Adaptive Channel Buffers in On-Chip Interconnection Networks— A Power and Performance Analysis

Avinash Karanth Kodi, *Member*, *IEEE*, Ashwini Sarathy, and Ahmed Louri, *Senior Member*, *IEEE*

Abstract—On-chip interconnection networks (OCINs) have emerged as a modular and scalable solution for wire delay constraints in deep submicron VLSI design. OCIN research has shown that the design of buffers in the router influences the energy consumption, area overhead, and overall performance of the network. In this paper, we propose a low-power low-area OCIN architecture by reducing the number of buffers within the router. To minimize the performance degradation due to the reduced buffer size, we use the already existing repeaters along the inter-router channels to double as buffers along the channel when required. At low network loads, the proposed adaptive channel buffers function as conventional repeaters, propagating the signals. At high network loads, the adaptive channel buffers function as storage elements in addition to the router buffers. The router buffers can be assigned either statically or dynamically to the incoming packets. Static allocation reserves equal buffer space partitioned among all of the incoming packets, whereas dynamic allocation reserves buffer space on a per-flit basis, enabling higher buffer occupancy. We evaluate the proposed adaptive channel buffers with both static and dynamic buffer allocation policies in the 90-nm technology node, using 8×8 mesh and folded torus network topologies. Simulation results using the SPLASH-2 suite benchmarks and synthetic traffic patterns show that, by reducing the router buffer size, our proposed architecture achieves nearly 40 percent savings in router buffer power, 30 percent savings in overall network power, and 41 percent savings in area, with only a marginal 1-5 percent drop in throughput under dynamic buffer allocation and about 10-20 percent drop in throughput for statically assigned buffers.

Index Terms—On-chip networks, interconnect design, adaptive channel buffers, low-power architecture.

1 Introduction

Technology scaling is expected to continue into the deep submicron regime for at least the next decade as projected by Moore's law and the more recent growth rate from the International Technology Roadmap for Semiconductors (http://www.itrs.net/). As the density of transistors on a chip increases, the trend toward integrating more functionality onto a single chip has given rise to the Chip Multiprocessor (CMP) paradigm [1], [2]. In CMP architectures, gate delays continue to scale down with successive technology generations while wire delays increase [3], [4]. With rapidly diminishing feature size, signals require several clock cycles to traverse from one edge of the chip to another. This increased wire delay problem in CMP architectures has led to the design and development of a more structured and scalable packet-switched On-Chip Interconnection Network (OCIN) paradigm [1], [2], [5], [6], [7], [8], [9], [10], [11], [12].

As OCINs are being targeted at complex systems such as CMPs, heterogeneous cores, portable and handheld devices, accurate estimation of their performance, power dissipation, and area overhead are essential during the design phase in order to avoid costly redesign. These on-chip networks are characterized by the *channels* for data transmission and the *routers* for storing, arbitration, and switching functions performed by input buffers, arbiters, and the crossbar, respectively. In a recent workshop on OCINs [13], it was shown that almost 46 percent of the router power was consumed by the input buffers and 54 percent of the router area was dominated by the crossbar [13]. Moreover, for every bit of information transmitted, the router consumed almost eight times the power of the link [13]. With the increasing need for low-power architectures, these power consumption and chip area trends for OCINs have initiated several research efforts, including

1. reducing the buffer power and area constraints [7], [8],
2. minimizing the crossbar power by segmented and cut-through crossbars [9], [10],
3. optimizing the performance by look-ahead routing [14], speculative virtual channel (VC) [15], and switch allocation (SA) [16],
4. regulating the link power by adopting dynamic voltage and frequency scaling [17], [18], [19], and
5. incorporating topological [5], [9] and routing optimizations [20].

As the input buffers account for significant router power budget and area, a straightforward optimization would be to reduce the number of input buffers. However, the network performance and flow control is primarily characterized by the input buffers [21]. A good flow control determines how

- *A.K. Kodi is with the Department of Electrical Engineering and Computer Science, Russ College of Engineering and Technology, Ohio University, 322 Stocker Center, Athens, OH 45701. E-mail: kodi@ohio.edu.*
- *A. Sarathy and A. Louri are with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, PO Box 210104, 1230 E. Speedway Blvd., Tucson, AZ 85721.*
E-mail: {sarathya, louri}@ece.arizona.edu.

Manuscript received 2 July 2007; revised 6 Mar. 2008; accepted 13 Mar. 2008; published online 28 Apr. 2008.
Recommended for acceptance by R. Marculescu.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TCSI-2007-07-0288.
Digital Object Identifier no. 10.1109/TC.2008.77.

a network's resources, such as the channel bandwidth and the buffer capacity, are allocated to packets traversing the network. Wormhole switching combined with VC flow control allowed the channel state to be decoupled from the channel bandwidth, thereby increasing the throughput and avoiding potential deadlocks in the network [21], [22]. For power and area constrained OCIN design, reducing the size of the input buffer will result in a reduction in either the number of VCs or the buffer depth, both of which are very critical for overall network performance.

Current wire design trends have shown that signal delay along a wire increases quadratically with the length of the wire [23]. Repeater insertion along the wire makes the delay linearly dependent on the wire length and is always required to meet the stringent timing constraints of high-speed Very Large Scale Integration (VLSI) designs [3], [6], [24], [25]. Research initiatives into optimizing the performance of these repeaters have shown that the repeaters can also be designed to sample and maintain data line voltage levels when required [26]. Therefore, with repeaters on the channel as potential buffer elements, it is possible to reduce the router buffer size and utilize the storage on the channel when required.

In this paper, we propose reducing the power consumption and area overhead of OCINs by employing circuit and architectural techniques at the channel and the router buffers, respectively. At the channel, we deploy circuit level enhancements to the existing repeaters so that they can double as buffers when required. We propose a novel control block that will enable the repeaters to adaptively function as buffers during congestion. At the router, we deploy architectural techniques such as static and dynamic buffer allocation to prevent performance degradation while sustaining or improving the performance of a generic router. Static allocation reserves equal buffer space partitioned among all the incoming packets, whereas dynamic allocation reserves buffer space on a per-flit basis (a flit is a basic flow control unit and a packet consists of several flits), enabling higher buffer occupancy.

The proposed adaptive channel buffers can be viewed as serial FIFO buffers as opposed to parallel FIFO buffers within the routers. This causes Head-of-Line (HoL) blocking, which can in turn lead to deadlocks in the network. Therefore, eliminating HoL blocking and preventing deadlocks is critical in our proposed architecture. HoL blocking is more pronounced in the static allocation scheme as flits of different packets are not mixed, to ensure easier control. Dynamic buffer allocation alleviates this problem of HoL blocking to some extent, through flexible flit placement within the router buffer and tighter flow control (limiting flit transmission by the number of credits available). In addition to the dynamic buffer allocation, other deadlock recovery mechanisms need to be employed in order to completely overcome the effects of HoL blocking. Our design can be extended to dynamically increase/decrease the number of VCs depending on the network load [8]. Another alternative is to provide a spare VC (and a corresponding buffer slot) that can work as a release path in case of deadlock recovery.

1.1 Related Work

As the input buffers within the routers affect the overall performance of the OCINs, several research initiatives have targeted the design of optimized buffers to improve the buffer utilization. Given that the depth of the buffers per VC is an important resource in the OCIN environment, an application-specific buffer management scheme that allocates the buffer depth to the VCs depending on the traffic pattern has been explored in [7]. While input buffering is commonly seen, the effect of repositioning the buffers either at the output or in the middle (between the input and output of the crossbar switch) has been studied [27] in the context of multiprocessor systems. Changing the buffer depth and organization has shown almost 85 percent savings in buffer resources using static buffer allocation. Static allocations do not utilize resources optimally, but greatly simplify the design and management of the buffers by regulating the buffer allocation. Therefore, in this work, static allocation has been considered as one of the potential solutions.

Dynamic buffer allocation has been explored by using link lists, circular buffers, and a table-based approach [8], [21], [28], [29]. Dynamically Allocated Multi-Queue (DAMQ) [29] buffers made use of link lists by fixing the number of VCs for each input port. This eased the operation for a given input port. As link lists were used to update the pointer logic to maintain the free list, it caused a three-cycle delay at every flit arrival/departure. For OCINs, where performance is of paramount importance, this three-cycle delay is unacceptable. Fully Connected Circular Buffers (FC-CBs) [28] avoided the link list approach and used registers to selectively shift some flits within the buffer. However, being fully connected, it required $P^2 \times P$ crossbar instead of the regular $P \times P$ crossbar. Moreover, it required some existing flits to be shifted when a new flit arrived. This requirement adds considerable latency, power, and area overhead over a more nonshifting approach such as the ViChaR design [8].

The motivation for ViChaR is that if there are few VCs and a large buffer depth, then, at high load, packets are blocked due to lack of VCs. In ViChaR, the number of VCs and the depth of buffers per VC are dynamically adjusted based on the traffic load. It was shown that a single-cycle latency was sufficient to manage the VCs, track empty slots, and dynamically allocate buffers to new flits. However, there are two disadvantages in the ViChaR design: 1) As there can be as many VCs as there are flit buffers, VC arbitration, SA, credit return, and slot tracker logic become more complicated and 2) while increasing the number of VCs arbitrarily can achieve increased throughput (as it prevents packets from blocking), it also has the side effect that the latency of the packet increases. The larger latency is simply the result of the increased interleaving of packets that occurs with more VCs, which tends to "stretch" the packets across the network [21]. Moreover, it has been shown in [30] that increasing the number of VCs is beneficial for uniform traffic, while increasing the depth is beneficial for nonuniform traffic. Therefore, in the proposed dynamic allocation scheme, we adopt a dynamic VC table-based approach with a fixed number of VCs, thereby achieving the flexibility of storing flit buffers dynamically without excessive control overhead.

