ARC 2017 - Program

13th International Symposium on Applied Reconfigurable Computing
Delft, The Netherlands, April 3 - 7, 2017

TIME SCHEDULE

	April 3 Monday	April 4 Tuesday	April 5 Wednesday	April 6 Thursday	April 7 Friday
09:00	Tutorial 1: PYNQ (Xilinx)	Keynote 1: Onur Mutlu (ETH Zurich)	Keynote 2: Walid Najjar (UC Riverside)	Keynote 3: Patrick Lysaght (Xilinx)	Tutorial: ρ-Vex
10:00	**BREAK**				
10:30	Tutorial 1: PYNQ (cont.) (Xilinx)	Session 1: Adaptive Architectures	Session 4: Design Space Exploration	Session 6: FPGA based design	Tutorial: ρ-Vex (cont.)
12:00	**LUNCH**				
14:00	Tutorial 1: PYNQ (cont.) (Xilinx)	Session 2: Embedded Computing and Security	Session 5: Fault Tolerance	Best Paper Announcement Session 7: Neural Networks	Tutorial: ρ-Vex (cont.)
15:30	**BREAK**				
16:00	Tutorial 1: PYNQ (cont.) (Xilinx)	Session 3: Simulation and Synthesis	Social Event (including dinner)	Session 8: Languages and Estimation Techniques	Tutorial: ρ-Vex (cont.)

* FP = Full Paper SP = Short Paper

ARC 2017 - Program

13th International Symposium on Applied Reconfigurable Computing
Delft, The Netherlands, April 3 - 7, 2017

MONDAY (03/04/2017)

9:00	**Tutorial 1 – PYNQ (Xilinx)** *PYNQ is an open-source framework that enables programmers who want to use embedded systems to exploit the capabilities of Xilinx Zynq All Programmable SoCs (APSoC). It allows users to exploit custom hardware in the programmable logic without having to use ASIC-style CAD tools. Instead the APSoC is programmed in Python and the code is developed and tested directly on the embedded system. The programmable logic circuits are imported as hardware libraries and programmed through their APIs, in essentially the same way that software libraries are imported and programmed.* *The framework combines four main elements: (1) the use of a high-level productivity language, Python in this case; (2) Python-callable hardware libraries based on FPGA overlays; (3) a web-based architecture incorporating the open-source Jupyter Notebook infrastructure served from Zynq's embedded processors; and (4) Jupyter Notebook's client-side, web apps. The result is a web-centric programming environment that enables software programmers to work at higher levels of design abstraction and to re-use both software and hardware libraries.* *This tutorial will give a hands-on introduction to PYNQ framework. It will feature the latest version of PYNQ with Python 3.6 and Asyncio support for processor and fabric interrupts. Several new overlays will be introduced along with examples of overlay creation and binding into the PYNQ framework.*
BREAK	
10:30	**Tutorial 1 – PYNQ (Xilinx) – Cont.**
LUNCH	
14:00	**Tutorial 1 – PYNQ (Xilinx) – Cont.**
BREAK	
16:00	**Tutorial 1 – PYNQ (Xilinx) – Cont.**

* FP = Full Paper SP = Short Paper

ARC 2017 - Program

13th International Symposium on Applied Reconfigurable Computing
Delft, The Netherlands, April 3 - 7, 2017

TUESDAY (04/04/2017)

