Heriot-Watt University
Edinburgh

Schloss Dagstuhl 2000 Seminar Report

Points from talks attended at Schloss Dagstuhl (June 26th to 30th, Germany).

Evolvable Hardware (EHW), Xin Yao

- EHW can be though of as:
  - The application of evolutionary computation techniques to circuit synthesis.
  - Hardware that is capable of on-line adaptation.
  - Hardware that changes architecture and behaviour depending on environment.
- Can be thought of as an optimisation algorithm for a hardware system.
- Direct Evolution
  - Evolvable algorithm to select and optimise component values in a circuit e.g. the value of a resistor.
- Indirect Evolution
  - Evolves an intermediate representation (e.g. trees) which specify hardware circuitry.
  - Structured Functional Description Language (SFL) circuits can be evolved e.g. 4 bit adder successfully evolved.
- Can evolve simultaneously:
  1. Circuit topology.
  2. Number of components.
  3. Component values.
- However, how do you represent as a chromosome? Create a circuit constructing primitive and network has variable size.
- Idea of this work is to show that evolution can be used to generate working circuits.
- Gate level approach runs into scaling problems.
- Advantages:
  - Explores a larger design space.

  - Does not assume priori knowledge.
  - Does not require exact specification - can design complete systems.
  - Constraints and special requirements can be imposed on evolution i.e. system must have a specific frequency response.
  - Some analogue circuits are too difficult or costly to design by humans.
- Challenges
  - Scalability is an issue. Time complexity of evolutionary algorithm for evolvable hardware.
  - Time is crucial since size of chromosome scales complexity in a polynomial manner.
- If we evolve hardware when do we stop evolution.
- This system is good at generalisation - dealing with new environments.
- Disaster prevention in fitness evaluation during on-line adaptation.
- Finally, EHW is good at discovering novel circuits that would not normally be designed.

Reconfigurable Computers in Space, Neil Bergman

- Reconfigurable computing has no killer applications terrestrially - but it does in space.
- Advantages:
  - After launch, spacecraft electronics are unavailable for physical upgrade or repair.
  - New circuit configurations can overcome design faults or change functionality.
  - The same circuitry can be used for different tasks reducing weight and power requirements.
  - Routing can be performed around failed parts.
  - FPGAs allow generic spacecraft circuit boards to be constructed rather than a series of specific ones.
- Disadvantages:
  - Ionising radiation can flip EPROM or SRAM bits.
  - Links to spacecraft are low bandwidth and high error rate. Can be a problem when downloading 1Mbit configuration files.
  - Limited by on-board configuration storage's susceptibility to radiation.

- Problems could be solved by triple redundancy voting circuits or CRC checking techniques.
- Need space friendly FPGAs - such parts are usually 10 years behind current technology.
- Space Friendly definition:
  - Can check themselves for errors.
  - Have special diagnostic logic.
  - Techniques needed to avoid "dodgy" circuitry.
  - Thin substrate needed - less volume for radiation interference.
  - Error detecting logic, fault secure logic or complementary logic required.
  - Single chip/MCM combination of FPGA, micro controller and flash memory.

Fault Tolerant Reconfigurable Computing, Heiko Schröder

- Due to radiation:
  - Single event upsets (common).
  - Latches stuck (need extra hardware).
  - Total loss (rare at 800km above Earth's surface).
- Methods used for fault tolerance:
  - Shadow processors.
  - Majority voting schemes.
  - Byzantine systems.
- Component grade hierarchy:
  - Industrial
  - Military
  - Radiation Tolerant
  - Radiation Hardened

Self Organisation of Reconfigurable Architectures, Uwe Tangen

- Hardware evolution:
  - Simulation of evolving hardware.
  - Instantiated hardware optimisation.

  - Biological evolution.
- DNA or RNA can be considered as just a bit string.
- Enzymes are the circuits.
- The problem is difficult to solve:
  - Combinatorial explosion of 2^{2^n}.
  - Mismatch between coding length and power.
  - Inflexibility of routing.
- Conclusions:
  - Self organisation in Si is possible.
  - Coding seems to be a central issue.

Experiences with Reconfigurable Computing and Further Work, Tobias Oppold

- Java is used to produce an object oriented breakdown of the system either in SW (micro controller) or in HW (FPGA).
- Java to VHDL to Netlist to Configuration Data.
- Tensilica: Xtensa processor - http://www.tensilica.com/
- Omnipath Processor (NEC) - Something in between a parallel processor and programmable logic. Multiple contexts with a switching time of ~2ns between contexts e.g. in one context an element is an adder, in another a XOR gate.

