MATHEMATISCHES FORSCHUNGSINSTITUT OBERWOLFACH

T a g u n g s b e r i c h t 27/1991

Mathematische Methoden des VLSI–Entwurfs und des Distributed Computing

23.6. bis 29.6.1991

The Conference on 'Mathematische Methoden des VLSI–Entwurfs und des Distributed Computing' was organized by M. J. Fischer (New Haven) and G. Hotz (Saarbrücken). After the meetings in 1983 and 1987, it was the third conference concerning VLSI–Design and the second one concerning Distributed Computing, respectively.

The conference was visited by 24 participants which came from Germany, USA, Israel and Australia. This relatively small number enabled the holding of very comprehensive lectures and left enough time for discussions. By this way, the conversation between researchers which are interested in a more theoretical view of the problems and those which pay more attention on practical aspects was advanced.

The lectures dealt with the following themes:

- layout problems (floorplanning, placement, routing);
- logic synthesis;
- test methods;
- design of VLSI–systems;
- parallel architectures;
- problems in regular networks;
- secret and reliable communication in distributed systems;
- synchronization problems;
- fault–tolerant computing;
- bioinformatics.

Beside these lectures, there were demonstrations of the systems CADIC (Th. Burch, Univ. d. Saarlandes) and HULDA (M. Weber, Humboldt–Univ. Berlin), and an interesting lecture concerning chess on a network of transputers (R. Feldmann, Univ. Paderborn).

Vortragsauszüge

B. Becker:

Synthesis for Testability: BDDs

We consider a special case of the synthesis for testability spectrum: circuits which are derived from (reduced ordered) Binary Decision Diagrams (BDDs). The testability of these circuits is characterized with respect to the cellular, stuck–at and path–delay fault model. Full testability can be achieved in (almost) all cases. Corresponding test sets can be computed efficiently.

Th. Burch:

CADIC, a Top–Down Design System for Integrated Circuits

In this talk, we present the graphical frontend of the design system CADIC which has been developed at the University of Saarbrücken within the Sonderforschungsbereich 124. This schematic editor allows the designer to specify whole families of circuits by using parametrized instances of macro–cells and wires. The editor also provides an easy way of specifying random–logic circuits. The design can base on any given library of basic–cells. In future, we will integrate the synthesis–tools of the design system into this editor in order to get a complete graphical design system.

R. Feldmann:

Some New Results for Binary Networks

(joined work with B. Monien, P. Myslivietz and S. Tschöke)

In the first part of the talk, we show that every Shuffle Exchange network of dimension 2^d ($SX(2^d)$) has a Hamiltonian path. This result is obtained inductively. We show that if you have a Hamiltonian path in $SX(k)$ that obeys some constraints then you obtain 2^k node disjoint paths of length 2^k in $SX(2k)$. These node disjoint paths can be combined to a Hamiltonian path in $SX(2k)$ using an embedding of the $SK(2k)$ into the DeBruijn network of dimension k ($DB(k)$).
In the second part of the talk, we develop algorithms to determine the shortest path between two nodes a and b in the Butterfly network ($BFN(n)$), Cube–Connected–Cycle network ($CCC(n)$), $DB(n)$ and $SX(n)$ running in time polylogarithmic in the number of nodes.

M. Fischer:

Multiparty Secret Key Exchange Using a Random Deal of Cards

(joined work with R. Wright)

A team of k players is formed within an m–person organization to work on a classified project. The team wishes to construct an n–bit secret key S to use for secure communication among the team members. Others in the organization are collectively

regarded as an eavesdropper, Eve, who is trying to discover S. Following the example of Fischer, Paterson and Rackoff, members of the organization are given hands of cards of prespecified sizes before the team is formed. The hands are dealt from a randomly shuffeled deck of distinct cards. We ask how the hands can be used to establish S. That is, we desire a protocol for the team by which each team member learns S, any n–bit string is equally likely to be S, and Eve learns nothing about S, even if she overhears all communication among the team members. We place no computational limitations on Eve, so standard public key cryptographic techniques do not apply.
Our main result is a randomized protocol that solves this problem whenever the desk is sufficiently large. We also examine a class of simple 1–bit secret key exchange protocols that work for particular collections of hand sizes. We show one of the protocols to be optimal within this class by applying an intricate game–theoretic argument to a closely–related combinatorial game. Finally, we provide some weak impossibility results that show, for example, the nonexistence of a 1–bit secret key exchange protocol for a team of three players, each of whom holds a single card from a three card desk, even when Eve has no cards.

