
CHALLENGES AND PROMISING RESULTS IN NoC PROTOTYPING USING FPGAs

ALTHOUGH A SIGNIFICANT AMOUNT OF THEORETICAL WORK SUPPORTS THE POTENTIAL OF NoC ARCHITECTURES, SUCH RESULTS NEED TO BE DEMONSTRATED BY ACTUAL IMPLEMENTATIONS BEFORE THE NoC PARADIGM BECOMES A REALITY. BESIDES DEMONSTRATING THE FEASIBILITY OF THE OVERALL APPROACH, PROTOTYPING ENABLES ACCURATE EVALUATION OF POWER, PERFORMANCE, AREA, AND VARIOUS DESIGN TRADE-OFFS. THIS ARTICLE PRESENTS FOUR NoC PROTOTYPES, DISCUSSES THE CHALLENGES ASSOCIATED WITH THEIR DESIGN, AND ASSESSES THE POTENTIAL OF THE NoC APPROACH.

Umit Y. Ogras
Radu Marculescu
Carnegie Mellon
University

Hyung Gyu Lee
Samsung Electronics

Puru Choudhary
Diana Marculescu
Michael Kaufman
Peter Nelson
Carnegie Mellon
University

..... Recently proposed as a modular, scalable solution to on-chip communication, the network-on-chip (NoC) paradigm achieves communication among different cores by transferring messages through the network infrastructure, eliminating the need for ad hoc global wires.¹ In contrast to traditional bus-based and point-to-point architectures, NoC architectures provide far greater bandwidth with moderate area overhead.² Furthermore, the NoC approach offers matchless opportunities for implementing globally asynchronous, locally synchronous (GALS) designs, which make clock distribution and timing-closure problems more manageable.³

In this article, we present four NoC prototypes and discuss the implementation effort and challenges associated with their prototyping. The first two prototypes implement multimedia applications while allowing accurate power-performance and

area evaluations through direct measurements. The other two prototypes illustrate the use of application-specific, long-range links for NoC performance optimization through FPGA- and ASIC-style implementations.

NoC prototyping

The potential of the NoC approach has stimulated significant research work on NoC design methodologies.^{1,4-6} Researchers usually evaluate these methodologies through simulation using synthetic traffic generators. Although simulation is a flexible approach for power-performance evaluations under various network parameters, it relies on many assumptions and approximations that affect the accuracy of the results. On the other hand, most studies dealing with concrete NoC implementations lack performance evaluation under real driver applications. Instead, they rely on simple

traffic generators,⁵⁻⁷ so their performance results are of little practical value. Relying solely on traffic generators could also result in inaccurate predictions, because network performance and power consumption depends heavily on the characteristics of the communication traffic generated by the applications running on that network. Consequently, prototypes that implement real applications are necessary to make the results of various theoretical studies meaningful.

Prototyping entails building a functional system model that lets the designer evaluate various aspects of a design, illustrate the main ideas, and provide a realistic projection about the final product implementation. Indeed, having a prototype available not only provides more accurate power-performance evaluations, but also lets designers make evaluations beyond the usual capabilities of simulation. For example, prototypes make accurate area estimations feasible, as well as hardware overhead estimates of various components and energy consumption measurements. The 80-core teraflops chip recently introduced by Intel is a good example of a NoC prototype being used for real product development.⁸ Other examples are the TRIPS prototype,⁹ which demonstrates scalable processor and memory system technologies for nanoscale chips, and the pioneering Raw tiled multicore architecture.¹⁰

This article presents four NoC prototypes, discusses the implementation challenges, and assesses the potential of the NoC approach for on-chip communication, summarized in Figure 1. The first two prototypes demonstrate the use of the NoC paradigm for implementing multimedia applications, in particular the Motion JPEG (MJPEG) and MPEG-2 encoders. Through direct measurements on the actual prototypes, we show the viability of the NoC approach and its superiority to traditional approaches. Our third example demonstrates the effectiveness of application-specific, long-range links for NoC performance optimization through FPGA prototyping. Finally, we present an ASIC-style implementation of a 4×4 mesh

