

Hardware Description Languages and their Applications

Specification, modelling,
verification and synthesis of
microelectronic systems

IFIP TC10 WG10.5 International Conference on
Computer Hardware Description Languages
and their Applications,
20-25 April 1997, Toledo, Spain

Edited by

Carlos Delgado Kloos

*Universidad Carlos III de Madrid
Spain*

and

Eduard Cerny

*Université de Montréal
Canada*

江苏工业学院图书馆
藏书章

Published by Chapman & Hall on behalf of the
International Federation for Information Processing (IFIP)



CHAPMAN & HALL

London · Weinheim · New York · Tokyo · Melbourne · Madras

Published by Chapman & Hall, 2-6 Boundary Row, London SE1 8HN, UK

Chapman & Hall, 2-6 Boundary Row, London SE1 8HN, UK

Chapman & Hall GmbH, Pappelallee 3, 69469 Weinheim, Germany

Chapman & Hall USA, 115 Fifth Avenue, New York, NY 10003, USA

Chapman & Hall Japan, ITP-Japan, Kyowa Building, 3F, 2-2-1 Hirakawacho, Chiyoda-ku, Tokyo 102, Japan

Chapman & Hall Australia, 102 Dodds Street, South Melbourne, Victoria 3205, Australia

Chapman & Hall India, R. Seshadri, 32 Second Main Road, CIT East, Madras 600 035, India

First edition 1997

© 1997 IFIP

Printed in Great Britain by TJ International, Padstow, Cornwall

ISBN 0 412 78810 1

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the UK Copyright Designs and Patents Act, 1988, this publication may not be reproduced, stored, or transmitted, in any form or by any means, without the prior permission in writing of the publishers, or in the case of reprographic reproduction only in accordance with the terms of the licences issued by the Copyright Licensing Agency in the UK, or in accordance with the terms of licences issued by the appropriate Reproduction Rights Organization outside the UK. Enquiries concerning reproduction outside the terms stated here should be sent to the publishers at the London address printed on this page.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

A catalogue record for this book is available from the British Library



Printed on permanent acid-free text paper, manufactured in accordance with ANSI/NISO Z39.48-1992 and ANSI/NISO Z39.48-1984 (Permanence of Paper).

Hardware Description Languages and their Applications

**Specification, modelling, verification and
synthesis of microelectronic systems**

Visit the IT & Applied Computing resource centre
www.IT-CH.com

IFIP – The International Federation for Information Processing

IFIP was founded in 1960 under the auspices of UNESCO, following the First World Computer Congress held in Paris the previous year. An umbrella organization for societies working in information processing, IFIP's aim is two-fold: to support information processing within its member countries and to encourage technology transfer to developing nations. As its mission statement clearly states,

IFIP's mission is to be the leading, truly international, apolitical organization which encourages and assists in the development, exploitation and application of information technology for the benefit of all people.

IFIP is a non-profitmaking organization, run almost solely by 2500 volunteers. It operates through a number of technical committees, which organize events and publications. IFIP's events range from an international congress to local seminars, but the most important are:

- the IFIP World Computer Congress, held every second year;
- open conferences;
- working conferences.

The flagship event is the IFIP World Computer Congress, at which both invited and contributed papers are presented. Contributed papers are rigorously refereed and the rejection rate is high.

As with the Congress, participation in the open conferences is open to all and papers may be invited or submitted. Again, submitted papers are stringently refereed.

The working conferences are structured differently. They are usually run by a working group and attendance is small and by invitation only. Their purpose is to create an atmosphere conducive to innovation and development. Refereeing is less rigorous and papers are subjected to extensive group discussion.

Publications arising from IFIP events vary. The papers presented at the IFIP World Computer Congress and at open conferences are published as conference proceedings, while the results of the working conferences are often published as collections of selected and edited papers.

Any national society whose primary activity is in information may apply to become a full member of IFIP, although full membership is restricted to one society per country. Full members are entitled to vote at the annual General Assembly, National societies preferring a less committed involvement may apply for associate or corresponding membership. Associate members enjoy the same benefits as full members, but without voting rights. Corresponding members are not represented in IFIP bodies. Affiliated membership is open to non-national societies, and individual and honorary membership schemes are also offered.

