Formal Verification of Analog and Mixed Signal Designs: Survey and Comparison

Mohamed H. Zaki[1],Sofiène Tahar[1]and Guy Bois[2]
[1]Dept. of Electrical & Computer Engineering, Concordia University
1455 de Maisonneuve W., Montreal, Quebec, H3G 1M8, Canada
[2]Genie Informatique, Ecole Polytechnique de Montreal
5255 Avenue Decelles, Montreal, Quebec, H3T 2B1, Canada
Email: {mzaki,tahar}@ece.concordia.ca, guy.bois@polymtl.ca

***Abstract*— Analog and mixed signal (AMS) circuits are important integrated circuits that are usually needed at the interface between the electronic system and the real world. In contrast to digital designs, verification of AMS systems is a challenging task that requires lots of expertise and deep understanding of their behavior. Researchers started lately studying the applicability of formal methods for the verification of AMS systems as a way to tackle the limitations of conventional verification methods like simulation. This paper surveys research activities in the formal verification of AMS designs as well as compares the different proposed approaches.**

I. Introduction

Analog and mixed signal (AMS) circuits are important integrated circuits that are usually needed at the interface between the electronic system and the real world. Among the important functionality of the analog circuits are processing analog signal and converting between analog and digital data representation. In contrast to digital designs, the verification of AMS circuits is a challenging task that requires lots of expertise and deep understanding of their behavior. The functionality of analog circuits is defined directly in terms of continuous electrical quantities and is usually sensitive to environment factors like signal noise, current leakage. Traditionally, simulation was used where the evaluation of the results is often done manually in an informal fashion and the search of the state space is not complete. Simulation based techniques were then complemented by symbolic techniques where the effect of parameters variations on the system behavior is analyzed. Although successful, challenging problems like non-linear effects make these techniques only suitable for simple designs. Researchers started lately studying the applicability of formal methods for the verification of AMS systems as a way to tackle the limitations of conventional techniques. In this paper, we provide a survey and comparison of of the research activities in the field of AMS design formal verification. The remaining of the paper is as follow. Equivalence checking methods applied to AMS designs are surveyed in Section II, followed by model checking and reachability techniques in Section III, run-time verification in Section IV and deductive methods in Section V, before concluding with a discussion in Section VI.

II. Equivalence Checking

Equivalence checking is a problem where we are given two systems models and are asked whether these systems are equivalent with respect to some notion of equivalence. For instance, in [1], the authors proposed applying equivalence checking between transfer function forms of the implementation and specification by transforming the transfer functions to the discrete Z-domain where they can be represented in terms of digital components and encoded into finite state machine (FSM) representation like Binary Decision Diagrams (BDDs). Verification of the digitized circuit implementation against the digitized transfer function consists in proving that the implementation FSM mimics the behavior of the specification FSM for all acceptable inputs. The methodology is only practical for linearized systems as transfer function generation for non linear circuits is very difficult in general. The discretization of the behaviors raises issues like the error analysis which must be accounted for. Also, the adequacy of such methodology to large circuits where the state space explosion arise quickly even with BDD encoding is a major issue. Realizing the coefficient of a transfer function exactly using actual components and devices is not always possible as tolerance region around nominal characteristic must be taken into account. This latter issue was investigated in [3], where the authors present an equivalence checking approach for linear analog circuits to prove that an actual circuit fulfills a specification in a given frequency interval for all parameter variations. Linear analog circuits can be described by transfer functions extracted from the netlist by symbolic analysis methods, resulting in a parameterized description of the circuit behavior. The main idea of the procedure is to compare by inclusion the value sets of the transfer functions of specification and implementation. To ensure soundness, the authors chose an over-approximation for the implementation transfer function while an under-approximation is chosen for the specification transfer function. In [2], the authors proposed an equivalence checking approach based on qualitative comparison between two representations of the nonlinear analog system described by sampling the state space and computing the differences in the vector fields for two systems. Direct comparison of vector fields is usually not possible. Therefore, the authors propose applying non linear transformations on the sample state spaces to make the comparison possible. Finding the correct transformations is a non trivial task and automation at this level is not always possible specially with the increasing complexity of the designs. In addition, no mention of soundness of the algorithm is discussed. In [4], Salem propose a technique for verifying VHDL-AMS designs