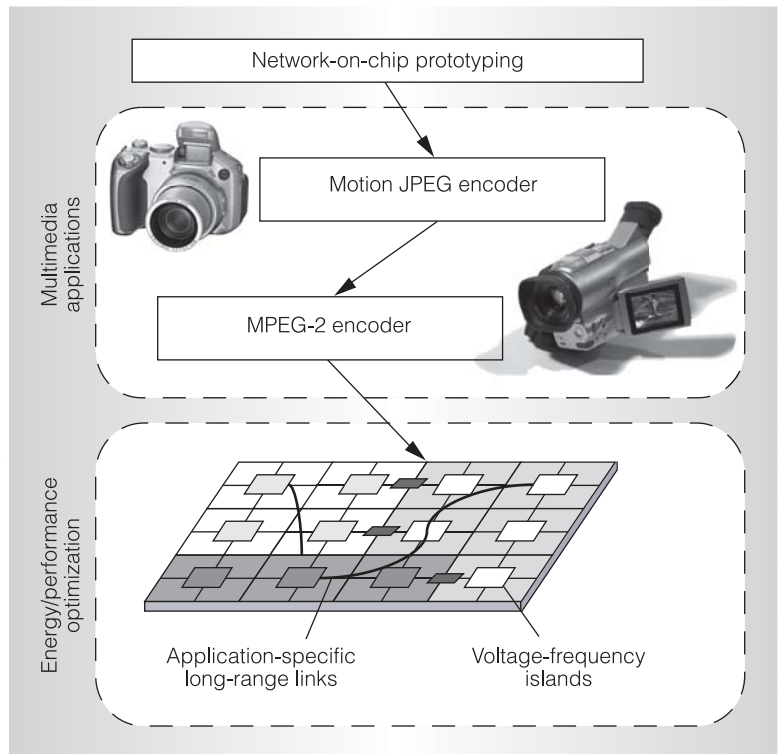


Figure 1. Overview of the network-on-chip (NoC) prototypes presented in this article.

network, with and without long-range links, to obtain a more accurate area evaluation.

Our prototyping work relies on resource reuse as an effective way to reduce design time and cost. For instance, we used the same on-chip router and standard network interface in all four prototypes. Using these building blocks, we could implement different network topologies and applications with minimal effort. Furthermore, we obtained various IP cores—such as RAM-based FIFO memories and discrete cosine transform, quantization, and zigzag-reordering modules—from the open Xilinx IP library (<http://www.xilinx.com/ipcenter/index.htm>). Using freely available cores saves a significant amount of time over custom implementations, which are challenging in terms of both design and test.

On-chip router design

On-chip routers are at the heart of NoC designs. Therefore, we first designed the output buffered on-chip router shared by the four prototypes we discuss in this article.

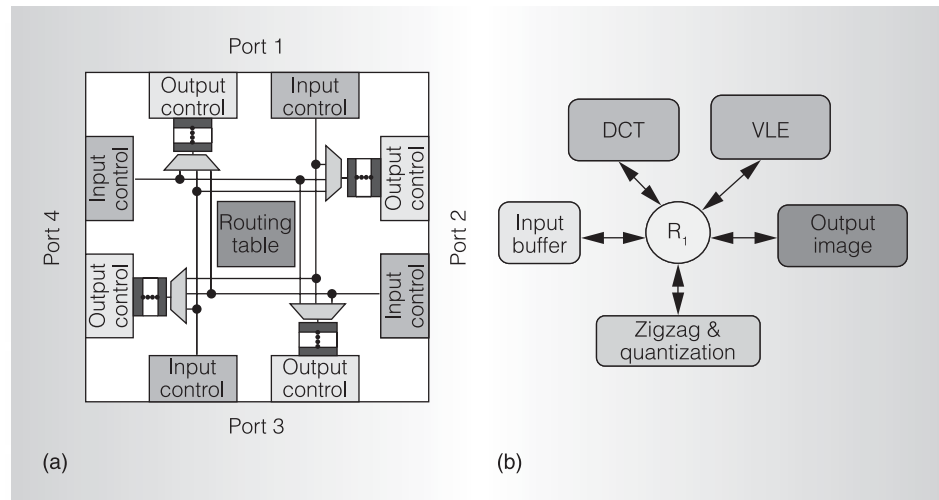


Figure 2. Simplified block diagram of the router (a), and NoC communication architecture of the motion JPEG (MJPEG) encoder (DCT: discrete cosine transform, VLE: variable length encoder) (b). (Implementation details of the router appear elsewhere.⁹)

Our design, whose simplified block diagram appears as Figure 2a, implements wormhole flow control because of its moderate buffer requirements. The router consists of four pipeline stages; hence, it takes four cycles to route the header flit. Then, the remaining flits of a packet, which can vary in size from 1 to 255 flits, follow the header in a pipelined fashion. The depth and width of the output buffers are parameterized. The router supports deterministic routing, and, for flexibility, the routing strategy is implemented as a lookup table. Finally, the routers can have different areas and number of ports, depending on the network topology and the routers' locations in the network. Table 1 summarizes the device utilizations.

