MATLAB/SIMULINK-Based High-Level Synthesis of Discrete-Time and Continuous-Time ΣΔ Modulators

J. Ruiz-Amaya, J.M. de la Rosa, F. Medeiro, F.V. Fernández, R. del Río, B. Pérez-Verdú and A. Rodríguez-Vázquez
Instituto de Microelectrónica de Sevilla – IMSE-CNM (CSIC)
Edificio CICA-CNM, Avda. Reina Mercedes s/n, 41012- Sevilla, SPAIN

Phone: +34 95 5056666, Fax: +34 95 5056686, E-mail: ruiz@imse.cnm.es

Abstract

This paper describes a tool that combines an accurate SIMULINK-based time-domain behavioural simulator with a statistical optimizer for the automated high-level synthesis of ΣΔ Modulators (ΣΔMs). The combination of high accuracy, short CPU time and interoperability of different circuit models together with the efficiency of the optimization engine makes the proposed tool an advantageous alternative for ΣΔM synthesis. The implementation on the well-known MATLAB/SIMULINK platform brings numerous advantages in terms of data manipulation, flexibility and simulation with other electronic subsystems. Moreover, this is the first tool dealing with the synthesis of ΣΔMs using both Discrete-Time (DT) and Continuous-Time (CT) circuit techniques(*).

1. Introduction

ΣΔ Analog to Digital Converters (ADCs) have demonstrated to be an attractive solution for the implementation of analog-digital interfaces in *systems-on-chip*. Compared to Nyquist-rate ADCs, ΣΔ architectures present a better performance in terms of resolution, speed and power consumption together with a high robustness against the unavoidable circuit parasitics and tolerances [1][2][3]. However, the need to design high-performance ΣΔ ADCs in adverse digital technologies together with the vertiginous rate imposed by the technology evolution has motivated the interest for CAD tools which can optimize the design procedure in terms of efficiency and short time-to-market. For this purpose, several tools for oversampling converter synthesis have been reported in the last years [1][4][5][6][7]. These tools use different synthesis strategies that can be roughly classified into two main categories [1][8]:

- *Knowledge-based synthesis-tools,* which are based on capturing the knowledge of experienced designers [4][5]. Although the execution times are very short, the results still must be optimized because design procedures are usually based on approximate equations and very simple models. Additionally, they are closed tools, e.g., limited to a reduced number of topologies and the addition of new ones is a very costly process and usually restricted to the tool developers.
- *Optimization-based synthesis tools* [1][6][7], which are based on an iterative optimization procedure with a performance evaluator in the loop. Performance evaluation can be done by means of equations or simulations. In the latter case, the characteristics of the simulator determine the accuracy and openness of the tool.

Most modern approaches [1][6][7] belong to the second class of tools. Fig.1 shows the conceptual block diagram of a conventional optimization-based synthesis tool. The design process of a ΣΔM starts from the high-level modulator specifications (resolution, signal bandwidth, etc.). The objective is to get the building block specifications (design parameters) that optimize the performance of the modulator; that is, those specifications which satisfy the modulator-level specifications with the minimum power consumption and silicon area. At each iteration of the optimization procedure, circuit performances are evaluated at a given point of the design parameter space. According to such an evaluation, a movement in the design parameter space is generated and the process is repeated again.

The iterative nature of the optimization procedure requires a very efficient mechanism for performance evaluation. Event-driven ΣΔ behavioral simulation is used in the synthesis approaches in [1][6][7]. This technique enables very efficient analysis while providing high accuracy levels.

In these tools, both, simulation engine and models, are implemented using a programming language like C. Modulator libraries are usually available, containing a limited

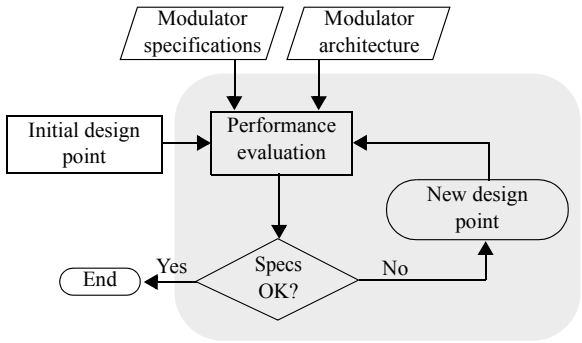

Fig. 1: Conceptual block diagram of an optimization-based ΣΔM synthesis tool.

