INTERNATIONALES BEGEGNUNGS- UND FORSCHUNGSZENTRUM FÜR INFORMATIK

Schloß Dagstuhl

Seminar Report 9808

Dynamically Reconfigurable Architectures

February 22 – 27, 1998

Overview

The Dagstuhl Seminar on “Dynamically Reconfigurable Architectures” brought together 40 participants from 8 different countries and from very diverse areas:

- “algorithms/RMesh people” designing efficient algorithms for processor arrays with reconfigurable bus systems,
- “FPGA people” using field programmable gate arrays for fast and flexible prototyping and designing tools supporting hardware/software codesign,
- “optical people” designing fascinating new microoptical systems for fast communication at incredible bandwidth.

All the 29 talks treated some aspect of dynamically reconfigurable architectures, some were survey talks giving an introduction into the different areas. "Dynamical reconfiguration" refers to reconfigurable interconnection systems, being used for "computing in space", and to reconfigurable logic cells or processing elements which can be programmed by look up tables (LUTs).

The talks emphasized that technological advances have opened up new ways of implementing complex systems in a way that blurred the barriers between hardware and software components development, and that existing design tools do not seem to be adequate for the necessary new design styles. For example, dynamically reconfigurable hardware allows to swap "soft cells" on demand in and out of hardware, requiring new 2-dimensional scheduling (or paging) strategies. It is even possible to let hardware evolve by itself, learning the required functions, or simulating biochemical processes. New advances in optical communication all of a sudden lead to feasible implementations of interconnection structures which so far had been seen as having only theoretical value.

The pleasant atmosphere of Schloß Dagstuhl was an important incentive for the lively interaction between the participants. We would like to thank all who contributed to the success of the seminar. Furthermore, we gratefully acknowledge the financial support by the TMR Programme of the European Community.

Karl-Heinz Brenner, Hossam ElGindy, Hartmut Schmeck, Heiko Schröder

Participants

Jürgen Becker, Darmstadt, Germany
Neil W. Bergmann, Brisbane, Australia
Gordon Brebner, Edinburgh, Great Britain
Karl-Heinz Brenner, Mannheim, Germany
Catherine Dezan, Brest, France
Adam Donlin, Edinburgh, Great Britain
Hossam ElGindy, Callaghan, Australia
Dietmar Fey, Jena, Germany
Reiner W. Hartenstein, Kaiserslautern, Germany
Gunter Haug, Karlsruhe, Germany
Michael Herz, Kaiserslautern, Germany
Thomas Hoffmann, Kaiserslautern, Germany
Michael Kaufmann, Tübingen, Germany
Rainer Kress, München, Germany
Manfred Kunde, Ilmenau, Germany
Loic Lagadec, Brest, France
Dominique Lavenier, Rennes, France
Christian Lengauer, Passau, Germany
William P. Marnane, Cork, Ireland
Martin Middendorf, Karlsruhe, Germany
Ronald Moore, Frankfurt, Germany
Ulrich Nageldinger, Kaiserslautern, Germany
Stephan Olariu, Norfolk, USA
Laurent Perraudeau, Rennes, France
Bernard Pottier, Brest, France
Andreas Reinsch, Jena, Germany
Sergej Sawitzki, Dresden, Germany
Manfred Schimmler, Braunschweig, Germany
Hartmut Schmeck, Karlsruhe, Germany
Heiko Schröder, Loughborough, Great Britain
John Snowdon, Edinburgh, Great Britain
Rainer G. Spallek, Dresden, Germany
Carsten Steckel, Karlsruhe, Germany
Ondrej Sykora, Bratislava, Slovakia
Uwe Tangen, Jena, Germany
Jürgen Teich, Zürich, Switzerland
Ralph Weper, Jena, Germany
Thomas Worsch, Karlsruhe, Germany
Eberhard Zehendner, Jena, Germany
Karl-Heinz Zimmermann, Hamburg, Germany

Contents

MARTIN MIDDENDORF
Models of Computation for Reconfigurable Arrays

RONALD MOORE
Automatic and Adaptive Mapping of Programs and Data onto (Heterogeneous) Hardware

STEPHAN OLARIU
List Ranking on the Reconfigurable Mesh and Related Problems

STEPHAN OLARIU
Algorithms for Arrays with Reconfigurable Busses

BERNARD POTTIER
Madeo: Object Oriented Programming, Modelization and Tools for FPGAs

SERGEJ SAWITZKI
CoMPARE: An Approach to Design of Dynamically Reconfigurable Processor Architecture

MANFRED SCHIMMLER
The Bus Concept of Systola 4096

HARTMUT SCHMECK
Sparse Matrix Multiplication on Dynamically Reconfigurable Arrays

HEIKO SCHRÖDER
Physical and Technical Limits for Reconfigurable Architectures

JOHN SNOWDON
Optics in Computing, Pespectives and Possibilities

CARSTEN STECKEL
A Simulator for Reconfigurable Meshes

ONDREJ SYKORA
Optical All–to–All Communication in some Product Graphs and Permutation Communications in All–Optical Rings

UWE TANGEN
Hardware Evolution on POLYP

KARL-HEINZ ZIMMERMANN
Polynomial Scheduling

GORDON BREBNER's personal summary
Dynamically Reconfigurable Architectures

Abstracts

The Impact of Reconfigurable Hardware Platforms on the Overhead in Parallelism

by JÜRGEN BECKER

The talk describes a new co-compiler and its "vertical" parallelization method for coarse-grained reconfigurable Hardware Platforms. This includes a general model for co-operating microprocessor / accelerator platforms and new parallelizing compilation techniques derived from it. To achieve optimized speed-ups and hardware resource utilization, novel vertical parallelization techniques involving parallelisn exploitation at four different levels (task-, loop-, statement- and operation-level) are explained, archieved by configurable Xputer-based accelerators.

The focus of the talk is on the new developed "Vertical Hyperplane Theorem" for parallelization of nested accelerator loops and optimizing loop transformation based theorem.

