

10281 Abstracts Collection

Dynamically Reconfigurable Architectures

— Dagstuhl Seminar —

Peter M. Athanas¹, Jürgen Becker², Jürgen Teich³ and Ingrid Verbauwhede⁴

¹ Virginia Polytechnic Institute, US
athanas@vt.edu

² KIT Karlsruhe, DE
becker@kit.edu

³ Universität Erlangen-Nürnberg, DE
teich@informatik.uni-erlangen.de

⁴ Katholieke Universiteit Leuven, BE
Ingrid.verbauwhede@esat.kuleuven.be

Abstract. From 11.07.10 to 16.07.10, the Dagstuhl Seminar 10281 “Dynamically Reconfigurable Architectures” was held in Schloss Dagstuhl – Leibniz Center for Informatics. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Dynamically Run-Time Reconfigurable Computing Architectures, Self-adaptive Systems, Computational Models, Circuit Technologies, System Architecture, CAD Tool Support, Reconfigurable/Adaptive Computing based on Nanotechnologies

10281 Summary – Dynamically Reconfigurable Architectures

Dynamic and partial reconfiguration of hardware architectures such as FPGAs and coarse grain processing arrays bring an additional level of flexibility in the design of electronic systems by exploiting the possibility of configuring functions on-demand during run-time. When compared to emerging software-programmable Multi-Processor System-on-a-Chip (MPSoC) solutions, they benefit a lot from lower cost, more dedication and fit to a certain problem class as well as power and area efficiency. This has led to many new ways of approaching existing research topics in the area of hardware design and optimization techniques. For example, the possibility of performing adaptation during run-time raises questions in the areas of dynamic control, real-time response, on-line power management and design complexity, since the reconfigurability increases the design space towards infinity.

Keywords: Dynamically Run-Time Reconfigurable Computing Architectures, Self-adaptive Systems, Computational Models, Circuit Technologies, System Architecture, CAD Tool Support, Reconfigurable/Adaptive Computing based on Nanotechnologies

Joint work of: Athanas, Peter M.; Becker, Jürgen; Teich, Jürgen; Verbauwhede, Ingrid

Extended Abstract: <http://drops.dagstuhl.de/opus/volltexte/2010/2892>

Brainstorming session: Dynamically Reconfigurable Architectures and Security

Starting questions:

- do we need dynamically reconfigurable architectures for security?
- do we need it for performance reasons, i.e. better execution time, less power or energy, smaller area?
- or do we need it to improve secure implementations? meaning to protect implementations from attacks?
- does dynamically reconfigurable architectures hurt or help security?

The three main open research problems as conclusion of the panel:

1. IP protection, IP distribution, evaluation etc. in a trusted and secure way. The more reconfigurable, remote reconfigurable, dynamically reconfigurable an architecture becomes, the more urgent this problem becomes.
2. Secure remote update. This holds for FPGA reconfigurations as well as embedded SW (both are "soft")
3. API's and interfaces between HW & SW. HW becomes softer, SW intermixed with HW.

Keywords: Security, dynamically reconfigurable architectures, FPGA, IP protection

Design and Implementation of an Object-Oriented DPR-Framework

Norbert Abel (Universität Heidelberg, DE)

Nowadays, two innovative future trends regarding hardware development and hardware description can be found. The first trend concerns the hardware itself. Modern Xilinx FPGAs provide the possibility to be reconfigured partially and dynamically - which is called dynamical partial reconfiguration (DPR). DPR opens a huge field of new functionalities on FPGAs.

However, using DPR means struggling with architectural details of the used FPGAs and the according synthesis and implementation tools. A developer would focus most of the time on DPR and only a small part of the time on the implementation of the actual modules - of course that is the opposite of what hardware engineers want to do.

The second trend concerns the way hardware is described. Many hardware developing groups are looking forward to an HDL which operates on the algorithmic level, since this would come with a significant increase in productivity. The aim is to be able to translate common software algorithms to hardware in an efficient way (which is called high-level synthesis or HLS).

Although both DPR and HLS are important future trends regarding hardware design, they develop quite independently. Today's software-to-hardware compilers focus on conventional hardware and therefore have to remove dynamic aspects such as the instantiation of calculating modules at runtime. Even object-oriented languages like SystemC do not support the dynamic instantiation of objects (that means the usage of new or delete outside of the constructor) for synthesis at all. On the other hand, DPR tools are working on the lowest possible layer regarding FPGAs: the bitfile level. Our research focuses on the design and the implementation of a Framework combining the two technologies, since this has the potential to kill two birds with one stone. Firstly, DPR can change the programming paradigm in future HDLs regarding dynamic instantiations. Dynamic parts would not have to be removed any longer but could be realized on the target FPGA using DPR. Secondly, a high-level language support of DPR technologies could help end its shadowy existence and turn it into a commonly used method.

Keywords: FPGA, DPR, HLS, Object-Orientation

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2836>

See also: Norbert Abel, Design and Implementation of an Object-Oriented Framework for Dynamic Partial Reconfiguration, FPL 2010, Milano, Italy, August 2010

Run-time Adaptation for Reconfigurable Embedded Processors

Lars Bauer (KIT - Karlsruhe Institute of Technology, DE)

State-of-the-art reconfigurable processors require that the application programmer (or compiler) determines during compile time which reconfigurations shall be performed and when they shall be performed, i.e. which accelerators shall be loaded to a particular part of the reconfigurable fabric at a certain time. The problem is that it is typically not known during compile time which applications execute at the same time (i.e. in a multi-tasking environment), demanding the reconfigurable fabric. Additionally, it is not necessarily known, which accelerators are demanded frequently and which are demanded rather seldom, because

this may depend on the input data for the application. Here, it would be desirable that the information about the actual application/system requirements that are only available during run time would be used to determine which re-configurations should be performed.