(*)This work has been supported by the EU ESPRIT IST Project 2001-34283/TAMES-2 and the Spanish CICYT Project TIC2001-0929/ADAVERE.

number of architectures. Although a text or graphical interface is usually provided to create new architectures, block models cannot be easily changed. On the other hand, the possible circuit techniques used to implement the modulators are constrained by the capabilities of the simulation engine and the available block models. For this reason, the synthesis tools in [1][6][7] are limited to discrete-time Switched Capacitor (SC) ΣΔ modulators.

To overcome these problems the proposed ΣΔ synthesis Tool has been implemented using the MATLAB/SIMULINK platform [9][10]. The embedded behavioural simulator is able to efficiently evaluate the performances of LowPass (LP) or BandPass (BP) ΣΔMs implemented using either SC, SwItched Current (SI) or CT circuit techniques. This enables the synthesis tool to deal with all those types of ΣΔMs.

The implementation on the MATLAB/SIMULINK platform provides a number of advantages: (a) it is a widely used platform, familiar to a large number of engineers, whereas special-purpose tools [6][7] require to learn a proprietary text-based or graphical interface; (b) it has direct access to very powerful tools for signal processing and data manipulation; (c) it has complete flexibility to create new ΣΔM architectures, and even to include different blocks, either continuous-time or discrete-time; and (d) it enables a high flexibility for the extension of the block library whereas adding new blocks or models to existing libraries in previous tools requires the qualified contribution of a programmer.

A complete list of non-idealities of the building blocks for all circuit techniques (SC, CT and SI) have been considered [1][2][3]. The price to pay for the implementation of the behavioral models of the ΣΔM blocks (integrators, quantizers, etc.) using elementary SIMULINK library blocks is a high computational cost. To palliate this problem the behavioral models are incorporated as SIMULINK S-functions [11]. This approach decreases the computational cost to acceptable levels for synthesis purposes.

The optimization tool at the core of the synthesis toolbox contains adaptive statistical optimization techniques for wide space exploration and deterministic techniques for fine tuning [1]. Besides, the addition of knowledge about specific architectures has been enabled. Such knowledge can be coded using a standard programming language: C or C++, is compiled at run-time and incorporated into the optimization process. This makes this synthesis toolbox an optimization-based synthesis tool but with the appealing features of knowledge-based systems.

This paper is organized as follows. The characteristics of the optimization core are briefly described in Section 2. Section 3 is devoted to the performance evaluator: the behavioural simulator SIMSIDES. Finally, Section 4 gives several simulation and synthesis examples of the proposed toolbox.

2. Optimization core

The presented synthesis tool uses an optimization core for design parameter selection as described above. Deterministic optimization methods, like those available in the MATLAB standard distribution [9], are not suitable because initially we may have little or no idea of an appropriate design point. Therefore, the optimization procedure is quickly trapped in a local minimum.

For this reason, we developed an optimizer which combines an adaptive statistical optimization algorithm inspired in simulated annealing (local minima of the cost function can then be avoided) with a design-oriented formulation of the cost function (which accounts for the modulator performances).

Fig.2 shows the flow diagram of the optimizer where starting from a modulator topology, e.g., a modulator whose design parameters (building block specifications) are not known and arbitrary initial conditions, a set of design parameter perturbations is generated. With the new design parameters, a set of simulations are performed to evaluate the modulator performance. From the simulation results, it automatically builds a cost function (that has to be minimized). The type and value of the perturbations as well as the iteration acceptance or rejection criteria depend on the selected optimization method. The optimization process is divided into two steps:

- The first step explores the design space by dividing it into a multi-dimensional coarse grid, resulting in a mesh of hypercubes (*main optimization*). A statistical method is usually applied in this step.
- Once the optimum hypercube has been obtained, a final optimization is performed inside this hypercube (*local optimization*). A deterministic method is usually used in this step.