Moreover, the performance analysis methods for analyzing microprocessor / accelerator(s) symbiosis in early implementation phases are sketched, which are needed for controlling the microprocessor / accelerator(s) partitioning process to optimize complete application times.

Identifying Characteristics of Algorithms which make them Suitable for Reconfigurable Computing Implementations

by NEIL W. BERGMANN

When FPGAs are incorporated within a stored program computer, the result is a machine where the programmer can design both the software for the machine and the hardware that will execute that software. Such a machine, where the hardware can be reconfigured or customised on a program-by-program basis, is called a reconfigurable computer or a custom computer.

This talk shows that:

- Custom computers allow fine-grain parallel processing of demanding data-rate signals such as real-time video.

- Even a single FPGA can provide 50-100 times speedups compared to conventional computers.

- LUT structures based on distributed arithmetic provide twice the space performance of conventional arithmetic.

Remaining challenges include the need to identify useful problems which exhibit performance enhancements through techniques like

- many small, parallel PEs
- memory data re-use
- direct access to multimedia data streams.

Dynamically Reconfigurable FPGAs

by GORDON BREBNER

This introductory talk surveys the general architectural principles of Field Programmable Gate Array (FPGA) technologies. First, the pre-history is reviewed, leading from externally-programmed, two-level array technologies (PAL, PLA) to internally-programmed, multiple-level FPGA technologies. The basic components of an FPGA architecture are: configurable function blocks; configurable switchboxes and local interconnect; non-local interconnect; and configuration memory. Complexities of function blocks vary, typically boolean functions of between two and five variables, plus one or two flip-flops. There may be extras, such as dedicated carry logic, or memory structures. Blocks are usually laid out as a regular two-dimensional array. Interconnection includes nearest-neighbour between blocks and/or separate routing channels with switchboxes at intersections. For dynamic reconfigurability, configuration memory should be SRAM-based, with random access (as opposed to serially-accessed, as in many current FPGA technologies). The Xilinx XC6200 has several novel features of interest to dynamic reconfiguration, and it is contrasted with other current technologies: Xilinx XC4000, Altera Flex 10K and Atmel AT40K. Future major research issues are: the distinction between 'hard' and 'soft' (circuitry can now be soft); FPGAs as system components; and the current wide variety of FPGA styles - is there an ideal FPGA architecture?

Soft Circuitry

by GORDON BREBNER

This talk is about a software-oriented approach to configurable circuitry. The idea is to seek factors common to stored circuitry and stored programs - essentially, both are state machines computing outputs from inputs. Some usual differences are: parallel vs. sequential; two- vs. one-dimensional; tighter vs. looser structure; and synchronous vs. asynchronous. Neither conventional hardware circuit design tools nor conventional high-level programming languages are ideal for software circuit design - work at Edinburgh is seeking an alternative, starting by investigating lower-level notations. As a system component, soft circuitry has a hardware and a software environment - this necessitates standard interfaces at various levels: physical, bit data and functional. 'Virtual cicuitry' is the soft circuitry analogue of virtual memory, with 'Swappable Logic Units' (SLUs) the analogue of pages/segments. Two operating system-provided environments investigated are the 'sea of accelerators' and the 'parallel harness' - both raise issues of dynamic SLU placement and routing, but are constrained enough to make this practical. Another interesting use of soft circuitry under investigation is 'mobile circuitry' - where applets are expressed as circuits in a network computing environment; a prototype has been produced. Two key issues are: how can circuits be represented in a portable (FPGA-independent) form? and what execution environment should be provided for circlets? At the moment, a fine-grain logic gate array model looks most promising.

Technologies for Realizing Dynamically Reconfigurable Optical Interconnections

by K.H. BRENNER

Optical interconnection technology has reached a high level of maturity for long distance interconnections. The receivers and transmitters can provide high modulation bandwidth to support communication at tens of GB/s. Fully packaged connector modules allow easy implementation of optically interconnected systems. Using WDM, the bandwidth of the transport medium can be further multiplied and enables fully interconnected systems which may be reconfigured by wavelength tuning. For short distance communication free-space systems are attractive since they offer also spatial bandwidth in addition to temporal bandwidth and they can be integrated into small threedimensional modules. The mechanisms for reconfiguration are either based on mechanical or electrooptical effects. For the mechanical case, reconfiguration times are limited to the order of a millisecond. Electrooptic effects enable switching in the nanosecond regime but the power requirements

are difficult to meet. An interesting possibility of electrooptic reconfiguration is opened up by the concept of a free-space holographic switch.

Runtime Reconfigurable Routing

by ADAM DONLIN

In the context of Virtual Circuitry systems - systems where the dynamically reconfigurable nature of FPGA's is exploited to provide time division multiplexing of circuitry components on a cell array - this talk addresses practical techniques to support flexible aspects of the traditional place and route cycle at runtime.

The timescales of place and route are typically in stark contrast to those of dynamic reconfiguration. - APR on the scale of minutes, even hours and reconfiguration on the scale of nano $\rightarrow$ miroseconds. To retain an element of flexibility we explore a continuum of routing frameworks which trade off degrees of parallelism and degrees of flexibility. Typically these frameworks support detailed interactions of circuitry components (sw_s) resident on the array. Three points on this continuum are highlighted. These represent a model which supports highly parallel interactions with an explicit routing framework - but with limited/coarse scale reconfigurations. To address practicality we consider the configuration overhead of the harness, the system overhead of calculating the harness/framework and the amount of array resources consumed. A midpoint model that supports increased flexibility yet sacrifices the degree of parallelism utilises an incomplete routing harness with key switching points in the routed connections left incomplete fine grain partial reconfiguration is exploited to rapidly and practically establish and break connections in the routing harness. A final model, exploiting the ultimate RISC processor architecture, provides a novel means of supporting highly flexible - yet serialised - communication/routing between circuitry components. Essentially, routing is characterised in software - potentially algorithmically.

