TOSCA: A Simulator for Switched-Capacitor Noise-Shaping A/D Converters

Valentino Librali, Victor F. Dias, Marco Ciaipponi, and Franco Maloberti, Senior Member, IEEE

Abstract—Today’s trends in mixed analog-digital circuit design show that oversampling techniques will probably be preferred over traditional approaches for implementing high-resolution data converters. Two main factors have limited their widespread use: common analytical as simulation tools are inadequate for exact analysis because these circuits are inherently nonlinear, and conventional electrical simulators are not suitable for simulation because very long transient analyses are required. A software tool named TOSCA (Tool for Oversampled Switched-Capacitor A/D Converter Analysis) has been developed in order to overcome these drawbacks. The simulator is behavioral, general purpose and fully user-friendly. Because a set of basic building blocks is available, generic switched-capacitor noise-shaping A/D converters can be analyzed simply by building a netlist file. Two hierarchical levels have been considered for circuit description: block-level for sub-circuits like quantizers and digital filters, and component-level for sub-circuits like the integrators where switches, capacitors and operational amplifiers are used as building elements. The developed models allow the most relevant non-ideal parameters of the components to be taken into account and a set of post-processing facilities allow extensive analysis of the circuits. The program is written in C language, uses dynamic memory allocation and is very fast.

I. INTRODUCTION

In recent years oversampled data converters have become very popular for implementing analog-to-digital (A/D) and digital-to-analog (D/A) interfaces in standard CMOS processes [1]–[6]. It is well known that their main advantage over conventional Nyquist-rate data converters comes from the possibility they have to trade speed for resolution. Because the analog processing required is limited to low-accuracy operations such as integration and low-resolution quantization and D/A conversion, these circuits allow to shift the resolution limits from the accuracy-matching properties of the components to the more convenient trade-off between speed and thermal noise [7]–[9]. Moreover, these advantages are considered crucial by circuit designers who must use CMOS technologies which are optimized for digital circuitry. Recent theoretical works and IC implementations indicate that with state-of-the-art technologies more than 16 bits of resolution can be achieved for digital-audio signals [6]–[8].

Up until now, two main factors have limited the efficient design of oversampled data converters: the intrinsic nonlinear nature of the circuits, and the lack of general purpose simulators suitable for the analysis of tens of thousands of sampling periods [10]–[16].

As illustrated in Fig. 1, existing tools for the simulation of oversampled data converters can be classified by considering three principal qualifying aspects [18]: the accuracy of the models used in the modulator and decimator (transistor-level, behavioral, or simple difference equation modeling), the ease and the flexibility of use of the tool (the tool can be or not be user-friendly and limited or not to the simulation of a finite number of modulator and decimator structures) and finally, the postprocessing algorithm used for the evaluation of system performance. An optimum tradeoff of the above-mentioned aspects determines the features of the simulator TOSCA. It has been developed in order to meet the following four key features.

1) Since the modulator performance is evaluated on the basis of the statistical average of a very large number of samples, it should be possible to simulate the long data stream, and consequently to perform the very long time analysis in reasonable CPU time.

2) Since the simulator is dedicated to the analysis of switched-capacitor networks, it should be able to take the nonideal effects of the components into consideration and to accurately evaluate the signal samples at the end of each clock phase.

3) Since the circuit performance is better expressed in terms of the signal-to-noise ratio (SNR), rather than non-linearity parameters, the simulator should contain a specific postprocessor. Code density tests could also be applied but at the expense of yet more time consuming simulations [19]–[20].

4) The simulator should be general purpose in order to be able to simulate generic modulator/decimator structures, and to have a simple interface for the description of the circuit and for the analysis commands (netlist interface as in some electrical simulators).

TOSCA is based on a set of building blocks which allow generic oversampled modulators and digital decimators to be built up as a block-level network [22]–[25]. For cases where flexibility inside a specific block is essential, the possibility for component level description has also
been considered. As an example, the topology of a switched-capacitor integrator can be specified as a subcircuit where switches, capacitors, and operational amplifiers are used as building elements, like in Swicap [28]. The models developed for the building blocks allow the most relevant nonideal parameters of the components to be taken into account, a fact which makes simulation results quite realistic. The simulator contains post-processing routines that allow extensive analysis of the circuits to be made. Available facilities include the calculation of signal-to-noise ratio (SNR) for sinusoidal and dc input signals, time and frequency domain probing of circuit node voltages as well as the calculation of their respective amplitude histograms.

