DRAFT Final Report:
Workshop on On- and Off-Chip Networks for Multi-Core Systems

William Dally
December 2006

Executive Summary

A workshop was held at Stanford University on December 6 and 7, 2006 to survey the state-of-the art in on- and off-chip interconnection networks, to perform a gap analysis on this technology, and to chart a research agenda for this field. The workshop brought together about 50 of the leading researchers studying on-chip interconnects from both academe and industry. Attendees included representatives of major computer companies such as Intel, AMD, Sun, and HP; academics from the US and Europe; and representatives of DARPA and NSF.

The workshop consisted of invited presentations, poster presentations, and working groups. The 15 invited presentations gave a forecast of technology for building on-chip networks, surveyed applications for on-chip networks, captured the current state of the art in on-chip networks, and identified some gaps in the current state of the art. 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 on-chip network 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 their findings. All of the presentation slides, posters, and videos of the talks for the workshop are available on-line at http://www.ece.ucdavis.edu/~ocin06/program.html.

The workshop identified on-chip interconnection networks (OCINs) as a key, enabling technology both for multi-core and many-core processors (also called CMPs), which are rapidly becoming the prevalent computing platform, and also for systems on a chip (or SoCs) which are commonplace in consumer embedded systems. The working groups, identified several gaps in existing knowledge about OCINs that if not remedied will prevent OCINs (and CMPs and SoCs) from realizing their potential.

Three issues stood out as being particularly critical challenges for OCINs: power, latency, and CAD compatibility. In fact, one industry participant in the workshop conjectured that the power and latency problems were so severe as to make buses (which do not scale) preferable to OCINs. First, the power of on-chip networks implemented using current techniques is too high (by a factor of 10) to meet the expected needs of future CMPs. Fortunately research to develop 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 using these networks to access on-chip memory. A research effort to develop speculative micro-architecture (which has the potential to reduce latency through a router to a single clock), circuit techniques (which may be able to increase signal velocity on channels), and network architecture (which can reduce the number of hops) may be able to address this issue. 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 may be able to provide compatibility.

To address these three primary issues, as well as several secondary issues, the workshop recommends a staged research program to advance the technology of on-chip networks. First, research is needed to develop optimized circuits for OCIN components: channels, buffers, and switches. This work will set the constraints and provide optimized building blocks for architecture and microarchitecture efforts. Second, architecture research is needed to address the primary issues of power and latency as well as to deal with other critical issues, such as congestion control. This work should address both network-level architecture (e.g., topology, routing, and flow-control) as well as router microarchitecture. Research on design methods is needed to encapsulate the components and architectures developed in libraries and generators that are compatible with modern CAD flows, making this research accessible to SoC designers. Finally, to facilitate the above research, the community needs to develop a set of standard benchmarks and evaluation techniques to enable realistic evaluation of proposed approaches and uniform comparison between approaches. A series of prototype OCINs should be built by the research community to provide a testbed for research in this area and to identify unanticipated research issues.

If the recommended course of research is pursued and is successful, OCINs are likely to realize their potential to provide high-bandwidth, low-latency, low-power interconnect for CMPs and SoCs – providing a key technology needed for the large-scale CMPs expected to dominate computing in the near future. If this research is not performed, OCINs will not be able to meet the needs of CMPs – leading to a serious on-chip bandwidth issue with future computers – and optimized OCINs will not be usable in SoCs due to CAD tool and design flow incompatibilities.

Workshop Overview

The workshop consisted of 15 invited talks, two poster sessions, two working group sessions, a panel, and a wrapup discussion. The invited talks were intended to give an overview of the current state-of-the art of OCIN technology and covered the following topics:

1. Applications of OCINs (3 talks)
2. Technology constraints on OCINs (3 talks)
3. OCIN Architecture and Design Technology (6 talks)
4. Working OCIN Prototypes (3 talks)

The invited talks served to give the working groups a common base of knowledge with which to start their deliberations. The posters complemented the talks by covering additional OCIN research topics. The panel served to start the discussion on the research issues that emerged from the invited talks at posters.

The workshop divided into five working groups each of which was charged with performing a gap analysis and developing a research agenda on a particular aspect of OCIN technology. The five groups titles and charges are:

1. Technology and Circuits for On-Chip Networks: How will technology (ITRS CMOS) and circuit design affect the design of on-chip networks. What key research issues must be addressed in this area?
2. Evaluation and Driving Applications for On-Chip Networks: How should on-chip networks be evaluated? What will be the dominant

workloads for OCNs in 5-10 years. What key research issues must be addressed in this area?

