Analysis and Implementation of Practical, Cost-Effective Networks on Chips

Se-Joong Lee, Kangmin Lee, and Hoi-Jun Yoo
Korea Advanced Institute of Science and Technology

Editor's note:
This article describes design issues in three chips that exploit star and mesh networks, with the objective of comparing area and energy costs. The authors present new solutions based on mesochronous communication and burst packet transactions.
—Giovanni De Micheli, Ecole Polytechnique Fédérale de Lausanne

Designers pursuing higher performance have raised SoC clock frequencies to several gigahertz, revealing many challenging issues in global clock distribution and synchronization. Another recent approach to high SoC performance, besides raising clock frequencies, involves the adoption of multiple processing cores to exploit parallelism. The second-generation Itanium 2 microprocessor, for example, has two cores. The first-generation Cell processor combines eight streaming processors to provide high-performance computing for multimedia and streaming workloads.[1] One mobile-application SoC has three ARM9 processor cores and one DSP core integrated for low power consumption while providing high-performance multimedia functionalities.[2] Clearly, this trend will persist and accelerate, and the number of cores integrated on a single chip will keep increasing.[3]

Use of a high-speed clock and the multicore architecture requires an on-chip bus architecture to evolve into a network architecture. Recent studies have focused on the network-on-a-chip (NoC) architecture. Based on a flexible network architecture, the NoC can provide not only sufficient bandwidth but also additional functionalities that high-performance SoCs require.

Researchers have specified network topology, packet format, and the bitwidth of link wires for NoCs,[4,5] and more-focused research on specific NoC features has recently been published.[6,7] However, most previous work doesn't deal with chip implementation issues. Researchers determined architectural specifications such as topology and link bitwidth without considering implementation constraints such as low power consumption and area overhead. Although the literature describes implementation issues such as a high-performance NoC design, guaranteed quality of service, and low-power system design,[8,9] physical design issues and quantitative analyses of the NoC architecture haven't been addressed, and more work on implementation issues is necessary to make NoCs viable.

This article addresses NoC implementation issues related to NoC chips designed at the Korea Advanced Institute of Science and Technology[10-12] and discusses our architectural decisions, approaches, and solutions.

Chip implementation issues

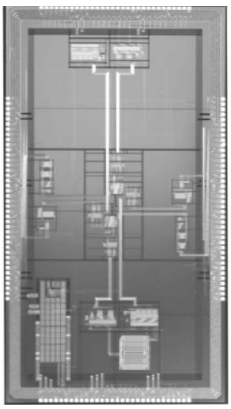
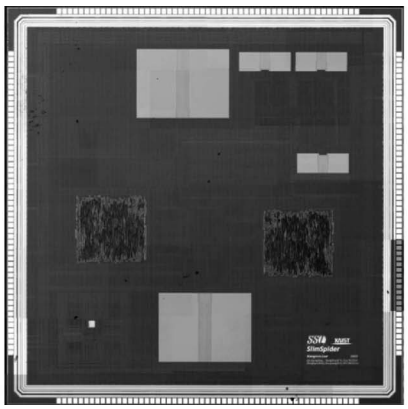
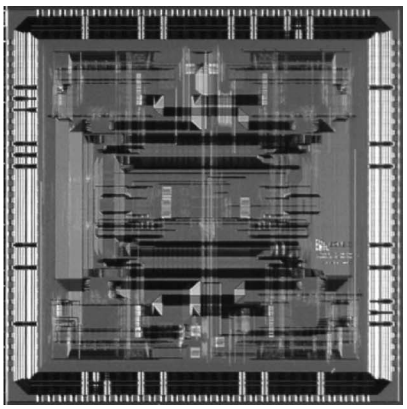
Table 1 summarizes the main features, design issues, and solutions for the three NoC chips that our group fabricated.

Circuit design

The first chip, Protone (prototype on-chip network), was designed with two physical-layer features: high-speed (800 MHz) mesochronous communication and on-chip serialization (OCS). Using 4:1 serialization, Protone transfers 80-bit packets through 20-bit links. The 4:1 serialization reduces Protone's network area by 57%, making it practical for use in SoC design. The distributed NoC building blocks are not globally synchronized, and the packets transfer with mesochronous communication. Because mesochronous communication eliminates the burden of global synchronization,

0740-7475/05/$20.00 © 2005 IEEE Copublished by the IEEE CS and the IEEE CASS

Table 1. Characteristics of the three implemented NoC chips.

	Protone	Slim-spider	Intelligent Interconnect System
Die photo			
Technology	0.38 μm	0.18 μm	0.18 μm
Die size	6.0 mm × 10.8 mm	5 mm × 5 mm	5 mm × 5 mm
Features and design issues	Mesochronous communication 800-MHz serialization Star topology	1.6-GHz serialization Low-power schemes Burst packet transaction Hierarchical star topology	Programmable synchronizer Packet bypassing scheme Configurable topology (mesh or tree)

800-MHz clocking is possible. Protone's implementation and successful measurement demonstrates the feasibility of high-performance on-chip serialized networking with mesochronous communication.

Compared with a parallel link, however, the serialized link increases link activity. When previous and current data are correlated, a parallel link saves power because it requires fewer transitions. Serialization breaks the correlation and loses the power savings. The second NoC chip, Slim-spider, solves this problem with an encoding scheme called Silent,[13] which suppresses unwanted transitions for correlated data transfer with negligible energy overhead.

Slim-spider uses a higher serialization ratio (10:1) and a higher clock frequency (1.6 GHz). Achieving 10:1 serialization at 1.6 GHz is technically significant. However, such high-ratio serialization seems inefficient in terms of power consumption: Its power analysis implies that serialization and deserialization units consume almost 20% of overall network power. Therefore, we must find an optimal serialization ratio that can minimize network area as well as power consumption. In this article, we analyze OCS in terms of the area overhead and energy consumption of network building blocks, and we propose an optimal serialization ratio in mesh and star topology networks.

The next physical design issue is mesochronous communication, which requires synchronizers between clock domains. Protone uses a FIFO synchronizer for each link wire. The FIFO synchronizer is useful when the phase difference between clock domains is unknown. However, its power and latency overhead is considerable: The synchronizers consume 23% of Protone's overall network power.

To avoid such overhead, the next chip implementation, the Slim-spider, uses a single-stage pipeline synchronizer. Intensive Spice simulations under various conditions show that this scheme eliminates all possible synchronization failures. Such an approach provides the best performance and lowest overhead. However, as the operation frequency and the number of network nodes increase, the full-custom solution is not practical. To resolve these problems, the third chip, the Intelligent Interconnect System (IIS), uses a more sophisticated synchronization scheme: a programmable delay synchronizer.

