Practical Simulation Flow for Evaluating Analog and Mixed-Signal Test Techniques

Manuel J. Barragan, Haralampos-G. Stratigopoulos, Salvador Mir, Hervé Le-Gall, Neha Bhargava, Ankur Bal

Practical Simulation Flow for Evaluating Analog and Mixed-Signal Test Techniques

Manuel J. Barragan[≠‖]
[≠]Université Grenoble Alpes, TIMA, F-38000, Grenoble, France
[‖]CNRS, TIMA, F-38000, Grenoble, France

Haralampos-G. Stratigopoulos[*]
[*]Sorbonne Universités, UPMC Univ. Paris 06, CNRS, LIP6
4 place Jussieu, 75252 Paris CEDEX 05, France

Salvador Mir[≠‖]
[≠]Université Grenoble Alpes, TIMA, F-38000, Grenoble, France
[‖]CNRS, TIMA, F-38000, Grenoble, France

Hervé Le-Gall[&]
[&]STMicroelectronics
12 Rue Jules Horowitz
38000 Grenoble, France

Neha Bhargava[#] and Ankur Bal[#]
[#]STMicroelectronics
Plot No. 1, Main Road, Knowledge Park III
Greater Noida, Uttar Pradesh 201308, India

***Abstract*—Alternative test techniques are continuously proposed for analog and mixed-signal circuits with the aim to reduce the standard test cost and complexity. One of the main reasons why the majority of these alternative test solutions have not been met with success is the lack of a proof that they will not sacrifice the accuracy of the standard test, resulting in intolerable test escapes and yield loss. In this paper, we target specifically circuits with long simulation times and we present a practical simulation flow that allows evaluating in a time-efficient manner alternative test solutions during their development phase by estimating the resultant parametric test metrics. The simulation flow can be used to provide feedback for refining an alternative test solution and to give the green light for implementing the alternative test solution in high-volume production. The simulation flow is demonstrated for evaluating a built-in self-test strategy for ΣΔ ADCs.**

***Index Terms*—Mixed-signal circuit testing, test metrics, Monte Carlo methods, statistical blockade, built-in self-test, ΣΔ ADC testing.**

I. Introduction

The standard industry practice for high-volume manufacturing testing of analog circuits is to measure directly the performances that are listed in the datasheet and compare them to their specifications. Here, by "analog" we refer in general to non-digital circuits, e.g. baseband analog, mixed-signal, RF, etc. This testing approach is straightforward and precise and the results are easily interpretable, yet it comes at a very high cost since accurate analog tests require long run times on costly automated test equipment.

To this end, there are significant efforts for developing alternative test approaches to replace the standard tests and reduce cost. These approaches include built-in self-test (BIST) where performances are measured on-chip [1], [2], design-for-test (DfT) where on-chip test structures are used to facilitate test access and test response observability [3], and defect-oriented test where the goal is to generate tests that guarantee detection of defects [4].

An alternative test approach is evaluated against a number of criteria, including whether (a) it can be automated, (b) it is generic and versatile, meaning that it can be applied to several types of analog circuits, (c) it can be easily migrated from one technology to another, (d) it is transparent to the circuit under test, meaning that it does not degrade the performances, (e) it incurs an affordable low-area overhead, (f) it requires an affordable number of extra test pins, (g) it incurs minimum design modifications without increasing design iterations, (h) it reduces the standard test cost without significant test development effort, and (i) it achieves in practice equivalent or at least comparable test quality as compared to the standard test approach.

Given an alternative test technique it is rather straightforward to argue about criteria (a)-(h). However, arguing about criterion (i) and offering an equivalence proof is not a straightforward task. Typically, test quality is measured in terms of test metrics, such as test escapes (i.e. faulty circuits passing the test) and yield loss (i.e. functional circuits failing the test). It is required to have efficient test metrics estimation methods to be able to evaluate newly proposed alternative test techniques during their development phase. The goal is to provide confidence about their effectiveness or summarily abandon them early in the development phase based on simulation and before moving to silicon prototypes and data collection in high-volume manufacturing. Unless we have a strong indication that an alternative test technique is effective, we take a risk by moving to high-volume production and we may find ourselves in front of an unpleasant result that the alternative test technique is actually ineffective. In this case, we would have lost significant resources only to start from scratch searching for another alternative test technique to reduce costs.

