INTEGRATION, the VLSI journal xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

There is a limit to everything: Automating AMS operating condition check generation on system-level

Georg Gläser[a,*], Martin Grabmann[a], Gerrit Kropp[a], Andreas Fürtig[b]

[a] IMMS Institut für Mikroelektronik- und Mechatronik-Systeme gemeinnützige GmbH Ilmenau, Germany
[b] Institute for Computer Science, Goethe Universität Frankfurt a. M., Germany

ABSTRACT

Analog/Mixed-Signal (AMS) design and verification strongly relies on more or less abstract models to make extensive simulations feasible. Maintaining consistent behavior between system model and implementation is crucial for a correct verification. Besides the functionality, this also involves the operating conditions: A faulty model might introduce false-positive verification results despite of e.g. an incorrect supply voltage. We propose a novel method to compare a model to its implementation and automatically generate a *model safe-guard unit* from transistor-level simulation data. By using the concept of system acceptance region, we transfer the information about valid operating conditions to a higher level of abstraction. We introduce this unit into a purely functional VerilogAMS model to check the validity in the simulated environment. This method can be used to significantly reduce the risk of erroneous verification results on system level.

We demonstrate our approach using a demodulator circuit of a passive RFID tag. The model is automatically augmented with additional checks derived from an exploration of the underlying circuits' parameter space. By comparing the risk of false-positive simulation results, we prove that the design risk can be significantly reduced. It is eliminated completely by introducing an additional safety-margin, which drives the coverage up to 100%.

1. Introduction

The verification of complex analog mixed-signal (AMS) systems heavily relies on the use of behavioral models to keep the simulation effort in a reasonable range. Contrary to the design process, the verification is usually following a bottom-up approach. Since full-chip analog simulations are not feasible due to enormous processing times, behavioral models are created. These models represent the functional aspects of the circuit in a very abstract way by only modeling for instance the input/output relations. As shown in Fig. 1, this abstraction step imposes the challenge of verifying the operating conditions (OC) of each component. If a behavioral model is used in conditions where the underlying transistor-level circuit is not working within specification (e.g. with disabled supply voltage), the verification result is not valid or even false-positive. Still, the check for environmental conditions is normally not the main focus of model generation. Therefore, the behavioral model must be augmented with further checks or assertions for the relevant OC. Capturing the full operating region with manual checks imposes a significant effort which is for instance a major obstacle for mixed-signal IP (intellectual property, i.e. a fully characterized building block). This results in very simple or even oversimplified checks or additional simulations using the transistor-level representation of the IP. In addition, there are only few methods to compare an abstract model e.g. in VerilogAMS to a transistor-level schematic since these representations are essentially different and may not even share the same simulation environment [1].

In this contribution, we show a methodology to automate the process of OC check generation and use a model coverage measure to compare it to the schematic level implementation. While our approach is not limited to certain AMS modeling languages, we focus on the task of realizing the OC check in VerilogAMS since it is widely supported in design tools and can be simulated together with transistor-level circuits. The first important step is to generate an implementation-agnostic representation of the valid operating region. This is done using the concept of system acceptance regions. As a meta-model for this OC acceptance region, we train an SVM-classifier *(Support Vector Machine)* following the ideas presented in Refs. [2] and [3]. These contributions use the classifier to efficiently sample and explore a component's parameter space. Based on the acquired information, we generate a *Model Safe-Guard Unit* (MSU) that is automatically inferred into a

* Corresponding author.
E-mail addresses: georg.glaeser@imms.de (G. Gläser), martin.grabmann@imms.de (M. Grabmann), gerrit.kropp@imms.de (G. Kropp), fuertig@em.informatik.uni-frankfurt.de (A. Fürtig).

https://doi.org/10.1016/j.vlsi.2018.02.016
Received 29 September 2017; Received in revised form 19 January 2018; Accepted 1 February 2018