3. CAD and Design Tools for On-Chip Networks: What CAD tools are needed to design on-chip networks and to design systems using on-chip networks? What research issues must be addressed in this area?
4. System Architecture for On-Chip Networks: What system architecture (topology, routing, flow control, interfaces) is optimal for on-chip networks? What are the research issues in this area?
5. Microarchitecture for On-Chip Networks: What microarchitecture is needed for on-chip routers and network interfaces to meet latency, area, and power constraints? What are the research issues in this area?

The remainder of this report gives the findings and recommendations of each of the working groups.

Technology and Circuits for On-Chip Networks

A small working group discussed technology constraints and the impact of circuits for on-chip networks. The group members' areas of interest spanned architecture, optics, networks, and circuits:

Dave Albonesi, Cornell University, architecture
Keren Bergman, Columbia University, photonic network architectures
Nathan Binkert, HP Labs, system architecture and networking
Shekhar Borkar, Intel, technology and circuits
Chung-Kuan Cheng, UC-San Diego, communication circuits and wires
Danny Cohen, Sun Labs, network architectures
Jo Ebergen, Sun Labs, circuits and clocking
Ron Ho, Sun Labs, communication circuits and wires

The conclusions of this group are that power is clearly a gap between today's technologies and those needed by on-chip networks in the future. This applies not only to the communciation channels in the network but also to the memories used for buffering in the network. Areas of particular research interest include

Circuits: Low-energy circuits for realizing network channels and routers.
Technology: 3D integration to enable shorter and thus lower-power channels
Technology: On-die photonics to improve latency, bandwidth, and maybe power
Technology: Re-optimizing the metal stacks on chips for routing and low power

The group considered two technology drivers for on-chip networks. First, enterprise-class systems of a large scale, assembled as CMP-style machines, would require a high-performance network to attain the throughput important to its applications. For these machines, users would be willing to spend on power in order to achieve performance, at least to reasonable levels, such as to the air-cooled limit for chips. Cost would be important, as it would set how many racks could be purchased for a data center, but it would not be the overriding factor.

Second, hand-held personal electronic systems, of the type embodied today by highly integrated cellphone/camcorder/MP3 player devices, would also need some routing network between elements of its System-on-a-Chip (SoC) design. In these

machines, cost is the primary driver: a plenary speaker from SonICs, Drew Wingard, had impressed on the audience that cost drives all chips down to no larger than 50mm^2—anything bigger is too expensive. These machines also have severe power constraints, such that active power could not exceed much more than 200mW, in order to maintain reasonable battery life.

Given the wide problem space of on-chip networks, the working group established a "back-of-the-envelope" analytical flow that first set the required bandwidth of a technology driver, checked the communication latency of a possible solution, and then verified the energy costs of that solution to see if it would work.

Other constraints mentioned but not yet explored included design time and cost, which reflect the problems of using exotic or innovative technologies that require the development of CAD and vendor ecosystems, as well as reliability and fault tolerance and how these can be quantified. The latter constraints become even more pressing for dynamically reconfigured routing networks, especially as workload dependencies may make routing paths highly variable and difficult to debug.

Enterprise-class CMP systems

We assumed that in the year 2015, in a 22nm technology, a reasonably optimistic design point might integrate 256 cores on a 400mm^2 die, in a 16x16 grid. The chip might run at 7GHz—about 30 gate delays per clock, on-par with modern SPARC cores—and use a 0.7V power supply. Projections of wire technologies estimate a latency using repeaters of 100ps/mm and a power cost of 0.25mW/Gbps/mm. Both of these are optimistic, the latency somewhat more so than the power.

The basic routing architecture could assume many forms: for example, Shekar Borkar, a member of the working group, had argued for a bus-like broadcast architecture in his plenary talk. However, for this discussion we agreed on a mesh-style routing grid. A mesh routing grid for 256 processors incorporates 480 total core-to-core links, each 1.25mm long. This includes 15 horizontal core-to-core links in each row and 16 total rows. The resulting product is then doubled to account for the vertical links.

We assumed a thermal limit of 150W, about 40W per square cm. While below the air-cooled limit, this is still an aggressive thermal budget, rarely exceeded in products today. Of this 150W, we allocated 20% (30W) to the network, leaving 80% for computation, storage, and off-chip I/O. A network consists of three components: the channels (wires), the switch, and the buffers. A series of plenary talks showed that generally, these were balanced in their power dissipation, so that each would be budgeted 10W in our hypothetical design.