M. Kaufmann:

Homotopic Planar Routing of Multiterminal Nets

(joined work with S. Gao)

In the area of VLSI–algorithms, homotopic routing is a quite new development which considers the detailed routing phase on the basis of a proceeding global routing of the nets. For the planar model, there are some efficient algorithms which work well for two–terminal net problems, but not for multiterminal nets. In this talk, we consider homotopic routing of multiterminal nets and present the first efficient algorithm to find a minimum–length solution by constructing Steiner trees.

R. Kolla:

Performance Optimization of Combinational Circuits

In this talk, we present a cell based approach to the problem of performance optimization of VLSI combinational circuits. In general, performance optimization is the problem of finding optimal area–time trade–offs for an implementation of a boolean function by an integrated circuit. It is too hard to solve this problem as a whole since it contains the synthesis problem as well as the mapping to a given technology. Therefore, we focus on the case where the structure of the circuit as a gate circuit is fixed and only the physical parameters are free over a finite set of possible implementations by a cell.
We show that even this restricted case is NP–complete. It remains NP–complete if the circuit is a tree but in this case there are dynamic programming methods with a pseudopolynomial time complexity which are tractable for practical instances. We

show how these exact algorithms can be used to compute approximate optimal solutions for combinational circuits. Finally, we report experimental results for benchmark circuits and point out generalizations of our approach.

M. Kunde:

Balanced Routing: Towards the Distance Bound on Grids

The problem of packet routing on an r–dimensional grid of processors with side length n is studied. Each processor is able to store $r \cdot f(n)$ packets, $f(n) < n^{1-1/r}$. The new class of balanced routing problems is introduced which includes such fundamental problems as partial h–h routing. On 3–dimensional $n \times n \times n$ grids (without wrap–arounds), partial permutations and $(1, f(n))$–balanced problems can be solved in $3.333n + O(n/\sqrt{f(n)})$ steps. For arbitrary r, $r \geq 3$, these problems can be done within $(2r - 3 + 1/r)n + o(n)$ steps, which is faster than the corresponding numbers of steps of previous deterministic and randomized algorithms. A further improvement is possible for $r \geq 4$. By a new rearrange technique, a step number of $(r + (r-2) \cdot (1/r)^{1/(r-2)})n + o(n)$ can be achieved. The number of steps is reduced to the half if the algorithms are adapted to tori of processors. For h–h routing, the algorithms can be generalized such that $O((h + r)n)$ steps are possible.

Th. Lengauer:

Bioinformatics: A Challenge for the 90's

This talk gives an introduction into the area of bioinformatics which is concerned with the following problems:

Molecular Modelling: Designing and analyzing biomolecules with the aid of the computer.

Sequence Analysis: The analysis of the relationship between biomolecular sequences (DNA, RNA, Peptide chains).

Molecular Dynamics: The analysis and prediction of the temporal interactions of biomoleculs in order to explain and predict their function.

Understanding DNA: The interpretation of the vast amount of genetic information stored in the chromosomes.

The field is in bad need of algorithmic advances. As a prerequisite, appropiate models have to be defined that are both faithful to reality and susceptible to efficient computation. The talk emphasizes the protein folding problem.

F. Mattern:

On the Relativistic Structure of Logical Time in Distributed Systems

A distributed system can be characterized by the fact that the global state is distributed and that a common time base does not exist. We show that the notion of time is an important concept in everyday life of our decentralized "real world" which helps to solve problems like getting a consistent population census ("snapshot") or

determining the potential causality between events. We argue that a linearly ordered structure of time is not (always) adequate for distributed systems and propose a generalized non–standard model of time which consists of vectors of clocks. These clock–vectors are partially ordered and form a lattice. By using timestamps and a simple clock update mechanism, the structure of causality is represented in an isomorphic way. Vector timestamps are an interesting generalization of Lamport's well–known logical clocks. Among other things, the new model of time leads to an interesting characterization of the global state problem and allows to determine whether two events are causally related or not. The model has a close analogy to Minkowski's relativistic space–time. We discuss this analogy and show that in Minkowski's model two broadcasts which are causally dependant are perceived in the same order by all observers. We then demonstrate that using time vectors this same notation of "causal order" can be implemented in distributed systems with asynchronous messages of unknown delays.