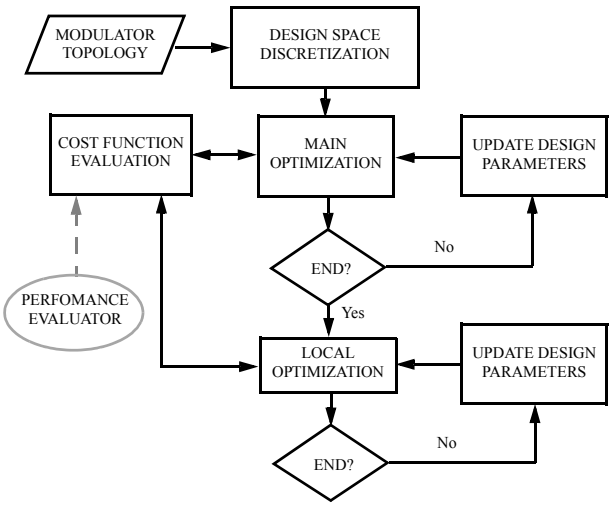

Fig. 2: Operation flow of the optimization core.

In addition, the optimization core is very flexible, in so far as the cost function formulation is very versatile: multiple targets with several weights, constraints, dependent variables, and logarithmic grids are permitted. This optimization procedure has been extensively tested with design problems involving behavioural simulators as well as electrical simulators [1][7].

This optimizer has been integrated in the MATLAB/SIMULINK platform by using the MATLAB engine library [9], so that the optimization core runs in background while MATLAB acts as a computation engine.

3. Behavioural simulator: SIMSIDES

As described in Section 1, the proposed synthesis tool uses a simulator as a performance evaluator. ΣΔMs are strongly non-linear circuits and, therefore, evaluation of their main performance specifications has to be carried out in the time-domain. Due to the oversampling nature of ΣΔMs, this means that long transient simulations (several thousands of clock cycles) are necessary to evaluate their main figures of merit.

Transistor-level simulations using SPICE-like simulators lead to excessive CPU times (typically from days to weeks [12]), because they base its analysis in numerical integration, with small integration steps and complex device models. To overcome this problem, different alternatives have been developed, which at the price of losing some accuracy in their results, reduce the simulation time [1]. One of the best accuracy-speed trade-offs is achieved by using the so-called *behavioural simulation* technique using functional models [6][7][13][14]. This approach requires that the circuit can be partitioned into basic blocks with independent functionality. This implies that an instantaneous block output cannot be related to itself, that is, either there is no global feedback loop, or, in case such a loop exists, there is a delay that avoids the instantaneous dependence. These blocks are described by equations that relate the outputs in terms of the inputs and the internal state variables. Thus, the accuracy of the simulation depends on how precisely those equations describe the real behaviour of each block.

This has been the technique used in our simulator, called SIMSIDES (SIMulink-based SIgma-DElta Simulator). This simulator has been implemented as a toolbox in the MATLAB/SIMULINK platform. Modelling and simulation of ΣΔMs in this platform was first reported in [15][16], although limited to SC architectures. Although very intuitive, the implementation of the behavioral models of each basic building block requires several sets of elementary SIMULINK blocks. This means a penalty in computation time which may become critical in an optimization-based synthesis process in which hundreds or thousands of simulations must be executed. To overcome this problem, behavioural models in SIMSIDES have been incorporated in SIMULINK by using S-functions [11]. As a consequence, the CPU time for the time-domain simulation of a DT/CT ΣΔM involving 65536 samples is typically a few seconds†. Besides, SIMSIDES is able to deal with any circuit technique: SC, SI or CT.

SIMSIDES contains a set of ΣΔM block libraries which are classified according to the modulator hierarchy level and the circuit technique. The basic building blocks modelled in SIMSIDES, as well as its non-idealities are summarized in Table 1. A detailed description of these non-idealities and their behavioural models can be found in [1], [2] and [3] for SC, CT and SI circuits, respectively.