Topology

Protone uses a star topology. For Slim-spider, we used a hierarchical star topology that divides the network into local and global networks, both with the star topology. For NoCs, the star topology provides full bandwidth and minimum latency. The nonblocking crossbar switch ensures that the network is nonblocking and therefore

guarantees full bandwidth. Even though the star topology requires longer link wires, queuing and switching latency dominate the end-to-end latency;[14] consequently, the star topology has the minimum latency.

Despite having the highest bisectional bandwidth and a small hop count, the star topology hasn't been used extensively,[3,4] because it has poor scalability. The large central switch and the long interconnection wires between processing units (PUs) and the switch appear inefficient in terms of area overhead and energy consumption. However, in on-chip implementations, the number of PUs is limited to a few tens; therefore, researchers must reconsider the scalability problem. Moreover, queuing buffers constitute a considerable portion of a network's overall area overhead and power consumption, and the star topology mitigates these factors by using as few queuing buffers as possible. Therefore, choosing a cost-efficient topology depends on having real implementation data on queuing buffers and switches.

Packet format and protocol

We have not addressed packet format and transaction protocols explicitly in previous papers. In this article we discuss packet format and protocol issues in relation to our example chips.

First, a definition of packet format must consider the physical channel structure. In addition, we describe protocols for multiple-outstanding addressing (MOA) and burst packet transactions. In Slim-spider, which integrates many IP blocks, dedicated protocols support packet transactions between microprocessors, SRAMs, and other IP blocks. One such transaction is the burst packet transaction, which supports burst mode accesses between microprocessors and memory units. This article covers burst packet transaction issues and presents our adopted solutions. In addition, we describe a packet bypassing technique devised to enhance burst packet transaction efficiency.

Topology analysis

Compared with the mesh architecture, the star topology has two drawbacks: the limited number of PUs that it can interconnect cost-effectively and its use of long link wires from PUs to the central switch. These drawbacks intensify as the number of PUs (N) and the average distance between neighboring PUs (L) increase. We find the particular N value of a star architecture that makes it more cost-effective than the mesh topology by using various L values. Using 0.18-micron technology implementation results, we evaluate the area and energy consumption of mesh and star networks to find the conditions that make the star topology more cost-efficient.

Number of PUs

The number of PUs on a chip is limited to a few tens. Even if N exceeds 100, PUs will be interconnected hierarchically: PUs communicating frequently with one another will be clustered, and a local network will interconnect them. A global network will interconnect the clusters. Interconnection within a cluster involves interconnecting a few tens of PUs. Global interconnection is similar but uses longer interconnections. Therefore, all the analyses in this article apply to a range of four to 25 PUs.

Phit size

Phit, or physical transfer unit, size determines the switch area. When a 5×5 switch with a 40-bit-wide channel is implemented in full-custom 0.18-micron technology, the switch size—including arbiters, crossbar switch fabric, and input ports with a 240-bit-capacity FIFO buffer—is about 0.8 mm^2. Considering that the popular ARM7EJ-S processor is 1.25 mm^2 in 0.18-micron technology (http://www.arm.com/products/CPUs/ARM7EJSCore.html), a 40-bit phit is a reasonable choice.

Area and energy comparison

A network consists of link wires, queuing buffers, and switches. On the basis of the area and energy consumption of the building blocks shown in Table 2, where the area and energy values were measured from actual implementation results, we evaluated the overall area for each PU and the energy consumption for one packet transaction for both the mesh and star networks, using N and L. (In the link area evaluation, we assume metal routing to be in orthogonal directions. We also assume network traffic to be uniformly distributed, with all PUs transferring packets to all other PUs at the same rate.)

The y-axes in Figures 1a and 1b show the area and energy ratios for the mesh and star networks. Mesh networks tend to become more cost-effective than star networks as N increases. However, when $N \leq 16$ and $L \leq 1$ mm, star network area is between 5% and 20% smaller than mesh area, although the networks' energy consumption is comparable.

The constraint $L \leq 1$ mm implies that the star network is cost-effective for a local network. In other words, the star topology is inefficient as a global network architecture in which cluster switches would be more than 1 mm apart.

However, energy consumption in link wires can be reduced by 50% if designers decrease the link supply voltage.[11] Such a reduction in link energy consumption requires doubling the L values in the energy consumption graph (Figure 1b). When adopting the low-power link, we conclude that the star network is cost-effective when $N \leq 9$ and $L \leq 3$ mm. This implies that the star topology with a proper low-power link scheme can serve as a cost-effective global network architecture with nine or fewer clusters.

Table 2. Process and design parameters for 0.18-micron CMOS technology.

Category	Description	Typical value	Symbol
Design parameters	Packet size (= default phit size)	80 bits	
Area (μm²)	Three-packet queuing buffer	3.30×10^4	A_{QB}
	80-bit 1 x 1 crosspoint switch fabric	7.74×10^3	A_{SF}
	One-input multiplexer-tree[11] arbiter*	8.58×10^2	A_{ARB}
	80-bit 1-mm metal-link routing	8.80×10^4	A_{LK}
Energy (J)	One-packet write and read	2.67×10^{-11}	E_{QB}
	80-bit 1 x 1 crosspoint switch fabric	9.06×10^{-12}	E_{SF}
	One-input multiplexer-tree arbiter	4.00×10^{-13}	E_{ARB}
	80-bit 1-mm metal-link routing	4.78×10^{-11}	E_{LK}

*A one-input arbiter does not exist. We obtained A_{ARB} from the area of an eight-input multiplexer-tree arbiter divided by 8.

On-chip serialization

Throughout the chip implementations, we applied an OCS technique to reduce the number of interconnect wires and the switch size. Such reductions affect a network's overall area and energy consumption.

Area and energy-consumption variation due to OCS

Table 3 summarizes the area and energy-consumption variations of the building blocks—links, queuing buffers, and switches—with respect to the serialization ratio (R_{SER}). (We define R_{SER} as the I/O bitwidth of the processing units divided by the phit size.) Wire capacitance and driving buffer size determine a link's energy consumption. Applying OCS reduces the number of wires in a link, so that the wires can have more space between them within the allowed routing area, thus reducing coupling capacitance and hence energy consumption. However, OCS increases a link's operation frequency, so the driver must be enlarged to support high-speed signaling. When R_{SER} is less than 4, the energy reduction in routing wires overwhelms the energy increment in link drivers. As R_{SER} exceeds 4, however, the situation is reversed. As a result, E_{LK} decreases as R_{SER} varies from 1 to 4, and E_{LK} increases for the higher R_{SER}.