To understand the challenge in estimating test metrics, we can distinguish two different scenarios, depending on where

the performance specifications lie with respect to their nominal expected value by design. Let us consider test escape without loss of generality. Assuming a performance that is normally distributed, if its specification lies at 3, 3.29, 3.89, 4, and 5 standard deviations (σ) from the nominal value, then, respectively, we will observe on average approximately 2700, 1000, 100, 63, and 0.57 parts per million (PPM) that are faulty due to process variations. If all specifications are set at beyond 4σ, then the circuit is robust by design, it practically never fails due to process variations, and any failure is due to defects. In this scenario, relying solely on defect-oriented test is an appropriate strategy [4]. However, in the case of aggressive designs in advance technology nodes where some specifications lie below 4σ, then we are also concerned about test escapes due to process variations (i.e. parametric test escape).

By default, parametric test escape has an upper bound equal to the faulty PPM. The most straightforward approach to obtain an estimate is to perform a Monte Carlo transistor-level simulation analysis, where in each run we save the values of performances and alternative tests. In the end, we can either estimate test escape based on relative frequencies and/or examine the correlation between performances and alternative tests at the tails of the distribution where test escape events occur. Since the true value may be in the order of a few hundred or tens PPM, with any reasonable size Monte Carlo sample we would not have come across any such events, thus erroneously estimating a zero test escape. Obtaining an estimate with a relatively good confidence may require the simulation of millions of Monte Carlo samples. Clearly, this is impractical for any realistic analog circuit, thus necessitating an alternative approach.

In the context of parametric test metrics evaluation, several fast Monte Carlo alternative approaches have been proposed to date. Approaches based on density estimation [5], Copulas theory [6], extreme value theory [7], importance sampling [8], and generation of parametric fault models [9], [10] require that the circuit can be simulated at transistor-level at least a few hundred or thousand times (e.g. amplifiers, filters, mixers, bandgap reference, etc.), thus they are not applicable for circuits with long simulation times (e.g. analog-to-digital converters (ADCs), phase locked loops (PPLs), etc.). For circuits with long simulation times, alternative approaches proposed to date result in parametric test metrics estimates that are not verifiable [11].

II. PRACTICAL SIMULATION FLOW FOR PARAMETRIC TEST METRICS EVALUATION

In this paper, we propose a practical simulation flow for evaluating parametric test metrics in the case of circuits with long simulation times. The practical simulation flow is illustrated in Fig. 1 and is presented next in more details in a bottom-up fashion.

A. Behavioral modeling

The first step is to develop a behavioral model of the circuit. A behavioral model is constructed by decomposing the circuit into independent sub-circuits, creating a separate model for each sub-circuit that captures its functionality, and then linking these models and manipulating the data flow so as to compute the circuit performances and the alternative tests. The behavioral model has as inputs a set of behavioral parameters that correspond to key performances of the sub-circuits as well as to noise sources.

For circuits with long simulation times, such a behavioral model is always available during the design phase and is used repeatedly to guide the design. The behavioral model helps to find the best trade-off between the performances of the different sub-circuits and, thereby, helps to dissociate as much as possible the design of the sub-circuits, speeding up design iterations. Thus, the development of the behavioral model does not introduce an extra effort towards our objective.

