

FPGA'12: Preliminary Program

Wednesday, 22 February

2:00 PM

Pre-Conference Workshop

FPGAs in 2032: Challenges and Opportunities in the next 20 years

Chair: Vaughn Betz, University of Toronto

Chair: Lesley Shannon, Simon Fraser University

Abstract: This year marks the 20th anniversary of the FPGA Symposium, so it is fitting that this workshop will look forward to the changes that the next 20 years are likely to bring to programmable systems. A panel of visionaries from industry and academia will present their thoughts on major research areas, challenges and opportunities that will emerge over the coming two decades. Questions abound, from what the software flow in 2032 will be, to what architectures will suit chips with 100 billion transistors, and what the fabrication technology will be.

Presenters:

Mr. Bob Blainey, IBM Fellow, Compiler and Next-Generation System Software

Dr. Ivo Bolsens, CTO and Senior Vice-President of Xilinx

Dr. Misha Burich, CTO and Senior Vice-President of R & D of Altera

Professor Peter Cheung, Head of the Department of Electrical and Electronic Engineering, Imperial College London

Dr. Michael Flynn, Chairman of Maxeler Technologies and Professor Emeritus at Stanford University

Mr. Shep Siegel, Founder and CTO of Atomic Rules

Dr. Steve Teig, President and CTO of Tabula

6:00 PM

Registration

7:00 PM

Reception

Thursday, 23 February

8:00 AM Continental Breakfast, Registration

8:40 AM Opening Remarks

9:00 AM **Session 1: Applications I**

Intra-Masking Dual-Rail Memory on LUT Implementation for Tamper Resistant AES on FPGA

Anh-Tuan Hoang and Takeshi Fujino, Ritsumeikan University, Japan

Speedy FPGA-Based Packet Classifiers with Low On-Chip Memory Requirements

Chih-Hsun Chou and Nian-Feng Tzeng, University of Louisiana, USA
Fong Pong, Broadcom Corporation

A real-time stereo vision system using a tree-structured dynamic programming on FPGA (short)

Minxi Jin and Tsutomu Maruyama, University of Tsukuba, Japan

Incremental Clustering Applied to Radar De-Interleaving: A Parameterized FPGA Implementation (short)

Scott Bailie, MIT Lincoln Laboratory, USA
Miriam Leeser, Northeastern University

X-ORCA: FPGA-Based Wireless Localization in the Sub-Millimeter Range (short)

Matthias Hinkfoth, Enrico Heinrich, Sebastian Vorköper, Volker Kühn and Ralf Salomon
University of Rostock, Germany

Communication Visualization for Bottleneck Detection of High-Level Synthesis Applications (short)

John Curreri, Intel, USA
Greg Stitt and Alan George, University of Florida, USA

10:00 AM **Break: Posters 1** (60 minutes)

11:00 AM **Session 2: Design Studies**

CONNECT: Re-Examining Conventional Wisdom for Designing NoCs in the Context of FPGAs

Michael Papamichael and James Hoe, Carnegie Mellon University, USA

A Performance and Energy Comparison of FPGAs, GPUs, and Multicores for Sliding-Window Applications

Jeremy Fowers, Greg Brown, Patrick Cooke and Greg Stitt, University of Florida

A Mixed Precision Monte Carlo Methodology for Reconfigurable Accelerator Systems

Gary Chun Tak Chow, Qiwei Jin, Anson Hong Tak Tse, Wayne Luk and David Thomas
Imperial College, London, UK
Philip Leong, University of Sydney, Australia

Saturating the Transceiver Bandwidth: Switch Fabric Design on FPGAs

Zefu Dai and Jianwen Zhu, University of Toronto

12:20 PM

Lunch (1 hour, 40 minutes)

2:00 PM

Session 3: CAD

The VTR Project: Architecture and CAD for FPGAs from Verilog to Routing

Jonathan Rose, Jason Luu, Jason Anderson and Opal Densmore, University of Toronto

Andrew Somerville and Kenneth B. Kent, University of New Brunswick

Chi Wai Yu, City University of Hong Kong

Jeffrey Goeders, University of British Columbia

Peter Jamieson, University of Miami, Ohio, USA

Compiling High Throughput Network Processors

Maysam Lavasani and Derek Chiou, University of Texas at Austin, USA

Larry Dennison, Lightwolf Technologies

Limit Study of Energy & Delay Benefits of Component-Specific Routing

Raphael Rubin and Andre DeHon, University of Pennsylvania

Nikil Mehta, Caltech

Analyzing and Predicting the Impact of CAD Algorithm Noise on FPGA Speed Performance and Power (short)

Warren Shum and Jason Anderson, University of Toronto

Impact of FPGA Architecture on Resource Sharing in High-Level Synthesis (short)

Stefan Hadjis, Andrew Canis, Jason Anderson, Jongsok Choi, Kevin Nam and

Stephen Brown, University of Toronto

Tomasz Czajkowski, Altera Corporation

A Fast Discrete Placement Algorithm for FPGAs (short)

Qinghong Wu, Synopsys Inc.

Ken McElvain, University of California, Berkeley

3:15 PM

Break: Posters 2 (60 minutes)

4:15 PM

Session 4: Architecture I

Rethinking FPGAs: Elude the Flexibility Excess of LUTs with And-Inverter Cones

Hadi P. Afshar, David Novo, Paolo lenne and Hind Benbihi, EPFL

Securing Netlist-Level FPGA Design Through Exploiting Process Variation and Degradation

Jason Zheng and Miodrag Potkonjak, UCLA

*Prototype and Evaluation of the CoRAM Memory Architecture for
FPGA-Based Computing (short)*

Eric Chung, Michael Papamichael, Gabriel Weisz, James Hoe and Ken Mai,
Carnegie Mellon University

7:00 PM **Banquet at the Monterey Bay Aquarium** (transportation is provided).

Friday, 24 February

8:00 AM **Breakfast/Registration**

9:00 AM **Session 5: Applications II**

*A Coarse-Grained Stream Architecture for Cryo-Electron Microscopy
Images 3D Reconstruction*

Wendi Wang, Bo Duan, Wen Tang, Chunming Zhang, Guangming Tan, Peiheng
Zhang and Ninghui Sun, ICT,CAS,China

*A Cycle-Accurate, Cycle-Reproducible Multi-FPGA System for
Accelerating Multi-Core Processor Simulation*

Sameh Asaad, Ralph Bellofatto, Bernard Brezzo, Chuck Haymes, Mohit Kapur, Benjamin
Parker, Thomas Roewer, Proshanta Saha, Todd Takken and Jose Tierno, IBM

FPGA-Accelerated 3D Reconstruction using Compressive Sensing (short)

Jianwen Chen, Jason Cong and Yi Zou, UCLA

*Reconfigurable Architecture and Automated Design Flow for Rapid
FPGA-based LDPC Code Emulation (short)*

Haoran Li, Youn Sung Park and Zhengya Zhang, University of Michigan, Ann Arbor

*Reliability of a Softcore Processor in a Commercial SRAM-Based FPGA
(short)*

Nathaniel Rollins and Michael Wirthlin, Brigham Young University

9:55 AM **Break: Posters 3** (60 minutes)

10:55 AM **FPGA-20 Presentation**

11:15 AM **Session 6: Tools and Abstractions**

Leveraging Latency Insensitivity to Ease Multiple FPGA Design

Kermin Fleming, MIT

Michael Adler, Joel Emer, Michael Pellauer and Angshuman Parashar, Intel

A Scalable Approach for Automated Precision Analysis

David Boland and George Constantinides, Imperial College, London

Optimizing SDRAM Bandwidth for Custom FPGA Loop Accelerators

Samuel Bayliss and George Anthony Constantinides, Imperial College, London

VirtualRC: A Virtual FPGA Platform for Applications and Tools Portability (short)

Robert Kirchgessner, Greg Stitt, Alan George and Herman Lam, University of Florida

12:20 PM **Lunch** (1 hour, 40 minutes)

2:00 PM **Session 7: Compute Engines and Run-Time Systems**

Multi-Ported Memories for FPGAs via XOR

Eric LaForest, Ming Gang Liu, Emma Rapati and J. Gregory Steffan

University of Toronto

Octavo: an FPGA-Centric Processor Family

Eric LaForest and J. Gregory Steffan, University of Toronto

Accelerator Compiler for the VENICE Vector Processor (short)

Zhiduo Liu, Aaron Severance, and Guy Lemieux, University of British Columbia

Satnam Singh, Google

FCache: A System For Cache Coherent Processing on FPGAs (short)

Vincent Mirian and Paul Chow, University of Toronto

A Lean FPGA Soft Processor Built Using a DSP Block (short)

Hui Yan Cheah, Suhaib A. Fahmy, Douglas Maskell, Nanyang Technological University

Chidamber Kulkarni, Xilinx Inc.

