Theme Article

Synchronized Progress in Interconnection Networks (SPIN): A New Theory for Deadlock Freedom

Aniruddh Ramrakhyani
Georgia Institute of Technology

Tushar Krishna
Georgia Institute of Technology

Paul V. Gratz
Texas A&M University

***Abstract*—Routing deadlocks, i.e., a cyclic dependence between buffered packets, are a fundamental network design challenge. Existing solutions require resource over provisioning. We propose a new theory for deadlock freedom, Synchronized Progress in Interconnection Networks (SPIN) that solves the problem through coordinated movement of deadlocked packets.**

One of the most fundamental challenges in the design of an interconnection network lies in ensuring routing deadlock freedom. A routing deadlock is defined as a cyclic buffer dependency chain where each packet in the chain must acquire a buffer resource held by some other packet in the chain such that no forward progress is possible. Routing deadlocks are a design flaw that can cause system breakdown due to indefinite stalls thereby killing chips. Thus, they must be appropriately dealt with to ensure functional correctness and system reliability.

The problem of routing deadlocks has received significant attention from the research community, and plethora of works have been proposed to either avoid or recover from them. All prior solutions can be classified into one of the four theoretical frameworks:

- Dally's Theory[1]: defines a strict order in which routing resources (links and/or buffers) can be acquired by network packets which ensures a cyclic dependence (and hence deadlock) is never created in the first place.

Digital Object Identifier 10.1109/MM.2019.2910249
Date of publication 11 April 2019; date of current version 8 May 2019.

0272-1732 © 2019 IEEE
Published by the IEEE Computer Society

- Duato's Theory[2]: adds an additional set of buffers to create a routing restricted escape-path that can be used to either avoid or recover out of deadlocks.
- Flow Control[3]: restricts packet injection when the number of empty buffers in the network fall below a threshold to ensure there is at least one free buffer in any dependence chain, thereby guaranteeing forward progress.
- Deflection Routing[4]: forces packet movement every cycle by requiring routers to assign every input flit to some output port (buffer) even if they get misrouted in the process.

A striking commonality among all these theories is that they all view interconnection networks as a classical *dining philosopher problem*, *i.e.*, two resources (the upstream and downstream buffers) need to be held by a given philosopher (packet) in order for progress to be made. Thus, a flit sitting in a buffer is unable to move forward to the downstream router because there is no free buffer available, and a deadlock occurs when this dependence loops back to the buffer of the original flit. Unfortunately, this view fundamentally leads to 1) loss in available bandwidth due to resource (buffer/link) usage restrictions (Dally's theory, Flow Control Deflection Routing) and 2) area and energy overheads of additional resources (buffers) required to avoid or recover out of deadlocks (Duato's Theory).

In this paper, we challenge the theoretical notion of viewing deadlocks in interconnection networks as a dining philosophers problem, where each packet must acquire the downstream buffer before it can relinquish the upstream buffer it resides in, which every solution to date is based on. We instead argue that a deadlock can be viewed as a lack of coordination among distributed entities because, unlike the dining philosopher's problem, a packet needs to hold only one buffer resource at a given moment. We propose a new theoretical framework called Synchronized Progress in Interconnection Network (SPIN) based on the following working principle: If every router in the deadlocked cycle sends out its blocked flit at the same time, without first waiting for the downstream router to become free, there will be safe forward progress. We call this simultaneous synchronized movement of all deadlocked packets in the loop a *spin*. Within a provable upper bound on the number of spins, one of the packets would exit the deadlocked loop thereby breaking the dependence chain. Figure 1 shows the scheme in action. Here, packets A-H are stuck in a cyclic dependence (and hence a deadlock) [Figure 1(a)] as indicated by the arrows which represent the direction that each packet must travel for its next hop. After two *spins* [Figure 1(b) and 1(c)], packets G & C exit the deadlock loop thereby breaking the dependence chain and, thus, resolving the deadlock.

We propose a new theoretical framework called Synchronized Progress in Interconnection Network (SPIN) based on the following working principle: If every router in the deadlocked cycle sends out its blocked flit at the same time, without first waiting for the downstream router to become free, there will be safe forward progress.

