PRECISION BEHAVIOURAL MODELLING OF CIRCUIT COMPONENTS FOR DATA CONVERTERS

S Brigati, V Liberati, F Maloberti

University of Pavia, Italy

ABSTRACT

This paper describes the use of behavioural modelling for accurate and efficient simulation of data converters. Examples of behavioural models of building blocks for different classes of converters are provided and analysed. The need for dedicated high-level simulation tools for particular system architectures is also discussed. Simulation results show the effectiveness of the proposed methods in the design of data converters.

1 INTRODUCTION

Data converters have been receiving special attention as they are key elements in mixed-mode application specific integrated circuits (ASICs). Present trends are towards integrated signal processing, partially in the analogue and partially in the digital domain.

The design of mixed-mode integrated circuits means efficient analysis tools must be available, capable of estimating the electrical performance of a whole system within a reasonable time. Data converters are mixed-mode circuits, since they process the signal in both analogue and digital domains. High resolution converters require an evaluation of the effects coming from non-idealities that limit the circuit performance. Thus the simulation tool must have a suitable degree of accuracy. This would require a device-level analysis for the analogue sections, but for large scale circuits containing digital parts this is impractical because the required CPU time increases very quickly the greater the size of the circuit. On the other hand, digital simulation tools cannot provide accurate information on the behaviour of analogue elements. For mixed-mode circuits, the use of two different simulators for separate analyses of analogue and digital sections does not allow us to evaluate the performance of the whole system. Therefore, the need for a unified mixed-mode simulation capability arises, becoming more and more urgent as system complexity grows.

Analogue blocks can be described according to different abstraction levels, as shown in Fig. 1: (a) at the device level, considering all the circuit elements; (b) with a macromodel, or (c) with a behavioural model which considers only the input-output relationship.

Macromodelling consists in the description of a complex circuit in a simpler way by using the components and the elementary building blocks available within a conventional simulator. Behavioural modelling requires the component or block behaviour to be described using a dedicated language which can be either compiled or interpreted by the simulator. In some cases, however, the development of dedicated simulation tools is needed to achieve particular results in terms of accuracy and efficiency when analysing limited classes of systems.

The choice of the modelling level must take into account the validation problem and the availability of technology parameters. Usually, silicon foundries provide device models, i.e., electrical parameters for circuit simulation at device level. The design is considered "error-free" when a standard circuit simulator indicates that the output lies inside the requested specifications for the given range of electrical parameters. Therefore the parameters of macromodels and behavioural models must correspond to the technology parameters supplied by the foundry.

Figure 1: Different abstraction levels for an analogue block: (a) device level; (b) macromodel; (c) behavioural model

The advantages of macromodelling are its independence from compilers and the easy inclusion of macromodels in the system description. Its main drawback is the difficulty of building very complex models since the number of available components is limited.

Behavioural modelling, on the other hand, is only possible when a specific description language is provided within the simulator or when the user is writing his own analysis routine which can be linked to an existing simulator. In this case, however, there is no limit to the complexity and to the accuracy of the models which can be written. Anyway, the more a model is complex and accurate, the longer the simulation time required.

The development of dedicated simulation tools requires much more effort than the previous solutions, but it may be necessary when conventional simulation tools are inefficient. This happens, for instance, during the analysis of oversampled data converters. As pointed out by Dias et al (1), oversampled circuits are strongly nonlinear, must be analysed in the time-domain over a very large number of clock periods, and need specific post-processing routines to evaluate system performance. All the above mentioned requirements are not met by a conventional simulator so a specific analysis tool is needed in this case.

Recently, many authors have proposed different approaches to the macromodelling and to the behavioural modelling of data converters: Casinovi and Sangiovanni-Vincentelli (2), Ruan (3), Liu et al (4)-(5), Gielen et al (6). An overview of CAD tools for data converter design was presented by Gielen and Franca (7). A survey of dedicated simulators for oversampled converters is presented in (1).

In this paper we analyse behavioural modelling of building blocks for data converters with particular reference to the trade-off between model precision and simulation efficiency. We provide examples of behavioural models of Nyquist-rate converter building blocks written in a specific language for a conventional simulator: the FAS language (8) interpreted by ELDO from Anacad Computer Systems (9). Moreover, we describe a dedicated high-level model used within the simulator TOSCA, developed by Liberati et al (10), for the analysis of Sigma-Delta A/D converters. Simulation results show the effectiveness of the proposed methods for the design of data converters.

