistors and unit resistors. This way, the performance of the core amplifiers, and consequently the ΣΔM, can be adapted to a different set of specifications with optimized power consumption [45].

Common-Mode Voltage Generator

The common-mode voltage V_{CM} is usually defined as a half of the supply voltage, that is, $V_{CM} = (V_{DD} - V_{SS})/2$. This operation can be easily performed by a resistor divider and a buffer as illustrated in the example shown in Figure 4.74. This circuit implements the required ratio of $1/2$ in a simple and robust way using two identical resistors and a simple OTA configured as a buffer. Similar to the case of the reference voltage generators, a large (off-chip) capacitor may be used in combination with an on-chip damping network in order to "clean" the generated voltage and keep it constant and stable despite the switching-noise activity propagated across the substrate.

4.5.3 Additional Digital Logic

Apart from the extra analog circuits formerly described, some ΣΔM ICs require the incorporation of additional digital circuits in order to carry out some specific tasks. Among others, the following digital signal processing may be necessary in some applications:

- Output digital buffers driving the modulator outputs in order to drive the load capacitance due to either the bonding pads (in stand-alone IC implementations) or the decimation filter connected at the modulator output.

- Multibit DAC linearization techniques, for instance, based on DEM algorithms such as DWA.
- Decoders used in quantizers, such as thermometric-to-binary decoders, frequently used in embedded flash ADCs.
- Serial-to-parallel registers, sometimes used for managing a large number of digital control signals, for instance, in reconfigurable ΣΔMs.
- Digital power-down signals, used for turning off/on some parts of the circuits when needed in order to optimize the power consumption.

The practical implementation of the aforementioned digital circuits will strongly depend on the specific purpose and application. The detailed description of the diverse techniques that can be used is beyond the scope of this book. Instead, the interested reader can find diverse examples of these logic circuits in a number of state-of-the-art references collected at the end of Chapter 5.

4.6 Layout Design, Floorplanning, and Practical Issues

As in any other mixed-signal IC, layout implementation is one of the most critical steps in the design process of ΣΔMs. The performance of a ΣΔM can be completely destroyed if the layout is not carefully designed and this will be illustrated later in this section. To this purpose, a number of design strategies and practical tricks must be followed. Some of these layout techniques are based on general rules used in the design of analog IC circuits, while others are specific to the design of ΣΔMs. This section gives an overview of the most important and critical recommendations to implement a high-quality ΣΔM layout.

4.6.1 Layout Floorplanning

At the very beginning of the layout design stage, it is essential to make an appropriate partitioning and placement or *floorplan* of the different parts that constitute the layout of the ΣΔM. This floorplan must take into account the recommendations discussed subsequently.

Divide the Layout into Different Parts or Regions

The most usual layout partition considers three different regions corresponding to the analog, mixed-signal, and digital parts of the modulator:

- The *analog part* should include all analog subcircuits of the ΣΔM core, such as OTAs, the preamplifiers used in the latched comparators, capacitors, resistors, inductors, as well as any other auxiliary analog circuit, that is, master bias current generator, reference voltage generator, etc. This part of the layout must also include critical parts that may affect the performance of the modulators, such as for instance some control circuits required to implement reconfiguration techniques, power-down switches, as well as any other control (analog) circuit.

- The *mixed-signal part* usually includes CMOS switches (in SC-ΣΔMs), the latches used in the comparators, as well as any other ΣΔM subcircuit handling both analog and digital signals.
- The *digital part* includes the clock-phase generator, digital buffers, as well as any other digital logic circuit required for the operation of the modulator, such as DEM logic, decoders used the quantizers, digital registers, etc.

It is very important to keep in mind that there is not a direct correspondence between the schematic building blocks and their corresponding parts in the layout of the modulator. For instance, an SC integrator is made up of three essential circuit elements: opamps, capacitors, and CMOS switches. However, the first two parts (opamps and capacitors) are placed in the analog region of the layout, whereas CMOS switches are included in the mixed-signal region. So the routing between these parts is also especially critical because their associated parasitics may severely degrade the performance of the modulator. Another example is the embedded multibit flash ADCs. These circuits are made up of a resistor ladder and a bank of comparators. The former are included in the analog part, while the latter are in turn subdivided into three different circuit blocks: preamplifiers, latches, and SR flip-flops (Figure 4.55), respectively included in the analog, mixed-signal, and digital sections of the layout.

Shield Sensitive ΣΔM Analog Subcircuits from Switching Noise

The placement of analog and digital parts should be carried out in such a way that there is an increased distance among the most sensitive analog blocks and the noisy digital parts. Nevertheless, given that the majority of standard CMOS technologies have a low-resistivity substrate, the switching activity of digital circuits may severely degrade the performance of the chip. Hence, additional layout techniques are frequently incorporated to attenuate the impact of noisy signals propagating across the substrate. Some of these techniques are the following:

- Use *guard rings* with dedicated bonding pads and pins surrounding each section of the circuit. Although these well-known techniques are not so effective in low-resistivity bulk epitaxial processes, it provides some attenuation at least for the noisy surface currents propagating in the epitaxial layer [46].
- Use *separate voltage supplies* for the different parts of the modulator. This strategy implies using dedicated power supplies (V_{DD} and V_{SS}), each one with their bonding pad and chip package pin for the analog, mixed-signal, and digital parts, as well as for the guard rings.
- Extensive use of *decoupling capacitors* throughout the chip for each supply voltage (V_{DD}) and its corresponding ground (V_{SS}). This well-known technique allows to keep supply voltages *clean*.

Use Buses to Distribute Signals Shared by Different Parts of the Modulator

There are a number of signals that are shared by a number of ΣΔM building blocks, and consequently, must be distributed across the entire chip. These signals include, among

others, the digital clock phases, DAC control signals, power-down signals, supply voltages, bias currents, reference voltages, and common-mode voltages. All these signals must be routed based on a U-shaped bus configuration similar to the one discussed in Figure 4.70.

Be *Obsessive* About Layout Symmetry and Details of Analog Parts

The layout of the critical parts of the modulator, particularly the subcircuits forming the loop filter, must be designed paying attention to any minor detail, following the most useful layout rules of high-performance analog circuits including, among others:

- Use *common-centroid* layout structures with unit circuit elements (capacitors, resistors, transistors) in order to maximize the matching performance.
- Use *fully-differential* topologies to reduce common-mode interferences.
- Use *multiple contacts* and vias in order to reduce the parasitic resistances associated with each connection and to avoid *catastrophic* failures derived from microfractures in the metal connections.
- Use *single-finger transistors* to build CMOS switches. This strategy helps to avoid crossings among digital signals (clock phases connected at the gate of transistors) and analog signals (connected at the drain/source terminals).[19]
- Optimize the *width of metal connections* taking into account the maximum current density flowing through each metal path and their parasitic resistances and capacitances.
- Use *stacked metal layers* to reduce the parasitic resistances.

As an illustration of the abovementioned recommendations, Figure 4.75 shows the layout floorplanning of a cascade 2-1 SC-ΣΔM. A number of the aforementioned rules are highlighted in the figure. Note that, in addition to previous recommendations, there are many other basic layout rules that must be followed in order to maximize the performance of analog circuits. The interested reader can find plenty of excellent papers and books dealing with this topic [44, 47, 48].

4.6.2 I/O Pad Ring

The design of the I/O pad ring enclosing the ΣΔM core layout is also very critical to guarantee a correct operation of the modulator. A number of practical rules must be followed in order to avoid any performance degradation caused by a nonideal behavior associated with the design, placement, and/or routing of bonding pads. Among others, the following recipes should be taken into account:

- *Divide the pad ring into different parts* (analog, mixed-signal, digital, etc.) to further improve the isolation among the different ΣΔM regions and to avoid switching noise coupling. To this purpose, divide the power ring using power-cut cells, that is, cells that provide a virtual cut through diodes in antiparallel configuration, as conceptually depicted in Figure 4.76 [49].

[19] If single-finger transistors are used, the analog signals can be directly routed to the transistor diffusions, while the digital signals can be connected to the transistor gate either using polysilicon or metal layers.

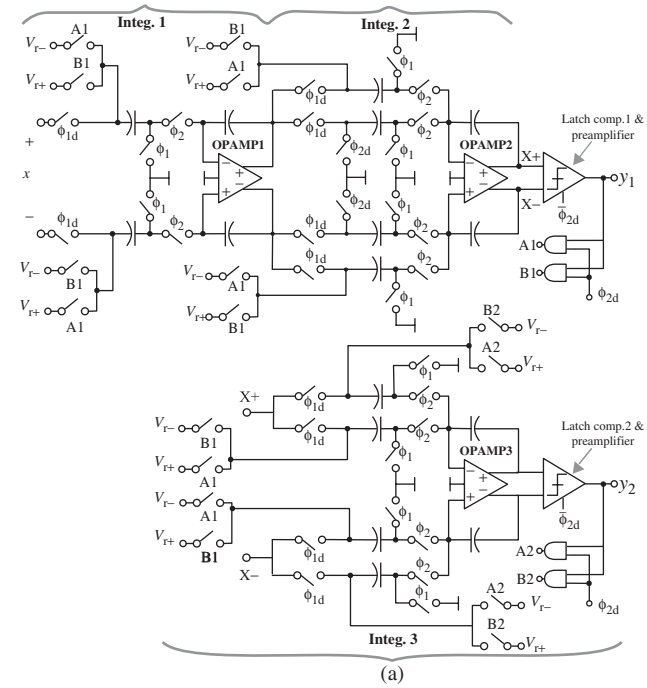


Figure 4.75 Example of layout floorplanning of ΣΔMs: (a) schematic of the modulator (cascade 2-1 SC-ΣΔM) and (b) layout floorplanning. Each layout region, as well as its corresponding guard rings, has dedicated supply voltages.

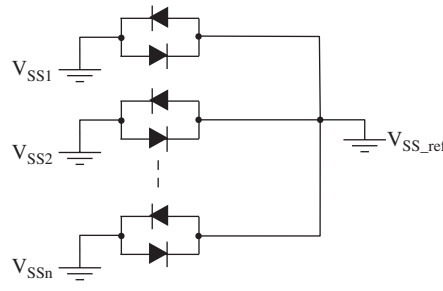


Figure 4.76 Conceptual illustration of the virtual connection among different grounds in the power ring using diodes in antiparallel configuration.

- *Place the switching* (*noisy*) *pads as far as possible* from the most sensitive analog pads.
- Use pads with *ESD protection* where needed, for instance for signals driving transistor gates.

As an illustration, Figure 4.77 shows the conceptual floorplanning of an I/O pad ring, highlighting some of the most important strategies stated earlier.

4.6.3 Importance of Layout Verification and Catastrophic Failures

As it is well known, the use of suitable layout CAD tools, such as Design Rule Checker (DRC) and Layout Versus Schematic (LVS) tools, are very useful for designers to guarantee that their layouts are free of errors. In addition, layout-extracted simulations including technology parasitics are also quite convenient to ensure a correct performance of the chip before sending it for fabrication.

Despite the usefulness of the aforementioned tools, in many practical situations (particularly in some industrial first-silicon prototypes) it is very common to work with technologies which are still under development.[20] However, one of the consequences of using such new technologies is that the design kits (i.e., the technology files including electrical device models, layout rules, etc.) are also under development, which adds extra effort to designers. For instance, it is relatively common for parasitic extraction tools such as Layout Parasitic Extractors (LPEs) to be unavailable and consequently, designers have to be very careful and conservative during the design process—especially at the layout stage.

In this scenario, it is especially critical to pay attention not only to the error messages provided by CAD tools (such as DRC/LVS) but also to the warning messages. The latter may look like insignificant problems—particularly for novel designers! However, catastrophic failures may be caused by these (apparently a priori) minor problems. To illustrate this issue, which may appear in many practical circumstances, let us consider again the SC-ΣΔM shown in Figure 4.75a. The circuit implementation of this modulator involves over 1000 transistors, which requires a tedious and careful design and layout verification.

[20] Many times, industrial partners working in a given project are interested in testing high-performance analog cells such as ΣΔMs in a cutting-edge technology.

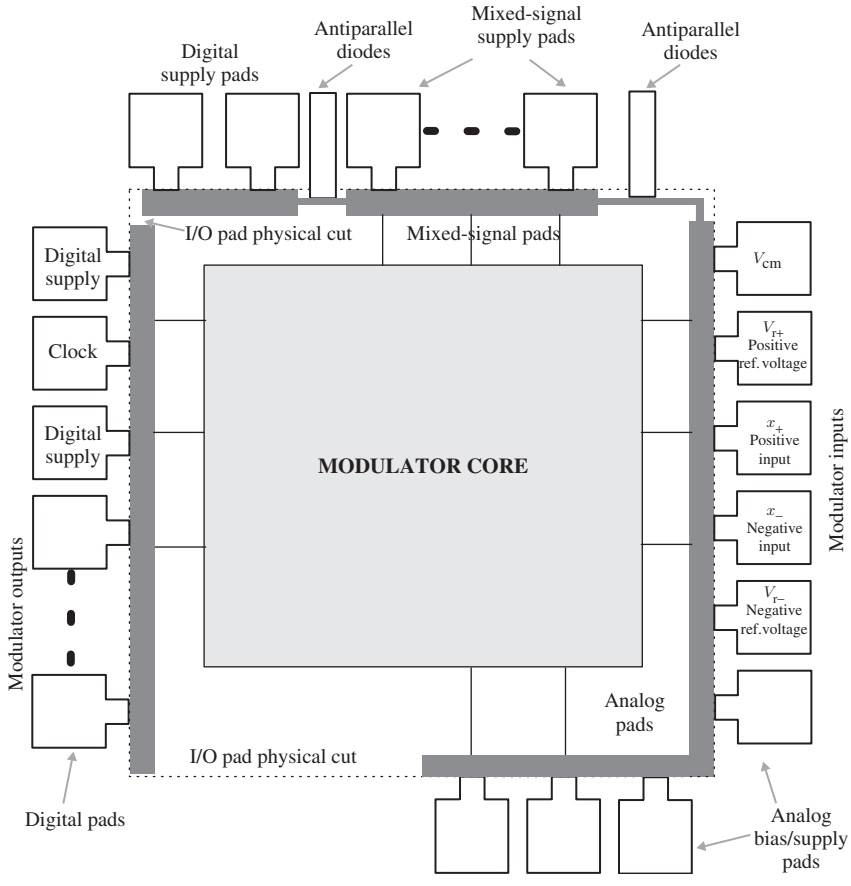
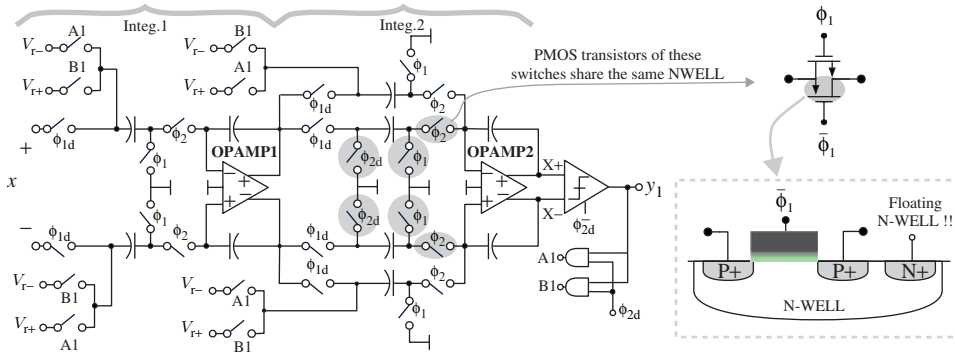


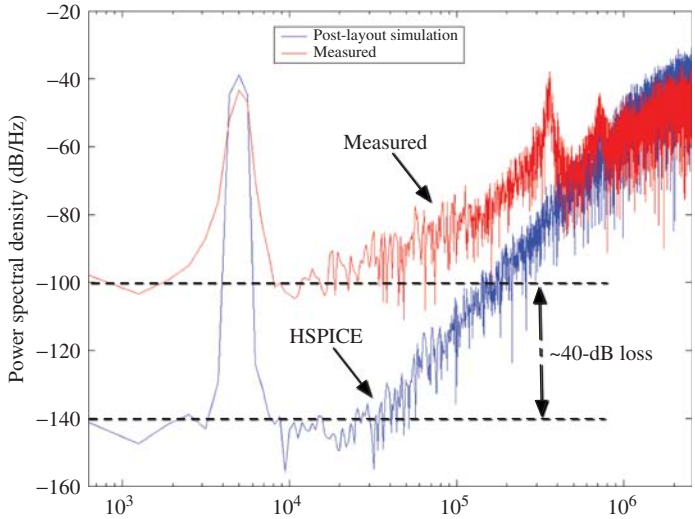
Figure 4.77 Conceptual illustration of an I/O pad ring which can be used in ΣΔMs.

Let us assume that, owing to a design error, the NWELL enclosing the pMOS transistors of the CMOS switches highlighted in Figure 4.78a are floating, that is, there is not any well contact. This error is displayed as a warning message by the DRC/LVS tools. Assuming that there is no LPE available, the layout-extracted netlist is essentially the same as that of the schematic. Therefore, transistor-level simulations will give good results, masking an error that obviously affects the signal transmitted from the first integrator to the second integrator. As a consequence, as illustrated in Figure 4.78b, experimental measurements reveal a severe performance degradation as compared to the transistor-level (HSPICE) simulations. That is, a single missing connection in an NWELL enclosing only six pMOS transistors may completely destroy the performance of a ΣΔM made up of more than 1000 transistors.

In conclusion, the layout phase of ΣΔMs is a critical stage of the design procedure and it must be carefully verified and checked using verification CAD tools without underestimating any warning message or apparently minor issue, which may become the cause of a catastrophic failure.



(a)



(b)

Figure 4.78 Illustrating the performance degradation in the SC-ΣΔM shown in Figure 4.75a caused by a floating NWELL: (a) SC schematic of the front-end stage of Figure 4.75a, highlighting the CMOS switches with floating NWELL and (b) comparison between experimental and simulated modulator output spectra, showing how the noise-shaping performance becomes severely degraded as a consequence of the floating NWELL.

4.7 Chip Package, Test PCB, and Experimental Set-Up

The last stage of the design phase of ΣΔMs deals with a number of tasks required to prepare the chip for testing its performance experimentally in a laboratory. The most important and critical considerations to take into account are the bonding diagram and chip package, the test PCB, and the experimental set-up. Some of these issues must be addressed before sending the chip for fabrication. This section describes the most

important practical issues to be considered regarding packaging, prototyping, and testing of ΣΔMs, assuming that they are going to be measured as stand-alone ICs. Similar recommendations can be followed in case the ΣΔM is embedded in a system-on-chip (SoC) implementation.

4.7.1 Bonding Diagram and Package

The bonding of the modulator chip has to be deeply analyzed, concerning the pad-ring placement described in previous section, the number of pads and pins assigned to supplies, guard rings, reference voltages, etc. Therefore, the selection of the most suitable chip package is a critical issue because the parasitics associated with the package and with its bonding connections to the chip may severely affect the performance of the modulator, particularly when high speed and/or high resolution are demanded.

Indeed, the effect of package and bonding-wire parasitics can be analyzed in detail by using dedicated CAD tools (such as Cadence Allegro), which allow to simulate the circuit considering these parasitic effects. Some tools, such as Cadence PKG, allow modeling of the package by an equivalent LCR circuit that is synthesized from the physical and electrical characteristics of a given package. Among others, the most important limitations come from the parasitic bonding inductances. It is recommendable to use surface-mount devices (SMDs) such as ball grid array (BGA) or quad flat package (QFP)—the latter being quite commonly used for testing ASIC prototypes in general and ΣΔ ICs in particular.

Some critical considerations must be also taken into account. On the one hand, bonding pads of the same type (analog, mixed-signal, digital, or digital I/Os) must be placed together and apart from the rest using *power-cut* diode cells similar to those shown in Figure 4.76. This allows power separation and avoids cross-talk. Moreover, supply pads and pins should be placed in parallel to reduce the total parasitic inductance due to the compensation given by the complementary mutual inductances [44]. On the other hand, double-bonding techniques and multiple pins are frequently used for the supplies of the different sections in order to reduce the inductance of the paths to the chip and decrease supply bounce. Note that if reference voltages are provided externally (off-chip), double bonding should be also used for the reference pads/pins in order to halve their bonding parasitic inductances as well.

4.7.2 Test PCB

In order to characterize the ΣΔM IC, a special-purpose PCB is needed in order to connect the chip to the different test instruments providing the necessary signals, biasing, and supply voltages, as well as to capture the modulator output data for further processing in a computer. Owing to the aforementioned practical limitations associated with the chip packages, other testing approaches are considered, particularly in very high-speed applications. For instance, the modulator die can be directly bonded to the test PCB, without using any package, in order to reduce the dimensions of bonding and consequently the parasitic inductances [9]. Alternatively, low-temperature co-fired ceramic (LTCC) substrates—commonly used in RF applications—can be used for reducing the effect of off-chip circuit element parasitics, as these elements can be embedded together with the modulator die into the same ceramic package.

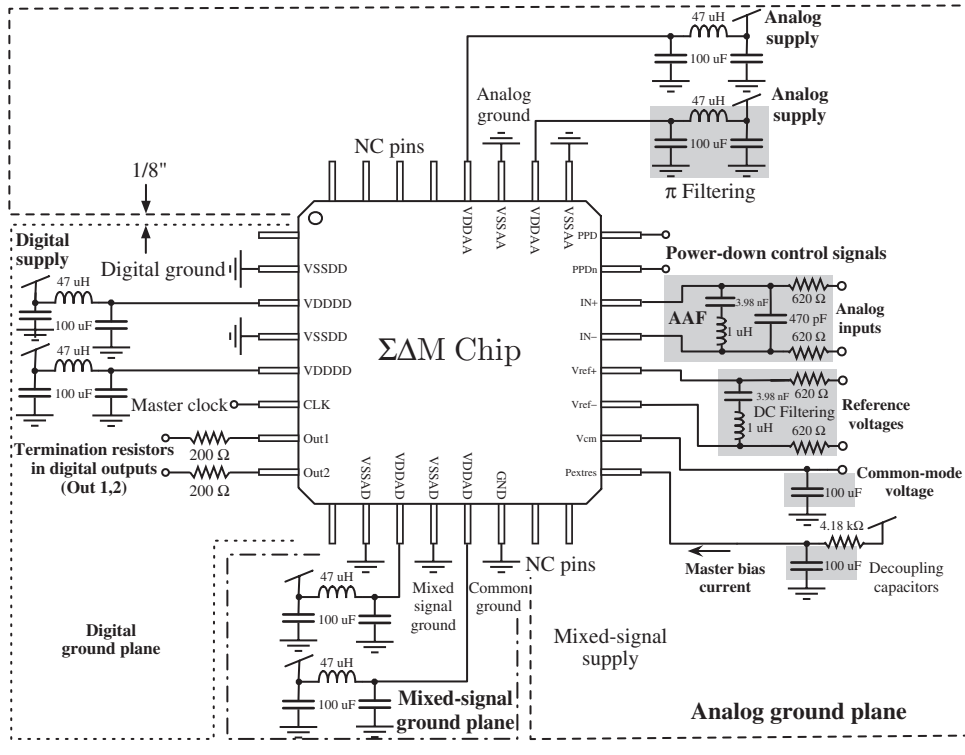


Figure 4.79 Conceptual schematic of a test PCB used for measuring the performance of the modulator in Figure 4.75. Note that this is a simplified version of the modulator presented in [50], which included a programmable-gain front-end integrator and a preamplifier connected to the modulator. The additional pins required to include these circuits have been omitted for the sake of simplicity. Moreover, the values of circuit elements in this test chip correspond to the signal specifications reported in [50]: $B_w = 20$ kHz and DR = 110 dB.

Let us consider the most common approach that consists of a chip package connected to the necessary circuits for testing in a multilayer PCB. Figure 4.79 shows the conceptual schematic of a PCB used for testing the ΣΔM of Figure 4.75 [50]. In this example, a 32-pin QFP package is used. Note that, in addition to the effect of circuit parasitics themselves, external electromagnetic interferences are injected into the ΣΔM chip through both inductive and capacitive coupling. In order to reduce the impact of off-chip circuit parasitics and to obtain a robust test circuit, the following circuit strategies are usually incorporated in the test PCB (some of them highlighted in Figure 4.79):

- *Separate the PCB into different areas or planes* corresponding to the analog, mixed-signal, and digital signals. Ground planes should be separated—usually with a gap larger than 1/8"—and connected only in one point. This way noisy return currents are minimized.
- *Use regulators to keep the values of the supply voltages stable*.

- *Use decoupling capacitors in the supply, biasing, and reference voltage lines*. It is a common practice to combine a large tantalum capacitor with a small ceramic capacitor connected in parallel, in such a way that the ceramic capacitor is placed as close as to the package pin. Both capacitors are usually connected together with one inductor in a π-filter configuration, as illustrated in Figure 4.79.
- *Keep digital signal paths as far as possible from the sensitive analog pins*.
- *Use termination resistors* for impedance coupling in the digital output lines.
- *Use ESD protection diodes* for sensitive input pins, particularly if they are not used in the corresponding bonding pads.

Apart from the aforementioned techniques, PCBs used for testing $\Sigma\Delta$Ms should contain an AAF. A low-order (typically first- or second-order) RC filter is enough in the majority of practical situations.

4.7.3 Experimental Test Set-Up

The test set-up and instruments used for measuring the performance of $\Sigma\Delta$Ms in the laboratory are also very important and need to be carefully planned. The number and type of laboratory equipment depend on different factors including the nature of the signal to be tested (low-pass, band-pass, sinewave, modulated signal, etc.), the target modulator specifications (in-band noise power, linearity, etc.), the type of performance metrics to be measured (output spectrum, SNR/SNDR, HD3, IM3, INL, etc.), and so forth.

Planning the Types and Number of Equipment Needed

It should be noted, however, that every additional circuit element or laboratory instrument included in the measurement set-up is a potential source of errors and interference that may degrade the modulator performance. For that reason, it is very important to think in advance—preferably during the design phase—of the kind of measurements that will be needed and the type and number of instruments which will be required. In many practical situations, the performance of the chip cannot be properly characterized experimentally because of limitations imposed by the lab equipment and instruments. Some examples of these limitations are the jitter error of the clock generators, the maximum frequency/linearity provided by the signal generator, the maximum capture rate of the logic analyzer, etc.

Generally speaking, regardless of the specific measurements to be carried out, at least the following instruments are commonly needed for testing $\Sigma\Delta$Ms:

- *Power supply generators* to generate the supply voltages, reference voltages, common-mode voltage as well as any other DC voltage or bias current required. Whenever possible, it is highly recommended to use a *voltage regulator circuit*—embedded in the test PCB—to generate all DC and bias signals from a single voltage supply to minimize the number of instruments and wires.
- *Analog (input) signal generators*, at least one sinewave generator, if possible providing balanced fully-differential signals with the required bandwidth and accuracy—in terms of noise and linearity. Note that the same common-mode voltage generator must be used for both the signal generator and the modulator chip. Otherwise, a systematic

offset will be introduced at the modulator inputs, which may severely degrade the performance of the modulator.

- *Clock generator* with the required performance in terms of frequency, logic levels, and clock jitter error.
- *Data acquisition systems*, such as logic analyzers or SoC test units (for instance, Agilent 93000), are essential for capturing the modulator output bitstreams, which are transferred to a personal computer or to a workstation in order to process the data.[21]

As mentioned previously, apart from the "essential" instruments, other laboratory equipment may eventually be needed, such as spectrum analyzers to check the frequency spectrum of a given signal "on the fly," multimeters to measure the DC operating point, etc.

Connecting Laboratory Instruments

The connection of the different instruments to the test PCB is critical, and must be implemented in such a way that their parasitics are minimized. Among others, the following recommendations must be followed:

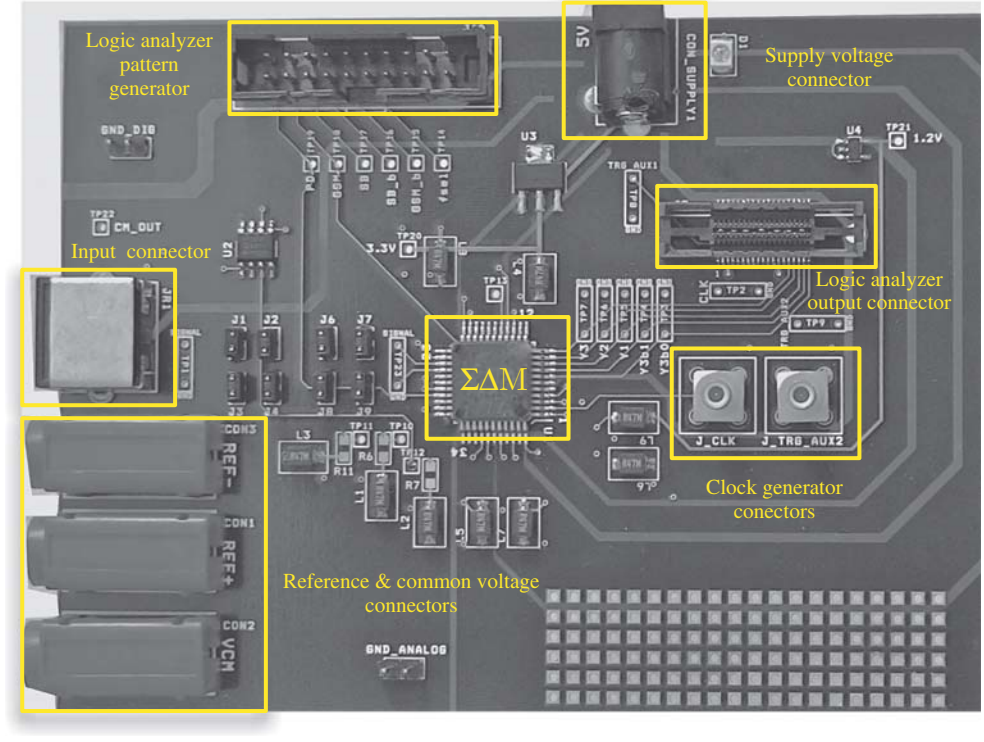


Figure 4.80 Illustrating some connectors of different instruments in a ΣΔM IC test PCB.

[21] Many logic analyzers at present have an embedded PC, so that the same instrument is used for capturing the ΣΔM output data and for processing this data, for instance, using MATLAB. Alternatively, some logic analyzers and SoC test units can be also used for generating the input signal waveforms, supply voltages, digital control signals, etc.

- *Use the appropriate connectors* for each instrument in the PCB, as illustrated in Figure 4.80.
- *Reduce the length and number of cables* used for connecting the instruments.
- Make sure that *the ground of each instrument is connected to its corresponding ground* in the PCB, that is, instruments providing analog signals should have their grounds connected to the analog ground in the PCB.
- Make sure that *all grounds are connected in star configuration* as conceptually depicted in Figure 4.81a. Do not use the scheme shown in Figure 4.81b, where the ground of one instrument is connected to the ground of another instrument and so on.

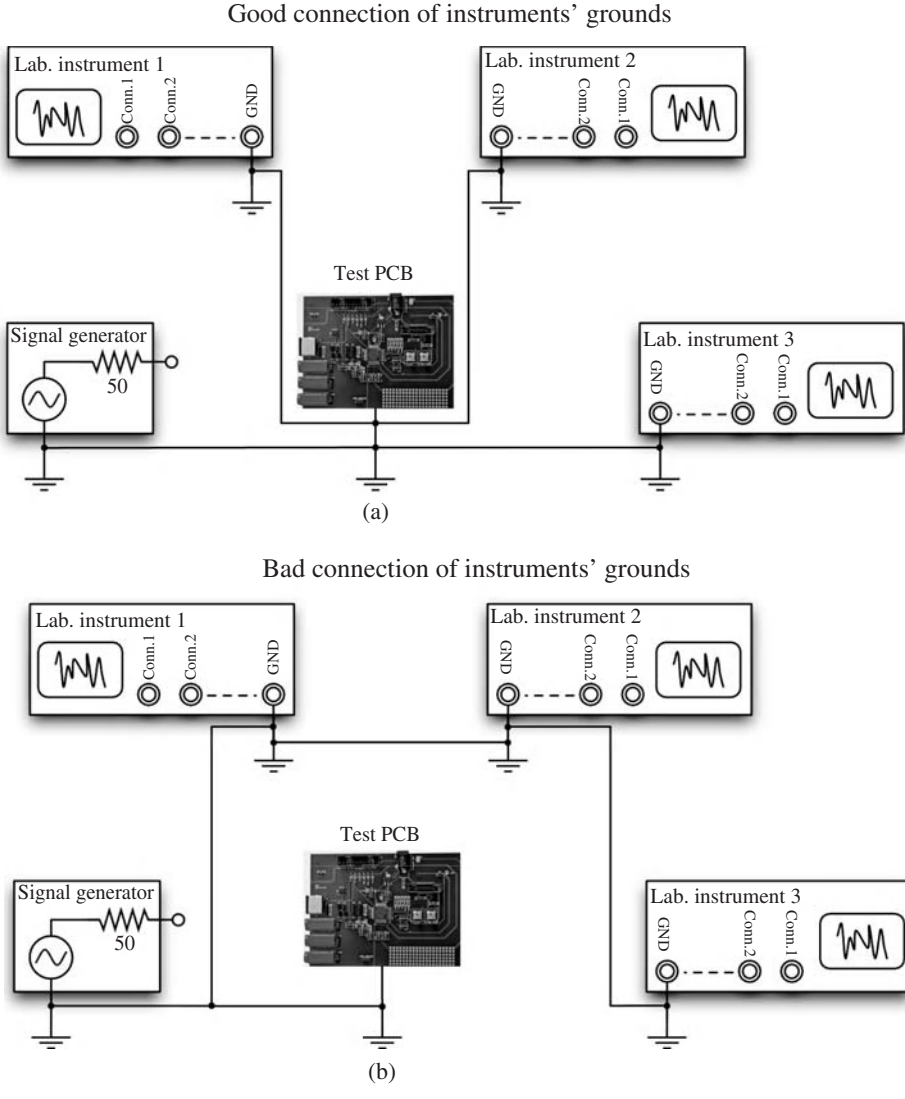


Figure 4.81 Illustrating the ground connection of different laboratory instruments to the test PCB.

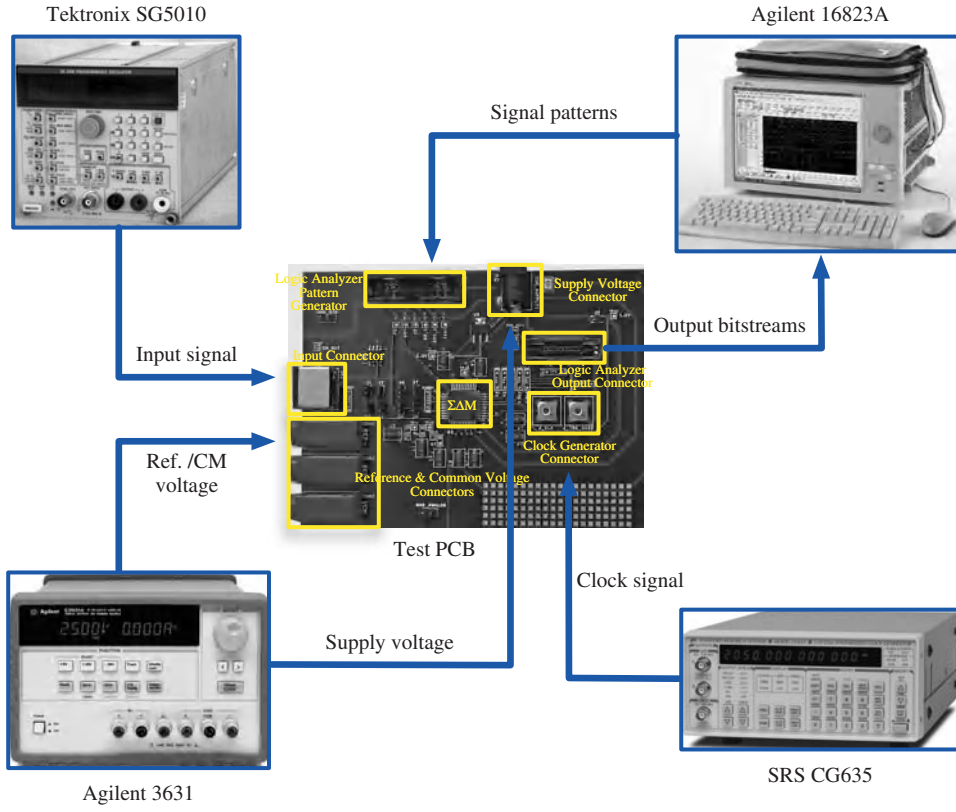


Figure 4.82 Conceptual example of a ΣΔM measurement set-up based on a logic analyzer.

Last but not least, another important issue to take into account in the set-up deals with turn-on/off sequence followed to switch on/off the different instruments involved. Thus, the turn-on sequence should start with the supply voltage, followed by the clock-signal generator, and finally, the analog signal generator. The turn-off sequence should be carried out in the opposite way.

Measurement Set-Up Example

Figure 4.82 shows the conceptual diagram of a measurement set-up based on a logic analyzer as the data acquisition system [9]. In this case, an Agilent A3631A unit generates the voltage supply, whereas the fully-differential input signal is generated by a Tektronix SG5010 audio oscillator. An optional SRS CG635 clock generator can be used for generating the clock signal, whereas an Agilent A16823B logic analyzer[22] is used for acquiring the modulator output bitstreams and also for generating the digital control signals required

[22] Depending on the electrical characteristics demanded for the clock signal, particularly the jitter error specification, it can be generated by a logic analyzer. If a very low-jitter clock signal is required, then an appropriate clock-signal generator should be used instead.

to test the modulator. After bitstream acquisition, data is post-processed in MATLAB. Note that, in the case of cascade modulators in which the DCL is not implemented on-chip, a procedure similar to that described in Section 4.3 can be followed to process the output results.

4.8 Summary

This chapter presented a design guide that includes a collection of practical recipes to be taken into account in the electrical design and characterization of ΣΔMs, going from macromodel representation of their building blocks to their transistor-level and physical-level implementation, including layout, packaging, and test set-up. A complete description of the macromodels frequently used for the main building blocks has been presented, giving some examples on how to use them in both SC and CT-ΣΔMs. At the transistor level, the most important design considerations of ΣΔM circuit blocks have been detailed, including amplifiers, transconductors, switches, comparators, and DACs. Several design examples and simulation test benches have been illustrated, emphasizing on their implementation in Cadence Design FrameWork II. Apart from the building-block design itself, a number of practical issues related to the electrical characterization of ΣΔMs, such as the injection of noise sources in transient electrical simulations and the post-processing of simulation results, have been also addressed.

The diverse aspects covered in this chapter close the design flow of ΣΔMs discussed in this book, going from ideal fundamentals and architecture considerations given in Chapter 1, to the impact of nonideal errors at system level discussed in Chapter 2, and to their application to the high-level synthesis and verification using behavioral models and simulation, which is detailed in Chapter 3. This systematic top-down/bottom-up design methodology is complemented in Chapter 5 with an exhaustive study of the state-of-the-art on ΣΔMs in order to help designers select the optimum ΣΔM architecture and circuit technique for their applications together with identifying the trends, challenges, and practical solutions which have been proposed in each case by the ΣΔM design community.

References

[1] K. Kundert and O. Zinke, *The Designer's Guide to Verilog-AMS*, Kluwer Academic Publishers, 2004.
[2] G. Suárez, M. Jiménez, and F. O. Fernández, "Behavioral Modeling Methods for Switched-Capacitor ΣΔ Modulators," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, vol. **54**, pp. 1236–1244, June 2007.
[3] J. M. Rabaey, SPICE 3 User Guide. [Online]. Available: http://bwrc.eecs.berkeley.edu/classes/icbook/spice/.
[4] A. Vladimirescu, *The SPICE Book*. John Wiley & Sons, 1994.
[5] Y. Tsividis, "Integrated Continuous-Time Filter Design—An Overview," *IEEE Journal of Solid-State Circuits*, vol. **29**, pp. 166–176, March 1994.
[6] S. Pavan, "Efficient Simulation of Weak Nonlinearities in Continuous Time Oversampling Converters," *IEEE Transactions on Circuits and Systems I—Regular Papers*, vol. **57**, pp. 1925–1934, August 2010.
[7] B. Razavi, *Principles of Data Conversion System Design*, IEEE Press, 1995.
[8] F. Maloberti, *Data Converters*, Springer, 2007.
[9] A. Morgado, R. del Río, and J. M. de la Rosa, *Nanometer CMOS Sigma-Delta Modulators for Software Defined Radio*, Springer, 2011.

[10] Verilog-A. *Language Reference Manual: Analog Extensions to Verilog HDL*, Open Verilog International, 1996.
[11] *Cadence Verilog-A Language Reference*, Cadence Design Systems Inc., 2006.
[12] J. González and E. Alarcón, "Current-Steering High-Speed D/A Converters for Communications," *Chapter 3 in CMOS Telecom Data Converters* (A. Rodríguez-Vázquez, F. Medeiro, and E. Janssens, Editors), Kluwer Academic Publishers, 2003.
[13] M. Ortmanns and F. Gerfers, *Continuous-Time Sigma-Delta A/D Conversion: Fundamentals, Performance Limits and Robust Implementations*, Springer, 2006.
[14] J. M. de la Rosa, B. Pérez-Verdú, and A. Rodríguez-Vázquez, *Systematic Design of CMOS Switched-Current Bandpass Sigma-Delta Modulators for Digital Communication Chips*, Kluwer Academic Publishers, 2002.
[15] Virtuoso® Spectre® Circuit Simulator User Guide, Cadence Design Systems Inc., 2010.
[16] *The HSPICE Documentation Set*, Synopsys Inc., 2005.
[17] W. Bennett, "Spectra of Quantized Signals," *Bell System Technical Journal*, vol. **27**, pp. 446–472, July 1948.
[18] F. Medeiro, B. Pérez-Verdú, and A. Rodríguez-Vázquez, *Top-Down Design of High-Performance Sigma-Delta Modulators*, Kluwer Academic Publishers, 1999.
[19] Y. Dong and A. Opal, "Time-Domain Thermal Noise Simulation of Switched Capacitor Circuits and Delta-Sigma Modulators," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. **19**, pp. 473–481, April 2000.
[20] J. Kasdin, "Discrete Simulation of Colored Noise and Stochastic Processes and $1/f^{\alpha}$: Power Law Noise Generation," *Proceedings of the IEEE*, vol. **83**, pp. 802–827, February 1995.
[21] *Virtuoso® Schematic Editor L User Guide*, Cadence Design Systems Inc., 2008.
[22] R. del Río, F. Medeiro, B. Pérez-Verdú, J. M. de la Rosa, and A. Rodríguez-Vázquez, *CMOS Cascade* $\Sigma\Delta$ *Modulators for Sensors and Telecom: Error Analysis and Practical Design*, Springer, 2006.
[23] Y. Geerts, M. Steyaert, and W. Sansen, *Design of Multi-bit Delta-Sigma A/D Converters*, Kluwer Academic Publishers, 2002.
[24] S. Narayanan, "Application of Volterra Series to Intermodulation Distortion of Transistor Feedback Amplifier," *IEEE Transactions Circuit Theory*, pp. 518–527, November 1970.
[25] P. Wambacq, G. Gielen, P. Kinget, and W. Sansen, "High-Frequency Distortion Analysis of Analog Integrated Circuits," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. **46**, pp. 335–345, March 1999.
[26] W. Yu, S. Sen, and B. Leung, "Distortion Analysis of MOS Track-and-hold Sampling Mixers Using Time-varying Volterra Series," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. **46**, pp. 101–113, February 1999.
[27] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*, IEEE Press, 2005.
[28] A. Morgado, R. del Río, J. M. de la Rosa, R. Castro-López, and B. Pérez-Verdú, "A 0.13 µm CMOS Adaptive Sigma-Delta Modulator for Triple-Mode GSM/Bluetooth/UMTS Applications," *Microelectronics Journal*, vol. **41**, pp. 277–290, 2010.
[29] R. Tortosa, A. Aceituno, J. M. de la Rosa, A. Rodríguez-Vázquez, and F. V. Fernández, "A 12-bit, 40MS/s Gm-C Cascade 3-2 Continuous-Time Sigma-Delta Modulator," Proc. of the IEEE Intl. Symp. on Circuits and Systems, pp. 1–4, 2007.
[30] P. Allen and D. Holberg, *CMOS Analog Circuit Design*, 2nd edn, Oxford University Press, 2002.
[31] A. Rodríguez-Vázquez et al., "Comparator Circuits," *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 577–600, John Wiley & Sons, 1999.
[32] B. Nikoli et al., "Improved Sense-Amplifier-Based Flip-Flop: Design and Measurements," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 876–884, June 2000.
[33] T. Kobayashi et al., "A Current-Controlled Latch Sense Amplifier and a Static Power-Saving Input Buffer for Low-Power Architecture," *IEEE Journal of Solid-State Circuits*, vol. **28**, pp. 523–527, April 1993.
[34] A. Yukawa, "A CMOS 8-bit High-Speed Converter IC," *IEEE Journal of Solid-State Circuits*, vol. **20**, pp. 775–779, June 1985.
[35] Y. Wang and B. Razavi, "An 8-Bit 150-MHz CMOS A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 308–317, March 2000.
[36] G. Yin et al., "A High-Speed CMOS Comparator with 8-b Resolution," *IEEE Journal of Solid-State Circuits*, vol. **27**, pp. 208–211, February 1992.

[37] W. Press et al., *Numerical Recipes in C. The Art of Scientific Computing*, 2nd edn, Cambridge University Press, 1992.

[38] S. Yan and E. Sánchez-Sinencio, "A Continuous-Time ΣΔ Modulator With 88-dB Dynamic Range and 1.1-MHz Signal Bandwidth," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 75–86, January 2004.

[39] P. Crombez et al., "A Single-Bit 500 kHz-10 MHz Multimode Power-Performance Scalable 83-to-67 dB DR CT ΔΣ Modulator for SDR in 90 nm Digital CMOS," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 1159–1171, June 2010.

[40] L. Breems, R. Rutten, and G. Wetzker, "A Cascaded Continuous-Time ΣΔ Modulator with 67-dB Dynamic Range in 10-MHz Bandwidth," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 2152–2160, December 2004.

[41] G. Mitteregger, C. Ebner, S. Mechnig, T. Blon, C. Holuigue, and E. Romani, "A 20-mW 640-MHz CMOS Continuous-Time ΣΔ ADC With 20-MHz Signal Bandwidth, 80-dB Dynamic Range and 12-bit ENOB," *IEEE Journal of Solid-State Circuits*, vol. **41**, pp. 2641–2649, December 2006.

[42] M. Bolatkale et al., "A 4GHz Continuous-Time ΔΣ ADC With 70dB DR and −74 dBFS THD in 125 MHz BW," *IEEE Journal of Solid-State Circuits*, vol. **46**, pp. 2857–2868, December 2011.

[43] K. Lee and R. Meyer, "A Current-Controlled Latch Sense Amplifier and a Static Power-Saving Input Buffer for Low-Power Architecture," *IEEE Journal of Solid-State Circuits*, vol. **20**, pp. 1103–1113, December 1985.

[44] B. Razavi, *Design of Analog CMOS Integrated Circuits*, McGraw-Hill, 2000.

[45] A. Morgado, R. del Río, J. M. de la Rosa, L. Bos, J. Ryckaert, and G. van der Plas, "A 100kHz-10MHz BW, 78-to-52dB DR,4.6-to-11mW Flexible SC ΣΔ Modulator in 1.2-V 90-nm CMOS," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 418–421, September 2010.

[46] M. Felder and J. Ganger, "Analysis of Ground-Bounce Induced Substrate Noise Coupling in a Low Resistive Bulk Epitaxial Process: Design Strategies to Minimize Noise Effects on a Mixed-Signal Chip," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. **46**, pp. 1427–1436, October 1999.

[47] F. Maloberti, "Layout of Analog and Mixed Analog-Digital Circuits," *Chapter 11 in Design of Analog-Digital VLSI Circuits for Telecommunications and Signal Processing* (J. Franca and Y. Tsividis, Editors), Prentice-Hall, 1994.

[48] Y. Tsividis, *Mixed Analog-Digital VLSI Devices and Technology: An Introduction*, McGraw-Hill, 1996.

[49] S. Cao et al., "ESD Design Strategies for High-Speed Digital and RF Circuits in Deeply Scaled Silicon Technologies," *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. **57**, pp. 2301–2311, September 2010.

[50] J. M. de la Rosa et al., "A CMOS 110-dB@40-kS/s Programmable-Gain Chopper-Stabilized Third-Order 2-1 Cascade Sigma-Delta Modulator for Low-Power High-Linearity Automotive Sensor ASICs," *IEEE Journal of Solid-State Circuits*, vol. **40**, pp. 2246–2264, November 2005.

5

Frontiers of ΣΔ Modulators: Trends and Challenges

Since the introduction of the ΣΔ modulation technique, there has been a pleiad of ICs published; implemented using many diverse technology processes, architectural, and circuit techniques, which have targeted a huge number of applications that span from sensors and instrumentation to ultra-low power, biomedical applications, and broadband communications [1–3].

An in-depth understanding of the performance of state-of-the-art ΣΔMs, their trends, challenges, and circuits and systems solutions, constitutes a powerful and empirical tool for designers to select the optimum ΣΔM architecture, circuit implementation, technology process, etc, for a given set of specifications and/or a particular application. With this objective in mind and following the practical approach considered throughout this book, this chapter gives an overview of reported cutting-edge ΣΔM ICs fabricated in CMOS technologies. The main purpose of this overview is to make an exhaustive analysis of the evolution trends, design challenges, and reported solutions in order to extract practical conclusions and guidelines which may be useful for designers in their own projects.

The study described here is inspired by previous surveys on state-of-the-art performance of ADCs [3–10]. In the majority of cases, these surveys cover different kinds of ADCs, not only ΣΔMs. Among others, the surveys published by Walden [4], Murmann [5], and more recently Jonsson [8, 9] and Manganaro [10], are particularly exhaustive and accurate. Compared to these works, the study described in this book focuses on ΣΔM ICs; emphasizing on the different architectures and circuits used in many different applications. Moreover, in addition to analyzing statistics and comparing data collected from publications and extracted from standard performance metrics, the study presented in this chapter gives an overview of cutting-edge ΣΔM circuits and systems techniques which are at the frontier of state-of-the-art performance, highlighting the incoming trends and challenges as well as the proposed efficient solutions.

The data used for this study was mainly collected from the *IEEE Journal of Solid-State Circuits* as well as the major conferences sponsored by the *IEEE Solid-State Circuits Society (SSCS)*, namely *International Solid-State Circuits Conference (ISSCC)*, *European Solid-State Circuits Conference (ESSCIRC)*, *Custom Integrated Circuits Conference (CICC)*, *Symposium on VLSI Circuits (VLSI)*, *Asian Solid-State Circuits Conference*

CMOS Sigma-Delta Converters: Practical Design Guide, First Edition. José M. de la Rosa and Rocío del Río.

(ASSCC), and *Radio Frequency Integrated Circuits Symposium (RFIC)*. In addition to these SSCS publications, some other remarkable ΣΔM ICs published in the *IEEE Transactions on Circuits and Systems (parts I and II)* and the *IEEE International Symposium on Circuits and Systems (ISCAS)* have also been collected in the survey. Overall, more than 300 ΣΔM ICs have been analyzed in detail and considered in this review. Although the works under study include papers published since 1990 to June 2012, data was more exhaustively collected in the last 10 years.

All data collected and analyzed in the survey is compiled in an Excel spreadsheet that is available online at http://www.imse-cnm.csic.es/~jrosa/CMOS-SDMs-Survey-IMSE-JMdelaRosa.xlsx. The database in this spreadsheet file is periodically kept up to date every six months and aims to be a complement to the popular and highly cited Murmann's ADC survey data collection [11].

Following this introduction, the chapter is organized as follows. Section 5.1 gives an overview of the state-of-the-art ΣΔMs, comparing their performance with Nyquist-rate ADCs. The diverse families of ΣΔM architectures and circuit techniques are exhaustively analyzed and compared in Section 5.2 in order to extract practical and empirical design guidelines. Section 5.3 reviews some of the most significant cutting-edge ΣΔM techniques, analyzing the design trends and challenges in the frontiers of ΣΔMs. Finally, the chapter is concluded with a classified description of state-of-the-art references.

5.1 Overview of the State of the Art on ΣΔMs

Tables 5.1–5.13 sum up the performance of the state-of-the-art ΣΔM ICs considered in this survey. For the sake of clarity and simplicity, the ICs included in the study have been classified according to their architecture/circuit characteristics into the following tables:

- SC-ΣΔMs: single-loop single-bit LP-ΣΔMs (Table 5.1), single-loop multibit LP-ΣΔMs (Table 5.2), cascade single-bit LP-ΣΔMs (Table 5.3), cascade multibit LP-ΣΔMs (Table 5.4), and BP-ΣΔMs (Table 5.5).
- CT-ΣΔMs: single-loop single-bit LP-ΣΔMs (Table 5.6), single-loop multibit LP-ΣΔMs (Table 5.7), cascade LP-ΣΔMs (Table 5.8), and BP-ΣΔMs (Table 5.9).
- ΣΔMs with time-coded quantization (Table 5.10).
- Hybrid ΣΔMs: CT/DT ΣΔMs, active/passive ΣΔMs, digital-based ΣΔMs, and ΣΔM/Nyquist-rate ADCs (Table 5.11).
- Reconfigurable ΣΔMs implemented using SC (Table 5.12) and CT circuits (Table 5.13).

