Advanced Automation in Formal Verification of Processors

Ulrich Kühne
Arbeitsgruppe Rechnerarchitektur
Universität Bremen

Dissertation
zur Erlangung des Grades eines Doktors
der Ingenieurwissenschaften
– Dr.-Ing. –

Vorgelegt im Juli 2009 an der Universität Bremen
Kolloquiumstermin: 25. September 2009

Erster Gutachter:	Prof. Dr. Rolf Drechsler, Universität Bremen
Zweiter Gutachter:	Prof. Dr. Wolfram Büttner, Abstract RT Solutions GmbH, München
Prüfungsausschuss:	Prof. Dr. Hans-Jörg Kreowski, Prof. Dr. Jan Peleska, Dr. Daniel Große, Markus Groß

Berichte aus der Informatik

Ulrich Kühne

Advanced Automation in Formal Verification of Processors

D 46 (Diss. Universität Bremen)

Shaker Verlag
Aachen 2009

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Zugl.: Bremen, Univ., Diss., 2009

Printed in Germany.

ISBN 978-3-8322-8619-4
ISSN 0945-0807

Shaker Verlag GmbH • P.O. BOX 101818 • D-52018 Aachen
Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9
Internet: www.shaker.de • e-mail: info@shaker.de

For Cécile

Abstract

This thesis addresses automation and enhanced user support in functional verification of digital systems with a special focus on processors. Verification is an important issue in the design process of digital systems. In contrast to traditional simulation based methods, formal verification can provide a mathematical correctness proof. But, up to today, these methods are difficult to use in practice, without a strong background in formal techniques. This thesis aims at improving the usability and increasing the productivity of formal hardware verification.

The functional verification of digital systems can be attacked bottom-up or top-down. For an ad hoc verification of single components or simple hardware systems, the specification is formalized step by step in a bottom-up manner. Thereby, the verification engineer is faced with the task to identify the implementation principles of the design and to capture the behavior in temporal logic properties. In this thesis, techniques are presented to speed up this process. This includes an easy to use coverage check that gives feedback about the quality of the written properties. The process of writing the properties can be supported by strengthening properties automatically. Another approach automatically generated properties for a given expected behavior of the design.

When facing the verification of a restricted class of designs, further automation can be achieved in a top-down flow. In particular, for processors, the task is to find a mapping between the instruction set architecture – the abstract programmer's view – and the implementation. In this thesis, it is examined how parts of this process can be automated. The presented approach provides a guided verification flow, starting with a formal architecture description that is then related to the implementation. The use of architectural models in a structured verification flow can increase the productivity significantly. Further applications of such high-level models include the automatic generation of instruction set simulators and the automatic synthesis of embedded software. These issues are also addressed in this thesis.

Zusammenfassung

Diese Arbeit beschäftigt sich mit Automation und erweiterter Benutzeruntertützung in der funktionalen Verifikation von digitalen Sytemen, mit einem Schwerpunkt auf Prozessoren. Die Verifikation ist ein entscheidender Faktor im Entwurfsprozess von Hardware-Systemen. Im Gegensatz zu traditionellen simulativen Methoden kann mittels formaler Verifikation ein mathematischer Korrektheitsbeweis erbracht werden. Allerdings ist der praktische Einsatz diese Techniken bis heute mit Schwierigkeiten verbunden, da sie teilweise erhebliches Vorwissen in formalen Methoden erfordern. Diese Arbeit zielt darauf ab, die Anwendbarkeit formaler Hardware-Verifikation zu verbessern und ihre Produktivität zu erhöhen.