Fig.3 shows the general structure of SIMSIDES. First, the modulator architecture is defined by connecting the building blocks of the SIMSIDES libraries. After the circuit diagram is created, the user sets some parameters and options which are taken into account by the tool to perform the time-domain simulation. Monte-Carlo simulation as well as parametric analysis are also possible. Output data generated by simulation consists of time-domain series which can be processed to get different figures of merit. Thus, histograms and output spectra are computed using the routines provided by the signal processing toolbox of MATLAB. Other analyses such as Signal-to-Noise Ratio (*SNR*), harmonic or intermodulation distortion, are done using a collection of functions specifically developed for SIMSIDES.

The quality of the synthesis approach is not only given by its efficiency in terms of CPU time. It also critically depends on the accuracy of the performance evaluator. To illustrate

TABLE 1: Basic building blocks and non-idealities modelled in SIMSIDES.

Circuit technique	Block		Non-idealities
SC	Integrators	Opamps	Finite and non-linear gain, dynamic limitations (incomplete settling error, harmonic distortion), output range, thermal noise.
		Switches	Thermal noise, finite and non-linear resistance.
		Capacitors	Non-linearity, mismatching.
	Resonators		Non-idealities associated to the integrators.
SI	Integrators		Finite and non-linear gain, finite output and input conductance, dynamic limitations (incomplete settling, harmonic distortion, charge injection), thermal noise.
	Resonators		Feedback gain error, non-idealities associated to the integrators.
CT	Integrators		Finite and non-linear gain, dynamic limitations (parasitic capacitors, high and low frequency poles), thermal noise, output range and lineal input range, offset.
	Resonators		Non-idealities associated to the integrators.
ALL	Clock		Jitter.
	Comparators		Offset, hysteresis.
	Quantizers /DACs		Integral non-linearity, gain error, offset, jitter noise, delay time.

†. All simulations shown in this paper were done using a PC with an AMD XP2400 CPU@2GHz @512MB-RAM.

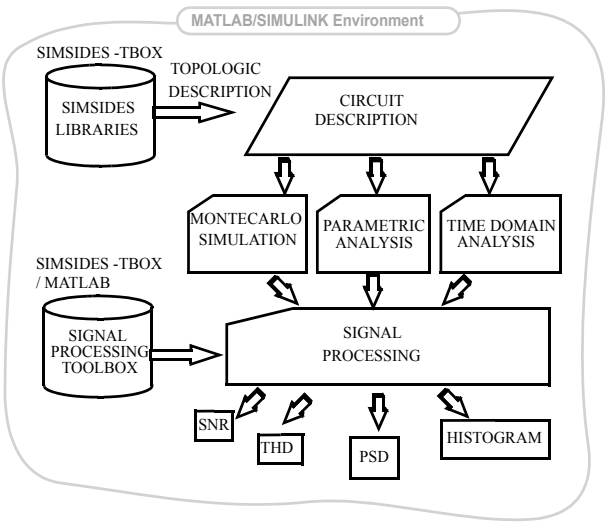

Fig. 3: Structure of SIMSIDES.

the accuracy achieved by the behavioural models in SIMSIDES, Fig.4 shows the output spectrum of a SC cascade 2-1-1 modulator intended for ADSL application [17]. The three plots correspond to a behavioural simulation using SIMSIDES, an electrical simulation using HSPICE (5 days of CPU time for 8192 samples) and real measurements taken from a chip prototype. A different signal frequency has been chosen for HSPICE for better comparison.

4. The ΣΔ Synthesis toolbox in operation

The proposed tool has been conceived as a MATLAB toolbox for the simulation and synthesis of ΣΔMs. The Graphical User Interface (GUI) included in the toolbox allows to navigate easily through all steps of the simulation, synthesis and post-processing of results. For illustration purposes, Fig.5 shows part of the toolbox GUI for architecture description.

By using this GUI, the user can either open an existing ΣΔM architecture or create a new one in the SIMULINK platform. When a simulation is finished, different performance figures such as output spectrum, in-band noise

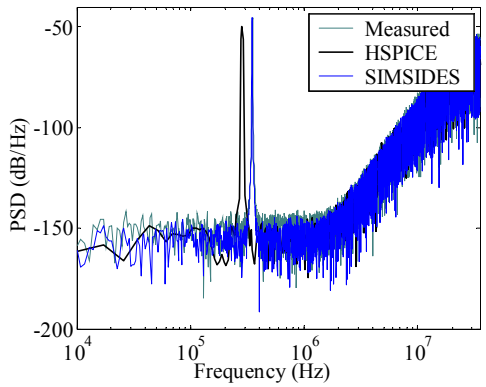

Fig. 4: Simulation of a SC cascade 2-1-1.

