Review of Methodologies for Pre- and Post-Silicon Analog Verification in Mixed-Signal SOCs

Georges Gielen*, Nektar Xama*, Karthik Ganesan[†], and Subhasish Mitra[†]

*Dept. Electrical Engineering, KU Leuven, Belgium
[†]Dept. Computer Science, Stanford University, California, USA

Abstract—The integration of increasingly more complex and heterogeneous SOCs results in ever more complicated demands for the verification of the system and its underlying subsystems. Pre-silicon design validation as well as post-silicon test generation of the analog and mixed-signal (AMS) subsystems within SOCs proves extremely challenging as these subsystems do not share the formal description potential of their digital counterparts. Several methods have been developed to cope with this lack of formalization during AMS pre-silicon validation, including model checkers, affine arithmetic formalisms and equivalence checkers. However, contrary to the industrial practice for digital circuits of using formal verification and ATPG tools, common industry practice for analog circuits still largely defaults to simulation-based validation and test generation. A new formal digital-inspired technique, called AMS-QED, can potentially solve these issues in analog and mixed-signal verification.

I. INTRODUCTION

The development of ever more complex and heterogeneous Systems-On-Chip (SOCs) nowadays poses ever higher demands on verifying the functionality of these SOCs while simultaneously addressing the growing complexity of the system. Applications such as Internet of Things or autonomous vehicles additionally require absolute reliability from the electronic systems. These requirements combined therefore create a demand for effective and scalable techniques to verify both the digital and the analog/mixed-signal (AMS) subsystems included in a mixed-signal SOC. In verification, a clear distinction is made between:

a) the pre-silicon design validation, which focuses on validating the correctness in functionality of a design, and
b) post-silicon test generation, which focuses on verifying the correctness in functioning of a fabricated (sub)system.

In both pre- and post-silicon verification, digital subsystems have a clear advantage. While many methods and tools have been developed and are widely used in industry to both formally validate and automatically generate tests (ATPG) for digital subsystems, the AMS verification tools are clearly lagging behind. Digital verification is at the stage where other considerations are coming into play, such as security [1].

In common industrial practice today, analog and mixed-signal circuits are still largely verified based on repeated-simulations methodologies, even though some formal methods and tools have been explored and presented
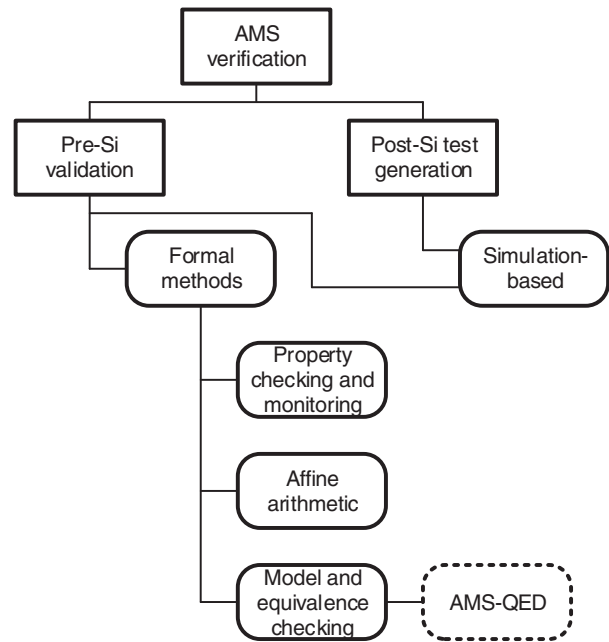

Fig. 1. Diagram indicating a classification of the different techniques for analog/mixed-signal pre- and post-silicon verification.

over the past years. The inherent problems of verifying analog circuits lie in the facts that:

a) there is no efficient or widely used method to formally describe their function, whereas commercial simulators are used extensively in all steps of the design;
b) the performance of analog circuits is parametrically sensitive to all kinds of variations (process, supply, temperature, aging), which makes their verification complicated; and
c) the impact of fabrication defects on analog circuit functionality is less predictable.

