Research Challenges for On-Chip Interconnection Networks

John D. Owens
University of California, Davis

William J. Dally
Stanford University

Ron Ho
Sun Microsystems

D.N. (Jay) Jayasimha
Intel Corporation

Stephen W. Keckler
University of Texas at Austin

Li-Shiuan Peh
Princeton University

On-chip interconnection networks are rapidly becoming a key enabling technology for commodity multicore processors and SoCs common in consumer embedded systems. Last year, the National Science Foundation initiated a workshop that addressed upcoming research issues in OCIN technology, design, and implementation and set a direction for researchers in the field.

VLSI technology's increased capability is yielding a more powerful, more capable, and more flexible computing system on single processor die. The microprocessor industry is moving from single-core to multicore and eventually to many-core architectures, containing tens to hundreds of identical cores arranged as chip multiprocessors (CMPs).[1] Another equally important direction is toward systems on a chip (SoCs), composed of many types of processors on a single chip. Microprocessor vendors are also pursuing mixed approaches that combine multiple identical cores with different cores, such as the AMD Fusion processors combining multiple CPU cores and a graphics core.

Whether homogeneous, heterogeneous, or hybrid, cores must be connected in a high-performance, flexible, scalable, design-friendly manner. The emerging technology that targets such connections is called an on-chip interconnection network (OCIN), also known as a network on chip (NoC), whose philosophy has been summarized as "route packets, not wires."[2] Connecting components through an on-chip network has several advantages over dedicated wiring, potentially delivering high-bandwidth, low-latency, low-power communication over a flexible, modular medium. OCINs combine performance with design modularity, allowing the integration of many design elements on a single die.

Although the benefits of OCINs are substantial, reaching their full potential presents numerous research challenges. In 2006, the National Science Foundation initiated a workshop to identify these challenges and to chart a course to solve them. The conclusions we present here are the work of all the attendees of the workshop, held last December at Stanford University. All the presentation slides, posters, and videos of the workshop talks are available online at http://www.ece.ucdavis.edu/~ocin06/program.html.

Published by the IEEE Computer Society.

0272-1732/07/$20.00 © 2007 IEEE

We found that three issues stand out as particularly critical challenges for OCINs: power, latency, and CAD compatibility. First, the power of OCINs implemented with current techniques is too high (by a factor of 10) to meet the expected needs of future CMPs. Fortunately, a combination of circuit and architecture techniques has the potential to reduce power to acceptable levels. Second, the latency of these networks is too large, leading to performance degradation when they are used to access on-chip memory. Research efforts to develop speculative microarchitectures that reduce latency through a router to a single clock, circuit techniques that increase signal velocity on channels, and network architectures that reduce the number of hops might overcome this problem. Third, many on-chip network circuit and architecture techniques are incompatible with modern design flows and CAD tools, making them unsuitable for use in SoCs. Research to provide library encapsulation of network components might provide compatibility.

The workshop identified five broad research areas and the key issues in each area:

- *OCIN technology and circuits*. How will technology (such as the CMOS roadmap from the *International Technology Roadmap for Semiconductors*) and circuit design affect on-chip network design?
- *OCIN microarchitecture*. What microarchitecture is needed for on-chip routers and network interfaces to meet latency, area, and power constraints?
- *OCIN system architecture*. What system architecture (topology, routing, flow control, interfaces) is best suited for on-chip networks?
- *CAD and design tools for OCINs*. What CAD tools are needed to design on-chip networks and systems using on-chip networks?
- *Evaluation and driving applications for OCINs*. How should on-chip networks be evaluated? What will be the dominant workloads for OCINs in five to 10 years?

About the workshop

The 2006 Workshop on On- and Off-Chip Interconnection Networks for Multicore Systems, held at Stanford University on 6 and 7 December 2006, brought together about 50 of the leading researchers from academia and industry studying on-chip interconnection networks (OCINs). The NSF-initiated workshop featured invited presentations, poster presentations, and working groups. The 15 invited presentations gave a technology forecast, surveyed applications, and captured the current state of the art and identified gaps in it. The posters covered related topics for which time did not allow a plenary presentation. Each of the five working groups met for a total of four hours to assess one aspect of OCIN technology, to perform a gap analysis, and to develop a research agenda for that aspect of on-chip networks. Each working group then presented a briefing on its findings.

We greatly appreciate the dedication and energy of the workshop participants in defining the research agenda we present in this article. The technology working group included Dave Albonesi, Cornell University; Keren Bergman, Columbia University; Nathan Binkert, HP Labs; Shekhar Borkar, Intel; Chung-Kuan Cheng, UC San Diego; Danny Cohen, Sun Labs; Jo Ebergen, Sun Labs; and Ron Ho, Sun Labs. The system architectures working group members included Jose Duato, Polytechnic University of Valencia; Partha Kundu, Intel; Manolis Katevenis, University of Crete; Chita Das, Penn State; Sudhakar Yalamanchili, Georgia Tech; John Lockwood, Washington University; and Ani Vaidya, Intel. The microarchitectures working group included Luca Carloni, Columbia University; Steve Keckler, University of Texas at Austin; Robert Mullins, Cambridge University; Vijay Narayanan, Penn State; Steve Reinhardt, Reservoir Labs; and Michael Taylor, UC San Diego. The design tools working group included Luca Benini, University of Bologna; Mark Hummel, AMD; Olav Lysne, Simula Lab, Norway; Li-Shiuan Peh, Princeton; Li Shang, Queens University, Canada; and Mithuna Thottethodi, Purdue. The evaluation working group included Rajeev Balasubramaniam, University of Utah; Angelos Bilas, University of Crete; D.N. (Jay) Jayasimha, Intel; Rich Oehler, AMD; D.K. Panda, Ohio State University; Darshan Patra, Intel; Fabrizio Petrini, Pacific National Labs; and Drew Wingard, Sonics.