We implemented the basic router using Verilog HDL in about two months. We used standard FIFO buffers from the Xilinx IP library to implement the output buffers; the remaining modules are custom designs. To test the routers' functionality, we instantiated a 4×4 mesh network. Then, we attached random-number generators to each router to generate random traffic. Finally, we simulated the resulting network using the ModelSim simulator (<http://www.model.com/>) to verify that all the packets reach their destinations successfully. The complete testing of the router functionality took about one month.

The on-chip routers are likely to be used with many existing IP cores designed with certain protocols in mind (for example, CoreConnect and AMBA). If the use of NoC architectures is to spread, designers will need wrappers that can efficiently interface with existing bus-based IPs and the NoC communication infrastructure. Toward this end, we designed a few simple wrappers that interface the router with the processing cores and perform packetization and depacketization operations.

NoC prototypes for multimedia applications

Our MJPEG and MPEG-2 encoder prototypes are important examples because they shed light on a rich class of multimedia applications with similar data flows and processing requirements. The huge amount of data in these multimedia applications makes the design of a communication

Router type	No. of slices
3-port	219
4-port	304
5-port	397
6-port	503

architecture for them very challenging. The NoC communication architecture stands out as a viable approach to satisfying the demands for higher performance, better quality, and lower energy consumption in embedded multimedia systems design.

MJPEG encoder

Our MJPEG encoder prototype consists of nonoptimized, off-the-shelf IP cores connected in a star network topology, shown in Figure 2b. We chose the star topology in this case because of the small number of nodes in the design. In addition, this topology supports a sufficiently large communication bandwidth with small area overhead. The network channel width and flit size is set to 16 bits. Each packet in the network contains one block in the current frame—that is, 8×8 pixels, each represented by 16 bits. Consequently, the packets are divided into 64 body flits and one header flit that carries the address information. Finally, the FIFO buffers in the router have a depth of 16 flits.

We designed the encoder using Verilog HDL and simulated it using ModelSim before synthesis. The encoder accepts $352 \text{ pixel} \times 288 \text{ pixel}$ images in bitmap format and outputs JPEG image format. Most of the design effort—about two weeks out of four weeks' total work—went into the variable-length encoder module so that the final output would comply with standard decoders. We used Xilinx ISE Foundation to synthesize, implement, and download the encoder design to the Xilinx Virtex-II FPGA. To make the layout close to a real silicon implementation, we used the floorplanning tool embedded in Xilinx ISE Foundation.

To verify the encoder's functionality, we initially stored the bitmap images in an external memory. Then, the encoder read the image from memory in a block-by-block fashion, performed the encoding, and wrote the compressed images back to memory. Finally, we uploaded the encoded images to a host computer and decoded them using a software decoder. This verification step took about two weeks.

Our MJPEG encoder's performance, energy consumption, and area figures make it

unique. Our design reaches a throughput of 582 frames per second, very close to the theoretical maximum throughput achievable with the given set of cores (that is, 582.8 frames per second, assuming a 100-MHz clock frequency). The entire encoder can be easily enhanced with the addition of more IP cores running the application in parallel. In addition, accurate measurements performed directly on the FPGA prototype using the tool presented by Lee et al.¹¹ show that the encoder has 0.4-W dynamic power consumption. At 4,901 slices, our design is comparable in area to state-of-the-art industrial designs (http://www.barco.com/projection_systems/downloads/BA112JPEG2000EFactSheet.pdf). Thus, it is suitable for high-end digital camera systems.

MPEG-2 encoder

The MPEG-2 standard exploits the spatial redundancy by encoding intra (I) frames, similar to the JPEG standard. Therefore, the MPEG-2 encoder reuses the discrete-cosine transform (DCT), quantization, variable-length-encoder, and SRAM modules from the MJPEG prototype. The MPEG standard also exploits the temporal redundancy between consecutive frames by encoding predicted (P) frames. Our prototype accomplishes this operation by encoding three P-frames following each I-frame via motion estimation and compensation modules. We implemented these two modules from scratch using Verilog HDL in about two weeks. Finally, we implemented inverse quantization and inverse DCT modules using application notes from Xilinx (http://www.xilinx.com/xlnx/xweb/xil_publications_index.jsp?category=Application+Notes). Like the MJPEG encoder, the MPEG-2 encoder was synthesized, implemented, and downloaded to Xilinx Virtex-II FPGA using Xilinx ISE Foundation. Then, to verify its functionality, we decoded its output using a standard software decoder. Drawing on the experience we gained during the test of the MJPEG encoder, we were able to test the MPEG-2 encoder in about one week.

