



A static verification approach for architectural integration of mixed-signal integrated circuits

Rajdeep Mukhopadhyay^{a,*}, Anvesh Komuravelli^a, Pallab Dasgupta^a, S.K. Panda^a,
Siddhartha Mukhopadhyay^b

^a Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India

^b Department of Electrical Engineering, Indian Institute of Technology, Kharagpur, India

ARTICLE INFO

Article history:

Received 21 August 2008

Received in revised form

10 March 2009

Accepted 19 May 2009

Keywords:

Static verification

Design integration

Mixed-signal circuit specification

Formal model

ABSTRACT

In this paper we present a static method for verifying the proper integration of analog and mixed-signal macroblocks into an integrated circuit. We consider the problem in a setting where there is no golden reference for verifying the validity of the interconnections between the blocks. The proposed verification methodology relies on an abstract modeling of the functional behavior of the blocks and a set of consistency criteria defined over the composition of these abstract models. A new formalism called *mode sequence chart* (MSeqC) has been presented for capturing the behavior of the blocks at a level of abstraction that is suitable for interconnection verification. We present rules to compose the MSeqCs of each block in an integrated design and present three criteria that indicate possible interconnection faults. We present a tool called AMS-IV (AMS-interconnection verification) that takes the design netlist as input, the MSeqC model of each design block as reference, and tests the three criteria.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

As designers attempt to integrate multiple pre-designed and pre-verified design blocks into an integrated circuit, the task of verifying that the integration has been done correctly has become a major challenge. The problem becomes significantly more complex when the component blocks are mixed-signal in nature, since mixed-signal simulation is prohibitively complex as compared to pure digital simulation.

In this paper we study the verification problem associated with integrating several complex mixed-signal circuit blocks into an integrated circuit. We have been studying this problem in the context of integrating power management units (PMUs) for portable devices, such as cell phones, PDAs, and laptops. A typical PMU contains several linear drop out regulators (LDOs), a few buck regulators, a battery charger and several other blocks for bias generation, UVLO, etc., and digital control. Typically the manufacturers of such PMUs have existing design IPs for these components, and the main challenge is to integrate a combination of these components (as required by the customer) into an

integrated PMU within very low time budget. The whole design fails if the components are not interconnected correctly, where interconnection (in the analog sense) also refers to proper polarity, drive strength, etc.

We require a functional specification as a reference in this verification problem, even if the goal is only to verify that the interconnections have been made correctly. This is due to the fact that there is a manual point of entry for the interconnections. There are broadly two practices for integrating multiple mixed-signal circuit blocks into an integrated chip. The most common form is to make the interconnections graphically using a schematic editor (such as Cadence Virtuoso schematic editor [25]). The other approach attempts to enter the interconnections textually, either directly using text formats (using HDLS such as Verilog [23], VerilogA [24], etc.) or indirectly using scripting languages. In both approaches the point of entry is human and therefore prone to similar types of errors. Therefore integration verification is not achievable without functional verification of the integrated design.

Formal verification methods for AMS designs based on equivalence checking [2,14,22], model checking [13,9,3,15], theorem-proving [7,11,10], etc. do not address the problem of verifying integration of large integrated AMS designs. On the other hand run-time verification methods such as [4,18,16] require simulation of the integrated AMS design. Typically the integrated designs that we refer to are so large that adequate functional verification through mixed-signal simulation of the integrated netlist is

* Corresponding author.

E-mail addresses: rajdeep.mukhopadhyay@gmail.com (R. Mukhopadhyay), anvesh@cse.iitkgp.ernet.in (A. Komuravelli), pallab@cse.iitkgp.ernet.in (P. Dasgupta), subrat@cse.iitkgp.ernet.in (S.K. Panda), smukh@ee.iitkgp.ernet.in (S. Mukhopadhyay).

infeasible in practice, as also observed by the author of [21]. At the same time the full functionality of the component blocks is typically not required for verifying the correctness of the integration. For example, it is possible to replace the component blocks with lightweight behavioral models and ramp up the simulation speed of the integrated design. In a related work we adopted this approach using Verilog A [24] models with reasonable success. Based on our experience, we have the following observations:

- (1) Developing Verilog A models for complex blocks like buck regulators is non-trivial. It is possibly too much of an effort to develop executable behavioral models for all legacy designs of the components.
- (2) Verilog A models are not suitable for static (formal) analysis. We believe that most interconnection faults can be detected statically against a well defined functional specification.
- (3) The industry seems to be moving towards platform based design methodologies where third party design IPs are integrated at the physical level (post layout). It is unlikely that behavioral models for such components will be made available.

