Mixed-Signal Characterization Environments for Analog to Digital Converters

Jason Abele, Brian Dupaix, John Sheridan Fisher and Steven B. Bibyk
Department of Electrical and Computer Engineering
The Ohio State University
Columbus, Ohio 43210
Email: abele.5@osu.edu, dupaixb@ece.osu.edu, fisherj@ece.osu.edu and bibyk@ece.osu.edu

***Abstract*— With analog blocks frequently representing the biggest risk in single substrate mixed-signal designs, intellectual property (IP) libraries of fabrication-proven and fully-characterized analog cells can offer a low-risk method to decrease time-to-market (TTM), in any group where TTM is the dominant factor of success. While the development of these libraries has received significant research attention, analog and mixed-signal designs are frequently developed from scratch without much benefit from external IP. We believe that a reason for this recalcitrance has been a lack of confidence in the external IP, and specifically the limited access to characterization that has been done to describe the IP. In this paper, we develop methods to characterize IP with a focus on decreasing TTM. Further, we show how these same characterizations yield performance metrics which are useful in calibrating abstract models to shorten the simulation time for large systems, as well as providing performance specifications for cell-level functional verification. The collection of simulation testbenches, interpretation of results, and control scripts is our "Characterization Environment." As an example, we develop a system of simulation libraries to characterize the behavior, sensitivity to parameter variation, and performance of analog library cells, which are combined for the fabrication of a delta-sigma modulator analog to digital converter ($\Delta\Sigma$ ADC). Finally, we show how the characterization of libraries of analog cells is a key form of IP necessary to promote the common use of external Analog IP libraries.**

I. Introduction

Analog IP is separable into two categories, hard IP and soft IP. Soft Analog IP can be modified by end users to meet individual design needs. Hard Analog IP, such as a mask-layout with abstract simulation models eases TTM by reducing design time and improving system simulation time. However, Hard Analog IP has limited effect in the analog and mixed-signal design communities. Abstract simulation models are intended to provide confidence that simulation result will match fabrication results. However, substrate noise and other proximity effects can only be determined after integration in a mixed-signal system. Because the figures of merit for IP cells vary depending on the application, determining appropriate model order reductions and calibration for abstract models is an area of continuing active research. Further, this kind of IP is only as durable as the targeted fabrication process and can even be affected by variations in the process line over time.

Beyond these problems with both process specific and abstract models, design intent contained in the contained in the testbenches and the interpretation of the simulation results is not transmitted to users of the IP. This collection of simulation testbenches, interpretation of results, and control scripts forms a "Characterization Environment." As analog synthesis and layout generation tools mature, Characterization Environments represent a key enabling component of soft Analog IP. A $\Delta\Sigma$ ADC example is used to illustrate the volume of characterization content which is left out of the exchange of Analog IP.

II. Characterization

Characterization Environments are not new to software or digital hardware projects. Indeed they have developed into formal and empirical verification methodologies for both of those disciplines. The best analog designers also practice retention and reuse of their Characterization Environments [1]. Extending the practice of developing libraries of Characterization Environments to the analog design community has broad benefit.

These libraries give design context to the models they characterize. Context can be used to bring new designers up to speed on projects and provides a record of best practices when developing design expertise. Better than white papers and books, these libraries offer an executable environment for exploring the accumulated design practices. The context also sets a common grammar for design reviews. For example, if design teams use the same characterization library to determine bandwidth, the ambiguity of noise equivalent bandwidth versus 99% bandwidth calculation can be easily resolved [2]. Also, the effect of an experienced designer's departure is reduced if the Characterization Environment is well documented. These are all reasons that design reuse and the market for IP flourishes in software and digital design. Unfortunately, Characterization Environments for analog and mixed-signal projects are still frequently designed from scratch.

