Report from Dagstuhl Seminar 17061

Wildly Heterogeneous Post-CMOS Technologies Meet Software

Edited by

Jerónimo Castrillón-Mazo[1], Tei-Wei Kuo[2], Heike E. Riel[3], and Matthias Lieber[4]

1 **TU Dresden, DE,** `jeronimo.castrillon@tu-dresden.de`
2 **National Taiwan University – Taipei, TW,** `ktw@csie.ntu.edu.tw`
3 **IBM Research Zurich, CH,** `hei@zurich.ibm.com`
4 **TU Dresden, DE,** `matthias.lieber@tu-dresden.de`

Abstract

The end of exponential scaling in conventional CMOS technologies has been forecasted for many years by now. While advances in fabrication made it possible to reach limits beyond those predicted, the so anticipated end seems to be imminent today. The main goal of the seminar 17061 "Wildly Heterogeneous Post-CMOS Technologies Meet Software" was to discuss bridges between material research, hardware components and, ultimately, software for information processing systems. By bringing together experts from the individual fields and also researchers working interdisciplinarily across fields, the seminar helped to foster a mutual understanding about the challenges of advancing computing beyond current CMOS technology and to create long-term visions about a future hardware/software stack.

Seminar February 5–10, 2017 – http://www.dagstuhl.de/17061

1998 ACM Subject Classification B.7.1 [Integrated Circuits]: Types and Design Styles – Advanced technologies, Algorithms implemented in hardware, Memory technologies, C.1.3 [Processor Architectures]: Other Architecture Styles – Analog computers, Data-flow architectures, Heterogeneous (hybrid) systems, Neural nets, C.3 [Special-purpose And Application-based Systems]: Real-time and embedded systems, Signal processing systems, C.4 [Performance Of Systems]: Design studies, Modeling techniques, D.3.2 [Programming Languages]: Language Classifications – Data-flow languages, Specialized application languages, D.3.4 [Programming Languages]: Processors – Code generation, Compilers, Run-time environments, D.4.7 [Operating Systems]: Organization and Design

Keywords and phrases 3D integration, compilers, emerging post-CMOS circuit materials and technologies, hardware/software co-design, heterogeneous hardware, nanoelectronics, neuromorphic computing, operating systems, programming languages, runtime support, system architecture

Digital Object Identifier 10.4230/DagRep.7.2.1

Wildly Heterogeneous Post-CMOS Technologies Meet Software, *Dagstuhl Reports*, Vol. 7, Issue 02, pp. 1–22
Editors: Jerónimo Castrillón-Mazo, Tei-Wei Kuo, Heike E. Riel, and Matthias Lieber
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Jerónimo Castrillón-Mazo
Tei-Wei Kuo
Heike E. Riel
Sayeef Salahuddin
Matthias Lieber

Topic and Structure

The end of exponential scaling in conventional CMOS technologies has been forecasted for many years by now. While advances in fabrication made it possible to reach limits beyond those predicted, the so anticipated end seems to be imminent today. An indication of this is the research boom, both in academia and industry, in emerging technologies that could complement or even replace CMOS devices. Examples for such emerging technologies include tunnel FETs, nonvolatile memories such as magnetoresistive RAM, 3D integration, carbon nanotube transistors, and graphene.

The main goal of this seminar was to discuss bridges between material research, hardware components and, ultimately, software for information processing systems. Given a new class of wildly heterogeneous systems that integrate different technologies, we want to reason about enabling hardware and software abstractions, from languages and system-software down to hardware mechanisms. The challenge of realizing an efficient wildly heterogeneous system can only be tackled by employing holistic and synergistic approaches in an interdisciplinary environment. By bringing together experts from the individual fields and also researchers working interdisciplinarily across fields, the seminar helped to foster a mutual understanding about the challenges of advancing computing beyond current CMOS technology and to create long-term visions about a future hardware/software stack.

The seminar was structured around four partially overlapping areas, namely: (i) far-fetched materials and physics such as spin, nanomagnets, phase transition, and correlated phenomena, (ii) near future materials (and software) such as phase-change memory, nanowires, nanotubes, and neuromorphic devices, (iii) low-level software layers for new technologies such as operating systems, runtime support, middleware, and HW/SW-co-designed firmware, and (iv) upper software layers such as new programming/specification languages, models, and software synthesis.

Important questions addressed by the seminar included:

- Materials/Devices: What are the current status and the roadmap of post-CMOS materials and technologies? What will be the expected characteristics of the new devices? Will new technologies enable a fundamentally different computing paradigm, e. g., beyond von Neumann? What are the challenges for proper benchmarking of different technologies?
- Hardware/Software Stack: How much of the hardware's heterogeneity and its characteristics should be exposed to programmers? How general may be a programming model/language for future (yet unknown) hardware? How to make software adapt itself to hardware with fluctuating resources? Which new applications can be enabled by emerging materials and technologies and what needs to be done at the software layers to make them viable?
- Analysis: How can we model the interactions across the layers of the hardware/software stack? What kind of formal operational models and analysis methods are needed for evaluating heterogeneous systems? Can system-level analysis of new technologies give insights to material scientists, disrupting the otherwise incremental innovation paradigm?

Main Conclusions

Summary

There will probably be no CMOS replacement for chips with billions of transistors in the next 20 years, but architectural advances at various levels (such as 3D transistors, 3D integration of memory and logic, specialization, and reconfigurability) will lead to performance improvements despite the scaling limitations of planar CMOS technology. New non-volatile memories (e. g., spin-based) bear the potential to radically change various areas of computing, such as data-intensive processing and neuromorphic computing. New hardware architectures will need rethinking today's software stack and our widely used programming models. Finally, even though some post-CMOS technologies will not replace high-end CMOS transistors, there is great potential in new, yet unknown, applications. Applications, backed by a strong commercial demand, will give some technologies the push to become viable. Examples are radio-frequency for carbon nanotubes, graphene based sensors, organic low-cost transistors for wearables, and memristors for neuromorphic computing.

