Evaluation of low-cost mixed-signal test techniques for circuits with long simulation times

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

Evaluation of Low-Cost Mixed-Signal Test Techniques for Circuits with Long Simulation Times

Haralampos-G. Stratigopoulos*, Manuel J. Barragan†‡, Salvador Mir†‡, Hervé Le Gall§,
Neha Bhargava¶, and Ankur Bal¶
*Sorbonne Universités, UPMC Univ. Paris 06, CNRS, LIP6, 4 place Jussieu, 75005 Paris, France
†Université Grenoble Alpes, TIMA, F-38000 Grenoble, France
‡CNRS, TIMA, F-38000 Grenoble, France
§ST Microelectronics, 12 Rue Jules Horowitz, 38000 Grenoble, France
¶ST Microelectronics, Plot No. 1, Knowledge Park III, 201308, Greater Noida, India

***Abstract*— The high cost of mixed-signal circuit testing has sparked a lot of interest for developing alternative low-cost techniques. Although it is rather straightforward to evaluate an alternative test technique in terms of test cost reduction, proving the equivalence between an alternative and the standard test technique in terms of test metrics, before actually deploying the alternative test technique in production, is very challenging. The underlying reason is the prohibitive simulation effort that is required. Existing test metrics evaluation methodologies are efficient only for circuits that can be simulated fast at transistor-level. In this paper, we propose a test metrics evaluation methodology for circuits with long simulation times that is based on a combination of behavioral modeling and statistical blockade. The methodology is demonstrated on a built-in self-test strategy for $\Sigma\Delta$ analog-to-digital converters.**

I. Introduction

Testing the mixed-signal functions of Systems-on-Chip (SoC) and Systems-in-Package (SiP) is responsible for a large fraction of the overall manufacturing cost [1]. The standard test technique followed today is specification testing where all the performances that are promised in the datasheet are measured sequentially [2]. The high test cost is due to the specialized automatic test equipment (ATE) that is required and the corresponding long test times. To reduce the test cost, various alternative test paradigms have been proposed, including structural defect-oriented test [3], built-in test [4], [5], built-off test [6], alternate test [7], and statistical learning [8]. The aim of any such alternative test technique is to alleviate the ATE requirements, for example, by performing the test on low-cost digital ATE, and to reduce the test time, for example, by reducing the number of test configurations, the volume of data that needs to be loaded and processed on the ATE, the pure electrical test time, etc.

Although it is typically an easy exercise to project test cost savings resulting by employing alternative test techniques, it is not as straightforward to prove that an alternative test technique maintains a test accuracy that is comparable, if not equivalent, to the test accuracy of the standard test technique. Ideally, the test accuracy, typically expressed by the test coverage and yield coverage metrics or, equivalently, by the test escape and yield loss metrics, should be evaluated during the design and test development phases through simulation. This is important so as to be able to adjust, refine, or abandon the alternative test technique at this early stage before moving to high-volume production. Otherwise, we would need to apply both the standard and alternative test techniques repeatedly on a high-volume of fabricated circuits, in order to examine how well they correlate. If at this late stage we conclude that the alternative test technique results in insufficient test accuracy, then we would have wasted significant resources and face a significant setback in the test development.

The challenge for evaluating alternative test techniques through simulation is to be able to perform the analysis in a time-efficient manner. Regarding catastrophic defects, the design of the simulation campaign to evaluate the test coverage is straightforward, however, in the case where the number of defects is too high, sophisticated defect sampling techniques must be used [9]. On the other hand, evaluating alternative test techniques against process variations is far more challenging. The reason is that the specifications are set to 3σ or higher and, thereby, a straightforward Monte Carlo simulation that generates failing instances to evaluate test coverage and marginally functional instances to evaluate yield loss is not a viable option since an intractable number of iterations are required. We use the term *parametric test metrics* to refer to the test coverage and yield loss in the case where only process variations are considered and catastrophic defects are excluded from the analysis.

Several fast alternatives to Monte Carlo simulation exist [10]. In the context of parametric test metrics evaluation, several approaches have been studied, including density estimation [11], [12], Copulas theory [13], extreme value theory [14], [15], importance sampling [16], and generation of parametric fault models [17], [18]. However, all these approaches make the tacit assumption that the circuit can be simulated at transistor-level at least a few hundreds or thousand times. Therefore, they are not applicable for circuits with long simulation times, such as analog-to-digital converters (ADCs) and phase locked loops (PPLs).