2 MACROBLOCK MODELLISATION

There is no single solution to the problem of circuit partitioning in macromodels. The choice of the block set has to be done according to common sense rules, preferring those blocks which are (or can be considered) standard cells: operational amplifiers, comparators, etc. Such block partitioning is suitable for most circuits. However, for particular classes of circuits the introduction of more complex building blocks may be opportune, for example the integrator in Sigma-Delta A/D converters.

The model for each block should be general enough to be adaptable to the user's specific problem. This goal is achieved by parameterisation of the model: the user can specify a set of parameters, which characterise a particular block implementation. As for the modelling of transistors in device-level simulation, the model parameters should have a "physical" meaning and should be easily evaluated for a practical circuit implementation, either by transistor-level simulation or by measurement.

The precision of a model indicates how close the model is to the real subsystem from the point of view of their input-output relationships. Once the model has been defined, the numerical values for its parameters can be obtained by minimising the distance between the output of the model and the output of the real subsystem. The choice of the optimum parameter values is an optimisation problem. When developing a model, the definition of its "best" parameter set is an identification problem. In principle, the model can be made as precise as one wants, by increasing the number of parameters, and therefore its complexity. In practice, the maximum available precision is limited by two factors related to complexity. The first depends on the computing resources: the more complex the model, the more CPU time required for the analysis. The second factor lies in the handling of the model. The increase in precision implies the need to determine a larger number of parameter values in order to reach the theoretical precision level. Thus parameter optimisation may become the bottleneck in the process, since either the user is required to have the same knowledge as the model programmer, or the optimisation must be carried out by a time-consuming automated procedure.

From the above considerations it follows that a good behavioural model requires a trade-off between accuracy, ease of understanding, easy extraction of parameters, and simulation time. In our modelling work, we decided to adopt — whenever it was possible — the parameter set that is commonly used by circuit designers to characterise a cell. This choice makes it easier to extract the parameter set from an analogue design. Design validation is done by using the worst case parameter values obtained from transistor-level simulation into behavioural analysis.

2.1 Single Ended Operational Amplifier Model

A single ended operational amplifier can be described by the model shown in Fig. 2. It consists of three sections: (i) a dynamic system, which accounts for differential gain, common-mode gain, power supply rejection ratio, and for their frequency dependence; (ii) a voltage saturation block, to consider non-linearities in the input-output characteristics; and (iii) a current limiter, which imposes a maximum value on the output current.
In the dynamic system, the input voltages $V_{in+}$ and $V_{in-}$ are converted into a differential signal $V_d = V_{in+} - V_{in-}$ and a common-mode signal $V_{cm} = \frac{1}{2}(V_{in+} + V_{in-})$.

Sub-blocks $A_d$ and $A_{cm}$ in Fig. 2 represent the differential and the common-mode gain respectively. They depend on the frequency and in our implementation their expressions in the $s$ domain are

$$A_d(s) = A_{d0} \frac{1 + \frac{s}{z_d}}{(1 + \frac{s}{p1})(1 + \frac{s}{p2})}$$  

for the differential gain, and

$$A_{cm}(s) = A_{cm0} \frac{(1 + \frac{s}{z_{cm1}})(1 + \frac{s}{z_{cm2}})}{(1 + \frac{s}{p1})(1 + \frac{s}{p2})}$$  

for the common-mode gain. $A_{d0}$ and $A_{cm0}$ are the dc gains for the differential and the common-mode signals respectively, $p1$ and $p2$ are the poles in the transfer functions, while $z_d$ and $z_{cm1}, z_{cm2}$ are the zeros of the differential and common-mode gain functions. It is assumed that the poles have the same values for both transfer functions, since the signal path is the same. The power supply gains are expressed with functions similar to (2), with different parameters. The output of the first block is the sum of the four contributions described above.

The given equations are good approximations for most op amps; their parameters can be easily obtained by transistor-level simulation in frequency domain. When necessary, the equations can easily be generalized by adding more poles and zeros. For the simulation in time domain, the $s$-domain relationships can be easily translated into difference equations using the backward Euler transformation.