Post-CMOS logic for compute-intensive applications

Currently, there is no alternative to CMOS on the horizon to realize logic for large von Neumann computing, due to lower projected performance and/or yield challenges. Candidates discussed on the seminar have been: tunnel FETs, III-V, 2D materials such as graphene, CNTs, or spintronics. This means that general purpose and high-performance computing will most probably be based on CMOS in the medium term. To workaround the CMOS scaling problem, architectural specialization will gain more and more importance leading to general purpose computing systems with (various) specialized accelerators. We already find them today in, e. g., mobile devices or GPU high-performance computing accelerators. Additionally, reconfigurable logic, such as FPGAs, and application-specific circuits have a high potential for performance gains. However, it is a big challenge to program such heterogeneous systems. Work towards solutions based on dataflow programming, memory access patterns, skeletons, and domain-specific languages have been discussed at the seminar. Additionally, operating system might need to adapt to allow, for example, accelerators to perform system calls.

Emerging memory technologies

In near future, new non-volatile memories will be available that could unify RAM and permanent storage, including MRAM and RRAM. While these could provide huge benefits for memory-intensive applications, the implications on architecture and software stack are not yet clear. For example, what will be the role of the file system in such an architecture? And how to deal with security aspects when every bit in RAM is permanent? Looking further into the future, the spin-based, non-volatile racetrack memory has the potential to compete with SRAM in terms of performance, while consuming considerably less energy. High-performance and energy-efficient non-volatile memories will also be important for neuromorphic devices.

Going 3D

3D integration enables the integration of heterogeneous technologies for logic, memory, communication, and sensing on a single chip. At the transistor level, 3D corrugated transistors were discussed as a promising direction to keep reducing the footprint while avoiding short-channel effects. Advancing today's die stacking technology through fine-grained vias linking the layers, will provide a substantial improvement for latencies and bandwidths in the systems. Bringing memory closer to logic will lower the memory wall (or even lead to a breakdown?). This means that many existing applications could be compute bound (again) and the processor architecture could be potentially simplified by removing the overhead that was added to workaround the memory wall, such as big caches and prefetchers, making place for additional compute units. In this optimistic scenario, general-purpose computing would receive a great (one-time) performance boost. For compilers and applications, we would have to rethink our way of optimizing code.

Computing beyond von Neumann

Architectural approaches beyond von Neumann were also discussed to speedup specific applications. Examples were neuromorphic computers, analog circuits, and dataflow machines. Of course these approaches cannot replace general purpose processors completely. A possible future architecture would combine classical von Neumann processors with non-von Neumann accelerators (on the same chip) to enable mixed programming. The recent industry adoption of machine learning drives the need for neuromorphic computers. While these systems already outperform general purpose processors today, new technologies such as non-volatile memories and analog spintronics promise even greater gains. Promising analog circuits were shown to perform well for concrete NP-complete problems such as SAT and graph coloring. Along these lines, a theoretical framework was introduced that may serve to abstractly compare the asymptotic energy efficiency between the analog and the digital realizations of a system. Finally, dataflow machines where discussed that stream data directly between computational units without the overhead of registers and caches, thereby removing the "Turing tax".

Special applications

Some of the materials considered in the seminar are very likely not able to compete with CMOS for logic, but have strengths in other electronic application areas such as sensors, radio frequency, and displays. Carbon nanotubes and graphene are promising materials for high-frequency antennas required for upcoming wireless communication systems. In the particular case of Graphene, it seems that the initial technological hype has passed, and engineering has taken over to produce new clever devices (e. g., nano-membranes for sensing). Organic electronics are already commercially available in displays and OLEDs. Their distinct features of flexibility, low production cost (printed electronics), and biodegradability could potentially open completely new application areas for logic, but not at comparable speed and efficiency to CMOS. These devices have also been deemed important for bio-compatibility. However, there is a long road ahead for testing and certifying actual devices in living tissue, which is not a trivial task, considering the wealth of molecules being investigated in this domain.

Co-design and design space exploration

Proper hardware/software co-design will be very important to achieve performance gains given the limits of CMOS and the prospective wildly heterogeneous and/or application-specific computing systems. Given a specific application problem, numerous implementation alternatives, from the algorithm down to the hardware architecture and technologies, might be feasible. Tools that help developers navigating the huge design space (e. g., using modeling and benchmarking techniques) and automate an efficient implementation as much as possible are needed. It appears to be that the large part of the software is less flexible than the hardware and much work has to be done to make software future-proof.

2 Table of Contents

Executive Summary

Overview of Talks

3 Overview of Talks

3.1 Stochastic power management in energy harvesting systems

Rehan Ahmed (ETH Zürich, CH)

Joint work of Rehan Ahmed, Bernhard Buchlil, Pratyush Kumar, Lothar Thiele

Ambient energy harvesting has been shown to have significant potential in increasing the lifetime of sensor motes and IoT devices. However, energy harvesting sources are variable in nature, and good prediction/power management strategies need to be designed so that the systems powered by them do not encounter battery depletion. In this work, we present a formal study on optimizing the energy consumption of energy harvesting embedded systems. To deal with the uncertainty inherent in these systems, we have developed a Stochastic Power Management (SPM) scheme, that builds statistical models of harvestable energy based on historical data, and uses these models to design an energy consumption profile. The proposed scheme, maximizes the minimum energy consumption over all time intervals, while giving probabilistic guarantees on not encountering battery depletion. We also present results of experimental evaluation. Through the results, we quantitatively establish that the proposed solution is highly effective at providing a guaranteed minimum service level.

3.2 Neuromorphic computing with Non-Volatile Memories

Stefano Ambrogio (IBM Almaden Center – San Jose, US)

Neuromorphic computing stands as an innovative solution for targeting tasks which are easily solved by the human brain, but that require high computational resources on current Von-Neumann computers. This talk presents a brief overview of the main research branches employing Non-Volatile Memories (NVM) as the synaptic element in neural networks for Machine Learning [1]. This research field has gained an increasing interest in the last years due to the performance opportunities that NVM could potentially provide, outperforming nowadays GPUs and CPUs [1, 2].