Functionally Verifying State Saving and Restoration in Dynamically Reconfigurable Systems (short)

Lingkan Gong and Oliver Diessel, University of New South Wales

3:00 PM **Break: Posters 4** (60 minutes)

4:00 PM

Session 8: Architecture II

A Configurable Architecture to Limit Wakeup Current in Dynamically-Controlled Power-Gated FPGAs

Assem A. M. Bsoul and Steven J. E. Wilton, University of British Columbia

Reducing the Cost of Floating-Point Mantissa Alignment and Normalization in FPGAs

Yehdhih Ould Mohammed Moctar, UC Riverside
Hadi P. Afshar, Nithin George, Paolo lenne, EPFL
Guy Lemieux, University of British Columbia
Philip Brisk, UC Riverside

Abstracts

Intra-Masking Dual-Rail Memory on LUT Implementation for Tamper Resistant AES on FPGA

Anh-Tuan Hoang and Takeshi Fujino, Ritsumeikan University, Japan

In the current countermeasure design trends against Different Power Analysis (DPA), the security at the gates level is concentrated for its independent to the security algorithm. Several Dual-rail with pre-charge logics (DPL) are indicated to achieve that goal. The design on ASIC can attain the goal with its backend design restriction on placement and routing. However, implement them into Field Programmable Gate Array (FPGA) without information leakage is still a problem due to the difficulty in place and route restriction on FPGA.

This paper describes our novel Masked Dual-rail Pre-charged Memory approach, called "Intra-Masking Dual-Rail Memory on LUT" and its implementation on FPGA for the tamper resistant AES. The proposed design packs all the unsafe nodes such as the unmasking and masking as well as the dual-rail memory, dual-rail buses into a single LUT, makes them balanced and independent with the placement and routing tools. The design is independent with the cryptographic algorithm, and so can be applied to the available cryptographic standards such as DES or AES as well as the future standards. It requires no special place and route constraints in the implementation. The Correlation Power Analysis (CPA) attack on 1,000,000 traces of AES implementation on FPGA shows that the secret information is well protected against the first order side-channel attack. Even the number of LUTs used as memory in this implementation is 7 times bigger than that of the conventional unprotected single-rail memory Table-lookup AES and 3 times bigger than the implementation based on the Composite Field, it occupies a smaller number of LUTs than all other advanced tamper resistant implementations such as the Wave Dynamic Differential Logic, the Masked Dual-Rail Pre-charge Logic, and the Threshold.

Speedy FPGA-Based Packet Classifiers with Low On-Chip Memory Requirements

Chih-Hsun Chou and Nian-Feng Tzeng, University of Louisiana, USA
Fong Pong, Broadcom Corporation

This article pursues speedy packet classification with low on-chip memory requirements realized on Xilinx Virtex-6 FPGA. Based on hashing round-down prefixes specified in filter rules (dubbed HaRP), our implemented classifier is demonstrated to exhibit an extremely low on-chip memory requirement (lowering the byte count per rule by a factor of 8.6 in comparison with its most recent counterpart [2]), taking only 50% of Virtex-6 on-chip memory to store every large rule dataset (with some 30K rules) examined. In addition, it achieves a higher throughput than any known FPGA implementation, reaching more than 200 MPPS (millions packet lookups per second) with 8 processing units and 8 memory banks in the HaRP pipeline to support the line rate over 130 Gbps under bi-directional traffic in the worst case with 40-byte packets. By reducing memory probes per lookup, enhanced HaRP can further boost the classification speed to 255 MPPS.

A real-time stereo vision system using a tree-structured dynamic programming on FPGA (short)

Minxi Jin and Tsutomu Maruyama, University of Tsukuba, Japan

Many hardware systems for stereo vision have been proposed. Their processing speed is very fast, and reaches to several hundred frames per second. However, the algorithms used in those systems are limited in order to achieve the high processing speed by simplifying the sequences of the memory accesses and operations, and the error rate by those systems can not compete with those by software programs. To achieve high accuracy on the depth map with simple hardware systems is important for using them in the practical field. In this paper, we describe an FPGA implementation of a tree-structured dynamic programming algorithm. The computational complexity of this algorithm is higher than those by previous hardware systems, but the processing speed of our system is still fast enough for real-time applications without using a large number of on-chip memory banks, and its error rate is competitive with software algorithms.

Incremental Clustering Applied to Radar De-Interleaving: A Parameterized FPGA Implementation (short)

Scott Bailie, MIT Lincoln Laboratory, USA

Miriam Leeser, Northeastern University

We introduce ICED: Incremental Clustering of Evolving Data. ICED is a novel incremental clustering algorithm designed to effectively cluster data whose characteristics change over time. ICED is an unsupervised clustering technique that assumes no prior knowledge of the incoming data, and supports removing clusters that contain stale data. ICED is designed to be configurable to adapt to different applications and was developed with an FPGA implementation in mind. A combination of compile time parameters including number of clusters and run time parameters including distance threshold and fade cycle length, gives the user control over the implementation. The FPGA implementation of ICED has been applied to an important radar application: pulse deinterleaving. ICED is the first implementation of incremental clustering on an FPGA which we are aware. The implementation runs 39 times faster than an equivalent C implementation on a 3GHz Intel Xeon processor, and is capable of processing radar data received in real time.

X-ORCA: FPGA-Based Wireless Localization in the Sub-Millimeter Range (short)

Matthias Hinkfoth, Enrico Heinrich, Sebastian Vorköper, Volker Kühn and Ralf Salomon
University of Rostock, Germany

Recent research has developed a new, entirely digital architecture, called α -SYS, that determines the phase shift of two periodic signals with a resolution as good as about 20 ps. This paper incorporates the α -SYS system into a wireless experimental setup to form a localization system. The practical experiments utilize a 2.484 GHz transmitter and runs the α -SYS core on a Cyclone II FPGA. The results indicate that this simple localization system easily yields a spatial resolution in the sub-millimeter range.

Communication Visualization for Bottleneck Detection of High-Level Synthesis Applications (short)

John Curreri, Intel, USA

Greg Stitt and Alan George, University of Florida, USA

High-level synthesis tools increase FPGA productivity by allowing developers to specify circuit behavior at a higher abstraction level. However, because a higher abstraction can decrease performance compared to register-transfer level designs, application developers need bottleneck detection techniques to help optimize high-level code. Currently, identifying bottlenecks can be particularly challenging due to limitations of existing high-level synthesis tools that provide few performance analysis capabilities. In this paper, we address this problem by introducing a communication-bottleneck detection tool that provides a developer with a high-level visualization of the communication bandwidth between all processes on both a CPU and FPGA, while graphically identifying bottlenecks via color coding. We evaluated the presented techniques for third-party Impulse-C implementations of Triple DES and Molecular Dynamics, which identified bottlenecks that we optimized in just several minutes to achieve speedups of 2.18x and 1.25x compared to the original FPGA execution on an XD1000 platform. Bottlenecks were also detected on a Backprojection application on the Novo-G supercomputing platform (24 nodes each with two GiDEL PROCStar III boards for a total of 192 FPGAs). The measured overhead of the tool was less than 2 percent resource overhead and 3 percent frequency overhead.

CONNECT: Re-Examining Conventional Wisdom for Designing NoCs in the Context of FPGAs

Michael Papamichael and James Hoe, Carnegie Mellon University, USA

An FPGA is a peculiar hardware realization substrate in terms of the relative speed and cost of logic vs. wires vs. memory. In this paper, we present a Network-on-Chip (NoC) design study from the mindset of NoC as a synthesizable infrastructural element to support emerging System-on-Chip (SoC) applications on FPGAs. To support our study, we developed CONNECT, an NoC generator that can produce synthesizable RTL designs of FPGA-tuned multi-node NoCs of arbitrary topology. The CONNECT NoC architecture embodies a set of FPGA-motivated design principles that uniquely influence key NoC design decisions, such as topology, link width, router pipeline depth, network buffer sizing, and flow control. We evaluate CONNECT across multiple router configurations, as well as entire CONNECT networks, and report FPGA synthesis results and network performance results. We also evaluate CONNECT against a high-quality publicly available synthesizable RTL-level NoC design intended for ASICs, both before and after applying proper RTL coding discipline for FPGA synthesis. Our evaluation shows a significant gain in specializing NoC design decisions to FPGAs' unique mapping and operating characteristics. For example, in the case of a 4x4 mesh configuration evaluated using a set of synthetic traffic patterns, we obtain comparable or better performance than the baseline NoC, while reducing logic resource cost by 58% .

A Performance and Energy Comparison of FPGAs, GPUs, and Multicores for Sliding-Window Applications

Jeremy Fowers, Greg Brown, Patrick Cooke and Greg Stitt, University of Florida

With the emergence of accelerator devices such as multicores, graphics-processing units (GPUs), and field-programmable gate arrays (FPGAs), application designers are confronted with the problem of searching a huge design space that has been shown to have widely varying performance and energy metrics for different accelerators, different application domains, and different use cases. To address this problem, numerous studies have evaluated specific applications across different accelerators. In this paper, we analyze an important domain of applications, referred to as sliding-window applications, when executing on FPGAs, GPUs, and multicores. For each device, we present optimization strategies and analyze use cases where each device is most effective. The results show that FPGAs can achieve speedup of up to 11x and 57x compared to GPUs and multicores, respectively, while also using orders of magnitude less energy.