Nevertheless, with the growing complexity levels and the tightening reliability and robustness constraints of the next generation of systems, pre- and post-silicon AMS design verification must also guarantee correctness with higher levels of confidence, hence necessitating the adoption of more formal tools. Additionally, the use of formal methods may also be expected to improve design productivity, like they have done in the case of digital systems.

This review paper gives a comparative overview of the

state of the art of different AMS verification techniques presented until now. A classification of the methods is shown in Fig. 1. Verification is classified into two main clusters depending on the moment when the verification is done: pre-silicon validation and post-silicon testing and test generation. Simulation-based techniques are widely used for both purposes, while formal methods are mainly used in the pre-silicon stage.

The remainder of this overview paper is organized as follows. Section II will address pre-silicon validation and describe the different AMS techniques used at this stage, ranging from formal (property checking, affine arithmetic, model and equivalence checking) to simulation-based techniques, as well as their combination. Section III will then describe the mainly simulation-based methods use for state-of-the-art post-silicon test generation. A new promising technique, called AMS-QED, that combines concepts from digital formal verification with AMS circuits, will then be introduced in section IV. Conclusions will be formulated in section V.

II. Pre-silicon verification: design validation

AMS circuit verification at design time can broadly be classified into two approaches: formal methods and simulation-based techniques.

a) Formal methods attempt to validate in some formal manner if the designed circuit is satisfying the correctness constraints. They typically step away from physics-based AMS simulations and try to work at a higher level of circuit abstraction, as this can lead to a significant speed-up in design validation time, be it that they can suffer from inaccuracies due to this abstraction.

b) Simulation-based techniques, on the other hand, start from the physics of the devices within the circuit and validate the design by sampling its performance through simulating the circuit a number of times under different inputs and conditions. This has the benefit that it takes many more physical details into account. However, since these simulations are time-intensive, this comes at the cost of a large validation time. In addition, the limited set of simulations carried out does not guarantee formally that the design is correct under all circumstances.

A. Formal methods

The formal methods can be classified in three groups: property checking and monitoring, affine arithmetic, and model or equivalence checking.

1) Property checking and monitoring: The first group of formal methods, i.e. property checking and monitoring, generalizes a property to be checked or monitored in a temporal logic language, such as Signal Temporal Logic in [2], which combines property checking with a standalone monitoring tool. The properties are described in STL, which can describe signal properties over a continuous range and can add time-based triggers to that description. Another work [3] proposes a different specification language with a
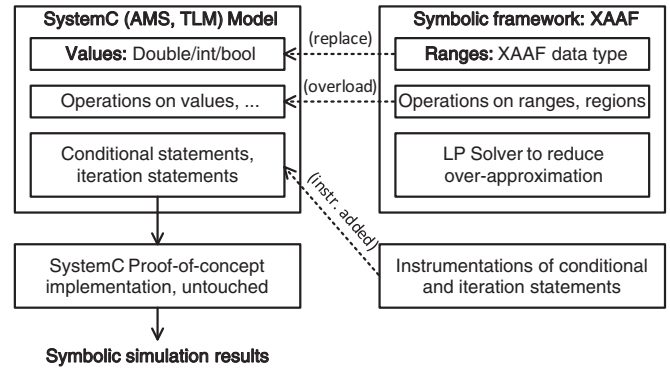

Fig. 2. Diagram of XAAF-based symbolic simulation (adapted from [6]).

complementing, fully automated design flow using property checking as well. In general, these works describe tools and a design flow that interleaves the actual design and the validation at the same time. The downside of property checking lies in how well the logic and languages can describe the actual signal and their desired behavior and how complete the set of properties is to validate the circuit.

2) Affine arithmetic: The second group of formal methods uses some form of affine arithmetics [4], which is an extension of interval analysis, with the added benefit that uncertainty values may be represented symbolically. These methods aim to prove formally that the actual circuit implementation is containing the specifications. An extension to affine arithmetics is introduced in [5], [6] as the Extended Affine Arithmetic Form (XAAF), that allows standard arithmetic as well as relational operators, which further allows control flow. Fig. 2 shows the diagram of how XAAFs can be used in the design simulation flow (in this figure a SystemC model flow) [6]. An LP solver is used to reduce over-approximation. Despite the large potential gain in simulation time, a very large effort has to be put into generating these forms. The authors assert though that this could be solved using automation [5].