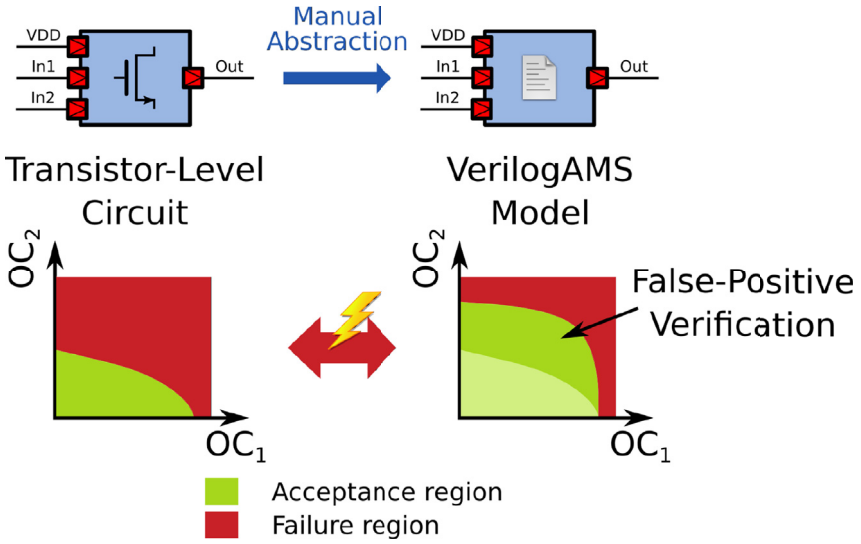

Fig. 1. A VerilogAMS model operated in conditions where the underlying circuit is not performing as expected may result in a false-positive verification.

VerilogAMS model. To extract information about OC and provide a method for reporting or impact modeling, we also introduce sensor and actuator modules along with our MSU. Our methodology results in an augmented VerilogAMS model to be used for simulation without need for further manual interaction. Since this model guarantees to operate only within the valid region of the implementation, the model coverage is driven to nearly 100%.

We demonstrate our method by examining the demodulator of an RFID *(Radio-Frequency Identification)* tag. This block is located in the communication system. It maps the analog waveforms to a digital signal for processing e.g. commands from the reader system. We extract the OC acceptance region of the block using the methods shown in Ref. [2] and pass the resulting data to our MSU generation scheme. To prove our result, we compare the resulting acceptance region of the augmented model to the transistor-level circuit using model coverage.

This paper is structured as follows. After a short review of the related work, we present the concept of system acceptance regions. This builds the base for the model coverage measure following the explanations given in Ref. [4]. To ensure a model coverage of 100%, we present in the following section the MSU generation scheme. The power of our approach will be shown in a case study by augmenting a model of a mixed-signal demodulator circuit. After evaluating the resulting coverage, we discuss the possible impact on simulation performance and possible future approaches.

2. Related work

The operation condition check (OCC) for single devices (i.e. transistors, resistors, etc.) is currently an essential step in the System-on-Chip (SoC) design process. It does not check system-dependent environmental conditions as for instance if the supply voltage is switched on. The OCC is used to ensure that all primitive devices used in a circuit are operating within the safe operating area for the simulated conditions. Since this method is focused on single devices, current tools as e.g. Ref. [5] do not provide extensions to higher levels of abstraction. These are addressed by current research:

In the digital design process, PSL [6] is typically used to formalize and check properties of logical signals in different types of models. One application is the use of assertions that enable fast falsification and error detection [7]. These methods try to abstract the device under test to an automaton or graph structure checking previously described properties.

In the analog domain, properties can be checked in a similar way. Analog chips are typically tested by hand-made input stimuli created by the designer. By discretization of the state space, model checking techniques can be applied [8] using ASL, the analog specification language. The authors of [9] present an assertion-based verification methodology, combining the digital, analog and software domain by automatically creating observer automata. The to be checked properties need to be specified in a mixed-signal assertion language (MSAL). This produces a huge amount of overhead and lacks of support of low-level analog properties [10]. creates traces through a symbolic simulation based on differential algebraic equations (DAE) for analog and discrete values of the digital part.

All these AMS approaches either require in-depth adaptions of the simulator or specialized manual descriptions of the targeted properties. They do no integrate easily into available applications. In addition, an automatic flow for mining operating regions for automatically generating checks has not been presented.

3. System acceptance regions

The base for the presented method are system acceptance regions, which we shortly introduce in this section. They first emerged in the form of *Shmoo plotting* [11]. This concept was used for the simulations of integrated circuitry by e.g. Ref. [2]. We briefly summarize the argumentation of this article.

Consider the setup shown in Fig. 2. A device under test (DUT) is exercised by a test fixture in a certain way to test whether it complies to a given specification for a given set of parameters. It is assumed, that the test is complete, i.e. it verifies that the regarded part of the specification holds in all relevant scenarios. We define the *system acceptance region* of a given DUT as the set of points in a regarded parameter space, where its behavior complies to its specification. Note that the DUT's behavior is classified only binary. That means, no target performance value such as for instance amplification is regarded but only whether it passes the test for specification compliance. This simplification neglects *borderline cases* where the DUT only partly fulfills the specification or some performance goal is only slightly missed.

While this concept seems fairly natural, it exposes some very useful features. It has in theory no limitation regarding the dimensionality of the parameter space under examination. This paves the way to very general approaches based on this concept. Still, the actual exploration of concrete acceptance regions is still a topic of research, see e.g. Refs. [3,12]. The most important property of the shown regions is the independence from the actual model implementation method. Transistor level circuits can be examined as well as very abstract real-number models in VerilogAMS. The only requirement that is imposed is the