

Muhammad Hassan
Daniel Große
Rolf Drechsler

Enhanced Virtual Prototyping for Heterogeneous Systems

 Springer

Enhanced Virtual Prototyping for Heterogeneous Systems

Muhammad Hassan • Daniel Große •
Rolf Drechsler

Enhanced Virtual Prototyping for Heterogeneous Systems

 Springer

Muhammad Hassan
DFKI GmbH
Bremen, Germany

Daniel Große
Johannes Kepler University of Linz
Linz, Austria

Rolf Drechsler
University of Bremen and DFKI GmbH
Bremen, Germany

ISBN 978-3-031-05573-7 ISBN 978-3-031-05574-4 (eBook)
<https://doi.org/10.1007/978-3-031-05574-4>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2023

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To Khushboo and Maryam,
Marie,
and
Anna.*

Preface

The successful co-design and verification of secure multi-disciplinary heterogeneous *systems-on-chips* (SOCs) with tight interactions between *hardware/software* (HW/SW) systems and their analog physical environment is an increasingly daunting task. In this regard, the emergence of *virtual prototypes* (VPs) at the abstraction of *electronic system level* (ESL) has modernized the design and verification of heterogeneous SOCs. A VP is essentially an executable abstract model of the entire HW platform and pre-dominantly created in C++-based system modeling language SystemC together with *transaction level modeling* (TLM) techniques and its mixed-signal extension SystemC AMS. The much earlier availability as well as the significantly faster simulation speed in comparison to the *register transfer level* (RTL) models and SPICE-level models are among the main benefits of VPs. Thus, virtual prototyping enables HW/SW co-design and verification very early in the design flow. Serving as reference for (early) embedded SW development and HW verification, the functional correctness and security validation of VPs are very important. Hence, a VP is subjected to rigorous functional verification. However, the modern VP-based verification flow still has weaknesses, in particular due to lack of methodologies to capture the complex interactions between digital and analog designs as well as unavailability of security validation techniques. This book proposes several novel approaches that cover varying verification aspects to strongly enhance the modern VP-based verification flow. The chapters of the book are essentially divided into four parts: The first part introduces a new verification perspective for VPs by using *metamorphic testing* (MT) as no reference models/value are needed for verification, unlike in modern VP-based verification flow. The second part enhances the code coverage closure methodologies in modern VP-based verification flow by considering *mutation analysis* and stronger coverage metrics like *data flow* coverage. The third part covers a set of novel, systematic, and lightweight functional coverage-driven verification methodologies to improve the coverage closure. The fourth and final part of the book showcases novel approaches to enable early security validation of VPs. All approaches are presented in detail and

are extensively evaluated with several experiments that clearly demonstrate their effectiveness in strongly enhancing the modern VP-based verification flow.

Bremen, Germany
Linz, Austria
Bremen, Germany
November 2021

Muhammad Hassan
Daniel Große
Rolf Drechsler

Acknowledgments

First, we would like to thank the members of the research group for *computer architecture* (AGRA) at the University of Bremen as well as the members of the research department for *cyber-physical systems* (CPS) at the *German Research Centre for Artificial Intelligence* (DFKI) in Bremen. We appreciate the great atmosphere and stimulating environment. Furthermore, we would like to thank all co-authors of the papers which formed the starting point for this book: Hoang M. Le, Vladimir Herdt, Mingsong Chen, and Mehran Goli. We especially thank Karsten Einwich and Thilo Vörtler from COSEDA Technologies GmbH for many interesting discussions and successful collaborations.