Preface

1 PRESENTATION

This book contains the papers that have been accepted for presentation at the XIII IFIP WG 10.5 International Conference on *Computer Hardware Description Languages and Their Applications 1997*, or CHDL'97 for short, in Toledo, 21–23 April 1997. CHDL has been held every other year since 1973, rotating between locations in Europe, North America and Asia. The conference originated under IEEE/ACM sponsorship and since 1981 has been organized by IFIP Working Group 10.2 (now WG 10.5). Previously, the conference has been held at the following locations:

- 1973: New Brunswick (New Jersey, USA)
- 1974: Darmstadt (Germany)
- 1975: New York (New York, USA)
- 1976: Palo Alto (California, USA)
- 1981: Kaiserslautern (Germany)
- 1983: Pittsburgh (Pennsylvania, USA)
- 1985: Tokyo (Japan)
- 1987: Amsterdam (The Netherlands)
- 1989: Washington (D.C., USA)
- 1991: Marseille (France)
- 1993: Ottawa (Canada)
- 1995: Chiba (Japan)

The CHDL topics have a significant history with over 100 HDLs being published in the 1970s. Since the mid-1980s, HDLs have become commonplace in system design and VLSI. Presently, we are in a consolidation phase, in which languages and standards are increasingly being used, at the same time as the scope is being broadened to include CAD tools and additional application areas (such as analog, microwave or system-level design).

Altogether 42 papers have been received, maintaining a very high average quality. The topics ranged from Specification to Implementation, including Verification and Synthesis. Out of these papers, 17 were selected for presentation and publication, arranged according to the following session topics:

- Specification and Design of Reactive Systems
- Verification Using Model Checking Techniques
- Formal Characterizations of Systems
- Analog Languages

- Languages in Design Flows
- Future Trends in Hardware Design
- HDLs for the XXI Century
- Formal Methods for Asynchronous and Distributed Systems

Furthermore, there are three invited presentations, the abstract of which appear here. Finally, the content of seven poster presentations is included in abbreviated form in this book. Two panel sessions have been scheduled, about *The Next HDL Paradigms* and *Analog and Mixed-Signal HDLs*.

CHDL'97 is held at the same location and dates as other workshops on closely related areas:

- Spring '97 Conference of the VHDL Users' Forum in Europe
- 2nd Workshop on Libraries, Component Modelling and Quality Assurance
- Esprit NADA workshop (about New Hardware Design Methods).

The first two are more practically oriented, whereas the third is of more theoretical nature, as compared with CHDL. Nevertheless, there are no clear boundaries, which is also shown by the fact that two papers accepted for CHDL have been placed in sessions together with papers of the VHDL Users' Forum. At any rate, we hope that this spectrum of offerings has enriched the perspectives of the participants to the combined event. And in the same way as in Toledo several cultures have coexisted, the Christian, the Jewish and the Arabic, leaving a wealth of buildings for us to admire, we hope that the multi-event in April 97 in Toledo will be of profit to different cultures and perspectives in the area of HDLs and give rise to insightful findings.

We would like to thank all who made this conference possible. First, IFIP, and in particular, TC10 and WG 10.5, who sponsored this event. Then, all the members of the programme committee (most of them from WG 10.5), who together with additional reviewers did an excellent job for the paper selection. Furthermore, we thank the members of the steering and organizing committees for doing their job with enthusiasm. We want also to express our recognition to the fine relationship maintained with the organizers of the parallel events. In particular, Eugenio Villar from the University of Cantabria is to be thanked for his co-operation in solving the problems that arose during the organization of the events. The authors of submitted papers, tutorialists, panelists and session chairs are to be thanked for their contribution. GRIAO (Université de Montréal) helped by providing a machine and the part-time help of an analyst. Finally, we would like to gratefully acknowledge the generous support provided by the *Universidad Carlos III de Madrid* and by the Spanish *Comisión Interministerial de Ciencia y Tecnología*.