First, the talk targets fully connected neural networks with Phase Change Memory, trained with the backpropagation algorithm. After introducing the working principle, recent results and comparison between devices used in analog or binary modes are provided [1, 3]. Then, the talk shows some networks trained with the Spike-Timing-Dependent-Plasticity biological protocol [4, 5], underlining the differences with the backpropagation algorithm and the need for extensive global studies in this field. Finally, the impact of device non-idealities on both backpropagation and STDP networks and algorithms is analyzed and some solutions are provided.

References

1 G. W. Burr et al. *Experimental Demonstration and Tolerancing of a Large-Scale Neural Network (165 000 Synapses) Using Phase-Change Memory as the Synaptic Weight Element.* IEEE Trans. Elec. Dev., 62(11):3498–3507, 2015.

2 G. W. Burr et al. *Large-scale neural networks implemented with non-volatile memory as the synaptic weight element: Comparative performance analysis (accuracy, speed, and power).* In IEDM 2015.

3 S. Yu et al. *Binary neural network with 16 Mb RRAM macro chip for classification and online training.* In IEDM 2016.

4 S. Kim et al. *NVM neuromorphic core with 64k-cell (256-by-256) phase change memory synaptic array with on-chip neuron circuits for continuous in-situ learning.* In IEDM 2015.

5 S. Ambrogio et al. *Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM.* IEEE Trans. Elec. Dev., 63(4):1508–1515, 2016.

3.3 M3: Integrating Arbitrary Compute Units as First-class Citizens

Nils Asmussen (TU Dresden, DE)

Joint work of Nils Asmussen, Marcus Völp, Benedikt Nöthen, Hermann Härtig, Gerhard Fettweis
Main reference N. Asmussen, M. Völp, B. Nöthen, H. Härtig, G. Fettweis, “M3: A Hardware/Operating-System Co-Design to Tame Heterogeneous Manycores”, in Proc. of the 21st Int’l Conf. on Architectural Support for Prog. Lang. and Oper. Systems (ASPLOS 2016), pp. 189–203, ACM, 2016.
URL http://dx.doi.org/10.1145/2872362.2872371

We are currently observing a trend towards more heterogeneous systems in order to meet the desired performance and energy efficiency. For example, DSPs, FPGAs and special purpose accelerators are employed next to general purpose cores. However, current operating systems are relying on processor features such as user/kernel mode and memory management units for protection and access to operating system services. These features are not necessarily available on all compute units (CUs), preventing an integration of arbitrary CUs as first-class citizens.

I will present a hardware/software co-design, consisting of a new hardware component and an operating system based on it. By introducing a common interface for all CUs, arbitrary CUs can be integrated as first-class citizens, where untrusted code can run on all CUs and all CUs can access operating system services such as file systems or network stacks.

3.4 Exploring Performance Portability using Memory-Oriented Programming Models

Tal Ben-Nun (The Hebrew University of Jerusalem, IL)

As heterogeneous computing architectures become ubiquitous, application programming complexity is increasing beyond the skill-set of the average scientist. Thus, it is imperative to devise a unified programming environment that enables efficient utilization of the underlying computational resources, without sacrificing simplicity. The talk will present a programming model that tackles one of the fundamental aspects of computing – memory access – which often recurs as a performance bottleneck in parallel applications. The talk will show that by categorizing algorithm inputs and outputs into access patterns, a wide variety of programs can be automatically optimized for various architectures and partitioned across multiple devices. Using the memory-oriented representation, both processing architectures (e. g., CPU, GPU, FPGA) and memory architectures (e. g., stacked memory, ReRAM) can potentially be utilized to their full extent. The presented memory-oriented programming model currently exhibits state-of-the-art performance on nodes with multiple GPUs and irregular algorithms, facilitating the development of efficient applications on architectures that range from mobile devices to supercomputers.

3.5 A hardware/software stack for emerging systems

Jerónimo Castrillón-Mazo (TU Dresden, DE)

This talk introduces the German cluster of excellence "Center for Advancing Electronics Dresden" (cfaed, http://cfaed.tu-dresden.de), which looks into a set of promising technologies that may augment or replace CMOS. Given a new class of wildly heterogeneous systems that integrate different technologies, the Orchestration sub-project of cfaed aims at devising hardware and software abstractions that would allow programming such complex systems [6]. Abstractions include those typically found in computing systems, ranging from hardware mechanisms up to software engineering approaches. These abstractions are paired with formal modelling for quantitative analysis, aiming at tradeoff analysis in heterogeneous systems.

We discuss general hardware mechanisms for isolation on tile-based systems, exemplarily demonstrated in the Tomahawk multicore platform [1]. We argue that tile-based systems offer a well-suited architectural template for integrating components implemented in different technologies. At the hardware level, components must only agree on the interfacing to the on-chip network via routers that provide isolation at the hardware level. We describe the M3 capability-based operating system (OS) [2] which builds on this hardware interface. With a micro-kernel approach, M3 provides access to system resources for tiles that cannot run a full-fledged OS (see abstract 3.3 by Nils Asmussen). As programming abstraction, we use dataflow programming models, architecture models and compilers to automatically generate low-level code for heterogeneous multi-cores [4]. Finally, for formal trade-off analysis, we have developed new theory to handle multiple objective functions and resolve nondeterministic choices in an optimal way [3].

In this talk, we report on early results of deploying the abstractions on a heterogeneous CMOS platform and an effort to bring up a system simulator that allows integrating models of components on Post-CMOS technologies. We briefly discuss promising architectural options with reconfigurable 1D transistors (e. g., with silicon nano-wires [5]). Finally, we share our experience when trying to bridge the broad interdisciplinary gap between material and computer scientists.

References

1 O. Arnold et al. *Tomahawk: Parallelism and heterogeneity in communications signal processing MPSoCs.* ACM Transactions on Embedded Computing Systems, 13(3s):107, 2014.