1-4244-0417-7/06/$20.00 ©2006 IEEE

based on combining equivalence checking, rewriting systems and simulation into a verification environment. The verification methodology consists of partitioning the specification and implementation codes into digital, analog and data converter components. Digital components are verified using classical equivalence checking while analog specification and implementation are simplified using mathematical terms rewriting rules and the outputs are fed to comparators to be verified by using simulation. This syntactic method can only be applied to simple designs where rewriting techniques can be easily applied. While the presented methodology is practically using existing techniques, it ignores the coupling between the analog and digital parts. In addition, it is not clear to which analog classes these techniques can be applied. While diverse, discussed ideas are limited to specific classes of circuits. With the exception of the work in [3], approximation and discretization techniques affect the soundness of the methodology. Error analysis and unified equivalence theory for AMS are important candidate for the success of such verification methodology. Table I draw a brief comparison among the above mentioned projects.

III. Model Checking and Reachability analysis

Model checking and reachability analysis of AMS designs have the potential of validating designs over a range of parameters and for all possible input signals all at once such that none of them drives the system into a bad state. An important issue is the solution of the system differential equations over a range of initial states.

Several methods for approximating reachable sets for continuous dynamics have been proposed. In [5], the authors tried to construct a finite-state discrete abstraction of electronic circuits by partitioning the continuous state space into fixed grid hypercubes and computing reachability relations between these cubes using numerical techniques. Language containment method is then applied between specification and implementation automata. In [6], the authors tried to overcome the expensive computational method in [5], by using discretization and projection techniques of the state space in what is called projectahedra, reducing the dimension of the state space. While the approach is less precise due to projection, it is still sound. Variant approaches of polyhedral based analysis was adapted in [7], [8]. In [7], the authors used d/dt for the verification of analog systems described with differential algebraic equations (DAEs). In order to tackle the state explosion problem due to the exhaustive analysis, they proposed in addition using techniques from optimal control (i.e., hybrid constrained optimization) in order to find bounds of the reachability. The idea is to formulate bounded horizon reachability as a hybrid constrained optimization problem that can be solved by techniques such as mixed-integer linear programming. In [8], the authors used the Checkmate tool for the verification of AMS designs. The tool is based on constructing abstractions of the continuous dynamics, using flow pipes approximations, which are sequences of polyhedra that follow the natural contour of the vector field. Therefore, the state space is partitioned along the waveforms that the system can generate for the given set of initial conditions and there is no need to discretize the entire state space. Checkmate specifications to be verified can be provided as ACTL temporal logic formulas. For the verification of systems like $\Delta - \Sigma$ modulator, which is described by discrete time components, a modification of the tool to support discrete time analysis was proposed. In [9], the authors proposed to use an automatic state space subdivision method, by discretizing the whole continuous state space into variables sized regions (hyperboxes) where each of these hyperboxes represents a homogeneous part of the state space and it is treated as a discrete state of the simplified system. Some kind of estimation techniques are then proposed to describe possible transitions between partitions under the condition of retaining the essential nonlinear behavior of the analog system. Different criteria take care of the resulting error during discretization and try to automatically minimize the error by choosing a suitable subdivision of the state space. The discretized state space is then encoded and CTL based model checking is applied. The proposed approach was implemented in a tool called $Amcheck$. In [10], the authors proposed extending their previous work for the verification of time constraints like rise and fall time of analog signals, slew rate of operational amplifier. The presented extensions are based on algorithms taking delay times into account during the discretization of the state space. In [11], the authors used the extension of petri nets for real time and hybrid systems (Timed Hybrid Petri Nets (THPN)) for modeling and verification of AMS designs. They present a method to translate differential equation models into THPNs and applied a conservative zone based algorithm for reachability analysis on the generated model. In [12], the authors compared verification using their methodology in [11] against simulation results, by examining the effect of variable delays caused by parasitic capacitances and interconnect capacitances on the performance and functionality of the circuits. In [13], the authors developed a bounded model checking prototype tool (*Property-Checker*) for the verification of the quasi-static behavior of AMS designs. The basic idea is based on validity checking of first order formulas over a finite interval of time. The prototype accepts as inputs the design as an automaton described in XML language and the formal specification in the form of properties both internally converted to first order logic formulas and the Property-Checker returns whether a given property holds for that mixed-signal circuit or not. In contrast to other approaches, the one presented in [13] trades-off accuracy with efficiency by basing the analysis on rational numbers rather than real numbers. Unlike equivalence checking, a common theme between above work (except [13]), is the overapproximation of the reachability analysis of the designs using computational techniques. Although expensive they guarantee the verification results. Other successful trends in the related hybrid system theory that can be explored in the model checking of AMS is the use of sound techniques like constraint solving techniques and logic based methods like quantifier elimination. In addition model checking in the frequency domain has yet to be developed. Tables II, III gives a comparison between the work presented in this section.