The Third Way - Neither Hard Nor Soft Ware, Gordon Brebner

- Hardware connotations: Logic, processors, datapaths, computer architectures, fixed I/O etc.
- Software connotations: Programs, scripts, GUIs, algorithms etc.: i.e. flexible.
- Alternative phrase: Run time reconfigurable systems.
- New technological influences:
  - System level integrated chips.
  - Configurable logic arrays.
  - Photonic technologies.
- Future
  - Networks of diverse programmable information handling components from chip level upwards.

    - Communication is the key problem.
- Death to the bus.
- Expressing diverse programmability:
    - 'Architecture' is the programmable thing.
    - 'Algorithm' refers to the programming.
- What is the optimal level of abstraction?
- Towards the third way:
    - Resist existing abstractions - could be too abstract.
    - Investigate incremental stepping stones towards future system design - don't solve everything too soon.
    - Invent new information handling models and system design processes.
- The barriers between computing, electronics and Photonics must be broken down.
- Many potential from the "disappearing computer" in the appliance or location.

Makimoto's Wave, the 2nd Design Crisis and the Future of Reconfigurable Computing, Reiner Hartenstein

- Microprocessor provides the most functionality, however the most silicon is used in accelerators.
- "Tail wagging the dog".
- Why not make accelerators reconfigurable.
- Logic synthesis i.e. hardware design by mapping onto a strange platform using CAD.
- Makimoto's Law: "Mainstream silicon application is changing every 10 years."

I2K-Microprossors and Reconfiguration, Klaus Waldschmidt

- Core of modern analogue and digital circuits.
- Size 'wall' of 0.1µm technology.
- Supply cannot be reduced under 1V.
- In semiconductors light propagates with one tenth of the speed it does in a vacuum.
- Memory capacity scales with the square of chip size. Access time is not increasing in the same way.

- Superscalar architectures use out of order processing.
- Complexity Effective Superscalar processor: Splits related processes instructions into FIFO pipelines. (Ref. Norman P. Jouppi).
- Prediction
  - To branch or not to branch.
  - Early estimation of control flow decisions.
- Configuration flow-cache for reconfiguration of functional units.
- Parallelism: co-operative load balancing at run time. Medium or coarse grain parallelism e.g. threads or microthreads.
- Configuration by programming language and compilers. Add circuit descriptions in (for example) VHDL using code.

Configurable System on Chip, Reiner Kress

- Configurable SoCs from Infineon AG.
- Designers need:
  - Configurable DSP instruction set.
  - Configurable SoC architecture.

Infrastructure of PCI Pamette, Mark Shand

- The Pamette is a reconfigurable memory mapped PCI board.
- Oriented to I/O and interfacing applications.
- PAM: Programmable Active Memory.
- Architecture has no processor on board - the system must have a host processor anyway.
- Compiler designed. Wanted to write code and not draw designs.
- Protect system against user misuse or program crash so it doesn't bring the host down.
- Bitstream compression works well (don't listen to Xilinx). Just try using gzip.
- http://www.research.digital.com/SRC/pamette

Asynchronous Run-Time Reconfiguration (RTR), Simon Taylor

- Problems: Lack of tools and difficult designs.
- Asynchrony - can split circuit across multiple FPGAs.

- Conventional FPGAs have problems that make asynchrony difficult. Asynchronous architectures are therefore required.
- 'Montage' can run synchronously or asynchronously.
- PGA-STC uses a programmable delay element to satisfy local timing constraints.
- STACC designed for implementation of self-timed reconfigurable systems.
- Clock is replaced by an array of timing cells. The advantage is that timing cells can be optimised for a specific task.

Embedding Express Graphs into Networks, Manfred Kunde

- Reducing the diameter of a network.
- Diameter is the longest distance between nodes i.e. maximum hop count.
- Motivated by ATM networks.
- First structure considered was ring. Can reduce average hops of N to $2.81\sqrt{N}$.
- Can also be used on mesh topology, H-tree or 3D H-tree.
- Graphs with Hamiltonian cycles allow you make full use of the ring concept.
- Must space node links evenly for optimal usage.