Using ModelSim, simulating the MPEG-2 encoder for one frame takes more than a day. When the design runs on the FPGA,

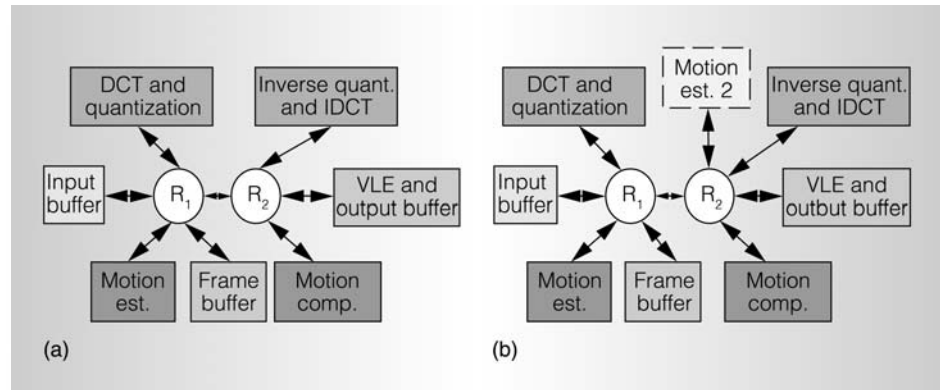


Figure 3. NoC communication architecture of the MPEG-2 encoder with a single motion estimation module (a) and with two motion estimation modules (b).

it takes only a few seconds, including the time to upload and download raw images to the FPGA. As such, FPGA prototyping allows the encoding of much longer video sequences.

Figure 3a shows the NoC architecture of the MPEG-2 encoder prototype. The width of the network channels is 16 bits, and the packets consist of 65 flits; thus, each packet carries information for an 8 pixel \times 8 pixel video block. Our implementation achieves an encoding rate of 45.6 frames per second for 352 pixel \times 288 pixel CIF frames, using a single motion estimation module. Furthermore, measurements performed on the FPGA prototype, listed in Table 2, reveal that the power consumption at a 100-MHz clock frequency is 1.92 W. The design uses 10,442 slices from the target FPGA.

The performance bottleneck in the MPEG-2 encoder is the motion estimation module. Thus, we expect a significant improvement in the design throughput by duplicating this module, as in Figure 3b. As we can see from Figures 3a and 3b, the modularity of the NoC approach allows the

new architecture to be derived from the initial architecture with little effort. The new design's throughput increases to 83.2 frames per second, which implies a speedup of about 82 percent. This improvement comes at a moderate area overhead of about 6 percent.

NoC prototypes for optimization

Although prototypes that implement real applications are desirable for obtaining the most reliable evaluation, it is impractical to implement all possible applications, especially large-scale systems that are of interest for optimization purposes. Therefore, we also created two prototypes that enable the evaluation of various performance and energy optimization techniques.

Performance optimization via long-range links

Network topologies such as a 2D mesh structure are attractive because of their regular structure, well-controlled electrical parameters, and simple layout. Nevertheless, these architectures can suffer from long packet latencies because they lack connections between remotely situated nodes. To remedy this situation, we recently demonstrated optimizing the performance of regular NoC architectures using application-specific, long-range links.⁶ The proposed approach inserts a few long-range links into a regular NoC architecture such that performance, measured as the critical traffic load at which the network becomes congested, is optimized subject to con-

Table 2. Performance, power consumption and area of the MPEG-2 encoder implementation on Xilinx Virtex-II FPGA.

No. of motion estimation modules	Performance (frames/s)	Power consumption (W at 100 MHz)	Area (no. of FPGA slices)
1	45.6	1.92	10,442
2	83.2	2.23	11,031

straints on the number of allowable additional links. In this section, we use FPGA- and ASIC-style NoC implementations to illustrate the feasibility of the approach and support our earlier results.⁶

We first connected the on-chip routers to compose a 4×4 mesh network using Verilog HDL. After testing the network's operation by hardware simulation performed using ModelSim, we determined application-specific long-range links for hotspot traffic patterns and various benchmarks from the E3S suite (<http://www.ece.northwestern.edu/~dickrp/e3s>). To implement the network with long-range links, we replaced the routers used in the original mesh network with six-port routers whenever necessary, and inserted the appropriate long-range links. We completed this process in less than a day by modifying the top-level Verilog module describing the mesh network. Similarly, we reused the test bench used for the mesh network to test the functionality of the network with long-range links. Finally, we synthesized, implemented, and downloaded the designs to Xilinx Virtex-II FPGA using Xilinx ISE Foundation.

Performance evaluation. Figure 4 shows the average packet latency as a function of the traffic injection rate before and after inserting the long-range links. At low traffic loads, the average packet latency drops by about 20 percent for the auto-industry application. As the packet injection rate increases, the reduction in latency becomes significantly larger. Similar behavior occurs for the hotspot traffic. In particular, the long-range links improve the critical traffic load by about 11 percent.

