

**Dubikhin V, Myers C, Sokolov D, Syranidis I, Yakovlev A. Advances in Formal Methods for the Design of Analog/Mixed-Signal Systems. In: DAC '17 Proceedings of the 54th Annual Design Automation Conference. 2017, Austin, TX, USA: Association for Computing Machinery.**

**Copyright:**

© Authors 2017. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *DAC '17 Proceedings of the 54th Annual Design Automation Conference*, <https://doi.org/10.1145/3061639.3072945>.

**DOI link to article:**

<https://doi.org/10.1145/3061639.3072945>

**Date deposited:**

07/08/2017

# Advances in Formal Methods for the Design of Analog/Mixed-Signal Systems

Vladimir Dubikhin  
Newcastle University  
Newcastle upon Tyne, UK  
v.dubikhin1@ncl.ac.uk

Chris Myers  
University of Utah  
Salt Lake City, USA  
myers@ece.utah.edu

Danil Sokolov  
Newcastle University  
Newcastle upon Tyne, UK  
danil.sokolov@ncl.ac.uk

Ioannis Syranidis  
Newcastle University  
Newcastle upon Tyne, UK  
ioannis.syranidis@gmail.com

Alex Yakovlev  
Newcastle University  
Newcastle upon Tyne, UK  
alex.yakovlev@ncl.ac.uk

## ABSTRACT

Analog/mixed-signal (AMS) systems are rapidly expanding in all domains of information and communication technology. They are a critical part of the support for large-scale high-performance digital systems, provide important functionalities in medium-scale embedded and mobile systems, and act as a core organ of autonomous electronics such as sensor nodes. Analog and digital parts are closely intermixed, hence demanding AMS design methods and tools to be more holistic. In particular, the emergence of "little digital" electronics inside or near analog circuitry calls for the increasing use of asynchronous logic. To cope with the growing complexity of AMS designs, formal methods are required to complement traditional simulation approaches. This paper presents an overview of the state-of-the-art in AMS formal verification and asynchronous design that enables the development of analog/asynchronous co-design methods. One such co-design methodology is exemplified by the LEMA-WORKCRAFT workflow currently under development by the authors.

## Keywords

Formal methods, verification, analog/mixed-signal circuits

## 1. INTRODUCTION

While digital design has shifted its research focus to the system level, *analog/mixed-signal* (AMS) design is undergoing an interesting metamorphosis. After years of surrendering many signal processing domains to digital electronics, AMS is now in a stronger position to once again take on digital in these domains. From being on the periphery of complex systems (i.e., restricted to ADCs, amplifiers, power blocks, RF, etc.), it is now moving into the very heart of such systems. This is exemplified by the emergence of: (i) many-

core systems, with numerous time and power domains, needing timing control and power regulation, equipped with sensors for process, voltage and temperature variations; (ii) autonomous self-powered sensor nodes, e.g. in *Internet-of-Things* (IoT), with energy harvesting and power electronics placed closely with the computation and communication electronics (cf. Ambiq's ultra low power MCU technology is based on a symbiosis of digital electronics and voltage regulators, <http://ambiqmicro.com/>). Furthermore, power management IC design is becoming an area of rapid growth in research — the size, performance and energy requirements, as well as the overall holistic nature of modern system engineering, all call for a much more radical innovation in power converter and controller design than ever before. Semico estimated (end of 2015) that "121 billion analog ICs will ship in 2015 and that by 2020 the analog IC market will be worth \$56.5 billion".

AMS engineers are increasingly involved in designing analog electronics with a significant portion of digital elements in them [1]. Examples include programmable and pipelined ADCs, complex power management circuits, such as multi-phase buck converters, switched-capacitor circuits, etc. These digital parts often perform functions such as calibration control, parameter configuration, switching control, "monitoring and knobbing", etc. Specific reasons for having more digital components are the increasing complexity and functionality of AMS, as well as the migration of mixed-signal towards ultra-deep sub-micron technologies.