SPIN eliminates the need for virtual channels (VCs) to achieve deadlock freedom and is the first scheme to provide fully adaptive routing for any topology, at the least buffer cost theoretically possible for a buffered routing algorithm: one buffer per message class. To take the advantage of this powerful capability of SPIN, we propose Fully Adaptive One-VC Routing with SPIN (FAvORS), a novel topology agnostic routing algorithm that provides deadlock free fully adaptive routing with just one-VC which is impossible across the known solutions today. We also present a low-cost, topology-agnostic, distributed implementation of SPIN allowing it to scale well to large networks.

QUALITATIVE COMPARISON OF SPIN

In Table 1, we qualitatively compare SPIN with the other deadlock freedom theories. The SPIN scheme is completely topology agnostic and requires no knowledge of the channel dependency graph (CDG) unlike Dally's and Duato's theories. In addition, as mentioned before, SPIN enables fully adaptive routing at the least buffer cost theoretically possible for a buffered routing algorithm. Moreover, unlike flow-control and

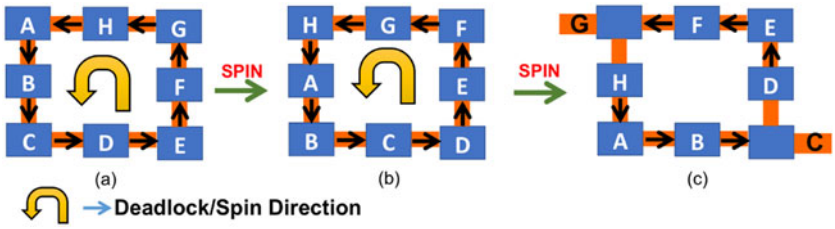


Figure 1. Deadlock freedom with SPIN.

deflection-routing-based schemes, SPIN does not place any packet routing or injection restrictions and thus is inherently livelock-free.

SPIN IMPLEMENTATION

Any implementation of SPIN needs three features: 1) deadlock detection; 2) coordinating the time for *spin*; and 3) performing the *spin*. We choose a deadlock recovery based approach motivated by the premise that deadlocks are rare events (see Section II-F in the original paper[1]).

We illustrate the details through a condensed example. Figure 2(a) shows a deadlocked network with following dependence: (A) ⇒ (B) ⇒ (C) ⇒ (D) ⇒ (A)

Deadlock Detection [Figure 2(a)]: Deadlocks are detected using counters that are placed at every node in the network at design time. Each counter has a state associated with it. In Figure 2(a), the counter at node 3 expires in state *Deadlock Detection (DD)* as packet C does not leave within threshold time (step 1). Next, node 3 sends out a special msg. (SM) called *probe* msg. to confirm the presence of deadlock and rule out false positives due to congestion (step 2). The *probe* msg. tracks the buffer dependence and loops back to node 3 due to the presence of a cyclic dependence (requirement for a deadlock). The deadlock path brought back by the *probe* is latched at the source node, node 3 (step 3). The presence of deadlock is now confirmed.

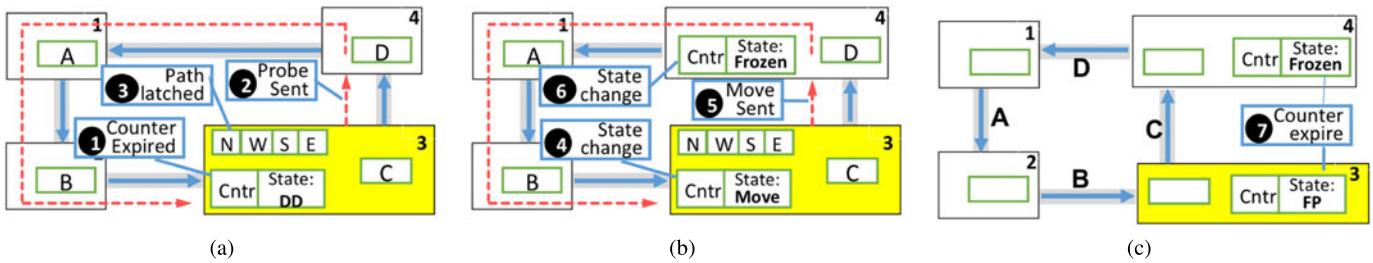
Coordinating the spin time [Figure 2(b)]: After receiving the probe back, the counter FSM at node 3 transitions to state *Move* (step 4). To convey the *spin* time to the deadlocked routers, node 3 sends out another SM called the *move* msg. along the path brought back by the *probe* (step 5). Upon receiving the *move* msg., the counter FSM at node 4 transitions to state *Frozen* and the counter is set to count to the *spin* cycle (step 6). The *move* msg. is processed similarly at other nodes and loops back to node 3.