3) Model and equivalence checkers: The last group of formal methods consists of model and equivalence checkers. The formal checkers use an AMS subsystem model to represent the subsystem. This model can be described by a certain logic, for which propositions can be made that relate to the specifications. The propositions can be expressed in logic statement similar to the first group of methods. In [7], for example, Bounded Linear Temporal Logic (BLTL) is used. Furthermore, stochastic model checking is done by stochastic sampling of transient simulations and testing one or more Bayesian hypotheses using this model [8]. Instead of BLTL, using purely Boolean models has also been proposed.

Next to the use of logic, a method for equivalence checking between an analog transistor-level netlist of a circuit and its behavioral model is presented in [9] for the complete reachable state space of that circuit. An automatic stimuli generation algorithm has been presented that generates inputs for the netlist to cover the entire reachable state space described by an Analog Transition System (ATS), i.e. a discretized form of the differential equations that define the circuit. It is also possible

to generate an ATS from the behavioral model, thus allowing a comparison between both systems. Because the state space is discretized, an efficient exploration is possible for that refinement of the discretization. The method is applicable in general, as it only needs a netlist description and a behavioral model.

B. Simulation-based validation

The industrial AMS IC design practice typically uses simulation-based validation as the default method. This has the benefit that the circuit is represented very reliably (up to the precision of the device models used and the parasitics incorporated in the netlist). However, the downside is that the simulation time can be excessively long, especially for transient simulations or when simulating for time-dependent reliability for instance. Fortunately, improvements in this regard are possible. For example, piecewise-linear (PWL) modeling allows speed-up of both DC and transient simulations, with limited loss of accuracy [10]. As transient simulations are especially time-consuming, several improvements have been proposed to accelerate this type of simulations. In [11], linearized state-space models are generated that approximate the behavior of the analog part, which can then be used to bypass the numerically intensive integration part of the transient simulations. A more complete technique for pre-silicon validation, called Built-In State Consistency Checking (BISCC), has been presented in [12]. This technique uses an algorithm that generates two input stimuli with identical initial condition, that cause an internal node voltage or branch current to be consistent for both stimuli, which is then checked during validation.

C. Combining simulations and formal methods

Several works have also been presented that describe more complete design methodologies containing a combination of both formal and simulation-based techniques. One of these is named Recursively Verifying and Modeling (RVM) [13], where an automated tool generates and validates behavioral models based on transistor-level simulations. This behavioral model is then used to verify the circuit's correctness. This methodology further employs a hierarchical modeling framework to cope with the availability and complexity of the IC systems to be verified. The methodology presented in [14] follows a somewhat similar structure by also having a hierarchical approach due to its circuit partitioning scheme, as well as having a model generator. After model generation, however, it does use a different technique for validation: it uses an equivalence checking technique. Both of these methodologies show that for AMS systems, pre-silicon validation can become more feasible when using a hybrid approach, involving both formal methods and simulation techniques.

III. POST-SILICON TESTING AND TEST GENERATION

For the post-silicon validation and testing, AMS test generation today primarily involves simulation-based
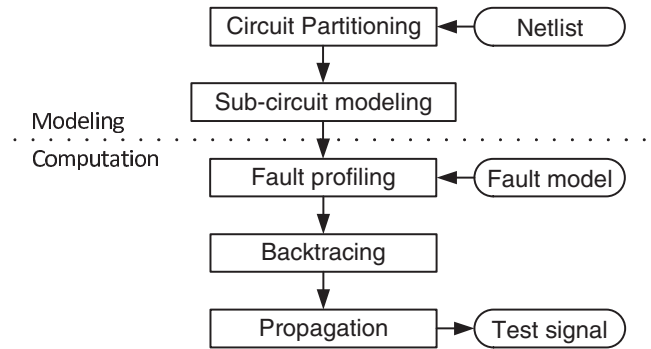

Fig. 3. Flow diagram of the ADAGE automated ATSG method [15].