Contents

1	Introduction	1
1.1	Electronic System-Level Design and Verification	3
1.2	Book Contribution	8
1.2.1	Contribution Area 1: AMS Metamorphic Testing Environment	9
1.2.2	Contribution Area 2: AMS Enhanced Code Coverage Verification Environment	10
1.2.3	Contribution Area 3: AMS Enhanced Functional Coverage Verification Environment	11
1.2.4	Contribution Area 4: Digital Early Security Validation	12
1.2.5	Contribution Summary	12
1.3	Book Organization	13
2	Preliminaries	15
2.1	SystemC	15
2.1.1	Basics	16
2.1.2	Transaction-Level Modeling (TLM)	18
2.2	SystemC AMS	19
2.2.1	Models of Computation (MOC)	20
2.2.2	Timed Data Flow (TDF)	20
3	AMS Metamorphic Testing Environment	23
3.1	MT-Based System-Level Verification Approach	24
3.1.1	Overview	25
3.1.2	Test-Case Generator	25
3.1.3	Metamorphic Relations	26
3.1.4	Core Properties	26
3.2	Metamorphic Testing for RF Amplifiers	26
3.2.1	MT Principle for RF Amplifiers	27
3.2.2	Identification of Metamorphic Relations	28
3.2.3	Experimental Evaluation	31
3.3	Metamorphic Testing for PLLs	36

3.3.1	Phase-Locked Loop	37
3.3.2	MT Principle for Mixed-Signal Interactions	38
3.3.3	Identification of MRs for PLLs	38
3.3.4	Experiments	41
3.4	Summary	43
4	AMS Enhanced Code Coverage Verification Environment	45
4.1	Software Driven Verification for IP Integration	46
4.1.1	SW Test Qualification Methodology	47
4.1.2	Consistency Demonstration Example	53
4.1.3	Experimental Results	58
4.2	Data Flow Testing for Digital Virtual Prototypes	63
4.2.1	SystemC Running Example	64
4.2.2	Def-Use Association and Data Flow Testing	66
4.2.3	Data Flow Testing for SystemC	66
4.2.4	Implementation Details	72
4.2.5	Experimental Results	73
4.3	Data Flow Testing for SystemC AMS Virtual Prototypes	74
4.3.1	SystemC AMS Motivating Example	74
4.3.2	Data Flow Testing for SystemC-AMS TDF Models	76
4.3.3	Implementation Details	83
4.3.4	Experimental Results	84
4.4	Summary	85
5	AMS Enhanced Functional Coverage Verification Environment	87
5.1	Preliminaries	90
5.1.1	Functional Coverage	90
5.1.2	AMS VP Verification Environment and Deficiencies	90
5.2	Enhanced Functional Coverage Verification Environment Setup	91
5.2.1	Running Example: LNA	92
5.2.2	Environment Setup	92
5.3	AMS Functional Coverage-Driven Verification Approach	98
5.3.1	Coverage Analysis	98
5.3.2	Industrial Case Study	101
5.4	Lightweight Coverage Directed Stimuli Generation	104
5.4.1	Revisiting Output Coverage Definition and Collection	104
5.4.2	Lightweight Coverage Analysis	106
5.4.3	Experimental Evaluation	113
5.5	Summary	122
6	Digital Early Security Validation	123
6.1	Static Information Flow Analysis	125
6.1.1	Approach Overview	125
6.1.2	Data Flow Driven Information Flow Analysis	129
6.1.3	Experimental Results	136
6.2	Dynamic Information Flow Analysis	138

- 6.2.1 Motivating Example and Threat Models 138
- 6.2.2 Dynamic IFT Methodology 142
- 6.2.3 Experimental Evaluation 149
- 6.3 Summary 154
- 7 Conclusion** 155
 - 7.1 Future Directions 157
- Bibliography** 159
- Index** 165

List of Algorithms

Algorithm 1	Proposed LCDG approach to capture <i>linear</i> , <i>non-linear</i> , and <i>unstable</i> behaviors	109
Algorithm 2	Direct security property generation	146
Algorithm 3	Indirect security property generation	146
Algorithm 4	Security violation detection	148