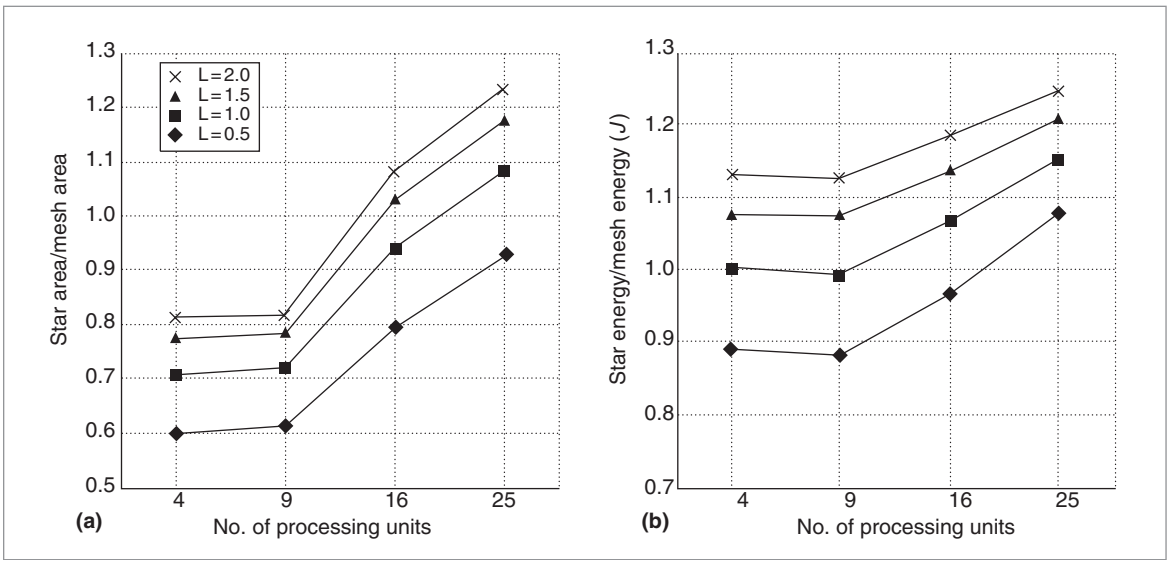


Figure 1. Area (a) and energy consumption (b) ratios between mesh and star networks with various numbers of processing units (PUs), where *L*, the average distance between neighboring PUs, ranges from 0.5 mm to 2.0 mm.

Table 3. Area and energy-consumption variations in relation to serialization ratio R_{SER}. (Values are normalized to R_{SER} = 1.)

	Area				Energy consumption			
R_{SER}	Link	Queuing buffer	Switch fabric	Arbiter	Link	Queuing buffer	Switch fabric	Arbiter
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2	0.75	1.00	0.25	1.00	0.75	1.13	0.54	1.00
4	0.50	1.00	0.06	1.00	0.64	1.30	0.32	1.00
8	0.31	1.00	0.02	1.00	0.80	1.60	0.20	1.00

Table 4. Design parameters for on-chip serialization.

Category	Description	Typical value				Symbol
Switch fabric	Coupling capacitance of a switch fabric wire (80-bit 1 × 1 switch)	24.3 fF				C_W
	Junction capacitance of a connector (80-bit 1 × 1 switch)	1.19 fF				C_J
	Gate capacitance of a switch fabric connector	1.22 fF				C_C
	Port bitwidth	80 bits				W
Link wire	R_{SER}	1	2	4	8	
	Driving buffer tapering factor (energy optimized per Spice simulation)	16	16	8	4	
	Routing wire space (μm)	0.5	1.0	1.5	2.0	

A queuing buffer's I/O bitwidth is the phit size. Because the storage cell must share the I/O wires, a small phit means more storage cells or greater capacitance connected to each I/O wire. Therefore, as R_{SER} increases, a queuing buffer's power consumption increases.

In a switch, arbiters are independent of OCS, which affects only a switch fabric. Switch fabric area decreases by the factor R_{SER}^2. As the switch fabric shrinks, its energy consumption also decreases. On the basis of the energy model presented by Wang et al.[15] and the parameters defined in Table 4, we express the energy required to transfer one packet, E_{SF}, as

$$E_{SF} = E_{SFW} + E_{SFC}$$

where

$$E_{SFW} = R_{SER} \times \frac{W}{R_{SER}} \times \left(\frac{C_W}{R_{SER}} + C_J \right) \times V^2$$

and

$$E_{SFC} = \frac{W}{R_{SER}} \times C_C \times V^2$$

E_{SFW} and E_{SFC} are the energy consumption on switch fabric wires and connectors, respectively, in a 1 × 1 switch. In the equations, E_{SFW} = number of phits per packet × phit bitwidth × wire capacitance × V^2; E_{SFC} = port bitwidth × connector gate capacitance × V^2; W is the port bitwidth; C_W is the coupling capacitance of a switch fabric wire; C_J is the junction capacitance of a connector; and C_C is the gate capacitance of a switch fabric connector. (Because the switch fabric connectors are controlled on a packet basis rather than phit by phit, the number of phits per packet is not multiplied in the E_{SFC} equation.) Because C_W is much larger than C_J, as shown in Table 4, the equations imply that E_{SF} is reduced effectively by the factor of $1/R_{SER}$. (This reduction also appears in Table 3.)

Optimal serialization ratio

On the basis of the area and energy analysis described in the "Topology analysis" section, we analyzed the serialized networks of mesh and star topologies. Figure 2 shows star and mesh energy consumption variation in relation to R_{SER} when N = 16. Both mesh and star areas are minimized when R_{SER} is 4. Because of the queuing buffer overhead and high-frequency operation in the links, further serialization is not energy efficient. This trend persists as N increases from 4 to 25. Both mesh and star areas decrease continuously as R_{SER} increases. However, once R_{SER} exceeds 4, any further serialization effect is negligible. Therefore, we choose 4:1 serialization, or a 20-bit phit size, as optimum.

The abundance of wire-line resources on a chip doesn't mean that we can use a wide channel. The wires must be organized by an OCS technique to opti-

mize network area and energy consumption. The benefits of OCS are more apparent in star networks, so that the star becomes more energy efficient when we apply 4:1 serialization, as Figure 2 shows.

Impacts on packet latency

Figure 3a shows a simplified packet transmission flow, and Figure 3b shows a timing diagram, both for a serialized mesochronous NoC. The network consists of an up sampler, a synchronizer, a switch, and a down sampler. Switches in a network use a high-frequency clock (Clk_{net}), and a mesochronous mechanism using synchronizers enables communication between switches. In this example, to enable 4:1 serialization, Clk_{net} is four times faster than the source and destination PU clocks. An up sampler and a down sampler interface the high-speed network with a PU. They perform synchronization, packet queuing, and serialization and deserialization.

