SYSTEM C
SystemC

Methodologies and Applications

edited by

Wolfgang Müller
Paderborn University, Germany

Wolfgang Rosenstiel
Tübingen University, Germany

and

Jürgen Ruf
Tübingen University, Germany

KLUWER ACADEMIC PUBLISHERS
NEW YORK, BOSTON, DORDRECHT, LONDON, MOSCOW
Contents

Foreword ix
Preface xiii

Chapter 1
A SystemC Based System On Chip Modelling and Design Methodology 1
Yves Vanderperren, Marc Pauwels, Wim Dehaene, Ates Berna, Fatma Özdemir
1.1. Introduction 1
1.2. An Overview of the Methodology 2
1.3. Requirements Capture and Use Case Analysis 3
1.4. Modelling Strategies 5
1.5. Iterative Development, Model Refinement and Verification 20
1.6. Conclusions 25

Chapter 2
Using Transactional Level Models in a SoC Design Flow 29
Alain Clouard, Kshitiz Jain, Frank Ghenassia, Laurent Maillet-Contoz, Jean-Philippe Strassen
2.1. Introduction 29
2.2. Overview of the System to RTL Design Flow 31
2.3. TLM, a Complementary View for the Design Flow 33
2.4. TLM Modeling API 44
2.5. Standard Usage of the SystemC API 49
2.6. Advanced TLM API Usages 51
2.7. Example of a Multimedia Platform 52
2.8. Example of ARM Processor Subsystem 58
2.9. Conclusions 63

Chapter 3
Refining a High Level SystemC Model 65
Bernhard Niemann, Frank Mayer, Francisco Javier, Rabano Rubio, Martin Speitel
3.1. Introduction and Motivation 65
3.2. The OFDM Demodulator 66
3.3. High Level SystemC Model 68
3.4. Refinement to ANSI C 79
3.5. Further Refinement — Operator Grouping 87
3.6. Summary 93
3.7. Conclusions 95
Chapter 4
An ASM Based SystemC Simulation Semantics
Wolfgang Müller, Jürgen Ruf, Wolfgang Rosenstiel

4.1. Introduction
4.2. Related Works
4.3. Abstract State Machines
4.4. SystemC Basic Concepts
4.5. SystemC Operations
4.6. SystemC Scheduler
4.7. Example
4.8. Conclusions

Chapter 5
SystemC as a Complete Design and Validation Environment
Alessandro Fin, Franco Fummi, Graziano Pravadelli

5.1. Introduction
5.2. Methodology Overview
5.3. Design Error Modeling
5.4. High Level Validation of SystemC Design
5.5. Efficient Fault Simulation of a SystemC Design
5.6. Experimental Results
5.7. Concluding Remarks

Chapter 6
System Level Performance Estimation
Nuria Pazos, Winthir Brunnbauer, Jürgen Foag, Thomas Wild

6.1. Introduction
6.2. State of the Art
6.3. Methodology
6.4. Implementation Procedure
6.5. Methodology Verification
6.6. Case Study. Results and Evaluation
6.7. Conclusions and Outlook

Chapter 7
Design of Protocol Dominated Digital Systems
Robert Siegmund, Uwe Proß, Dietmar Müller

7.1. Introduction
7.2. Specification of Data Communication Protocols
7.3. An SVE Model of the USB 2.0 Protocol
7.4. Synthesis from SVE Protocol Specifications
7.5. Summary

Chapter 8
Object Oriented Hardware Design and Synthesis Based on SystemC 2.0
Eike Grimpe, Wolfgang Nebel, Frank Oppenheimer, Thorsten Schubert