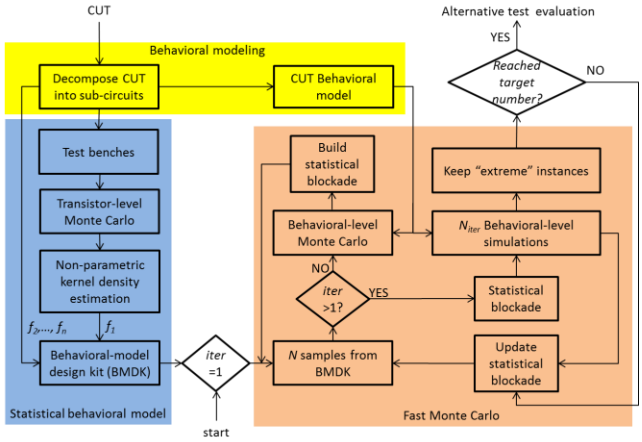

Figure 1: Practical simulation flow.

B. Statistical behavioral model

The next step is to be able to run Monte Carlo simulations at behavioral level. For this purpose, we propose to rely on a behavioral model design kit (BMDK), similar to how the process design kit (PDK) is employed in Monte Carlo transistor-level simulations.

The behavioral parameters that correspond to sub-circuit performances are correlated, drawing upon the correlations that exist amongst the low-level process parameters, as these are defined in the PDK. Our approach is to estimate their joint probability density function (PDF), denoted by f_1. To address the scenario where f_1 does not follow a known parametric form (e.g. Gaussian, etc.), we use non-parametric kernel density estimation (KDE), which makes no assumption about the underlying form of the distribution [5].

Estimating f_1 requires an initial Monte Carlo sample of behavioral parameters. The larger this sample is, the more accurate the estimate will be. However, a single transistor-level simulation of the circuit to generate a single sample may take more than one day to complete. The simulation burden can be drastically reduced by performing Monte Carlo simulations at transistor-level independently for each sub-circuit, using appropriate test benches to extract the performances that are

directly employed as behavioral parameters. The key is to use the same Monte Carlo seed for every test bench to ensure that the correlations between the behavioral parameters are captured effectively.

The behavioral parameters that correspond to noise sources are independent and are modeled by appropriate PDFs, denoted by $f_2 \ldots f_n$.

The BMDK consists of the PDFs $f_1 \ldots f_n$. One run in the behavioral-level Monte Carlo simulation consists of obtaining an independent sample from $f_1 \ldots f_n$ to create an instance of behavioral parameters, creating the behavioral model instance, and subsequently running the behavioral-level simulation.

C. Fast Monte Carlo

Although behavioral-level simulation is considerably faster than transistor-level simulation, it is in the order of minutes and, thereby, Monte Carlo behavioral-level simulation is still not time-efficient enough for the purpose of evaluating parametric test metrics.

To speed up behavioral-level Monte Carlo simulation, we employ the statistical blockade technique [10]. The underlying idea is to decide before simulating a behavioral model instance if this simulation is likely to result in an "extreme" circuit. An extreme circuit is defined as a circuit that lies towards the tails of the design distribution where performances fail and where test escape and yield loss events may occur. If an extreme circuit is likely to be generated, then the simulation is allowed. Otherwise, the simulation is blocked and we proceed by sampling another behavioral model instance from the BMDK. In other words, we aim at focusing the simulation effort on extreme circuits that are useful for the purpose of evaluating parametric test metrics.

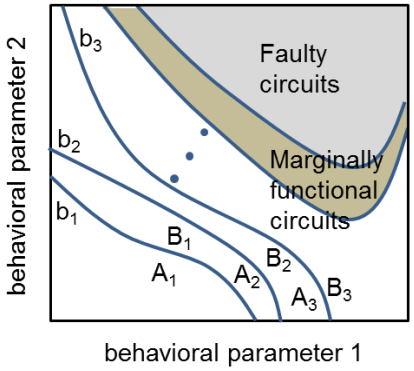

Figure 2: Statistical blockade algorithm.