Performing the spin [Figure 2(c)]: The counter at all deadlocked nodes expire together in

Table 1. Comparison of deadlock freedom theories.

Theory	Packet Injection/ Scheduling Restrictions	Acyclic CDG Required	Topology Dependent	VC cost for:				Livelock Freedom cost
				Minimal Deterministic Routing		Fully Adaptive Routing		
				Mesh (NxM)	Dragon-fly	Mesh (NxM)	Dragon-fly	
Dally's Theory[1]	No	Yes	Yes	1	2^5	6	3^5	None
Duato's Theory[2]	No	No*	Yes**	1	2	2	3	None
Flow Control[3]	Yes	No	Yes	2^6	2	2^6	2	None
Deflection Routing[4]	Yes†	No	No	Not possible‡	Not possible‡	0°	0°	High
SPIN	No	No	No	1	1	1	1	None

*Only an acyclic connected subgraph.
**Need to know topology to design acyclic CDG within the escape virtual channel.
†Cannot inject if the number of packets currently at router is equal to the number of its output ports.
‡Minimal routing cannot be guaranteed by design.
°Cost is 0 assuming bufferless design. Routing is adaptive in the sense that it prevents network hotspots from forming due to deflections. However, deflections may be out of unfavorable ports so routing is not "fully adaptive."



(a) (b) (c)

Figure 2. Walk-through example and router microarchitecture of SPIN. (a) I: Deadlock Detection (*Probe* Traversal). (b) II: Communicate SPIN cycle (*Move* Traversal). (c) III: The SPIN.

the *spin* cycle (step 7). Upon counter expiry, routers push out the deadlocked packets on the respective requested output ports (step 8) leading to simultaneous one-hop forward movement of all deadlocked packets. The *spin* is now complete. However, one *spin* may not be enough to resolve the deadlock. If so, the steps discussed previously can be repeated. However, the implementation introduces optimizations to speed up the case of deadlocks requiring multiple spins to resolve (Section IV-B-4 in the original paper[1]).

The implementation has a low overhead. The SMs. use the same links as regular flits and are not buffered anywhere. Thus, the design adds no extra links or buffers for their handling. In addition, as the recovery is carried out in a distributed manner, thus there is no central coordinator limiting the timing.

Why is the counter state important? The counter state indicates what phase the router is in the recovery process thereby allowing it to gracefully deal with race conditions such as when multiple routers detect the deadlock and send out probe msgs. Section IV-C in the original paper lists some such scenarios and how the design handles them.

FAVORS ROUTING ALGORITHM

SPIN is the first theory to enable true one-VC fully adaptive deadlock-free routing for any topology. We exploit this capability by designing a novel routing scheme: **F**ully **A**daptiv**e** **O**ne-VC **R**outing with **S**pin (FAvORS) that leverages SPIN for deadlock freedom. To select among the routing options at each hop, the algorithm uses "no. of cycles the VC at output port has been active for (since it last became free)" (t_{active}) as a proxy for contention at the next hop router. This information can be obtained cheaply from the VC credit. The algorithm has two variants.

Minimal Adaptive Routing: It routes packets using only minimal paths. At each hop, the algorithm randomly selects any minimal path with a free-VC at next hop. If none of the minimal paths have a free-VC, the algorithm selects the output port with least t_{active}.