In this talk, concepts and strategies are presented to increase the run-time adaptivity of reconfigurable processors significantly. As foundation, a novel hierarchical composition of Special Instructions is presented that allows switching between different performance/area trade-offs efficiently during run time. To determine which trade-off shall be chosen (and thus, to provide adaptivity), light-weight approaches for online monitoring, dynamic instruction-set selection, reconfiguration-sequence scheduling, and accelerator replacement are discussed and a comparison with state-of-the-art reconfigurable processors is provided.

Joint work of: Bauer, Lars; Shafique, Muhammad; Henkel, Jörg

rASIP: Reconfigurable Application-Specific Instruction-Set Processor

Mladen Berekovic (TU Braunschweig, DE)

DSP system designers have been deploying application-specific instruction set processors or ASIPs for some time to overcome performance bottlenecks and improve power and efficiency. However, this approach has the disadvantage of being relatively inflexible when it comes to the support of new application domains. Also, in typical system-on-chip designs, several different ASIPs are deployed, each of them targeting and optimized for a different function block.

This makes design of such systems very time-consuming and application-specific.

A way to overcome these limitations is to deploy a reconfigurable array instead of a "special function unit" hence allowing for a larger flexibility in mapping algorithms. This also allows for the use of a standardized ASIP template that can be used to implement several different function blocks such as audio and video. This approach borrows a lot of concepts from coarse-grain reconfigurable computing and is termed as rASIP or reconfigurable ASIP. The concept is being demonstrated within the context of a European Artemis project Smart for use in low-power video sensor nodes with support for streaming video and encryption.

Keywords: ASIP, reconfigurable, coarse-grain-array CGRA

Low-Power Reconfigurable Architectures for High-Performance Mobile Nodes

Mladen Berekovic (TU Braunschweig, DE)

Modern embedded systems have an emerging demand on high performance and low power circuits.

Traditionally special functional units for each application are developed separately. These are plugged to a general purpose processors to extend its instruction set making it an application specific instruction set processor. As this strategy reaches its boundaries in area and complexity reconfigurable architectures propose to be more flexible. Thus combining both approaches to a reconfigurable application specific processor is going to be the upcoming solution for future embedded systems.

Keywords: reconfiguration, ASIP, RASIP, low power, high performance, video encoding, encryption, wireless sensor node, mobile device

Joint work of: Hanke, Matthias; Kranich, Tim; Berekovic, Mladen; Papaefstathiou, Yannis

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2837>

Platform Based Reconfigurable Computing Design

Neil W. Bergmann (The University of Queensland, AU)

Reconfigurable computing design is hard. Experienced embedded systems engineers can still take many weeks or months to get a reconfigurable system-on-chip working, even without the complication of dynamic reconfiguration. DR makes design exponentially harder. The required knowledge base is broad and deep, and very vendor-specific.

The search for a "one-size-fits-all" design methodology is most likely flawed.

This talk proposes a design methodology based on platforms. A platform is a design framework which already has all the infrastructure for a reconfigurable system on chip - all that needs to be added is the application specific content - specific hardware and software module designs. A platform is necessarily domain specific - a platform for high-speed image processing is not the same as one for low-power wireless sensors.

Some thoughts on possible frameworks in terms of design languages, design abstractions, operating systems, network on chip, I/O, and dynamic reconfigurability will be presented.

Keywords: Design tools, FPGA, reconfigurable computing

API to assist the assembly of 2D reconfigurable systems

Lars Braun (KIT - Karlsruhe Institute of Technology, DE)

Partial dynamic reconfigurable (PDR) systems designed with state-of-the-art tool chains, like the Early Access Partial Reconfiguration (EAPR) Flow from Xilinx, do not exploit the flexibility provided by dynamic and partial reconfiguration features as supported by state-of-the-art FPGA devices.

For example the utilized chip area and the location for a dynamic area on the chip is traditionally fixed during design-time. Thereby the shape and the size of the area is fixed by the largest module. If a smaller module is placed on the region of a bigger one, chip area remains unutilized. These restrictions are only some examples for the current status for the support of development and run-time tools for reconfigurable hardware architectures. A new approach will be presented for exploiting the full capability of reconfigurable hardware architectures more efficiently than traditional solutions. This is achieved by a new concept of using micro blocks for the communication infrastructure as well as for the functional elements on the FPGA. In addition, a Mesh-based Network on Chip (NoC) which is specifically designed for the constraints given by the FPGA completes this approach. This paper will present the current status of this approach and provides some ideas about the possible tool chain to support designers in creating such a PDR system.

Dynamically adaptive behaviours

Gordon Brebner (Xilinx - San José, US)

There has been much research on dynamically reconfigurable architectures over the past two decades, with promising results. However, there has also been a persistent decoupling of this architecture work from real applications. This talk will consider the question of how dynamically adaptive behaviour can arise naturally in current or future applications, and how this might be coupled to underlying architectures via apt design methodologies and tools. It will be illustrated by an example from very high speed optical networking.

Why FPGAs should have binary compatibility

Luigi Carro (UFRGS - Porto Alegre, BR)

We discuss the role of FPGAs in the near future and some challenges for them to reach the general purpose arena.