The generous support of the National Science Foundation (through the Computer Architecture Research and Computer Systems Research programs) and the University of California Discovery Program made the workshop possible. Bill Dally and John Owens chaired the workshop, Timothy Pinkston and Jan Rabaey provided suggestions for workshop direction, and Jane Klickman provided expert logistic and administrative support.

Technology-driving applications

At the workshop, we considered two representative technology-driving applications for on-chip networks.

Applications for CMP systems

Large-scale, enterprise-class systems assembled as CMP-style machines require a high-performance network to attain the throughput important to their applications. For these machines, users will be willing to spend on power to achieve performance, at least to reasonable levels, such as to the air-cooled limit for chips. Cost will be important because it will determine how many racks can be purchased for a data center, but it will not be the overriding

factor. With the emergence of graphics-based applications targeted to the end user, even desktop systems will have general- and special-purpose computing cores and other platform elements integrated on a die. These designs, which require an appropriate on-die interconnect, push technology limits with their need for high bandwidth and low latency under power and area constraints.

Representative applications of OCINs on CMPs include

- *Data centers, including transaction-processing systems and Web servers.* CMPs address the need for further server consolidation, assuming memory bandwidth doesn't limit performance.
- *High-performance computation*. This not only encompasses traditional scientific applications but has expanded to real-time simulation, financial tasks, and bioinformatics.
- *Recognition/mining/synthesis*. Recognition tasks include facial recognition and other computer vision tasks; data mining includes text, image, or speech search. Mined or other data is synthesized to create new models.
- *Medical and health*. Examples are MRI and CT image processing.
- *Desktop computers*. Applications include computationally demanding media and gaming applications such as video and graphics.

Applications for embedded systems

The second driving application is embedded systems. For example, handheld personal electronic systems, of the same type as today's highly integrated cellphone-camcorder-MP3 devices, will require routing networks between elements of their SoC designs. Most CMP applications are also suitable for embedded applications, although perhaps at a smaller scale. The embedded space also includes portable applications that demand computing power coupled with efficiency. Next-generation portable applications include civilian devices such as firefighter communication devices that include real-time monitoring, local weather prediction, and video feedback to a central control location. Communication devices for soldiers will have similar computation, storage, and communication requirements. Other possible applications include real-time medical communication devices, handheld gaming devices, and PDAs.

The primary driver in these systems is cost, followed by active power dissipation (about 200 mW is necessary for a reasonable battery life). Although performance is important for these systems, perhaps more important is their ability to easily connect diverse IP blocks from different designers or vendors into a single system, motivating improved design styles and simple system integration.

The design and performance goals of high-performance systems differ from those of embedded systems. The research community should acknowledge these differences by pursuing research that addresses broad problems across many program domains, as well as more specific research in only one domain.

Technology and circuits

The most important technology constraint for on-chip networks is power consumption. A clear gap exists between today's technologies and what future on-chip networks will be using, not only for communication channels but also for memories used for network buffering.

Other constraints include design productivity and cost, reflecting the problems of using exotic or innovative technologies that require the development of CAD and vendor ecosystems. Still other constraints are reliability and fault tolerance, which are harder to quantify. The latter constraints are even more pressing for dynamically reconfigured routing networks because workload dependencies can make routing paths highly variable, not easily repeatable, and difficult to debug. The technology working group at the Stanford workshop focused on power for enterprise-class CMP machines and personal handheld devices, using some basic "back-of-the-envelope" analysis. For a performance-oriented CMP server, the group first set bandwidth and latency targets required for a typical application and then considered

whether the resulting energy costs would be feasible. For a battery-operated handheld device, the group first set total power dissipation and then calculated how much bandwidth that would support. Both calculations showed clear technology gaps and thus research directions of interest.

Enterprise-class CMP systems

The technology working group imagined a next-generation CMP of the year 2015. Figure 1 shows this design. In a 22-nm technology, a reasonably optimistic design point might integrate 256 cores on a 400-mm^2 die, in a 16 × 16 grid. A mesh routing grid for the 256 cores incorporates 480 total links (15 horizontal core-to-core links in each row or column and 32 total rows and columns), each 1.25 mm long.

The chip, running at 0.7 V, could run at 7 GHz—about 25 gate delays per clock, on par with modern cores. Optimistic wire technology projections estimate a latency using repeaters of 100 ps/mm and a power cost of 0.25 mW/Gbps/mm.[3]

One potential application for such a CMP is data mining. For this application class, we begin with a representative bisection bandwidth requirement of 2 Tbytes/s.[4] With 16 links spanning the chip's bisector, this implies a 1-terabit per second (Tbps) bandwidth requirement per individual link. At a base clock rate of 7 GHz, achieving 1 Tbps requires each link to have 145 bits, or perhaps 72 bits using double-data-rate circuits. The repeated latency of each 1.25-mm link is 125 ps, which enables a single clock cycle per link hop. Long-distance transfers would benefit from multithreaded cores, so that communication across the entire chip would not stall total forward progress.