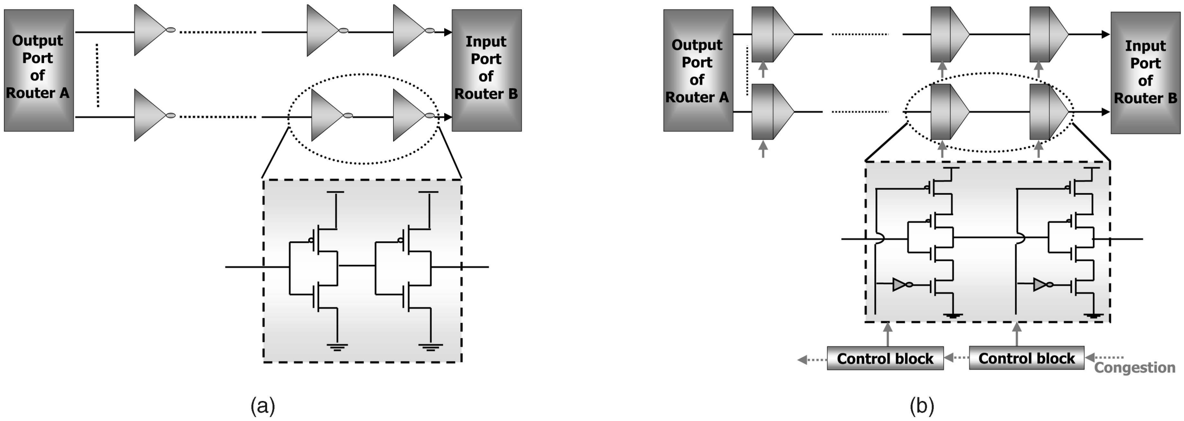


Fig. 1. (a) A conventional repeater-inserted channel between two routers. (b) Proposed channel using three-state repeaters that function as adaptive channel buffers during congestion.

1.2 Our Contributions

The distinct contributions of our proposed work are given as follows:

1. *Control block circuit for adaptive channel buffers.* The proposed control circuit technique achieves two distinct features: a) It can be designed to operate accurately at high clock speeds and b) it consumes significantly low power as it can be disabled in the absence of congestion.
2. *Dynamic and static buffer allocation with congestion control.* In both the static and dynamic buffer allocation schemes utilized in our proposed architecture, the changes made pertain only to the input buffers and the allocation of the input buffer space to the incoming flits. As opposed to other OCIN designs where performance is improved at the cost of major changes to the entire router architecture, we minimize the need for redesign and make necessary changes only at the input buffer.
3. *Combination of circuit and architectural techniques.* The combination of circuit and architectural techniques using channel buffers and router buffers (with static and dynamic buffer allocations) is unique to our proposed architecture. This combination allows us the flexibility to reduce the number of router buffers without significantly degrading the throughput and latency of the network.
4. *Detailed power and performance evaluation.* Power and area estimations in the 90 nm technology node at 500 MHz and 1.0 V show a 30 percent reduction in overall network power and a 41 percent reduction in area when half of the input buffers in the router are removed.

Cycle accurate network simulation on 8×8 mesh and folded torus network topologies show only a marginal 1-5 percent loss in throughput for dynamic buffer allocation and a 10-20 percent drop in throughput with static buffer allocation. Performance evaluation on an 8×8 mesh network running SPLASH-2 suite benchmarks showed less than 1 percent drop in performance and 20-30 percent overall network power savings.

The remainder of this paper is organized as follows: The implementation of the channel buffers and the associated control logic is explained in Section 2. The design of the router buffers using static and dynamic buffer allocations is explained in Section 3. Performance evaluation for the proposed architecture in terms of power consumption, area overhead, and network simulation is presented in Section 4. We conclude in Section 5.

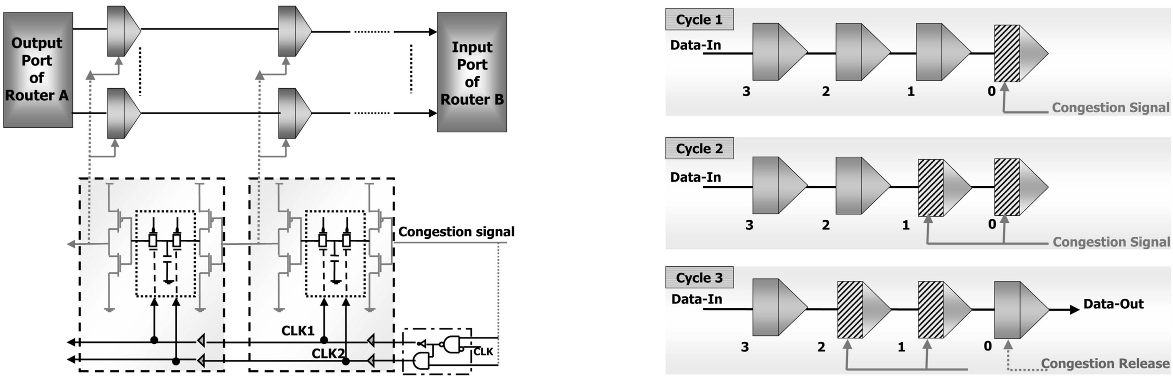
2 Design of Channel Buffers

2.1 Proposed Channel Buffer Implementation

In this section, we detail the implementation of the proposed channel buffers and the associated control logic. Fig. 1a shows a conventional repeater-inserted channel between two routers. The inverters (functioning as repeaters) are sized and spaced according to the first-order RC wire delay model described in [3]. Fig. 1b shows the proposed channel with the conventional repeaters replaced by the three-state repeaters, which can also function as buffers when required. A single stage of the three-state repeaters comprises of a three-state repeater-inserted segment along all of the wires in the channel. Each such repeater stage receives a control input from the corresponding control block. In the absence of congestion, the control logic is turned OFF and the three-state repeaters function like the conventional repeaters shown in Fig. 1a. Data moves through the channel without being held by the three-state repeaters. When the control block is turned ON by the incoming congestion signal, the three-state repeaters function as channel buffers and the data bit is held in position. Once congestion is alleviated, the control logic turns OFF and the three-state repeaters continue to function as conventional repeaters. The presence of channel buffers can thus reduce the number of input buffers required in the router to achieve significant savings in power and area for a given network performance.

2.2 Proposed Control Block Implementation

The proposed control block enables the three-state repeaters to function as channel buffers during congestion. A single control block is sufficient to control the functionality of all of the repeaters in one stage. Fig. 2a shows the circuit-level



(a)

(b)

Fig. 2. (a) Proposed control blocks interfaced to the channel buffer stages. (b) Data-flow control in the proposed channel buffers during congestion.

implementation of the proposed control block. The incoming congestion signal is delayed by one clock cycle at each control block, using a simple switched capacitor. In the next clock cycle, the channel buffer stage is tri-stated and the congestion signal travels to the next control block. Hence, each channel buffer stage is successively turned OFF to hold the data in position until the congestion-release signal arrives. The design of the congestion control line with the proposed control blocks shown in Fig. 2a provides the following advantages: 1) The control circuit behaves as a delay module as well as a repeater for the congestion signal. Unlike conventional repeaters, the control circuit shown in Fig. 2a operates accurately at variable clock speeds and retains the signal stability even at high clock speeds and 2) the control block can be turned OFF by the clocking circuitry when there is no congestion, thus reducing the power consumption along the congestion control line.

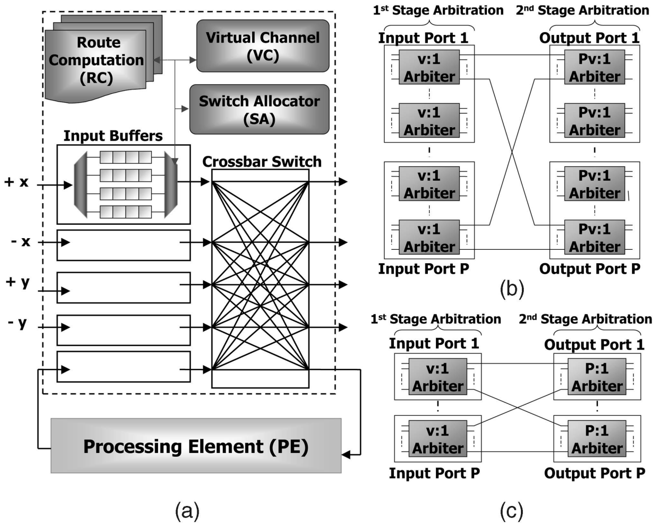
Fig. 2b illustrates the data-flow control along the channel using four stages of channel buffers and the corresponding control blocks. For simplicity, only a single wire in the channel has been shown. During cycle 1, the incoming congestion signal causes the data bit to be held by the zeroth channel buffer, while the remaining stages continue to function as repeaters. After the delay in the control block, the congestion signal travels to the next stage in cycle 2 and causes it to hold the data bit in position. The remaining two stages still continue to function as repeaters. Cycle 3 shows the congestion-release signal arriving at the zeroth stage. The zeroth stage outputs the data held and functions as a repeater while the congestion signal reaches the third stage, causing it to buffer the data. Thus, the channel buffers are successively switched to function as buffers during congestion and then successively released to continue as repeaters once congestion is alleviated.