Using Optoelectronic Interconnects for Run-Time Reconfigurable Arrays, Dietmar Fey

- Motivation:
  - Increase in chip density $\alpha = 2$.
  - Increase in chip size $\beta = 2$.
  - Before: 5×5 elements. After: $5 \times 5 \times 2^2 \times 2^2 = 400$ elements.
- MQWs can also be used as detectors.

Prototyping Environment for Reconfigurable Processors, Serjei Savitzki

- Reconfigurable processors: PRISC, nP, Spyder, OneChip, Garp, CoMPARE, NAPA1000, Proteus.
- Approaches to prototyping:
  - Dedicated environment multi-FPGA boards with on-board memory ($10,000 to $100,000).

  - Commercial multi-FPGA boards that operate inside computer using dedicated software ($1,000 to $10,000).
  - Evaluation module - available from manufacturer only with specific card tools (<$1,000).
- Requirements for test environment:
  - Flexibility - Reconfiguration times should be kept short.
  - Scalability - Unsatisfactory mapping as it scales.
  - Observability - Can watch what the FPGA is doing at various stages: i.e. state information.

Evaluating DSPs with Dynamically Reconfigurable Processing Units, Steffen Köhler

- Processing of streaming data efficiently done by application specific DSP.
- DSP only suitable for a narrow range of applications.
- Performance enhancements only possible through hardware exchange.
- Could increase number of processing units (PUs), improve clock rate or introduce new computational topologies.
- RPU: Reconfigurable Processing Units.

The P2NC Project or How Much Parallelism is There?, Yosi Ben-Asher

- Consider the problem of speedup in sequential programs by means of parallelism.
- P2NC compiles sequential programs into boolean circuits and then executes them in parallel using a probabilistic algorithm.
- Parallelism is best exposed when programs are compiled into circuits.
- Modern processors are based on instruction level parallelism (ILP).
- Goals:
  - To produce upper bounds on parallelism when using ILP.
  - Evaluate potential improvement for compilation to circuits.
  - Probabilistic method used by P2NC can lead to new type of architecture that is not von-Neumann based.
- Metrics:
  - Depth of parallelism (DP) - the ratio between the size of a circuit and its depth.
  - Potential speedup (PS) - ability to speedup.

  - Execution of a program using probabilistic methods to evaluate the circuit.
- The circuit evaluation problem (CVP) is P complete and even its approximation is known to be hard.
- Expose parallelism by:
  - Loop unrolling.
  - Compiler/hardware scheduling where compiler/hardware groups independent instructions.
  - False memory dependencies.
  - Branch prediction - speculative execution of iterations based on predictions of the branch that will be taken.
  - Pipeline parallelism - not relevant as there is no load/fetch/decode. Everything is in one circuit.

Acceleration of Satisfiability Algorithms by Reconfigurable Hardware, Marco Platzner

- Application areas:
  - CAD of digital circuits, synthesis, optimisation, test pattern generation and verification.
  - Cryptography (encoding and decoding).
- Problem is NP complete.
- Software solves by using a backtracking tree search algorithm.
- Hardware does not change asymptotic runtime complexities.
  - Less powerful strategies than software.
  - Fast deduction by exploiting parallelism in problem instances.
- Instance specific reconfiguration - new hardware for each problem instance. Circuit generation and FPGA configuration at runtime.
- Required application characteristics:
  - Fine grained parallelism instance.
  - Long runtimes with each new instantiation.
- Optimisations:
  - Order variables.
  - Choose assignment order.
  - Check for special cases.

Cellular Automata with Reconfigurable Busses, Thomas Worsch

- Cellular Automata: Finite state machine operating on data and interconnected using some architecture to one of its neighbours.

Why and How Should We Use FPGAs to Run Mobile Code, Frederic Raimbault

- Specialised virtual configurable arrays (SVCA).
- Dynamically reconfigurable virtual machine (DRVM).
  - Improves runtime for mobile code.
- Mobile code - an evolving concept.
  - Transferred in place - client/server model (SQL).
  - Loaded on demand (HTML, appalets).
  - Autonomous agents (Aglets, Odessey, Voyager).
- Mobile code is useful for:
  - Reducing network load.
  - Disconnected (off-line) network mode.
- Already used in e-commerce and telecommunications runtime services.
- Java (OO and Java virtual machine) benefits:
  - Dynamic classes and safe memory handling.
- Java drawbacks:
  - Software complexity (installation is 70MB).
  - Resource hungry.
  - Performance: 10-100 times slower.
  - Raised security problem - executed on host.
  - Runtime rigidity - monolithic core, complex entangled data structures.
- Solution: put code on runtime reconfigurable hardware.