2 N. Asmussen et al. *M3: A hardware/operating-system co-design to tame heterogeneous manycores.* In ASPLOS 2016.

3 C. Baier et al. *Probabilistic model checking and non-standard multi-objective reasoning.* In FASE 2014.

4 J. Castrillon and R. Leupers. *Programming Heterogeneous MPSoCs: Tool Flows to Close the Software Productivity Gap.* Springer, 2014.

5 M. Raitza et al. *Exploiting transistor-level reconfiguration to optimize combinational circuits.* In DATE 2017.
6 M. Völp et al. *The Orchestration Stack: The Impossible Task of Designing Software for Unknown Future Post-CMOS Hardware.* In PMES at SC 2016.

3.6 Random Thoughts/Examples about Neuromorphic Computing and Emerging Devices

Yiran Chen (Duke University – Durham, US)

Joint work of Yiran Chen, Hai (Helen) Li, Qing Wu, Wei Wen, Beiye Liu, Chaofei Yang, Miao Hu

Human brain is the most sophisticated organ that nature ever builds. Building a machine that can function like a human brain, indubitably, is the ultimate dream of a computer architect. Although we have not yet fully understood the working mechanism of human brains, the part that we have learned in past seventy years already guided us to many remarkable successes in computing applications, e. g., artificial neural network and machine learning. The recently emerged research on "neuromorphic computing", which stands for hardware acceleration of brain-inspired computing, has become one of the most active areas in computer engineering. Our presentation starts with a background introduction of neuromorphic computing, followed by some design examples of hardware acceleration schemes of learning and neural network algorithms on IBM TrueNorth and memristor-based computing engines. At the end, we will share our prospects on the future technology challenges and advances of neuromorphic computing.

References

1 W. Wen et al. *A new learning method for inference accuracy, core occupation, and performance co-optimization on TrueNorth chip.* In DAC 2016.
2 M. Hu et al. *Memristor Crossbar-Based Neuromorphic Computing System: A Case Study.* IEEE Trans. Neural Netw. Learning Syst., 25(10):1864–1878, 2014.
3 C. Yang et al. *Security of neuromorphic computing: thwarting learning attacks using memristor's obsolescence effect.* In ICCAD 2016.

3.7 Bridging the gap between single device fabrication and system design for emerging device technologies

Martin Claus (TU Dresden, DE)

Main reference M. Schröter, M. Claus, P. Sakalas, M. Haferlach, D. Wang, "Carbon Nanotube FET Technology for Radio-Frequency Electronics: State-of-the-Art Overview", IEEE J-EDS, 1(1):9–20, 2013.
URL http://dx.doi.org/10.1109/JEDS.2013.2244641

Research on emerging electronics is in many cases restricted to single-device fabrication as a proof of concept study for specific two-dimensional (e. g. MoS2) and one-dimensional materials (e. g. CNTs). Depending on the device architecture, these devices capture the functionality (switching and amplification) [1] of incumbent Silicon-based transistors or add functionality such as reconfigurability at the transistor level [2].

However, for evaluating the performance of these materials and new device functionality, circuit and system design studies comprising hundreds or even millions of devices are essential. Since the fabrication of these systems is far beyond the technological possibilities for most emerging technologies, the circuit and system evaluation relies on simulations. Due to the inherent complexity of emerging devices, holistic multi-scale simulations are required [3, 4, 5].

The talk will focus on one-dimensional materials and devices for high-performance computing as well as reconfigurable systems. The significance of holistic multi-scale simulations for technology development as well as circuit and system design based on physics-based compact modeling will be highlighted.

References

1 M. Schröter et al. *Carbon Nanotube FET Technology for Radio-Frequency Electronics: State-of-the-Art Overview*, IEEE Journal of the Electron Devices Society, 1(1):9–20, 2013.

2 G. Darbandy, M. Claus and M. Schröter. *High-Performance Reconfigurable Si Nanowire Field-Effect Transistor Based on Simplified Device Design*, IEEE Transactions on Nanotechnology, 15(2):289–294, 2016.

3 M. Claus et al. *COOS – A wave-function based Schrödinger-Poisson solver for ballistic nanotube transistors*, Journal of Computational Electronics, 13:(2):689–700, 2014.

4 S. Mothes, M. Claus and M. Schröter. *Toward Linearity in Schottky Barrier CNTFETs*, IEEE Transactions on Nanotechnology, 14(2):372–378, 2015.

5 M. Schröter et al. *A Semiphysical Large-Signal Compact Carbon Nanotube FET Model for Analog RF Applications*, IEEE Transactions on Electron Devices, 62(1):52–60, 2015.

3.8 System-Level Design Optimization for Integration with Silicon Photonics

Ayse Coskun (Boston University, US)

System-level design tools and optimization methods are essential for enabling computer engineers experiment with emerging technologies. Similarly, a system-level view is necessary for researchers working with new technologies to work with constraints imposed by different applications or architectures. This talk discusses a cross-layer methodology for designing power-efficient many-core systems with on-chip silicon photonic networks. The proposed methodology enables optimizing the layout [1] or the runtime operation [2] of a target system to reduce the power overhead and/or guardbanding associated with silicon photonics integration on chip.

Through this specific example of integration with silicon photonics, another aim of the talk is to demonstrate a way for enabling early integration of emerging technologies into system design, including when using 2.5D/3D stacking to integrate (broadly) heterogeneous technologies together. The talk also discusses various open design automation and tooling challenges in designing systems with emerging technologies and in heterogeneous system design and runtime management.

References

1 A. K. Coskun et al. *Cross-layer Floorplan Optimization for Silicon Photonic NoCs in Many-core Systems.* In DATE 2016.

2 J. Abellan et al. *Adaptive Tuning of Photonic Devices in a Photonic NoC Through Dynamic Workload Allocation.* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 36(5):801–814, 2017.

3.9 Matching computer science tools and new technology

Erik P. DeBenedictis (Sandia National Labs – Albuquerque, US)

New devices provide new computational capabilities, yet we will need new computer science tools to exploit the capabilities in algorithms. The talk will highlight two examples, but others can be discussed informally.

