Xama, Nektar; Andraud, Martin; Gomez, Jhon; Esen, Baris; Dobbelaere, Wim; Vanhooren, Ronny; Coyette, Anthony; Gielen, Georges

Machine Learning-based Defect Coverage Boosting of Analog Circuits under Measurement Variations

Published in:
ACM Transactions on Design Automation of Electronic Systems

DOI:
10.1145/3408063

Published: 01/10/2020

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Xama, N., Andraud, M., Gomez, J., Esen, B., Dobbelaere, W., Vanhooren, R., Coyette, A., & Gielen, G. (2020). Machine Learning-based Defect Coverage Boosting of Analog Circuits under Measurement Variations. *ACM Transactions on Design Automation of Electronic Systems*, *25*(5), Article 47. https://doi.org/10.1145/3408063

Machine Learning-Based Defect Coverage Boosting of Analog Circuits under Measurement Variations

NEKTAR XAMA, KU Leuven, Department of Electrical Engineering ESAT-MICAS
MARTIN ANDRAUD, Aalto University, Department of Electronics and Nanoengineering
JHON GOMEZ, KU Leuven, Department of Electrical Engineering ESAT-MICAS
BARIS ESEN, KU Leuven, Department of Electrical Engineering ESAT-MICAS
WIM DOBBELAERE, ON Semiconductor
RONNY VANHOOREN, ON Semiconductor
ANTHONY COYETTE, ON Semiconductor
GEORGES GIELEN, KU Leuven, Department of Electrical Engineering ESAT-MICAS

Safety-critical and mission-critical systems, such as airplanes or (semi-)autonomous cars, are relying on an ever-increasing number of embedded integrated circuits. Consequently, there is a need for complete defect coverage during the testing of these circuits, in order to guarantee their functionality in the field. In this context, reducing the escape rate of defects during production testing is crucial, and significant progress has been made to this end. However, production testing using automatic test equipment is subject to various measurement parasitic variations, which may have a negative impact on the testing procedure and therefore limit the final defect coverage. To tackle this issue, this paper proposes an improved test flow targeting increased analog defect coverage, both at system- and block-level, by analyzing and improving the coverage of typical functional and structural tests under these measurement variations. To illustrate the flow, the technique of inserting a pseudo-random signal at available circuit nodes and applying machine learning techniques to its response is presented. A DC-DC converter, derived from an industrial product, is used as a case study to validate the flow. In short, results show that system-level tests for the converter suffer strongly from the measurement variations and are limited to just under 80% coverage, even when applying the proposed test flow. Block-level testing, on the other hand, can achieve only 70% fault coverage without improvements, but is able to consistently achieve 98% of fault coverage at a cost of at most 2% yield loss with the proposed machine-learning based boosting technique.

CCS Concepts: • **Computing methodologies → Neural networks**; • **Hardware → Design for testability**; *Integrated circuits*; *Methodologies for EDA*.

Additional Key Words and Phrases: Machine Learning for Test, AMS IC Test, Test under Measurements Variations

Authors' addresses: Nektar Xama, nektar.xama@esat.kuleuven.be, KU Leuven, Department of Electrical Engineering ESAT-MICAS, Kasteelpark Arenberg 10, Heverlee, Belgium, B-3001; Martin Andraud, Aalto University, Department of Electronics and Nanoengineering, School of electrical engineering, P.O. Box 15500, Aalto, Finland, FI-00076; Jhon Gomez, KU Leuven, Department of Electrical Engineering ESAT-MICAS; Baris Esen, KU Leuven, Department of Electrical Engineering ESAT-MICAS; Wim Dobbelaere, ON Semiconductor, Belgium; Ronny Vanhooren, ON Semiconductor, Belgium; Anthony Coyette, ON Semiconductor, Belgium; Georges Gielen, KU Leuven, Department of Electrical Engineering ESAT-MICAS, Kasteelpark Arenberg 10, Heverlee, Belgium, B-3001.

XXXX-XXXX/20xx/6-ART $15.00
https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

ACM Reference Format:
Nektar Xama, Martin Andraud, Jhon Gomez, Baris Esen, Wim Dobbelaere, Ronny Vanhooren, Anthony Coyette, and Georges Gielen. 20xx. Machine Learning-Based Defect Coverage Boosting of Analog Circuits under Measurement Variations. 1, 1 (June 20xx), 27 pages. https://doi.org/xx.xxxx/xxxxxxx.xxxxxxx

1 INTRODUCTION

The trend towards the increasing use of Integrated Circuits (ICs) in cyber-physical systems shows no signs of slowing down [16]. Following this trend, automotive applications stand at the top of the list of fast, continuously growing IC sectors as the industry is preparing for the arrival of autonomous vehicles. Indeed, modern vehicles rely on an increasing number of ICs, in particular analog and mixed-signal (AMS) circuits, to run safety-critical functions, such as collision detection or communication with other agents in the field.

Consequently, ensuring that all these ICs are fully functional in the field becomes an increasingly important constraint for IC manufacturers. Manufacturing flaws, however, can cause a multitude of defects in the production of an IC. Among them, catastrophic defects can result in unwanted open or short circuits within the IC. They pose a serious threat to the designed functionality because they modify the inner circuit topology and cause a wrong behavior. Detecting all these hard defects before sending the IC in the field is therefore crucial. Specifically considering analog circuits, detecting catastrophic defects is not an easy task, as the fabrication of ICs is subjected to a lot of variations. This has been an extensive research topic for more than a decade, e.g. [5, 6, 10, 11, 15, 21, 22, 28, 29, 32, 36]. First in line are process variations, which are usually taken into account in the design process by simulating (a) process corners for inter-wafer variations and (b) Monte-Carlo variations for intra-wafer variations. The objective of these simulations is to ensure that the designed circuit meets the specifications under these variations, as the design is validated across all process corners and random process variations. The specifications are subsequently assessed after chip fabrication during the testing of the IC, typically performed in industry with an Automated Test Equipment (ATE). In addition to the manufacturing variations, other variations will also directly affect the test results. Specifically, measurement variations of the ATE and the test setup, e.g. the variable contact resistance of the test probe pins, even when using some force control, can modify the test results and can have an impact on the overall fault coverage for a particular IC. In the current literature, the impact of these measurement variations is rarely analyzed in the context of defect detection and is therefore not taken into account in fault coverage metrics.

In this paper, we will develop a novel design test framework aimed at AMS ICs, that is able to take into account measurement variations during testing and compensate for them while maximizing the fault coverage, specifically by inserting additional measurements to the standard test flow and by using machine learning algorithms. Hence, we propose:

- The development of a new model for measurement variations in the context of multi-site ATE testing, which consists in adding test techniques step by step, evaluating them and comparing them in terms of the fault coverage. This flow is enriched by a model that enables simulating the measurement variations, integrating them in the test flow and evaluating their impact on the fault coverage. This test flow is extendable to compare alternative test methodologies and to boost the fault coverage when needed;
- A comprehensive evaluation of the impact of the measurement variations on system-level and block-level testing methodologies;
- A boosting technique, aimed at compensating for these variations and increasing the final fault coverage. This technique is based on (a) the inclusion of new tests, such as pseudo-random stimuli measurements in the example presented later on, in addition to block-level testing, and (b) the use of a machine learning algorithm.

These three contributions have also been integrated in a complete AMS test flow that is suitable for industrial environments. We will illustrate this test flow by comparing several test techniques, such as system-level test, block-level test and other measurements, on a representative case study consisting of an analog DC-DC converter derived from an industrial product. This comparison will allow to evaluate the impact of measurement variations on the fault coverage for the selected test techniques. These faults are the simulation models used for the defects introduced during IC fabrication. It will show that system-level testing, although intrinsically better that block-level testing in terms of fault coverage, is more sensitive to measurement variations. Using the proposed boosting technique, the fault coverage is increased from 70-80% to up to 98% on the considered converter case study under measurement variations, which is a significant improvement compared to standard test techniques typically used for this circuit. This technique also removes the need for system-level testing of the converter, as the faults are all covered by block-level testing and the functionality is measured as well.

The remainder of this paper is organized as follows. Section 2 will give the state of the art of standard industrial practices in analog circuit testing, explaining briefly the main test methodologies used in literature. Sections 3 and 4 will then go into the details of our proposed framework, able to model and integrate measurement variations in the test analysis and design process. Section 5 will detail the design of the case study used to illustrate the proposed methodology, which is a DC-DC converter. Section 6 will then present a comparative analysis between several test methodologies in terms of fault coverage, without and with measurement variations considered. Section 7 will detail the proposed enhanced test to increase the fault coverage of the case study circuit using pseudo-random test excitation in combination with machine learning-based test processing. Section 8 will conclude the paper.

2 OVERVIEW ON EXISTING FAULT DETECTION METHODOLOGIES FOR ANALOG CIRCUITS

Numerous test strategies have been developed and applied to analog circuits. This section will briefly review the main ones, before subsequently applying them on a case study circuit.

2.1 General fault detection principles

The objective of fault detection methods is to detect and screen out (ideally all) defective circuits during or right after the manufacturing process. The details of how faulty circuits are simulated will be given later in Section 3.2.2. Generally speaking, any IC must fulfill a given number of specifications, which are assessed during the testing phase, and fabricated circuits that do not satisfy these specifications are rejected. In practice, one of the main complications for the development of an effective analog fault detection method is that the manufacturing and test processes themselves are subjected to numerous variations. We will divide them into two distinct categories:

a) *process variations*, due to variations in the manufacturing phase, and
b) *measurement variations*, due to variations occurring in the testing phase.

2.1.1 Fault detection under process variations. Process variations are the result of variations in the manufacturing process. They can be modelled in the design stage, usually through Monte-Carlo and corner simulations, from which detection boundaries are deduced between faulty and non-faulty (i.e. good) dies [12]. Their distribution is usually Gaussian and the spread depends on the actual process used in production. These parameters are usually supplied together with the technology that is used at the design stage.

An illustrative example of how a fault detection method works under process variations is given in Fig. 1. Here, the Probability Distribution Function (PDF) of a given test under process variations

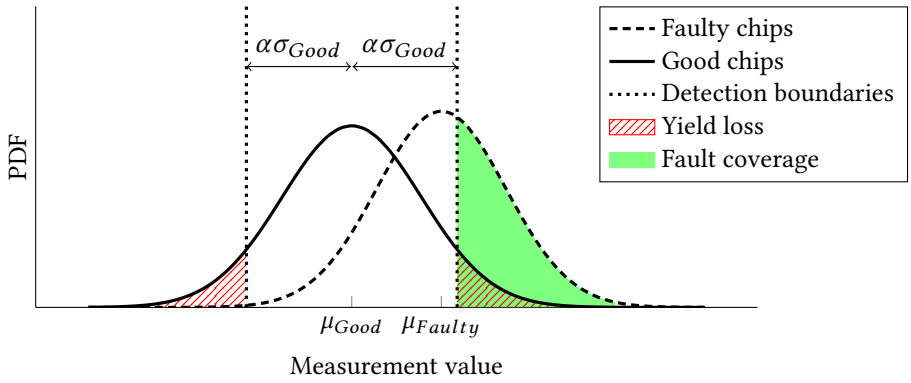

Fig. 1. Graph showing the yield loss on a distribution of good dies and the fault coverage on a simplified distribution of dies with a fault.

is represented using a Gaussian distribution. The PDF of "good" chips (i.e. devices passing the test) is represented with the thick line, and the PDF of faulty chips (i.e. devices failing the test) is represented by the thick dotted line. For each test, a detection boundary is defined, expressed in terms of the mean (μ_{Good}) and the standard deviation (σ_{Good}) of this Gaussian distribution. Using this, we can define the *yield loss*, shown in dashed red, as the percentage of good chips (chips without defects) that have some measurement result outside of the test limits. These limits are usually expressed as a number of standard deviations away from the mean of the measured values and are usually related to the product design specifications. Due to the variations in the production process, the circuit can be designated as faulty, even though it does not contain any defect. This means that either the boundaries are set too tight or that the process produces circuits that are out of the preset specifications. We consider the yield loss as the fraction of good circuits rejected because the test procedure considers them as out of the test bounds. The fraction of fabricated chips that remains within the test bounds is called the *yield*. Conversely, the *fault coverage* is defined as the fraction of faulty circuits that are out of the test bounds, i.e. faulty circuits tested and classified as faulty.