As shown in Figure 1, such digital (on-top or within analog) electronics are "little digital" as opposed to "big digital" (i.e. traditional computational) electronics. Designing "little digital" is difficult because it should seamlessly integrate with the analog parts, which are dynamic and notoriously difficult to automate. Using standard design flows such as RTL, which is driven by a clock, is not a good option for "little digital" because analog circuits have their own notion of timing and events. The clocked operation mode, natural for the data processing (in "big digital"), might lead to either low responsiveness or power consumption overheads in control modules of mixed-signal systems. On the one hand, the operating frequency must be sufficiently high to promptly react to changes in analog sensor readings. On the other hand, high clocking frequency can potentially result in wasted clock cycles if the sensors' readings change slowly.

For these reasons, the use of asynchronous logic for dig-

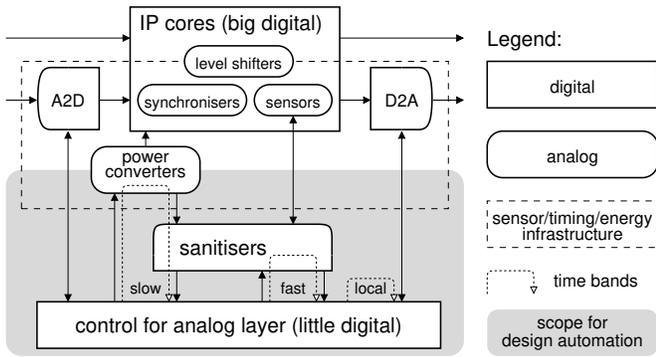


Figure 1: AMS system with "little digital" control.

ital control has the potential to significantly improve the quality of the analog electronics. Indeed, today we are seeing that asynchronous design adds 'holistic value' to the AMS system, going beyond the traditional scope of pure digital domain, namely better power conversion efficiency, lower output ripple, faster response to analog events, reduced inductor size [2]. Asynchronous digital control also has the advantage of improved *electromagnetic compatibility* (EMC) as compared with clocked designs, since it naturally spreads the power spectrum of operation [3]. Finally, asynchronous design, if supported by user-friendly tools, is an attractive option for AMS engineers who accept the nature of 'continuous time' inherent in asynchronous designs.

Currently, analog design with "little digital" is largely done by analog engineers without any formal steps from the specification to netlists. No synthesis tools are in common use, and validation with conventional simulation can take days or longer. Furthermore, analog electronics, e.g. in power regulators, is interfaced closely with "little digital" components, such as sampling latches and groups of logic gates that need to eventually produce activation to gate drives (e.g. PWM and PFM). These parts are prone to glitches and hazards if designed by hand. These glitches can lead to catastrophic system failures. To address these design challenges, researchers are investigating the application of formal methods to the design of AMS circuits and systems. This paper gives a brief introduction to the recent advances in this research domain – they form a baseline for future development of an analog/asynchronous co-design methodology.

## 2. AMS FORMAL VERIFICATION

Traditionally, circuit simulation has been used for AMS system verification. Growing system complexity and technology scaling has led to increased simulation time and probability of failure. As an alternative to simulation-based verification, numerous researchers have been exploring the application of formal verification methods to AMS circuits. Formal verification utilizes exhaustive algorithmic techniques to ensure that a design implementation satisfies the properties given in its specification [4]. These properties are often expressed using temporal logic, while the model for the design can be expressed formally in a variety of ways including automata, Petri nets, etc. Formal verification then proceeds to exhaustively check that the properties are satisfied. In the end, if the formal representation of the system is correct and the set of properties precisely characterize the specification

requirements, then the designer can have a higher confidence of correct operation.

In the digital design space, formal verification has increasingly supplemented standard simulation-based techniques, which has been instrumental in enabling designs consisting of hundreds of millions of devices. Unfortunately, despite recent progress, formal verification of AMS designs is still lagging in industry, as SPICE simulation-based verification remains the prominent verification method for circuit blocks such as operational amplifiers, comparators, ADCs, and VCOs. Although commercial simulators, such as Cadence APS, can exploit multi-core computing environments, AMS verification remains fundamentally the same as when the SPICE simulator was first introduced many decades ago.