A Mixed Precision Monte Carlo Methodology for Reconfigurable Accelerator Systems

Gary Chun Tak Chow, Qiwei Jin, Anson Hong Tak Tse, Wayne Luk and David Thomas
Imperial College, London, UK
Philip Leong, University of Sydney, Australia

This paper introduces a novel mixed precision methodology applicable to any Monte Carlo (MC) simulation. It involves the use of data-paths with reduced precision, and the resulting errors are corrected by auxiliary sampling. An analytical model is developed for a reconfigurable accelerator system with a field-programmable gate array (FPGA) and a general purpose processor (GPP). Optimisation based on mixed integer geometric programming is employed for determining the optimal reduced precision and optimal resource allocation among the MC data-paths and correction datapaths. Experiments show that the proposed mixed precision methodology requires up to 11 % additional evaluations while less than 4 % of all the evaluations are computed in the reference precision; the resulting designs are up to 7.1 times faster and 3.1 times more energy efficient than baseline double precision FPGA designs, and up to 163 times faster and 170 times more energy efficient than quad-core software designs optimised with the Intel compiler and Math Kernel Library. Our methodology also produces designs for pricing Asian options which are 4.6 times faster and 5.5 times more energy efficient than NVIDIA Tesla C2070 GPU implementations.

Saturating the Transceiver Bandwidth: Switch Fabric Design on FPGAs

Zefu Dai and Jianwen Zhu, University of Toronto

Driven by the demand of communication systems, field programmable gate array (FPGA) devices have significantly enhanced their aggregate transceiver bandwidth, reaching terabits per second for the upcoming generation. This paper asks the question whether a single-chip switch can be built that saturates the available transceiver bandwidth.

In answering this question, we propose a new switch organization, called Grouped Crosspoint Queued switch, that brings significant memory efficiency over the state-of-the-art organizations. This makes it possible to build high bandwidth, high radix switches directly on FPGA that rivals ASIC performance. The proposal was validated at small scale by a 16x16 160Gps switch on the available Virtex-6 device, and simulated at a larger scale of fat-tree switching network with 5Tbps capacity.

The VTR Project: Architecture and CAD for FPGAs from Verilog to Routing

Jonathan Rose, Jason Luu, Jason Anderson and Opal Densmore

Ken Kent, University of New Brunswick

Chi Wai Yu, City University of Hong Kong

Peter Jamieson, University of Miami, Ohio, USA

As the semiconductor industry moves towards large-scale heterogeneous chips, it is likely that one portion of these systems will contain an FPGA. To facilitate the development of FPGA architectures and CAD tools for those systems, as well as pure-play FPGAs, there is a need for a large scale, publicly available software suite that can synthesize circuits into hypothetical human-described FPGA architectures. These circuits should be captured at the HDL level, or higher, and pass through logical and physical synthesis. Such a tool must provide detailed modeling of area, performance and energy to enable architecture exploration. As software flows themselves evolve to permit design capture at even high levels of abstraction, this downstream full-implementation flow will always be required. This paper describes the current status and new release of an ongoing effort to create such a flow - the 'Verilog to Routing' (VTR) project, which is a broad collaboration of researchers. There are three core tools: ODIN II for Verilog Elaboration and front-end hard-block synthesis, ABC for logic synthesis, and VPR for physical synthesis and analysis. ODIN II now has a simulation capability to help verify that its output is correct, as well as specialized synthesis at the elaboration step for multipliers and memories. ABC is used to optimize the 'soft' logic of the FPGA. The VPR-based packing, placement and routing is now fully timing-driven (the previous release was not) and includes new capability to target complex logic blocks. In addition we have added a set of four large benchmark circuits to a suite of previously-released Verilog HDL circuits. We illustrate the use of the new flow it to show how it can help architect a floating-point unit in an FPGA, in an off-the-shelf flow, and contrast it with a prior, longer effort that was required to do the same thing.

Compiling High Throughput Network Processors

Maysam Lavasani and Derek Chiou, University of Texas at Austin, USA
Larry Dennison, Lightwolf Technologies

Gorilla is a programming model, a canonical architecture, and a tool chain that generates very high performance, low power solutions for data parallel applications with fine grain irregularity. Irregularity simultaneously destroys performance and increases power consumption on many data parallel processors such as GPGPUs. Gorilla achieves high performance and low power through the use of FPGA-tailored parallelization techniques and application specific hardwired accelerators, processing engines, and communication mechanisms. Automatic compilation from C and the intrinsic flexibility of FPGAs provide programmability. We demonstrate Gorilla's capability by using it to generate a family of core-router network processors processing up to 100Gbps (200MPPS for 64B packets) supporting any mix of IPv4, IPv6, and multi-labeled MPLS packets on a single FPGA with off-chip lookup tables. We ran a 40Gbps version of Gorilla on a Xilinx Virtex-6 FPGA, verifying for performance and correctness, and measured power consumption comparable to full custom, commercial network processors. We also demonstrate how Gorilla can be used to save FPGA resources by generating merged virtual routers.

Limit Study of Energy & Delay Benefits of Component-Specific Routing

Raphael Rubin and Andre DeHon, University of Pennsylvania
Nikil Mehta, Caltech

As feature sizes scale toward atomic limits, parameter variation continues to increase, leading to increased margins in both delay and energy. The possibility of very slow devices on critical paths forces designers to reduce clock speed and operate at higher voltages than desired in order to meet timing. With post-fabrication configurability, FPGAs have the opportunity to use slow devices on non-critical paths while selecting fast devices for critical paths. This requires the extraction of delay information on a component-specific basis and the ability to use this information to map each FPGA differently--- both potentially expensive operations. To motivate this work, we first want to understand the potential benefit we might gain from component-specific mapping. We quantify the margins associated with parameter variation in FPGAs over a wide range of predictive technologies (45nm--12nm) and gate sizes and show how these margins can be significantly reduced by delay-aware, component-specific routing. For the Toronto 20 benchmark set we show that component-specific routing can improve delay by 1.2-1.8x and reduce energy for energy minimal designs by 1.3-1.9x. We further show that these benefits increase as technology scales.

Analyzing and Predicting the Impact of CAD Algorithm Noise on FPGA Speed Performance and Power (short)

Warren Shum and Jason Anderson, University of Toronto

FPGA CAD algorithms are heuristic, and generally make use of cost functions to gauge the value of one potential circuit implementation over another. At times, such algorithms must decide between two or more implementation options of apparently equal cost. This work explores the variations in circuit quality (i.e. noise) that arise when CAD algorithms are altered to choose randomly when faced with such equal-cost alternatives. Noise sources are identified in logic synthesis and technology mapping algorithms, and experimental results are presented which show standard deviations of 3.3% and 3.7% from the mean in post-routed delay and power. As a means of dealing with this variation, early timing and power prediction metrics can be applied after technology mapping to find the best circuits in the presence of noise. These metrics are able to find circuits with 1.3-1.8% lower critical path delay and 1.4% lower dynamic power, on average, when applied to a set of noise-injected circuits.

Impact of FPGA Architecture on Resource Sharing in High-Level Synthesis (short)

Stefan Hadjis, Andrew Canis, Jason Anderson, Jongsok Choi, Kevin Nam and Stephen Brown, University of Toronto
Tomasz Czajkowski, Altera Corporation

Resource sharing is a key area-reduction approach in highlevel synthesis (HLS) in which a single hardware functional unit is used to implement multiple operations in the highlevel circuit specification. Sharing a functional unit normally involves adding multiplexers to its inputs, which are costly to implement in FPGAs. Hence the prevailing sentiment is that sharing is rarely useful for FPGAs, except when the shared resource is either very large, or is relatively scarce. In this paper, we re-visit the resource sharing question in the FPGA context and show that the utility of sharing depends on the underlying FPGA logic element architecture. Specifically, we show that different sharing trade-offs exist when 4-LUTs vs. 6-LUTs are used. We further show that certain multi-operator patterns (e.g. add followed by subtract) occur multiple times in programs, thereby creating additional opportunities for sharing larger composite functional units – units comprised of patterns of interconnected operators. The sharing cost/benefit analysis is used to inform decisions made in the binding phase of an HLS tool, whose RTL output is targeted to Altera commercial FPGA families: Stratix IV (uses 6-LUTs) and Cyclone II (uses 4-LUTs). Results show resource sharing in HLS can reduce area by up to 38% for Stratix IV designs (8-12%, on average), and 47% for Cyclone II designs (7-16%, on average).

A Fast Discrete Placement Algorithm for FPGAs (short)

Qinghong Wu, Synopsys Inc.

Ken McElvain, University of California, Berkeley

Good FPGA placement is crucial to obtain the best Quality of Results (QoR) from FPGA hardware. Although many published global placement techniques place objects in a continuous ASIC-like environment, FPGAs are discrete in nature, and a continuous algorithm cannot always achieve superior QoR by itself. Therefore, discrete FPGA-specific detail placement algorithms are used to improve the global placement results. Unfortunately, most of these detail placement algorithms do not have a global view. This paper presents a discrete placer that fills the gap between the two placement steps. It works like simulated annealing, but leverages various acceleration techniques. It does not pay the runtime penalty typical of simulated annealing solutions. Experiments show that with this placer, final QoR is significantly better than with the global-detail placer approach.

