International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com
Volume 7, Issue 5 (June 2013), PP. 17-22

Randomized Partially-Minimal Routing Architecture for 3-D Mesh Network on Chips

B.Raja Sekhar Reddy[1], G.Ramesh[2]
[1]ECE Department, GPR Engineering College, Kurnool, Andhra Pradesh, India
[2]ECE Department, Assistant Professor, GPR Engineering College, Kurnool, Andhra Pradesh, India

Abstract:- Programmable many-core processors are poised to become a major design option for many embedded applications. In the design of power-efficient embedded many-core processors, the architecture of the on-chip network plays a central role. Many designs have relied on 2D mesh architecture as the underlying communication fabric. With the emergence of 3D technology, new on-chip network architectures are possible. The increasing viability of 3-D silicon integration technology has opened new opportunities for chip architecture innovations. One direction is in the extension of 2-D mesh-based tiled chip-multiprocessor architectures into three dimensions. This paper focuses on efficient routing algorithms for such 3-D mesh networks. Existing routing algorithms suffer from either poor worst-case throughput (DOR, ROMM) or poor latency (VAL). Although the minimal routing algorithm O1TURN proposed in already achieves near-optimal worst-case throughput for 2-D mesh networks, the optimality result does not extend to higher dimensions. For 3-D and higher dimensional meshes, the worst-case throughput of O1TURN degrades tremendously. The main contribution of this paper is a new oblivious routing algorithm for 3-D mesh networks called randomized partially-minimal (RPM) routing. RPM provably achieves optimal worst-case throughput for 3-D meshes when the network radix is even and within a factor of optimal worst-case throughput when is odd. Finally, whereas VAL achieves optimal worst-case throughput at a penalty factor of 2 in average latency over DOR, RPM achieves (near) optimal worst-case throughput with a much smaller factor of 1.33. For practical asymmetric 3-D mesh configurations where the number of device layers are fewer than the number of tiles along the edge of a layer, the average latency of RPM reduces to just a factor of 1.11 to 1.19 of DOR. Additionally, a variant of RPM called randomized minimal first (RMF) routing is proposed, which leverages the inherent load-balancing properties of the network traffic to further reduce packet latency without compromising throughput.

Keywords:- 3-D ICs, on-chip networks, routing algorithms.

I. INTRODUCTION

There has been considerable discussion in recent years on the benefits of 3-D silicon integration in which multiple device layers are stacked on top of each other with direct vertical interconnects tunnelling through them. 3-D integration promises to address many of the key challenges that arise from the semiconductor industry's relentless push into the deep Nano-scale regime. Recent advances in 3-D technology in the area of heat dissipation and micro-cooling mechanisms have alleviated thermal concerns regarding stacked device layers. Among the benefits, 3-D integration promises the ability to provide huge amounts of communication bandwidth between device layers and integrate disparate technologies in the same chip.

The increasing viability of 3-D technology has opened new opportunities for chip architecture innovations. One direction is in the extension of two-dimensional (2-D) tiled chip-multiprocessor architectures into three dimensions. Many proposed 2-D tiled chip-multiprocessor architectures have relied on a 2-D mesh network topology as the underlying communication fabric. Extending mesh-based tiled chip-multiprocessor architectures into three dimensions represents a natural progression for exploiting 3-D integration. The focus of this paper is on providing efficient routing for such 3-D mesh networks.

As in the case of 2-D mesh networks, throughput and latency are important performance metrics in the design of routing algorithms. Ideally, a routing algorithm should maximize both worst-case and average-case throughput and minimize the length of routing paths. Although dimension-ordered routing (DOR) achieves minimal-length routing, it suffers from poor worst-case and average-case throughput because it offers no route diversity. On the other hand, the routing algorithm proposed by Valiant (VAL) achieves optimal worst-case throughput by load balancing globally across the entire network. However, it suffers from poor average-case throughput and long routing paths. ROMM provides another alternative that achieves minimal routing and good

average-case throughput by considering route diversity in the minimal direction, but it suffers from poor worst-case throughput.