IV. Run-Time Verification

Model checking of AMS circuits is computationally expensive and therefore suffers from the state-space explosion

TABLE I
EQUIVALENCE CHECKING

	[1]	[2]	[3]	[4]
Type of Systems	Linear	Non linear	Linear	Non linear AMS
Models	Transfer function	ODE - DAE	Transfer function	ODE - DAE
Analysis Regions	Transient response	Near operating point transient analysis	Near operating point	Functional analysis
Analysis Domain	Frequency	Time	Frequency	Time
Techniques and Analysis	OBDDs comparisons	Qualitative analysis	Interval analysis	Rewriting, SAT simulation
Tools	N/A	MAPLE	MAPLE	M-CHECK
Case Studies	Low Pass filter	CMOS inverter, Opamp	Band pass filter, opamp	D/A converter

TABLE II
MODEL CHECKING

Project	[5]	[6]	[8]	[7]
Type of Systems	Nonlinear	Non linear	Non Linear	Non linear
Models	ODE	ODE	HA/ ODE - DAE	HA/ODE -DAE
Analysis Region	No restriction	No restriction	No restriction	No restriction
Techniques and Analysis	Simulation lang. containment	Projection numerical appro.	Numerical approx.	Numerical approx., MILP
State Space partitions	Fixed size hyperCubes	Projectaherda	Convex polyhedra	Orthogonal polyhedra
Tools	COSPAN	Matlab/ Coho	Checkmate	d/dt
Case Studies	Interlock Circuits	Van der Pool Oscillator	Tunnel Diode $\Delta - \Sigma$ mod	Low Pass Filter $\Delta - \Sigma$ mod

TABLE III
MODEL CHECKING (CONT')

Project	[9], [10]	[11], [12]	[13]
Type of Systems	Non linear	Non linear	AMS
Models	ODE, DAE	THPN/ODE	piecewise linear automaton
Analysis Regions	No restriction	No restriction	Steady State
Techniques and Analysis	Numerical analysis	Numerical approx.	Bounded MC
State Space partitions	HyperCubes	Convex polygons	
Temporal Logic	CTL-AT	ACTL	FOL
Tools	Amcheck	ATACS	CVCL, Property checker
Case Studies	Schmidt trigger, Opamp , VCO	Tunnel diode PLL	Sequential circuit