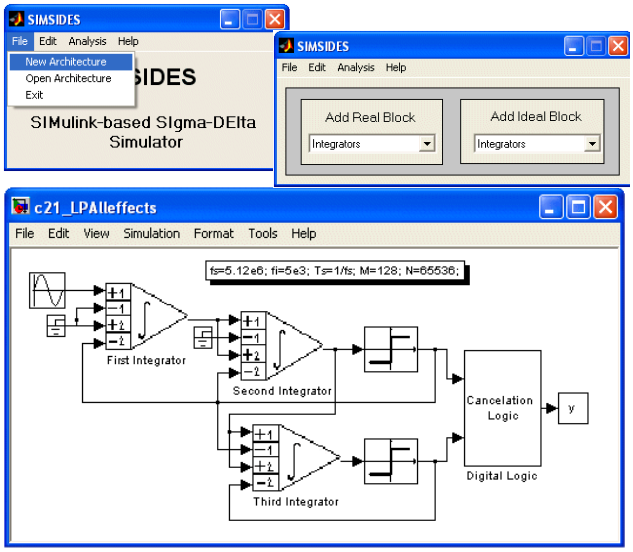

Fig. 5: Illustrating the GUI for editing modulator topologies of the ΣΔ Synthesis Toolbox.

power, harmonic distortion, etc. can be computed from the output data through the analysis/data processing menu. In addition, parametric analysis and MonteCarlo simulations can be performed.

High-level synthesis is started from the synthesis menu, where constraints, performance specifications, design parameters, optimization algorithms, etc., can be specified. Then, the optimization core starts the exploration of the design space to find out the optimum solution by using SIMSIDES results for performance evaluation.

To illustrate the simulation and synthesis capabilities of this toolbox two ΣΔM architectures have been selected:

- An SC 2-1 cascade single bit ΣΔM (SC 2-1 sb) (Fig.6(a)).
- A CT 5th-order LP ΣΔM (CT 5th-order LP) (Fig.6(b)).

One of the most important degrading factors in SC cascades ΣΔMs is the mismatch error. This is illustrated in the MonteCarlo simulation of Fig.7(a), where a random Gaussian mismatch error with zero mean and 0.5% standard deviation has been assumed. Each plot corresponds to a parametric analysis of the *SNR* versus the input amplitude. About 450 simulations with 32768 samples were done to obtain this figure and it only took 541s CPU time. Apart from mistmatch error, thermal noise can degrade considerably the resolution. Fig.7(b) illustrates the performance degradation of the 2-1 sb modulator caused by all thermal noise sources.

On the other hand, one of the major advantages of CT ΣΔMs lies in that they can achieve higher sampling frequencies. However, CT ΣΔMs degrade drastically their performance as a result of two important errors: clock jitter noise and delay time between the quantizer clock edge and DAC response. Fig.8(a) shows the impact of the delay on the output spectrum of the modulator of Fig.6(b). Two different cases

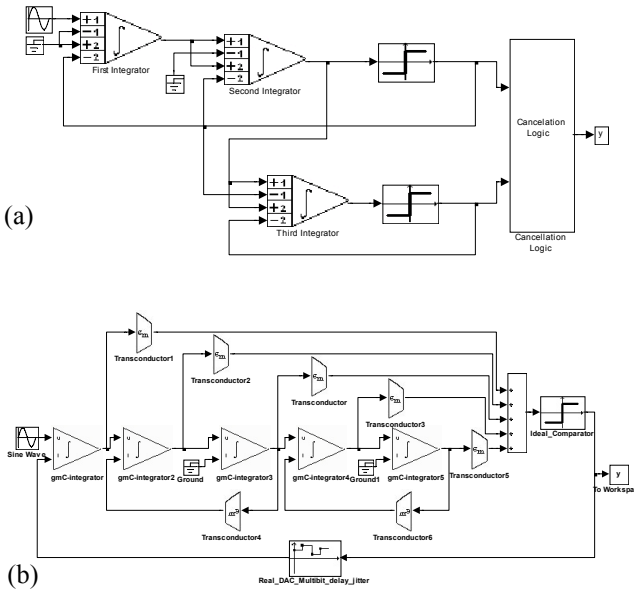

Fig. 6: Block diagrams of the (a) SC 2-1 sb ΣΔM and (b) CT 5th-order LP ΣΔM.