A working group participant, Dave Albonesi, had earlier examined data-mining applications and reported a bisection bandwidth requirement of 2TBps. While this was not the highest number he found, it was representative of the class of benchmarks he examined. At 2TBps, or equivalently 16Tbps, that sets 1Tbps per individual link across any vertical or horizontal system bisection. At a base clock rate of 7GHz, a 1Tbps throughput sets each link to have 145 bits, or perhaps 72 bits but using double-data-rate circuits. With systems today regularly using data buses 128 bits wide, this is not an onerous constraint.

The latency of each link, set by a length of 1.25mm and a wire propagation at 100ps/mm, is 125ps, which is lower than a clock cycle. This enables a single cycle per

link hop, which appeared to be reasonable, especially for short-distance communciation. Long-distance transfers would push for cores with sufficient multi-threading capabilities that communication across the entire chip would not stall total forward progress.

The power for the network channels, calculated at peak throughput, assumes every single link is fully active at its peak bandwidth. At 1Tbps and 480 total links, this results in 480Tbps over each 1.25mm link, or 150W at 0.25mW/Gbps/mm. This easily exceeds the 10W budget we had pre-planned for the network channel, by over an order of magnitude. While the assumption of full bandwidth in every link is unrealistic, many systems do exhibit relatively high utilization in short bursts.

The group also considered the buffering required in a router. We considered flits in the network that would be fairly wide, as they might be for coherency traffic between cores carrying cache lines. Thus we assumed 16B wide flits. Each routing point would have five bidirectional ports: one for each compass direction, and a fifth for the local core. Thus total traffic per cycle would be on the order of 160B, or 1280b.

The depth of the memory is set by timing. At a minimum, we expected to have to store four flits deep. This would allow for two cycles of flow control to the next core, one cycle there and one cycle back; plus two cycles of some error checking, such as a cyclic redundancy check. In many routers, designers then multiply this buffering by another factor of eight (or so) for safety, to ensure that the local storage never limits the channel utilization. This brought our storage to 1280b * 4flits * 8 = 40Kb/router. At 256 routers per chip, this led to a 10Mb distributed memory.

The latency of each 40Kb memory block should be under 150ps to maintain single-cycle access. Shekhar Borkar pointed out that given scaling trends in SRAM design, this may be difficult without resorting to register-file based memories, which carry a high area and power premium. However, we assumed a basic low-power 6T SRAM cell using "sub-DRC" design rules for compactness to give a 0.16μm^2 cell.

We took a swag at the power of a memory by assuming about 15% of the area would represent switched capacitance, or about 2fF/cell. This is an admittedly crude estimate, as it approximates both full-swing wordlines as well as bitlines swinging at less than the full supply. Over 10Mb of memory, this leads to 20nF, or about 70W of total power for full utilization. Again, this vastly exceeds the 10W budget we had proposed, even using the low-power SRAM cells instead of the faster register file-based memories.

Personal electronics device

The personal electronics device driver is of great interest not only to the home consumer who uses her cellphone on a daily basis, but also to professional users who need high bandwidth, a moderate amount of computing, and some storage on a highly integrated hand-held device. One example from the workshop sponsors was for forest firefighters armed with with real-time monitoring, local weather prediction, and video feedback to a central control location. Another example discussed was for the next-generation foot-based war-fighter, with strikingly similar computation, storage, and communication requirements.

In this design space, we considered a tighter constraint of 5% total power to be network power. Unlike enterprise-class CMP machines, hand-held systems would have highly specialized, almost one-off network designs between their various SoC blocks.

Thus we would expect the total power budget of the communication to be smaller as well. At 5% of a 200mW limit (driven by battery life), this gives us 5mW for the channels.

The requirement for (approximately) 50mm^2 chips leads to link lengths of 7mm. At our benchmark of 0.25mW/Gbps/mm, this leads to a total bandwidth on the chip of only 2.8Gbps. This is a remarkably small number for future systems; it's only slightly higher than what is needed for a pair of HDTV video feeds, and almost certainly inadequate for tomorrow's computing requirements.

Research agenda

Based on the previous discussions, we felt that the key constraint was power, both for communicating data across channels as well as for storage and switching in the network routers. In addition, memory scaling trends make the prospect of a large, reliable, and fast memory distributed across an on-chip network difficult.

Many researchers have examined the issues of low-power communication on wires, with several publications and test chips built to explore these circuits. Typically, they reduce power by reducing voltage swing on the wires. This is an important on-going area of research, not only for basic circuit design issues, but also to enable a CAD and design ecosystem and infrastructure. Without that support, low-power wire circuits will never be accepted and generally used in ASIC design flows. They need to be a "drop-in replacement" to be of high value to the design community.