Specifically, a chip M_i is said to be faulty for a measurement i with mean $\mu_{Good,i}$ and standard deviation $\sigma_{Good,i}$, if:

$$\left| \frac{M_i - \mu_{Good,i}}{\sigma_{Good,i}} \right| > \alpha_i \tag{1}$$

where α_i is the detection boundary scaling factor for that measurement. The detection boundary positions are determined by the parameter α_i, which quantifies how many standard deviations away from the good mean the tested chip must be for a measurement i to be regarded as faulty. By choosing different values for the parameter α_i, the detection boundary between good and faulty can be moved. This results in a trade-off, since some circuits of the faulty population will have their distribution of measurements close to or even inside the good population, as shown by the example faulty population in Fig. 1. Depending on where the boundary is set, a different portion of the faulty distribution will be detected, shown in green, but this comes at the cost of *yield loss* (i.e. good circuits detected as faulty), shown in dashed red.

Although many techniques have been presented to deal with process variations (see Subsection 2.2), the impact of other types of variations, such as measurement variations, on the measurements during test have not been analyzed extensively in the literature.

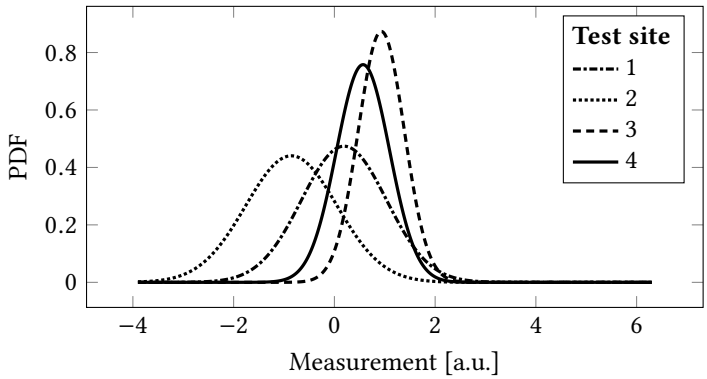

Fig. 2. Distribution for measured values for the same measurement over four parallel test sites on an industrial product of a single wafer.

2.1.2 Fault detection under ATE measurement variations. Measurement variations refer to changes that may occur during the testing phase. These variations are for instance the variable contact resistance when probing the IC with tester probes. One of the objectives of this paper is to integrate such measurement variations in the test flow, to evaluate their impact on the overall test results, and to compensate for them. To illustrate the actual impact of measurement variations, the experimental PDFs of four parallel test sites for the same measurement of an industrial-scale product are shown in Fig. 2. An industrial ATE is capable of testing multiple chips on a silicon wafer at the same time, using multiple test sites next to each other. Each site contains its own set of measurement probes, which are routed on a large Printed Circuit Board (PCB) to the measurement circuitry of the ATE itself. It is clear from Fig. 2 that each test site has its own bias due to the different measurement parasitics, which will be modelled and explained in more detail in Section 3.1.

To mitigate the effect of such measurement variations, their impact needs to be modelled, evaluated and compensated. In the following section, we will give an overview of the different concepts and techniques that will be used to achieve this.

2.2 Overview of the standard fault detection techniques

When testing AMS ICs, most of the fault detection strategies in industry consider in priority (or exclusively) process-induced variations. In this regard, we can broadly distinguish two types of testing methodologies, as illustrated in Fig. 3: system-level and block-level testing.

2.2.1 System-level testing: the classical approach for IC testing in the industry is to perform system-level testing. The general principle is to apply to the Circuit Under Test (CUT) a series of full-system tests, using the available pins of the circuit, for example primary inputs, outputs, supplies, etc. These tests allow for a straightforward assessment of the different circuit specifications, as they resemble typical circuit usage conditions. In addition, system-level testing does not require any additional on-chip structures to be included, avoiding any circuit area overhead. However, it may be difficult to detect all possible defects of the circuit because the controllability and observability are limited to what can be accessed from the outside pins. It is typically not possible to test every part of the IC separately, which in the end may limit the fault coverage in the inaccessible parts [22], and lead to potential undetected faults.

2.2.2 Block-level testing: to access more parts of the IC, block-level testing is used, with the objective to individually test all blocks within the IC separately and to maximize the fault coverage.

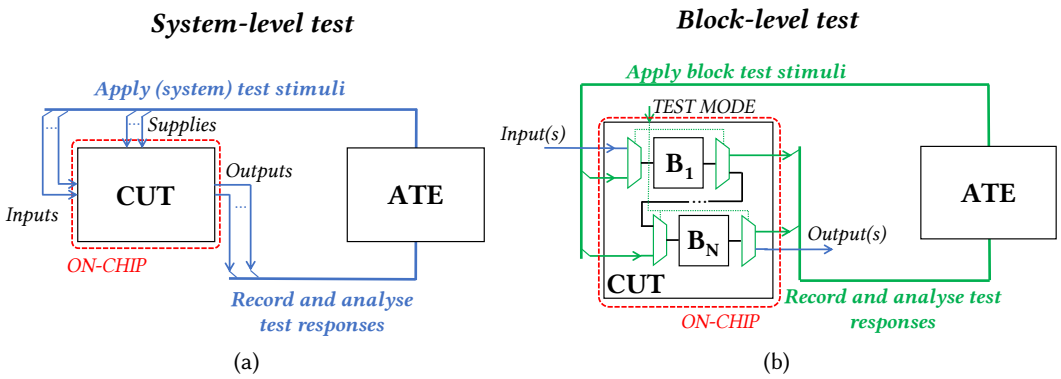

Fig. 3. Illustration of (a) system-level test and (b) block-level test for AMS ICs.

This testing approach requires additional Design-for-Test (DfT) structures, added to the circuit to allow the separation and individual measurement of each circuit block. The DfT process is usually carried out along with the IC design, in collaboration between test engineers and designers, to enable maximum controllability and observability in all circuit blocks, maximizing the fault coverage and minimizing the impact on the IC performances. The added DfT structures may introduce overhead (e.g. in chip area, power, etc.) that must be taken into account and minimized. Another possible drawback of solely using block-level testing is that system characteristics must be deduced from block characteristics. It is up to the circuit designers and the test engineers to map and combine them during the test process. Recently, automatic frameworks such as [8] have been developed to select DfT structures and test stimuli for AMS circuits to maximize the fault coverage at minimum overhead.

The use of these test methodologies does not always lead to an optimal fault coverage. Specifically, some faults may stay hidden, even with the use of multiple DfT blocks. In this context, this paper presents solutions to further increase the fault coverage. This research is divided into two different contributions:

1) implementation of additional measurements targeted to detect the remaining faults, and
2) development of statistical or machine learning based methods to enhance fault detection.

These two contributions will be detailed in the next subsections.

2.3 Additional tests for better fault coverage

2.3.1 Use of additional measurements: a first approach to enhance the test process is the use of extra additional measurements. For instance, measuring power supply currents to spot additional faults has proven its efficiency, first for digital ICs, where it is a standard test named IDDQ [25], and then adapted for analog circuits [28]. It can be used also to evaluate other metrics, such as the neighborhood current ratio (NCR), which evaluates the ratio of the IDDQ of the current chip to the mean IDDQ values of the neighboring dies, and can identify potential outlier circuits [5].

2.3.2 Use of additional test stimuli and responses: improving the fault coverage can also be performed through additional test stimuli and test responses. In this context, Built-In Self-Test (BIST), initially targeted to reduce the test cost by integrating the test generation and response analysis on-chip, leads to the implementation of several new measurement techniques. A large variety of BIST techniques have been proposed for specific analog circuits, such as phase-locked

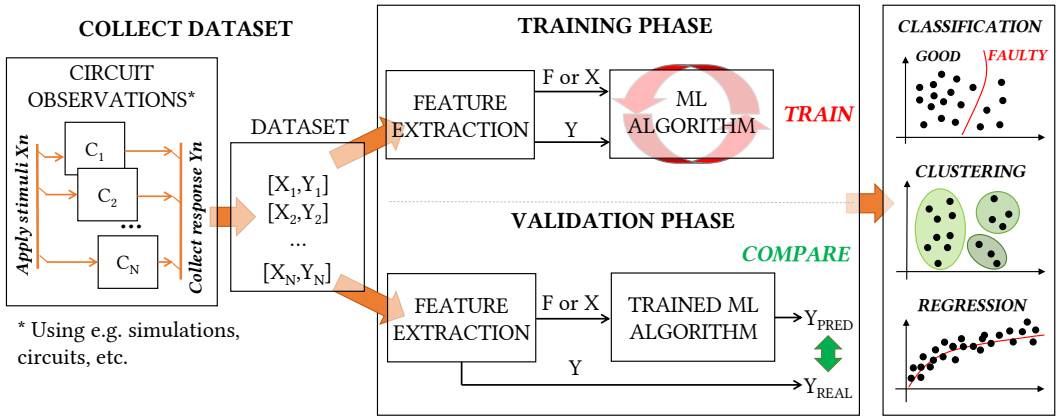

Fig. 4. General illustration of a typical machine learning based testing strategy.

loops [17], data converters [13], DC-DC converters [4], or more general circuits [10]. The CUT can, for instance, be analyzed in the frequency domain, e.g. by applying random test inputs and measuring the frequency response of the circuit [29]. The CUT can also be set in forced oscillation mode, such that its response can be analyzed to detect defects [10].

2.4 Machine learning techniques for fault detection

In order to further increase the fault coverage, advanced statistical and machine learning algorithms have been proposed. These algorithms are indeed able to detect complex correlations between variables, by, for instance, mapping measurements to circuit performances and identifying previously hidden faults when processing the results from (extra) measurements, as presented in this paper.

There exist numerous works proving the efficiency of machine learning and deep learning algorithms to different test applications, such as fault detection [6, 11, 21, 28, 29, 31, 32, 32, 36, 37], fault diagnosis and yield learning [15], test escape detection [19, 27], or alternate testing and calibration [1–3, 20, 24]. The term *alternate test* refers to a cost-effective methodology where the complete test process is replaced by low-cost measurements, correlated to the actual performances of the circuit with machine learning algorithms [35]. This algorithm can then compensate for process variations if performance tuning knobs are included in the learning process. A general review of machine learning applications for IC testing, including fault detection, is available in [30].