Multi-Ported Memories for FPGAs via XOR

Eric LaForest, Ming Gang Liu, Emma Rapati and J. Gregory Steffan

University of Toronto

Multi-ported memories are challenging to implement with FPGAs since the block RAMs included in the fabric typically have only two ports. Any design that requires a memory with more than two ports must therefore be built out of logic elements or by combining multiple block RAMs. The recently-proposed Live Value Table (LVT) design provides a significant operating frequency improvement over conventional approaches. In this paper we present an alternative approach based on the XOR operation that provides multi-ported memories that use far less logic but more block RAMs than LVT designs, and are often smaller and faster for memories that are more than 512 entries deep. We show that (i) both designs can exploit multipumping to trade speed for area savings, (ii) that multipumped XOR designs are significantly smaller but moderately slower than their LVT counterparts, and (iii) that both the LVT and XOR approaches are valuable and useful in different situations, depending on the constraints and resource utilization of the enclosing design.

Octavo: an FPGA-Centric Processor Family

Eric LaForest and J. Gregory Steffan, University of Toronto

Overlay processor architectures allow FPGAs to be programmed by non-experts using software, but prior designs have mainly been based on the architecture of their ASIC predecessors. In this paper we develop a new processor architecture that from the beginning accounts for and exploits the predefined widths, depths, maximum operating frequencies, and other discretizations and limits of the underlying FPGA components. The result is Octavo, an eight-pipeline-stage eight-threaded processor that operates at the BRAM maximum of 550MHz on a Stratix IV FPGA. Octavo is highly parameterized, allowing us to explore trade-offs in datapath and memory width, memory depth, and number of supported thread contexts.

Accelerator Compiler and the VENICE Soft Vector Processor (short)

Zhiduo Liu, Aaron Severance, and Guy Lemieux, University of British Columbia
Satnam Singh, Google

This paper describes the compiler design for VENICE, a new soft vector processor (SVP) compactly implemented on an FPGA. The compiler is based on adding a new back-end target to Microsoft Accelerator, a highly data parallel library for C++ and C#. This allows us to automatically compile high-level programs into VENICE assembly code, thus avoiding the process of writing inline assembly code used by previous SVPs. Also, unlike previous SVPs which were focussed on scaling efficiently to a large number of parallel ALUs, we take a new direction with VENICE by optimizing it particularly for a small number of ALUs while also improving performance. As a result, VENICE is both smaller and faster than previous soft vector processors, offering over 2× better performance-per-logic block than the previous best, VEGAS [3]. Experimental results show the compiler can generate scalable parallel code with execution times that are comparable to hand-written VENICE assembly code. On data-parallel applications, VENICE at 100MHz on a DE3 runs at speeds comparable to a 3.4GHz Intel i7-2600 processor, beating it in performance on three of seven benchmarks by up to 3.2×.

FCache: A System For Cache Coherent Processing on FPGAs (short)

Vincent Mirian and Paul Chow, University of Toronto

Much like other processing units in the world today, FPGAs are becoming increasingly larger and contain large amounts of reconfigurable logic. This makes FPGAs an acceptable platform for multiprocessor systems. However in today's world of FPGA computing, very limited infrastructure facilitates the creation of shared memory systems for FPGAs. This paper introduces FCache, a distributed system for cache coherent processing on FPGAs. Its underlining infrastructure simplifies the creation of multiprocessor shared memory systems on FPGAs.

FCache has many features embedded in its design. It maps the conventional shared bus to two networks on the FPGA, which exploits parallelism in its cache coherent system. It manages coherency with messages and maintains consistency by timestamping these messages. Its modularized cache design allows for ease in implementing various other cache configurations. Its simple interface allows for heterogeneous processors to co-exist in a single instance of its kind. FCache also contains embedded functionality (such as locking and unlocking a mutex variable), which facilitates the programming of these heterogeneous processors for parallel applications.

Results show that FCache services more than one main memory request. However, unbalanced network utilization is observed. This gives incentive to implement cache data transfer over the under utilized network in FCache which would further increase performance.

Leveraging FPGA DSP Blocks for General Computation (short)

Hui Yan Cheah, Suhaib A. Fahmy, Douglas Maskell, Nanyang Technological University
Chidamber Kulkarni, Xilinx Inc.

As Field Programmable Gate Arrays (FPGAs) have advanced, the capabilities and variety of embedded resources have increased. One of the main driving application domains during the early years of FPGAs was signal processing. Engineers realised the ease with which inherent parallelism in such algorithms could be exploited to achieve significant acceleration. Hence, FPGA vendors began tailoring their architecture to such applications. First, embedded hard multipliers were added, then these were developed into embedded multiplier-accumulator blocks, and now these embedded digital signal processing (DSP) blocks have advanced to the point of supporting a wide range of operations. Here we explore how these processors can be leveraged for general computation with minimal addition of extra logic, and while attempting to leverage existing compilation techniques. We show that the DSP48E1 blocks in Xilinx Virtex-6 devices support a wide range of standard machine instructions and how they can be designed into the core of a basic processor.

Functionally Verifying State Saving and Restoration in Dynamically Reconfigurable Systems (short)

Lingkan Gong and Oliver Diessel, University of New South Wales

Dynamically reconfigurable systems increase design density and flexibility by allowing hardware modules to be swapped at run time. Systems that employ checkpointing, periodic or phased execution, preemptive multitasking and resource defragmentation, may also need to be able to save and restore the state of a module that is being reconfigured. Existing tools verify the functionality of a system that is undergoing reconfiguration. These tools can also be employed if state is accessed using application logic. However, when state is accessed via the configuration port, functional verification is hindered because the FPGA fabric, which in the system mediates the transfer of state between the application logic and the configuration port, is not being simulated. We describe how to efficiently simulate those aspects of the fabric that are used in accessing state via the configuration port. To the best of our knowledge, this work is the first to allow cycle-accurate simulation of a system partially reconfiguring both its logic and state and a case study shows that our method is effective in detecting device independent design errors.

A Coarse-Grained Stream Architecture for Cryo-Electron Microscopy Images 3D Reconstruction

Wendi Wang, Bo Duan, Wen Tang, Chunming Zhang, Guangming Tan, Peiheng Zhang and Ninghui Sun, ICT,CAS,China

The wide acceptance of bioinformatics, medical imaging and multimedia applications, which have a data-centric favor to them, require more efficient and application-specific systems to be built. Due to the advances in modern FPGA technologies recently, there has been a resurgence in research aimed at accelerator design that leverages FPGAs to accelerate large-scale scientific applications. In this paper, we exploit this trend towards FPGA-based accelerator design and provide a proof-of-concept and comprehensive case study on FPGA-based accelerator design for a single-particle 3D reconstruction application. The proposed stream architecture is built by first offloading computing-intensive software kernels to dedicated hardware modules, which emphasizes the importance of optimizing computing dominated data access patterns. Then configurable computing streams are constructed by arranging the hardware modules and bypass channels to form a linear deep pipeline. The efficiency of the proposed stream architecture is justified by the reported 2.54x speedup over a 4-cores CPU. In terms of power efficiency, our FPGA-based accelerator introduces a 7.33x and a 3.4x improvement over a 4-cores CPU and an up-to-date GPU device, respectively.

A Cycle-Accurate, Cycle-Reproducible Multi-FPGA System for Accelerating Multi-Core Processor Simulation

Sameh Asaad, Ralph Bellofatto, Bernard Brezzo, Chuck Haymes, Mohit Kapur, Benjamin Parker, Thomas Roewer, Proshanta Saha, Todd Takken and Jose Tierno, IBM

Software based tools for simulation are not keeping up with the demands for increased chip and system design complexity. In this paper, we describe a cycle-accurate and cycle-reproducible large-scale FPGA platform that is designed from the ground up to accelerate logic verification of the Bluegene/Q compute node ASIC, a multi-processor SOC implemented in IBM's 45 nm SOI CMOS technology. This paper discusses the challenges for constructing such large-scale FPGA platforms, including design partitioning, clocking & synchronization, and debugging support, as well as our approach for addressing these challenges without sacrificing cycle accuracy and cycle reproducibility. The resulting fullchip simulation of the Bluegene/Q compute node ASIC runs at a simulated processor clock speed of 4 MHz, over 100,000 times faster than the logic level software simulation of the same design. The vast increase in simulation speed provides a new capability in the design cycle that proved to be instrumental in logic verification as well as early software development and performance validation for Bluegene/Q.

FPGA-Accelerated 3D Reconstruction using Compressive Sensing (short)

Jianwen Chen, Jason Cong and Yi Zou, UCLA

The radiation dose associated with computerized tomography (CT) is significant. Compressive sensing methods provide mathematic approaches to reduce the radiation exposure, without sacrificing image quality. However, the computational requirement of the algorithm is prohibitive, and much higher than conventional image reconstruction algorithm like Feldkamp-Davis-Kress (FDK) algorithm. This paper describes an FPGA implementation of one compressive sensing algorithm with applications on CT image reconstruction. The ray tracing forward and backward projection procedures have abundant random off-chip accesses, as well as load-balancing issues, and is a good fit of the multi-FPGA platform that excels in interleaved memory access. Our FPGA EM kernel is 50% faster than the GPU implementation on Tesla C1060. Moreover, our FPGA kernel can process two independent images at the same time, and thus the kernel throughput is 3X of Tesla C1060 and slightly better than the Fermi GTX480 GPU. Moreover, our EM kernel is deployed in a hybrid (CPU+GPU+FPGA) computer, and we show that the hybrid approach delivers a better performance and energy than GPU-only solutions.

