Guest Editors' Introduction: The Network-on-Chip Paradigm in Practice and Research

André Ivanov
University of British Columbia

Giovanni De Micheli
École Polytechnique Fédérale de Lausanne

■ **It is our pleasure to introduce** this special issue on networks on chips (NoCs). Large, complex multiprocessor-based SoC platforms are already well into existence, and, according to common expectations and technology roadmaps, the emergence of billion-transistor chips is just around the corner. The complexity of such systems calls for a serious revisiting of several on-chip communication issues. In this special issue, we focus on an emerging paradigm that effectively addresses and presumably can overcome the many on-chip interconnection and communication challenges that already exist in today's chips or will likely occur in future chips. This new paradigm is commonly known as the *network-on-chip paradigm*. The articles featured in this issue come from outstanding experts from around the world, from both industry and academia. Together, the articles reveal and discuss a wide range of issues specifically pertinent to NoCs. They also provide perspective based on actual practice, as well as more-speculative perspectives. To achieve a good degree of self-containment in this issue, we've included a more tutorial/survey type of article to lead a group of four specific and detailed articles.

The NoC paradigm is one, if not the only one, fit to enable the integration of an exceedingly large number of computational, logic, and storage blocks in a single chip (otherwise known as a SoC). Notwithstanding this school of thought, the adoption and deployment of NoCs face important issues relating to design and test methodologies and automation tools. In many cases, these issues remain unresolved.

On-chip interconnection network

Set-top boxes, wireless base stations, high-definition TV, and mobile handsets are just a few applications that have arisen because of multiprocessor SoCs. With such chips, the constraints for performance, power consumption, reliability, error tolerance and recovery, cost, and so forth can be extremely severe. One design characteristic that lies at the core of all these critical specifications is the on-chip interconnection network. Many experts advocate regularity in such networks as opposed to continuing with the more traditional ad hoc networks that have evolved over the past decades of IC design. Hence, much research and practical interest has recently focused on regular networks implemented on chip, often influenced by the parallel-computing field. When integrated on chip in the form of micronetworks, these regular networks are referred to as NoCs.

Effective on-chip implementation of network-based interconnect paradigms requires developing and deploying a whole new set of infrastructure IPs and supporting tools and methodologies. For example, NoCs require switches and router blocks, as well as corresponding communication formats and protocols. The design complexity of conventional SoCs is already soaring, so it's understandable that the development of SoCs based on nontraditional models might at first appear overwhelming and hence unnecessary or undesirable. However, when the specifications of these systems reach levels at which traditional methodologies and architectures are incapable of meeting the requirements, system architects and project managers obviously have no recourse except

0740-7475/05/$20.00 © 2005 IEEE

Timing is right for GALS SoC design

Philippe Martin, Arteris

Moore's law's relentless pace, allowing the integration of many IP blocks on a single SoC, increases on-chip communication bandwidth requirements and drives up the operating frequency of the connections between IP blocks. At the same time, shrinking process geometries reduce the cross section of wires: The wires' resistance per length increases while their capacitance does not scale down accordingly. Compared to gates, wires are becoming slower, as Figure A illustrates.

Bus-based, synchronous communication structures in SoCs that are larger than 10 × 10 mm and operate at several hundreds of megahertz have tight timing constraints, slow wires, and require tight clock-skew control. The combination of these obstacles creates timing closure issues that are increasingly difficult to fix on the global wires between IP blocks.

At the same time, SoC devices with dozens of clocks are already today's reality. Power management requires turning off or slowing down a device's temporarily inactive functions while keeping its critical functions active and at full speed. In addition, many IP blocks have their own natural clock rate, driven by the components with which they interface (DRAM chips, for example) or by the real-time data flows they serve (such as display controllers and I/Os).

Considering the timing closure issues, there is no point in maintaining globally synchronous bus protocols between cores that, in the end, do not share the same clock. Just as on-board buses—such as peripheral component interconnect (PCI)—have moved to a point-to-point high-speed network implementation (PCI-Express), on-chip communications will transition to point-to-point networks on chips (NoCs) between locally clocked IP blocks or subsystems. This is the globally asynchronous, locally synchronous (GALS) paradigm.