The statistical blockade decision is based on a classification boundary in the space of behavioral parameters, as illustrated in Fig. 2. This classification boundary is updated in subsequent iterations so as to fine-tune the decision and save significant simulation time. In the first iteration, we perform a Monte Carlo behavioral-level simulation with N runs and we calculate the performance median. The N samples are divided into two groups, one group A_1 having performance lower than the median and one group B_1 having performance larger than the median. The classification boundary b_1 is allocated to separate groups A_1 and B_1 in the space of behavioral parameters. Without loss of generality, let us assume that the performance has an upper specification. In this case, the group B_1 is the more extreme, in the sense it lies closer to the area of faulty circuits that violate the specification. In the second iteration, we sample N behavioral model instances based on the BMDK and we classify them based on the classification boundary b_1. Obtaining a sample using the BMDK and classifying the sample can be done very quickly. What is time consuming is simulating the sample. We decide to simulate only the N_2 out of N samples that are classified to belong to group B_1, since these samples are likely to be the more extreme. In this way, we block N-N_2 simulations of non-extreme samples. Then, the N_2 instances are divided into a group A_2 having performance lower than the median and a group B_2 having performance larger than the median, where the median is recalculated based on the N_2 instances. The classification boundary is updated to classification boundary b_2 that separates groups A_2 and B_2 and is closer to the area of faulty circuits. In subsequent iterations we move the classification boundary to increase the chances that a faulty circuit will be simulated, while at the same time minimizing the overall simulation effort by blocking simulations that are highly unlikely to produce a faulty circuit. In practice, in a few iterations and with an affordable number of simulations, we end up simulating an extreme set of faulty circuits and marginally functional circuits (i.e. circuits that pass marginally the specification) that has sufficient size to evaluate parametric test metrics.

D. Parametric test metrics evaluation

Given the extreme set of faulty and marginally functional circuits, we can study the correlation between the performances and alternative tests at the tails of the design distribution where test escape and yield loss may occur. A high degree of correlation implies that the alternative tests track the performances and, thereby, can replace the standard tests without sacrificing test quality. Parametric test metrics can also be projected to PPM and confidence intervals by fitting an extreme value model to the extreme set of circuits [7].

III. CASE STUDY

Our case study is a BIST strategy for measuring the Signal-to-Noise and Distortion Ratio (SNDR) of ΣΔ ADCs, originally proposed in [2].

The BIST relies on injecting into the input of the modulator a ternary digital test stimulus that encodes a high-resolution sine-wave with programmable amplitude to excite the complete full scale of the ΣΔ ADC. The test stimulus is generated on-chip by combining a digital ΣΔ bitstream encoding a high-resolution sine-wave with a delayed version of itself. The digital ΣΔ bitstream is generated through software and is loaded beforehand into a circular shift-register. The test stimulus is injected into the input of the modulator through a 3-level Digital-to-Analog Converter (DAC). This requires adding four switches at the input of the modulator, which is a minor and non-intrusive modification.

The ADC in our case study is an 18-bit 40nm CMOS ΣΔ ADC designed by STMicroelectronics. Fig. 3 shows a

simplified schematic of the modified 2:1 MASH modulator, including the ternary digital test stimulus generator. The 3-level DAC interfaces the three digital symbols +1, 0, and -1 in the ternary digital test stimulus to the three analog differential levels $+V_{ref}$ = REF^{+}-REF^{-}, 0, and $-V_{ref}$ = REF^{-}- REF^{+} of the modulator.

Fig. 4 shows a microphotograph of the fabricated chip. The chip also includes a response analyzer, which computes the SNDR based on a purely digital sinewave fitting algorithm, and the BIST control circuitry, which manages test execution and provides a standard digital access to load different test configurations and read out test results. The BIST is fully digital with the exception of the modification at the input of the modulator. The area overhead is 7.1% of the area of the complete ΣΔ ADC.