Partial Rearrangements of Space-Shared FPGAs

by HOSSAM ELGINDY

Dynamically reconfigurable field programmable gate arrays naturally support high performance computing applications, and more interestingly can be reconfigured on-line to reduce total hardware requirements. This talk reports on joint work, with Oliver Diessel, on methods for supporting the presence of multiple circuits on a single or many FPGA

chips. We present an approach based on partial rearrangement of a subset of executing tasks to alleviate the fragmentation problem. Methods for overcoming the NP-hard problems of identifying feasible rearrangements and scheduling are reported.

Computing with Linear Arrays with Pipelined Optical Buses

by HOSSAM ELGINDY

Pipelined optical bus systems are capable of performing arbitrary permutation between attached nodes, and thus function as a reconfigurable network. This talk reports on early work into the use of such network to efficiently implement commonly used algorithms. In particular, a polylogarithmic worst-case running time sorting algorithm is presented.

An attempt to define the computation model based on realistic/practical consideration is also pursued.

Reconfigurable Architectures by Means of Optoelectronic Circuits

by DIETMAR FEY

Current trends in optoelectronic very large scale integrated (OE-VLSI) circuits are presented as well as first experiences, gained with an own designed OE-VLSI circuit. A parallel optoelectronic interface for VLSI circuits can help to overcome pin limitation and delays caused by long on-chip wires. Optical interconnects offer the following advantageous features compared to electrical interconnects in higher connectivity and information density, higher interconnection bandwidths, a lower signal and clock skew and the possibility to realize 3-D architectures and circuits. Especially the last point is the most interesting one for circuit designers and computer architects. For example, a parallel 3-D optical interface for a FPGA eases dynamic and partial reconfigurability.

The most promising OE-VLSI technologies appropriate for reconfigurable hardware are SEED (self electro optic effective device) based on VCSEL (vertical cavity surface emitting laser diode) based systems. In such systems either an array of SEEDs, which is an externally controllable modulator, or an array of interlaced VCSELs on photo diodes is flip-chip bounded on a CMOS circuit. Currently SEED OE-VLSI systems represent the most advanced technology. VCSEL based systems require a more sophisticated manufacturing. However, it is assumed that the future will belong to VCSEL based systems because the optical interconnection scheme is easier than in SEED based systems.

In the talk different possibilities for reconfigurable hardware using parallel optical interconnections are shown. The main focus lies on an own designed test circuit in SEED OE-VLSI technology using a 0,5 mm CMOS process. This circuit contains look-up tables (LUTs) that can be dynamically and simultaneous reconfigured via optical input pads. Simulations in SPICE show an access rate up to 250 MHz for all LUTs. The size of a 2×2 LUT is 24000 mm^2 leading to a density of about 4000 LUTs per cm^2.

A Revival of Systolic Arrays by Course Granularity Reconfigurable Circuits

by R. HARTENSTEIN (joint work with M. Herz, T. Hoffmann and U. Nageldinger)

In a typical FPGA only one out of 100 - 200 transistors realy serves the application, whereas the rest is reconfigurability overhead. The paper introduces the mesh-connected reconfigurable KRESS ARRAY, having a logic integration density being better by about three orders of magnitude. The paper also briefly recalls the methods having been implemented for mapping an application onto a KRESS ARRAY. (Very High Level Synthesis). It illustrates the feasibility that this coarse granularity dynamically reconfigurable computing platform has the potential to overcome general purpose computing.

Reconfigurable Logic as Shared Resource in Multipurpose Computers

by GUNTER HAUG (joint work with W. Rosenstiel)

One of the intended applications of the Xilinx XC6200 family is to be used as reconfigurable coprocessor to accelerate software. This approach has failed so far because of the lack of software support. First, there was no programming method similar to conventional coding. Second, there was no sufficient run time environment.

Our approach is, to provide both, a design flow software designers are familiar with, and an integrated run time environment.

The runtime environment is an extended version of the Linux operating system. In this system the reconfigurable coprocessor is a shared resource administrated by the kernel like all other resources. If the scheduler decides to perform a context switch the configuration of the coprocessor is also changed.

The design flow starts with only one C–specification. Functions, that appear in hts_fork calls are intended to go into hardware. So the design flow is split into two branches: software and hardware. The hardware branch is performed by high level synthesis, logic synthesis, placement and routing. The software branch is mainly generation of communication functions and compilation. After both branches are complete, the results - compiled program and configuration bits - can be linked together to one binary capable to run on the extended operating system.

Dynamically Reconfigurable Architectures in Embedded Systems

by RAINER KRESS

Reconfigurable architectures are more and more used in current design flows. Traditionally, reconfigurable architectures are used for emulation and rapid prototyping of embedded systems. In the product planning phase, animated prototypes based on field-programmable gate array (FPGA) technology give customers a better look and feel of the final product.

Next generation embedded systems place new demands on an efficient methodology for their design and verification. These systems have to support interaction over networks, multiple concurrent applications, changing operating conditions, and changing customer requirements. Therefore, besides existing requirements like low cost and high performance, new demands like adaptivity and reconfigurability, also during run-time, arise.

When developing complex systems, intelligent trade-offs between hardware and software components are necessary to deliver the design best satisfying performance and cost constraints. Therefore, designers need a complete development framework that facilitates such trade-offs. A new concept for tool-assisted design exploration and fast prototyping of hardware/software systems has been proposed. Starting from a system specification in Java, a novel design flow has been presented which is targeted to next generation embedded systems including reconfigurable hardware. It supports trade-off evaluation, automated interface generation and verification of the design by co-emulation for traditional as well as dynamically reconfigurable architectures.

Sorting and Routing on ℓ-Constrained Reconfigurable Meshes

by MANFRED KUNDE

In an ℓ-constrained reconfigurable mesh a data item can be transported in one step on a bus of length at most ℓ. We show that we can embed ℓ independent $\frac{n}{\ell}\times\frac{n}{\ell}$ static meshes into one $n \times n$ reconfigurable mesh such that the maximal bus length is ℓ. Using the embedded $\frac{n}{\ell}\times\frac{n}{\ell}$ meshes for performing the so-called all-to-all mapping h-h sorting can be solved in $hn + o(n)$ steps provided $h\ell \geq 8$. The technique can be extended to r-dimensional reconfigurable meshes with side length n. For optimal results the bus length ℓ should fulfill $h\ell \geq 40$. Following the placements of queens in the modular ℓ-queens problems on an $\ell \times \ell$ chessboard the embedding method also works for reconfigurable meshes with diagonals. h-h sorting can then be done in $\frac{h}{3}n + o(n)$ steps with ℓ-constrained buslength with $\ell \geq \frac{13}{h}$.

