Bernd Becker, Randal Bryant,
Christoph Meinel (editors):

Computer Aided Design and Test

Dagstuhl-Seminar-Report; 56
15.02.-19.02.93 (9307)

ISSN 0940-1121

Tel.: +49-6871 - 2458
Fax: +49-6871 - 5942

Das Internationale Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Bezugsadresse: Geschäftsstelle Schloss Dagstuhl
Universität des Saarlandes
Postfach 1150
D-66041 Saarbrücken, Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: office@dag.uni-sb.de

INTERNATIONALES BEGEGNUNGS- UND FORSCHUNGSZENTRUM FÜR INFORMATIK
Schloß Dagstuhl

Dagstuhl-Seminar-Report

Computer Aided Design and Test

14.02.-19.02.1993

The second workshop on "Computer Aided Design and Test" at the IFBI was organized by Bernd Becker (Frankfurt), Randal E. Bryant (Carnegie Mellon) and Christoph Meinel (Trier). There were 36 participants from 5 countries. The organizers took the chance to bring together research groups from different areas in computer science, electrical engeneering and industry. The common interest of all participants were problems related to the field of electronic design automation with special emphasis on research done in the area of (high level and logic) synthesis, verification and testing.

Twenty-three lectures were presented at the workshop covering the following topics:

- High level synthesis and timing
- Verification and timing
- OBDDs and OBDD-based algorithms
- generalizations of OBDDs
- Delay fault testing and synthesis
- Testable (fault tolerant) design
- Random testability
- Fault simulation and test generation

The participants are very grateful for this opportunity to meet in such an excellent atmosphere that provided ample time for fruitful discussions, be it on new ideas and future developments, or on the differing viewpoints of e.g. a practitioner and a theoretical computer scientist.

Contents

Rademacher-Walsh Coefficients and BDD Based Data Structures

F. Bauernöppel
Humboldt - University, Berlin

Computing the Rademacher-Walsh Spectrum of a Boolean function is a prerequisite for symmetry based logic synthesis. Therefore, it is important to have efficient algorithms for computing RW coefficients of Boolean functions directly from their internal representation in a logic synthesis tool. We show that RW coefficients can be computed efficiently from a BP1 branching program, and give evidence that no such algorithm does exist for (non-disjoint) cute representations of the Boolean function. Finally, we compute the full spectrum of n-bit adders, using the above techniques. It is shown that there are exactly $c_n = 2c_{n-1} + 2$ non-zero coefficients ($c_1 = 4$). Similar results are shown for the number of coefficients that are non-zero for certain output bits.

(Problems to Obtain) First Experimental Results with BP1s

B. Becker **R. Krieger** **K.-J. Englert**
J.W.G. - University, Frankfurt

We study the problem to obtain a "good" BP1-representation for a given Boolean circuit C. For a given type a type-synthesis strategy is proposed to keep the size of the BP1s obtained during the construction process small. A first simple heuristic is given to define a tree-like type for a circuit C. A BP1- package generalized from the Bryant-BDD-package is used to obtain first encouraging results for some of the benchmark-circuits by application of the above mentioned methods.

Retiming and High-Level Synthesis

R. Camposano
GMD mbH, St. Augustin

Retiming optimizes a design moving registers (storing elements) through combinational logic. This technique has received a lot of attention lately and has been combined, for example, with logic synthesis. But retiming has not had a broad impact in practice yet, mainly for two reasons. On the one hand, finite state machines of reasonable size can be optimized using more powerful techniques that rely on state enumeration such as state minimization and encoding; the position of registers in finite state machines is mostly 'good' already in the first design. On the other hand, registers in data paths are often loaded conditionally (not in every cycle), which impedes straight-forward retiming. This presentation shows how to optimize data paths with conditionally loaded registers by combining retiming with high-level synthesis. We first introduce an extended model for retiming and derive the exact conditions which allow retiming with conditionally loaded registers. We then define high-level transformations such as bus-replication and schedule-compression. These transformations together with retiming allow the high-level optimization of a design. The presented techniques are illustrated with two examples and results down to the logic level are given.