Approaches applicable to circuits with long simulation times are proposed in [19], [20]. The underlying idea to speed up the analysis is to replace transistor-level simulations with behavioral-level simulations. The behavioral parameters are extracted by following a divide and conquer approach to

H.-G. Stratigopoulos was with Université Grenoble Alpes, CNRS, TIMA, Grenoble, France. He is now with Sorbonne Universités, UPMC Univ. Paris 06, CNRS, LIP6, France.

decompose the circuit into individual blocks that can be simulated independently at transistor-level. In [19], the Monte Carlo analysis is performed at behavioral-level by sampling an estimate of the joint probability density function (PDF) of behavioral parameters to quickly generate millions of "synthetic" circuit instances so as to be able to estimate test metrics with parts-per-million (PPM) using relative frequencies. In [20], a hyper-rectangular in the space of behavioral parameters is defined based on the minimum and maximum values of the behavioral parameters. Thereafter, uniform sampling is performed within this hyper-rectangular to generate "synthetic" faulty and marginally functional circuit instances so as to be able to study the correlation between alternative and standard test techniques beyond the nominal design point. The disadvantage of the approach in [19] is that the PDF is estimated based on centered circuit instances, thus, the tails of the PDF where the faulty and marginally functional circuit instances occur are inaccurately estimated, resulting, in turn, in inaccurate parametric test metrics. The disadvantage of the approach in [20] is that the extracted faulty and marginally functional circuit instances are not necessarily realistic and feasible circuit instances, thus, there is little confidence in the resultant correlation between alternative and standard test techniques. In this paper, we propose a methodology that also uses behavioral modeling as a mean to speed up the analysis, yet it builds upon the ideas proposed in [18] to speed up further the analysis so as to be able to generate a realistic and feasible set of faulty and marginally functional circuit instances based on which the correlation between alternative and standard test techniques can be studied with confidence.

The rest of the paper is structured as follows. In Section II, we provide an overview of the proposed methodology and we present the detailed algorithm. In Section III, we present our case study which is a built-in self-test (BIST) technique for $\Sigma\Delta$ ADCs. In Section IV, we demonstrate our methodology on this case study. Finally, Section V concludes the paper.

II. Parametric Test Metrics Evaluation Methodology

A. Overview

Our approach to parametric test metrics evaluation is to generate in a time-efficient manner two sets of circuit instances, namely a *fault model*, which contains a set of circuit instances that fail the specification limit of at least one performance, and a set of *marginally functional circuit instances* that have at least one performance that lies close to its specification limit. These two sets that comprise *extreme* circuit instances are used to examine the correlation between the alternative and standard test techniques at the tails of the design distribution where test escape and yield loss events occur.

Our aim is to stick to the Monte Carlo approach so as to cover effectively the design space instead of running selective corner simulations, which inevitably could lead to misleading conclusions. However, this entails several challenges that we have to overcome. To speed up simulation, we first replace the transistor-level design with a behavioral model that captures effectively the circuit functionality, as well as the non-idealities (i.e. noise sources, non-linearity, etc.). 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 and achieve the best possible performance trade-offs. Thus, the development of the behavioral model does not introduce an extra effort towards our objective.

The next step is to be able to run Monte Carlo simulations at behavioral-level. Our approach is to estimate the joint PDF of behavioral parameters and use it as a behavioral model design kit (BMDK), similar to how the process design kit (PDK) is employed in Monte Carlo transistor-level simulations. To address the scenario where the PDF does not follow a known parametric form (i.e. Gaussian, etc.), we use non-parametric kernel density estimation (KDE) which makes no assumption about the underlying form of the distribution [21].

To estimate the joint PDF of behavioral parameters, we require an initial Monte Carlo sample of behavioral parameters which draws upon the correlations that exist amongst the low-level process parameters, as these are defined in the actual PDK of the technology. Our approach is to run a Monte Carlo simulation at transistor-level and use appropriate test benches to extract the behavioral parameters. For this purpose, taking into consideration that a single transistor-level simulation of the complete circuit may take more than one day to terminate, we decompose the circuit into individual simpler blocks that can be simulated independently of each other at transistor-level, much faster than the complete circuit itself. Test benches using one block at a time are designed to extract all the relevant behavioral parameters. The key is to use the same Monte Carlo seed for every test bench such that the same global process parameter vectors are visited in each Monte Carlo run. This ensures that the correlations between the behavioral parameters are captured efficiently.

To perform a Monte Carlo simulation at behavioral level, we use the BMDK to sample a behavioral parameter vector in each run. Although behavioral simulation is considerably faster than transistor-level simulation, it is in the order of minutes and, thereby, it is still not time-efficient enough for the purpose of evaluating parametric test metrics. To address this challenge, we employ the statistical blockade technique [22], [23]. The underlying idea is to decide whether a circuit instance is extreme before actually running the behavioral simulation since any non-extreme circuit instance is worthless for evaluating parametric test metrics. If a circuit instance is unlikely to be extreme, then the behavioral simulation is blocked and we proceed by sampling a new behavioral parameter vector. The decision block that is introduced into the Monte Carlo loop is fine-tuned in the course of the algorithm so as to increase the confidence on whether to run a behavioral simulation and, thereby, save a maximum of simulation effort.