Keywords: Software compatibility, reconfigurable general purpose processing

Reconfigurability for Variability

Peter Y. K. Cheung (Imperial College London, GB)

Our recent work on how to use reconfigurability to alleviate the problems caused by process variation and degradation is presented. Techniques for efficient online delay measurement that allow us to characterize the delay of any path on an FPGA with pico-second resolution is described.

Such delay information is used to help reconfiguring a device which would otherwise be too slow for the task in hand - an approach known as "late binding". It will further show our latest work on both measuring and overcoming the problem of process degradation (or time-dependent variability). The talk will conclude by suggesting a number of interesting unsolved issues relating to this topic.

Keywords: Reconfigurable architecture, process variation, process degradation, online delay test, late binding, timing error, BIST

Towards a reconfigurable hardware architecture for implementing a LDPC module suitable for software radio systems

Rene Cumplido (INAOE - Puebla, MX)

Error correction is a key piece in modern digital communications. This area is directed towards the recovery of multiple errors generated when a signal is transmitted using noisy channels. In the last years, LDPC (Low Density Parity Check) codes have attracted attention of researchers because of their excellent error correcting performance using message-passing algorithms. In addition to a required performance, actual radios require to communicate with multiple radios for supporting the increased needed of integration between devices. In this sense, Software Defined Radio (SDR), an enabling technology in many areas of communications, allows to build multi standard radios for communicating with other radios using any communication standard. Reconfigurable implementations of LDPC codes are an indispensable requirement for enabling future radios. In this paper, some open problems in designing and implementing such LDPC components are presented and discussed.

Keywords: LDPC codes, Software Defined Radio, Hardware Implementation

Joint work of: Cumplido, Rene; Campos, Juan Manuel; Feregrino, Claudia; Perez-Andrade, Jose Roberto

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2895>

Dilating network-on-chip based dynamic hardware configurations

Oliver Diessel (Univ. of New South Wales, AU)

This research aims to discover more productive approaches to utilizing field-programmable gate arrays for implementing performance critical systems, such as in video surveillance, mobile communications, and electronic warfare, where the requirements change rapidly over time.

The motivation for and preliminary results on the dilated placement of communication task graphs into a regular network topology will be described. Further work will be discussed.

Keywords: Placement, Dynamic reconfiguration, Run-time support

Joint work of: Diessel, Oliver; Hredzak, Branislav

Towards Dilated Placement of Dynamic NoC Cores

Oliver Diessel (Univ. of New South Wales, AU)

Instead of mapping application task graphs in a compact manner onto reconfigurable devices using a network-on-chip for interconnecting application cores, we propose dilating the mappings as much as the available latencies on critical connections allow. In a dilated mapping, the unused resources between an application's configured components can be used to provide additional flexibility when the configuration needs to change. We motivate the reasons for dilating application task graphs targeted at reconfigurable devices; derive a simulated annealing approach to dilating the placement of such graphs; and present preliminary results of applying the algorithm to synthetic test cases. The method appears to result in successful and meaningful graph dilation and could be further tuned to satisfy desired power constraints.

Keywords: Modular reconfiguration, networks-on-chip, application mapping, dilation

Joint work of: Hredzak, Branislav; Diessel, Oliver

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2834>

Evaluation and comparison of FPGA designs and other thoughts on FPGA design security

Saar Drimer (University of Cambridge, GB)

The FPGA research community can improve the level of contributions, enable meaningful comparisons, and promote reuse by creating incentives for implementers to share their source code and datasets at the time of publication. In order to do so, we should add a reproducibility scale and score to manuscripts' review process so that the effort for creating reproducible research is appropriately rewarded – this score could, in some cases, compensate for low originality scores. Further, a committee of academic and industry members should be formed to guide implementers and reviewers of common pitfalls and how to report results in a way that helps meaningful evaluations. This guide will be continuously updated. Specifically in the FPGA security field, interesting topics to explore are

practical secure remote update of FPGA bitstreams, and role/identity-based authentication in order to allow different levels of access to portions of the FPGA fabric – this may well open up a new kind of FPGA applications. Finally, it is likely that we would soon witness increased use of FPGAs as an adversarial tool, as more man-in-the-middle type of attacks will require timing-critical operations which are hard to achieve with cheap microcontrollers.

Keywords: FPGA, security

Design and Programming of Reconfigurable Multiprocessor Systems

Diana Goehring (Fraunhofer IOSB, DE)

The efficient programming of reconfigurable architectures is until today a great challenge especially for application developers, which are not familiar with hardware description languages such as VHDL or Verilog. There exists several commercial and academic C-based High Level Synthesis tools like ImpulseC or CatapultC as well as some MATLAB-based tools like Matlab HDL-Coder or Xilinx System Generator. Their disadvantage is that these tools only support a subset of ANSI-C, C++ or Matlab and that they only can be used to generate accelerator functionalities. This means the framework around this accelerator, e.g. connections with external devices like cameras, memories, PCI interfaces, still has to be designed by the user. Therefore, application engineers often prefer to use multiprocessor systems or general purpose graphic processor units like NVidia Tesla GPUs, which support C/C++-based programming models like CUDA, OpenMP or MPI resulting in a shorter design cycle for a sufficient speedup. The disadvantage of these architectures is their high power consumption which makes them inefficient for embedded systems. This results in a trend towards FPGA-based multiprocessor systems, which combine the benefits of both a simpler programming model together with the lower power consumption of the FPGAs due to the design time optimization of the MPSoC. By further exploiting the runtime adaptation of FPGAs like the Xilinx FPGAs, which support dynamic and partial reconfiguration, the RAMPSoC (Runtime Adaptive Multi-Processor System-on-Chip) approach was born. RAMPSoC allows adapting the hardware architecture of the MPSoC at design and at runtime to the application requirements. Therefore an efficient performance per Watt ratio can be achieved. To program such a flexible MPSoC and to hide the complexity of the underlying hardware a novel design methodology was developed, which currently supports applications written in C, C++ or C combined with the MPI standard, which is well known in the supercomputing community. The design methodology generates both the hardware architecture and the partitioning of the software applications at design time. A special purpose operating system called CAP-OS (Configuration Access Port Operating System) receives the partial configuration bitstreams and software executables and is responsible for the

scheduling of the tasks, the resource management and the configuration of the system at runtime. This talk will present the RAMPSoC approach, the current status of the design methodology and the CAP-OS.