List of Figures

Fig. 1.1	Early SW development leveraging shift left concept	2
Fig. 1.2	VP placement in the complete SOC design flow	4
Fig. 1.3	General AMS VP verification environment	5
Fig. 1.4	Four stages of modern simulation-based VP verification flow	6
Fig. 1.5	Proposed enhancements in VP verification flow	8
Fig. 1.6	Contributions of book: enhanced VP verification flow	9
Fig. 2.1	SystemC architecture with TLM and AMS extensions [1, 71]	16
Fig. 2.2	TLM: initiator, target, socket, and interconnect	18
Fig. 2.3	Simulation speed of modeling languages in comparison to SPICE [9]	19
Fig. 2.4	Second-order LPF	21
Fig. 3.1	Emerging verification techniques	24
Fig. 3.2	Metamorphic testing example—SUM function	25
Fig. 3.3	Graphical illustration of MR for amplifier <i>output = 7 × input</i> . (a) Base test-stimulus: $x(t) = \sin(2\pi 5000t)$. (b) Amplifier output at base test-stimulus. (c) The follow-up test-stimulus: $3 \times x(t) = 3 \sin(2\pi 5000t)$. (d) Amplifier output at the follow-up test-stimulus	28
Fig. 3.4	<i>I_{LNA}</i> Output power vs. input power (dBm). Overshoot of output power because of non-linearity approximation	33
Fig. 3.5	Fault-detection quality of MT-based verification	34
Fig. 3.6	Gain output of FLNA	35
Fig. 3.7	Power output of FLNA	35
Fig. 3.8	PLL top level diagram	37
Fig. 3.9	Graphical illustration of MR for charge pump (CP)	39
Fig. 3.10	PLL faulty behavior—dead-zone effect revealed by MR4	42
Fig. 3.11	FFT of PLL faulty behavior	42
Fig. 3.12	Phase frequency detector (PFD) without a delay element	43

Fig. 3.13	PLL behavior after addition of delay element in PFD	43
Fig. 4.1	Enhanced structural coverage verification	46
Fig. 4.2	SW qualification methodology	52
Fig. 4.3	SC_THREAD find_max with the original coverage results for tests 1 and 2	55
Fig. 4.4	Consistency ex.: coverage results for C2 example	56
Fig. 4.5	Consistency ex.: coverage results for C4 example	57
Fig. 4.6	An overview of our data flow testing approach for SystemC	66
Fig. 4.7	Sensor system—T: Temperature, H: Humidity, XS: X Sensor (X=T,H), Z^{-1} = delay, AMUX: Analog mux, ADC: Analog-to-Digital Converter, X_LED: Light-emitting Diode, G: Gain	76
Fig. 4.8	Proposed data flow testing methodology overview	77
Fig. 5.1	Enhanced functional coverage verification	88
Fig. 5.2	AMS enhanced functional coverage verification environment—LNA: Low Noise Amplifier, BPF: Band-Pass Filter, res: resolution, ires: interval resolution	91
Fig. 5.3	Graphical illustration of <i>resolution</i> parameter. (a) Stimuli parameter resolution definition with 0.2 V amplitude difference. (b) Stimuli parameter resolution definition with 0.2 Hz frequency difference	93
Fig. 5.4	Overview on AMS functional coverage-driven approach—res: resolution, ires: interval resolution	99
Fig. 5.5	Gain (G_1) vs. input power	101
Fig. 5.6	Case study: RF transmitter and receiver model	102
Fig. 5.7	Case study—coverage closure w.r.t. resolution	103
Fig. 5.8	Overview on LCDG approach—res: resolution, ires: interval resolution, L/NL: linear/non-linear coverage goals, O/U: overshoot/undershoot coverage goals, SCG: secondary-extension coverage goals	105
Fig. 5.9	Gain curve (output power vs. input power)—overshooting behavior due to Taylor series approximation	121
Fig. 6.1	Early security validation	124
Fig. 6.2	Motivating example	126
Fig. 6.3	Information flow analysis overview	128
Fig. 6.4	One access path for example shown in Listing 6.1	132
Fig. 6.5	The architecture of the motivation example <i>RISC-32 SOC</i>	140
Fig. 6.6	The architecture of the proposed methodology	142
Fig. 6.7	Data extraction. (a) part of the generated <i>Debug symbols</i> and (b) part of the generated <i>GDB command file</i> of the <i>RISC-32 SOC</i> model	143