In all cases, the main features of each reported IC are summarized in terms of the following performance metrics: DR (measured in bits), B_w, OSR, technology process, supply voltage, and power consumption.[1] In the case of BP-ΣΔMs (Tables 5.5 and 5.9), the notch frequency f_n is also given. A schematic description of each ΣΔM topology is sketched in the tables, highlighting the loop-filter order, the number of stages (in cascade ΣΔMs), the number of bits of the embedded quantizer, the type of quantization technique (in time-based quantizers), the operation mode and standard covered (in the case of reconfigurable ΣΔMs), etc. A more complete description of the modulators—not shown

[1] The data corresponding to the power consumption accounts only for the ΣΔM, excluding the power consumed by the decimation filter.

Table 5.1 State-of-the-art SC single-loop single-bit LP-ΣΔMs

Ref.	DR (bit)	B_w (Hz)	OSR	Architecture	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[12]	13.8	2.00E+04	100	3rd-ord	0.18 μm/0.7	3.60E−05	0.06
[12]	12.3	8.00E+03	125	2nd-ord	0.35 μm/1.2	5.60E−06	0.07
[13]	13.5	2.00E+04	50	4th-ord	0.13 μm/0.9	6.00E−05	0.13
[14]	18.5	2.40E+04	—	6th-ord	0.35 μm/5–3.3	2.30E−01	12.93
[15]	14.4	2.00E+04	100	3rd-ord	0.18μm/1	1.30E−04	0.15
[16]	10.5	5.00E+04	65	3rd-ord	90 nm/0.65	2.70E−05	0.19
[17]	16.3	2.50E+04	100	2nd-ord	0.18μm/0.7	8.70E−04	0.21
[18]	12.5	1.60E+04	48	3rd-ord	0.5 μm/0.9	4.00E−05	0.22
[12]	12.2	1.20E+02	42	3rd-ord	0.35 μm/1.5	3.80E−07	0.35
[19]	13.0	8.00E+03	64	2nd-ord	0.18 μm/0.65	4.55E−05	0.35
[20]	16.0	2.00E+04	64	4th-ord	0.5 μm/1.5	1.00E−03	0.38
[21]	15.3	2.50E+05	96	5th-ord	0.8 μm/3.3	4.30E−02	2.13
[22]	14.6	1.00E+03	64	5th-ord	0.35 μm/1.5	2.00E−05	0.40
[23]	9.8	1.00E+04	70	3rd-ord	0.13 μm/0.25	7.50E−06	0.41
[24]	13.7	1.25E+06	50	5th-ord	0.25 μm/2.4	1.40E−02	0.42
[25]	14.4	1.35E+05	48	4th-ord	0.25 μm/2.7	2.84E−03	0.50
[26]	12.2	1.00E+06	31	2nd-ord	0.25 μm/2.4	5.00E−03	0.55
[27]	11.8	3.00E+05	166	2nd-ord	0.13 μm/1.2	2.98E−04	0.14
[19]	11.0	1.60E+04	32	2nd-ord	0.18 m/0.65	4.55E−05	0.69
[28]	13.1	1.00E+05	520	2nd-ord	0.13 μm/1.5	1.28E−03	0.73
[29]	14.4	2.50E+04	100	3rd-ord	0.35 μm/1	9.50E−04	0.88
[30]	10.0	2.00E+04	100	3rd-ord	0.18 μm/0.7	3.60E−05	0.88
[31]	8.7	8.35E+05	30	2nd-ord	45 nm/1.1	6.30E−04	0.89
[32]	12.2	8.00E+03	64	2nd-ord	0.18 μm/0.7	8.00E−05	1.06
[33]	16.7	1.10E+04	64	4th-ord	0.5 μm/2.5	2.50E−03	1.07
[28]	9.9	5.00E+05	104	2nd-ord	0.13 μm/1.5	1.28E−03	1.33
[34]	11.5	7.80E+04	64	4th-ord	0.18 μm/0.5	8.60E−04	1.90
[35]	16.4	4.80E+03	256	2nd-ord	0.7 μm/5	1.71E−03	2.06
[28]	8.1	1.00E+06	52	2nd-ord	0.13 μm/1.5	1.28E−03	2.32
[36]	11.5	2.00E+04	64	2nd-ord	0.18 μm/1.8	4.20E−04	3.63
[37]	12.0	3.40E+03	74	2nd-ord	0.7 μm/1.5	1.01E−04	3.63
[38]	16.0	2.50E+04	256	2nd-ord	1 μm/5	1.38E−02	4.21
[39]	8.7	3.84E+06	24	3rd-ord	0.25 μm/2.5	1.35E−02	4.23
[40]	14.3	3.00E+03	128	2nd-ord	0.5 μm/1.5	5.50E−04	4.54
[41]	12.0	8.00E+03	64	3rd-ord	1.2 μm/2	3.40E−04	5.19
[42]	15.3	3.50E+03	286	2nd-ord	0.6 μ/1.8	2.00E−03	7.08
[43]	13.0	8.00E+03	64	2nd-ord	0.25 μm/1.8	1.00E−03	7.63
[44]	15.3	2.50E+05	64	5th-ord	0.6 μm/5	2.10E−01	10.41
[45]	8.5	2.50E+02	16	3rd-ord	0.35 μm/1.8	2.20E−06	12.15
[46]	13.4	9.77E+04	128	2nd-ord	1.2 μm/5	2.59E−02	12.27
[47]	12.0	5.00E+04	102	2nd-ord	0.35 μm/1	5.60E−03	13.67
[48]	21.0	4.00E+02	320	4th-ord	3 μm/10	2.50E−02	14.90
[47]	13.0	2.00E+04	256	2nd-ord	0.35 μm/1	5.60E−03	17.09
[39]	14.0	2.00E+04	192	3rd-ord	0.25 μm/2.5	1.15E−02	17.55
[49]	20.0	4.00E+02	320	4th-ord	0.6 μm/5	1.60E−02	19.07
[50]	13.5	2.50E+05	64	4th-ord	1.5 μm/5	1.60E−01	27.62
[51]	10.3	1.00E+04	25–125	4th-ord	0.18 μm/1.8	7.50E−04	29.13
[52]	14.5	8.00E+03	256	2nd-ord	3 μm/5	1.20E−02	32.37
[53]	13.8	1.00E+03	250	3rd-ord	2 μm/5	9.40E−04	32.95
[54]	14.2	9.77E+03	256	2nd-ord	2 μm/5	1.30E−02	35.37

The ΣΔM ICs are sorted by FOM. Engineering notation is used for B_w and power consumption.

Table 5.2 State-of-the-art SC single-loop multibit LP-ΣΔMs

Ref.	DR (bit)	B_w (Hz)	OSR	Architecture	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[55]	13.7	1.00E+05	16	2nd-ord(9L)	0.18 μm/1.5	1.40E−04	0.05
[56]	13.8	2.00E+04	63	4th-ord(3L)	0.13 μm/0.5	3.50E−05	0.06
[57]	15.0	2.40E+04	64	3rd-ord(5b)	65 nm/0.6	1.33E−04	0.09
[58]	12.7	1.00E+06	24	2nd-ord(—)	0.18 μm/1.5	1.35E−03	0.10
[59]	13.9	1.10E+06	60	2nd-ord(5L)	0.18 μm/1.8	5.40E−03	0.16
[60]	15.0	1.60E+04	128	3rd-ord(3b)	0.18 μm	1.70E−04	0.16
[61]	16.3	2.50E+04	100	2nd-ord(18L)	0.18 μm/0.7	6.80E−04	0.17
[62]	13.2	4.20E+06	12	3rd-ord(4b)	0.18 μm/1.5	1.30E−02	0.17
[63]	10.7	1.94E+06	20	2nd-ord(5L)	90 nm/1.2	1.20E−03	0.19
[64]	14.0	2.20E+06	33	2nd-ord(4b)	0.18 μm/1.8	1.38E−02	0.19
[65]	13.3	1.90E+06	16	2nd-ord(4b)	0.18 μm/1.5	8.10E−03	0.21
[66]	15.0	2.40E+04	128	3rd-ord(4b)	0.18 μm/1	3.50E−04	0.22
[67]	12.3	1.25E+06	16	3rd-ord(9L)	0.18 μm/1.2	3.30E−03	0.25
[68]	11.0	1.60E+04	32	2nd-ord(—)	0.18 μm/1	1.70E−05	0.26
[69]	13.4	1.10E+06	47	2nd-ord(3b)	0.13 μm/1.5	7.00E−03	0.30
[70]	10.2	2.00E+07	8	3rd-ord(4b)	0.18 μm/1.8	1.60E−02	0.35
[22]	14.6	1.00E+03	16	5th-ord(1.5b)	0.35 μm/1.8	2.00E−05	0.40
[71]	13.0	2.50E+06	12	4th-ord(4b)	0.18 μm/1.6	1.92E−02	0.47
[72]	9.8	6.00E+06	8	3rd-ord(5b)	0.18 μm/1.8	6.18E−03	0.58
[73]	14.0	1.25E+06	32	5th-ord(1.5b)	0.25 μm/2.5	2.40E−02	0.59
[74]	11.3	4.00E+06	13	4th-ord(4b)	90 nm/1.2−3	1.18E−02	0.60
[75]	13.7	1.25E+07	8	5th-ord(4b)	0.18 μm/1.8	2.00E−01	0.60
[69]	14.4	3.00E+05	175	2nd-ord(3b)	0.13 μm/1.5	8.00E−03	0.63
[76]	10.7	1.00E+06	20	2nd-ord(4b)	90 nm/1.3	2.10E−03	0.63
[77]	12.8	2.00E+05	65	2nd-ord(5b)	0.13 μm/1.5	2.40E−03	0.82
[77]	12.0	2.00E+05	65	2nd-ord(5b)	0.13 μm/1.2	1.40E−03	0.85
[76]	12.5	2.00E+05	50	2nd-ord(4b)	90 nm/1.3	2.10E−03	0.91
[78]	15.0	2.40E+04	128	3rd-ord(3L)	0.13 μm/0.9	1.50E−03	0.95
[79]	17.2	2.00E+04	154	2nd-ord(4b)	0.13 μm/3.3	9.90E−03	1.61
[80]	13.8	3.00E+05	96	2nd-ord(3b)	0.18 μm/1.8	1.50E−02	1.75
[81]	17.0	3.13E+04	128	2nd-ord(5b)	0.18 μm/3.3	1.47E−02	1.79
[82]	15.8	1.25E+06	24	3rd-ord(4b)	0.65 μm/5	2.95E−01	2.07
[83]	12.7	2.20E+06	8	2nd-ord(4b)	0.35 μm/3.3	6.20E−02	2.18
[84]	12.7	2.20E+06	8	5th-ord(4b)	0.35 μm/3.3	6.20E−02	2.18
[77]	8.0	2.00E+06	12	2nd-ord(5b)	0.13 μm/1.5	2.90E−03	2.81
[85]	12.8	6.25E+05	18	2nd-ord(6b)	0.18 μm/2.7	3.00E−02	3.30
[86]	18.7	2.00E+04	128	5th-ord(4b)	0.35 μm/5−3.3	6.80E−02	3.99
[85]	11.7	1.92E+06	12	2nd-ord(6b)	0.18 μm/2.7	5.00E−02	4.00
[87]	19.0	4.00E+02	512	2nd-ord(3b)	2 μm/5	2.18E−03	5.19
[88]	13.7	6.25E+05	12	4th-ord(4b)	0.25 μm/2.5	1.00E−01	6.01
[89]	20.3	2.00E+04	128	5th-ord(33L)	0.35 μm/5	3.30E−01	6.39
[88]	13.0	1.00E+06	12	4th-ord(4b)	0.25 μm/2.5	1.05E−01	6.41
[85]	13.5	2.00E+05	58	2nd-ord(6b)	0.18 μm/2.7	3.00E−02	6.47
[82]	12.0	6.25E+06	8	3rd-ord(4b)	0.65 μm/5	3.80E−01	7.42
[90]	13.7	2.50E+05	16	4th-ord(4b)	1.2 μm/5	5.80E−02	8.96
[91]	16.0	4.50E+01	512	1st-ord(3b)	0.35 μm/2.6	6.00E−05	10.17
[92]	19.3	4.80E+04	64	7th-ord(3L)	0.8 μm/5	7.60E−01	12.26
[93]	16.7	2.00E+04	64	2nd-ord(5b)	0.5 μm/3.3	7.04E−02	16.53
[94]	16.2	2.40E+04	64	2nd-ord(5b)	0.5 μm/3.3	6.86E−02	18.72

Table 5.2 State-of-the-art SC single-loop multibit LP-ΣΔMs (cont.)

Ref.	DR (bit)	B_w (Hz)	OSR	Architecture	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[85]	15.3	1.80E+04	639	2nd-ord(6b)	0.18 μm/2.7	3.00E−02	20.37
[95]	18.0	2.00E+04	154	5th-ord(17L)	0.35 μm/5	3.00E−01	27.83
[96]	15.7	2.00E+04	64	2nd-ord(3b)	1.2 μm/5	6.75E−02	32.82
[97]	16.0	1.95E+04	128	3rd-ord(1b-5b)	2 μm/5	8.50E−02	33.26
[98]	14.3	1.00E+03	16	5th-ord(17L)	0.35 μm/1.8	9.00E−03	223.09

The ΣΔM ICs are sorted by FOM. Architecture is expressed in terms of the modulator order and the number of bits/levels of the embedded quantizer. For instance, 2nd-ord(4b) represents a second-order loop filter with 4-bit quantizer and 5th-ord(33L) means a fifth-order architecture with 33-level quantizer.

Table 5.3 State-of-the-art SC cascade single-bit LP-ΣΔMs

Ref.	DR (bit)	B_w (Hz)	OSR	Architecture	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[99]	12.8	2.00E+04	64	2-2	0.13 μm/1	1.80E−05	0.06
[99]	11.50	4.00E+04	64	2-2	0.13 μm/1.2	1.80E−02	0.15
[19]	12.2	1.60E+04	32	2-1	0.18 μm/0.65	6.18E−05	0.42
[19]	13.0	8.00E+03	64	2-1	0.18 μm/0.65	6.18E−05	0.47
[100]	12.2	1.00E+06	31	2-2	0.25 μm/2.4	5.00E−03	0.53
[101]	13.6	1.00E+06	56	3-1	65 nm/1	1.70E−02	0.67
[102]	16.1	2.50E+04	80	2-1	0.8 μm/1.8	2.50E−03	0.71
[103]	14.3	2.00E+04	100	2-2	0.18 μm/1	6.60E−04	0.82
[104]	13.5	1.80E+05	36	2.2	0.4 μm/1.8	5.00E−03	1.20
[105]	18.1	2.00E+04	128	2-1	0.35 μm/3.3	1.47E−02	1.31
[106]	14.1	2.00E+05	100	2-1	130 nm/1.2	1.10E−02	1.57
[107]	15.0	1.10E+06	24	2-1-1	0.5 μm/3.3	2.00E−01	2.77
[108]	15.7	1.60E+05	64	2-1	1.2 μm/5	6.50E−02	3.82
[109]	14.8	1.00E+06	24	2-1-1	1 μm/5	2.30E−01	4.03
[110]	14.0	1.10E+06	24	2-2-2	0.35 μm/3.3	1.50E−01	4.16
[111]	16.7	2.20E+04	128	2-1	0.6 μm/3	2.20E−02	4.86
[112]	17.0	2.50E+04	128	2-1	1 μm/5	4.70E−02	7.17
[113]	18.1	1.25E+04	64	2-2	0.6 μm/5	7.50E−02	10.68
[114]	15.0	9.00E+04	64	1-1-1	1.5 μm/5	7.60E−02	12.89
[115]	18.1	2.40E+04	128	2-2	0.7 μm/5	5.00E−01	35.91
[116]	14.8	2.50E+04	64	2-2	3 μm/5	7.40E−02	51.03

The cascade topology is expressed in terms of the stage orders. For instance, notation 2-2 is used for representing a fourth-order cascade ΣΔM made up of two second-order stages.

Table 5.4 State-of-the-art SC cascade multibit LP-ΣΔMs

Ref.	DR (bit)	B_w (Hz)	OSR	Architecture	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[117]	12.2	5.00E+06	13	2(4b)-2(4b)	0.13 μm/1.2	8.90E−03	0.19
[118]	11.4	2.00E+07	11	2-2(4b)	90 nm/1.2	2.79E−02	0.26
[119]	11.3	2.50E+06	8	1(4L)-1(4L)-1(4L)-1(4L)	65 nm/1.2	3.73E−03	0.29
[120]	12.5	6.25E+05	16	2(3b)-2(2b)	0.18 μm/1.2	2.10E−03	0.29
[121]	12.0	3.13E+06	8	0-3(17L)	0.18 μm/1.8	1.90E−02	0.74
[76]	9.4	2.00E+06	10	2(4b-dual)	90 nm/1.3	2.10E−03	0.79
[122]	10.4	1.92E+06	100	2-1(5L)	0.13 μm/1.2	4.30E−03	0.83
[123]	14.0	2.20E+06	16	2(3L)-1(3L)-1(3L)	0.25 μm/2.5	6.25E−02	0.87
[124]	10.8	2.00E+07	8	2-2(4b)	90 nm/1.4	7.80E−02	1.09
[122]	13.4	1.00E+05	195	2-1(5L)	0.13 μm/1.2	2.40E−03	1.11
[125]	15.0	2.00E+06	16	2(5b)-2(3b)-1(3b)	0.5 μm/2.5	1.50E−01	1.14
[126]	15.0	1.25E+06	8	2(4b)-1(4b)-1(4b)	0.5 μm/5	1.05E−01	1.28
[127]	13.0	2.20E+06	16	2-1-1(3b)	0.25 μm/2.5	7.17E−02	1.99
[127]	13.8	1.10E+06	32	2-1-1(3b)	0.25 μm/2.5	6.58E−02	2.10
[127]	12.7	2.20E+06	16	2-1-1(3b)	0.25 μm/2.5	6.58E−02	2.25
[128]	13.4	2.00E+04	64	2-2(1.5b)	0.35 μm/0.6	1.00E−03	2.31
[127]	13.7	1.10E+06	32	2-1-1(3b)	0.25 μm/2.5	7.17E−02	2.45
[129]	16.4	1.00E+06	8	2(5b)-2(3b)	0.25 μm/—	4.75E−01	2.75
[130]	13.0	1.10E+06	16	2-1-1(3b)	0.7 μm/5	5.50E−02	3.05
[131]	14.6	1.10E+06	29	2-1-1(2b)	0.35 μm/3.3	1.80E−01	3.29
[132]	13.0	7.81E+05	32	2-2(3b)	0.35 μm/2.5	5.00E−02	3.91
[127]	13.0	1.10E+06	16	2-1-1(4b)	0.35 μm/3.3	7.37E−02	4.09
[38]	12.0	1.05E+06	24	2-1(3b)	1 μm/5	4.10E−02	4.77
[127]	12.0	2.00E+06	16	2-1-1(4b)	0.35 μm/3.3	7.83E−02	4.78
[133]	10.2	8.33E+06	3	1-1-1-1-1-1-1-1(3L)	0.18 μm/1.8	9.50E−02	5.02
[110]	13.0	1.10E+06	24	2-2(5b)	0.35 μm/3.3	9.90E−02	5.49
[134]	8.5	4.00E+07	4	2(1.5b)-2(4b)	0.13 μm/1.2	1.75E−01	6.04
[135]	13.0	7.00E+05	16	2-2-2(3L)	0.7 μm/3.3	8.10E−02	7.06
[136]	14.3	1.00E+05	16	2(1.5b)-2(1.5b)-2(1.5b)	1.2 μm/5	4.00E−02	10.26

The number of bits per levels of the quantizer is given in parentheses. This way, notation 2-1(5L) is used for denoting a cascade 2-1 ΣΔM topology with 5-level quantizer in the second stage. If no parentheses are shown, single-bit quantization is assumed in a given modulator stage.

Table 5.5 State-of-the-art SC BP-ΣΔMs

Ref.	DR (bit)	f_n (Hz)	B_w (Hz)	OSR	Architecture	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[137]	14.4	4.00E+07	2.50E +06	12	4th-ord	0.18 μm/1.8	1.50E−01	1.39
[138]	13.3	2.00E+07	1.25E+06	32	4-4	0.35 μm/3	3.70E−02	1.47
[139]	11.7	4.00E+07	1.00E+06	6	4th-ord	0.18 μm/1.8	1.60E−02	2.46
[140]	13.2	1.00E+07	2.00E+05	33	2-0	0.25 μm/2.1	1.00E−02	2.66
[138]	11.7	2.00E+07	1.76E+06	23	4-4	0.35 μm/3	3.70E−02	3.16
[141]	15.7	1.26E+07	3.10E+05	65	4th-ord	0.18 μm/1.8	1.15E−01	3.61
[142]	13.4	1.07E+07	6.00E+04	0	4th-ord	0.25 μm/1	8.45E−03	6.65
[143]	12.0	1.60E+07	2.00E+06	16	6th-ord	0.25 μm/2.5	1.10E−01	6.71
[142]	12.0	1.07E+07	1.00E+05	0	4th-ord	0.25 μm/1	8.45E−03	10.03
[142]	12.7	1.07E+07	6.00E+04	0	4th-ord	0.25 μm/1	8.45E−03	10.58
[144]	11.7	3.25E+06	2.00E+05	33	4-2	0.8 μm/3	1.44E−02	10.82
[145]	14.5	1.07E+07	4.00E+05	46	4th-ord(4b)	0.15 μm/3.3	2.08E−01	11.22
[145]	15.3	1.07E+07	2.00E+05	93	4th-ord(4b)	0.15 μm/3.3	2.08E−01	12.89
[142]	10.0	1.07E+07	2.00E+05	0	4th-ord	0.25 μm/1	8.45E−03	20.63
[146]	12.6	5.66E+05	2.50E+05	20	2-2(3b)	0.25 μm/2.5	7.70E−02	24.81
[147]	9.0	2.00E+06	3.00E+04	133	4th-ord	2 μm/3.3	8.00E−04	26.04
[148]	11.7	2.00E+07	2.70E+05	148	4th-ord	0.35 μm/3	5.60E−02	31.17
[149]	12.2	2.00E+07	2.00E+05	200	4th-ord	0.6 μm/3.3	7.20E−02	38.26
[150]	9.7	3.25E+06	2.00E+05	32	3rd-ord	0.35 μm/3.3	1.87E−02	56.21
[148]	6.7	2.00E+07	3.84E+06	10	4th-ord	0.35 μm/3	5.60E−02	70.13
[151]	12.0	1.07E+07	2.00E+05	93	6th-ord	0.35 μm/3.3	1.16E−01	70.80
[145]	18.3	1.07E+07	3.00E+03	6167	4th-ord(4b)	0.15 μm/3.3	2.08E−01	107.41
[152]	10.8	3.75E+06	2.00E+05	25	4th-ord	0.8 μm/5	1.30E−01	182.29
[153]	6.8	1.07E+07	2.00E+05	107	2nd-ord	0.35 μm/1	1.20E−02	269.23

Table 5.6 State-of-the-art CT single-loop single-bit LP-ΣΔMs

Ref.	DR (bit)	B_w (Hz)	OSR	Architecture	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[154]	13.5	3.60E+07	50	4th-ord	90 nm/1.2	1.50E−02	0.018
[155]	14.5	1.00E+06	32	4th-ord	0.18 µm/1.8	2.00E−03	0.04
[156]	15.6	2.40E+04	128	3rd-ord	0.18 µm/1.8	1.10E−04	0.05
[157]	15.0	2.40E+04	127	3rd-ord	0.18 µm/1.8	1.10E−04	0.07
[156]	14.8	2.40E+04	128	3rd-ord	0.18 µm/1.8	1.22E−04	0.09
[158]	14.6	1.00E+06	32	4th-ord	0.18 µm/1.8	4.70E−03	0.09
[159]	12.8	2.00E+04	64	4th-ord	0.13 µm/0.6	2.86E−05	0.10
[160]	9.6	1.56E+07	32	4th ord	0.13 µm/1.2	4.00E−03	0.16
[161]	9.9	6.00E+07	50	3rd-ord	45 nm/1.4-1.8	2.00E−02	0.17
[162]	11.3	4.00E+06	18	3rd-ord	65 nm/0.95-1.25	3.60E−03	0.18
[163]	10.8	1.00E+07	32	3rd-ord	90 nm/1.2	6.80E−03	0.19
[164]	11.7	4.00E+06	64	3rd-ord	90 nm/1.2	5.50E−03	0.21
[165]	11.8	4.00E+07	12	4th ord	90 nm/1.2	6.96E−02	0.24
[166]	10.1	1.00E+07	15	5th-ord	0.11 µm/1.1	5.32E−03	0.25
[164]	12.7	1.92E+06	64	3rd-ord	90 nm/1.2	6.44E−03	0.26
[167]	14.0	6.00E+05	213	4th-ord	90 nm/1.3	5.40E−03	0.27
[168]	12.8	1.95E+06	32	5th-ord	65 nm/2.5	8.55E−03	0.30
[169]	9.7	1.50E+07	50	5th-ord	45 nm/1.1	9.00E−03	0.37
[164]	11.5	1.92E+06	64	3rd-ord	90 nm/1.2	4.34E−03	0.39
[164]	13.5	5.00E+05	90	3rd-ord	90 nm/1.2	5.00E−03	0.43
[170]	12.5	1.00E+06	140	2nd-ord	0.18 µm/1.8	6.00E−03	0.52
[171]	11.8	6.40E+06	32	3rd-ord	0.13 µm/1.2	2.52E−02	0.54
[171]	10.2	1.70E+07	12	3rd-ord	0.13 µm/1.2	2.52E−02	0.61
[172]	10.7	1.25E+06	64	4th-ord	0.13 µm/1.2	2.70E−03	0.65
[173]	11.3	2.00E+06	38	4th-ord	0.18 µm/1.8	6.60E−03	0.65
[174]	11.8	2.50E+04	48	3rd-ord	0.5 µm/1.5	1.35E−04	0.76
[175]	12.3	5.00E+05	64	5th-ord	0.18 µm/1.8	4.40E−03	0.87
[176]	13.3	1.00E+05	65	4th-ord	0.35 µm/2.5	1.80E−03	0.89
[177]	11.2	4.00E+06	50	3rd-ord	0.18 µm/1.8	1.69E−02	0.92
[178]	10.4	1.30E+06	50	4th-ord	0.11 µm/1.2	3.42E−03	0.99
[179]	9.4	5.00E+04	32	3rd-ord	0.5 µm/1.5	7.50E−05	1.11
[180]	11.3	3.15E+06	81	2nd-ord	90 nm/1.2	1.25E−02	1.26
[181]	12.4	1.23E+06	813	2nd-ord	0.18 µm/1.8	1.80E−02	1.35
[178]	11.4	4.28E+05	150	4th-ord	0.11 µm/1.2	3.42E−03	1.51
[182]	10.5	2.50E+06	16	2nd-ord	1.2 µm/3	1.20E−02	1.66
[183]	12.0	2.50E+04	64	3rd-ord	0.18 µm/0.5	3.70E−04	1.81
[184]	11.3	2.50E+04	48	3rd-ord	0.5 µm/1.5	2.50E−04	1.98
[185]	10.0	3.10E+06	64	5th-ord	0.6 µm/3.3	1.60E−02	2.52
[186]	13.0	4.00E+03	64	4th-ord	0.5 µm/2.2	2.00E−04	3.05
[187]	10.0	1.20E+06	16	1st-ord	0.35 µm/3.3	1.22E−02	4.96
[188]	11.7	2.56E+02	111	2nd-ord	0.18 µm/1.4	1.33E−05	7.97
[189]	9.6	1.00E+06	25	2nd-ord	2 µm/5	1.66E−02	10.70
[190]	10.0	2.50E+04	48	3rd-ord	0.5 µm/1.5	7.50E−04	14.65
[191]	12.8	1.00E+04	10 000	3rd-ord	0.18 µm/1.8	4.30E−03	31.21
[192]	12.8	1.00E+02	1500	2nd-ord	0.7 µm/5	2.00E−03	1373.36

Table 5.7 State-of-the-art CT single-loop multibit LP-ΣΔMs

Ref.	DR (bit)	B_w (Hz)	OSR	Architecture	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[193]	15.2	2.40E+04	64	3rd-ord(4b)	0.18 µm/1.8	9.00E−05	0.05
[194]	11.7	2.50E+07	16	3rd-ord(4b)	90 nm/1.2	8.50E−03	0.05
[195]	13.0	2.00E+07	16	3rd-ord(4b)	0.13 µm/1.2	2.00E−02	0.06
[196]	11.4	1.00E+07	15	5th-ord(3b)	110 nm/1.1	5.32E−03	0.10
[197]	10.2	2.50E+07	10	3rd-ord(4b)	90 nm/1.2	8.00E−03	0.14
[198]	13.0	2.00E+06	32	3rd-ord(3b)	65 nm/1.2	4.52E−03	0.14
[199]	12.0	2.00E+06	26	3rd-ord(4b)	0.13 µm/1.5	3.00E−03	0.18
[200]	12.2	1.00E+06	32	3rd-ord(4b)	0.13 µm/1.2	2.20E−03	0.23
[201]	11.3	5.00E+06	10	3rd-ord(4b)	0.13 µm/1.2	6.00E−03	0.25
[202]	11.3	1.50E+07	10	3rd-ord(4b)	0.18 µm/1.8	2.07E−02	0.27
[203]	11.5	1.00E+07	13	3rd-ord(3.5b)	130 nm/1.2	1.80E−02	0.31
[204]	13.2	8.00E+06	16	4th-order(4b)	65 nm/1.3	5.00E−02	0.33
[205]	12.7	2.00E+07	23	4th-ord(4b)	0.13 µm/1.5	8.70E−02	0.34
[206]	13.0	1.20E+07	10	3rd-ord(6b)	0.5 µm/2.5	7.50E−02	0.38
[207]	11.3	1.25E+08	16	3rd-ord(4b)	45 nm/1.1,1.8	2.60E−01	0.41
[208]	9.7	2.50E+07	405	5th-ord(4b)	0.18 µm/1.8	1.80E−02	0.44
[209]	11.0	2.00E+07	24	3rd-ord(3b)	130 nm/1.3	3.80E−02	0.46
[210]	13.3	1.00E+07	32	5th-ord(3b)	0.18 µm/1.8	1.00E−01	0.50
[211]	14.0	1.00E+05	130	2nd-ord(3b)	65 nm/1	2.10E−03	0.64
[212]	13.9	2.50E+06	12	5th-ord(4b)	0.25 µm/2.5	5.00E−02	0.65
[207]	10.5	1.25E+08	16	3rd-ord(4b)	45 nm/1.1	2.60E−01	0.72
[211]	10.2	1.92E+06	16	2nd-ord(3b)	65 nm/1	3.20E−03	0.72
[213]	10.0	2.00E+06	26	3rd-ord(3b)	0.13 µm/1.2	3.00E−03	0.73
[214]	14.0	1.20E+05	54	4th-ord(3b)	0.13 µm/1.25	3.00E−03	0.76
[215]	10.0	1.92E+06	48	1st-ord(5b)	130 nm/1.2	3.10E−03	0.79
[216]	10.0	1.92E+06	48	1st-ord(5b)	0.13 µm/1.2	3.10E−03	0.80
[217]	10.8	2.00E+07	16	3rd-ord(4b)	0.13 µm/1.2	5.80E−02	0.82
[218]	12.5	6.00E+05	42	3rd-ord(2b)	90 nm/1.5	6.00E−03	0.86
[219]	11.5	2.00E+06	32	5th-ord(3L)	0.18 µm/1.8	1.10E−02	0.95
[220]	14.9	2.00E+04	300	2nd-ord(4b)	45 nm/1.1	1.20E−03	0.98
[221]	11.0	1.50E+07	10	4th-ord(4b)	0.13 µm/1.5	7.00E−02	1.14
[222]	11.0	1.50E+07	10	4th-ord(4b)	0.13 µm/1.5	7.50E−02	1.22
[223]	10.0	2.00E+07	16	3rd-ord(4b)	0.13 µm/1.2	5.80E−02	1.40
[224]	14.4	1.00E+06	16	3rd-ord(5b)	0.5 µm/–	6.20E−02	1.43
[225]	8.7	2.00E+07	16	2nd-ord(3b)	0.25 µm/2.5	3.20E−02	1.92
[226]	8.9	2.00E+07	5	3rd-ord(4b)	0.18 µm/1.8	1.03E−01	5.39

Table 5.8 State-of-the-art CT cascade LP-ΣΔMs

Ref.	DR (bit)	B_w (Hz)	OSR	Architecture	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[227]	12.5	2.00E+07	8.5	2-2 I/Q	90 nm/1.2	5.60E−02	0.24
[228]	11.7	1.00E+07	5	1(3L)-1(1b)	0.18 µm/2	4.80E−02	0.72
[229]	11	1.80E+07	10	2-1-1(4b)	0.18 µm/1.8	1.83E−01	2.48
[230]	11	1.80E+07	10	2-1-1(4b)	0.18 µm/1.8	1.83E−01	2.48
[231]	10.87	2.00E+07	8	2-2(4b) I/Q	0.18 µm/1.8	2.16E−01	2.89
[231]	10.87	1.00E+07	8	2-2(4b)	0.18 µm/1.8	1.22E−01	3.27

I/Q is used for denoting a quadrature topology.

Table 5.9 State-of-the-art CT BP-ΣΔMs

Ref.	DR (bit)	f_n (Hz)	B_w (Hz)	OSR	Architecture	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[232]	11.3	2.4–2.5E+09	2.00E+07	80	6th-ord	40 nm	2.00E−02	0.19
[233]	14.7	1.30E+05	2.00E+05	65	3rd-ord	0.25 µm/1.8	2.70E−03	0.25
[234]	10.0	2.40E+09	6.00E+07	25	6th-ord	90 nm/1	4.00E−02	0.33
[235]	9.9	8.50E+01	5.00E+06	95	2nd-ord	0.18 µm/1.8	6.00E−03	0.36
[236]	9.7	2.00E+08	2.40E+07	16	4th-ord	65 nm/1.25	1.20E−02	0.39
[237]	12.0	—	3.84E+06	20	5th-ord	0.18 µm/2.9	1.41E−02	0.45
[237]	13.5	—	1.23E+06	32	5th-ord	0.18 µm/2.9	1.31E−02	0.46
[238]	11.0	2.00E+06	1.00E+06	24	2nd-ord(4b)	0.18 µm/1.8	2.20E−03	0.54
[239]	12.0	0–1E+09	1.50E+08	12	6th-ord	65 nm/1	5.50E−01	0.80
[240]	13.3	1.00E+05	2.70E+05	24	4th-ord	0.25 µm/2	4.60E−03	0.83
[241]	14.7	4.40E+07	8.50E+06	16	4th-ord	0.18 µm/2.9	3.75E−01	0.83
[242]	8.3	2.44E+09	2.80E+07	64	4th-ord	0.13 µm/1.2	1.50E−02	1.00
[243]	9.3	2.25E+08	4.00E+06	13	4th-ord	65 nm/1	1.30E−02	2.67
[244]	13.3	1.07E+07	2.00E+05	53	5th-ord	0.25 µm/2.5	1.10E−02	2.73
[245]	12.7	1.00E+07	2.00E+04	250	2nd-ord	1.2 µm/3.3	2.50E−04	3.42
[246]	6.9	2.44E+09	8.00E+07	38	4th-ord	40 nm/1.1	5.28E−02	3.60
[247]	11.3	2.00E+08	1.00E+07	40	4th-ord	0.18 µm/1.8	1.60E−01	3.72
[234]	10.0	2.44E+09	6.00E+07	60	6th-ord	90 nm/1.2	4.00E−02	4.08
[242]	8.2	2.44E+09	2.50E+07	64	4th-ord	0.13 µm/1.2	1.90E−02	4.65
[237]	15.0	1.00E+05	2.00E+04	650	5th-ord	0.18 µm/2.9	9.10E−03	6.94
[248]	16.0	1.07E+07	2.00E+05	104	5th-ord	0.18 µm/1.8	2.10E−01	8.01
[249]	9.3	4.09E+06	2.00E+06	31	2nd-ord	0.25 µm/1.8	2.05E−02	8.36
[250]	9.0	9.00E+08	9.00E+06	55	4th-ord	65 nm/1.2	8.00E−02	8.62
[249]	8.1	4.09E+06	4.00E+06	16	2nd-ord	0.25 µm/1.8	2.05E−02	9.67
[251]	7.7	2.22E+09	8.00E+07	160	4th-ord	40 nm/1.1	1.64E−01	9.86
[248]	14.0	1.07E+07	5.00E+05	42	5th-ord	0.18 µm/1.8	2.10E−01	12.82
[252]	8.4	4.73E+07	3.84E+06	25	4th-ord	0.35 µm/3.3	4.50E−02	16.99
[248]	19.3	1.07E+07	3.00E+03	6950	5th-ord	0.18 µm/1.8	2.10E−01	54.22
[253]	8.3	7.97E+08	1.00E+06	1600	4th-ord	0.13 µm/1.2	3.00E−02	58.06
[254]	10.8	1.07E+07	2.00E+05	100	6th-ord	0.5 µm/5	6.00E−02	84.13
[255]	6.7	2.44E+09	1.00E+06	1628	2nd-ord	0.13 µm/1.3	2.60E−02	126.40
[252]	9.2	4.73E+07	2.00E+05	473	2nd-ord	0.35 µm/3.3	4.50E−02	195.30
[256]	6.7	7.00E+07	2.00E+05	700	2nd-ord	0.5 µm/2.5	3.90E−02	937.79
[257]	7.2	1.00E+08	2.00E+05	1000	4th-ord	0.35 µm/3.3	1.65E−01	2805.49
[258]	6.0	1.00E+09	5.00E+05	4000	4th-ord	0.18 µm/1.8	2.90E−01	4531.25
[257]	8.7	1.00E+08	2.00E+05	1000	4th-ord	0.35 µm/2.7–3.3	3.30E−01	5678.56

Table 5.10 State-of-the-art ΣΔMs with time-coded quantization

Ref.	DR (bit)	B_w (Hz)	OSR	Architecture	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[259]	13.2	2.00E+07	45	1st-ord—VCO	0.13 μm/1.5	2.00E−02	0.05
[260]	12.8	1.00E+07	30	2nd-ord—VCO	90 nm/1.2	1.60E−02	0.11
[261]	12.5	4.00E+06	12.5–150	1(4b)—VCO	0.13 μm/1.2	6.10E−03	0.13
[262]	12.7	2.50E+07	8	5th-order—VCO	0.18 μm/1.8	4.80E−02	0.14
[263]	9.8	2.00E+07	16	3rd-ord—TEQ	65 nm/1	7.00E−03	0.19
[264]	11.2	1.80E+07	32	Ring-oscillator	65 nm/1	1.70E−02	0.20
	11.8	9.00E+06	64	Modulator		1.70E−02	0.26
[265]	11.7	2.00E+07	25	2nd-ord(5b)—VCO	0.13 μm/1.2	4.00E−02	0.30
	12.6	4.50E+06	128			1.70E−02	0.30
	11.5	4.50E+06	64			8.00E−03	0.31
[266]	9.7	2.00E+07	6	2nd-ord—TDC	65 nm/1.3	1.05E−02	0.32
[267]	13.00	1.00E+05	25–250	1-1-1—TDC	0.13 μm/1.2	7.00E−04	0.43
[268]	9.7	1.00E+06	78	2nd-ord—TDC	90 nm/1.2	1.30E−03	0.80

TDC stands for time-to-digital converter; TEQ stands for time-encoding quantizer.

Table 5.11 State-of-the-art hybrid ΣΔMs

Ref.	DR (bit)	B_w (Hz)	OSR	Hybrid Technique	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[269]	16.7	2.42E+04	135	CT-DT	40 nm/1	5.00E−04	0.10
[270]	13.8	5.00E+06	8	ΣΔM-pipeline	0.18 μm/1.8	2.20E−02	0.15
[271]	13.0	5.00E+06	9	Active–passive	55 nm/1.3	1.30E−02	0.16
[272]	12.2	5.00E+06	16	ΣΔM-cyclic	0.18 μm/1.8	7.90E−03	0.16
[273]	12.2	1.56E+06	8	ΣΔM/two-step	0.18 μm/1.2	2.60E−03	0.18
[274]	11.2	1.20E+06	15000	Active–passive	65 nm/1.4	1.16E−03	0.21
[275]	10.9	2.00E+06	38	Active–passive	0.28 μm/1.5	2.70E−03	0.35
[276]	12.5	2.00E+05	375	Digital-based	65 nm/1.3	9.50E−04	0.41
[228]	11.7	1.00E+07	5	ΣΔM-cyclic	0.18 μm/2	4.80E−02	0.72
[215]	10.0	1.92E+06	48	ΣΔM-SAR	130 nm/1.2	3.10E−03	0.79
[277]	8.9	5.00E+06	8	Digital-based	0.18 μm/1.8	4.00E−03	0.85
[278]	11.0	3.13E+06	16	CT-DT	65 nm/1.1	1.10E−02	0.86
[279]	12.5	7.50E+06	16	CT-DT	0.18 μm/1.2	8.90E−02	1.02
[277]	12.4	2.00E+05	16	Digital-based	0.18 μm/1.8	4.00E−03	1.89
[280]	12.2	1.00E+07	4	ΣΔM-pipeline	0.18 μm/3.3	2.40E−01	2.55
[281]	15.9	4.80E+04	128	CT-DT	0.35 μm/3.3	1.80E−02	3.07
[282]	16.5	2.00E+04	256	CT-DT	65 nm/3.3	1.50E−02	4.10
[283]	16.7	2.00E+04	128	CT-DT	0.18 μm/3.3	3.73E−02	8.76
[284]	9.5	4.00E+06	500	Active–passive	90 nm/1.2	5.40E−02	9.32
[285]	13.0	1.10E+04	64	CT-DT	0.5 μm/1.8	1.70E−03	9.43
[286]	14.5	1.25E+06	8	ΣΔM-pipeline	0.6 μm/5	5.50E−01	9.49

Table 5.12 State-of-the-art SC adaptive/reconfigurable ΣΔMs

Ref.	Standard	DR (bit)	B_w (Hz)	OSR	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[85]	AMPS	15.0	1.8E+04	639	180 nm/2.7	3.00E−02	22.1
	GSM	13.2	2.0E+05	58		3.00E−02	5.4
	CDMA	12.5	6.75E+05	17		3.00E−02	2.9
	UMTS	11.4	1.92E+06	12		5.00E−02	2.8
[287]	GSM/EDGE	14.3	1.0E+05	130	130 nm/1.2,3.3	2.39E−02	0.7
	UMTS	12.8	1.92E+06	12		2.45E−02	0.3
	WLAN	10.8	1.0E+07	6		4.45E−02	0.6
[288]	GSM	14.1	2.0E+05	100	130 nm/3.3	2.52E−02	
	Bluetooth	13.2	1.0E+06	20		2.50E−02	
	WCDMA	10.2	4.0E+06	10		4.45E−02	
[289]	GSM	14.1	1.0E+05	130	130 nm/1.2	2.00E−03	0.6
	UMTS	12.5	1.92E+06	16		5.20E−03	0.2
		12.2	5.0E+06	16		1.36E−02	0.3
	LTE	11.6	1.0E+07	12		2.02E−02	0.3
		11.4	2.0E+07	5		3.47E−02	0.3
[290]	GSM	13.8	1.0E+05	250	90 nm/1.2	3.40E−03	1.2
	Bluetooth	12.5	5.0E+05	90		3.70E−03	0.6
	UMTS	10.7	2.0E+06	20		6.80E−03	1
[291]	GSM	12.7	1.0E+05	200	90 nm/1.2	4.60E−03	3.5
	Bluetooth	11.3	5.0E+05	80		5.30E−03	2.1
	GPS	11.6	1.0E+06	60		6.20E−03	1
	UMTS	10.7	2.0E+06	30		8.00E−03	1.2
	DVB-H	10.1	4.0E+06	15		8.00E−03	0.9
	WiMAX	8.5	5.0E+06	12		1.10E−02	1.6
	GSM	13.2	1.0E+05	256		4.90E−03	0.99

For the sake of clarity, and considering that reconfigurable ΣΔMs feature different performance depending on their operating mode, the ICs in this table have been sorted by the year of publication.

Table 5.13 State-of-the-art CT adaptive/reconfigurable ΣΔMs

Ref.	Standard	DR (bit)	B_w (Hz)	OSR	Tech./supply voltage (V)	Power (W)	FOM (pJ/conv)
[292]	GSM	15	2.00E+05	64	180 nm/1.6,2.9	9.00E−03	0.69
	Bluetooth	13.5	1.23E+06	32		8.20E−03	0.29
	UMTS	12	3.84E+06	40		7.60E−03	0.24
[293]	GSM	13.3	2.00E+05	65	90 nm/1.1,1.3	1.40E−03	0.35
	Bluetooth	12.2	1.00E+06	100		3.40E−03	0.37
	UMTS	8.3	1.00E+07	20		7.00E−03	1.07
[294]	EDGE	14.3	1.35E+05	96	65 nm/2.5	2.60E−03	0.47
	CDMA	13.3	6.15E+05	62		3.10E−03	0.25
	UMTS	11.8	1.92E+06	40		3.70E−03	0.26
[295]	WLAN	10	5.00E+06	32	90 nm/1.2	6.80E−03	0.24
	DVB	11.3	4.00E+06	64		5.50E−03	0.27
	UMTS	11.2	1.92E+06	64		4.34E−03	0.5
	Bluetooth	12.5	1.00E+06	90		5.00E−03	0.85
		11.2	1.00E+06	32		1.70E−02	3.7
		10.83	2.30E+06	32		1.70E−02	1.94
[296]		10.88	5.00E+06	32	65 nm/1.2,2.5	1.70E−02	0.91
		11.83	1.00E+06	64		1.70E−02	2.32
		11.66	2.30E+06	64		1.70E−02	1.13
		12.63	1.00E+06	128		1.70E−02	1.34
		11.5	1.00E+06	64		8.00E−03	0.75
		8.84	1.50E+07	7		1.05E−02	0.76
		9.84	1.00E+07	10.4		1.05E−02	0.57
[297]		10.9	2.50E+06	20.8	65 nm/1.3	8.50E−03	0.74
		10.2	5.00E+06	41.6		8.50E−03	1.04
	GSM	13.32	2.00E+05	128	90 nm/1	2.80E−03	0.68
	Bluetooth	12.33	5.00E+05	96		2.60E−03	0.5
	UMTS	11.66	2.00E+06	32		3.60E−03	0.28
[298]	DVB-H	11.16	4.00E+06	24		4.90E−03	0.27
	WLAN	9.01	2.00E+07	16		8.50E−03	0.41
[299]		10.5	8.00E+06	25	0.18 μm/1.8	4.76E−02	0.81
		9.11	1.60E+07	12.5		4.76E−02	1.29
[300]	UMTS	9.7	1.92E+06	31	180 nm/1.8	8.90E−03	0.95
	DVB	9.8	4.0E+06	25		1.21E−02	0.84

ICs are sorted by date of publication.

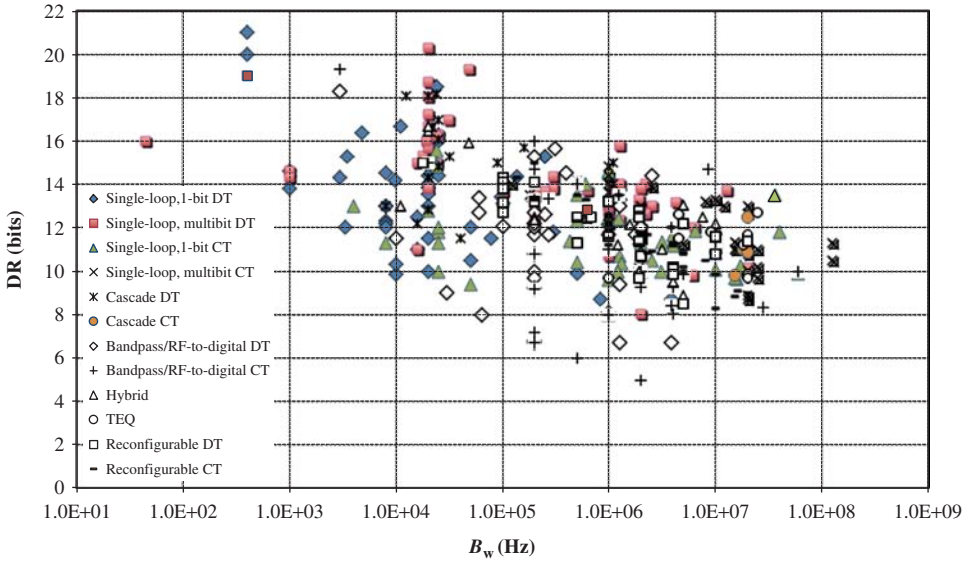


Figure 5.1 DR-versus-B_w conversion region of state-of-the-art CMOS ΣΔMs.

in the tables of this chapter for the sake of simplicity – can be found in the spreadsheet available on the Web.

5.1.1 DR-versus-B_w Conversion Region

Figure 5.1 represents DR versus B_w for the different families of ΣΔMs.[2] It can be noted in Figure 5.1 that a wide DR-versus-B_w *conversion region* is covered, ranging over seven decades in frequency and 16 bit in the DR axis. The highest DR is 21 bit, which was reported in 1994 by Kerth *et al.* who designed a fourth-order single-loop single-bit SC-ΣΔM in a 3-μm CMOS technology which digitized 400-Hz signals for seismic applications [48]. On the other hand, the highest value of B_w was reported by Shibata *et al.* at ISSCC in February 2012. This chip consisted of a DC-to-1-GHz tunable notch-frequency sixth-order multibit LP/BP CT-ΣΔM, implemented in a 65-nm CMOS technology, achieving 12-bit DR within a 150 MHz signal bandwidth [239].

Figure 5.2 compares the *conversion region* achieved by the state-of-the-art ΣΔMs with the one achieved by Nyquist-rate CMOS ADCs. The data in this figure was collected from Murmann's ADC survey [11], as well as from that used in Figure 5.1. It can be observed from Figure 5.2 that ΣΔ ADCs dominate the state-of-the-art DR-versus-B_w front in a wide frequency range: from hundreds of hertz to hundreds of megahertz. Beyond this range, Nyquist-rate are more competitive, reporting state-of-the-art performance in

[2] Some authors use to represent B_w versus DR or SNDR, often referred to as *aperture plot* [5, 10].

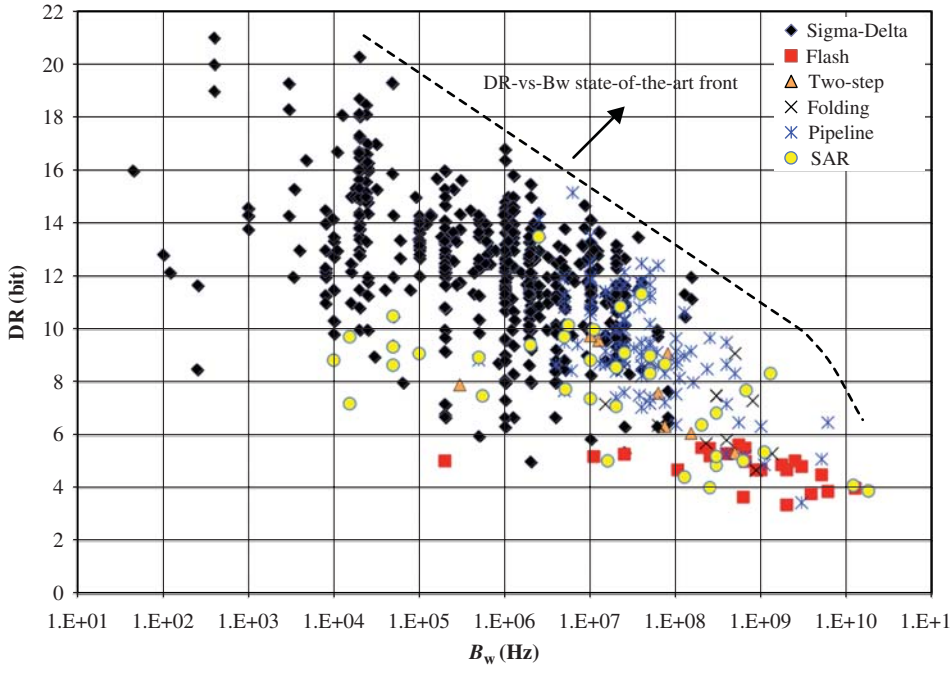


Figure 5.2 Comparison of the DR-versus-B_w conversion region of ΣΔM and Nyquist-rate ADCs. In all cases, only CMOS ICs are considered.

high-frequency applications with signal bandwidths ranging from hundreds of megahertz to dozens of gigahertz.

Analyzing the evolution over time of the B_w handled by ΣΔMs, it can be observed that the tendency is toward extending their range of application, particularly increasing the digitized signal bandwidth, although these designs—placed in the so-called *technology front*—are sometimes less power efficient, as will be discussed later. Figure 5.3 represents the B_w reported by ΣΔMs over time of publication for the years 1990–2012. It can be observed how ΣΔMs have evolved from low-frequency (high-resolution) to high-frequency (medium-to-low resolution) applications. Note also that the range of B_w is spread over time from two orders of magnitude to over six orders of magnitude. This trend benefits from the technology downscaling and it is expected to continue in the coming years.