A. Capturing Characterization

Borrowing from the verification model used in the digital hardware design process [3] we propose a structure for analog and mixed-signal characterization libraries as depicted in Figure 1. There are four basic components: biasing, stimuli, results extraction, and constraints. This model provides a gradual transition from current practice into an environment ready for empirical verification. Current practice usually mixes

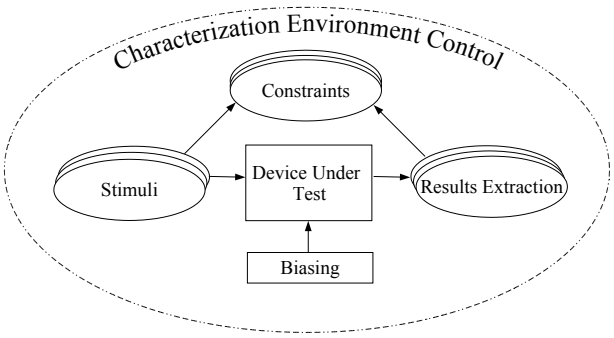

Fig. 1. Characterization Environment

some set of preconditions and biasing with input stimuli to form a testbench. The simulator is then used to refine or select the stimuli and capture the results. Results are usually post-processed and plotted for hand measurement or visual inspection.

A first step to improving this practice involves formalizing the designer's understanding of expected simulation results. A designer has an expectation of the shape of a Bode plot from the AC simulation of an OTA. For example, the gain-bandwidth (GBW), phase margin and pole frequency can be easily extracted from open-loop simulation results of an OTA [4]. Results Extraction captures the designer's expectations with automated post-processing of simulation data and stores the frequency where $V_{output}/V_{input} = 1$ (GBW) or $180 - phase$ at the GBW frequency (phase margin). Results extraction can be accomplished using functions built-in to the simulator, through the use of computer algebra systems, or in mathematics simulation languages like the mixed-signal hardware description languages (HDL-AMS). Additional specific examples will be explored in the example $\Delta\Sigma$ ADC.

The second step involves separating biasing and preconditions from the stimuli. Since the biasing and preconditions are more tightly coupled to specific implementations of a particular cell, this activity is useful to abstract the stimuli for use with a broader class of IP cells. The biasing, preconditions and stimuli generally remain in the same simulation language as the cell, though some simulation environments allow mixing of HDL-AMS, circuit model and other descriptions. With the stimuli and results extraction abstracted from implementation dependence, a vocabulary for hierarchical constraints has been developed.

Finally, constraints are written to compare stimuli and results against a specification to complete the Characterization Environment. Constraints can be run as part of the simulation or developed as a post processing step. For the OTA example, one useful constraint would compare the GBW, phase margin and pole frequency from AC stimulus against the specification, producing errors if constraints are not met.

In the remaining sections, we describe a Characterization Environment for a $\Delta\Sigma$ ADC. The characterizations have been cast in math and pseudo code for clarity outside of specific simulation and test environments.

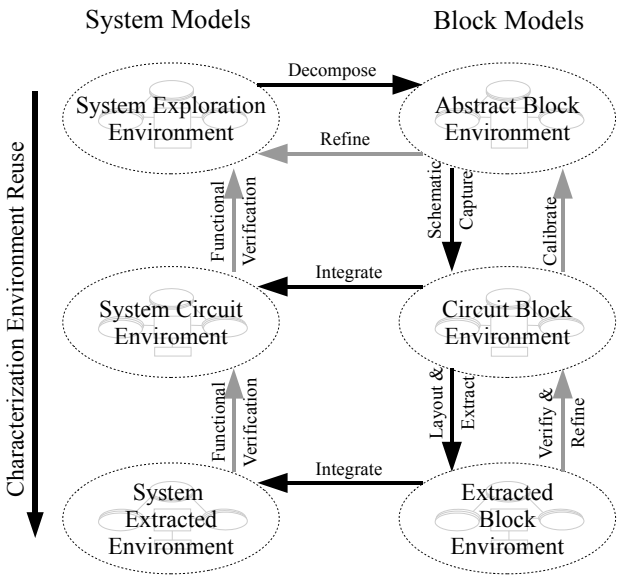

Fig. 2. Characterization Environment Workflow

III. Characterization of a $\Delta\Sigma$ ADC

$\Delta\Sigma$ ADCs have enjoyed significant research attention. They represent a class of data converters which relax component requirements by sampling at a frequency higher than the Nyquist criteria and applying noise shaping to move quantization noise out of the band of interest [5]. Although they can be implemented in a continuous time fashion, a discrete-time architecture using switched capacitor filters is more common and shall serve as our architecture. As a starting point, we chose specifications of 16-bit resolution and a low oversampling ratio (OSR) of 16 at 100Msps for a wide Nyquist equivalent bandwidth of 3MHz.