Another area of research would integrate multiple chips together in a 3D (or at least 2.5D) stack. By breaking apart a wide, single, monolithic chip into a stack of many smaller chips, total routes can be made significantly shorter, saving both total latency as well as total power. Several companies and universities are currently examining issues related to chip stacking and vertical integration, but improving on-chip network design using such 3D stacking represents a research gap today.

Using photonics on chips represents another research vector. Optics have achieved traction in chip-to-chip communication paths, but not yet for on-chip environments, in part due to integration difficulties and also due to the costs of translating between optical and electrical domains. However, given their extremely low latency—15 to 20X faster than repeated wires—optics on chips represent an intriguing area of open research for building very low-latency routing networks.

Finally, we might also consider re-optimizing basic technology parameters, such as the metal stackup in modern processes. For on-chip routing networks, with a preponderance of long wires and a relative dearth of transistors (at least compared to modern microprocessors), we may benefit from trading off dense, higher-capacitance lower metal layers in exchange for lower-capacitance but coarser upper metal layers. Similar tradeoffs may show up as we re-examine underlying technologies specifically for routing networks.

Evaluation and Driving Applications for On-Chip Networks

The working group comprised of 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 working group decided to look at this topic by looking at the following sub-problems: a) identify the applications/workloads which are likely to place special requirements on the interconnect, b) characterize them from computer architectural and programming model perspectives, c) derive the on-chip network requirements from the characterization, d) identify a research agenda from the requirements.

We identified two distinct application environments affecting the networks: one for CMP- and the other for System-on-Chip (SoC)-style architectures. Further, the former pushes technology limits in terms of need for high bandwidth, low latency under power- and area constraints while the latter calls for improved design styles with "integratable" IP blocks and many types of engines on chip.

Applications

We identified the following classes of applications for CMP style architectures: A) Datacenter: traditional ones such as transaction processing workloads (TPC-C, -H, etc.), webserver, etc. CMPs would further drive the need for server consolidation on to the single socket, provided the main memory bottleneck problem (arising from limited pinout on the chip) is solved. We purposely chose to consider this problem to be an orthogonal one. B) HPC (High Performance Computing) – the field of these applications has expanded to include real time simulation, financial applications (e.g., options trading), bioinformatics, in addition to the traditional ones of interest to science and engineering. C) RMS (Recognition, Mining, Synthesis): We expect these to form an *important and emerging* class given, for example, the increased emphasis on security. An example would be the *recognition* of multiple faces going through a security checkpoint. *Mining* could take the form of text, image, or speech search to match a scenario. Mined data could be *synthesized* to create new models. D) Healthcare – MRI, being an example – typically, instances where imaging and 3D are involved. E) Desktop/laptops: Traditional uses with extended use for multiple video streams and games (including multi-player and multi-site). For SoC style architectures, the class of embedded applications would enlarge to encompass – a) the PDA being used for games (single, multi-player, multi-site), videos, image search,etc., b) use in healthcare by medical personnel with real-time needs (for e.g., at patient beds)

Architectural Characterization / Programming Models

How do these applications affect the on-chip network? For that one needs to understand the type and nature of traffic that is demanded by these apps (access/sharing patterns and communication/synchronization). The access patterns seem to cover both heavily cacheable traffic (read-only, read-write sharing), which places a burden on the on-chip interconnect, and streaming traffic from DRAM or I/O which places primary burden on the external interfaces (affecting pinout, which falls outside the scope of this workshop) and a secondary burden on the on-chip network. Further, there are the usual problems such as bursty traffic – there are well studied mechanisms for congestion management but what overhead do they entail?

With increasing integration, we expect that the CMP will have additional agents (other than the CPU, cache, memory controller) such as specialized engines that have special traffic needs not falling into cache-line sized accesses, for example. The interconnect has to efficiently support *diverse* traffic patterns. This need is further

exacerbated in SoC architectures where a large number of diverse IP blocks is the rule, rather than the exception.

The team saw the possible need to support synchronization/communication primitives in the network for both coherence style traffic (e.g., to efficiently broadcast and collect invalidations at the home nodes) and message passing style traffic (e.g., to broadcast data). In the former scenario, with hundreds of processing elements, even directory based systems would not scale without such support in the interconnect.

With the integration of specialized engines, in addition to different traffic patterns, there is the need for QoS (quality of service) guarantees or even soft real time constraints. With server consolidation workloads, the single CMP has to be dynamically partitioned into several systems but with the need to support performance isolation (one partition's traffic should not affect the performance of another partition) and fault isolation (a partition going down should not bring down another partition). Finally, security concerns require that different parts of the system which are running separate applications be effectively isolated. Interestingly, all these three seemingly diverse scenarios have a commonality from the on-chip network's perspective. Since the network is *shared*, all these scenarios require some form of network isolation – either virtual or physical.