3 Design of Router Buffers

3.1 OCIN Router Architecture

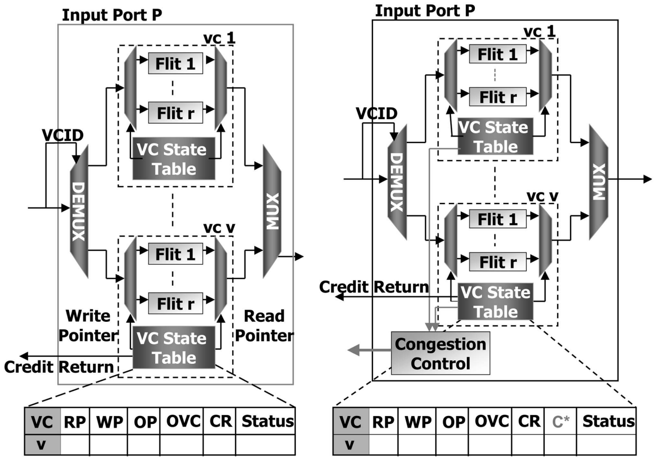
In packet-switched OCINs, every processing element (PE) is connected to an OCIN component (router), as shown in Fig. 3a, with most OCINs commonly adopting network topologies such as mesh or torus for regularity and modularity [20], [21], [27], [28]. In wormhole switching, each packet that arrives on the input port progresses through the router pipeline stages [Routing Computation (RC), VC allocation (VA), SA, and Switch Traversal (ST)] before it is delivered to the appropriate output port [21]. At each intermediate router, only the header flit of every packet is responsible for the RC and VA pipeline stages. Fig. 3b shows the VA stage arbitration for a P port, v VC/port, and r flit buffers/VC router architecture with the total buffers/port, $z = vr$ [16]. Every flit (including header and body flits) of the packet competes for access to the crossbar in the SA stage. Fig. 3c shows the SA stage arbitration [8]. After switch traversal, the flit is transferred on the channel between the routers in the Link Traversal (LT) stage and the process repeats.

The input buffer organization of a parallel FIFO buffer is shown in Fig. 4a. Each input VC is associated with a VC state table [8], [21] which ensures that the incoming flits are routed to the correct output port (OP). The VC Identifier (VCID) of the incoming flit allows the input demultiplexer (DEMUX) to switch to the correct input VC. The write pointer (WP) points to an empty flit buffer to write the incoming data. The read pointer (RP) points to the next flit to be transmitted to the crossbar. OP is provided by the RC



(a) (b) (c)

Fig. 3. (a) A generic OCIN router architecture. (b) Virtual channel arbitration. (c) Switch allocation.



(a) (b)

Fig. 4. (a) Generic static buffer allocation. (b) Proposed static buffer allocation with congestion control.

stage; output VC (OVC) is provided by the VA stage. Credits (CR) indicates the total amount of storage available at the downstream router. The status field at the end indicates the current status of the VC—idle, waiting, routing, VA, SA, ST, and others.

3.2 Statically Allocated Router Buffers

Static allocation of router buffers simplifies the overall design of the router by partitioning the buffer space among the VCs. An incoming flit is directed to an available buffer slot among the buffers assigned to that VC. This prevents mixing of flits from different packets and reduces the control overhead for the buffers. In the generic OCIN design, the total number of input buffers is vr per input port. With the wires doubling as buffers, we have additional c buffers in the channel. Therefore, the total storage available becomes $vr + c$. The number of credits available at each VC is $\lfloor (vr + c)/z \rfloor$. This allows routers to send additional flits into the network, even if the storage is in the channel instead of in the router buffer. The proposed statically allocated router buffer architecture with congestion control is shown in Fig. 4b. Other than the congestion control unit, all other functionalities are identical to the generic router architecture. Every VC state table maintains an additional field C^*, which indicates congestion. As the buffer implemented is a FIFO register, if the WP does not point to a null buffer and $\mathrm{WP} = \mathrm{RP}$, then the C^* field is set. This activates the congestion control, which in turn holds the data in the network channel itself. When a flit is read from the buffer, the RP moves to the next buffer, clearing the congestion C^* field and allowing data flits to enter into the router.

From the perspective of implementation, this nominal change does not impact the design of the network router architecture. Moreover, significant power savings and area gain can be obtained. However, from the perspective of performance, this design leads to HoL blocking in the channel buffers at high network load. When the congestion field C^* is set for a particular VC, the corresponding flits are held in the network channel. These flits block the flits headed toward other VCs, although the other VCs may have their C^* field cleared. Therefore, unavailability of buffers in any one of the VCs causes flits headed to all other VCs to be blocked. A more attractive alternative is dynamic allocated router buffers explained in Section 3.3.

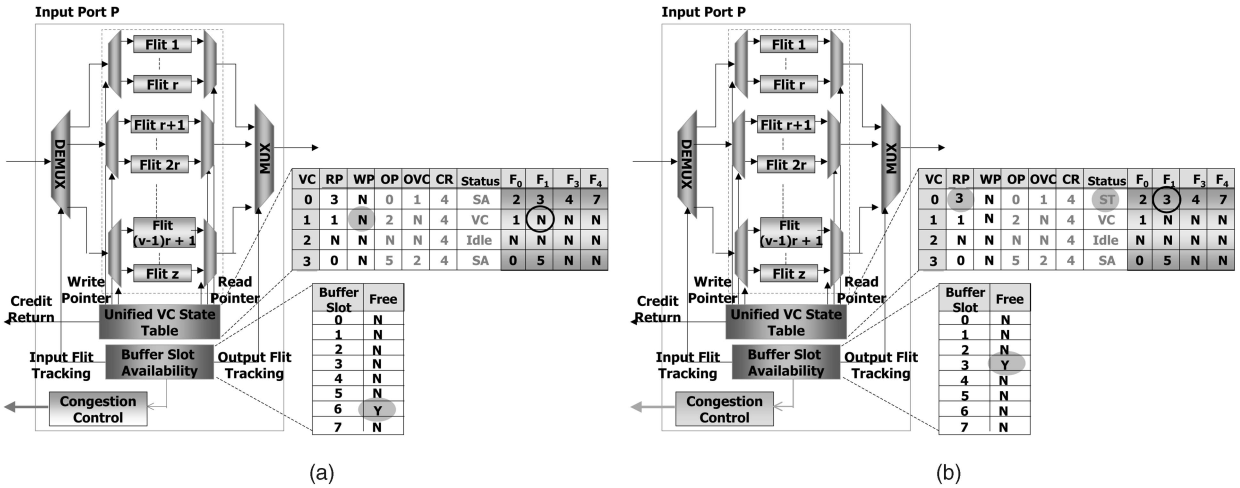
3.3 Dynamically Allocated Router Buffers

Dynamic allocation of router buffers maximizes the throughput of the network as an incoming flit can be directed to any available slot in the entire buffer space. This technique allows the buffer space to be shared by flits belonging to different packets and significantly reduces HoL blocking. In designing dynamically allocated router buffers, our goal is to maximize the throughput of the network without increasing the router latency. Link list [29] and circular buffers [28] have either the latency penalty or the crossbar scaling issue. As ViChaR's [8] table-based approach had solved several issues pertaining to latency and scalability, we have adopted a similar idea but limited the number of VCs.

Figs. 5a and 5b illustrate the proposed dynamically allocated router buffer architecture. Here, $v = 4$ (VCs/port), $z = 8$ (total buffer slots/port), and $c = 8$ (channel buffers). We adopt a unified buffer structure and augment the architecture with a "Unified VC State Table" (UVST), whose size is minimal and does not grow with the number of VCs. The maximum size of the UVST is $O(v)$ as compared to that in ViChaR, which is $O(vr)$. When a new flit arrives, its VCID cannot be used to switch as all buffer slots are unified. Therefore, we use the "Buffer Slot Availability" (BSA) tracking system to allocate/deallocate arriving/departing flits with buffer slots. The input DEMUX switches to the buffer slot provided by the BSA at the input flit tracking. For a departing flit, the BSA deallocates the buffer slot using the output flit tracking and adds it to the list of free slots maintained in the buffer slot table (shown in the inset of Figs. 5a and 5b). The number of buffer slots depends on the maximum number of credits available for a particular VC. For purposes of fairness, the number of credits is equally divided between the VCs as $\lfloor (z + c)/v \rfloor$ per VC slot. When the BSA does not find a nonnull pointer in its base table, it triggers the congestion signal.

In the example shown in Fig. 5a, VCs 0, 1, and 3 are allocated and are currently in the SA, VC, and SA stages, respectively. Suppose the newly arriving flit has a VCID of 1. First, the BSA tracks the input flit to the appropriate buffer slot. As the only free slot in the BSA table is 6 (highlighted in Fig. 5a), it allocates the incoming flit to slot 6. As there are no more free slots, the BSA enables the congestion control signal. The UVST then updates the F_1 slot for VC 1 (circled in Fig. 5a). Fig. 5b shows the congestion being released. Here, VC 0 has been allocated the switch and its status is updated to ST. This causes the RP to read the flit from buffer slot 3 into the crossbar. BSA tracks the output flit, clears the field for three, and releases the congestion signal.