Such a chip would devote 20 percent of a 150-W power budget to an on-chip interconnect network consisting of three components: channels (wires), buffers, and switch. In our hypothetical design, we budget 10 W for each component. We consider the first two components in more detail.

Network channel power. We calculate network channel power at peak throughput, assuming every single link is fully active at its peak bandwidth. At 1 Tbps and 4.0 total links, this results in 4.0 Tbps over 1.25-mm links, or 150 W at 0.25 mW/Gbps/mm. This exceeds our allocated 10-W budget for the network channels by more than an order of magnitude. Although the assumption of full bandwidth in every link is unrealistic, many systems do exhibit relatively high utilization in short bursts. Even with lowered total activity factors of 25 percent, our network channel power is still unacceptably high.

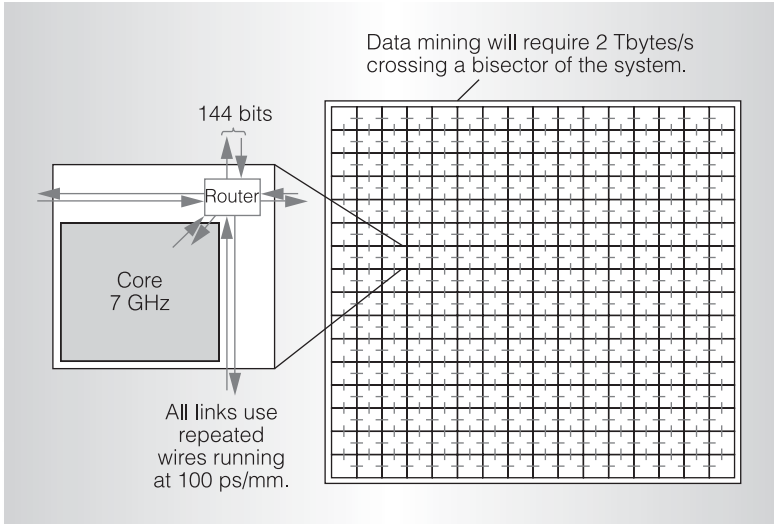

Figure 1. A CMP machine in 2015: 256 cores in a 16 × 16 grid.

Memory power. Each node requires a router with buffering. Bandwidth requirements for data mining call for 145-bit buses, which would attach to five two-way ports per router (bidirectional ports to the north, east, west, and south, and to the local core). To allow flow control and error checking such as cyclic redundancy checks, the buffers should store at least four flits deep; designers often use an additional multiplicative safety margin factor of eight to ensure that local storage never limits channel utilization. This leads to more than 45 Kbits per router of local storage, or more than 10 Mbits of chipwide storage.

For single-cycle access, the access latency of each 45-Kbit memory block should be under 150 ps. A basic low-power, six-transistor (6T) SRAM cell, plus its amortized portion of the decoder and sense-amp peripheral circuits, would require about 0.16 μm^2. Assuming that about 15 percent

of the area would represent switched capacitance, each cell has a load of 2 fF. Over 10 Mbits of memory, this leads to 20 nF, or about 70 W of total power at 100 percent activity. Again, this vastly exceeds the proposed 10-W budget. Using faster but more power-hungry register files would exacerbate the power budget problem.

Personal electronics device

The personal electronics device driver is of great interest not only to the home consumer who uses a cell phone daily but also to professional users who need high bandwidth, a moderate amount of computing, and some storage in a highly integrated hand-held device. The technology working group expected a more specialized OCIN for this design space, and thus considered a tighter constraint of 5 percent total power to be network power. At 5 percent of a 200-mW limit (driven by battery life), network channels can consume only 5 mW each.

Assuming a 50-mm^2 chip, link lengths must be around 7 mm. With an expected power consumption of 0.25 mW/Gbps/mm, this hypothetical system can sustain a total on-chip bandwidth of only 2.8 Gbps. This is remarkably small for future systems; it only slightly exceeds the appetite of a pair of HDTV video feeds and is almost certainly inadequate for tomorrow's computing requirements.

Research agenda

The dominant thread across both application scenarios is power, for communicating data across channels as well as for storage and switching in the network routers. In addition, memory-scaling trends underscore the difficulty of distributing a large, reliable, and fast memory across an on-chip network. Four fruitful research areas can help mitigate these difficulties:

- *Reducing power by reducing the voltage swing on wires*. In addition to fundamental circuit design research, equally important is developing a CAD ecosystem and design infrastructure for low-voltage signaling. ASIC design flows mandate a drop-in replacement for standard signaling to achieve market acceptance in the design community.
- *Integrating multiple chips in a 3D (or at least 2.5D) stack*. Breaking apart a wide, single, monolithic chip into a stack of many smaller chips can make total routes significantly shorter, saving total latency as well as total power.
- *Using photonics on chips*. Optics has achieved traction in chip-to-chip communication paths but not yet in on-chip environments because of integration difficulties and the costs of translating between optical and electrical domains. However, given the potentially low power and extremely low latency of optic connections—15 to 20 times faster than repeated wires—optics on chips is an intriguing area of open research for building routing networks.
- *Reoptimizing basic technology parameters such as the metal buildup in modern processes*. On-chip routing networks, with a preponderance of long wires and a relative dearth of transistors (at least compared with modern microprocessors), might benefit from trading off dense, higher-capacitance lower metal layers for lower-capacitance, coarser upper metal layers. Similar trade-offs might emerge as we reexamine underlying technologies specifically for routing networks.