For the case of 2-D mesh networks, a novel described a routing algorithm called O1TURN that achieves both minimal-length routing and near-optimal worst-case throughput. O1TURN simply chooses between two possible minimal-turn paths (XY and YX) for routing. Despite the simplicity, it was shown that O1TURN achieves optimal worst-case throughput when the network radix is even and within a factor of optimal worst-case throughput when is odd. However, as observed in, the near-optimal worst-case throughput property of O1TURN does not extend to higher dimensions.1 perhaps surprisingly; the worst-case throughput of O1TURN degrades tremendously for higher dimensional meshes. For example, in the 3-D case for a mesh, the worst-case throughput of O1TURN degrades to just 30% of optimal. The corresponding worst-case throughput values for DOR and ROMM are even less at around 13% and 26% of optimal, respectively.

In this paper, we introduce a new oblivious routing algorithm called *Randomized Partially-Minimal* (RPM) routing that achieves near-optimal worst-case throughput, higher average case throughput than existing routing algorithms, and good average latency. Conceptually, RPM works as follows: In the 3-D case, we use Z to denote the "vertical" dimension and XY to denote the two "horizontal" dimensions. RPM works by first routing a packet in the minimal direction to a random intermediate "layer" or "plane" in the vertical dimension; i.e., it first routes a packet in the minimal direction to a random intermediate Z position. It then routes the packet on the XY layer using either minimal XY or YX routing. Finally, it routes the packet in the minimal direction in the Z vertical dimension to its final destination. The entire Z-XY-Z or Z-YX-Z path makes at most three turns. Effectively, RPM load-balances traffic uniformly across all vertical layers and routes traffic minimally in the two horizontal dimensions.

Although RPM is worst-case throughput optimal, it is not minimal in terms of latency as it routes packets non-minimally in one of the three dimensions. For certain traffic patterns which are inherently load-balanced, the latency of RPM can be further reduced by preferentially choosing intermediate routing layers along the minimal direction, without causing an overall imbalance in the traffic routed to different layers. We propose a variant of RPM called *randomized minimal first* (RMF) routing that uses destination-aware intermediate layer selection to preferentially select intermediate layers in the minimal direction. This helps in reducing latency over RPM.

II. RANDOMIZED PARTIALLY-MINIMAL ROUTING

The basic idea behind RPM is fairly simple. Conceptually, RPM works by load-balancing flits uniformly across all *k* vertical layers along the Z dimension, just like VAL, but only along one dimension. RPM then routes flits on each XY plane using minimal XY or YX routing with equal probability. Finally, RPM routes flits to their final destinations along the Z dimension. Fig. 1 depicts two possible RPM routing paths. In particular, let (x_1, y_1, z_1) be the source, (x_2, y_2, z_2) be the destination, and $z^{\wedge}$ be the randomly chosen intermediate Z position. The two corresponding Z-XY-Z and Z-YX-Z routing paths are $(x_1, y_1, z_1) \rightarrow (x_1, y_1, z^{\wedge}) \rightarrow (x_2, y_1, z^{\wedge}) \rightarrow (x_2, y_2, z^{\wedge}) \rightarrow (x_2, y_2, z_2)$ and $(x_1, y_1, z_1) \rightarrow (x_1, y_1, z^{\wedge}) \rightarrow (x_1, y_2, z^{\wedge}) \rightarrow (x_2, y_2, z^{\wedge}) \rightarrow (x_2, y_2, z_2)$, respectively, with at most three turns. When $x_1 = x_2$ and $y_1 = y_2$, the traffic is just uniformly randomized along the Z dimension. In this case, when $z^{\wedge}$ is greater than both z_1 and z_2, or when $z^{\wedge}$ is less than z_1 and z_2, a loop is formed in the path $z_1 \rightarrow z^{\wedge} \rightarrow z_2$. These loops can be removed online before routing a packet to reduce hop count. When the source and destination are the same, no routing is necessary. It should be noted that although we use load-balancing along the Z dimension for this description, RPM can be equivalently defined by load-balancing uniformly along any one dimension and routing minimally in the remaining two dimensions.

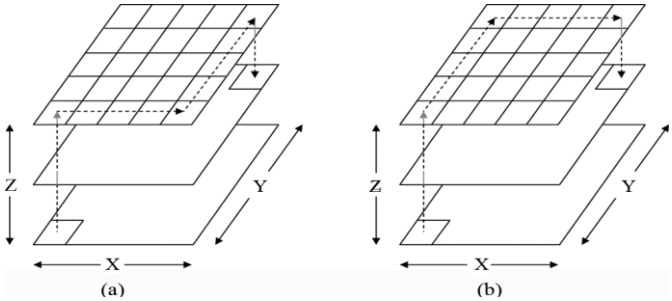

Fig.1: Examples of RPM routing (a) Z-XY-Z path (b) Z-YX-Z path.