The circuit description is contained in a netlist file. This file is divided into sections containing timing conditions, modulator architecture, integrator topology, decimator structure and analysis commands. Analysis results are stored in separate text files which can easily be used by graphic postprocessors.

The paper is organized as follows. Section II introduces the simulator set up and structure and defines the conditions that a converter should meet in order to be broken up into independent sub-circuits and the resulting network to be computable. Section III describes the models developed for the different analog and digital building blocks, paying special attention to nonideal effects present in the integrator, the quantizer and the digital decimator. Section IV illustrates the post-processing facilities available and gives details concerning the algorithm developed for the calculation of the SNR. Finally, in Section V a case study is considered which demonstrates the flexibility and easiness of use of the simulator, and the potentialities it offers [25].

II. SIMULATION APPROACH

A. One-Way Block Modeling

The simulation of an oversampled data converter requires the analysis of a very large number of samples. Transistor level simulators, such as SPICE [26], are not suitable for this kind of simulation because they require many millions of time steps. This would result in several days (or weeks) of CPU time for each simulation. By contrast, behavioral simulators have proved to be a reliable alternative for the fast and accurate simulation of oversampled switched-capacitor A/D converters [10]-[16]. This approach requires to partition the circuit into a set of independent sub-circuits, and to describe each of these blocks with explicit equations that relate the outputs in terms of the inputs and the internal state variables. In general, it is possible to state that a part of a system is an independent sub-circuit when the following conditions are met: firstly, the output of the sub-circuit has not an instantaneous effect on its own inputs and secondly, the internal state variables and the outputs can be updated simply by knowing the previous state of the block and the current inputs. These two conditions guarantee that a network can be expressed by a set of explicit equations which relate all the circuit nodes. In addition, if a permutation of the network nodes exists which allows the set of equations to be written in a triangular form, then the network is said to be computable and a direct solution can be found for each circuit node [27]. This means that the nodes of the circuit can be calculated consecutively in each clock phase using explicit equations. This is a key feature for greatly reducing the overall computation time. Switched-capacitor noise-shaping A/D converters usually conform to all these conditions.

B. Set-Up

The basic setup of TOSCA is shown in Fig. 2. It is worth noting that each block has a direct correspondence to a real hardware system, in which signal, clock generators, modulator, decimator and post-processing blocks are considered separately. With this approach several data entry points (EP) are available in the simulator which allow software and hardware blocks to be interchanged together. In this way the simulator is able to generate data for testing real IC circuits, or can accept real signals as input and process them as if they were simulated data.

An input netlist for TOSCA follows the same setup indicated in Fig. 2. It is composed of four principal sections: TIMING, for the specification of the clock signals, MODULATOR, for the description of the modulator and integrator architectures, DECIMATOR, for the specification of digital filtering and decimation operations and, finally, ANALYSIS, for the specification of analysis commands. The latter refers to postprocessing routines which allow the system to be characterized with commonly used parameters.

Let us consider in Fig. 3 the schematic diagram of a two-stage second-order sigma-delta (ΣΔ) modulator (MASH), cascaded with a digital decimator. The basic building blocks where the overall converter has been broken up are indicated. They are a signal source, two integrators and comparators, delays, adders, and a digital decimator. All of these blocks are independent sub-circuits in the sense defined above.
Figs. 4(a)–4(h) show the set of basic building blocks considered in TOSCA. They correspond to the following.

- Signal generator—(a);
- integrator—(b);
- adder—(c);
- multiplier—(d);
- delay—(e);
- Quantizer—(f);
- FIR and IIR digital decimator—(g); at the highest level of description,
- Operational amplifier, Switch and Capacitor—(h);

at the level in which the topology for the integrators is specified.

The circuit is translated into a netlist using a dedicated syntax whose format is similar to the ones used in common simulators [26], [28], [29]. For the converter considered above, Fig. 5 illustrates the two files containing the input netlist for TOSCA: mash.net, specifying the converter comprising the modulator and the decimator architectures as well as timing and analysis commands, and ni.sci specifying the topology of the two integrators. Different topologies could also be used for the two integrators.