Nonminimal Adaptive Routing: The source router has a choice between routing minimally or nonminimally through an intermediate node. The algorithm tries to avoid the latency overhead of nonminimal routing by trying to route minimally first. If any of the minimal paths have a free VC at the next hop, the packet gets routed minimally all the way to destination using the minimal adaptive routing algorithm described previously. Else to choose between minimal and nonminimal routing option, the source node first picks a random intermediate node and then selects the option with lower (Hop-count + (t_{active})) sum, i.e., nonminimal routing option would be selected if

$$H_{\text{min}} + t_{\text{active-min}} > H_{\text{non-min}} + t_{\text{active-non-min}}. \quad (1)$$

This decision is taken only once at the source node which limits the number of misroutes to one thereby making the algorithm livelock free. In the nonminimal routing case, the packet gets routed to the intermediate node first and then to the destination node using minimal adaptive routing (described before) in both the phases.

Algorithm Intuition: The scheme favors minimal routing due to the latency overhead of nonminimal routing. Nonminimal paths should only be considered in a heavily loaded network. If there exists a free-VC at the outport of the minimal path, it indicates that the network is lightly loaded and hence the algorithm routes minimally.

Table 2. Network configurations.

Topology	Design	Adaptive	Minimal	Theory	Type
1024-node Dragonfly	UGAL	Full	No	Dally	Avoidance
	Minimal	No	Yes	SPIN	Recovery
	FAvORS_NMin	Full	No	SPIN	Recovery
8x8 2-D Mesh	Westfirst	Part	Yes	Dally	Avoidance
	EscapeVC	Full	Yes	Duato	Avoidance
	Static-Bubble [7]	Full	Yes	Flow-Ctrl	Recovery
	FAvORS_Min	Full	Yes	SPIN	Recovery

The algorithm uses number of cycles for which the outport VC has been active as a proxy for port contention at the next router. If there is no contention, the VC at next hop router would return to the idle state fast. Further, to spread the traffic uniformly and prevent routing hotspots, the intermediate node is chosen randomly.

EVALUATIONS

Simulation Methodology

We quantify the SPIN's performance by comparing with state-of-the-art deadlock free routing schemes on two popular topologies: a 1024 node off-chip dragon fly[5] with a group size of 8, and on-chip 8 × 8 mesh. All simulations are carried out using the cycle-accurate gem5[8] full-system simulator with Garnet2.0[9] network model. The mesh has 1-cycle routers and 1-cycle links; the dragon fly has 1-cycle routers,[5] 1-cycle intragroup, and 3-cycle intergroup links.

Baselines

Table 2 lists the state-of-the-art baseline designs along with the type (Deadlock Avoidance/Recovery) and the underlying theory they are based on. For off-chip 1024-node Dragon-fly topology, we use a deadlock avoidance scheme with UGAL[10] routing that requires a change of VC every time a global link is traversed.[5] We quantify the performance of our FAvORS algorithm on Dragon-fly by comparing it with UGAL and minimal routing.

For 8 × 8 mesh, we choose a mix of avoidance and recovery schemes. WestFirst design uses west-first routing in all VCs while escape-VC design uses west-first routing in the escape VC and fully adaptive routing in rest. Static Bubble[7] uses adaptive routing in all VCs and strategically places additional buffers that are used for recovery. We also compare minimal version of FAvORS routing algorithm with all these baselines.

Performance

1024-Node Off-Chip Dragon-Fly. Figure 3 plots latency versus injection rate using synthetic traffic patterns with a mix of 3-VC and 1-VC configuration from Table 2.

For the 3-VC case, the baseline design (Deadlock avoidance based UGAL routing) requires packets to change the VC on each use of the intergroup link to avoid deadlocks while SPIN places no routing restrictions. Consequently, while both designs perform identically at low loads (0.01 injection rate), SPIN provides 50%, 20%, and 83% more saturation throughput with bit-complement, transpose, and tornado traffic patterns, respectively. The effect of VC-use restrictions on saturation throughput can be visualized better with the neighbor traffic pattern where a router only generates requests for its 1-hop away neighbor. Here despite the presence of 3-VCs, the traffic in the baseline design can only use one-VC (for the minimal path which will be the dominant routing method for this traffic pattern) thereby severely limiting the throughput. Consequently, SPIN provides 25% improvement in saturation throughput.