Microarchitectures and system architectures

Having identified the fundamental circuit and technology issues, we turned to higher-level design issues: OCIN microarchitectures and system architectures. The individual presentations and the group discussion made clear that the best network microarchitecture depends strongly on an application's bandwidth and latency needs. A survey of several recent prototypes and products confirms that even for today's technologies, OCIN design varies widely, including

- high-bandwidth mesh networks connecting dozens of components,[5,6]
- ring and star networks for modest bandwidth communication between nearby IP blocks,[7] and

- shared-bus and crossbar architectures for SOC applications.[8]

Some workshop attendees made cases for simple networks (networks with highly concentrated, lower-bandwidth links such as a bus or segmented bus) for applications with limited bandwidth demand. The increase in on-chip wire density might even extend the range for which such networks are feasible. However, other attendees claimed that scaling to higher bandwidths requires routed networks with less-concentrated links rather than highly concentrated bus-oriented networks. The difference in opinions among the attendees shows that further work is necessary to determine optimal network designs for applications of varying bandwidth demands.

Workshop members did agree that latency and power are the two most critical cross-cutting design challenges for OCIN architectures. They also discussed several other important research directions, including programmability, managing reliability and variability, and scaling on-chip networks to new technologies.

Latency

Minimizing latency in on-chip networks is critical to approaching the characteristics of traditional chip-level bus interconnects, which have typically been small in scale and low in latency. Low-latency networks make the system designer's and the programmer's jobs easier because low overhead reduces the need to avoid communication and encourages efforts in exploiting concurrency.

Network interfaces. Efficient, lightweight OCIN interfaces are critical for overall latency reduction because the transmission time on wires and in routers in today's networks is often dominated by software overheads into and out of the networks. We see a need for thin network abstractions that expose hardware mechanisms for use by application-level programmers. These networks should be tightly coupled to the computation or storage elements attached to them, but they should also be general purpose to provide portability and utility across various uses. Virtualizing the network interface is a promising approach for providing atomicity and security, but such interfaces must not unduly add to latency. Research on remote queues,[9] automatic method invocation on message arrival,[10] and integrated microarchitectural networks[6] has previously appeared, but more work on both the hardware and the software sides of network interfaces is needed.

Routers. Innovations in router architecture and microarchitecture are needed to reduce OCIN latencies while maintaining reasonable area and power budgets. Reducing the number of pipeline stages in the router is critical, as is congestion control with bounded or limited router buffering. Recent work in speculative router architectures pushing router pipelines to a single stage is promising, but more research is needed in speculative microarchitectures to improve accuracy and efficiency.[11] Another promising research area is flow-control algorithms and microarchitectures that identify and accelerate critical traffic without substantially affecting the latency of less critical traffic. Research on better network and interface support for out-of-order message delivery to further the aims of adaptive routing is also promising. Improved efficiency and performance might be accessible to networks that exploit some form of static or stable information from the application. Potential examples are circuit-switched networks or a hybrid packet- and circuit-switched network, if circuit configuration time can remain small.

Exploiting wire density. The abundance of on-chip wires changes the trade-offs in network design. As mentioned, increased wire density can extend the viability of concentrated networks (such as bus-like networks) by allowing more links between network endpoints. Increased wire density can also open opportunities for innovations in OCIN topologies supported by higher-degree routers. Finally, wire densities will likely reduce the importance of virtual channels, because physical channels might no longer be the critical network resource. Such shifts in relative technology costs demand examination and innovation in OCINs.

Power

Power has become a major concern in system design and must be budgeted and traded off among different system parts, including the communication infrastructure. As described earlier, not all systems using on-chip networks will operate at the same power-performance point. Promising areas of research on power techniques for various deployment domains include

- power-efficient designs that limit router complexity and unnecessary work,
- adaptive power management that lets networked systems shift power between computation and communication on the basis of the application or application phase, and
- dynamic voltage and frequency modulation in the network.

Programmability

For an effective concurrent SoC or multicore system, a programmer needs a fast and robust on-chip network transport, fast and easy-to-use network interfaces, and predictable network performance.

Modeling and measurement. In effect, today's networks are black boxes to programmers, who find it very difficult to reason about network bottlenecks when writing and optimizing their programs. To solve this problem, we recommend research into network modeling and measurement techniques for use by application programmers. Network modeling means developing cost models for network latency under different traffic patterns and workloads to enable programmers to predict how an application will perform. The community should not be surprised if sacrificing peak network performance for a greater degree of predictability is desirable. Measurement means network hardware, such as performance counters, and tools that can synthesize the measurements into feedback that helps programmers understand how an application uses the network. Many tools have been developed over the past decade to help programmers understand program performance on uniprocessors; it is time to embark on the creation of such tools for OCINs.