Dynamic buffer management is not as rigid as the static allocation scheme and tends to eliminate the HoL blocking to some extent. If the number of VCs increased/decreased dynamically as in ViChaR, then the HoL blocking can be completely eliminated. Although our design can be extended to incorporate dynamic VA such as ViChaR, this has



(a)

VC	RP	WP	OP	OVC	CR	Status	F_0	F_1	F_3	F_4
0	3	N	0	1	4	SA	2	3	4	7
1	1	N	2	N	4	VC	1	N	N	N
2	N	N	N	N	4	Idle	N	N	N	N
3	0	N	5	2	4	SA	0	5	N	N

Buffer Slot	Free
0	N
1	N
2	N
3	N
4	N
5	N
6	Y
7	N

(b)

VC	RP	WP	OP	OVC	CR	Status	F_0	F_1	F_3	F_4
0	3	N	0	1	4	ST	2	3	4	7
1	1	N	2	N	4	VC	1	N	N	N
2	N	N	N	N	4	Idle	N	N	N	N
3	0	N	5	2	4	SA	0	5	N	N

Buffer Slot	Free
0	N
1	N
2	N
3	Y
4	N
5	N
6	N
7	N

Fig. 5. Proposed dynamic buffer allocation with congestion control, showing the congestion being set in (a) and congestion being released in (b), for $v = 4$ (VCs/port), $z = 8$ (total buffers/port), and $c = 8$ (channel buffers).

not been explored in this paper. Our objective has been to reduce the HoL blocking that exists due to the channel buffers. These channel buffers can be viewed as serial FIFO buffers as opposed to the parallel FIFO buffers used within the routers. Therefore, eliminating the HoL blocking is critical in our new design. Static allocation of buffer slots simplifies the overall design as it requires minimum extension over a generic OCIN router architecture. Dynamic allocation of buffer slots along with the table-based design significantly reduces the HoL blocking. This achieves much higher throughput with significant savings in power consumption and chip area.

4 Performance Evaluation

In this section, we evaluate the router buffers and the proposed channel buffers in terms of power dissipation, area overhead, and overall network performance. We consider 8×8 mesh and folded torus topologies with a four-stage pipelined router design. Each router has $P = 5$ input ports (four for each direction and one for the PE). The baseline design considered has four VCs per input port, with each VC having four flit buffers in the router, for a total of 80 flit buffers ($= 5 \times 4 \times 4$). Each packet consists of four flits and each flit is 128 bits wide. For the design with channel buffers, we consider four different cases where some or all of the repeaters along the channel are replaced by channel buffers. The notation followed for the different cases is of $vn_V - rn_R - cn_C$, where n_V is the number of VCs per input port, n_R is the number of router buffers per VC, and n_C is the number of channel buffers. For example, the baseline is denoted as $v4 - r4 - c0$, implying four VCs per input port, four router buffers per VC, and zero channel buffers. For a fair comparison with the baseline, the number of buffers eliminated from the router is added to the set of channel buffers. In each case, the design is implemented in Verilog and synthesized using the Synopsys Design Compiler tool with the TSMC 90 nm technology library, with a supply voltage of 1 V and an operating frequency of 500 MHz.

4.1 Channels

The channels between the routers are implemented in the semiglobal or intermediate metal layers, as the local metal layers are reserved for the processor and the global metal layers are used by the power/clock distribution signals [9].

4.1.1 Channel Power Estimation

The power per segment of the repeater-inserted channel is given by

$$P_{segment} = P_{dynamic} + P_{leakage} + P_{short\text{-}ckt}, \tag{1}$$

where $P_{dynamic}$ is the switching power, $P_{leakage}$ is the power due to the subthreshold leakage current, and $P_{short\text{-}ckt}$ is the power due to the short-circuit current. The power per segment is multiplied by the number of segments and the channel width to obtain the total channel power dissipation for a flit traversal. When a conventional repeater is replaced by a channel buffer, there is an additional capacitance, C_{buf}, due to the added transistors, as shown in Fig. 1b. The components of the total power increase due to the additional capacitance, as given by

$$\widetilde{P}_{dynamic} = \alpha \times \left[k(C_o + C_p + C_{buf}) + \ell C_w\right] \times V_{DD}^2 \times freq, \tag{2}$$

$$\widetilde{P}_{leakage} = 2 \times \left[1/2 \times V_{DD} \times \left(I_{off}(W_N + W_P)k\right)\right], \tag{3}$$

$$\widetilde{P}_{short\text{-}ckt} = \alpha \times \widetilde{t}_{rise} \times W_N \times k \times V_{DD} \times I_{sc} \times freq, \tag{4}$$

where $\widetilde{P}_{dynamic}$ is the dynamic power, $\widetilde{P}_{leakage}$ is the leakage power, $\widetilde{P}_{short\text{-}ckt}$ is the short-circuit power of a channel buffer inserted segment along the channel, α is the activity factor, k is the repeater sizing, ℓ is the repeater spacing, V_{DD} is the supply voltage, $freq$ is the operating frequency, C_o and C_p are the device diffusion and gate capacitances, respectively, C_w is the wire capacitance per unit length, I_{sc} is the device short-circuit current, I_{off} is the leakage current, W_N (W_P) is the width of the NMOS (PMOS) transistor in the channel buffer, and $\widetilde{t}_{rise}$ is the rise time of the short-circuit current pulse in the channel buffer.

There is one control block for every stage of the channel buffers. The control block is switched "ON" during congestion and power $P_{ctrl\text{-}blk}$ is dissipated within the block due to the inverters (P_{inv}) and the switched capacitor ($P_{sw\text{-}cap}$). The power P_{clk} consumed by the block supplying the clock signals to the control blocks is the sum of the dynamic, leakage, and short-circuit powers of the individual gates in the block. In the absence of congestion, the channel buffers function like repeaters and the power dissipated per segment of the channel is $\widetilde{P}_{segment_{repeater}}$ and is given by

$$\widetilde{P}_{segment_{repeater}} = \widetilde{P}_{dynamic} + \widetilde{P}_{leakage} + \widetilde{P}_{short\text{-}ckt}. \qquad (5)$$

During congestion, the channel buffers store the data in position and the power dissipated is

$$\widetilde{P}_{segment_{chl-buffer}} = \widetilde{P}_{leakage} + P_{ctrl\text{-}blk} + P_{clk}. \qquad (6)$$

The channels between the routers are assumed to be 2 mm long for the mesh network. The average channel length doubles in the case of the folded torus network [9] and, hence, the channels are 4 mm long. Compared to off-chip networks, OCINs have abundant wiring resources that can be efficiently utilized to improve the network performance [2], [6]. Channels with widths such as 128 [8], [10] and 256 bits [21] have been explored in the context of OCINs. Therefore, in our design, we employ channels that are capable of handling 128-bit wide flits. In obtaining the power values, the power-optimal repeater insertion methodology described in [23] has been used. In the baseline design, there are eight conventional repeaters along each wire of the 128-bit wide channels. The total power consumed by the channel per flit traversal is 2.45 mW for the 8×8 mesh and 3.94 mW for the 8×8 folded torus. When all eight conventional repeaters are replaced by channel buffers, the total power consumed in the channel for every flit traversal is found to be 3.55 mW for the mesh and 5.04 mW for the folded torus. In the presence of congestion, the power dissipated by each control block is found to be 2.089 μW. The power due to the block supplying the clocks to the control blocks is 3.82 μW. The additional control logic thus consumes only a small fraction of the total power dissipated in the channels.

4.1.2 Channel Area Estimation

The area of the channels between the routers is determined by the area of the repeaters and the wires. The repeaters and the wires utilize different metal layers and their area overheads are independent of each other [24]. The area consumed by the wires is given by the product of the channel bit width, the wire pitch, and the wire spacing in the given technology. In the 8×8 mesh network, 128 wires in each channel occupy an area of 0.1536 mm^2. The area doubles to 0.3072 mm^2 for the 8×8 folded torus network [9]. This area is constant across all of the design cases considered since the bit width of the channel remains constant. The repeater area is given by

$$Area_{repeaters} = k \times Area_{min} \times N_R \times N_W, \qquad (7)$$

where $Area_{min}$ is the area of a minimum-sized inverter at the 90 nm technology considered, N_W is the bit width of the channel, and N_R is the number of repeaters along the channel. When the conventional repeaters are replaced by the channel buffers, the area increases due to the additional transistors in the channel buffers. The area overhead due to the control block for each channel buffer stage is the sum of the individual transistor areas in the block and is negligible compared to the overall channel buffer area. The area occupied by the repeater stages along a 128-bit channel is found to be 32 $\mu\mathrm{m}^2$ in case of the baseline and 80 $\mu\mathrm{m}^2$ when all eight conventional repeaters are replaced by channel buffers.

4.2 Router

4.2.1 Router Power Estimation

This section summarizes the power estimation for the buffers, the crossbar, and the arbiter in the router. The router buffers are implemented as FIFO registers with the associated control logic.

The dynamic power consumed by the router buffer is the sum of the power expended in writing a flit into the buffer and the power consumed to read out the flit from the buffer, as indicated by the WP and the RP from the VC state table. The leakage power consumption of the buffer is the product of the supply voltage and the total leakage current in the buffer. When the number of VCs or the buffer depth per VC is reduced, the size and number of components within the buffer are also reduced, decreasing the power consumption.

Considering both the write and read operations in the router buffer, the total power consumed for a flit traversal in the buffer is found to be 19.54 mW for the baseline design with 16 buffer slots and no channel buffers. When the buffer size is reduced to 50 percent of the baseline, power consumed per flit traversal decreases to 11.57 mW, saving 40.77 percent in the buffer power alone.

A two-stage matrix arbiter design [31] is considered with the first stage selecting one output from the v VCs of a port and the second stage arbitrating among the Pv inputs from each of the P ports. In the case of four VCs, the two-stage arbiter consumes a power of 0.15 mW for a single arbitration task. When the number of VCs is decreased to 3, the power consumed by the arbiter reduces to 0.09 mW per arbitration. The switch in the router consumes 0.31 mW per flit traversal, in the case of four VCs per port, and 0.27 mW per flit traversal in the case of three VCs per port.