9:00	**Keynote 1 – Rethinking Memory System Design (and the Computing Platforms We Design Around It)**
	Speaker: Onur Mutlu (ETH Zurich) *The memory system is a fundamental performance and energy bottleneck in almost all computing systems. Recent system design, application, and technology trends that require more capacity, bandwidth, efficiency, and predictability out of the memory system make it an even more important system bottleneck. At the same time, DRAM and flash technologies are experiencing difficult technology scaling challenges that make the maintenance and enhancement of their capacity, energy efficiency, and reliability significantly more costly with conventional techniques. In fact, recent reliability issues with DRAM, such as the RowHammer problem, are already threatening system security and predictability.* *In this talk, we first discuss major challenges facing modern memory systems in the presence of greatly increasing demand for data and its fast analysis. We then examine some promising research and design directions to overcome these challenges and thus enable scalable memory systems for the future. We discuss three key solution directions: 1) enabling new memory architectures, functions, interfaces, and better integration of memory and the rest of the system, 2) designing a memory system that intelligently employs emerging non-volatile memory (NVM) technologies and coordinates memory and storage management, 3) reducing memory interference and providing predictable performance to applications sharing the memory system. If time permits, we will also touch upon our ongoing related work in combating scaling challenges of NAND flash memory.*
	BREAK
10:30	**Session 1 – Adaptive Architectures**
	Improving the Performance of Adaptive Cache in Reconfigurable VLIW Processor (FP) *Sensen Hu, Anthony Brandon, Qi Guo and Yizhuo Wang*
	LP-P^{2}IP: A Low-power Version of P^{2}IP Architecture using Partial Reconfiguration (FP) *Álvaro Avelino, Valentin Obac, Naim Harb, Carlos Valderrama, Glauberto Albuquerque and Paulo Possa*
	NIM: An HMC-based Machine for Neuron Computation (SP) *Geraldo F. Oliveira, Paulo C. Santos, Marco A. Z. Alves and Luigi Carro*
	VLIW-based FPGA Computation Fabric for Medical Imaging (SP) *Joost Hoozemans, Rolf Heij, Jeroen van Straten and Zaid Al-Ars*
	LUNCH
14:00	**Session 2 – Embedded Computing and Security**
	Hardware Sandboxing: A Novel Defense Paradigm Against Hardware Trojans in Systems on Chip (FP) *Christophe Bobda, Joshua Mead, Taylor Whitaker, Charles Kamhoua and Kevin Kwiat*
	Rapid Development of Gzip with MaxJ (FP) *Nils Voss, Tobias Becker, Oskar Mencer and Georgi Gaydadjiev*
	On the Use of (Non-)Cryptographic Hashes on FPGAs (SP) *Andreas Fiessler, Daniel Loebenberger, Sven Hager and Björn Scheuermann*
	An FPGA-based Implementation of a Pipelined FFT Processor for High-Speed Signal Processing Applications (SP) *Ngoc-Hung Nguyen, Sheraz Khan, Cheol-Hong Kim and Jong-Myon Kim*
	BREAK
16:00	**Session 3 – Simulation and Synthesis**
	Soft timing closure for soft programmable logic cores: The ARGen approach (FP) *Théotime Bollengier, Loïc Lagadec, Mohamad Najem, Jean-Christophe Le Lann and Pierre Guilloux*
	FPGA Debugging with MATLAB using a Rule-based Inference System (FP) *Habib Ul Hasan Khan and Diana Göhringer*
	Hardness Analysis and Instrumentation of Verilog Gate Level Code for FPGA-based Designs (SP) *Abdul Rafay Khatri, Ali Hayek and Josef Börcsök*
	A Framework for High Level Simulation and Optimization of Coarse-Grained Reconfigurable Architectures (SP) *Muhammad Adeel Pasha, Umer Farooq, Muhammad Ali and Bilal Siddiqui*

* FP = Full Paper SP = Short Paper

ARC 2017 - Program

13th International Symposium on Applied Reconfigurable Computing
Delft, The Netherlands, April 3 - 7, 2017

WEDNESDAY (05/04/2017)