Keywords: Reconfigurable Computing, Multiprocessor Systems, Design Methodology, Programming model

Why reconfigurable architectures are useful

Reiner Hartenstein (TU Kaiserslautern, DE)

Computing has to be re-invented because of two main problem areas, both being related to power efficiency. Technical limits of power dissipation per processor chip caused the transition to multicore architectures. Financial limits will be reached within about a decade or slightly more by rising energy prices and rapid growth of the electricity consumption of the entirety of all kinds of computers everywhere worldwide. If we do not find a timely effective solution we will run into a severe economic crisis.

A key issue is the tremendous inefficiency of what we call "software", i. e. running on instruction-stream-driven architectures. Improvements by orders of magnitude can be obtained by migration to data streams in the context of massive software to configure migrations. Data-stream-driven reconfigurable architectures are useful by providing the basis to reinvent computing for avoiding the future unaffordability of its electricity bill. The talk discusses how to implement a rescue campaign.

Keywords: Power-efficiency, overhead, paradigm shift

Architectural Vulnerability Factor Estimation with Backwards Analysis

Robert Hartl (TU München, DE)

SEUs (Single Event Upsets) in memories of synchronous circuits will be a challenging problem for reliable safety-critical systems. There are several approaches for estimation of fault-tolerance and sensitivity of these circuits, which require high computational effort and special circuit models or deliver too pessimistic results. We present a novel and deterministic method to determine the AVF (Architectural Vulnerability Factor) of any RT-level circuit using a standard simulation model. The method called Backwards Analysis (BA) uses stimulus values in time-reversed order to calculate the impact of several masking effects (logic masking, information lifetime, timing derating, transitive masking) in one single algorithm. BA delivers exact results in several levels of detail in acceptable runtime and shows the sensitive parts of a design. These results could be used for reliability assessment and help limiting the hardware efforts for selective hardening of the circuit.

Keywords: Architectural Vulnerability Factor, AVF, Single Event Upset, SEU, Backwards Analysis, derating, logic masking

Joint work of: Hartl, Robert; Rohatschek, Andreas; Stechele, Walter; Herkersdorf, Andreas

Architectural Vulnerability Factor Estimation with Backwards Analysis

Robert Hartl (TU München, DE)

Single-Event-Upsets in synchronous register-based designs are a severe problem for safety-critical applications. Exact and detailed error rate estimations are needed to determine a system's level of reliability. Available methods for estimation consider only special effects, use special reliability models or are computationally intensive. We present an innovative method that is able to calculate the architectural vulnerability factor (AVF) of any RT-level circuit description by applying time-reversed stimulus values. This method, which we call Backwards Analysis, considers all major masking effects (logic masking, information lifetime, timing derating, transitive masking) in a single algorithm and delivers results in several levels of detail from average AVF through sensitivity waveforms. The results show the critical parts and states of a design, which could be used for reliability assessment and selective hardening of the circuit to reach a target failure rate.

Keywords: Architectural Vulnerability Factor, Backwards Analysis, Single Event Upset, Critical Path Tracing

Joint work of: Hartl, Robert; Rohatschek, Andreas; Stechele, Walter; Herkersdorf, Andreas

See also: Hartl, R., Rohatschek, A., Stechele, W., Herkersdorf, A.: Architectural Vulnerability Factor Estimation with Backwards Analysis; Proc. 13th EUROMICRO Conference on Digital System Design (DSD10), 2010

Research challenges and opportunities for FPGA interconnection networks

Yajun Ha (National Univ. of Singapore, SG)

With the increasing process variations in advanced technologies, delay defects are gaining a larger impact on FPGA timing yield. If the delay defect areas can be quickly and accurately located, FPGA timing yield can be improved by avoiding them. Conventional delay testing methods do not take into account the spatial information of variability-induced delay faults, thus cannot accurately locate the delay defects to a well restricted areas. Based on the superb locality preserving

feature of space-filling curves, we propose a method to locate delay faults and generate a delay variation map (DVM) with scalable resolutions. The method uses Hilbert curves to guide the test configurations of FPGAs. It is able to work on FPGAs with arbitrary dimensions and embedded hard IP cores. Compared with normal test approaches, our method achieved around 60% increase in delay faults locating resolution.

The Optimization of Interconnection Networks in FPGAs

Yajun Ha (National Univ. of Singapore, SG)

Scaling technology enables even higher degree of integration for FPGAs, but also brings new challenges that need to be addressed from both the architecture and the design tools side. Optimization of FPGA interconnection network is essential, given that interconnects dominate logic. Two approaches are presented, with one based on the time-multiplexing of wires and the other using hierarchical interconnects of high-speed serial links and switches. Design tools for both approaches are discussed. Preliminary experiments and prototypes are presented, and show positive results.