Carlos Delgado Kloos
Leganés (Madrid/Spain)

January 1997

Ed Cerny
Montréal (Canada)

2 COMMITTEES

2.1 Steering Committee

- **General Chair:**
Carlos Delgado Kloos (Universidad Carlos III de Madrid, Spain)
- **Programme Chair:**
Eduard Cerny (Université de Montréal, Canada)
- **Tutorial Chair:**
Przemyslaw Bakowski (IRESTE, France)
- **Asia-Pacific Representative:**
Masaharu Imai (Osaka University, Japan)
- **Exhibition Chair:**
Serafin Olcoz (SIDSA, Spain)

2.2 Programme Committee

- | | |
|---------------------------------|-------------------------------------|
| ● David Agnew, Canada | ● François Anceau, France |
| ● Przemyslaw Bakowski, France | ● Mario Barbacci, USA |
| ● Howard Barringer, UK | ● Graham Birtwistle, UK |
| ● Dominique Borrione, France | ● Raul Camposano, USA |
| ● Eduard Cerny, Canada | ● Edmund Clarke, USA |
| ● Francisco Corella, USA | ● Werner Damm, Germany |
| ● Carlos Delgado Kloos, Spain | ● Nikil Dutt, USA |
| ● Hans Eveking, Germany | ● Norbert Fristacky, Czech Republic |
| ● Masahiro Fujita, USA | ● Ganesh Gopalakrishnan, USA |
| ● Werner Grass, Germany | ● Reiner Hartenstein, Germany |
| ● Graham Hellestrand, Australia | ● Masaharu Imai, Japan |
| ● Steven Johnson, USA | ● Thomas Kropf, Germany |
| ● David Luckham, USA | ● Paul Menchini, USA |
| ● Jean Mermet, France | ● Wolfgang Nebel, Germany |
| ● Adam Pawlak, France | ● Franz Rammig, Germany |
| ● Peter Schwarz, Germany | ● Jørgen Staunstrup, Denmark |
| ● Flavio Wagner, Brazil | ● Ronald Waxman, USA |
| ● Akihiko Yamada, Japan | ● Michael Yoeli, Israel |

2.3 Reviewers

We would like to thank:

David Agnew, François Anceau, Przemyslaw Bakowski, Mario Barbacci, Howard Barringer, S. Berezin, Bachir Berkane, Graham Birtwistle, Dominique Borrione, Peter T. Breuer, S. Campos, Raul Camposano, Eduard Cerny,

Ching-Tsun Chou, Edmund Clarke, Francisco Corella, Werner Damm, Carlos Delgado Kloos, Nikil Dutt, Hans Eveking, Norbert Fristacky, Masahiro Fujita, Ganesh Gopalakrishnan, Werner Grass, Reiner Hartenstein, Graham Hellestrand, Teruo Higashino, Masaharu Imai, Steven Johnson, Thomas Kropf, David Luckham, Andrés Marín López, Natividad Martínez Madrid, Paul Menchini, Jean Mermet, M. Minea, Wolfgang Nebel, Adam Pawlak, Sreeranga P. Rajan, Franz Rammig, Luis Sánchez Fernández, Peter Schwarz, Jørgen Staunstrup, Kazuko Takahashi, Flavio Wagner, Ronald Waxman, Akihiko Yamada, Ying Xu, Michael Yoeli, J. Zejda

... and all other students and colleagues whose names did not come to our attention and who helped with the reviewing process.

2.4 Organizing Committee

- Laura Acebes García, Univ. Carlos III de Madrid
- Peter T. Breuer, Univ. Carlos III de Madrid
- Carlos Delgado Kloos, Univ. Carlos III de Madrid
- Salvador López Mendoza, Univ. Carlos III de Madrid
- Andrés Marín López, Univ. Carlos III de Madrid
- Natividad Martínez Madrid, Univ. Carlos III de Madrid
- Luis Sánchez Fernández, Univ. Politécnica de Madrid
- Aurora Sánchez Garrido, Univ. Carlos III de Madrid

2.5 Financial support

- Universidad Carlos III de Madrid
- Comisión Interministerial de Ciencia y Tecnología