5.1.2 Conversion Energy and Figures of Merit

Apart from their main specifications (DR (or SNDR) and B_w), the efficiency of ADCs is quantified in terms of the power consumption required to achieve these specifications. Taking all these *ingredients* into account, the amount of *energy* per converted sample E—also referred to as *conversion energy* [5]—can be calculated as [5, 9],

$$E(\mathrm{J}) \equiv \frac{P_w(\mathrm{W})}{\mathrm{DOR}\ (\mathrm{S/s})} \tag{5.1}$$

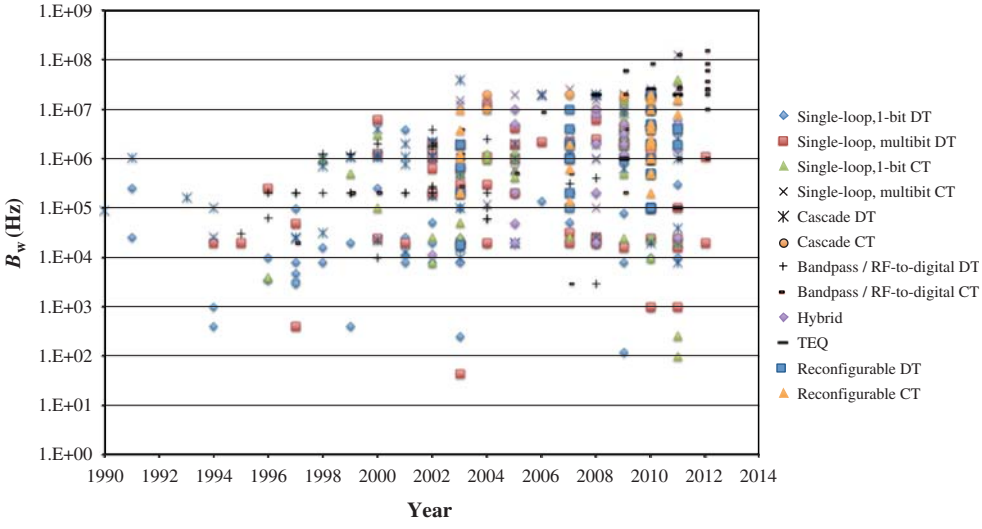


Figure 5.3 B_w spread of ΣΔMs over time for the years 1990–2012.

where P_w is the power consumption and DOR $\equiv 2B_w$ represents the digital output rate of the ADC, that is, the Nyquist rate, measured in samples per second (S/s).

Thus, it is common to analyze and graphically represent the efficiency of ADCs in the form of plots of *conversion energy* versus DR or SNDR—also known as *energy plots* [5, 10]. As an illustration, Figure 5.4 shows the *energy plot* of state-of-the-art ΣΔMs. Note that, in order to compute the overall performance achieved by a given ΣΔM, both

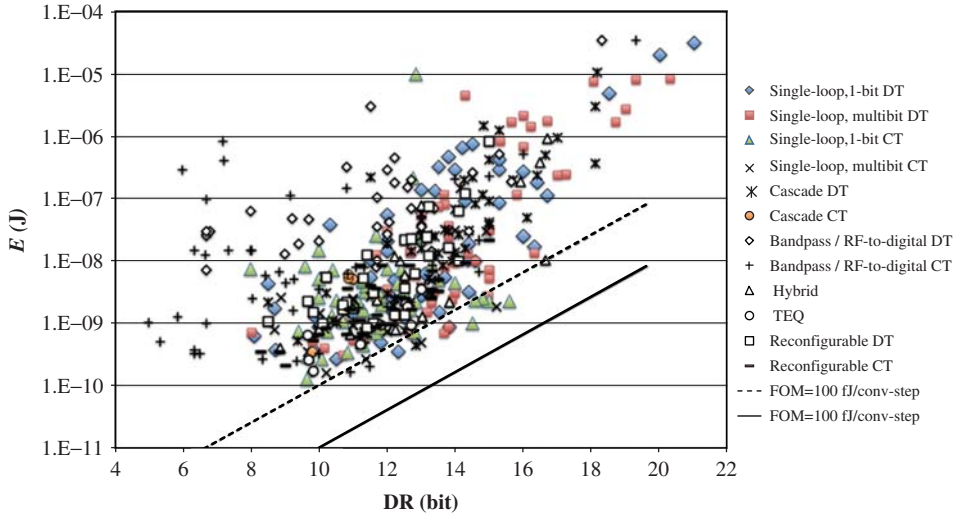


Figure 5.4 Energy plot of state-of-the-art ΣΔMs.

magnitudes involved in Figure 5.4—that is, the conversion energy E and the resolution (expressed in terms of either DR or SNDR)—must be taken into account. As expected, there is a direct relationship between DR and E, so that the larger the resolution, the more conversion energy is needed.

It is therefore convenient to express the trade-off highlighted in Figure 5.4 between the conversion energy and DR in a single performance metric to quantify the performance of a given ΣΔM and to compare the performance of different ADC architectures. Such a performance metric is often referred to as *figure of merit* (FOM), and accounts in a single formula the three basic specifications of an ADC, namely effective resolution, signal bandwidth, and power consumption.

Choosing the appropriate FOM is always a matter of debate and discussion. Among others, the following expressions that relate E with ENOB have been most frequently used for comparing the performance of ADCs [4, 102, 301]

$$\mathrm{FOM}_1|_{\mathrm{pJ/conv}} \equiv \frac{E(\mathrm{J})}{2^{\mathrm{ENOB(bit)}}} \cdot 10^{12} = \frac{P_{\mathrm{w}}(\mathrm{W})}{2^{\mathrm{ENOB(bit)}} \cdot \mathrm{DOR(S/s)}} \cdot 10^{12} \tag{5.2}$$

$$\mathrm{FOM}_2 \equiv 2kT \cdot \frac{3 \cdot 2^{2 \cdot \mathrm{ENOB(bit)}}}{E(\mathrm{J})} = 2kT \cdot \frac{3 \cdot 2^{2 \cdot \mathrm{ENOB(bit)}} \cdot \mathrm{DOR(S/s)}}{P_{\mathrm{w}}(\mathrm{W})} \tag{5.3}$$

where k is the Boltzmann constant and T is the chip temperature (measured in Kelvin).

Note that FOM_1 emphasizes power consumption, whereas FOM_2 emphasizes effective resolution. Therefore, the smaller the FOM_1 value and the larger the FOM_2 value, the "better" the ADC is. In this chapter, FOM_1 has been selected because it has been the most commonly used FOM in the recent years by the majority of the ΣΔM community. It should be noted that according to the definition of ENOB given in Equation 1.20, DR was used for computing ENOB and consequently for the value of FOM_1 as shown in Tables 5.1–5.13. However, in those ΣΔM ICs presenting a strongly nonlinear behavior, the effective resolution (i.e., ENOB) is indeed limited by the peak SNDR, as stated in Chapter 1 and hence, its corresponding FOM_1 is worse than the one shown in the tables. However, the purpose of this chapter is to analyze the overall performance and trends of ΣΔ ADCs, instead of the specifications achieved by a particular design. From this perspective, and considering that not all authors reported all data required to compute FOM_1, the analysis was based on expressions of FOM_1 derived from DR, instead of from SNDR.

Thus, following an approach similar to that proposed by Murmann [5], the results reported by state-of-the-art performance ΣΔMs are compared in Figure 5.4 with two straight lines for the numerical examples of $\mathrm{FOM}_1 = 100$ fJ/conversion-step and 10 fJ/conversion-step. Note that the majority of ΣΔMs reaching the lowest values of FOM_1 are placed between both lines and correspond to CT circuit implementations, showing the best efficiency of these kinds of ΣΔMs for resolutions in the order of 11–15-bit DR within 1–100 MHz bandwidths, as will be discussed later in this chapter.

Figure 5.5 represents FOM_1 versus B_{w} for the different ΣΔM architectures included in Tables 5.1–5.13. It can be seen how CT-ΣΔMs obtain better performance than SC-ΣΔMs for $B_{\mathrm{w}} > 1$ MHz, while SC-ΣΔMs are more efficient for lower values of B_{w}. In terms of architectures, it can be noted how LP-ΣΔMs obtain better values of FOM_1 than BP-ΣΔMs. This is due to the fact that CT BP-ΣΔMs digitize signals placed at a notch frequency f_{n}, which in the majority of cases is in the order of hundreds of megahertz

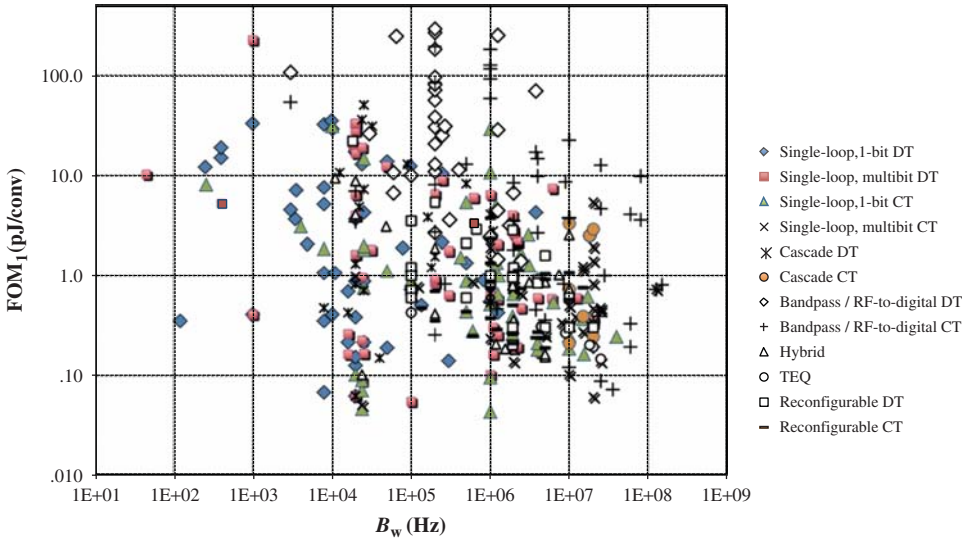


Figure 5.5 FOM_1 versus B_w corresponding to the state-of-the-art ΣΔMs under study.

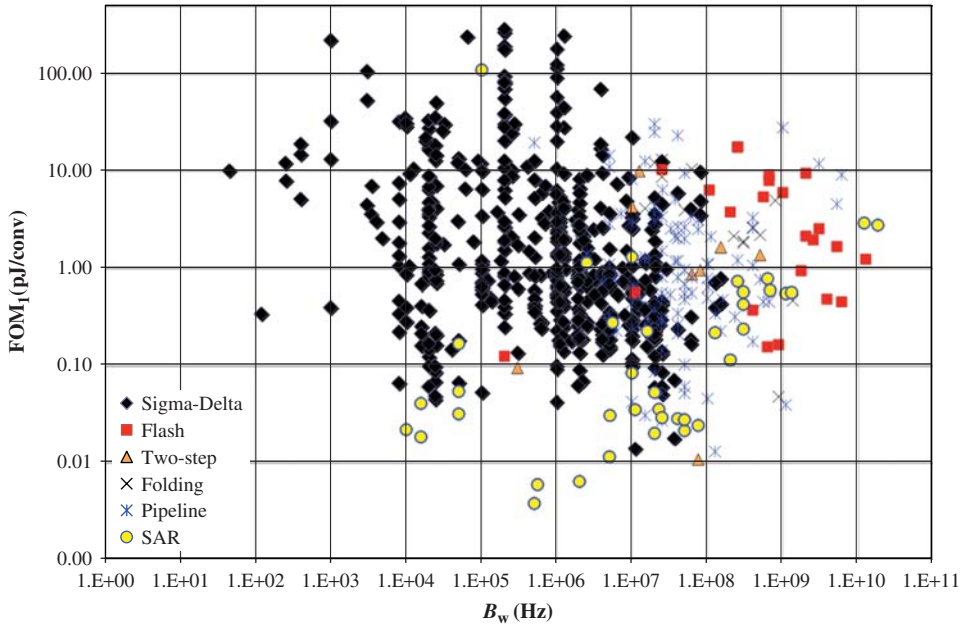


Figure 5.6 Comparison of ΣΔ ADCs and Nyquist-rate ADCs in terms of FOM_1 versus B_w.

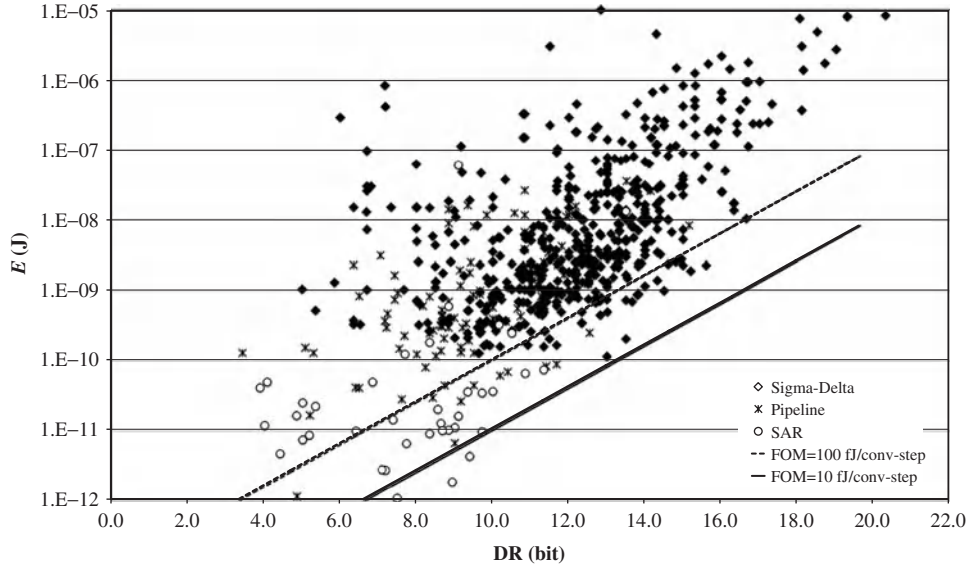


Figure 5.7 Comparison of energy plot of state-of-the-art ΣΔMs, SAR, and pipeline ADCs.

and even in the gigahertz range (Table 5.9). However, this parameter is not considered in FOM_1, which penalizes its performance compared to LP-ΣΔMs as will be discussed later.

Figure 5.6 compares the value of FOM_1 achieved by state-of-the-art ΣΔMs with that achieved by Nyquist-rate ADCs. Note that ΣΔMs show a competitive performance in a wide range of B_w, from hundreds of hertz to hundreds of megahertz. Only SAR ADCs achieve lower values of FOM_1 within the interval $10\,\text{kHz} < B_w < 10$ MHz. Pipeline ADCs are the best choice for those applications with $100\ \text{MHz} < B_w < 1$ GHz, whereas flash ADCs dominate for $B_w > 1$ GHz.

In terms of conversion energy, (as illustrated in Figure 5.7), ΣΔ ADCs are more efficient than SAR and pipeline ADCs for applications requiring DR > 12 bit, while SAR ADCs require the lowest conversion energy within the 8–12-bit resolution interval.

5.2 Empirical and Statistical Analysis of State-of-the-Art ΣΔMs

Analyzing in detail the data included in Tables 5.1–5.13, a number of interesting conclusions can be drawn. This section aims to identify practical pieces of information that can be used as guidelines for designers to make a better decision on the best architecture and circuit technique for a given set of specifications.

5.2.1 SC versus CT State-of-the-Art ΣΔMs

Although SC circuit technique has been traditionally used in most reported ΣΔMs, more and more ICs are being implemented using CT circuits, especially in those applications targeting broadband signals and/or requiring low power consumption.

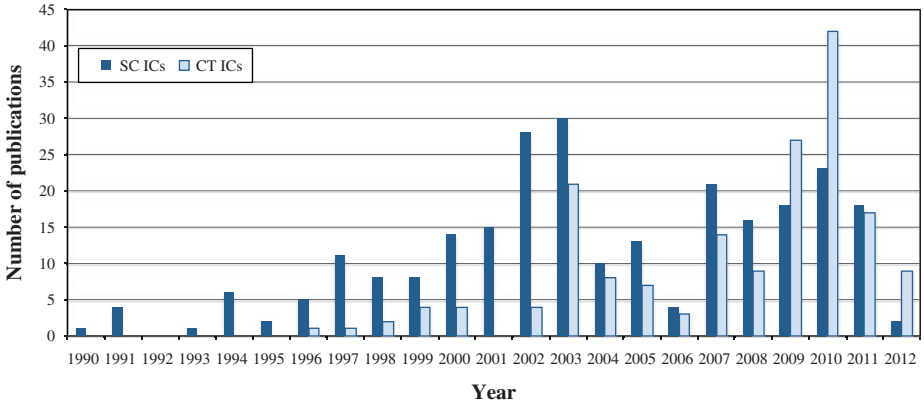


Figure 5.8 Number of SC- and CT-ΣΔMs published for the years 1990–2012.

Overall, approximately 57% of ΣΔM ICs included in this survey are implemented using SC circuits, whereas 43% of them are CT-ΣΔMs. However, as illustrated in Figure 5.8, there is a tendency, in recent years, to use CT techniques. This trend is expected to continue as this kind of ΣΔMs will benefit from the technology downscaling process and the increase in transition frequency. Indeed, the decrease in voltage supplies (and consequently the voltage headroom) together with the increasing use of digital signal processing and calibration techniques are favoring the implementation of CT-ΣΔMs in nanometer CMOS processes, increasing their operating frequency more and more [10].

Figure 5.9a represents the *aperture plot* [5] (i.e., B_w vs DR) corresponding to the state-of-the-art SC- and CT-ΣΔMs. Note that two *state-of-the-art fronts* (highlighted in the figure) can be clearly identified. One front—dominated by CT-ΣΔMs—goes from approximately 10-bit to 14-bit DR, covering a B_w range from 10 to 150 MHz. The other front—covered by SC-ΣΔMs—goes from 14-bit to 21-bit DR with B_w ranging from 400 Hz to 2 MHz. Therefore, as could be logically expected, it can be *empirically* deduced from these results that CT-ΣΔMs are more suitable for medium-resolution (10–14 bit) and medium–high bandwidths (10–100 MHz), whereas SC-ΣΔMs are more appropriate for applications requiring high-resolutions (15–20 bit) within low-medium bandwidths (100 Hz–1 MHz).

The resolution-versus-bandwidth regions covered by the state-of-the-art fronts drawn in Figure 5.9a define the application areas where SC- and CT-ΣΔMs are more efficient. This is illustrated in Figure 5.9b, where the energy plot is compared for both circuit techniques. Overall, it can be concluded that CT-ΣΔMs are more efficient than SC-ΣΔMs for $B_w > 1$MHz.

5.2.2 Gm-C versus Active-RC State-of-the-Art CT-ΣΔMs

As already discussed in this book, CT-ΣΔMs can be implemented using either active-RC or Gm-C integrators. Active-RC integrators have the advantages of better linearity

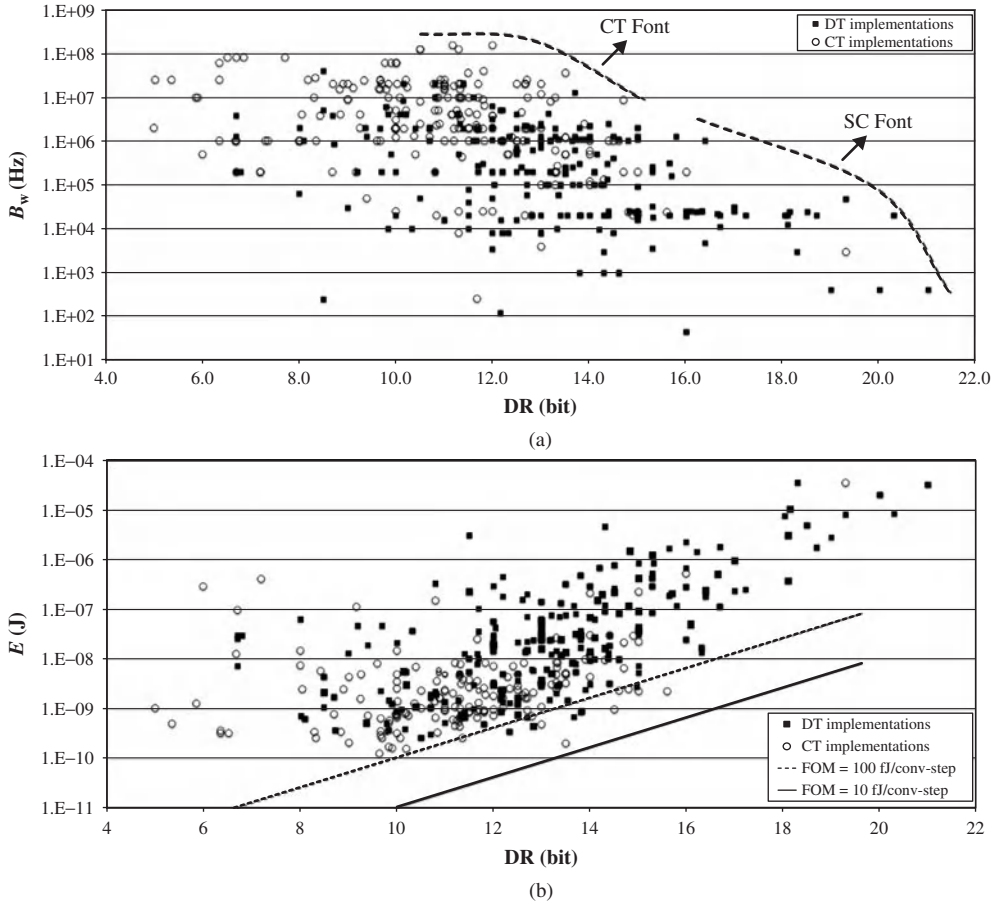


Figure 5.9 Comparison between state-of-the-art SC- and CT-ΣΔMs: (a) aperture plot (B_w versus DR) and (b) energy plot (E versus B_w).

and larger signal swing, whereas Gm-C integrators are potentially faster with less power consumption, at the price of reducing their linearity as compared to active-RC integrators [302]. In practice, the majority of state-of-the-art CT-ΣΔMs have been implemented using active-RC integrators. Sometimes an active-RC front-end integrator is chosen for its better linearity, whereas the rest integrators of the ΣΔM loop-filter are of Gm-C type. This is the most common situation in LP-ΣΔMs, while most BP-ΣΔMs operating at notch frequencies in the order of hundreds of megahertz or in the gigahertz band, have been implemented using either Gm-C or Gm-LC integrators.

Figure 5.10a illustrates the conversion region covered by the different CT circuit techniques by showing the aperture plot of state-of-the-art CT-ΣΔMs. For the sake of clarity, the different families of CT-ΣΔMs have been organized into three categories: Gm-C, active-RC, and active-RC/Gm-C. The latter assumes that only the front-end integrator is active-RC while the remaining ones in the modulator chain are implemented using Gm-C

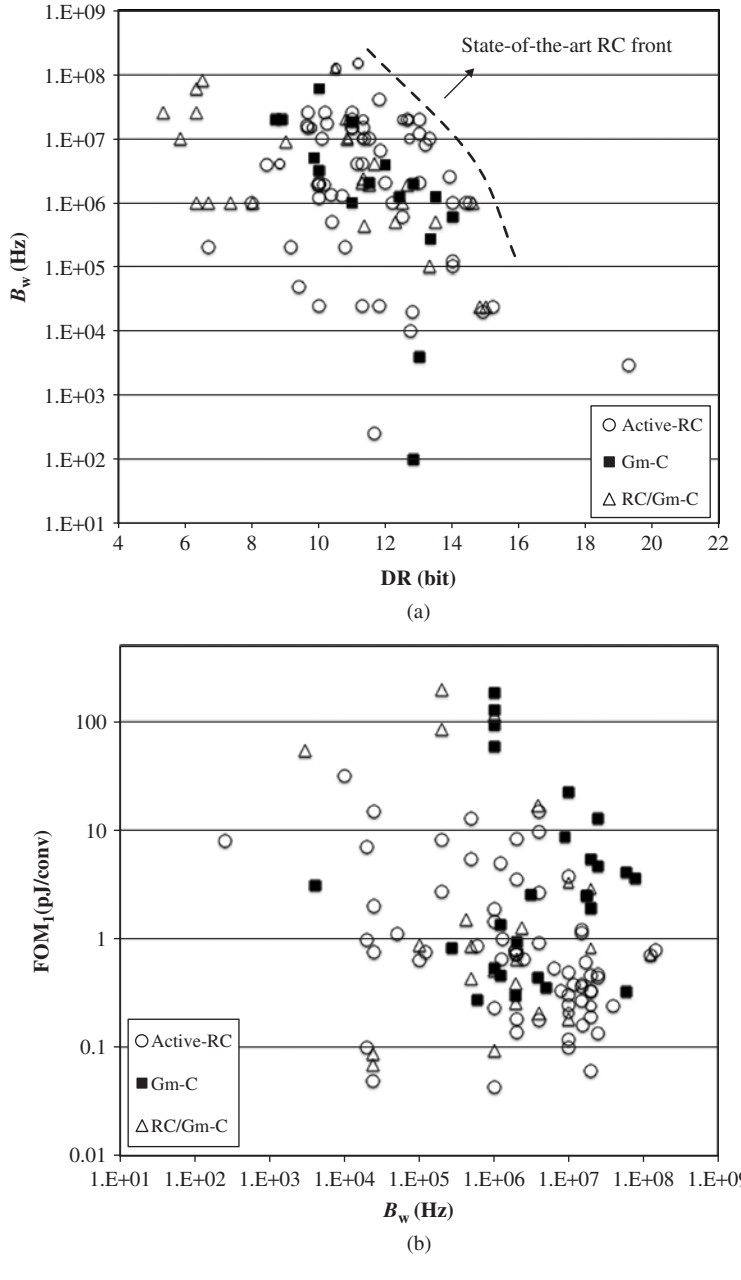


Figure 5.10 Gm-C versus active-RC CT-ΣΔMs: (a) aperture plot and (b) FOM_1 versus B_w.

techniques. It can be noted that the state-of-the-art front (highlighted in the figure) is dominated by active-RC implementations.

In terms of FOM_1, active-RC CT-ΣΔMs have demonstrated to be more efficient than Gm-C implementations, as illustrated in Figure 5.10b. It can be shown how active-RC ΣΔMs obtain lower values of FOM_1 in a wide range of B_w—from 100 kHz to almost 100 MHz.

5.2.3 *Technology Used in State-of-the-Art ΣΔMs*

A variety of CMOS processes can be distinguished in Tables 5.1–5.13. These technologies have been used in the last 20 years by state-of-the-art ΣΔMs—going from micrometer CMOS (3-μm technology node) down to nanometer processes (40 nm). Supply voltages have scaled down with technology evolution from >6.5 to 0.5 V. As an illustration, Figure 5.11 shows an histogram of the CMOS technologies employed by the ΣΔMs considered in this survey. Note that 180-nm CMOS process has been the technology node most commonly employed in recent years, mainly using a single 1.8-V supply voltage.

As discussed in Section 5.1.1, ΣΔMs have demonstrated state-of-the-art performance in many diverse processes, taking advantage of the technology downscaling to create the conditions to increase their conversion region. This is illustrated in Figure 5.12, where FOM_1 is represented over the technology node. Note that the minimum value of FOM_1 achieved by ΣΔMs decreases with the technology downscaling, reaching the lowest values for 180-nm CMOS processes.

5.2.4 *Single-Loop versus Cascade State-of-the-Art ΣΔMs*

Single-loop architectures have been more used than cascade topologies. Overall, approximately 80% of the state-of-the-art ΣΔM ICs considered in this survey are single-loop topologies, whereas only 20% of them are cascades. The latter have been mainly

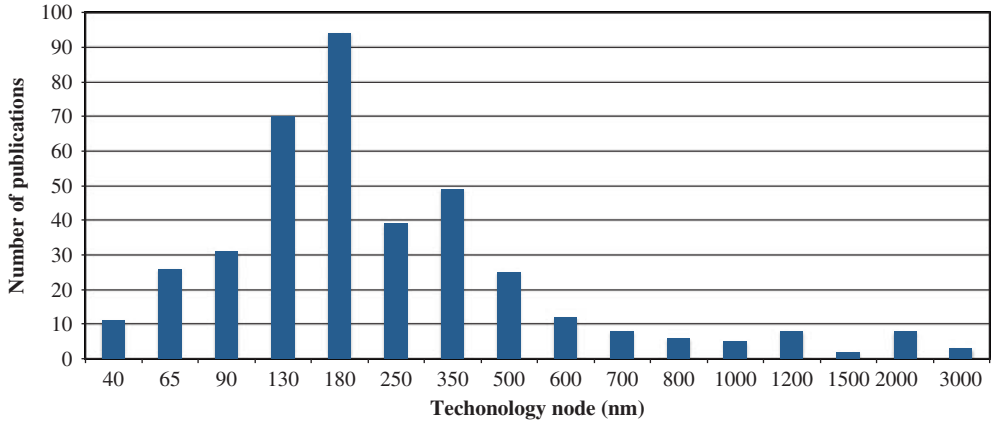


Figure 5.11 Histogram of CMOS technologies used by state-of-the-art ΣΔMs.

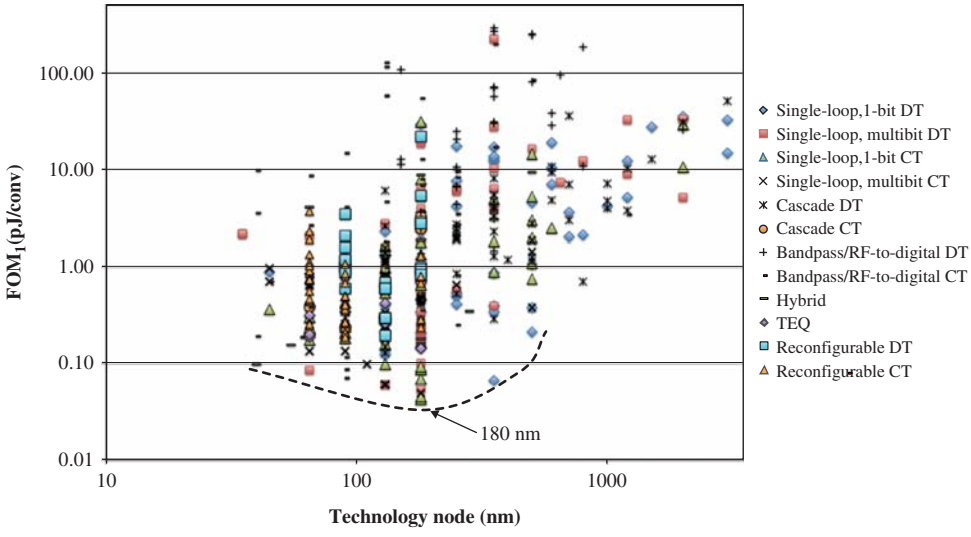


Figure 5.12 FOM_1 versus CMOS process nodes from 3 μm to 40 nm.

implemented using SC circuits, although recent synthesis methods are making the implementation of CT cascade ICs possible.

In terms of the loop-filter order of single-loop architectures, the second-order topology was the most used by SC-ΣΔMs, in 46% of single-bit architectures and 51% of multibit topologies (Tables 5.1 and 5.2). In the case of CT implementations, third- and fourth-order topologies were preferred by single-bit topologies, in 39 and 28% of cases, respectively (Table 5.6). In the case of multibit implementations, loop-filter orders higher than three were implemented in 83% of ICs, taking advantage of the better stability properties of high-order (> 3) implementations with multibit quantization.

As far as the cascade topologies are concerned (Tables 5.3, 5.4, and 5.8), the majority of implementations used a fourth-order loop-filter with two stages in a 2-2 configuration or three stages in a 2-1-1 configuration. Third-order 2-1 implementations are also used by SC cascades in 22% of cases, as shown in Tables 5.3 and 5.4.

Figure 5.13 compares the aperture plot of single-loop and cascade state-of-the-art ΣΔMs. It is noted how single-loop dominates almost all ranges of applications from low-bandwidth high-resolution to broadband low-medium resolution. However, as discussed in Section 5.3.1, enhanced techniques are improving the performance of cascade architectures, which may be very suited to implement reconfigurable/adaptive ADCs, thus taking advantage of the very modular nature of the cascade topology.

5.2.5 Single-Bit versus Multibit State-of-the-Art ΣΔMs

Multibit quantization is in general more employed than single-bit quantization. The majority of cascade topologies include dual quantization with a multibit quantizer in the last stage in order to attenuate the impact of the nonlinearity of the feedback DAC. However, increasingly more cascade ICs are including internal quantizers with B ranging from

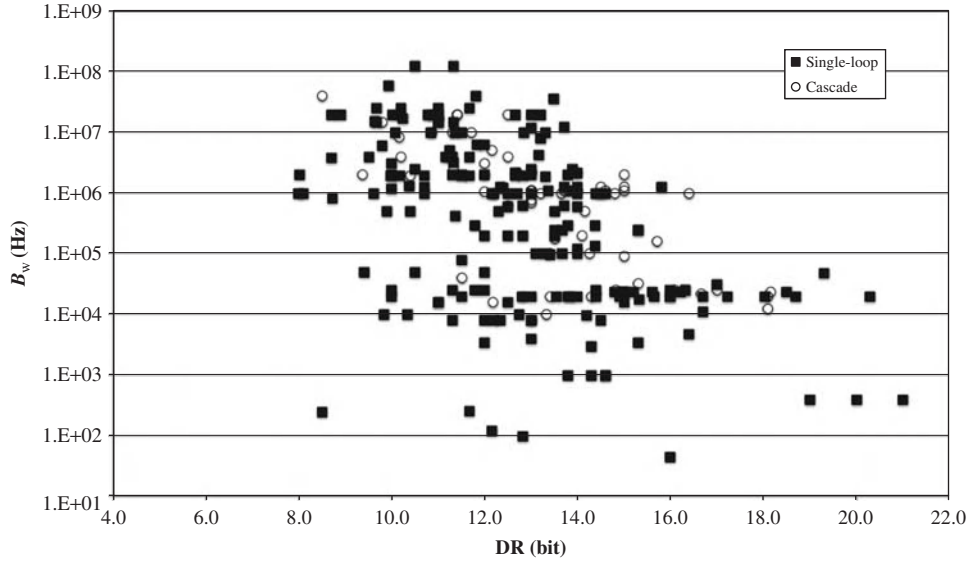
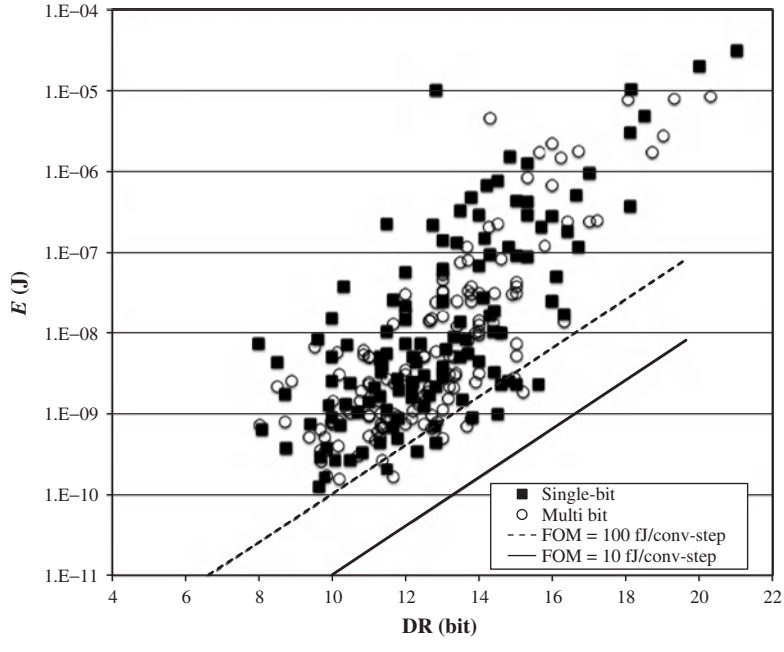


Figure 5.13 Aperture plot of single-loop and cascade state-of-the-art ΣΔMs. Band-pass, hybrid, and reconfigurable ΣΔMs are not included in this plot.

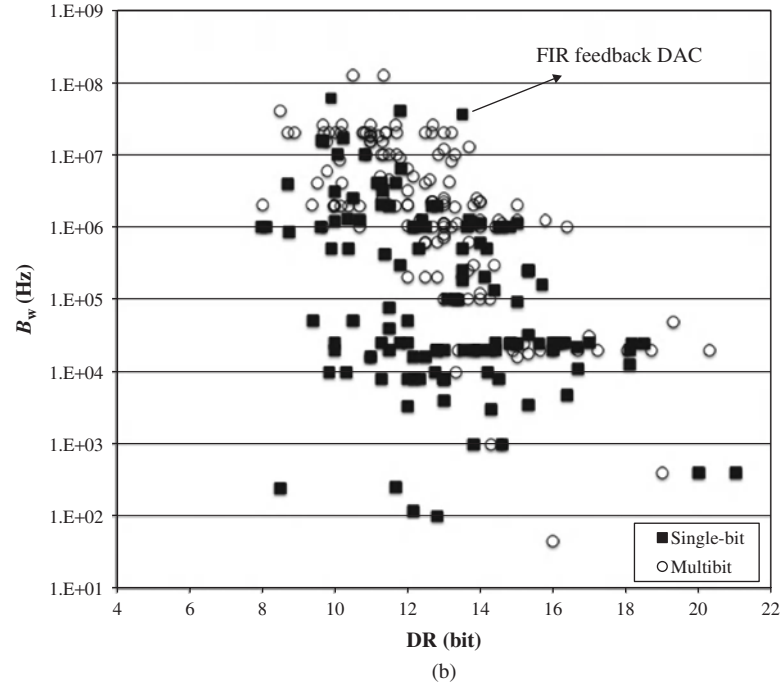
2 to 5 bit in all stages combined with proper linearization techniques. In addition, a number of ΣΔM ICs are including trilevel quantizers to benefit from the extra level provided by fully differential circuit implementation of the embedded flash ADCs, while keeping an inherent linear behavior of the feedback DAC.

In terms of conversion energy, there are more single-bit ΣΔMs than multibit ΣΔMs achieving a $FOM_1 < 100$ fJ/conversion-step, as illustrated in the energy plot shown in Figure 5.14a. Although multibit implementations can also achieve a good efficiency for resolutions in the order of 12–16 bit, there are more single-bit solutions reaching state-of-the-art performance within this range.

Figure 5.14b compares the aperture plot of single-bit and multibit ΣΔMs. It is shown how multibit modulators dominate the state-of-the-art fronts except for those applications with resolutions over 20 bit within signal bandwidths below 1 kHz. Indeed, a number of CT-ΣΔMs intended for broadband communications use multibit quantization. Apart from the obvious benefits in terms of the increased number of bits, multibit quantization (and NRZ DAC) is also appealing in practice for its lower sensitivity to clock jitter error, as discussed in Chapter 2. However, the price to pay for using multibit quantization is the inherent nonlinear operation of the multibit feedback DAC, what forces using linearization techniques, with the subsequent penalty in power consumption, speed limitation, and circuit complexity. To address this problem, some authors propose the use of alternative implementations of the modulator feedback waveforms—such as DACs with finite impulsive response (FIR) [154, 161]—to reduce the sensitivity to clock jitter and to relax the ΣΔM loop-filter linearity specifications, giving rise to a more competitive performance.



(a)



(b)

Figure 5.14 Single-bit versus multibit state-of-the-art ΣΔMs: (a) energy plot and (b) aperture plot.

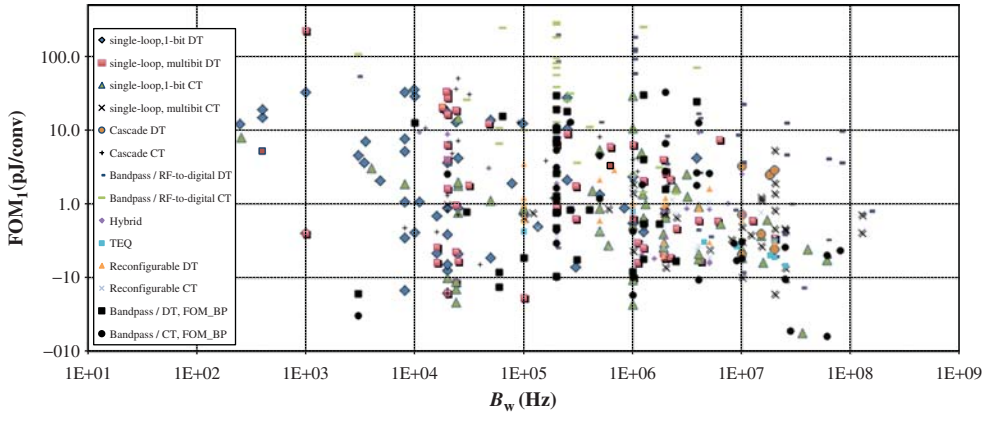


Figure 5.15 Comparison of LP- and BP-ΣΔMs: FOM_1 and FOM_{BP} versus B_w.

5.2.6 *Low-Pass versus Band-Pass State-of-the-Art ΣΔMs*

As stated in Section 5.1.2, LP-ΣΔMs obtain better performance in terms of FOM_1 than BP-ΣΔMs. This is better illustrated in Figure 5.15, which compares FOM_1 of LP- and BP-ΣΔMs. Figure 5.16a shows the aperture plot of LP- and BP-ΣΔMs. Note that the state-of-the-art front is dominated by LP-ΣΔMs, although the highest values of B_w have been reached by BP-ΣΔMs.

Comparing the energy plot of both types of ΣΔMs, it can be shown that LP-ΣΔMs consume less energy per conversion-step than BP-ΣΔMs, as illustrated in Figure 5.16b, where all ΣΔMs with $\text{FOM}_1 < 100$ fJ/conversion-step are of LP type. However, it should be noted that FOM_1 might not be adequate for quantifying the efficiency of BP-ΣΔMs, because B_w may not always be representative of the operating frequency of the modulator. For instance, looking at Table 5.9, it can be noted that there are several ICs digitizing signals with B_w in the order of 10–20 MHz but placed at center (notch) frequencies in the order of gigahertz. For that reason some authors propose alternative FOMs such as the following one [303]:

$$\text{FOM}_{\text{BP}}|_{\text{pJ/conv}} \equiv \frac{P_w(\text{W})}{2^{\text{ENOB(bit)}} \cdot (f_n + \frac{B_w}{2})} \cdot 10^{12} \tag{5.4}$$

which takes into account not only B_w, but also the notch frequency f_n. As illustrated in Figure 5.15, the use of FOM_{BP} increases the number of state-of-the-art performance BP-ΣΔMs, although the comparison may be not fair in this case for LP-ΣΔMs.

Figure 5.16c compares the performance of LP- and BP-ΣΔMs, based on the following FOM proposed by Schreier and Temes [2]:

$$\text{FOM}_3|_{\text{dB}} \equiv \text{SNDR(dB)} + 10\log_{10}[B_w(\text{Hz})/P_w(\text{W})] \tag{5.5}$$

To be consistent with the rest of the FOMs considered in this chapter, SNDR is replaced with DR in the above expression. Note that FOM_3 increases its value for better performing designs. The horizontal dashed line in Figure 5.16c was called the *architecture front*

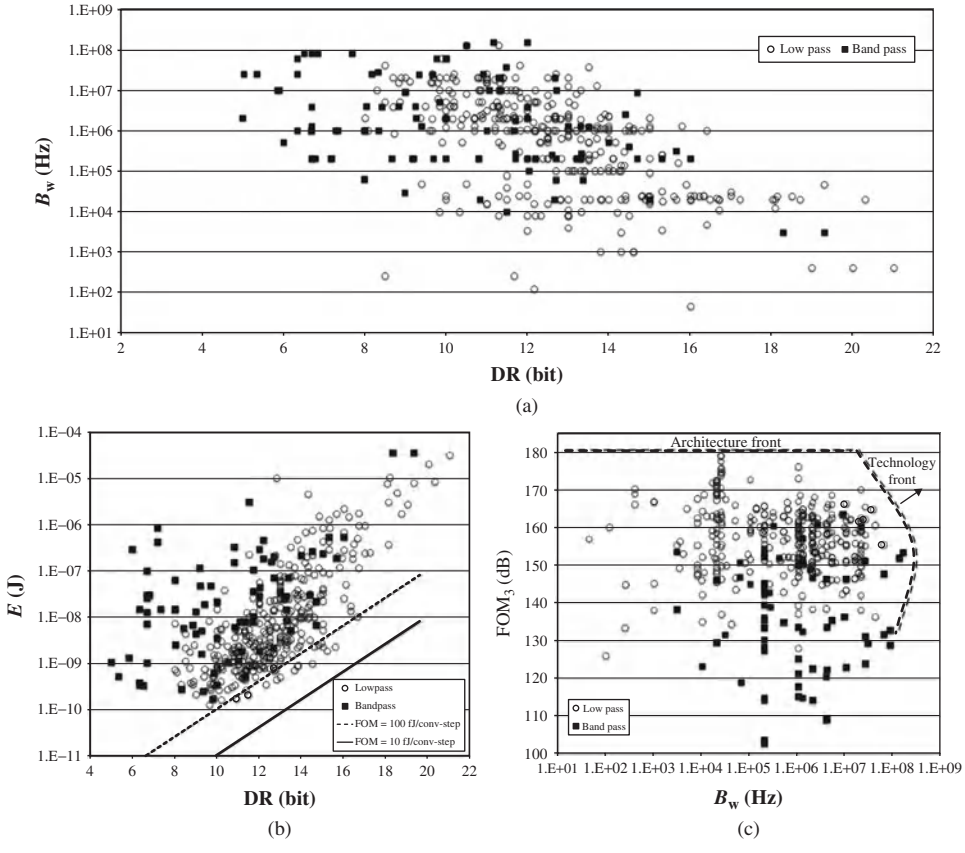


Figure 5.16 LP-ΣΔMs versus BP-ΣΔMs: (a) aperture plot, (b) energy plot, and (c) FOM_3 versus B_w.

by Schreier and Temes, while the vertical front is named *technology front*. The former is absolutely dominated by LP-ΣΔMs, while the most advanced architectures in the technology front are BP-ΣΔM topologies. The latter push the technology process to its limits in terms of digitized signal bandwidth and sampling frequency, at the price of being less efficient in terms of power dissipation. As discussed in the next section, the new generations of BP-ΣΔMs are taking advantage of technology downscaling to push ΣΔMs forward in terms of speed and efficiency.

5.3 Cutting-Edge ΣΔM Architectures and Techniques

To conclude this survey, this section presents a review of some of the most relevant cutting-edge ΣΔM topologies that are in the *frontiers* of the ΣΔM designs, identifying their most significant trends, challenges, and practical solutions. The following architecture/circuit techniques are overviewed:

- Sturdy MASH (SMASH) architectures

- Hybrid ΣΔM architectures, including ΣΔM-Nyquist ADCs, active–passive CT-ΣΔMs, CT/DT circuit implementations
- Multirate ΣΔMs
- Multibit ΣΔMs with time-coded quantization
- ΣΔMs implemented with digital-based circuit techniques
- Adaptive/reconfigurable ΣΔMs
- Ultra-high-speed ΣΔMs for RF digitization.

A detailed explanation of the aforementioned ΣΔMs is beyond the scope of this book. Instead, this section provides an overview of their most significant features, with emphasis on their advantages/drawbacks, design trends, and challenges.

5.3.1 SMASH ΣΔM Architectures

One of the main limitations of cascade ΣΔMs is their higher sensitivity to mismatch than their single-loop counterparts. This may explain why a vast majority of state-of-the-art ΣΔM ICs (~80%) were implemented using single-loop architectures as stated in Section 5.2.4. This limitation has motivated ΣΔ designers to look for alternative implementations of cascade ΣΔMs that reduce the sensitivity to noise leakages.

An interesting alternative cascade ΣΔ topology is the so-called *Sturdy* MASH (SMASH) [304, 305]. The main idea behind this kind of cascade ΣΔMs consists of replacing the DCL by the own analog processing provided by the loop-filter integrators together with an additional interstage digital feedback path. As a result, the modulator output can be obtained from a direct digital addition (or subtraction) operation of the different stage outputs in the cascade, with no need for a DCL as in conventional cascades, and the subsequent elimination of matching requirements between the analog and the digital filtering.

Figure 5.17 shows the conceptual block diagram of a two-stage SMASH ΣΔM.[3] Assuming a linear model for the embedded quantizers, the Z-transform of the modulator output can be written as [305],

$$Y(z) = \mathrm{STF}_1(z)X(z) + \mathrm{NTF}_1(z)[1 - \mathrm{STF}_2(z)]E_1(z) - \mathrm{NTF}_1(z)\mathrm{NTF}_2(z)E_2(z) \quad (5.6)$$

where $\mathrm{NTF_i}$ and $\mathrm{STF_i}$ represent the NTF and STF of the ith stage in the cascade, respectively. Note that the overall NTF of the cascade modulator—that is, the NTF filtering the second-stage quantization error $E_2(z)$—is the same as that of a conventional cascade ΣΔM. However, the first-stage quantization error is not completely canceled. Instead, it is filtered by $\mathrm{NTF}_1(z)[1 - \mathrm{STF}_2(z)]$.

Figure 5.18 illustrates how to implement the concept of SMASH ΣΔMs in a fourth-order 2-2 topology [305]. Note that the DCL is replaced by direct feedback paths from the output of the second stage to the input of the first stage, although an additional feedback DAC is required. Analyzing Figure 5.18 considering linear models for the quantizers, it is easy to show that the Z-transform of the modulator output is given by:

$$Y(z) = z^{-2}X(z) + (1 - z^{-1})^4 E_1(z) - (1 - z^{-1})^4 E_2(z) \quad (5.7)$$

[3] The concept of SMASH ΣΔMs has been also applied to multistage cascade architectures [306].

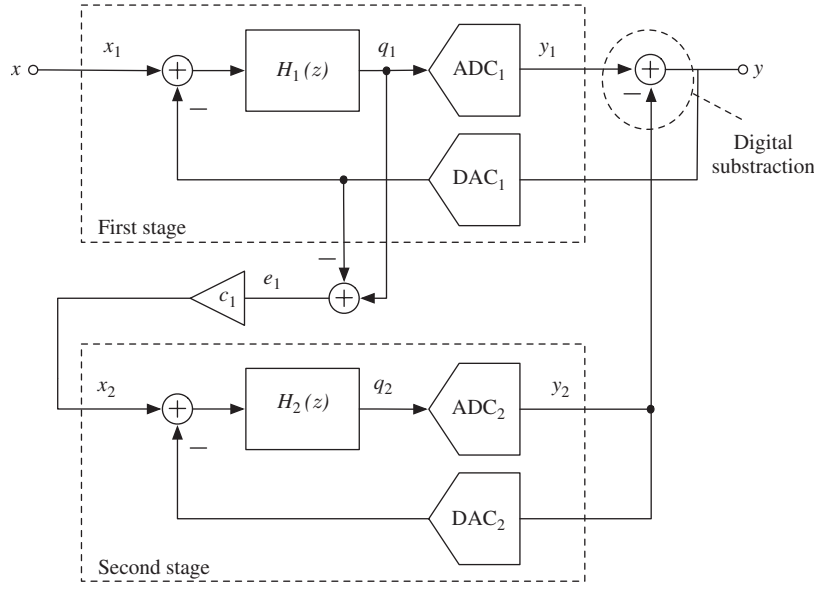


Figure 5.17 Conceptual block diagram of a two-stage SMASH ΣΔM [304, 305].

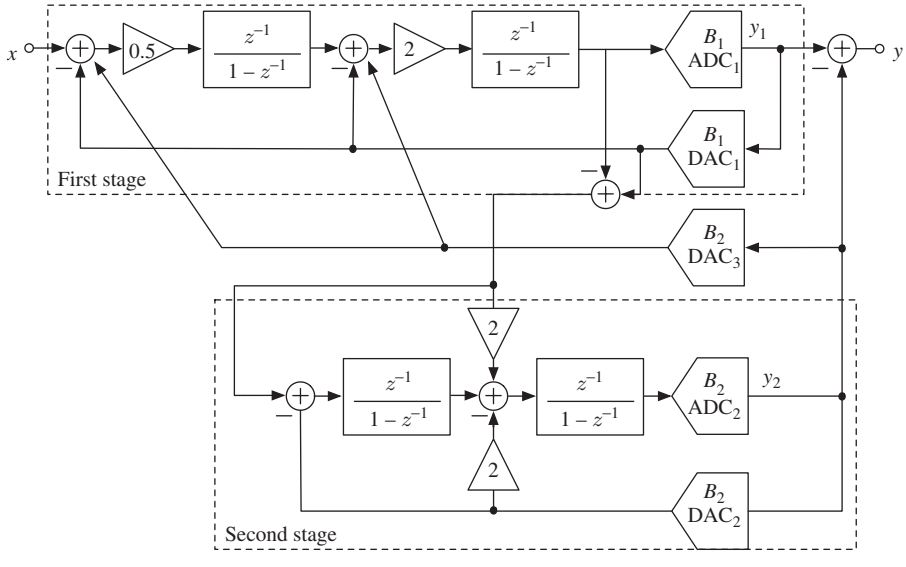


Figure 5.18 Block diagram of a 2-2 SMASH ΣΔM proposed in [305].

Note from the above expression that a fourth-order high-pass filtering is obtained for both $E_1(z)$ and $E_2(z)$. In this case, $\mathrm{STF}_2(z) = 2z^{-1} - z^{-2}$, which according to Equation 5.6, does not give a complete cancelation of $E_1(z)$. However, $E_1(z)$ can be ideally canceled if $\mathrm{STF}_2 = 1$, that is, if unity STF (USTF) [307, 308] is implemented in the second stage of the SMASH topology. Indeed, the performance of SMASH ΣΔMs can be notably improved if USTF is used in all stages of the modulator, as proposed in [288]. As stated in Chapter 1, the main advantage of using USTF is that integrators ideally process quantization error only and hence, their output swing can be reduced and the tolerance to amplifiers nonlinearities is increased. However, one of the main drawbacks of applying USTF to CT-ΣΔMs is that the resulting modulators do feature a worse implicit AAF function.