Keywords: field-programmable gate array, architecture, computer-aided design

Joint work of: Chen, Xiaolei; Ha, Yajun

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2842>

A Tool for the Support of Experiments with Online Adaptivity

Christian Hochberger (TU Dresden, DE)

In this talk I will briefly introduce the concept of AMIDAR processors. AMIDAR stands for adaptive microinstruction driven architecture. It forms a good basis for processors that can be reconfigured on the fly at runtime. I will then show recent results of our research (what speedup can we achieve, how many resources are required for this).

The main part of my talk will be focused on the simulator that we use to carry out our experiments. I will show its structure and I will give a short live presentation, where I let different code fragments run with varying parameters.

Keywords: Online Adaptivity, Coarse Grain Reconfigurable Array, Reconfigurable Processor, AMIDAR

Application Requirement Aware Processors

Michael Huebner (KIT - Karlsruhe Institute of Technology, DE)

Dynamic and partial reconfiguration of FPGAs is a well known technique in runtime adaptive system design. With this technique, parts of a configuration can be substituted while other parts stay operative without any disturbance. The advantage is the fact, that the spatial and temporal partitioning can be exploited with the goal to increase performance and to reduce power consumption due to the re-use of chip area. A novel methodology for the inclusion of the configuration access port into the data path of a processor core, in order to adapt the internal architecture and to re-use this access port as data- sink and source, leads to a high degree of flexibility in novel processors. The inclusion of the configuration access port into the data path of the processor enables the further abstraction of the complexity of dynamic FPGA hardware. The access to the hardware is realized from a software perspective with standardized libraries and therefore definitely increases the attractiveness to a wider community including also non experts in reconfigurable hardware. These processors are able to provide the optimal microarchitecture as well and the suitable instruction set architecture for a given application. Furthermore, due to run-time adaptivity, the processors architecture can be tailored to the current requirements of the application or system status. The situation tailored architecture can be realized e.g. through an adaptive pipeline balancing, the use of ipc (instruction per cycles) variation to reduce power consumption, the exploitation of dynamic instruction level parallelism and related pipeline adaptation, and the adaptive issue queue for reduced power at high performance. The examples are only a small overview for a novel generation of adaptive processor architectures which is application specific but reconfigurable.

Keywords: Reconfigurable ASIP, FPGA, dynamic and partial reconfiguration

Compiling Geometric Algebra Computations into Reconfigurable Hardware Accelerators

Andreas Koch (TU Darmstadt, DE)

Geometric Algebra (GA), a generalization of quaternions, is a very powerful form for intuitively expressing and manipulating complex geometric relationships common to engineering problems. The actual evaluation of GA expressions, though, is extremely compute intensive due to the high-dimensionality of data being processed. On standard desktop CPUs, GA evaluations take considerably longer than conventional mathematical formulations. GPUs do offer sufficient throughput to make the use of concise GA formulations practical, but require power far exceeding the budgets for most embedded applications. While the suitability of low-power reconfigurable accelerators for evaluating specific

GA computations has already been demonstrated, these often required a significant manual design effort. We present key components of a proof-of-concept compile flow combining symbolic and hardware optimization techniques to automatically generate accelerators from the abstract GA descriptions without user intervention that are suitable for high-performance embedded computing. The presentation will address the hardware-independent front- and middle-ends, the hardware synthesis back-end, and the underlying library of parametrizable module generators.

Compiling Geometric Algebra Computations into Reconfigurable Hardware Accelerators

Andreas Koch (TU Darmstadt, DE)

Geometric Algebra (GA), a generalization of quaternions and complex numbers, is a very powerful framework for intuitively expressing and manipulating the complex geometric relationships common to engineering problems.

However, actual processing of GA expressions is very compute intensive, and acceleration is generally required for practical use. GPUs and FPGAs offer such acceleration, while requiring only low-power per operation.

In this paper, we present key components of a proof-of-concept compile flow combining symbolic and hardware optimization techniques to automatically generate hardware accelerators from the abstract GA descriptions that are suitable for high-performance embedded computing.

Keywords: Geometric Algebra, FPGA, High-Level-Compiler Gaalop

Joint work of: Huthmann, Jens; Müller, Peter; Stock, Florian; Hildenbrand, Dietmar; Koch, Andreas

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2838>

Advances in Component-based System Design and Partial Run-time Reconfiguration

Dirk Koch (University of Oslo, NO)

With passing over the 1M LUT barrier, FPGA technology is heading into new challenges and opportunities. While the present ASIC-like design methodology and tools will struggle to scale with such huge devices, providing partial run-time reconfiguration will become obligatory for dealing with long configuration times and the increasing vulnerability to single event upsets.

Within the COSRECOS project, we address these issues by developing methods and tools that allow to compose systems rapidly by plugging together fully physically implemented components. Moreover, by allowing a hot-swapping of

such components, the tremendous advantages of partial run-time reconfiguration can be utilized.

This talk will give an overview of recent trends, our present research activities, and will discuss open issues.

Keywords: FPGA design, partial reconfiguration, component-based design

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2841>

Advances and Trends in Dynamic Partial Run-time Reconfiguration

Dirk Koch (University of Oslo, NO)

The progress in silicon industry has resulted in a tremendous increase in device capacity of FPGAs. The smallest devices of the upcoming Altera Stratix-5 FPGAs as well as the announced Xilinx Virtex-7 FPGAs provide more than double the amount of logic and embedded memory as the flagship devices of the one decade old Stratix or Virtex-II series FPGAs. By passing the one million LUTs border, high density FPGAs are sufficient to host 250 softcore CPUs plus the required peripherals.