In the 1-VC version, UGAL cannot be used as it requires at least three VCs. We run the fully adaptive nonminimal version of the FAvORS algorithm, and contrast it against a minimal routing algorithm. Both the designs use SPIN for deadlock

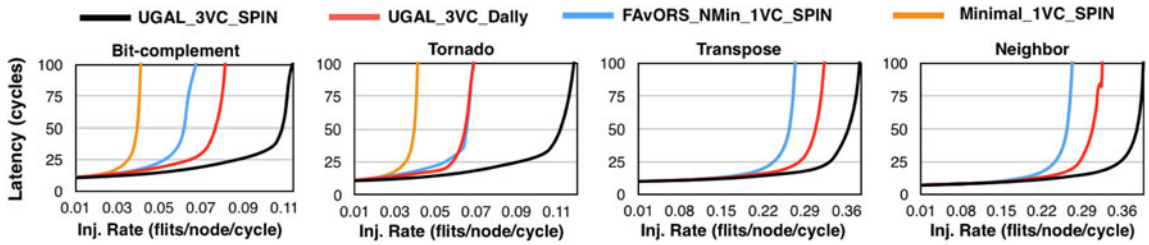


Figure 3. Network performance of 1024-node Dragon Fly (configs described in Table 2).

freedom. FAvORS-NMin outperforms minimal routing by offering 62% and 78% higher throughput with bit-complement and tornado traffic patterns, respectively. The saturation throughput is identical with neighbor traffic pattern as FAvORS reduces to minimal in a highly loaded 1-vc network with all packets traveling only 1 hop.

8 × 8 On-Chip 2-D Mesh. Figure 4 plots latency versus injection rate using synthetic traffic patterns with a mix of 3-VC and 1-VC configurations from Table 2 for 8 × 8 Mesh. Here too, SPIN outperforms other baselines by providing 79%, 16%, and 68% higher saturation throughput than west-first routing and 6%, 18%, and 8% higher throughput than the escape VC design with bit-reverse, uniform random, and transpose traffic patterns, respectively, using 3-VCs. In the 1-VC case, minimal adaptive version of FAvORS provides 80%, 20%, and 18% higher saturation throughput than west-first routing with transpose, bit-reverse, and bit-rotation traffic patterns, respectively. With tornado traffic pattern, minimal adaptive reduces to west-first routing and thus both designs perform identically.

Accuracy and Overheads of SPIN

Link Utilization. Figure 5(a) plots the link utilization of flits and the SMs with uniform random traffic pattern on a 8 × 8 mesh with 3-VCs. At low loads (0.01), the probability of deadlock is low and hence no SMs are sent out (0% combined link utilization). At medium (0.2) and high (0.5) loads, the combined link utilization of SMs never exceeds 5%; thus, the links are either idle or being used by flits at almost all times. It is interesting to note the increase in link idle time at high load with the flit utilization dropping to 33% due to increased frequency of deadlocks at high load. This indicates that the links are mostly idle in case of frequent deadlocks and thus can be used by the SMs.

Area Overhead. Figure 5(b) compares the area overhead of SPIN with alternate designs when implemented in a NoC RTL generator[11] using 15-nm Nangate standard cell library[12]. Compared to the West-first routing design, SPIN has a 4% area overhead which is significantly less than 100% and 10% area overhead for Escape-VC[2] and Static Bubble[7] designs, respectively.

APPLICATIONS and IMPACT

Deadlock Freedom Solutions

As discussed previously, existing state-of-the-art solutions have high implementation overhead in terms of energy and area. This is because all prior frameworks have viewed deadlocks as a lack of resource (buffers) problem