Specialised Virtual Configurable Arrays, Dominique Lavenier

- Just ideas, still no implementation.
- Wish to design "hardware" support that is independent of FPGA family architecture.
- Use CLBs as routing resources.

Hardware/Software Elements, Bernard Pottier

- Hardware resources are increasing faster and faster - design cost remains linear to this increase.
- Design time is critical.
- Main idea is homogeneous use of object models.
- Redesigning everything down to hardware with one methodology is attractive because:
    - No loss of semantics due to software layering.
    - Avoid layer instability.
    - Zero black box operation.
- Portability on reconfigurable parts is key.
- C model does not fit on-chip multiprocessing and has complex syntax.

Lava, Satnam Singh

- Synthesis not possible with Lava - only a structured definition tool.
- No CLB locations - a little bit of architecture independence.
- Gives you back the composite circuit.
- Humans better at placing (layout) than the computer.
- Computer better at routing.

The FURI Runtime Reconfigurable Environment, Adam Donlin

- The ultimate RISC (URISC).
- Single instruction processor - move memory to memory.
- Migrate compute elements to system bus.
- Flexible URISC (FURI).
- Exploit dynamicism - expand and contract system bus.
- Self modifying circuitry: memory mapped configuration SRAM. Self modification considered a taboo in modern systems.
- Must be able to configure on-line - partially reconfigurable architectures are essential.
- Non-technical requirement - open architecture FPGA.
- Can be static (predefined configuration schedule) or dynamic (primitive operating system).
- Offline store of circuitry available on request.

Reconfigurable Functionality - The OS Perspective, Michael Dales

- Add configurable units to CPU ALU.
- OS needs to be altered to support addition so no processes reconfigure units being used by other processes.
- Single large resource is bad.
- Resource management - divide the available resources.
- Loading circuits - what happens in an interrupt occurs?
  - Delay interrupt?
  - Back off and restart?
- Checkpoints during loading to handle interrupts.

Operating Systems Support for Dynamically Reconfigurable Architectures, Oliver Diessel

- Motivation:
  - Growing size of reconfigurable logic resource.
  - Growing range and integration of applications.
  - Orientation towards real time tasks.
- How do you manage and support logic resources in a multitasked environment.
  - Space/time.
  - Fixed/variable partitioning.
- Fixed partition size advantages:
  - Easy to design for.
  - Fault tolerant.
  - Manageable.
  - Location independent.
- Fixed partition size disadvantages:
  - Task performance can suffer.
  - Fragmentation.
- Variable partition size advantages:
  - Optimal performance i.e. right amount or resource allocated.
  - Better utilisation.

- Variable partition size disadvantages:
  - Scheduling complexity to cope with dynamic resource availability.
  - External fragmentation.
- Coping with variable size tasks:
  - Wait for or pre-empt executing tasks.
  - Move executing tasks to free up space (garbage collection).
  - Adapt incoming task to space available at runtime. Requires fast partitioning, placement and routing.
- Long term goals are to develop accurate and efficient indicators to accept or reject a task.

Task Rearrangement of Partially Reconfigurable FPGAs With Restricted Buffer, Martin Middendorf

- Examines the placement of tasks in a reconfigurable array.
- Total Reconfiguration: All tasks stopped and rearranged.
- Partial Reconfiguration: Single tasks relocated while others execute.
- Input data streams of the suspended tasks must be buffered.
- Use genetic algorithm to schedule tasks.

Devil's Advocate - The Von-Neumann Bottleneck and Other Myths, Mark Shand

- Reconfigurable can do real time i.e. nanoseconds to 10s of microseconds.
- Reconfigurable usually thought of as power efficient - but watch out though for the Transmeta.
- Right granularity is important: Bit, functional unit, CPU, multiprocessor.
- Instruction set is market driven.

Dynamically Reconfigurable Logic and System on Chip, Patrick Lysaght

- Dynamically reconfigurable: Devices that can be selectively reconfigured while running.
- In other words, the floor plan is a function of time.
- With more logic and more tasks on an FPGA it is less likely that all tasks execute simultaneously.
- Difference between adjacent logic families in FPGAs is increasing.
- Deep sub-micron (DSM) is a device with a transistor pitch <0.5μm.

- DPCS: Delay and power calculation system.