Improved Test Generation for Multiple Faults in Programmable Logic Arrays

Yongzhang Chen
University of Karlsruhe

A method which generates tests for multiple crosspoint faults in PLAs has been realized through both extensively analysis and optimum utilization of dominance relations and partial dominance relations among these faults. For redundant as well as irredundant PLAs, complete test sets for multiple missing crosspoint faults can be generated at an expense which is close to that for single faults. The resulting test sets are compact, nevertheless all multiple as well

as single missing crosspoint faults are detected. This algorithm has been implemented. Running on an Apple MAC II, tests for 51 well-known benchmark PLAs from Berkeley have been generated. For some of the benchmark PLAs obviously improved results have been achieved compared with other well-known results in the literature.

Rapid Prototyping of Robust Path-Delay-Fault Testable Circuits Derived from Binary Decision Diagrams

B. Becker **R. Drechsler**
J.W.G. - University, Frankfurt

We investigate the testability properties of Boolean circuits derived from Binary Decision Diagrams using the robust path-delay-fault model. We present a method to obtain fully testable circuits by extra inputs. These circuits can easily be mapped to multiplexer based field programmable gate arrays. Thus, it is possible to generate a fully testable prototype from the BDD-description of a Boolean function. The resulting circuit is not larger in size than the BDD-description.

Formal Verification of Timing Conditions

H. Eveking
J.W.G. - University, Frankfurt

We consider signal-expressions consisting of signals and a number of basic operators like delay, edge-detection, stability. Fundamental properties like the orthogonality of the delay- and other operators lead to a number of basic equalities between signal-expressions. It is shown how timing conditions like set-up and hold time conditions as well as basic circuits like edge-triggered flipflops and gates are described in SMAX, a small and axiomatized HDL. Signal-expressions are included in the language. Formal verification of timing conditions is discussed in terms of the transformation of internal into external timing conditions of a system. The derivation of the clock-period and clock-sheer conditions of a system from its internal timing conditions serves as an example. The method is not restricted to synchronous systems. The false path problem is discussed in this framework. Finally, some results using symbolic model-checking for the verification of small circuits with transport or inertial delays are given.

Exploiting Symmetry in Temporal Logic Model Checking

E. M. Clarke **T. Filkorn** **S. Jha**

Very often, reactive systems occuring in practice exhibit a considerable amount of symmetry. In this paper, we give techniques to reduce the state space searched during symbolic model checking in the presence of symmetry. Exploiting symmetry has been studied in the context of model checking with explicit state representation, but because BDDs are used during symbolic model checking additional complications arise. In the paper we have tried to formalize the notion of symmetry and identify problems which arise while using symmetry during symbolic model checking. We also studied the complexity of various basic steps, like the computation of the orbit relation, which are encountered while using symmetry during symbolic model checking. We also tested our ideas on a simple cache-coherency protocol based on the Futurebus+ IEEE standard. Finally, we have identified problems whose solution is crucial in exploiting symmetry.

Systems Test and Validation

W. Geisselhardt
University of Duisburg

A procedure was presented to derive tests to validate a system at various levels of abstraction the design is stepping through. It is based on a VHDL description of the system at its respective level. These descriptions start behavioral at the system level, turning more and more structural reaching lower levels such as Chip- and RT-level. Our test generation procedure makes use of the control flow and data flow, the graphs of which are derived for internal use. From these a structural test of the VHDL-program describing the systems behavior is derived using symbolic execution. These tests serve the designer (1) to validate the model, the basis of further design steps, and (2) to exercise the real system. What a hierarchical description looks like and how the test generation works, is explained by an example, the I8251 USART. This comparatively small example was used to keep things managable, it does not mean a limitation in size or complexity. The procedure has been implemented by the program FunTestIC.