Reconfigurable Co-Processors: from Nested Loops to FPGA Systolic Arrays

by DOMINIQUE LAVENIER

With the exponential growth in the level of integration, we are already seeing the arrival of commercial and specialized micro-processor with on-chip reconfigurable co-processors (e.g. National Semi-Conductor: NAPA 1000). There are very few tools for programming such processors, and definitively no high end tools. It is our contention that reconfigurable computing will not get wide acceptance unless tools become mature and are available to an ordinary programmer with little additional input. Hence, the tools must also exploit automatic parallelization in addition to standard computation technology.

Since regular systolic arrays have proved to be successful for reconfigurable computing, the tools will be based on the parallelization of critical loops on the Reconfigurable Co-processor. Our group is working on techniques for accelerating application programs on co-processor consisting of reconfigurable components.

Real Time OSP and Dynamically Reconfigurable FPGA

by WILLIAM MARNANE

FPGAs have the logic resources and speed capabilities to implement real time DSP algorithms, such as the Autoregressive Modified Covariance Spectral Estimator. However such complex algorithms require arithmetic functions such as division as well as the standard multiply accumulate operations. Due to the area restriction in FPGAs, a bit serial approach is required. This in turn results in a complex control path. No benefit is gained from the reconfigurability of the FPGA and the question is raised as to how we exploit this reconfigurability in real time DSP applications. The answer is in the elimination of the control path and achieves control through reconfiguration. However current design tools do not support this method and new techniques are required. Some central benefit can be gained in the meantime, through preserving regularity when mapping to the FPGA, through the elimination of directly storing constants in the circuit, and through the exploitation of run time dependencies. Future techniques that generate the control reconfiguration stream should exploit the well known synthesis techniques for systolic arrays.

Models of Computation for Reconfigurable Arrays

by MARTIN MIDDENDORF

An overview over various models of Dynamically Reconfigurable Processor Arrays is given. Also results are reviewed that compare the computational power of the different models. Main elements of these models are the processing elements (PEs), the buses, the topology of the underlying fixed connections and the possible connections that can be dynamically set up between the ports of a PE by using switches. The results reviewed include a characterization of the problems that can be solved in constant time by Reconfigurable Networks (RNs) using different types of switches. Also simulation results between RNs, PRAMs and Reconfigurable Meshes (RMs) as well as self-simulation results for RMs are considered. Finally several models which are related to reconfigurable arrays are discussed.

Automatic and Adaptive Mapping of Programs and Data onto (Heterogeneous) Hardware

by RONALD MOORE

Increasingly, the main barrier to wider acceptance of parallel computation is seen to be ease of programming. Programming parallel computers will remain significantly more difficult than programming conventional computers as long as the programmer bears the responsibility for distributing computation and data across the available processors. As such, parallel programming is currently only attractive to users whose need for performance justifies the high cost of developing parallel software. This problem is waiting for reconfigurable architectures as well.

At the Johann Wolfgang Goethe-University in Frankfurt, we have developed two architectures which use an associative communication crossbar to support automatically distributed data and computation. The crossbar can be seen as a data-driven self-reconfiguring bus.

The first of these is called ADARC (Associative Dataflow ARChitecture). A hardware prototype with 12 processors has been built, and scalability results are encouraging. The second architecture is in the initial development stages and is called SDAARC (Self-Distributing Associative ARChitecture). In SDAARC, each processor and local memory are augmented to function as an "attraction memory site" - one large, slow cache in a Cache Only Memory Architecture (COMA). COMA techniques used in the literature to distribute data are extended to distribute computation as well.

Finally, the author speculates that this system could be extended to allow almost seamless integration of heterogeneous processors, e.g. both conventional microprocessors and also Application Specific Instruction set Processors (ASIP's).

List Ranking on the Reconfigurable Mesh and Related Problems

by STEPHAN OLARIU

Finding a vast array of applications, the list ranking problem has emerged as one of the fundamental techniques in parallel algorithm design. Surprisingly, the best previously-known algorithm to rank a list of n items on a reconfigurable mesh of size $n \times n$ was running in $O(\log n)$ time. It was open for more than eight years to obtain a faster algorithm for this important problem.

Our main contribution is to provide the first breakthrough: we propose a deterministic list-ranking algorithm that runs in $O(\log^* n)$ time as well as a randomized one running in O(1) expected time, both on a reconfigurable mesh of size $n \times n$. Our results open the door to an entire slew of efficient list-ranking-based algorithms on reconfigurable meshes.

Algorithms for Arrays with Reconfigurable Busses

by STEPHAN OLARIU

This talk is essentially a survey or results - old and new - about reconfigurable architectures with a special focus on the reconfigurable mesh. The topics covered include:

- computer arithmetic
- graph algorithms
- computational geometry
- simulation of the PRAM

Algorithms in each of these areas are outlined. The hope is that the audience will get a taste for what the basic results are.

Madeo: Object Oriented Programming, Modelization and Tools for FPGAs

by BERNARD POTTIER (joint work with L. Lagadec)

Object technology is investigated for system integration and architecture synthesis for FPGAs. Sequential or combinational executions, communications, FPGA controls, ... are modeled in classes having specific cross compilers. Smalltalk-80 syntax is modified in order to provide types of different strength.

In the case of architecture synthesis, we use block closures to represent atomic computations in hardware. Then, these blocks are usable to describe algorithms as an example for regular architectures. A logic generator (LG) has been implemented. This LG requires the specification of definition sets (DS) for each block parameter. Each object appearing in a DS must be an instance of a class having support for binary encoding.

The synthesis process build a table with every possibility in the product of DS and the corresponding results. All these objects are translated to binary to produce a PLA that is split according to a target FPGA technology logic grain.