9:00	**Keynote 2 – Acceleration Through Hardware Multithreading**
	Speaker: Walid Najjar (University of California Riverside, US) *Long memory latencies, as measure in CPU clock cycles, is probably the most daunting challenge to modern computer architecture. In multicore designs the long memory latency is mitigated with the use of massive cache hierarchies. This solution pre-supposes some forms of temporal or spatial localities. Irregular applications, by their very nature, suffer from poor data locality that results in high cache miss rates and long off-chip memory latency. Latency masking multithreading, where threads relinquish control after issuing a memory request, has been demonstrated as an effective approach to achieving a higher throughput. Multithreaded CPUs are designed for a fixed maximum number of threads tailored for an average application. FPGAs, however, can be customized to specific applications. Their massive parallelism is well know, and ideally suited to dynamically manage hundreds, or thousands, of threads. Multithreading, in essence, trades off memory bandwidth for latency. In this talk I describe how latency masking multithreaded execution on FPGAs can achieve a higher throughput that CPUs and/or GPUs on two sets of applications: sparse linear algebra and database operations.*
BREAK	
10:30	**Session 4 – Design Space Exploration**
	Parameter Sensitivity in Virtual FPGA Architectures (FP) *Peter Figuli, Weiqiao Ding, Shalina Percy Delicia Figuli, Kostas Siozios, Dimitrios Soudris and Jürgen Becker*
	Custom Framework for Run-time Trading Strategies (FP) *Andreea Ingrid Funie, Liucheng Guo, Xinyu Niu, Wayne Luk and Mark Salmon*
	Exploring HLS Optimizations for Efficient Stereo Matching Hardware Implementation (SP) *Karim M. A. Ali, Rabie Ben Atitallah, Nizar Fakhfakh and Jean-Luc Dekeyser*
	Architecture Reconfiguration as a Mechanism for Sustainable Performance of Embedded Systems in case of Variations in Available Power (SP) *Dimple Sharma, Victor Dimitriu and Lev Kirischian*
LUNCH	
14:00	**Session 5 – Fault Tolerance**
	Exploring Performance and Soft Error Recovery in Dual-Core LockStep ARM A9 Processor Embedded into Xilinx Zynq-7000 APSoC (FP) *Ádria Oliveira, Lucas Antunes Tambara and Fernanda Kastensmidt*
	Applying TMR in Hardware Accelerators Generated by High-Level Synthesis Design Flow for Mitigating Multiple Bit Upsets in SRAM-based FPGAs (FP) *André Flores Dos Santos, Fabio Benevenuti, Lucas Tambara, Jorge Tonfat and Fernanda Lima Kastensmidt*
BREAK	
16:00	**Social Event**

* FP = Full Paper SP = Short Paper

ARC 2017 - Program

13th International Symposium on Applied Reconfigurable Computing
Delft, The Netherlands, April 3 - 7, 2017

THURSDAY (06/04/2017)