Focusing on analog fault detection, methods presented in the literature typically consist in using machine learning algorithms, such as neural networks or support vector machines, to highlight and classify faults in complement to the current test process or coupled with additional measurements applied to the CUT. This is illustrated in Fig. 4. First, a *dataset* is constituted using a set of N observations. For instance, a dataset can be a fault model dictionary, created with Monte-Carlo simulations of a circuit under process variations, partially with specific defects introduced in the circuit. It can also be enriched with test data from fabricated circuits, such as pre-production wafers. This entire dataset is subsequently used to train and validate a machine learning model, that can perform various tasks. The most typical tasks consist in classification (e.g. giving pass/fail labels for ICs in a fault detection case), clustering or multi-class classification (e.g. build distinct classes depending on the type of faults in a fault diagnosis case), or regression (e.g. explicitly learn the relationship between given measurements and circuit performances in an alternate test case). In this paper, we will focus on machine learning applications for fault detection, i.e. pass/fail detection.

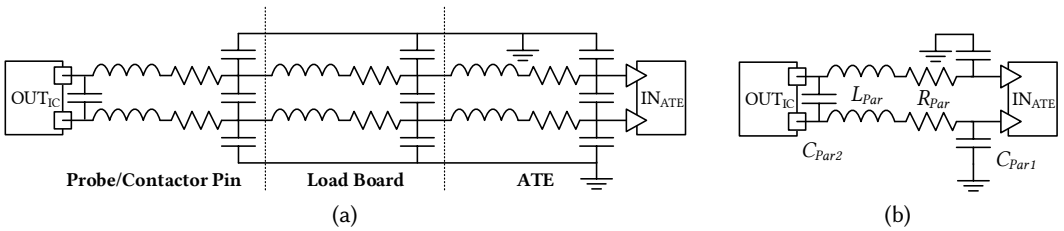

Fig. 5. Detailed (a) and simplified (b) models for the measurement parasitics on ATE probes, as present during production IC testing.

Among the first examples in literature, in [28], neural network algorithms have been combined with supply current measurements to detect faults in an amplifier. The fault detection algorithm is first trained based on a fault dictionary, observing the simulated reaction of the circuit to a supply current ramp, and identifying clusters of faults with a multi-class classifier. In the fault detection phase, a ramp is applied to the CUT and the trained classifier is used to detect faults. In [29], the analysis of the random test responses is performed using artificial neural networks to classify between good and faulty circuits, based on a dataset with faulty and fault-free devices. More recent developments in the machine learning area allow to detect more advanced non-linear boundaries [32], enable to spot and filter out more efficiently outliers in a distribution of ICs using alternate test [31], or are integrated in a more classical test flow [36]. The term *outlier* is used here to define circuits exhibiting a behavior quite different from regular devices, thus having a high probability to contain a fault. Standard testing and alternate testing can also be combined, as in [11], where the machine learning model used to classify between good, faulty, and marginal dies can be updated over time to take into account new information in the data.

3 MODELING, EVALUATION AND COMPENSATION OF MEASUREMENT VARIATIONS

The description of the impact of measurement variations on AMS testing in the previous section leads to three targets:

(1) model the effect of the measurement variations;
(2) integrate this model in a test flow that evaluates its impact;
(3) automate the test process to take these variations into account with compensation and enhancements.

In this section, these three points will be addressed. In Section 4, the resulting test flow that achieves these targets will be presented.

3.1 Measurement variations during production test

Due to the nature of production testing, measurement variations are unavoidable during production tests. ATEs always have some parasitic elements that influence the measurements. The effect of this on an actual industrial product has been shown in Fig. 2. Three main sources of parasitic elements are causing this effect: the probe needle or contactor pin, the load board, and the ATE traces. With multi-site testing, each test site has its own set of probe needles and is routed slightly differently on the PCB to the ATE, causing measurement variations for the same measurement. Each of these three elements can approximately be modeled by an inductor-resistor-capacitor chain of lumped elements, as shown in Fig. 5a. Moreover, there is also coupling possible between two neighboring

parasitic chains, as depicted by capacitors in Fig. 5a. For the analysis in this paper, the measurement parasitic model has been simplified to the equivalent circuit in Fig. 5b, by contracting the three chains to one single chain. The exact values for R_{Par}, L_{Par} and C_{Par1} are unknown in the analysis and are therefore randomly sampled from a $[1-10]$ uniform distribution, with unit values Ω, nH and pF, respectively, whereas C_{Par2} has a range of $[5-10]$ pF. These measurement parasitics are added to each probe pad in the analysis. The values for the resistance are based on [18], which reports contact resistances within the range mentioned. The inductor and capacitor values are based on values for PCB parasitics [34]. The parasitic elements are all assumed to be uncorrelated.

The simplification is equivalent to an RLC filter with added capacitive coupling between the nodes. As the PDFs of all parasitic components of Fig. 5b are flat, the resulting effect on measurement variations can vary, as is clear from Fig. 2.

3.2 Integration in the test flow

We will use several aspects of AMS testing to evaluate the impact of the measurement variations, while at the same time accounting for the process variations. Different types of tests, measurements and test setup schemes are explored that will aid in assessing the effects of the measurement variations.

3.2.1 Specification and additional tests. Both system- and block-level test methodologies contain specification tests, i.e. tests that measure some functional specification. In the system-level test method, the function is typically related to the system, e.g. the conversion speed of a voltage converter. This is also true for block-level tests, only the functions here are related to the different blocks of the system. These can typically be amplifiers, oscillators, etc., which can have tests that measure for instance the bandwidth or oscillation frequency. Specification tests are typically not designed specifically with fault coverage in mind; yet, they are consistently used in industrial test settings. Their fault coverage performance for the considered design case will be analyzed in Sections 6 and 7. Additional measurements are used to quantify structures that are usually present in most AMS circuits as introduced in Section 2.3.1. These can measure currents and voltages that should always be present and can be done regardless of the test methodology applied. The DC current consumption is used as an additional analog measurement in our case. This extra measurement does not introduce a significant increase in test time, because it can happen in parallel to the specification tests. Besides having a very low test cost, this extra measurement has the additional benefit that it can be simulated efficiently by DC simulations. These are orders of magnitudes faster than the transient simulations required for the other typical tests.

3.2.2 Fault coverage simulations. We adopt a defect-based test methodology for analog circuits. The assessment of the test methods is quantified by their analog defect coverage for catastrophic defects. These are defects that cause short or open circuits between transistor terminals. In the simulations, these defects are modelled by faults inside CMOS transistors as described by the 5-fault model [26]. This model contains the following faults:

- open circuit at source side;
- open circuit at drain side;
- gate-source short circuit;
- gate-drain short circuit;
- drain-source short circuit.

For test analysis, open circuits are modelled as a high-resistive connection (1 GΩ), whereas short circuits are modelled as low-resistive connections (100 Ω). The fault coverage (*FC*) is expressed as the fraction of all the possible faults (N_{faults}) that the test program can detect ($Ft_{detect,i} = 1$ if

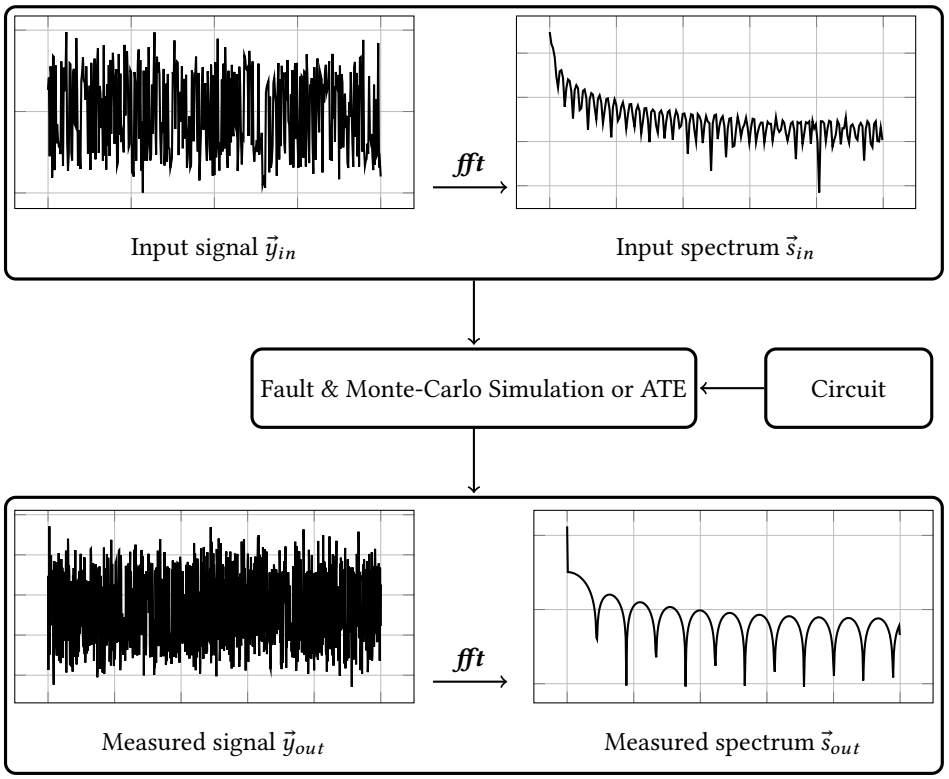

Fig. 6. Block diagram showing the test data generation.

fault i is detected):

$$FC = \frac{\sum_{i=1}^{N_{faults}} Ft_{detect,i}}{N_{faults}} \tag{2}$$

As the number of transistors in a circuit is determined (including possible DfT circuitry), so is the number of defects. We will further assume that each fault has equal probability of occurrence (although this could be included if the proper process information would be available). Thus, the calculation of the fault coverage reduces to Equation 2. Since this work is based on simulations, we will use the fault coverage as target criterion for the remainder of this paper.

3.2.3 Test boundary selection. The fault coverage will be calculated in two situations:

1) the coverage when all threshold boundary scaling parameters $\alpha_i = 6$ with respect to the mean of the population without measurement variations; and
2) the maximum achievable coverage, by selecting α_i such that the coverage is maximized with respect to the mean of the entire population.

For this first approach using $\alpha_i = 6$ is typical in industry, while the last approach is rather used as a metric of optimality, as the selection of these parameters is highly dependent on the industrial test situation. It also corresponds to Dynamic Part Average Testing (DPAT) [12]. Its original purpose is to account for wafer to wafer process variations, which also causes a shift in the measurements. DPAT uses only the samples of the considered wafer to determine tighter and better-centered test

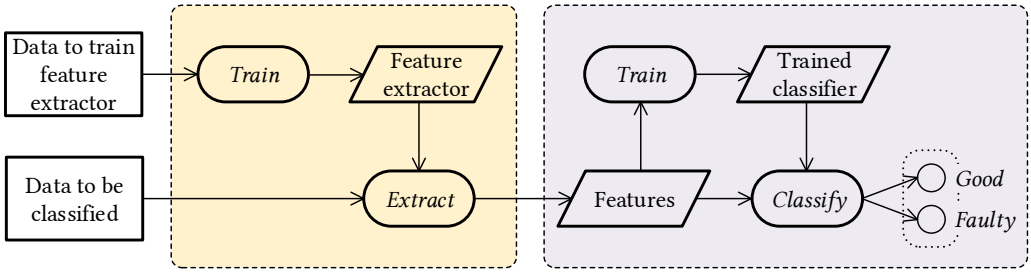

Fig. 7. Illustration of the classification process

boundaries following equation (1). In the frame of this paper, it is assumed that the simulated dies are on a single wafer and thus the test boundaries depend on the entire simulated population.

Using DPAT has two additional benefits. Firstly, the tests become invariant to process corners: without DPAT simulations would also be required across process corners, heavily increasing the required simulation time. Secondly, the technique only needs data computation, which means that it adds no cost in terms of test time or silicon area.