Associative Parallel Processing for Logic Event-Driven Simulation (APPLES), Damian Dalton

- Minimise support tasks.
- Identify essential element operations.
- Retain functionality.
- Normally only 5-10% of gates in a circuit are active.
- Current simulators evaluate all gates and therefore have 90-95% redundancy.
- Problem lies in active gate identification.

Cyclic Cutwidth of Meshes, Ondrej Sýkora

- Motivation for meshes:
    - Natural data structure in many problems (e.g. the matrix).
    - Constant degree.
    - Many algorithms developed.
    - As parallel or distributable computing systems: implementable with short wires therefore little signal propagation delay.

Keith J. Symington

List of Participants

Dagstuhl seminar 00261

Date: 25.06.2000 - 30.06.2000

Title: Dynamically Reconfigurable Architectures

Webpage: http://www.dagstuhl.de/DATA/Participants/00261.html

Marcus Bednara
Universität Paderborn
FB 14 - Elektrotechnik
Pohlstr. 47-49
Postfach 162 E
D-33098 Paderborn
D
phone: +49-5251-60-3921
fax: +49-5251-60-4221
e-mail: bednara@date.upb.de
url: http://www-date.uni-paderborn.de/

Yosi Ben-Asher
Haifa University
Dept. of Computer Science
IL-32000 Haifa
IL
phone: +972-4-8240338
e-mail: yosi@cs.haifa.ac.il
url: http://cs.haifa.ac.il/yosi

Neil W. Bergmann
Queensland University of Technology
CRC - Satellite Systems
2 George Street
GPO Box 2434
QLD 4001 Brisbane
AU
phone: +61-7-3864-2785
fax: +61-7-3864-1516
e-mail: n.bergmann@qut.edu.au
url: http://www.crcss.qut.edu.au

Gordon Brebner
University of Edinburgh
Division of Informatics
James Clerk Maxwell Building
Mayfield Road
EH9 3JZ Edinburgh
GB
phone: +44-131-650-5180
fax: +44-131-667-7209
e-mail: G.Brebner@ed.ac.uk

Michael Dales
University of Glasgow
Dept. of Computing Science
17 Lilybank Gardens
G12 8RZ Glasgow
GB
phone: +44-141-330-6297
fax: +44-141-330-4913
e-mail: michael@dcs.gla.ac.uk
url: http://www.dcs.gla.ac.uk/~michael

Damian Dalton
University College Dublin
Dept. of Computer Science
IRL - 4 Dublin
IRL
phone: +353-1-7062477
fax: +353-1-269-7262
e-mail: damian.dalton@ucd.ie
url: http://www.cs.ucd.ie/staff/ddalton/

Oliver Diessel
The University of New South Wales
School of Computer Science and Engineering
NSW 2052 Sydney
AU
phone: +61-2-9385-5922
fax: +61-2-9385-5995
e-mail: odiessel@cse.unsw.edu.au
url: http://www.cse.unsw.edu.au/~odiessel

Adam Donlin
University of Edinburgh
Dept. of Computer Science
James Clerk Maxwell Building
Mayfield Road
EH9 3JZ Edinburgh
GB
phone: +44-131-650-5144
fax: +44-131-667-7209
e-mail: adamd@dcs.ed.ac.uk
url: http://www.dcs.ed.ac.uk/~adamd/

Rolf Drechsler
Universität Freiburg
Institut für Informatik
Am Flughafen 17
D-79110 Freiburg
D
phone: +49-761-203-8145
fax: +49-761-203-8142
e-mail: drechsle@informatik.uni-freiburg.de
url: http://www.informatik.uni-freiburg.de/~drechsle

Hossam El Gindy
University of New South Wales
School of Computer Science and Engineering
Office EE Bldg. Room 339
NSW 2052 Sydney
AU
phone: +61-2-9385-4034
fax: +61-2-9385-5995
e-mail: elgindy@cse.unsw.edu.au
url: http://www.cse.unsw.edu.au/school/people/info/elgindy.html

Dietmar Fey
Universität Siegen
Inst. für Rechnerstrukturen
Hölderlinstr. 3
D-57068 Siegen
D
phone: +49-271-740-2474
fax: +49-271-740-2473
e-mail: fey@rs.uni-siegen.de
url: http://www.rs.uni-siegen.de