B. Detailed algorithm

In this Section, we present the detailed algorithm for parametric test metrics evaluation. In Section IV, the various steps

of the algorithm will be demonstrated on our case study using figures where necessary.

step 1 Build a behavioral model of the circuit.
step 2 Decompose the circuit into individual blocks.
step 3 Build test benches for each individual block for extracting all relevant behavioral parameters.
step 4 Run n Monte Carlo simulations at transistor-level independently for each test bench using the same seed and extract the behavioral parameters.
step 5 Use the sample of n behavioral parameters from step 4 to build the joint PDF of behavioral parameters using non-parametric KDE. This joint PDF serves as the BMDK.

The behavioral model from step 1 and the BMDK from step 5 are the inputs to the main loop of the algorithm that is based on statistical blockade. The algorithm is run separately for each performance of interest. Let us denote a performance of interest by P and let us assume without loss of generality that it has a lower specification limit, denoted by s. Suppose that we target generating a fault model of size at least N. As a by-product, the algorithm returns a set of marginally functional circuit instances of size $M > N$.

step 6 Sample the BMDK to generate $n^{'}$ independent behavioral parameter vectors. Set $i = 1$.
step 7.i Run $n^{'}$ behavioral-level simulations using the $n^{'}$ behavioral parameter vectors and compute the performance P in each run.
step 8.i The circuit instances that satisfy $P < s$ are put into the fault model while the circuit instances that satisfy $s < P < s + |\epsilon|$ are put into the set of marginally functional circuit instances. If the size of the fault model exceeds N, then the algorithm terminates.
step 9.i Consider the ordered sequence of performances from step 7.i, denoted by $P^{(1)} \leq \cdots \leq P^{(j)} \leq \cdots \leq P^{(n^{'})}$, and calculate the q%-quantile as $u_i = P^{(j)}$, where $j = \left\lceil n^{'} q/100 \right\rceil$. In other words, $\Pr\{P \leq u_i\} \approx q/100$. We create two sets, namely the set S_{-i} that includes the $\left\lceil n^{'} q/100 \right\rceil$ circuit instances that satisfy $P \leq u_i$ and the set S_{+i} that includes the rest $n^{'} - \left\lceil n^{'} q/100 \right\rceil$ circuit instances that satisfy $P > u_i$. The set S_{-i} contains the most extreme circuit instances with performance that lies closer to the specification limit.
step 10.i Train a nonlinear binary classifier to allocate a boundary b_i that separates the sets S_{-i} and S_{+i} in the space of behavioral parameters.
step 11.i Apply the statistical blockade technique to generate a sample of $n^{'}$ behavioral parameter vectors that is more extreme than the sample used in step 7.i. Specifically, we sample the BMDK sequentially and we examine the footprint of each generated behavioral parameter vector with respect to boundary b_i. Sampling stops when we identify $n^{'}$ behavioral parameter vectors that lie in the area that corresponds to S_{-i}. The underlying idea is that a behavioral parameter that lies in the area that corresponds to S_{+i} more likely will result in a performance that satisfies $P > u_i$ if we were to run a behavioral-level simulation. For this reason, these behavioral parameters are skipped since our objective is to generate circuit instances that satisfy $P < s$ and given that, at least in the first iterations of the algorithm, $s < u_i$. In short, the aim of the statistical blockade is to concentrate the simulation effort in areas of the behavioral parameter space that are more likely to result in extreme circuit instances violating the specification limit and circumvent running a behavioral-level simulation if, with very high probability, the resultant performance will be much higher than the specification limit.
step 12.i Set $i = i+1$. Go to step 7.i where the behavioral-level simulations use the $n^{'}$ behavioral parameter vectors generated in step 11.i-1.

In successive iterations of the loop, the threshold u_i reduces and the boundary allocated by the classifier in step 10.i moves towards the true boundary that separates the functional from the faulty circuit instances in the space of behavioral parameters. In this way, the sample that is generated in step 11.i is more and more extreme and its geometric mean is approaching gradually the specification limit. After few iterations, the fault model size reaches N and the algorithm terminates. A set of marginally functional circuit instances of size M is also made available.

The parameters $n^{'}$, q, and ϵ are user-defined. $n^{'}$ should be selected as high as possible given the simulation time budget. For selecting q, there is trade-off between simulation time and how well the area of failing circuit instances is approximated. By setting $q = 50$, we maintain balanced training sets S_{+i} and S_{-i} at each iteration and we populate heavily the area around the boundary that separates them. Therefore, the allocation of the boundary b_i is more accurate and the area of failing circuit instances is well approximated in the end. However, the boundary moves slowly towards the true boundary that separates the functional from the faulty circuit instances in the space of behavioral parameters and, thereby, the algorithm takes more time to terminate. By setting q at a small value, say $q = 10$, the boundary moves faster towards the true boundary and, thereby, the algorithm terminates faster, however, the set S_{+i} may overshadow the set S_{-i}, risking not to approximate equally well the area of failing circuit instances in the end. Finally, ϵ defines the area of marginal functionality. For a large enough and practical value of ϵ, for example, setting ϵ equal to one standard deviation, a circuit instance that fails the specification limit is sampled with lower probability than a marginally functional circuit instance and, thereby, at any time during the course of algorithm, a set of marginally functional circuit instances of size $M > N$ is maintained.