The first example is the inability of computational complexity theory to capture the advantage of some new devices. For example, RRAM arrays used in neural networks can have non-unity (energy) cost for scalar multiplication, including unexpectedly low energy. Essentially, the energy of a multiply depends on how the result is used later on. However, the straightforward interpretation of computational complexity theory assumes unit cost for arithmetic (of a given precision). I claim this is why computer science community assumes physicists are in error when they make certain exotic claims about devices. The misunderstanding then blocks development of algorithms using the new devices to best advantage. The proposed resolution is to use a complexity measure for algorithms based on minimum energy in units of kT. I'll present examples of the problem and resolution.

The second issue is the bias towards the von Neumann architecture in computer architecture tools. The HPC computer architecture community uses an iterative process called "codesign" in an attempt to improve architectures for the "post Moore's Law era". This means simulating proposed new architectures against frequently used algorithms or instruction traces, iteratively modifying the architecture to get better performance or energy efficiency. Due to artifacts of the von Neumann architecture in the simulation inputs, if somebody applies codesign to a new non-von Neumann architecture, the feedback process will very quickly restore the architecture to the von Neumann model. I claim this is why we are overwhelmed with minor variants of the von Neumann architecture while not having effective ways to exploit new physics. The remedy is to replace codesign with feedback loop that does not include artifacts of the von Neumann architecture. I will give examples.

3.10 AnyDSL: Building Domain-Specific Languages for Productivity and Performance

Sebastian Hack (Universität des Saarlandes, DE) and Roland Leißa (Universität des Saarlandes, DE)

Joint work of Sebastian Hack, Klaas Boesche, Roland Leißa, Philipp Slusallek
Main reference S. Hack, K. Boesche, R. Leißa, R. Membarth, P. Slusallek, "Shallow embedding of DSLs via online partial evaluation", in Proc. of the 2015 ACM SIGPLAN Int'l Conf. on Generative Prog.: Concepts and Experiences (GPCE 2015), pp. 11–20, ACM, 2015.
URL http://dx.doi.org/10.1145/2814204.2814208

To achieve good performance, programmers have to carefully tune their application for the target architecture. Optimizing compilers fail to produce the "optimal" code because their hardware models are too coarse-grained. Even more, many important compiler optimizations are computationally hard even for simple cost models. It is unlikely that compilers will ever be able to produce high-performance code automatically for today's and future machines.

Therefore, programmers often optimize their code manually. While manual optimization is often successful in achieving good performance, it is cumbersome, error-prone, and unportable. Creating and debugging dozens of variants of the same original code for different target platform is just an engineering nightmare.

An appealing solution to this problem are domain-specific languages (DSLs). A DSL offers language constructs that can express the abstractions used in the particular application domain. This way, programmers can write their code productively, on a high level of abstraction. Very often, DSL programs look similar to textbook algorithms. Domain and machine experts then provide efficient implementations of these abstractions. This way, DSLs enable the programmer to productively write portable and maintainable code that can be compiled to efficient implementations. However, writing a compiler for a DSL is a huge effort that people are often not willing to make. Therefore, DSLs are often embedded into existing languages to save some of the effort of writing a compiler.

In this talk, I will present the AnyDSL framework we have developed over the last three years. AnyDSL provides the core language Impala that can serve as a starting point for almost "any" DSL. New DSL constructs can be embedded into Impala in a shallow way, that is just by implementing the functionality as a (potentially higher-order) function. AnyDSL uses online partial evaluation remove the overhead of the embedding.

To demonstrate the effectiveness of our approach, we generated code from generic, high-level text-book image-processing algorithms that has, on each and every hardware platform tested (Nvidia/AMD/Intel GPUs, SIMD CPUs), beaten the industry standard benchmark (OpenCV) by 10-35 %, a standard that has been carefully hand-optimized for each architecture over many years. Furthermore, the implementation in Impala has one order of magnitude less lines of code than a corresponding hand-tuned expert code. We also obtained similar first results in other domains.

3.11 Design Space Exploration: Getting the Most out of Accelerators

Xiaobo Sharon Hu (University of Notre Dame, US)

For many applications, it is a non-trivial task to actually achieve the high performance and energy efficiency promised by heterogeneous platforms. Introduction of a variety of programmable (such as GPU) or trainable/tunable (such as neural network and constrained optimization solver based) accelerators further exacerbate the problem. One reason is the lack of reliable prediction of the system's performance/energy before application implementation. Another reason is that a heterogeneous platform presents a large design space for workload partitioning among different processing units. Yet another reason is the complicated data usage patterns occurring in many applications.

This talk uses a medical image analysis application as a motivational example to show how different types of accelerators (particularly fully convolutional neural networks and Boolean satisfiability solver) can be employed to solve the problem efficiently and the challenges faced by the design exploration effort. I then present our effort in developing a framework to assist application developers to identify workload partitions that have high potential leading to high performance or energy efficiency for CPU+GPU system *before actual implementation.* The framework can further be used to estimate the performance or energy of given workload partitions. I end the talk with some insights on how such a framework together with our benchmarking approach may be leveraged to help explore the design space of heterogeneous systems with neural network and SAT solver based accelerators.

3.12 Architecture and software for when there's no longer plenty of room at the bottom

Paul H. J. Kelly (Imperial College London, GB)

Joint work of Paul H. J. Kelly, David Ham, Fabio Luporini, Lawrence Mitchell, Mike Giles, Gihan Mudalige, Istvan Reguly, Doru Bercea, Graham Markall, Florian Rathgeber, Maciej Piechotka
Main reference F. Rathgeber et al., "Firedrake: Automating the Finite Element Method by Composing Abstractions", ACM Transactions on Mathematical Software (TOMS), 43(3):24, ACM, 2017.
URL https://doi.org/10.1145/2998441

In 1959, Richard Feynmann wrote his prescient article "Plenty of room at the bottom", demonstrating just how far contemporary computers were from fundamental physical limits. The 58 years of exponential progress since then have brought us much closer to such limits, and there is much debate about where they really lie. What is clear is that we're a lot closer. We are confronted more and more with fundamental physical concerns, particularly with regard to the communication latency, bandwidth and energy. This talk offered a reflection on how this impacts how we think about algorithms and how we design high-performance software. Along the way I discussed the "Turing Tax" – the price we pay for running a programs on a universal, general-purpose machine. I also sketched some of the experience from our lab on delivering software tools that help abstract locality, expose the algorithmic-level design space, and enable tight control over data movement even in code based on irregular data such as unstructured meshes.