Energy consumption evaluation. We used the cycle-accurate energy measurement tool for FPGAs presented by Lee et al.¹¹ to evaluate the energy consumption. More specifically, we used a synthetic workload, in which the job of each processing element (PE) is to send 50 packets (consisting of 32 flits each) to certain destination nodes. After that, we performed two sets of experiments to assess the effect of the long-range links on energy consumption.

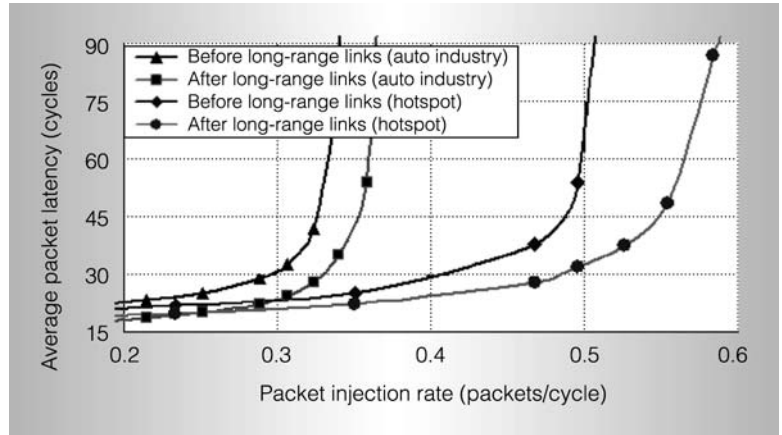


Figure 4. Comparison of the average packet latency for hotspot and auto-industry benchmarks before and after the insertion of long-range links.

In the first set of experiments, the PEs inject packets at uniform intervals of time until all 50 packets are sent. Since the time interval is set to be the same with and without long-range links, the total time to complete the job does not change considerably—7,694 cycles for the regular mesh network compared to 7,674 cycles after inserting the long-range links (see Table 3). Our measurements show that the insertion of long-range links reduces energy consumption by about 2.2 percent.

In the second set of experiments, the packet injection was bursty; that is, the PEs produce and send packets as long as the downstream router can accept them. It takes 5,120 cycles for the mesh network to complete the job, whereas the network with long-range links finishes the job in only 3,416 cycles. The corresponding energy consumption is found to be 17.14 μJ without the long-range links, and 11.72 μJ after the insertion of long-range links. The

Table 3. Impact of long-range links on energy consumption.

Traffic type	Network	Execution		
		time (cycles)	Energy (μJ)	Power (mW)
Constant rate (0.082 packets/cycle)	4×4 mesh	7,694	28.02	364.2
	Mesh with long-range links	7,674	27.39	356.9
Bursty	4×4 mesh	5,120	17.14	334.7
	Mesh with long-range links	3,416	11.72	343.2

decrease in the energy consumption is caused by the reduction in the execution time. Although after inserting the long-range links the power consumption increases from 334.7 mW to 343.2 mW (at a 100-MHz clock frequency), the speedup in job completion can be exploited to maximize the duration of the possible power-down mode and thus save energy.⁶

ASIC-style implementation and detailed area comparisons. Inserting long-range links results in an area overhead due to additional links and extra ports needed in some of the routers. According to the measurements performed on the FPGA prototype, the 2D mesh network occupies 6,683 slices, whereas the network with long-range links occupies 7,152 slices; this implies an overhead of about 7 percent.

To assess the area overhead more accurately and illustrate the impact of the long-range link on regularity, a two-student team developed an ASIC-style implementation using 90-nm technology. Although this implementation had no actual processing cores attached to the routers, to make the layouts more realistic, the two-student team intentionally created a large PE that occupied about two-thirds of the available space of each node.

First, we used the SoC Encounter tool (http://www.cadence.com/products/digital_ic/soc_encounter/index.aspx) to lay out a single node with the objective of making it as rectangular as possible. This included both arranging the router's ports to easily connect the north/south and east/west directions and leaving space around the extra port for the connection of a long-range link. This step took three weeks. The major difficulties concerned learning how to use the SoC Encounter tool to meet timing constraints and to analyze power consumption. Altering the Verilog code developed for the FPGA prototype to be used with the SoC Encounter tool was also challenging. After this step, it took two weeks to lay out the regular mesh network using the layout of a single node. It was straightforward for the placer and router to handle the floorplan, because the connections are simply neighbor to neighbor, with no obstructions.

We spent one more week optimizing the layout for performance and area. Finally, we took two days to lay out the mesh network with long-range links. There were no routing difficulties in adding the long-range links, even with the large PE that the long-range links had to travel across. Overall, adding the long-range links was significantly easier than implementing the initial mesh layout.