III. Case Study

A. BIST for ΣΔ ADCs using digital ternary stimulus

Our case study is a fully-digital, low-cost BIST strategy for measuring the Signal-to-Noise and Distortion Ratio (SNDR) of ΣΔ ADCs, originally proposed in [24], [25]. Fig. 1 shows the

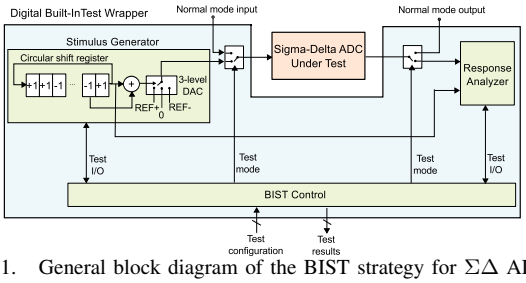

Fig. 1. General block diagram of the BIST strategy for $\Sigma\Delta$ ADCs.

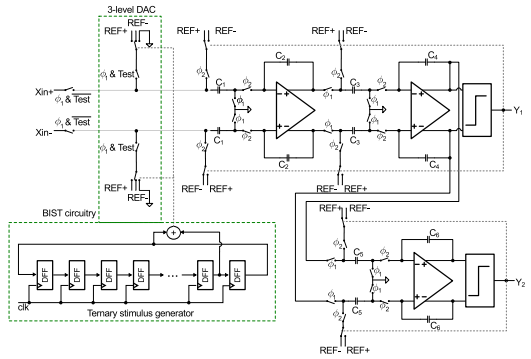

Fig. 2. Simplified schematic of the 2:1 MASH modulator including built-in test circuitry.

general block diagram of the BIST strategy. The test wrapper is composed of three digital blocks, namely the Stimulus Generator, Response Analyzer, and BIST Control. During test mode, the $\Sigma\Delta$ ADC under test is disconnected from the main signal path and is connected to the Stimulus Generator and the Response Analyzer. The Stimulus Generator provides an optimized digital test stimulus to the input whereas the Response Analyzer computes the SNDR based on purely digital algorithms. The BIST Control block manages test execution and provides a standard digital access to load different test configurations and read out test results.

The key point in the dynamic characterization of any ADC is to provide an appropriate high-linearity input test stimulus –at least 2 effective bits above the nominal linearity of the ADC– that can excite the complete full scale (FS) of the ADC. This particular BIST strategy makes use of a ternary test stimulus that encodes a high-linearity sine-wave at an amplitude close to the FS of the $\Sigma\Delta$ ADC. This ternary test stimulus is generated on-chip by combining a digital $\Sigma\Delta$ bitstream encoding a high-resolution sine-wave with a delayed version of itself. The digital $\Sigma\Delta$ bitstream is generated through software and is loaded into the circular shift register. As a result of this combination, the shaped high-frequency quantization noise in the digital $\Sigma\Delta$ bitstream can be greatly attenuated, improving the spectral performance of the ternary test stimulus. The resulting ternary test stimulus is then fed into the $\Sigma\Delta$ ADC through a 3-level Digital-to-Analog Converter (DAC). The Response Analyzer compares the output response to the reference digital $\Sigma\Delta$ bitstream and computes the SNDR based on the well-known sine-wave fitting algorithm. The interested reader is referred to [24], [25] for an in-depth description of the BIST strategy.

B. $\Sigma\Delta$ ADC and behavioral modeling

The ADC under test in our case study is a 18-bit $\Sigma\Delta$ ADC designed in a 40nm CMOS technology and provided by STMicroelectronics. Fig. 2 shows a simplified schematic of the modulator of the $\Sigma\Delta$ ADC which is a fully-differential 2:1 MASH. Fig. 2 includes the circular shift register of the Stimulus Generator and shows how the input stage of the modulator is modified for injecting the ternary test stimulus. The required 3-level DAC, shown in Fig. 1, has been merged together with the input stage of the modulator simply by adding four input switches [25]. This DAC interfaces the three digital symbols '$+1$', '0', and '-1' in the ternary test stimulus to the three analog differential levels $+V_{ref} = \text{REF}^+ - \text{REF}^-$, 0, and $-V_{ref} = \text{REF}^- - \text{REF}^+$, respectively, in order to retrieve the analog sine-wave encoded in the ternary test stimulus and inject it at the input of the modulator. The decimation filter of the $\Sigma\Delta$ ADC and the Response Analyzer are completely digital blocks and they are not considered in our analysis.