The key challenge with the application of digital domain formal methods to the analog domain is the continuous nature of voltage and current state variables. Therefore, formal approaches in the AMS space must deal with a potentially infinite state space. One approach to address this problem is to use theorem proving methods. Theorem proving methods use axioms and inference rules, such as *satisfiability modulo theories* (SMT) solvers, to create mathematical proofs that the specifications are met by a model. While there has been some success [5], these methods require a high amount of user guidance and expertise. Alternatively, the state space can be discretized into a finite set of equivalence classes, and automated state space exploration methods can be applied. The state space methods can be further split into equivalence checking and model checking methods. Equivalence checking methods compare the outputs of two different models under similar input conditions [6], while model checking analyzes the behavior of the design using exhaustive exploration of the entire reachable state space [7]. The key challenge of the state exploration methods is balancing the trade-off between the accuracy of the state space discretization and the number of states needed for the representation.

The remainder of this section focuses on the latest advances in formal verification methods and tools that are relevant to the needs of developing a holistic approach to AMS design advocated in this paper. Comprehensive surveys of formal verification methods can be found in [7] and more recently in [8].

### 2.1 Theorem Proving

METITARSKI is an automatic theorem prover based on a combination of resolution and a decision procedure for the theory of real closed fields. It is designed to prove theorems involving real-valued special functions such as log, exp, sin, cos and sqrt. In particular, it is designed to prove universally quantified inequalities involving such functions. METITARSKI has been lately used to determine the possibility of oscillation of a tunnel diode oscillator and the change in gain due to component tolerances for an operational amplifier [9]. In [10] Narayanan et al. adopted METITARSKI toolset to verify saturation property of an Op-Amp under noise and process variation conditions.

### 2.2 DC Operating Point Analysis

These approaches assume the inputs are held steady and try to find a unique equilibrium point. One such approach is implemented in the FSPICE tool, which solves the multiple DC operating points problem by setting up and solving a *satisfiability* (SAT) problem [11]. Other techniques for DC

analysis can be found in [12, 13, 14], and [15]. The latter, for example, applies evolutionary computing for the detection of multiple equilibrium points.

### 2.3 Equivalence Checking

These approaches attempt to show that two representations of an AMS circuit produce the same response to the same inputs. In [1], a new flow is proposed to enable a top-down design approach for analog components. Analog cells are described using SystemVerilog and compared against their implementation at the transistor level, while digital blocks are validated using existing tools for digital components. This validation method has been used to test analog cells of a single-slope ADC and a serial link receiver.

### 2.4 Symbolic Simulation

Authors of [16] present an extension to symbolic simulation approach utilizing affine arithmetic to allow the representation of control flow and discrete changes. The proposed methodology is used to verify the stability property of a 3rd order  $\Sigma\Delta$  modulator.

Another approach using affine arithmetic is described in [17] tackles the problems of device mismatch and process variation. They reformulate the basic *modified nodal analysis* (MNA) equations in order to include vectors containing parameter expressions based on affine arithmetic. The result of the simulation is not a single trace but a range capturing all potential simulation results obtained by varying a parameter in a certain range. The methodology is applied to an analog bandpass filter and the results are compared to a Monte Carlo simulation. The simulation takes less time, however the algorithm tends to diverge when strong nonlinearities combined occur.

### 2.5 State space guided simulation

In [18], the authors propose a property verification and equivalence checking methodology for analog circuit blocks based on a novel algorithm for formal automatic input stimuli generation. Therewith, it overcomes the incompleteness of transient simulation and the designer-unfriendliness of formal approaches by combining a formal approach and conventional transient circuit simulation. This method is applied to a Sallen-Key biquad lowpass filter with a cut-off frequency of 1000 Hz. Using a property specification for overshoot behavior and an automatic evaluation on the simulation result, complete and therefore formal property verification coverage could be obtained without user-interaction.

### 2.6 Reachability Analysis

The COHO tool performs reachability analysis using state spaces represented as projections of high-dimensional polyhedra onto high-dimensional spaces [19, 20]. This method was successfully used to verify the correctness of a high speed toggle element and an arbiter. Verification of cyclic properties can also be performed by proving the existence of a cyclic invariant.