Reconfigurable Architecture and Automated Design Flow for Rapid FPGA-based LDPC Code Emulation (short)

Haoran Li, Youn Sung Park and Zhengya Zhang, University of Michigan, Ann Arbor

Multitude of design freedoms of LDPC codes and practical decoders require fast simulations. FPGA emulation is attractive but inaccessible due to its design complexity. We propose a library and script based approach to automate the construction of FPGA emulations. Code parameters and design parameters are programmed either during run time or by script in design time. We demonstrate the architecture and design flow using the LDPC codes for the latest wireless communication standards: each emulation model was auto-constructed within one minute and the peak emulation throughput reached 3.8 Gb/s on a BEE3 platform.

Reliability of a Softcore Processor in a Commercial SRAM-Based FPGA (short)

Nathaniel Rollins and Michael Wirthlin, Brigham Young University

Softcore processors are an attractive alternative to using radiation-hardened (rad-hard) processors in space-based applications. Compared to commercial processors, rad-hard processors are slow, consume more area, consume more power, and are extremely expensive. Compared to rad-hard processors, softcore processors are fast, exible, less expensive, and recon gurable. Unlike traditional processors, the logic and routing of a softcore processor are vulnerable to the effects of single-event upsets (SEUs). This paper applies two common SEU mitigation techniques, TMR with checkpointing and DWC with checkpointing, to the LEON3 Softcore processor. The improvement in reliability over an unmitigated version of the processor is measured using three metrics: the architectural vulnerability factor (AVF), mean time to failure (MTTF), and mean instructions to failure (MITF). Using con guration memory fault injection, we found that DWC with checkpointing improves the MTTF and MITF by 10 and 8 respectively. TMR with checkpointing improves the MTTF and MITF by 22 and 19 respectively. This paper also quanti es the SEU vulnerability of each of the major components of the LEON3 processor using fault injection.

A Configurable Architecture to Limit Wakeup Current in Dynamically-Controlled Power-Gated FPGAs

Assem A. M. Bsoul and Steven J. E. Wilton, University of British Columbia

A dynamically-controlled power-gated (DCPG) FPGA architecture has recently been proposed to reduce static energy dissipation during idle periods. During a power mode transition from an off state to on state, the wakeup current drawn from power supplies causes a voltage droop on the power distribution network of a device. If not handled appropriately, this voltage droop could cause malfunction of the design and/or the device. In DCPG FPGAs, the amount of wakeup current is not known beforehand as the structures of power-gated modules are application dependent; thus, a configurable solution is required to handle wakeup current. In this paper we propose a programmable wakeup architecture for DCPG FPGAs. The proposed solution has two levels: a fixed intra-region level and a configurable inter-region level. The architecture ensures that different power gating regions in a power-gated module can be turned on such that the wakeup current constraints are not violated. We study the area and power overheads of the proposed solution. Our results show that the area overhead of the proposed inrush current limiting architecture is less than 2% for a power gating region of size 3x3 or 4x4 tiles, and the leakage power saved is more than 85% in a region of size 4x4 tiles.

Reducing the Cost of Floating-Point Mantissa Alignment and Normalization in FPGAs

Yehdhih Ould Mohammed Moctar, UC Riverside
Hadi Parandeh-Afshar, Nithin George, Paolo Ienne, EPFL
Guy Lemieux, University of British Columbia
Philip Brisk, UC Riverside

In floating-point datapaths synthesized on FPGAs, the shifters that perform mantissa alignment and normalization consume a disproportionate number of LUTs. Shifters are implemented using several rows of small multiplexers; unfortunately, multiplexer-based logic structures map poorly onto LUTs. FPGAs, meanwhile, contain a large number of multiplexers in the programmable routing network; these multiplexers are placed under static control of the FPGA's configuration bitstream. Some of the multiplexers in the FPGA's Input Interconnect Blocks (IIBs) are exposed to the user logic under dynamic control. By mapping portions of the shifters onto these modified multiplexers, the number of Configurable Logic Blocks (CLBs) required to implement the shifters required to implement mantissa alignment and normalization for floating-point FPGAs are reduced by 67%. If dynamic multiplexing is not required, the IIBs containing modified multiplexers can be configured as normal. The area overhead incurred by modifying an IIB is small, and there is no need to modify every IIB in the FPGA. No modifications to the FPGA's global routing network are necessary, and experiments show that there is no negative impact in terms of clock frequency or routability for benchmarks that do not use the dynamic multiplexers.

Rethinking FPGAs: Elude LUT Flexibility Excess with And-Inverter Cones

Hadi P. Afshar, David Novo, Paolo Ienne and Hind Benbihi, EPFL

Look-Up Tables (LUTs) are universally used in FPGAs as the elementary logic blocks. They can implement any logic function and thus covering a circuit is a relatively straightforward problem. Naturally, flexibility comes at a price, and increasing the number of LUT inputs to cover larger parts of a circuit has an exponential cost in the LUT complexity. Hence, rarely LUTs with more than 4--6 inputs have been used. In this paper we argue that other elementary logic blocks can provide a better compromise between hardware complexity, flexibility, delay, and input and output counts. Inspired by recent trends in synthesis and verification, we explore blocks based on And-Inverter Graphs (AIGs): they have a complexity which is only linear in the number of inputs, they sport the potential for multiple independent outputs, and the delay is only logarithmic in the number of inputs. Of course, these new blocks are extremely less flexible than LUTs; yet, we show (i) that effective mapping algorithms exist, (ii) that, due to their simplicity, poor utilization is less of an issue than with LUTs, and (iii) that a few LUTs can still be used in extreme unfortunate cases. We show first results indicating that this new logic block combined to some LUTs in hybrid FPGAs can reduce delay up to 25--35% and area by some 16% on average. Yet, we explored only a few design points and we think that these results could still be improved by a more systematic exploration.

Securing Netlist-Level FPGA Design through Exploiting Process Variation and Degradation

Jason Zheng and Miodrag Potkonjak, UCLA

The continuously widening gap between the NRE and RE cost of producing IC products in the past few decades gives high incentives to unauthorized cloning and reverse-engineering of ICs. Existing IC DRM schemes often demand high overhead in area, power, and performance, or require non-volatile storage. Our goal is to develop a novel IP protection technique that not only protects circuit designs from cloning and reverse engineering, but also offers capabilities to remotely disable the device. In this paper we show a proof-of-concept implementation of the basic elements of the technique, as well as a case study of applying the anti-cloning technique to a nontrivial FPGA design.

The Feasibility and Effectiveness of the CoRAM Memory Architecture for FPGA-Based Computing (short)

Eric Chung, Michael Papamichael, Gabriel Weisz, James Hoe and Ken Mai,
Carnegie Mellon University

The CoRAM memory architecture for FPGA-based computing augments traditional sea-of-gates fabric with a natural and effective interface abstraction for applications to interact with the off-chip environment, most critically the main memory. The two central tenets of the CoRAM memory architecture are (1) the deliberate separation of concerns between computation versus data marshaling and (2) a multithreaded software abstraction to express data marshaling control. In this paper, we report our study of a concrete instance of the CoRAM architecture where we developed and examined in detail the control thread programming environment and the required microarchitecture-level mechanisms. For this work, we developed a working language and compiler for the control threads and a full RTL-level implementation of a CoRAM microarchitecture instance that can be synthesized for standard cells or prototyped on FPGAs. Our results argue for the CoRAM architecture's effectiveness in providing ease of programming and portability for application development without undue overhead in performance and hardware cost. Synthesis results suggest that CoRAM support can be added with a modest 2% overhead relative to the die area with a power overhead of 7% under pessimistic assumptions.

Easing Multiple FPGA Design with Latency Insensitive Bounded Dataflow Networks

Kermin Fleming, MIT

Michael Adler, Joel Emer, Michael Pellauer and Angshuman Parashar, Intel

Traditionally, hardware designs partitioned across multiple FPGAs have had low performance, due to the inefficiency of maintaining cycle-by-cycle timing among discrete FPGAs. In this paper, we present a mechanism by which complex designs may be efficiently and automatically partitioned among multiple FPGAs using explicitly programmed latency-insensitive links. We describe the automatic synthesis of an area efficient, high performance network for routing these inter-FPGA links. By mapping a diverse set of large research prototypes onto a multiple FPGA platform, we demonstrate that our tool obtains significant gains in design feasibility, compilation time, and even wall-clock performance.