Feasible Boolean Manipulation with Read-Once-Only Branching Programs (FBDDs)

J. Gergov **Ch. Meinel**
University of Trier

A central issue in the solution of many computer aided design problems is to find concise representations for circuit designs and their functional specifications. OBDDs introduced by Bryant recently emerged as a popular representation for various CAD applications. However, there exist many functions of practical interest (e.g. indirect storage access function (ISA), hidden weighted bit function) which provably require OBDDs of exponential size but can be represented efficiently ($O(n^2)$ size) with FBDDs. We prove that instead of OBDDs one can efficiently work with FBDDs, too. In particular, we generalize the linear ordering of the OBDDs to types in the case of FBDDs and show

that each Boolean function has a canonical representation as reduced FBDD of complete type τ. Further, we show how the OBDD manipulation algorithms can be applied to FBDDs.

The Fault Graph and Its Application to Combinational Fault Simulation

R. Hahn B. Becker H. Hengster
J.W.G. - University, Frankfurt

The fault graph is a data structure to store faults and to manage relations between faults. It allows to exploit these relations for testing purposes and to control the order of faults considered. To demonstrate the effectiveness of the fault graph we developed an algorithm for fault simulation in combinational circuits that is based on this data structure. Using the fault graph the algorithm is very simple. Experiments show that it is superior to a sophisticated fanout oriented fault simulation algorithm.

On Numerical Weight Optimization for Random Testing

J. Hartmann
University of Saarbrücken

Pseudo-random testing has become a wide-spread technique to implement self-tests. To make this method usable for circuits containing random pattern resistant faults, input probabilities can be weighted. In this talk, new results on numerical weight optimization are presented. First, two new cost functions are

derived which are aimed at minimizing the expected test length and at maximizing the expected fault coverage, respectively. In the second part, optimizations based on precomputed tests are considered. Applying numerical methods to such optimizations yields a new method which improves existing strategies.

The Formalization of an HDL

W. A. Hunt
Computational Logic, Inc.

We have used the Boyer-Moore logic to formalize a Hardware Description Language (HDL). The semantics of the HDL are given as an embedding of the HDL in the Boyer-Moore logic. We have defined a predicate that recognized well- formed circuit descriptions. A simulator has been defined that gives a four- valued logical value to each circuit signal. We have used this HDL to describe an implementation of the FM9001 microprocessor, which we have proved to be correct with respect to a behavioral specification.

A BDD-Based Algorithm for Computation of Exact Fault Detection Probabilities

R. Krieger **B. Becker** **R. Sinković**
J.W.G. - University, Frankfurt

Signal and fault detection probabilities are widely used in the area of testing. Due to the computational complexity, in most cases only approximated values are computed. In the talk, we describe a system called PLATO (Probalistic Logic Analyzing Tool) which allows the computation of the exact values for many

combinational circuits. The implemented algorithms use the recently developed BDD-packages as data structure. Beside the description of our algorithm we pay attention to general problems arising with the use of BDDs as data structure. Some new heuristics (decomposition in supergates, Shannon decomposition of combinational circuits) are presented to deal with these problems.

Synthesis of Fanout-Free Functions

R. Kolla
University of Bonn

The problem of timing driven synthesis of Boolean functions for VLSI-cell libraries is unsolved, at least if one is interested in exact methods. A popular approach is to synthesize a technology independent circuit first and then to use tree matching techniques for the fanout free parts of this circuit. We show that for fanout free parts, the synthesis of the associated fanout free function is feasible, at least for some simplifications. The key problem is the problem to synthesize a big fanin. We show that there is an exact but exponential time algorithm if we assume a unit delay model. We hope that this algorithm will be tractable for practical problem sizes. Generalization to more realistic delay models and to tractable approximative methods are pointed out. Implementation and practical experiments for benchmark circuits are current work.