We developed a range of Characterization Environments within our design flow from Figure 2. While working with models of the entire system, we start with the System Exploration Environment to evaluate architecture and system design decisions. This Characterization Environment will be reused and expanded to evaluate the system with circuit and extracted models in the Circuit Block Environment and Extracted Block Environment, respectively. Each block in the system hierarchy is characterized in its own environment for the level of abstraction that corresponds with its available models.

A. System Characterization

To begin exploring the system level model, we find that it is important to choose a language that is very flexible and offers a rich library of analysis tools suited to the system being designed. Historically, computer algebra systems like MATLAB have fit these requirements. However, analog mixed-signal hardware description languages (HDL-AMS) are also attractive for their ease of cosimulation with circuits and other HDLs. In order to explore the system architectures, we implemented a variety of system models as finite difference

equations. We were interested in evaluating the effects of architecture design decisions on resolution.

Our next step was to define a simulation for the System Exploration Environment. Each simulation in the environment is composed of a set of input stimuli, constraints and results extraction. We chose a sinusoid input stimuli with a frequency of f_{in} and a magnitude of A_{in} allows us to extract the desired results. A constraint monitored the system state for quantizer overload, which is a common problem with $\Delta\Sigma$ ADCs [5]. The noise-shaping properties of a $\Delta\Sigma$ ADC rely on the quantization noise to be bounded by $\pm LSB/2$ (least significant bit). Hence we write a constraint which compares the input of the quantizer, v, to the analog feedback, y, for each quantizer at each sample. If the error signal $|y-x| > LSB/2$ then the quantizer is overloaded. This error state is noted and the simulation can be halted.

Results were extracted using the Finite Fourier Transform (FFT) of the outputs and measuring peak Signal to Noise Ratio (SNR). In order to use the (FFT) on the outputs from a transient (time-domain) simulation, careful consideration of simulation is required to ensure that the FFT will not be the limiting factor in our System Exploration Environment. The FFT produces sampled spectrum which is periodic and discrete with respect to frequency. The resulting spectrum is said to be "binned" with a maximum frequency resolution shown in Equation 1 where f_s is the sampling frequency, N_s is the number of samples, and T_{sim} is the length of the transient simulation.

$$Resolution = \frac{f_s}{f_s T_{sim}} = \frac{1}{T_{sim}} \tag{1}$$

Thus it can be seen that the length of the transient simulation, determines the frequency resolution. However, it is also important to note that in oversampled converters, only $\frac{1}{OSR}$ of the bins are within our band of interest. Knowing that inherent non-linearities in our frequency response lead to harmonics in the output, we expect to find integer multiples of f_{in} as components of the output. We choose a fully differential architecture for the benefit of canceling even harmonics. Thus the output will contain frequency components at both f_{in} and $3f_{in}$ which we would like to measure. This frequency separation of $2f_{in}$ defines our critical frequency resolution. Due to signal "leakage [6]," the output frequency spectrum will contain impulses which have spread based on the window function applied. Thus for a rectangular window, we require at least three "bins" between sample and its expected harmonic to resolve the impulses. Thus the transient simulation needed a minimum resolution:

$$\frac{1}{T_{sim}} = \frac{2f_{in}}{3bins} \Rightarrow T_{sim} = \frac{3bins}{2f_{in}} \tag{2}$$

With an FFT of sufficiently high resolution, the SNR can be extracted. By varying A_{in} of the input and extracting SNR, peak SNR is the maximum SNR in the linear region.