The practical interest of this work is illustrated by a XC6200 editor that provides a programmable interface for geometric operations (translations, rotations, ...). We give a demonstration of this editor.

CoMPARE: An Approach to Design of Dynamically Reconfigurable Processor Architecture

by SERGEJ SAWITZKI (joint work with R. G. Spallek)

We propose CoMPARE (Common Minimal Processor Architecture with Reconfigurable Extension), a simple microprocessor architecture. It uses a LUT-based reconfigurable logic array as an extension to a conventional ALU to build special instructions for acceleration of different applications. The integration of the reconfigurable resources at the instruction level allows the common programming model for microprocessors to be maintained, requiring only a modification of the compiler back-end. Several simulation runs with the behavioral VHDL model have shown that even a comparatively simple reconfigurable extension increases the performance by a factor of 2-4 for different applications. A prototype based on Xilinx XC6200 is currently under development. The timing analysis of the RTL model suggests that an implementation of CoMPARE on a Xilinx XC6264 FPGA will run at 11 MHz.

The Bus Concept of Systola 4096

by MANFRED SCHIMMLER

Systola 1024 is a parallel architecture with 1024 processors integrated on an add-on-board for PCs. The talk focuses on the strengths and weaknesses of this machine and on the concept of its sucessor, the Systola 4096.

The underlying model is the instruction systolic array (ISA). One of the most powerful features of an ISA is the capability of performing aggregate functions. These are global operations that can be executed with a constant number of instructions.

The ISA model is explained together with its implementation in Systola 1024. Some limits are pointed out. These include limitations in right-to-left communication, bus limitations, I/O-limitations and memory limitations. The concepts of Systola 4096 are presented that overcome these limitations.

Sparse Matrix Multiplication on Dynamically Reconfigurable Arrays

by HARTMUT SCHMECK (joint work with H. ElGindy, M. Kunde, M. Middendorf, H. Schröder and G. Turner)

The sparseness of an n×n-matrix A can be characterized by its maximal number of nonzero elements per row (r_A) or per column (c_A), or by its total number of nonzero elements divided by n (k_A). If c_A or r_A are constant, A is called column or row sparse, rsp., if both are constant, it is called uniformly sparse, and if only k_A is constant, it is called weakly sparse. Whereas on standard mesh-connected arrays sparse matrices cannot be multiplied faster than full matrices, the availability of segmentable buses on the dynamically reconfigurable array can lead to significant speed improvements. We present algorithms for computing C=A·B on an n×n reconfigurable array in time $O(c_A \cdot c_B)$, $O(r_A \cdot r_B)$, and $O(r_A \cdot c_B)$ which means constant time for appropriate column or row sparse operands. For A column sparse and B row sparse, a lower time bound of $\Omega\left(\sqrt{\max(c_A, r_B) \cdot n}\right)$ is obtained and a corresponding lower bound of $\Omega\left(\sqrt{k_A \cdot n}\right)$, if A is weakly sparse. For these cases we derive algorithms running in time $O\left(\sqrt{k_A \cdot r_B \cdot n}\right)$, $O\left(\sqrt{k_A \cdot c_B \cdot n}\right)$, and $O\left(\sqrt{c_A \cdot r_B \cdot n}\right)$, i.e. there still is a gap between the upper and lower bounds. If both A and B are weakly sparse, the best known lower and upper bounds are $\Omega\left(\sqrt{\max(k_A, k_B) \cdot n}\right)$ and O(n), i.e. in this case there is no speed improvement (so far) over the standard systolic algorithm. (*remark: during the seminar H. ElGindy and M. Middendorf improved the lower bound to* $O\left(max(k_A, k_B) \cdot \sqrt{n}\right)$). The presented algorithms can be transformed into a polyalgorithm choosing the most efficient variant depending on the parameters c_A, c_B, r_A, r_B, k_A, and k_B.

Physical and Technical Limits for Reconfigurable Architectures

by HEIKO SCHRÖDER

Looking back into the recent history of parallel computing shows several trends that are likely to continue into the near future:

1. High power computing (i.e. Tera-flop and beyond) can only be reached through massive parallelism and distributed memory.
2. No increase in investment into this area can be expected from industry or research institutions in the near future.
3. The number of manufacturers of such machines is decreasing.

4. Parallel computing will nevertheless occupy important niche markets with massively parallel special purpose architectures.

The market is certainly very competitive. Recently it has been demonstrated that reconfigurable mesh architectures with and without the use of optical connections can be built. In order to predict the exact performance of such machines and thus to be able to decide whether such concepts have a chance on the market, a range of technical and physical limitations have to be taken into account. The simplest of these is the speed of light.

Such architectures promise constant diameter (better than the hypercube) and super-linear bisection width (better than the hypercube) – but are there enough applications that can make good use of these features?

Optics in Computing, Perspectives and Possibilities

by JOHN SNOWDON

The progress of Silicon electronics towards faster processors and greater number of off-chip connections is projected to lead to a requirement for aggregate data I/O-rates in excess of 1 $Tbits^{-1}$ within ten years. Free space optics is perhaps the only technology capable of meeting this demand within a reasonable power budget. Additionally free-space optics relays are capable of carrying several hundred thousand data channels. This implies that many chip outputs (e.g. 10 000 channels) may be interlaced within such a relay.

These relays are called optical highways and have the potential to support large numbers of wide point to point interconnects to implement topologies such as hypercubes, crossbars and cluster crossbars between many processors. The theoretical bandwidth available in such a highway is > 100.000 $Tbits^{-1}$.

In this talk, we present our latest experimental work. Two demonstrators have been under construction:

1. A special purpose sorter using a bitonic sorting algorithm and a perfect shuffle hard wired optical interconnection. This can sort 1024 16 bit numbers in 15 ms and demonstrates 200 $Gbits^{-1}$ of off-chip optical data rate.

2. An optical crossbar switch employing VCSEL technology. In addition a fast (order of clock cycles) beam steering device is discussed.

