Research Statement

Michael K. Papamichael

My research interests lie in the broad area of computer architecture with emphasis on hardware acceleration, reconfigurable computing, and on-chip interconnects. I deeply enjoy bringing my architecture ideas to life and enabling others to also do so. To this end, I have worked extensively on tools and methodologies that facilitate rapid hardware design and exploration. An already matured effort that captures well my research interests and approach is the CONNECT Network-on-Chip (NoC) framework, which offers a scalable composable NoC architecture and the associated tools to deliver fully working NoC designs that others can use. The publicly available CONNECT NoC RTL generator (http://bit.ly/connect_noc_gen) is actively used in multiple projects and by many researchers around the world, who have collectively generated thousands of networks and published several papers using CONNECT-generated NoCs.

The Repercussions of Disruptive Changes in the Semiconductor Industry

We are currently witnessing fundamental and disruptive changes in the semiconductor industry. On the one hand, our inability to further scale supply voltage is leading to the breakdown of classical CMOS scaling as described by Dennard [1]. As a result, after decades of continuous growth, transistor efficiency, which has been the primary driving force behind performance improvements in general-purpose computing, has now reached a stalemate [2, 3]. On the other hand, still rapidly growing transistor counts driven by Moore’s Law, coupled with recent technology advances and trends, such as die stacking and the increased presence of reconfigurable logic, are enabling the development of massive, diverse Systems-on-Chip (SoCs) with unique and demanding communication requirements.

The Need to Facilitate Hardware Specialization. In an effort to continue increasing performance in the power-constrained setting of the post-Dennard era, there is a growing interest in hardware specialization [3, 4]. While hardware specialization is key to power-efficient computing, as my extensive experience with hardware prototyping has also made me realize first-hand, hardware development today is notoriously hard. Hardware design has not fundamentally changed since the introduction of modern Hardware Description Languages (HDLs) in the 1980s and the proliferation of IP-based methodologies and Application Specific Integrated Circuit (ASIC)-based design more than two decades ago. In fact, in the majority of cases hardware design is still a very primitive low-level process, which essentially pertains to transcribing schematics or specification documents to a structural register-transfer-level representation. As a result, even though there is great demand for mapping applications to specialized hardware to achieve higher power efficiency, hardware development today is not only limited to experts, but takes more time, requires larger design teams and is more expensive than ever. My research aims at lowering the barrier-to-entry for building hardware and allowing application-experts to more easily and efficiently realize their ideas in hardware.

The Need to Rethink Interconnects for Future SoCs. Modern SoCs that host hundreds or even thousands of communicating modules and make use of modern fabrication technologies, such as 3D stacking, are posing major challenges and create unique opportunities for interconnect design. At the same time, the increased use of reconfigurable logic, whether in the form of high-capacity Field Programmable Gate Array (FPGA) devices or reconfigurable fabric hosted within larger SoCs, is blurring the lines between “hard” components that are fixed at fabrication and “soft” components that can be repurposed dynamically at run-time. My research aims at rethinking aging ASIC-driven Network-on-Chip (NoC) architectures to develop a systematic and flexible NoC infrastructure that can 1) properly map to and take advantage of modern underlying fabrication and packaging technologies, such as die stacking, and 2) exploit reconfigurability to better support and adapt to application-specific communication needs.

Recent and Current Research

My PhD research is focused on Pandora and CONNECT. Pandora [9, 10] is a novel knowledge-encapsulating IP1 design paradigm that embodies a set of principles aimed at lowering the barrier-to-entry for building hardware accelerators and allowing application-experts to more easily and efficiently realize their ideas in hardware. CONNECT [5, 8] is a Network-on-Chip (NoC) architecture and RTL generation framework (http://bit.ly/connect_noc_gen) that serves as a demonstration vehicle for the Pandora principles. In addition to these

1 An IP or IP block refers to a pre-made, pre-validated, reusable hardware module that implements a specific function (e.g., CPU, Network-on-Chip, etc.).
projects, while at Carnegie Mellon, I have also worked on multiple aspects of the CoRAM FPGA memory abstraction [6, 7], acceleration of architectural simulations using FPGAs [11, 12, 13], and multi-core memory scheduling [14, 15].