3.13 Towards Future-Proof (Parallel?) Programming Models

Christoph W. Kessler (Linköping University, SE)

Joint work of Usman Dastgeer, Johan Enmyren, August Ernstsson, Lu Li, Christoph Kessler
Main reference A. Ernstsson, L. Li, C. Kessler, "SkePU 2: Flexible and type-safe skeleton programming for heterogeneous parallel systems", Int. J. Parallel. Prog., pp. 1–19, Springer, 2017.
URL http://dx.doi.org/10.1007/s10766-017-0490-5

In the coming years, Moore's Law is expected to slow down and current CMOS-based hardware technology to eventually hit physical limits. However, there is, in spite of many exciting new developments, no disruptive replacement technology ready yet to take over after CMOS in a short-term range. Instead, it is likely that the already existing trends towards more parallel/distributed, less coherent, more heterogeneous, less fault-tolerant and more reconfigurable architectures will accelerate further in the coming years. Also, as hardware performance growth declines, an increasingly large share of performance boost must come from improvements in the software, e. g. by more adaptive algorithms and data structures, and more powerful optimizing compiler and runtime system techniques. At the same time, we have to care about portability and programmer productivity to sustain a scalable software market.

In this talk we consider two architecture-independent, high-level (parallel) programming models that can be effectively mapped to today's already quite diverse computer architectures – and hopefully also to coming generations of computing technology:

(1) Skeleton Programming – high-level, customizable general-purpose or domain-specific program building blocks representing frequently occurring patterns of control and data flow, exposing a sequential-looking, compositional programming interface, with adaptive implementations for a broad range of parallel, distributed and heterogeneous target systems. As an example, we briefly review SkePU [1, 2].

(2) Coarse-Grain Dataflow Programming – expressing a computation as a graph of tasks or actors with explicit dependences, which can be configured and mapped statically or dynamically to the resources of a parallel, distributed and heterogeneous target system.

We survey these two complementary approaches, with some techniques in compiler and runtime support for today's architectures, and motivate why they may still be useful with tomorrow's (mostly unknown) computer hardware and can thus lead to more future-proof software.

References

1 C. Kessler et al. *SkePU: Autotunable Multi-Backend Skeleton Programming Framework for Multicore CPU and Multi-GPU Systems.* https://www.ida.liu.se/labs/pelab/skepu/, 2016.

2 A. Ernstsson, L. Li, C. Kessler. *SkePU 2: Flexible and type-safe skeleton programming for heterogeneous parallel systems.* Int. J. Parallel. Prog., 2017.

3.14 Optimization through Hardware and Software Co-Designs

Tei-Wei Kuo (National Taiwan University – Taipei, TW)

This presentation is inspired by the changing of landform by new memory technology, huge driving forces of various applications and heterogeneous computing. There are even more opportunities for system optimization right now than ever. Such optimization opportunities also unsurprisingly exist from the application layer all the way to the system and hardware layers. Excellent examples are software-controlled cache and smart storage devices. In the past decades, we have been experiencing huge impacts due to storage innovation (with a good example on solid-state disks). Some emerging non-volatile memory is now bringing innovation to traditional memory management. Because of that, we soon see the blurring of system boundaries in the memory architecture. With those in mind, challenges and opportunities are coming.

3.15 Future challenges and opportunities for adaptive HPC applications

Matthias Lieber (TU Dresden, DE)

Due to the limits of CMOS technology, high performance computers have experienced a large growth in complexity over the last 15 years: concurrency is exploding (millions of cores in the top systems today), heterogeneous architectures that include accelerators are common, reliability has become a major concern, and performance variability of CPUs can be observed. On the other side, the scientific and engineering applications that run on these systems are also becoming more complex and adaptive, often leading to workload variations over runtime. Dynamic load balancing is used to redistribute dynamic workloads to reduce wasted time in (necessary) synchronization points. Load balancing on such highly parallel computers with heterogeneous compute resources in each node is challenging and trade-offs have to be made regarding workload balance, communication optimization, migration reduction, and the actual costs for making such load balancing decisions. Solutions that focus on some aspects of the problem have been demonstrated, for example very fast load balancing methods can be implemented with space-filling curves and task-based programming models (such as PaRSEC and StarPU) enable load balancing in heterogeneous computing environments. The limits of CMOS technology and possible solutions that have been discussed during this seminar lead to future challenges but also opportunities for high performance computing. Regarding logic, there is not yet a clear successor for CMOS on the horizon. That means that in near future performance gains can only be achieved through architectural improvements and specialization, such as FPGAs and ASICs. However, the increasing heterogeneity will complicate programming as well as runtime workload management. Regarding memory, non-volatile memories will very likely enable improved fault-tolerance mechanisms and accelerate data-intensive applications. 3D stacking of (several layers of memory) on top of logic, as already available in some accelerators, will lead to improved memory performance, potentially reducing the gap between memory and compute performance.

3.16 Neural network-based accelerators: do device, circuits, architectures, or algorithms provide the best “bang for our buck”?

Michael Niemier (University of Notre Dame, US)

Joint work of Michael Niemier, X. Sharon Hu

Researchers are increasingly looking for new ways to exploit the unique characteristics of emerging devices (e. g., non-volatility) as well as architectural paradigms that transcend the energy efficiency wall. In this regard, there has been great interest in hardware implementations of various types of neural networks. One obvious example is IBM’s TrueNorth chip, which realizes a configurable, spiking neural network (SNN) in hardware [1]. Additionally, cellular neural networks (CeNNs) are now under investigation via the Semiconductor Research Corporation’s benchmarking activities [2, 3] as (i) they can solve a broad set of problems [4] (e. g,. image processing, associative memories, etc.), and (ii) can exploit the unique properties of both spin- and charge- based devices [5, 6]. However, in all cases, we must consider what application spaces/problem sets/computational models ultimately benefit from hardware realizations of neural networks, and if hardware implementations can ultimately outperform alternative architectures/models for the same problem.