Die funktionale Verifikation kann auf verschiedene Weisen angegangen werden, bottom-up von der Implementierung zur Spezifikation oder top-down von der Spezifikation ausgehend. In der ad hoc Verifikation einzelner Module und einfacher Systeme wird die Spezifikation schrittweise in einem bottom-up Ansatz formalisiert. Dabei ist es die Aufgabe des Verifikations-Ingenieurs, die Funktionsweise der Implementierung zu verstehen und das Verhalten des Systems in temporaler Logik zu beschreiben. In dieser Arbeit werden Techniken vorgestellt, um diesen Prozess zu vereinfachen und zu beschleunigen. Mit einer einfach anzuwendenen Coverage-Technik kann die Qualität eines Eigenschaftssatzes überprüft werden. Das Schreiben der Eigenschaften kann durch automatische Stärkung einzelner Eigenschaften unterstützt werden. Ein weiterer Ansatz beschäftigt sich mit der automatischen Generierung von Eigenschaften, die ein erwartetes Verhalten der Implementierung beschreiben.

Ein höherer Automationsgrad kann bei der funktionalen Verifikation von eingeschränkten Klassen von Systemen erreicht werden. Für Prozessoren besteht die Verifikationsaufgabe darin, eine Abbildung zwischen der Instruktionssatzarchitektur – der abstrakten Sicht des Programmierers – und der Implementierung zu finden. In dieser Arbeit wird untersucht, wie dieser Prozess teilweise automatisiert werden kann. Der entwickelte Ansatz stellt einen werkzeugunterstützten Verifikationsablauf zu Verfügung, ausgehend von einer formalen Architekturbeschreibung. Durch die Nutzung von Architekturmodellen in einem klar strukturierten Ablauf lässt sich die Produktivität der Verifikation erhöhen. Weitere Anwendungen solcher abstrakter Modelle sind die automatische Generierung von Instruktionssatzsimulatoren und die automaische Synthese eingebetteter Software-Programme, welche ebenfalls Gegenstand dieser Arbeit sind.

Danksagung

Mit dieser Arbeit beende ich meine Promotion im Fach Informatik. Dass es so weit kommen konnte, dafür bin einigen Leuten zum Dank verpflichtet.

An erster Stelle danke ich Rolf Drechsler, dem Leiter der Arbeitsgruppe Rechnerarchitektur und Betreuer meiner Promotion. Er hat mich bereits während meines Studiums an der Universität Bremen für die Forschung begeistert und mich seitdem stets in meinen Zielen unterstützt, hat zahlreiche Veröffentlichungen gelesen und korrigiert und mich immer wieder motiviert. Auch den Mitabeitern der Arbeitsgruppe danke ich für die tolle Zusammenarbeit. Mit Daniel Große hatte ich viele fruchtbare Diskussionen, aus denen auch viele gemeiname Arbeiten entstanden sind. Genauso danke ich allen anderen dort, die zur Verlängerung meiner Publikationsliste beigetragen haben: André Sülflow, Görschwin Fey, Robert Wille und Stefan Frehse.

Während meiner Promotion hatte ich die Möglichkeit, einige Zeit in der Stadt München zu verbringen. In Kooperation mit der Firma OneSpin Solutions sind dort große Teile dieser Arbeit entstanden. An dieser Stelle danke ich Wolfram Büttner, der mich dort sehr freundlich aufgenommen hat und der auch einer der Gutachter dieser Arbeit ist. Jörg Bormann danke ich für die Betreuung und seine Verbesserungsvorschläge für meine Dissertation. Auf Grundlage seiner Arbeiten ist die Eigenschaftsgenerierung für Prozessoren entstanden. Ein herzliches Dankeschön geht an Sven Beyer, mit dem ich in München zusammengearbeitet habe und der in Fragen der Verifikation stets Rat wusste und ebenfalls meine Dissertation gelesen hat. Außerdem danke ich Sebastian Skalberg und Christian Pichler, die mich zu Beginn der Kooperation in München in ihre Untersuchungen zur Simulatorgenerierung einbezogen haben.

Vielen Dank schließlich an die Mitglieder des Prüfungsausschusses, die meinem Promotionskolloquium in der Universität Bremen beiwohnten, an Hans-Jörg Kreowski, Jan Peleska, Daniel Große und Markus Groß.

Contents