CONTENTS

Preface	vii
PART ONE Specification and Design of Reactive Systems	1
1 Synchronous languages for hardware and software reactive systems <i>G. Berry</i>	3
2 Towards a complete design method for embedded systems using Predicate/Transition-Nets <i>B. Kleinjohann, J. Tacke and C. Tahedl</i>	4
PART TWO Verification Using Model Checking Techniques (+ Poster Abstracts)	25
3 Simplifying data operations for formal verification <i>F. Balarin and K. Sajid</i>	27
4 CTL and equivalent sublanguages of CTL <i>K. Schneider</i>	40
5 Verifying linear temporal properties of data intensive controllers using finite instantiations <i>R. Hojati, D. Dill and R.K. Brayton</i>	60
6 A high-level language for programming complex temporal behaviors and its translation into synchronous circuits (poster abstract) <i>C.-T. Chou, J.-L. Huang and M. Fujita</i>	74
7 System-level hardware design with μ -charts (poster abstract) <i>J. Philipps and P. Scholz</i>	77
8 Interface synthesis in embedded hardware-software systems (poster abstract) <i>M. Auguin, C. Belleudy and G. Gogniat</i>	80
9 TripleS-a formal validation environment for functional specifications (poster abstract) <i>J.-P. Soininen, J. Saarikettu, V. Veijalainen and T. Huttunen</i>	83
10 SOFHIA: a CAD environment to design digital control systems (poster abstract) <i>R.J. Machado, J.M. Fernandes and A.J. Proença</i>	86
11 Compiling the language BALSA to delay insensitive hardware (poster abstract) <i>A. Bardsley and D. Edwards</i>	89
12 High-level synthesis of structured data paths (poster abstract) <i>C. Mandal and R.M. Zimmer</i>	92
PART THREE Formal Characterizations of Systems	95
13 Characterizing a portable subset of behavioural VHDL-93 <i>K. Thirunarayan and R. Ewing</i>	97

14 Algebra of communicating timing charts for describing and verifying hardware interfaces <i>B. Berkane, S. Gandrabur and E. Cerny</i>	114
15 A formal proof of absence of deadlock for any acyclic network of PCI buses <i>F. Corella, R. Shaw and C. Zhang</i>	134
PART FOUR Analog Languages	157
16 Behavioural modelling of sampled-data with HDL-A and ABSynth <i>V. Moser, H.-P. Amann and F. Pellandini</i>	159
PART FIVE Languages in Design Flows	179
17 Hardware description languages in practical design flows <i>R. Camposano</i>	181
18 VHDL generation from SDL specification <i>J.-M. Daveau, G. Fernandes Marchioro and A.A. Jerraya</i>	182
19 Exploiting isomorphism for speeding up instance-binding in an integrated scheduling allocation and assignment approach to architectural synthesis <i>B. Landwehr, P. Marwedel, I. Markhof and R. Dömer</i>	202
PART SIX Future Trends in Hardware Design	213
20 Verification of large systems in silicon (special talk) <i>C. Ussery and S. Curry</i>	215
21 The Shall Design test Development model for hardware systems <i>M. Heuchling, W. Ecker and M. Mrva</i>	240
22 Modular operational semantic specification of transport triggered architectures <i>J. Mountjoy, P. Hartel and H. Corporaal</i>	260
PART SEVEN Formal Methods for Asynchronous and Distributed Systems	281
23 The world of I/O: a rich application area for formal methods (invited talk) <i>F. Corella</i>	283
24 Abstract modelling of asynchronous micropipeline systems using Rainbow <i>H. Barringer, D. Fellows, G. Gough and A. Williams</i>	285
25 A new partial order reduction algorithm for concurrent system verification (short talk) <i>R. Nalumasu and G. Gopalakrishnan</i>	305
PART EIGHT VHDL	315
26 VHDL power simulator: power analysis at gate level <i>L. Kruse, D. Rabe and W. Nebel</i>	317
27 Object oriented extensions to VHDL. the LaMI proposal <i>J. Benzakki and B. Djafri</i>	334
Index of contributors	348
Keyword index	349