This talk will discuss strategies for quantitatively assessing neural network co-processors. To facilitate discussion, as a representative case study, we will consider work with CeNNs. More specifically, we will discuss (i) algorithm development where processing tasks are mapped to CeNNs or more conventional CPUs/GPUs (e. g., for image recognition, CeNNs can be highly efficient for feature extraction tasks given the architecture’s parallel nature; for more mathematical operations, CPUs may be more efficient.) (ii) Next, given the analog nature of a CeNN, and the inherent nature of inference applications, algorithmic accuracy must be evaluated at multiple levels (e. g., we must address overall algorithmic quality, and any impact on algorithmic quality due to lower precision hardware.) (iii) Algorithms must then be mapped to a suitable hardware architecture (e. g., parallel CeNNs vs. a CeNN that is used serially). (iv) Finally, we must compare energy, delay, and accuracy projections to the best von Neumann algorithm for the same application-level problem. Using targeting tracking as a case study, we will discuss (a) strategies for algorithmic refinement, and (b) where we derive the most substantial benefits for metrics of interest (energy and delay) – i. e., from devices, circuits, architectures, and/or algorithms.

We will also present preliminary results as to how CeNNs can be leveraged to accelerate convolutional neural networks. As a case study, we present preliminary data for the MNIST digit recognition problem. We will compare/contrast our projections to other architectures and algorithms – e. g., the DropConnect algorithm [7] (with power profiling done on an Intel i5 processor; devices have similar feature sizes to those used in CeNN simulations), IBM True North [8], etc.

References

1 P. A. Merolla et al. *A million spiking-neuron integrated circuit with a scalable communication network and interface.* Science, 345(6197):668–673, 2014.
2 SRC benchmarking center, 2016. https://www.src.org/program/nri/benchmarking/
3 SRC-DARPA STARnet initiative, 2016. https://www.src.org/program/starnet/
4 L. O. Chua and T. Roska. *Cellular Neural Networks and Visual Computing: Foundations and Applications.* Cambridge University Press, 2002.

5 C. Pan and A. Naeemi. *A proposal for energy-efficient cellular neural network based on spintronic devices.* IEEE Transactions on Nanotechnology, 15(5):820–827, 2016.

6 B. Sedighi et al. *A CNN-inspired mixed signal processor based on tunnel transistors.* In DATE 2015.

7 S. Esser et al. *Backpropagation for energy-efficient neuromorphic computing.* In NIPS 2015.

8 L. Wan et al. *Regularization of Neural Networks using DropConnect.* In ICML 2013.

3.17 Scenario-based, System-level Embedded Systems Design

Andy D. Pimentel (University of Amsterdam, NL)

Main reference A. D. Pimentel, "Exploring Exploration: A Tutorial Introduction to Embedded Systems Design Space Exploration", IEEE Design & Test, 34(1):77–90, 2017.
URL http://dx.doi.org/10.1109/MDAT.2016.2626445

Modern embedded systems are becoming increasingly multifunctional and, as a consequence, they more and more have to deal with dynamic application workloads. This dynamism manifests itself in the presence of multiple applications that can simultaneously execute and contend for resources in a single embedded system as well as the dynamic behavior within applications themselves. Such dynamic behavior in application workloads must be taken into account during the design of multiprocessor system-on-a-chip (MPSoC)-based embedded systems. In this talk, I will present the concept of application workload scenarios to capture application dynamism and explain how these scenarios can be used for searching for optimal mappings of a multi-application workload onto an MPSoC. To this end, the talk will address techniques for both design-time mapping exploration as well as run-time mapping of applications.

3.18 Organic electronics: devices for the electronic gadgets age

Sebastian Reineke (TU Dresden, DE)

Organic electronics is a rapidly growing research field that currently develops in the shadow of existing CMOS technology. Ultimately, it targets for applications beyond standard electronics we know today. Success has been achieved already with organic light-emitting diodes (OLEDs), which we find today in most of our mobile displays, and organic solar cells. However, those technologies may soon become commodities, where organic electronics will touch new ground with its potential to deliver scalable, low-cost, customizable (in form and function), and disposable devices. Only future will tell, in which sectors organic electronics will be most successful, but definitely, the route to success proves to be bumpy due to the lack of concerted developments. In this talk, I will give a brief introduction to the conventional approach in organic electronics using the example of OLEDs. Here, I will summarize the general research challenges, give details to some recent device concepts, and assemble a collection of potential scenarios of use.

The main conclusion of this presentation and the discussion related to it is the fact that organic electronics is a very front-end rich technology platform. With attributes like flexible, ultra-lightweight, plastic etc., organic electronics will open up new application scenarios, where the next big challenge is the seamless integration of back-end electronics components made of organic devices that are needed to run such novel front-end applications. Only with the knowledge of the future systems, questions of software development can be addressed in a meaningful fashion. Here, the big question is how diverse the future organic electronics will look like.

3.19 Towards Next Generation of Computing

Heike E. Riel (IBM Research Zurich, CH)

In the past 50 years computing was driven by "smaller & denser" resulting in "faster & cheaper". Cost per function has decreased tremendously, while system performance and reliability have been improved significantly. Dimension scaling alone is no longer sufficient and various paths are pursued in order to increase system performance. In order to further extend core logic and memory technology roadmaps by miniaturization significant innovation in materials, devices and architectures is required. Key technologies which are investigated to continue the roadmap are, e. g., gate-all-around nanowire technologies, III-V semiconducting nanowires for high-mobility field-effect transistors (FETs), III-V nanowires heterostructure tunnel FETs as steep slope devices or carbon nanotube field-effect transistors. In parallel other technologies to build new architectures such as heterogeneous integration, 3D packaging, system-on-chip, silicon photonics and others are pushed to increase system level performance. Yet despite all of these innovative technologies, increasing the density of transistors will cease when length-scales reach atomic dimensions. This raises the fundamental question of what is next? What is the future of information technology beyond scaling and traditional computing? In that regard completely new computing paradigms are developed such as quantum computing and neuromorphic technologies.