A Scalable Approach for Automated Precision Analysis

David Boland and George Constantinides, Imperial College, London

The freedom over the choice of numerical precision is one of the key factors that can only be exploited throughout the datapath of an FPGA accelerator, providing the ability to trade the accuracy of the final computational result with the silicon area, power, operating frequency, and latency. The field of word-length optimization has developed with the aim of tapping into this potential, and various publications have demonstrated that performance benefits can be obtained by tuning the precision in specific algorithms. However, in order to perform such optimizations automatically and create reliable hardware for general algorithms, a tool is required to verify that the hardware will meet an error or range specification for a given precision. Unfortunately, existing tools to perform this task typically suffer either from a lack of tightness of bounds or require a large execution time when applied to large scale algorithms. In this work, we propose an approach that can both scale to larger examples and obtain tighter bounds, within a smaller execution time, than the existing methods. The approach we describe also provides a user with the ability to trade the quality of bounds with execution time of the procedure, making it suitable within a word-length optimization framework for both small and large-scale algorithms. %and also allow a much greater control of this trade-off to allow a user to create superior hardware designs.

We demonstrate the use of our approach on instances of iterative algorithms to find the solution to a system of linear equations. We show that because our approach can track how the relative error decreases with increasing precision, unlike the existing methods, we can use it to create hardware with guaranteed error properties. This results in a saving of 25% of the area in comparison to optimizing the precision using competing analytical techniques, whilst requiring a smaller execution time than the existing methods, and saving almost 80% of area in comparison to adopting IEEE double precision arithmetic.

Optimizing SDRAM Bandwidth for Custom FPGA Loop Accelerators

Samuel Bayliss and George Anthony Constantinides, Imperial College, London

Memory bandwidth is critical to achieving high performance in many FPGA applications. The bandwidth of SDRAM memories is, however, highly dependent upon the order in which addresses are presented on the SDRAM interface. We present an automated tool for constructing an application specific on-chip memory hierarchy which presents requests to the external memory with an ordering which optimizes off-chip memory bandwidth for fixed on-chip memory resource. Within a class of algorithms described by affine loop nests, this approach can be shown to reduce both the number of requests made to external memory and the overhead associated with those requests. Data presented shows a trade-off between the use of on-chip resources and achievable off-chip memory bandwidth where a range of improvements from 3.6x to 4x gain in efficiency on the external memory interface can be gained at a cost of a 1.05x to 1.60x increase in the ALUTs dedicated to address generation circuits in an Altera Stratix III device.

VirtualRC: A Virtual FPGA Platform for Applications and Tools Portability (short)

Robert Kirchgessner, Greg Stitt, Alan George and Herman Lam, University of Florida

Numerous studies have shown that field-programmable gate arrays (FPGAs) are capable of significant speedup for important application domains at a fraction of the power of microprocessors and graphics-processing units (GPUs). Despite these benefits, the difficulties of FPGA application design have limited the use of FPGAs as application accelerators. One significant challenge is dealing with the lack of code portability across different FPGA platforms, which prevents design reuse techniques that have significantly improved productivity for other devices. In this paper, we present a novel method of enabling FPGA application and tool portability using a framework for FPGA platform virtualization, referred to as VirtualRC. We show that VirtualRC achieves portability with a modest performance overhead of 5-6% and area overhead of approximately 1%, while also demonstrating portability of eleven RTL applications and two high-level synthesis tools across three physical platforms.

Poster Session 1

Accelerating Short Read Mapping on an FPGA

Yupeng Chen, *Nanyang Technological University* Bertil Schmidt, *Johannes Gutenberg University Mainz* Douglas Leslie Maskell, *Nanyang Technological University*

The explosive growth of short read datasets produced by high throughput DNA sequencing technologies poses a challenge to the mapping of short reads to a reference genome in terms of sensitivity and execution speed. Existing methods often use a restrictive error model for computing the alignments to improve speed, whereas more flexible error models are generally too slow for large-scale applications. Although a number of short read mapping software tools have been proposed, designs based on hardware are relatively rare. In this paper, we present a hybrid system for short read mapping utilizing both software and field programmable gate array (FPGA)-based hardware. The compute intensive semi-global alignment operation is accelerated on the FPGA. The proposed FPGA aligner is implemented with a parallel block structure to gain computational efficiency. We also propose a block-wise alignment algorithm to approximate the score of the conventional dynamic programming algorithm. Our performance comparison shows that the FPGA achieves an average speedup of 38 for the alignment operation on a Xilinx Virtex5 FPGA compared to the GASSST software implementation. For the overall execution time, our hybrid system achieves an average speedup of 2.4 compared to GASSST at comparable sensitivity and an average speedup of 1.8 compared to the popular BWA software at a significantly better sensitivity.

The Masala Machine: Accelerating Thread-Intensive and Explicit Memory Management Programs with Dynamically Reconfigurable FPGAs

Mei Wen, Nan Wu, Qianming Yang, Chunyuan Zhang, Liang Zhao, *National University of Defense Technology*

A uniform FPGA-based architecture, an efficient programming model and a simple mapping method are paramount for PPGA technology to be more widely accepted. This paper presents MASALA, a dynamically reconfigurable FPGA-based accelerator specifically for parallel programs written in thread-intensive and explicit memory management (TEMM) programming models. The system uses TEMM programming model to parallelize the demanding application, including decomposing the application into separate thread blocks, decoupling compute and data load/store etc. Hardware engines are included into the MASALA by using partial dynamic reconfigure modules, each of which encapsulates Thread Process Engine implementing the thread functionality in hardware. A data dispatching scheme is also included in MASALA to enable the explicit communication among multiple memory hierarchies such as between inter- hardware engines, the host processor and hardware engines. At last, the paper illustrates a Multi-FPGA prototype system of the presented architecture: MASALA-SX. A large synthetic aperture radar (SAR) image formatting experiment shows that the MASALA architecture facilitates the construction of a TEMM program accelerator by providing it with greater performance and less power consumption than current CPU platforms, but without sacrificing programmability, flexibility and scalability.

Timing Yield Improvement of FPGAs Utilizing Enhanced Architectures and Multiple Configurations Under Process Variation

Fatemeh Sadat Pourhashemi, Morteza Saheb Zamani, *Amirkabir University of Technology*

Designing with field-programmable gate arrays (FPGAs) can face with difficulties due to process variations. Some techniques use reconfigurability of FPGAs to reduce the effects of process variations in these chips. Furthermore, FPGA architecture enhancement is an effective way to degrade the impact of variation. In this paper, various FPGA architectures are examined to identify which architecture can achieve larger parametric yield improvement utilizing multiple configurations as opposed to single configuration. Experimental results show that by increasing cluster size from 4 to 10, yield improvement increases from 2.82X to 4.48X. However, changing look-up table (LUT) size from 4 to 7 results in yield improvement degradation from 2.82X to 1.45X, using 10 configurations compared to single configuration over 20 MCNC benchmark circuits. These results indicate that multi- configuration technique causes larger timing yield improvement in FPGAs with larger cluster size and smaller LUT size.

A Field Programmable Array Core for Image Processing

Declan Walsh, Piotr Dudek, *The University of Manchester*

Massively parallel processor arrays have been shown to be an effective and suitable choice for image processing tasks [1]. More recently, some of the state of the art processor arrays have been used for real-time machine vision tasks such as intelligent transport system applications [2] or video processing on mobile applications [3] providing a much more powerful solution than a conventional processor. A number of Single Instruction Multiple Data (SIMD) processor arrays have been implemented on FPGAs [4]-[6], which are particularly suited to implementing such processor architectures because of their similarities of both being arrays of fine grained logic elements. In this work, we propose an FPGA implementation of a processor array where the processing elements (PEs) are as small as possible, while providing local memory sufficient for processing greyscale images. The PE is then replicated to form an array. A 32×32 PE array is implemented on a Xilinx Virtex 5 XC5VLX50 FPGA using the four-neighbour connectivity with the possibility to scale up using a larger FPGA. The processor array operates at a frequency of 96 MHz and executes a peak of 98.3 giga operations per second (GOPS) (bit-serial operations). A binary edge detection algorithm is performed in 52.08 ns. Uploading and downloading a binary image in a 32×32 array takes an extra 687.5 ns. Sobel edge detection of an 8-bit greyscale image is performed in 5.33 μ s. Uploading and downloading an 8-bit greyscale image in a 32×32 array takes 5.36 μ s. With larger FPGAs being available in the future, the array sizes comparable to state of the art custom designed ICs can be implemented on these FPGAs.

EmPower: FPGA Based Emulation of Dynamic Power Management Algorithms for Multi-Core Systems on Chip

Sundaram Ananthanarayanan, *Anna University* Chirag Ravishankar, Siddharth Garg, Andrew Kennings, *University of Waterloo*

Dynamic power management for multi-core system on chip (MPSoC) platforms has become an increasingly critical design problem. In this paper, we present EmPower, an FPGA based emulation, validation and prototyping framework for dynamic power management research targeted at MPSoC platforms. EmPower supports two advanced power management features – per-core dynamic frequency scaling and clock gating, and power-aware thread migration. We also provide two fully-functional parallel applications for benchmarking—video encoding and software-defined radio. Our experimental results indicate that EmPower provides up to 36X improvement in run-time compared to cycle-accurate software simulations, and enables accurate and efficient exploration of the design space of power management algorithms.