Direct transistor-level simulation of the complete system in Fig. 2 takes up very long times. Specifically, a single simulation to obtain the SNDR of the 2:1 MASH modulator for a given input amplitude with an accuracy of 1dB takes more than one day to complete on dedicated simulation servers. To this end, according to steps 1-3 of the algorithm in Section II-B, we developed a realistic behavioral model following the guidelines provided in [26], [27] and we decomposed the modulator into simpler building blocks that can be efficiently and separately simulated at transistor-level. The main sources of dynamic non-linearity and noise in the modulator that result in SNDR variations are modeled explicitly as behavioral parameters in the behavioral model. In particular, we consider the finite gain-bandwidth product (GBW), limited slew-rate (SR), finite open-loop gain, reference voltage of the modulator, and noise of the amplifier in the first integrator and we set up transistor-level test benches including only the first integrator to extract these behavioral parameters. In addition, we consider jitter noise and kT/C noise. Notice that in behavioral-level Monte Carlo simulations all noise sources are treated as random variables and only the GBW, SR, open-loop gain, and reference voltage of the modulator vary according to their distribution in the BMDK reflecting process variations in the PDK of the technology. 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 [26], [27]. In addition, the behavioral model excludes other non-idealities, such as charge injection, clock feedthrough, capacitor non-linearity, non-linearity of the 3-level DAC, kickback noise from comparators to integrators, etc., since, based on our evaluations, their impact on the dynamic performance is far less significant compared to the aforementioned modeled non-idealities and given that a behavioral model cannot include all possible non-idealities because the equations would be unsolvable [26], [27].

Fig. 3 shows the SNDR of the complete 2:1 MASH modulator as a function of the input amplitude at the nominal design point using both transistor-level and behavioral-level

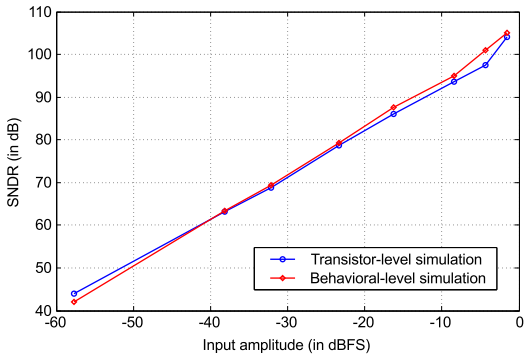

Fig. 3. SNDR vs. input amplitude curves obtained by transistor-level and behavioral-level simulation.

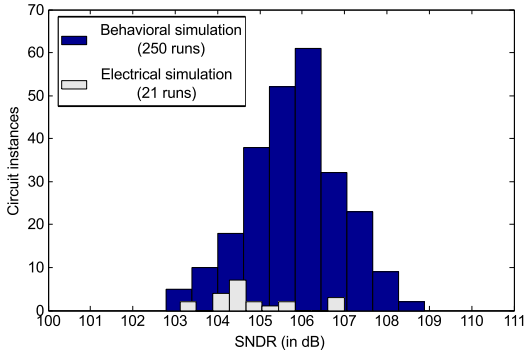

Fig. 4. SNDR histograms obtained by transistor-level and behavioral-level Monte Carlo simulations using, respectively, the PDK and the BMDK.

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. Regarding simulation time, it took about one week to run the set of transistor-level simulations, while the set of behavioral-level simulations was completed in about 1.5 minutes.

In Fig. 4, we consider the maximum input amplitude and we compare the SNDR histogram obtained by transistor-level Monte Carlo simulation using the technology PDK with the SNDR histogram obtained by behavioral-level Monte Carlo simulation using the developed BMDK. In the case of transistor-level Monte Carlo simulation, it took about one month to perform the 21 runs, while in the case of behavioral-level simulation it took only 50 minutes to perform 250 runs. As it can be observed, the average SNDR in the case of transistor-level simulations is 104.8 dB with a standard deviation 1.07 dB, while in the case of behavioral-level simulations the average SNDR is 105.7 dB with a standard deviation 1.09 dB. Although the statistical significance of the SNDR histogram obtained by transistor-level Monte Carlo simulation is limited by the low number of runs, which are still very time-consuming and prohibitive to perform in practice, the two SNDR histograms show a good agreement.

IV. Results

The first step in evaluating the BIST strategy is to simulate it at the nominal design point. The results are shown in Fig. 5 for two different operating scenarios. Solid-line curves correspond to a low sampling frequency (f_s) below one tenth of the GBW

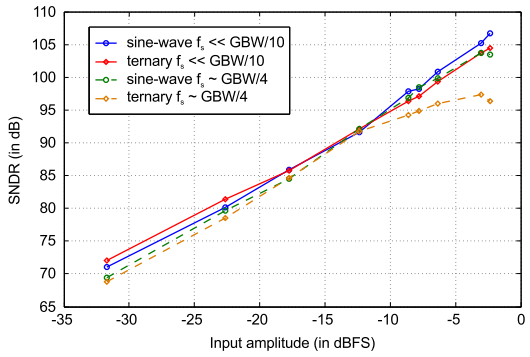

Fig. 5. SNDR measurements as a function of the test stimulus amplitude.