In this paper we present a formalism, called *mode sequence charts* (MSeqC), for modeling the component blocks of an integrated circuit at a very high level of abstraction. A mode sequence chart captures the macromodes at which a mixed-signal block works, and the admissible transitions between these modes. Each state of the chart is annotated with a set of assumptions that are made about the environment when the component is in that mode, and a set of properties that the component guarantees while operating in that mode.

There exists a number of formalisms to specify components of a system at various levels of abstractions. In [5] an automata based language for specifying software components have been proposed. An extension of hybrid automata [1] to model interaction among various hybrid systems is proposed in [17]. StateCharts [12], which is another formalism to model complex systems is based on extension of state machine models. Message sequence chart [19] is a graphical and textual language for specifying interaction among various communicating system components.

Fundamentally, the solution provided in this paper is based on discrete mode abstraction. Consider a hybrid automaton. In each

mode, such an automaton has a set of invariants and a dynamic law. By ignoring the mode dynamic, and partitioning the invariants into environment invariants (assumptions) and component invariants (guarantees) one obtains a Kripke structure (finite state machine) that has a set of states (the modes) and a set of atomic propositions (the invariants) associated to each mode. This structure is amenable to discrete model checking and compositional reachability analysis, which is used to address the interconnection problem.

Mode sequence charts are a particularly tailored formalism for defining the underlying Kripke structure at a level of abstraction which is not alien to analog designers, and can also be analyzed using static model checking tools.

Each global state of the integrated circuit is a composition of compatible states of the mode sequence charts of the components. Interconnection faults manifest themselves in terms of incompatibility of local states corresponding to a desired global state, and in terms of unreachability of desired global states. The main contributions of this paper are as follows:

- (1) We introduce mode sequence charts as a high level formalism to be used in interconnection verification.
- (2) We present the composition rules for integrating the mode sequence charts of the components based on the interconnections entered by the designer in the schematic.
- (3) We present three different formal criteria for verifying the correctness of the interconnections. Interconnection faults are typically manifested in terms of violations of these criteria.
- (4) We have developed a tool called analog and mixed-signal interconnection verifier (AMS-IV) tool that essentially checks the three criteria and produce three corresponding reports. These reports are useful to determine whether the circuit has any interconnection fault.

In the next section we illustrate our basic idea of interconnection verification with a toy example. Section 3 formally presents *mode sequence chart* and Section 4 describes the interconnection model. In Section 5 we propose three interconnection fault detection criteria and explain them with examples. Section 6 presents the algorithm and implementation of the AMS-IV tool briefly. In Section 7 we present a case study on a voltage mode buck regulator circuit. Section 8 concludes the paper with a few possible topics for future research.

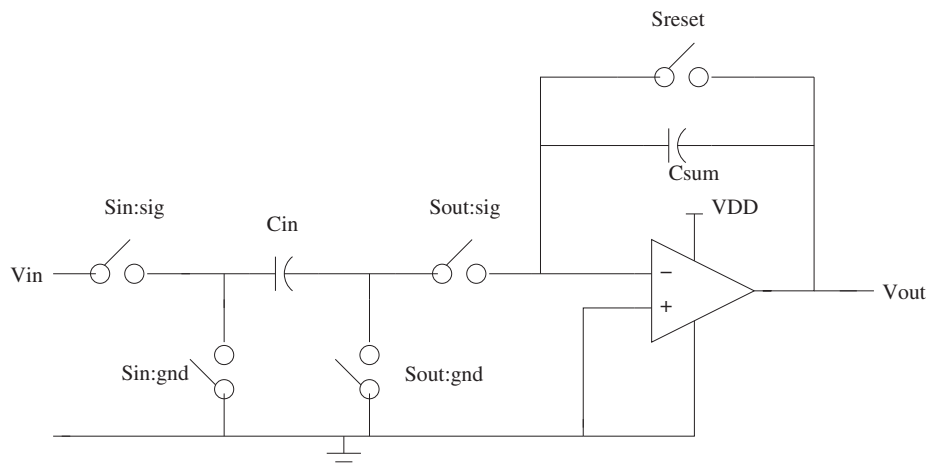


Fig. 1. A switched capacitor integrator circuit [20].

Download English Version:

<https://daneshyari.com/en/article/538489>

Download Persian Version:

<https://daneshyari.com/article/538489>

[Daneshyari.com](https://daneshyari.com)