OCS reduces the synchronization delay. Typically, worst-case synchronization latency is one clock cycle, which implies that using a high-frequency clock reduces synchronization delay. OCS also reduces the delay in a switch. Without OCS, a switch requires two clock cycles for processing packet ingress and egress. With the application of 4:1 OCS, the switch uses a 4× higher clock frequency. The reduced cycle time means the switch operations should be pipelined. In an actual implementation, the switch operation consists of three pipelined stages: packet processing, arbitration, and switching. Therefore, the switch latency is 3 Clk_{net} cycles, or the equivalent of a 0.75-cycle time for a nonserialized network clock. Serialized packet processing can result in timing overhead. However, the switch performs cut-through switching, so packet processing begins as soon as the first phit arrives at the switch's input port. As a result, there is no overhead in the serialized packet processing.

The additional delay resulting from OCS is T_{DES}, which is 1 Clk_{DES} cycle. T_{DES} occurs only at the end of packet transmission. Therefore, the reduction in delay time in the synchronization and the switches is enough to compensate for the additional T_{DES}.

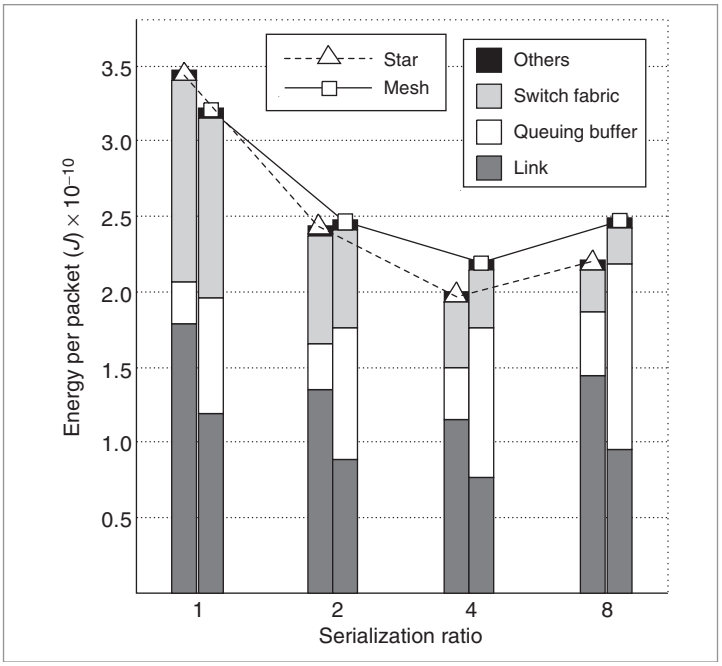


Figure 2. Mesh and star energy variation in relation to R_{SER} when the number of processing units (N) = 16.

Packet format and protocol

Here we turn to issues of packet format, multiple-outstanding addressing, and the burst packet transaction protocol.

Aligned-packet format

An aligned-packet format helps provide an efficient hardware implementation. Typically, independently defined packet, flit, and phit formats support the concept of a layered architecture. However, such independently defined formats cause a packet processing burden for a NoC. Because a NoC's data link, network, and even its transport layers should be implemented with hardware, misalignment among packet, flit, and phit formats makes parsing and processing operations difficult.

Instead of the typical packet definition, shown in Figure 4a (p. 429), we propose an aligned-packet definition that defines packet format in relation to the physical layer structure, as shown in Figure 4b. This packet format defines a fixed-length packet, and each packet field has dedicated link wires. This scheme has two advantages over the typical packet definition scheme: The packet parsing procedure is very simple, and additional link wires can easily increase a field's bitwidth. A disadvantage of this scheme, however, is that link use is inefficient if some fields are disabled. For example, if a packet carries a payload 2 that is empty, the link wires corresponding to payload 2—that is, the link wires from the 14th to the 21st link, or simply link <21:14>—remain idle, but other packet transactions can't use the idle link wires. The typical packet definition, on the other hand, can use the channel resource efficiently. Even though a shorter packet is transferred, all the link wires are used and the packet's transmission time is shorter than that of a long, or aligned, packet. Nevertheless, the disad-