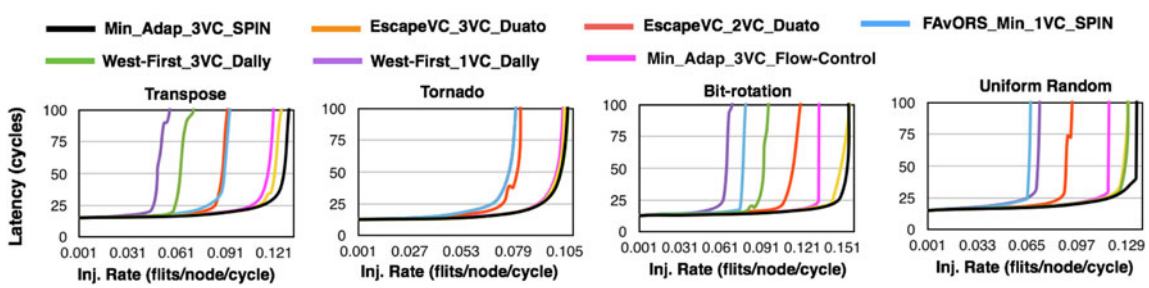


Figure 4. Network performance of 8 × 8 2-D Mesh (configs described in Table 2).

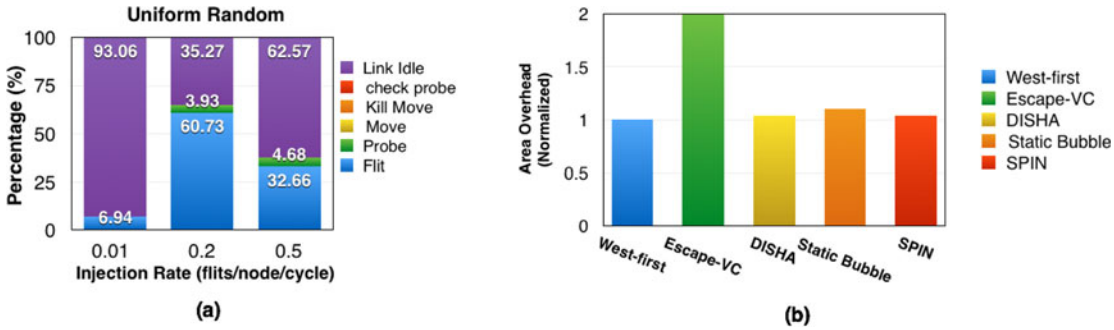


Figure 5. Network performance of 8 × 8 2-D Mesh (configs described in Table II). (a) Link Utilization (b) Area Overhead.

and, thus, have focused on either intelligently managing the available resources or adding more to solve the problem.

Dally's theory places routing/buffer usage restrictions leading to increased packet latency and loss in saturation throughput. Duato's theory adds buffers to create the escape-VC path and consequently suffers from high area and energy overhead of the escape-VC buffers. Implementations of Flow control theory require expensive token-capturing schemes to control packet injection besides the energy overhead of circulating the token. Solutions based on Deflection routing need to solve livelocks due to packet misrouting.

SPIN is free from all these overheads due to the fundamental difference in the way it views deadlocks. Consequently, it is able to provide higher saturation throughput and guarantees routing deadlock freedom at a much lower energy and area cost.

Interconnection Networks

As SPIN is topology agnostic and makes no assumption about the underlying routing algorithm, it can be plugged into any system to make it deadlock free. Apart from the traditional interconnection networks, such as the on-chip Mesh (used in compute chips) or off-chip Dragon-fly (used in super-computers) topologies, SPIN can also be employed to provide deadlock-freedom for datacenter topologies like Jellyfish, a random graph-based topology that allows easy incremental expansion of datacenter networks. As SPIN is low cost and plug-n-play, it will work with any random graph thereby increasing the efficacy of the topology in solving the problem of network expansion post deployment.

In addition, we see SPIN as a promising choice for providing deadlock freedom in the design of custom NoCs using NoC generators like FlexNoc (by Arteris) and for on-chip networks within domain-specific accelerators like Eyeriss. Another interesting application for SPIN is NoC power-gating and on-chip resiliency. Waning silicon reliability and increasing need for power-gating attributed to the demise of Dennard's scaling have led to a surge in instances of irregular on-chip topology that changes dynamically, thereby requiring solution for deadlock free routing over these topologies.

Systems Research

We would also like to emphasize that SPIN as a theoretical framework has much broader applications than just interconnection networks. SPIN's basic premise of being able to release and acquire a resource via clever orchestration can be applied across the system stack where deadlocks are a fundamental challenge. For example, SPIN can be used to schedule orchestrated movement in coherence protocols, enabling protocol design without requiring VCs. Deadlocks occurring at OS level can also leverage SPIN by fundamentally changing how locks are acquired and released. This will have a profound impact in industry and academia.