We expect that CMPs will support a "mixed-mode" of programming, i.e., both coherent shared memory and message passing) because of the range of applications that would run. Clearly, this would require efficient support in the interconnect for both cache-sized line transfers and variable length message transfers.

Network Requirements / Evaluation Metrics

Based on the architectural characterization, the team came up with what needs to be supported in the network:

- Efficient data transfer support at various granularties for coherent and message-passing paradigms and for different types of specialized engines. SoC architectures, also need this support, although the granularities will vary.
- Support for partitioning. This includes QoS guarantees (through separate virtual channels or completely partitioned sub-networks), performance isolation (so that partitions do not share routing paths, for e.g.), isolation for security (through partitioned sub-networks, for e.g.).
- Clean, efficient, and common network interfaces to support multiple programming models.
- Possible support for synchronization/communication primitives such as multi-cast, barriers (interestingly, there was a poster paper in the workshop which pushed coherence decisions into the network)

The team also felt strongly that it is very important that a fairly good set of evaluation metrics be defined –the usual latency, bandwidth measures have to be evaluated under power, energy, and thermal constraints, under area constraints (Silicon is at a premium and on-chip network area is of concern). In addition, there is also a need for standardization of metrics – for example, one team member (Drew Wingard) pointed out that even the term "network latency" is used in so many ways as to render any two comparisons with skepticism.

Research Agenda

Applications will demand both network performance and functionality. Many techniques that are well studied for off-chip networks can be applied to on-chip networks. Given that the range of applications that CMPs and SoCs and, consequently, the on-chip networks need to support are so diverse, *how we support them efficiently under power, energy, thermal, and area constraints is the key challenge*.

Much of the research agenda falls out of the listing of the topics under "Network Requirements". A few additional research issues and a more general call to action items are mentioned below:

- With the need for dynamic partitioning (and fault tolerance arising from process variability/reliability – a topic addressed by another group), the network topology does not remain static. As a result, subnetworks which have different topologies than the static one are created on the fly. What is the support needed at the hardware and system software level to support such reconfiguration?
- Development of analytical models to predict the real time guarantees of the (SoC) architecture being designed.
- Monitoring network performance under constraints (for e.g., once the network utilization has crossed a threshold, how does a particular class of traffic behave?) so that one can study the effectiveness of network wide policies.
- Tools: We expect that realistic full system simulation, especially execution based, will not be possible given the current set of tools and methodologies. Many groups in both academia and industry are resorting to emulation through the use of FPGAs to overcome the simulation speed problem. A concerted effort across multiple research disciplines in computer engineering is required for a realistic study of workloads on CMPs and SoCs.
- There is also a need to define a suite of workloads/benchmarks for comparing different systems. The suite should also specify the *mix of workloads to be run concurrently* and should provide common evaluation criteria for comparison. This requires a concerted effort by groups in academia and industry interested in CMP and SoC architectures.

Design Tools for On-Chip Networks

The working group comprised of Luca Benini (U. of Bologna, Italy), Mark Hummel (AMD), Olav Lysne (Simula Lab, Norway), Li-Shiuan Peh (Princeton University, USA), Li Shang (Queens University, Canada) and Mithuna Thottethodi (Purdue University, USA). Seven research challenges were outlined as critical research challenges in the development of design tools for on-chip networks targeting both many-core processor chips as well as systems-on-a-chip.

1. The interface of network synthesis with system-level constraints and design. As chips move towards multi-core and many-core in future technologies, system-level constraints become increasing complex and requirements more multi-faceted. It is essential for the on-chip network synthesis tool to be able to interface effectively with these. The foremost challenge lies in the accurate characterization and modeling of the system traffic – such as that imposed by a shared memory system on-chip, or a platform-specific chip. In both general-purpose and embedded domains, heterogeneous chips are

becoming increasingly likely, making synthesis tougher. There is also a need for synthesis tools to handle both hard and soft constraints within the same framework.

2. Hybrid custom and synthesized tool flow. With general-purpose processors typically leading the embedded market with aggressive, innovative microarchitectures and designs that are custom-designed, it is critical for design tools to be able to leverage these high performance designs within the existing tool flow to ease adoption into mass market embedded devices. Questions arise as to whether specialized libraries for networks can be constructed and how that permeates throughout the entire CAD tool flow. This is particularly important to facilitate fast transfer of research into industry products, benefiting the mass market.