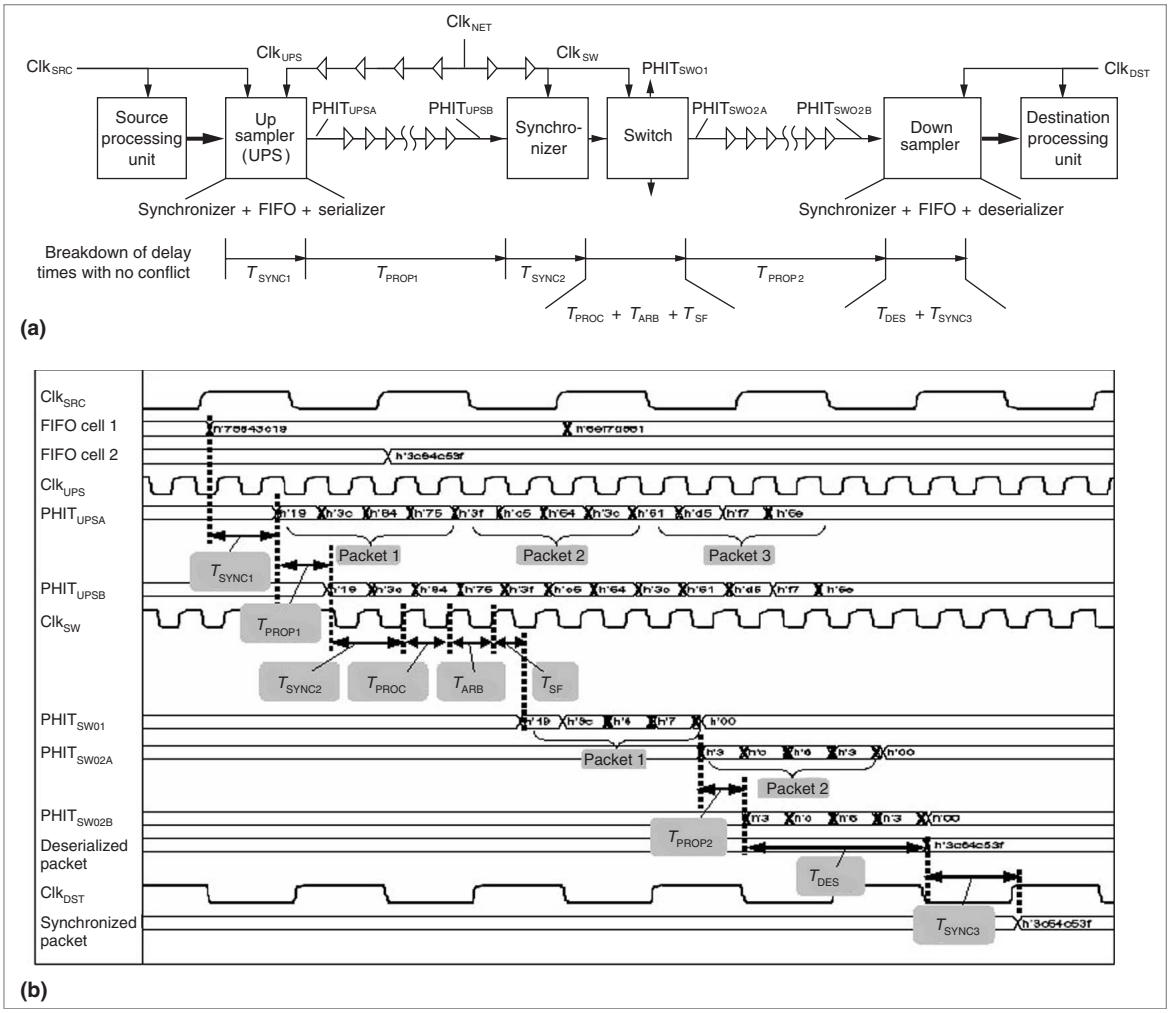


Figure 3. Packet transmission flow (a) and details of delay time breakdown (b). The breakdown uses the following categories for delay time: synchronization (T_{SYNC}), propagation (T_{PROP}), packet processing (T_{PROC}), arbitration (T_{ARB}), switch fabric (T_{SF}), and deserialization (T_{DES}).

vantages of the typical packet definition scheme—a complex packet-parsing procedure and an inflexible bitwidth adjustment—persuaded us to use the aligned-packet definition.

In the implementation results for Slim-spider, which uses a typical nonaligned-packet format, the parsing overhead occurs as a deserialization unit's power consumption at a destination. In a 1:4 deserialization design, control logic and data path power consumption increase 3× and 1.3×, respectively, compared with the aligned format. As a result, deserialization power consumption increases by about 50%.

Multiple-outstanding-addressing protocol

When a source PU issues a read command to a destination, MOA enables additional issues of read commands before the previous read transaction completes. The MOA scheme, a well-known technique in advanced bus architectures, hides a communication channel's latency and increases throughput. Using the MOA scheme in a network architecture, however, requires resolving the packet-ordering issue. Some researchers—for example, Millberg et al.,[16]—believe that using packet sequence numbers can provide a solution, although this requires a reordering buffer at each PU. To avoid the considerable hardware overhead of reordering buffers, we found possible occurrences of packet order change and designed a network interface (NI) to prevent such occurrences.

Figure 5a (p. 430) shows how, in our design, the NI manages the packet sequence. The NI checks the destination of a read command packet, and if successive

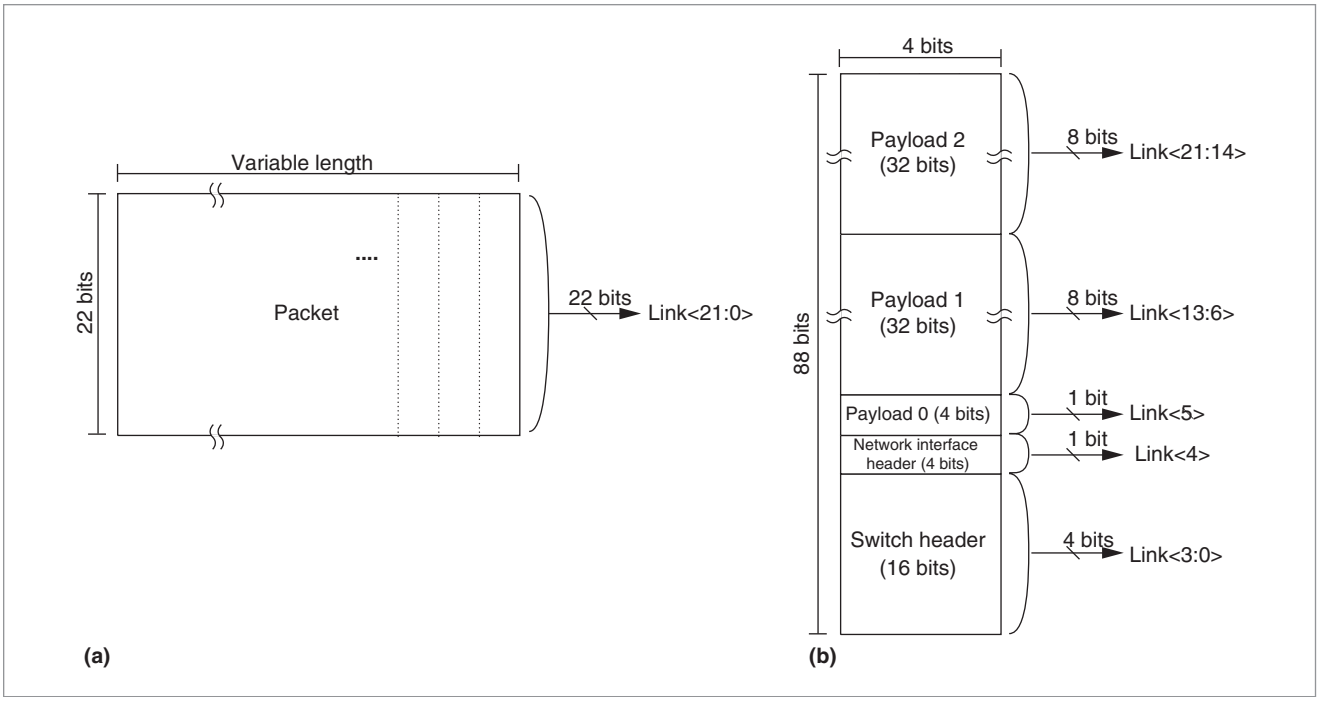


Figure 4. Packet format definitions: typical (a) and aligned (b).

read command packets have the same destination, the NI admits MOA. Because in our design the network provides a single fixed path for a single destination, packet sequence doesn't change in the network. However, if two packets have different destinations, their latency should be different, so that the order of packet issue and arrival might not be the same. Therefore, if a subsequent read command packet targets a different destination, the NI holds the source's packet transmission, thereby blocking MOA. The NI holds the transmission until all previous packet transactions complete.

Burst packet transaction

In burst read and write operations, we assume that successive packets have a continuous address space. Therefore, a burst packet flow requires protection from intrusion by other packets until it arrives at its destination. In our design, we protect burst write packet flow from interruption by using a holding-arbiter operation, as Figure 5b shows. A packet has a specific field in a header, the hold-arbiter field. For the burst write packet transaction, source NI enables the hold-arbiter fields of burst packets, except for the last packet. When the hold-arbiter field is enabled, a switch's arbiter holds its grant output. Thus, the output ports assigned to the burst packet flow aren't switched to other inputs until the last packet is routed.