List of Tables

Table 4.1	Summary of SystemC specific mutation operators	49
Table 4.2	Consistency analysis results for a mutation	50
Table 4.3	Consistency example: coverage legend	54
Table 4.4	IRQMP SW test qualification results	61
Table 4.5	GPTimer SW test qualification results	63
Table 4.6	Data flow associations for the SystemC example in Listing 4.6 sorted by classification	70
Table 4.7	SystemC-AMS TDF models-specific data flow associations—reference Listing 4.7	81
Table 4.8	Case study: car window lifter system and Buck-boost converter data flow associations	84
Table 5.1	LNA: Parameter A (amplitude) coverage report	96
Table 5.2	LNA: Parameter F (frequency) coverage report	96
Table 5.3	LNA: Input power coverage report	96
Table 5.4	LNA gain (G_1) output coverage report	97
Table 5.5	LNA gain (G_1) coverage report. Case C2: addition of bins in gain coverpoint between 18.8 dB and 19.9 dB	100
Table 5.6	LNA gain (G_1) coverage report. Case C1: static parameter refinement on input stimuli	102
Table 5.7	Cross-coverage bug-free DUV (excerpt)—checker vs. input power vs. gain (G_1)	102
Table 5.8	LNA gain (G_2) output coverage report—iteration 1	106
Table 5.9	LNA gain (G_2)—overshoot/undershoot coverage goals	106
Table 5.10	LNA gain (G_2)—secondary-extension goals	107
Table 5.11	Excerpt of LNA gain (G_2) output trace after iteration 1	107
Table 5.12	LNA gain (G_2) output coverage report—iteration 2	112
Table 5.13	LNA gain (G_2) output coverage report—iteration 4	112
Table 5.14	LNA gain (G_2) output coverage report—iteration 6	113
Table 5.15	LNA gain (G_2)—secondary-extension goals	114
Table 5.16	Cross-coverage (excerpt)—checker vs. input power vs. gain	114

Table 5.17	Cross-coverage (excerpt)—frequency vs. input power vs. gain	114
Table 5.18	Specifications of seven industrial LNAs	115
Table 5.19	LNAs case study—gain (G) progress over multiple iterations	117
Table 5.20	LNAs case study—1 dB compression point progress over multiple iterations	118
Table 5.21	LNAs case study—input third-order intercept point (IIP3) progress over multiple iterations	119
Table 5.22	LNAs case study—input second-order intercept point (IIP2) progress over multiple iterations	120
Table 6.1	Experimental results for all case studies	151

Chapter 1

Introduction



Internet-of-Things (IOT) and *5G* (fifth generation technology standard for broadband cellular networks) have enabled a plethora of possibilities which were once unimaginable. While *5G* provides the high-speed connectivity and ubiquitous coverage, it is the smart IOT devices that gather and transport the data that fuel the promise and potential of IOT. These smart devices are a prime example of heterogeneous *System-On-Chips* (SOCs), which comprise two parts: (1) *Mixed-Signal Hardware* (HW) where analog world meets the digital world, (2) and *Software* (SW), the invisible layer that connects us to the physical reality. Heterogeneous SOC are among the fastest growing market segments in the electronics and semiconductor industry. Driven by growth opportunities in various application domains, many semiconductor vendors are adapting and shifting their focus from separate *Integrated Circuits* (ICs) performing one functionality, toward a more integrated solution of *Radio Frequency* (RF) and high-performance *Analog/Mixed-Signal* (AMS) designs. Due to this industry shift, most SOC today are heterogeneous containing analog sensors, mixed-signal converters, digital processors running SW on top, and RF transceivers, tightly integrated on a single die. While this shift has resulted in high-performance, efficient, and low-area devices, e.g., Apple M1 SOC [6], it has significantly increased the efforts required to develop and verify these highly complex devices and achieving the required *Time-To-Market* (TTM) simultaneously.