C. Structure

Fig. 6 shows the simplified flowchart of the program. Firstly, the input files are parsed and the information contained in the netlist is read and transferred into internal data structures. Error messages, if any, are displayed at this stage. Secondly, a computability check is performed and the network nodes are reordered according to the
needs of the updating algorithm. Next, the length of the sequence to be analyzed is calculated. It is determined by the number of output samples required, by the decimation factor and by the length of the digital filters. Then, memory is allocated. Because dynamic memory allocation is used, no structural limitations are imposed on circuit complexity or on the length of the analysis to be made. Finally, analysis is performed in three consecutive steps. In the first step the behavioral models are generated once per simulation run and the modulator section is solved for the entire analysis length. In the second step the decimator is simulated and the output sequence is stored. Finally, in the third step the required postprocessing analysis is made and output files are generated. These files can then be used by common graphic postprocessors.

III. BEHAVIORAL MODELING

This section describes the models developed for the building blocks in TOSCA. Tables I and II summarize the complete set of blocks available in the modulator and decimator sections, respectively.

A. Signal Generator

As shown in Table I (see also Fig. 4(a)), TOSCA gives the user the possibility for using dc, sinusoidal and square waveform generators. The syntax used allows amplitude, frequency and phase in sinusoidal and square waveform generators to be specified. Generic signal generators can also be provided through external data files, being in this case referred to in the input netlist as user-defined signal generators.

B. Integrator

With TOSCA the user has the possibility for using three distinct classes of switched-capacitor integrators: ideal, standard and user-defined. For the former, the simple difference equation:

\[ y(n) = y(n-1) + \sum g_k x_k(n) \]  

is used. Coefficients \( g_k \) are the integration gains that apply to each of the inputs \( x_k(n) \), and \( y(n) \) is the output sequence.

For standard integrators, the simulator refers to the scheme shown in Fig. 4(b), where multiple SC branches, both inverting and non-inverting can be considered at the input. Finally, for user-defined integrators the simulator refers to a topology supplied by the user in an additional netlist file—see the \( \text{ni} . \text{scit} \) netlist in Fig. 5(b). In this case, switches, capacitors, and amplifiers can be used to introduce the desired integrator topology to the simulator. In the two latter cases, the user has the possibility of specifying a set of optional parameters which allow the most relevant nonideal characteristics of the operational amplifier to be taken into account. These parameters are finite dc-gain, bandwidth, slew-rate, input offset voltage, output swing and dc-gain nonlinearity.

In order to consider the effect of the nonideal parameters, the developed behavioral model considers three distinct modes of operation: linear, for modeling dc-gain, bandwidth, and offset voltage; current-limited, for modeling slew-rate; and finally, nonlinear or distortion for output swing and dc-gain nonlinearity modeling. Apart from an initialization and an updating step in the simulation algorithm, standard and user-defined integrators refer to the same basic behavioral model. The generation of the behavioral model follows three principal steps. In the first, the charge conservation principle is applied to each of the internal nodes of the integrator (virtual groundline (VG) nodes) and a relationship like

\[ \nu_o(t) + a \nu_i(t) = b \]  

relating the output \( \nu_o(t) \) and the differential input \( \nu_i(t) \) voltages of the amplifier is obtained. Parameter \( a \) contains the information concerning the topology of the integrator while \( b \) depends on the input signals and the present state
of the integrator. Relationship (2) is referred to as the circuit equation since it is independent from the particular operational amplifier considered. Because standard integrators have a known topology with a single VG node, the resulting circuit equation is known a priori and thus defined explicitly in the simulator. By contrast, user-defined integrators can have several internal nodes, other than the virtual ground of the amplifier. For each of these nodes the charge conservation principle is applied and the resulting set of relationships is next reduced to a single circuit equation like (2). Since the structure of the integrator is not known by the simulator, it is necessary to store and update the internal node voltages of the integrator in a way to permit the corresponding parameters for the circuit equation to be calculated. For that purpose, a matrix relation:

$$Q_i[n] = C_i V_i[n]$$

is used for each integrator in the circuit. Here, $Q_i[n]$ is the charge vector which applies to each capacitor in the integrator, $V_i[n]$ is the node voltages vector comprising the inputs, the output and the internal nodes, and $C_i$ is the capacitance matrix of the circuit.