Frontiers of Feasible Boolean Manipulation in Terms of Branching Programs (BDDs)

Ch. Meinel
University of Trier

We investigate the question whether and to what extend the solution of central tasks of digital logic circuit design of a given Boolean function f benefits from a representation of f in terms of certain restricted branching programs. On the basis of this research we purpose the use of read-once-only Branching Programs (BP1) instead of the often much more extensive OBDD-representations. First experimental evidence of the power of this approach is given by means of a BP1-package that has been developed by modification of Bryants OBDD-package.

Detection of Symmetry of a Boolean Function using OBBDs

D. Möller **M. Weber**
Humboldt - University, Berlin

We present a new approach to detect symmetry of combinational functions represented as Reduced Ordered Binary Decision Diagrams (ROBDD). It is shown that the structure of ROBDDs representing symmetric functions is of a special kind. Using this knowledge we develop two heuristics to determine pairs of variables which cannot be symmetric. Information about asymmetry is used to achieve faster algorithms to detect symmetry.

Delay Fault Coverage, Test Set Size, and Performance Tradeoffs

W. K. Lam A. Saldanha R. K. Brayton
A. L. Sangiovanni-Vincentelli
University of California - Berkeley CA

The main disadvantage of the path delay fault model is that to achive 100% testability every path must be tested. Since the number of paths is usually exponential in circuit size, this implies that analysis and synthesis techniques for 100% path delay fault testability are infeasible on most circuits. We prove that 100% delay fault testability is not necessary to guarantee the speed of a combinational circuit. There exist path delay faults which can never impact the circuit delay (computed using any correct timing analysis method) unless some other path delay faults also affect it. These are termed robust dependent delay faults and need not be considered in delay fault testing. Necessary and sufficient conditions under which a set of path delay faults is robust dependent are proved; this yields more accurate and larger delay fault coverage estimates than previously used. Next, assuming only the existence of robust delay fault tests for a very small set of paths (linear in circuit size), we show how the circuit speed (clock period) can be selected such that 100% robust delay fault coverage is achieved. This leads to a quantitative tradeoff between the testing effort (test set size) for a circuit and the verifiability of its performance. Finally, under a bounded delay model, we show that the test set size can be reduced while maintaining the delay fault coverage for the specified circuit speed. Examples and experimental results are given to show the effect of these three techniques on the amount of delay fault testing necessary to guarantee correct operation.

Graph Driven BDDs - A New Data Structure for Boolean Functions

D. Sieling **I. Wegener**
University of Dortmund

Data structures for Boolean functions should admit a concise representation of a large class of Boolean functions and efficient algorithms on this representation. We extend the well-known OBDD-data-structure by introducing generalized variable orderings which allow in contrast to OBDDs different variable orderings for different inputs. A generalized variable ordering can be described by a so-called oracle graph and it can be shown that the representation of Boolean functions by the new data structure, which we call LBDDs, is unique up to isomorphisms for each Boolean function, if we fix the oracle graph. LBDDs have the same expressive power as read-once Branching Programs and the most important operations on Boolean functions can be performed on LBDDs almost as efficiently as on OBDDs. Algorithms for the reduction and the synthesis problem are presented. The time complexity of the synthesis algorithm can be reduced, if we use a more structured type of graph driven BDDs which we call WBDDs.

Does ITE Work for BP1s?

A. Slobodová
University of Trier

OBDDs introduced by Bryant are very restricted Binary Decision Diagrams (BDDs) widely used in CAD-applications. Unfortunatelly, they are not concise enough to represent all functions of practical interest in a managable size. Gergov and Meinel introduced a notion of BP1-type that is a generalization of the ordering of tested variables to a graph. This approach yields a new data structure for Boolean function manipulation, that is provably more compact than OBDDs and simultaneously this representation preserves all desired features of OBDDs (existence of the canonical form that implies a good equivalence test,

feasible synthesis and evaluation ...). We explain in detail how the corresponding algorithms for BP1s work and give a hint to an implementation. The last part of the talk is devoted to the modifications of the OBDD-package (of Bryant and Brace) we have done in order to obtain a BP1-package. The crucial is the modification of the if-then-else operator (ITE) that is a building block of the OBDD-package.