techniques, that simulate both the fault-free and the faulty circuits, as the representation of the system needs to be as close to reality as possible. As such, these techniques are similar to those explained in subsection II-B. Techniques from formal methods are however used to boost the time taken by AMS test generation. For example, interval analysis and PWL models are used in the ADAGE methodology [15] to automatically generate analog input test signals (ATSG) and select the test output signals to catch as many as possible of the potential hard defects that can occur in the circuit. A flow diagram of this ADAGE ATSG method is shown in Fig. 3. The subcircuit modeling step uses PWL models and fault profiling uses interval analysis to make the backtracing step (for fault activation) possible and the forward propagation step (for fault observation) much faster compared to standard simulations. The work not only maximizes the detection coverage of hard defects in the AMS system under test, but - if full coverage cannot be reached with the given inputs and outputs - it also automatically inserts extra Design-for-Test (DfT) structures to further increase the test coverage. The DfT circuitry demonstrated in [15] consists of adding extra "stress" transistors to pull specific nodes to the supply or ground level, thus introducing extra test states that enable an increase in test coverage.

Other techniques contain algorithms that use the distinctive signature property of the signals in order to perform post-silicon validation and test generation. For example, in [16], a signature-based methodology is presented to isolate electrical bugs and to compensate for yield loss. This method resembles equivalence checking, because it compares the behavioral model of the circuit under test to the fabricated circuit. The input stimuli are generated in such a way that electrical bugs are differentiated by demanding that the response is maximally different between good and buggy samples. The technique in [16] is mainly aimed at RF circuits, as their inherent operation already contains signature behavior. For these circuits, the BISCC technique described in subsection II-B can also be used for test generation, as this technique results in a set of input stimuli that can be used in a test environment. This technique further proposes DfT structures that sense and compare internal nodes to check their values and thus provide test observability.
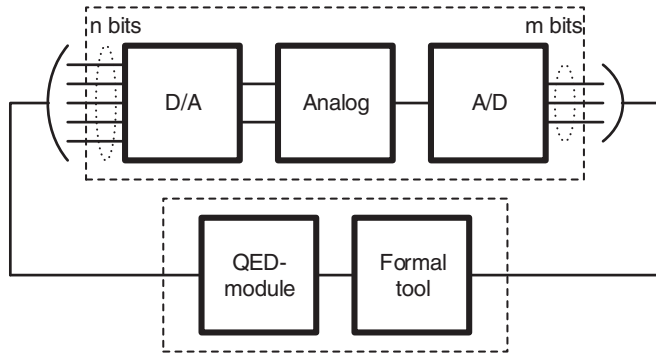
Fig. 4. Flow diagram of AMS-QED.

IV. AMS-QED

As digital verification techniques have proven to be significantly more effective compared to AMS verification, drawing inspiration from successful digital techniques can prove vital in improving AMS verification. A recent successful digital technique is called Quick Error Detection (QED) [17]. It is designed to very efficiently detect bugs by introducing a set of specific QED transformations that detect logic and electric errors in the digital circuit. Significant reductions in error detection time have been shown on a commercially available multicore processor [17]. QED was originally developed for post-silicon validation of processor cores, with the QED transformations tailored towards the Instruction Set Architecture (ISA) of the processor being validated. More recently, the QED concept has been expanded more broadly. In particular, Symbolic Quick Error Detection (SQED) [18] is a bounded-model checking based formal verification technique that detects logic bugs during pre-silicon verification.

Using the principles from SQED, the technique can potentially be extended to apply to AMS subsystem verification. This newly proposed AMS-QED method is conceptually shown in Fig 4. Here the Bounded Model Checker or formal tool drives a QED module that contains AMS equivalents of the QED transformations. The transformations are used to generate the input of the AMS block and are transformed into the analog domain to be used in the analog subsystem. The output as a result of this input is then digitized and used as input of the formal tool. This digitization, in essence, transforms the analog circuit into an equivalent digital one. The technique requires a model for the transformed digital circuit, that includes the characteristics of this transformation. The AMS-QED methodology is currently in full development.

V. CONCLUSIONS

This survey paper has presented an overview and classification of methods for the pre- and post-silicon verification of analog/mixed-signal circuits. Despite today's common industrial practice of primarily using simulations, several formal methods have been developed, ranging from model checking and affine arithmetic to equivalence checking, in order to formally verify the correctness of analog circuits.