**Pandora.** Pandora pertains to a novel IP design paradigm that aims at reining in the growing complexity of modern highly-parameterized IP generators. To achieve this goal, in Pandora, IP blocks not only capture the microarchitectural and structural view of a design but also encapsulate additional dimensions of knowledge that the IP author has to offer, which can come in the form of: (1) detailed IP design space characterization to help the user understand the effects of parameter choices with respect to hardware implementation and IP-specific metrics, (2) application-level goal-oriented parameterization that is meaningful to the IP user and automatically sets low-level structural parameters to achieve the desired design optimizations, and (3) purpose-built domain-aware simulation-time and runtime monitoring mechanisms to assist functional and performance debugging. In addition to reducing complexity and boosting productivity, the Pandora approach also dramatically reduces the combined total effort, because work that would potentially otherwise be repeated by each IP user, is now only performed once and can be leveraged by others.

For the purposes of Pandora, I have: (1) developed DELPHI [9] for performing fast and efficient IP characterization (power, area, frequency) across multiple technology nodes, (2) developed and evaluated Nautilus [10], an IP optimization engine, which uses modified genetic algorithms that incorporate IP author knowledge to perform automated guided design space search, (3) developed a set of flexible reusable simulation-time and run-time instrumentation components to aid in hardware functional and performance debugging (scheduled for public release in early 2015), and (4) extended the CONNECT Network-on-Chip generator to demonstrate many of the Pandora design principles, such as application-specific goal-oriented NoC optimization [7].

In DELPHI [10] I led a collaborative effort with students and faculty from MIT and UC Berkeley to develop a flexible open framework that leverages the DSENT modeling engine (http://sites.google.com/site/mitdSENT/) for fast, easy and efficient characterization of RTL hardware designs. In the context of Pandora, DELPHI can be used to accelerate the IP characterization process and rapidly map an IP’s design space with respect to implementation characteristics or other IP-specific metrics of interest. The DELPHI flow processes Verilog or VHDL RTL hardware descriptions to generate a technology-independent DSENT design model, which can then be used to perform very fast—one to two orders of magnitude faster than full RTL synthesis—estimation of hardware performance characteristics, such as frequency, area, and power. The full DELPHI framework (including a fully open-source flow) will soon be available for download or for use through a web-based portal.

As a next step towards realizing the Pandora vision, I also developed Nautilus [11], which builds on top of the DELPHI characterization engine to help IP users perform parameter optimization and navigate an IP’s design space in a fast and automated manner. At the core of Nautilus is a modified genetic algorithm I developed, that allows embedding of IP-author knowledge pertaining to the IP design space. This knowledge, coming in the form of “hints”, captures the IP author’s intuition about how IP parameters relate to the various metrics of interest; the goal is to help steer the optimization search process more quickly toward profitable regions of or directions in the design space. Our evaluations across multiple IPs show that author-guided instances of Nautilus can achieve the same quality of results up to an order of magnitude faster than a baseline genetic algorithm.

**CONNECT.** In CONNECT [5, 8] I proposed a Network-on-Chip architecture that 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, and in many cases go against conventional wisdom stemming from traditional ASIC-oriented NoC designs. Compared against a high-quality publicly available synthesizable RTL-level NoC design intended for ASICs, CONNECT consistently offers lower latencies and is able to achieve comparable network performance at one-half the FPGA resource cost; or alternatively, three to four times higher network performance at approximately the same FPGA resource cost. To support this research I developed and released the CONNECT NoC generation engine that can produce synthesizable RTL designs of multi-node NoCs of arbitrary topology. Since its release, the CONNECT NoC generator has gained significant traction (hundreds of users, thousands of generated networks), and has already been used to publish numerous papers by many researchers around the world.

My work on CONNECT is also strongly tied to the CoRAM and Shrinkwrap projects. The CoRAM project investigates a portable memory and I/O abstraction for FPGA platforms that allows users to focus on the performance-critical parts of their applications and spend less time developing board- and device-specific infrastructure, which is traditionally a necessary and painstaking process in FPGA development. For the purposes of CoRAM [6], I developed a custom version of CONNECT that still serves as the primary backend interconnect for CoRAM applications. In Shrinkwrap [7] I collaborated with Eric Chung at Microsoft Research to experiment with
Future Research

My future research agenda is well-aligned with my current research trajectory in facilitating hardware design and studying novel interconnects. It is organized in two thrusts pertaining to: (1) reining in hardware design complexity to enable ubiquitous transparent hardware acceleration for the masses, and (2) rethinking conventional aging ASIC-driven NoC architecture in the face of emerging SoC platforms.