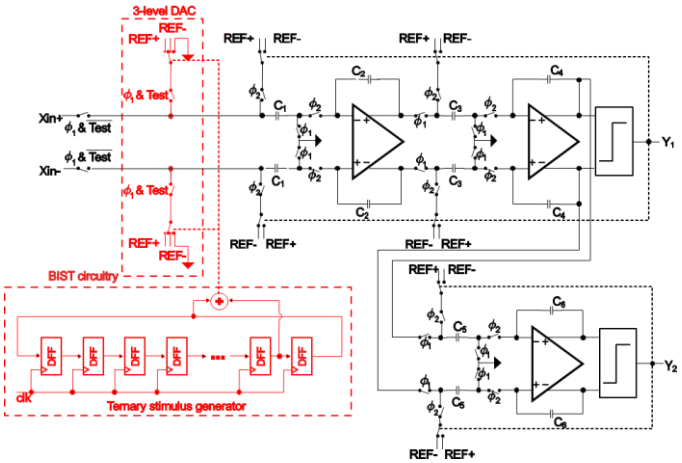

Figure 3: ΣΔ modulator DfT.

Fig. 5 shows the measured SNDR by the BIST and the standard test for different input amplitudes using a sampling frequency at one tenth of the GBW and an oversampling ratio of 128. The standard test consists of applying a high-resolution analog sine-wave test stimulus. The constant offset is due to the noise contribution and input signal attenuation of the anti-aliasing filter that is placed before the modulator and, thereby, it is not included in the BIST path. As it can be seen, the two curves are very well correlated, pointing to the equivalence between the BIST and standard test for the functional prototype chip. However, to give the green light to implement the BIST in high-volume production, this equivalence should be proven for any chip that will be produced, including corner chips. The practical simulation flow proposed in this work is employed to perform this study.

IV. RESULTS USING THE PRACTICAL SIMULATION FLOW

The simulation analysis considers only the modulator since the rest of the blocks (i.e. test stimulus generator, response analyzer, BIST control circuitry, decimation filter of the ADC, etc.) are fully digital. A single simulation to obtain the SNDR for a given input amplitude with an accuracy of 1dB takes more than one day to complete.

To this end, we developed a realistic behavioral model of the modulator following the guidelines provided in [12]. The sub-circuits of the modulator are the integrators, the comparators, the 3-level DACs, and the bandgap reference voltage generators. The behavioral parameters include the gain-bandwidth product (GBW), slew-rate (SR), open-loop gain, and noise in the operational amplifier of the integrators, the saturation levels of the comparators, the reference voltages, the sampling jitter noise, and the KT/C noise. These behavioral parameters capture the main sources of dynamic non-linearity and noise in the modulator that result in SNDR variations. The behavioral model excludes other non-idealities, such as charge injection, clock feedthrough, capacitor nonlinearity, non-linearity of the 3-level DAC, kickback noise from comparators to integrators, etc. Based on our evaluations, their impact on the dynamic performance is far less significant. Besides, a behavioral model cannot include all possible non-idealities because the equations would be unsolvable [12]. Finally, the dynamic performance of the modulator is mainly limited by the first integrator, thus, for simplicity, the second and third integrators, as well as the comparators, are modeled as ideal blocks [12].

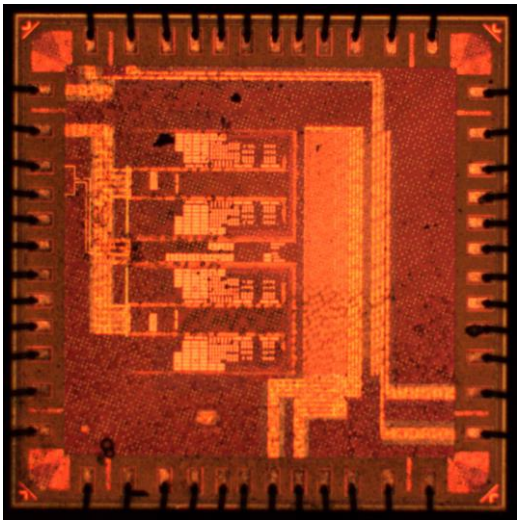

Figure 4: Photo of fabricated chip.