The PHAVER tool [21] operates on *linear hybrid automata* (LHA) which by definition contain both discrete and continuous components. Similar to differential inclusions, LHA are characterized by a set of states and linear inequalities defining transitions. For the computation of the reachable states, PHAVER uses a polyhedral representation and over-approximation based on affine dynamics.

The SPACEEX tool [22] provides an extensible verification platform for hybrid systems. The tool consists of three main components: an analysis core, a command line program, that analyses the system; a web interface, which provides ability to specify initial states and other analysis parameters, run the analysis core, and visualize the output graphically; and a model editor, a graphical editor for creating models of complex hybrid systems out of nested components. SPACEEX relies on hybrid automata for model description and support functions [23] for state space exploration. This system has been used to model and verify the behavior of several benchmarks [24, 25].

Another work [26] presents a state space analysis method for verifying both the transient and invariant specifications for a PLL using zonotopes by describing reachable sets. The behavioral model of the charge-pump PLL is a hybrid automaton with linear continuous dynamics and uncertain parameters. Furthermore, authors claim that their methodology computes accurate over-approximations of reachable sets for hybrid systems when there are a large number of discrete state transitions. The methodology is applied to the verification of locking time and stability of a 27GHz PLL designed in 32nm CMOS SOI technology. The novel reachability analysis method efficiently provides an upper bound on the worst-case lock time in the presence of random phase error and charge pump current variations.

The LEMA tool [27] includes multiple reachability analysis methods. It includes both an explicit state method originally used for timing verification that leverages *zones* and *difference bound matrices* (DBMs) to represent the continuous state space [28] and implicit state methods that use *binary decision diagrams* (BDDs) and SMT solvers [29]. The properties are specified using the *Language for Analog/Mixed-Signal Properties* (LAMP) [30]. Finally, the AMS circuits are represented using a *labeled Petri net* (LPN) model [29, 28]. Finally, the LEMA tool includes model generation methods to produce these LPN models from simulation data [31, 32, 33]. The LEMA tool has been utilized to verify a number of AMS circuits including DACs, phase interpolators, voltage controlled oscillators, etc.

## 3. ANALOG/ASYNCHRONOUS CO-DESIGN

In order to support the design of "little digital" circuits, we are leveraging the recent advances in AMS formal verification just described and coupling them with the significant advances in asynchronous design over the last years [34]. While most of these advances have been in the domain of pure digital design, such as pipelines, processors, networks-on-chip, and interfaces, there have been some work that considers analog dynamics. For example, much research has been in the areas of digital components exhibiting various analog phenomena, such as metastability, namely arbiters and synchronizers [35]. Also, one of the earlier advancements in the scope of asynchronous design for analog circuits is the design of asynchronous control logic for an ADC [36].

The WORKCRAFT [37] toolset for the design capture, simulation, synthesis, and verification of interpreted graph models is being developed specifically targeting analog/asynchronous co-design. One of the key features is its ability to create and manipulate *signal transition graphs* (STG). STG, a special type of an LPN, provides excellent capabilities for capturing concurrent behavior of asynchronous circuits, as well as necessary pragmatic design

notation [38]. WORKCRAFT provides a convenient mechanism for the verification of constructed STGs and subsequent high-level synthesis, using one of the back-end tools: PETRIFY [39] or MPSAT [40].

In [41], this software is used to create an asynchronous control for a buck converter, based on its timing diagram specification. The resulting circuit proves to be more power efficient, as well as containing fewer complex gates than an equivalent synchronous design. Moreover, since the input-output latency of the new circuit depends only on the delay of a single gate, the resulting responsiveness is also improved. Other examples of using STGs and WORKCRAFT in this domain include the design of the control of a capacitor switching in novel power regulation methods for energy-harvesting systems [3] and controllers for switched capacitor converters [42].

New specification formats for asynchronous controllers of power converters, specification styles (such as concepts [43]) and variety of compositional techniques to capture internal and external time-sequencing based on tokens have also been developed. Special components for analog-to-asynchronous interfaces, such as WAIT [41] and WAITX [44] have been designed and formally verified to be free from hazards. New visualization methods have been implemented in WORKCRAFT for the behavior of asynchronous logic using timing diagrams, to facilitate easier comprehension of asynchronous circuits by analog designers. Finally, the basics of behavioral mining and formal verification of analogue circuits using the LEMA tool described in the previous section have been explored in [45].