4.2.2 Router Area Estimation

The areas of the router buffers, the arbiter, and the switch are obtained from the synthesized designs using the Synopsys Design Compiler tool and the TSMC 90 nm technology library. In the case of the baseline design with 16 buffer slots in the router, the buffer area is 81,407 $\mu\mathrm{m}^2$. A 50 percent decrease in the buffer size leads to a 40.88 percent reduction in the buffer area.

The area of the crossbar is given by the number of input/output signals that it should accommodate. The area of the arbiter is the area occupied by the two levels of NOR gates [31] and is minimal compared to the wire-dominated area of the crossbar.

TABLE 1
Power Estimation for Various Channel and Router Buffer Configurations in an 8×8 Mesh and 8×8 Folded Torus Networks

vn_V- rn_R- cn_C	Router Buffer Power (mW)	Mesh Channel + Control Block Power (mW)	Folded Torus Channel + Control Block Power (mW)	Mesh Total Power (Router Buffer + Channel + Control Block) (mW)	Mesh % Change in Total Power	Folded Torus Total Power (Router Buffer + Channel + Control Block) (mW)	Folded Torus % Change in Total Power
v4-r4-c0	19.54	2.45 + 0	3.94 + 0	21.99	–	23.48	–
v5-r3-c1	19.29	2.81 + 0.005	4.28 + 0.005	22.10	+0.50	23.57	+0.03
v3-r4-c4	15.09	2.90 + 0.012	4.39 + 0.012	18.00	-18.14	19.49	-16.99
v4-r3-c4	14.51	2.90 + 0.012	4.39 + 0.012	17.42	-20.78	18.91	-19.46
v4-r2-c8	11.57	3.55 + 0.020	5.04 + 0.020	15.14	-31.15	16.63	-29.17

Power values are for one flit traversal. n_V is the number of VCs per input port, n_R is the number of router buffers per VC, and n_C is the number of channel buffers.

4.3 Comparison of the Different Cases

Table 1 shows a comparison of the power estimations per flit traversal for various channel and router buffer configurations. The first configuration shown is the baseline case $(v4 - r4 - c0)$ and uses no channel buffers. Change in power in each of the other cases is expressed as a percentage increase $(+)$ or a percentage decrease $(-)$ with respect to the baseline. The total power per flit traversal for the baseline is 21.99 mW in the mesh network. When 50 percent of the router buffers are removed and all of the repeaters along the channel are replaced with channel buffers, the total power per flit traversal reduces to 15.14 mW as seen in the $v4 - r2 - c8$ case, giving 31.15 percent savings in total network power per flit traversal. The corresponding savings in total power in the case of the folded torus network is 29.17 percent.

4.4 Simulation Methodology

A cycle-accurate on-chip network simulator was used to conduct a detailed evaluation of the proposed channel and router buffer architecture in both 8×8 mesh and 8×8 folded torus networks. The test configurations are represented in the results as $vn_V - rn_R - cn_C$, where n_V is the number of VCs per input port, n_R is the number of router buffers per VC, and n_C is the number of channel buffers. For simplicity, they will be referred to as $n_V - n_R - n_C$ in the following discussion. The test configurations evaluated were $n_V - n_R - n_C = 4 - 4 - 0$ (baseline), 4-3-4, 4-2-8, 3-4-4, and 5-3-1. For synthetic traffic patterns, packets were injected according to Bernoulli process based on the network load for a given simulation run. The network load is varied from 0.1 to 0.9 of the network capacity. The simulator was warmed up under load without taking measurements until steady state was reached. Then, a sample of injected packets was labeled during a measurement interval. The simulation was allowed to run until all of the labeled packets reached their destinations. For the SPLASH-2 suite benchmarks [32], the network traces were gathered by running the benchmarks on the Rice Simulator for ILP Multiprocessors (RSIM) [33] for 64 nodes. RSIM models Modified, Exclusive, Shared, and Invalid (MESI) directory-based cache coherence protocol, with the home node assigned based on the first touch policy. The access patterns for the protocol implementation with precise timing information are gathered and then simulated on our proposed cycle-accurate network simulator.

We tested our hypothesis of using static and dynamic buffer allocation schemes on several traffic patterns such as: 1) Uniform Random, where each node randomly selects its destinations with equal probability and 2) Permutation Patterns, where each node selects a fixed destination based on permutations. We evaluated the performance on the following permutation patterns: Bit-Reversal (BR), Butterfly (BU), Matrix Transpose (MT), Complement (CO), Tornado (TO), Perfect Shuffle (PS), Neighbor (NE); and 3) SPLASH-2 suite benchmarks covering a spectrum of memory sharing and access patterns [32], including FFT with input data set $64K$ points; LU with 256×256, 16×16 block; MP3D with 48,000 molecules; Radix with $1M$ integers, 1,024 radix, and Water-nsquared with 512 molecules.

4.5 Simulation Results and Discussion

The following discussion presents the simulation results (input buffer power consumed, throughput, average latency, overall network power, and the occurrence of congestion in the network) for the individual cases, a comparison of the throughput and the input buffer power for all of the synthetic traffic patterns considered, and a comparison of the performance and overall network power for the SPLASH-2 suite benchmarks.

Input buffer power. Fig. 6 shows the total power dissipated by the input buffers in the router, with static and dynamic buffer allocations for uniform (UN) and CO traffic patterns in the 8×8 mesh and folded torus networks for a network load of 0.5.

For the mesh topology, the power savings in the 4-3-4 configuration using dynamic buffer allocation is nearly 24 percent. The power savings for the 4-2-8 configuration (reducing the buffer depth from 4 to 2) is about 40 percent. The 3-4-4 and 5-3-1 configurations show 22 percent and 10 percent savings in buffer power alone. Therefore, in the dynamic case, the power savings by reducing the buffer in half is almost 40 percent. Under static buffer allocation for the mesh network, a reduction of the buffer size in half causes a power savings of almost 53 percent. The power savings observed for the 4-3-4 configuration is 33 percent while, for the 5-3-1 configuration, power decreases by nearly 13 percent. Similar results are observed for the folded torus topology with the 4-2-8 configuration achieving a power savings of almost 39 percent for the dynamic case and 53 percent for the static case. Therefore, in both the static and dynamic cases, significant power savings is obtained by reducing the buffer size.

Throughput, latency, and power. Figs. 7 and 8 show the throughput, average latency, and overall network power for UN and CO traffic patterns using static and dynamic buffer allocation for varying network load, for the 8×8 mesh and