PART ONE

Specification and Design of Reactive Systems

Synchronous Languages for Hardware and Software Reactive Systems

Gerard Berry

Ecole des Mines de Paris

B.P. 207, F-06904 Sophia-Antipolis CDX, France, berry@cma.cma.fr

Abstract

Synchronous languages are dedicated to hardware, software, or mixed reactive systems that maintain a continuous interaction with their environment. They come in two classes: data-oriented languages such as Lustre and Signal, which are targeted to data-intensive applications such as continuous control or signal processing, and control-oriented languages or visual formalisms such as Esterel, Statecharts, or Argos, which are tailored for discrete controllers. Data-oriented languages are equational and involve combinational or delayed operations over multiply-clocked signals. Control-oriented languages have explicit primitives for sequencing, concurrency, and process preemption.

The synchronous languages are defined by precise mathematical semantics based on a simple zero-delay model, unlike most HDLs that are based on more intricate simulation models. The semantics makes it precise what a program means, how it can be compiled, and how to verify its properties.

We analyze the synchronous programming primitives and we present the semantical framework. We show how to synthesize RTL-level circuits from synchronous programs and how to use circuit synthesis and optimization techniques to produce efficient software codes as well. We present verification techniques and algorithms for synchronous programs. We discuss combinational cycles in synchronous programs, and we exactly characterize those that have a synchronous behavior. Finally, we present some industrial applications of synchronous languages.

More detailed information can be obtained from the Esterel web page at <http://www.inria.fr/meije/esterel> or by sending an email message to <esterel-request@cma.inria.fr>.

Keywords

Synchronous languages, reactive systems

Towards a Complete Design Method for Embedded Systems Using Predicate/Transition-Nets*

B. Kleinjohann

C-LAB

Fürstenallee 11, 33102 Paderborn, Germany, bernd@c-lab.de

J. Tacke

C-LAB

Fürstenallee 11, 33102 Paderborn, Germany, theo@c-lab.de

C. Tahedl

Heinz Nixdorf Institut

Fürstenallee 11, 33102 Paderborn, Germany, tahedl@uni-paderborn.de

Abstract

In this paper, we present a new approach to embedded system design based on modeling discrete and also continuous system parts with high level Petri-Nets. Our investigations concentrate on a complete design flow, analysis on high level Petri-Nets and their meaning for hardware/software partitioning of real-time embedded systems. The concepts for hybrid modeling of discrete and continuous systems are applied in an example in the domain of mechatronic systems.

Keywords

Design Method, Embedded Systems Design, Hardware/Software-Codesign, High Level Petri Nets

1 INTRODUCTION

In recent years embedded systems have gained increasing importance. This development is rooted in the growing number of systems containing embedded systems. Simultaneously the focus in design has shifted towards criteria requiring a common design process of the whole system. While classical hardware/software-codesign just considered optimizing the contrary aims of maximizing speed and minimizing costs, thinking about the fulfillment of hard real-time restrictions, minimization of the absolute size and weight of the assembly group, reduction of energy-consumption, and

*Supported by DFG-Sonderforschungsbereich 376 "Massive Parallelität: Algorithmen, Entwurfsmethoden, Anwendungen"

especially safety and reliability are getting more important now. Especially for the systems' reliability it is important to consider not only each single component on its own but its behavior within the whole context. Many functional errors only expose themselves when all individual components work together in the whole context, observed over time. The individual components are typically partitioned into hardware components (ASICs, FPGAs) and software components (parallel real-time software on one or several general purpose processors) in order to ensure a realization fulfilling all requirements.

When modeling embedded systems it is important to handle concurrency. As shown in (Dittrich 1994) Petri-Nets are very well suited for the specification of concurrent systems. To handle the complexity they use hierarchical Petri-Nets as specification language in order to support hardware/software-codesign. In our work we also decided to use an extended form of Petri-Nets, namely Extended Predicate/Transition-Nets (Extended Pr/T-Nets). Our work differs from their approach as we use Petri-Nets that support the notion of time and have a more powerful hierarchy semantic. We additionally consider verification and partitioning methods using the known methods for formal analysis of Petri-Nets.