Keywords: FPGA, Partial Run-time Reconfiguration

Joint work of: Koch, Dirk; Torresen, Jim

A hypermorphic architecture template offering hardware efficient exploitation of ILP/DLP/TLP

Ralf Koenig (KIT - Karlsruhe Institute of Technology, DE)

The dynamic run-time complexity of today's and future embedded applications is steadily increasing. In order to address this challenge from a hardware perspective, various reconfigurable Multiprocessor System-on-Chip (MPSoC) architectures have been developed. Thereby, most of today's reconfigurable architectures show intrinsic characteristics that offer a unique strength when processing algorithms of a certain application domain but are less efficient otherwise. Consequently, all those cores are loosely connected - by a bus or NoC - at architecture design time for integration on an MPSoC that match the requirements of a certain application domain best. In that way MPSoCs perform very efficient with respect to a distinct application domain. However, they become inefficient for executing a wide range of applications as the MPSoCs are unable to dynamically balance a brought range of instruction-, data-, and thread-level parallelism (ILP, DLP, TLP). We are approaching this challenge by a novel, hypermorphic architecture concept. The architecture template is composed out of tightly integrated reconfigurable fabrics offering two-fold characteristics at run-time. They

either can dynamically be combined to realize hardware accelerators for computational intensive kernels on word and sub-word granularity. In absence of loop- or data-level parallelism, they otherwise offer the creation of VLIW processors of variable issue-width. This concept offers to efficiently exploit the available hardware for different compositions of ILP, DLP, and TLP. Also with respect to the programming paradigm, this concept offers interesting aspects. While today's heterogeneous MPSoCs most often lack fundamental programming support such as an integrated toolchain or even compilers for the reconfigurable fabrics, this concept allows for a much more developer friendly implementation process. At first the available C/C++ code can steadily be optimized to run on VLIW processor instances, also exploiting TLP, while only the most demanding kernels will have to be realized by hardware accelerators.

System Adaptivity in a FlexPath Network Processor

Michael Meitinger (TU München, DE)

Based on observations on current NP implementations and relevant Internet traffic scenarios, a new NP architecture is defined that makes use of reconfigurable packet processing paths in order to improve the system performance: the FlexPath Network Processor (FlexPath NP). We propose to extend state-of-the-art processor-centric NP architectures with specific hardware units in order to classify the incoming traffic into separate processing classes. For each traffic class, we can provide an optimized processing path, i.e. a functional unit traversal sequence within the NP. In addition, we propose to offload significant shares of the traffic to a dedicated hardware path in order to bypass the CPU cluster and save precious programmable processing resources. We also address the problem of multi-processor load balancing in the context of multi-core network processors. In this talk, we present different levels of system adaptivity within the FlexPath NP.

Secure remote reconfiguration of FPGAs

Nele Mentens (K.U. Leuven, BE)

This paper presents a solution for secure remote reconfiguration of FPGAs. Communicating the bitstream has to be done in a secure manner to prevent an attacker from reading or altering the bitstream. We propose a setup in which the FPGA is the single device in the system's zone-of-trust. The result is an FPGA architecture that is divided into a static and a dynamic region. The static region holds the communication, security and reconfiguration facilities, while the dynamic region contains the targeted application.

Keywords: FPGA, cryptography, security, remote configuration

Joint work of: Mentens, Nele; Vliegen, Jo; Braeken, An; Touhafi, Abdellah; Wouters, Karel; Verbauwhede, Ingrid

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2839>

Managing Runtime Reconfiguration Decisions

Thilo Pionteck (Universität Lübeck, DE)

Partially reconfigurable hardware accelerators enable the offloading of computationally intensive tasks from software to hardware at runtime. Beside handling the technical aspects, finding a proper reconfiguration point in time is of great importance for the overall system performance. Determination of a suitable point reconfiguration demands the evaluation of the performance degradation during runtime reconfiguration and expected performance benefit after reconfiguration. Three different approaches to determine a proper point of reconfiguration are discussed. Delays and weighted transitions are used to reduce the number of reconfigurations while keeping system performance at a maximum. Evaluation is done with a simulation model of a runtime reconfigurable network coprocessor. Results show that the number of reconfigurations can be reduced by about 35% for a given application scenario. By optimizing runtime reconfiguration decisions, the overall system performance is even higher than compared to pure threshold based reconfiguration decision schemes.

Keywords: Runtime reconfiguration decisions, reconfiguration management

How to enjoy the variability of your FPGA

Patrick Schaumont (Virginia Polytechnic Institute - Blacksburg, US)

A PUF can be used to extract a non-volatile secret key from an FPGA fabric by exploiting process manufacturing variations. In this talk, we present the requirements for the implementation of Physical Unclonable Functions in FPGA.

They include security requirements, such as the nature and the amount of challenge/response pairs in the PUF. They also include quality metrics such as Uniqueness and Reliability. The focus of the talk is on three observations related to the implementation of PUF in FPGA. First, we note that a PUF should be analyzed in terms of the population of chips, not in terms of a single design.

Second, we note that most existing PUF architectures do not map well into the FPGA fabric. Third, we point out that the use of PUF as a root-of-trust in non-volatile FPGA is very tricky, and requires an access-protected bitstream.

Keywords: Hardware Security, Physical Unclonable Functions, Authentication

A mathematical approach towards hardware design

Gerard J. M. Smit (University of Twente, NL)

Today the hardware for embedded systems is often specified in VHDL. However, VHDL describes the system at a rather low level, which is cumbersome and may lead to design faults in large real life applications. There is a need of higher level abstraction mechanisms.