S. Moran:

The Distributed Counter Problem

(joined work with I. Yadin)

A counter (mod m) holds an integer in the range $[0, \cdots, m-1]$, and enables two basic operations:
(1) INCREMENT: Increment its value by one (mod m), and
(2) LOOK: Get its current value.
A *distributed* counter is a counter in a shared memory environment, which can be INCREMENTed and/or LOOKed by many processors. Most known implementations of distributed counters forbid concurrency: When a processor P wishes to perform INCREMENT/LOOK, P first locks the access to the counter, then it modifies/gets its value as needed, and finally P unlocks the access to the counter. This approach ensures the correctness of the implementation, but may create bottleneck effects when many processors wish to access the counter simultaneously. Moreover, in the presence of fail–stop failures the counter may get deadlocked.
In this talk, we study implementations which enable many processors to access the counter concurrently. Such implementations raise problems in ensuring (and even in defining) the correctness of the INCREMENT/LOOK operations. We define two types of correctness – *static correctness*, which requires that a LOOK operation returns the correct value only if it is not concurrent with any INCREMENT operation, and *dynamic correctness*, which requires that a LOOK operation returns a correct value in all cases. Then we present lower and upper bounds on the number of bits required to implement a distributed counter under these correctness requirements, as a function of the number of processors that are allowed to INCREMENT the counter. In some cases we present implementations which are very efficient in both space and time, while in other cases we show that any implementation must be very inefficient.

W. J. Paul:

On the Cost Effectiveness of PRAMs

In the first part of this lecture, we introduce a model which permits to treat computer architecture as a formal optimization problem. In the second part of this lecture, we investigate the cost effectiveness of PRAMs. We present and analyze a reengineered version of the "Fluent Machine". The resulting machine has a surprisingly good price / performance ratio even if compared with distributed memory machines (DMMs) on a workload which is ideal for DMMs.

R. Pinter:

Symbolic Layout Improvement Using String Matching Based Local Transformations

(joined work with S. Ben–Yehuda)

Symbolic layout often serves as an intermediate form during layout generation. We present a methodology, borrowed from compiler optimization practice, and a specific technique, based on efficient string matching algorithms, to improve the quality of symbolic layout. The method constitutes the application of numerous local replacement rules to the original symbolic layout, thereby producing a new layout with better performance characteristics and that would facilitate better eventual compaction. The transformations are enabled by the recognition of patterns in the layout, which is performed efficiently using a linear–time string matching algorithm. The ideas have been demonstrated by an experimental tool, which was applied to a number of channel routes, resulting in significant improvements.

R. Pinter:

Realizing Expression Graphs with $Xilink^R$ Elements

(joined work with I. Levin)

Xilink networks comprise programmable functional elements (of 4 inputs and 1 output) connected by a configurable network (a grid of busses). We consider the problem of mapping an expression graph (that represents a combinational network) to a minimal number of Xilink elements. In the present work, we only look at the topology of the expression graph, i.e., we ignore the operations so, e.g., no algebraic simplification is performed during the mapping (it could have been made at earlier stages, of course). We present two results:

Trees (of arbitrary degree) can be mapped optimally in linear time.

The problem becomes NP–complete for DAGs, even if they have only 1 root and the maximal in–degree of its nodes is 3.

We are currently working on heuristics for DAGs.
In addition, several open problems remain, such as:
– consider semantics during the mapping process;

– allow functional elements with two outputs (realizing, say, two functions of three inputs each);
– consider other optimization criteria (depth, skew, ...);
– integrate layout considerations into the mapping.

S. Pinter:

Scheduling Instructions for Data–Flow Machines

(joined work with R. Hardon)

Data–Flow computers execute programs by dividing a data flow graph into instruction templates which are scheduled as early as possible. Implementing this scheme involves communication overheads which affect the running time of the program. In this paper, we present a model for data flow machines which includes both communication and execution times. With this model, we derive lower and upper bounds on the execution time of programs represented as trees and DAGs. We provide algorithms for optimally partitioning a program into sets of instruction templates, for both tree and DAG like programs realizing the minimum execution time. The algorithms are of time complexity $O(|V|^2)$ and $O(|V|^5)$, respectively.

