Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 153 (2006) 3–7
www.elsevier.com/locate/entcs

Analog Circuit Verification: a State of an Art

Oded Maler[1,2]

CNRS-VERIMAG
Centre Equation, 2 av. de Vignate
38610 Gières, France

Abstract

Extending formal verification methodology toward analog circuits is a very challenging task that will occupy researchers for some time. To put this challenge in context we sketch some of the history of digital circuit verification as well as more recent attempts to adapt it to continnuous and hybrid systems.

Keywords: analog circuit verification, hardware verification

1 Introduction

We start by situating the role of formal verification in the design process of circuits. Property-based system design means that a system is characterized by a set of properties it should satisfy. These properties specify, in a formal language, which traces of I/O behaviors the system may exhibit while interacting with its external environment. There are basically two approaches for validating a system with respect to a given property. Both of them are based on transforming the property into a *property monitor*, a mechanism that checks whether a given behavior (sequence of I/O events) satisfies the property. This monitor can be viewed either as an automaton accepting exactly the set of satisfying behaviors or as a procedure working recursively both on the length of the sequence and on the syntactic structure of the property.

[1] This work was partially supported by the European Community project IST-2003-507219 PROSYD (Property-based System Design).
[2] Email: Oded.Maler@imag.fr

1571-0661/$ – see front matter
doi:10.1016/j.entcs.2006.02.020

In *simulation/testing* (also known as "dynamic verification" in the circuit jargo) a model of the system is used to generate simulation traces, and each of those is checked by the monitor. If one of the traces violates the property the system is incorrect. However since the system is to be exposed to a potentially infinite (or prohibitively large) number of inputs, it is impossible to complete this procedure for all possible inputs. A large effort in this domain is about the systematic generation of test-cases that somehow "cover" all interesting classes of behaviors.

The goal of *algorithmic verification* is more ambitious. The transition graph of the system is explored in order to show that *all* the sequences it can generate are accepted by the property monitor. A major problem here is that of state-explosion as the number of states of the system is exponential in the number of state-variables (memory holding elements, in the case of digital circuits). Much of the current research in verification is about finding clever ways to cope with this situation and about developing the methodological aspects of property-based system design. This workshop is concerned with the extension of this methodology to analog and mixed-signal systems.

2 Historical Context

To assess the contrast between the respective situations in the digital and analog domains it is worth recalling the evolution of (digital) formal verification up to the present:

- Early work on program verification (1965-1977).
- Introduction of Temporal Logic as a formalism for specifying properties of reactive systems (Pnueli 1977).
- Work on deductive (partly-manual) verification for TL (1977-present).
- First model-checking algorithms for fully automatic verification (Queille and Sifakis 1981, Clarke and Emerson 1981).
- First workshop on computer-aided verification (Grenoble, 1989).
- First symbolic model-checker that could treat systems with state-space too large to be enumerated (McMillan 1992).
- The development of industrial-strength verification tools (1993-present)
- The Intel bug and the proliferation of formal verification into the semiconductor industry (1995).
- The development of "industrial" versions of temporal logic such as Sugar at IBM and ForeSpec at Intel (1994-1998).
- Accelera discussions that culminate in the PSL standard (1998-2004).

As we can see, it took almost 30 years to push theoretical ideas into industrial-strength tools and along the way, in addition to impressive algorithmic development, cultural gaps between theoreticians and practitioners had to be bridged. Verification is founded upon logic, automata and semantics which are part of theoretical computer science. To a certain extent some knowledge of these topics is part of the background of digital designers via Boolean logic and sequential finite-state machines. In the other direction, most theoretical computer scientists have a basic understanding of digital circuits, at least at the gate and flip-flop level of abstraction. Conferences like *CAV: Computer-Aided Verification*, as well as collaborations between researchers and industry, contributed a lot to the creation of a common verification culture.

The situation in the analog domain is radically different. The electrical behavior of transistors in digital circuits is just a means to realize logical gates. This implies that the functional correctness of a circuit can be validated using an abstract model that corresponds to a sequential machine, without worrying too much about the physical realization. Such physical details may influence "non-functional" properties of the circuit such as timing or power consumption, but the logical abstraction is robust with respect to such variations. In contrast, the functionality of analog circuits is defined directly in terms of continuous electrical quantities. As such they are much more sensitive to unavoidable (and sometimes random) perturbations from the overall operating environment such as voltage from the power supply, temperature, noise from nearby electronics or noise from fundamental sources such as the physics of the insides of the basic transistors. One unpleasant consequence of this situation is that important parameters of the circuit dynamics are determined only after physical placement or even fabrication.