3. Design validation. A critical hurdle in the deployment of on-chip networks lies in validation of its operation – how designs can be ensured to be robust, in the face of process variations and tight cost budgets.

4. Impact of CMOS scaling and new interconnect technologies. There is a clear need for new timing, area, power, thermal and reliability models for future CMOS processes, circuits and architectures, so design tools can be effective as CMOS scales. New interconnect technologies need to factor this in order to ease adoption – models and libraries should be made available together with proposals of new interconnects. Proposed modeling infrastructure should also be extensible to ensure the plugging in of these new technologies and interconnects.

5. End-user feedback design tool chain. As the scale and complexity of networks increases, there is a need for design tools that feed back to the designer and aid them in the design. For instance, network characteristics can be fed back to the designer to allow him to quickly iterate. Research in this domain can potentially leverage the end-user feedback design toolchain research in other network domains such as the Internet, though there are clearly substantial differences in the requirements for the specific domain of on-chip networks.

6. Dynamic reconfigurable network tools. Not only do general-purpose many-core chips need to support a wide variety of traffic and applications, networks-on-chips in MPSoC platforms increasingly have to support a large 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, in a fashion 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. This is no longer tenable for future large-scale networks and systems. There is thus a need for research into analytical methods: such as formal methods and queuing analysis-based tools for estimating network power-performance. While prior research can be leveraged, the key distinct features of on-chip networks (such as physical constraints, link-level flow control) make it necessary to explore new analysis approaches.

In short, we see the above challenges critically impacting the immense embedded MPSoC market as well as the general-purpose computing market (Challenges 2, 3, 4, 5 are particularly relevant to the general-purpose market). Overcoming these challenges will enable complex, correct network designs that will otherwise be impossible, and facilitate the adoption of on-chip networks.

System and Micro Architecture for On-Chip Networks

The working group on system architectures consisted of 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 working group on microarchitectures included of Luca Carloni (Columbia University), Steve Keckler (The University of Texas at Austin), Robert Mullins (Cambridge University), Vijay Narayanan (Penn State), Steve Reinhardt (Reservoir Labs), and Michael Taylor (UC San Diego).

Collectively, we identified latency and power as the most critical technical challenges for on-chip networks. The groups also discussed several other important research directions, including programmability, managing reliability and variability, and scaling on-chip networks to new technologies. We found that latency and power are cross-cutting issues that span these other areas of research and must be considered in all aspects of on-chip network design.

Latency

Minimizing latency in on-chip networks is critical as these networks will likely be used as replacements for chip-level bus interconnects that have typically been small in scale and low in latency. Low latency networks makes the system designer's and the programmer's job easier as low overhead reduces the need to avoid communication and the effort to exploit concurrency.

Efficient light-weight on-chip network interfaces are critical for overall latency reduction, as the transmission time on the wires and in the routers in today's networks is often dominated by the software overheads into and out of the networks. We see a need for thin network abstractions that expose hardware mechanisms that can be used by application-level programmers. These networks should be tightly coupled into the compute or storage elements that attach to them, but should be general purpose to provide portability and utility across different use scenarios. We also discussed virtualization of the network interface to provide atomicity and security, but recognize that such interfaces cannot unduly add to latency. While research on such topics as remote queues and automatic method invocation on message arrival have previously been proposed, more research on the hardware and software side of network interfaces is needed.

Router architecture and microarchitecture innovations are necessary to reduce latency of on-chip networks, but without blowing out area or power budgets. Reducing the number of pipeline stages in the router is clearly critical, as is congestion control with bounded or limited router buffering. We suggest new research into flow control algorithms and microarchitectures that identify and accelerate critical traffic without substantially affecting the latency of less critical traffic. Better support in networks and interfaces for out-of-order message delivery could provide new capabilities in this arena.

The abundance of on-chip wires may reduce the importance of virtual channels, as traditional package pin-limits do not exist. Networks that can exploit some form of static or stable information from the application may be able to provide faster service without increasing power consumption. One interesting avenue of research is a reconsideration of circuit switched networks or a hybrid packet/circuit switched network if the circuit configuration time can be kept small. While speculation during routing has been used to reduce router latency toward a single cycle, more research is needed in speculative microarchitectures to improve accuracy and efficiency. Finally, while most on-chip networks to date have been implemented as meshes, the jury is still out on the right network topology.