8.1. Introduction
8.2. Related Work
8.3. High Level Hardware Modeling with SystemC
<table>
<thead>
<tr>
<th>Contents</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8.4. The SystemC Plus Approach</td>
<td>222</td>
</tr>
<tr>
<td>8.5. Hardware Synthesis from Object Oriented Models</td>
<td>233</td>
</tr>
<tr>
<td>8.6. Conclusions and Outlook</td>
<td>245</td>
</tr>
<tr>
<td>Chapter 9</td>
<td></td>
</tr>
<tr>
<td>Embedded Software Generation from SystemC for Platform Based Design</td>
<td>247</td>
</tr>
<tr>
<td>Fernando Herrera, Víctor Fernández, Pablo Sánchez, Eugenio Villar</td>
<td></td>
</tr>
<tr>
<td>9.1. Introduction</td>
<td>247</td>
</tr>
<tr>
<td>9.2. System Specification Methodology</td>
<td>253</td>
</tr>
<tr>
<td>9.3. SW Generation</td>
<td>259</td>
</tr>
<tr>
<td>9.4. SW/SW Communication and Driver Generation</td>
<td>267</td>
</tr>
<tr>
<td>9.5. Example of Software Generation in SystemC 2.0</td>
<td>268</td>
</tr>
<tr>
<td>9.6. Impact of SystemC 3.0 Release</td>
<td>270</td>
</tr>
<tr>
<td>9.7. Conclusions</td>
<td>272</td>
</tr>
<tr>
<td>Chapter 10</td>
<td></td>
</tr>
<tr>
<td>SystemC-AMS: Rationales, State of the Art, and Examples</td>
<td>273</td>
</tr>
<tr>
<td>Karsten Einwich, Peter Schwarz, Christoph Grimm, Christian Meise</td>
<td></td>
</tr>
<tr>
<td>10.1. Introduction</td>
<td>274</td>
</tr>
<tr>
<td>10.2. Modeling and Simulation of Analog Circuits and Systems</td>
<td>275</td>
</tr>
<tr>
<td>10.3. Problem Specific Simulation Techniques</td>
<td>282</td>
</tr>
<tr>
<td>10.4. Overview of SystemC-AMS Extensions</td>
<td>292</td>
</tr>
<tr>
<td>10.5. Conclusions</td>
<td>296</td>
</tr>
<tr>
<td>Chapter 11</td>
<td></td>
</tr>
<tr>
<td>Modeling and Refinement of Mixed-Signal Systems with SystemC</td>
<td>299</td>
</tr>
<tr>
<td>Christoph Grimm</td>
<td></td>
</tr>
<tr>
<td>11.1. Introduction</td>
<td>299</td>
</tr>
<tr>
<td>11.2. Basic Methods and Definitions</td>
<td>301</td>
</tr>
<tr>
<td>11.3. Implementation in SystemC 2.0</td>
<td>305</td>
</tr>
<tr>
<td>11.4. Refinement of Signal Processing Systems</td>
<td>310</td>
</tr>
<tr>
<td>11.5. A Simple Example for Modeling and Refinement</td>
<td>316</td>
</tr>
<tr>
<td>11.6. Conclusions</td>
<td>320</td>
</tr>
<tr>
<td>References</td>
<td>325</td>
</tr>
<tr>
<td>Index</td>
<td>343</td>
</tr>
</tbody>
</table>
Foreword

SystemC started with a vision of being more than a language definition. It also embraced the idea of a community of system level designers and modellers, sharing their ideas and models, and contributing to the evolution of both the language and the methodologies of how to use it. While the Open System C Initiative (OSCI) organisation is able to drive the language definition forward, it is the user community that is making strong contributions to SystemC modelling approaches and methodologies.

When we wrote our book on SystemC1 in 2001–2, this user community was clearly emerging, with a number of very interesting presentations being made by a wide variety of industrial and academic users in major design and design automation conferences, and user group meetings. Most notable among these was the very strong European presence, especially as represented at the very active European Users group organised by Professor Wolfgang Rosenstiel of the University of Tübingen. This group has been holding twice–yearly meetings since early 2000.

This book, *SystemC – Methodologies and Applications*, represents the state of the art in system level design methods and modelling approaches for using SystemC in answering key design questions. The editors of the book, Wolfgang Müller, Wolfgang Rosenstiel and Jürgen Ruf, have done an excellent job in pulling together contributions from leading European groups into the volume.

When we consider what is needed to turn a language such as SystemC into an everyday part of system modelling and design practice, we come up with several requirements:

- Design methodologies for complex systems, involving combinations of hardware and software
- Modelling abstractions to allow creation of more effective system level verification

---

Rigorous language semantics to allow the creation of appropriate design tools and to clarify modelling approaches.

- System level performance analysis approaches.
- Methods for improving designer productivity.
- Extensions to enable heterogeneous system modelling

The appearance of this book is rather timely from an industry perspective. SystemVerilog is emerging as a major standard for higher abstraction in HDL design, modelling and verification. It unifies design and verification capabilities (testbench, assertions, coverage etc.) in a single language and provides an implementation path from higher abstraction to gates. This offers a unique opportunity to build higher abstraction links between SystemC and SystemVerilog and unify transaction level modeling paradigms. Many key concepts of SystemC discussed in this book will facilitate such opportunities.

The eleven chapters in the book nicely match the above requirements. The first three chapters deal with applied design methodologies accompanied by case studies. The first chapter presents a comprehensive methodology for SoC modelling and design. Although complete automation was not possible, there are several advantages gained through the use of this structured design approach. The second chapter addresses the concept of “transaction level modelling” (TLM) and describes a SystemC based TLM modelling and analysis environment, demonstrated on several examples including a multimedia and ARM based platform. Important simulation speedup results are presented through using SystemC based TLM abstractions. Chapter 3 describes a refinement flow from high level SystemC towards implementable ANSI C targeted to an ARC based system. This refinement includes estimation strategies for SystemC based models to allow more optimal SW implementation code. The example used is an OFDM Demodulator.