problem that makes exhaustive verification very hard and limitations in memory and/or time resources. In order to cope with these problems, run-time verification (logical monitoring) methods were developed where no computational model is needed prior to the verification, avoiding state space explosion. By employing logical monitors, an efficient analysis of the results is achieved, avoiding exhaustive inspection, by testing whether a given behavior satisfies a property. This process can be performed in two different fashions: *Offline monitoring* starts after the whole sequence is given. *Online monitoring* is interleaved with the process of reading the sequence and is similar to the way the sequence is read by an automaton. In [14], Maler *et al.*, proposed an offline methodology for monitoring the simulation of continuous signals described by differential equations. The work is based on extending the PSL logic to support monitoring analog signals, by defining the syntax and semantics of metric timed linear temporal logic (MTL) and extended it with predicate over real to form the signal temporal logic (STL). No proposed test case generation technique is proposed. As the methodology is still in progress, it is only suitable for monitoring single trajectories. The main practical extension is to monitor flows instead. A similar idea was proposed in [16], where the authors use an extended temporal logic, AnaCTL (CTL for analog circuit verification), for monitoring the transient behavior of non-linear analog circuits. The transient response of a circuit under all possible input waveforms is represented as an FSM created by means of repeated SPICE simulations, bounding and discretizing the continuous state space of an analog circuit. Exhaustive simulation is again a drawback as the created FSM is not guaranteed to cover the total transient behavior leading to soundness problem. An online monitoring technique was proposed in [15], where the authors used linear hybrid automata as template monitors for time domain features of oscillatory behavior, such as bounds on signal amplitude and jitter For the automata with an error state, the reachability computation can be stopped as soon as this state is reachable. The monitors are used within the PHAver tool where nonlinear circuit equations are modelled with piecewise affine differential inclusions, where inaccuracies in approximation can be compensated by overlapping the sections. Although appealing, several issues must be addressed to make run-time verification useful. Among these issues are monitoring properties in frequency domain, synthesizing monitors from the specification as well as developing testcase generation approaches to guide the verification. Table IV summarizes the main characteristics of the described projects.

V. DEDUCTIVE METHODS

Theorem provers are formal systems that were developed to prove design properties using formal deduction based on a set of inference rules. Even though these deductive methods are not constrained by any decidability frontiers, their application requires expertise and significant human intervention which makes their application to complex systems very difficult. A lot of research has been focusing on extending theorem provers with decision procedures for verification assistance

TABLE IV
RUN-TIME VERIFICATION

	[14]	[16]	[15]
Type of Systems	Non linear	Non linear	Piecewise affine
Models	ODE	ODE	ODE
Monitors	STL	Ana CTL	Linear HA
Monitoring Type	offline	offline	online
Analysis Regions	No restriction	Transient response	No restriction
Analysis Domain	time dom.	time dom.	time dom.
Techniques	Numerical Simulation	Numerical Simulation	Numerical Approx.
Tools	Matlab	Spice simulator	PHAver
Case Studies	Sine wave signals	VCO Opamp	Oscillator Circuits

and automation as well as formalizing important theories like the real analysis theory. Some primary efforts on verifying AMS systems using theorem provers started recently. In [17], Gosh and Vermin used the PVS theorem prover to formally prove the equivalence of the piecewise-linear models of synthesized analog designs to their user given behavioral specifications in a VHDL-AMS subset relative to the DC and low frequency behaviors properties. One drawback of this work is that the linearization of nonlinear behavior is ad-hoc and no formal error analysis is given. The ideas presented are a good starting point for a methodology to verify analog designs, yet important extensions should be studied more, especially related to AC analysis. Similar but more elaborative research was done in [18], where the author was mainly interested in verifying the DC behavior of the circuits. The approach relies upon specifying the behaviors of analog components (such as transistors) by conservative approximation techniques based on piecewise-linear predicates on voltages and currents. Theorem proving was initially used to check for the implication relation between the implementation and the specification [19]. In order to automate the verification process, the author proposed afterwards using constraint based techniques instead and described a decision procedure to achieve his goal [18]. The deductive based verification of AMS systems is still premature and facing a long line of challenges including analysis in time and frequency domains, AC analysis, error analysis. Also, the formalization of the necessary theories needed in the analysis is still underdeveloped and efforts have to be done to integrate theorem proving within the mainstream verification tools. In table V, we highlight some main points of the work surveyed.