The first challenge in this regard is the HW and SW dependency. Conventionally, HW and SW were developed in isolation and only met each other at the late integration and testing stages. As a consequence, a sequential dependency between HW and SW development phases always existed as shown in Fig. 1.1a. Hence, SW could only be tested properly once the first silicon prototypes of the SOC were available. In particular, HW dependent SW such as device drivers and low-level kernel code could only be written after the silicon design had been completed. One solution to lessen the TTM widely adopted by industry is to move away from complete in-house HW development and instead use larger amounts of pre-

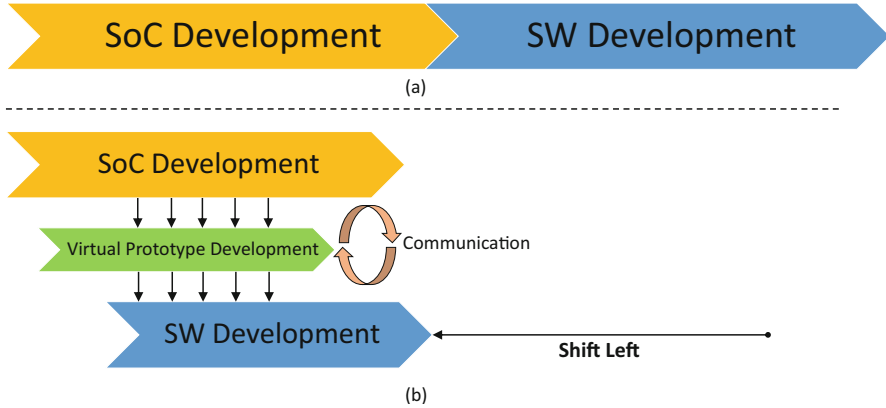


Fig. 1.1 Early SW development leveraging shift left concept

verified third-party *Intellectual Property* (IP). It allows them to focus more on the *Unique Selling Point* (USP) of their HW and SW. With the increased SW feature-rich functionality and complex interactions between different HW components, the design complexity has increased manifold making the design verification of heterogeneous SOCs an increasingly daunting task.

The second challenge in verification of heterogeneous SOCs is the slow joint simulation speed of *Register Transfer Level* (RTL) and SPICE (Simulation Program with Integrated Circuit Emphasis) models for the digital and analog/RF part of the SOC [9]. Traditionally, analog/RF verification methodology was ad hoc by nature and these IPs were always verified by separate teams. It was driven by directed tests run over sweeps, corners, and Monte Carlo analysis. Unfortunately, it has not changed much until now and creates a bottleneck in the design and verification process. On the other hand, digital IPs had formalized verification methodologies used by separate teams. This included executable verification plans, constrained-random stimulus generation [119], testbench automation, assertions, and coverage metrics. These pre-verified analog, RF, and digital IPs were later on integrated together in a mostly digital SOC design and SW was executed on top to test if everything works as expected. However, due to multi-functional nature of the heterogeneous SOCs, the analog and RF IPs in particular have become very complex with significant digital control logic. Furthermore, the interaction of analog/RF and digital IPs has increased significantly, in particular when SW running on top is considered as well [77]. Hence, the traditional verification approaches are no longer adequate. The joint simulation, while slow, is still considered a golden standard because of high accuracy and cannot be ignored. However, it is too slow for chip-level simulations, unless it is used extremely selectively.

Third, design space and architectural exploration is restricted. Given the system requirements, finding the optimal system configuration is a tiring task. Each