Figure 5 shows the resulting layouts of the mesh network, with and without the long-range links. The floorplan of the network with long-range links is surprisingly regular, and the irregularity in this implementation is fairly small. As such, adding or removing a link has a small effect on the layout. The mesh network contains 175,672 standard cells, and it occupies 20.84 mm² (see Table 4). The network with long-range links is 21.26 mm², with 181,234 standard cells. The final routing and timing of the two designs are almost identical.

The cycle time for both designs is currently at 3 ns, or a 333-MHz clock speed. Both designs meet this timing with some slack. In both cases, the critical path limiting the design is within the node and is not due to the internode links. The long-distance links are not within the critical path at this speed. The longest delay due to long-range links is 1.25 ns. This extra delay can likely be dropped with the addition of some buffer or a full logical repeater.

Evaluation of the two designs shows that they have similar power figures. The regular mesh has an average total power of 990 mW, whereas the mesh with long-range links has an average total power of 1,055 mW without clock gating. The reason for this power difference is that the long-range link layout has more logic, and thus consumes more power.

Our work demonstrates the use of application-specific long-range links for NoC performance optimization using both FPGA- and ASIC-style implementations. As future work, we plan to develop a GALS-based NoC prototype with multiple voltage-frequency islands (VFI) to evaluate various energy optimization tech-

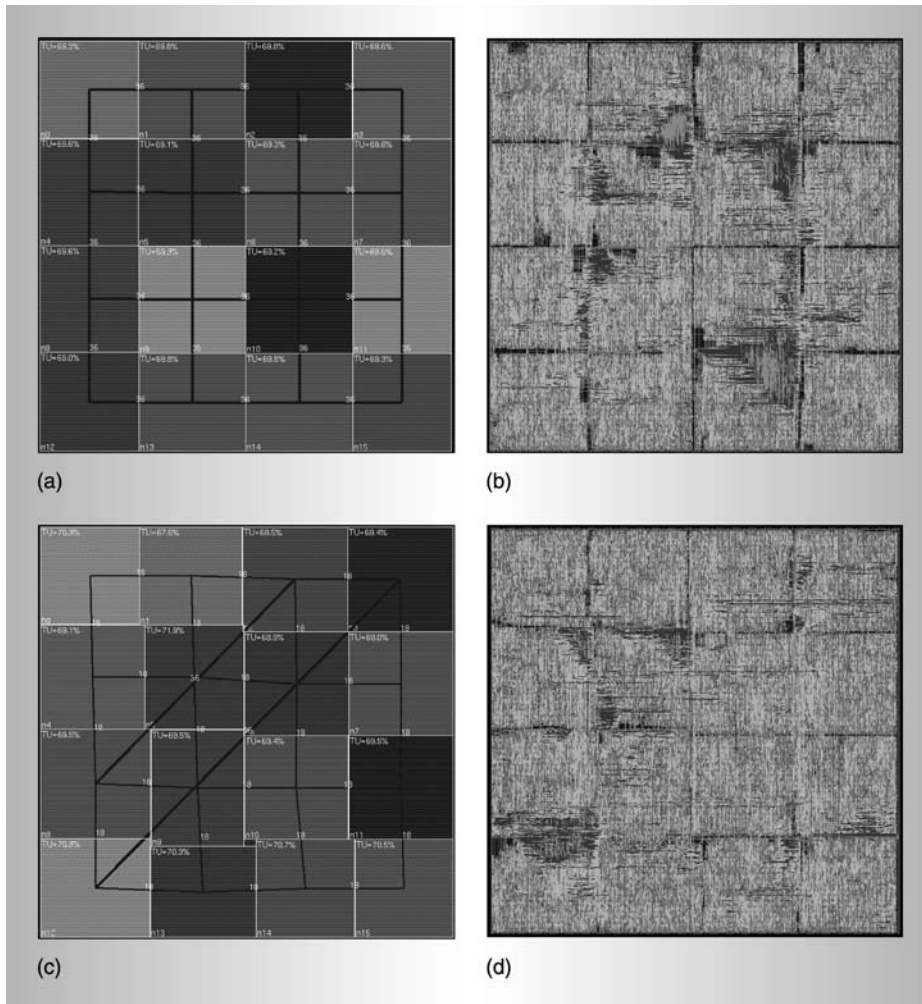


Figure 5. Floorplan (a) and layout (b) of the mesh network without the long-range links; floorplan (c) and layout (d) after the addition of long-range links.

niques.¹² In this architecture, communication within the islands will be synchronous, whereas the interisland communication will be asynchronous. The basic structure of the router will remain the same as that shown in Figure 2. However, unlike fully synchronous designs, in the GALS-based implementation, all data and control signals

across different domains must be converted from one VFI to another. We plan to accomplish this through block RAM-based mixed-clock FIFO buffers from the Xilinx library. Finally, we will use the delay-locked loops present in the FPGA device to generate the individual clock signals in each voltage island.