The voltage saturation block accounts for gain nonlinearities and output swing limitations. Its input is the output of the dynamic system, whose value is not limited. In the ideal case, the input-output characteristic of the voltage saturator should be a straight line with unit derivative in the range $(V_{th}, V_{oh})$, and a constant value ($V_{ol}$ or $V_{oh}$) outside the linear range, as shown in the second block of Fig. 2 (shaded line). To account for gain variations with the output dc level, the model implemented is composed of three intervals: a quasi-linear region, described by a polynomial function, and two exponential regions having $V_{ol}$ and $V_{oh}$ as asymptotic values, respectively. To avoid convergence problems in the simulation, the input-output characteristic of the voltage saturator must be continuous with a continuous derivative. In the practical implementation of this block, we used a third-order polynomial, to ensure a sufficient degree of accuracy. The parameter values can be obtained by minimising the distance between the characteristic of the model and the dc transfer curve obtained with a device-level simulation.

The current limiter accounts for slew-rate limitations, imposing a minimum and a maximum value for the output current ($i_{min}$ and $i_{max}$). A hard clipping model has been used for this block.

### 2.2 Comparator Model

Comparators used in data converters are usually followed by latches and perform the comparison during one phase, while the other phase is reserved for reset and auto-zero operations. The symbol of a comparator

![Comparator Symbol](image)

Figure 3: Symbol of a latched comparator with reset
block is motivated by speed requirements when simulating Sigma-Delta converters.

The scheme of a generic two-phase SC integrator is shown in Fig. 5. Besides the integrating capacitance \( C_o \), the block can have one or more input capacitive branches, either dc connected or switched. Damping branches may also be used. In the implementation of this model within TOSCA (10), there is no limit to the number of input branches.

The model for the integrator op amp is simpler than the one presented in Sect. 2.1, however it takes into account the most important non-idealities: input offset, finite dc gain, finite gain-bandwidth product, non-linear gain, output swing limitation and slew-rate.

The voltage at the op amp virtual ground is

\[
v^-(t) = -\frac{1}{A_o} v_d(t) - \frac{1}{A_o \omega_p} \frac{dv_d(t)}{dt} + V_{offset}
\]

where \( A_o \) and \( A_o \omega_p \) are the dc gain and the gain-bandwidth product of the op amp, and \( V_{offset} \) is the input offset voltage, which is assumed to be constant. In the linear operation mode, the output voltage is obtained as a superposition of the contributions of each input branch. Each contribution is calculated by applying the charge conservation principle at the virtual ground node.

Bandwidth limitations give rise to an exponential waveform at the integrator output during a clock phase. Denoting the initial and the asymptotic voltages with \( V_i \) and \( V_f \) respectively, the output voltage in time domain is

\[
v_d(t) = V_i + V_f - e^{-\frac{t}{\tau}}
\]

where \( \tau \) is the time constant of the integrator. When the slope of the output voltage exceeds the slew-rate \( SR \), the output varies linearly in time until it exits the slew-rate condition:

\[
v_d(t) = V_i + V_f \cdot \tau
\]

3 SIMULATION EXAMPLES

The models discussed in the previous section were used in the computer evaluation of data converters. In this section we shall present two examples: a Nyquist-rate flash A/D converter simulated with ELDO, and a band-pass Sigma-Delta A/D converter simulated with TOSCA.

3.1 Nyquist-rate flash ADC

A 10-bit pipelined A/D converter was designed to process signals from nuclear detectors, by Brigati et al (11). Its first stage is the 5-bit flash A/D converter shown in Fig. 6.
To evaluate the frequency limitations, the converter was simulated with ELDO. Fig. 7 shows the results of a time domain analysis. The sampling frequency is 40 MHz and the input signal is a full-dynamic ramp (dashed line). Different values for the capacitive loads of the comparators were assumed. To model this effect, different time constants were used in the comparator array. Fig. 7 shows that this results in some missing codes in the input-output converter characteristic (solid line). This effect is not present at a lower sampling frequency. The simulation required 11 min of CPU time on a SPARCstation 2.

3.2 Sigma-Delta band-pass ADC

The schematic diagram of a second order band-pass Sigma-Delta A/D modulator is shown in Fig. 8. It is the single-ended representation of a fully-differential structure. The integrator was modelled as described in
4 CONCLUSIONS

This paper has discussed the use of behavioural modelling for the computer analysis of data converters.

Models for an op amp, a comparator and an SC integrator have been described. Two simulation examples of a Nyquist-rate ADC and of a Sigma-Delta ADC have been presented. It has been shown that accurate results can be obtained within a limited CPU time.

In the future, behavioural simulation will be used more and more to analyse complex mixed-mode systems. Therefore, it will become necessary to employ a standard description language for analogue blocks, such as VHDL for digital systems.

5 ACKNOWLEDGEMENT

Work supported by CEC, Esprit Project 5056 AD2000.

6 REFERENCES