A Simulator for Reconfigurable Meshes

by CARSTEN STECKEL (joint work with M. Middendorf, H. ElGindy, and H. Schmeck)

The benefits of a simulation tool supporting algorithm development by visual and numerical output was presented. The tool simulates SIMD mesh connected processor arrays with conditional instruction control. Switches within the processors connect the four ports North, South, East and West to form interconnecting buses. The buses can be written to once and read from multiple times within one computational cycle. The structure of this cycle is free in a sense, that the developer is able to simulate different abstract machine models. This freedom is realized through an "end of computational cycle"-marker which is placed into the RAM based assembler instruction code. The dimensions of the mesh or torus, the number of registers, the width of the buses, and the radius that a signal can move per computational cycle can be set algorithm and model specific. Several examples of algorithms were presented to demonstrate the simulators abilities for visualizing algorithms and producing numerical output.

Optical All–to–All Communication in some Product Graphs and Permutation Communications in All–Optical Rings

by ONDREJ SYKORA (joint work with Mike Paterson (partly), Heiko Schröder and Imrich Vrto)

Optical networks is a very fastly developing new area of research. It is a key technology in communication networks and it is expected that it will dominate such important applications like video conferencing, scientific visualisation, real-time medical imaging, high speed super-computing, distributed computing (covering local to wide area). Networks which use optical transmission and maintain optical data paths through the nodes are called all-optical networks. All-optical networks exploit photonic technology for implementation of both switching and transmission functions so that the signals in the networks can be maintained in optical form and not to be converted during the transmission. This fact allows for much higher transmission rates since there is no overhead due to conversions to and from the electronic form during transmission. This is enabled by the technology of wavelength-division multiplexing. Wavelength-division multiplexing (WDM) partitions the optical bandwidth into a number of channels, and allows multiple laser beams to be propagated concurrently along the same optical fiber, on

distinct light channels (wavelengths). It is evident that the number of wavelength (well known as the optical bandwidth) is a limiting factor. To study the problem we model all-optical network as a symmetric directed graph. Vertices of the symmetric directed graph are assigned to the terminals and arcs to the optical fibers connecting the terminals of the network. A request to send a message from one terminal to the other is then the corresponding ordered pair of nodes. To a request one has to assign an directed path connecting the two nodes of the request. One directed path uses one wavelength. The set of requests which should be realized in the same time is usually called an instance of connectivity requests. To each instance of connectivity requests one has to assign a set of directed paths. If two directed paths share an arc they have to use different wavelengths. Our problem is to find a solution for an instance of connectivity requests such that the number of wavelengths is minimized. There are considered some classes of instances of connectivity requests.

An important instance is the so called all-to-all instance where all possible connectivity requests are realized.

We prove exact results on the number of wavelengths for equal odd sided 2-dimensional toroidal meshes and for rectangular 2-dimensional toroidal meshes with one side longer or equal than the double times the other. We proved also a couple of results for ordinary meshes. These results complete the results recently got for even sided square toroidal meshes by B. Beauquier.

We study also the wavelength problem and arc (edge) congestion problem for communicating permutation instances on a ring. We show that the numbers of wavelengths in the directed case $\vec{\omega}$, in the undirected case ω, and the arc congestion $\vec{\pi}$ and the edge congestion π for a permutation instance I_1 on an n-vertex ring C_n satisfy:

$$\vec{\omega}(C_n, I_1) \le \lceil \tfrac{n}{3} \rceil,\ \vec{\pi}(C_n, I_1) \le \lceil \tfrac{n}{4} \rceil,\ \omega(C_n, I_1) \le \lceil \tfrac{n}{2} \rceil,\ \pi(C_n, I_1) \le \lceil \tfrac{n}{2} \rceil$$

All bounds are the best possible for worst-case instances. From the algorithmic point of view, determining $\vec{\omega}(C_n, I_1)$ and $\omega(C_n, I_1)$ is NP-hard, which can be shown by a modification of the NP-hardness proof of the wavelength problem in rings, for general instances.

Hardware Evolution on POLYP

by UWE TANGEN

Programmable hardware is presented based on Field Programmable Gate Array technology which allows the implementation of a large spatial medium for the study of logical and physical models of evolution. In a first phase, a massively parallel computer was designed, NGEN, with configurable interface to a UNIX host workstation, broadband interconnect up to 30 and 144 tabula rasa processors (FPGAs) configurable down to the individual gate level. With a description and implementation of an evolution model simulating a real biochemically coupled isothermal amplification system (3SR, CATCH) the feasibility of this FPGA approach is shown.

As a successor of NGEN, POLYP, with actually 288 MBytes, 20 ns access time, distributed memory and 96 XC6264 chips from Xilinx and optical broadband interconnect is presented. With a hierarchical control structure POLYP is especially designed to tackle problems of hardware evolution.

A first design implementation for evolving a simple 8 bit wide bus is shown and discussed.

POLYP now offers for the first time the investigation of the important biological problem of the interaction between the description - via DNA - and functionality - via enzymes - in biology under evolutionary conditions in electronic hardware.

Polynomial Scheduling

by KARL-HEINZ ZIMMERMANN

In my talk, an introduction to the scheduling of recurrence equations with polynomial functions is given. It is well-known from approximation theory that each real-valued function on a finite domain is a polynomial function. Moreover, it is known that uniform recurrences defined on fat domains can be optimally scheduled by linear function. This is, however, wrong for affine recurrences.

Given an n-dimensional recurrence over a domain D, we derive a min-max problem such that in the outer minimization varies over a class of polynomial functions, e.g., linear, quadratic or cubic polynomials, and the inner maximization provides the latency of the schedules. This problem is a non-smooth convex optimization problem. We have used a bundle-trust method to solve this problem. For this, we need to provide in each iteration a solution of the inner maximization problem (we have done already this for quadratic or

convex functions) and a subgradient (this is here a trivial problem). For affine or uniform recurrences we have established constraints in order to reduce the number of constraints on causality.

The applicability of our method is demonstrated by an affine recurrence modeling optimal string parenthezisation.