MICRO

Table 4. Evaluation of the ASIC-style NoC implementations with and without long-range links.

Network	Area (mm ²)	No. of cells	Maximum frequency (MHz)	Power consumption (mW)
Mesh without long-range links	20.84	175,672	333	990
Mesh with long-range links	21.26	181,234	333	1,055

Acknowledgments

This research was supported by the Gigascale Systems Research Focus Center, one of five research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation program, and in part by Semiconductor Research Corporation Contract 2004-HJ-1189.

References

1. G. De Micheli and L. Benini (Eds.), *Networks on Chips: Technology and Tools (Systems on Silicon)*, Morgan Kaufmann, 2006.
2. H.G. Lee et al., "On-Chip Communication Architecture Exploration: A Quantitative Evaluation of Point-to-Point, Bus and Network-on-Chip Approaches," *ACM Trans. Design Automation of Electronic Systems*, vol. 12, no. 3, Aug. 2007, pp. 21-40.
3. J. Muttersbach, T. Villager, and W. Fichtner, "Practical Design of Globally Asynchronous Locally Synchronous Systems," *Proc. Int'l Symp. Advanced Research in Asynchronous Circuits and Systems*, IEEE CS Press, 2000, pp. 52-59.
4. J. Hu and R. Marculescu, "Energy- and Performance-Aware Mapping for Regular NoC Architectures," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 4, Apr. 2005, pp. 551-562.
5. E. Rijpkema, K. Goossens, and A. Radulescu, "Trade-offs in the Design of a Router with Both Guaranteed and Best-Effort Services for Networks on Chip," *Proc. Design, Automation and Test in Europe Conf. (DATE 03)*, IEEE CS Press, 2003, pp. 350-355.
6. U.Y. Ogras et al., "Communication Architecture Optimization: Making the Shortest Path Shorter in Regular Networks-on-Chip," *Proc. Design, Automation and Test in Europe Conf. (DATE 06)*, IEEE CS Press, 2006, pp. 712-717.
7. K. Lee et al., "A 51mW 1.6GHz On-Chip Network for Low-Power Heterogeneous SoC Platform," *Proc. Int'l Solid-State Circuits Conf. (ISSCC 04)*, IEEE CS Press, 2004, pp. 152-161.
8. S. Vangal et al., "An 80-Tile 1.28TFLOPS Network-on-Chip in 65nm CMOS," *Proc. Int'l Solid-State Circuits Conf. (ISSCC 07)*, IEEE CS Press, 2007, pp. 98-100.
9. D. Burger et al., "Scaling to the End of Silicon with EDGE Architectures," *Computer*, vol. 37, no. 7, July 2004, pp. 44-55.
10. M.B. Taylor et al., "The Raw Microprocessor: A Computational Fabric for Software Circuits and General Purpose Programs," *IEEE Micro*, vol. 22, no. 2, Mar.-Apr. 2002, pp. 25-35.
11. H.G. Lee et al., "Cycle-Accurate Energy Measurement and Characterization of FPGAs," *Analog IC and Signal Processing*, vol. 42, no. 3, Mar. 2005, pp. 239-251.
12. U.Y. Ogras et al., "Voltage-Frequency Island Partitioning for GALS-Based Networks-on-Chip," *Proc. 44th Design Automation Conf. (DAC 07)*, ACM Press, 2007, pp. 110-115.

Umit Y. Ogras is a postdoctoral research associate in the Department of Electrical and Computer Engineering at Carnegie Mellon University. His research interests focus on communication-centric design methodologies for nanoscale systems on chips (SoCs), with a special interest in network-on-chip communication architectures. Ogras has a PhD in electrical and computer engineering from Carnegie Mellon University.

Radu Marculescu is an associate professor in the Department of Electrical and Computer Engineering at Carnegie Mellon University. His research interests include developing design methodologies and software tools for SoCs, on-chip communication, low-power design, and ambient intelligence. Marculescu has a PhD from the University of Southern California.

Hyung Gyu Lee works in the Software Laboratories at Samsung Electronics. His research interests include embedded system design and implementation, system-level low-power systems, and on-chip network design in SoCs. He has a PhD in computer science and engineering from Seoul National University, Korea.

Puru Choudhary is a senior design engineer at Marvell Technology Group. His research interests include low-power design and the design of GALS systems. Choud-

hary has an MS in electrical and computer engineering from Carnegie Mellon University.