Network robustness. Network robustness includes low-overhead support for deadlock avoidance, mechanisms for quality of service for traffic of different priorities, and network-based tolerance of unexpected failures. One promising mechanism for handling unusual network events in a lightweight fashion is network-driven exceptions that can be handled in software by general- or special-purpose processing elements. Network microarchitectures should be scalable across generations of systems, and a related challenge is interfacing on-chip networks to off-chip, board-level, rack-level, and systemwide networks. Unifying the protocols across these different transport layers can make the protocols easier to build and easier for programmers to reason about.

Network services. Incorporating more intelligence into the network and its protocols can ease programmer burden and simplify system design. Recently, researchers have discussed incorporating support for cache coherence in the network layer.[12] Other possible research areas include security and encryption services. Whether breaking down abstraction barriers between the transport layer and the memory layer is viable, and what other opportunities exist for creating high-level network-based services remain open questions.

Reliability and variability

With shrinking transistor and wire dimensions, reliability and variability have become significant challenges for IC designers. Past research has examined methods of providing network reliability. Now on-chip networks will need new lightweight mechanisms for link-level and end-to-end service guarantees. One example is self-monitoring links and switches that detect failures and intelligently reconfigure themselves. Both high-performance and embedded systems will require power-, latency-, and area-efficient, error-tolerant designs to provide useful on-chip network infrastructure.

Fabrication variation. Fabrication process variability, either on dies or across wafers, can prevent a single static design from achieving high performance and low power for all fabricated devices. Postfabrication network tuning is a promising way to tolerate fabrication faults as well as speed variations of different network elements. Some form of network self-test, along with configuration—perhaps in the same way on-chip memories employ redundant rows—might prove useful. Another method might be to exploit elasticity in the network links to tolerate variations in router speeds, perhaps using self-timed or asynchronous circuits and microarchitectures.

Traffic variation. Another form of variability arises from the different types of traffic delivered by different applications or different phases of the same application. Applications differ in message length, message type (data, synchronization, and so forth), message patterns (regular streams, unstructured, and so forth), and message injection rates (steady or bursty). Again, the abundance of on-chip wires provides an opportunity to specialize or replicate networks to improve latency or efficiency across multiple types of loads. Identifying the proper set of on-chip communication primitives and designing networks that implement them will be a valuable line of inquiry.

Technology scaling

Network design has always been subject to technology constraints, such as package pin bandwidth. Although wire count constraints are less important on chip, smaller feature sizes affect the relative cost of communication and computation. Faster computation relative to wire flight time motivates more intelligent routing algorithms designed to minimize message hop count and network congestion. Combined with the likelihood of large numbers of on-chip networked elements, this trend indicates a need for research into technology-driven and scalable router, switch, and link designs. As emerging technologies, such as 3D die integration, on-chip optical communication, and any of many possible postsilicon technologies become viable, new opportunities and constraints will further drive the need for innovation in interconnection networks. We must make early investments in characterizing changing and emerging technologies from the perspective of on-chip networks as well as new network designs motivated by such shifts in technology.

OCIN design tools

The desire for flexible, high-performance OCINs compatible with modern chip design approaches motivates new approaches to the design tools that will create them. The design tools working group identified seven key research challenges in the development of CAD tools targeting multicore processor chips and SoCs. Figure 2 shows an overview of these research challenges.

1. *Interface of network synthesis with system-level constraints and design*. As chips move toward multicore design in future technologies, system-level constraints become increasingly complex, and requirements become more multifaceted. It is essential for OCIN synthesis tools to interface effectively with these constraints and requirements. The foremost challenge is the accurate characterization and modeling of system traffic, such as that imposed by a shared-memory SoC or a platform-specific chip.
2. *Hybrid custom and synthesized tool flow*. General-purpose processors typically lead the embedded market with aggressive, innovative microarchitectures and custom designs. It is therefore critical for design tools to leverage these high-performance designs within the existing tool flow for easy adoption into mass-market embedded devices. Can we construct specialized libraries for networks, and how can we integrate them into the entire CAD tool flow? This is particularly important for facilitating fast transfer of research into products benefiting the mass market.
3. *Design validation*. A critical hurdle in deploying on-chip networks is validat-