The burst read packet transaction doesn't require this protection. Even if burst read packets are multiplexed with other packets, as shown in Figure 5c, the source receives burst read packets sequentially without any intrusion by other packet flows. Different burst read packet flows will not be routed to a single PU at the same time because, according to the MOA rule, a PU cannot initiate more than one burst read flow.

Packet bypassing technique

A packet-switched network uses channel resources more efficiently than a circuit-switched network because the route from a source to a destination is pipelined. The advantage increases as the route becomes longer. However, route length in current SoCs is marginal, so this alone does little for a packet-switched network. Clearly, though, if a NoC uses a high-frequency clock—say, 1 GHz in 0.18-micron technology—and small packets, the packet-switched network works well as a global interconnect architecture. However, if the NoC has a lower clock frequency for low power consumption or uses burst packet transfer, repetitive processes such as synchronization, packet queuing, and arbitration in all the intermediate switches are redundant and inefficient.

We solved this problem in the IIS chip implementation by using adaptive switching mode selection. A switch provides packet-switching or circuit-switching functionality on the basis of a packet flow's burst length.

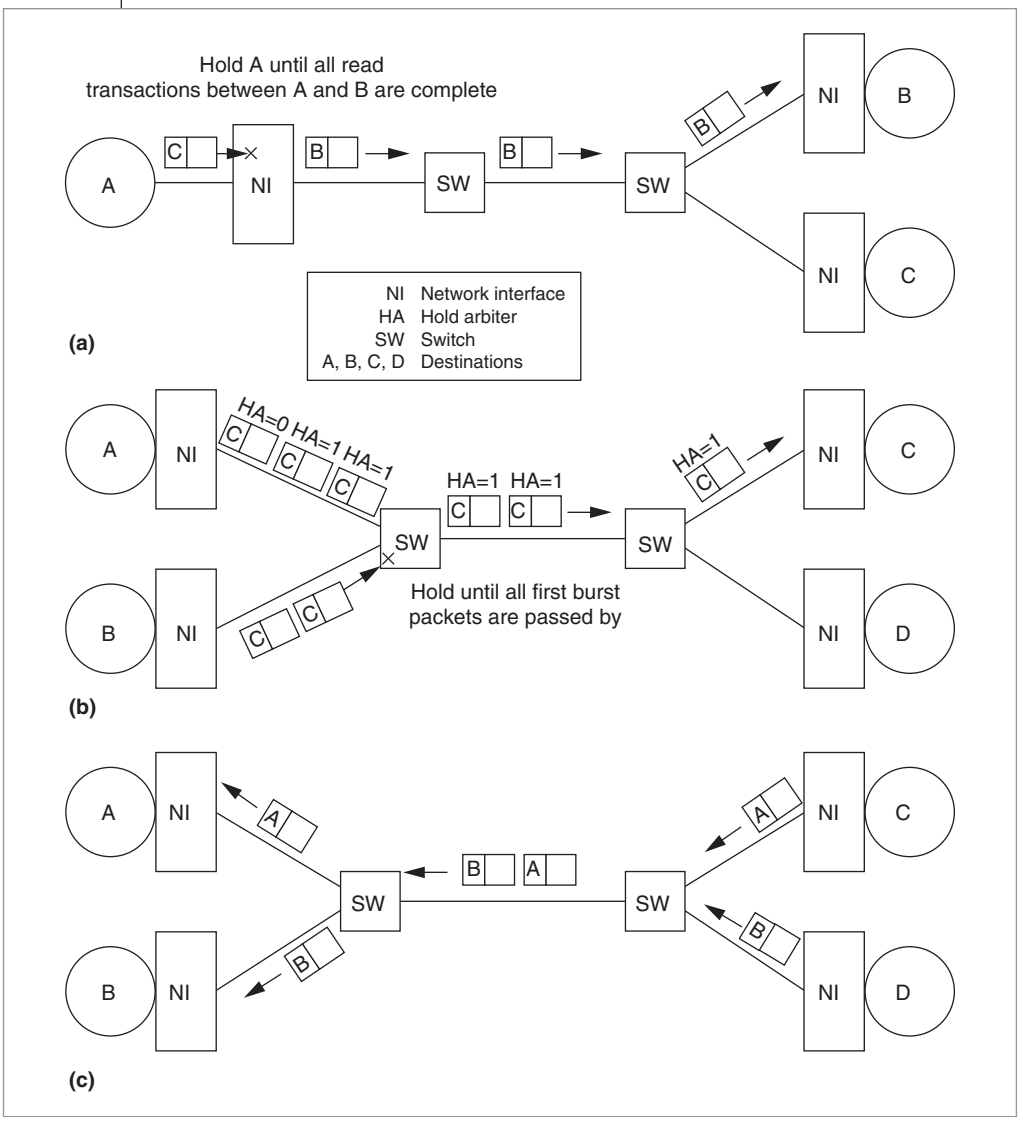


Figure 5. Packet protocols: multiple-outstanding addressing (MOA) blocking for different destination access (a), burst write packet transmission (b), and burst read packet transmission (c).

When a level 1 switch is in the circuit-switching mode, a synchronizer, a FIFO buffer, and the input port's packet-parsing logic are bypassed, and the packets are routed to the prescheduled output port. This effectively reduces delay and energy consumption for a packet transfer. Compared with the packet-switching operation, circuit switching in a NoC's 40-bit 5×5 switch has 85% less delay and consumes 70% less energy.

There are two issues in level 2 switch design: variation of synchronization timing and data-link-layer flow control. In packet-switching mode, the level 2 switch in Figure 6 receives packets from the level 1 switch. When circuit switching is enabled, however, the level 2 switch receives packets from the NI. Therefore, synchronization timing should be changed (as described in the next section), and the flow control protocol of the data-link layer must be reestablished. Level 2 switches and NIs are designed with two flow control units for the two different switching modes.

Figure 6 shows the overall operation. The NoC has level 1 and level 2 switches. A level 1 switch supports both switching modes; a level 2 switch supports only packet switching. The first packet of a burst packet flow enables the level 1 switches' circuit-switching mode, so that the remaining packets bypass the level 1 switches. This mechanism effectively reduces the number of switches along the end-to-end route, and the level 2 switches still provide the advantages of a packet-switched network. Because a level 1 switch doesn't parse entering packets, the command for escaping from the circuit-switching mode comes from the sideband signal, which is asserted by the NI and the level 2 switches.

Synchronization

Multiple-clock-domain systems require signal synchronization among clock domains. Considering the NoCs' operating circumstances, we solve the synchronization problem with a programmable delay synchronizer.

Phase difference in a mesochronous NoC

One NoC contribution to a SoC design is to ease the burden of global synchronization by using mesochro-

nous communication, meaning that network blocks share the same clock but there is no phase difference compensation. Without a mechanism to compensate for the phase difference, nondeterministic operation, such as metastability, would impair system stability.