Power

We recognize that not all systems that employ on-chip networks operate at the same power/performance point. For example, high-performance systems need fast on-chip networks for high-speed data transport, while embedded systems have tight power budgets and value on-chip networks more for their ability to easily connect IP from different designers or vendors. The research community should acknowledge these differences and pursue research that solves problems in both domains. Because power envelopes for both high-performance and embedded systems are tightly constrained, power has clearly become an item that must be budgetted and traded off among different parts of the system. These tradeoffs expose areas for research such as adaptive power management for networked systems that can shift power between computation and communication on an application or application phase basis. Techniques such as dynamic voltage and frequency modulation in the network could prove fruitful. All innovations in architectures and microarchitectures that reduce latency must take power into consideration.

Programmability

To make effective use of a concurrent SoC or multicore system, a programmer needs (1) a fast and robust on-chip network transport, (2) fast and easy-to-use network interfaces, and (3) predictable network performance and a means to reason about it. 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. While network microarchitectures should also be scalable across generations of systems, one related challenge is how to interface on-chip networks to off-chip, board-level, rack-level, and system-wide networks. Unifying the protocols across these different transport layers may make them easier to build and easier for programmers to reason about. Another question for further research is how much intelligence to push into the network. Recently researchers have discussed incorporating support for cache coherence in the network layer; other possible areas include security and encryption. Whether breaking down such 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 research questions.

Today's networks are effectively black boxes to programmer, and programmers find it very difficult to reason about network bottlenecks when writing and optimizing their programs. To rectify these problems, we recommend research efforts into network modeling and measurement for use by application programmers. Network modeling means developing cost models for network latency under different traffic patterns and loads that a programmer can use to predict how an application will perform. We might find that a precondition to such models are network architectures that behave in a predictable fashion; it might actually better to sacrifice some network performance on stochastic loads to achieve more predictable network behavior. Measurement means hardware, such as performance counters, in the network and tools that can synthesize the measurements into feedback that a programmer can use to understand how an application uses the network. Many tools have been developed over the last decade to help a programmer understand program performance on a uniprocessor. It is time to embark on such tools for on-chip networks.

Reliability/Variability

With shrinking transistor and wire dimensions, reliability and variability have become significant challenges for designers of integrated circuits. While past research has examined methods to provide network reliability, on-chip networks will need new light-weight means for link-level and end-to-end guarantees of service. One example is self-monitoring links and switches that detect failures and intelligently reconfigure themselves. Power, latency, and area-efficient error tolerant designs will be required to provide useful on-chip network infrastructure in both the high-performance and embedded spaces.

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

Another form of variability arises from the different types of traffic delivered by different applications or by different phases of the same application. Variations may manifest in message length, message type (data, synchronization, etc.), message patterns (regular streams, unstructured, etc.), and message injection rates (steady vs. 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. While wire count constraints are less important on-chip, smaller feature sizes affect the relative of cost of communication and computation. Faster computation relative to wire flight time makes viable more intelligent routing algorithms

designed to minimize both 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 one of many possible post silicon technologies, become viable, new opportunities and constraints will further drive the need for innovation in interconnection networks. We recommend early investment into characterizing changing and emerging technologies from the perspective of on-chip networks as well as into new network designs motivated by such shifts in technology.

Recommendations

Based on the gap analysis performed by the study groups, we recommend that NSF start an aggressive research program to close the identified gaps. On-chip interconnection networks are a critical technology that is required to enable both future many-core (CMP) processors and future SoCs for embedded applications. To make sure that this technology is in place when needed, the following key research tasks should be performed:

1. **Low-Power Circuits for On-Chip Interconnection Networks:** To close the power gap, research is needed to develop low-power circuits for channels, switches, and buffers. If successful, this research will reduce the power needed by on-chip networks by an order of magnitude, allowing it to fit in expected power envelopes for future CMPs and SoCs.
2. **Low-Latency Network and Router Architecture:** To make the latency of on-chip interconnection networks competitive with dedicated wiring research is needed to reduce the delay of routers (possibly to one cycle), and to reduce the number of hops required by a typical message. Circuit research to reduce the latency of channels may also be valuable in closing this gap. If successful, this work will enable on-chip networks to match the latency of dedicated wiring.
3. **Encapsulating on-chip Network Components:** To make on-chip network technology accessible to SoC designers, this technology must be encapsulated in a way that is compatible with standard CAD flows (e.g., as parameterized hard macros). Tools to automatically synthesize on-chip networks from these macros (as well as blocks of standard logic) are also needed to make the technology accessible. If successful, this research will remove one of the largest roadblocks to adoption of on-chip networks in SoCs.
4. **Develop Prototype On-Chip Networks**: To expose unanticipated research issues and to serve as a baseline for future research, and as a test bed for new on-chip network components, optimized prototype on-chip networks should be designed, constructed, and evaluated. This work will also serve as a proof of concept for on-chip networks, reducing their perceived risk and facilitating transfer of this technology to industry.
5. **Develop Standard Benchmarks and Evaluation Methods**: To keep on-chip interconnection network research focused on the real problems, a set of standard benchmarks and evaluation methods are needed. Standard benchmarks also allow

direct comparison of research results and facilitate exchange of information between researchers in the field.