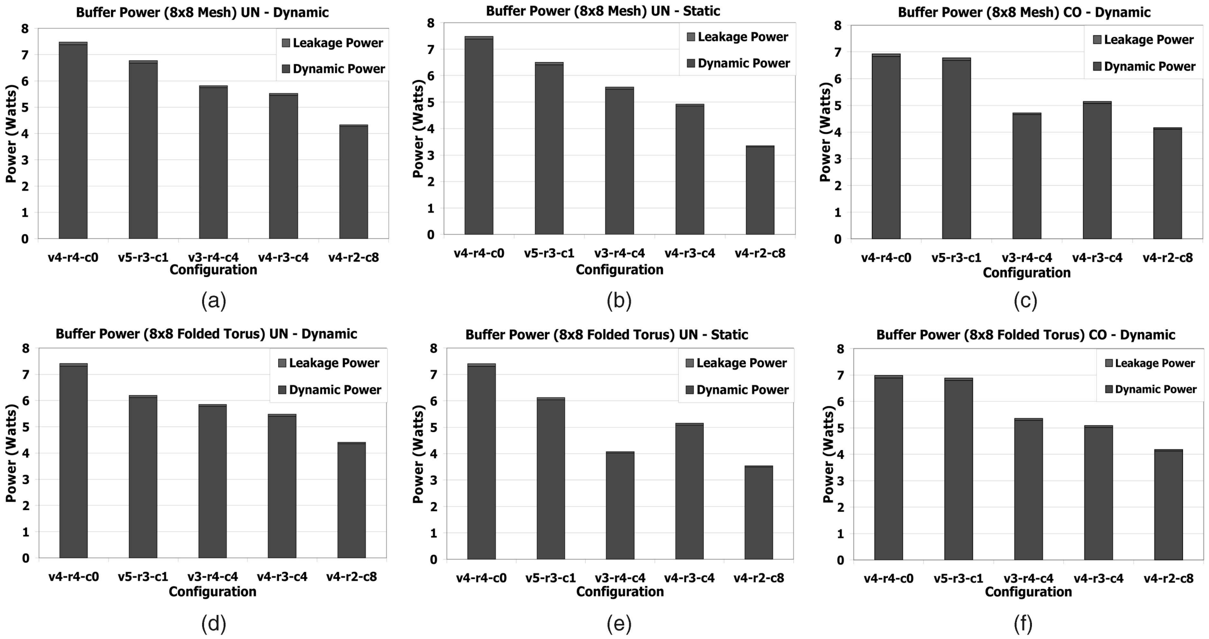
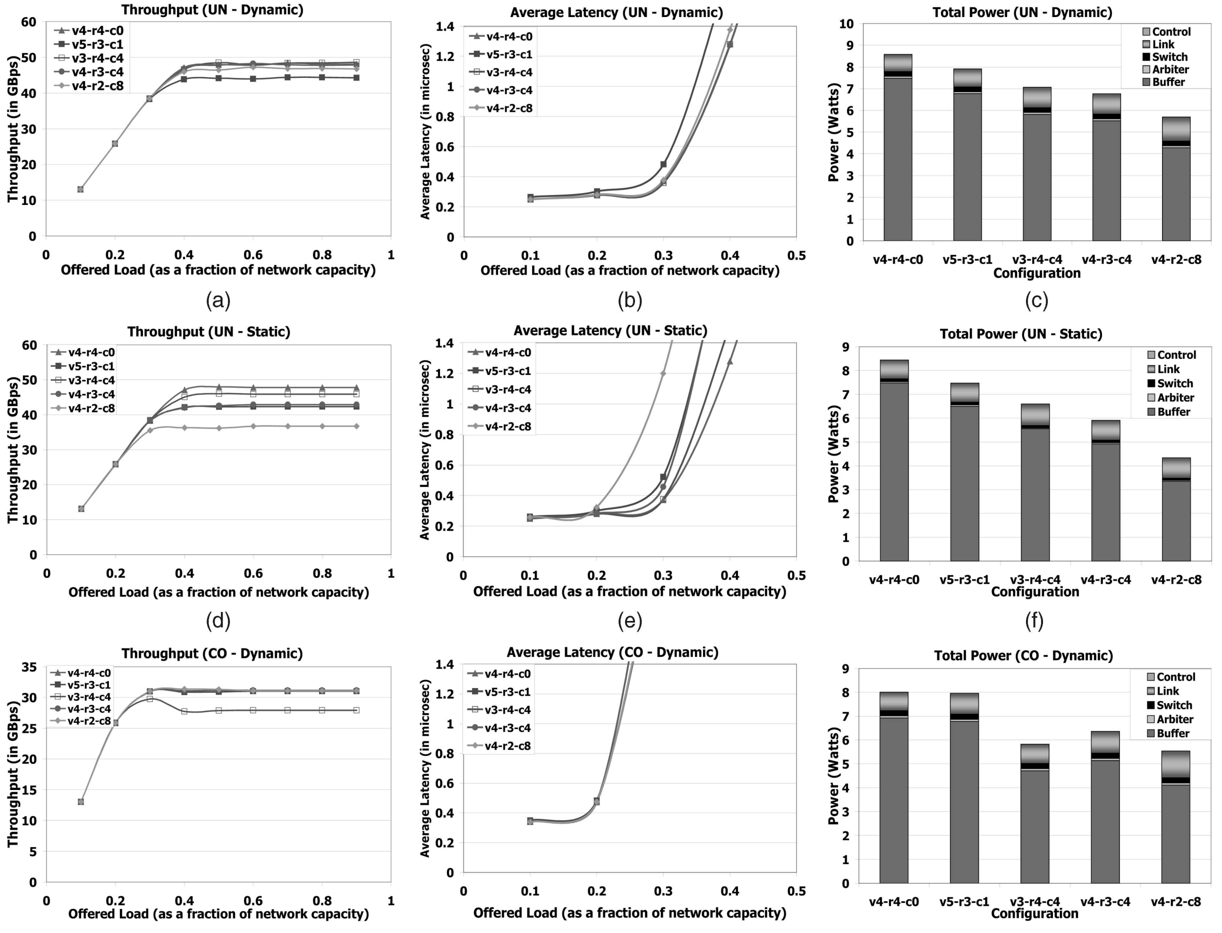


Fig. 6. Total power dissipated by the input buffers in the router with static and dynamic buffer allocations for UN and CO traffic patterns in 8×8 mesh and folded torus networks for a network load of 0.5. The configurations tested were $vn_V - rn_R - cn_C$, where n_V is the number of VCs per input port, n_R is the number of router flit buffers per VC, and n_C is the number of channel buffers.

folded torus networks, respectively. From Fig. 7, for the dynamic buffer allocation in the mesh topology, the throughput shows almost similar performance for 4-4-0, 3-4-4, and 4-3-4 under uniform traffic. The decrease in the number of VCs for the 3-4-4 or the buffer depth for the 4-3-4 does not significantly affect the throughput. The more interesting point is 4-2-8, which shows only about 3 percent drop in performance. This result is significant as we can save nearly 41 percent of the buffer size and yet achieve similar performance as the baseline configuration by dynamically allocating the buffer resources to flits. At high network loads, the congestion signal prevents or throttles the data movement into the router buffer. As we have additional buffers in the channel, the flow of data flits is not hampered even though we have fewer buffers in the routers. For the 5-3-1 configuration, the drop in throughput is about 6 percent as compared to the baseline. The increased number of VCs does not yield any tangible benefits in this case. For all configurations except 5-3-1, the network saturates at 0.35. Under the complement traffic pattern, the throughput is almost the same for all configurations except the 3-4-4, which shows a decrease of about 5 percent. From Fig. 7, for the static buffer allocation in the mesh network, the performance degradation compared to the baseline increases as the depth of the buffer is reduced. The 4-2-8 configuration shows almost 20 percent reduction in throughput. The HoL blocking causes the network throughput to degrade as flits get stuck behind a blocked flit. The 5-3-1 and 4-3-4 configurations show a 12.5 percent drop in throughput, whereas the 3-4-4 shows only a 6 percent drop. The average network latency shown in Fig. 7 reflects the HoL effects on various configurations in the mesh network. When the depth of the buffer reduces, it affects the throughput and increases the network latency. The total power consumed in the mesh network is shown in Fig. 7 for a network workload of 0.5. In the case of dynamic buffer allocation, for uniform traffic, the 4-2-8 configuration shows a decrease of almost 30 percent of the overall network power for the mesh network. The 4-3-4 and 3-4-4 configurations show a reduction of almost 20 percent of the network power, while the 5-3-1 configuration shows the network power reducing by almost 10 percent. Therefore, by reducing the buffer size in half for the 4-2-8 configuration, we achieve almost 30 percent reduction in total network power including the channel, the input buffers, the crossbar switch, and the arbiter. All of the configurations achieve a reduction in power compared to the baseline. The power dissipation trends in the mesh network for the static case show that in the 4-2-8 configuration, the power savings is almost 50 percent, which is nearly 20 percent more savings than the dynamic case. However, this savings comes at the cost of the reduced throughput for the static allocation.

From Fig. 8, for the dynamic buffer allocation in the folded torus topology, all of the configurations except 5-3-1 show similar performance for both traffic patterns. This is significant since a reduction in the buffer depth or a decrease in the number of VCs has not degraded the performance compared to the baseline. For the static case shown in Fig. 8, throughput drops by about 16 percent for the 4-2-8 configuration. The average latency plots shown in Fig. 8 for the folded torus network indicate that the network saturates at about 0.4 under uniform traffic (for all configurations except the 5-3-1) and at about 0.2 under complement traffic. Fig. 8 also shows the total power consumed in the folded torus network for a network workload of 0.5 under uniform and complement traffic. In the case of dynamic buffer allocation, for uniform traffic, the 4-2-8 configuration shows a decrease of about 30 percent. All of the configurations achieve a reduction in power



(a) (b) (c)

(d) (e) (f)

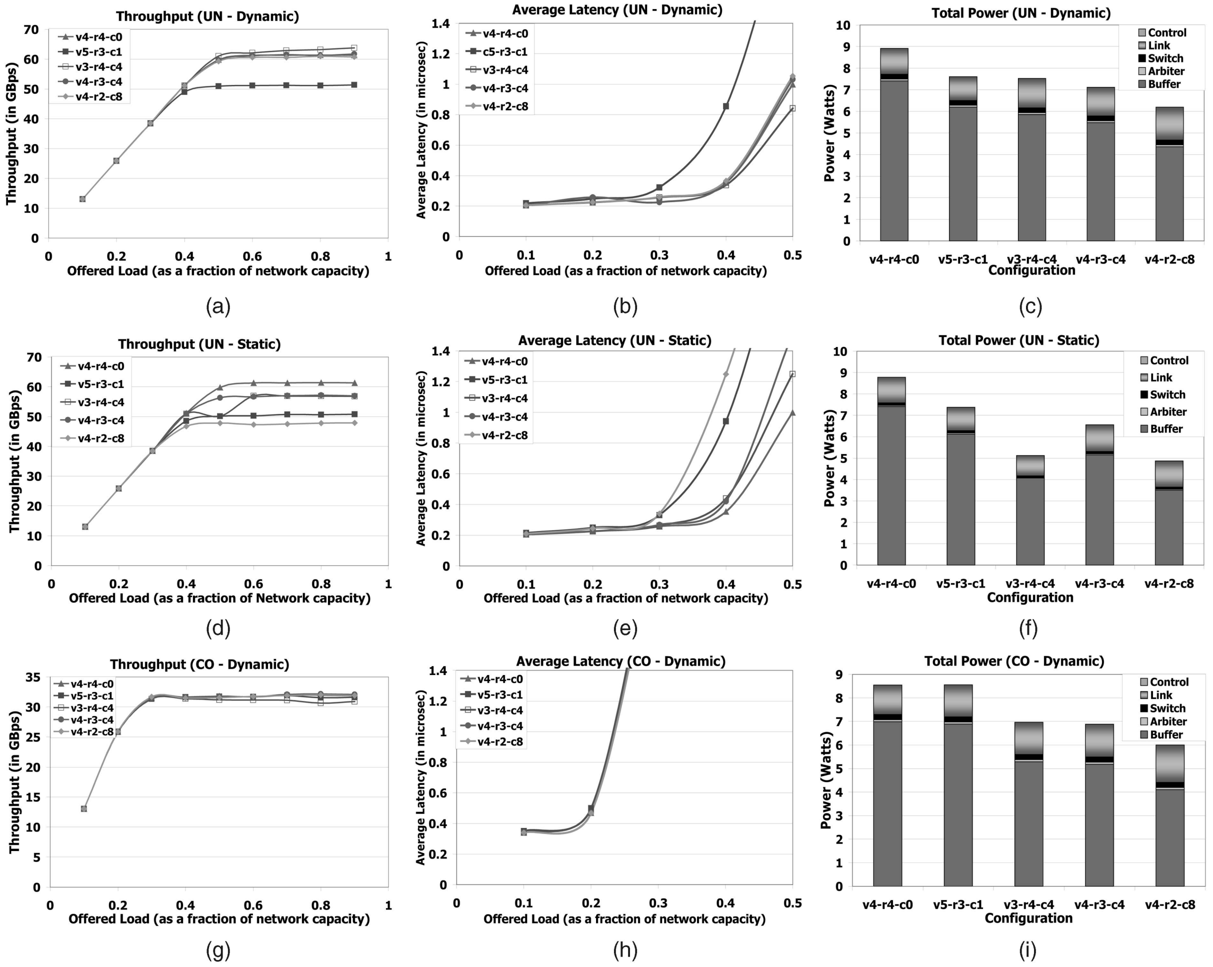
(g) (h) (i)

Fig. 7. Throughput, latency, and overall network power for UN and CO traffic patterns using static and dynamic buffer allocations for varying network loads for an 8×8 mesh network.

compared to the baseline, as seen in the case of the mesh network.