Figure 5.19 shows a ΣΔM architecture that extends the underlying principle of SMASH ΣΔMs to the implementation of USTF in both stages, thus taking advantage of both strategies (SMASH and USTF) to achieve higher resolution in wideband applications with lower values of OSR. In this example, the Z-transform of the modulator output is given by:

$$Y(z) = X(z) - \frac{1}{d}(1 - z^{-1})^4 E_2(z) \tag{5.8}$$

Note that contrary to conventional SMASH ΣΔMs, the quantization error in the front-end stage is ideally canceled, while avoiding any digital filtering. In addition, the use of a scaling factor d (which is usually implemented as a power of 2) will further reduce

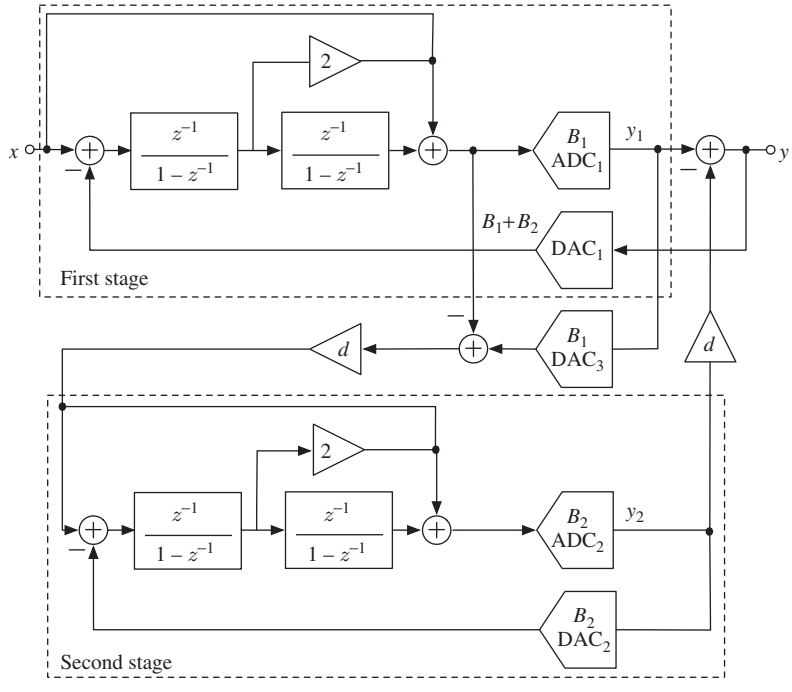


Figure 5.19 SMASH ΣΔM with USTF proposed in [288].

the in-band noise power of the second-stage quantization error at the output. Moreover, using USTF in both stages of the modulator allows relaxing of the output swing and gain demands in the amplifiers, while keeping a better sensitivity to mismatch than simple SMASH and cascade topologies. However, the price to pay is that a front-end DAC with a full scale larger than that of the quantizers in the cascade (with resolutions B_1 and B_2) is required to account for the summation of the digital outputs of the stages. Nevertheless, the location of the digital adder helps increase considerably the robustness to capacitor mismatches as demonstrated in [288].

5.3.2 *Hybrid ΣΔMs*

An interesting approach in the design of high-performance ΣΔMs consists of using *hybrid* circuit and/or architecture techniques to improve the efficiency of the digitization as compared to *conventional* ΣΔMs. The term *hybrid* is used here to denote those kinds of ΣΔMs that are not, strictly speaking, *pure* ΣΔMs, according to the classification criteria given in Chapter 1. Thus, several categories of hybrid ΣΔMs (H-ΣΔMs)—some of them listed in Table 5.11—have reported state-of-the-art performance.

Hybrid ΣΔM-Nyquist ADCs

A first category of H-ΣΔM architectures is that which results from combining a Nyquist-rate ADC—usually a pipeline, SAR, or cyclic ADC—and a ΣΔM [216, 228, 272, 273, 280, 286]. In the majority of cases, the basic strategy followed in this kind of H-ΣΔMs consists of replacing the embedded multibit flash ADC with another kind of Nyquist-rate ADC architecture, namely pipeline [280, 286], two-step [273], SAR [216], or cyclic [228, 272]. Essentially, the main advantage of this strategy is to provide a way of increasing the number of levels of the internal quantizer without the prohibitive exponential growth of the power consumption and silicon area of flash ADCs.

The idea of hybrid ΣΔM-Nyquist ADCs has been implemented in both single-loop and cascade architectures, featuring a competitive performance in different application scenarios. As an illustration, Figure 5.20 shows the conceptual block diagram of one of the first reported hybrid ΣΔM-Nyquist ICs, which consists of in a cascade of a 5-bit second-order ΣΔM and a 12-bit fourth-stage pipeline ADC [286].

One of the main inconveniences of combining ΣΔMs and Nyquist-rate ADCs is that as the number of bits of the Nyquist-rate subADC increases; the circuit complexity, sensitivity to circuit errors, and latency increases as well, with the risk of losing the benefits provided by ΣΔMs in terms of system simplicity and robustness against circuit nonidealities. Therefore, although very competitive performance can be achieved by replacing multibit flash ADCs with other alternative Nyquist-rate ADCs, the mentioned design trade-offs have motivated ΣΔM designers to explore alternative implementations to conventional multibit quantization, as discussed in Section 5.3.4.

Hybrid Active/Passive ΣΔMs

Another kind of H-ΣΔMs corresponds to a family of CT-ΣΔMs in which the loop filter is implemented by hybrid active–passive circuit elements. Thus, part of the loop-filter

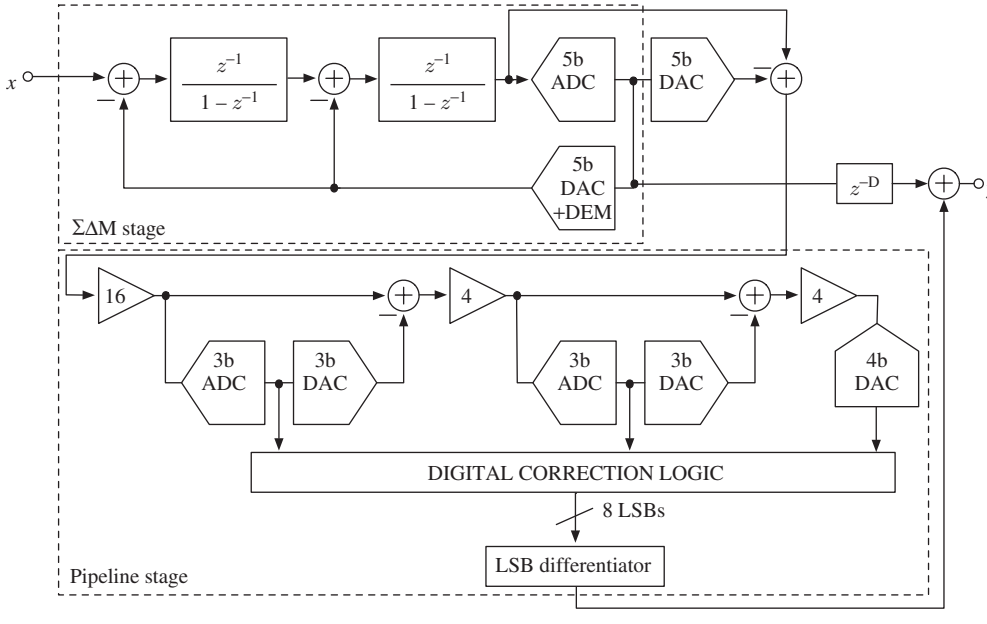


Figure 5.20 Example of the first reported hybrid ΣΔM-Pipeline ADCs [286].

integrators (resonators in the case of BP-ΣΔMs) are implemented as passive-RC networks, while others are implemented as active-RC or Gm-C circuits. The motivation for using this approach is to reduce the number of OTAs and hence reduce the power dissipation, as OTAs are the circuit blocks that consume most of the power in a ΣΔM. As already discussed in this book, the design issues associated with loop-filter OTAs aggravate in nanometer CMOS technologies with reduced supply voltage, and particularly in wideband applications where the dynamic requirements of the operational amplifiers yield in many cases to very power-hungry circuit solutions.

In this scenario, replacing some active integrators with passive-RC networks in the ΣΔM loop-filter results in a more efficient solution. Apart from the power saving, integrated passive-RC networks have better linearity than their active counterparts. However, the main limitation of using passive integrators is that they do not provide any gain that makes the loop filter more sensitive to their error mechanisms. This limitation can be partially palliated if the active integrators are used at the front-end of the modulator, so that the effect of thermal noise of the passive-RC networks and other circuit nonideal effects can be reduced by the preceding active integrators. Figure 5.21 shows an example proposed by Song *et al.* in [275]. The modulator architecture—consisting of a fifth-order single-loop architecture—uses five integrators, two of them implemented by passive-RC networks. These passive integrators are placed in the second and fourth positions of the modulator chain, so that their nonideal effects can be attenuated by the gain of active integrators.

Prompted by the above mentioned potential benefits, passive-RC networks are becoming a popular circuit solution for the implementation of low-power wideband CT-ΣΔMs.

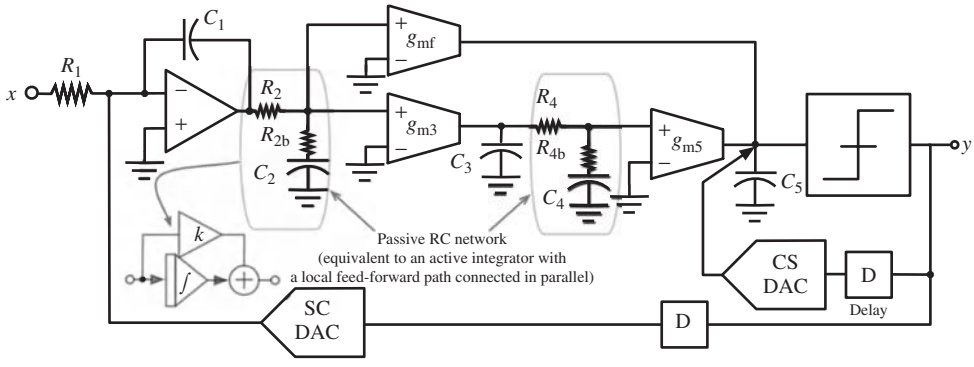


Figure 5.21 Example of the hybrid active–passive ΣΔM reported in [275].

A good example is the fourth-order BP-ΣΔM reported by Chae *et al.* [236], which digitizes 24-MHz band-pass signals located at 200 MHz with 58-dB SNDR while consuming 12 mW. This circuit is implemented by using single-opamp high-Q resonators that incorporate positive feedback to achieve high Q and are implemented by combining an active LPF and a passive HPF. Another interesting example was proposed by Srinivasan *et al.* [161], who reported a third-order CT-ΣΔM made up of a front-end passive-RC integrator followed by two Gm-C integrators. The chip—fabricated in a 45-nm CMOS technology—digitizes 60-MHz signals with 61-dB SNDR and 20 mW power consumption, while clocked at 6 GHz—one of the highest clock frequencies reported by ΣΔM ICs to date. This trend is expected to continue, with the demand for higher and higher data rates and the technology downscaling, which will facilitate the integration of high-quality passive devices together with conventional active circuits.

Hybrid CT/DT ΣΔMs

The third category of H-ΣΔMs included in Table 5.11 corresponds to those ΣΔMs in which the loop filter is implemented by both SC and CT integrators, thus taking advantage of both circuit techniques. As shown earlier in this chapter, CT-ΣΔMs operate at faster rates than their SC counterparts with relatively lower power dissipation. However, CT-ΣΔMs are more sensitive than SC-ΣΔMs to some circuit and architecture error mechanisms, such as circuit-element tolerances, excess loop delay, and clock jitter error.

Some authors have tried to circumvent the mentioned limitations of CT-ΣΔMs by using the so-called hybrid CT/DT ΣΔMs, in which some parts of the loop filter—usually the front-end building blocks—are implemented using CT circuits, thus providing potentially faster operation and embedded antialiasing filtering with reduced power consumption [279, 281–283, 294, 309, 310]. Thus, hybrid CT/DT ΣΔMs have been demonstrated using both single-loop [281, 283] and cascade ICs [279]. The most common situation in practice consists of using a CT front-end integrator, whereas the remaining integrators

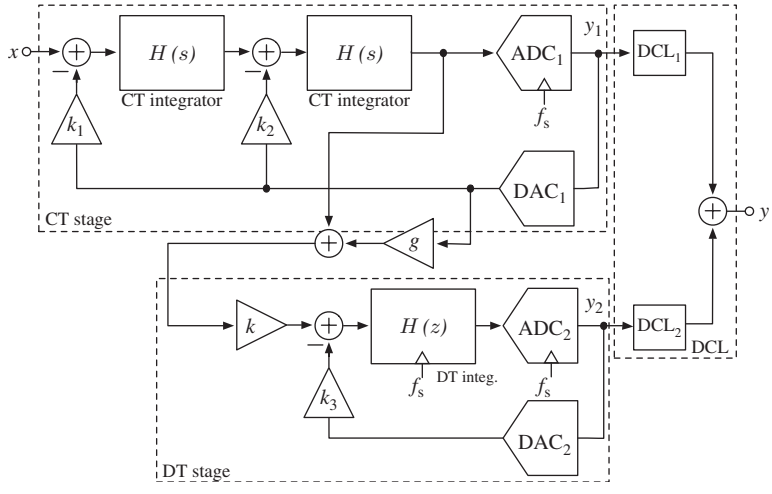


Figure 5.22 Example of the hybrid CT/DT cascade ΣΔMs reported in [279].

in the modulator loop are implemented using SC circuit techniques. As an illustration, Figure 5.22 shows one of the first practical implementations of a hybrid CT/DT cascade ΣΔM, proposed in [279]. This modulator is a cascade of a second-order (active-RC) CT-ΣΔM stage with a first-order SC stage, considering 4-bit quantization in both stages.

In spite of the mentioned advantages, reported hybrid CT/DT ΣΔM ICs do not really exploit the potential capability of CT circuits integrated in CMOS processes to operate up to the gigahertz range with reasonable linearity. This is partially due to the fact that the maximum sampling rate of hybrid CT/DT ΣΔMs is indeed limited by the SC part of the modulator loop filter. A possible solution to alleviate this restriction could be using the so-called multirating, as described in the next section.

5.3.3 Multirate ΣΔMs

The so-called multirate (MR) ΣΔMs are a particular kind of ΣΔMs in which a different sampling frequency is used at different parts of the modulator. The most common approach is based on using a lower sampling frequency in the building blocks placed at the front-end of the modulator—where most of the power is consumed—and a higher sampling rate in the subsequent subcircuits, where the dynamic requirements can be relaxed [311]. This concept can be applied to either single-loop or cascade architectures and implemented using SC [311] and CT circuits [312]. Without loss of generality, this section focuses on cascade implementations because the modular nature of these topologies makes the implementation of MR-ΣΔMs more simple and robust. However, similar considerations can be derived for single-loop MR-ΣΔM topologies.

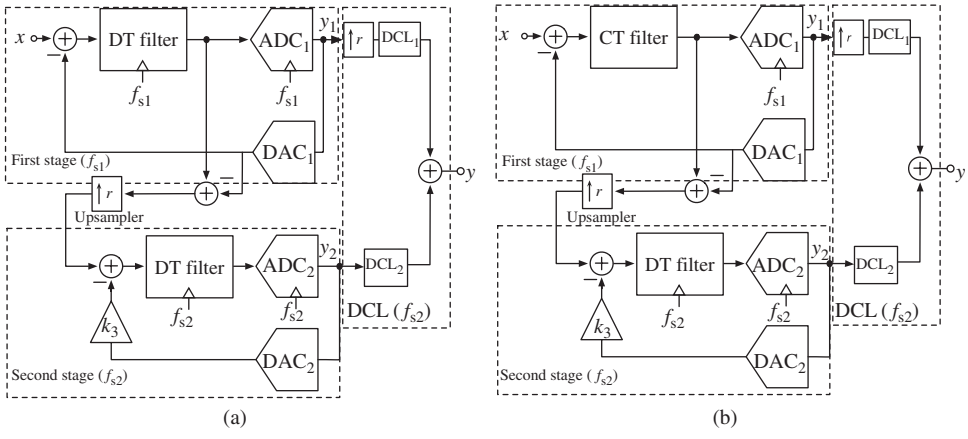


Figure 5.23 Conceptual block diagram of a conventional (upsampling) cascade MR-ΣΔM: (a) DT scheme and (b) hybrid CT/DT scheme.

Upsampling Cascade MR-ΣΔMs

Figure 5.23a shows the conceptual implementation of a conventional cascade (two-stage) MR-ΣΔM. [4] For the sake of generality, multibit quantization will be assumed in all stages of the cascade, with B_i being the number of bits of the internal quantizer in the ith stage. The sampling frequency f_{si} of the different modulator blocks is depicted in the figure. The most common situation in conventional MR-ΣΔMs is that the front-end stage operates at f_{s1}, whereas the remaining ith stages are sampled at $f_{si} > f_{s1}$. This approach—also referred to as upsampling MR-ΣΔM [313]—benefits from increasing values of OSR in the back-end stages—where the dynamic requirements are less demanding than in the front-end stages [311, 312, 314].

The operation behind the modulator in Figure 5.23a is conceptually the same as in a conventional monorate cascade ΣΔM. All stage outputs are combined by the DCL transfer functions—clocked at f_{s2}—so that ideally only the quantization error of the last stage remains and it is shaped by an NTF whose order (L) is the sum of the orders of all stages in the cascade (L_i). However, as a consequence of using several sampling frequencies, additional *upsampler* blocks—represented conceptually in Figure 5.23a—are required.

Assuming that $\mathrm{STF}_i(z) = z^{-L_i}$ and $\mathrm{NTF}_i(z) = (1 - z^{-1})^{L_i}$, and defining $\mathrm{DCL}_i(z)$ so that $\mathrm{DCL}_1(z) = \mathrm{STF}_2(z)$ and $\mathrm{DCL}_2(z) = \mathrm{NTF}_1(z^r)$, it can be shown that the IBN power at the output of the modulator is approximately given by [5]

$$\mathrm{IBN}_{\mathrm{USMR}} \simeq \frac{\Delta^2}{12} \frac{\pi^{2L}}{(2L+1)\mathrm{OSR}_1^{2L+1} r^{2(L-L_1)+1}} \tag{5.9}$$

where $r \equiv f_{s2}/f_{s1}$, Δ represents the quantization step of the last quantizer, and $\mathrm{OSR}_1 \equiv f_{s1}/(2B_w)$ [313].

[4] For the sake of simplicity, two-stage cascade architectures are discussed in this section without loss of generality. This analysis can be extended to N-stage cascade MR-ΣΔMs as shown in [313].

[5] Subscript USMR is used for noting UpSampling MultiRate in order to differentiate from the downsampling multirate (DSMR) concept discussed later in this section.

Hybrid CT/DT Cascade MR-ΣΔMs

The concept of MR-ΣΔMs can be extended to hybrid CT/DT implementations as conceptually depicted in Figure 5.23b, that represents a cascade two-stage MR H-ΣΔM. The circuit nature (either CT or DT) of the different modulator blocks as well as their corresponding sampling frequencies are highlighted. The use of a complete CT stage (instead of just the front-end integrator for instance) maximizes the embedded AAF, while taking advantage of the relaxed dynamic requirements of the front-end CT circuits as compared to their DT counterparts [279].

The analysis of Figure 5.23b can be carried out by applying a DT–CT transformation to the front-end stage of Figure 5.23b. The resulting MR H-ΣΔM is equivalent to the original MR DT-ΣΔM. As stated in Chapter 1, this CT–DT equivalence can be guaranteed because of the DT nature of the (open) loop transfer function from the front-end quantizer output to the sampled quantizer input [315, 316]. This way, considering this CT–DT equivalence and assuming a linear model for the quantizers, it can be shown that the IBN of the modulator in Figure 5.23b is also given by Equation 5.9.

Downsampling Hybrid CT/DT Cascade MR-ΣΔMs

Using lower values of the OSR in the front-end CT stage of Figure 5.23b may be beneficial when the subsequent SC stages can operate at high enough values of OSR. However, the use of high values of OSR in the back-end stages is not practical when digitizing wideband signals, for instance in the order of 10–100 MHz. In these cases, sampling rates in the order of gigahertz may be necessary to digitize such signals with low-medium effective resolutions (8–10 bit). Therefore, the use of back-end SC stages may be not feasible because of the prohibitive values of GB required for the operational amplifiers.

The mentioned problems can be alleviated if the concept of multirating is redefined just in the opposite way as it was conceived, that is, going from an *upsampling* MR system—in which the front-end stage rate is increased in the subsequent stages—to a *downsampling* MR system—in which the front-end (CT) stage operates at the highest rate, thus benefiting from their potentially faster operation [313].

Figure 5.24 shows a conceptual block diagram of a *downsampling* (two-stage) cascade MR H-ΣΔM architecture proposed in [313]. In contrast to conventional (*upsampling*) MR H-ΣΔMs, the back-end (DT) stage operates at a rate lower than that of the front-end (CT) stage, that is, $f_{s2} < f_{s1}$. The main drawback of this approach is the aliasing caused by the downsampling processing, which requires using an interstage AAF. However, as shown in [313], the operation of the AAF can be completely translated to digital domain by using two additional digital blocks of the transfer functions which are named $H_1(z)$ and $H_2(z)$. In this way, Figure 5.24 has the same number of analog building blocks as a conventional cascade ΣΔM.

Therefore, the operation behind the modulator in Figure 5.24 is essentially the same as in conventional cascade ΣΔMs. The main difference is that the DCL transfer functions are designed so that they must remove not only the quantization error of the front-end stage $E_1(z)$, but also its aliased components. To this purpose, $H_1(z)$ and $H_2(z)$ must be reconfigurable and programmable according to the value of the downsampling ratio ($p \equiv f_{s1}/f_{s2} > 1$). These functions are completely implemented in the digital domain, without any extra analog hardware required, and can be easily synthesized for different values of

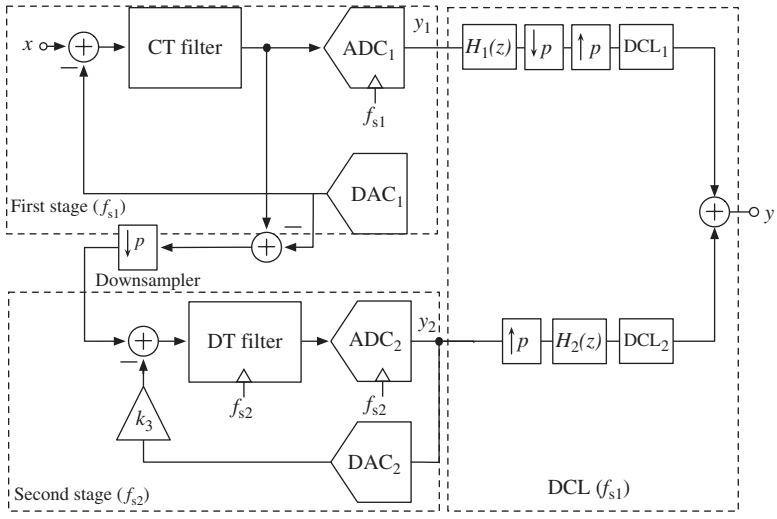


Figure 5.24 Conceptual block diagram of a downsampling cascade hybrid CT/DT MR-ΣΔM [313].

p as detailed in [313]. As a consequence, the resulting MR-ΣΔMs are potentially faster, less sensitive to circuit error mechanisms, and more power efficient than conventional upsampling MR architectures.

Compared to single-rate cascade *pure* CT-ΣΔMs, downsampling MR H-ΣΔMs may be less power efficient if the same specifications are required. However, under such conditions, downsampling MR H-ΣΔMs becomes less demanding in terms of some circuit nonidealities such as finite DC gain and circuit-element tolerances, as demonstrated in [313].

5.3.4 Multibit ΣΔMs with Time-Coded Quantization

In addition to increasing the bandwidth of digitized signals, significant efforts have been carried out in recent years to improve the performance of the embedded multibit quantizers used in most state-of-the-art ΣΔMs. As already discussed in different parts of this book, multibit quantizers are usually implemented by flash ADCs, which are generically made up of a resistive ladder and a bank of comparators. As already mentioned, the number of comparators increases exponentially with the number of bits B, with the subsequent penalty in silicon area and power dissipation.

An alternative technique to reduce the number of comparators is based on the so-called *tracking quantizer* [199] that uses a counter and a switching matrix to connect the reference voltage to the comparators inputs according to the value of the quantizer input ranges. However, although the power consumption is reduced, the efficient implementation of conventional multibit quantizers based on flash architectures is becoming a design challenge because of the reduction in voltage supplies associated with CMOS technology

downscaling. As a consequence, the dynamic range may become severely degraded by the effect of comparator offset and hysteresis [317].

The mentioned problems have motivated the exploration of alternative circuit implementations of multibit quantizers embedded in ΣΔMs. The majority of these techniques are based on changing the concept of signal quantization, going from an amplitude coding to a time coding. The main benefit of this approach is that the resulting quantization methods are more suited for the implementation in low-voltage (<1 V) nanometer CMOS processes [318], although special care must be taken in terms of those errors that are directly related to the timing, such as clock jitter error in CT-ΣΔMs.

Figure 5.25 shows the conceptual block diagram of the three basic alternative approaches for the implementation of *time-coded* quantizers which have been successfully reported in state-of-the-art ΣΔM ICs, some of them listed in Table 5.10. The first one—depicted in Figure 5.25a—consists of using a voltage-controlled oscillator (VCO) instead of a multibit quantizer [259–262, 265]. The idea is based on translating the quantized information in the amplitude domain to the time domain by means of a voltage-to-frequency conversion that can be performed by a VCO circuit as proposed in [265]. The principle of operation behind this approach is based on the use of a ring oscillator to count the number of edges within a given time period. The result is directly related to the input signal, thus obtaining a digital representation of the input amplitude.

The main limitation of VCO-based quantization is the inherent nonlinear operation associated with the voltage-to-frequency conversion implemented in the VCO. This limiting factor can be mitigated by using a phase detector instead of a frequency detector as proposed in [259], which provides also a first-order noise shaping. Alternatively, VCO-based quantizers can be embedded in the back-end stage of a cascade ΣΔM, such that their nonlinearities are attenuated by the noise shaping of precedent stages in the cascade. An interesting implementation of this idea, proposed by Asl *et al*. [261], consists of a two-stage cascade upsampling MR ΣΔM, in which the back-end stage (operating at a higher rate) is implemented by a VCO-based quantizer, thus benefiting from its potentially higher operation speed without being penalized by its intrinsic nonlinearity.

An alternative to VCO-based quantizers consists of using pulse-width modulation (PWM) [209], as conceptually depicted in Figure 5.25b. The PWM block converts the voltage at the output of the loop filter into a pulse $p(t)$ with corresponding width and a time-to-digital converter (TDC) generates digital codes that are discrete representations of the edges of the $p(t)$ signal, that is, a time-quantized representation. One of the main limitations of PWM-based quantizers is their limited dynamic range [317]. It can be enhanced by using the so-called *time-enconding quantizer* (TEQ) as illustrated in Figure 5.25c [171]. Essentially, the idea underlying TEQ-based ΣΔMs consists of using a modulator (a CT-ΣΔM in the figure) whose loop filter is split into two loops, one of which is made to oscillate, thus providing an independent control of the oscillation as a difference with PWM-based quantization schemes [187].

5.3.5 *Mostly Digital ΣΔMs*

Time-to-digital conversion techniques have not only been used for replacing embedded flash ADCs, but also for the main building blocks of a ΣΔM. A good example of using voltage-controlled ring oscillators as an essential part of ΣΔMs was proposed by Taylor

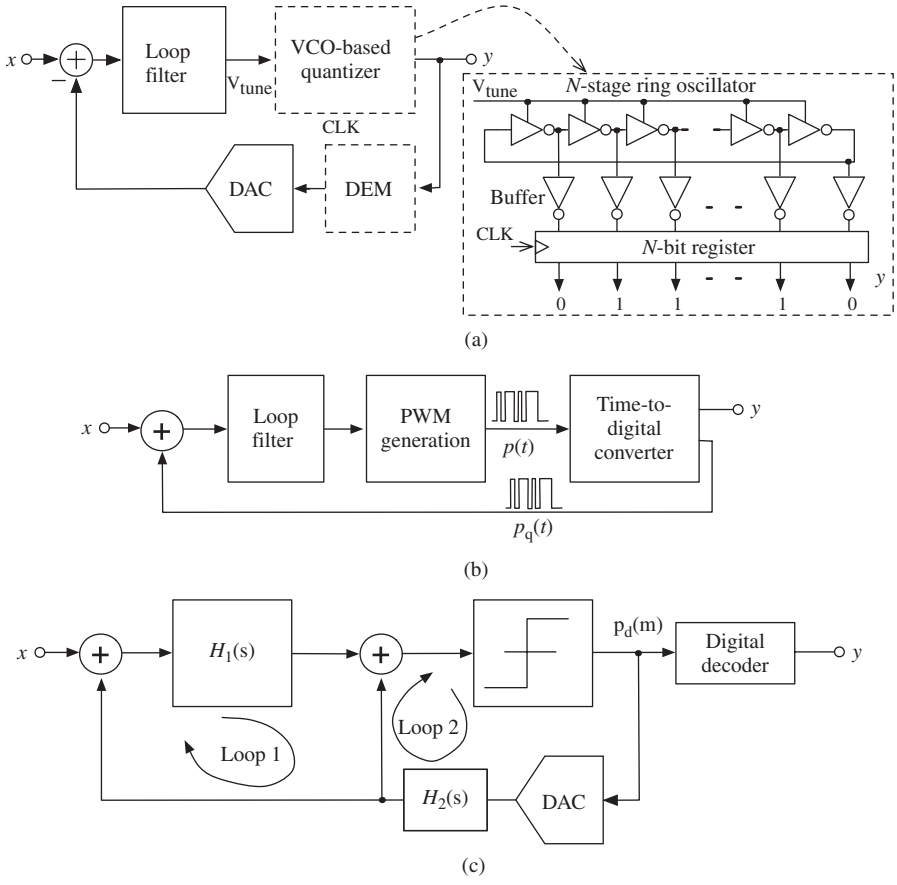


Figure 5.25 Conceptual block diagrams of some ΣΔMs with *time-coded* quantization: (a) VCO-based scheme proposed in [265]. The implicit barrel-shift DEM is not shown for the sake of simplicity; (b) PWM-based quantization reported in [209]; and (c) TEQ-based ΣΔM architecture proposed in [171].

and Galton in [264]. The proposed modulator—mostly implemented by digital circuit techniques—consists of a digitally background-calibrated voltage-controlled ring oscillator (VCRO), in which its inverters are sampled at the required output sample-rate without containing any analog integrators, comparators, or feedback DACs. The main advantage of this approach is that as the ΣΔM is essentially implemented by digital circuits, it may benefit from the CMOS technology downscaling in terms of speed, with the subsequent advantages for low-voltage low-power wideband applications. Moreover, the operation of VCRO-based ΣΔMs essentially depends on the speed of its digital blocks, thus trading accuracy for signal processing. The main limitation of this approach is its higher sensitivity to timing errors, such as clock jitter error and the aforementioned nonlinear behavior of VCOs. These problems are overcome in [264] by using digital background calibration and self-canceling dither techniques, yielding to state-of-the-art results.

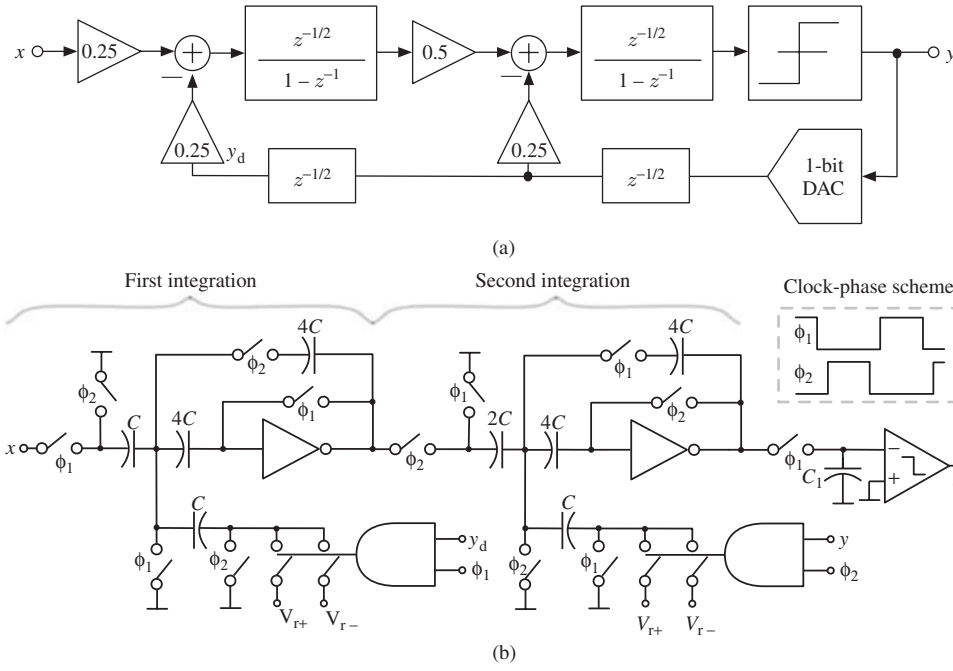


Figure 5.26 Example of the inverter-based second-order single-loop ΣΔM proposed in [12]: (a) block diagram and (b) conceptual single-ended schematic.

Apart from the mentioned *time-coded* circuit techniques, there have been some other approaches to *digitize* ΣΔM loop filters: by replacing their critical analog building block—essentially the OTAs—with alternative *digital-based* solutions. This is the case of the comparator-based [36] and inverter-based [12, 276] integrators used in ΣΔMs, in which their embedded OTAs are replaced by comparators and inverters, respectively. As an illustration, Figure 5.26 shows the block diagram of the inverter-based second-order single-loop ΣΔM proposed by Chae and Han in [12]. The modulator—implemented using very simple circuits—achieves 63-dB peak-SNDR within 8-kHz bandwidth, while consuming only 5.6 μW from a 1.2-V supply. These results demonstrate that the use of *digital-based* circuit techniques may become very energy efficient while keeping a cutting-edge performance. Indeed the use of inverters to emulate the transconductance operation, such as the well-known Nauta's transconductor [319], as well as improved versions of this idea, are gaining popularity in the analog design community, for instance in the design of widely tunable filters [320].

However, despite the aforementioned benefits of replacing OTAs with digital-based circuit solutions, it should be noted that simple circuits such as inverters, cannot be seen as the *panacea* for all analog design problems in ΣΔMs. In practice, there is always a trade-off between the required electrical performance of a given building block and the circuit complexity in terms of number of transistors, with the subsequent penalty in power consumption, silicon area, impact of parasitics, etc. Therefore, a simple inverter circuit cannot replace a high-performance OTA. However, depending on the modulator

specifications and target application, digital-based integrators could be a suited circuit solution. For instance, this might be the case in some biomedical applications demanding very low power consumption and/or wideband applications requiring ultra-high-speed loop-filter circuits. In both cases, the required DR is typically below 60–70 dB, which might not be very demanding in terms of finite DC gain of loop-filter building blocks and, consequently, the use of simple digital-based circuit solutions may result competitive. Nevertheless, the use of digital techniques is still at an early stage and it is expected that circuit improvements derived from technology downscaling will make possible that ΣΔMs become more and more digital in a near future.

5.3.6 Adaptive/Reconfigurable ΣΔMs

The integration of entire wireless telecom systems into mainstream low-cost *digitally oriented* nanometer CMOS technologies together with the continuous increasing number of standards, operation modes, and applications, is driving the need of the so-called *adaptive/reconfigurable* ADCs, which are a particular kind of ADC that can reconfigure its performance metrics to different sets of specifications with adaptive power consumption. However, the practical implementation of this kind of ADCs involves a number of practical issues and trade-offs at both architecture and circuit level, which must be taken into account to optimize the performance in terms of power budget. This problem is indeed aggravated as efficiency is required in an increasingly wide tunable region in the resolution-versus-bandwidth plane.

In this application scenario, ΣΔMs are very suited for the implementation of this kind of ADC as the variation of their basic parameters—OSR, L, and B—can easily contribute to adapting the ADC performance to different requirements with optimized power consumption and large hardware reuse [321].

Reconfiguration at Architecture Level

The aforementioned reasons explain why state-of-the-art multistandard, multimode ADCs have been mainly implemented using ΣΔMs. Indeed, a number of adaptive/reconfigurable ΣΔM ICs have been reported in the literature, implemented with both SC and CT circuit techniques, as summarized in Tables 5.12 and 5.13, respectively. The majority of the ΣΔMs included in these tables are able to digitize more than three standards [85, 287–295, 297], achieving cutting-edge performance while digitizing signals with B_w ranging from hundreds of kilohertz to tens of megahertz. Although changing the sampling frequency has been the reconfiguration strategy most commonly applied, the mandatory use of low values for OSR in wideband standards has forced ΣΔMs to achieve the required DR by resorting to different strategies, including adaptive high-order shaping [287, 292], switchable cascade topologies [290, 291, 297], multirating [290], programmable notches within the signal band [287, 300], and multibit embedded quantizers with adjustable number of bits [85].

Adaptation at Circuit Level

The mentioned architecture reconfiguration strategies are frequently combined with circuit/transistor-level techniques in order to increase the energy efficiency of

adaptive/reconfigurable ΣΔMs. Among others, one of the most commonly used circuit techniques consists of implementing switchable unit cells, such as OTAs [295, 300], or unit capacitors [237, 298], in which a different number of unit elements is connected depending on the required electrical performance of a given ΣΔM building block. Apart from the former circuit-level reconfiguration techniques, programmable bias current generators are frequently used in order to adapt the bias current and, subsequently, the power consumption of the amplifiers (and the preamplifiers used for the comparators) to the different operation modes covered by the reconfigurable ΣΔM [291, 298].

Concurrency

Although both single-loop and cascade topologies report state-of-the-art performance, cascade ΣΔMs are a priori more suitable architectures for the implementation of adaptive/reconfigurable ADCs. Indeed, the modular nature of cascade topologies allows to easily turn different stages either on or off depending on the loop-filter order L required for a given set of specifications, while keeping system stability and low sensitivity to circuit nonidealities. In addition to reconfiguring its performance parameters, a multimode ΣΔ ADC must digitize signals corresponding to different standards in a parallel or concurrent way—that is, GSM and Bluetooth signals, and/or UMTS and WLAN signals.

Concurrency can be easily implemented in a cascade ΣΔM as conceptually illustrated in Figure 5.27, where a concurrent switchable network is used for allowing the modulator to be configured as several *submodulators* working in parallel, each one processing a different input signal. A number of control signals are needed to enable the switches that correspond to the desired operation mode in each case, either concurrent or cascaded. Moreover, a programmable DCL is required, which should be adapted to the concurrent operation and/or the number of stages working as either isolated single-loop modulators or as parts of a cascade. The price to pay is an increase in the complexity of the digital control circuits as compared to conventional ΣΔMs. However, the implementation of digital circuits benefits from technology downscaling, which may result in a more efficient solution than using separate ADCs operating in parallel. Thus, the majority of state-of-the-art reconfigurable ΣΔMs focus their efficiency in achieving reconfigurability and adaptability. The increasing need of implementing concurrency in modern mobile handheld terminals has motivated doing more research on this topic and the first experimental evidences of implementing this feature have already been reported [322].

5.3.7 Ultra-High-Speed CT-ΣΔMs for RF Digitization

As previously explained in this chapter, more and more CT-ΣΔMs are demonstrating to be a competitive solution for the implementation of power-efficient ADCs operating in the gigahertz range [154, 161, 181, 207, 232, 239, 251, 323]. This feature has prompted the interest of RF circuit designers to use CT-ΣΔMs in *digital-intensive* RF transceivers. Thus, in addition to digitize signals, CT-ΣΔM ICs might take advantage of their CT circuit nature to merge some RF functionalities such as blocker-rejection filtering, frequency-mixing process, channel-selection and antialiasing filtering, and so on [155, 176, 203, 233, 284]. All these functionalities can be embedded within the ΣΔM feedback loop as conceptually depicted in Figure 5.28a, thus resulting in more compact RF receivers, with

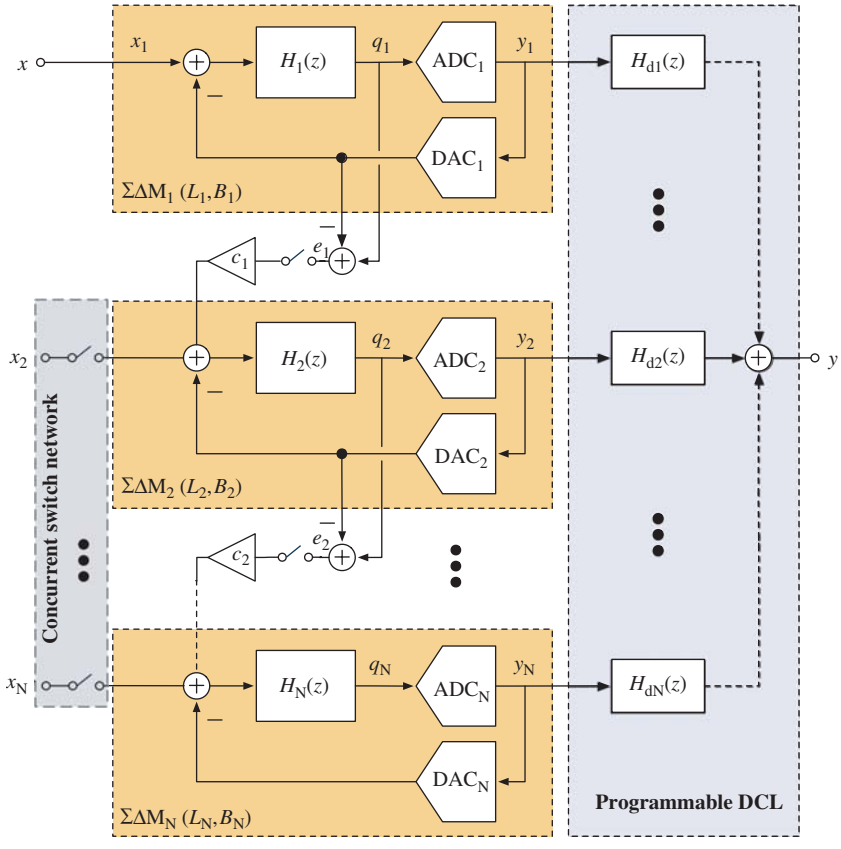


Figure 5.27 Conceptual block diagram of a reconfigurable cascade ΣΔM with concurrent operation.

a reduced complexity in the analog hardware and the subsequent benefits in terms of sensitivity to circuit imperfections and power/area scalability with technology.

A fully digitized RF receiver, conceptually illustrated in Figure 5.28b, should place the analog-to-digital interface as close as possible to the antenna—an approach commonly referred to as *RF-to-baseband digitization* or *RF-to-digital conversion* [251]. Although this is the ideal implementation of the so-called *software-defined radio* (SDR) paradigm [324], its practical application scenario is still far from a consumer product deployment. This is mostly limited by the unfeasible *power-hungry* requirements demanded by the ADC, particularly in terms of signal bandwidth, linearity, and dynamic range, supporting signal levels from the sensitivity to the full strength. In order to reduce the power consumption, some authors propose using BP-ΣΔMs to digitize IF signals [232, 236], by implementing an IF-digitization scheme as that shown in Figure 5.28a. This way, after being preamplified by the LNA and downconverted from RF to IF in the mixer, the incoming signal is digitized by a CT BP-ΣΔM. A good example of this compact and power-efficient solution is reported by Harrison *et al.* in [232]. The modulator is integrated in a 40-nm CMOS

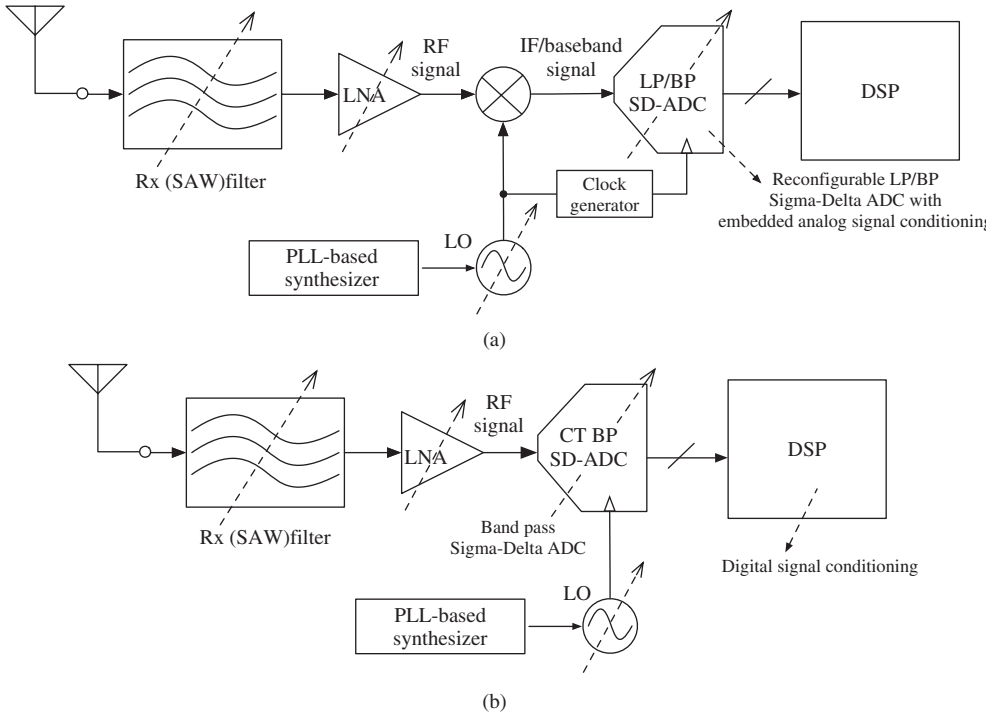


Figure 5.28 Conceptual block diagrams of *digital-intensive* RF receivers using: (a) reconfigurable LP/BP-ΣΔM with embedded analog baseband signal conditioning. The ΣΔM can be implemented using either SC or CT circuits depending on the operating frequency. (b) RF-to-baseband CT BP-ΣΔM, where RF signals are directly digitized and most signal processing is implemented by the DSP.

technology and digitizes 20-MHz signals placed at 700 – 800 MHz with 70-dB SNDR, dissipating 20 mW when clocked at 3.2 GHz.

In spite of the strong specifications required by RF-to-digital converters, the recent advances in BP-ΣΔM ICs—particularly some ICs implemented using CT circuits listed in Table 5.9—are pushing RF digitization forward, taking significant steps toward future implementations of SDR mobile devices based on the conceptual scheme of Figure 5.28a. It is interesting to note that one of the ICs in Table 5.9, reported by Martens *et al.* in [251], is clocked at 8.88 GHz, which is the highest sampling rate reported to date by CMOS ΣΔMs. The chip presented in [251] is a fourth-order CT BP-ΣΔM with a notch-frequency placed at $f_s/4$, that is, 2.2 GHz. This high-speed operation can be achieved by combining six time-interleaving quantizers with a polyphase decimation filter, which allows RF digitization at 2.2 GHz with 48-dB DR within 80-MHz bandwidth, while consuming 164 mW.

The majority of these state-of-the-art CT BP-ΣΔMs intended for RF-to-baseband digitization use a fixed center frequency or *notch frequency* and combine diverse strategies to relax the ADC requirements—such as out-of-band embedded filtering [250] or *subsampling* [234]. A good example was proposed by Ryckaert *et al.*, who reported the

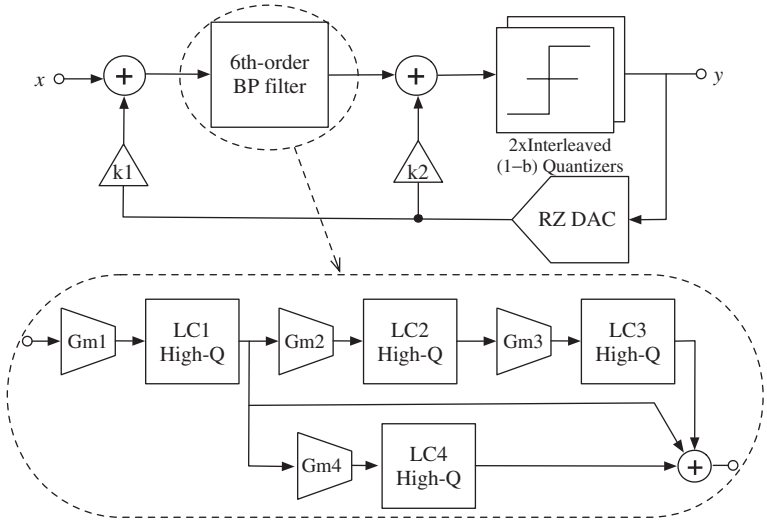


Figure 5.29 Example of a six-order Gm-LC BP-ΣΔM for RF digitization proposed by Ryckaert *et al.* [234]. It combines subsampling and two time-interleaved comparators to digitize 60-MHz signals placed at 2.4 GHz when clocked at 3 GHz.

design of a six-order BP-ΣΔM—conceptually depicted in Figure 5.29—which uses the subsampling technique [325] for digitizing signals centered at 2.4 GHz with $f_s = 3\,\text{GHz}$, that is, a ratio of 4/5. The loop filter—implemented by Q-enhanced LC resonators—is made tunable by varying tank capacitors such that the operation of the ADC can be adjusted to the desired center frequency. However, a wider tuning range is required in order to implement an SDR receiver.

Indeed, one of the problems associated with conventional BP-ΣΔMs with a fixed notch-frequency is that the RF receiver requires a programmable frequency synthesizer (Figure 5.28) in order to place the signal band within the pass-band of the BP-ΣΔM. This issue has motivated the interest for reconfigurable BP-ΣΔMs with wide tunable notch-frequency. On the basis of this strategy, Yamamoto *et al.* reported in [326] one of the widest tuning ranges for the notch-frequency achieved by SC BP-ΣΔMs, ranging from DC to 12.6 MHz. This tuning range can be significantly increased by using CT-ΣΔMs, as reported by Shibata *et al.* in [239], where a sixth-order reconfigurable LP/BP-ΣΔM IC features a DC-to-1 GHz tunable notch-frequency, at the price of consuming 550 mW.

Another interesting approach of using CT BP-ΣΔMs with a reconfigurable 0.8–2 GHz notch-frequency was reported by Gupta *et al.* in [253, 323]. In this case, a second-order topology is integrated together with a quadrature phase-locked loop (QPLL) to allow quadrature phase synchronization between a raised-cosine feedback DAC and the embedded quantizer. As shown in Table 5.9, the chip digitizes 1-MHz signals placed at 797 MHz with a DR of 8.3 bit, while consuming 41 mW, out of which 11 mW are consumed by the QPLL.

In conclusion, the trend toward RF digitization is expected to continue, as revealed by the increasing number of CT BP-ΣΔMs operating in the gigahertz range that are becoming commonplace. The main issue of this approach is the power consumption that

must be reduced in order to be competitive with conventional approaches based on a direct-conversion receiver made up of an analog signal conditioning (basically an LNA, a mixer, and a baseband filter) followed by a reconfigurable LP-ΣΔM [321].

5.4 Classification of State-of-the-Art References

For the sake of clarity, this chapter is concluded by grouping the references on the state-of-the-art ΣΔM ICs, according to the same classification criteria followed in Tables 5.1–5.13:

- SC single-loop single-bit LP-ΣΔMs: [12–54]
- SC single-loop multibit LP-ΣΔMs: [22, 55–98]
- SC cascade single-bit LP-ΣΔMs: [19, 99–116]
- SC cascade multibit LP-ΣΔMs: [38, 76, 110, 117–136]
- SC BP-ΣΔMs: [137–153]
- CT single-loop single-bit LP-ΣΔMs: [154–192]
- CT single-loop multibit LP-ΣΔMs: [193–226]
- CT cascade LP-ΣΔMs: [227–231, 297]
- CT BP-ΣΔMs: [232–258]
- ΣΔMs with time-coded quantization: [259–268]
- Hybrid ΣΔMs: [215, 228, 269–286]
- SC reconfigurable ΣΔMs: [85, 287–291, 300]
- CT reconfigurable ΣΔMs: [292–299].

5.5 Summary

This chapter presented an analysis of the state of the art on CMOS ΣΔMs, making a systematic classification of their architectures, circuit techniques, and target applications. On the basis of the practical approach followed in this book, the empirical data extracted from this statistical survey has been applied to identify design trends and challenges in order to help designers select the most suitable ΣΔM among different families for a given application and/or range of specifications. With this practical objective in mind, a number of design trade-offs have been discussed at different abstraction levels, namely architecture level (i.e., single-loop vs cascade, single-bit vs multibit, low-pass vs band-pass), circuit-level (i.e., SC vs CT, Gm-C vs active-RC), and technology process. The conclusions derived from the presented statistical and empirical analysis can be used as a practical guide to be incorporated in the top-down/bottom-up systematic design procedure described in this book.

The chapter is concluded with a review of cutting-edge ΣΔM ICs, considering diverse circuits and systems techniques, namely: advanced cascade topologies, hybrid implementations, MR architectures, time-coded quantization, digital-based loop-filter implementations, adaptive/reconfigurable architectures, and ultra-high-speed ΣΔMs.

As a final conclusion, it could be said that compared to other kinds of ADCs, ΣΔM techniques cover the widest resolution-versus-bandwidth region, targeting more and more applications and taking advantage of CMOS technology downscaling. This trend is expected to continue with the development and improvement of some architecture and circuit techniques—particularly using CT circuits, time-code quantization, and an

intensive use of digital-assisted analog circuits—which will all push ΣΔMs forward, increasing their dominance in a growing number of data-conversion applications.

References

[1] P. M. Aziz *et al*., "An Overview of Sigma-Delta Converters," *IEEE Signal Processing Magazine*, vol. **13**, pp. 61–84, January 1996.

[2] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*, IEEE Press, 2005.

[3] J. M. de la Rosa, "Sigma-Delta Modulators: Tutorial Overview, Design Guide, and State-of-the-Art Survey," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. **58**, pp. 1–21, January 2011.

[4] R. H. Walden, "Analog-to-Digital Converter Survey and Analysis," *IEEE Journal on Selected Areas in Communications*, vol. **17**, pp. 539–550, April 1999.

[5] B. Murmann, "A/D Converter Trends: Power Dissipation, Scaling and Digitally Assisted Architectures," *Proc. of the IEEE Custom Integrated Circuits Conf.*, pp. 105–112, 2008.