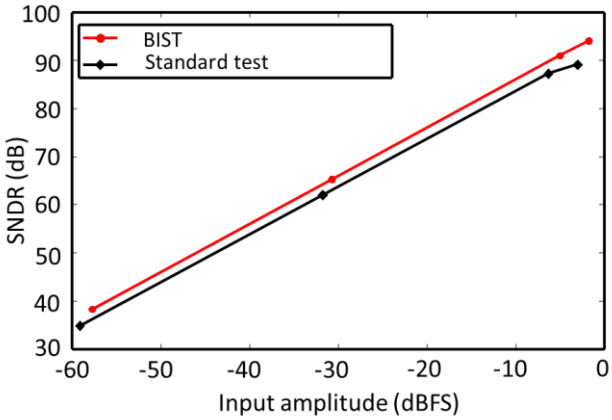

Figure 5: Equivalence between BIST and standard test for a prototype chip.

Fig. 6 shows the SNDR as a function of the input amplitude at the nominal design point using both transistor-level and behavioral-level simulations. As it can be observed, there is a very good agreement between the transistor-level and behavioral-level simulations across the whole amplitude range, demonstrating the accuracy of the behavioral model.

For each sub-circuit, a Monte Carlo simulation with 1000 runs is performed using appropriate test benches and the actual PDK so as to extract all the relevant performances that are employed as behavioral parameters. The same Monte Carlo seed is shared for all test benches. The 1000 behavioral parameter instances are used to estimate the joint PDF f_1 using non-parametric KDE. Fig. 7 projects behavioral parameter instances from transistor-level simulation and "synthetic" behavioral parameter instances resulting from BMDK sampling onto a 3-dimensonal behavioral parameter space. As it can be seen, the simulated and synthetic behavioral parameter instances are practically indistinguishable, demonstrating the accuracy of the BMDK.

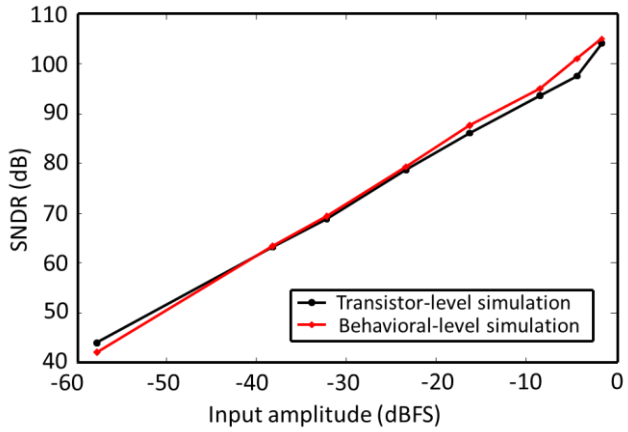

Figure 6: Comparison between transistor-level and behavioral-level simulations.

The correlation between the BIST and standard test is studied for an input amplitude of -2.3 dBFS that is very close to the full scale, a sampling frequency at one fourth of the GBW, and an oversampling ratio of 128. We chose this operating point because it is a challenging scenario for the BIST strategy. The SNDR specification is set at 4σ and equals *spec*=101.19 dB.

Fig. 8 shows the histogram of SNDR values of behavioral model instances that are simulated during multiple iterations of the statistical blockade algorithm. As it can be seen, in subsequent iterations the histogram moves to the left, closer to the SNDR specification. This demonstrates the effectiveness of the classifier in distinguishing extreme from non-extreme behavioral model instances. After 4 iterations, we have generated our target number of more than 100 faulty circuits that violate the SNDR specification. In the course of the algorithm, we have also generated a large number of marginally functional circuits whose SNDR value lies between *spec* and *spec*+ε, where ε=0.8 dB.