NoCs use layered communication protocols, decoupling the physical transmission of bits through point-to-point wires from higher-level aspects, such as IP socket transactions. The communications network consists of switching elements (SEs) that route packets between IP components that implement transactions. The network protocol makes no assumption about the granularity of the clock distribution or even the existence of clocks.

Although using no clocks in an SE would seem to be the most radical way of solving the timing closure issues, clockless logic is slower (because of the necessary handshake procedures) and consumes more area (because of the necessary redundancy) than its clocked equivalent. A much more efficient GALS approach uses clocked SEs and IP components, assembled into larger synchronous clusters:

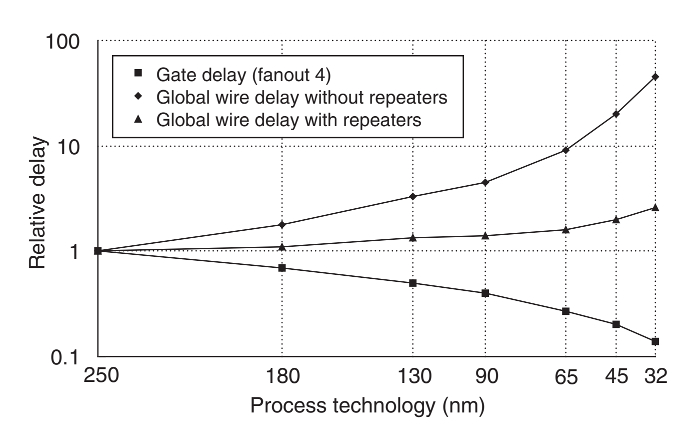

Figure A. Relative evolution of wire and gate delays (source: *2003 International Technology Roadmap for Semiconductors,* Sematech, 2003).

novel approaches and architectures. As Martin points out in the "Timing is right for GALS SoC design" sidebar, NoCs are an example of such newer (and arguably enabling) solutions. Nevertheless, others will argue that NoCs are only "solutions in waiting" at this stage because of the lack of maturity in the paradigm itself and its associated tools.

This special issue illustrates how, to date, engineers have successfully deployed NoCs to meet certain very aggressive specifications. At the same time, the articles reveal many issues and challenges that require solutions if the NoC paradigm will indeed become a panacea or quasi-panacea for tomorrow's SoCs. Given the many issues at stake, we believe this special issue will shed important and relevant light on this emerging novel paradigm.