3.3 Introduce compensation and enhancement techniques in the test flow

As additional test measurement, we propose to inject a pseudo-random noise signal, measure the response to this pseudo-random signal, and classify between good and faulty circuits based on these measurements. This signal activates the circuit over a wide spectral band and allows to find faulty circuits based on their response to this signal. Our implementation is based on ATE capabilities rather than previously proposed BIST implementations [29]. The general flow is shown in Fig. 6. A pseudo-random input signal $\vec{y}_{in}$ is applied to one or more nodes of the circuit and the response $\vec{y}_{out}$ of the circuit to this signal is simulated (in this paper) or measured by the ATE (in industry). It is also the ATE that, in practice, will determine the bandwidth and the power of the pseudo-random signal.

From the spectrum $\vec{s}_{out}$ of the measured output, a set of features is extracted. This extraction is based on [9], where low-dimensional features are extracted from high-dimensional time signals. In our work, we use both proposed techniques, where the extracted features correspond to the Principal Components of the signals, and an autoencoder to extract features. In this paper, only the good (fault-free) population of circuits are used to train the feature extractor. These features are then used to train a binary classifier, such that classification is possible between good and faulty circuits. The entire feature extraction and classification process is illustrated in Fig. 7. The input layer of the neural net consists of the features, whereas the output is only the two classes: *Good* and *Faulty*. As an extension, a Support Vector Machine (SVM) binary classifier will also be used. This classification step is implemented in the Machine Learning Toolbox of MATLAB.

4 PROPOSED TEST FLOW

The proposed flow for analog fault coverage improvement under measurement variations is illustrated in Fig. 8. This flow is specifically targeted to deal with the issues caused by measurement variations and therefore to increase the fault coverage in the context of an industrial ATE test environment. The general principle is to analyze, compare and combine several test strategies (additional measurements or stimuli, etc.), when necessary, in order to compensate simultaneously for process and measurement variations. The overall flow is built in such a way that it can be

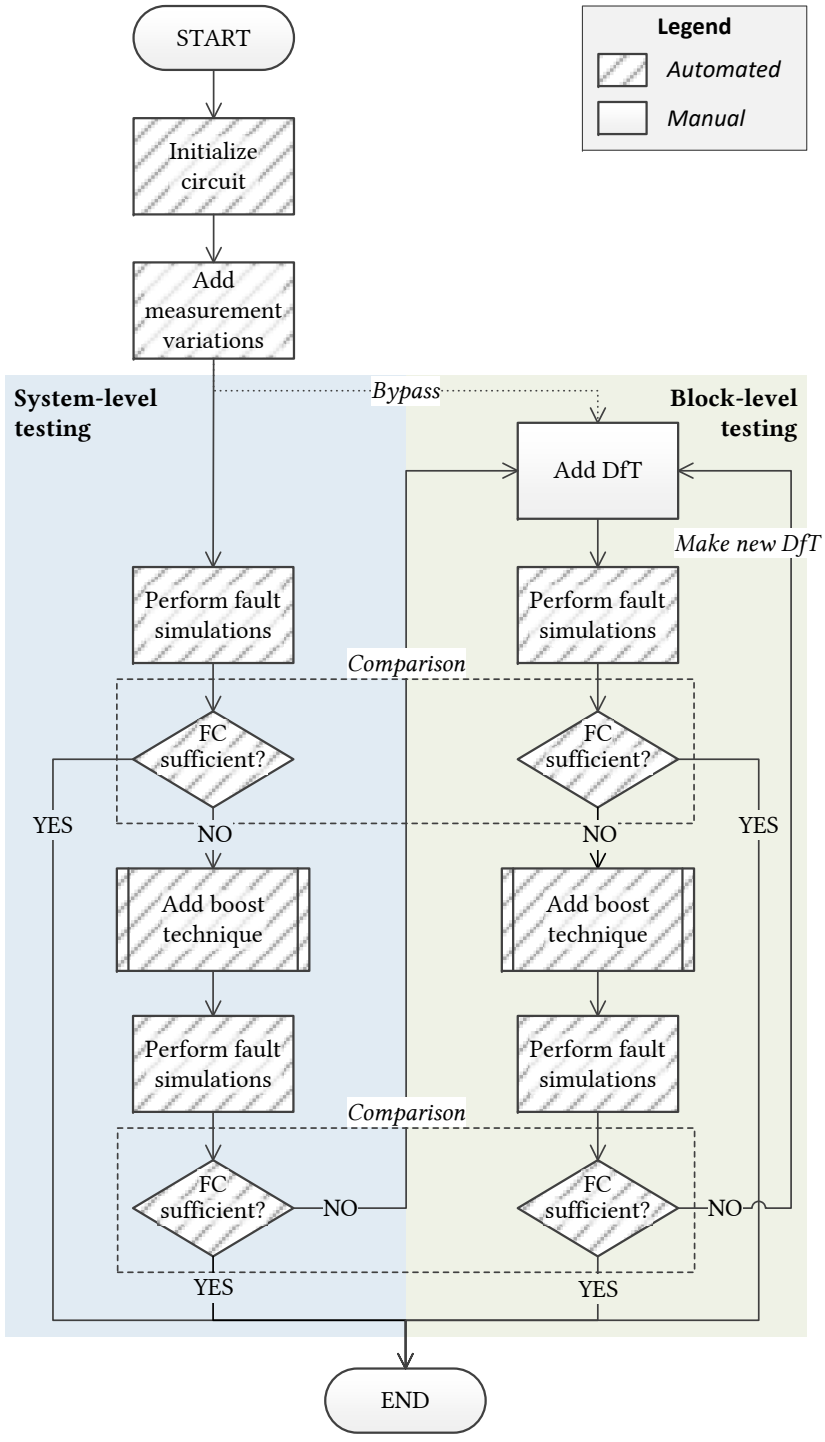

Fig. 8. Proposed AMS test fault coverage (FC) improvement flow.

integrated into the design flow of an IC. When designing ICs with attention to testing them, this flow can be used to determine optimal testing schemes and the potential need for DfT circuitry during the design stage. Integrating this into the design flow also has the benefit of allowing to design the DfT circuitry in such a way as to have minimal impact on the circuit in normal use.

The proposed test flow is split in two main branches: system-level (left) and block-level (right) testing methodologies. Both branches can be traversed sequentially or simultaneously. This can either follow the design procedure of an IC, where first a system is created, then the fault coverage is assessed, and the choice remains to invest in DfT and follow the second branch. The first branch can also be entirely bypassed, when DfT is necessary (while debugging, for example). It is also possible to obtain a comparison between both methodologies by following them simultaneously. The latter is the approach taken in this paper. All steps of this methodology, except for the generation of DfT blocks, can be automated, for instance to compare the fault coverage of two testing techniques, or to stop the process when a given fault coverage is reached.

In details, the proposed test flow is composed of the following major steps:

- **Initialize the circuit:**
  This step consists in e.g. preparing simulations with the required test stimuli and storing future test results.
- **Add measurement variations in the circuit:**
  These variations can be added at simulation level by introducing parasitic components to the circuit, as explained in more details in section 3.1.
- **First branch: system-level testing (left side of Fig. 8):**
  - **Perform the fault simulation with system-level tests.**
    The first tests that are performed by the methodology are system-level tests, because they are straightforward to implement and do not need any DfT structures. The proposed fault coverage of system-level test uses functional tests (e.g. specification-based tests, depending on the circuit type and topology), as well as structural tests (e.g. DC supply current measurements) as they do not require any change in the circuit topology.
  - **Add fault coverage boosting technique for system-level testing (Fig. 9).**
    If the coverage is insufficient, the next step calculates possible improvements by adding our proposed boosting technique. This technique is also configurable and can be performed in an automated way. The basic principle is to add tests (e.g. alternate measurements) and measurements in combination with machine learning techniques to extend the fault coverage. Several machine learning algorithms (referred to as ML in Fig. 9) can be compared to select the one that leads to the best fault coverage. In the proposed methodology, we use a pseudo-random noise signal as an additional test technique, and compare two feature selection methods (encoder with neural network and principal component analysis), with two types of classifiers (neural networks and support vector machines) to choose the optimal strategy.
- **Second branch: Block-level testing (right side of Fig. 8):**
  - **Add DfT structures for block-level testing.**
    If the coverage is insufficient with system-level test, or if the objective is to compare system-level and block-level test under measurement variations, this step intends to design and incorporate DfT structures into the circuit for block-level testing. In this work, this is the only step that is not automated in the test flow, as DfT structures heavily depend on the type of circuit and the topology.

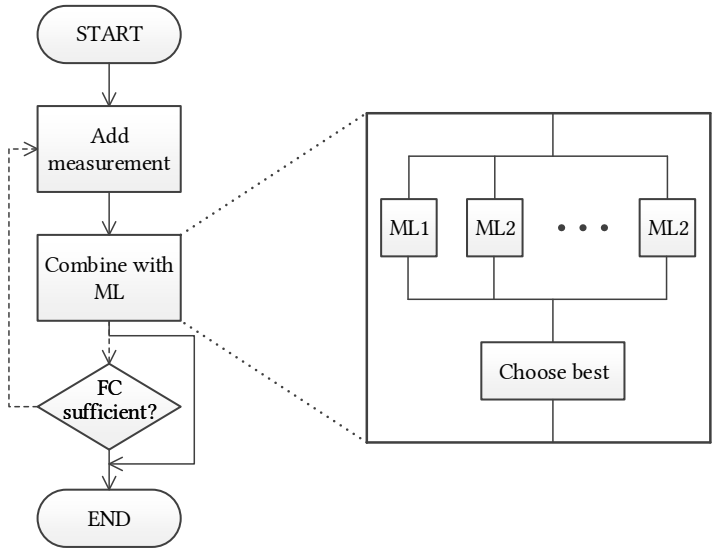

Fig. 9. Detail on the proposed boosting techniques.

- **Perform the fault simulation with block-level testing.**
  Generate the block-level fault coverage using functional and structural tests. This fault simulation is linked to the chosen DfT strategy which depends on the considered circuit.
- **Add fault coverage boosting technique for block-level testing.**
  If the coverage is insufficient, the next step calculates possible improvements by adding our proposed boosting technique. This is the same technique as for the system-level phase. If the coverage is not high enough, the same branch can be followed using additional DfT circuitry.

Furthermore, by performing concurrently the system-level and block-level testing schemes, the proposed flow enables a direct comparison between the two testing methodologies. Another interesting feature of this approach is that the influence of measurement variations for the two branches of the flow can also directly be compared to evaluate which test suffers the most from these variations. In the following subsection the DfT structures will be detailed in the context of our DC-DC converter case study.