These developments can be illustrated in a simple AMS circuit design example. A basic power regulator consisting of an analog block and a digital controller, as shown in the schematic in Figure 2. The controller determines the state of NMOS and PMOS transistors in response to under-voltage (UV) and over-current (OC) conditions. These conditions are detected and signaled by special sensors, implemented as comparators in combination with buffering latches. The formal specification of digital control is given in STG format in Figure 3. This STG can be synthesized into a logic implementation using WORKCRAFT.

The described specification of the buck controller offers possibilities for optimization, depending on the behavior of analog environment. One possibility is the elimination of concurrency between the acknowledgment and over-current signals, provided transistors always switch faster than coil current ramps up. Another opportunity for optimization is to completely remove one charging scenario, depending on the output capacitance parameter. These opportunities can be extracted from the simulation traces of the analog part using LEMA in the form of LPNs and converted to the STG format for subsequent resynthesis. The details of this procedure can be found in [45].

Further advancements are needed to turn this approach into versatile and robust design flow applicable for wider academic and industrial use. The next section describes some of the remaining challenges.

## 4. DISCUSSION

While there has been much progress, there are gaps that require more research to make formal methods a reality for industrial-scale designs:

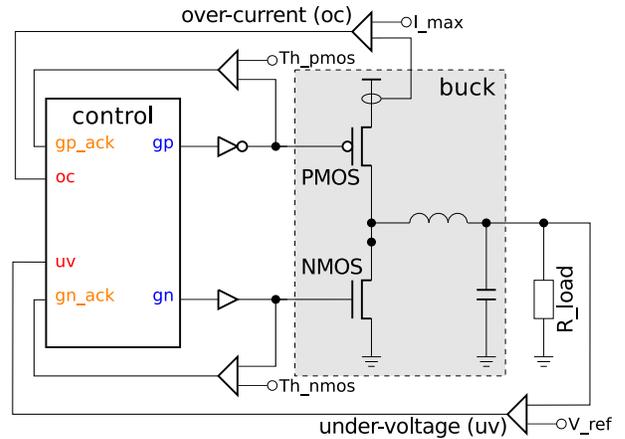


Figure 2: Buck schematic.

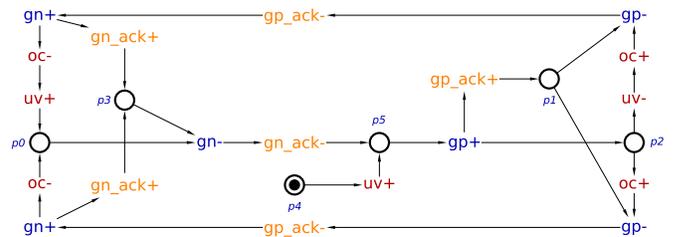


Figure 3: STG specification of buck control.

- Integration of formal modeling and verification into the analog-asynchronous co-design process. Currently designers of analog parts perform validation separately (mostly through simulations) from the asynchronous domain (asynchronous circuits are often correct by construction, but this rapidly changes as we go into deep submicron). As a result, designers spend days if not weeks checking all relevant scenarios in which analog blocks are stimulated by asynchronous logic, and vice versa, tuning asynchronous specifications to the required number of modes for the analog parts. The problem in this aspect concerns analog behavioral mining, extensive visualization, and composition of multiple scenarios into a complete synthesizable specification of the asynchronous control. For example, the above buck design example can be made multi-phase and involve checking additional conditions such as zero-crossing.
- AMS systems with digital control have a variety of feedbacks with different "time bands" and corresponding timing requirements: (a) analog-dynamics signaling, such as over-current or under-voltage events (100ns-10 $\mu$ s); (b) control activation feedbacks, such as acknowledging gate-drive signals (1-10ns) — both (a) and (b) can sometimes overlap; (c) internal asynchronous-fabric events, such as signals between logic gates and handshakes (10-100ps). As a result, clocked digital logic does not offer the best possible response time, as well as it does not scale well into deep-submicron, where gate and wire delay ratios and PVT variability parameters are changed, and hence the notion of hazard-freedom changes (more care