III. ROUTER IMPLEMENTATION

In this section, we discuss how RPM can be efficiently integrated into a typical on-chip router. We first explain how RPM can be made deadlock-free using virtual channels. We then present details about the modifications needed in existing on-chip routers to implement RPM.

A. Virtual Channels and Deadlocks

In general, RPM can be defined by load-balancing uniformly along any one dimension and routing minimally in the two remaining dimensions (Z-XY/YX-Z, X-YZ/ZY-X or Y-XZ/ZX-Y). For practical asymmetric mesh topologies where the number of device layers (Z dimension) is expected to be fewer than the number of nodes along the edge of a layer (X and Y dimensions), two-phase routing along the short Z dimension and minimal routing along the longer X and Y dimensions results in highest average-case throughput. On the other hand, for symmetric 3-D mesh topologies, a randomized version of RPM where Z-XY/YX-Z, X-YZ/ZY-X, and Y-XZ/ZX-Y routings are used with equal probability yields the highest average-case throughput as it balances channel load along all three dimensions. Randomization does not change the worst-case throughput of RPM because each one of the three routing algorithms combined has the same near-optimal worst-case throughput.

Virtual channels (VCs) are needed in on-chip routers to avoid cyclic resource dependencies, like buffer dependencies, which can result in deadlocks. If RPM is implemented by load-balancing only along one dimension (let's say the Z dimension), two virtual channels per physical channel are sufficient to achieve deadlock-free routing. One approach is to divide the input buffers on links along the Z dimension into two VCs—VC-0 reserved for phase-1 of Z routing and VC-1 reserved for phase-2 of Z routing. The buffers on links along the X and Y dimensions are also divided into two VCs—VC-0 reserved for packets using X-Y routing and VC-1 reserved for packets using Y-X routing. This VC allocation scheme ensures deadlock-free operation as the corresponding channel dependency graph is acyclic.

The randomized version of RPM, which involves load-balancing packets along each dimension with equal probability and routing minimally along the two remaining dimensions, can be made deadlock-free using three VCs. One VC allocation approach for this case is to let packets start in VC-0 and increment the VC number after every YX, ZY or ZX turns. Since the routing paths have at most two of these three possible turns, three VCs are sufficient. This VC allocation scheme results in an acyclic channel dependency graph for randomized RPM.

B. RPM Router

1) Baseline 3-D Router: Fig. 2 shows the architecture of a typical 7-port router for 3-D mesh networks. This architecture is a direct extension of 5-port routers used in 2-D mesh networks, with the addition of two extra ports for vertical communication. At each input port, buffers are organized as separate FIFO queues, one for each VC. Flits entering the router are placed in one of these queues depending on their VC ID. The router is generally pipelined into five stages comprising route computation, VC allocation, switch allocation, switch traversal and link traversal. The route computation stage determines the output port of a packet based on its destination. This is followed by VC allocation where packets acquire a virtual channel at the input of the downstream router. A packet that has acquired a VC arbitrates for the switch output port in the switch arbitration stage. Flits that succeed in switch arbitration traverse the crossbar before finally traversing the output link to reach the downstream router. Head flits proceed through all pipeline stages while the body and tail flits skip the route computation and VC allocation stages and inherit the output port and VC allocated to the head flit. The tail flit releases the reserved VC after departing the upstream router.

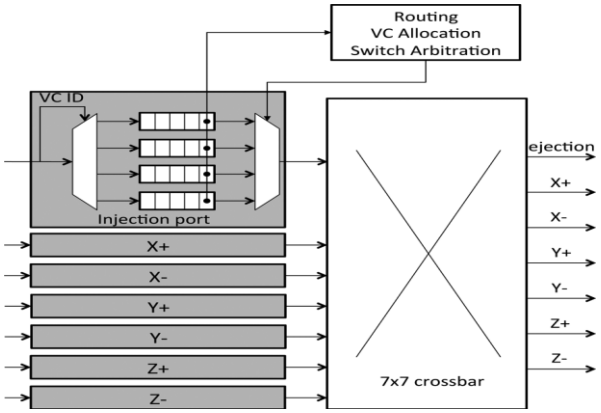

Fig.2: Baseline router architecture.