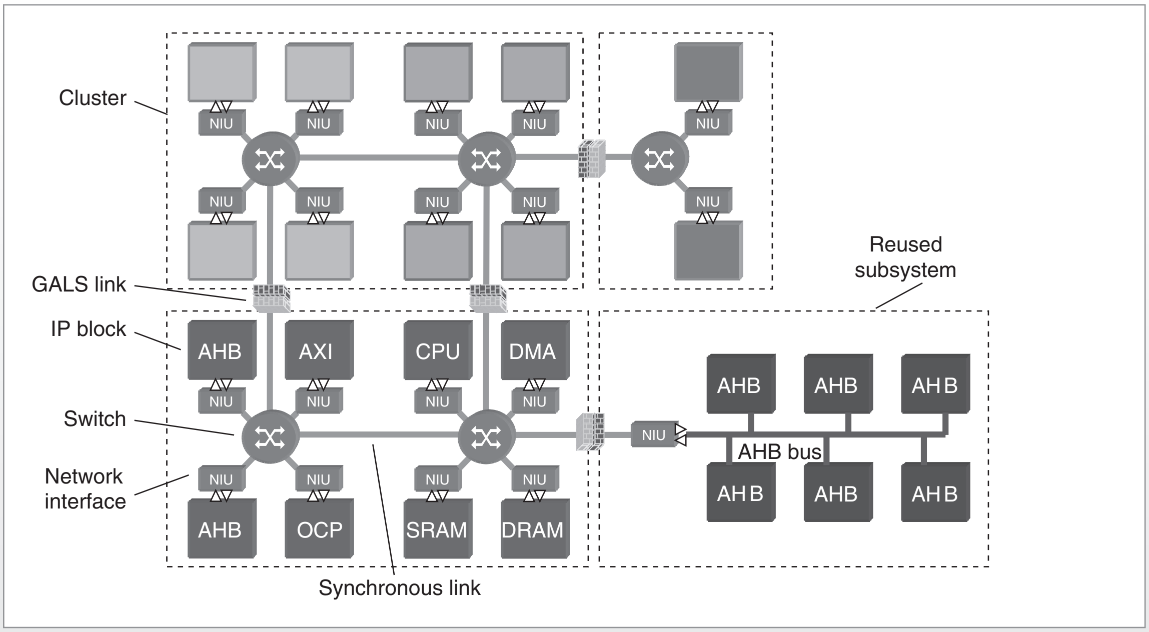

Figure B. Example NoC-based SoC structure.

- Clusters share the same clock, and designers adjust SE pipeline stages to the cluster's clock frequency. This clustering minimizes latency—a critical factor between tightly coupled components, such as processors and their associated memories.
- An asynchronous link can connect an individual IP block to its associated SE. Designers can then finely optimize the IP block's operating frequency (for power consumption, for example).
- Connecting clusters with asynchronous links eliminates the need for timing closure at the SoC level. A synthesis-friendly implementation makes asynchronous links compatible with ASIC design flows.

Figure B illustrates the structure of such a NoC-based SoC.

This GALS approach guarantees SoC-level timing closure while minimizing latency and increasing throughput on application-specific critical communication paths. Using these techniques, Arteris has demonstrated overall operating frequencies exceeding 750 MHz on a 10 × 10-mm chip split into nine clusters. Arteris designed the chip using Artisan's standard-cell libraries and standard synthesis-based design flow with TSMC's 90-nm process.

Contrary to centralized and synchronous techniques, the globally asynchronous NoC architecture readily scales with increasing SoC complexity and process changes. Once adopted, NoCs and GALS systems will be here to stay.

Philippe Martin is the product marketing director of Arteris. Contact him at philippe.martin@arteris.com.

This special issue

The five articles in this special issue cover NoC-related issues in existing practice as well as more advanced issues in research. The first article, "Design, Synthesis, and Test of Networks on Chips" by Pande et al., somewhat sets the stage for the following articles. Tutorial- or survey-like in nature, this article reviews the state of the art in NoC technology in terms of design, automatic synthesis, and post-manufacturing testing—stressing both challenges and emerging solutions to those challenges. The authors conclude their comprehensive survey by reiterating their belief that the NoC communication paradigm constitutes an enabling solution for large, complex SoCs consisting of many embedded functional and storage blocks. They also remark that the issues under current debate regard design trade-offs and performance optimizations—largely the subject of the subsequent articles.

The next article, "Æthereal Network on Chip: Concepts, Architectures, and Implementations," very clearly presents the state of the art in industry. Goossens, Dielissen, and Rădulescu describe the Æthereal NoC, a specific NoC developed at Philips Research Laboratories, which aims at high-performance multimedia embedded systems. They describe the rationale and concepts underlying this specific network, illustrating how it can meet quality-of-service specifications for real-time applications or high resource utilization objectives for best-effort service applications. For example, they describe a contention-free routing based on time-division-multiplexing slots that, in turn, offer throughput and latency guarantees. The authors discuss in detail how to address this optimization problem in the design flow; they also describe how to program the slot allocations at runtime, assuming two different programming models. The authors then describe router architectures and implementations and illustrate the trade-offs between them. Ultimately, they argue that using the Æthereal NoC allows system architects to evaluate and trade off programming models, performance, and cost to achieve balanced SoC solutions.

The third article, "Analysis and Implementation of Practical, Cost-Effective Network on Chips" by S.-J. Lee, K. Lee, and Yoo, is also very real in the sense that it focuses on the design and evaluation of real ICs based on existing practice. The authors summarize the main features and design issues of three NoC-based ICs, describing the fabrication of ICs for mesochronous communications, high-speed serialization, and programmable synchronization. The authors discuss the design trade-offs they made and the resulting cost and performance. The basic architecture they used for their chips is based on a star topology; they also considered the mesh architecture. Based on their findings, the authors argue that this topology is the most cost-effective when used as a local network architecture. Furthermore, when combined with low-power link schemes, the star topology can also be useful as the basis for a global network architecture. The authors also describe in detail the trade-offs regarding packet format, size, and corresponding protocols.