[6] H. S. Lee and C. G. Sodini, "Analog-to-Digital Converters: Digitizing the Analog World," *Proceedings of the IEEE*, vol. **96**, pp. 323–334, February 2008.

[7] B. E. Jonsson, "On CMOS Scaling and A/D-Converter Performance," *Proc. of the NORCHIP Conf.*, pp. 1–4, November 2010.

[8] B. E. Jonsson, "A Survey of A/D-Converter Performance Evolution," *Proc. of the IEEE Intl. Conf. on Electronics, Circuits, and Systems*, pp. 766–769, December 2010.

[9] B. E. Jonsson, "An Empirical Approach to Finding Energy Efficient ADC Architectures," *Proc. of the Intl. Workshop on ADC Modeling, Testing and Data Converter Analysis and Design and IEEE ADC Forum*, June 2011.

[10] G. Manganaro, *Advanced Data Converters*, Cambridge University Press, 2012.

[11] B. Murmann, ADC Performance Survey 1997–2012. [Online]. Available: http://www.stanford.edu/∼murmann/adcsurvey.html, 2012.

[12] Y. Chae and G. Han, "Low Voltage, Low Power, Inverter-Based Switched-Capacitor Delta-Sigma Modulator," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 369–373, May 2009.

[13] J. Roh *et al*., "A 0.9-V 60-μW 1-Bit Fourth-Order Delta-Sigma Modulator With 83-dB Dynamic Range," *IEEE Journal of Solid-State Circuits*, vol. **43**, pp. 361–370, February 2008.

[14] C. B. Wang *et al*., "A 113-dB DSD Audio ADC Using a Density-Modulated Dithering Scheme," *IEEE Journal of Solid-State Circuits*, vol. **28**, pp. 114–119, January 2003.

[15] L. Yao, M. Steyaert, and W. Sansen, "A 1-V 140-μW 88-dB Audio Sigma-Delta modulator in 90-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 1809–1818, November 2004.

[16] S. Gambini and J. Rabaey, "A 100 kS/S 65 dB DR ΣΔ ADC with 0.65V Supply Voltage," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 202–205, September 2007.

[17] H. Park *et al*., "A 0.7-V 100-dB 870-μW Digital Audio ΣΔ Modulator," *IEEE Symp. VLSI Circuits. Digest of Technical Papers*, pp. 178–179, 2008.

[18] V. Peluso *et al*., "A 900-mV Low-Power ΔΣ A/D Converter with 77-dB Dynamic Range," *IEEE Journal of Solid-State Circuits*, vol. **12**, pp. 1887–1897, December 1998.

[19] J. Sauerbrey *et al*., "0.65V Sigma-Delta Modulators," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 1021–1024, May 2003.

[20] A. L. Coban and P. E. Allen, "A 1.5V 1.0 mW Audio ΔΣ Modulator with 98 dB Dynamic Range," *IEEE ISSCC Digest of Technical Papers* pp. 50–51, February 1999.

[21] P. Rombouts *et al*., "A 250-kHz 94-dB Double-Sampling ΣΔ Modulation A/D Converter With a Modified Noise Transfer Function," *IEEE Journal of Solid-State Circuits*, vol. **10**, pp. 1657–1662, October 2003.

[22] J. Xu *et al*., "Power Optimization of High Performance ΔΣ Modulators for Portable Measurement Applications," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, 2010.

[23] F. Michel and M. Steyaert, "A 250 mW 7.5μW 61 dB SNDR CMOS SC ΔΣ Modulator Using a Near-Threshold-Voltage-Biased CMOS Inverter Technique," *IEEE ISSCC Digest of Technical Papers*, pp. 476–477, February 2011.

[24] Z. Cao *et al*., "A 14 mW 2.5 MS/s 14 bit ΣΔ Modulator Using Split-Path Pseudo-Differential Amplifiers," *IEEE Journal of Solid-State Circuits*, vol. **42**, pp. 2169–2179, October 2007.

[25] N. Klemmer and E. Hegazi, "A DLL-Biased, 14-Bit ΔΣ Analog-to-Digital Converter for GSM/GPRS/EDGE Handsets," *IEEE Journal of Solid-State Circuits*, vol. **41**, pp. 330–338, February 2006.

[26] K. A. Donoghue *et al*., "A Digitally Calibrated 5-mW 2-MS/s 4th-Order ΔΣ ADC in 0.25-μm CMOS with 94 dB SFDR," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 422–425, September 2010.

[27] B. Nowacki *et al*., "A 1.2V 300μW Second-Order Switched-Capacitor ΔΣ Modulator Using Ultra Incomplete Settling with 73 dB SNDR and 300 kHz BW in 130 nm CMOS," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 271–274, September 2011.

[28] F. Chen, S. Ramaswamy, and B. Bakkaloglu, "A 1.5V 1 mA 80 dB Passive ΣΔ ADC in 0.13μm Digital CMOS Process," *IEEE ISSCC Digest of Technical Papers*, vol. **54-55**, pp. 244–245, February 2003.

[29] M. Dessouky and A. Kaiser, "Very Low-Voltage Digital-Audio ΔΣ Modulator with 88-dB Dynamic Range Using Local Switch Bootstrapping," *IEEE Journal of Solid-State Circuits*, vol. **36**, pp. 349–355, March 2001.

[30] Y. Chae *et al*., "A 0.7V 36μW 85 dB-DR Audio ΔΣ Modulator Using Class-C Inverter," *IEEE ISSCC Digest of Technical Papers*, pp. 490–491, February 2008.

[31] T. Musah *et al*., "A 630μW Zero-Crossing-Based ΔΣ ADC Using Switched-Resistor Current Sources in 45 nm CMOS," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2009.

[32] J. Sauerbrey *et al*., "A 0.7-V MOSFET-Only Switched-Opamp ΣΔ Modulator in Standard Digital CMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. **37**, pp. 1662–1669, December 2002.

[33] M. Snoeij *et al*., "A 4th-Order Switched-Capacitor Sigma-Delta A/D Converter Using a High-Ripple Chebyshev Loop Filter," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 615–618, May 2001.

[34] J. Wang *et al*., "A 0.5V Feedforward Delta-Sigma Modulator with Inverter-Based Integrator," *Proc. of the IEEE European Solid-State Circuits Conf.*, September 2009.

[35] F. Medeiro *et al*., "Quick Design of High-Performance ΣΔ Modulators using CAD Tools: A 16.4 b 1.71 mW CMOS ΣΔM for 9.6 ksample/s A/D Conversion," *Proc. of 2nd IEEE-CAS Region 8 Workshop on Analog and Mixed IC Design)*, pp. 22–27, 1997.

[36] M. C. Huang and S. I. Lu, "A Fully Differential Comparator-Based Switched-Capacitor ΔΣ Modulator," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 369–373, May 2009.

[37] V. Peluso *et al*., "A 1.5-V 100-μW ΔΣ Modulator with 12-b Dynamic Range Using the Switched-Opamp Technique," *IEEE Journal of Solid-State Circuits*, vol. **32**, pp. 943–952, July 1997.

[38] B. P. Brandt and B. A. Wooley, "A 50-MHz Multibit Sigma-Delta Modulator for 12-b 2-MHz A/D Conversion," *IEEE Journal of Solid-State Circuits*, vol. **26**, pp. 1746–1756, December 1991.

[39] T. Burger and Q. Huang, "A 13.5-mW 185-MSample/s ΔΣ Modulator for UMTS/GSM Dual-Standard IF Reception," *IEEE Journal of Solid-State Circuits*, vol. **36**, pp. 1868–1878, December 2001.

[40] D. Senderowicz *et al*., "Low-Voltage Double-Sampled ΣΔ Converters," *IEEE Journal of Solid-State Circuits*, vol. **32**, pp. 1907–1919, December 1997.

[41] S. Au and B. H. Leung, "A 1.95-V, 0.34-mW, 12-b Sigma-Delta Modulator Stabilized by Local Feedback Loops," *IEEE Journal of Solid-State Circuits*, vol. **32**, pp. 321–328, March 1997.

[42] J. Gilo *et al*., "A 1.8V 94 dB Dynamic Range ΔΣ Modulator for Voice Applications," *IEEE ISSCC Digest of Technical Papers*, pp. 230–231, February 1996.

[43] T. Tille and R. Meyer, "A 1.8-V MOSFET-Only ΣΔ Modulator Using Substrate Biased Depletion-Mode MOS Capacitors in Series Compensation," *IEEE Journal of Solid-State Circuits*, vol. **36**, pp. 1041–1047, July 2001.

[44] P. Maulik *et al*., "A 16-Bit 250-kHz Delta-Sigma Modulator and Decimation Filter," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 458–467, April 2000.

[45] A. Gerosa *et al*., "A Fully-Integrated Two-Channel A/D Interface for hate Acquisition of Cardiac Signals in Implantable Pacemakers," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 157–160, September 2003.

[46] C. Thanh *et al*., "A Second-Order Double-Sampled Delta-Sigma Modulator Using Individual-Level Averaging," *IEEE Jounal of Solid-State Circuits*, vol. **32**, pp. 1269–1273, August 1997.

[47] M. Keskin *et al*., "A 1-V 10-MHz Clock-Rate 13-Bit CMOS ΔΣ Modulator Using Unity-Gain-Reset Opamps," *IEEE Journal of Solid-State Circuits*, vol. **37**, pp. 817–824, July 2002.

[48] D. A. Kerth *et al*., "A 120-dB Linear Switched-Capacitor Delta-Sigma Modulator," *IEEE ISSCC Digest of Technical Papers*, pp. 196–197, February 1994.

[49] D. B. Kasha *et al*., "A 16-mW, 120-dB Linear Switched-Capacitor Delta-Sigma Modulator With Dynamic Biasing," *IEEE Journal of Solid-State Circuits*, vol. **34**, pp. 921–926, July 1999.
[50] F. Op't Eynde *et al*., "A CMOS Fourth-Order 14 b 500 k-Sample/s Sigma-Delta ADC Converter," *IEEE ISSCC Digest of Technical Papers*, pp. 62–63, February 1991.
[51] H.-B. Le *et al*., "A Regulator-Free 84 dB DR Audio-Band ADC for Compact Digital Microphones," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, 2010.
[52] B. E. Boser and B. A. Wooley, "The Design of Sigma-Delta Modulation Analog-to-Digital Converters," *IEEE Journal of Solid-State Circuits*, vol. **23**, pp. 1298–1308, December 1988.
[53] S. Nadeem *et al*., "16-Channel Oversampled Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. **29**, pp. 1077–1085, September 1994.
[54] T. V. Burmas *et al*., "A Second-Order Double-Sampled Delta-Sigma Modulator Using Additive-Error Switching," *IEEE Journal of Solid-State Circuits*, vol. **31**, pp. 284–293, March 1996.
[55] A. Pena-Perez *et al*., "A 84 dB SNDR 100 kHz Bandwidth Low-Power Single Op-Amp Third-Order $\Delta\Sigma$ Modulator Consuming 140μW," *IEEE ISSCC Digest of Technical Papers*, pp. 478–479, February 2011.
[56] Z. Yang *et al*., "A 0.5-V 35-μW 85-dB DR Double-Sampled $\Delta\Sigma$ Modulator for Audio Applications," *IEEE Journal of Solid-State Circuits*, vol. **47**, pp. 722–735, March 2012.
[57] L. Liu *et al*., "A 95 dB SNDR Audio $\Delta\Sigma$ Modulator in 65 nm CMOS," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2011.
[58] N. Maghari and U.-K. Moon, "A Third-Order DT $\Delta\Sigma$ Modulator Using Noise-Shaped Bi-Directional Single-Slope Quantizer," *IEEE Journal of Solid-State Circuits*, vol. **46**, pp. 2882–2891, December 2011.
[59] K. Lee *et al*., "A Power-Efficient Two-Channel Time-Interleaved $\Sigma\Delta$ Modulator for Broadband Applications," *IEEE Journal of Solid-State Circuits*, vol. **42**, pp. 1206–1215, June 2007.
[60] L. Liu *et al*., "A 1-V 15-Bit Audio $\Delta\Sigma$-ADC in 0.18μm CMOS," *IEEE Transactions on Circuits and Systems I—Regular Papers*, vol. **59**, pp. 915–925, May 2012.
[61] H. Park *et al*., "A 0.7-V 870-μW Digital-Audio CMOS Sigma-Delta Modulator," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 1078–1088, April 2009.
[62] K. Lee *et al*., "A Noise-Coupled Time-Interleaved $\Delta\Sigma$ ADC with 4.2 MHz BW, -98 dB THD, and 79 dB SNDR," *IEEE ISSCC Digest of Technical Papers*, pp. 494–495, February 2008.
[63] J. Koh, Y. Choi, and G. Gómez, "A 66 dB DR 1.2V 1.2 mW Single-Amplifier Double-Sampling 2nd-order $\Delta\Sigma$ ADC for WCDMA in 90 nm CMOS," *IEEE ISSCC Digest of Technical Papers*, pp. 170–171, February 2005.
[64] S. Kwon and F. Maloberti, "A 14 mW Multi-bit $\Delta\Sigma$ Modulator with 82 dB SNR and 86 dB DR for ADSL2+," *IEEE ISSCC Digest of Technical Papers*, pp. 161–162, February 2006.
[65] K. Lee *et al*., "An 8.1-mW, 82 dB Delta-Sigma ADC with 1.9-MHz BW and −98 dB THD," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 2202–2211, August 2009.
[66] L. Liu *et al*., "A 1V 350μW 92 dB SNDR 24 kHz $\Delta\Sigma$ Modulator in 0.18μm CMOS," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, 2010.
[67] N. Maghari *et al*., "A +5 dBFS Third-Order Extended Dynamic Range Single-Loop $\Delta\Sigma$ Modulator," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2010.
[68] K. Tiew *et al*., "A 0.06-mm^2 Double-Sampling Single-OTA 2nd-Order $\Delta\Sigma$ Modulator in 0.18-μm CMOS Technology," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, 2011.
[69] R. Gaggl, M. Inversi, and A. Wiesbauer, "A Power Optimized 14-Bit SC $\Delta\Sigma$ Modulator for ADSL CO Applications," *IEEE ISSCC Digest of Technical Papers*, pp. 82–83, February 2004.
[70] J. Chae *et al*., "A 63 dB 16 mW 20 MHz BW Double-Sampled $\Delta\Sigma$ Analog-to-Digital Converter with An Embedded-Adder Quantizer," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2010.
[71] Y. Wang *et al*., "A 2.5MHz BW and 78 dB SNDR Delta-Sigma Modulator Using Dynamically Biased Amplifiers," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2008.
[72] E. Bonizzoni *et al*., "Third-Order $\Sigma\Delta$ Modulator with 61-dB SNR and 6-MHz Bandwidth Consuming 6 mW," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 218–221, September 2008.
[73] R. Reutemann, P. Balmelli, and Q. Huang, "A 33 mW 14 b 2.5 MSample/s $\Sigma\Delta$ A/D Converter in 0.25μm Digital CMOS," *IEEE ISSCC Digest of Technical Papers*, vol. **1**, p. 316, 2002.
[74] Y. Fujimoto *et al*., "A 100 MS/s 4 MHz Bandwidth 70 dB SNR $\Delta\Sigma$ ADC in 90 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 1697–1708, June 2009.
[75] P. Balmelli and Q. Huang, "A 25-MS/s 14-b 200-mW $\Sigma\Delta$ Modulator in 0.18-μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 2161–2169, December 2004.

[76] J. Yu and F. Maloberti, "A Low-Power Multi-Bit ΣΔ Modulator in 90-nm Digital CMOS Without DEM," *IEEE Journal of Solid-State Circuits*, vol. **40**, pp. 2428–2436, December 2005.
[77] G. Gomez and B. Haroun, "A 1.5V 2.4/2.9 mW 79/50 dB DR ΣΔ Modulator for GSM/WCDMA in a 0.14μm Digital Process," *IEEE ISSCC Digest of Technical Papers*, February 2002.
[78] M. G. Kim *et al*., "A 0.9 V 92 dB Double-Sampled Switched-RC Delta-Sigma Audio ADC," *IEEE Journal of Solid-State Circuits*, vol. **43**, pp. 1195–1206, May 2008.
[79] Y. Kim *et al*., "A 105.5 dB, 0.49 mm^2 Audio ΣΔ Modulator Using Chopper Stabilization and Fully Randomized DWA," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2008.
[80] R. Gaggl *et al*., "A 85-dB Dynamic Range Multibit Delta-Sigma ADC for ADSL-CO Applications in 0.18-μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **38**, pp. 1105–1114, July 2003.
[81] J. Wu *et al*., "Multi-bit Sigma Delta ADC with Reduced Feedback Levels, Extended Dynamic Range and Increased Tolerance for Analog Imperfections," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2007.
[82] Y. Geerts *et al*., "A High-Performance Multibit ΔΣ CMOS ADC," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 1829–1840, December 1999.
[83] J. Chen *et al*., "A 94 dB SFDR 78 dB DR 2.2MHz BW Multi-bit Delta-Sigma Modulator with Noise Shaping DAC," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2007.
[84] A. J. Chen and Y. P. Xu, "Multibit Delta-Sigma Modulator With Noise-Shaping Dynamic Element Matching," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 1125–1133, June 2009.
[85] M. R. Miller and C. S. Petrie, "A Multibit Sigma-Delta ADC for Multimode Receivers," *IEEE Journal of Solid-State Circuits*, vol. **38**, pp. 475–482, March 2003.
[86] Y. Yang *et al*., "A 114-dB 68-mW Chopper-Stabilized Stereo Multibit Audio ADC in 5.62 mm^2," *IEEE Journal of Solid-State Circuits*, vol. **38**, pp. 2061–2068, December 2003.
[87] O. Nys and R. K. Henderson, "A 19-bit Low-Power Multibit Sigma-Delta ADC Based on Data Weighted Averaging," *IEEE Journal of Solid-State Circuits*, vol. **32**, pp. 933–942, July 1997.
[88] T. Kuo *et al*., "A Wideband CMOS Sigma-Delta Modulator With Incremental Data Weighted Averaging," *IEEE Journal of Solid-State Circuits*, vol. **37**, pp. 11–17, January 2002.
[89] Y. Yang *et al*., "A Single Die 124 dB Stereo Audio Delta Sigma ADC With 111 dB THD," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 252–255, September 2007.
[90] R. T. Baird *et al*., "A Low Oversampling Ratio 14-b 500-kHz ΔΣ ADC With a Self-Calibrated Multibit DAC," *IEEE Journal of Solid-State Circuits*, vol. **31**, pp. 312–320, March 1996.
[91] J. Johansson *et al*., "A 16-bit 60μW Multi-Bit ΣΔ Modulator for Portable ECG Applications," *Proc. of the IEEE European Solid-State Circuits Conf.*, September 2003.
[92] J. Gilo *et al*., "A 5V, 118 dB ΔΣ Analog-to-Digital Converter for Wideband Digital Audio," *IEEE ISSCC Digest of Technical Papers*, pp. 218–219, February 1997.
[93] E. Fogleman and I. Galton, "A Dynamic Element Matching Technique for Reduced-Distortion Multibit Quantization in Delta-Sigma ADCs," *IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing*, vol. **48**, pp. 158–170, February 2001.
[94] E. Fogleman *et al*., "A 3.3-V Single-Poly CMOS Audio ADC Delta-Sigma Modulator with 98-dB Peak SINAD and 105-dB Peak SFDR," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 297–307, March 2000.
[95] A. Prasad *et al*., "A 120 dB 300 mW Stereio Audio A/D Converter with 110 dB THD+N," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 191–194, September 2004.
[96] F. Chen and B. H. Leung, "A High Resolution Multibit Sigma-Delta Modulator with Individual Level Averaging," *IEEE Journal of Solid-State Circuits*, vol. **30**, pp. 453–460, April 1995.
[97] J. Sauerbrey *et al*., "A Dual-Quantization Multi-Bit Sigma Delta Analog/Digital Converter," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 437–440, May 1994.
[98] J. Xu *et al*., "A 9μW 88 dB DR Fully-Clocked Switched-Opamp ΔΣ Modulator with Novel Power and Area Efficient Resonator," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2011.
[99] L. Wang and L. Theogarajan, "An 18μW 79 dB-DR 20 kHz-BW MASH ΔΣ Modulator Utilizing Self-Biased Amplifiers for biomedical Applications," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2011.
[100] K. A. Donoghue *et al*., "A Digitally Corrected 5-mW 2-MS/s SC ΔΣ ADC in 0.25-μm CMOS With 94-dB SFDR," *IEEE Journal of Solid-State Circuits*, vol. **46**, pp. 2673–2684, November 2011.
[101] K. Cornelissens and M. Steyaert, "A 1-V 84-dB DR 1-MHz Bandwidth Cascade 3-1 Delta-Sigma ADC in 65-nm CMOS," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 332–335, 2009.

[102] S. Rabii and B. A. Wooley, "A 1.8V Digital-Audio Sigma-Delta Modulator in 0.8μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **32**, pp. 783–796, June 1997.
[103] C. Kuo *et al*., "A Low-Voltage Fourth-Order Cascade Delta-Sigma Modulator in 0.18-μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **57**, pp. 2450–2461, September 2010.
[104] O. Oliaei *et al*., "A 5-mW Sigma-Delta Modulator With 84-dB Dynamic Range for GSM/EDGE," *IEEE Journal of Solid-State Circuits*, vol. **37**, pp. 2–10, January 2002.
[105] J. M. de la Rosa *et al*., "A CMOS 110-dB@40-kS/s Programmable-Gain Chopper-Stabilized Third-Order 2-1 Cascade Sigma-Delta Modulator for Low-Power High-Linearity Automotive Sensor ASICs," *IEEE Journal of Solid-State Circuits*, vol. **40**, pp. 2246–2264, November 2005.
[106] A. Morgado, R. del Río, and J. M. de la Rosa, "An Adaptive ΣΔ Modulator for Multi-Standard Hand-Held Wireless Devices," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, pp. 232–235, 2007.
[107] Y. Geerts *et al*., "A 3.3-V, 15-bit, Delta-Sigma ADC with a Signal Bandwidth of 1.1 MHz for ADSL Applications," *IEEE Journal of Solid-State Circuits*, vol. **34**, pp. 1829–1840, July 1999.
[108] G. Yin *et al*., "A 16-b 320-kHz CMOS A/D Converter Using Two-Stage Third-Order ΣΔ Noise Shaping," *IEEE Journal of Solid-State Circuits*, vol. **28**, pp. 640–647, June 1993.
[109] A. M. Marques *et al*., "A 15-b Resolution 2-MHz Nyquist Rate ΔΣ ADC in a 1-μm CMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. **33**, pp. 1065–1075, July 1998.
[110] J. C. Morizio *et al*., "14-bit 2.2-MS/s Sigma-Delta ADC's," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 968–976, July 2000.
[111] G. J. Gomez, "A 102-dB Spurious-Free DR ΣΔ ADC Using a Dynamic Dither Scheme," *IEEE Journal of Solid-State Circuits*, vol. **47**, pp. 531–535, June 2000.
[112] L. A. Williams and B. A. Wooley, "A Third-Order Sigma-Delta Modulator with Extended Dynamic Range," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 193–202, March 1994.
[113] C. B. Wang, "A 20-bit 25-kHz Delta-Sigma A/D Converter Utilizing a Frequency-Shaped Chopper Stabilization Scheme," *IEEE Journal of Solid-State Circuits*, vol. **36**, pp. 566–569, March 2001.
[114] M. Rebeschini, N. R. van Bavel, P. Rakers, R. Greene, J. Caldwell, and J. R. Haug, "A 16-b 160 kHz CMOS A/D Converter Using Sigma-Delta Modulation," *IEEE Journal of Solid-State Circuits*, vol. **25**, pp. 431–440, April 1990.
[115] I. Fujimori *et al*., "A 5-V Single-Chip Delta-Sigma Audio A/D Converter with 111 dB Dynamic Range," *IEEE Journal of Solid-State Circuits*, vol. **32**, pp. 329–336, March 1997.
[116] G. Miao *et al*., "An Oversampled A/D Converter Using Cascaded Fourth Order Sigma-Delta Modulation and Current Steering Logic," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 412–415, May 1998.
[117] R. Zanbaghi *et al*., "A 75 dB SNDR, 10MHz Conversion Bandwidth Stage-Shared 2-2 MASH ΔΣ Modulator Dissipating 9 mW," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2011.
[118] P. Malla *et al*., "A 28 mW Spectrum-Sensing Reconfigurable 20MHz 72 dB-SNR 70 dB-SNDR DT ΔΣ ADC for 802.11 n/WiMAX Receivers," *IEEE ISSCC Digest of Technical Papers*, February 2008.
[119] K. Yamamoto and A. C. Carusone, "A 1-1-1-1 MASH Delta-Sigma Modulator Using Dynamic Comparator-Based OTAs," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2011.
[120] N. Maghari *et al*., "74 dB SNDR Multi-Loop Sturdy-MASH Delta-Sigma Modulator Using 35 dB Open-Loop Opamp Gain," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 2212–2221, August 2009.
[121] A. Gharbiya and D. A. Johns, "A 12-bit 3.125MHz Bandwidth 0-3 MASH Delta-Sigma Modulator," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 2010–2018, July 2009.
[122] A. Dezzani and E. Andre, "A 1.2-V Dual-Mode WCDMA/GPRS ΣΔ Modulator," *IEEE ISSCC Digest of Technical Papers*, February 2003.
[123] T.-H. Chang *et al*., "A 2.5-V 14-bit, 180-mW Cascaded ΣΔ ADC for ADSL2+ Application," *IEEE Journal of Solid-State Circuits*, vol. **42**, pp. 2357–2368, November 2007.
[124] J. Paramesh *et al*., "An 11-Bit 330MHz 8X OSR $\Sigma - \Delta$ Modulator for Next-Generation WLAN," *IEEE Symposium VLSI Circuits. Digest of Technical Papers*, December 2006.
[125] K. Vleugels, S. Rabii, and B. Wooley, "A 2.5-V Sigma-Delta Modulator for Broadband Communications Applications," *IEEE Journal of Solid-State Circuits*, vol. **36**, pp. 1887–1899, December 2001.
[126] I. Fujimori *et al*., "A 90-dB SNR 2.5-MHz Output-Rate ADC Using Cascaded Multibit Delta-Sigma Modulation at 8× Oversampling Ratio," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 1820–1828, December 2000.
[127] R. del Río *et al*., *CMOS Cascade ΣΔ Modulators for Sensors and Telecom: Error Analysis and Practical Design*, Springer, 2006.

[128] G.-C. Ahn *et al*., "A 0.6-V 82-dB Delta-Sigma Audio ADC Using Switched-RC Integrators," *IEEE Journal of Solid-State Circuits*, vol. **40**, pp. 1868–1878, December 2005.
[129] R. Brewer *et al*., "A 100 dB SNR 2.5MS/s Output Data Rate ΔΣ ADC," *IEEE ISSCC Digest of Technical Papers*, pp. 172–173, February 2005.
[130] F. Medeiro *et al*., "A 13-bit, 2.2-MS/s, 55-mW Multibit Cascade ΣΔ Modulator in CMOS 0.7-μm Single-Poly Technology," *IEEE Journal of Solid-State Circuits*, vol. **34**, pp. 748–760, June 1999.
[131] S. K. Gupta and V. Fong, "A 64-MHz Clock-Rate ΣΔ ADC With 88-dB SNDR and −105-dB IM3 Distortion at a 1.5-MHz Signal Frequency," *IEEE Journal of Solid-State Circuits*, vol. **37**, pp. 1653–1661, December 2002.
[132] H. Lampinen and O. Vinio, "An Optimization Approach to Designing OTAs for Low-Voltage Sigma-Delta Modulators," *IEEE Journal of Solid-State Circuits*, vol. **50**, pp. 1665–1671, December 2001.
[133] T. C. Caldwell and D. A. Johns, "An 8th-Order MASH Delta-Sigma With an OSR of 3," *Proc. of the IEEE European Solid-State Circuits Conf.*, September 2009.
[134] A. Tabatabaei *et al*., "A Dual Channel ΣΔ ADC with 40MHz Aggregate Signal Bandwidth," *IEEE ISSCC Digest of Technical Papers*, February 2003.
[135] A. R. Feldman *et al*., "A 13-Bit, 1.4-MS/s Sigma-Delta Modulator for RF Baseband Channel Applications," *IEEE Journal of Solid-State Circuits*, vol. **33**, pp. 1462–1469, October 1998.
[136] I. Dedic, "A Sixth-Order Triple-Loop ΣΔ CMOS ADC with 90 dB SNR and 100 kHz Bandwidth," *IEEE ISSCC Digest of Technical Papers*, pp. 188–189, 1994.
[137] F. Ying and F. Maloberti, "A mirror image free two-path bandpass ΣΔ modulator with 72 dB SNR and 86 dB SFDR," *IEEE ISSCC Digest of Technical Papers*, pp. 84–85, February 2004.
[138] T. O. Salo *et al*., "80-MHz Bandpass ΔΣ Modulators for Multimode Digital IF Receivers," *IEEE Journal of Solid-State Circuits*, vol. **38**, pp. 464–474, March 2003.
[139] J. Galdi *et al*., "Two-Path Band-Pass ΣΔ Modulator with 40-MHz IF 72-dB DR at 1-MHz Bandwidth Conuming 16 mW," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 248–251, September 2007.
[140] R. Maurino and C. Papavassiliou, "A 10 mW 81 dB Cascaded Multibit Quadrature ΣΔ ADC with a Dynamic Element Matching Scheme," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 451–454, September 2005.
[141] K. Yamamoto *et al*., "A Delta-Sigma Modulator with a Widely Programmable Center Frequency and 82-dB Peak SNDR," *Proc. of the IEEE Custom Integrated Circuits Conf.*, pp. 65–69, September 2007.
[142] C.-H. Kuo and S.-I. Liu, "A 1-V 10.7-MHz Fourth-Order Bandpass ΔΣ Modulators Using Two Switched Opamps," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 2041–2045, November 2004.
[143] A. Tabatabaei and A. Wooley, "A Two-Path Bandpass Sigma-Delta Modulator with Extended Noise Shaping," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 1799–1809, December 2000.
[144] A. Hairapetian, "An 81 MHz IF receiver in CMOS," *IEEE Journal of Solid-State Circuits*, vol. **31**, pp. 1981–1986, December 1996.
[145] T. Yamamoto, M. Kasahara, and T. Matsuura, "A 63 mA 112/94 dB DR IF Bandpass ΔΣ Modulator With Direct Feed-Forward Compensation and Double Sampling," *IEEE Journal of Solid-State Circuits*, vol. **43**, pp. 1783–1794, August 2008.
[146] T. Ueno *et al*., "A Fourth-Order Bandpass $\Delta - \Sigma$ Modulator Using Second-Order Bandpass Noise-Shaping Dynamic Element Matching," *IEEE Journal of Solid-State Circuits*, vol. **37**, pp. 809–816, July 2002.
[147] B.-S. Song, "A Fourth-Order Bandpass Delta-Sigma Modulator with Reduced Number of Op Amps," *IEEE Journal of Solid-State Circuits*, vol. **30**, pp. 1309–1315, December 1995.
[148] T. O. Salo *et al*., "A 80-MHz Bandpass ΣΔ Modulator for a 100-MHz IF Receiver," *IEEE Journal of Solid-State Circuits*, vol. **37**, pp. 798–808, July 2002.
[149] A. K. Ong and B. A. Wooley, "A Two-Path Bandpass ΣΔ Modulator for Digital IF Extraction at 20MHz," *IEEE Journal of Solid-State Circuits*, vol. **32**, pp. 1920–1934, December 1997.
[150] W.-T. Cheng *et al*., "A 75 dB Image Rejection IF-Input Quadrature Sampling SC ΣΔ Modulator," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 455–458, September 2005.
[151] P. Cusinato *et al*., "A 3.3-V CMOS 10.7-MHz Sixth-Order Bandpass ΣΔ Modulator with 74-dB Dynamic Range," *IEEE Journal of Solid-State Circuits*, vol. **36**, pp. 629–638, April 2001.
[152] S. A. Jantzi *et al*., "Quadrature Bandpass ΔΣ Modulation for Digital Radio," *IEEE Journal of Solid-State Circuits*, vol. **32**, pp. 1935–1950, December 1997.

[153] V. S. Cheung *et al*., "A 1-V 10.7-MHz Switched-Opamp Bandpass ΣΔ Modulator Using Double-Sampling Finite-Gain-Compensation Technique," *IEEE Journal of Solid-State Circuits*, vol. **37**, pp. 1215–1225, October 2002.

[154] P. Shettigar and S. Pavan, "A 15 mW 3.6GS/s CT-ΣΔ ADC with 36MHz Bandwidth and 83 dB DR in 90 nm CMOS," *IEEE ISSCC Digest of Technical Papers*, pp. 156–157, February 2012.

[155] K. Philips *et al*., "A Continuous-Time ΣΔ ADC With Increased Immunity to Interferers," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 2170–2177, December 2005.

[156] S. Pavan and P. Sankar, "Power Reduction in Continuous-Time Delta-Sigma Modulators Using the Assisted Opamp Technique," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 1365–1379, July 2010.

[157] S. Pavan and P. Sankar, "A 110μW Single Bit Audio Continuous-Time Oversampled Converter with 92.5-dB Dynamic Range," *Proc. of the IEEE European Solid-State Circuits Conf.*, September 2009.

[158] F. Munoz, K. Philips, and A. Torralba, "A 4.7 mW 89.5 dB DR CT Complex ΔΣ ADC with Built-In LPF," *IEEE ISSCC Digest of Technical Papers*, February 2005.

[159] J. Zhan *et al*., "A 0.6-V 82-dB 28.6-μW Continuous-Time Audio Delta-Sigma Modulator," *IEEE Journal of Solid-State Circuits*, vol. **46**, pp. 2326–2335, October 2011.

[160] A. Jain and S. Pavan, "A 4 mW 1GS/s Continuous-Time ΔΣ Modulator with 15.6MHz Bandwidth and 67 dB Dynamic Range," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 259–262, September 2011.

[161] V. Srinivasan *et al*., "A 20 mW 61 dB SNDR (60MHz BW) 1 b 3rd-Order Continuous-Time Delta-Sigma Modulator Clocked at 6GHz in 45 nm CMOS," *IEEE ISSCC Digest of Technical Papers*, pp. 158–159, February 2012.

[162] K. Matsukawa *et al*., "A 69.8 dB SNDR 3rd-order Continuous-Time Delta-Sigma Modulator with an Ultimate Low-Power Tuning System for a Worldwide Digital TV-Receiver," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2010.

[163] P. Crombez *et al*., "A Single-Bit 6.8 mW 10MHz Power-Optimized Continuous-Time ΔΣ with 67 dB DR in 90 nm CMOS," *Proc. of the IEEE European Solid-State Circuits Conf.*, September 2009.

[164] P. Crombez *et al*., "A 500 kHz-10MHz Multimode Power-Performance Scalable 83-to-67 dB DR CTΣΔ in 90 nm Digital CMOS with Flexible Analog Core Circuitry," *IEEE Symp. VLSI Circuits. Digest of Technical Papers*, pp. 70–71, 2009.

[165] X. Xing *et al*., "A 40MHz 12 bit 84.2 dB-SFDR Continuous-Time Delta-Sigma Modulator in 90 nm CMOS," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, 2011.

[166] K. Matsukawa *et al*., "A Fifth-Order Continuous-Time Delta-Sigma Modulator With Single-Opamp Resonator," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 697–706, April 2010.

[167] A. Das *et al*., "A 4th-order 86 dB CT ΔΣ ADC with Two Amplifiers in 90 nm CMOS," *IEEE ISSCC Digest of Technical Papers*, February 2005.

[168] D. Kim *et al*., "A Continuous-Time, Jitter Insensitive ΣΔ Modulator Using a Digitally Linearized Gm-C Integrator with Embedded SC Feedback DAC," *IEEE Symp. VLSI Circuits. Digest of Technical Papers*, pp. 38–39, 2011.

[169] R. H. Veldhoven *et al*., "Technology portable, 0.04 mm^2, GHz-rate ΣΔ Modulators in 65 nm and 45 nm CMOS," *IEEE Symp. VLSI Circuits. Digest of Technical Papers*, pp. 72–73, 2009.

[170] B. Putter, "ΣΔ ADC with Finite Impulse Response Feedback DAC," *IEEE ISSCC Digest of Technical Papers*, February 2004.

[171] E. Prefasi *et al*., "A 0.1mm^2, Wide Bandwidth Continuous-Time ΣΔ ADC Based on a Time Encoding Quantizer in 0.13μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 2745–2754, October 2009.

[172] J. Zhang *et al*., "A 1.2-V 2.7-mW 160MHz Continuous-Time Delta-Sigma Modulator with Input-Feedforward Structure," *Proc. of the IEEE Custom Integrated Circuits Conf.*, pp. 475–478, September 2009.

[173] R. van Veldhoven, "A 3.3 mW ΣΔ Modulator for UMTS in 0.18μm CMOS with 70 dB Dynamic Range in 2MHz Bandwidth," *IEEE ISSCC Digest of Technical Papers*, February 2002.

[174] F. Gerfers, M. Ortmanns, and Y. Manoli, "A 1.5-V 12-bit Power-Efficient Continuous-Time Third-Order ΣΔ Modulator," *IEEE Journal of Solid-State Circuits*, vol. **38**, pp. 1343–1352, August 2003.

[175] K. Philips, "A 4.4 mW 76 dB Complex ΣΔ ADC for Bluetooth Receivers," *IEEE ISSCC Digest of Technical Papers*, February 2003.

[176] L. Breems *et al*., "A 1.8-mW CMOS ΣΔ Modulator with Integrated Mixer for A/D Conversion of IF Signals," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 468–475, April 2000.

[177] Y.-C. Chang *et al*., "A 4MHz BW 69 dB SNDR Continuous-Time Delta-Sigma Modulator with Reduced Sensitivity to Clock Jitter," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, pp. 265–268, 2011.
[178] T. Nagai *et al*., "A 1.2V 3.5 mW ΔΣ Modulator with a Passive Current Summing Network and a Variable Gain Function," *IEEE ISSCC Digest of Technical Papers*, February 2005.
[179] L. Samid and Y. Manoli, "A Micro Power Continuous-Time ΣΔ Modulator," *Proc. of the IEEE European Solid-State Circuits Conf.*, September 2003.
[180] L. S. M Anderson, "Design and Measurement of a CT ΔΣ ADC With Switched-Capacitor Switched-Resistor Feedback," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 473–483, February 2009.
[181] E. H. Dagher *et al*., "A 2-GHz Analog-to-Digital Delta-Sigma Modulator for CDMA Receivers With 79-dB Signal-to-Noise Ratio in 1.23-MHz Bandwidth," *IEEE Journal of Solid-State Circuits*, vol. **38**, pp. 1819–1828, November 2004.
[182] C.-H. Lin *et al*., "A 5MHz Nyquist Rate Continuous-Time Sigma-Delta Modulator for Wideband Wireless Communication," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 368–371, May 1999.
[183] K.-P. Pun, S. Chatterjee, and P. Kinget, "A 0.5-V 74-dB SNDR 25-kHz Continuous-Time Delta-Sigma Modulator With a Return-to-Open DAC," *IEEE Journal of Solid-State Circuits*, vol. **42**, pp. 496–507, March 2007.
[184] F. Gerfers *et al*., "Implementation of a 1.5V Low-Power Clock-Jitter Insensitive Continuous-Time ΣΔ Modulator," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 652–655, May 2002.
[185] L. Luh *et al*., "A 400MHz 5th-Order CMOS Continuous-Time Switched-Current ΣΔ Modulator," *Proc. of the IEEE European Solid-State Circuits Conf.*, September 2000.
[186] E. van der Zwan and E. Dijkmans, "A 0.2 mW CMOS ΣΔ Modulator for Speech Coding with 80 dB Dynamic Range," *IEEE Journal of Solid-State Circuits*, vol. **31**, pp. 1873–1880, December 1996.
[187] L. Hernández-Corporales *et al*., "A 1.2-MHz 10-bit Continuous-Time Sigma-Delta ADC Using a Time Ecoding Quantizer," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. **56**, pp. 16–20, January 2009.
[188] F. Cannillo *et al*., "1.4V 13μW 83 dB DR CT-ΣΔ Modulator with Dual-Slope Quantizer and PWM DAC for Biopotential Signal Acquisition," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 267–270, September 2011.
[189] L. Luh *et al*., "A 50-MHz Continuous-Time Switched-Current ΣΔ Modulator," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 579–582, May 1998.
[190] M. Ortmanns *et al*., "A Continuous-Time Sigma-Delta Modulator with Switched Capacitor Controlled Current ModeFeedback," *Proc. of the IEEE European Solid-State Circuits Conf.*, September 2003.
[191] H. Zare-Hoseini *et al*., "A Low-Power Continuous-Time ΔΣ *Modulator for Electret Microphone Applications,*" Proc. of the IEEE Asian Solid-State Circuits Conf., 2010.
[192] N. Sarhangnejad *et al*., "A Continuous-Time ΣΔ Modulator with a Gm-C Input Stage, 120-dB CMRR and -87 dB THD," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, 2011.
[193] S. Pavan, *et al*., "A Power Optimized Continuous-Time Delta-Sigma ADC for Audio Applications," *IEEE Journal of Solid-State Circuits*, vol. **43**, pp. 351–360, February 2008.
[194] P. Witte *et al*., "A 72 dB-DR ΔΣ CT Modulator Using Digitally Estimated Auxiliary DAC Linearization Achieving 88 fJ/conv in a 25MHz BW," *IEEE ISSCC Digest of Technical Papers*, pp. 154–155, February 2012.
[195] G. Mitteregger, C. Ebner, S. Mechnig, T. Blon, C. Holuigue, and E. Romani, "A 20-mW 640-MHz CMOS Continuous-Time ΣΔ ADC With 20-MHz Signal Bandwidth, 80-dB Dynamic Range and 12-bit ENOB," *IEEE Journal of Solid-State Circuits*, vol. **41**, pp. 2641–2649, December 2006.
[196] K. Matsukawa *et al*., "A 5th-Order Delta-Sigma Modulator with Single-Opamp Resonator," *IEEE Symp. VLSI Circuits. Digest of Technical Papers*, pp. 68–69, 2009.
[197] J. G. Kauffman *et al*., "An 8.5 mW Continuous-Time ΔΣ Modulator With 25MHz Bandwidth Using Digital Background DAC Linearization to Achieve 63.5 dB SNDR and 81 dB SFDR," *IEEE Journal of Solid-State Circuits*, vol. **46**, pp. 2869–2881, December 2011.
[198] S.-J. Huang and Y.-Y. Li, "A 1.2V 2MHz BW 0.084 mm^2 CT ΔΣ ADC with -97.7 dBc THD and 80 dB DR Using Low-Latency DEM," *IEEE ISSCC Digest of Technical Papers*, pp. 172–173, February 2009.
[199] L. Dorrer *et al*., "A 3-mW 74-dB SNR 2-MHz Continuous-Time Delta-Sigma ADC With a Tracking ADC Quantizer in 0.13-μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **40**, pp. 2416–2427, December 2005.

[200] J.-C. Tsai *et al*., "A Continuous-Time ΔΣ ADC with Clock Timing Calibration," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, pp. 369–370, 2008.
[201] M. Ranjbar *et al*., "A Robust STF 6 mW CT ΔΣ Modulator with 76 dB Dynamic Range and 5MHz Bandwidth," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2010.
[202] K. Reddy and S. Pavan, "A 20.7 mW Continuous-Time ΔΣ Modulator with 15MHz Bandwidth and 70 dB Dynamic Range," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 210–213, September 2008.
[203] H. Kim *et al*., "Adaptive Blocker Rejection Continuous-Time ΣΔ ADC for Mobile WiMAX Applications," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 2766–2779, October 2009.
[204] Y. S. Shu, B. S. Song, and K. Bacrania, "A 65 nm CMOS CT ΔΣ Modulator with 81 dB DR and 8MHz BW Auto-Tuned by Pulse Injection," *IEEE ISSCC Digest of Technical Papers*, February 2008.
[205] J. G. Kauffman *et al*., "A 78 dB SNDR 87 mW 20MHz Bandwidth Continuous-Time ΔΣ ADC With VCO-Based Integrator and Quantizer Implemented in 0.13μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 3344–3358, December 2009.
[206] M. Moyal *et al*., "A 700/900 mW/Channel CMOS Dual Analog Front-End IC for VDSL with Integrated 11.5/14.5 dBm Line Drivers," *IEEE ISSCC Digest of Technical Papers*, February 2003.
[207] M. Bolatkale *et al*., "A 4GHz Continuous-Time ΔΣ ADC With 70 dB DR and −74 dBFS THD in 125 MHz BW," *IEEE Journal of Solid-State Circuits*, vol. **46**, pp. 2857–2868, December 2011.
[208] X. Chenand *et al*., "A 18 mW CT ΔΣ Modulator with 25MHz Bandwidth for Next Generation Wireless Applications," *Proc. of the IEEE Custom Integrated Circuits Conf.*, pp. 73–76, September 2007.
[209] V. Dhanasekaran *et al*., "A 20MHz BW 68 dB DR CT ΔΣ ADC Based on a Multi- Bit Time-Domain Quantizer and Feedback Element," *IEEE ISSCC Digest of Technical Papers*, pp. 174–175, February 2009.
[210] W. Yang *et al*., "A 100 mW 10MHz-BW CT ΔΣ Modulator with 87 dB DR and 91 dBc IMD," *IEEE ISSCC Digest of Technical Papers*, February 2008.
[211] M. Vadipour *et al*., "A 2.1 mW/3.2 mW Delay-Compensated GSM/WCDMA ΣΔ Analog-Digital Converter," *IEEE Symp. VLSI Circuits. Digest of Technical Papers*, pp. 180–181, 2008.
[212] Z. Li and T. S. Fiez, "A 14 Bit Continuous-Time Delta-Sigma A/D Modulator With 25MHz Signal Bandwidth," *IEEE Journal of Solid-State Circuits*, vol. **42**, pp. 1873–1883, September 2007.
[213] L. Dorrer *et al*., "10-Bit, 3 mW Continuous-Time Sigma-Delta ADC for UMTS in a 0.12 μm CMOS process," *Proc. of the European Solid-State Circuits Conf.*, September 2003.
[214] M. Schimper *et al*., "A 3 mW Continuous-Time ΣΔ Modulator for EDGE/GSM with High Adjacent Channel Tolerance," *Proc. of the European Solid-State Circuits Conf.*, pp. 183–186, September 2004.
[215] M. Ranjbar *et al*., "A Low-Power 1.92MHz CT ΔΣ Modulator With 5-bit Successive Approximation Quantizer," *Proc. of the IEEE Custom Integrated Circuits Conf.*, pp. 5–8, September 2009.
[216] M. Ranjbar *et al*., "A 3.1 mW Continuous-Time ΔΣ Modulator With 5-Bit Successive Approximation Quantizer for WCDMA," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 1479–1491, August 2010.
[217] J.-G. Jo *et al*., "A 20MHz Bandwidth Continuous-Time ΣΔ Modulator with Jitter Immunity Improved Full-Clock Period SCR (FSCR) DAC and High Speed DWA," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, 2010.
[218] P. Fontaine, A. N. Mohieldin, and A. Bellaouar, "A Low-Noise Low-Voltage CT ΔΣ Modulator with Digital Compensation of Excess Loop Delay," *IEEE ISSCC Digest of Technical Papers*, pp. 498–499, 2005.
[219] Y. Aiba *et al*., "A Fifth-Order Gm-C Continuous-Time ΔΣ Modulator With Process-Insensitive Input Linear Range," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 2381–2391, September 2009.
[220] L. Dorrer *et al*., "A Continuous-Time ΔΣ ADC for Voice Coding with 92 dB DR in 45 nm CMOS," *IEEE ISSCC Digest of Technical Papers*, pp. 502–503, February 2008.
[221] S. Patón, A. di Giandomenico, L. Hernández, A. Wiesbauer, T. Poetscher, and M. Clara, "A 70-mW 300-MHz CMOS Continuous-Time ΣΔ ADC With 15-MHz Bandwidth and 11 Bits of Resolution," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 1056–1063, July 2004.
[222] A. D. Giandomenico *et al*., "A 15MHz Bandwidth Sigma-Delta ADC with 11 Bits of Resolution in 0.13μm CMOS," *Proc. of the IEEE European Solid-State Circuits Conf.*, September 2003.
[223] J.-G. Jo *et al*., "A 20-MHz Bandwidth Continuous-Time Sigma-Delta Modulator With Jitter Immunity Improved Full Clock Period SCR (FSCR) DAC and High-Speed DWA," *IEEE Journal of Solid-State Circuits*, vol. **46**, pp. 2469–2477, November 2011.

[224] S. Yan and E. Sánchez-Sinencio, "A Continuous-Time ΣΔ Modulator With 88-dB Dynamic Range and 1.1-MHz Signal Bandwidth," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 75–86, January 2004.
[225] J. Arias *et al*., "A 32-mW 320-MHz Continuous-Time Complex Delta-Sigma ADC for Multi-Mode Wireless-LAN Receivers," *IEEE Journal of Solid-State Circuits*, vol. **41**, pp. 339–351, February 2006.
[226] T. C. Caldwell and D. A. Johns, "A Time-Interleaved Continuous-Time ΔΣ Modulator with 20MHz Signal Bandwidth," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 447–450, September 2005.
[227] L. Breems, R. Rutten, R. van Veldhoven, and G. van der Weide, "A 56 mW Continuous-Time Quadrature Cascaded ΣΔ Modulator With 77 dB DR in a Near Zero-IF 20 MHz Band," *IEEE Journal of Solid-State Circuits*, vol. **42**, pp. 2696–2705, December 2007.
[228] C. C. Lee and M. P. Flynn, "A 14 b 23MS/s 48 mW Resetting ΣΔ ADC with 87 dB SFDR 11.7 b ENOB & 0.5 mm^2 Area," *IEEE Symp. VLSI Circuits. Digest of Technical Papers*, pp. 182–183, 2008.
[229] J. Kamiishi *et al*., "A Self-Calibrated 2-1-1 Cascaded Continuous-Time ΔΣ Modulator," *Proc. of the IEEE Custom Integrated Circuits Conf.*, pp. 9–12, September 2009.
[230] Y.-S. Shu, J. Kamiishi, K. Tomioka, K. Hamashita, and B.-S. Song, "LMS-Based Noise Leakage Calibration of Cascaded Continuous-Time ΣΔ Modulators," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 368–379, February 2010.
[231] L. Breems, R. Rutten, and G. Wetzker, "A Cascaded Continuous-Time ΣΔ Modulator with 67-dB Dynamic Range in 10-MHz Bandwidth," *IEEE Journal of Solid-State Circuits*, vol. **39**, pp. 2152–2160, December 2004.
[232] J. Harrison *et al*., "An LC Bandpass ΔΣ ADC with 70 dB SNDR Over 20MHz Bandwidth Using CMOS DACs," *IEEE ISSCC Digest of Technical Papers*, pp. 146–147, February 2012.
[233] S. B. Kim *et al*., "A 2.7 mW, 90.3 dB DR Continuous-Time Quadrature Bandpass Sigma-Delta Modulator for GSM/EDGE Low-IF Receiver in 0.25 μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 891–900, March 2009.
[234] J. Ryckaert, J. Borremans, B. Verbruggen, L. Bos, C. Armiento, J. Craninckx, and G. van der Plas, "A 2.4 GHz Low-Power Sixth-Order RF Bandpass ΔΣ Converter in CMOS," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 2873–2880, November 2009.
[235] B. D. Vuyst and P. Rombouts, "A 5-MHz 11-bit Delay-Based Self-Oscillating ΣΔ Modulator in 0.025 mm^2," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2010.
[236] H. Chae *et al*., "A 12 mW Low-Power Continuous-Time Bandpass ΔΣ Modulator with 58 dB SNDR and 24MHz Bandwidth at 200MHz IF," *IEEE ISSCC Digest of Technical Papers*, pp. 148–149, February 2012.
[237] R. H. M. Veldhoven, "A Triple-Mode Continuous-Time ΣΔ Modulator With Switched-Capacitor Feedback DAC for a GSM-EDGE/CDMA2000/UMTS Receiver," *IEEE Journal of Solid-State Circuits*, vol. **38**, pp. 2069–2076, December 2003.
[238] M. S. Kappes, "A 2.2-mW CMOS Bandpass Continuous-Time Multibit $\Delta - \Sigma$ ADC With 68 dB of Dynamic Range and 1-MHz Bandwidth for Wireless Applications," *IEEE Journal of Solid-State Circuits*, vol. **38**, pp. 1098–1104, July 2003.
[239] H. Shibata *et al*., "A DC-to-1GHz Tunable RF ΔΣ ADC Achieving DR=74 dB and BW=150MHz at f_0 =450MHz Using 550 mW," *IEEE ISSCC Digest of Technical Papers*, pp. 150–151, February 2012.
[240] F. Esfahani *et al*., "An 82 dB CMOS Continuous-Time Complex Bandpass Sigma-Delta ADC for GSM/EDGE," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 1049–1052, May 2003.
[241] R. Schreier *et al*., "A 375-mW Quadrature Bandpass ΔΣ ADC With 8.5-MHz BW and 90-dB DR at 44 MHz," *IEEE Journal of Solid-State Circuits*, vol. **41**, pp. 2632–2640, December 2006.
[242] A. Ashry *et al*., "A 3.6GS/s, 15 mW, 50 dB SNDR, 28MHz Bandwidth RF ΣΔ ADC with FoM of 1 pJ/bit in 130 nm CMOS," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2011.
[243] P. M. Chopp and A. A. Hamoui, "A 1V 13 mW Frequency-Translating ΔΣ ADC with 55 dB SNDR for a 4MHz Band at 225MHz," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2011.
[244] E. J. van der Zwan *et al*., "A 10.7-MHz IF-to-Baseband ΣΔ A/D Conversion System for AM/FM Radio Receivers," *IEEE Journal of Solid-State Circuits*, vol. **35**, pp. 1810–1819, December 2000.
[245] F. Chen and B. Leung, "A 0.25-mW Low-Pass Passive Sigma-Delta Modulator with Built-In Mixer for a 10-MHz IF Input," *IEEE Journal of Solid-State Circuits*, vol. **32**, pp. 774–782, June 1997.
[246] J. Rychaert *et al*., "A 6.1GS/s 52.8 mW 43 dB DR 80MHz Bandwdith 2.4GHz RF Bandpass ΔΣ ADC in 40 nm CMOS," *Proc. of the IEEE Radio Frequency Integrated Circuits Symp.*, pp. 443–446, June 2010.