R. Reischuk:

Lower Bounds for Fault–Tolerant Computations

We consider Boolean circuits and decision trees in which gates may make errors with probability ε. It is shown that most n–input functions require noisy circuit size resp. tree depth $\Omega(n \cdot \log n)$ to be realized with error probability at most $\delta < 1/2$. More specific, we prove that the critical number $crit(f)$ of a function f implies the lower bound $\Omega(crit(f) \cdot \log crit(f))$ for circuits and static decision trees. In the dynamic decision tree model, a random function has complexity $\Theta(n \cdot \log n)$.

H. Schröder:

Load Balancing Algorithms

Load balancing in mesh–connected architectures is evaluated under the linear model of VLSI computation. The system investigated in has N processors with k storage cells each. The processors can produce or consume entries of their storage in events called 'synchronized access'. It is shown that the average balancing time in between synchronized accesses has the lower bound $T = \Omega(\sqrt{N}/\sqrt{k})$ for $k = O(N)$. Algorithms that meet these lower bounds in the range $2 \leq k \leq 2N$ are presented. These are of little practical value. Only for $k = cN$ (c = constant) practically feasible algorithms are presented.

J. Sieck:

Placement in Computer Aided Schematics

(joined work with M. May)

A schematic is a mapping of real or abstract objects including their interconnections and possibly some lettering into the plane. Schematics are mainly characterized by their structure. Examples of typical schematics are: block diagrams, flow charts, technological schematics, logic diagrams, Petri nets, graphs and networks.
Fundamental layout requirements are:
- maintainance of the main signal flow;
- grouping together strongly interconnected symbols;
- short interconnection lines;
- few intersections and bends in the line routing.

Scheme decomposition, symbol placement and line routing are the major steps in the automatic design of graphical schematics. Because the general placement problem for netlike schematics and also typical subproblems (grid placement, rowplacement) are NP–hard, it is necessary to develop heuristic algorithms.
In the talk, we discuss a general placement algorithm which generates schematics in every case and some special algorithms for the placement of schematics with special properties.

U. Sparmann:

Structure Based Test Methods for Arithmetic Circuits

Today's manufacturing processes for VLSI circuits are very sensible to disturbances and thus a large percentage of the produced chips is physically defect, even if the design was correct. Thus, there has to be a test phase, where the good chips are separated from the faulty ones. Since the problem of generating a test set needed for this phase is NP–complete, universal test generation algorithms which can be applied to arbitrary circuits often deliver only poor results for large modules. Fortunately, large circuits like RAMs, PLAs or iterative logic arrays are very regular in structure and their structural properties can be used to derive efficient test sets for them.
In the talk, results about such a structure based test of fast realizations for the most essential arithmetic operations in computers have been presented. For the example of parallel prefix computation over finite monoids, it was shown how to exactly characterize the testability of this operation by taking advantage of algebraic properties of the monoid operation and structural properties of the computation network. Application of this theory results in the construction of time and test optimal adders and incrementers.
New problems arise when considering the test problem for heterogenous circuits which consist not only of one regular module but of a large number of modules which may be regular or irregular in structure:
1) How to handle irregular parts?
2) How to apply tests over the surrounding circuitry?

Solutions to these problems have been presented for floating point addition and multiplication. As a result, we obtain completely testable realizations with very small hardware overhead.

L. Stockmeyer:

Bounds on the Time to Reach Agreement in the Presence of Timing Uncertainty

(joined work with H. Attiya, C. Dwork and N. Lynch)

This work concerns the real–time complexity of reaching agreement, in the presence of processor failures and uncertain information about time. In particular, we focus on situations where the time to detect faults is greater than the time to deliver messages. For example, in one model studied, it is assumed that the amount of real time between any two consecutive steps of any correct processor is at least c_1 and at most c_2; thus, $C = c_1/c_2$ is a measure of timing uncertainty. An upper bound of d on message delivery time is also assumed. Assuming that processors fail by stopping, processor failures can be detected by timeouts. Letting T denote the worst–case time to detect a failure, a standard timeout procedure gives T roughly equal to Cd. The straightforward simulation of a rounds–based agreement algorithm could use f consecutive timeouts if there are f faults, taking time at least fT, i.e., fCd. One result is an agreement algorithm in which the worst–case time for a timeout is incurred at most once, yielding a running time of approximately $2f\delta + T$, where δ is an upper bound on the message delay that actually occurs in a given execution. A lower bound of $(f-1)d + Cd$ on the time to reach agreement in the timing–based model (in the case $\delta = d$) is also shown.