Dynamically Reconfigurable Architectures

GORDON BREBNER's personal summary

(a) Learnt from algorithms / RMESH people:
- efficient algorithms are very delicate
- central concern is communication in space and time
- dynamic reconfigurability offers novel communication possibilities

(b) Learnt from (already knew!) from FPGA people:
- circuits are a software implementation possibility, alongside programs
- one particular aim is automatic partitioning into circuits + programs
- dynamic reconfigurability offers the mechanism for circuits as software

(c) Learnt from optical people:
- technology offers novel communication in time and space
- dynamic reconfigurability offers:
  - novel ways of configuring circuitry
  - novel ways of interconnection

(d) Learnt from general discussion:
- solutions invented should have some practical significance
- problems from applications should advice, but not drive, research

My overall architecture picture now:
- PEs may be like processors or like "circuit emulators"
- PEs interconnected, maybe optically
- dynamic reconfiguration within PEs and of interconnect

The big problem for this general architecture model:
- How can delicate algorithms be programmed in a high-level way, without expert knowledge of: programming, circuit design, parallel/distributed communication?

Dagstuhl-Seminar 9808 **List of Participants**

Jürgen **Becker**
Technische Universität Darmstadt
Institut für Datentechnik
FG Mikroelektr. Systeme
Karlstr. 15
64283 Darmstadt
Germany
tel: +49-6151-164337
fax: +49-6151-164936
e-mail: becker@mes.tu-darmstadt.de
http://www.microelectronic.e-technik.tu-darmstadt.de

Neil W. **Bergmann**
Queensland University of Technology
Space Centre for Satellite Navigation
2 George Street
GPO Box 2434
QLD 4001 Brisbane
Australia
tel: +61-7-3864-2785
fax: +61-7-3864-1516
e-mail: n.bergmann@qut.edu.au

Gordon **Brebner**
University of Edinburgh
Dept. of Mathematics
James Clerk Maxwell Building
Mayfield Road
EH9 3JZ Edinburgh
Great Britain
tel: +44-131-650-5180
fax: +44-131-667-7209
e-mail: gordon@dcs.edinburgh.ac.uk

Karl-Heinz **Brenner**
Universität Mannheim
Fakultät Mathematik und Informatik
B6,26 - Zi. 320
68131 Mannheim
Germany
tel: +49-621-292-3004
fax: +49-621-292-1605
e-mail: karl-heinz.brenner
@informatik.uni-mannheim.de

Catherine **Dezan**
Université de Bretagne Occidentale
Dept. Informatique
6 avenue Victor Le Gorgeu
BP 809
F-29287 Brest
France
fax: +33-2-98-01-62-16
e-mail: Catherine.Dezan@univ-brest.fr

Adam **Donlin**
University of Edinburgh
Dept. of Mathematics
James Clerk Maxwell Building
Mayfield Road
EH9 3JZ Edinburgh
Great Britain
tel: +44-131-650-5144
fax: +44-131-667-7209
e-mail: adamd@dcs.ed.ac.uk
http://www.dcs.ed.ac.uk/~adamd

Hossam **ElGindy**
University of Newcastle
Dept. of Electrical and Computer Engineering
EA-G19
University Drive
NSW 2308 Callaghan
Australia
tel: +61-49-21-5740
fax: +61-49-21-6993
e-mail: hossam@olive.newcastle.edu.au

Dietmar **Fey**
Friedrich-Schiller-Universität
LST für Rechnerarchitektur
Ernst-Abbe-Platz 1-4
07743 Jena
Germany
tel: +49-3641-8-46373
fax: +49-3641-8-46372
e-mail: dietmar.fey@uni-jena.de

Reiner W. **Hartenstein**
Universität Kaiserslautern
FB Informatik
Lehrstuhl Rechnerstrukturen
Postfach 3049
67653 Kaiserslautern
Germany
tel: +49-631-205-2606
fax: +49-631-2052640
e-mail: hartenst@rhrk.uni-kl.de

Gunter **Haug**
FZI Karlsruhe
Haid-und-Neu-Strafle 10-14
75131 Karlsruhe
Germany
tel: +49-721-9654-412
fax: +49-721-9654-413
e-mail: haug@fzi.de

Michael **Herz**
Universität Kaiserslautern
FB Informatik
LST Rechnerstrukturen
Postfach 3049
67653 Kaiserslautern
Germany
tel: +49-631-205-2644
fax: +49-631-205-2640
e-mail: herz@computer.org
http://www.xputers.informatik.uni-kl.de

Thomas **Hoffmann**
Universität Kaiserslautern
FB Informatik
LST Rechnerstrukturen
Postfach 3049
67653 Kaiserslautern
Germany
tel: +49-631-205-2625
fax: +49-631-205-2640
e-mail: hoffmant@rhrk.uni-kl.de
http://www.xputers.informatik.uni-kl.de

Michael **Kaufmann**
Universität Tübingen
Wilhelm-Schickard-Institut für Informatik
Sand 13
72076 Tübingen
Germany
tel: +49-7071-297404
fax: +49-7071-67540
e-mail: mk@informatik.uni-tuebingen.de

Rainer **Kress**
Siemens AG
Corporate Technology / Dept.
Microelectronics / ZT ME 5
Otto-Hahn-Ring 6
81739 München
Germany
tel: +49-89-636-50806
fax: +49-89-636-44950
e-mail: rainer.kress@mchp.siemens.de

Manfred **Kunde**
Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
FG Automaten und Formale Sprachen
Postfach 327
98684 Ilmenau
Germany
tel: +49-3677-692-766
fax: +49-3677-69 12 37
e-mail: kunde@theoinf.tu-ilmenau.de

Loic **Lagadec**
Université de Bretagne Occidentale
Dept. Informatique
6 avenue Victor Le Gorgeu
BP 809
F-29287 Brest
France
fax: +33-2-98-01-62-16

Dominique **Lavenier**
Universit» de Rennes
IRISA
Campus de Beaulieu
Avenue du Général Leclerc
F-35042 Rennes
France
tel: +33-2-99 84 72 17
fax: +33-2-99 84 71 71
e-mail: avenier@irisa.fr