Reiner W. Hartenstein
Universität Kaiserslautern
FB Informatik
Lehrstuhl Rechnerstrukturen
Postfach 3049
D-67653 Kaiserslautern
D
phone: +49-631-205-2606
fax: +49-631-2052640
e-mail: hartenst@rhrk.uni-kl.de
url: http://xputers.informatik.uni-kl.de

Rainer Kress
Infineon Technologies AG
Corporate Development
Otto-Hahn-Ring 6
81739 München
D
phone: +49-89-234-50806
fax: +49-89-234-44950
e-mail: rainer.kress@infineon.com
url: http://www.infineon.com

Manfred Kunde
Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
FG Automaten und Formale Sprachen
Postfach 100565
D-98684 Ilmenau
D
phone: +49-3677-692-766
fax: +49-3677-69 12 37
e-mail: kunde@theoinf.tu-ilmenau.de
url: http://eiche.theoinf.tu-ilmenau.de

Steffen Köhler
TU Dresden
Fakultät Informatik
D-01062 Dresden
D
phone: +49-351-463-8452
fax: +49-351-463-8324
e-mail: stk@ite.inf.tu-dresden.de

Dominique Lavenier
Université de Rennes 1
Campus de Beaulieu
Avenue du Général Leclerc
F-35042 Rennes
F
phone: +33-2-99 84 72 17
fax: +33-2-99 84 71 71
e-mail: lavenier@irisa.fr
url: http://iww.irisa.fr/api/Lavenier/

Patrick Lysaght
University of Strathclyde
Dept. of E&EE
204 George Street
G1 1XW Glasgow
GB
phone: +44-1548-2249
fax: +44-1552-4968
e-mail: p.lysaght@eee.strath.ac.uk
url: http://www.eee.strath.ac.uk/staff/

Falko Mattasch
Friedrich-Schiller-Universität
LST für Rechnerarchitektur
Fakultät für Mathematik u. Informatik
Ernst-Abbe-Platz 1-4
D-07743 Jena
D
e-mail: mattasch@informatik.uni-jena.de

Martin Middendorf
Universität Karlsruhe
Institut AIFB
D-76128 Karlsruhe
D
phone: +49-721-608-3705
fax: +49-721-693-717
e-mail: mmi@aifb.uni-karlsruhe.de
url: http://www.aifb.uni-karlsruhe.de/~middendorf

Viorel Onofrei
Technical University
Dept. of Computer Science
B-dul Dimitrie Mangeron Nr. 53A
RO-6600 Iasi
RO
phone: +40-32-232-430
fax: +40-32-214-290
e-mail: onofrei@tuiasi.ro
url: http://eureka.cs.tuiasi.ro/~onofrei

Tobias Oppold
Universität Tübingen
Wilhelm-Schickard-Institut für Informatik
Raum 126
Sand 13
D-72076 Tübingen
D
phone: +49-7071-29-78977
fax: +49-7071-29-5062
e-mail: oppold@informatik.uni-tuebingen.de
url: http://www-ti.informatik.uni-tuebingen.de/~oppold/

Marco Platzner
ETH Zürich
Institut für Techn. Informatik und Kommunikationsnetze
TIK - ETZ G-85
Gloriastr.35
CH-8092 Zürich
CH
phone: +41-1-632-7544
fax: +41-1-632-1035
e-mail: platzner@tik.ee.ethz.ch
url: http://www.tik.ee.ethz.ch/~platzner

Bernard Pottier
Université de Bretagne Occidentale
Dept. Informatique
6 avenue Victor Le Gorgeu
BP 809
F-29287 Brest
F
phone: +33-2-98 01 62 17
fax: +33-2-98 01 62 16
e-mail: pottier@univ-brest.fr
url: http://penarvir.univ-brest.fr/~pottier/

Frederic Raimbault
Université de Bretagne Sud
UFR SSI
1 Rue de la Loi
F-56000 Vannes
F
e-mail: frederic.raimbault@univ-ubs.fr

Sergej Sawitzki
TU Dresden
Fakultät Informatik
Dü 242
D-01062 Dresden
D
phone: +49-351-4638458
e-mail: ss9@irz.inf.tu-dresden.de
url: http://www.inf.tu-dresden.de/~ss9/

Hartmut Schmeck
Universität Karlsruhe
Institut AIFB
D-76128 Karlsruhe
D
phone: +49-721-608-4242
fax: +49-721-608-6581
e-mail: schmeck@aifb.uni-karlsruhe.de
url: http://www.aifb.uni-karlsruhe.de/~schmeck