On the Check Base Selection Problem for Fast Adders

U. Sparmann
University of Saarbrücken

To achieve on-line error detection by residue codes in arithmetic circuits, a number X is encoded as a tuple consisting of X and its residue modulo a check base b. The check base selection problem consists in choosing b such that the most important errors can be detected, and b is as small as possible in order to minimize the area overhead induced by error detection. This problem has been investigated for the powerful class of Parallel Prefix Computation adders which is well suited for VLSI implementation and offers the basic time/area trade-off for addition. It has been shown that for these adders error analysis can be done very efficiently by a simple structure analysis of the circuit. As a consequence, the check base selection problem can be solved automatically for individual adders. For regular adder families an analytical approach exploiting their regularity has been applied. As a result, important adder families have been characterized where on-line error detection can be achieved by minimum check bases.

Experiences with Biased Random Test Pattern Generation to Meet the Demand for High Quality BIST

H. Grützner **C. W. Starke**
IBM Laboratory Böblingen

It is of much interest to find solutions for a high quality Built-In Selftest (BIST) with stuck-fault coverage > 99%. We investigated 3 approaches for LFSR based biased random pattern testing. They can be mapped into custom-made finite state machines satisfying the requirements in terms of hardware overhead on the chips for BIST. The first approach chooses patterns with constant weights for all chip inputs, the second one uses weights for clusters of inputs. With above techniques the test coverage was improved up to 2% compared to the results with flat random patterns. This was demonstrated for two CMOS VLSI chips. The hardware overhead was < 1%. The third approach uses weight merging algorithm, in which an optimal weight set is determined by merging previously calculated optimal weights. Thus leads to a fault coverage of appr. 99%, the hardware overhead is in the range of 3% as also demonstrated for one VLSI chip.

A Modified Wallace-Tree Multiplier with Reduced Area

R. Oelschlägel **M. Weber** **B. Zaddach**
Humboldt - University, Berlin

We present a modified Wallace-tree multiplier which is as fast as the traditional Wallace-tree architecture but saves more than one third of the transitions. The multiplier bases on a combination of two well known ideas: The Wallace-tree and the radix-4 encoding of one of the operants of the multiplication. The obtained results are of basically practical interest. In our research we've attached great importance to the existence of efficient layout realizations and to the tractability of layout generation. Further, we've tried to give a special respect to the electrical peculiarities of the architecture, especially to the increasing load capacity of signals which are to be driven for increasing bit size of the argu-

ments. At present we have a generator which generates a symbolic layout for a given bit size of the operants. The architecture was intensively tested; for bit sizes between 8 and 128 and for a large number of input series we've made SPICE simulations using the 1.5 μm Euro-Chip CMOS technology. The obtained results are quite encouraging.

On the Importance of Variable Orderings for Reduced OBDDs and a Remark on Multiplication

I. Wegener
University of Dortmund

The sensitivity of a Boolean function is defined as the quotient of the reduced OBDD with respect to a worst and an optimal ordering of the variables. It is proven that the sensitivity of a random function is $1 + O(n^2 2^{-n/3})$ for all but a fraction of $O(2^{-n/4})$ of all functions. But for important functions the situation is different, the sensitivity is exponential and the fraction of good orderings is exponentially small. A model is formulated where the observation can be generalized to all but a small fraction of functions with small OBDD size. Two other results are mentioned. In MBDDs it is possible to test the variables k-times with respect to a given ordering. Despite of the existence of null chains (paths which are not computation paths) all typical operations are possible in polynomial size (k is the exponent). It is argued that no data structure supporting some of the important operations can represent the graph of multiplication in polynomial size. Otherwise the public key cryptosystem RSA is unsafe!