In order to implement a new routing algorithm like RPM in place of existing routing algorithms like DOR, only the route computation and VC allocation stages of the router pipeline need to be modified.

2) Choosing an Intermediate Layer: RPM involves load-balancing packets to a randomly chosen intermediate layer, some extra logic needs to be added to the baseline architecture to choose an intermediate layer. In most

NoCs there is a strict requirement that all flits of a packet have to arrive at the destination in-order. Hence, load-balancing across layers has to be carried out at the granularity of packets and not flits.

The logic for picking a random intermediate layer can be implemented using a simple linear feedback shift register (LFSR), which is often used to generate a pseudo-random sequence. Let us assume that packets are load-balanced uniformly along the Z dimension. If k_z is the number of layers in the topology, an LFSR with $\log_2 k_z$ bits is sufficient to generate a pseudo random sequence of k_z layers. In order to increase the randomness of the LFSR sequence for small values of, a pseudo-random sequence with greater than $\log_2 k_z$ bits can be generated and the intermediate layer can be chosen by performing a modulo- k_z operation on the generated sequence. For example, in a 8x8x4 mesh topology with 256 nodes, layer load balancing can be carried out using the last two bits of a 8-bit a pseudo-random sequence. To ensure that the random number generators at the different nodes work independently the LFSR at a node can be initialized to the unique 8-bit node address. In general, for a network with N nodes, a $\log_2 N$ bit LFSR can be used to pick an intermediate layer. Each LFSR can then be initialized to a unique initial state (which is the unique $\log_2 N$ bit node address) resulting in different pseudo-random sequences at different nodes. In the special case when the X and Y coordinates of the destination are same as the corresponding coordinates of the source, the decision of the LFSR is overridden and the intermediate layer is forced to be the Z coordinate of the destination. If the pseudo-random sequence generation process and the packet injection process (determining packet size) are independent, over a period of time the number of flits sent to different layers is expected to be equal.

If more accurate load-balancing is desired, a more sophisticated credit-based load balancing scheme can be employed for multi-flit packets. In this technique, every node in the network maintains k_z credit counters, one for each layer, all initialized to 0. The first layer with non-negative credits is always chosen to route a packet. When layer l^* is selected to route an M flit packet, the credit counter corresponding to l^* is decremented by $M - M/k_z$ since layer l^* receives an excess of $M - M/k_z$ flits compared to ideal flit-level load-balancing across k_z At the same time, counters corresponding to all other layers are incremented by M/k_z to account for the flit deficit with respect to ideal flit-level load-balancing. This layer selection approach requires k_z counters at each router to keep track of the credits associated with each layer. The counter size depends on the maximum packet length and the number of layers in the topology. If the maximum packet length is L flits, the credit values can range from $-L$ to $(k_z - 1)\ L$. As explained earlier, when the X and Y coordinates of a packet's destination are same as those of the source, the Z coordinate of the destination is forced to be the intermediate layer in order to remove loops. The counters remain unchanged in this special case as no extra load is added to links along the X and Y dimensions of the destination layer.

After an intermediate layer is chosen at the time of packet injection, the intermediate layer number must be included in the packet header to enable the route computation stage to route packets to the appropriate layer. The intermediate layer selection can be carried out one cycle in advance to avoid increasing the critical path delay of the router.

3) Choosing XY/YX Routing: RPM uses minimal XY or YX routing with equal probability on each horizontal layer. The logic to choose XY or YX routing paths can be the same as the approach described in, which uses a single signed counter to keep track of the excess/deficit of flits using XY or YX routing. The decision to use XY or YX routing can be taken before packet injection, in parallel with intermediate layer selection, and the routing decision can be stored as a part of the packet header. This approach avoids adding extra delay at intermediate routers.

4) Routing and VC Allocation: In a baseline router using dimension-ordered routing, the routing can be decomposed into the X, Y and Z dimensions. The route computation stage first routes a packet along the X dimension, followed by the Y dimension and finally, the Z dimension. For packets at the *injection*, X+ and X- inputs of a router (see Fig. 2), the route computation logic needs to determine the X, Y and Z offsets to a packet's final destination and choose the first productive dimension.4 For packets at the Y+ and Y- inputs, the routing decision is based on the Y and Z offsets to the packet's final destination and for packets at the Z+ and Z- inputs, the decision is based only on the Z offset.