5 CASE STUDY: DC-DC CONVERTER

A number of analog benchmark circuits are available in the literature for testing purposes, e.g. in [33]. However, they typically lack proper definition of the process variations of the technology. Thus, in order to illustrate the proposed methodology, a custom DC-DC converter has been designed and serves as a case study throughout this paper. The circuit is industrially relevant. Indeed, cyber-physical systems need multiple clean supply voltages to guarantee the designed functionality. For instance, a standard vehicle power supply today usually consists of batteries and alternators that provide a voltage in the range from 12 V to over 14 V. Robust DC-DC converters are thus necessary to provide all other ICs with a supply voltage that is as stable and clean as possible. It is therefore important that the DC-DC converters are thoroughly tested for defects. DC-DC converters are rarely the case study for developing test strategies. Whereas, in [4] a BIST methodology has been

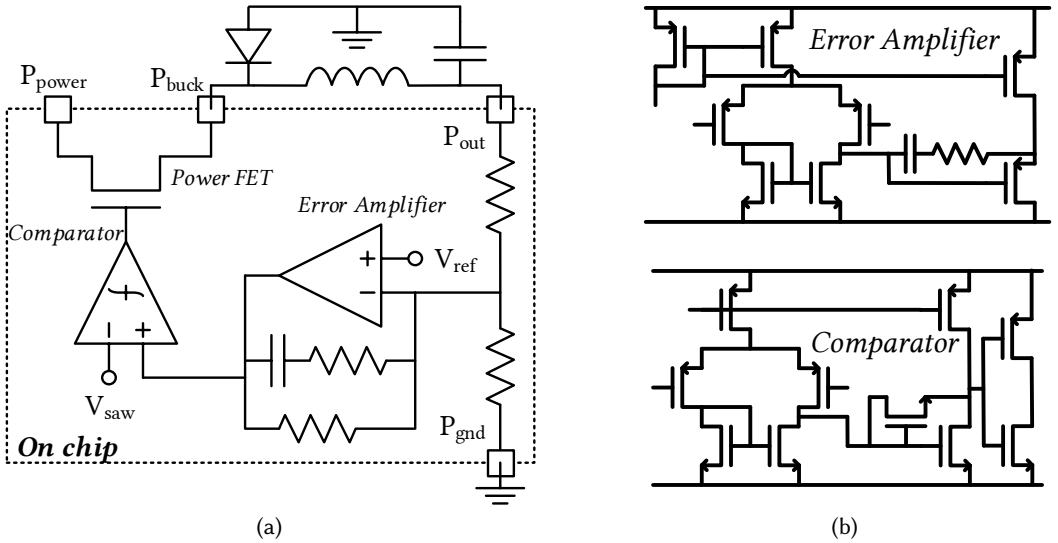

Fig. 10. Principal schematic diagram of the DC-DC buck converter (totaling 25 transistors) and external components (a), and the simplified schematics of the two main blocks containing most of these transistors (b).

developed, targeting the detection of parametric defects, while the test effectiveness of detecting catastrophic defects in DC-DC converters has hardly been analyzed.

5.1 Design of the converter

The design is derived from a full industrial product embedded in automotive systems, with the aim of having a working circuit to which different testing techniques can be applied. ON Semiconductor's 0.35 µm I3T50 technology is used in the implementation, because the products on which this circuit is based on, are part of the company's automotive IC portfolio. The target design of the circuit is as close as possible to the original design of ON Semiconductor's products.

The switching buck converter is chosen as the topology, which is based on [23]. This topology is also common in several products. The complete schematic is shown in Fig. 10. It uses feedback to control the duty cycle of a pulse-width modulated signal that operates at a frequency of 2 MHz. This, in turn, operates a large power transistor (FET). The feedback consists of measuring the difference between the converter output and a reference voltage level (V_{ref}). The circuit has been designed to convert 3.3 V down to 2.4 V. The reference (V_{ref}) and sawtooth (V_{saw}) voltage inputs of the error amplifier and comparator are taken to be ideal sources. The blocks in the circuit of Fig. 10 feature 25 transistors, as shown in Fig. 10b.

5.2 System-level test methodology

System-based functional tests are executed when the entire converter is on and converting. This means that there are additional off-chip components needed to emulate the nominal use case. These components (a diode, an inductor and a capacitor) are typical for a buck converter and are outside of the chip, as shown in Fig. 10. Different conditions or condition changes can be introduced during the tests. In our case, changing the supply and the load are used as tests, where each is switched

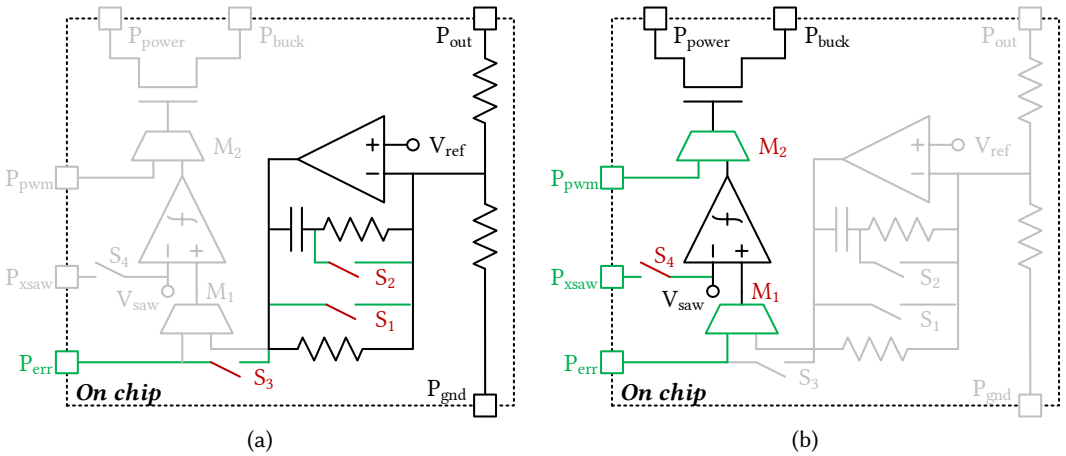

Fig. 11. Schematic diagrams of the DC-DC buck converter in two block test modes: (a) amplifier, and (b) comparator and power FET.

from its respective minimum (3 V and 1 kΩ) to its maximum (3.6 V and 1 MΩ) value and vice versa. In the first situation, a change to the supply is induced after the circuit has settled. The same is also done in the second situation by changing the load. In each situation, the supply or load are changed. For each test three measurements are simulated: settled voltage before the change, the overshoot or undershoot (maximum of minimum voltage obtained), and the settled voltage after the change. This results in a total of six measurements per test. The test boundaries that determine whether a sample is defective or not will be given in Section 6.

5.3 Design-for-test structures and block-level test methodology

While other works focus on adding BIST [4] to test DC-DC converters, the flow presented in this work proposes to design and insert a number of Design-for-Test (DfT) structures into the circuit, such that more internal blocks are accessible. These DfT structures come in the form of multiplexers (M_i) and switches (S_i), and are shown in Fig. 11. Their purpose is to allow the ATE equipment to reach signals inside the converter. These signals, which otherwise are not accessible, are routed by the switches and multiplexers to specific test pads. The switches can also put the converter in selected test states for specifically designed tests. 28 DfT transistors are added in total.

For the block-level functional tests, the DC-DC converter is divided into three distinct blocks, as indicated in Fig. 10. These blocks are the error amplifier, the comparator, and the power FET. The error amplifier is tested in separate test conditions, whereas the test conditions for the comparator and the power FET are combined in joint test states. The active parts for each test mode are

Table 1. Number of simulated samples.

Type (total amount)	Intra-die variations	Measurement variations	Total # of simulations
Good (1)	98	7	686
Faulty (265)	1	7	1855

highlighted in Fig. 11. All block-level tests follow a general pattern: an input voltage is applied to the block and the output of the block is measured. Different from system-level testing, this testing requires extra DfT structures. The applied inputs can also change after some time, such that different conditions can be tested for that block. The output is then measured by the ATE through the DfT structures. The measurements in the block-level tests are either settled voltages, settling times for the error amplifier tests, or voltage levels that cause a sufficiently high current to flow through the power FET for the other tests.

5.4 Fault simulations under measurement variations for the designed DC-DC converter

To model the measurement variations, each fault is simulated for 7 different, randomly sampled sets of measurement parasitic values, determined by the parasitic model values described in Section 3.1. The parasitics are added to each probe pad P_i: for the system-level approach there are only four pads (shown in Fig. 10), while for the block-level approach, three additional probe pads are introduced in addition to the four existing pads.

The total number of fault simulations is given in Table 1. The total number of faulty circuits also includes the faults in the DfT structures. Note that these are omitted in the analysis of the system-level tests, as these transistors are never used in this case.

6 BASELINE TEST OF THE DC-DC CONVERTER

This section will present the baseline comparison between system-level testing in Section 6.1 and block-level testing in Section 6.2, while the effects of the additional measurements and the machine learning techniques will be introduced in Section 7. First, only the system- and block-level specification tests will be used; then DC supply current measurements will be added to complete the baseline. The fault coverage results are summarized in Table 2.

6.1 Fault coverage analysis with system-level testing

The coverage is determined using the test detection boundaries explained in Section 3.2.3: first $\alpha = 6$ is used for each of the measurements. This results in a fault coverage of 73.6% without measurement variations. When measurement variations are included, the coverage increases to 76.1%, while at the same time causing a yield loss of 4.5%. The effect of the measurement variations is that some measurement shifts and spreads more, such that extra yield loss is incurred for fixed detection boundaries, as well as an increase in coverage, because some faulty circuits shift outside the test boundaries at the same time. Secondly, the maximum achievable fault coverage is calculated. This maximum is 76% and is reduced by 0.5% when taking measurement parasitics into account, due to the same spreading effect.

6.2 Effects of measurement variations on block-level testing

Block-level testing shows very little impact from measurement variations. The effect of the measurement variations on the coverage is very limited and shows an increase in coverage by only 0.6%, because there is a small shift of the distribution of some measurement here as well. This effect is more pronounced in defective circuits, which results in very high coverage levels of the functional blocks. For $\alpha = 6$ this is 89.6%, while it is possible to get to 92.8% as upper limit. However, a major drawback becomes clear when the coverage of the added DfT structures is included in the coverage calculation. When these are included, the coverage performance drops to 62.9% with $\alpha = 6$ and 65.2% as maximum achievable coverage.

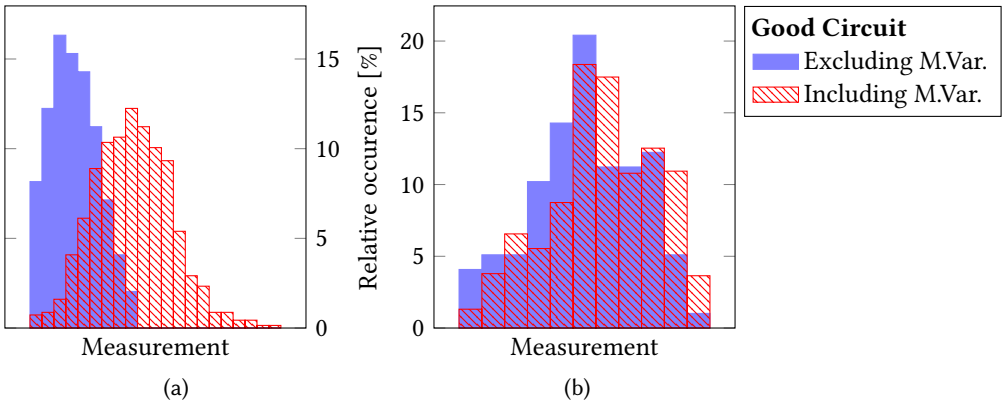

Fig. 12. Histograms of (a) a system-level and (b) a block-level measurement on the fault-free converter without and with measurement parasitic variations.

6.3 System-level test versus block-level test

It is clear that system-level testing does experience some negative effects from the measurement variations, whereas block-level testing shows very limited change. The coverage of the extra DfT circuitry used in block-level testing, however, highly reduces the coverage levels below those of system-level testing.