The second step considers the effects associated with the amplifier. Here the three modes of operation defined above apply. In the linear mode of operation the amplifier is assumed to be a one-pole circuit whose behavior is described by the first-order differential equation [30]:

$$v_d(t) = -\frac{1}{A_0} v_d(t) - \frac{1}{A_0 \omega_p} \frac{dv_d(t)}{dt}$$

(4)

where $A_0$ and $A_0 \omega_p$ are the dc-gain and the gain-bandwidth product, respectively—see Fig. 7(a). Offset voltage, $V_{\text{offset}}$, is modeled with a constant voltage source at the input terminals—see Fig. 7(b).

$$v^-(t) = v_d(t) + V_{\text{offset}}$$

(5)

The solution of (2), (4), and (5) determines the transient of the integrator during the corresponding clock phase, allowing the output voltage to be written in an exponential form like

$$v_d(t) = V_1 + V_2 e^{-t/\tau}$$

(6)

where factors $V_1$ and $V_2$ are calculated once per clock phase and are dependent on the integrator topology, on the input signals, and on the state of the integrator at the beginning of the clock phase. Factor $\tau$ is the time constant of the integrator in the same clock phase. Equation (6) defines a direct and unique solution for the output voltage at any instant inside the clock phase interval. However, because the network is of sampled-data type, (6) needs only to be evaluated at the end of the clock phase.

Fig. 8 shows how the linear and the current-limited modes of operation are modeled together. The control block monitors the switching between the two modes based on the following. First, the slope $k/\tau$ of the time response defined by (6) is compared with the slewing-rate limit, $SR$. The linear mode of operation mode is chosen if the condition $|k/\tau| \leq |SR|$ is verified (Fig. 9, curve (a)) whereas the equation:

$$v_d(t) = V_{\text{omin}} + SR t$$

(7)

is used if $|k/\tau| > |SR|$ (Fig. 9, curve (b)). Here, $V_{\text{omin}}$ stands for the state of the integrator at the beginning of the time slot. The time required for the amplifier to switch back to the linear mode of operation is determined by the continuous derivative condition in the output voltage ($T_{sr} = k/|SR| - \tau$).

The third mode of operation models the nonlinear dc-gain and clipping characteristic of the amplifier. The dc-gain characteristic is modeled by the polynomial [18]

$$A(V_o) = A_0 + \alpha_1 V_o + \alpha_2 V_o^2 + \cdots + \alpha_{k-1} V_o^{k-1}$$

(8)

where $\alpha_i$ are the distortion coefficients of the amplifier and are supplied by the user in the integrator card. The output is set to $V_{\text{omax}}$ or $V_{\text{oamp}}$, respectively, when the negative and the positive saturation levels of the amplifier are reached.

The routine that implements this mode of operation follows three steps. Firstly, the dc-gain is set to the initial value $A_0$ and the output voltage of the integrator is calculated according to the two above operation modes. Next, the output voltage is applied to the polynomial equation (8) and the new value for the gain is evaluated. Finally, the first two steps are iteratively repeated until a given degree of accuracy is reached. Simulations indicate that very few iterations are necessary to guarantee a satisfactory accuracy.

User-defined integrators require an additional step which is the updating of the internal nodes voltages. After
the output node has been calculated, the set of relationships used to determine the circuit equation is solved (2), and the internal nodes of the integrator are calculated. Next, the node voltages vector $V[n]$ in (3) is filled and the corresponding matrix relation is expanded in order to allow the charge vector $Q[n]$ to be updated. At this point the simulator is able to initialize the following clock phase.

C. Adder, Multiplier, and Delay

The simulator allows common mathematical operators like adders, multipliers, and delays to be used as basic building blocks—see Fig. 4(c)–4(e). With the former of those blocks a signal can be generated resulting from the weighted sum of $k$ different signals or node voltages. This block can also be used as amplifier or attenuator. The multiplier block allows two signals to be multiplied, and the delay block permits to implement a delay over a node voltage. The latter block is the equivalent of a D-type flip-flop.