The faulty circuits and two limited random sets of marginally functional and centered circuits (i.e. with SNDR values around the nominal design point) are simulated next at behavioral level to extract the SNDR considering both the BIST and the standard test. The results are plotted in Fig. 9. The correlation is excellent across the design space and almost linear with a knee that lies well into the region of functional circuits. We observe that the SNDR extracted by the BIST is lower, resulting in an offset in the opposite direction as compared to Fig. 5. This is explained by the fact that the sampling frequency is higher, i.e. one fourth instead of one tenth of the GBW. As a result, the ternary digital test stimulus, unlike the ideal analog sine-wave, induces settling errors in the first integrator of the modulator, thus degrading the SNDR value. We observe also that there is some overlapping due to some marginally faulty circuit instances now satisfying the SNDR specification and, vice versa, some marginally functional circuit instances now violating the SNDR specification. Finally, we observe that there is a spread in the points. The overlapping and the spread are both due to noise in the modulator and the limited number of samples that are used to compute the Fast Fourier Transform (FFT) from which the SNDR is derived. Fig. 9 is a strong indication that the BIST strategy achieves high fault coverage and low yield loss and, thereby, we can pursue it further and proceed with implementing it in high-volume production. Strictly speaking the methodology does not provide a formal proof that the BIST is equivalent to the standard test in terms of test accuracy, since there are possible sources of inaccuracy related to the behavioral model, the BMDK, and the statistical blockade decisions. A formal proof can only be claimed after obtaining high-volume silicon data. The purpose of the methodology is to confine significantly the risk that the BIST will be proven ineffective in the end.

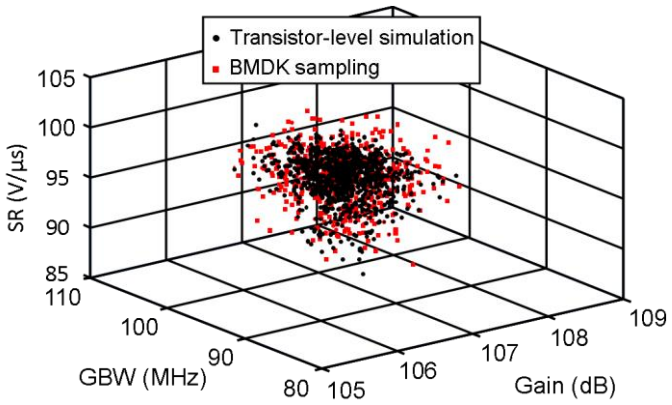

Figure 7: Comparison between simulated and sampled behavioral parameters.

Regarding computation time, it took 50 minutes to run the 1000 transistor-level simulations of the sub-circuits, 1 minute to derive the BMDK, a few seconds to obtain one sample from the BMDK, and less than 1 minute to run a behavioral-level simulation. The majority of the time is consumed for running the behavioral-level simulations in the statistical blockade loop. Overall, it took about 6 hours to produce the data in Fig. 9, in order to complete the BIST evaluation.

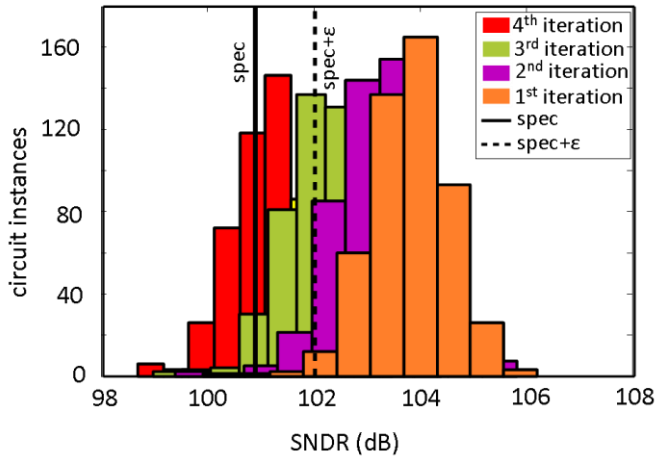

Figure 8: Progression towards the tails of the distribution.