TABLE V
THEOREM PROVING

	[17]	[19], [18]
Type of systems	Piecewise linear	Piecewise linear
Modelling	set of predicates over real	set of predicates over real
Domain Analysis	Time	Time
Error Analysis	No	No
Tool	PVS	N / A
Case Studies	Adder, Telephone Receiver	TTL

VI. CONCLUSION

In this paper, we summarized the research activities in the application of formal methods for the verification of AMS systems. We tried to be as exhaustive as possible into collecting the different related work as well as giving a comparison among the research presented. The lack of extensive research is due to the complexity of the verification process and the challenging problems mostly inherited from the hybrid systems. The formal verification of AMS design is a relatively young research field and still under-developed, which is a bad and a good sign at the same time. It is bad because this shows the lack of interest and this can be deduced due to the different scientific background between AMS engineers, control engineers and computer scientists. However, this can motivate interdisciplinary collaborations. The good news is that room for exploration is yet wide open. Among the interesting directions is developing AMS theory with high order logic, process algebraic languages for AMS designs, developing specification logics for frequency properties among others. Another important direction is incorporating formal verification within the design flow, hence complementing simulation, testing and symbolic analysis.

REFERENCES

[1] A. Balivada, Y. V. Hoskote, J. A. Abraham: Verification of transient response of linear analog circuits. IEEE VLSI Test Symposium, 1995: 42-47.

[2] L. Hedrich and E. Barke, A Formal Approach to Nonlinear Analog Circuit Verification. IEEE/ACM Intl. Conference on CAD, November 1995: 123-127.

[3] L. Hedrich and E. Barke, A Formal Approach to Verification of Linear Analog Circuits with Parameter Tolerances. IEEE/ ACM Design, Automation and Test in Europe, 1998: 649-654.

[4] A. Salem. Semi-formal verification of VHDL-AMS descriptors. IEEE Int. Symposium on Circuits and Systems, 2002.

[5] R.P. Kurshan and K.L. McMillan. Analysis of digital circuits through symbolic reduction. IEEE Trans. on Computer-Aided Design 10:13501371, 1991.

[6] Mark R. Greenstreet, Ian Mitchell. Reachability Analysis Using Polygonal Projections. HSCC, LNCS 1569, springer: 103-116,1999.
HSCC, LNCS 1386, springer: 159-174, 1998.

[7] T. Dang, A. Donze, O.Maler: Verification of analog and mixed-signal circuits using hybrid system techniques. In: Formal Methods in Computer-Aided Design, LNCS 3312, springer: 14-17, 2004.

[8] S.Gupta, B.H. Krogh, R.A. Rutenbar: Towards formal verification of analog designs, IEEE/ACM International Conference on Computer Aided Design: 210 - 217, 2004.

[9] W. Hartong, *et al.*: "Formal Verification for Nonlinear Analog Systems: Approaches to Model and Equivalence Checking," Advanced Formal Verification, Kluwer: 205-245, 2004, .

[10] D. Grabowski, *et al.*: Time Constrained Verification of Analog Circuits using Model-Checking Algorithms,Workshop on Formal Verification of Analog Circuits, 2005.

[11] S. Little, *et al.*: Verification of Analog and Mixed-Signal Circuits Using Timed Hybrid Petri Nets. ATVA, LNCS 3299, springer: 426-440, 2004.

[12] C. Myers, *et al.*: The Case for Analog Circuit Verification. Workshop on Formal Verification of Analog Circuits, 2005.

[13] M. Freibothe, *et al.*: Formal Verification of the Quasi-Static Behavior of Mixed-Signal Circuits by Property Checking, Workshop on Formal Verification of Analog Circuits, 2005

[14] O. Maler, D. Nickovic: Monitoring Temporal Properties of Continuous Signals. FORMATS/FTRTFT, LNCS 3253, Springer: 152-166, 2004.

[15] G. Frehse, B. Krogh, R. Rutenbar, O. Maler Time Domain Verification of Oscillator Circuit Properties, Workshop on Formal Verification of Analog Circuits, 2005

[16] T. R. Dastidar, P. P. Chakrabarti: A Verification System for Transient Response of Analog Circuits Using Model Checking. VLSI Design: 195-200, 2005.

[17] A. Ghosh and R. Vemuri: Formal Verification of Synthesized Analog Circuits. In ACM/IEEE Int. Conference on Computer Design: 40-45, 1999.

[18] K. Hanna: Reasoning About Analog-Level Implementations of Digital Systems. Formal Methods in System Design 16(2): 127-158,2000.

[19] K. Hanna: Reasoning about Real Circuits. TPHOLs, LNCS 859, Springer: 235-253, 1994.