In the embedded systems group of the University of Twente we are working on systematic and transformational methods to design hardware architectures, both multi core and single core. The main line in this approach is to start with a straightforward (often mathematical) specification of the problem. The next step is to find some adequate transformations on this specification, in particular to find specific optimizations, to be able to distribute the application over different cores. The result of these transformations is then translated into the functional programming language Haskell since Haskell is close to mathematics and such a translation often is straightforward. Besides, the Haskell code is executable, so one immediately has a simulation of the intended system.

Next, the resulting Haskell specification is given to a compiler, called CëaSH (for CAES LAnguage for Synchronous Hardware) which translates the specification into VHDL. The resulting VHDL is synthesizable, so from there on standard VHDL-tooling can be used for synthesis. In this work we primarily focus on streaming applications: i.e. applications that can be modeled as data-flow graphs.

At the moment the CëaSH system is ready in prototype form and in the presentation we will give several examples of how it can be used. In these examples it will be shown that the specification code is clear and concise. Furthermore, it is possible to use powerful abstraction mechanisms, such as polymorphism, higher order functions, pattern matching, lambda abstraction, partial application. These features allow a designer to describe circuits in a more natural and concise way than possible with the language elements found in the traditional hardware description languages.

In addition we will give some examples of transformations that are possible in a mathematical specification, and which do not suffer from the problems encountered in, e.g., automatic parallelization of nested for-loops in C-programs.

Keywords: Hardware design, mathematical specification, streaming applications

Joint work of: Smit, Gerard J. M.; Kuper, Jan; Baaij, Christiaan P. R.

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2840>

Lessons Learned from last 4 Years of Reconfigurable Computing

Walter Stechele (TU München, DE)

Partial dynamic reconfiguration of FPGAs was investigated for video-based driver assistance applications during the last 4 years. High-level application software was combined with dynamically reconfigurable hardware accelerators in selected scenarios, e.g. vehicle lights detection, optical flow motion detection. From the beginning of the project, various research challenges have been targeted, including hardware/software partitioning between embedded RISC and accelerators, granularity of reconfigurable regions, as well as the impact of the reconfiguration process on system performance. This article will review the status of these research challenges and present an outlook on future challenges, including reconfiguration look ahead. Challenges will be illustrated on robotic vision scenarios with dynamically changing computational load from soft real-time and hard real-time applications.

Keywords: Reconfigurable computing, vision-based driver assistance

Joint work of: Stechele, Walter; Claus, Christopher; Laika, Andreas

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2835>

An ADL-based Software Framework targeting a hypermorphic reconfigurable processor architecture

Timo Stripf (KIT - Karlsruhe Institute of Technology, DE)

Through reconfiguration of the microarchitecture, hypermorphic processors offer a wide range of adaptability. On one side the hardware fabrics can be used to establish optimized accelerators for computational intensive algorithms. On the other, the same fabrics also allow to constitute full interrupt-capable processor instances. Thereby, based on the characteristics of an application or thread, the ISA of a processor instance can be changed in order to e.g. increase performance or reduce resource consumption. In that way, the architecture allows to flexibly exploit a wide range of instruction-, data-, and thread-level parallelism (ILP, DLP, TLP).

In the past less focus was put on the programmability of reconfigurable architectures. Thus, they are lacking compilers that allow generating code from high-level languages. Consequently, software development becomes a manual, time-consuming task. In contrast, along with the micro architecture development we also research software development methodologies for hypermorphic processor architectures closing the gap between high level programmability and reconfigurability. Therefore, a novel software toolchain is required supporting the new degree of freedom offered by the reconfigurable instruction format and

set respectively. Especially, a flexible code generator as part of the compiler is necessary supporting multiple instruction formats even within one application. Therefore, we developed ADL-based extensions to the LLVM compiler infrastructure in order to enable user retargetability.

In a top-down approach application development can start from an available C/C++ implementation. Thereby, the compiler framework supports the developer by detecting partitions of different ILP inside one application. This information can be used to compile the partitions to optimized instruction formats enabling efficient utilization of the available hardware. In that way, an application gradually can be optimized also offering the flexibility to exploit TLP beside of ILP. If required the performance can further be increased by including architecture-specific, hand-optimized hardware accelerators.

In addition, the ADL is further used for generating binary utilities and an instruction set simulator (ISS) embedded in the software toolchain suitable for design-space exploration (DSE). Especially, the simulator is required for compiler validation, performance estimation, and application characterization.

Dynamic data folding through run-time reconfiguration

Dirk Stroobandt (Gent University, BE)

We present a new run-time reconfiguration technique for FPGAs, called parameterizable reconfiguration, that is perfectly tailored to dynamic data folding applications. The technique takes care of the notion that slowly varying inputs (called parameters) do not change their value in a certain time interval and the FPGA implementation can therefore be made a lot more resource efficient. Whenever a parameter changes its values, this is reflected in a reconfiguration of the FPGA function. Parameterizable reconfiguration allows a very fast run-time reconfiguration by preparing the design (off-line) in such a way that the parameter change only requires an online Boolean function evaluation and subsequent FPGA reconfiguration. All NP hard problems (synthesis, placement and routing) remain to be done off-line and are no longer in the critical path of the run-time reconfiguration. This opens up new possibilities for a lot of applications with slowly varying inputs.

Keywords: FPGA, parameterizable run-time reconfiguration, dynamic data folding

Joint work of: Stroobandt, Dirk; Bruneel, Karel

Reconfigurable Architectures in 2020

Juergen Teich (Universität Erlangen-Nürnberg, DE)

We discuss different ways how reconfigurable computing devices will be part of future Many-Core System-on-a-Chip (MPSoC) solutions such as the integration as co-processors of standard processors tile.