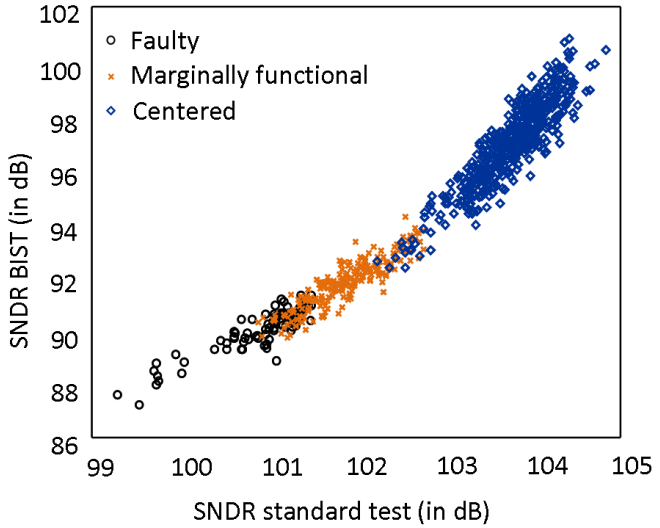

Figure 9: Correlation between standard and alternative test.

V. CONCLUSIONS

We presented an overview of a practical simulation flow that has been developed for evaluating alternative test solutions specifically for analog and mixed-signal circuits with long simulation times. The simulation flow is semi-automated requiring only a behavioral model of the circuit and Monte Carlo transistor-level simulation data from its sub-circuits. The behavioral model and the decomposition of the circuit into sub-circuits are tasks already completed by the designer. The same simulation flow can be used for circuits with fast simulation times, in which case it is fully automated. In particular, referring to the blocks in the simulation flow in Fig. 1, the "behavioral modelling" and "statistical behavioral model" blocks are removed and behavioral-level simulations are replaced with transistor-level simulations in the "fast Monte Carlo" block. The practical simulation flow was demonstrated on a ΣΔ ADC for proving a BIST strategy.

ACKNOWLEDGMENT

This research has been carried out in the framework of the project ENIAC ELESIS.

REFERENCES

[1] J. Kim, "On-chip measurement of jitter transfer and supply sensitivity of PLL/DLLs," *IEEE Transactions on Circuits and Sysems II: Express Briefs*, vol. 56, no. 6, pp. 449–453, 2009.

[2] M. Dubois *et al.*, "Ternary stimulus for fully digital dynamic testing of SC ΣΔ ADCs, " *IEEE International Mixed-Signals, Sensors, and Systems Test Workshop*, 2012.

[3] A. Coyette *et al.*, "Automated testing of mixed-signal integrated circuits by topology modification," *IEEE VLSI Test Symposium*, 2015.

[4] S. Sunter *et al.*, "Practical random sampling of potential defects for analog fault simulation," *IEEE International Test Conference*, 2014, paper 2.3.

[5] N. Kupp *et al.*, "On proving the efficiency of alternative RF tests, " *IEEE/ACM International Conference on Computer-Aided Design*, , pp. 762-767

[6] A. Bounceur *et al.*, "Estimation of analog parametric test metrics using copulas, " *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 9, pp. 1400-1410, 2011.

[7] H.-G. Stratigopoulos, "Test metrics model for analog test development, " *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 1116-1128, 2012

[8] E. Yilmaz and S. Ozev, "Adaptive-learning-based importance sampling for analog circuit DPPM estimation," *IEEE Design & Test*, vol. 32, no. 1, pp. 36 – 43, 2015.

[9] M. B. Yelten et al., "Scalable and efficient analog parametric fault identification," *IEEE/ACM International Conference on Computer-Aided Design*, 2013, pp. 387 – 392.

[10] H.-G. Stratigopoulos and S. Sunter, "Fast Monte Carlo-based estimation of analog parametric test metrics," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1977–1990, 2014.

[11] M. Dubois et al., "Evaluation of digital ternary stimuli for dynamic test of ΣΔ ADCs, " *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2014.

[12] P. Malcovati et al., "Behavioral modeling of switched-capacitor sigmadelta modulators," *IEEE Transactions on Circuits and Systms I: Fundamental Theory and Applations*, vol. 50, no. 3, pp. 351–364, 2003.