Acknowledgment

This workshop was made possible by the generous support of NSF through the Computer Architecture Research and Computer Systems Research programs and from the University of California Discovery Program. Workshop co-chair John Owens contributed to all aspects of workshop organization and content. Our steering committee, which included Li-Shiuan Peh, Timothy Pinkston, and Jan Rabaey set workshop direction and made many suggestions that led to the success of the workshop. Jane Klickman expertly handled all workshop logistics.

List of Attendees

Albonesi, Dave	albonesi@csl.cornell.edu	Cornell
Balasubramonian, Rajeev	rajeev@cs.utah.edu	Univ. of Utah
Benini, Luca	lbenini@deis.unibo.it	University of Bologna
Bergman, Keren	bergman@ee.columbia.edu	Columbia
Bilas, Angelos	bilas@csd.uoc.gr	University of Crete
Binkert, Nathan	binkert@hp.com	HP Labs
Bolsens, Ivo	ivo.bolsens@xilinx.com	Xilinx
Borkar, Shekhar	shekhar.y.borkar@intel.com	Intel
Carloni, Luca	luca@cs.columbia.edu	Columbia
Cheng, Chung-Kuan	kuan@cs.ucsd.edu	UCSD
Chien, Andrew	andrew.chien@intel.com	Intel
Cohen, Danny	danny.cohen@sun.com	Sun
Conway, Pat	pat.conway@amd.com	AMD Principal Member of Technical Staff (Sunnyvale)
Dally, Bill	billd@csl.stanford.edu	Stanford
Darema, Federica	fdarema@nsf.gov	NSF Program Manager
Das, Chita	das@cse.psu.edu	Penn State
Duato, Jose	jduato@disca.upv.es	UPV
Ebergen, Jo	jo.ebergen@sun.com	Sun
Foster, Michael	mfoster@nsf.gov	NSF
Harrod, Bill	William.Harrod@darpa.mil	DARPA
Hiller,Jon	jhiller@stassociates.com	Science and Technology Associates
Ho, Ron	ronald.ho@sun.com	Sun
Horowitz, Mark	horowitz@ee.stanford.edu	Stanford
Hummel, Mark	mark.hummel@amd.com	AMD Fellow (Boston)
Jayasimha, Jay	jay.jayasimha@intel.com	Intel
Katevenis, Manolis	katevenis@ics.forth.gr	University of Crete
Keckler, Steven	skeckler@cs.utexas.edu	U.T. Austin
Kozyrakis, Christos	christos@ee.stanford.edu	Stanford
Kubiatowicz, John	kubitron@cs.berkeley.edu	Berkeley

Kundu, Partha	partha.kundu@intel.com	Intel
Lockwood, John	lockwood@arl.wustl.edu	Washington University
Lysne, Olav	olavly@simula.no	Simula Research Labs,Oslo
Mora, Gaspar	gmora@gap.upv.es	UPV
Mullins, Robert	Robert.Mullins@cl.cam.ac.uk	Cambridge
Narayanan, Vijay	vijay@cse.psu.edu	Penn State
Oehler, Rich	rich.oehler@amd.com	AMD Corporate Fellow (New York)
Owens, John	jowens@ece.ucdavis.edu	UC Davis
Panda, D.K.	panda@cse.ohio-state.edu	Ohio State
Patra, Darshan	priyadarsam.patra@intel.com	Intel
Peh, Li-Shuan	peh@ee.princeton.edu	Princeton
Petrini, Fabrizio	fabrizio.petrini@pnl.gov	Pacific Nat'l Labs
Pinkston, Timothy	tpinkston@nsf.gov	NSF/USC
Reinhardt, Steve	stever@eecs.umich.edu	Michigan
Shang, Li	li.shang@queensu.ca	Queens University, Canada
Silla, Federico	fsilla@disca.upv.es	UPV
Taylor, Michael B.	mbtaylor@cs.ucsd.edu	UCSD
Thottethodi, Mithuna	mithuna@purdue.edu	Purdue
Vaidya, Aniruddha	aniruddha.vaidya@intel.com	Intel
Wingard, Drew	wingard@sonicsinc.com	Sonics
Yalamanchili, Sudhakar	sudha@ece.gatech.edu	Georgia Tech
Yoon, Barbara	barbara.yoon@darpa.mil	DARPA