D. Quantizer

As the name itself indicates, this block implements the quantization of an analog signal—see Fig. 4(f). The output is an integer number that belongs to a set of $2^N$ possible values, $N$ being the resolution in bit. The user can adopt two distinct classes of quantizers: comparators and generic $N$-bit quantizers. Comparators refer to a model in which nonideal parameters like constant and state-dependent offset voltage are available. Generic $N$-bit quantizers can be of three distinct types: floor, with which integer numbers in the interval $-2^{N-1}$ up to $2^{N-1} - 1$ are covered, ceiling, that refers to code numbers in the interval from $-2^{N-1} - 1$ up to $2^{N-1}$ and, finally, symmetric, where the code number zero is excluded.

E. Decimator

The decimator in an oversampled A/D converter is the part of the circuit where the sampling rate is traded for resolution [31]. Because a lowpass filtering is performed before rate compression, aliasing of high frequency quantization noise can be made negligible and thus an increase of resolution in the output words is possible. As known, the maximum achievable resolution is determined by the amount of noise power that is initially left in band by the modulator.

The decimator section in TOSCA permits to implement any filtering and rate-compression operation. It is possible to define generic cascades of decimators [32], each decimator being specified by a filtering function $H(z)$, and a rate reduction factor $M$—see Table II and Fig. 4(g). Three distinct types of transfer functions can be implemented: all-pass, i.e., a simple rate compressor, $k$-th order moving average filters (Sinc$^k$), and user-defined FIR and IIR filters [27]. In the latter of these cases the filter coefficients must be provided through an external file. The filters are implemented in the time domain using the generic equation:

$$y(n) = \sum_{k=0}^{P-1} a_k x(n - k) - \sum_{k=1}^{D-1} b_k y(n - k)$$

where $a_k$ and $b_k$ are the coefficients of the filter. The following nonideal effects can be taken into account: roundoff of the coefficients and the finite nature of the registers, adders, and multipliers. The user can specify the number of bit in the input words and in the coefficients, and the length of the registers in the adder and multiplier operators. All these facilities allow the design of entire decimation cascades and optimize each of the filters accordingly with the needs of the particular modulator structure.

IV. THE POSTPROCESSOR

An important feature of the proposed simulator is the set of postprocessing facilities available for evaluating the most useful parameters in oversampled data converters. Table III summarizes the postprocessing facilities available in TOSCA. Beside the algorithms for the calculation of the SNR, the simulator also allows the node voltages to be represented in time and frequency domains, as well as their discrete probability density function. For frequency-domain analysis, standard and user-defined cosine windows are also available [34], [35].

There are essentially two types of parameters that can be used to characterize A/D converters: parameters related with the transfer characteristic of the converter, like differential and integral nonlinearity [33], and parameters like the SNR, the dynamic range, and the total harmonic distortion which are related to the spectral properties of the signal. The latter of these are usually preferred to characterize oversampled data converters because they require the analysis of a smaller number of output samples [19].

For the evaluation of the SNR it is necessary, as shown in Fig. 10, to separate the noise from the output signal. For this, a suitable replica $x[n]$ of the applied test signal must be evaluated and subtracted from the converter output $y(n)$ first, and the ratio between the signal and the noise power calculated next. This ratio can be evaluated in the frequency domain using signal and noise spectra, or in the time domain using their corresponding expectation magnitudes. In any case, two major practical considerations limit the application of these algorithms: firstly, it is not a simple task to determine the optimum replica of the signal in such a way as to distinguish the substantial noise from the fictitious contributions arising from irrel-
TABLE III
POST-PROCESSING FACILITIES AVAILABLE FOR THE ANALYSIS