Dagstuhl-Seminar 9307:

List of Participants

Wolfram **Büttner**
SIEMENS AG - ZFEBT SE 1
Zentralabt. Forschung und Entwicklung
Otto-Hahn-Ring 6
D-81739 München
tel.: +49-89-6364 6310

Frank **Bauernöppel**
Humboldt Universität
Fachbereich Informatik
Unter den Linden 6-8
D-10117 Berlin
bauernoe@informatik.hu-berlin.de
tel.: +49-30-28 051 94 App. 17

Bernd **Becker**
Universität Frankfurt
Fachbereich Informatik (20)
Robert-Mayer-Str. 11-15
D-60054 Frankfurt
becker@kea.infomatik.uni-frankfurt.dbp.de
tel.: +40-69-7988250

Jochen **Bern**
Universität Trier
Fachbereich IV - Informatik
Zurmaiener Str. 98-100
D-54292 Trier
bern@uni-trier.de
tel.: +49-651-201-2836

Beate **Bollig**
Universität Dortmund
Fachbereich Informatik
Postfach 50 05 00
D-44221 Dortmund
bollig@nirvana.informatik.uni-dortmund.de
tel.: +49-231-755-2067

Raul **Camposano**
GMD Gesellschaft für Mathematik
und Datenverarbeitung mbH
Schloß Birlinghoven
Postfach 13 16
D-53731 St. Augustin
camposano@gmd.de
tel.: +49-2241-14 28 74

Yongzhang **Chen**
Universität Karlsruhe
Institut für Rechnerentwurf und
Fehlertoleranz
Kaiserstr. 12
D-76128 Karlsruhe
yongzhan@ira.uka.de
tel.: +49-721-6083961

Rolf **Drechsler**
Universität Frankfurt
Fachbereich Informatik (20)
Robert-Mayer-Str. 11-15
D-60054 Frankfurt
drechsler@kea.informatik.uni-frankfurt.de
tel.: +49-69-798-8472

Klaus-Jürgen **Englert**
Universität Frankfurt
Fachbereich Informatik (20)
PF 11 19 32
D-60054 Frankfurt

Hans **Eveking**
Universität Frankfurt
Fachbereich Informatik (20)
Robert-Mayer-Str. 11-15
D-60054 Frankfurt
eveking@informatik.uni-frankfurt.de
tel.: +49-69-798-8149

Thomas **Filkorn**
SIEMENS AG - ZFE BT SE1
Zentralabt. Forschung und
Entwicklung
Otto-Hahn-Ring 6
D-81739 München
filkorn@inf21.zfe.siemens.de
tel.: +49-89-636-45760

Winfried **Görke**
Universität Karlsruhe
Institut für Rechnerentwurf und
Fehlertoleranz
Kaiserstr. 12
D-76128 Karlsruhe
goerke@ira.uka.de
tel.: +49-721-608-39 09

Michael **Galetzka**
Fraunhofer Inst. f. Integrierte
Schaltungen Erlangen
Außenstelle Dresden
Zeunerstr. 38
D-01069 Dresden
galetzka@eas.iis.fhg.de
tel.: +49-351-463-3211

Walter **Geisselhardt**
Universität-GHS Duisburg
Fachbereich 9 DV
Postfach 10 16 29
D-47048 Duisburg
gd@du9ds3.uni-duisburg.de
tel.: +49-203-379-27 29

Jordan **Gergov**
Universität Trier
Fachbereich IV - Mathematik
Postfach 3825
D-54286Trier
gergov@uni-trier.de
tel.: +49-651-201-2832

Ralf **Hahn**
Universität Frankfurt
Fachbereich Informatik (20)
Robert-Mayer-Str. 11-15
D-60054 Frankfurt
hahn@kea.informatik.uni-frankfurt.de
tel.: +49-69-798-23 16

Joachim **Hartmann**
Universität des Saarlandes
Fachbereich 14 - Informatik
Postfach 11 50
D-66041 Saarbrücken
hartmann@cs.uni-sb.de
tel.: +49-681-302-34 64

Manfred **Henftling**
TU-München
Postfach 2024 20
D-80290 München
mah@regent.e-technik.tu-muenchen.dbp.de
tel.: +49--89-5517-430