CONCLUSION

In this paper, we presented a new deadlock freedom framework called SPIN that challenges the theoretical notion of viewing network routing deadlocks as a lack of resource problem, instead

viewing them as a lack of coordination among distributed entities. We provide a scalable implementation of this framework and also propose for the first time a fully adaptive routing algorithm with only one VC using this framework for deadlock freedom. Quantitatively, SPIN provides comparable performance to other state-of-art schemes at 35%–52% lower area and 38%–55% lower power on a 1024-node off chip Dragon-fly and on-chip 8 × 8 mesh. The underlying theory of the framework has broad impact in the areas of interconnection networks and systems research.

Quantitatively, SPIN provides comparable performance to other state-of-art schemes at 35%–52% lower area and 38%–55% lower power on a 1024-node off chip Dragon-fly and on-chip 8 × 8 mesh.

REFERENCES

1. W. J. Dally and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. Comput.*, vol. 39, no. 5, pp. 547–553, May 1987.
2. J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 4, no. 12, pp. 1320–1331, Dec. 1993.
3. C. Carrion *et al.*, "A flow control mechanism to avoid message deadlock in k-ary n-cube networks," in *Proc. 4th Int. Conf. High-Perform. Comput.*, 1997, pp. 322–329.
4. C. Fallin *et al.*, "Chipper: A low-complexity bufferless deflection router," in *Proc. IEEE 17th Int. Symp. High Perform. Comput. Archit.*, 2011, pp. 144–155.
5. J. Kim *et al.*, "Technology-driven, highly-scalable dragonfly topology," in *Proc. 35th Annu. Int. Symp. Comput. Archit.*, 2008, pp. 77–88.
6. C. Xiao, M. Zhang, Y. Dou, and Z. Zhao, "Dimensional bubble flow control and fully adaptive routing in the 2-D mesh network on chip," in *Proc. IEEE/IPIP Int. Conf. Embedded Ubiquitous Comput.*, Shanghai, China, Dec. 17–20, 2008, Volume I, pp. 353–358.
7. A. Ramrakhyani and T. Krishna, "Static bubble: A framework for deadlock-free irregular on-chip topologies," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit.*, 2017, pp. 253–264.
8. N. Binkert *et al.*, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, pp. 1–7, Aug. 2011.
9. N. Agarwal *et al.*, "GARNET: A detailed on-chip network model inside a full-system simulator," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, 2009, pp. 33–42.
10. A. Singh, "Load-balanced routing in interconnection networks," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 2005.
11. H. Kwon and T. Krishna, "Opensmart: Single-cycle multi-hop NoC generator in BSV and chisel," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, 2017, pp. 195–204.
12. M. Martins *et al.*, "Open cell library in 15nm freePDK technology," in *Proc. Symp. Int. Symp. Phys. Des.*, 2015, pp. 171–178.

Aniruddh Ramrakhyani is a performance modelling engineer at Apple, where he works on optimizing Media SoCs for Apple products. His research interests lie in the area of interconnection networks, hardware–software co-design and machine learning accelerators. He has an MS in electrical and computer engineering from Georgia Institute of Technology. He is a member of the ACM and the IEEE. Contact him at aniruddh.ramrakhyani@gmail.com.

Paul V. Gratz is an associate professor with the Department of Electrical and Computer Engineering, Texas A&M University. His research interests include high–performance computer architecture, processor memory systems, and on-chip interconnection networks. He had a PhD in electrical and computer engineering from the University of Texas at Austin. He is a Senior Member of the IEEE and a member of the ACM. Contact him at pgratz@gratz1.com.

Tushar Krishna is an assistant professor in the School of Electrical and Computer Engineering, Georgia Institute of Technology. His research interests include computer architecture, on-chip interconnection networks, and deep learning accelerators. He has a PhD in electrical engineering and computer science from MIT. He received the NSF CRII Award in 2018. He is a member of the IEEE and the ACM. Contact him at tushar@ece.gatech.edu.