Congestion variation. Fig. 9 shows the occurrence of congestion in the 8×8 mesh and folded torus networks for static and dynamic buffer allocations under uniform traffic. Congestion in the network indicates that the channel buffers are enabled to hold the data along the links. The baseline configuration (4-4-0) does not make use of channel buffers and, therefore, the occurrence of congestion is shown to be zero for the baseline under all the cases considered. In both the mesh and the folded torus topologies, the 4-2-8 configuration shows the highest occurrence of congestion. This is due to the reduction of the router buffer size in half compared to the baseline. Reduction in the router buffer size causes the flits to be blocked due to insufficient buffer space as the network load increases. Static buffer allocation shows a higher occurrence of congestion than the dynamic case by about 18 percent for the 4-2-8 configuration. Although the 4-2-8 configuration shows the maximum congestion in the network, the corresponding network performance does not drop significantly as the adaptive channel buffers hold the data along the links and prevent loss of data due to congestion. The 5-3-1 configuration shows a low occurrence of congestion compared to the other configurations due to a higher number of VCs. The folded torus topology shows a low congestion occurrence compared to the mesh network due to the flexibility provided by the additional end-around links.

Throughput and router buffer power for all traffic patterns. Figs. 10a and 10c show the power consumed at the input buffers of the router and the throughput achieved at a network load of 0.5 for the 8×8 mesh network, with static and dynamic buffer allocations for all of the synthetic traffic patterns including UN, CO, PS, BU, BR, MT, NE, and TO for three configurations, namely, 4-4-0, 4-3-4, and 4-2-8. From Fig. 10a, it can be observed that, irrespective of whether the buffer allocation is static or dynamic, power savings is obtained for all of the traffic patterns. The power saving seen with static allocation is slightly more than the power saving observed with dynamic allocation. For the Complement traffic pattern, static buffer allocation provides 57 percent savings in router buffer power for the 4-2-8 configuration as compared to the baseline, whereas, with dynamic buffer allocation, the savings decreases to 40 percent. Fig. 10c shows no appreciable decrease in throughput for the dynamic case for all of the traffic patterns. Dynamic buffer allocation provides the flexibility for the flits to be allocated to any available buffer slot and is not as restrictive as the static allocation.

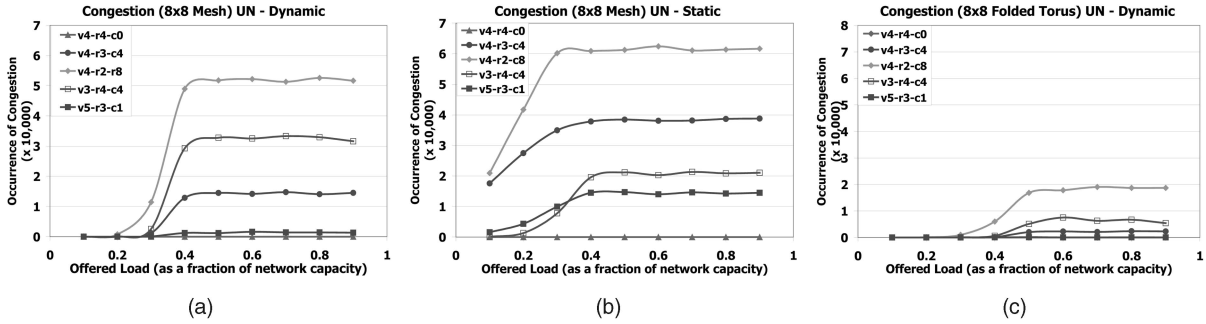


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 8. Throughput, latency, and overall network power for UN and CO traffic patterns using static and dynamic buffer allocations for varying network loads for an 8×8 folded torus network.



(a) (b) (c)

Fig. 9. Congestion in the 8×8 mesh and the 8×8 folded torus networks for static and dynamic UN traffic.

Throughput and power for SPLASH-2 suite benchmarks. Figs. 10b and 10d show the normalized total power consumed and the normalized execution time for the selected SPLASH-2 suite benchmarks for the 4-4-0, 4-3-4, and 4-2-8 configurations with dynamic buffer allocation. The normalization is carried out with respect to the baseline 4-4-0 configuration. From Fig. 10b, the power savings from the proposed 4-3-4 and 4-2-8 configurations are 20 percent and 30 percent, respectively. From Fig. 10d, the 4-3-4 and 4-2-8 configurations do not show a significant drop in performance; in fact, the drop is less than 1 percent. Therefore, dynamic allocation with channel buffers does not degrade performance and provides significant power savings for all SPLASH-2 suite benchmarks.

5 Conclusion

As recent research on OCINs has shown that the design of the router buffers influences the energy consumption, area overhead, and overall performance of the network, our proposed architecture attempts to reduce the size of the buffers within the routers. As this impacts performance, we have provided additional adaptive channel buffers which can be used to store data along the channels only when required. A combination of circuit and architectural

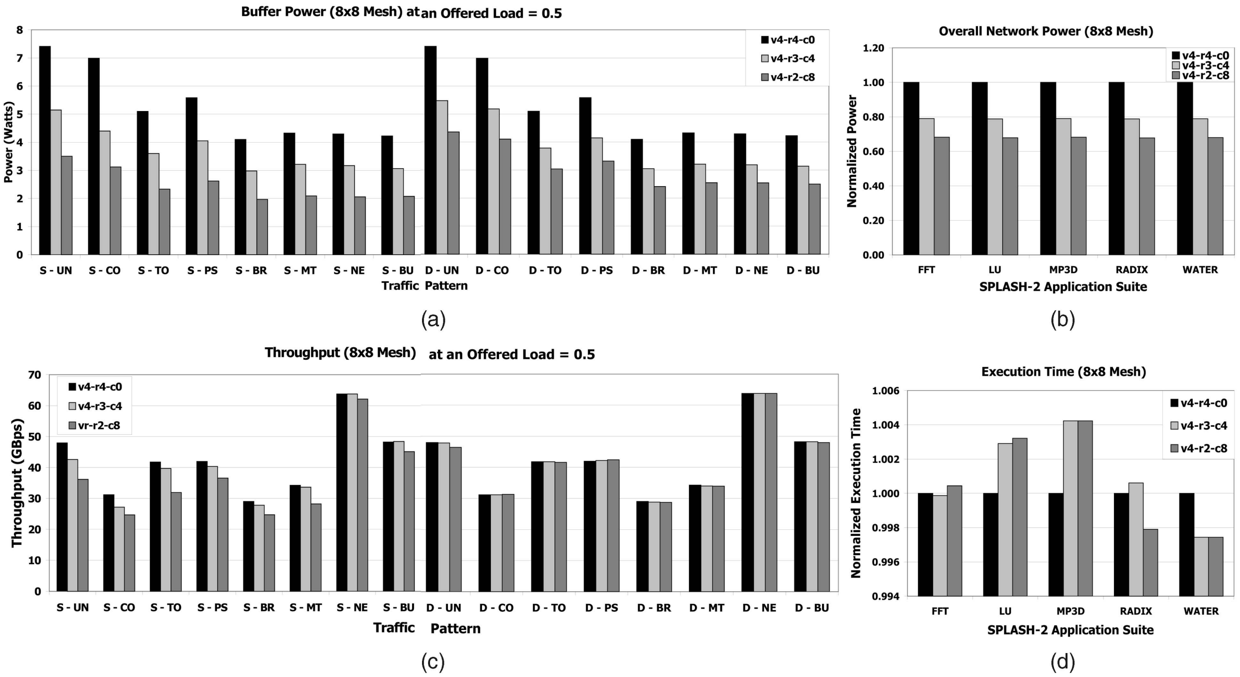


Fig. 10. (a) Buffer power for synthetic traffic. (b) Normalized overall network power for SPLASH-2 benchmarks. (c) Throughput for synthetic traffic. (d) Normalized execution time for SPLASH-2 benchmarks for the 8×8 mesh network. Synthetic traffic patterns (UN, CO, TO, PS, BR, MT, NE, and BU) are considered under both static (S) and dynamic (D) buffer allocation schemes.

techniques unique to our proposed architecture allows us to reduce the router buffer size without significantly degrading the network performance. Simulation results using the SPLASH-2 application suite as well as synthetic traffic patterns in the 90 nm technology node show that, by eliminating some of the router buffers, our proposed architecture achieves nearly 40 percent savings in router buffer power, 30 percent savings in overall network power, and a 41 percent savings in area, with only a 1-5 percent drop in throughput for dynamically assigned buffers and a 10-20 percent drop in throughput for static buffer allocation.