Fig. 12 shows the effect of the measurement variations in the form of histograms. These show a typical measurement for each test methodology, excluding and including measurement parasitic variations. Fig. 12a shows the impact of the measurement variations on the system-level test. The mean value is shifted and the standard deviation is increased, which explains the high yield loss in Table 2 and the reduction in maximum achievable coverage. The block-level measurements, in comparison, only experience a small shift in the histogram and a negligible increase in standard deviation, as shown in Fig. 12b. This very limited shift can be explained by the absence of a feedback loop that regulates the system. Measurements in system-level test are always taken when it is regulating, thus the effect is present in each measurement. Block-level measurements do not contain such complex feedback loops and are thus much more resilient to measurement variations. The

Table 2. System- and block-level functional test fault coverage (**FC**) and yield loss (**YL**) levels with fixed parameters $\alpha = 6$ and the maximum coverage without yield loss.

Test method	**Measurement variations**	**Test bounds fixed $\alpha = 6$**		**Max coverage without YL**	
		FC [%]	**YL [%]**	**FC [%]**	**YL [%]**
System-level	No	73.6	0	76.0	0
	Yes	76.1	4.47	75.5	0
Block-level (excluding DfT)	No	89.6	0	92.8	0
	Yes	89.6	0	92.8	0
Block-level (including DfT)	No	62.3	0	63.8	0
	Yes	62.9	0	65.2	0

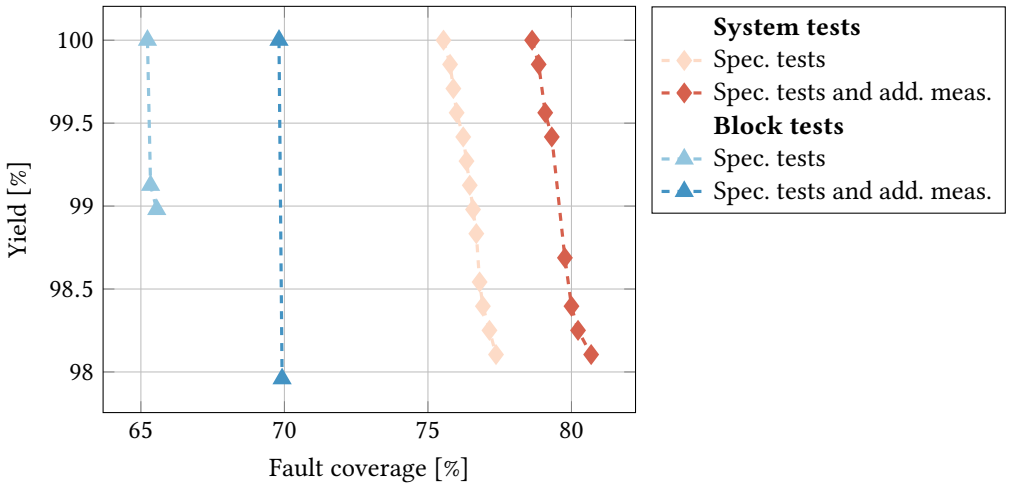

Fig. 13. Trade-off between the maximum achievable fault coverage and the yield for the specification tests and the baseline which consists of both specification (spec.) tests and the additional measurements (add. meas.) for system-level testing and block-level testing with DfT circuitry.

limited shift of the block-level measurements also explains the limited effect of the measurement variations on the block-level test coverage values shown in Table 2.

6.4 Adding the DC current supply measurement to form the baseline

In order to complete the baseline fault coverage performance in both cases, the DC current measurement at the supply is added as extra measurement. This results in an increase of the fault coverage to 78.4%, from previously 75.5%, for system-level testing, while block-level testing experiences a much larger benefit, as the coverage increases to almost 70%.

6.5 Fault coverage results

As a target, the maximum achievable fault coverage will be used and extended to allow only a very limited yield loss to be traded for a possible improvement in fault coverage. In typical industrial test situations, allowing a limited yield loss can prevent defective circuits from being shipped. Furthermore, such a trade-off is the optimal way of comparing the fault coverage performance using machine learning to the previous techniques, because there is no α_i in the machine learning approach. The trade-off between fault coverage and yield is shown in Fig. 13 for the case of only the functional tests and including DC measurements. Trading yield for fault coverage only allows an increase of around 1% extra fault coverage for 1% yield loss with system-level testing. Block-level testing has almost no benefit in fault coverage when allowing yield loss. The benefits of trading in yield are not substantial enough to reach the state of the art in fault coverage. Therefore, other techniques are necessary. However, the fault coverage levels of system-level tests are well below those of recently presented methods [7, 14], which achieve over 90%. To achieve these levels, we apply our fault coverage boosting technique in the next section.

7 FAULT COVERAGE BOOSTING TECHNIQUES

So far, we have passed a part of both branches of the flow simultaneously and can conclude the following points:

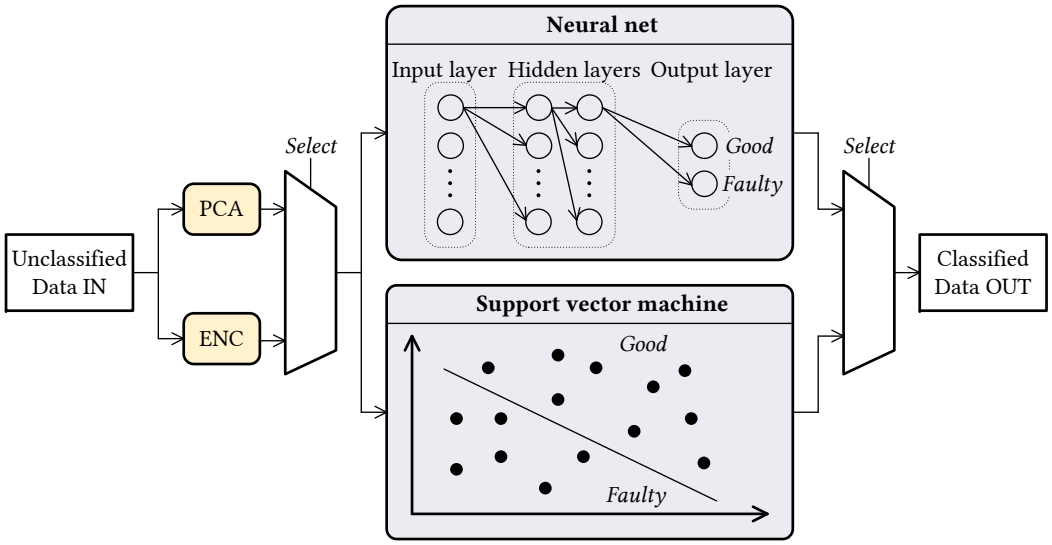

Fig. 14. Detailed implementations of the machine learning techniques. PCA and encoders are used to reduce the signal dimensions into small features. NN and SVM are compared in terms of their performance.

- Block-level testing performs worse than system-level testing due to the coverage of the DfT circuitry.
- Block-level testing is much less sensitive to measurement variations.
- There is room to improve the fault coverage (FC) versus yield loss (YL) trade-off, as it reaches a maximum of 80% FC at 1.5% YL.

In this section, we will continue with the flow and apply the boosting technique, which was introduced in Fig. 9.

7.1 Implementation of the pseudo-random measurement

As a first step, the test sequence needs to be enriched with additional measurements to enable catching more defects. In this work, we use a frequency analysis of the circuit's response to a pseudo-random signal, implemented as follows. The available probe pads are used as the insertion points for the pseudo-random signal used as extra input. The signal is inserted at the supply voltage for the system-level tests and at the DfT injection points for the block-level testing. The bandwidth (BW) of the pseudo-random signal itself is determined by the capabilities of the arbitrary waveform generator on a typical ATE. In our case a BW of 20 MHz is used. The power of the signal is limited in such a way that the amplitude of the inserted signal does not exceed the technology's safe operating regions.

7.2 Classify circuit instances with machine learning algorithms

Following up on the measurement results obtained with the pseudo-random signal, a machine learning algorithm is used to extract information from the test results and to classify circuits between good and faulty instances. In order to find optimal results, several techniques can be used to build the feature extraction block and the classifier in this case. We will compare (a) two feature extraction methods, namely an encoder composed of a neural network (NN) encoder and a method based on Principal Component Analysis (PCA), and (b) two types of classifier, namely

a feedforward NN and support vector machines (SVM). These are shown in Fig. 14. Keeping the generality of the proposed test approach, other types of feature extraction and dimension reduction schemes can be used as well as different machine learning or classification algorithms are possible in this case, we provide here examples in the context of our case study.

Feature selection methods.

- **Principal Component Analysis.**
  PCA is a statistical technique enabling to reduce data dimensionality, by extracting the principal components (or vectors) of the data while preserving a maximum of information. This is necessary since the measured signals are time series with very high dimensions (more than 5000). These principal vectors are then used as features for the machine learning algorithm. In our case, the PCA matrix used for the feature extraction is based only on the Monte-Carlo simulations of the good population, considering only process variations.
- **Automatic feature extraction with encoder.**
  Features can also be extracted using deep learning techniques. Encoders based on neural networks are popular among them, and have recently been used in the context of test escape detection [19, 27]. Their principle is similar to PCA but enables to extract non-linear features. The data is the same as used for the PCA.

Types of classifiers.

- **Neural network.**
  Neural networks are one of the most popular machine learning techniques, which have proven their ability to learn in numerous contexts. In this work, we will use a feedforward neural network, fully connected, trained with the usual back propagation algorithm. It is composed of 4 layers: the input layer, whose size depends on the number of features, two hidden layers of 50 neurons each, and an output layer with one node per class to predict.
- **Support vector machines.**
  The general principle of SVMs is to find a boundary (in the form of a hyperplane) between the two different classes, so that data points of the different classes lie as far as possible from each other. This boundary can be moved by changing support vectors, whose value are determined during training. In our case, Radial Basis Functions are used as kernel functions for the SVM.

7.3 Comparison of feature extractions and machine learning algorithms

7.3.1 Details on feature extraction with PCA. In order to reduce the dimensionality of the measured or simulated (in our case) signals, the PCA features $\vec{g}_i$ are extracted using a PCA matrix W_{PCA}, constructed from the data. These distinctive features are based on a set of T training spectra $\vec{s}_i$, which contain K frequency bins. These training vectors make up the matrix X. The L eigenvectors $\vec{v}_j$ with the largest eigenvalues $\vec{\lambda}_j$ of XX^T are then computed and normalized to produce $\vec{u}_j$:

$$\vec{u}_j = \frac{1}{sqrt(K\lambda_j)} X^T \vec{v}_j \tag{3}$$

Using these normalized eigenvectors, W_{PCA} is constructed as $W_{PCA} = [\vec{u}_1 \vec{u}_2 \dots \vec{u}_L]$. With this PCA matrix, features $\vec{g}_i$ are extracted from the spectra of the measurements using:

$$\vec{g}_i = W_{PCA}^t \vec{s}_i \tag{4}$$

The reduction of signal dimension to a small signature depends on which signals are applied and how long. Considering the ATE capabilities, the signals are kept in the same order of magnitude

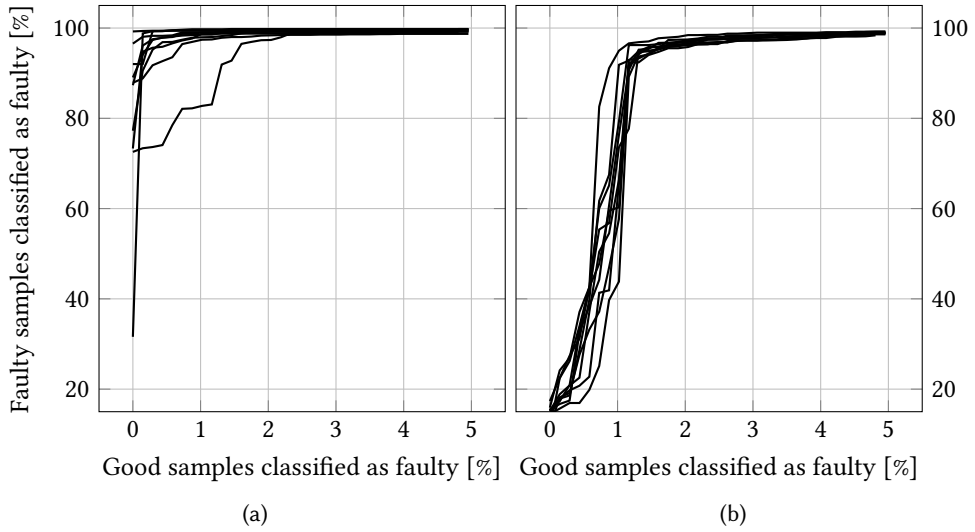

Fig. 15. ROC curves of repeatedly training and classifying with (a) NN and (b) SVM using different parts of the data for training and validation in block-level tests (including DfT) using 10 features per measured signal.

as the other measurements. In our case this operation results in the reduction of dimension by a factor 10^3 per measurement to around the order of 10 per signal.