R.Strong:

New Latency Bounds for Atomic Broadcast

(joined work with D. Dolev and F. Cristian)

We discuss bounds on the time required to reach agreement in a distributed system as a function of the failure model. Our model consists of a set of n processors that communicate via messages and possess clocks that are approximately synchronized. We assume (1) an upper bound d on message transmission and processing delay as measured on any correct clock in the system; (2) an upper bound e on the difference in reading between two correct clocks at the same real time; and (3) an upper bound ρ on the rate of drift of any correct clock with respect to real time.
Simple Atomic Broadcast is a single input, multiple output agreement satisfying:

1) if value z is given as input to correct processor p at time T on p's clock, then all correct processors eventually deliver $< T, p, z >$ as output; and
2) if any correct processor delivers $< T, p, z >$, then all correct processors deliver $< T, p, z >$.

The latency of an atomic broadcast is the worst case difference between T and the local clock time of delivery of $< T, p, z >$, expressed as a function of the number f of processor failures to be tolerated. (For simplicity, we assume that the network is completely connected and that only processors fail.)

failure model	lower bound	upper bound
	Previous Results	
omission	$e + (f+1)d$	$e + (f+1)d$
generalized timing	$e + (f+1)d$	$(f+1)(e+d)$
Byzantine	$e + (f+1)d$	$(f+1)(e+d)$
	New Results	
generalized timing	$k(e+d)$	$ke + \alpha(n, f, \rho, d)$
Byzantine	$2(e+d)$	$2e + \beta(n, f, \rho, d)$

where $k = \lfloor n/(n-f) \rfloor + \lfloor (n-1)/(n-f) \rfloor$.

W. Vogelgesang:

Performance Driven k–Layer Wiring

(joined work with M. Kaufmann and P. Molitor)

Given a grid based wire layout, the objective of the layer assignment problem we investigate is to minimize the interconnect delay by taking into account the conductivity of interconnection wires and vias. For MOS circuits with two or more layers for interconnections, the problem is shown to be NP–hard. It remains NP–hard for the special case of two layers having the same conductivity ($PDW(=)_2$) as well as for the special case of two layers with extremely different conductivities ($PDW(\neq)_2$). However, $PDW(\neq)_2$ can be reduced to a generalized integer flow problem which gives hope of good heuristics and approximations, possibly, not only for $PDW(\neq)_2$ but also for the general case. Furthermore, $PDW(\neq)_2$ can be solved in polynomial time when the routing grid is sufficiently fine.

M. Weber:

Data Structures and Algorithms for Routing

A router designing environment is presented which contains tools to describe the geometry of channels and the corresponding net information as well as special graphs, tools for verification, evaluation, classification of routing tasks and wirings.
The work at this system especially aimes at:

- providing a pool of data structures and tools to support the implementation of routers as much as possible;
- treating all three dimensions equally without any preference;
- winning a wide range of possibilities to express restrictions and degrees of freedom in routing problems and to make use of them in the course of solution;
- implementing a variety of routing strategies as 'experts' of a black–board–system.

A special Left–Edge like Router is presented which works in some cases better than the known routers of this type.

G. Zimmermann:

A Genetic Algorithm for the Compaction of Floorplans

(joined work with A. Heß and K. Glasmacher)

In hierarchical VLSI design methods, floorplans are generated by placing and global routing for flexible cells. The shapes of these cells have to be estimated as well as the widths of the routing channels between cells. These all are then layed out such that they fit best into the floorplan. The remaining problem, called chip assembly, is the correction of the floorplan such that all channels are routable and the area is minimized. Since the minimal width of the channels is a function of a lateral offset of the cells along the channel, this is more complex than the classical 2–dim. compaction problem. The width function can be measured. If we assume floorplans with slicing structures, we can independently choose one offset for each cut line. For such a set of choices, we can construct a legal floorplan. The total area of this floorplan is used as "fitness" value in a genetic algorithm that tries to find the best of all sets of offsets. Crossbreedings can be obtained by mixing these sets from two good solutions and generate children. Mutations are possible by randomly changing offsets in a set. With populations of about 100 different sets of offsets and 100 generations, good results have been achieved for a number of real layouts. The question remains if better algorithms can be found to solve the stated compaction problem.