Heiko Schröder
Nanyang Technological University
School of Computing Engineering
Nanyang Avenue
639798 Singapore
SGP
phone: +65-790 6045
fax: +65-791 5927
e-mail: asheiko@ntu.edu.sg

Mark Shand
Compaq Systems Research Center
130 Lytton Ave.
CA94301 Palo Alto
USA
phone: +1-650-853-2236
fax: +1-650-853-2237
e-mail: shand@pa.dec.com
url: http://www.research.digital.com/SRC/staff/shand/bio.html

Satnam Singh
Xilinc Inc.
2100 Logic Drive
CA 95124 San José
USA
e-mail: satnam.singh@xilinx.com
url: http://www.xilinx.com

John Snowdon
Heriot-Watt University
Dept. of Physics
Riccarton Campus
EH14 4AS Edinburgh
GB
phone: +44-1-31-451-3026
fax: +44-1-31-451-3136
e-mail: j.f.snowdon@hw.ac.uk

Rainer G. Spallek
TU Dresden
Fakultät Informatik
Inst. für Technische Informatik
D-01062 Dresden
D
phone: +49-351-463-8243
fax: +49-351-463-8324
e-mail: rgs@ite.inf.tu-dresden.de

Ondrej Sykora
Loughborough University of Technology
Dept. of Computer Studies
LE 11 3TU Loughborough
GB
phone: +44-1509-222692
fax: +44-1509-211586
e-mail: O.Sykora@lboro.ac.uk
url: http://www.lboro.ac.uk

Keith Symington
Heriot-Watt University
Dept. of Physics
Riccarton Campus
EH14 4AS Edinburgh
GB
phone: +44-131-451-3040
fax: +44-131-451-3136
e-mail: kjsymington@iee.org

Uwe Tangen
GMD
BioMip - Biomolecular Information Processing
Schloss Birlinghoven
D-53754 St. Augustin
D
phone: +49-2241-14 15 30
fax: +49-2241-14 15 11
e-mail: tangen@gmd.de
url: http://www.gmd.de/

Simon Taylor
University of Edinburgh
Division of Informatics
James Clerk Maxwell Bldg.
Mayfield Road
EH9 3JZ Edinburgh
GB
fax: +44-131-667-7209
e-mail: sim@dcs.ed.ac.uk

Jürgen Teich
Universität Paderborn
FB 14 - Elektrotechnik
Pohlstr. 47-49
Postfach 162 E
D-33098 Paderborn
D
phone: +49-5251-603-002
fax: +49-5251-603-424
e-mail: teich@date.uni-paderborn.de
url: http://www-date.uni-paderborn.de/

Jozsef Vasarhelyi
University of Miskolc
Dept. of Automation
Egyetemvaros
H-3515 Miskolc
H
fax: +36-46-431-822/+40-64-194-822
e-mail: vajo@mazsola.iit.uni-miskolc.hu

Bernd Vettermann
Fachhochschule Mannheim
Hochschule für Technik & Gestaltung
Windeckstr. 110
D-68163 Mannheim
D
phone: +49-621-292-6313
fax: +49-621-292-6454
e-mail: b.vettermann@fh-mannheim.de

Klaus Waldschmidt
Universität Frankfurt
FB 20 Informatik
Techn. Informatik
Robert-Mayer-Str. 11-15
PF 11 19 32
D-60054 Frankfurt
D
phone: +49-69-798-28248
fax: +49-69-798-22351
e-mail: waldsch@ti.informatik.uni-frankfurt.de

Thomas Worsch
Universität Karlsruhe
Fakultät für Informatik
Am Fasanengarten 5
Postfach 6980
D-76128 Karlsruhe
D
phone: +49-721-608-43 11
fax: +49-721-69 86 75
e-mail: worsch@ira.uka.de
url: http://liinwww.ira.uka.de/~worsch/

Xin Yao
University of Birmingham
School of Computer Science
Edgbaston
B15 2TT Birmingham
GB
phone: +44-121-414-3747
fax: +44-121-414-4281
e-mail: X.yao@cs.bham.ac.uk
url: http://www.cs.bham.ac.uk/~xin/

Eberhard Zehendner
Universität Jena
Institut für Informatik
Ernst-Abbe-Platz 1-4
D-07743 Jena
D
phone: +49 3641 946385
fax: +49 3641 946372
e-mail: zehendner@acm.org
url: http://www2.informatik.uni-jena.de/~nez/