Next, we describe one possible high-level implementation of the route computation stage for symmetric 3-D mesh networks where packets are load-balanced only along the *Z* dimension. Route computation is closely tied to the VC allocation scheme used in the router. Packets are routed using either Z-XY-Z or Z-YX-Z routing paths and two VCs are needed to make RPM deadlock-free in the asymmetric mesh case, as described in Section III-A. We assume that the input buffers at all router ports are divided into two sets of VCs, VC set 0 and VC set 1. Each VC set can in turn have one or more VCs. The two VC sets at different physical channels are associated with different RPM routing segments, as summarized in Table I. A packet is injected into either VC set 0 or VC set 1 at the *injection* port. The routing decisions for packets at different input ports and input VCs are taken as follows:

- For packets at any VC set of the *injection* port and VC set 0 of the Z+ and Z- ports, the offset along the Z dimension to the intermediate layer is determined along with the X and Y offsets to the final destination. If the packet has reached the destination, it is simply ejected from the network. If the Z offset to the intermediate layer is non-zero, packets are routed along the Z dimension on VC set 0. On the other hand, if the Z offset is zero and a packet is chosen to use XY routing, the packet is forwarded to the output port along the X dimension on VC set 0, if the X offset is non-zero. If the X offset is also zero, the packet is forwarded to the output port along the Y dimension on VC set 0. Alternatively, if the packet is chosen to use YX routing, it is forwarded to the output port along the Y dimension on VC set 1, if the Y offset is non-zero. If the Y offset is also zero, the packet is forwarded to the output port along the X dimension on VC set 1.

TABLE 1
RPM VC ALLOCATION

Physical Channel	Virtual Channel	Routing Segment
Injection	Set 0, Set 1	Packet Injection
Z+, Z-	Set 0	Z Phase-1
Z+, Z-	Set 1	Z Phase-2
X+, X-, Y+, Y-	Set 0	XY Routing
X+, X-, Y+, Y-	Set 1	YX Routing

- For packets at VC set 0 of X+ and X- ports, the X, Y and Z offsets to the final destination are computed. If either X or Y dimensions are productive, packets are forwarded on VC set 0 using XY routing. Once packets reach the X and Y coordinates of the destination; they are either ejected, if the Z offset is 0, or forwarded along the appropriate Z output port on VC set 1. Packets at VC set 0 of Y+ and Y- ports are routed similarly, the only difference being that the X offset can be ignored for these packets as they are guaranteed to have reached the X coordinate of the destination.
- For packets at VC set 1 of Y+ and Y- ports, the Y, X and Z offsets to the final destination are computed. If either Y or X dimensions are productive, packets are forwarded on VC set 1 using YX routing. Once packets reach the Y and X coordinates of the destination, they are either ejected, if the Z offset is 0, or forwarded along the appropriate Z output port on VC set 1. Packets at VC set 1 of X+ and X- ports are routed similarly, the only difference being that the Y offset can be ignored for these packets as they are guaranteed to have reached the destination Y coordinate.

For symmetric mesh topologies, a randomized version of RPM that needs three sets of VCs is used, as discussed in Section III-A. In addition to choosing an intermediate layer and the order of dimension traversal within a layer, the randomized variant also needs to select one of X, Y or Z dimensions for load-balancing packets. This decision has to be taken during packet injection and stored in the packet header. The route computation at intermediate routers can then be divided into three parallel planes to handle, X-YZ/ZY-X, Y-XZ/ZX-Y, and Z-XY/YX-Z routing, based on the routing plane chosen for a packet at the time of injection. The VC allocation stage looks at the input port, the input VC set, and the output port of a packet to determine the output VC set. The output VC set is equal to input VC set + 1 if the packet is making a YX, ZY or ZX turn. Otherwise, the output VC set is equal to the input VC set.

IV. RMF ROUTING

RPM achieves near-optimal worst-case throughput in 3-D mesh networks by load-balancing packets uniformly across all vertical layers and routing minimally on the horizontal layers. However, RPM is not minimal in terms of latency since it needs to route packets to a randomly chosen intermediate layer. As described in Section III-B2, the intermediate layer selection process in RPM is oblivious to the packet's destination, which may result in routing in non-minimal directions. Non-minimal routing can be avoided to some extent by using a packet's destination in the layer-selection process and preferentially choosing an intermediate layer in the minimal direction. In his section, we present a destination-aware layer selection technique that can reduce the latency of RPM when the traffic is inherently load-balanced, such as uniform random traffic. We refer to this variant of RPM as RMF routing.