The general definition of mesochronous communication implies that the phase difference between clock domains is unknown. In a hardwired chip, however, the phase difference is bounded to an *unknown but deterministic* difference. In addition, the deterministic phase difference varies according to a chip's working environment. The phase difference between two communicating blocks can vary because of

- the packet bypassing technique (physical distance variation),
- supply voltage variation (electrical distance variation), or
- network clock frequency variation (reference timing variation).

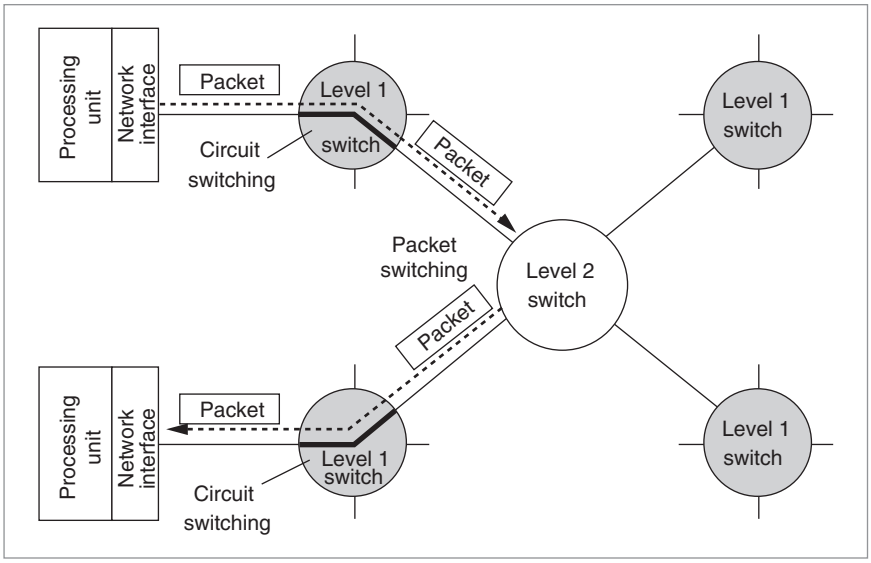


Figure 6. Packet bypassing operation.

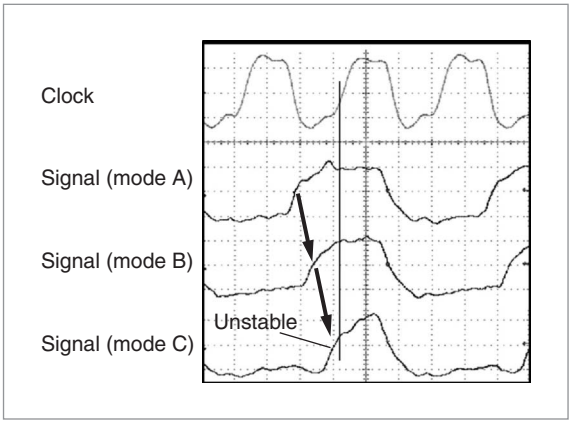


Figure 7. Quantized phase variation.

With the packet bypassing technique enabled, a switch's input port receives packets not only from its neighboring switch but also from distant switches. This means that the physical distance between two communicating blocks can change. Supply voltage control is a means of power management in low-power-application SoCs. In case of supply voltage variation, electrical distances between blocks vary, so that phase difference changes. A network clock frequency variation also changes the clock phase difference. Clock frequency scaling is useful for managing overall power consumption on the basis of the network bandwidth requirements. For example, Slim-spider uses three steps to control network frequency for low-power applications. When the clock frequency changes, reference timing also changes, which shifts the phase.

So, clearly, phase difference can vary. Having information about clock frequency, supply voltage, and network configuration lets us control phase difference properly. Because the combinations of phase-varying factors are limited, we can think of the phase difference as *unknown but quantized*. The measured waveforms in Figure 7 show a signal's quantized phase variation in relation to the network configuration. A receiver fetches the input signal at the clock's positive edge, and a transmitter deasserts the signal if the receiver fetches it successfully. As the network configuration or mode changes, the input signal's rising edge also changes. Eventually, this rising edge lies near the rising edge of the clock in network mode C, causing unstable synchronization.

Programmable delay synchronizer

To deal with the quantized phase differences, we devised a programmable delay synchronizer, shown in Figure 8. This synchronizer consists of variable-delay elements, a phase detecting unit, and a register file. During the initialization period, the phase detecting unit fetches input signals at three different sampling timings. (This resembles the oversampling technique,

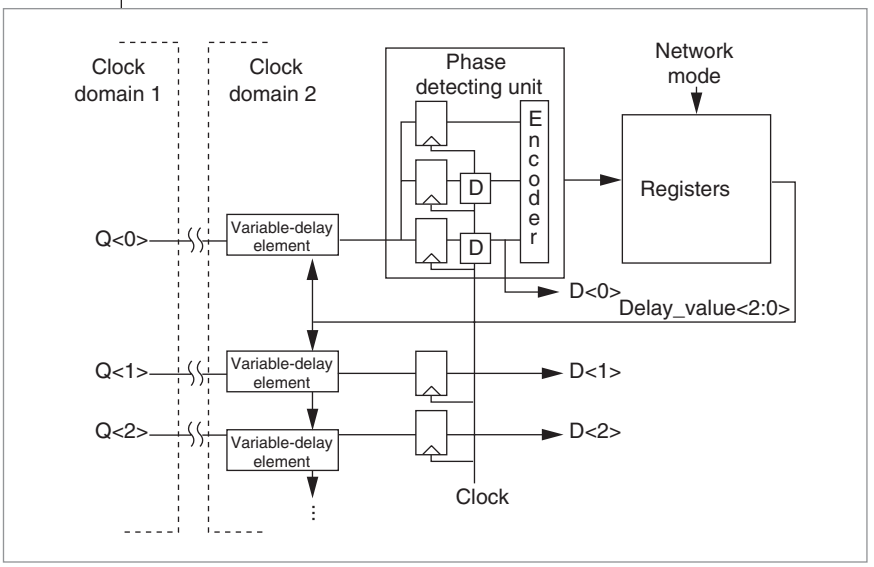


Figure 8. Block diagram of programmable delay synchronizer. The letter D within a box signifies a delay element. D<0>, D<1>, and D<2> are notations for input signals of clock domain 2. Q<0>, Q<1>, and Q<2> represent output signals of clock domain 1.