9:00	**Keynote 3 – Enabling Software Engineers to Program Heterogeneous, Reconfigurable SoCs**
	Speaker: Patrick Lysaght (Xilinx Research Labs, San Jose, California) *In this talk, modern software trends will be explored with a focus on how we can enable software developers to exploit the benefits of reconfigurable hardware. This talk introduces PYNQ, a new open-source framework for designing with Xilinx Zynq devices, a class of All Programmable Systems on Chip (APSoCs) which integrates multiple processors and Field Programmable Gate Arrays (FPGAs) into single integrated circuits. The main goal of the framework is to make it easier for designers of embedded systems to use APSoCs in their applications. The APSoC is programmed in Python and the code is developed and tested directly on the embedded system. The programmable logic circuits are imported as hardware libraries and programmed through their APIs, in essentially the same way that software libraries are imported and programmed. The framework combines three main elements:* • *the use of a high-level productivity language, Python in this case* • *Python-callable hardware libraries based on FPGA overlays* • *a web-based architecture incorporating the open-source Jupyter Notebook infrastructure served from Zynq's embedded processors* *The result is a programming environment that is web-centric so it can be accessed from any browser on any computing platform or operating system. It enables software programmers to work at higher levels of design abstraction and to re-use both software and hardware libraries for reconfigurable computing. The framework is inherently extensible and integrates coherently with hardware–dependent code written in C and C++. The talk concludes with an outline of areas for continued development, and a call for community participation.*
	BREAK
10:30	**Session 6 – FPGA Based Designs**
	FPGA Applications in Unmanned Aerial Vehicles - A Review (FP) *Mustapha Bouhali, Farid Shamani, Zine Elabadine Dahmane, Abdelkader Belaidi and Jari Nurmi*
	Genomic Data Clustering on FPGAs for Compression (FP) *Enrico Petraglio, Rick Wertenbroek, Flavio Capitao, Nicolas Guex, Christian Iseli and Yann Thoma*
	A Quantitative Analysis of the Memory Architecture of FPGA-SoCs (FP) *Matthias Göbel, Ahmed Elhossini, Chi Ching Chi, Mauricio Alvarez Mesa and Ben Juurlink*
	LUNCH
14:00	**Best Paper Announcement**
14:10	**Session 7 – Neural Networks**
	Optimizing CNN-based Object Detection Algorithms on Embedded FPGA Platforms (FP) *Ruizhe Zhao, Xinyu Niu, Yajie Wu, Wayne Luk and Qiang Liu*
	An FPGA Realization of a Deep Convolutional Neural Network using a Threshold Neuron Pruning (FP) *Tomoya Fujii, Shimpei Sato, Hiroki Nakahara and Masato Motomura*
	Accuracy Evaluation of Long Short Term Memory Network Based Language Model with Fixed-Point Arithmetic (SP) *Ruochun Jin, Jingfei Jiang and Yong Dou*
	FPGA Implementation of a Short Read Mapping Accelerator (SP) *Mostafa Morshedi and Hamid Noori*
	BREAK
16:00	**Session 8 – Languages and Estimation Techniques**
	dfesnippets: An Open-Source Library for Data flow Acceleration on FPGAs (FP) *Paul Grigoras, Pavel Burovskiy, James Arram, Xinyu Niu, Kit Cheung, Junyi Xie and Wayne Luk*
	A Machine Learning Methodology for Cache Recommendation (FP) *Osvaldo Navarro, Jones Mori, Javier Ho mann, Fabian Stuckmann, and Michael Hübner*
	ArPALib: A Big Number Arithmetic Library for Hardware and Software implementations. A Case Study for the Miller-Rabin Primality Test (SP) *Jan Macheta, Agnieszka Dąbrowska-Boruch, Paweł Russek and Kazimierz Wiatr*

* FP = Full Paper SP = Short Paper

FRIDAY (07/04/2017)

9:00	**Tutorial 2 – ρ-Vex (Computer Engineering Laboratory, TUDelft)** *On the last day of ARC, a tutorial is organized to familiarize the participants with the rVEX platform that is developed at Delft University of Technology. It is an open-source implementation of a design-time reconfigurable and run-time parametrizable VLIW processor. Design-time reconfigurability is realized by the highly generic VHDL code. It comes with a complete toolchain, simulator, debug & trace hardware and interfacing software.* *The tutorial will highly 2 use cases of the platform; - The FPGA prototype of the dynamic core - An FPGA overlay fabric consisting of 64 cores running on 200MHz targeting streaming image processing workloads* *There will also be room for participants to port their application of interest to one (or both) the platforms to experiment with either the reconfigurable properties or the streaming fabric under guidance of the rVEX developers. We have an industrial grade compiler, floating point emulation, math and C standard libraries and a simply Linux port, so we expect to be able to run most applications that are not too complex.* *For more information about the platform, see http://rvex.ewi.tudelft.nl* *A full release (4.1) is available on the site if you wish to do some experiments before the tutorial.*
BREAK	
10:30	**Tutorial 2 – ρ-Vex (Computer Engineering Laboratory, TUDelft) – Cont.**
LUNCH	
14:00	**Tutorial 2 – ρ-Vex (Computer Engineering Laboratory, TUDelft) – Cont.**
BREAK	
16:00	**Tutorial 2 – ρ-Vex (Computer Engineering Laboratory, TUDelft) – Cont.**

* FP = Full Paper SP = Short Paper