Now that we have an idea of the length of simulation based on the resolution required in the FFT, we can begin to build the Characterization Environment control logic for the System Exploration Environment. This control can be as simple as a loop of transient simulations for the appropriate duration, which varies the amplitude of the input sinusoid. After each transient simulation, the data probed from the outputs can be passed into the results extraction to determine the SNR from a single transient simulation. This data can be tabulated in the control loop and passed as a vector to another results extraction block, where the peak SNR is determined. All of the results should be logged for comparison with the more detailed models. After determining that a 6^{th} order modulator composed of a cascade of three 2^{nd} order modulators can meet our specifications, the system is decomposed into blocks (Figure 3).

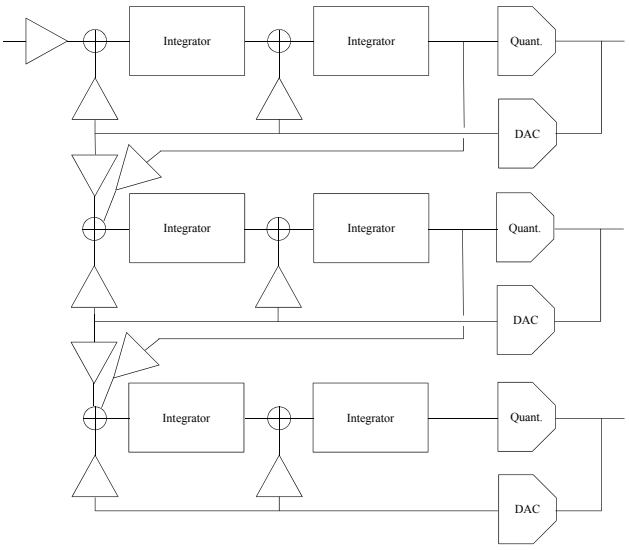

Fig. 3. $\Delta\Sigma$ Block Diagram

B. Block Environments

At this point, the system has been decomposed into blocks and each block in the system needs a set of specifications. There are three basic blocks whose specifications need to be characterized: an integrator, a quantizer, and a DAC. We chose an architecture with a 1-bit quantizer and DAC to ease the linearity design requirement. Linearity and settling time are the two most critical performance metrics of these blocks, both of which are dependent on their load, parasitics and supply noise. For that reason, modeling of these non-idealities is best revisited after detailed circuit design when models can be calibrated to their environment. The integrator, especially the first in the signal path, is the most critical block [7]. The integrators are further decomposed into an opamp and switched-capacitor filter. The opamp contains the majority of the non-ideal performance within our design. These effects are explored in detail by Wu [7] and others. If a rich library of parameterized abstract models is not already available, it is preferable to consider block specifications analytically and continue to detailed circuit design. After extracting performance parameters from the circuits, the abstract models

can be revisited. By delaying the creation of abstract block models, design time can be focused on the high value blocks which contribute the most to system simulation length [8] or present the least conservative margin between specification and characterized performance. Careful attention to the design of of the detailed block Characterization Environment allows reuse for checking calibrated abstract model performance against extracted results from the circuits.

At the detailed block level, blocks are expressed as circuits and netlists suitable for a variety of SPICE-style simulators [9]. Many of the commercial SPICE variants provide tools to control simulations, provide stimuli, probe the circuit and process results. For the example of such an environment built around the opamp, Cadence provides a detailed whitepaper [4]. We prefer to keep our environments free of single vendor extensions to the extent that it can be reasonably managed. Thus, although we may need to use vendor extensions for simulation control or probing the circuit, results extraction is kept primarily in libraries of MATLAB code to facilitate reuse across Characterization Environments.