have been considered: a fixed delay, which is independent on quantizer input voltage magnitude; and a signal-dependent delay, which is practically constant for large quantizer input voltages, but rises for decreasing inputs [2]. Fig.8(b) illustrates the performance degradation caused by jitter noise

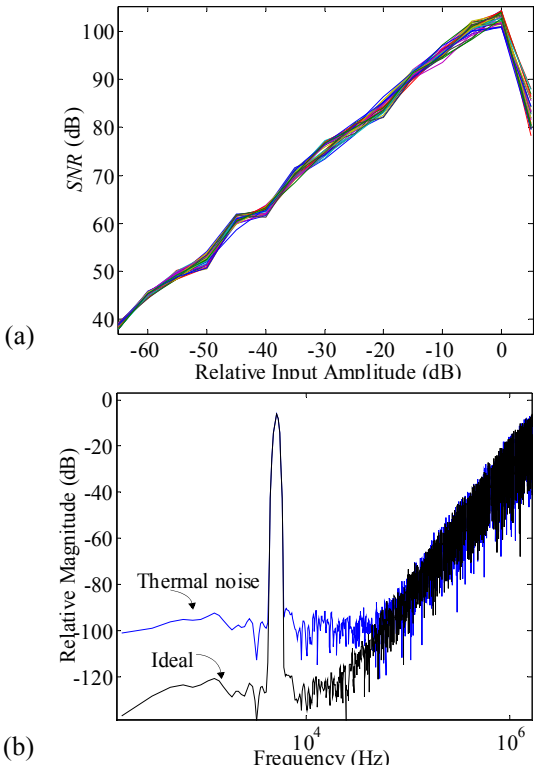

Fig. 7: Effect of (a) mismatch error and (b) thermal noise on SC 2-1 sb ΣΔM.

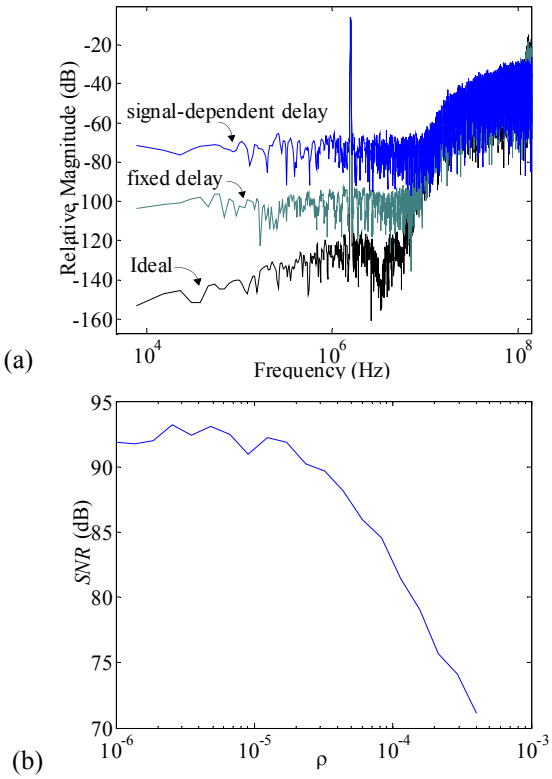

Fig. 8: Effect of (a) loop delay time and (b) clock jitter noise on CT 5th-LP ΣΔM.

where a random Gaussian jitter noise with zero mean and standard deviation $\sigma_{jit} = \rho T_s$ has been assumed.

To show the capabilities of the Synthesis Toolbox, the high-level sizing of the modulators in Fig.6 is performed. The modulator specifications are: 15bits@20kHz for the SC 2-1 sb ΣΔM and 12bits@6.25MHz for the CT 5th-order LP ΣΔM. The objective is to meet those specifications with the minimum power consumption and silicon area.Once design parameters, design specifications, and constraints have been specified through the toolbox GUI, a wide exploration of the design space is performed by the optimizer. At each point of the design space a SIMSIDES simulation is done to evaluate the modulator performances.