Christiane **Henno**
Robert Bosch GmbH
Abt. KA/DIC1
Tübinger Straße 123
D-72762 Reutlingen

Ulrich **Hertrampf**
Universität Trier
Fachbereich IV - Informatik
Postfach 3825
D-54286 Trier
hertramp@uni-trier.de
tel.: +49-651-201-28 38

Günter **Hotz**
Universität des Saarlandes
Fachbereich 14 - Informatik
Postfach 11 50
D-66041 Saarbrücken
hotz@cs.uni-sb.de
tel.: +49-681-302 2414

Warren **Hunt**
Computational Logic Inc.
Suite 290
1717W. Sixth Street
Austin TX 78703-4776
USA
hunt@cli.com
tel.: +1-512-322-99 51

Reiner **Kolla**
Universität Bonn
Institut für Informatik
Römerstr. 164
D-53117 Bonn
kolla@silpion.informatik.uni-bonn.de
tel.: +49-228-550-373

Rolf **Krieger**
Universität Frankfurt
Fachbereich Informatik (20)
Robert-Mayer-Str. 11-15
D-60054 Frankfurt
krieger@kea.informatik.uni-frankfurt.de
tel.: +49-69-798-81 55

Dirk **Möller**
Humboldt Universität
Fachbereich Informatik
Unter den Linden 6-8
D-10117 Berlin
dmoeller@informatik.hu-berlin.de
tel.: +49-30- 2805 194 App.17

Christoph **Meinel**
Universität Trier
Fachbereich IV - Informatik
Postfach 3825
D-54286Trier
meinel@uni-trier.dbp.de
tel.: +49-651-201-28 27

Paul **Molitor**
Universität des Saarlandes
Fachbereich 14 - Informatik
Postfach 11 50
D-66041 Saarbrücken
tel.: +48-681-302 2064
molitor@cs.uni-sb.de
fax.: +48-681-302 4421

S.M. **Reddy**
University of Iowa
Dept. of Electrical Engeneering and
Computer Science
Iowa City IA 52242
USA

Alexander **Saldanha**
University of California at Berkeley
Dept. of Electrical Engineering (EECS)
and Computer Science
207-66 Cory Hall
Berkeley CA 94720
USA
saldanha@ic.berkeley.edu
tel.: +1-510-642-3626

Björn **Schieffer**
Universität des Saarlandes
Fachbereich 14 - Informatik
Postfach 11 50
D-66041 Saarbrücken

Detlef **Sieling**
Universität Dortmund
Fachbereich Informatik
Postfach 50 05 00
D-44221 Dortmund
sieling@nirvana.informatik.uni-dortmund.de
tel.: +49-231-755-2067

Anna **Slobodova**
Universität Trier
Fachbereich IV - Mathematik
Postfach 3825
D-54286 Trier
anna@kleopatra.uni-trier.de
tel.: +49-651-201-2833

Uwe **Sparmann**
Universität des Saarlandes
Fachbereich 14 - Informatik
Postfach 11 50
D-66041 Saarbrücken
sparmann@cs.uni-sb.de
tel.: +49-681-302-3414

C. W. **Starke**
IBM Deutschland GmbH
Abt. 3169
Schönaicher Str. 220
D-71032 Böblingen
tel.: +49-7031-16-2461

Bernd **Straube**
Fraunhofer-Institut f. Integrierte
Schaltungen Erlangen
Außenstelle Dresden
Zeunerstr. 38
D-01069 Dresden
straube@eas.iis.fhg.de
tel.: +49-351-463-32 11

Michael **Weber**
Humboldt Universität
Fachbereich Informatik
Unter den Linden 6-8
D-10117 Berlin
weber@informatik.hu-berlin.de
tel.: +49-30-209-30

Ingo **Wegener**
Universität Dortmund
Fachbereich Informatik
Postfach 50 05 00
D-44221 Dortmund
wegener@cantor.informatik.uni-dortmund.de
tel.: +231-755-27 77