Chapter four is a change of pace. Here, SystemC simulation semantics are formally defined using distributed abstract state machines. Such a rigorous definition will aid in creating advanced application tools, and determining and defining model interoperability with other languages. Two chapters exploring other aspects of system level modelling with SystemC follow this. Chapter 5 explores the issues of design error modelling and addresses emulation techniques interfacing SystemC to hardware emulators. Chapter 6 examines classical system level performance estimation, with SoC multi-processing architectures for networking applications. Their case study of a TCP/IP packet processing application and multi–processor platform indicate a significant speedup of simulation throughput with very low loss of accuracy, thus supporting higher level design space exploration.

The next three chapters deal with research aspects in the area of system level synthesis – automated transformations between different abstraction levels. A
protocol specification language implemented as a proposed extension library to SystemC is discussed in chapter 7. The authors furthermore elaborate on an automatic translation method between communication abstractions as a means to develop automatic synthesis of protocol controllers. Chapter 8 addresses object-oriented hardware design and synthesis techniques while the following chapter deals with embedded software generation. It targets relatively simple hardware platforms using the eCos operating system as an example, and points to possible improvements in SystemC 3.0 to better support the SW modelling and code generation.

The final two chapters in the book deal with heterogeneous systems modelling, in particular, research carried out in the area of analogue and mixed signal extensions to SystemC. Chapter 11 concentrates on methods for design, simulation and refinement of complex heterogeneous signal processing systems using SystemC, with examples for extensions of SystemC for analogue modelling. It describes in detail the coupling between different design domains, and the refinement of a simple signal processing application.

The scope, depth and range of the work represented in these chapters indicate the vigour and vitality of the European SystemC community and are an excellent harbinger of more methodologies to come in the future.

This book is an extremely valuable addition to the literature on SystemC. Designers, modellers, and system verifiers will all benefit from the lessons taught by all the contributors. In the area of SoC design, SW development is clearly in the critical path. This is where SystemC can play to its strengths enabling designers to rapidly create system level abstractions of algorithms and architectures that have a vital role in system level design and analysis. They also provide the SW designer with an executable model of the SoC platform in order to start SW development and integration as early as possible. The fact that SystemC is based on C++ also helps to pave the road to SW.

This book will also be of great benefit to students and researchers who wish to understand more about how SystemC has been used, how it can be used, the underlying language semantics, and possible future evolution, in step with SystemVerilog. We hope this will just be the first in a long series of books reporting on user experiences and methodologies using SystemC, for the benefit of the entire SystemC community.

Grant Martin
Berkeley
Thorsten Grötker
Aachen

March 2003
Preface

We put great effort into the selection of authors and articles to present a high quality survey on the state of the art in the area of system design with SystemC. Organised into 11 self-contained readings, we selected leading SystemC experts to present their work in the domains of modelling, analysis, and synthesis. The different approaches give a comprehensive overview of SystemC methodologies and applications for HW/SW designs including mixed signal designs. We know that any collection lacks completeness. This collection mainly results from presentations at European SystemC User Group meetings (www-ti.informatik.uni-tuebingen.de/~systemc). We believe that it gives a representative overview of current work in academia and industries and serves as a state–of–the–art reference for SystemC methodologies and application.

Any book could never be written without the help and valuable contributions of many people. First of all we would like to thank Mark de Jongh, Cindy Zitter, and Deborah Doherty from Kluwer who helped us through the process. Many thanks also go to the contributing authors and their great cooperation through the last weeks. For the review of the individual articles and valuable comments, we acknowledge the work of Axel Braun (Tübingen University), Rolf Drechsler (Bremen University), G"orschwin Fey (Bremen University), Uwe Gl"asser (Simon Fraser University), Daniel Große (Bremen University), Prakash Mohan Peranandam (Tübingen University), Achim Rettberg (C–LAB), Axel Siebenborn (Tübingen University), Alain Vachoux (EPFL), as well as many other colleagues from C-LAB, Paderborn University, and Tübingen University.

Wolfgang Müller
Paderborn

Wolfgang Rosenstiel, J"urgen Ruf
T"ubingen

March 2003

***

Wolfgang Müller dedicates this book to Barbara, Maximillian, Philipp, and Tabea.
Wolfgang Rosenstiel dedicates this book to his family and the SystemC community.
J"urgen Ruf dedicates this book to his wife Esther and his children Nellie and Tim.

xiii