<table>
<thead>
<tr>
<th>Command</th>
<th>Type</th>
<th>Options</th>
<th>Obs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window</td>
<td>common</td>
<td></td>
<td>coeff. list</td>
</tr>
<tr>
<td></td>
<td>cosine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>voltage</td>
<td>clock phase</td>
<td></td>
</tr>
<tr>
<td></td>
<td>histogram</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>Power Spect.</td>
<td>clock phase</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Amp. Spect.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td>ac sweep</td>
<td>source to</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dc sweep</td>
<td>sweep mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>frequency</td>
<td>SNR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sweep</td>
<td>DR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>S/THD</td>
<td></td>
</tr>
</tbody>
</table>

and the signal power, and the ratio:

$$\text{SNR} = \frac{E[x^2(n)]}{E[p^2(n)]}$$

(11)
evaluated as the SNR. Because the DFT routine is used, adequate windows can be applied to the signal to better estimate tones in the presence of noise, while the use of dynamic length sequences allows the adjustment of the input test frequency to one of the bins from the DFT basis set. In this case the whole Nyquist band is considered for noise integration.

V. A DESIGN EXAMPLE

In this section simulation results showing the main features of TOSCA are presented. With this example we intend to show how the noise-shaping A/D converter illustrated above in Fig. 3 can be analyzed. Special attention will be paid to pointing out the three principal features of the tool: how the different postprocessing facilities can be used to perform extensive analysis of the converter, how the models developed for the basic building blocks can be used to define the specifications of the components and, finally, how fast the simulator is. For simulation purposes the decimator has been assumed as a cascade of two filters, respectively, a third-order moving average filter that decimates by 32, and a 59-tap low-pass FIR filter that decimates by four. The output rate is assumed to be 8 kHz. The netlist presented above in Fig. 5 is the input for the simulator. A DECstation 3100 was chosen as working machine, and a typical analysis length of $2^{16}$ input-rate clock periods was considered (two-phase clock).

The calculation of the frequency spectrum is one of the most common and useful analyses made for testing noise-shaping A/D converters. With this kind of plot circuit designers are able to analyze the spectrum of the quantization noise, the effectiveness of the noise shaping, the noise level in the signal band as well as the presence of signal harmonics. As an example, Fig. 12 shows the amplitude spectrum of the signal at the output of the modulator when all the components are assumed to be ideal. Here we can easily identify the signal and the shaping of the noise performed by the modulator. The same analysis could also be made for nonideal components and for any other node in the circuit. The plot in Fig. 12 has been calculated in a CPU time of 75 s (including the $2^{16}$-points FFT).

Fig. 13 instead shows the histogram analysis to the output of the two integrators (18 s of CPU time per curve). This kind of analysis is useful, for example, for defining
the swing specifications for the amplifiers and also to help designers in optimizing the integration gain together with the feedback reference voltages (D/A output levels) [36].

SNR plots are the principal analyses for designers to characterize a noise-shaping A/D converter. This is because these plots contain all the information concerning the performance of the converter, namely the dynamic range, the maximum achievable SNR and the signal-to-noise + harmonic distortion ratio. The plot in Fig. 14 shows the SNR performance of the converter as a function of the amplitude of a sinusoidal input signal. Curve (a) stands for an ideal case condition simulation, and indicates a dynamic range of 90 dB and a maximum SNR of 97 dB, compared with the 94 dB expected from theoretical calculations with the linearized model. Each point in the plot required 17 s of CPU time. In the same plot (Fig. 14) curves (b) and (c) show the effect of using amplifiers with limited output swing, respectively, ±1.75 and ±1.5,
Fig. 14. SNR performance of the converter as a function of the amplitude of sinusoidal input signal; (a) ideal components; (b) and (c) swing limited operational amplifiers, ±1.75 and ±1.5 of the maximum amplitude of the input, respectively.

Fig. 15. SNR performance of the converter as a function of the amplitude of sinusoidal input signal; (a) ideal components; (b) 60 dB and (c) 40 dB dc-gain amplifiers.

relative to the maximum amplitude of the input test signal. Together with the histogram analysis plotted above in Fig. 13 it is then possible to define the minimum output swing required by the amplifiers for which there is no significant degradation of the SNR performance.