Hongzhong **Wu**
Universität des Saarlandes
Fachbereich 14 - Informatik
Postfach 11 50
D-66041 Saarbrücken
wu@cs.uni-sb.de
tel.: +49-681-302-20 64

Zuletzt erschienene und geplante Titel:

E. Novak, St. Smale, J.F. Traub (editors):
Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report; 50; 12.10.-16.10.92 (9242)

J. Encarnação, J. Foley (editors):
Multimedia - System Architectures and Applications, Dagstuhl-Seminar-Report; 51; 02.11.-06.11.92 (9245)

F.J. Rammig, J. Staunstrup, G. Zimmermann (editors):
Self-Timed Design, Dagstuhl-Seminar-Report; 52; 30.11.-04.12.92 (9249)

B. Courcelle, H. Ehrig, G. Rozenberg, H.J. Schneider (editors):
Graph-Transformations in Computer Science, Dagstuhl-Seminar-Report; 53; 04.01.-08.01.93 (9301)

A. Arnold, L. Priese, R. Vollmar (editors):
Automata Theory: Distributed Models, Dagstuhl-Seminar-Report; 54; 11.01.-15.01.93 (9302)

W. Cellary, K. Vidyasankar , G. Vossen (editors):
Versioning in Database Management Systems, Dagstuhl-Seminar-Report; 55; 01.02.-05.02.93 (9305)

B. Becker, R. Bryant, Ch. Meinel (editors):
Computer Aided Design and Test , Dagstuhl-Seminar-Report; 56; 15.02.-19.02.93 (9307)

M. Pinkal, R. Scha, L. Schubert (editors):
Semantic Formalisms in Natural Language Processing, Dagstuhl-Seminar-Report; 57; 23.02.-26.02.93 (9308)

W. Bibel, K. Furukawa, M. Stickel (editors):
Deduction , Dagstuhl-Seminar-Report; 58; 08.03.-12.03.93 (9310)

H. Alt, B. Chazelle, E. Welzl (editors):
Computational Geometry, Dagstuhl-Seminar-Report; 59; 22.03.-26.03.93 (9312)

H. Kamp, J. Pustejovsky (editors):
Universals in the Lexicon: At the Intersection of Lexical Semantic Theories, Dagstuhl-Seminar-Report; 60; 29.03.-02.04.93 (9313)

W. Strasser, F. Wahl (editors):
Graphics & Robotics, Dagstuhl-Seminar-Report; 61; 19.04.-22.04.93 (9316)

C. Beeri, A. Heuer, G. Saake, S. Urban (editors):
Formal Aspects of Object Base Dynamics , Dagstuhl-Seminar-Report; 62; 26.04.-30.04.93 (9317)

R. Book, E.P.D. Pednault, D. Wotschke (editors):
Descriptional Complexity: A Multidisciplinary Perspective , Dagstuhl-Seminar-Report; 63; 03.05.-07.05.93 (9318)

H.-D. Ehrig, F. von Henke, J. Meseguer, M. Wirsing (editors):
Specification and Semantics, Dagstuhl-Seminar-Report; 64; 24.05.-28.05.93 (9321)

M. Droste, Y. Gurevich (editors):
Semantics of Programming Languages and Algebra, Dagstuhl-Seminar-Report; 65; 07.06.-11.06.93 (9323)

Ch. Lengauer, P. Quinton, Y. Robert, L. Thiele (editors):
Parallelization Techniques for Uniform Algorithms, Dagstuhl-Seminar-Report; 66; 21.06.-25.06.93 (9325)

G. Farin, H. Hagen, H. Noltemeier (editors):
Geometric Modelling, Dagstuhl-Seminar-Report; 67; 28.06.-02.07.93 (9326)

Ph. Flajolet, R. Kemp, H. Prodinger (editors):
"Average-Case"-Analysis of Algorithms, Dagstuhl-Seminar-Report; 68; 12.07.-16.07.93 (9328)