7.3.2 PCA and machine learning algorithms. The performance of the classification can be shown in Fig. 15, with the ROC (Receiver Operating Characteristic) curves of a tenfold repeated training of the same neural network or SVM, using the same settings, but using different training subsets, i.e. different divisions of the data in the training and validation subsets. Only the results when using block-level testing are shown here for simplicity. Only true and false positives of the classification as faulty are shown, as this is a binary classification. There are two main conclusions from Fig. 15:

(1) the performance without yield loss can have outliers, while in general the coverage will be around 90%, and
(2) the highest achievable coverage is consistently higher than 98% when allowing at most 2% yield loss.

From this it can be concluded that, at minimal yield loss, a consistently high test coverage performance can be achieved using the neural network approach, which will be used in the following subsection.

7.3.3 PCA and NN: optimal number of features. The impact of the number of features, corresponding to the number of Principal Components, has also been investigated. Fig. 16 shows the summary of this analysis on the block-level tests. Here, using a different number of components, the training and validation of the neural net have been repeated ten times and the average is given in the figure. The result is that using less than 10 principal components gives a sub-optimal fault coverage, because more than 20% yield loss is needed to achieve the same fault coverage performance as when using more features. Beyond 10 principal components, there is no further improvement, any change visible is stochastic, and repeated experiments show that any line with more than 10 components lies around the 10-component line. Therefore, only 10 principal components should be used in this case study, to keep the numerical calculations contained while achieving optimum

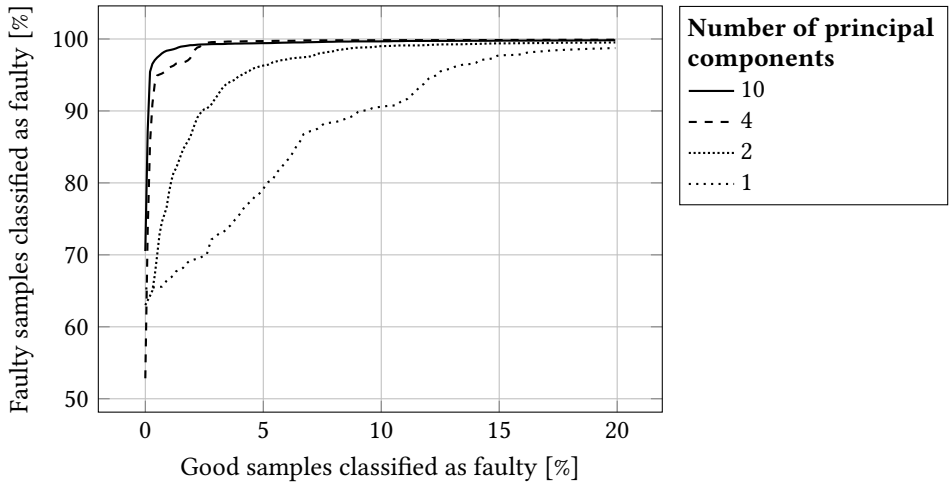

Fig. 16. Average ROC curves over 10 runs using a different number of principal components as features in block-level testing (including DfT).

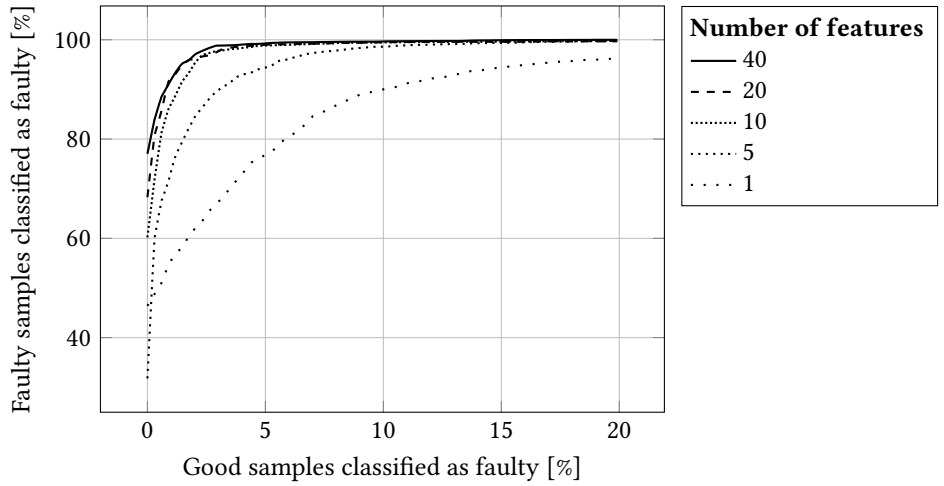

Fig. 17. Average over 10 runs using different number of features with NN encoder.

fault coverage. The consistency in training using the PCA matrix based on good circuits without measurement variations, shows that the principal components of the good circuit are adequate for classifying faulty circuits and that they experience little or no effect under measurement variation.

Deep-learning approach with autoencoder. Instead of using PCA analysis to extract features, an encoder can be used. Combined with a neural network decoding the data, we obtain a complete autoencoder, which has been used recently in the context of testing [19, 27]. We also applied the autoencoder to our case study, in order to evaluate if better features could be extracted to improve the performance of the algorithm. However, the results were worse than when using PCA, as illustrated in Fig. 17. When using 10 features or more, the maximum FC with a yield loss of

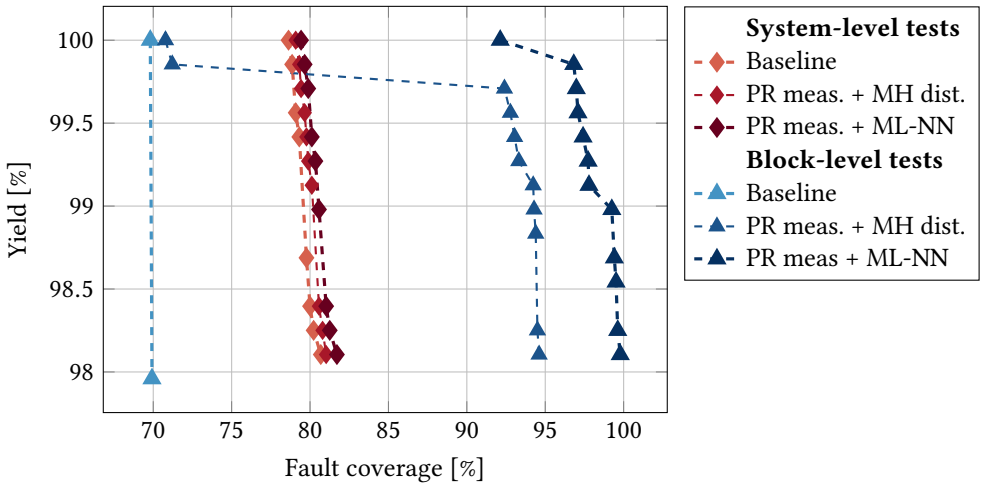

Fig. 18. Trade-off between the maximum achievable fault coverage and the yield for the baseline case and when using pseudo-random signal injection (**PR**) together with a mahalanobis-based metric (**MH**) and machine learning using a neural network (**NN**) for system-level and block-level testing (include DfT).

maximum 2% is around 90%, significantly lower that the PCA counterpart. This can be explained by the fact that PCA focuses on linear components in a more guided way, which are sufficient to describe the relationships between the measurements and the test results. The encoder does not have any prior information on the data and thus leads to lower performance.

7.4 Final fault coverage performance

Summarizing the findings of the previous subsections, the following settings are used to compute the final fault coverage results:

- Feature extraction with PCA; 10 principal components used as features.
- Classification with a neural network with two hidden layers of 50 neurons.
- For the classification performance, the entire population is divided in two halves: one for training the neural network, and one for validating it.

These final settings are evaluated in complement to system-level tests and block-level tests. For the sake of completeness, we also evaluate the impact of the pseudo-random measurements (referred as **PR meas.**) and the machine learning neural network (referred as **ML-NN**) separately. It enables to highlight the ability of machine learning on blocks to push further the fault coverage metrics. The results are summarized in Fig. 18.

Pseudo-random measurement without machine learning. It is possible to use the pseudo-random measurement without machine learning. This can be done, for instance, by computing a metric based on the Manalanobis distance (referred to as **MH**) from the PCA vectors. In detail, feature fingerprints $h_i = \{\vec{g}_i, \sigma_i^2\}$ are constructed by using the variance σ_i^2 of the features $\vec{g}_i$, which are required in the next step. The Mahalanobis distance $\vec{d}_i$ is used as a metric to score how close a fingerprint is to the good circuit, which is represented by the simulated reference feature set. This distance $\vec{d}_i$ between the reference fingerprint h_{ref} and a test's fingerprint h_{test} is defined as:

$$\vec{d}_i = \sqrt{\frac{\left(\vec{g}_{ref} - \vec{g}_{test}\right)^t \left(\vec{g}_{ref} - \vec{g}_{test}\right)}{\min\left(\sigma^2_{test}, \sigma^2_{ref}\right)}} \tag{5}$$

The final classification is made between good and faulty circuits based on these calculated distances. The threshold is, again, optimized such that a proper trade-off can be constructed between fault coverage and yield. Other non-machine learning schemes can be used as well, as the flow allows the use of multiple classification schemes. Some others (random forest and SVMs) were tried as well and their performance is comparable to that of **MH**. Due to their similarity, these are not shown in the comparison in Fig. 18.

Pseudo-random measurement with machine learning. Using the neural network, features are directly presented at the input of the neural network, which is trained to learn the boundary between the two classes.

Block-level versus system-level tests. Considering system-level testing, the ML boosting technique has a limited effect on the overall fault coverage, as illustrated in Fig. 18. This can be explained because it is only possible to apply the pseudo-random signal on top of the supply voltage in this case. This results in a fault coverage approaching 80% which is minimal with respect to the baseline. The trade-off for yield is also the same as the baseline. This limited impact can be explained due to the power supply rejection properties of the circuit: as fluctuations of the supply are suppressed, so is the effect of injecting a test signal at the supply.