Figs. 15 and 16 show some more simulation results concerning the analysis of the effects of the non-ideal characteristics associated with the operational amplifiers. The parameters considered are the dc-gain, the bandwidth and the slew-rate. Fig. 15 plots the SNR performance of
the converter when finite dc-gain values are assumed for the amplifiers. These results have been found to be in close agreement with published works, which in general refer to the MASH structure as being very sensitive to the variations from the ideal behavior of the integrator [1], [36]. The results presented in Fig. 16 correspond instead to the analysis of the effects of the speed parameters of the amplifier. Because this type of analysis is possible, users are able to define the minimum speed requirements for the amplifier, and consequently the maximum allowable operating frequency of the modulator. Curve (a) stands for the ideal converter case, whereas curves (b) and (c) stand for the case of two distinct pairs of bandwidth-slew rate values, 25 MHz, 25 V/μs and 5 MHz, 5 V/μs, respectively. A typical SNR analysis where all the possible nonideal parameters of the components are taken into account is made in a CPU time of 28 s per point in the plot, i.e., approximately twice the ideal case condition. In the cases where user-defined integrators are used the processing time is increased. CPU time depends on the number of integrators used in the circuit as well as on its complexity in terms of the number of internal nodes. Common offset and dc-gain compensated switched-capacitor integrators used in ΣΔ applications (typically with two internal virtual ground nodes), require a CPU time that is approximately five times larger than that of the standard integrator case.

VI. CONCLUSIONS

This paper has presented a simulator which is well suited to the analysis of switched-capacitor noise-shaping A/D converters. The tool is fully user-friendly, fast and accurate enough to analyze high-resolution converters. We have shown that in general SC modulators can be partitioned into a set of independent sub-circuits and that a computable block-level network can be defined. For the simulation of these building blocks we adopted the behavioral modeling approach. These building blocks are signal sources, integrators, quantizers, adders, multipliers, delays and digital filters. Accurate models were developed which allow the most relevant non-ideal characteristic of the components to be taken into account. For SC integrators, a low-level description has also been considered. At this level, users are able to introduce their own integrator topologies using switches, capacitors, and amplifiers as building elements. Another important feature of the simulator is the post-processing facilities available for analysis. Beside the algorithms that allow the SNR to be calculated, we have seen that the simulator also allows the signals to be analyzed in time and frequency domains as well as to represent the discrete probability density function associated with a given node. Finally, a design example has been considered. We have shown that the tool is easy to use, flexible, accurate and of great potential for helping users in the initial design stage of noise-shaping A/D converters. This simulator is presently being tested all over the world both in industrial and research environments.

REFERENCES


Valentino Liberali received the Laurea degree in electronics from Università degli Studi di Pavia, Pavia, Italy in 1986. From 1987 to 1990 he was with Italian Nuclear Physics Institute (INFN) working on the development and testing of low-noise electronics for particle detectors. In 1990 he joined the Department of Electronics of the University of Pavia, Italy, where presently he is Assistant Professor. His main research interests are the design of analog/digital interfaces and the behavioral modelling and simulation of CMOS integrated circuits.

Victor F. Dias received the Licenciatura and the Mestrado in electrical and computer engineering from Instituto Superior Técnico, Lisbon, Portugal, in 1986 and 1989, respectively, and the Doctorato di Ricerca in electronics and informatics from Università degli Studi di Pavia, Pavia, Italy in 1992. Since March 1992 he has been an Assistant Professor at IST, Lisbon, Portugal, and a researcher at INESC-Lisbon where he is involved in the development of mixed analog/digital ASIC's. His current research interests are in mixed analog-digital integrated circuits based on oversampled sigma-delta modulation techniques and dedicated software tools for simulation and testing.

Marco Ciapponi received the Laurea degree in electronics from Università degli Studi di Pavia, Pavia, Italy in 1991. During the last year of his studies he was concerned with the development of a part of the program TOSCA. He is currently working with Bofti s.r.l., Sondrio, Italy.

Franco Maloberti (A’84–SM’87) received the Laurea degree in physics in 1968 from University of Parma. From 1968 to 1969 he was a research assistant with the University of Aquila. In 1969 he joined the University of Pavia, Department of Electronics, where he is currently a full Professor and head of the university’s microelectronic laboratory. He has published more than 90 papers, holds 11 patents and is co-editor of the book Analog/Digital ASIC’s, Circuit Techniques, Design Tools and Applications, (London, UK: IEE Press).

Prof. Maloberti is a member of AIE (Italian Electrotechnical and Electronic Society) and a senior member of IEEE.