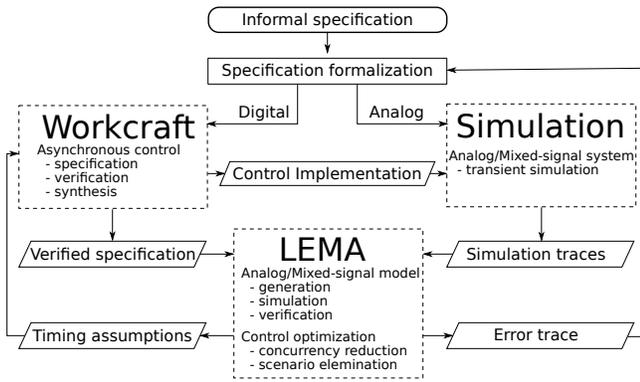


Figure 4: LEMA-Workcraft workflow.

needed about isochronic forks, non-zero delay input inverters, relative timing constraints, etc [38]). The issue of asynchronous-analog interface and synchronization is also underdeveloped.

To summarize, these two groups of challenges call for new CAD tools to act on constructing AMS circuits holistically and cooperatively. These tools will need significant common grounds in terms of modeling formalisms, variable representations and algorithmic techniques. At the same time, the more traditional radical view about formal methods, to fully replace simulations, is not realistic in practice. Formal methods should complement simulations and hence new types of tools, acting in-between, such as automated model generators, are required. To be more specific, one possible candidate of an analog/asynchronous co-design flow is presented in Figure 4. It puts emphasis on model generation (from simulation), sharing of related model representations (LPNs in LEMA and STGs in WORKCRAFT), between analog and digital parts, and iterative specification synthesis and verification. The details of the flow and examples from the design of power converters can be found in our recent paper [45].

## Acknowledgments

This research was supported by EPSRC grant "A4A: Asynchronous design for Analogue electronics" (EP/L025507/1) and National Science Foundation Grant No. CCF-1117515.