For block-level testing, however, the improvements are substantial. The achieved fault coverage levels exceed those of system-level testing by far. This can be explained by the fact that the response follows a signal path that passes through the DfT structures, thereby testing these as well. First, we can observe that using the pseudo-random measurements without machine learning already improves the fault coverage significantly above the baseline, considering a yield loss higher than 0.2%, with a general fault coverage around 95% at 2% yield loss. In addition, the use of machine learning on top of the pseudo-random measurements significantly increases the overall fault detection performance. It results in coverage levels around 90% without yield loss, while improving substantially and passing 98% fault coverage at the 2% yield loss point. This proves the ability of the proposed technique to boost the fault coverage, in this case using block-level testing as a baseline, while at the same time performing all the specification measurements. Using this method on top of system-level testing has no extra benefit as it would only cost testing time, while everything is already measured by the block-level tests.

8 CONCLUSIONS

This paper has presented a flow designed to optimize the test fault coverage of AMS ICs. Most steps can be automated. Both system-level and block-level testing methodologies are compared in terms of their fault coverage. Both methodologies contain both functional and structural measurements. The flow takes into account measurement variations and looks for a test approach such that the tests are minimally impacted by these measurement variations. The flow improves the fault coverage by adding extra measurements, such as injecting pseudo-random signals into the circuit, and applying machine learning classification to the response of these signals.

As a case study to illustrate the flow, an industry-oriented DC-DC converter has been used. When considering only functional and structural tests, system-level testing has the highest fault coverage, while block-level testing has a poor coverage due to the extra DfT structures. With the extra measurements added, system-level tests perform worse for the DC-DC converter than

block-level tests, achieving just under 80% fault coverage without any yield loss. Allowing for yield loss only improves the fault coverage by 1% per 1% yield loss. Block-level testing, on the other hand, shows a fault coverage of more than 90% without yield loss, at the expense of some extra DfT structures. Allowing for up to 2% yield loss increases the coverage to over 98%. These results have been obtained by using a fully connected neural net classifier on features extracted from the fault simulations through Principal Component Analysis.

ACKNOWLEDGEMENTS

This research has been conducted within the framework of the SAFE-IC2 and NoRMA projects partially funded by VLAIO, the Flemish Agency for Innovation by Science and Technology (code: IWT-150433).

REFERENCES

[1] M. Andraud, L. Galindez, Y. Lu, Y. Makris, and M. Verhelst. 2018. On the use of Bayesian Networks for Resource-Efficient Self-Calibration of Analog/RF ICs. In *2018 IEEE International Test Conference (ITC)*. 1–10. https://doi.org/10.1109/TEST.2018.8624893

[2] M. Andraud, H. Stratigopoulos, and E. Simeu. 2016. One-Shot Non-Intrusive Calibration Against Process Variations for Analog/RF Circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers* 63, 11 (Nov 2016), 2022–2035.

[3] D. Banerjee and A. Chatterjee. 2014. Signature driven hierarchical post-manufacture tuning of RF systems for performance and power. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 23, 2 (2014), 342–355.

[4] N. Beohar, P. Bakliwal, S. Roy, D. Mandal, P. Adell, B. Vermeire, B. Bakkaloglu, and S. Ozev. 2015. Disturbance-free BIST for loop characterization of DC-DC buck converters. In *Proceedings IEEE 33rd VLSI Test Symposium (VTS)*. 1–6. https://doi.org/10.1109/VTS.2015.7116250

[5] S. Biswas and B Cory. 2012. An Industrial Study of System-Level Test. *IEEE Design Test* 29, 1 (Jan/Feb 2012), 19–27.

[6] P. Collins, S. Yu, K. R. Eckersall, B. W. Jervis, I. M. Bell, and G. E. Taylor. 1994. Application of Kohonen and supervised forced organisation maps to fault diagnosis in CMOS opamps. *Electronics Letters* 30, 22 (Oct 1994), 1846–1847. https://doi.org/10.1049/el:19941281

[7] A. Coyette, B. Esen, W. Dobbelaere, R. Vanhooren, and G. Gielen. 2016. Automatic test signal generation for mixed-signal integrated circuits using circuit partitioning and interval analysis. In *Proceedings IEEE International Test Conference (ITC)*. 1–10. https://doi.org/10.1109/TEST.2016.7805867

[8] A. Coyette, B. Esen, N. Xama, G. Gielen, W. Dobbelaere, and R. Vanhooren. 2018. ADAGE: Automatic DfT-Assisted Generation of Test Stimuli for Mixed- Signal Integrated Circuits. *IEEE Design Test* 35, 3 (June 2018), 24–30. https://doi.org/10.1109/MDAT.2018.2799800

[9] Boris Danev, Thomas S Heydt-Benjamin, and Srdjan Capkun. 2009. Physical-layer Identification of RFID Devices. In *Usenix Security Symposium*. 199–214.

[10] S. R. Das, J. Zakizadeh, S. Biswas, M. H. Assaf, A. R. Nayak, E. M. Petriu, W. Jone, and M. Sahinoglu. 2007. Testing Analog and Mixed-Signal Circuits With Built-In Hardware: A New Approach. *IEEE Transactions on Instrumentation and Measurement* 56, 3 (June 2007), 840–855. https://doi.org/10.1109/TIM.2007.894223

[11] S. Deyati, B. J. Muldrey, and A. Chatterjee. 2016. TRAP: Test Generation Driven Classification of Analog/RF ICs Using Adaptive Probabilistic Clustering Algorithm. In *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*. 463–468. https://doi.org/10.1109/VLSID.2016.118

[12] W. Dobbelaere, R. Vanhooren, W. De Man, K. Matthijs, A. Coyette, B. Esen, and G. Gielen. 2016. Analog fault coverage improvement using final-test dynamic part average testing. In *Proceedings IEEE International Test Conference (ITC)*. 1–9. https://doi.org/10.1109/TEST.2016.7805829

[13] M. Dubois, H. G. Stratigopoulos, and S. Mir. 2012. Ternary Stimulus for Fully Digital Dynamic Testing of SC S? ADCs. In *2012 IEEE 18th International Mixed-Signal, Sensors, and Systems Test Workshop*. 5–10. https://doi.org/10.1109/IMS3TW.2012.12

[14] B. Esen, A. Coyette, N. Xama, W. Dobbelaere, R. Vanhooren, and G. Gielen. 2017. A very low cost and highly parallel DfT method for analog and mixed-signal circuits. In *Proceedings 22nd IEEE European Test Symposium (ETS)*. 1–2. https://doi.org/10.1109/ETS.2017.7968225

[15] K. Huang, H. Stratigopoulos, and S. Mir. 2010. Fault diagnosis of analog circuits based on machine learning. In *2010 Design, Automation Test in Europe Conference Exhibition (DATE 2010)*. 1761–1766. https://doi.org/10.1109/DATE.2010.5457099

[16] ICInsights. 2018. Automotive Electronic Systems Growth Strongest Through 2021. http://www.icinsights.com/news/bulletins/Automotive-Electronic-Systems-Growth-Strongest-Through-2021/. [Online; 25-March-2018].

[17] J. Kim. 2009. On-Chip Measurement of Jitter Transfer and Supply Sensitivity of PLL/DLLs. *IEEE Transactions on Circuits and Systems II: Express Briefs* 56, 6 (June 2009), 449–453.

[18] January Kister and Steve Hopkins. 2007. Electrical Contact Resistance - the Key Parameter in Probe Card Performance. In *Proceedings IEEE Semiconductor Test Workshop (SWTW), San Diego, CA, USA*.

[19] F. Lin and K. Cheng. 2017. An artificial neural network approach for screening test escapes. In *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. 414–419. https://doi.org/10.1109/ASPDAC.2017.7858358

[20] Y. Lu, K. S. Subramani, H. Huang, N. Kupp, and Y. Makris. 2015. A Comparative Study of One-Shot Statistical Calibration Methods for Analog/RF ICs. *Proc. of IEEE International Test Conference*, 1–10.

[21] Y. Maiden, B. W. Jervis, N. Dutton, and S. Lesage. 1997. Diagnosis of multifaults in analogue circuits using multilayer perceptrons. *IEE Proceedings - Circuits, Devices and Systems* 144, 3 (June 1997), 149–154. https://doi.org/10.1049/ip-cds:19971146

[22] L. S. Milor. 1998. A Tutorial Introduction to Research on Analog and Mixed-Signal Circuit testing. *IEEE Transactions of Circuits and systems II - Analog and Digital Processing* 45, 10 (Oct 1998), 1389–1407.

[23] N. Mohan, T. M. Undeland, and W. P. Robbins. 2002. *Power electronics: Converters, Applications, and Design* (3 ed.). John Wiley & Sons, New York.

[24] V. Natarajan, A. Banerjee, S. Sen, S. Devarakond, and A. Chatterjee. 2015. Yield recovery of RF transceiver systems using iterative tuning-driven power-conscious performance optimization. *IEEE Design & Test of Computers* 32, 1 (2015), 61–69.

[25] R. Rajsuman. 2000. Iddq testing for CMOS VLSI. *Proc. IEEE* 88, 4 (April 2000), 544–568. https://doi.org/10.1109/5.843000

[26] Ricardo Reis, Marcelo Soares Lubaszewski, and Jochen AG Jess. 2007. *Design of systems on a chip: design and test*. Springer.

[27] M. Shintani, M. Inoue, and Y. Nakamura. 2018. Artificial Neural Network Based Test Escape Screening Using Generative Model. In *2018 IEEE International Test Conference (ITC)*. 1–8. https://doi.org/10.1109/TEST.2018.8624821

[28] S. A. S. Somayajula, E. Sanchez-Sinencio, and J. Pineda de Gyvez. 1996. Analog fault diagnosis based on ramping power supply current signature clusters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 43, 10 (Oct 1996), 703–712.

[29] R. Spina and S. Upadhyaya. 1997. Linear circuit fault diagnosis using neuromorphic analyzers. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 44, 3 (Mar 1997), 188–196. https://doi.org/10.1109/82.558453

[30] H. Stratigopoulos. 2018. Machine learning applications in IC testing. In *2018 IEEE 23rd European Test Symposium (ETS)*. 1–10. https://doi.org/10.1109/ETS.2018.8400701

[31] H. Stratigopoulos, S. Mir, E. Acar, and S. Ozev. 2009. Defect Filter for Alternate RF Test. In *2009 14th IEEE European Test Symposium*. 101–106. https://doi.org/10.1109/ETS.2009.32

[32] H. G. Stratigopoulos and Y. Makris. 2005. Nonlinear decision boundaries for testing analog circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 24, 11 (Nov 2005), 1760–1773. https://doi.org/10.1109/TCAD.2005.855835

[33] Stephen Sunter and Peter Sarson. 2017. A/MS benchmark circuits for comparing fault simulation, DFT, and test generation methods. *2017 IEEE International Test Conference (ITC)* (2017), 1–7.

[34] Texas Instruments. 2004. IV. High-Speed Layout. In *European High Speed Analog Design & Solution Seminar*.

[35] P. N. Variyam, S. Cherubal, and A. Chatterjee. 2002. Prediction of Analog Performance Parameters Using Fast Transient Testing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 21, 3 (2002), 349–361.

[36] E. Yilmaz, S. Ozev, and K. M. Butler. 2011. Adaptive multidimensional outlier analysis for analog and mixed signal circuits. In *2011 IEEE International Test Conference*. 1–8. https://doi.org/10.1109/TEST.2011.6139178

[37] S. Yu, B. W. Jervis, K. R. Eckersall, I. M. Bell, A. G. Hall, and G. E. Taylor. 1994. Neural network approach to fault diagnosis in CMOS opamps with gate oxide short faults. *Electronics Letters* 30, 9 (April 1994), 695–696. https://doi.org/10.1049/el:19940472