Even though their effectiveness has been shown for various cases, they are not yet widespread used in industry. A new formal method, called AMS-QED, is inspired on recent successful digital methods, and aims at efficient AMS verification with broad applicability.

REFERENCES

[1] X. e. a. Guo, "Pre-silicon Security Verification and Validation: A Formal Perspective," in *DAC*. New York, NY, USA: ACM, 2015, pp. 145:1–145:6.

[2] O. Maler and D. Ničković, "Monitoring properties of analog and mixed-signal circuits," *International Journal on Software Tools for Technology Transfer*, vol. 15, no. 3, pp. 247–268, Jun 2013.

[3] M. Ma, L. Hedrich, and C. Sporrer, "ASDeX: a formal specification for analog circuit enabling a full automated design validation," *Design Automation for Embedded Systems*, vol. 18, no. 1, pp. 99–118, Mar 2014.

[4] C. Radojicic, F. Schupfer, M. Rathmair, and C. Grimm, "Assertion-based verification of signal processing systems with affine arithmetic," in *Proceeding of the 2012 Forum on Specification and Design Languages*, Sept 2012, pp. 20–26.

[5] C. Radojicic and C. Grimm, "Formal verification of mixed-signal designs using extended affine arithmetic," in *PRIME*, June 2016.

[6] E. B. et al., "Embedded tutorial: Analog-/mixed-signal verification methods for AMS coverage analysis," in *DATE*, 2016, pp. 1102–1111.

[7] Y. Wang, A. Komuravelli, P. Zuliani, and E. M. Clarke, "Analog circuit verification by statistical model checking," in *16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011)*, Jan 2011, pp. 1–6.

[8] A. V. Karthik, S. Ray, P. Nuzzo, A. Mishchenko, R. Brayton, and J. Roychowdhury, "ABCD-NL: Approximating Continuous non-linear dynamical systems using purely Boolean models for analog/mixed-signal verification," in *ASP-DAC*, Jan 2014, pp. 250–255.

[9] S. Steinhorst and L. Hedrich, "Equivalence checking of nonlinear analog circuits for hierarchical AMS System Verification," in *2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC)*, Oct 2012, pp. 135–140.

[10] Y. Zhang, S. Sankaranarayanan, and F. Somenzi, "Piecewise linear modeling of nonlinear devices for formal verification of analog circuits," in *2012 Formal Methods in Computer-Aided Design (FMCAD)*, Oct 2012, pp. 196–203.

[11] S. H. et al., "Efficient generation of analog circuit models for accelerated mixed-signal simulation," in *International SOC Conference*, 2012.

[12] S. Deyati, B. Muldrey, and A. Chatterjee, "BISCC: Efficient pre through post silicon validation of mixed-signal/RF systems using built in state consistency checking," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 274–277.

[13] C. J. R. Shi, "Mixed-signal system-on-chip verification using a recursively-verifying-modeling (RVM) methodology," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 1432–1435.

[14] B. C. Lim, J. Jang, J. Mao, J. Kim, and M. Horowitz, "Digital analog design: Enabling mixed-signal system validation," *IEEE Design Test*, vol. 32, no. 1, pp. 44–52, Feb 2015.

[15] A. Coyette, B. Esen, N. Xama, G. Gielen, W. Dobbelaere, and R. Vanhooren, "ADAGE: Automatic DfT-Assisted Generation of Test Stimuli for Mixed- Signal Integrated Circuits," *IEEE Design Test*, vol. 35, no. 3, pp. 24–30, June 2018.

[16] A. Chatterjee, S. Deyati, B. Muldrey, S. Devarakond, and A. Banerjee, "Validation signature testing: A methodology for post-silicon validation of analog/mixed-signal circuits," in *ICCAD*, Nov 2012, pp. 553–556.

[17] T. H. et al., "QED: Quick Error Detection tests for effective post-silicon validation," in *ITC*, 2010.

[18] E. Singh, D. Lin, C. Barrett, and S. Mitra, "Logic Bug Detection and Localization Using Symbolic Quick Error Detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2018.