## 5. REFERENCES

- [1] B. C. Lim, J. E. Jang, J. Mao, J. Kim, and M. Horowitz. Digital analog design: Enabling mixed-signal system validation. *IEEE Design & Test*, 32(1):44–52, 2015.
- [2] D. Sokolov, V. Dubikhin, V. Khomenko, D. Lloyd, A. Mokhov, and A. Yakovlev. Benefits of asynchronous control for analog electronics: multiphase buck case study. In *Proc. Design, Automation & Test in Europe (DATE)*, 2017.
- [3] D. Shang, X. Zhang, F. Xia, and A. Yakovlev. Asynchronous design for new on-chip wide dynamic range power electronics. In *Proc. Design, Automation & Test in Europe (DATE)*, 2014.
- [4] D. Perry and H. Foster. *Applied formal verification: For digital circuit design*. Electronic Engineering. McGraw-Hill, 2005.
- [5] Y. Peng and M. Greenstreet. *Integrating SMT with Theorem Proving for Analog/Mixed-Signal Circuit Verification*, pages 310–326. 2015.
- [6] M. Horowitz, M. Jeeradit, F. Lau, S. Liao, B. Lim, and J. Mao. Fortifying analog models with equivalence checking and coverage analysis. In *Design Automation Conference*, pages 425–430, 2010.
- [7] M. H. Zaki, S. Tahar, and G. Bois. Formal verification of analog and mixed signal designs: A survey. *Microelectron. J.*, 39(12):1395–1404, 2008.
- [8] I. Syranidis. *State space characterisation under parameter variations and application to bifurcation based voltage sensing*. PhD thesis, Newcastle University, 2014. Available as tech. rep. No. 185 at <http://async.org.uk/tech-reports/>.
- [9] W. Denman, B. Akbarpour, S. Tahar, M. H. Zaki, and L. C. Paulson. Formal verification of analog designs using MetiTarski. In *Proc. Formal Methods in Computer-Aided Design*, pages 93–100, 2009.
- [10] R. Narayanan, B. Akbarpour, M. Zaki, S. Tahar, and L. Paulso. Formal verification of analog circuits in the presence of noise and process variation. In *Proc. Design, Automation & Test in Europe (DATE)*, pages 1309–1312, 2010.
- [11] S. K. Tiwary, A. Gupta, J. R. Phillips, C. Pinello, and R. Zlatanovici. First steps towards SAT-based formal analog verification. In *Proc. International Conference on Computer-Aided Design (ICCAD)*, pages 1–8, 2009.
- [12] M. R. Greenstreet and S. Yang. Verifying start-up conditions for a ring oscillator. In *Proc. ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pages 201–206, 2008.
- [13] M. H. Zaki, I. M Mitchell, and M. R. Greenstreet. DC operating point analysis – a formal approach. In *Proc. Frontiers in Analog CAD (FAC)*, 2009.
- [14] M. Tadeusiewicz and S. Halgas. A method for finding multiple DC operating points of short channel CMOS circuits. *Circuits, Systems, and Signal Processing*, 32(5):2457–2468, 2013.
- [15] M. Zwolinski and D. A. Crutchley. Using evolutionary and hybrid algorithms for DC operating point analysis of nonlinear circuits. In *Proc. of Congress on Evolutionary Computation (CEC)*, pages 753–758, 2002.
- [16] C. Radojicic and C. Grimm. Formal verification of mixed-signal designs using extended affine arithmetic. In *Proc. PhD Research in Microelectronics and Electronics (PRIME)*, 2016.
- [17] D. Grabowski, C. Grimm, and E. Barke. Semi-symbolic modeling and simulation of circuits and systems. In *Proc. International Symposium on Circuits and Systems (ISCAS)*, pages 983–986, 2006.
- [18] L. Steinhorst, S. and Hedrich. Improving verification coverage of analog circuit blocks by state space-guided transient simulation. In *Proc. International Symposium on Circuits and Systems (ISCAS)*, pages 645–648, 2010.
- [19] C. Yan and M. R. Greenstreet. Circuit level verification of a high-speed toggle. In *Proc. Formal Methods in Computer Aided Design (FMCAD)*, pages 199–206, 2007.
- [20] C. Yan and M. R. Greenstreet. Formal verification of