Table 2 and Table 3 show the results of the high-level synthesis for both modulators. The optimization procedures required 817 iterations for the SC 2-1 sb modulator and 674 iterations for the CT 5th-LPΣΔ taking 16.4 minutes and 52.1 minutes of CPU time, respectively. Finally, Fig.9 illustrates both output spectra and Signal-to-Noise plus Distortion Ratio *(SNDR)* corresponding to the high-level sizing provided by the synthesis toolbox.

Conclusions

A tool for the synthesis of CT and DT ΣΔMs in the MATLAB/SIMULINK environment has been described. Based

TABLE 2: High-level synthesis results for SC 2-1 sb ΣΔM.

OPTIMIZED SPECS FOR:		15bits@20kHz	
		Integ. I	Integ. II-III
Modulator	Sampling frequency (MHz)	5.12	
	Oversampling ratio	128	
Integrators	Sampling capacitor C_i(pF)	6	1.5
	Feed-back capacitor C_o(pF)	24	3
	MOS switch-ON resistance (kΩ)	≤ 0.84	≤ 1.7
Opamps	DC-gain (dB)	≥ 58.5	≥ 56
	DC-gain non-linearity (V^{-2})	≤ 22%	≤ 22%
	Output swing (V)	2.7	
	Input noise PSD (nV/sqrt(Hz))	≤ 8.1	≤ 278
	Output current (mA)	≥ 0.5	≥ 0.23
	Input transconductance (mA/V)	≥ 0.5	≥ 0.14
Comparators	Hysteresis (V)	≤ 0.2	
Technology	Cap. non-linearity (ppm/V^2)	≤ 89	

TABLE 3: Synthesis results for CT 5th-order LP ΣΔM.

OPTIMIZED SPECS FOR:		12bits@6.25MHz	
		Integ. I	Other Integ.
Modulator	Sampling frequency (MHz)	300	
	Oversampling ratio	40	
Transconductors	Transconductance (mA/V)	0.6	0.15
	DC-gain (dB)	≥ 34	≥ 42
	Parasitic output capacitor (pF)	≤ 0.66	≤ 0.04
	Input linear swing (V)	≥ 0.5	≥ 0.32
	HD3 (dB)	≤ –50	≤ –30
DAC	Clock jitter (ps)	≤ 0.5	
	Excess loop delay time (ns)	≤ 0.77	

on the combination of an accurate and efficient SIMULINK-based time-domain behavioural simulator and an advanced statistical optimizer, the proposed tool allows to efficiently map the modulator specifications into building-block specifications in reasonable computation times. To the best of our knowledge, this is the first tool that is able to synthesize an arbitrary ΣΔM architecture using any circuit technique (SC, SI or CT). The implementation in the MATLAB/SIMULINK platform brings also numerous advantages with a relatively low penalty in computation time.

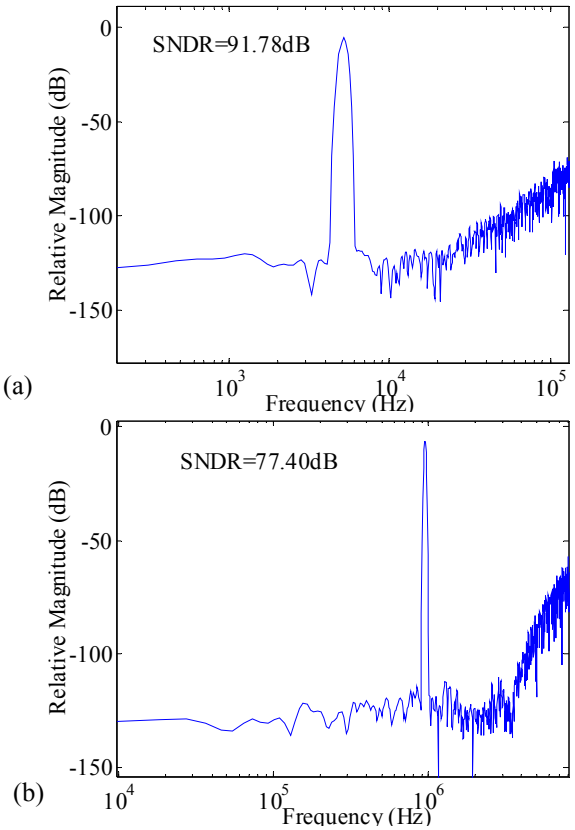

Fig. 9: Output spectra and *SNDR* of the synthesized (a) SC 2-1 sb and (b) CT 5th-order LP ΣΔMs.

