High-Level Verification

Sudipta Kundu · Sorin Lerner · Rajesh K. Gupta

High-Level Verification

Methods and Tools for Verification of System-Level Designs

With Chapter 6 contributed by Malay K. Ganai
and Chapter 8 contributed by Zachary Tatlock

Springer

Sudipta Kundu
Synopsys Inc.
NW Thorncroft Dr. 2174
97124 Hillsboro
USA
sudiptakundu@gmail.com

Rajesh K. Gupta
Department of Computer Science
and Engineering
University of California, San Diego
Gillman Drive 9500
92093-0404 La Jolla
USA
rgupta@ucsd.edu

Sorin Lerner
Department of Computer Science
and Engineering
University of California, San Diego
Gillman Drive 9500
92093-0404 La Jolla
USA
lerner@ucsd.edu

ISBN 978-1-4419-9358-8 e-ISBN 978-1-4419-9359-5
DOI 10.1007/978-1-4419-9359-5
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011928550

Preface

Given the growing size and heterogeneity of Systems on Chip (SOC), the design process from initial specification to chip fabrication has become increasingly complex. The growing complexity provides incentive for designers to use high-level languages such as C, SystemC, and SystemVerilog for system-level design. While a major goal of these high-level languages is to enable verification at a higher level of abstraction, allowing early exploration of system-level designs, the focus so far has been on traditional testing techniques such as random testing and scenario-based testing.

This book focuses on the rapidly growing area of high-level verification. We envision a design methodology that relies upon advances in synthesis techniques as well as on incremental refinement of the design process. These refinements can be done manually or through elaboration tools. In this book, we discuss verification of specific properties in designs written using high-level languages as well as checking that the refined implementations are equivalent to their high-level specifications. The novelty of each of these techniques is that they use a combination of formal techniques to do scalable verification of system designs completely automatically.

The verification techniques fall into two categories: (a) methods for verifying properties of high-level designs and (b) methods for verifying that the translation from high-level design to a low-level Register Transfer Language (RTL) design preserves semantics. Taken together, these two parts guarantee that properties verified in the high-level design are preserved through the translation to low-level RTL. By performing verification on the high-level design, where the design description is smaller in size and the design intent information is easier to extract, and then checking that all refinement steps are correct, we describe a hardware development methodology that provides strong and expressive guarantees that are difficult to achieve by directly analyzing the low-level RTL code.

We expect the reader to gain appreciation and knowledge of the recent advances in raising the abstraction for design verification tasks. While the complexity of the problem is not lost on a typical reader, this book would ultimately present a positive outlook on the engineering solutions to the problem that the reader can use in practice.

Acknowledgments

Acknowledgment is not a mere formality but a genuine opportunity to express the sincere thanks to all those; without whose active support and encouragement this book would not have been successful.

This book would not have been possible without contribution from Dr. Malay K. Ganai and Mr. Zachary Tatlock. Dr. Ganai shared his expertise on bounded model checking in Chap. 6 and Mr. Tatlock shared his knowledge on once-and-for-all verification techniques in Chap. 8.

Significant part of this book is based on Dr. Sudipta Kundu's doctoral dissertation work. To this end, we express our thanks to Prof. Ranjit Jhala, Prof. Ingolf Krueger, Prof. Bill Lin, Dr. Yuvraj Agarwal, Federic Doucet, Ross Tate, and Pat Rondon and to all the faculty members, staffs, and graduate students of the Department of Computer Science and Engineering for their continuous help and support.

Our sincere thanks to Chao Wang, Nishant Sinha, Aarti Gupta, and all other members of the System LSI and Software Verification group at NEC Laboratories America for uncountably many interesting discussion and explanation of key concepts in Verification.

We thank Prof. Alan J. Hu for his valuable insights and comments on the initial version of the translation validation part of this book.

We also thank the members of the MESL and Progsys group for their ceaseless effort, constant encouragement, and also for the endless good times during the making of this book.

We also agree that words are not enough to express our indebtedness and gratitude toward our families, to whom we owe every success and achievements of our life. Their constant support and encouragement under all odds has brought us where we stand today.

Contents

Acronyms

ATP	Automated Theorem Prover
BDD	Binary Decision Diagram
BMC	Bounded Model Checking
CCFG	Concurrent Control Flow Graph
CDFG	Control Data Flow Graph
CEGAR	Counter Example Guided Abstraction Refinement
CFG	Control Flow Graph
CSG	Conflict Sub-Graph
CSP	Communicating Sequential Processes
CSSA	Concurrent Static Single Assignment
CTL	Computation Tree Logic
FIFO	First In First Out
FSM	Finite State Machine
FSMD	Finite State Machine with Datapath
GALS	Globally Asynchronous Locally Synchronous
HDL	Hardware Description Language
HLD	High-Level Design
HLS	High-Level Synthesis
HLV	High-Level Verification
HTG	Hierarchical Task Graph
ICs	Integrated Circuits
LLS	Language of Labeled Segments
LTL	Linear Temporal Logic
MAT	Mutually Atomic Transaction
MC	Model Checking
OSCI	Open SystemC Initiative
PEC	Parameterized Equivalence Checking
POR	Partial-Order Reduction
RTL	Register Transfer Level
SAT	SATisfiability
SMC	Symbolic Model Checking
SMT	Satisfiability Modulo Theory
TLM	Transaction Level Modeling
TV	Translation Validation