Christian **Lengauer**
Universität Passau
FB Mathematik / Informatik
94030 Passau
Germany
tel: +49 851 509 3070
fax: +49 851 509 3092
e-mail: lengauer@fmi.uni-passau.de

William P. **Marnane**
University College Cork
Dept. of Electrical Eng. &
Microelectronics
Cork
Ireland
tel: +353-21-902041
fax: +353-21-271-698
e-mail: liam@rennes.ucc.ie

Martin **Middendorf**
Universität Karlsruhe
Institut AIFB
76128 Karlsruhe
Germany
tel: +49-721-608-37 05
fax: +49-721-69 37 17
e-mail: middendorf@aifb.uni-karlsruhe.de

Ronald **Moore**
Universität Frankfurt
FB 20 Informatik
Robert-Mayer-Str. 11-15
PF 11 19 32
60054 Frankfurt
Germany
tel: +49-69-798-22121
fax: +49-69-798-22351
e-mail: moore@ti.informatik.uni-frankfurt.de
http://www.ti.informatik.uni-frankfurt.de/TIhome.html

Ulrich **Nageldinger**
Universität Kaiserslautern
FB Informatik
LST Rechnerstrukturen / Bau 12/451
Postfach 3049
67653 Kaiserslautern
Germany
tel: +49-631-205-2657
fax: +49-631-205-2640
e-mail: nageldin@informatik.uni-kl.de
http://www.xputers.informatik.uni-kl.de

Stephan **Olariu**
Old Dominion University
Dept. of Computer Science
VA 23529-0162 Norfolk
USA
tel: +1-757-683-4417
fax: +1-757-683-4900
e-mail: olariu@cs.odu.edu

Laurent **Perraudeau**
Université de Rennes
IRISA
Campus de Beaulieu
Avenue du Général Leclerc
F-35042 Rennes
France
tel: +33-2-99 84 74 61
fax: +33-2-99 84 71 71
e-mail: Laurent.Perraudeau@irisa.fr

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

Andreas **Reinsch**
Friedrich-Schiller-Universität
LST für Rechnerarchitektur
R. 3223
Ernst-Abbe-Platz 1-4
07743 Jena
Germany
tel: +49-3641-946-381
fax: +49-3641-946-378
e-mail: reinsch@uni-jena.de

Sergej **Sawitzki**
TU Dresden
Fakultät Informatik
Dürerstrafle 24
01062 Dresden
Germany
tel: +49-351-4638428
e-mail: ss9@irz.inf.tu-dresden.de
http://www.inf.tu-dresden.de/~ss9

Manfred **Schimmler**
TU Braunschweig
Institut für Datenverarbeitungsanlagen
Hans-Sommer-Str. 66
38106 Braunschweig
Germany
tel: +49-531-391-3735
fax: +49-531-391-4587
e-mail: Schimmler@ida.ing.tu-bs.de

Hartmut **Schmeck**
Universität Karlsruhe
Institut AIFB
76128 Karlsruhe
Germany
tel: +49-721-608-42 42
fax: +49-721-69 37 17
e-mail: schmeck@aifb.uni-karlsruhe.de

Heiko **Schröder**
Loughborough University of Technology
Dept. of Computer Studies
Haslesgrave Bldg. - Room N 0.28
LE 11 3TU Loughborough
Great Britain
tel: +44-1-509-228225
fax: +44-1-509-211586
e-mail: H.Schroder@lboro.ac.uk

John **Snowdon**
Heriot-Watt University
Dept. of Physics
EH14 4AS Edinburgh
Great Britain
tel: +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
Dürerstrafle 24
01062 Dresden
Germany
tel: +49-351-463-8243
fax: +49-351-463-8324
e-mail: rgs@ite.inf.tu-dresden.de

Carsten **Steckel**
Universität Karlsruhe
Institut AIFB
76128 Karlsruhe
Germany
tel: +49-721-966 3806
fax: +49-721-966 3807
e-mail: carsten.steckel@stud.uni-karlsruhe.de

Ondrej **Sykora**
Slovak Academy of Sciences
Dept. of Informatics
Computing Center
Dubravask cesta 9
84235 Bratislava
Slovakia
tel: +421-7-3783212
fax: +421-7-375859
e-mail: sykora@savba.sk
http://www.ifi.savba.sk

Uwe **Tangen**
Institut für Molekulare Biotechnologie
Beutenbergstr. 11
07745 Jena
Germany
tel: +49-3641-656143
fax: +49-3641-656191
e-mail: utangen@imb-jena.de
http://www.imb-jena.de

Jürgen **Teich**
ETH Zürich
Inst. für Techn. Informatik
Gloriastr.35
CH-8092 Zürich
CH
tel: +41 1 632 7037
fax: +41 1 632 1035
e-mail: teich@tik.ee.ethz.ch
http://www.tik.ee.ethz.ch

Ralph **Weper**
Universität Jena
Institut für Informatik
Ernst-Abbe-Platz 1-4
07743 Jena
Germany
tel: +49 3641 9 46376
fax: +49 3641 9 46372
e-mail: weper@informatik.uni-jena.de
http://www2.informatik.uni-jena.de/~weper

Thomas **Worsch**
Universität Karlsruhe
Fakultät für Informatik
Am Fasanengarten 5
Postfach 6980
76128 Karlsruhe
Germany
tel: +49-721-608-43 11
fax: +49-721-69 86 75
e-mail: worsch@ira.uka.de

Eberhard **Zehendner**
Universität Jena
Institut für Informatik
Ernst-Abbe-Platz 1-4
07743 Jena
Germany
tel: +49 3641 946385
fax: +49 3641 946372
e-mail: eberhard.zehendner@uni-jena.de nez@uni-jena.de
http://www2.informatik.uni-jena.de/~nez

Karl-Heinz **Zimmermann**
TU Hamburg-Harburg
Technische Informatik VI
- Verteilte Systeme
Martin-Leuschel-Ring 16
21071 Hamburg
Germany
tel: +49-40-7718-3155
fax: +49-40-7718-2798
e-mail: k.zimmermann@tu-harburg.de
http://www.tu-harburg.de/ti6/~ti6khz