The fourth article, "Analysis of Error Recovery Schemes for Networks on Chips" by Murali et al., focuses on the critical issue of reliability, or more specifically, that of error recovery schemes to ensure reliable operations of high-performance SoCs. Decreasing transistor sizes and reduced voltage swings for power consumption minimization and speed, combined with a multitude of noise sources create a vast range of possible logic and timing errors. Ensuring that high performance and highly reliable systems and applications do not suffer from such errors requires incorporating resilience. Hence, Murali et al. analyze different error recovery schemes for NoCs, including end-to-end, switch-to-switch, and hybrid error detection/correction schemes. They focus the comparison on power consumption, error detection capability, and impact on network performance with the overall objective of equipping designers of complex SoCs with a set of well-parameterized alternatives for meeting reliability and performance specifications.

In the last article, "Dynamic Interconnection of Reconfigurable Modules on Reconfigurable Devices," Bobda and Ahmadinia extend the NoC paradigm to that of a dynamic one—a dynamic NoC (DyNoC). That is, they look at NoCs from a different perspective, namely that of dynamically reconfiguring the interconnections of modules on reconfigurable devices like FPGAs. The authors present two approaches addressing this problem of online reconfiguration. First, they present a circuit-routing solution based on the concept of a reconfigurable multiple bus on chip (RMBoC) but argue that this might be a practical solution only for today's column-wise reconfigurable FPGA devices. They present a second solution that targets devices with unlimited reconfiguration capabilities. This second approach, DyNoC, is a more generic 2D dynamic model. Bobda and Ahmadinia demonstrate the feasibility of the approaches through analysis and examples. In their conclusion, the authors acknowledge that issues regarding clearing certain regions of the network require further investigation.

We sincerely hope you will enjoy this special issue and that it will meet your needs and expectations. We believe the NoC topic will continue to be of growing interest and importance for a wider range of researchers and practitioners. We anticipate this issue will remain an important reference publication for future research and advancements in this field and closely related fields.

We enjoyed working on the shaping of this issue, and we acknowledge that the high-quality, dedicated efforts of many individuals made this special issue possible. We thank all those who contributed. In particular, we are grateful to the authors for their original contributions, as well as to the reviewers who contributed to significant improvements of the submissions through their detailed and critical feedback. We wish to thank Rajesh Gupta, editor in chief of *D&T*, for supporting and helping us create this special issue for the past several months. We also owe very special thanks to the editorial staff of the IEEE Computer Society for their fine job in editing and assembling this issue.

We will be happy to answer your questions about the articles in this special issue or to direct your questions to the individuals most fit to answer your queries. Happy reading! ■

André Ivanov is a professor in the Department of Electrical and Computer Engineering at the University of British Columbia. His research interests include IC testing; design for testability; and BIST for digital, analog, and mixed-signal circuits and SoCs. Ivanov has a BS, an MS, and a PhD, all in electrical engineering, from McGill University. He is the chair of the IEEE Computer Society's Test Technology Technical Council, a senior member of the IEEE, a Fellow of the British Columbia Advanced Systems Institute, and a Professional Engineer of British Columbia.

Giovanni De Micheli is a professor and director of the Integrated Systems Center at École Polytechnique Fédérale, Lausanne, Switzerland. His research interests include design technologies for integrated circuits and systems, with particular emphasis on synthesis, system-level design, hardware-software codesign, and low-power design. He is a Fellow of the ACM and the IEEE, and recipient of the 2003 IEEE Emanuel Piore Award.

■ Direct questions and comments about this special issue to André Ivanov, Dept. of Electrical and Computer Engineering, Univ. of British Columbia, 2356 Main Mall, Vancouver, BC, V6T 1Z4, Canada; ivanov@ece.ubc.ca.

For further information on this or any other computing topic, visit our Digital Library at http://www.computer.org/publications/dlib.