Thrust 1 – Unlocking the Potential of Hardware Acceleration. The overarching goal of this thrust is well aligned with my work in Pandora and aims at tackling the problem of increasing complexity in hardware design. The goal is to develop technologies that will enable the more widespread adoption of hardware acceleration for sustained performance and power efficiency. Today, a computer vision or signal processing expert can relatively easily and quickly experiment in software (e.g., using Matlab), but doing so in hardware is out of the question. My hope is to bridge this gap, so that in the near future application-experts are able to experiment with their ideas in hardware, the same way that they are able to do so in software today.

While efforts such as DELPHI and NAUTILUS are steps in the right direction, the path to ubiquitous seamless hardware acceleration is still unclear and will require breakthroughs in multiple areas of computer science and engineering spanning beyond computer architecture. Tackling important issues such as programmability, portability, composability, interface standardization and graceful fallback mechanisms will require carefully orchestrated advances at the intersection of computer architecture with programming languages, operating systems, runtime environments, as well as electronic design automation flows and CAD tools.

To fully realize the potential of hardware acceleration, we will need programming languages, perhaps domain-specific, that can better capture and expose fine-grain parallelism and patterns that are amenable to hardware acceleration; CAD tools and flows that can rapidly, and potentially on-the-fly, convert and accelerate portions of software using on-chip reconfigurable logic; operating and runtime systems that can manage, virtualize and schedule execution across a diverse set of components in future heterogeneous architectures --- whether these come in the form of conventional general-purpose processors, fixed-function hardened accelerators, or patches of soft reconfigurable logic; and fast flexible low-latency on-chip interconnect solutions that take into consideration emerging fabrication technologies, such as 3D stacking, and can exploit the potential on-chip presence of reconfigurable fabric.

Thrust 2: Application-Specific Interconnects in the Face of Emerging Technologies. In this thrust I plan to leverage my extensive experience in interconnect development [5, 7, 8], simulation [11, 12, 13] and application-specific customization [7] to re-examine conventional aging ASIC-driven NoC design practices in the face of recent and emerging advances in Integrated Circuit (IC) fabrication, such as die stacking and the inclusion of reconfigurable fabric.

In particular, recent advances in packaging technologies have allowed vertical integration of dies in the form of silicon interposers and 3D die stacking. Such technologies are of particular interest for future NoCs, because: (1) the NoC defines a clean “natural” design boundary that maps well to the limited wire availability between stacked dies (compared to within a die), (2) NoCs are designed to tolerate increased latency incurred when travelling from one die to another, and (3) an NoC (or parts thereof) can be built on an interposer based on older, higher-yielding technology nodes without necessarily compromising the overall system performance. Ultimately, research in this area can pave the way towards modular reuse of pre-designed/pre-fabricated NoC interposers, marking a radical departure from current SoC development practices and dramatically lowering the cost and complexity of building a chip.

At the same time, the need for higher power efficiency combined with the desire for post-fabrication customizability are leading to the integration of reconfigurable fabric in modern platforms (e.g., Intel Xeon+FPGA, IBM-Power8 CAPI, Microsoft Catapult, Xilinx Zynq). From an NoC perspective, the presence of reconfigurable fabric creates unique opportunities for research in: (1) generating custom application-specific NoCs, (2) dynamically

---

2 Silicon interposers are large dies built using older and better yielding fabrication technologies that, in their simplest form, act as a passive wiring substrate to host and connect multiple smaller dies.
reconfiguring the NoC (e.g., different topology overlays) or adjusting NoC parameters to better serve temporal changes in communication patterns, (3) enabling feedback-driven adjustment of network performance for Quality-of-Service through smart resource allocation (e.g., increasing network buffers or switching to a different switch scheduler), and 4) using the reconfigurable fabric to create bridge interfaces for integrating IPs from different vendors that adhere to different industry-standard interfaces.

Research Philosophy

Over the years, I have been fortunate to work on and lead several projects with a strong prototyping component, including extensive hardware development, hardware-software codesign, as well as simulation. This experience has made me deeply appreciate the value of performing thorough low-level detailed evaluation of ideas. Albeit slower and more tedious than purely simulation-based approaches, prototyping can not only yield more trustworthy and reliable results, but also, more importantly, often expose subtle but fundamental issues and fallacies that could otherwise go unnoticed. Moreover, I am a strong proponent of openly sharing knowledge, results and infrastructure to this end, I have put significant effort on disseminating my research results by packaging, maintaining, supporting, publicly releasing, and holding tutorials for many of the projects I have worked on, such as ProtoFlex, CoRAM, CONNECT, and DELPHI. Ultimately, in my experience, this hands-on and open approach to research can lead to more pragmatic solutions and generate more useful, high-impact research artifacts.

References