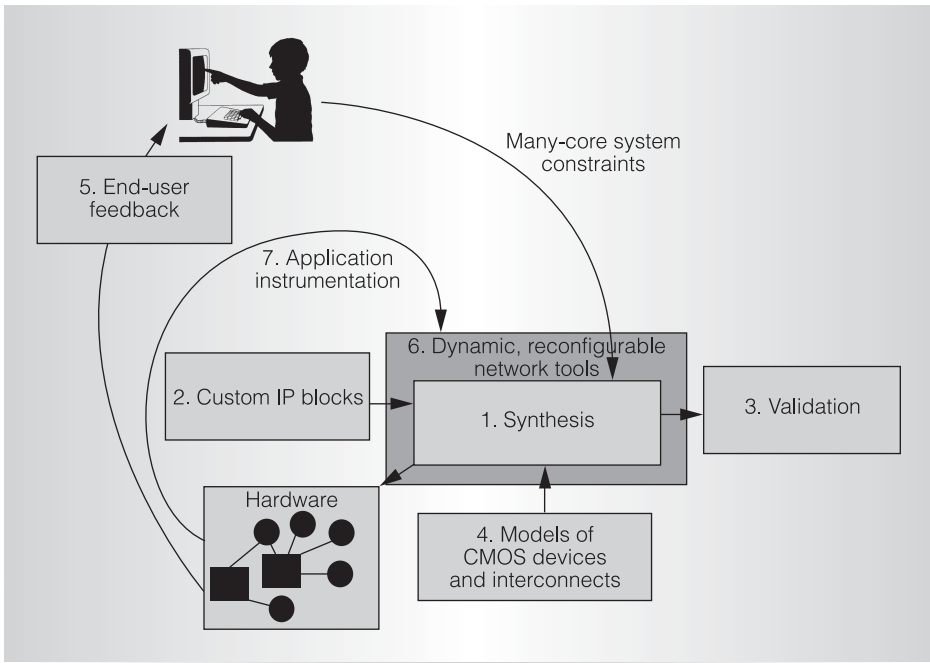

Figure 2. Overview of CAD challenges in on-chip network design and how the subcomponents interact and form the envisioned next-generation CAD tools for on-chip networks.

ing their operation. The key questions are how can we ensure designs are robust in the face of process variations and tight cost budgets, and how can we factor validation cost into the CAD design tool chain?

4. *Impact of CMOS scaling and new interconnect technologies*. For design tools to be effective as CMOS scales, we need new timing, area, power, thermal, and reliability models for future CMOS processes, circuits, and architectures. New interconnect technologies must meet this need to ease adoption. Models and libraries should be available with proposals of new interconnects. This modeling infrastructure should also be extensible to ensure integration of new technologies and interconnects.
5. *Design tool chain with end-user feedback*. As network scale and complexity increase, new design tools must provide feedback to help designers. For instance, feedback of network characteristics would allow designers to quickly iterate their designs. Research in this domain can potentially leverage design tool feedback research in other network domains such as the Internet, although there are clearly substantial differences in the OCIN domain's requirements.
6. *Dynamic, reconfigurable network tools*. Not only must general-purpose multicore chips support a wide variety of traffic and applications, OCINs in SoC platforms also must increasingly support a wide variety of applications to facilitate fast time to market. So dynamic reconfigurable network tools will be very useful, allowing soft router cores that can be configured on the fly to match different application profiles, similar to just-in-time software compilation.
7. *Beyond simulation*. Today's network design tools rely heavily on network simulation to drive power and performance estimates. For future large-scale networks and systems, however, simulation will no longer be tenable because of their complexity. Thus, we see a need for research into analytical methods, such as formal methods and queuing

analysis-based tools for estimating network power-performance. Although researchers can leverage prior work, the key distinct features of on-chip networks (such as physical constraints and link-level flow control) motivate new analysis approaches as well.

The seven design tool challenges will critically affect both the embedded-SoC and the general-purpose computing markets. Overcoming these challenges will enable complex, correct network designs that would otherwise be impossible and facilitate the adoption of on-chip networks.

Evaluation and driving applications for OCINs

The evaluation working group began by identifying the applications and workloads (described earlier) most likely to drive interconnect requirements and then characterized those workloads in terms of the architecture and programming model. From that characterization, we studied the network requirements and pinpointed a research agenda to address them.

Architectural characterization and programming models

How do the driving applications affect an OCIN? These applications have diverse access patterns. For example, one pattern is heavily cacheable traffic (read-only and read-write sharing), which places a significant performance burden on the on-chip interconnect. Another pattern is streaming traffic from DRAM or I/O, which places the primary burden on external interfaces (mainly because of pinout limitations) and a secondary burden on the on-chip network. A second difficulty is the traffic's bursty nature and the additional pressure that places on congestion management mechanisms.

With increasing integration, we expect that single-chip devices will have a diverse set of data producers and consumers attached to the OCIN. These might include specialized engines such as shader, texture, and fixed-function units.[13] Packetization at cache-line granularity would be inefficient for a subset of traffic generated by such units. Hence, the interconnect not only must efficiently support diverse traffic patterns but also must possibly meet quality-of-service guarantees or even soft real-time constraints. SoC architectures in which a large number of diverse IP blocks is the rule rather than the exception exacerbate these needs.[14]

Supporting multiple cores on a single chip also reveals new management problems. With server consolidation workloads, a single CMP must be dynamically partitioned into several systems. But it must also support performance isolation (one partition's traffic shouldn't affect another partition's performance) and fault isolation (a partition reset shouldn't force reset of another partition). In addition, security concerns require that different system parts running separate applications be effectively isolated. Interestingly, many of these seemingly diverse scenarios have a commonality from the on-chip network's perspective: Because the network is shared, all these scenarios require some form of network isolation—either virtual or physical.

We also forecast a need to support synchronization or communication primitives in the network for coherence-style traffic (for example, to efficiently broadcast and collect invalidations at the home nodes) and message-passing traffic (for example, to broadcast data). In the first scenario, with hundreds of processing elements, even directory-based systems wouldn't scale without such interconnect support.