V. CONCLUSION

In this paper, we proposed a new oblivious routing algorithm for 3-D mesh networks called RPM routing. Although minimal routing with near-optimal worst-case throughput has already been achieved for the 2-D mesh case using an algorithm called O1TURN, the optimality of O1TURN does not extend to 3-D or higher dimensions. RPM provably achieves optimal worst-case throughput for 3-D meshes when the network radix k is even and within a factor of $1/k^2$ of optimal worst-case throughput when k is odd. Moreover, RPM significantly

outperforms DOR, ROMM, O1TURN and VAL in average-case throughput by 90%–109%, 45%–54%, 28%–35%, and 24%–52%, respectively, on the different symmetric and asymmetric mesh topologies evaluated. Finally, whereas VAL achieves optimal worst-case throughput at a penalty factor of 2 in average latency over DOR, RPM achieves (near) optimal worst-case throughput with a much smaller penalty of 1.33. In practice, the average latency of RPM is expected to be closer to minimal routing because 3-D mesh networks are not expected to be symmetric in 3-D chip designs. For practical asymmetric 3-D mesh configurations where the number of device layers is far fewer than the number of nodes along the edge of a layer, the average latency of RPM reduces to just a factor of 1.11 of DOR. Finally, we also proposed a variant of RPM called RMF routing which uses the knowledge of a packet's destination while load balancing traffic to intermediate layers. RMF leverages the inherent load-balancing properties of the network traffic to reduce packet latency.

REFERENCES

[1] Agarwal, L. Bao, J. Brown, B. Edwards, M. Mattina, C.-C. Miao, C. Ramey, and D.Wentzlaff, "Tile processor: Embedded multicore for networking and multimedia," presented at the Hot Chips'19, Stanford, CA, 2007.

[2] Black, D. Nelson, C. Webb, and N. Samra, "3-D processing technology and its impact on IA32 microprocessors," in *Proc. Int. Conf. Comput. Des.* 2004, pp. 316–318.

[3] W. R. Davis *et al.*, "Demystifying 3-D ICs: the pros and cons of going vertical," *IEEE Des. Test Comput.*, vol. 22, no. 6, pp. 498–510, Nov./ Dec. 2005.

[4] M. Kawano *et al.*, "A 3-D packaging technology for 4 gbit stacked DRAM with 3 Gbps data transfer," in *IEEE Int. Electron Devices Meeting*, Dec. 2006, pp. 1–4.

[5] P. Gratz, K. Changkyu, R. McDonald, S. W. Keckler, and D. Burger, "Implementation and evaluation of on-chip network architectures," presented at the Int. Conf. Comput. Design, San Jose, CA, Oct. 2006.

[6] T. Kgil, A. Saidi, N. Binkert, R. Dreslinski, S. Reinhardt, K. Flautner, and T.Mudge, "Picoserver: Using 3-D stacking technology to enable a compact energy efficient chip multiprocessor," in *Proc. 12th Int. Conf. Archit. Support Program. Lang. Oper. Syst. (ASPLOS-XII)*, 2006, pp. 117–128.

[7] K. W. Lee, T. Nakamura, T. Ono, Y. Yamada, T. Mizukusa, H. Hashimoto, K. T. Park, H. Kurino, and M. Koyanagi, "Three-dimensional shared memory fabricated using wafer stacking technology," in *Int. Electron Devices Meeting Tech. Dig.*, 2000, pp. 165–168.

[8] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir, "Design and management of 3-D chip multiprocessors using network-in-memory," in *Proc. Int. Symp. Comput. Archit.*, 2006, pp. 130–141.

[9] T. Nesson and S. L. Johnsson, "ROMM routing on mesh and torus networks," in *ACM Symp. Parallel Algorithms Archit.*, 1995, pp. 275–287.

[10] D. Seo,A.Ali,W.-T. Lim, N. Rafique, and M. Thottethodi, "Near-optimal worst-case throughput routing for two-dimensional mesh networks," presented at the Int. Symp. Comput. Arch., Madison, WI, Jun. 2005.

[11] L. Xue *et al.*, "Three-dimensional integration: technology, use, and issues for mixed-signal applications," *IEEE Trans. Electron Devices*, vol. 50, no. 3, pp. 601–609, Mar. 2003.