of the amplifiers in the integrators, while dashed-line curves correspond to a high f_s at around one fourth of the GBW. The plot shows the SNDR as a function of the amplitude of the test stimulus for two different types of test stimuli, namely an ideal (e.g. very high-resolution) sine-wave corresponding to the standard test technique and a ternary test stimulus that encodes an ideal sine-wave corresponding to the BIST strategy. In both operating scenarios, the oversampling ratio (OSR) was set to 128 such that the expected SNDR is the same. This implies that the test stimuli have a higher frequency in the high f_s scenario than in the low f_s scenario. In the low f_s scenario, we observe that the BIST provides an SNDR measurement that practically coincides across the input amplitude range with the true SNDR obtained by using the ideal sine-wave as test stimulus. In the high f_s scenario, the BIST provides an SNDR that is consistently lower for input amplitudes above -10 dBFS, however, the two SNDR measurements track each other very well and, thereby, the BIST strategy can still be used by assuming an appropriate amplitude-dependent test limit. The fact that the BIST strategy provides a lower SNDR is explained by the input-dependent settling behavior in the first integrator of the modulator. In particular, the ternary test stimulus, unlike the ideal sine-wave, induces settling errors in the first integrator, thus degrading the SNDR value.

Once the BIST strategy is evaluated at the nominal design point and is proven to be equivalent to the standard test that employs a high-resolution sine-wave, the next step is to prove that the equivalence also holds true for extreme circuit instances with excessive process variations that result in a SNDR value that marginally passes or fails the specification limit. Unless this analysis is performed, we do not have sufficient confidence for applying the BIST strategy in high-volume production test. This analysis is performed by applying the proposed parametric test metrics evaluation methodology. Next, we will show this analysis for the high-f_s scenario and an input amplitude at -2.3 dBFS. We chose this operating point because it appears to be the most challenging for the BIST strategy.

Referring to the algorithm in Section II-B, steps 1-3 were discussed in Section III-B. In step 4, we perform $n = 10^3$ transistor-level Monte Carlo simulations for the different test benches, in order to generate $n = 10^3$ behavioral-level circuit instances. These data are used in step 5 to fit a non-parametric

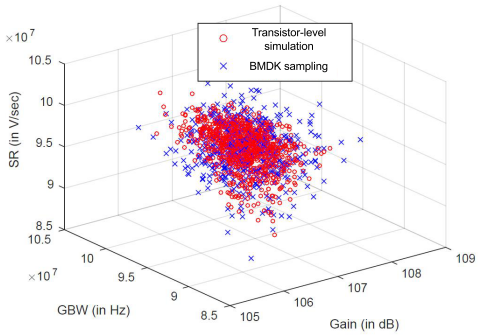

Fig. 6. Circuit instances generated through transistor-level simulation and non-parametric density sampling projected onto the space of three behavioral parameters.

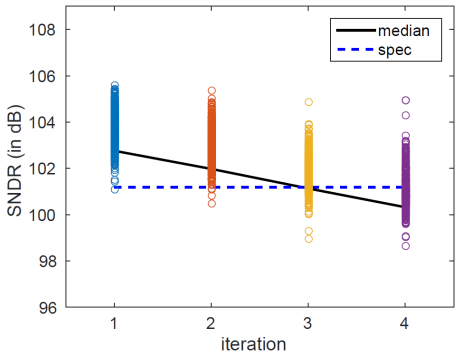

Fig. 7. SNDR values of circuit instances simulated in each iteration.

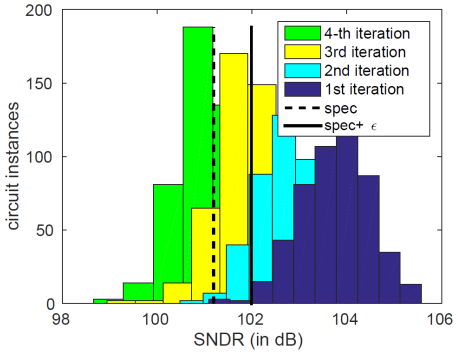

Fig. 8. Histograms of circuit instances simulated in each iteration.

joint PDF of the behavioral parameters that serves as the BMDK. The BMDK can be sampled very fast to generate new behavioral-level circuit instances. Fig. 6 projects circuit instances generated by transistor-level simulation and BMDK sampling onto the space of three behavioral parameters. We excluded from this scatter plot and from the scatter plots that will follow the reference voltage of the modulator since it exhibits the smallest variations and turns out not to be responsible for SNDR failures. As it can be observed from Fig. 6, the two populations are practically indistinguishable and, for the purpose of test metrics evaluation, BMDK sampling can replace transistor-level simulation.

Next, we run the statistical blockade loop in steps 6-11, in order to generate the set of extreme circuit instances. The SNDR has a lower specification that is set at 101.2dB, which corresponds to 4σ. We aim at generating $N = 100$ circuit instances that fail the specification. The margin ϵ is set at 0.8dB. Finally, we use $q = 10$ and in each iteration of the loop we simulate $n^{'} = 500$ circuit instances.