Because of these diverse application requirements and the equally diverse programming styles that will create software for these processors, we expect CMPs to support both coherent shared-memory and message-passing programming modes. This motivates efficient support in the interconnect for cache-sized line transfers and variable-length message transfers.

Network requirements and evaluation metrics

On the basis of the architectural characterization, we recommend four areas of emphasis in OCIN design and implementation:

- Efficient data transfer support at various granularities for coherent and

message-passing paradigms and for different types of specialized engines.
- Support for partitioning, including quality-of-service guarantees (perhaps through separate virtual channels or completely partitioned subnetworks), performance isolation (so that partitions don't share routing paths in the OCIN), and isolation for security (through partitioned subnetworks).
- Clean, efficient common network interfaces to support multiple programming models.[15]
- Possible support for synchronization and communication primitives such as multicast and barriers.

A further research challenge is to define evaluation metrics, such as latency and bandwidth, under the constraints of chip area, power, energy, and heat dissipation. Another need is to standardize the evaluation metrics so that architectural implementations can be unambiguously compared.

Beyond the design issues mentioned earlier, we also pose the following research questions:

- With the need for dynamic partitioning[16] and possibly fault tolerance resulting from process variability or the need for reliability, network topology doesn't remain static. Dynamic partitioning thus creates subnetworks with different topologies than the static one. What support is needed at the hardware and system software levels to support such dynamic reconfiguration?
- How can we develop analytical models to predict the real-time guarantees of the architecture being designed? SoC designs have a particular need for these models.
- How can we monitor network performance under constraints to study the effectiveness of networkwide policies? For instance, once network utilization has crossed a threshold, how does a particular class of traffic behave?
- We recognize that realistic full-system simulation, especially execution-based simulation, will not be possible given the current set of tools and methodologies. Many groups in academia and industry are resorting to emulation through the use of FPGAs to overcome the simulation speed problem. Is that sufficient? A concerted effort across multiple research disciplines in computer engineering is necessary for a realistic study of CMP and SoC workloads.
- How can we compare different systems under similar workloads? The community must develop a suite of workloads and benchmarks for such a comparison (such as the SPEC suite used by the CPU community[17]). The suite should specify the mix of workloads to run concurrently and should provide common evaluation criteria for comparison. This requires a cooperative effort by groups in academia and industry interested in CMP and SoC architectures. There has been initial activity in this direction in the SoC community[18] and a call to action in the CMP community.[19]

OCINs are a critical technology that will enable the success of future CMPs and SoCs for embedded applications. To make sure that this technology is in place when needed, we recommend a staged research program to carry out the following key tasks:

Develop low-power circuits and architectures. To close the power gap, research should develop optimized circuits for OCIN components: channels, buffers, and switches, as well as architectures targeted for low power. This research can reduce OCIN power consumption by an order of magnitude, allowing it to fit in the expected power envelopes for future CMPs and SoCs. This work will set the constraints and provide optimized building blocks for architecture and microarchitecture efforts.

Develop low-latency network and router architectures. Architecture research must address the primary issues of power and latency, as well as critical issues such as congestion control. This work should

address network-level architecture (topology, routing, and flow control), as well as router microarchitecture. It should reduce the delay of routers (possibly to one cycle) and reduce the number of hops required by a typical message. Circuit research to reduce channel latency can also help close the latency gap. This work will enable OCINs to match the latency of dedicated wiring.

Encapsulate OCIN components. To make OCIN technology accessible to SoC designers, research on design methods must encapsulate the OCIN components and architectures in libraries and generators that are compatible with standard CAD flows—for example, as parameterized hard macros. Tools that automatically synthesize OCINs from these macros (as well as from blocks of standard logic) are also needed. This research will remove one of the largest roadblocks to adoption of on-chip networks in SoCs.

Develop prototype OCINs. The research community should design, construct, and evaluate optimized prototypes, which can expose unanticipated problems, provide a baseline for future research, and serve as a testbed for new OCIN components. This work will also serve as a proof of concept for OCINs, reducing their perceived risk and facilitating transfer of this technology to industry.

Develop standard benchmarks and evaluation methods. To keep OCIN research focused on real problems, the community should develop standard benchmarks and evaluation methods. Standard benchmarks allow direct comparison of research results and facilitate information exchange between researchers.

If our recommended research course is successful, OCINs are likely to realize their potential to provide high-bandwidth, low-latency, low-power interconnect for CMPs and SoCs. OCINs will provide a key technology needed for the large-scale CMPs expected to dominate computing in the near future. Without this research, OCINs won't meet the needs of many next-generation CMP applications—leading to a serious on-chip bandwidth issue for future computers—and optimized OCINs won't be usable in SoCs because of CAD tool and design flow incompatibilities. MICRO