ACKNOWLEDGMENTS

This research was partially supported by US National Science Foundation Grants CCR-0538945 and ECCS-0725765. The authors would like to thank Dr. Dong Sheng (Brian) Ma and Minkyu Song for their assistance with the switched capacitor control block. The authors would also like to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] L. Benini and G.D. Micheli, "Networks on Chips: A New SoC Paradigm," *Computer,* vol. 35, pp. 70-78, 2002.

[2] W.J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," *Proc. Design Automation Conf.,* June 2001.

[3] R. Ho, K.W. Mai, and M.A. Horowitz, "The Future of Wires," *Proc. IEEE,* vol. 89, pp. 490-504, Apr. 2001.

[4] L.P. Carloni and A.L. Sangiovanni-Vincentelli, "Coping with Latency in SOC Design," *IEEE Micro,* vol. 22, no. 5, pp. 24-35, Sept./Oct. 2002.

[5] P.P. Pande, C. Grecu, A. Ivanov, and R. Saleh, "Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures," *IEEE Trans. Computers,* vol. 54, no. 8, pp. 1025-1040, Aug. 2005.

[6] S. Heo and K. Asanovic, "Replacing Global Wires with an On-Chip Network: A Power Analysis," *Proc. Int'l Symp. Low Power Electronics and Design,* pp. 369-374, Aug. 2005.

[7] J. Hu and R. Marculescu, "Application-Specific Buffer Space Allocation for Network-on-Chip Router Design," *Proc. IEEE/ACM Int'l Conf. Computer Aided Design,* pp. 354-361, Nov. 2004.

[8] C.A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M.S. Yousif, and C.R. Das, "ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers," *Proc. 39th Ann. Int'l Symp. Microarchitecture,* pp. 333-344, Dec. 2006.

[9] J. Balfour and W.J. Dally, "Design Tradeoffs for Tiled CMP On-Chip Networks," *Proc. 20th ACM Int'l Conf. Supercomputing,* pp. 187-198, June 2006.

[10] H.S. Wang, L.S. Peh, and S. Malik, "Power-Driven Design of Router Microarchitectures in On-Chip Networks," *Proc. 36th Ann. ACM/IEEE Int'l Symp. Microarchitecture,* pp. 105-116, Dec. 2003.

[11] S. Kumar, A. Jantsch, M. Millberg, J. Oberg, J.P. Soininen, M. Forsell, K. Tiensyrja, and A. Hemani, "A Network on Chip Architecture and Design Methodology," *Proc. IEEE CS Ann. Symp. VLSI,* p. 117, Apr. 2002.

[12] P. Guerrier and A. Greiner, "A Generic Architecture for On-Chip Packet-Switched Interconnections," *Proc. Conf. Design, Automation and Test in Europe,* pp. 250-256, Mar. 2000.

[13] P. Kundu, "On-Die Interconnects for Next Generation CMPs," *Proc. Workshop On- and Off-Chip Interconnection Networks for Multicore Systems,* Dec. 2006.

[14] J. Kim, C.A. Nicopoulos, D. Park, N. Vijaykrishnan, M.S. Yousif, and C.R. Das, "A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks," *Proc. 33rd Ann. Int'l Symp. Computer Architecture,* pp. 4-15, June 2006.

[15] R. Mullins, A. West, and S. Moore, "Low-Latency Virtual Channel Routers for On-Chip Networks," *Proc. 31st Ann. Int'l Symp. Computer Architecture,* pp. 188-197, June 2004.

[16] L.S. Peh and W.J. Dally, "A Delay Model and Speculative Architecture for Pipelined Routers," *Proc. Seventh Int'l Symp. High-Performance Computer Architecture,* pp. 255-266, Jan. 2001.

[17] E.J. Kim, K.H. Yum, G.M. Link, N. Vijaykrishnan, M. Kandemir, M.J. Irwin, M. Yousif, and C.R. Das, "Energy Optimization Techniques in Cluster Interconnects," *Proc. Int'l Symp. Low Power Electronics and Design,* pp. 459-464, Aug. 2003.

[18] L. Shang, L.S. Peh, and N.K. Jha, "Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks," *Proc. Seventh Int'l Symp. High-Performance Computer Architecture,* pp. 91-102, Feb. 2003.

[19] Q. Wu, P. Juang, M. Martonosi, L.S. Peh, and D.W. Clark, "Formal Control Techniques for Power-Performance Management," *IEEE Micro,* vol. 25, no. 5, Sept./Oct. 2005.

[20] J. Hu and R. Marculescu, "DyAD—Smart Routing for Networks-on-Chip," *Proc. 41st IEEE/ACM Design Automation Conf.,* June 2004.

[21] W.J. Dally and B. Towles, *Principles and Practices of Interconnection Networks.* Morgan Kaufmann, 2004.

[22] W.J. Dally, "Virtual-Channel Flow Control," *Proc. 17th Ann. Int'l Symp. Computer Architecture,* pp. 60-68, June 1990.

[23] K. Banerjee and A. Mehrotra, "A Power-Optimal Repeater Insertion Methodology for Global Interconnects in Nanometer Designs," *IEEE Trans. Electron Devices,* vol. 49, no. 11, pp. 2001-2007, Nov. 2002.

[24] M.A. El-Moursy and E.G. Friedman, "Optimum Wire Sizing of RLC Interconnect with Repeaters," *Integration, the VLSI J.,* vol. 38, no. 2, pp. 205-225, Dec. 2004.

[25] M.L. Mui, K. Banerjee, and A. Mehrotra, "A Global Interconnect Optimization Scheme for Nanometer Scale VLSI with Implications for Latency, Bandwidth, and Power Dissipation," *IEEE Trans. Electron Devices,* vol. 51, no. 2, pp. 195-203, Feb. 2004.

[26] M. Mizuno, W.J. Dally, and H. Onishi, "Elastic Interconnects: Repeater-Inserted Long Wiring Capable of Compressing and Decompressing Data," *Proc. IEEE Int'l Solid-State Circuits Conf.,* pp. 346-347, Feb. 2001.

[27] Y.M. Boura and C.R. Das, "Performance Analysis of Buffering Schemes in Wormhole Routers," *IEEE Trans. Computers,* vol. 46, pp. 687-694, 1997.

[28] N. Ni, M. Pirvu, and L. Bhuyan, "Circular Buffered Switch Design with Wormhole Routing and Virtual Channels," *Proc. Int'l Conf. Computer Design,* pp. 466-473, Oct. 1998.

[29] Y. Tamir and G.L. Frazier, "High-Performance Multiqueue Buffers for VLSI Communication Switches," *Proc. 15th Ann. Symp. Computer Architecture,* pp. 343-354, May-June 1988.

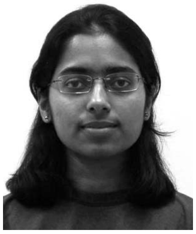
[30] M. Rezazad and H. Sarbazi-azad, "The Effect of Virtual Channel Organization on the Performance of Interconnection Networks," *Proc. 19th Int'l Parallel and Distributed Processing Symp.,* Apr. 2005.

[31] H.S. Wang, X. Zhu, L.S. Peh, and S. Malik, "Orion: A Power-Performance Simulator for Interconnection Networks," *Proc. 35th Ann. ACM/IEEE Int'l Symp. Microarchitecture,* pp. 294-305, Nov. 2002.

[32] C.S. Woo, M. Ohara, E. Torrie, J.P. Singh, and A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerations," *Proc. 22nd Ann. Int'l Symp. Computer Architecture,* pp. 24-37, June 1995.

[33] V. Pai, P. Ranganathan, and S.V. Adve, "RSIM Reference Manual Version 1.0," Dept. of Electrical and Computer Eng., Rice Univ., July 1997.

Avinash Karanth Kodi received the MS and PhD degrees in electrical and computer engineering from the University of Arizona, Tucson, in 2003 and 2006, respectively. He is currently an assistant professor of electrical engineering and computer science at Ohio University, Athens. His research interests include computer architecture, optical interconnects, chip multiprocessors (CMPs), and network-on-chips (NoCs). He is a member of the IEEE.



Ashwini Sarathy received the BE degree in telecommunications engineering from the Visvesvaraya Technological University, Belgaum, India, in 2004. She is currently working toward the MS degree in electrical and computer engineering at the University of Arizona, Tucson. Her research interests include computer architecture and network-on-chips (NoCs), with particular emphasis on modeling and simulation of power-efficient NoC architectures.

Ahmed Louri received the PhD degree in computer engineering from the University of Southern California (USC), Los Angeles, in 1988. He is currently a full professor of electrical and computer engineering at the University of Arizona, Tucson, and the director of the High Performance Computing Architectures and Technologies (HPCAT) Laboratory. His research interests include computer architecture, network-on-chips (NoCs), parallel processing, power-aware parallel architectures, and optical interconnection networks. He served as the general chair of the 2007 IEEE International Symposium on High-Performance Computer Architecture (HPCA), Phoenix. He has also served as a member of the technical program committees of several conferences, including the ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS) and the OSA/IEEE Conference on Massively Parallel Processors using Optical Interconnects, among others. He is a senior member of the IEEE, a member of the IEEE Computer Society, and a member of the Optical Society of America (OSA).