[247] C.-Y. Lu *et al*., "A Sixth-Order 200MHz IF Bandpass Sigma-Delta Modulator With Over 68 dB SNDR in 10MHz Bandwidth," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 1122–1136, June 2010.
[248] P. G. R. Silva *et al*., "An IF-to-Baseband ΣΔ Modulator for AM/FM/IBOC Radio Receivers With a 118 dB Dynamic Range," *IEEE Journal of Solid-State Circuits*, vol. **42**, pp. 1076–1089, May 2007.
[249] S.-B. Kim *et al*., "Continuous-Time Quadrature Bandpass Sigma-Delta Modulator for GPS/Galileo Low-IF Receiver," *Proc. of the IEEE Radio Frequency Integrated Circuits Symp.*, pp. 127–130, June 2007.
[250] K. Koli *et al*., "A 900-MHz Direct Delta-Sigma Receiver in 65-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 2807–2818, December 2010.
[251] E. Martens *et al*., "RF-to-Baseband Digitization in 40 nm CMOS With RF Bandpass ΔΣ Modulator and Polyphase Decimation Filter," *IEEE Journal of Solid-State Circuits*, vol. **47**, pp. 990–1002, April 2012.
[252] R. Yu and Y. P. Xu, "Bandpass Sigma-Delta Modulator Employing SAW Resonator as Loop Filter," *IEEE Transactions on Circuits and Systems I—Regular Papers*, vol. **54**, pp. 723–735, April 2007.
[253] S. Gupta *et al*., "A QLL-Timed Direct-RF Sampling Band-Pass ΣΔ ADC with a 1.2 GHz Tuning Range in 0.13μm CMOS," *Proc. of the IEEE Radio Frequency Integrated Circuits Symp.*, June 2011.
[254] J. V. Engelen and R. van de Plassche, *BandPass Sigma-Delta Modulators: Stability Analysis, Performance and Design Aspects*, Kluwer Academic Publishers, 1999.
[255] N. Beilleau, H. Aboushady, F. Montaudon, and A. Cathelin, "A 1.3V 26 mW 3.2GS/s Undersampled LC Bandpass ΣΔ ADC for a SDR ISM-band Receiver in 130 nm CMOS," *Proc. of the IEEE Radio Frequency Integrated Circuits Symp.*, 2009.
[256] I. Hsu and H. C. Luong, "A 70-MHz Continuous-Time CMOS Band-pass ΣΔ Modulator for GPS Receivers," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 750–753, May 2000.
[257] H. Tao and J. M. Khoury, "A 400-MS/s Frequency Translating Bandpass Sigma-Delta Modulator," *IEEE Journal of Solid-State Circuits*, vol. **34**, pp. 1741–1752, December 1999.
[258] K. Thomas *et al*., "A 1GHz CMOS Fourth-Order Continuous-Time Bandpass Sigma-Delta Modulator for RF Receiver Front-End A/D Conversion," *Proc. of the IEEE Asia and South Pacific Design Automation Conf.*, pp. 665–670, 2005.
[259] M. Park and M. H. Perrott, "A 78 dB SNDR 87 mW 20 MHz Bandwidth Continuous-Time ΣΔ ADC With VCO-Based Integrator and Quantizer Implemented in 0.13μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 3344–3358, December 2009.
[260] K. Reddy and B. Haroun, "A 16 mW 78 dB-SNDR 10MHz-BW CT-ΔΣ ADC Using Residue-Cancelling VCO-Based Quantizer," *IEEE ISSCC Digest of Technical Papers*, pp. 152–153, February 2012.
[261] S. Z. Asl *et al*., "A 77 dB SNDR, 4MHz MASH ΔΣ Modulator with a Second-Stage Multi-rate VCO-Based Quantizer," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2011.
[262] C.-Y. Lu *et al*., "A 25MHz Bandwidth 5th-Order Continuous-Time Low-Pass Sigma-Delta Modulator With 67.7 dB SNDR Using Time-Domain Quantization and Feedback," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 1795–1808, September 2010.
[263] E. Prefasi *et al*., "A 7 mW 20MHz BW Time-Encoding Oversampling Converter Implemented in a 0.08 mm^2 65 nm CMOS Circuit," *IEEE Journal of Solid-State Circuits*, vol. **46**, pp. 1562–1574, July 2011.
[264] G. Taylor and I. Galton, "A Mostly Digital Variable-Rate Continuous-Time ADC ΔΣ Modulator," *IEEE ISSCC Digest of Technical Papers*, pp. 298–299, February 2010.
[265] M. Z. Straayer *et al*., "A 12-Bit, 10-MHz Bandwidth, Continuous-Time ΣΔ ADC With a 5-Bit, 950-MS/s VCO-Based Quantizer," *IEEE Journal of Solid-State Circuits*, vol. **43**, pp. 805–814, April 2008.
[266] V. Dhanasekaran *et al*., "A Continuous-Time Multi-bit ΔΣ ADC Using Time Domain Quantizer and Feedback Element," *IEEE Journal of Solid-State Circuits*, vol. **46**, pp. 639–650, March 2011.
[267] Y. Cao *et al*., "A 0.7 mW 13 b Temperature-Stable MASH ΔΣ TDC with Delay-Line Assisted Calibration," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, pp. 361–364, 2011.
[268] B. Young *et al*., "A 2.4 ps Resolution 2.1 mW Second-Order Noise-Shaped Time-to-Digital Converter with 3.2 ns Range in 1MHz Bandwidth," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2009.
[269] T.-Y. Lo, "A 102 dB Dynamic Range Audio Sigma-Delta Modulator in 40 nm CMOS," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, pp. 257–260, 2011.
[270] O. Rajaee *et al*., "A 79 dB 80MHz 8X-OSR Hybrid Delta-Sigma/Pipeline ADC," *IEEE Symp. VLSI Circuits. Digest of Technical Papers*, pp. 74–77, 2009.
[271] C.-Y. Ho *et al*., "A 75.1 dB SNDR, 80.2 dB DR, 4th-Order Feed-Forward Continuous-Time Sigma-Delta Modulator with Hybrid Integrator for Silicon TV-tuner Applications," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, pp. 261–264, 2011.

[272] C. C. Lee and M. P. Flynn, "A 14 b 23MS/s 48 mW Resetting ΣΔ ADC," *IEEE Transactions on Circuits and Systems I—Regular Papers*, vol. **58**, pp. 1167–1177, June 2011.
[273] O. Rajaee *et al*., "Low-OSR Over-Ranging Hybrid ADC Incorporating Noise-Shaped Two-Step Quantizer," *IEEE Journal of Solid-State Circuits*, vol. **46**, pp. 2458–2468, November 2011.
[274] G. K. Balachandran *et al*., "A 1.16 mW 69 dB SNR (1.2MHz BW) Continuous-Time ΣΔ ADC with Immunity to Clock Jitter," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2009.
[275] T. Song *et al*., "A 2.7-mW 2-MHz Continuous-Time ΣΔM With a Hybrid Active-Passive Loop Filter," *IEEE Journal of Solid-State Circuits*, vol. **43**, pp. 330–341, February 2008.
[276] R. H. M. Veldhoven *et al*., "An Inverted-Based Hybrid ΣΔ Modulator," *IEEE ISSCC Digest of Technical Papers*, vol. 492-493–„ February 2008.
[277] J. H. Shim, I.-C. Park, and B. Kim, "A Third-Order ΣΔ Modulator in 0.18μm CMOS with Calibrated Mixed-Mode Integrators," *IEEE Journal of Solid-State Circuits*, vol. **40**, pp. 918–925, April 2005.
[278] Y. Kim *et al*., "An 11 mW 100MHz 16X-OSR 64 dB-SNDR Hybrid CT/DT ΔΣ ADC with Relaxed DEM Timing," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2009.
[279] S. Kulchycki *et al*., "A 77-dB Dynamic Range, 7.5-MHz Hybrid Continuous-Time/Discrete-Time Cascade ΣΔ Modulator," *IEEE Journal of Solid-State Circuits*, vol. **43**, pp. 796–804, April 2008.
[280] A. Bosi *et al*., "An 80MHz 4 x Oversampled Cascaded ΔΣ-Pipelined ADC with 75 dB DR and 87 dB SFDR," *IEEE ISSCC Digest of Technical Papers*, vol. **54–55**, February 2005.
[281] K. Nguyen *et al*., "A 106 dB SNR Hybrid Oversampling ADC for Digital Audio," *IEEE ISSCC Digest of Technical Papers*, pp. 176–177, February 2005.
[282] M. Choi *et al*., "A 101-dB SNR Hybrid Delta-Sigma Audio ADC Using Post Integration Time Control," *Proc. of the IEEE Custom Integrated Circuits Conf.*, pp. 89–92, 2008.
[283] P. Morrow *et al*., "A 0.18μm 102 dB-SNR Mixed CT SC Audio-band ΔΣ ADC," *IEEE ISSCC Digest of Technical Papers*, pp. 177–178, February 2005.
[284] R. Winoto and B. Nikolic, "A Highly Reconfigurable 400-1700MHz Receiver Using a Down-Converting Sigma-Delta A/D with 59-dB SNR and 57-dB SFDR over 4-MHz Bandwidth," *Proc. of the IEEE Symp. on VLSI Circuits*, pp. 142–143, 2009.
[285] O. Bajdechi *et al*., "A 1.8-V ΔΣ Modulator Interface for an Electret Microphone With On-Chip Reference," *IEEE Journal of Solid-State Circuits*, vol. **37**, pp. 279–285, March 2002.
[286] T. L. Brooks *et al*., "A Cascaded Sigma-Delta Pipeline A/D Converter with 1.25MHz Signal Bandwidth and 89 dB SNR," *IEEE Journal of Solid-State Circuits*, vol. **32**, pp. 1896–1906, December 1997.
[287] T. Christen, T. Burger, and Q. Huang, "A 0.13μm CMOS EDGE/UMTS/WLAN Tri-Mode ΣΔ ADC with -92 dB THD," *IEEE ISSCC Digest of Technical Papers*, pp. 240–241, February 2007.
[288] A. Morgado, R. del Río, and J. M. de la Rosa, "A New Cascade ΣΔ Modulator for Low-Voltage Wideband Applications," *IET Electronics Letters*, vol. **43**, pp. 910–911, August 2007.
[289] T. Christen and Q. Huang, "A 0.13 μm CMOS 0.1-20 MHz Bandwidth 86-70 dB DR Multi-Mode DT ΔΣ ADC for IMT-Advanced," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 414–417, September 2010.
[290] L. Bos *et al*., "Multirate Cascaded Discrete-Time Low-Pass ΔΣ Modulator for GSM/Bluetooth/UMTS," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 1198–1208, June 2010.
[291] A. Morgado, R. del Río, J. M. de la Rosa, L. Bos, J. Ryckaert, and G. van der Plas, "A 100 kHz-10MHz BW, 78-to-52 dB DR,4.6-to-11 mW Flexible SC ΣΔ Modulator in 1.2-V 90-nm CMOS," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 418–421, September 2010.
[292] R. van Veldhoven, "A Triple-Mode Continuous-Time ΣΔ Modulator with Switched-Capacitor Feedback DAC for a GSM-EDGE/CDMA2000/UMTS Receiver," *IEEE Journal of Solid-State Circuits*, vol. **38**, pp. 2069–2076, December 2003.
[293] S. Ouzounov *et al*., "A 1.2V 121-Mode CT Delta-Sigma Modulator for Wireless Receivers in 90 nm CMOS," *IEEE ISSCC Digest of Technical Papers*, pp. 242–243, February 2007.
[294] B. Putter, "A 5th-Order CT/DT Multi-Mode ΣΔ Modulator," *IEEE ISSCC Digest of Technical Papers*, pp. 244–245, February 2007.
[295] P. Crombez *et al*., "A Single-Bit 500 kHz-10 MHz Multimode Power-Performance Scalable 83-to-67 dB DR CT ΔΣ Modulator for SDR in 90 nm Digital CMOS," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 1159–1171, June 2010.
[296] A. Tabatabaei and A. Wooley, "A Mostly-Digital Variable-Rate Continuous-Time Delta-Sigma Modulator ADC," *IEEE Journal of Solid-State Circuits*, vol. **45**, pp. 2634–2646, December 2010.

[297] J. Sauerbrey *et al*., "A Configurable Cascaded Continuous-Time ΔΣ Modulator with up to 15MHz Bandwidth," *Proc. of the IEEE European Solid-State Circuits Conf.*, pp. 426–429, September 2010.

[298] Y. Ke *et al*., "A 2.8-to-8.5 mW GSM/Bluetooth/UMTS/DVB-H/WLAN Fully Reconfigurable CT ΔΣ with 200 kHz to 20MHz BW for 4G Radios in 90 nm Digital CMOS," *IEEE Symp. VLSI Circuits. Digest of Technical Papers*, pp. 153–154, 2010.

[299] V. Singh *et al*., "A 16MHz BW 75 dB DR CT ΔΣ ADC Compensated for More than One Cycle Excess Loop Delay," *Proc. of the IEEE Custom Integrated Circuits Conf.*, September 2011.

[300] C.-Y. Ho *et al*., "A Quadrature Bandpass Continuous-Time Delta-Sigma Modulator for a Tri-Mode GSM-EDGE/UMTS/DVB-T Receiver," *IEEE Journal of Solid-State Circuits*, vol. **46**, pp. 2571–2582, November 2011.

[301] F. Goodenough, "Analog Techniques of all Varieties Dominate ISSCC," *IEEE Journal of Solid-State Circuits*, vol. **44**, pp. 96–111, February 1996.

[302] M. Ortmanns and F. Gerfers, *Continuous-Time Sigma-Delta A/D Conversion: Fundamentals, Performance Limits and Robust Implementations*, Springer, 2006.

[303] A. Rodríguez-Vázquez, F. Medeiro, and E. Janssens, *CMOS Telecom Data Converters*, Kluwer Academic Publishers, 2003.

[304] P. Benabes, A. Gauthier, and R. Kielbasa, "New High-Order Universal ΣΔ Modulator," *IET Electronics Letters*, vol. **31**, pp. 1575–1577, January 1995.

[305] N. Maghari *et al*., "Sturdy MASH ΔΣ Modulator," *IET Electronics Letters*, vol. **42**, pp. 1269–1270, October 2006.

[306] N. Maghari *et al*., "Multi-Loop Efficient Sturdy MASH Delta-Sigma Modulators," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 1216–1219, May 2008.

[307] P. Benabes, A. Gauthier, and D. Billet, "New Wideband Sigma-Delta Convertor," *IET Electronics Letters*, vol. **27**, pp. 1575–1577, August 1993.

[308] J. Silva, U. Moon, J. Steensgaard, and G. C. Temes, "Wideband Low-Distortion Delta-Sigma ADC Topology," *IET Electronics Letters*, vol. **37**, pp. 737–738, June 2001.

[309] H. Kwan *et al*., "Design of Hybrid Continuous-Time Discrete-Time Delta-Sigma Modulators," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 1224–1227, May 2008.

[310] J. M. de la Rosa, A. Morgado, and R. del Río, "Hybrid Continuous-Time/Discrete-Time Cascade ΣΔ Modulators with Programmable Resonation," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 2249–2252, May 2009.

[311] F. Colodro and A. Torralba, "Multirate ΣΔ Modulators," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. **49**, pp. 170–176, March 2002.

[312] M. Ortmanns *et al*., "Multirate Cascaded Continuous-Time ΣΔ Modulators," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 4225–4228, May 2002.

[313] J. G. García-Sánchez and J. M. de la Rosa, "Multirate Downsampling Hybrid CT/DT Cascade Sigma-Delta Modulators," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. **59**, pp. 285–294, February 2012.

[314] M. Maghami and M. Yavari, "Multirate Double-Sampling Hybrid CT/DT Sigma-Delta Modulators for Wideband Applications," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 2253–2256, May 2009.

[315] P. Benabes, "Accurate Time-Domain Simulation of Continuous-Time Sigma-Delta Modulators," *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. **56**, pp. 2248–2258, October 2009.

[316] S. Pavan, "Systematic Design Centering of Continuous Time Oversampling Converters," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. **57**, pp. 158–162, March 2010.

[317] L. Hernández and E. Prefasi, "Analog-to-Digital Conversion Using Noise Shaping and Time Encoding," *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. **55**, pp. 2026–2037, August 2008.

[318] L. Hernández and A. Wiesbauer, "Exploiting Time Resolution in Nanometre CMOS Data Converters," *Proc. of the IEEE Intl. Symp. on Circuits and Systems*, pp. 1069–1072, May 2010.

[319] B. Nauta, "A CMOS Transconductance-C Filter Technique for Very High Frequencies," *IEEE Journal of Solid-State Circuits*, vol. **27**, pp. 142–153, February 1992.

[320] F. Houfaf *et al*., "A 65 nm CMOS 1-to-GHz Tunable Continuous-Time Low-pass Filter for High-Data-Rate Communications," *IEEE ISSCC Digest of Technical Papers*, pp. 362–363, February 2012.

[321] A. Morgado, R. del Río, and J. M. de la Rosa, *Nanometer CMOS Sigma-Delta Modulators for Software Defined Radio*, Springer, 2011.

[322] A. Morgado *et al*., "High-Efficiency Cascade ΣΔ Modulators for the Next Generation Software-Defined-Radio Mobile Systems," *IEEE Transactions on Instrumentation and Measurement*, vol. **61**, pp. 2860–2869, November 2012.

[323] S. Gupta *et al*., "A 0.8-2GHz Fully-Integrated QPLL-Timed Direct-RF-Sampling Bandpass ΣΔ ADC in 0.13μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. **47**, pp. 1141–1153, May 2012.

[324] J. Mitola, "The Software Radio Architecture," *IEEE Communications Magazine*, pp. 26–38, May 1995.

[325] A. Naderi, M. Sawan, and Y. Savaria, "On the Design of Undersampling Continuous-Time Bandpass Delta-Sigma Modulators for Gigahertz Frequency A/D Conversion," *IEEE Transactions on Circuits and Systems—I: Regular Papers*, vol. **55**, pp. 3488–3499, December 2008.

[326] K. Yamamoto, A. C. Carusone, and F. P. Dawson, "A Delta-Sigma Modulator With a Widely Programmable Center Frequency and 82-dB Peak SNDR," *IEEE Journal of Solid-State Circuits*, vol. **43**, pp. 1772–1782, August 2008.

A

SIMSIDES User Guide

SIMSIDES (SIMulink-based SIgma-DElta Simulator) is a time-domain behavioral simulator for ΣΔMs developed as a toolbox in the MATLAB/SIMULINK environment. SIMSIDES can be used for simulating any arbitrary ΣΔM architecture, implemented with both DT and CT circuit techniques. To this end, a complete list of ΣΔM building blocks (integrators, resonators, quantizers, embedded DACs, etc.) is included in the toolboox. The behavioral models of these building blocks take into account the most critical error mechanisms of different circuit techniques including SC, SI, and CT circuits. These models, validated through transistor-level electrical simulations and by experimental measurements taken from a number of silicon prototypes, have been incorporated into the SIMULINK environment as C-MEX S-functions. As detailed in Chapter 3, this approach drastically increases the computational efficiency in terms of CPU time and accuracy of the simulation results.

The behavioral models included in SIMSIDES have been compiled and tested in a number of operating systems, including Apple OS X, UNIX (Solaris), Linux, and Microsoft Windows. Both 32-bit and 64-bit system platforms have been successfully tested in the majority of them.

Although SIMSIDES was originally developed using MATLAB 6.5 and SIMULINK 5, the toolbox has been updated and successfully used in a number of MATLAB/SIMULINK versions in the last years, the last one being MATLAB 2010b. This appendix provides a user guide of SIMSIDES, giving an overview of the most significant features of the simulator.

A.1 Getting Started: Installing and Running SIMSIDES

A free copy of SIMSIDES can be downloaded from the following website:

`http://www.imse-cnm.csic.es/simsides`

After completing the online registration form and accepting the terms and conditions for using SIMSIDES, a zip file named `simsides.zip` is downloaded. The following steps must be followed to install the toolbox:

1. Uncompress the `simsides.zip` file to a directory on your computer hard disk. Let us assume that the directory is named `SIMSIDES`.

CMOS Sigma-Delta Converters: Practical Design Guide, First Edition. José M. de la Rosa and Rocío del Río.

2. Start MATLAB program.
3. Set the MATLAB search path in order to add the `SIMSIDES` directory. To do this, go to `File` menu in MATLAB and select `Set Path`. The `Set Path` dialogue box opens, listing all folders on the search path. From this dialogue box, click the button `Add with Subfolders` and select the `SIMSIDES` directory to add to the search path. In order to reuse the newly modified search path including SIMSIDES directory and subdirectories, click `Save`, and finally click `Close`. This procedure—illustrated in Figure A.1a—must be done only the first time SIMSIDES is installed in the hard disk.

In order to start SIMSIDES, type `simsides` at the MATLAB prompt and the SIMSIDES main window is displayed, as illustrated in Figure A.1b.

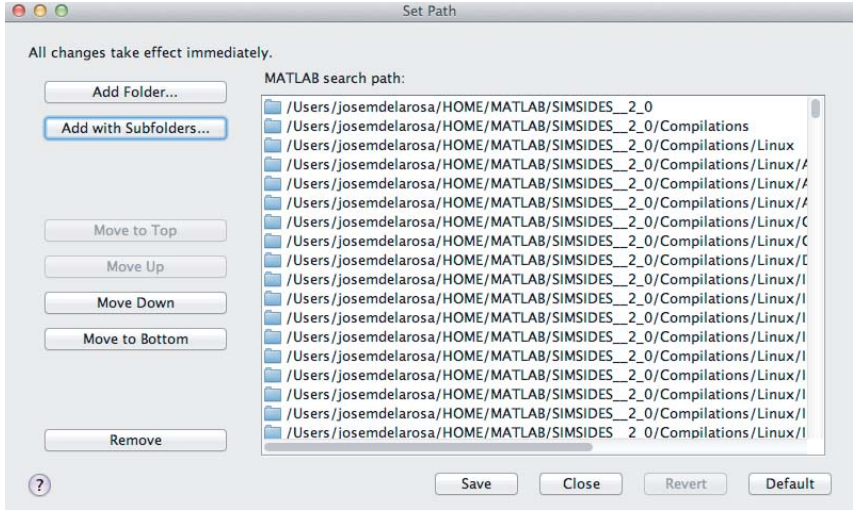
A.2 Building and Editing ΣΔM Architectures in SIMSIDES

To create a new ΣΔM architecture in SIMSIDES, select `File` and then `New Architecture` in the main menu, and a new SIMULINK model window is displayed. Alternatively, an existing ΣΔM architecture can be opened by selecting `File -> Open Architecture` as illustrated in Figure A.2.

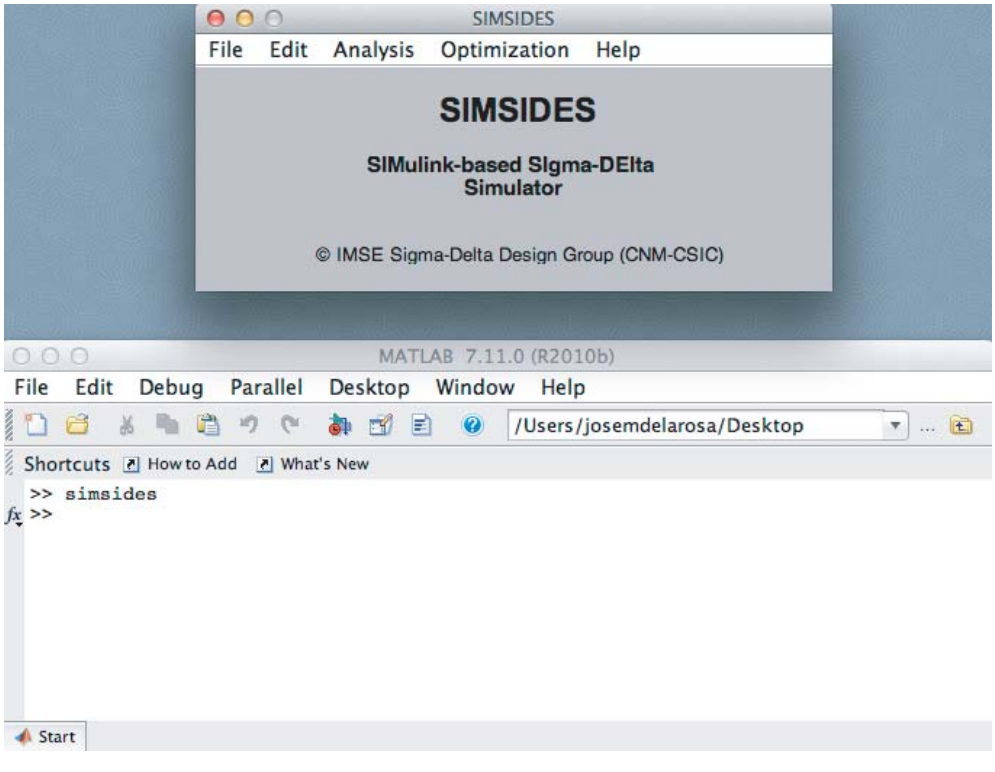
In order to define a ΣΔM block diagram in SIMSIDES, the required building blocks can be incorporated from the `Edit` menu as shown in Figure A.3. Both SIMULINK and SIMSIDES library models can be included by selecting `Edit -> SIMULINK Library` or `Edit -> Add block`, respectively. The latter option allows users to browse through all SIMSIDES library models. This way, clicking on `Edit -> Add block` a new window is displayed where the user can select either ideal or real building blocks, by choosing either `Add Ideal Block` or `Add Real Block` menus, respectively. In both cases, building-block models are organized in a set of sublibraries, namely integrators, quantizers and comparators, D/A converters, resonators, and auxiliary blocks. The latter are only available in real libraries.

Some model libraries are grouped in sublibraries that contain different models corresponding to different kinds of circuit implementations. For instance, if library `Real Integrators` is selected, a new window is displayed where the user can select the circuit technique (CT, SC, or SI) as well as the type of integrator (i.e., either FE or LD in the case of SC and SI integrators, and Gm-C, Gm-MC, active-RC, MOSFET-C in the case of CT integrators). As an illustration, Figure A.3 shows different sublibraries contained in the `Real Integrators` library. A complete list of model libraries and sublibraries, as well as a description of their models, is given in Appendix B.

Once the ΣΔM block diagram is completed and the different building-block model parameters have been defined in the MATLAB workspace, the modulator can be simulated in SIMULINK following the same procedure as for the simulation of an arbitrary model in SIMULINK; that is, choosing `Simulation -> Start` menu in the SIMULINK model window.

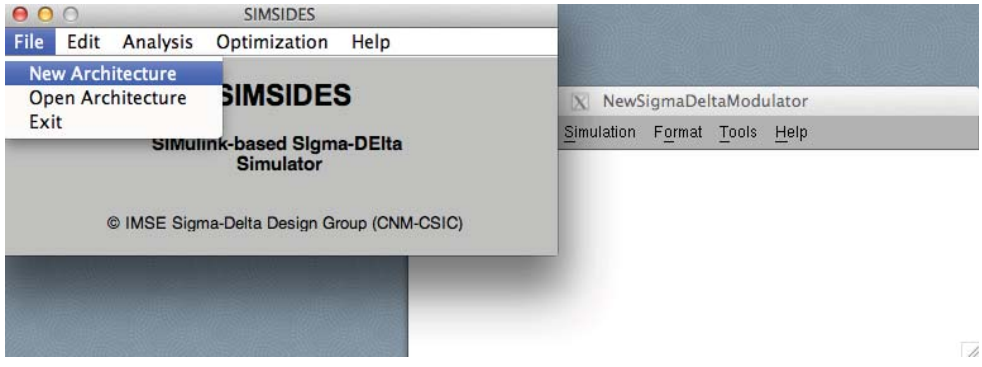


(a)

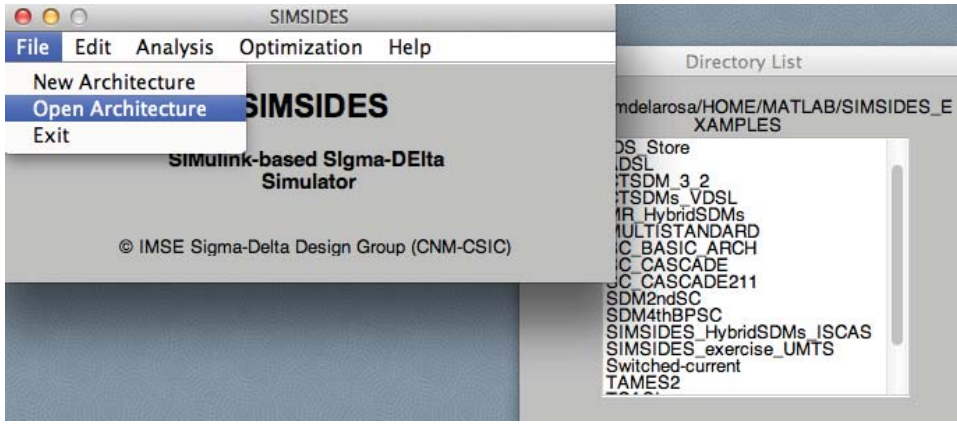


(b)

Figure A.1 Installing and starting SIMSIDES: (a) setting the MATLAB path and (b) starting SIMSIDES at the MATLAB prompt.



(a)



(b)

Figure A.2 Building and editing ΣΔMs in SIMSIDES: (a) creating a new ΣΔM architecture and (b) opening an existing model.

A.3 Analyzing ΣΔMs in SIMSIDES

Simulation output data can be post-processed in SIMULINK using the `Analysis` menu. As illustrated in Figure A.4, the `Analysis` menu includes the following submenus:

- `Node Spectrum Analysis`, that computes and plots the FFT magnitude spectrum of a given signal.
- `Integrated Power Noise`, used for calculating and graphically representing the IBN within a given signal bandwidth.
- `SNR/SNDR`, that computes the SNR and/or SNDR within the band of interest, considering both LP- and BP-ΣΔMs.
- `Harmonic Distortion`, that computes dynamic harmonic distortion figures, such as THD and intermodulation distortion figures.
- `Histogram`, used for representing histograms and analyzing the input/output swing in ΣΔM building blocks.

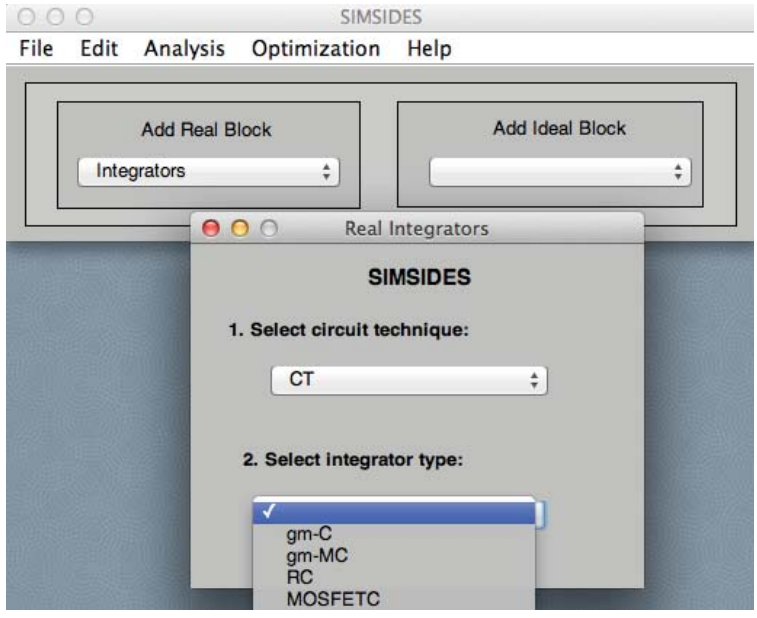


Figure A.3 Illustrating different sublibraries included in the Real Integrators library.

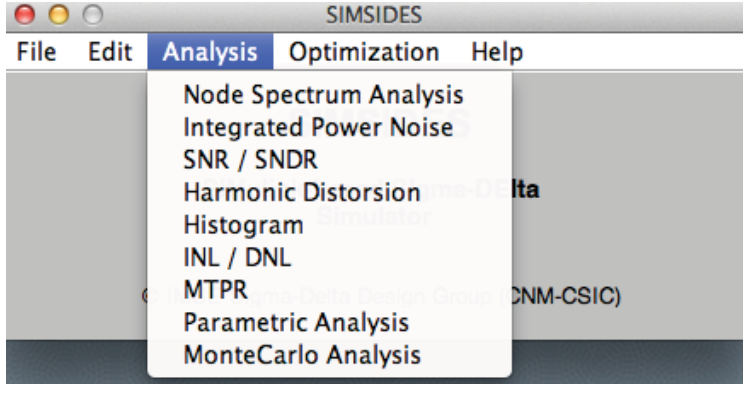


Figure A.4 Analysis menu in SIMSIDES.

- `INL/DNL`, that calculates static harmonic distortion.
- `MTPR`, used for computing multitone power ratio (MTPR).
- `Parametric Analysis`, that allows to simulate the impact of a given model parameter on the performance of ΣΔMs.
- `Monte Carlo Analysis`, to perform Monte Carlo simulations.

The required parameters and details involving the aforementioned analysis menus are described below.

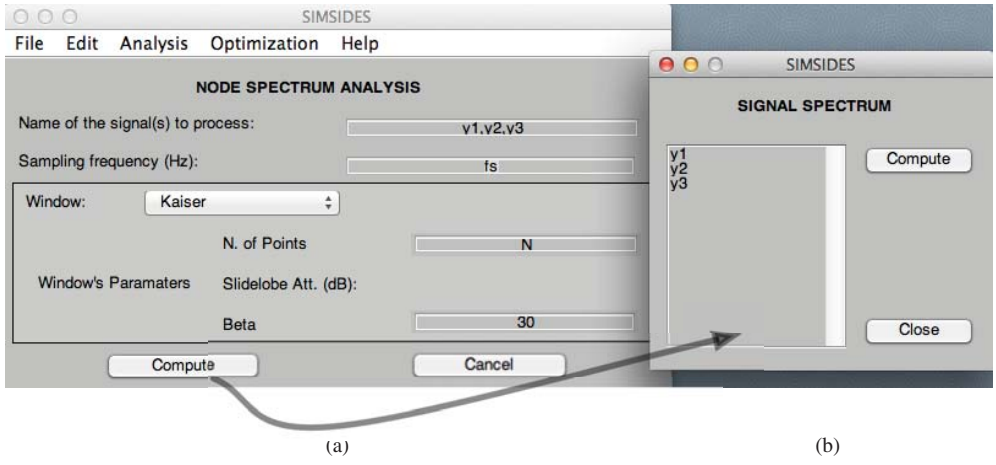


Figure A.5 Node spectrum analysis menu.

Node Spectrum Analysis

Figure A.5a shows the SIMSIDES `Node Spectrum Analysis` window. The following parameters are required to compute the FFT magnitude spectrum:

- `Name of the signal(s) to process`, where different variable names can be introduced, separated by commas. These variables can be output data generated in the simulations (for instance, the modulator output data stream), which have been previously saved in the MATLAB workspace by using the `To Workspace` SIMULINK block.
- `Sampling frequency`; that is, the sampling frequency in hertz.
- `Window`, that defines the window function used for computing the FFT. The main window functions available in MATLAB can be selected, namely Kaiser, Barlett, Blackman, Hamming, Hanning, Chebyshev, Boxcar, and Triangular.
- `Number of Points`; that is, the number of points (`N` in Figure A.5) for the selected window function and for FFT computation.
- `Window Parameters`, where other parameters required to define the window function are defined (such as `Beta` parameter used in Kaiser windows).

Once these parameters have been defined, the output spectrum can be computed by clicking on the `Compute` button and then selecting the signal to be processed from the new window that is displayed (`Signal Spectrum` window shown in Figure A.5b).

Integrated Power Noise

Figure A.6 shows the SIMSIDES `Integrated Power Node` window used for computing the IBN of any arbitrary data sequence obtained from simulations. To compute IBN, the following parameters are required:

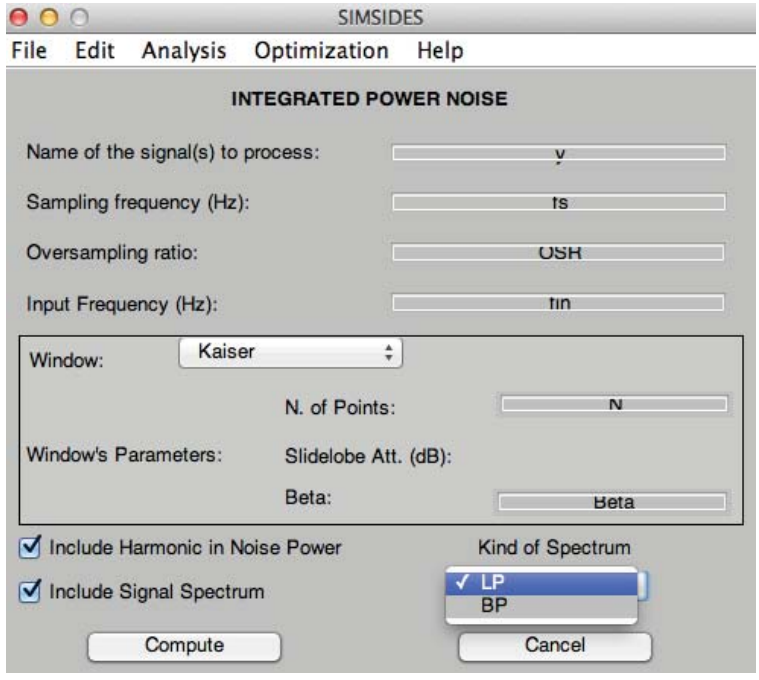


Figure A.6 Integrated power noise menu.

- `Name of the signal(s) to process`.
- `Sampling frequency`; that is, the sampling frequency in hertz.
- `Oversampling ratio`; that is, the value of OSR that defines the signal bandwidth in which the IBN is computed.
- `Input frequency`, where it is assumed that a single-tone input signal is applied.
- `Window Parameters`; that is, the parameters required to define the window function used for computing the IBN.
- `Kind of Spectrum`, that specifies the signal nature; that is, low-pass (LP) or band-pass (BP).

After defining all parameters described above, the IBN is computed by clicking on the `Compute` button. Harmonic distortion can be also taken into account in the calculation of the IBN by clicking the `Include Harmonic in Noise Power` button. The signal spectrum can be also plotted together with the IBN by choosing the `Include Signal Spectrum` option.

SNR/SNDR

Figure A.7 shows the SIMSIDES `SNR/SNDR` window. The parameters required to calculate the SNR/SNDR of a given signal are essentially the same as those used for computing IBN—described in the previous section. In this case, either the SNR or the SNDR is

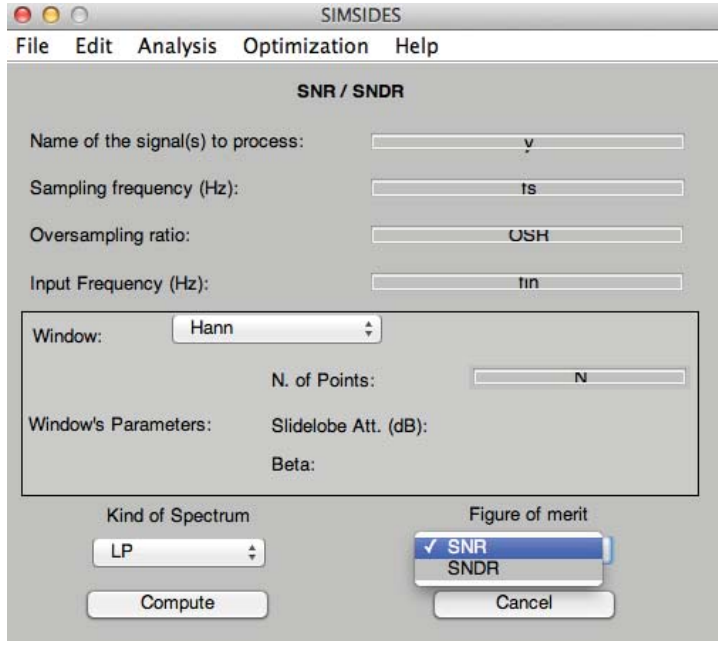


Figure A.7 SNR/SNDR menu.

computed depending on the `Figure of merit` selected. Note that this kind of analysis calculates the SNR/SNDR for a given value of the input signal amplitude. If an SNR-versus-amplitude curve is required, a parametric analysis should be chosen as will be described later.

Harmonic Distortion

Figure A.8 shows the SIMSIDES `Harmonic Distortion` window that is used for computing the harmonic distortion power. Two different figures of merit can be calculated, namely THD and third-order intermodulation distortion (`IM3` in Figure A.8). The latter requires using a two-tone input signal. For that reason, there is an additional parameter named `Input2 Frequency` that defines the frequency of the second input tone.

Integral and Differential Nonlinearity

The `INTEGRAL AND DIFFERENTIAL NON-LINEARITY` menu, illustrated in Figure A.9, is used for characterizing the static linearity in SIMSIDES. The analysis is based on either `Histograms` or `Input Ramp Waveform`—as selected by the user. Other parameters required to do this analysis are the `Input Amplitude` and the `Number of bits`, which specifies the ideal resolution of the A/D conversion, expressed in bits.

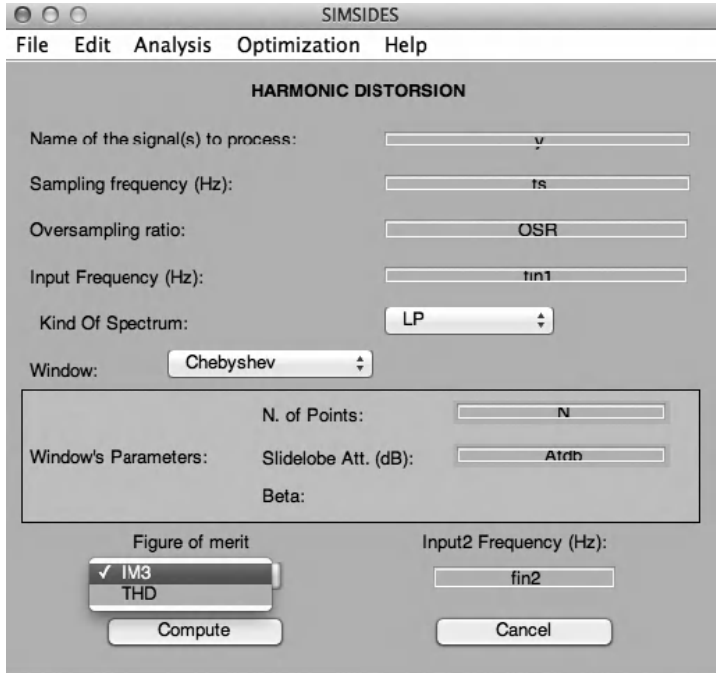


Figure A.8 Harmonic distortion analysis menu.

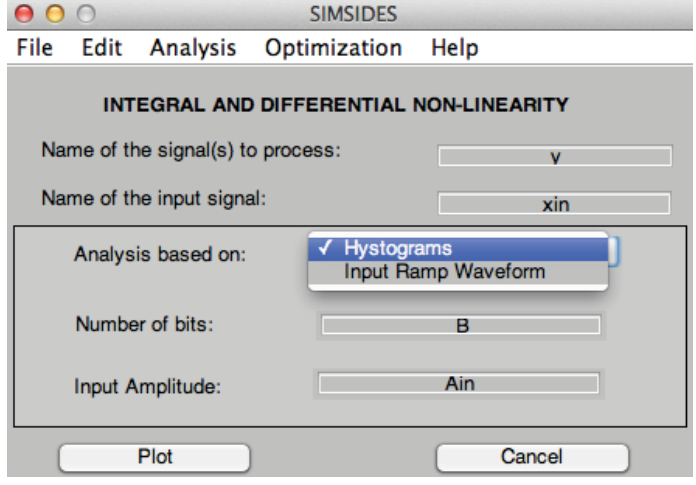


Figure A.9 Integral and differential nonlinearity analysis menu.

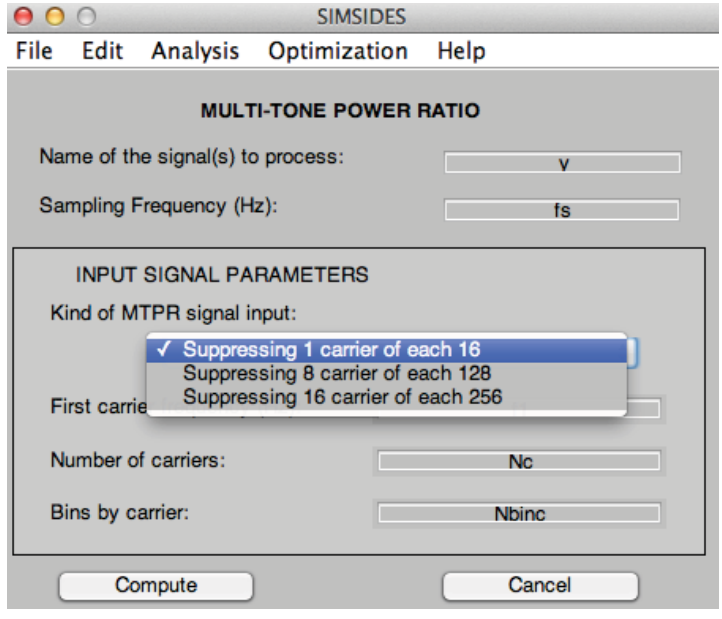


Figure A.10 Multitone power ratio analysis menu.

Multitone Power Ratio

SIMSIDES can also analyze the harmonic distortion in those telecom applications such as ADSL, where a discrete multitone (DMT) signal is used. In this case, the linearity of the system is measured by a figure named multitone power ratio (MTPR). The corresponding SIMSIDES menu—shown in Figure A.10—allows to compute MTPR for DMT input signals of different types:

- `Suppressing 1 carrier of each 16`; that is, 1 out of 16 carrier channels are suppressed.
- `Suppressing 8 carrier of each 128`; that is, 8 out of 128 carrier channels are suppressed.
- `Suppressing 16 carrier of each 256`; that is, 16 out of 256 carrier channels are suppressed.

In addition, the following parameters are also needed to compute MTPR:

- `Number of carriers`, that stands for the number of carrier channels in which the DMT signal is divided.
- `Bins by carrier`; that is, the number of bins assigned to each carrier channel in the FFT.

Histogram

Histograms of signals that have been previously saved on the MATLAB workspace can be computed using the `HISTOGRAM` menu (illustrated in Figure A.11), where the `Number`

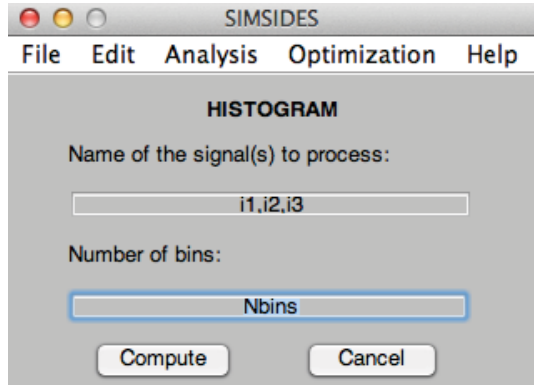


Figure A.11 Histogram analysis menu.

`of bins` specifies the number of intervals in which the signal range will be divided to compute the histogram.

Parametric Analysis

Figure A.12 shows the SIMSIDES `PARAMETRIC ANALYSIS` menu. This menu is used for analyzing the impact of varying a model parameter on the performance of ΣΔMs. Either one parameter or two parameters can be varied simultaneously by selecting the `Second Parameter` option. For each parameter, the following data must be specified:

- `Parameter Name`; that is, the name of the model parameter to be varied. This model parameter can be a variable used in a ΣΔM building-block model (such as for instance I_o, g_m, etc.) or a simulation parameter, such as the input signal amplitude, sampling frequency, and so on.

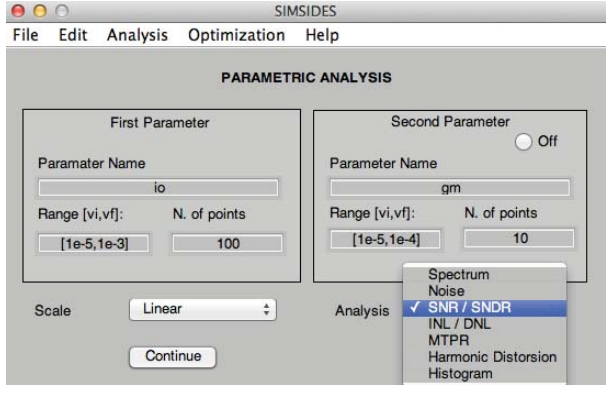


Figure A.12 Parametric analysis menu.

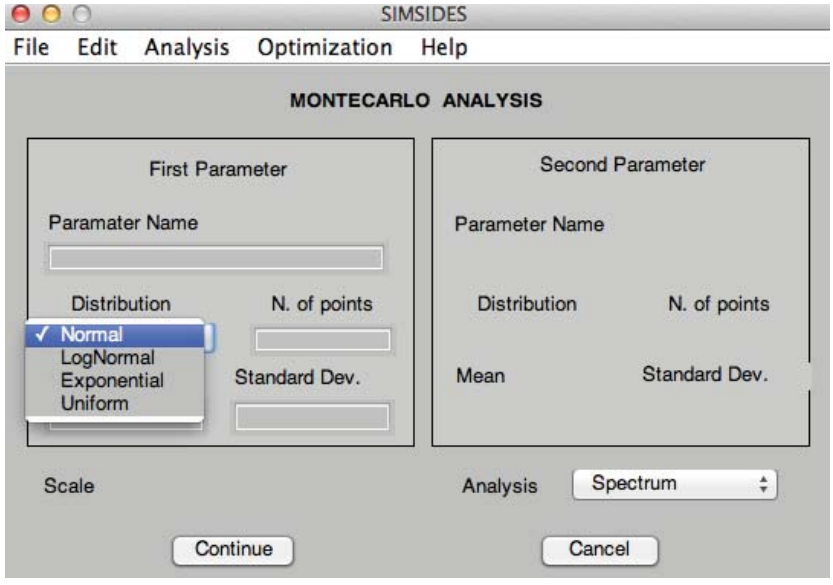


Figure A.13 Monte Carlo analysis menu.

- `Range [vi,vf]`, that defines the variation range, defined by an interval with a lower value given by `vi` and a upper value of `vf`.
- `N. of points`; that is, the number of points in which the variation interval is divided.
- `Scale`, that specifies if the variation range is either linear or logarithmic.
- `Analysis`, that specifies the type of analysis to be carried out, including output spectrum, IBN, SNR/SNDR, INL, MTPR, harmonic distortion, histograms, and so on.

Monte Carlo Analysis

Figure A.13 shows the SIMSIDES menu to run a Monte Carlo analysis. This is a particular case of parametric analysis, which has essentially the same functionalities and model parameters. The only difference is that the variation of the parameters involved in the Monte Carlo analysis are randomly varied according to a probability distribution with a mean value and a standard deviation that are specified in the analysis menu. Different types of probability distributions can be chosen, including `Normal`, `Log-Normal`, `Exponential`, and `Uniform` distributions.

A.4 Example

This section illustrates the use of SIMSIDES through a simple example in which several kinds of analyses will be carried out to show the main features of the toolbox. Figure A.14 shows the block diagram of the modulator under study, which consists of a third-order cascade 2-1 DT-$\Sigma\Delta$M with single-bit quantization in both stages.

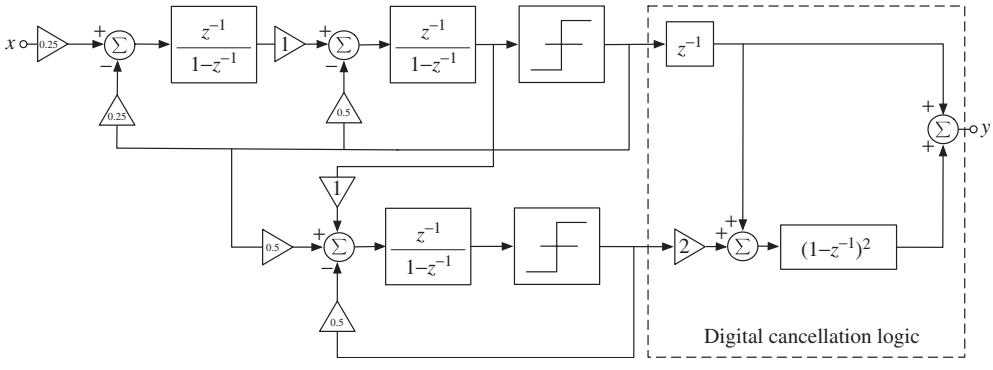


Figure A.14 Z-domain block diagram of a cascade 2-1 DT-ΣΔM.