Or, will we see architectures with Islands of individual FPGA modules being interconnected by a high speed network on a chip (NoC)? Or, will fine grain reconfigurable architectures such as FPGAs vanish in the era of 1000 processors on a chip?

We try to give a prognosis based on other currently available technology such as GPUs and discuss possible architectural evolutions based on the dominating factors of a) efficiency, b) flexibility (programmability) and c) productivity.

A new project to address run-time reconfigurable hardware systems

Jim Torresen (University of Oslo, NO)

Last autumn, we started a new project named Context Switching Reconfigurable Hardware for Communication Systems (COSRECOS). In this talk, I would like to present how we plan to address the challenge of changing hardware configurations while a system is in operation. The overall goal of the project is to contribute in making run-time reconfigurable systems more feasible in general.

This includes introducing architectures for reducing reconfiguration time as well as undertaking tool development. Case studies by applications in network and communication systems will be a part of the project. Comments to the planned outline are much welcome.

Joint work of: Koch, Dirk; Torresen, Jim

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2894>

The GAP Processor - a Processor with a Two-dimensional Execution Unit

Sascha Uhrig (Universität Augsburg, DE)

One of the main characteristics of (dynamically) reconfigurable systems is the need for a control processor. Additionally, special knowledge is required to use the reconfigurable architecture efficiently. Standard applications cannot profit from these architecture except if they are recompiled with some automatic support the reconfigurable system.

In this talk we will present the so-called Grid Alu Processor (GAP) architecture that is optimized for the execution of sequential instruction streams generated by a standard compiler like the GCC. A special control processor is not required to make use of the reconfigurable architecture.

The GAP comprises an in-order superscalar pipeline front-end enhanced by a configuration unit able to dynamically issue standard machine instructions to the functional units, which are organized in a two-dimensional array.

In contrast to well-known reconfigurable architectures no special synthesis tools are required and (nearly) no configuration overhead occurs.

The gain of the proposed processor architecture is obtained by the asynchronous execution of most instructions inside the array, the possibility to issue multiple depending instructions in the same cycle and the acceleration of loops.

Keywords: Reconfigurable processor, asynchronous execution, 2D ALU grid

Joint work of: Uhrig, Sascha; Shehan, Basher; Jahr, Ralf

Mojette implementations and applications in mobile communication

Jozsef Vasarhelyi (University of Miskolc, HU)

The Mojette transform (MOT) is an exact discrete Radon transform defined for a set of specific projections. This method was introduced by Jean-Pierre Guédon, although it is a simple transform using only additions, but it can be used from image processing applications up to distributed databases due to its properties. These properties will be introduced not only for the Mojette but also for the Inverse Mojette transform (IMOT). The paper presents the implementations of the transform in .NET environment and based on different projections and different realizations. Afterwards the HW implementation of the Mojette with FPGA is introduced and compared in speed computation with other implementations. Also describes possible applications of the Mojette transform in mobile communications on a simple example of a movie rental system with different approaches, to make it possible for the users to watch clips, trailers or even movies while they are not at home.

Keywords: Mojette transform, Security, distributed databases, image processing

Joint work of: Vasarhelyi, Jozsef; Szoboszlai, Peter; Serfözö; Peter, Turan, Jan

Full Paper:

<http://www.ieeexplore.org>

Full Paper:

<http://www.iccc.uni-miskolc.hu>

See also: 53. Serfözö P., Vászárhelyi, J., Szoboszlai P., Turan, J.; Performance requirements of the Mojette transform for internet distributed databases and image processing, IEEE OPTIM 2008. 11th International Conference on Optimization of Electrical and Electronic Equipment, 2008., Brasov, Romania, IEEE Digital Object Identifier 10.1109/OPTIM.2008.4602504, 22-24 May 2008 pp.:87 - 92;

Security and dynamically reconfigurable architectures: an introduction

Ingrid Verbauwhede (K.U. Leuven, BE)

Dynamically reconfigurable architectures are or will become platforms of choice for many applications, including cryptography and security applications.

Because they can adapt on the spot to the changing requirements of the application they are well suited for implementing crypto algorithms for small embedded platforms as well as high throughput applications in the cloud. From this end, it is probably no different from other application domains.

But new forms of programmability or reconfiguration, also introduce new ways of attacking the devices. This includes both active attacks and passive side channel leakage attacks.

Therefore from a security viewpoint it is important to identify the different forms of reconfiguration.

This presentation aims at linking secure embedded systems with dynamically reconfigurable architectures.

Keywords: Secure embedded systems, cryptography, dynamically reconfigurable architectures

New Directions for IP Core Watermarking and Identification

Daniel Ziener (Universität Erlangen-Nürnberg, DE)

In this talk, we present watermarking and identification techniques for FPGA IP cores. Unlike most existing watermarking techniques, the focus of our techniques lies on ease of verification, even if the protected cores are embedded into a product. Moreover, we have concentrated on higher abstraction levels for embedding the watermark, particularly at the logic level, where IP cores are distributed as netlist cores. With the presented watermarking methods, it is possible to watermark IP cores at the logic level and identify them with a high likelihood and in a reproducible way in a purchased product from a company that is suspected to have committed IP fraud. The investigated techniques establish the authorship by verification of either an FPGA bitfile or the power consumption of a given FPGA.

Keywords: IP protection, IP cores, watermarking

Joint work of: Ziener, Daniel; Teich, Jürgen

Full Paper: <http://drops.dagstuhl.de/opus/volltexte/2010/2843>