References

[1] F. Medeiro, B. Pérez-Verdú, and A. Rodríguez-Vázquez: *Top-Down Design of High-Performance Modulators*, Kluwer, 1999.

[2] J. A. Cherry and W. M. Snelgrove: *Continuous-Time Delta-Sigma Modulators for High-Speed A/D Conversion: Theory, Practice and Fundamental Performance Limits*, Kluwer, 2000.

[3] J. M. de la Rosa, B. Pérez-Verdú and A. Rodríguez-Vázquez: *Systematic Design of CMOS Switched-Current Bandpass Sigma-Delta Modulators for Digital Communications Chips*, Kluwer, 2002.

[4] G. F.M. Beenker, J.D. Conway, G. G. Schrooten and A. G.J. Slenter: "Analog CAD for Consumer ICs", *in Proc. Workshop on Advances in Analog Circuit Design,* pp. 343-355, 1992.

[5] M.F. Mar and R.W. Brodersen: "A Design System for On-Chip Oversampling A/D Interfaces", *IEEE Transactions on VLSI Systems,* Vol. 3, pp. 345-354, September 1995.

[6] K. Francken, P. Vancorenland and G. Gielen: "DAISY: A Simulation-Based High-Level Synthesis Tool for ΔΣ Modulators", *Proc. IEEE Int. Conf. Computer-Aided Design*, pp. 188-192, 2000.

[7] F. Medeiro, B. Pérez-Verdú, A. Rodríguez-Vázquez and J.L. Huertas: "A vertically Integrated tool for Automated Design of ΣΔ Modulators", *IEEE Journal of Solid-State Circuits,* Vol. 30, No. 7, July 1995.

[8] G.G. E. Gielen and R.A. Rutenbar: "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits", *Proceedings of the IEEE,* Vol. 88, pp. 1825-1852, December 2000.

[9] The MathWorks Inc.:Using MATLAB Version 6", July 2002.

[10] The MathWorks Inc.: "Using Simulink Version 5", July 2002.

[11] The MathWorks Inc.: "Writing S-Functions Version 5", July 2002.

[12] V. F. Dias, V. Liberali and F. Maloberti: "Design Tools for Oversampling Data Converters: Needs and Solutions", *Microelectronics Journal*, Vol. 23, pp. 641-650, 1992.

[13] C. H. Wolff and L. Carley: "Simulation of Δ–Σ Modulators Using Behavioral Models", *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 376-379, 1990.

[14] V. Liberali, V.F. Dias, M. Ciapponi and F. Maloberti, "TOSCA: a Simulator for Switched-Capacitor Noise-Shaping A/D Converters," *IEEE Trans. Computer-Aided Design*, Vol. 12, pp. 1376-1386, Sept. 1993.

[15] S. Brigati, F. Francesconi, P. Malcovati, D. Tonieto, A. Baschirotto and F. Maloberti: "Modeling Sigma-Delta Modulator Non-idealities in SIMULINK", *Proc. IEEE Int. Symp. Circuits and Systems*, pp. II.384-387, 1999.

[16] P. Malcovati, S. Brigati, F. Francesconi, F. Maloberti, P. Cusitano and A. Baschirotto: "Behavioural Modeling of Switched-Capacitor Sigma-Delta Modulators", *IEEE Trans. on Circuits and Systems-I*, pp. 352-364, March 2003.

[17] F. Medeiro et al.: "Design of a Broadband ΣΔ Modulator in 2.5V CMOS", *Proc. Design, Automation and Test in Europe: Designers' Forum*, pp. 219-223, 2002.