Poster Session 2

Adaptive FPGA-Based Robotics State Machine Architecture Derived with Genetic Algorithms

Jesus Savage, Rodrigo Savage, *Universidad Nacional Autonoma de Mexico*
Marco Morales-Aguirre, Angel Kuri-Morales, *Instituto Tecnologico Autonomo de Mexico*

This paper discusses how to generate mobile robots' behaviors using genetic algorithms, GA. The behaviors are built using state machines implemented in a programmable logic device, an FPGA, and they are encoded in such a way that a state machine architecture executes them, controlling the overall operation of a small mobile robot. The behaviors generated by the GA are evaluated according to a fitness function that grades their performance. Basically, the fitness function evaluates the robot's performance when it goes from an origin to a destination. In our approach each individuals' chromosome represents, given a set of inputs coming from the sensors and the current state, the next state and outputs that controls the robot's movements. For each generation the GA needs to evaluate population's individuals, doing this with the real robot it would require to much time, that would be impossible to do. Thus, the GA needs a simulator, as close as it can be to the real robot and its environment. The simulator gets the individuals' chromosomes and executes the algorithm state machine represented by them, it simulates the movements of the robot depending of the output generated in the present state and the simulated robot's sensors. Our objective was to prove that GA is a good option as a method for finding behaviors for mobile robots' navigation and also that these behaviors can be implemented in FPGAs.

A Novel Full Coverage Test Method for CLBs in FPGA

Yong Fu, Chi Wang, Liguang Chen, Jinmei Lai, *Fudan University*

FPGA's configurability makes it difficult for FPGA's manufacturers to fully test it. In this paper, a full coverage test method for FPGA's Configurable Logic Blocks (CLBs) is proposed, through which all basic logics of FPGA's every CLB can be fully tested. Innovative test circuits are configured to build repeatable logic arrays for look-up tables, distributed random access memories, configurable registers and other logics. The programmable interconnects needed to connect CLBs in these test circuits are also repeatable, making the configuration process much easier and the test speed much faster. The test method is implemented on different scales of Xilinx Virtex chips, where 19 test configuration circuits are needed to achieve 100% coverage for all CLBs. Besides, the method is transplantable and independent of FPGA's array size. To evaluate the test method reliably and guide the process of test vectors generation, a fault simulator - Turbofault is used to simulate FPGA's test coverage.

Constraint-Driven Automatic Generation of Interconnect for Partially Reconfigurable Architectures

Andre Seffrin, *Center for Advanced Security Research Darmstadt*
Sorin A. Huss, *TU Darmstadt*

Dynamic partial reconfiguration allows the exchange of hardware configurations on FPGAs at run-time. Within a reconfigurable system that supports several different modules, resource requirements for interconnect between these modules may be considerably high. Enabling communication via a crossbar may require too many resources. State-of-the-art modelling methods for partial dynamic reconfiguration already support the fine-grained description of interaction between the partial modules. We propose both an online and an offline method for automatically generating interconnect according to such communication constraints, aiming at a low resource usage. The online algorithm determines an appropriate port assignment for the partial modules by means of a greedy approach and exploits port overlaps. The offline algorithm employs simulated annealing in order to find a proper port assignment and also incorporates the scheme for exploiting port overlaps. Constraint-generated interconnect requires significantly less resources than a crossbar, even if only a random port assignment is used. Proper port assignment by the online method reduces these requirements by an additional 10%, and using the offline method reduces them by an additional 30% on average. Online port assignment is faster than the offline method by several orders of magnitude. The interconnect generation tool introduced in this work takes textual input of communication constraints and automatically generates a corresponding hardware description in VHDL.

Thermal-Aware Logic Block Placement for 3D FPGAs Considering Lateral Heat Dissipation

Juinn-Dar Huang, Ya-Shih Huang, Mi-Yu Hsu, Han-Yuan Chang, *National Chiao Tung University*

Three-dimensional (3D) integration is an attractive and promising technology to keep Moore's Law alive, whereas the thermal issue also presents a critical challenge for 3D integrated circuits. Meanwhile, accurate thermal analysis is very time-consuming and thus can hardly be incorporated into most of placement algorithms generally performing numerous iterative refinement steps. As a consequence, in this paper, we first present a fine-grained grid-based thermal model for the 3D regular FPGA architecture and also highlight that lateral heat dissipation paths can no longer be assumed negligible. Then we propose two fast thermal-aware placement algorithms for 3D FPGAs, Standard Deviation (SD) and MineSweeper (MS), in which rapid thermal evaluation instead of slow detailed analysis is utilized. Moreover, both take the lateral heat dissipation into consideration and focus on distributing heat sources more evenly within a layer in a 3D FPGA to avoid creating hotspots. Experimental results show that SD and MS achieve 12.1%/7.6% reduction in maximum temperature and 82%/56% improvement in temperature deviation compared with a classical thermal-unaware placement method only at the cost of minor increase in wirelength and delay. Moreover, MS merely consumes 4% more runtime for producing thermal-aware placement solutions.

Power-Aware FPGA Technology Mapping for Programmable-VT Architectures

Wei Ting Loke, Xilinx Inc. & National University of Singapore

Yajun Ha, National University of Singapore

In this paper, we present a framework for leakage power reduction in FPGAs with programmable-VT architectures, with focus on dual-VT technology mapping. The use of Reverse Back Bias (RBB) circuit techniques is recognized as one of the possible strategies in mitigating leakage power, a critical problem in circuits deploying deep submicron process technologies. FPGAs with the ability to tune LUT VT via RBB offer the potential of reducing leakage power with no sacrifice to circuit speed. Today, Alteras Stratix line of FPGAs offer some levels of VT programmability, but with optimizations limited to the post-P&R stage. We present a novel technology mapper (RBBMap), logic block packer (RBBPack) and placement-and-routing tool (RBBVPR) that together demonstrate the advantages in moving RBB optimizations upwards to the technology mapping level. Compared to an existing power-optimized technology mapping tool Emap, our framework offers an average of 44.41% savings in average logic block leakage power and 30.88% savings in average total energy consumption. We also illustrate why our work is potentially superior to another comparable work DVMap-2 that utilizes a dual-VDD approach.

FPGA-RR: An Enhanced FPGA Architecture with RRAM-Based Reconfigurable Interconnects

Jason Cong, Bingjun Xiao, University of California, Los Angeles

In this study, we explore the use of Resistive RAMs (RRAMs) as candidates for programmable interconnects in FPGAs. An RRAM cell can be programmed between high resistance state and low resistance state, with an on/off ratio close to MOSFET. It provides an opportunity to use an RRAM as a routing switch at a much smaller area cost than its CMOS counterpart. RRAMs can be fabricated over CMOS circuits using CMOS-compatible processes to have a more compact gate array. Our recent work (presented in NanoArch'2011) demonstrated significant potential of area, delay, and power reduction from using RRAMs in FPGAs. But some design problems remain open. The programming of RRAM switches integrated in interconnects is one important problem. We show that the high-level architecture of programming circuits for RRAM switches should be modified to avoid potential logic hazard. Also the programming cells used in previous works have an area overhead even larger than RRAM itself. We manage to reduce this overhead significantly with utilization of the non arbitrary pattern of RRAM integration in FPGA interconnects. In addition we suggest a novel buffering solution for FPGA interconnects in light of the low area cost of RRAM-based routing switch. We propose on-demand buffer insertion, where buffers can be connected to interconnects via RRAMs to dynamically reflect the demand of the netlist to map onto FPGA. Compared to conventional buffering solution which are pre-determined during fabrication and can only be optimized for general case, our solution shows further area savings and performance improvement. The resulting FPGA architecture using RRAM for programmable interconnects is named FPGA-RR. We provide a complete CAD flow for FPGA-RR.

Poster Session 3

Efficient In-System RTL Verification and Debugging Using FPGAs

Proshanta Saha, Chuck Haymes, Ralph Bellofatto, Bernard Brezzo, Mohit Kapur, Sameh Asaad, IBM T.J. Watson Research Center

FPGAs have become indispensable in processor design, bring-up and debug. Traditionally FPGAs have been used in prototyping, allowing end-users to emulate functionality of a specific component of a processor. However, as the complexity of processors grows, another aspect of processor design, RTL verification, has become a prime target for acceleration using FPGAs. Software-only RTL simulation and verification tools are no longer sufficient for many verification tasks as they often incur long execution time penalties. Software simulation time for a basic Linux kernel bring-up on a BlueGene/Q [1] processor, with 16 user PowerPC A2 cores, for example, could easily exceed several years.

An important feature of RTL verification acceleration using FPGAs is its fast debugging capabilities. The ability to quickly and accurately pinpoint the location of an anomaly in an RTL source is highly desirable. This paper proposes efficient in-system debugging techniques on FPGAs for RTL verification. We show how a network of over 45 Virtex 5 LX330 FPGAs can be efficiently used to read out state information of the BlueGene/Q processor. We also demonstrate how the new in-system debugging technique is 250x faster than comparable methods.

Parallel FPGA Placement Based on Individual LUT Placement

Chris C. Wang, Guy G. F. Lemieux, University of British Columbia

This work describes a novel approach to FPGA placement. Most conventional FPGA CAD flow clusters circuits into CLBs prior to placing it. We show that it is possible to achieve 28% and 21% improvement in wirelength and minimum channel width respectively, while suffering only 1.8% in critical path delay by placing individual LUTs directly. By utilizing a good parallel placer, the novel approach can achieve speedups over the conventional uni-processor placers as well.