The loop terminates after four iterations when we have finally generated fault model of size $N = 100$. Fig. 7 shows the SNDR values of the samples of circuit instances simulated across the four iterations. Each circle point corresponds to one circuit instance. As it can be observed, in subsequent iterations the SNDR median drops linearly and the number of circuit instances in the fault model increases. Fig. 8 provides another visual illustration. It shows that in subsequent iterations the SNDR histograms move to the left, closer to the specification limit, and a larger part of the tail of the histograms lies beyond the specification. Fig. 8 also shows that the number M of marginally functional circuits that are simulated across the four iterations is, as expected, much larger than N.

Fig. 9 projects the circuits instances in the first iteration and the fault model onto the space of the same three behavioral parameters plotted in Fig. 6. As it can be observed, circuit instances in the fault model have a combination of low GBW and low SR. Since the fault model is generated in a Monte Carlo sense, we conclude that failures most likely will be due to a combination of low GBW and low SR.

Fig. 10 plots the SNDR measurements obtained by the BIST strategy and the standard test technique on the fault model and the marginally functional circuit instances, as well as on the circuit instances in the initial Monte Carlo run, which, by default, are expected to be centered around the nominal design point. As it can be observed, the two SNDR measurements correlate very well not only around the nominal design point, but, most importantly at the tails of the distribution where test escape and yield loss events may occur. The correlation is slightly nonlinear with a knee that lies well into the region of functional circuits. We observe also that some marginally faulty circuit instances are now shown to satisfy the SNDR specification of 101.2dB and, vice versa, some marginally functional circuit instances are now shown to violate the SNDR specification of 101.2dB. These circuit instances correspond to the overlapping area of the faulty and marginally functional populations in Fig. 10. Finally, we observe that there is a spread in the points around the correlation curve. 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. 10 is a strong proof that the BIST strategy achieves high fault coverage and low yield loss and, thereby, we can proceed with implementing it in high-volume production test. Additionally, given the extreme circuit instances, the extreme value theory can be used to quantify the fault coverage and yield loss probabilities [14], [15], but this out of the scope of this paper.

Regarding computation time, it took 50 minutes to run the $n = 10^3$ transistor-level simulations of the individual blocks of the decomposed circuit in step 4, 1 minute to estimate the joint PDF in step 5 and derive the BMDK, a few seconds to sample one behavioral parameter vector from the BMDK in

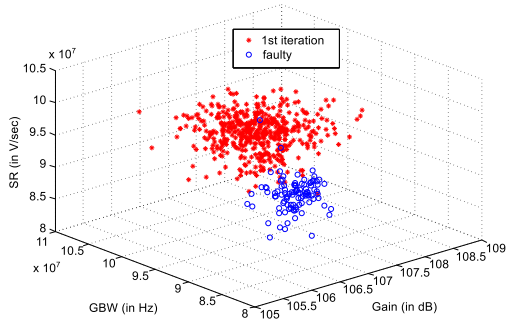

Fig. 9. Circuit instances in the first iteration and faulty circuit instances projected onto the space of three behavioral parameters.

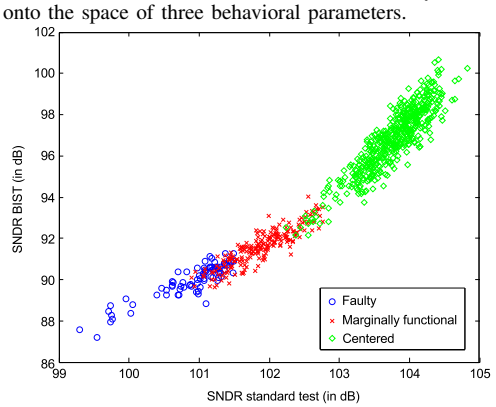

Fig. 10. True SNDR value vs. BIST measurement for faulty, marginally functional, and centered circuit instances.

step 6, and less than one 1 minute to run a behavioral-level simulation in step 7i. The majority of the time is consumed for running the behavioral-level simulations in the statistical blockade loop in steps 7i-12i. Overall, it took about 6 hours to produce the data for the plots in Fig. 6-10, in order to complete the evaluation.

V. Conclusions

We proposed a generic methodology for evaluating an alternative test technique against the standard test technique. The methodology specifically targets circuits with long transistor-level simulation times that cannot be handled by existing methodologies. It is based on running Monte Carlo simulations at behavioral-level while respecting the joint PDF of process parameters defined in the PDK. The simulation campaign is further accelerated by employing the statistical blockade technique. The output of the methodology is a set of extreme circuit instances that marginally pass or fail the specifications. This set allows to readily study the equivalence of the alternative and standard test techniques far from the nominal design point where test escapes and yield loss events may occur. The methodology was illustrated step by step for the evaluation of a BIST strategy for a 40nm 18-bit $\Sigma\Delta$ ADC.

Acknowledgement

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

References

[1] F. Poehl, F. Demmerle, J. Alt, and J. Obermeir, "Production test challenges for highly integrated mobile phone SoCs," in *Proc. IEEE European Test Symposium*, 2010, pp. 17–22.

[2] M. Burns and G. W. Roberts, *An Introduction to Mixed-Signal IC Test and Measurement*. Oxford University Press, 2nd ed., 2011.