Diana Marculescu is an associate professor of electrical and computer engineering at Carnegie Mellon University. Her research interests include energy-aware computing, CAD tools for low-power systems, and emerging technologies such as electronic textiles and ambient intelligent systems. Marculescu has a PhD in computer engineering from the University of Southern California.

Michael Kaufman is pursuing an MS in electrical and computer engineering at Carnegie Mellon University. He has a BS in the same field from the same university.

His research interests include computer architecture and hardware design.

Peter Nelson is pursuing an MS in electrical and computer engineering at Carnegie Mellon University. He has a BS in the same field from the same university. His research interests include computer architecture and hardware design.

Direct questions and comments about this article to Umit Y. Ogras, Carnegie Mellon University, Dept. of Electrical and Computer Engineering, 5000 Forbes Ave, Pittsburgh, PA 15213-3890; umit@cmu.edu.

For more information on this or any other computing topic, please visit our Digital Library at <http://www.computer.org/csdl>.

IEEE computer society

PURPOSE: The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field.

MEMBERSHIP: Members receive the monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

COMPUTER SOCIETY WEB SITE: www.computer.org

OMBUDSMAN: Email help@computer.org.

Next Board Meeting: 9 Nov. 2007, Cancún, Mexico

EXECUTIVE COMMITTEE

President: Michael R. Williams*

President-Elect: Rangachar Kasturi; * **Past President:** Deborah M. Cooper; * **VP, Conferences and Tutorials:** Susan K. (Kathy) Land (1ST VP); * **VP, Electronic Products and Services:** Sorel Reisman (2ND VP); * **VP, Chapters Activities:** Antonio Doria; * **VP, Educational Activities:** Stephen B. Seidman; † **VP, Publications:** Jon G. Rokne; † **VP, Standards Activities:** John Walz; † **VP, Technical Activities:** Stephanie M. White; * **Secretary:** Christina M. Schober; * **Treasurer:** Michel Israel; † **2006–2007 IEEE Division V Director:** Oscar N. Garcia; † **2007–2008 IEEE Division VIII Director:** Thomas W. Williams; † **2007 IEEE Division V Director-Elect:** Deborah M. Cooper; * **Computer Editor in Chief:** Carl K. Chang; † **Executive Director:** Angela R. Burgess†

* voting member of the Board of Governors

† nonvoting member of the Board of Governors

BOARD OF GOVERNORS

Term Expiring 2007: Jean M. Bacon, George V. Cybenko, Antonio Doria, Richard A. Kemmerer, Itaru Mimura, Brian M. O'Connell, Christina M. Schober

Term Expiring 2008: Richard H. Eckhouse, James D. Isaak, James W. Moore, Gary McGraw, Robert H. Sloan, Makoto Takizawa, Stephanie M. White

Term Expiring 2009: Van L. Eden, Robert Dupuis, Frank E. Ferrante, Roger U. Fujii, Ann Q. Gates, Juan E. Gilbert, Don F. Shafer

EXECUTIVE STAFF

Executive Director: Angela R. Burgess; **Associate Executive Director:** Anne Marie Kelly; **Associate Publisher:** Dick Price; **Director, Administration:** Violet S. Doan; **Director, Finance and Accounting:** John Miller

COMPUTER SOCIETY OFFICES

Washington Office. 1730 Massachusetts Ave. NW, Washington, DC 20036-1992
Phone: +1 202 371 0101 • Fax: +1 202 728 9614 • Email: hq.ofc@computer.org

Los Alamitos Office. 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314

Phone: +1 714 821 8380 • Email: help@computer.org

Membership and Publication Orders:

Phone: +1 800 272 6657 • Fax: +1 714 821 4641 • Email: help@computer.org

Asia/Pacific Office. Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan

Phone: +81 3 3408 3118 • Fax: +81 3 3408 3553

Email: tokyo.ofc@computer.org

IEEE OFFICERS

President: Leah H. Jamieson; **President-Elect:** Lewis Terman; **Past President:** Michael R. Lightner; **Executive Director & COO:** Jeffrey W. Raynes; **Secretary:** Celia Desmond; **Treasurer:** David Green; **VP, Educational Activities:** Moshe Kam; **VP, Publication Services and Products:** John Baillieul; **VP, Regional Activities:** Pedro Ray; **President, Standards Association:** George W. Arnold; **VP, Technical Activities:** Peter Staecker; **IEEE Division V Director:** Oscar N. Garcia; **IEEE Division VIII Director:** Thomas W. Williams; **President, IEEE-USA:** John W. Meredith, P.E.

revised 25 June 2007