Constraints on the detailed blocks can be evaluated "online" during the simulation, or "offline" by comparing probed results. By implementing "online" constraints in a language available within the circuit simulator, such as Verilog-A, critical simulation errors can abort failed simulations before completion. However, this can limit their utility in abstract environments which may not have electrical simulation. Casting the constraints "offline" in MATLAB or similar eases reuse, at the expense of simulation control. The value of reusing constraints in abstract models is dependent on whether abstract models calculate the signals compared in the constraint. An example constraint on the integrator evaluates the resolution of the settling at each sample. The resolution of the $\Delta\Sigma$ ADC will be affected by integrator settling which fails to meet the design specification of the system. This constraint can also be extended to the DAC, looking for complete settling on the feedback inputs of the integrators.

C. Revisit System Characterization

After the block characterizations, the complete system system Characterization Environments require attention to evaluate performance with the detailed blocks. Although system characterization with the complete circuit or extracted model may be useful before fabrication, the prohibitive run time of such simulations (days to weeks) warrants removing it from the design loop. Instead, calibrated abstract models are used to selectively replace detailed block models, reducing system characterization time and improving iteration time in the characterization workflow (Figure 2). By reusing the set of Characterization Environments developed for system and block models, a form of functional verification is performed by comparing extracted results from characterization runs at different levels of detail.

IV. Conclusion

The preceding is an overview of the characterization methods employed in the design and fabrication of a $\Delta\Sigma$ ADC. We outlined some key concerns in designing Characterization Environments as a way to capture metrics of behavior, performance and sensitivity to parameter variation. They can also be used to improve system simulation speed by replacing circuit models with calibrated abstract models. By reusing these environments at different levels of abstraction, designers develop confidence that the fabricated design matches the metrics from system exploration. Reusing the Characterization Environments across multiple similar designs or as designs are migrated to new fabrication processes spreads confidence an expertise throughout a group. Our examples continue to develop into a library as part of ongoing research.

In developing a library of characterizations, abstraction from process and cell implementation is an important goal. Abstracted characterizations can be readily reused in new processes and cell designs. Although the characterization libraries developed in this paper are aimed at simulations, they can be implemented in test equipment languages, like LabView, while the design is being fabricated. Thus tests can be ready to run when the fabricated design samples arrive, shortening the time until the performance of the samples is known. Computer evaluated results extraction and constraint comparison opens the door to regression testing. Regression testing allows engineering changes to be automatically evaluated at many levels of hierarchy without direct intervention of the designers in analyzing large system simulation results. Another enhancement to a regression testing environment would automatically tie regression test results to design revision history.

Characterization libraries capture knowledge about the context within which a system is designed to operate. This allows analog or mixed-signal systems to be evaluated for suitability in a larger project. Automation of Characterization Environments can be used to free designers from shepherding simulations and focus on design and characterization enhancement while Characterization Environments generate reports about extracted results and constraints.

References

[1] G. Gielen and R. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits," *Proceedings of the IEEE*, no. 88, 2000.

[2] B. Sklar, *Digital Communitcations*. Englewood Cliffs, New Jersey: Pretence Hall, Inc, 1988, sec. The Bandwidth Dilemma, pp. 43–44.

[3] J. Bergeron, *Writing Testbenches: Functional Verification of HDL Models*. 101 Philip Drive, Assinippi Park, Norwell Massachusetts 02061 USA: Kluwer Academic Publishers, 2001.

[4] C. D. S. Inc., "Functional verification of a differential operational amplifier," URL: http://www.cadence.com/whitepapers/.

[5] P. Aziz, H. Sorensen, and J. vn der Spiegel, "An overview of sigma-delta converters," *Signal Processing Magazine, IEEE*, no. 13, 1996.

[6] F. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, no. 66, 1978.

[7] J.-Y. Wu, "Roundtrip Design Strategy of High-Speed Delta-Sigma A/D Converters," Ph.D. dissertation, The Ohio-State University, 2000.

[8] D. Shah and D. Lee, "Guidelines to maximize the performance of verilog-ams/vhdl-ams behavioral modeling," *Chip Design*, pp. 17–28, Apr. 2005.

[9] L. Nagel, "Spice2: A computer program to simulate semiconductor circuits," Memo UCB/ERL M520, Tech. Rep., May 1975.