Dataflow-Driven Execution Control in a Coarse-Grained Reconfigurable Array

Robin Panda, Scott Hauck, University of Washington

Coarse Grained Reconfigurable Arrays (CGRAs) are a promising class of architectures for accelerating applications using a large number of parallel execution units for high throughput. While they are typically very efficient for a single task, all functional units are required to perform in lock step; this makes some classes of applications more difficult to program and efficiently use resources. Other architectures like Massively Parallel Processor Arrays (MPPAs) are better suited for these applications and excel at executing unrelated tasks simultaneously, but the amount of resources dedicated to a single task is limited.

We are developing a new architecture with the design flexibility of an MPPA and the throughput of a CGRA. A key to the flexibility of MPPAs is the ability for subtasks to execute independently instead of in lock step with all other tasks on the array. Adding this capability requires special control circuitry for architectural support in a CGRA. We describe the modifications required and our solutions. Additionally, we also describe the CAD tool modification and application developer concerns for utilizing the resulting hybrid CGRA/MPPA architecture.

OpenCL Memory Infrastructure for FPGAs

S. Alexander Chin, Paul Chow, University of Toronto

Programming models assist developers in creating high performance computing systems by forming a higher level abstraction of the target platform. OpenCL has emerged as a standard programming model for heterogeneous systems and there has been recent activity combining OpenCL and FPGAs. This work introduces memory infrastructure for FPGAs and is designed for OpenCL style computation, complementing previous work. An Aggregating Memory Controller is implemented in hardware and aims to maximize bandwidth to external, large, high-latency, high-bandwidth memories by finding the minimal number of external memory burst requests from a vector of requests. A template processing array with soft-processor and hand-coded hardware elements was also designed to drive the memory controller. The Aggregating Memory Controller is described in terms of operation and future scalability and the created processing array is described as a flexible structure that can support many types of processing solutions. A hardware prototype of the memory controller and processing array was implemented on a Virtex-5 LX110T FPGA. Two micro-benchmarks were run on both the soft-processor elements and the hand-coded hardware cores to exercise the memory controller. Results for effective memory bandwidth within the system show that the high-latency can be hidden using the Aggregating Memory Controller by increasing the number of threads within the processing array.

Operation Scheduling and Architecture Co-Synthesis for Energy-Efficient Dataflow Computations on FPGAs

Colin Yu Lin, Ngai Wong, Hayden Kwok-Hay So, The University of Hong Kong

Compiling high-level user applications for execution on FPGAs often involves synthesizing dataflow graphs beyond the size of the available on-chip computational resources. One way to address this is by folding the execution of the given dataflow graphs onto an array of directly connected simple configurable processing elements (CPEs). Under this scenario, the performance and energy-efficiency of the resulting system depends not only on the mapping schedule of the compute operations on the CPEs, but also on the topology of the interconnect array that connects the CPEs. This paper presents a framework in which the operation scheduler and the underlying CPE interconnect network topology are co-optimized on a per-application basis for energy-efficient FPGA computation. Given the same application, more than 2.5x difference in energy-efficiency was achievable by the use of different common regular array topologies to connect the CPEs. Moreover, by using irregular application-specific interconnect topologies derived from a genetic algorithm, up to 50% improvement in energy-delay-product was achievable when compared to the use of even the best regular topology. The use of such framework is anticipated to serve as part of a rapid high-level FPGA application compiler since minimum hardware place-and-route is needed to generate the optimal schedule and topology.

Poster Session 4

Post-Silicon Debugging Targeting Electrical Errors with Patchable Controllers

Masahiro Fujita, Hiroaki Yoshida, University of Tokyo

Due to continuous increase of design complexity in SoC development, the time required for post-silicon verification and debugging keeps increasing especially for electrical errors and subtle corner case bugs, and it is now understood that some sort of programmability in silicon is essential to reduce the time for post-silicon verification and debugging. Although an easiest way to achieve this is to use FPGA for entire circuits, performance especially in terms of power efficiency compared with pure hardwired logic may be significantly inferior. Here, we discuss partial use of such in-field programmability in control parts of circuits for post-silicon debugging processes for electrical errors and corner case logical bugs. Our method deals with RTL designs in FSM (Finite State Machine with Datapath) by adding partially in-field programmability, called 'patch logic', in their control parts. With our patch logic we can dynamically change the behaviors of circuits in such a way to trace state transition sequences as well as values of internal values periodically. Our patch logic can also check if there is any electrical error or not periodically. Assuming that electrical errors occur very infrequently, an error can be detected by comparing the equivalence on the results of duplicated computations. Through experiments we discuss the area, timing, and power overhead due to the patch logic and also show results on electrical error detection with duplicated computations..

Algorithm and Architecture Optimization for Large Size Two-Dimensional Discrete Fourier Transform

Berkin Akin, Peter A. Milder, Franz Franchetti, James C. Hoe, Carnegie Mellon University

We present a poster showcasing our FPGA implementations of two-dimensional discrete Fourier transform (2D-DFT) on large datasets that must reside off-chip in DRAM. These memory-bound large 2D-DFT computations are at the heart of important scientific computing and image processing applications. The central challenge in creating high-performance implementations is in the carefully orchestrated use of the available off-chip memory bandwidth and on-chip temporary storage. Our implementations derive their efficiency from a combined attention to both the algorithm design to enable efficient DRAM access patterns and datapath design to extract the maximum compute throughput at a given level of memory bandwidth. The poster reports results including a 1024x1024 double-precision 2D-DFT implementation on an Altera DE4 platform (based on a Stratix IV EP4SGX530 with 12 GB/s DRAM bandwidth) that reached over 16 Gflop/s, achieving a much higher ratio of performance-to-memory-bandwidth than both state-of-the-art CPU and GPU implementations.

Early Timing Estimation for System-Level Design Using FPGAs

Hugo Andrade, Arkadeb Ghosal, Rhishikesh Limaye, Sadia Malik, Newton Petersen, Kaushik Ravindran, Trung Tran, Guoqiang Wang, Guang Yang, National Instruments Corp.

FPGA devices provide flexible, fast, and low-cost prototyping and production solutions for system design. However, as the design complexity continues to rise, the design and synthesis iterations become a labor intensive and time consuming ordeal. Consequently, it becomes imperative to raise the level of abstraction for FPGA designs, while providing insight into performance metrics early in the design process. In particular, an important design time problem is to determine the maximum clock frequency that a circuit can achieve on a specific FPGA target before full synthesis and implementation. This early quantification can greatly help evaluate key design characteristics without reverting to tedious runs of the full implementation flow. In this work, we focus on the predictability of timing delay of circuits composed of high-level blocks on an FPGA. We are well aware of difficulties in tackling uncertainties in early timing estimation, e.g., an inherent gap between a high-level representation and gates/wires; extremely difficult delay estimation due to the randomness in physical design tools, etc. We show that the estimation uncertainties can be mitigated through a carefully characterized timing database of primitive building blocks and refined timing analysis models. We primarily focus on applications composed of data-intensive word-level arithmetic computations from the DSP domain and specified using static dataflow models. Our experiments indicate that for these applications, timing estimates can be obtained reliably within a good error margin on average and in the worst case. As future work, we plan to fine tune the timing database by modeling resource utilization effects and inter-primitive/actor routing delay via variants of Rent's rule and related efforts. We are also interested in exploring dynamic sub-cycle timing characterization.

Scalable Architecture for 135 GBPS IPV6 Lookup on FPGA

Yi-Hua E. Yang, Futurewei Technologies

Oguzhan Erdem, Middle-East Technical University Viktor K. Prasanna, University of Southern California

High-speed IP lookup remains a challenging problem in next generation routers due to the ever increasing line rate and routing table size. In addition, the evolution towards IPv6 also requires long prefix length, sparse prefix distribution, and potentially very large routing tables. Previous solutions have relied on complex trie compression as well as Bloom filters to achieve high forwarding rate for IPv6. In this paper, we propose a novel Combined Length-Infix Pipelined Search (CLIPS) architecture for IPv6 routing table lookup on FPGA. CLIPS solves the longest prefix match (LPM) problem by combining both prefix length and infix pattern search. Binary search in prefix length is performed on the 64-bit routing prefix of IPv6 down to an 8-bit length range in $\lceil \log_2(64/8) \rceil = 3$ phases; each phase performs a fully-pipelined infix pattern search with only one external memory access. A fourth and the last phase then finds the LPM (if any) within the 8-bit length range in a compressed multi-bit trie.

We describe the algorithms and data structures used for the CLIPS construction, run-time operation, dynamic update and false-positive avoidance. The proposed solution improves the on-chip memory efficiency on FPGA and maximizes the external SRAM utilization; additional properties for ensuring the practicality of our scheme include the modular construction, easy dynamic update, and simple resource allocation. Using a state-of-the-art FPGA, our CLIPS prototype supports up to 2.7 million IPv6 prefixes when employing 33 Mbits of BRAM and 4 channels of external SRAM. The prototype achieves a sustained throughput of 264 million IPv6 lookups per second, or 135 Gbps with minimum size (64-byte) packets.