The behavior of analog circuits is simulated and analyzed using models of continuous dynamical systems specified by differential and algebraic equations. Systematic mathematical support for these activities exists typically only for linear systems. The culture of designers may include many non-digital parts of electrical engineering, including signal processing and its mathematical basis (Fourier analysis, information theory), linear algebra and linear systems theory, numerical computations and, of course, intimate knowledge of the behavior of real transistors and of the physics of the particular application domains in which the circuits are used. Most of these domains are outside the background of computer scientists. The history of the attempts to export formal verification methodology toward models admitting real-valued variables and dense time is much shorter and is outlined below:

- First attempts to define specification formalisms and computational models for real-time systems (1985-1995).

- Positive results concerning the verification of timed automata (Alur and Dill 1990).
- A first proposal to verify hybrid systems (Maler, Manna and Pnueli 1991).
- Development of the algorithmic approach for the analysis of timed and "linear" hybrid automata (Henzinger et al); First tools: Kronos (timed automata, Verimag), Hytech (hybrid automata, Berkeley); Negative undecidability results that show that exact verification is impossible even for systems with few continuous variable and very simple dynamics (1992-1998).
- Development of the first tools for *approximate* verification of continuous and hybrid systems having non-trivial continuous dynamics with few state variables (Checkmate, CMU 1999; **d/dt**, Verimag 2000; Hysdel, ETH 2000); Control systems analysis is the main application domain.
- First attempts to apply formal verification to analog circuits (2002-present).

As one can see, the verification of systems having continuous variables is, provably, much more difficult than that of finite-state systems, and that verification tools for such systems are still in their infancy. Before trying to export verification methodology to analog circuits we should first clarify the motivation for doing so. Chips with analog or RF functions on them are notoriously difficult to get right on the first try, so anything we do that helps uncover more functional flaws or "bad behaviors" in subtle corners of the performance space is regarded as worthwhile. Designers of such circuits do use mathematical models for analytical and numerical computations, but the treatment of these models is more in the engineering and applied mathematics tradition, without the careful semantic and methodological concepts developed for modeling digital concurrent systems. Since the importance of analog components grows as systems become more integrated on a chip and more embedded in the physical world, any contribution toward accelerating their development will have significant economic consequences. It is believed that formal methods will occupy some complementary niche among the currently-used design methods for analog systems, as they already do for digital systems.

3 Workshop Overview

This issue consists of the papers presented in the first workshop on the domain. The contributions reflect some of the diversity of analog circuit verification, ranging from formal model checking to small-signal sensitivity analysis. They also illustrate the fundamental challenges: countering the state-explosion problem, trading speed against accuracy in abstractions, and describing properties of practical interest in formalisms suitable for verification. A variety

of circuits are considered: oscillator circuits such as a tunnel-diode oscillator and a voltage-controlled oscillator, a sequential mixed-signal circuit, a Schmitt trigger, a switched-capacitor integrator and a surfing pipeline based on self-resetting Domino XOR-Gates.

In the paper by Frehse et al., models defined by linear bounds and continuous time dynamics are analyzed in a formally sound fashion over an infinite time horizon. Monitor automata are used to verify circuit properties such as jitter using conventional reachability techniques.

A slightly more abstract view is taken in the paper by Freibothe et al., where similar models, but with discrete time dynamics, are analyzed over a finite time horizon, which is also referred to as bounded model-checking. This approach allows to consider more complex circuit architectures.

In the approach of Grabowski et al., accuracy is traded against speed by establishing a non-conservative finite state abstraction of the system, which can then be analyzed by efficient model-checking techniques. Circuit properties are described using a special temporal logic that allows one to include timing imformation.

Hybrid timed petri nets are used as modeling paradigm in the paper by Myers et al., which demonstrates the advantages of verification vs. simulation with an illustrative example. It demonstrates that standard simulation techniques, such as randomized or extreme point simulations, may fail to detect unwanted behavior that can be found using formal verification.

In the paper by Yang and Greenstreet, a small signal sensitivity analysis is applied to a nonlinear model of surfing circuits to guarantee their robustness. The results are used to improve the large signal stability analysis based on optimization.