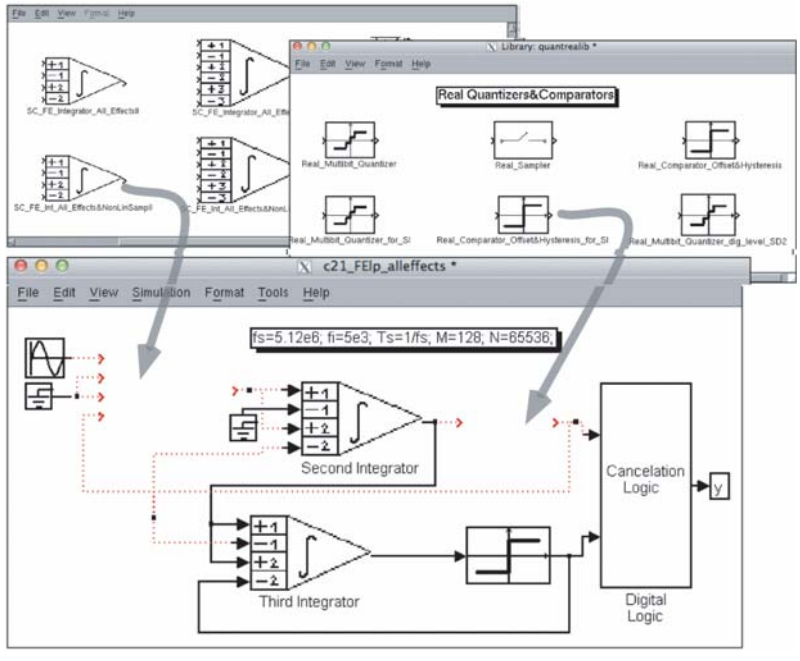
Creating the Cascade 2-1 ΣΔM Block Diagram in SIMSIDES

The modulator block diagram shown in Figure A.14 can be implemented by using the model libraries available in SIMSIDES. To this end, the same procedure as described in Section A.2 is followed:

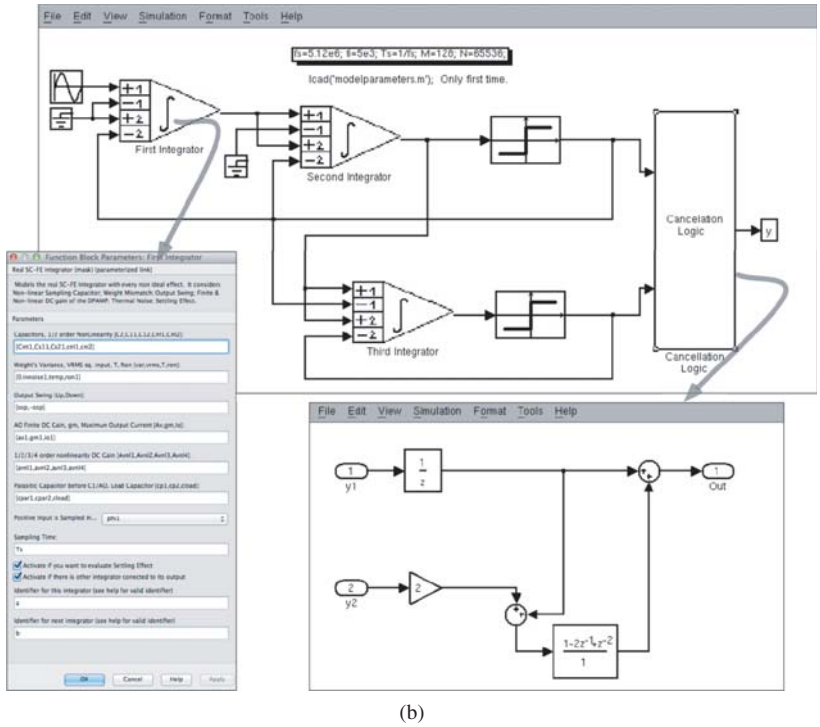
- Go to SIMSIDES main menu, select `File -> New Architecture` and introduce a name for the new ΣΔM architecture.
- Include the integrators and comparators from the SIMSIDES model libraries. To do this, select `Edit -> Add Block`. In this example, the FE integrators in Figure A.14 are implemented by using the `SC_FE_Integrator_All_Effects` blocks from the `Real Integrators` library, whereas single-bit quantizers are modeled by the `Real_Comparator_Offset&Hysteresis` comparator block available in `Quantizers&Comparators` library. These building blocks can be incorporated in the new architecture by simply dragging and dropping the models from their corresponding SIMSIDES libraries, as illustrated in Figure A.15a.
- Incorporate the remaining building blocks from the SIMULINK model library. To do this, go to `Edit -> Simulink Library` and drag the required models. In this example, the following blocks are required: `Sine Wave` and `Ground` blocks from `Sources` library, `Unit Delay` and `Discrete Filter` block from the `Discrete` library, and `To Workspace` from `Sinks` library.
- Finally, once all required blocks have been included in the new architecture, they are properly connected to implement the required ΣΔM architecture shown in Figure A.15b.

Setting Model Parameters

The modulator parameters and model parameters required to simulate the block diagram of Figure A.15 can be either set up in the MATLAB command window or they can be alternatively saved in an M-file that is loaded when needed. As an illustration, Figure A.16 shows the M-file used for setting up all model parameters of Figure A.15, that also includes



(a)



(b)

Figure A.15 SIMSIDES block diagram of the ΣΔM shown in Figure A.14: (a) building and editing the block diagram and (b) complete modulator block diagram in SIMSIDES.

```
% SDM parameters:
% Sampling Frequency(fs), Input Frequency (fi), Sampling Time (Ts)
% OverSampling Ratio (OSR=M); Number of points (N)
fs=5.12e6; fi=5e3; Ts=1/fs; M=128; N=65536;
% Model parameters
kt=0.026*1.6e-19; % Boltzmann constant

% First Integrator's parameters
Cint1=24e-12; % integration capacitor For gain=1
Cs11=6e-12; % sampling capacitor (branch 1)
Cs21=6e-12; % sampling capacitor (branch 2)
innoise1=0; % rms value of the input equivalent noise
ao1=2.63e3; % open-loop OTA DC gain
gm1=4.5e-3; % transconductance
io1=0.977e-3; % maximum OTA output current
ron1=60; % sampling switch-on resistance

% Second- and Third- Integrators
Cint2=3e-12;
Cs12=1.5e-12;
Cs22=1.5e-12;
innoise2=0;
ao2=1.38e3;
gm2=0.87e-3;
io2=0.25e-3;
ron2=650;

% Common integrator parameters
temp=175; % temperature
osp=2.7; % output swing
cn11=0; % capacitor first-order non-linear coef.
cn12=25e-6; % capacitor second-order non-linear coef.
avn11=0; % DC gain first-order non-linear coef.
avn12=15e-2; % DC gain second-order non-linear coef.
avn13=0; % DC gain third-order non-linear coef.
avn14=0; % DC gain fourth-order non-linear coef.
cpar1=0.6e-12; % parasitic (opamp) input capacitance
cpar2=0.6e-12;
cload=2.28e-12; % opamp (intrinsic) load capacitance
% Comparators
vref=2; % DAC reference voltage
hys=30e-3; % comparator hysteresis
```

Figure A.16 M-file including all model parameters required to simulate the ΣΔM in Figure A.15b.

a brief description of the different parameters and variables included. For the sake of completeness, Table A.1 includes the values of all building-block parameters as they are described in the SIMSIDES user masks, as well as other auxiliary block parameters (such as those used in `Sine Wave` and `To Workspace` blocks) that are required during simulation. In addition to these model parameters, simulation parameters must be set up to run a simulation. To do this, go to `Simulation -> Simulation Parameters` menu and define the following parameters:

- Simulation Time: `Start Time: 0.0`; `Stop Time: (N-1)*Ts`
- Solver options: `Type: Variable Step`; `Max Step Size: Auto`

Note that integrator building blocks are identified in order to properly compute the equivalent load capacitances required for the incomplete settling error model.

Table A.1 Building-block model parameters used for simulating the ΣΔM in Figure A.15b

Building block	Parameter description	Value/variable
Input sine wave	Sine type	Time based
	Amplitude	0.5
	Bias	0
	Frequency (rad s^{-1})	2*pi*fi
	Phase (rad)	0
	Sample time	0
	Interpret vector parameters	Selected
First Integrator	Integration and Sampling Capacitors (Branch 1, Branch 2)	[Cint1,Cs11,Cs21]
	Capacitor nonlinear coefficients	[cnl1,cnl2]
	Weight's variance, rms eq.input noise, temperature	[0,innoise1,temp]
	OTA DC gain, transconductance, max. output current	[ao1,gm1,io1]
	Positive/negative output swing	[osp,−osp]
	Switch on-resistance	ron1
	OTA DC gain nonlinear coefficients	[avnl1,2,3,4]
	Parasitic capacitances before the OTA	[cpar1,cpar2]
	Load capacitance	cload
	Positive input 1 is sampled at...	phi1
	Sampling Time	Ts
	Identifier for this integrator	a
	Identifier for the next integrator	b
Second, Third Integrators	Integration and sampling capacitors (Branch 1, Branch 2)	[Cint2,Cs12,Cs22]
	Capacitor nonlinear coefficients	[cnl1,cnl2]
	Weight's variance, rms eq.input noise, temperature	[0,innoise2,temp]
	OTA DC gain, transconductance, max. output current	[ao2,gm2,io2]
	Positive/Negative Output swing	[osp,−osp]
	Switch on-resistance	ron2
	OTA DC gain nonlinear coefs.	[avnl1,2,3,4]
	Parasitic capacitances before the OTA	[cpar1,cpar2]
	Load capacitance	cload
	Positive Input 1 is sampled at...	phi1
	Sampling Time	Ts
	Identifier for this integrator (second integrator)	b
	Identifier for this integrator (third integrator)	c
	Identifier for the next integrator	c
Comparators	Vhigh, Vlow	[vref, −vref]
	Offset, hysteresis	[0,hys]
	Phase ON	phi1
	Sampling time	Ts
	Identifier for this quantizer	quant1
To workspace (*y*)	Variable name	y
	Limit data points to last	N
	Decimation	1
	Sample time	Ts
	Save format	Array

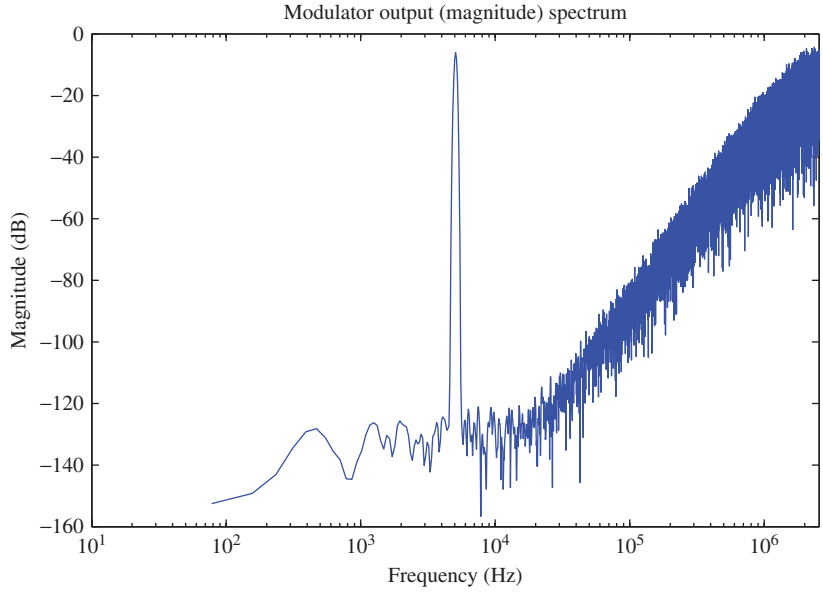


Figure A.17 Output spectrum (magnitude) of the ΣΔM in Figure A.15b.

Computing Output Spectrum

The output spectrum of the ΣΔM can be computed in SIMSIDES by following the next steps:

- Set up model parameters by using the M-file shown in Figure A.16.
- Simulate the modulator in Figure A.15b from the menu `Simulation -> Start`.
- Once the simulation is finished, go to `Analysis -> Node Spectrum Analysis` menu in SIMSIDES.
- Define the parameters requested in that menu. In this example, the sampling frequency is defined as `fs` and a Kaiser window function is used with a number of points `N` and `Beta = 20`.
- Click on `Compute` and then `Plot`, and the output spectrum shown in Figure A.17 is displayed.

SNR versus Input Amplitude Level

Figure A.18 shows the SNDR versus input amplitude level (or SNDR curve) of the ΣΔM in Figure A.15b. This figure has been obtained by using the `Analysis` menu and choosing `SNR/SNDR` analysis. In this example, the following parameters are used:

- `Parameter Name: Ain`, where `Ain` is the `Amplitude` parameter defined in the `Input Sine Wave` block in Figure A.15b.
- `Range [vi,vf]: [1e-6,2]`

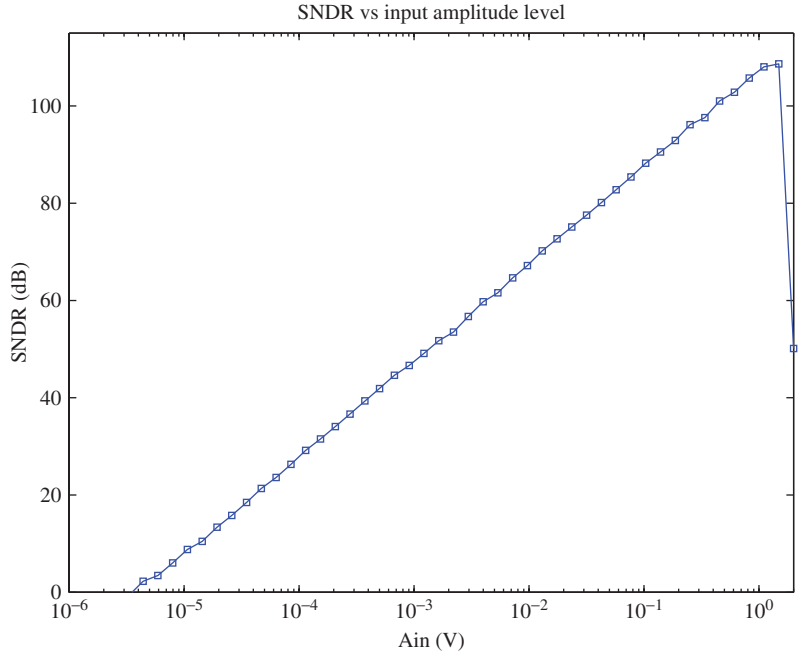


Figure A.18 SNDR versus input amplitude level of the ΣΔM in Figure A.15b.

- `N. of points: 50`
- `Scale: Logarithmic`
- `Analysis: SNR/SNDR`
- `Second Parameter: Off`

Once the aforementioned parameters are set up, click on `Continue` and the `SNR/SNDR` window menu shown in Figure A.7 is displayed. The requested parameters (i.e., sampling frequency, oversampling ratio, etc.) are set up according to the values given in Figure A.16, namely:

- `Name of the signal(s) to process: y`
- `Sampling frequency (Hz): fs`
- `Oversampling ratio: M`
- `Input Frequency (Hz): fi`
- `Window: Kaiser`
- `N. of Points: N`
- `Beta: 20`
- `Kind of Spectrum: LP`
- `Figure of merit: SNDR`

After setting up the aforementioned parameters, click on `Compute` and then `Plot` to obtain the curve given in Figure A.18.

Parametric Analysis Considering Only One Parameter

The `Parametric Analysis` menu can be used for studying the effect of a given model parameter on the modulator performance. For instance, let us consider the effect of the OTA transconductance g_m of the front-end integrator in Figure A.15b. In order to analyze the impact of this parameter on the effective resolution of the modulator, go to `Parametric Analysis` menu and set up the following parameters:

- `Parameter name: gm1`, that stands for g_m of the front-end integrator block in Figure A.15b.
- `Range [vi,vf]: [1e-5,1e-3]`
- `N. of points: 50`
- `Scale: Linear`
- `Analysis: SNR/SNDR`
- `Second Parameter: Off`

Once these parameters are defined, click on `Continue` and proceed similarly to previous examples in order to compute the SNDR. Figure A.19 shows the results of this analysis, by depicting the SNDR versus `gm1`.

Parametric Analysis Considering Two Parameters

The `Parametric Analysis` menu can be used also for implementing parametric analyses considering the variation of two different parameters. As an example, Figure A.20

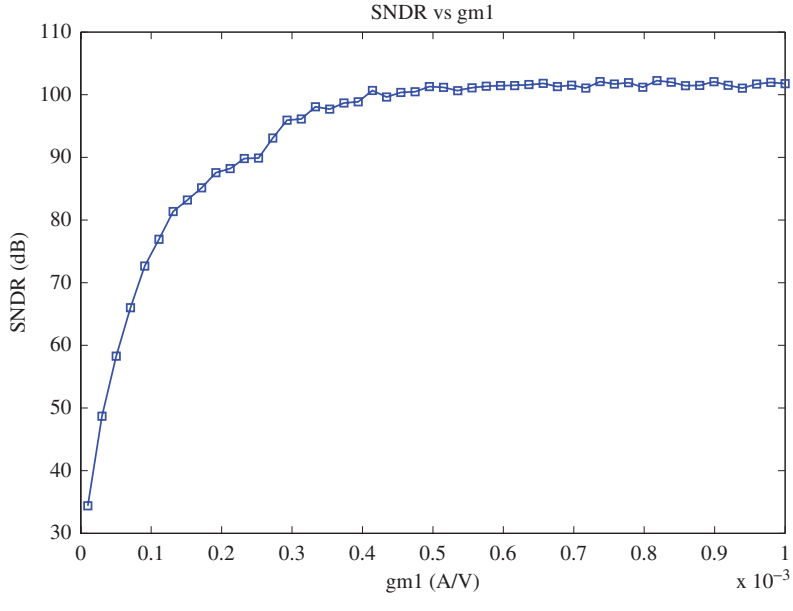


Figure A.19 Using parametric analysis to study the effect of a single model parameter: SNDR versus transconductance of the front-end amplifier for the ΣΔM in Figure A.15b.

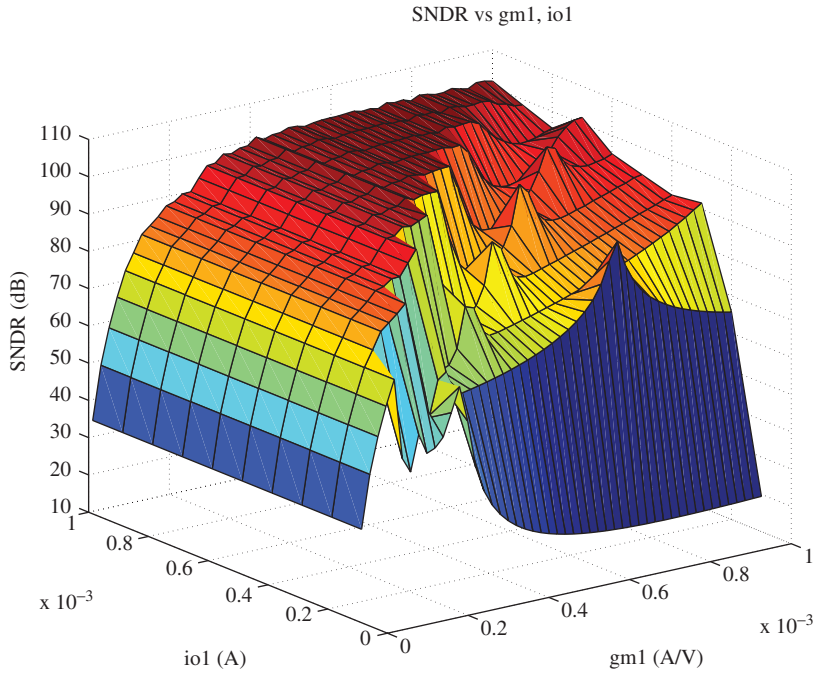


Figure A.20 Parametric analysis considering the effect of two parameters (g_{m1} and I_{o1}) on the SNDR.

shows the effect of both the OTA transconductance g_{m1} and the maximum output current I_{o1} of the front-end amplifier on the SNDR of the ΣΔM in Figure A.15b.

In order to obtain the graph in Figure A.20, the following parameters are set up in the `Parametric Analysis` menu:

- `Parameter name: io1`, that stands for the maximum output current I_o of the front-end integrator.
- `Range [vi,vf]: [1e-4,1e-3]`
- `N. of points: 10`

Computing Histograms

Finally, to conclude this example, Figure A.21 illustrates the histograms of the integrators outputs in the front-end stage of the modulator in Figure A.15b.

These histograms have been obtained by using the `Analysis -> Histograms` menu from SIMSIDES and setting up the following model parameters:

- `Name of the signal(s) to process: y1,y2`, that are the names given to the output of the integrators saved into the MATLAB workspace by using `To Workspace` blocks from the SIMULINK elementary library.
- `Number of bins: 100`

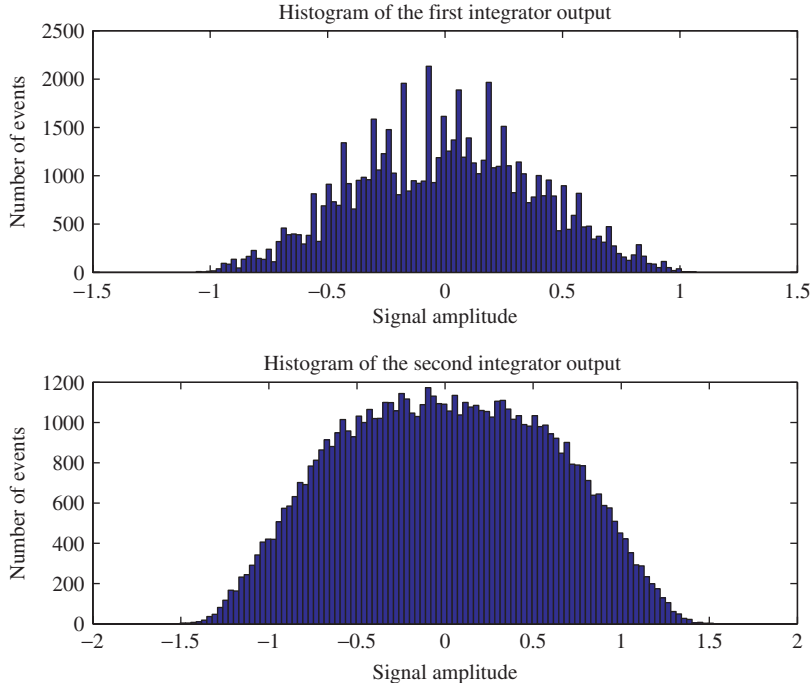


Figure A.21 Illustrating the use of histograms of the modulator in Figure A.15b.

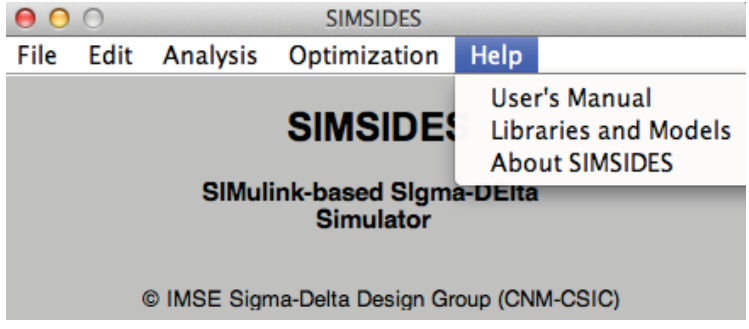


Figure A.22 Help menu.

A.5 Getting Help

SIMSIDES includes a help menu (illustrated in Figure A.22) from which this user guide can be opened by selecting `Help -> User Manual` in the SIMSIDES main window. In addition, a complete list of all behavioral models (and their corresponding parameters) included in SIMSIDES—described in Appendix B—can be also obtained from this menu by selecting `Help -> Libraries and Models`.

B

SIMSIDES Block Libraries and Models

This appendix contains a compilation of all ΣΔM building blocks and libraries included in SIMSIDES. A brief description of their purpose and functionality, as well as their main model parameters, is provided.

B.1 Overview of SIMSIDES Libraries

Table B.1 compiles all libraries included in SIMSIDES together with a brief description of their contents. These libraries are divided into two main categories: Ideal Libraries and Real Libraries. The former contains ideal building blocks, whereas the latter includes behavioral models that incorporate circuit-level nonidealities. The libraries containing integrators and resonators are subdivided into several specific sublibraries, which include in turn building-block models corresponding to different circuit-level implementations. For instance, SC integrators are subdivided into FE and LD integrators; CT integrators are subdivided into Gm-C, active-RC, etc.

B.2 Ideal Libraries

As shown in Table B.1, SIMSIDES includes four ideal libraries, namely: integrators, resonators, quantizers, and DACs. The building blocks contained in these libraries are described in the following sections.

B.2.1 Ideal Integrators

There are three kinds of ideal integrators in this library, namely `Ideal_CT_Integrator`, `Ideal_FE_Integrator`, and `Ideal_LD_Integrator`.

CMOS Sigma-Delta Converters: Practical Design Guide, First Edition. José M. de la Rosa and Rocío del Río.

Table B.1 Overview of SIMSIDES libraries

Ideal Libraries	**Sublibraries**	**Building Blocks**
Integrators	—	Ideal DT/CT integrators
Resonators	—	Ideal resonators
Quantizers & Comparators	—	Ideal quantizers
D/A Converters	—	Ideal DACs
Real Libraries	**Sublibraries**	**Building Blocks**
Integrators	SC FE integrators	Forward-Euler SC integrators
	SC LD integrators	Lossless-Direct SC integrators
	SI FE integrators	Forward-Euler SI integrators
	SI LD integrators	Lossless-Direct SI integrators
	Gm-C integrators	Gm-C integrators
	Gm-MC integrators	Miller OTA integrators
	RC integrators	Active-RC integrators
	MOSFET-C integrators	MOSFET-C integrators
Resonators	SC FE resonators	Resonators based on FE SC integrators
	SC LD resonators	Resonators based on LD SC integrators
	SI FE resonators	Resonators based on FE SI integrators
	SI LD resonators	Resonators based on LD SI integrators
	Gm-C resonators	Resonators based on Gm-C integrators
	Gm-LC resonators	Resonators based on Gm-LC integrators
Quantizers & Comparators	—	Nonideal single-bit & multibit quantizers
D/A Converters	—	Nonideal single-bit and multibit DACs
Auxiliary Blocks	—	Adders, latches, DEM blocks, etc

Building-Block Model Purpose and Description

`Ideal_CT_Integrator` block models the ideal S-domain transfer function of a CT integrator, given by,

$$\mathrm{ITF}(s) = g \cdot \frac{1}{s} \tag{B.1}$$

where g represents the integrator gain (also referred to as integrator weight).

`Ideal_FE_Integrator` and `Ideal_LD_Integrator` blocks model the ideal Z-domain transfer functions of DT FE and LD integrators, respectively given by:

$$\begin{aligned} \mathrm{ITF}_{\mathrm{FE}}(z) &= g \cdot \frac{z^{-1}}{1 - z^{-1}} \\ \mathrm{ITF}_{\mathrm{LD}}(z) &= g \cdot \frac{z^{-1/2}}{1 - z^{-1}} \end{aligned} \tag{B.2}$$

Model Parameters

The following model parameters are included in the dialogue box of the abovementioned models:

- `Gain`, that defines the integrator weight g.
- `Sampling Time`; that is, the sampling period of both the `Ideal_FE_Integrator` and `Ideal_LD_Integrator` blocks.

B.2.2 Ideal Resonators

This library includes diverse ideal resonator blocks described here.

Ideal_LD_Resonator

This block consists of a DT resonator made up of two LD integrators connected in a feedback loop as illustrated in Figure B.1a. Its model parameters are the following:

- `Af`; that is, the forward loop gain.
- `Afb`, that represents the feedback loop gain.
- `Sampling Time`.

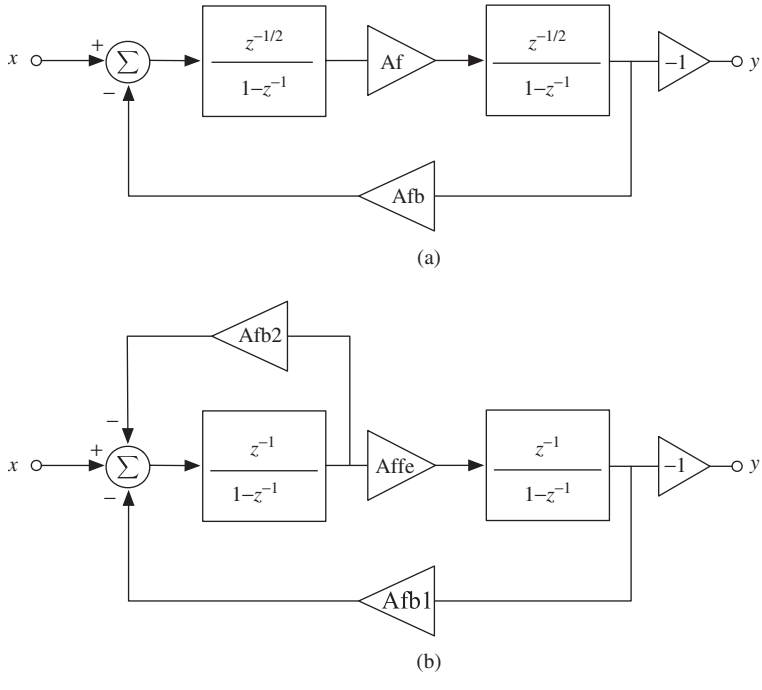


Figure B.1 Z-domain block diagram of: (a) ideal_LD_Resonator and (b) ideal_FE_Resonator.

Ideal_FE_Resonator

This block models a DT resonator made up of two FE integrators connected in a feedback loop as shown in Figure B.1b. In this case, the model parameters are the following:

- `Affe`, that represents the feed-forward loop gain.
- `Afb1`; that is, the global feedback loop gain.
- `Afb2`; that is, the local feedback loop gain.
- `Sampling Time`.

Ideal_CT_Resonator

This model corresponds to a CT (biquad) resonator with an *S*-domain transfer function given by:

$$\mathrm{RTF}(s) = \frac{(\pi/2)\cdot s}{s^2 + (\pi/2)^2} \tag{B.3}$$

The above transfer function is implemented by using the `Transfer Fcn` block from the `Continuous` SIMULINK library.

B.2.3 Ideal Quantizers

This library includes several building blocks that model single-bit and multibit/multilevel quantizers intended for both voltage-mode (SC/CT) and current-mode (SI) $\Sigma\Delta$Ms.

Ideal_Comparator

This block models the input/output DC characteristic of an ideal comparator, given by a sign function as,

$$v_o = \begin{cases} V_{\text{high}} & v_i \geq 0 \\ V_{\text{low}} & v_i < 0 \end{cases} \tag{B.4}$$

where v_i and v_o represent the input and output voltages, respectively, and V_{high} and V_{low} represent the analog values of the logic one and logic zero, respectively.

The model parameters of the `Ideal_Comparator` block are the following:

- $\mathtt{V_{high}}$, $\mathtt{V_{low}}$, that stand respectively for V_{high} and V_{low}.
- `Phase`, that models the clock phase in which the input signal is sampled. Two nonoverlapping clock phases, denoted as `phi1,2` are assumed.
- `Sampling Time`; that is, the clock signal period.
- `Identifier for this Quantizer`, that defines an identification name for the block (used by some dynamic errors such as the incomplete settling error in SC circuits).

Ideal_Comparator_for_SI

This block models a current-mode comparator used in SI-$\Sigma\Delta$Ms. The behavioral model is exactly the same as that used in the `Ideal_Comparator` block, except that in this

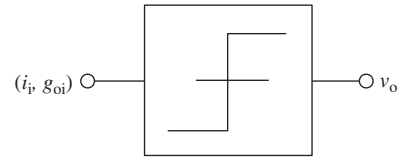


Figure B.2 Input and output signals in an Ideal_Comparator_for_SI model.

case the input signal is a current-mode signal that is modeled as a matrix made up of two vector elements, namely: the current signal itself and the output conductance of the current-mode building block (i.e, integrator, resonator, etc.) connected at the input of the comparator.

This way, the information provided to the model at each sampling time is a vector of two elements as illustrated in Figure B.2, where i_i is the input current, v_o is the output voltage, and g_{oi} is the output conductance of the building block connected at the comparator input.

Ideal_Multibit_Quantizer

This building block includes the ideal behavioral model of a multibit quantizer with a DC input/output *midrise* characteristic as illustrated in Figure B.3a.

The model parameters of this block are the following ones:

- `V_high, V_low`, that represent, respectively, the maximum and minimum values of the quantizer output FS range—see Figure B.3a.
- `Number of bits` of the quantizer.
- `Xmax-Xmin`, that defines the input FS range of the quantizer.

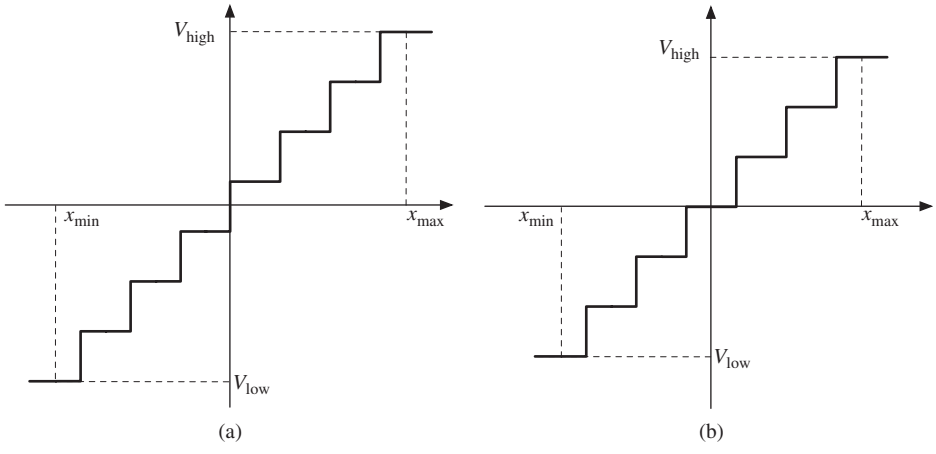


Figure B.3 Illustrating the input/output DC characteristic of: (a) multibit (3-bit) *midrise* quantizer and (b) multilevel (7-level) *midtreat* quantizer.

- `Phase ON`, that is the clock phase in which the input signal is sampled, considering a two-phase clock signal generator.
- `Sampling Time`, that denotes the clock period.

Ideal_Multibit_Quantizer_for_SI

This block models a current-mode multibit quantizer with the same input/output characteristic as an `Ideal_Multibit_Quantizer`, but considering that the input signal has two components—i_i and g_{oi}—in the same way as in Figure B.2.

Ideal_Multibit_Quantizer_levels

This building block models a multilevel quantizer in which the input/output characteristic is defined as a function of the number of levels, instead of the number of bits as in the `Ideal_Multibit_Quantizer` block. Thus, the same parameters are used, except for the number of bits, which is replaced by `Number of levels`. If this parameter is even, a *midrise* quantization characteristic similar to that shown in Figure B.3a is implemented. Otherwise, a *midtreat* characteristic similar to that depicted in Figure B.3b is provided.

Ideal_Multibit_Quantizer_levels_SD2

This building block includes the same behavioral model as the previous one, but the output is a thermometric-coded bit array. This is used in combination with multilevel DACs with DEM techniques, as illustrated in Section B.8.

Ideal_Sampler

This building block models an ideal S&H circuit which can be used in CT-$\Sigma\Delta$Ms, in which the signal is sampled at the input of internal quantizers.

Their model parameters are the following:

- `Sampling Time`, that denotes the clock period.
- `Input clock phase`, that defines the sampling clock phase.

B.2.4 Ideal D/A Converters

The blocks included in this library model different kinds of ideal DACs, namely:

- `Ideal_DAC`, that is used for modeling ideal single-bit DACs for either SC- or CT-$\Sigma\Delta$Ms. They are simply modeled as a voltage gain, named `Gain` in the model.
- `Ideal_DAC_for_SI`, that models a single-bit DAC for SI-$\Sigma\Delta$Ms.
- `Ideal_DAC_dig_level_SD2`, that consists of a multilevel ideal DAC.

The last two models and their associated parameters are described below.

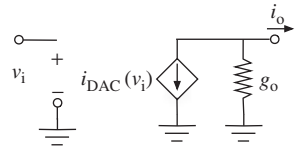


Figure B.4 Equivalent circuit of Ideal_DAC_for_SI model.

Ideal_DAC_for_SI

Figure B.4 shows the equivalent circuit of the `Ideal_DAC_for_SI` block. It consists of a voltage-controlled current source in parallel with a finite output conductance g_o. The current source i_{DAC} is a sign function of the input voltage v_i given by,

$$i_{DAC}(v_i) = \begin{cases} +I_{ref} & v_i = +v_{ref} \\ -I_{ref} & v_i = -v_{ref} \end{cases} \tag{B.5}$$

where I_{ref} and v_{ref} represent the modulator FS reference current and voltage, respectively. The model parameters included in the `Ideal_DAC_for_SI` model are the following:

- `Gain`, that models the DAC gain; that is, I_{ref}/v_{ref}.
- `Gout`, that represents g_o.
- `Sampling Time`, that models the clock period.
- `Input Clock Phase`; that is, the clock phase at which the DAC input is sampled.

Ideal_DAC_dig_level_SD2

This block transforms a thermometric-coded digital input into its corresponding analog level. The model parameters used in this behavioral model are:

- `Vhigh`, that defines the upper limit of the quantization FS range.
- `Vlow`, that corresponds to the lower limit of the quantization FS range.
- `Number of levels` of the embedded quantizer.

B.3 Real SC Building-Block Libraries

SIMSIDES includes two libraries of SC integrators and two libraries of SC resonators. These libraries are described subsequently.

B.3.1 Real SC Integrators

There are two SC integrator model libraries in SIMSIDES: one including FE SC integrator models and the other one including LD SC integrators. In both cases, integrator models are classified according to the nonideal effects that are included in the model and the number of SC branches connected at the integrator input. This way, for each model there

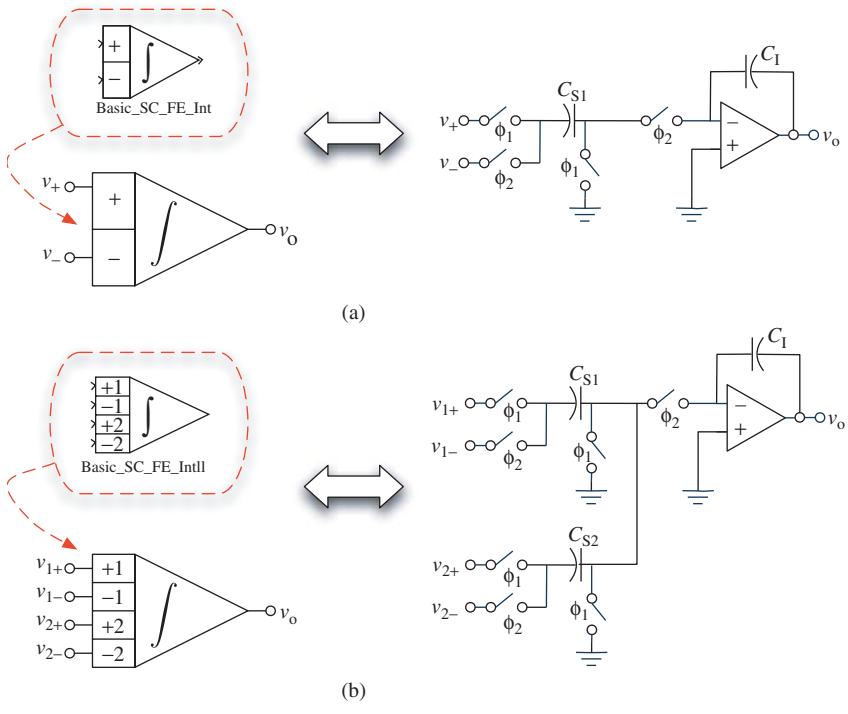


Figure B.5 SC integrator symbol in SIMSIDES: (a) one-branch integrator and (b) two-branch integrator.

are four building blocks using the same behavioral model except for the number of input SC branches.

As an illustration, Figure B.5 shows the symbol used in SIMSIDES for one-branch SC FE integrators (Figure B.5a) and two-branch SC FE integrators (Figure B.5b), together with their equivalent SC circuits. Note that although single-ended conceptual schematics are shown in this figure, fully-differential circuits are assumed in the behavioral models.

Both integrators in Figure B.5 use the same behavioral model that consists of an ideal SC FE integrator with output swing limitation. The behavioral model corresponding to a one-branch SC FE integrator is named `Basic_SC_FE_Int`, while the model of the two-branch SC FE integrator is named `Basic_SC_FE_IntII`. Following this nomenclature, `Basic_SC_FE_IntIII` and `Basic_SC_FE_IntIV` models are used for three- and four-branch SC FE integrators, respectively.

Table B.2 lists all SC integrator models available in SIMSIDES including a brief description of the nonidealities included in each of them. Note that the model names included in Table B.2 correspond to one-branch integrators. The same models are available for integrators with up to four input branches.

The model named `SC_FE_Int_1b_SD2` represents one-branch SC FE integrators with all circuit nonideal effects, including the degradation caused by the switch on-resistance on the integrator GB and SR. In this model, notation `1b` is used for denoting

Table B.2 Library of SC (FE/LD) integrators included in SIMSIDES

Model name	Circuit effects included
`Basic_SC_FE_Int` `Basic_SC_LD_Int`	Output swing limitation
`SC_FE_Int_Non_linear_C` `SC_LD_Int_Non_linear_C`	Output swing limitation, capacitor nonlinearity.
`SC_FE_Int_Weight_Mismatch` `SC_LD_Int_Weight_Mismatch`	Output swing limitation, capacitor mismatch.
`SC_FE_Int_Non_Linear_Sampling` `SC_LD_Int_Non_Linear_Sampling`	Output swing limitation, nonlinear switch on-resistance.
`SC_FE_Int_FiniteDCgain` `SC_LD_Int_FiniteDCgain`	Finite OTA DC gain, output swing limitation, parasitic OTA caps.
`SC_FE_Int_Finite&Non_LinearDCGain` `SC_LD_Int_Finite&Non_LinearDCGain`	Finite nonlinear OTA DC gain, output swing limitation parasitic OTA caps.
`SC_FE_Int_Noise` `SC_LD_Int_Noise`	OTA thermal noise, output swing limitation, parasitic/load OTA caps.
`SC_FE_Int_Settling` `SC_LD_Int_Settling`	Incomplete settling error, output swing limitation, parasitic/load OTA caps.
`SC_FE_Integrator_All_Effects` `SC_LD_Integrator_All_Effects`	switch on-resistance, capacitor nonlinearity and mismatch, settling error, finite (nonlinear) DC gain, thermal noise, parasitic/load capacitors, output swing limitation.
`SC_FE_Integrator_All_Effects&NonLinSamp` `SC_LD_Integrator_All_Effects&NonLinSamp`	switch nonlinear on-resistance, capacitor nonlinearity and mismatch, settling error, finite (nonlinear) DC gain, thermal noise, parasitic/load capacitors, output swing limitation.
`SC_FE_Int_1b_SD2`	switch on-resistance and its effect on GB and SR, capacitor nonlinearity and mismatch, settling error, finite (nonlinear) DC gain, thermal noise, parasitic/load capacitors, output swing limitation.
`SC_FE_Int_1b_DEM_SD2`	switch on-resistance and its effect on GB and SR, array of unit sampling capacitors, capacitor nonlinearity and mismatch, settling error, finite (nonlinear) DC gain, thermal noise, parasitic/load capacitors, output swing limitation.

one input SC branch. Similarly, other models the names of which include `nb` denote *n* input SC branches.

The model named `SC_FE_Int_1b_DEM_SD2` includes the same nonideal effects as `SC_FE_Int_1b_SD2` model but it also allows modeling the sampling capacitor as an array of unit capacitors. This array is used in combination with multilevel quantizers and DACs that incorporate DWA/DEM algorithms as described in Section B.8. In this way, the number of unit capacitors in the array must be exactly the same as the number of DAC levels. Moreover, this array should include the DAC mismatch error that can be modeled as a Gaussian distribution. Both `SC_FE_Int_1b_SD2` and `SC_FE_Int_1b_DEM_SD2` models provide detailed information about the integrator equivalent input-referred thermal noise, equivalent load capacitances, and transient response model parameters. All these pieces of information are displayed in the MATLAB command window after simulation.

Table B.3 lists the most important parameters used by the SC integrator behavioral models in SIMSIDES, as well as a brief description of all of them.

B.3.2 Real SC Resonators

SIMSIDES has two SC resonator model libraries corresponding to FEI-based resonators and LDI-based resonators. All building-block models correspond to the Z-domain block diagrams shown in Figure B.1 but they are implemented by using the SC integrator models described in the previous section. As an illustration, Figure B.6 shows the SIMSIDES block diagram of an SC LDI-based resonator (Figure B.6a) and an SC FEI-based resonator (Figure B.6b), which correspond to Figures B.1a and B.1b, respectively.

Following the same philosophy as that used in SC integrators, the behavioral models of SC resonators in SIMSIDES are classified attending to the number of input SC branches and the circuit nonideal effects included in the models. As an illustration, Figure B.7 shows an excerpt of both SC resonators libraries. Note that each row of blocks includes the same circuit nonideal effects, with the only difference among them being the number of input branches.

Table B.4 lists all SC resonator models available in SIMSIDES, including a brief description of the nonidealities considered in each of them. The parameters used in these models are the same as those included in SC integrator models—listed in Table B.3. In addition to these parameters, the resonator gain can also be defined by the user by setting a parameter named `Gain` (see Figure B.6), that can be defined in the model dialogue box.

B.4 Real SI Building-Block Libraries

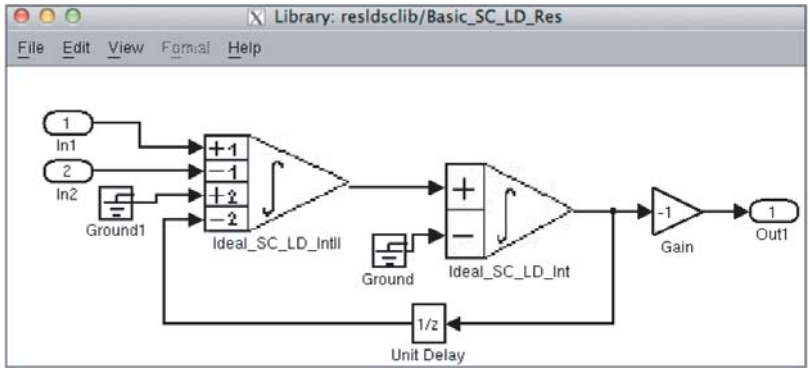
SIMSIDES includes all necessary building blocks for the simulation of SI-$\Sigma\Delta$Ms. This section describes SI integrators and resonators, as well as their main model parameters.

B.4.1 Real SI Integrators

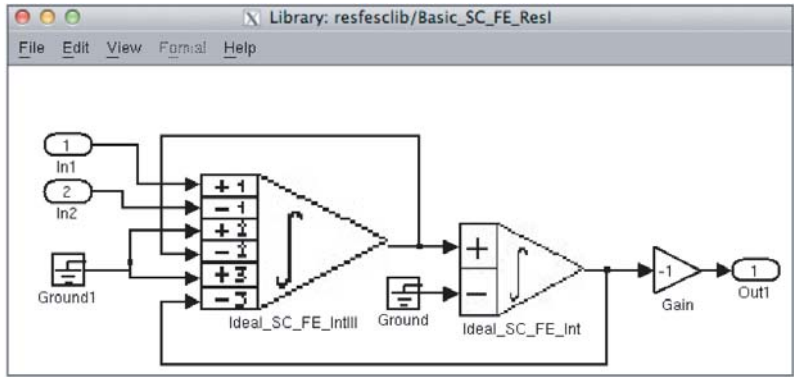
Following the same classification criteria as that followed for SC building-block models, there are two libraries of SI integrators in SIMSIDES: one including FE SI integrators and the other one including LD SI integrators. Figure B.8 shows the symbol used in

Table B.3 Model parameters used in SIMSIDES SC (FE/LD) integrators

Parameter name (in alphabetical order)	Brief description
`Array of sampling capacitors for DEM branch`	Array of unit capacitors used with multilevel DACs with DEM
`B (switch parameters)`	MOS large-signal transconductance (analytic model)
`Bandwidth (BW)`	Input signal bandwidth
`Capacitor (first/second)-order nonlinearity`	Capacitor (first/second) order nonlinearity
`Finite and Linear Ron`	switch on-resistance, linear model
`Finite DC Gain of the AO`	Finite OTA DC gain
`g (switch parameters)`	Finite switch on-conductance (analytic model)
`Identifier for this integrator`	Identifier used for settling error model
`Input Equivalent Thermal Noise`	OTA input-referred thermal noise
`Input parameters [A,fi,ph] (switch)`	Amplitude, frequency, and phase of the sinewave input (table look-up model)
`Integration/Sampling Capacitor`	Integration/sampling capacitors
`Integration additional load`	Additional load capacitance at the integration phase
`Load Capacitor (cload)`	Integrator load capacitance
`Maximum output current (Io)`	OTA maximum output current
`Nonlinearity of the DC Gain`	OTA DC gain nonlinear coefficients
`Output Swing Up/Down`	Maximum/minimum output swing limits
`Parasitic Capacitor before the AO (Cp)`	Parasitic capacitance at the OTA input
`pcoef (switch parameters)`	nonlinear coefficients of the switch on-resistance (table look-up model)
`Positive Input is Sampled in...`	Input-switch clock phase
`Ron`	switch on-resistance
`Sampling additional load`	Additional load capacitance at the sampling phase
`Sampling Time`	Clock signal period
`Switch on-resistance (Ron)`	Switch on-resistance
`Temp`	Temperature (K)
`Transconductance of the AO (gm)`	OTA transconductance
`Variance`	Variance of the capacitor mismatch error



(a)



(b)

Figure B.6 Block diagram of SC resonators in SIMSIDES: (a) LDI-based resonator and (b) FEI-based resonator.

SIMSIDES for SI integrators together with a conceptual schematic of an FE SI integrator and an LD SI integrator. In both cases, different models are included in SIMSIDES, which are classified attending to the number of nonideal effects that are taken into account, as detailed in Table B.5.

Apart from the building blocks in Table B.5, a current-mode *buffer* block, named `Buffer`, is included in all SIMSIDES SI libraries. This block transforms an input current vector into an output matrix made up of two vector elements: the input current signal itself in parallel with an output conductance, named `Gout of the source` in the model. Figure B.9 illustrates the operation of the `Buffer` block, by showing the SIMSIDES block symbol and its equivalent circuit.

B.4.2 Real SI Resonators

As in the case of SC circuits, two different types of SI resonators are modeled in SIMSIDES, namely SI FEI-based resonators and LDI-based resonators. Both block diagrams

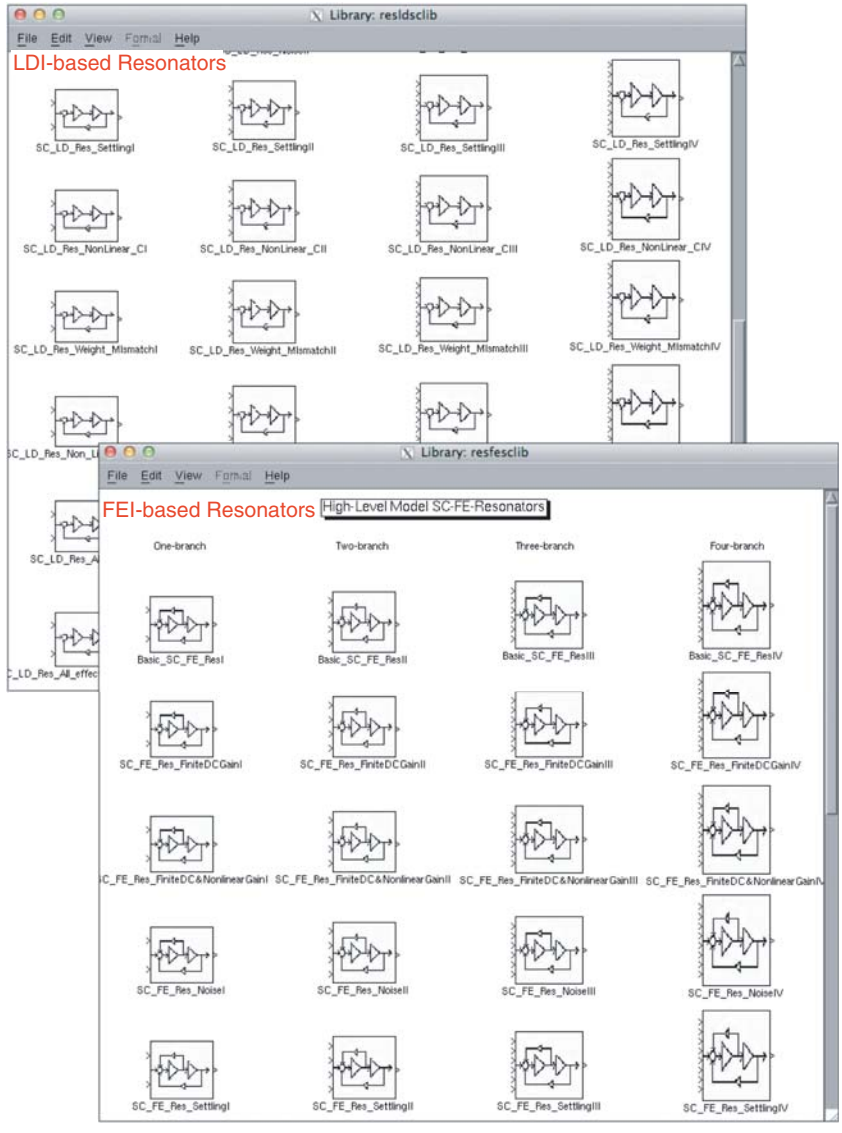


Figure B.7 Excerpt of SC resonator libraries in SIMSIDES.

are depicted in Figure B.10. In both cases, integrator gain parameters—denoted as `Affe`, `Afb1`, and `Afb2`—can be defined by the user in the block dialogue window, as well as their associated gain errors—respectively denoted in the model as `MU1`, `MU2`, `MU3`.