[3] H. Hashempour, J. Dohmen, B. Tasic, B. Kruseman, C. Hora, M. Beurden, and Y. Xing, "Test time reduction in Analogue/Mixed-signal devices by defect oriented testing: An industrial example," in *Proc. Design, Automation & Test in Europe Conference*, 2011.

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

[5] D. Mannath, D. Webster, V. Montano-Martinez, D. Cohen, S. Kush, T. Ganesan, and A. Sontakke, "Structural approach for built-in tests in RF devices," in *Proc. IEEE International Test Conference*, 2010, paper 14.1.

[6] J. Ferrario, R. Wolf, S. Moss, and M. Slamani, "A low-cost test solution for wireless phone RFICs," *IEEE Communications Magazine*, vol. 41, no. 9, pp. 82–88, 2003.

[7] S. S. Akbay, J. L. Torres, J. M. Rumer, A. Chatterjee, and J. Amtsfield, "Alternate test of RF front ends with IP constraints: Frequency domain test generation and validation," in *Proc. IEEE International Test Conference*, 2006, pp. 4.4.1–4.4.10.

[8] K. Huang, N. Kupp, C. Xanthopoulos, J. M. Carulli, and Y. Makris, "Low-cost analog/RF IC testing through combined intra- and inter-die correlation models," *IEEE Design & Test*, vol. 32, no. 1, pp. 53–60, 2015.

[9] S. Sunter, K. Jurga, P. Dingenen, and R. Vanhooren, "Practical random sampling of potential defects for analog fault simulation," in *Proc. IEEE International Test Conference*, 2014, paper 2.3.

[10] F. Gong, H. Yu, Y. Shi, and L. He, "Variability-aware parametric yield estimation for Analog/Mixed-signal circuits: Concepts, algorithms, and challenges," *IEEE Design & Test*, vol. 31, no. 4, pp. 6–15, 2014.

[11] A. Bounceur, S. Mir, E. Simeu, and L. Rolindez, "Estimation of test metrics for the optimisation of analogue circuit testing," *Journal of Electronic Testing: Theory and Applications*, vol. 23, no. 6, pp. 471–484, 2007.

[12] N. Kupp, H.-G. Stratigopoulos, Y. Makris, and P. Drineas, "On proving the efficiency of alternative RF tests," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2011, pp. 762–767.

[13] A. Bounceur, S. Mir, and H.-G. Stratigopoulos, "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.

[14] 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.

[15] H.-G. Stratigopoulos, P. Faubet, Y. Courant, and F. Mohamed, "Multi-dimensional analog test metrics estimation using extreme value theory and statistical blockade," in *Proc. Design Automation Conference*, 2013.

[16] 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.

[17] M. B. Yelten, S. Natarajan, B. Xue, and P. Goteti, "Scalable and efficient analog parametric fault identification," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2013, pp. 387 – 392.

[18] 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.

[19] M. Dubois, H.-G. Stratigopoulos, and S. Mir, "Hierarchical parametric test metrics estimation: A $\Sigma\Delta$ converter BIST case study," in *Proc. IEEE International Conference on Computer Design*, 2009, pp. 78–83.

[20] M. Dubois, H.-G. Stratigopoulos, S. Mir, and M. J. Barragan, "Evaluation of digital ternary stimuli for dynamic test of $\Sigma\Delta$ ADCs," in *Proc. IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2014.

[21] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1986.

[22] A. Singhee, J. Wang, B. H. Calhoun, and R. A. Rutenbar, "Recursive statistical blockade: An enhanced technique for rare event simulation with application to SRAM circuit design," in *Proc. IEEE International Conference on VLSI Design*, 2008, pp. 131–136.

[23] A. Singhee and R. A. Rutenbar, "Statistical blockade: Very fast statistical simulation and modeling of rare circuit events and its application to memory design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 8, pp. 1176–1189, 2009.

[24] L. Rolindez, S. Mir, J.-L. Carbonero, D. Goguet, and N. Chouba, "A stereo $\Sigma\Delta$ ADC architecture with embedded SNDR self-test," in *Proc. IEEE International Test Conference*, 2007, paper 32.1.

[25] M. Dubois, H.-G. Stratigopoulos, and S. Mir, "Ternary stimulus for fully digital dynamic testing of SC $\Sigma\Delta$ ADCs," in *IEEE International Mixed-Signals, Sensors, and Systems Test Workshop*, 2012.

[26] P. Malcovati, S. Brigati, F. Franscesconi, F. Maloberti, P. Cusinato, and A. Baschirotto, "Behavioral modeling of switched-capacitor sigma-delta modulators," *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 50, no. 3, pp. 351–364, 2003.

[27] A. Fornasari, P. Malcovati, and F. Maloberti, "Improved modeling of signa-delta modulator non-idealities in SIMULINK," in *IEEE International Symposium on Circuits and Systems*, vol. 6, 2005, pp. 5982–5985.