Often embedded systems do not only contain discrete but also continuous system parts. In order to model those parts with a continuous behavior we transform them into a form that can be denoted by the event-based Petri-Net semantics. This transformation is basically a discretization described in (Briellmann 1995). In contrast to other methods that contain continuous time control, e.g., (Grimm *et al.* 1996), this allows us to analyze properties for hardware/software-partitioning purposes in one common model. Nevertheless a discretization means a loss of information in the model. For our purposes this disadvantage is not relevant as analysis or optimizations that need the original continuous informations (e.g. differential equations) are done on a higher level and an earlier design step. Using a combined graph-based model for analog and digital system parts as described in (Grimm *et al.* 1996) has the disadvantage that the analyzing facilities are limited to structural properties of the graph.

In (Tanir *et al.* 1995) Pr/T-Nets are used for analysis purposes either, but do not serve as a common model. The final implementation of the system is not compiled from the Pr/T-Net model but from their common specification language DSL.

In (Esser 1996) object oriented Petri-Nets are used as a common model for hardware/software-codesign. For this type of nets it will be very hard to develop analysis methods. Actually it is possible to simulate object oriented Petri-Nets for design validation.

As an example for an embedded system with discrete and continuous system parts we use a decentralized traffic management system, based on communicating autonomous units. Within the continuous system parts we have to deal with controllers for certain properties of the vehicle. Discrete system parts are responsible for event-oriented decisions within each vehicle and for communication between the vehicles to organize crossroad management. When considering each subsystem in isolation, the following problems would arise:

Coupling. By now every single area of application has developed its own well understood techniques for modeling, combined with corresponding methods and tools for analysis and simulation. Already at this level many predictions about the temporal and functional behavior of each subsystem can be made. To validate the behavior of the whole system, the individual models have to be coupled. Often the models are given in different domain-specific modeling languages, so coupling can only be realized either on the level of simulation or within a hybrid language, that usually does not offer any facilities for further analysis. Coupling of simulators allows a hybrid simulation in the sense of a combined simulation of all subsystems, while each subsystem may be formulated in a different language for a specific simulator. On the level of simulation temporal and functional behavior and performance can be studied and validated. But not all errors can be found by simulation because of the exhaustive number of possible simulation runs. Hence it is desirable to run a formal analysis of the static and dynamic properties for formal verification purposes. The prerequisite for this kind of analysis is that all models are given in an uniform language.

Analysis techniques. For every new modeling language the corresponding formal analysis methods always have to be re-developed and re-implemented. They are mostly based on classical analysis methods, e.g., the well-known ones for Petri-Nets, and represent an extension or adaptation to the specific areas of application. In order to avoid the expenditure of deducing new analysis techniques and to reuse existing methods, it is apparently more effective to use a known modeling language for the analysis that offers the desired methods suitable for the tasks at hand. So, we decided to use a special form of high-level Petri-Nets in our approach.

Continuous Systems. A lot of methods that have evolved in the area of hardware/software-codesign only consider discrete parts when partitioning the system into hardware and software components. As we can see in our application example continuous parts, e.g., controllers of continuous behavior, play an important role, too. For the purpose of performance estimations or formal verification it is necessary to have a common model for the discrete and continuous system parts. Thereby it is sufficient to transform continuous parts into a discrete form on an implementation-like level. On this level temporal and structural properties of these components are still existent.

Global Optimization. A further problem in a separated design process of different subsystems is that each individual domain has proprietary methods of optimization, but no facility for a global optimization of the joint system exists. Especially when coupling different subsystems it may be useful and more cost-effective to export some functionality from one into another subsystem. An overall analysis of the joint system may reveal states that can never be reached and therefore can be eliminated from the design.

In the following sections we will give a brief introduction to our common formal model and an overview of our proposed design flow. We then describe strategies for hardware/software partitioning and formal analysis methods. Our application example shows how to model a hybrid system using extended Pr/T-Nets and how to apply some analysis methods for verification purposes.