Table B.6 lists all SI resonator blocks included in SIMSIDES together with a brief description of the error mechanisms taken into account in each model. These errors and their associated model parameters, which are the same as those used in SI integrators, are described here.

Table B.4 Library of SC (FE/LD) resonators included in SIMSIDES

Model name	Circuit effects included
`Basic_SC_FE_Res` `Basic_SC_LD_Res`	Output swing limitation
`SC_FE_Res_NonLinear_C` `SC_LD_Res_NonLinear_C`	Output swing limitation, capacitor nonlinearity.
`SC_FE_Res_Weight_Mismatch` `SC_LD_Res_Weight_Mismatch`	Output swing limitation, capacitor mismatch.
`SC_FE_Res_Non_Linear_Sampling` `SC_LD_Res_Non_Linear_Sampling`	Output swing limitation, nonlinear switch on-resistance.
`SC_FE_Res_FiniteDCgain` `SC_LD_Res_FiniteDCgain`	Finite OTA DC gain, output swing limitation, parasitic OTA caps.
`SC_FE_Res_FiniteDC&NonLinearGain` `SC_LD_Res_FiniteDC&NonLinearGain`	Finite nonlinear OTA DC gain, output swing limitation, parasitic OTA caps.
`SC_FE_Res_Noise` `SC_LD_Res_Noise`	OTA thermal noise, output swing limitation, parasitic/load OTA caps.
`SC_FE_Res_Settling` `SC_LD_Res_Settling`	Incomplete settling error, output swing limitation, parasitic/load OTA caps.
`SC_FE_Res_All_effects` `SC_LD_Res_All_effects`	switch on-resistance, capacitor nonlinearity and mismatch, settling error, finite (nonlinear) DC gain, thermal noise, parasitic/load capacitors.
`SC_FE_Res_All_effects&NonLinSamp` `SC_LD_Res_All_effects&NonLinSamp`	switch nonlinear on-resistance, capacitor nonlinearity and mismatch, settling error, finite (nonlinear) DC gain, thermal noise, parasitic/load capacitors, output swing limitation.

B.4.3 SI Errors and Model Parameters

This section gives a brief description of the main errors and model parameters included in SI model libraries of SIMSIDES. For the sake of clarity, these errors are linked to some of the building blocks in which they are included.

Basic_SI_FE(LD)_Integrator and Basic_SI_FE(LD)_Resonator

The model parameters included in these blocks are:

- `Integrator Gain`: integrator weight (see Figure B.8).
- `Iomax(=-Iomin)`: maximum/minimum integrator output current.
- `MU1,MU3`: first-order and third-order coefficients of the gain nonlinearity.

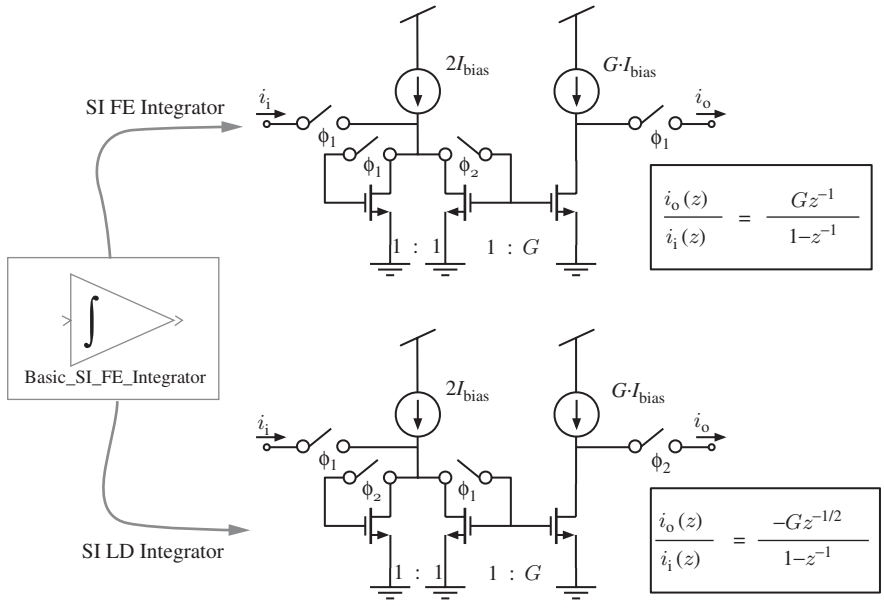


Figure B.8 SI integrator symbol used in SIMSIDES and its corresponding conceptual schematics.

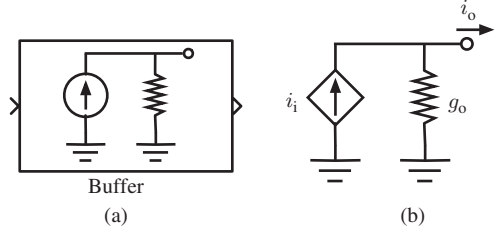


Figure B.9 SI buffer used in SIMSIDES: (a) symbol and (b) equivalent circuit.

The Z-domain transfer function of this integrator block is given by the following expression,

$$i_o(z) = (1 - \text{MU1}) \cdot i_{oi}(z) + \text{MU3} \cdot i_{oi}^3(z) \tag{B.6}$$

where $i_{oi}(z)$ is the Z-transform of the ideal output current.

The abovementioned expression is also used for both LDI- and FEI-based SI resonators, although in this case the nonlinear coefficients are named `SHI1, SHI3` instead of `MU1, MU3`.

SI_FE(LD)_Int_Finite_Conductance

These blocks include the effect of finite input–output conductance ratio error. To this purpose, the equivalent circuit shown in Figure B.11 is solved during sampling phase.

Table B.5 Library of SI (FE/LD) integrators included in SIMSIDES

Model name	Circuit effects included
`Ideal_SI_FE_Integrator` `Ideal_SI_LD_Integrator`	Ideal SI (FE/LD) integrator.
`Basic_SI_FE_Integrator` `Basic_SI_LD_Integrator`	Output current limits, nonlinear gain.
`SI_FE_Int_Finite_Conductance` `SI_LDI_Finite_Conductance`	Output current limits, finite nonlinear output conductance, input voltage limits, thermal noise.
`SI_FE_Int_Finite_Conductance&Settling` `SI_LDI_Finite_Conductance&Settling`	Output current limits, finite nonlinear input/output conductance, input voltage limits, settling error, thermal noise.
`SI_FE_Int_Finite_...&Settling&Charge_Injection` `SI_LDI_Finite_...&Settling&Charge_Injection`	Output current limits, finite nonlinear input/output conductance, input voltage limits, settling error, thermal noise, charge injection error.

This circuit corresponds to the SI LD integrator[1] of Figure B.8 on clock phase ϕ_1. In this circuit, the input signal is modeled as an ideal current in parallel with a finite conductance, denoted as g_{oi}. Memory-cell transistor 1 (M1) is configured in the hold phase and is modeled by its drain current i_{d1} in parallel with an output conductance g_o. In contrast, memory-cell transistor 2 (M2) is operating in sampling phase and it is modeled by the parallel connection of its output conductance with an input resistor. This resistor is a nonlinear function of the drain current i_{d2} of M2, given by,

$$v_i(i_{d2}) \simeq \mathrm{A1} \cdot i_{d2} + \mathrm{A3} \cdot i_{d2}^3 \tag{B.7}$$

where `A1, A3` represent the first-order and third-order nonlinear coefficients of the input resistance. These parameters are defined in the models together with the following ones:

- `Gout`: output conductance g_o of memory cells.
- `Vmax/Vmin`: maximum/minimum values of v_i in Equation (B.7).
- `Ibias`: integrator bias current (see Figure B.8).

[1] A similar circuit is used for SI FE integrators.

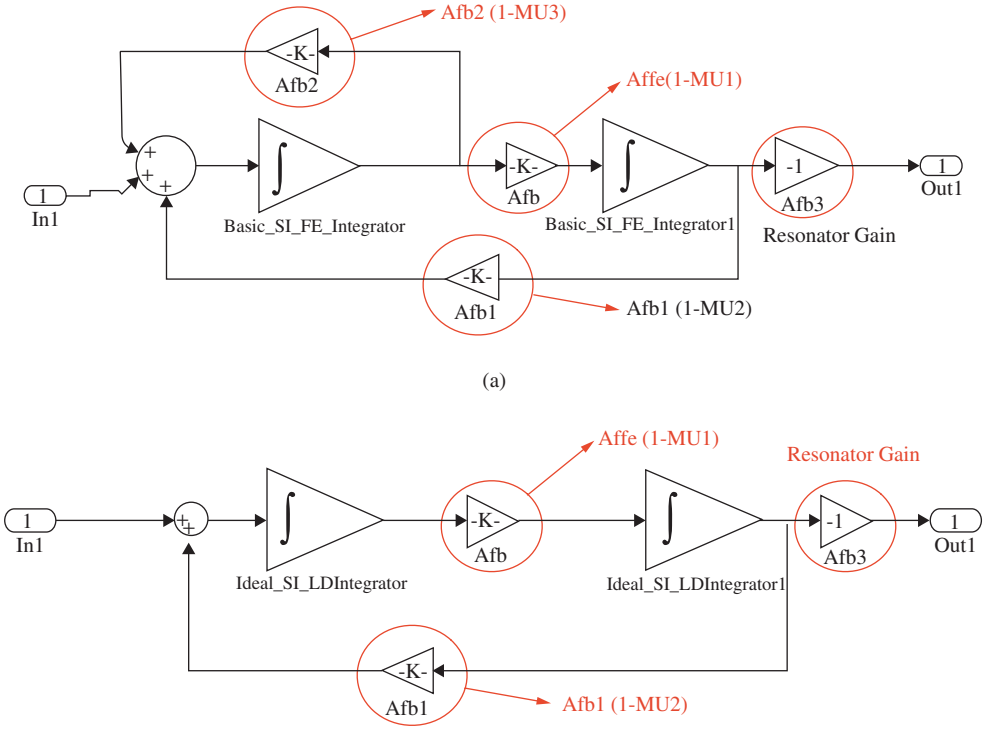


Figure B.10 Block diagram of SI resonators in SIMSIDES: (a) FEI-loop resonator and (b) LDI-loop resonator.

SI_FE(LD)_Int_Finite_Conductance&Settling&ChargeInjection

Those SI building blocks including nonlinear incomplete settling and charge injection require the following additional model parameters:

- `Gmo`: operating-point small-signal transconductance of memory transistors.
- `Cgs`: gate-to-source capacitance of memory transistors.
- `Eq`: charge injection error.

The charge injection error `Eq` is defined as a relative error on the voltage stored in the gate-to-source capacitance, given by:

$$v_{\text{gs, nonideal}} = (1 - \text{Eq}) \cdot v_{\text{gs, ideal}} \tag{B.8}$$

B.5 Real CT Building-Block Libraries

Figure B.12 shows the CT building-block model libraries included in SIMSIDES. There are four libraries of CT integrators and two libraries of CT resonators, which are classified

Table B.6 Library of SI (FE/LD) resonators included in SIMSIDES

Model name	Circuit effects included
`Ideal_SI_FE_Resonator` `Ideal_SI_LD_Resonator`	Ideal SI (FE/LD) resonator.
`Basic_SI_FE_Resonator` `Basic_SI_LD_Resonator`	Output current limits, nonlinear gain.
`SI_FE_Res_Finite_Conductance` `SI_LD_Resonator_Finite_Conductance`	Output current limits, finite nonlinear output conductance, input voltage limits, thermal noise.
`SI_FE_Res_Finite_Conductance&Settling` `SI_LD_Res_Finite_Conductance&Settling`	Output current limits, finite nonlinear input/output conductance, input voltage limits, settling error, thermal noise.
`SI_FE_Res_Finite_...&Settling&Charge_Injection` `SI_LD_Res_Finite_...&Settling&Charge_Injection`	Output current limits, finite nonlinear input/output conductance, input voltage limits, settling error, thermal noise, charge injection error.

attending to the circuit nature of the building blocks, namely Gm-C, Gm-MC, Gm-LC, active-RC, and MOSFET-C.

B.5.1 Real CT Integrators

Tables B.7–B.10 list all models[2] included in CT integrator libraries shown in Figure B.12, together with a brief description of the nonideal effects included. An explanation of the most representative blocks and their associated model parameters is given in the following sections.

Model Parameters Used in Transconductors and Gm-C Integrator Building Blocks

Building blocks listed in Table B.7 are used in SIMSIDES to model transconductors and Gm-C integrators, considering the effect of different circuit-level nonideal effects. This section gives a description of the most significant model parameters included in these blocks.

[2] RC_Int_1,2,3in models allow setting up of transistor-level parameters such as channel-length modulation, gate-to-source overdrive voltage, saturation voltage, supply voltage, etc.

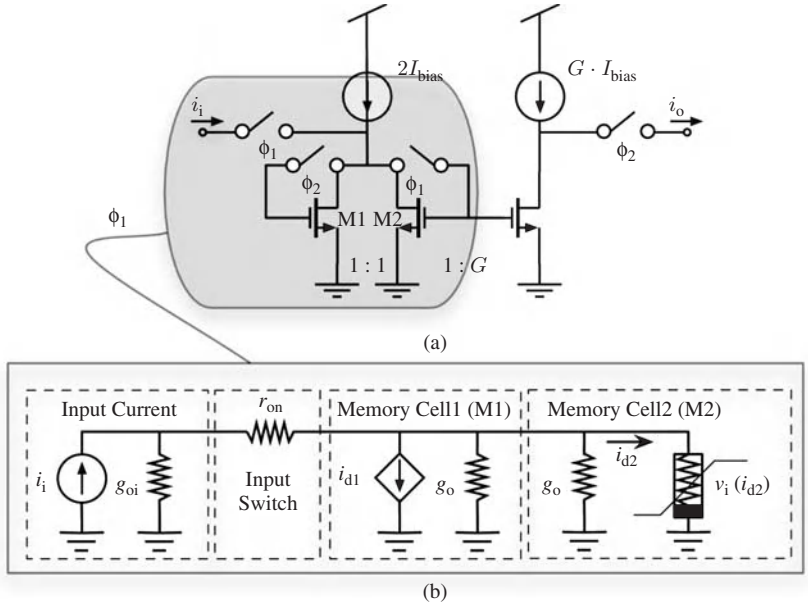


Figure B.11 Modeling finite input–output conductance ratio error in SIMSIDES: (a) SI LD integrator and (b) equivalent circuit during sampling phase (ϕ_1).

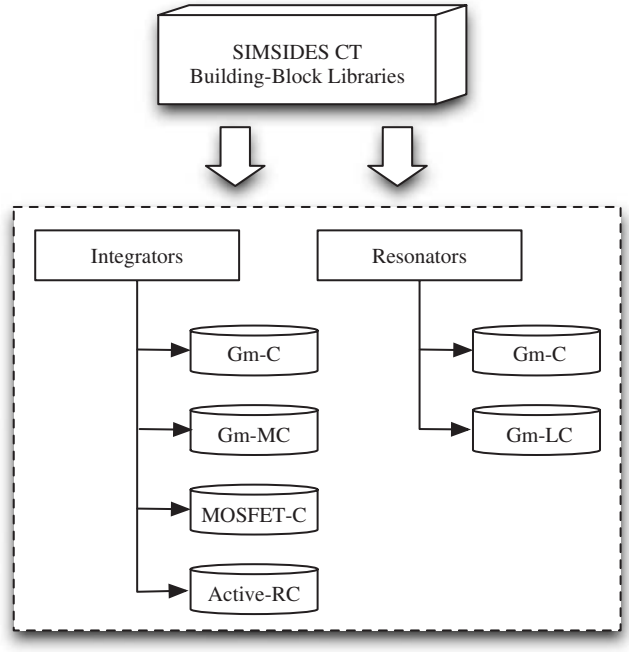


Figure B.12 Classification of SIMSIDES CT model libraries.

Table B.7 Gm-C integrator library models in SIMSIDES

Model name	Circuit effects included
`Ideal_OTA_C_CTint`	Ideal Gm-C integrator.
`Transconductor`	Input saturation voltage, nonlinear transconductance.
`gm_no_noise_new`	Output saturation voltage, third-order intercept point.
`1pole_gm`	Gm-C output impedance.
`OTA_C_CT_1pole`	Input/output saturation voltage, finite OTA DC gain, nonlinear transconductance, one-pole dynamic, time-constant error, nonlinear transconductance, thermal noise.
`OTA_C_CT_2poles` `OTA_C_CT_2polesb`	Input/output saturation voltage, Finite OTA DC gain, nonlinear transconductance, two-pole dynamic, time-constant error, nonlinear transconductance, thermal noise.

Table B.8 Gm-MC integrator library models in SIMSIDES

Model name	Circuit effects included
`Gm_MC_CTInt_1pole`	Input/output saturation voltage, finite OTA DC gain, parasitic capacitances, one-pole dynamic, thermal noise.
`Gm_MC_CTInt_2poles`	Input/output saturation voltage, finite OTA DC gain, parasitic capacitances, two-pole dynamic, thermal noise.
`Gm...1pole&Large_signal_distortion`	Input/output saturation voltage, output current limit, finite OTA DC gain, parasitic capacitances, one-pole dynamic.
`Gm...2poles&Large_signal_distortion`	Input/output saturation voltage, output current limit, finite OTA DC gain, parasitic capacitances, two-pole dynamic.
`Gm...1pole&Small_Signal_Distortion`	Input/output saturation voltage, output current limit, finite OTA DC gain, nonlinear transconductance, parasitic capacitances, one-pole dynamic.
`Gm...2poles&Small_Signal_Distortion`	Input/output saturation voltage, output current limit, finite OTA DC gain, nonlinear transconductance, parasitic capacitances, two-pole dynamic.

Table B.9 Active-RC integrator library models in SIMSIDES

Model name	Circuit effects included
`RC_CTInt_1pole`	OTA output swing limitation, finite OTA DC gain, parasitic capacitances, capacitance voltage coefficient, one-pole dynamic, thermal noise.
`RC_CTInt_2poles`	OTA output swing limitation, finite OTA DC gain, parasitic capacitances, capacitance voltage coefficient, two-pole dynamic, thermal noise.
`RC...1pole&Large_signal_distortion`	OTA output swing limitation, output current limit, finite OTA DC gain, parasitic capacitances, capacitance voltage coefficient, one-pole dynamic, thermal noise.
`RC...2poles&Large_signal_distortion`	OTA output swing limitation, output current limit, finite OTA DC gain, parasitic capacitances, capacitance voltage coefficient, two-pole dynamic, thermal noise.

`RC_Int_1in`
`RC_Int_2in`
`RC_Int_3in`
	OTA output swing limitation, finite OTA DC gain, nonlinear trans., slew rate, parasitic capacitances, one-pole dynamic, thermal noise.

Table B.10 MOSFET-C integrator library models in SIMSIDES

Model name	Circuit effects included
`MOSFET_C_CTInt_1pole`	OTA output swing limitation, finite OTA DC gain, parasitic capacitances, capacitance voltage coefficient, one-pole dynamic, thermal noise.
`MOSFET_C_CTInt_2poles`	OTA output swing limitation, finite OTA DC gain, parasitic capacitances, capacitance voltage coefficient, two-pole dynamic, thermal noise.
`MOS...1pole&Large_signal_distortion`	OTA output swing limitation, output current limit, finite OTA DC gain, parasitic capacitances one-pole dynamic, thermal noise.
`MOS...2poles&Large_signal_distortion`	OTA output swing limitation, output current limit, finite OTA DC gain, parasitic capacitances two-pole dynamic, thermal noise.

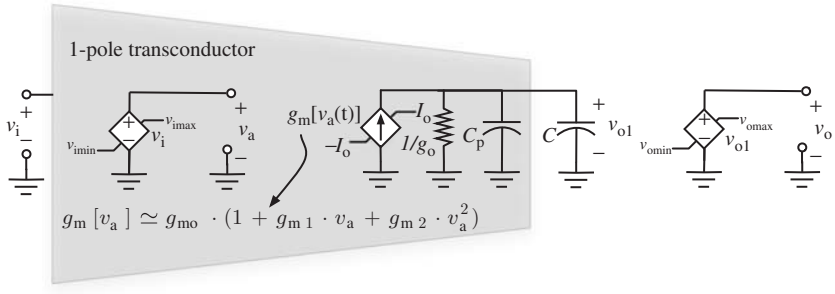


Figure B.13 One-pole Gm-C integrator model used in SIMSIDES.

`Transconductance of the OTA` and `Integration Capacitor`, define, respectively, the transconductance g_m and the integration capacitor C of the Gm-C integrator, the ITF of which is obtained by replacing $g = g_m/C$ in Equation (B.1).

`[Upper,Lower] bound saturation voltage` defines the maximum and minimum values of the output voltage v_o, as illustrated in Figure B.13.

`Input Voltage [Upper, Lower] saturation limit` defines the maximum and minimum values of the input voltage, denoted as v_{imax} and v_{imin}, respectively, in Figure B.13.

`[Second, Third] order distortion coefficient` represents the second- and third-order nonlinear transconductance coefficients $g_{m(1,2)}$, where it is assumed that the transconductance depends on the Gm-C integrator input voltage v_i as,

$$g_m \simeq g_{mo} \cdot (1 + g_{m1} \cdot v_i + g_{m2} \cdot v_i^2) \tag{B.9}$$

with g_{mo} being the nominal value of the transconductance. The maximum current provided by the transconductor is defined as I_o.

`DC voltage gain` represents the finite OTA DC gain, defined as g_m/g_o, with g_o being the output conductance of the Gm-C integrator.

`Integration constant time error`, defined as C_p/C, with C_p being the parasitic capacitance at the output of the Gm-C integrator.

`High-frequency pole` that defines the value of the high-frequency pole when a two-pole dynamic model is considered.

Gm-MC Integrators

SIMSIDES contains a library of Gm-C integrators based on the connection of a transconductance element and a Miller capacitance, also referred to as Gm-MC integrators. Figure B.14a shows the conceptual schematic of the Gm-MC integrators modeled in SIMSIDES. Several models that account for the effect of different circuit nonidealities are included in this library. The most accurate one, named `Gm_MC_CTInt_1poleSmall_Signal_Distortion`, is modeled by the equivalent circuit shown in Figure B.14b that involves a two-pole dynamic model. In addition to the model parameters included in Gm-C integrators, Gm-MC integrator models use the additional parameters described below.

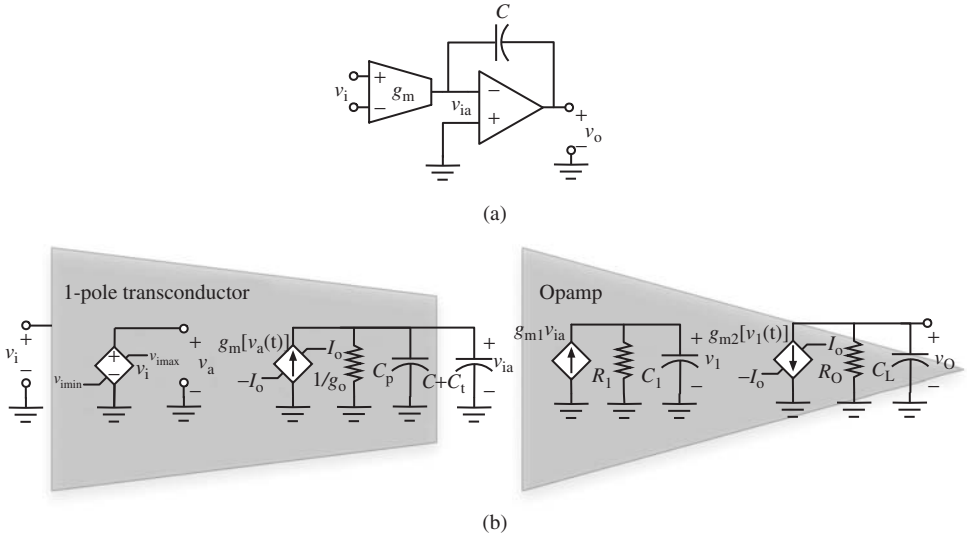


Figure B.14 Two-pole Gm-MC integrator model used in SIMSIDES: (a) conceptual schematic and (b) equivalent circuit of the transconductor and the opamp.

`Output transconductor parasitic, Integration capacitor ratio (Cp/C)` that represents C_P/C (see Figure B.14b).

`Output Operational parasitic, Integration capacitor ratio (CI/C)`; that is, C_L/C.

`Operational parasitic Output, OTA output capacitor ratio (Ct/C)`; that is, C_t/C.

`Transconductor and Op. Amplifier Unity gain frequency (Hz) [Gb1, Gb2]`, that represent the GB of the transconductor and of the opamp in Figure B.14b, respectively.

`High-frequency pole`, that defines the value of the high-frequency (nondominant) pole when a two-pole dynamic is considered—given by $1/(R_1C_1)$ in Figure B.14b.

`Origin transconductance`, that refers to the operating-point transconductance g_{mo} of the nonlinear characteristic given in Equation (B.9).

Active-RC Integrators

Among all active-RC integrator models listed in Table B.9, the most accurate and complete one is named `RC_CTInt_2poles&Large_signal_distortio`. Figure B.15 shows the conceptual schematic (Figure B.15a) and its corresponding equivalent model (Figure B.15b). There are two versions of this model: one based on a linear input resistance R and another in which R is a nonlinear function of the input voltage given by,

$$R(v_i) \simeq R \cdot (1 + R_1 \cdot v_i + R_2 \cdot v_i^2) \tag{B.10}$$

where $R_{(1,2)}$ represents the first- and second-order nonlinear coefficients.

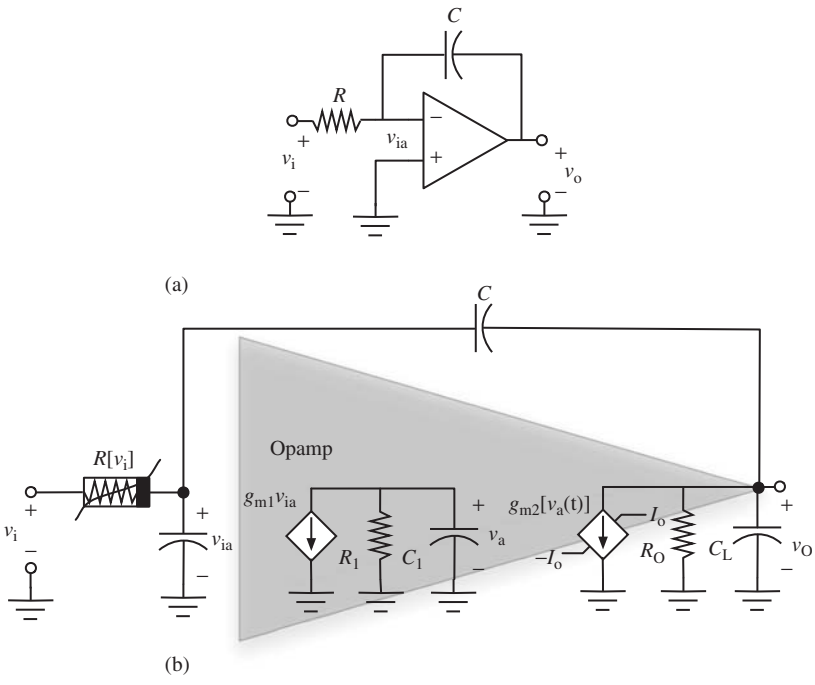


Figure B.15 Two-pole active-RC integrator model used in SIMSIDES: (a) conceptual schematic and (b) equivalent circuit.

Apart from the model parameters used by the CT building blocks described in previous sections, the following parameters are used in active-RC integrator models:

`Output resistance—integration resistance ratio`, that represents R_o/R in Figure B.15b.

`Integrator Ideal Unity gain frequency (RC)`, defined as $1/(RC)$.

`Opamp ideal Unity gain frequency`, that is the GB of the opamp in Figure B.15b.

`High-frequency pole (Hz)`, defined as $1/(R_1C_1)$ (see Figure B.15b).

MOSFET-C Integrators

In addition to active-RC integrator models, SIMSIDES includes also a library of MOSFET-C integrators the conceptual schematic of which is shown in Figure B.16. Essentially, these building blocks are the same as those used for modeling active-RC integrators, except that the integrator resistance R is replaced by a MOSFET transistor.

B.5.2 Real CT Resonators

Tables B.11 and B.12 list all models included in CT resonator libraries shown in Figure B.12, together with a brief description of their nonideal effects. These libraries

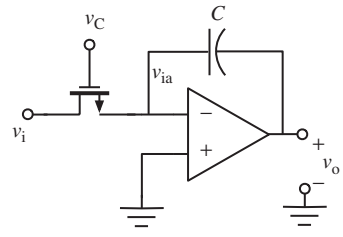


Figure B.16 Conceptual schematic of a MOSFET-C integrator similar to that modeled in SIMSIDES by the building blocks listed in Table B.10.

Table B.11 Gm-C resonator library models in SIMSIDES

Model name	Circuit effects included
`Ideal_gmC_CT_Resonator`	Ideal Gm-C resonator.
`gmC_CT_Res_1pole`	Finite OTA DC gain, time-constant error, one-pole dynamic, thermal noise.
`gmC_CT_Res_2poles` `gmC_CT_Res_2polesfull`	Finite OTA DC gain, time-constant error, two-pole dynamic, thermal noise.
`gmC_CT_Res_1pole_larged`	Input/output saturation voltage, output current limit, finite OTA DC gain, time-constant error, one-pole dynamic.
`gmC_CT_Res_2poles_larged`	Input/output saturation voltage, output current limit, finite OTA DC gain, nonlinear transconductance, time-constant error, two-pole dynamic.
`gmC_CT_Res_1pole_small&larged`	Input/output saturation voltage, output current limit, finite OTA DC gain, nonlinear transconductance, time-constant error, one-pole dynamic.
`gmC_CT_Res_2poles_small&larged`	Input/output saturation voltage, output current limit, finite OTA DC gain, nonlinear transconductance, time-constant error, two-pole dynamic.

include different building blocks that are classified according to the accuracy of their models as well as to the circuit nonidealities that are taken into account. As an illustration, Figure B.17 depicts both CT resonator libraries included in SIMSIDES, namely Gm-C resonators and Gm-LC resonators. Figure B.18 shows the conceptual schematic of both kinds of CT resonators. The model parameters involved in the building blocks shown in Figure B.17 are detailed here.

Table B.12 Gm-LC resonator library models in SIMSIDES

Model name	Circuit effects included
`Ideal_gmLC_CT_Resonator`	Ideal Gm-LC resonator.
`gmLC_CT_Res_1pole`	Input/output saturation voltage, inductance quality factor and series parasitic resistance, finite OTA DC gain, time-constant error, one-pole dynamic, thermal noise.
`gmLC_CT_Res_2poles`	Input/output saturation voltage, inductance quality factor and series parasitic resistance, finite OTA DC gain, time-constant error, two-pole dynamic, thermal noise.
`gmLC...1pole_large_dist`	Input/output saturation voltage, inductance quality factor and series parasitic resistance, output current limitation, finite OTA DC gain, time-constant error, one-pole dynamic, thermal noise.
`gmLC...2poles_large_dist`	Input/output saturation voltage, inductance quality factor and series parasitic resistance, output current limitation, finite OTA DC gain, time-constant error, two-pole dynamic, thermal noise.
`gm...1pole_small&large_dist`	Input/output saturation voltage, inductance quality factor and series parasitic resistance, output current limitation, nonlinear transconductance, finite OTA DC gain, time-constant error, one-pole dynamic, thermal noise.
`gmLC...2poles_small&larged`	Input/output saturation voltage, inductance quality factor and series parasitic resistance, output current limitation, nonlinear transconductance, finite OTA DC gain, time-constant error, two-pole dynamic, thermal noise.

Gm-C Resonators

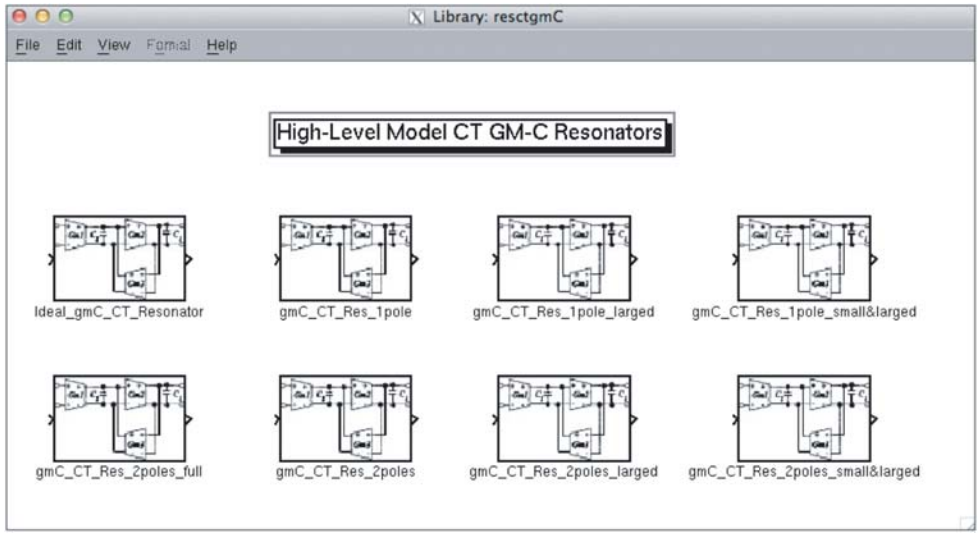
Apart from those parameters used in Gm-C integrators, the following model parameters are used in the Gm-C resonator model library:

`Transconductance of the first, second, and third OTA (gm1,gm2,gm3)`; that is, g_{m1}, g_{m2}, and g_{m3} in Figure B.18a.

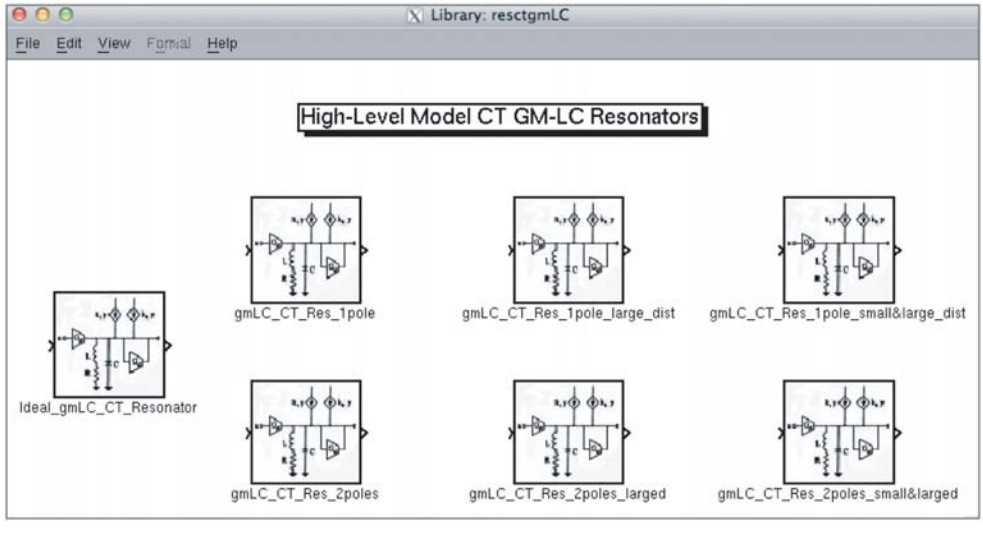
`Capacitors`; that is, C_1, C_2 in Figure B.18a.

`Nonlinear transconductance coefficients [gmnl1, gmnl2]`, that represents nonlinear coefficients $g_{m(1,2)}$ in Equation (B.9).

`DC Gain of OTAs [Av1,Av2,Av3]`, that defines the finite OTA DC gain of transconductors in Figure B.18a.



(a)



(b)

Figure B.17 CT resonator libraries included in SIMSIDES: (a) Gm-C resonators and (b) Gm-LC resonators.

`Percentual integration constant time error [et1, et2];` that is, the time constant errors associated with both feed-forward transconductances in Figure B.18a, given by,

$$\epsilon_{t1} = \frac{C_{p1} + C_{p3}}{C_1} \quad , \quad \epsilon_{t2} = \frac{C_{p2}}{C_2} \tag{B.11}$$

where C_{pi} represents the parasitic capacitance of the ith transconductance in Figure B.18a.

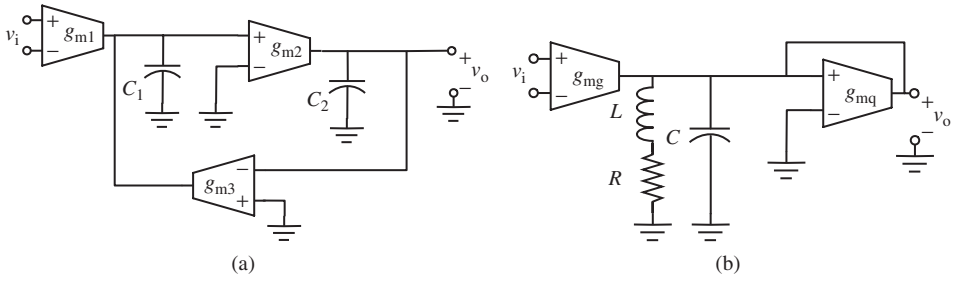


Figure B.18 Conceptual schematics of the CT resonators modeled in SIMSIDES: (a) Gm-C resonator and (b) Gm-LC resonator.

Gm-LC Resonators

The main model parameters used by Gm-LC resonator blocks are described in this section. Essentially, these parameters deal with the inductor element in Figure B.18b and its associated resonant frequency.

`Frequency resonance`; that is, the resonant frequency of the Gm-LC resonator.

`Inductor Q`; that is, the Q-factor of the inductor in Figure B.18b.

`Series Resistance`; that is, the parasitic resistance R of the inductor.

The rest of the model parameters used in Gm-C resonators have the same meaning as those used in Gm-C integrators and resonators.

B.6 Real Quantizers and Comparators

Table B.13 lists the building blocks included in the real `Quantizers&Comparators` SIMSIDES library, together with a brief description of their operation and main circuit nonidealities. In addition to the ideal parameters described in Section B.2.3, additional model parameters are required to model the different circuit nonidealities. These error parameters are listed in Table B.14.

Note that apart from comparators and quantizers, there is a building block named `Real_Sampler` that is used for modeling the S&H circuits that are connected at the input of embedded quantizers in CT-$\Sigma\Delta$Ms. One of the most critical errors associated with this building block is the clock jitter. It is modeled as an uncertainty in the sampling time δt corresponding to a stationary process with zero mean and standard deviation defined by the user (see Table B.14).

B.7 Real D/A Converters

Table B.15 lists the different building blocks included in the real `D/A Converters` SIMSIDES library, together with a brief description of their operation and main circuit errors.

Error parameters associated with the models listed in Table B.15 have the same meaning as those used in multibit quantizers, except for the selectable NRZ/RZ/HRZ DAC

Table B.13 Real Quantizers and Comparator models included in SIMSIDES

Model name	Circuit effects included
`Real_Comparator_Offset&Hysteresis`	Voltage-mode comparator with offset, (random & deterministic) hysteresis.
`Real_Comparator_Offset&Hysteresis_for_SI`	Current-mode comparator with offset and nonlinearity (INL).
`Real_Multibit_Quantizer`	Voltage-mode multibit quantizer with gain error, offset (random & deterministic) hysteresis.
`Real_Multibit_Quantizer_for_SI`	Current-mode multibit quantizer with gain error, offset, INL, (random & deterministic) hysteresis.
`Real_Multibit_Quantizer_dig_level_SD2`	Voltage-mode multilevel quantizer with gain error, offset, INL, (random & deterministic) hysteresis.
`Real_Sampler`	Sampling & Hold circuit with clock jitter error.

Table B.14 Error model parameters used in SIMSIDES Real Quantizers

Parameter name (in alphabetical order)	Brief description
`Gain Error in LSB`	Gain error measured in LSB.
`Jitter typical deviation`	Standard deviation of clock jitter error.
`Kind of Hysteresis`	Comparator hysteresis. It may be either deterministic or random hysteresis.
`INL in LSB`	Integral Nonlinearity error measured in LSB.
`Number of levels`	Number of quantizer levels.
`Offset`	Offset error.
`Offset Error in LSB`	Offset error measured in LSB.
`Seed for random jitter generation`	Seed number used for generating random jitter error.

waveform and the delay error. The latter can be chosen to be either a constant delay or a signal-dependent delay, given by,

$$\text{delay}(v_i) = \text{d0} + \frac{\text{d1}}{\text{x1} \cdot |v_i|} < \text{dmax} \tag{B.12}$$

where `d0, d1, x1, dmax` are model parameters set by the user.

Table B.15 Real DAC models included in SIMSIDES

Model name	Circuit effects included
`Real_DAC_Multibit`	Voltage-mode multibit DAC with offset, gain error, and INL error.
`Real_DAC_Multibit_SI`	Current-mode multibit DAC with offset, gain error, and INL error.
`Real_DAC_Multibit_delay_Jitter`	Voltage-mode multibit DAC with offset error, gain error, INL error, delay error, and clock jitter error.
`Real_DAC_Multibit_delay_Jitter_SI`	Current-mode multibit DAC with finite output conductance, offset error, gain error, INL error, delay error, and clock jitter error.
`Real_DAC_pulse_types`	Voltage-mode multibit DAC with selectable NRZ/RZ/HRZ output waveform.
`Real_DAC_Multibit_pulse_types`	Voltage-mode multibit DAC with selectable NRZ/RZ/HRZ output waveform, gain error, offset error, and INL error.
`Real_DAC..._delay_jitter`	Voltage-mode multibit DAC with selectable NRZ/RZ/HRZ output waveform, gain error, offset error, INL error, delay error, and clock jitter error.

B.8 Auxiliary Blocks

In addition to the building blocks described in previous sections, SIMSIDES includes a library named `Auxiliary blocks` that contains some other blocks (such as adders, DEM algorithms, and digital latches) also needed to simulate ΣΔMs. Table B.16 lists the models included in the mentioned library together with a brief description of their operation. The most significant parameters used by these models are listed in Table B.17.

As an illustration on the use of some of the auxiliary blocks, Figure B.19a shows the SIMSIDES block diagram of a second-order feed-forward SC-ΣΔM, which includes an embedded 16-level quantizer and DAC with a selectable DEM algorithm.

The block diagram in Figure B.19a includes the following building blocks:

- `Mux_SD2`. This block samples the modulator input signal in a number of unit capacitors, which corresponds to the number of unit elements used in the multilevel DAC minus one. As shown in Figure B.19b, the only model parameter of this block is named `Number of elements`, which equals 15 in this example.
- `SC_FE_Int_All_Effects_DEM_SD2`. This block is used for modeling the front-end integrator and includes all error mechanisms with values defined in the block dialogue box in Figure B.19c. This model includes also a parameter named `Array`

Table B.16 Auxiliary building-block models used in SIMSIDES

ANALOG ADDERS	
Model name	**Brief description**
`Analog_Adder_Ideal_SD2`	Ideal SC passive adder with parasitic input capacitance and load capacitance.
`Analog_Adder_real_SD2`	Real SC passive adder with parasitic input capacitance and load capacitance, switch on-resistance, settling error, capacitor nonlinearity, and thermal noise.
DIGITAL ADDERS	
Model name	**Brief description**

`Dig_add_generic_2outs`
`Dig_add_3L_5L_13L`
`Dig_add_3L_3L_5L_2outs`
`Dig_add_3L_3L_7L_2outs`
`Dig_add_3L_5L_9L_2outs`
`Dig_add_3L_5L_13L_2outs`
	Digital subtraction of a M_1-Level thermometric-coded signal and a M_2-Level thermometric-coded signal, which is scaled by a factor of `d`. The result is a $(M_1 + M_2/\mathrm{d})$-level thermometric-coded digital output.

DIGITAL LATCHES	
Model name	**Brief description**
`D_latch_simplest` `D_latch`	Digital "D" latches.
DAC WITH DEM ALGORITHMS	
Model name	**Brief description**
`DEM_id_SD2`	Ideal DEM algorithm.
`DAC - DEM - V04`	DAC block with a selectable DEM algorithm. There are three options: No DEM, DWA, Pseudo-DWA.
`Mux_SD2`	Building block used for sampling an input (analog) signal by a number of different branches corresponding to the number of DAC unit capacitors.

`of sampling capacitors for DEM branch`. The number of capacitors in the array must be exactly the same as the number of DAC levels, that is, the number of quantization levels. As an illustration, Figure B.20 shows the MATLAB code used for generating different alternative capacitor arrays together with other parameters used for simulating the block diagram in Figure B.19a. Note that this capacitor array must include also the DAC element mismatch, defined as a Gaussian probability distribution.

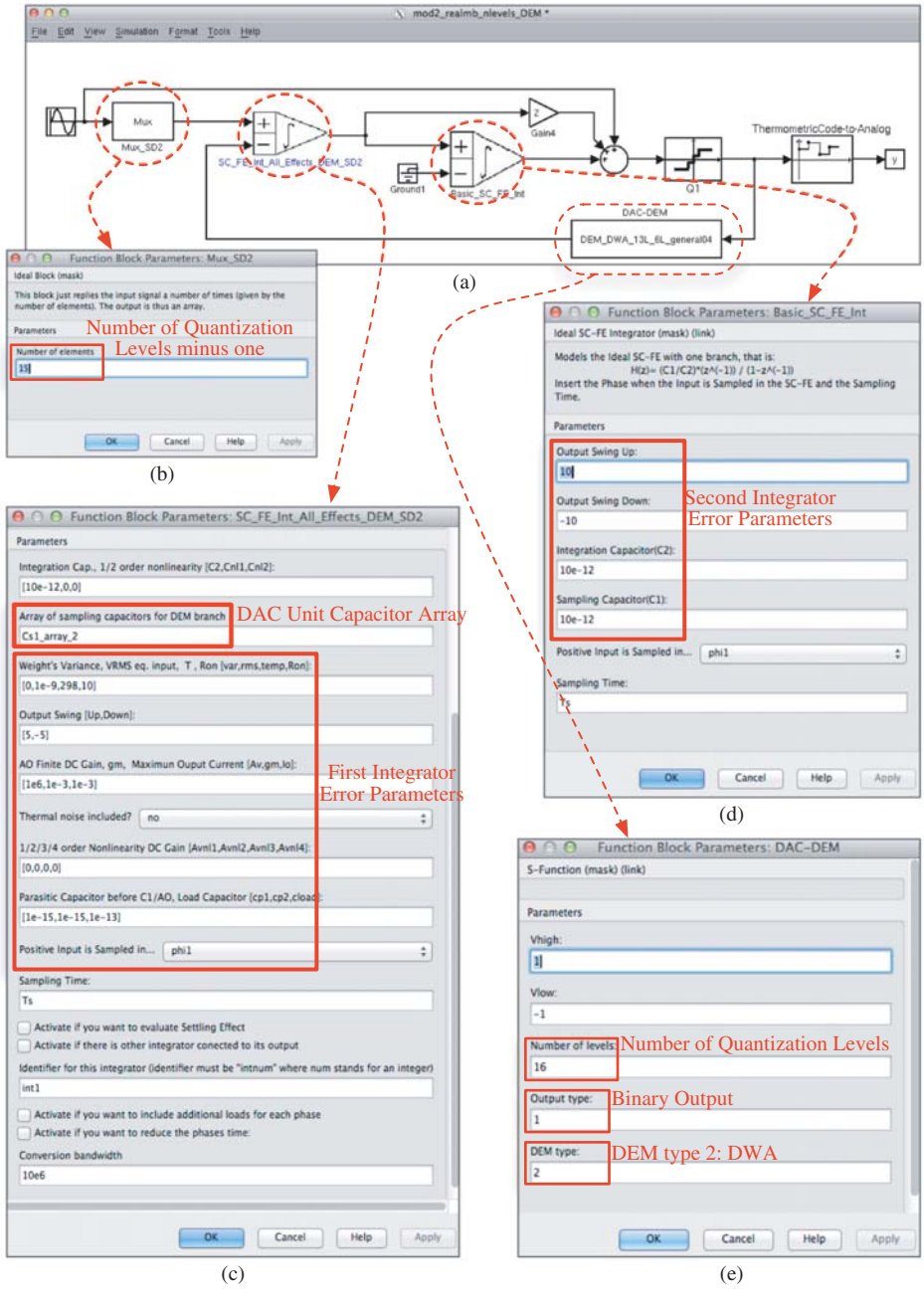


Figure B.19 Example of a second-order feed-forward SC-ΣΔM with 16-level quantization and DEM: (a) SIMSIDES block diagram, (b) Mux_SD2 dialogue box, (c) first integrator block dialogue box, (d) second integrator block dialogue box, and (e) DAC-DEM block dialogue box.

Table B.17 Error model parameters used in SIMSIDES Auxiliary Blocks

Parameter name (in alphabetical order)	Brief description
`Comparator Input Capacitor (C)`	Parasitic capacitance at the comparator/quantizer input.
`DEM type`	DEM algorithm: (1) No DEM, (2) DWA, (3) Pseudo-DWA. (Default = 1)
`Input Capacitor (C)`	Input capacitance of the analog adder.
`Nonlinearities of the capacitors`	Capacitance nonlinear coefficients in an analog adder.
`Number of elements`	Number of DAC unit elements.
`Output type`	Digital output code: (1) Binary output, (2) Trilevel output including common mode (Default = 1).
`Time interval between sampling and comparison (delta)`	Delay between the time instant when the adding operation is performed and the time instant when comparison time takes place.

```
%%%%%%%%%% Modulator: simulation parameters %%%%%

fs=35.2e6; % Sampling frequency
Ts=1/fs; % Sampling time
fi=137.5e3; % Input frequency
OSR=64; % OverSampling Ratio
N=65536; % Number of Points

%%%%%%%%%% DEM Parameters and Capacitor Feedback Arrays %%%%%%%%%%

nlevels=16; % Number of quantization levels
Cs1=10e-12; % Sampling Capacitor
sigma=[1/100]; % Capacitor mismatch standard deviation

%% Capacitor array

n_elements_1=(nlevels-1);
% Binary array
n_elements_2=floor((nlevels)/2);
% Tri-level array
Cs1_array_1=Cs1/n_elements_1*(ones(1,n_elements_1));
% No mismatch - binary
Cs1_array_2=Cs1/
n_elements_1*(ones(1,n_elements_1)+sigma*randn(1,n_elements_1));
% With mismatch - binary
Cs1_array_3=Cs1/
n_elements_2*(ones(1,n_elements_2)+sigma*randn(1,n_elements_2));
% With mismatch - tri_level
Cs1_array_4=Cs1/n_elements_2*(ones(1,n_elements_2));
% No mismatch - tri_level
```

Figure B.20 MATLAB code used for defining capacitor arrays and DEM parameters of Figure B.19e.

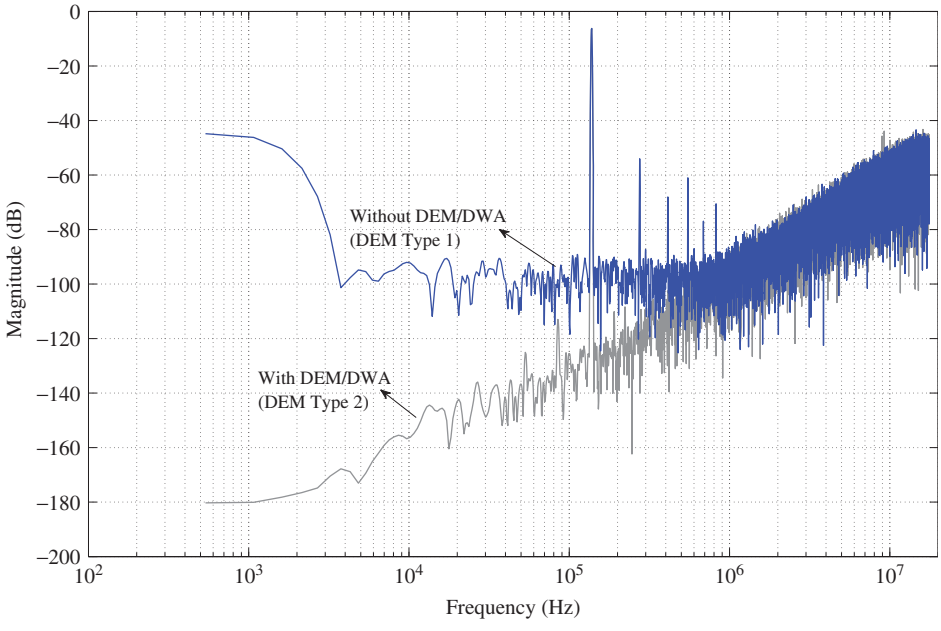


Figure B.21 Effect of DEM on the output spectrum of the ΣΔM of Figure B.19a.

- `Basic_SC_FE_Int`. This block is used for modeling the second integrator, considering only ideal values of the output swing and capacitors, as shown in Figure B.19d.
- `Real_Multibit_Quantizer_dig_level_SD2`. This block, named `Q1` in the example, is used for modeling the quantizer considering the nonideal effects listed in Table B.13. In this example, these nonideal effects have not been taken into account. The output of this block is a thermometric-coded bit array. This array is transformed into an analog signal for further processing by using the block named `ThermometricCode-to-Analog`.
- `DAC-DEM`. This block uses the `DAC - DEM - V04` model (see Table B.16). The most important model parameters of these block are highlighted in Figure B.19e.

As an illustration, Figure B.21 shows the output spectra of the ΣΔM in Figure B.19a considering a capacitor mismatch of 1%. It is clear how the performance of the modulator becomes severely degraded by DAC nonlinearity if DEM is not activated (`DEM type 1`).

Index

CMOS Sigma-Delta Converters: Practical Design Guide, First Edition. José M. de la Rosa and Rocío del Río.