3.20 Making better transistors: beyond yet another new materials system

Mark Rodwell (University of California – Santa Barbara, US)

In ~55 years of IC development, industry has concentrated on making switches smaller; universities have mostly concentrated on the complementary role of making them from new materials. Thus has university electron device research progressed from silicon to germanium and SiGe, the arsenide, phosphide and then antimonide III-V's, then carbon nanotubes, and today 2D semiconductors. Beyond SiGe, there seems but little hope that these more recent materials might benefit transistors in computer ICs.

Perhaps we should focus instead on improving their shape? Corrugating a FET channel, in the style of a folded piece of paper, produces a device with transport distance much larger than its footprint; we can use this to improve electrostatics and suppress source-drain tunneling currents in few-nm-footprint transistors. Corrugating the channel in the perpendicular direction increases the drive current per unit IC area, which we then might trade for lower-voltage, lower-power operation. With FETs, making low-resistance yet small contacts is as much a problem as is making short gates: should we corrugate the metal-semiconductor interface to reduce the interface resistance?

Or, should we change their band structure? In tunnel FETs, we can add several heterojunctions, and so increase the desired on-state tunneling currents while decreasing the unwanted off-state leakage currents. Yet, I can offer nothing beyond this single example, albeit one that I presently find of great interest.

Finally, perhaps we might change their function? One focus of this workshop is to explore the merging of logic and memory. We might do this at either within the transistor or within low-level logic design. It is not yet clear to this circuit designer that a logic-plus-memory transistor would be markedly more useful than, for example, simple merged logic using pass transistors and gate capacitances. I will do my best to examine their potential utility.

3.21 Logic Synthesis for Post-CMOS Technologies

Eleonora Testa (EPFL – Lausanne, CH), Giovanni De Micheli (EPFL – Lausanne, CH), and Mathias Soeken (EPFL – Lausanne, CH)

Traditionally, logic synthesis tools have concentrated on the optimisation of circuits based on CMOS logic primitives such as AND and OR. Since recently many emerging nanotechnologies are based on logic models different from standard CMOS, new logic synthesis approaches need to be considered. Most of the promising nanodevices, such as Resistive Random Access Memories (RRAMs) and Spin Wave Devices (SWDs), are based on majority logic and are characterised by nontrivial technological constraints. Both aspects are fundamental when designing new logic synthesis tools. In this talk, we present how many emerging technologies can benefit from a majority-based logic synthesis approach [1, 2]. We will concentrate on a new data structure that provides the necessary abstraction for Boolean functions optimization and manipulation [3]. Further, we will illustrate how SAT-based methods can be used to address the technological constraints.

References

1 E. Testa et al. *Inversion optimization in majority-inverter graphs.* In NANOARCH 2016.

2 O. Zografos et al. *Design and benchmarking of hybrid CMOS-spin wave device circuits compared to 10nm CMOS.* In IEEE-NANO 2015.

3 L. Amarù et al. *Majority-inverter graph: a novel data-structure and algorithms for efficient logic optimisation.* In DAC 2014.

3.22 Architectural Requirements for Intransitive Trust and Fault and Intrusion Tolerance

Marcus Völp (University of Luxembourg, LU)

Looking at our software stack today, we are facing a functionality / code size dilemma: functionality comes with a certain amount of code and the more code, the more vulnerabilities and possibilities for attackers to compromise the computer systems we all depend on. In this talk, I review designs for intransitive trust relationships, which allow critical applications to use functionality without trusting all the code that provides this functionality. Common patterns for intransitive trust involve ciphers to protect data integrity and confidentiality. Another involves replication and voting to hide Byzantine behavior of a minority of compromised replicas behind consensus of a healthy majority. I derive architectural implications and raise as questions how the strong isolation assumptions of intransitive trust design patterns can be realized in todays and upcoming wildly-heterogeneous systems.

Participants

- Rehan Ahmed
ETH Zürich, CH
- Stefano Ambrogio
IBM Almaden Center – San Jose, US
- Nils Asmussen
TU Dresden, DE
- Tal Ben-Nun
The Hebrew University of Jerusalem, IL
- Jerónimo Castrillón-Mazo
TU Dresden, DE
- Jian-Jia Chen
TU Dortmund, DE
- Yiran Chen
Duke University – Durham, US
- Martin Claus
TU Dresden, DE
- Ayse Coskun
Boston University, US
- Suman Datta
University of Notre Dame, US
- Erik P. DeBenedictis
Sandia National Labs – Albuquerque, US
- Gianluca Fiori
University of Pisa, IT
- Shunsuke Fukami
Tohoku University, JP
- Balazs Gerofi
RIKEN – Tokyo, JP
- Sebastian Hack
Universität des Saarlandes, DE
- Xiaobo Sharon Hu
University of Notre Dame, US
- Paul H. J. Kelly
Imperial College London, GB
- Christoph W. Kessler
Linköping University, SE
- Tei-Wei Kuo
National Taiwan University – Taipei, TW
- Roland Leißa
Universität des Saarlandes, DE
- Max Lemme
Universität Siegen, DE
- Matthias Lieber
TU Dresden, DE
- Michael Niemier
University of Notre Dame, US
- Stuart Parkin
MPI – Halle, DE
- Andy D. Pimentel
University of Amsterdam, NL
- Sebastian Reineke
TU Dresden, DE
- Heike E. Riel
IBM Research Zurich, CH
- Mark Rodwell
University of California – Santa Barbara, US
- Eleonora Testa
EPFL – Lausanne, CH
- Marcus Völp
University of Luxembourg, LU