L. Zuck:

Real–Time Sequence Transmission Problems

(joined work with D. Wang)

In the sequence transmission problem, the transmitter wishes to reliably communicate a sequence of data items (messages) to another process, the receiver. We study a real–time version of the sequence transmission problem (RSTP) where the messages are taken from a binary domain and we assume three constants, c_1, c_2 and $d, c_1 \leq c_2 << d$, such that each process takes a step at least every c_1 and at most every c_2 units of time, and each packet which is sent is delivered within d time units. We define the effort of a solution to be the average time it takes the receiver to learn a message. We study the effort of solutions to RSTP as a function of c_1, c_2, d and k – the size of the transmitters packet alphabet. We show tight bounds on the effort of solutions to RSTP for both cases of silent and "talkactive" receivers.

Berichterstatter: Wolfgang Vogelgesang

Tagungsteilnehmer

Prof.Dr. Bernd Becker
Fachbereich Informatik
Universität Frankfurt

W-6000 Frankfurt
GERMANY

Thomas Burch
Fachbereich 14 - Informatik
Universität des Saarlandes

W-6600 Saarbrücken
GERMANY

Dr. Rainer Feldmann
FB17, Mathematik/Informatik
Universität Paderborn
Warburger Str. 100
Postfach 1621

W-4790 Paderborn
GERMANY

Prof.Dr. Michael J. Fischer
Department of Computer Science
Yale University
P.O. Box 2158, Yale Station

New Haven , CT 06520-2158
USA

Prof.Dr. Günter Hotz
Fachbereich Informatik
Universität des Saarlandes
Im Stadtwald 15

W-6600 Saarbrücken 11
GERMANY

Dr. Michael Kaufmann
Max-Planck-Institut für Informati
Im Stadtwald

W-6600 Saarbrücken
GERMANY

Dr. Reiner Kolla
Fachbereich Informatik
Universität des Saarlandes
Im Stadtwald 15

W-6600 Saarbrücken 11
GERMANY

Dr. Manfred Kunde
Institut für Informatik
TU München
Arcisstr. 21
Postfach 20 24 20

W-8000 München 2
GERMANY

Prof.Dr. Thomas Lengauer
FB17, Mathematik/Informatik
Universität Paderborn
Warburger Str. 100
Postfach 1621

W-4790 Paderborn
GERMANY

Dr. Friedemann Mattern
Fachbereich Informatik
Gebäude 36
Universität Kaiserslautern
Postfach 3049

W-6750 Kaiserslautern
GERMANY

Prof.Dr. Shlomo Moran
Computer Science Department
TECHNION
Israel Institute of Technology

Haifa 32000
ISRAEL

Prof. Dr. Heiko Schröder
Electrical Engineering and Computer Science
University New Castle

NSW 2308
AUSTRALIA

Prof.Dr. Wolfgang J. Paul
Fachbereich Informatik
Universität des Saarlandes
Im Stadtwald 15

W-6600 Saarbrücken 11
GERMANY

Dr. Jürgen Sieck
Institut für Informatik in Entwurf und Fertigung im ZKI
Kurstr. 33

O-1086 Berlin
GERMANY

Dr. Ron Y. Pinter
Scientific Center
IBM Israel Ltd.
Technion City

Haifa 32000
ISRAEL

Uwe Sparmann
Fachbereich Informatik
Universität des Saarlandes
Im Stadtwald 15

W-6600 Saarbrücken 11
GERMANY

Dr. Rüdiger Reischuk
Institut für Theoretische Informatik
Technische Hochschule Darmstadt
Alexanderstr. 10

W-6100 Darmstadt
GERMANY

Dr. Larry Stockmeyer
IBM Almaden Research Center
K53/802
650 Harry Rd.

San Jose , CA 95120
USA

Dr. Brigitte Röthlein
PM-Magazin
In der Neher-Str. 9

W-8000 München 80
GERMANY

Dr. Ray Strong
IBM Almaden Resarch Center
K 53/8012
650 Harry Road

San Jose , CA 95120-6099
USA

Gefördert durch
DFG Deutsche Forschungsgemeinschaft

Dipl. Inform. Wolfgang Vogelgesang
Fachbereich 14 - Informatik
Universität des Saarlandes

W-6600 Saarbrücken
GERMANY

Dr. Michael Weber
Fachbereich Informatik
Humboldt-Universität Berlin
Postfach 1297
Unter den Linden 6

O-1086 Berlin
GERMANY

Prof.Dr. Gerhard Zimmermann
Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049

W-6750 Kaiserslautern
GERMANY

Prof.Dr. Lenore Zuck
Department of Computer Science
Yale University
P.O. Box 2158, Yale Station

New Haven , CT 06520-2158
USA