References

1. ''Special Session: Thousand-Core Chips,'' 44th Design Automation Conf., 2007, http://www2.dac.com/data2/44th/44acceptedpapers.nsf/websessions/42.
2. W.J. Dally and B. Towles, ''Route Packets, Not Wires: On-Chip Interconnection Networks,'' *Proc. 38th Conf. Design Automation* (DAC 01), ACM Press, 2001, pp. 684-689.
3. R. Ho, K. Mai, and M. Horowitz, ''Managing Wire Scaling: A Circuit Perspective,'' *Proc. IEEE Int'l Interconnect Technology Conf.*, IEEE Press, 2003, pp. 177-179.
4. K. Bergman et al., ''Optical On-Chip Networks for High-Performance, Energy-Efficient Multi-Core Architectures,'' poster session, Workshop On- and Off-Chip Interconnection Networks for Multicore Systems, Dec. 2006, http://www.ece.ucdavis.edu/~ocin06/posters.html.
5. Intel's Teraflops Research Chip, 2007, http://techresearch.intel.com/articles/Tera-Scale/1449.htm.
6. P. Gratz et al., ''Implementation and Evaluation of a Dynamically Routed Processor Operand Network,'' *Proc. 1st Int'l Symp. Networks-on-Chip* (NOCS 07), 2007, pp. 7-17.
7. ''STMicroelectronics Unveils Innovative Network-on-Chip Technology for New System-on-Chip Interconnect Paradigm,'' press release, Dec. 2005, http://www.st.com/stonline/press/news/year2005/t1741t.htm.
8. ''Sonics Defines SoC Interconnect Choices,'' press release, June 2006, http://findarticles.com/p/articles/mi_m0EIN/is_2006_June_26/ai_n16499393.
9. E.A. Brewer et al., ''Remote Queues: Exposing Message Queues for Optimization and Atomicity,'' *Proc. 7th Ann. ACM Symp. Parallel Algorithms and Architectures* (SPAA 95), ACM Press, 1995, pp. 42-53.
10. W.J. Dally et al., ''The Message-Driven Processor: A Multicomputer Processing Node with Efficient Mechanisms,'' *IEEE Micro*, vol. 12, no. 2, Mar.-Apr. 1992, pp. 23-39.
11. R. Mullins, A. West, and S. Moore, ''Low-Latency Virtual-Channel Routers for On-

Chip Networks,'' *Proc. 31st Ann. Int'l Symp. Computer Architecture* (ISCA 04), IEEE CS Press, 2004, pp. 188-197.
12. N. Eisley, L.-S. Peh, and L. Shang, ''In-Network Cache Coherence,'' *Proc. 39th Ann. IEEE/ACM Int'l Symp. Microarchitecture* (Micro 06), IEEE CS Press, 2006, pp. 321-332.
13. J. Held, J. Bautista, and S. Koehl, ''From a Few Cores to Many: A Tera-Scale Computing Research Overview,'' 2006, http://download.intel.com/research/platform/terascale/terascale_overview_paper.pdf.
14. L. Benini and G. De Micheli, ''Networks on Chips: A New SoC Paradigm,'' *Computer*, vol. 35, no. 1, Jan. 2002, pp. 70-78.
15. M. Katevenis, ''Towards Light-Weight Intra-CMP Network Interfaces,'' Workshop on On- and Off-Chip Interconnection Networks for Multicore Systems, Dec. 2006. http://www.ece.ucdavis.edu/-ocin06/program.html.
16. J. Duato et al., ''Part I: A Theory for Deadlock-Free Dynamic Reconfiguration of Interconnection Networks,'' *IEEE Trans. Parallel and Distributed Systems*, vol. 16, no. 5, May 2005, pp. 412-427.
17. J.L. Henning, ''SPEC CPU2006 Benchmark Descriptions,'' *SIGARCH Computer Architecture News*, vol. 34, no. 4, Sept. 2006, pp. 1-17.
18. C. Grecu et al., ''An Initiative towards Open Network-on-Chip Benchmarks,'' 2007 http://www.ocpip.org/socket/whitepapers/NoC-Benchmarks-WhitePaper-15.pdf.
19. J. Rattner, ''Cool Codes for Hot Chips,'' Hot Chips 18, 2006, http://www.hotchips.org/archives/hc18/2_Mon/HC18.Keynote%20One/HC18.Keynote1.pdf.

John D. Owens is an assistant professor of electrical and computer engineering at the University of California, Davis. His research interests include GPU computing and, more broadly, commodity parallel hardware and programming models. Owens has a PhD in electrical engineering from Stanford University.

William J. Dally is the Willard R. and Inez Kerr Bell Professor of Engineering and the chairman of the Department of Computer Science at Stanford University. His research interests include high-speed signaling, computer architecture, network architecture, and programming systems. Dally has a PhD in computer science from California Institute of Technology.

Ron Ho is a distinguished engineer at Sun Microsystems. His research interests include off-chip and on-chip communication technologies. Ho has a PhD in electrical engineering from Stanford University.

D.N. (Jay) Jayasimha is a principal engineer in the Corporate Technology Group at Intel Corporation. His research interests include multiprocessor architectures, interconnection networks, and performance analysis. Jayasimha has a PhD from the University of Illinois at Urbana-Champaign.

Stephen W. Keckler is an associate professor of computer sciences and electrical and computer engineering at the University of Texas at Austin. His research interests include computer architecture, interconnection networks, and parallel processor architectures. Keckler has a PhD in electrical engineering and computer science from the Massachusetts Institute of Technology.

Li-Shiuan Peh is a guest editor of this special issue. Her biography appears on page 5.

Direct questions and comments about this article to John D. Owens, Dept. of Electrical and Computer Engineering, University of California, Davis, One Shields Ave., Davis, CA 95616; jowens@ece.ucdavis.edu.