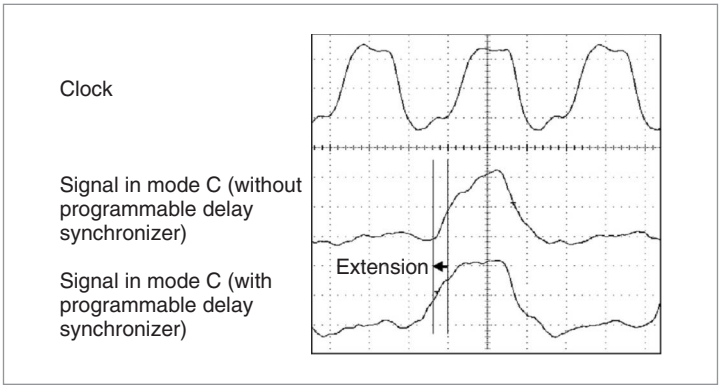


Figure 9. Measured waveforms of programmable delay synchronizer operation.

but it doesn't use a higher-frequency clock. Instead, it uses delay elements to generate the sampling timing.) The phase detecting unit uses the original clock and delayed clocks, and it finds the best timing point at which to sample the input data. The phase detecting unit converts the timing into the proper delay time—that is, the delay time that would result in enough of a sampling timing margin for the original clock if the variable-delay elements were programmed to have the proper delay time. Then, the register file stores the delay time information.

If the programmable delay synchronizer finishes the programming, the network changes its configuration or operating environment and repeats the programming for the altered operating condition. This process repeats until the network undergoes all possible conditions. After initialization, phase differences between clock domains are compensated for adaptively according to the chip's working environment. Figure 9 shows the operation of the programmable delay synchronizer. When it is used in mode C, reduced delay time is applied to the input signal, so that the timing that is fetched is effectively advanced compared with normal operation. This ensures a sufficient timing margin to fetch the input signal.

THE COST-EFFECTIVE DESIGN described in this article aims at making the NoC practical. However, NoCs still suffer from long end-to-end latency. Synchronization overhead in the mesochronous communication together with arbitration delay in all the switches produce this serious latency, which limits the NoC's application to latency-insensitive systems such as data-intensive multimedia processing systems. Future efforts at latency reduction will make the NoC more attractive to many SoC designers. ■

■ References

1. D. Pham et al., "The Design and Implementation of a First-Generation CELL Processor," *Proc. IEEE Int'l Solid-State Circuits Conf.* (ISSCC 05), IEEE Press, 2005, pp. 184-185.
2. S. Torii et al., "A 600MIPS 120mW 70 µA Leakage Triple-CPU Mobile Application Processor Chip," *Proc. IEEE Int'l Solid-State Circuits Conf.* (ISSCC 05), IEEE Press, 2005, pp. 136-137.
3. M. Taylor et al., "A 16-Issue Multiple-Program Counter Microprocessor with Point-to-Point Scalar Operand Network," *Proc. IEEE Int'l Solid-State Circuits Conf.* (ISSCC 03), IEEE Press, 2003, pp. 170-171.
4. W.J. Dally et al., "Route Packets, Not Wires: On-Chip Interconnection Networks," *Proc. Design Automation Conf.* (DAC 01), ACM Press, 2001, pp. 684-689.
5. S.-J. Lee et al., "Packet-Switched On-Chip Interconnection Network for System-on-Chip Applications," *IEEE Trans. Circuits and Systems Part II,* vol. 52, no. 6, June 2005, pp. 308-312.
6. S. Murali et al., "SUNMAP: A Tool for Automatic Topolo-

gy Selection and Generation for NoCs," *Proc. Design Automation Conf.* (DAC 04), ACM Press, 2004, pp. 914-919.
7. H. Jingcao et al., "DyAD—Smart Routing for Networks-on-Chip," *Proc. Design Automation Conf.* (DAC 04), ACM Press, 2004, pp. 260-263.
8. E. Rijpkema et al., "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* (DATE 03), IEEE CS Press, 2003, pp. 350-355.
9. D. Bertozzi et al., "Xpipes: A Network-on-Chip Architecture for Gigascale System-on-Chip," *IEEE Circuits and Systems Magazine*, vol. 4, no. 2, 2004, pp. 18-31.
10. S.-J. Lee et al., "An 800MHz Star-Connected On-Chip Network for Application to Systems on a Chip," *Proc. IEEE Int'l Solid-State Circuits Conf.* (ISSCC 03), IEEE Press, 2003, pp. 468-469.
11. K. Lee et al., "A 51mW 1.6GHz On-Chip Network for Low Power Heterogeneous SoC Platform," *Proc. IEEE Int'l Solid-State Circuits Conf.* (ISSCC 04), IEEE Press, 2004, pp. 152-153.
12. S.-J. Lee et al., "Adaptive Network-on-Chip with Wave-Front Train Serialization Scheme," *Proc. Symp. VLSI Circuits*, IEEE Press, 2005, pp. 104-107.
13. K. Lee et al., "SILENT: Serialized Low Energy Transmission Coding for On-Chip Interconnection Networks," *Proc. Int'l Conf. Computer-Aided Design* (ICCAD 04), 2004; http://www.iccad.com/archive.html.
14. K. Kim et al., "An Arbitration Look-Ahead Scheme for Reducing End-to-End Latency in Networks on Chip," *Proc. Int'l Symp. Circuits and Systems* (ISCAS 05), IEEE Press, 2005, pp. 2357-2360.
15. H.-S. Wang et al., "A Power Model for Routers: Modeling Alpha 21364 and InfiniBand Routers," *IEEE Micro*, vol. 23, no. 1, Jan.-Feb. 2003, pp. 26-35.
16. M. Millberg et al., "The Nostrum Backbone—A Communication Protocol Stack for Networks on Chip," *Proc. 17th Int'l Conf. VLSI Design*, IEEE CS Press, 2004, pp. 693-696.

Se-Joong Lee is a researcher in the Semiconductor System Laboratory at the Korea Advanced Institute of Science and Technology (KAIST). His research interests include high-speed on-chip interconnects and networking in SoCs. Lee has a BS, an MS, and a PhD in electrical engineering from KAIST. He is a student member of the IEEE.

Kangmin Lee is a PhD candidate in electrical engineering at KAIST. His research interests include the theory, architecture, and VLSI design of high-speed network switches and on-chip interconnection networks. Lee has a BS and an MS in electrical engineering from KAIST. He is a student member of the IEEE.

Hoi-Jun Yoo is on the Department of Electrical Engineering faculty at KAIST, where he leads the System Integration and IP Authoring Research Center. His research interests include SoC design, IP authoring, high-speed and low-power memory circuits and architectures, design of embedded-memory logic, optoelectronic ICs, and novel devices and circuits. Yoo has an MS and a PhD in electrical engineering from KAIST. He is a member of the IEEE.

■ Direct questions and comments about this article to Se-Joong Lee, Dept. of Electrical Eng., KAIST, Guseong-dong, Yuseong-gu, Daejeon, Korea; shocktop@eeinfo.kaist.ac.kr.

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