- an arbiter circuit. In *Proc. IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 165–175, 2010.
- [21] G. Frehse, B. H. Krogh, and R. A. Rutenbar. Verifying analog oscillator circuits using forward/backward abstraction refinement. In *Proc. Design, Automation & Test in Europe (DATE)*, pages 257–262, 2006.
- [22] G. Frehse, C. Le Guernic, A. Donze, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SPACEEX: Scalable verification of hybrid systems. In G. Gopalakrishnan and S. Qadeer, editors, *Computer Aided Verification*, volume 6806 of *LNCS*, pages 379–395. Springer, 2011.
- [23] C. Le Guernic and A. Girard. Reachability analysis of hybrid systems using support functions. In A. Bouajjani and O. Maler, editors, *Computer Aided Verification*, volume 5643 of *LNCS*, pages 540–554. Springer, 2009.
- [24] L. V. Nguyen and T. T. Johnson. Benchmark: DC-to-DC switched-mode power converters (buck converters, boost converters, and buck-boost converters). In G. Frehse and M. Althoff, editors, *Proc. International Workshop on Applied Verification for Continuous and Hybrid Systems (ARCH)*, volume 34 of *EPiC Series in Computing*, pages 19–24, 2015.
- [25] I. B. Makhlof and S. Kowalewski. Networked cooperative platoon of vehicles for testing methods and verification tools. In G. Frehse and M. Althoff, editors, *Proc. Int. Workshop on Applied Verification for Continuous and Hybrid Systems (ARCH)*, volume 34 of *EPiC Ser. in Comp.*, pages 37–42, 2015.
- [26] M. Althoff, A. Rajhans, B. H. Krogh, S. Yaldiz, X. Li, and L. Pileggi. Formal verification of phase-locked loops using reachability analysis and continuization. *Commun. ACM*, 56(10):97–104, 2013.
- [27] A. N. Fisher, S. Batchu, K. Jones, D. Kulkarni, S. Little, D. Walter, and C. J. Myers. LEMA: A tool for the formal verification of digitally-intensive analog/mixed-signal circuits. In *Proc. International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1017–1020, 2014.
- [28] S. Little, D. Walter, C. J. Myers, R. Thacker, S. Batchu, and T. Yoneda. Verification of analog/mixed-signal circuits using labeled hybrid Petri nets. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(4):617–630, 2011.
- [29] D. Walter, S. Little, C. J. Myers, N. Seegmiller, and T. Yoneda. Verification of analog/mixed-signal circuits using symbolic methods. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(12):2223–2235, 2008.
- [30] A. Fisher, D. Kulkarni, and C. Myers. A new assertion property language for analog/mixed-signal circuits. In M.-M. Louerat and T. Maehne, editors, *Languages, Design Methods, and Tools for Electronic System Design*, volume 311 of *Lecture Notes in Electrical Engineering*, pages 45–65. Springer International Publishing, 2015.
- [31] S. Little, D. Walter, K. Jones, C. J. Myers, and A. Sen. Analog/mixed-signal circuit verification using models generated from simulation traces. *Int. J. Found. Comput. Sci.*, 21(2):191–210, 2010.
- [32] S. Batchu. Automatic extraction of behavioral models from simulations of analog/mixed-signal (AMS) circuits. Master’s thesis, 2010.
- [33] D. Kulkarni. Improved model generation and property specification for analog/mixed-signal circuits. Master’s thesis, 2013.
- [34] A. Yakovlev, P. Vivet, and M. Renaudin. Advances in asynchronous logic: From principles to GALS & NoC, recent industry applications and commercial CAD tools. In *Proc. Design, Automation & Test in Europe (DATE)*, pages 1715–1724, 2013.
- [35] D. J. Kinniment. *Synchronization and arbitration in digital systems*. Wiley, 2008.
- [36] D. J. Kinniment, B. Gao, A. Yakovlev, and F. Xia. Towards asynchronous A-D conversion. In *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC)*, pages 206–215, 1998.
- [37] D. Sokolov, V. Khomenko, and A. Mokhov. Workcraft: Ten years later. In A. Mokhov, editor, *This asynchronous world. Essays dedicated to Alex Yakovlev on the occasion of his 60th birthday*. 2016. Available online <http://async.org.uk/ay-festschrift/paper25-Alex-Festschrift.pdf>.
- [38] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavango, and A. Yakovlev. *Logic synthesis of asynchronous controllers and interfaces*. Springer-Verlag Berlin Heidelberg, 2002.
- [39] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavango, and A. Yakovlev. Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers. *IEICE Transactions on Information and Systems*, 80(3):315–325, 1997.
- [40] V. Khomenko. A usable reachability analyser. Technical report, Newcastle University, 2009.
- [41] D. Sokolov, V. Khomenko, A. Mokhov, A. Yakovlev, and D. Lloyd. Design and verification of speed-independent multiphase buck controller. In *Proc. IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 29–36, 2015.
- [42] S. Mileiko, A. Kushnerov, D. Sokolov, and A. Yakovlev. Self-timed control of two-phase switched capacitor converters. In *IEEE International Conference on the Science of Electrical Engineering (ICSEE)*, pages 1–4, 2016.
- [43] J. Beaumont, A. Mokhov, D. Sokolov, and A. Yakovlev. Compositional design of asynchronous circuits from behavioural concepts. In *ACM/IEEE Int. Conf. on Formal Methods and Models for Codesign (MEMOCODE)*, pages 118–127, 2015.
- [44] V. Khomenko, D. Sokolov, A. Mokhov, and A. Yakovlev. WAITX: An arbiter for non-persistent signals. In *Proc. IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, 2017.
- [45] V. Dubikhin, D. Sokolov, A. Yakovlev, and C. J. Myers. Design of mixed-signal systems with asynchronous control. *IEEE Design & Test*, 33(5):44–55, 2016.