

A Voting Approach for Adaptive Network-on-Chip Power-Gating

Jiayi Huang, *Member, IEEE*, Shilpa Bhosekar, Rahul Boyapati, Ningyuan Wang, Byul Hur, *Member, IEEE*, Ki Hwan Yum, *Member, IEEE*, Eun Jung Kim, *Member, IEEE*

Abstract—Scalable Networks-on-Chip (NoCs) have become the standard interconnection mechanisms in large-scale multicore architectures. These NoCs consume a large fraction of the on-chip power budget, where the static portion is becoming dominant as technology scales down to sub-10nm node. Therefore, it is essential to reduce static power so as to achieve power- and energy-efficient computing. Power-Gating as an effective static power saving technique can be used to power off inactive routers for static power saving. However, packet deliveries in irregular power-gated networks suffer from detour or waiting time overhead to either route around or wake up power-gated routers. In this paper, we propose *Fly-Over (FLOV)*, a voting approach for dynamic router power-gating in a light-weight and distributed manner, which includes FLOV router microarchitecture, adaptive power-gating policy, and low-latency dynamic routing algorithms. We evaluate FLOV using synthetic workloads as well as real workloads from PARSEC 2.1 benchmark suite. Our full-system evaluations show that FLOV reduces the power consumption of NoC by 31% and 20%, respectively, on average across several benchmarks, compared to the baseline and the state of the art while maintaining the similar performance.

Index Terms—Networks-on-Chip, power-gating, routing algorithm, voting.

1 INTRODUCTION

CHIP multiprocessors (CMPs) tends to scale to hundreds of cores, which is a promising solution to extract huge performance improvement using parallel programming paradigms. However, it has become hard to win the performance as Moore's law predicted by simply shrinking transistor sizes and using high performance on-chip packaging due to the failure of Dennard Scaling. Given the very large number of transistors, it is also challenging to design processors to meet the power and thermal constraints. Thus, future CMP designs have to work under much stricter power envelopes.

Scalable Networks-on-chip (NoCs) such as 2D meshes, are *de facto* communication fabrics in large CMPs. Studies show that NoCs consume a significant portion, ranging from 10% to 36%, of the total on-chip power [1], [2], [3]. Therefore, it is highly desirable to achieve power-efficient NoC designs for future CMPs. Static power consumption for the chip is also increasing drastically, while the feature size becomes smaller and the operating voltage gets closer to the near-threshold level. Previous studies show that the percentage of static power in the total NoC power consumption increases from 17.9% at 65nm, to 35.4% at 45nm,

to 47.7% at 32nm and to 74% at 22nm [4], [5]. Based on this trend, as we reach towards sub-10nm feature sizes, static power can become the major portion in NoC power consumption. Power-gating is an effective circuit technique to mitigate the worsening impact of on-chip static power consumption by cutting off supply current to idle chip components. Due to low average core utilization in most modern workloads [6], significant research has presented efficient mechanisms for power-gating cores with marginal impact on performance [7], [8], [9]. Some studies have used power-gating for selected router components in a fine-grained fashion using topology reconfiguration [10], [11]. There has been other research on power gating [4], [12], [13], [14], [15], which can reduce NoC static power consumption. Another important issue in power-gating NoC is routing packets in an irregular network. Prior work either tries to route around off routers in a power-gated network [11], [12] or uses the shortest path by early waking up off routers on the path [13], [16]. These solutions may still incur detour or wakeup latency and power overhead.

Previous research proposes router power-gating either by reacting to network traffic [4] or based on the power state of the attached core [12]. NoRD decouples nodes and routers while maintaining network connection by forming a ring across the network [4]. However, the ring may incur considerable packet latency penalty in low traffic load. Significant research in Operating System (OS) level has shown notable static power savings in CMPs by power-gating idle cores and consolidating the thread executions to fewer cores [7], [8], [9], [17]. Router Parking (RP) [12] power-gates routers whose attached cores are power-gated, but requires a centralized fabric manager for network reconfiguration which creates a huge synchronization overhead, and the whole network has to stall until the reconfiguration is completed. Moreover, RP creates a single point of failure

- J. Huang, B. Hur, K.H. Yum and E.J. Kim are with Texas A&M University, College Station, TX, 77845.
E-mail: {jyhuang,yum,ejkim}@cse.tamu.edu, byulmail@tamu.edu.
- Shilpa Bhosekar is with Ambarella, Santa Clara, CA 95051.
E-mail: shilpa.bits@tamu.edu.
- Rahul Boyapati is with Intel Corporation, Santa Clara, CA 95054.
E-mail: rahul.boyapati@intel.com.
- Ningyuan Wang is with Google LLC, Mountain View, CA 94043.
E-mail: nywang@google.com.

Manuscript received 16 May 2020; revised 23 Sept. 2020; accepted 4 Oct. 2020.
Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Jiayi Huang.)
Recommended for acceptance by R. Marculescu.
Digital Object Identifier no. 10.1109/TC.2020.3033163

if the centralized fabric manager goes down. In addition, RP may incur additional detour latency overhead of packet delivery in an irregular power-gated network.

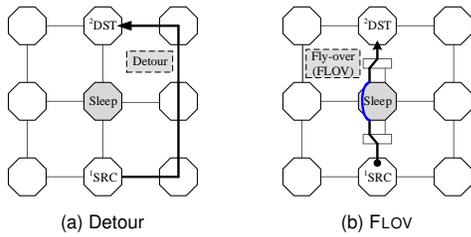


Fig. 1: Routing examples of detour (a) and FLOW (b) with a power-gated router in the middle. Each octagon is a router. ¹SRC and ²DST denote source and destination, respectively

In this paper, we propose *Fly-Over* (FLOW), a dynamic and distributed power-gating mechanism through router voting that eliminates the centralized control for router power gating. FLOW tries to power-gate routers as soon as the attached cores are powered off by the OS in a distributed manner. In FLOW, the packets can be delivered by flying over the power-gated router without the need of the detour as shown in Fig. 1. Since router power-gating may create interconnect partitions without communication paths, FLOW links are provided to enable incoming packets to travel straight through power-gated routers for network connectivity. FLOW includes FLOW router microarchitecture, a voting-based power-gating policy with handshake protocols, and its adaptive routing algorithms. For FLOW router microarchitecture, we augment the baseline router with FLOW links for *flying over* power-gated routers. Based on FLOW router, we present two power-gating modes. The first one works under restricted conditions, called *restricted FLOW* (R-FLOW), where no consecutive routers in a row/column can be power-gated at the same time. The second one is called *generalized FLOW* (G-FLOW), where two or more consecutive routers in a row/column can be power-gated simultaneously. These two power-gating modes show the trade-off of performance and power saving, where R-FLOW has better throughput than G-FLOW while G-FLOW saves more power than R-FLOW. In order to dynamically adapt to the traffic load to guarantee the performance, each router adjusts its power-gating mode among G-FLOW, R-FLOW and NO-FLOW (no power-gating) through router voting. In addition, based upon our prior work [18], we propose a better adaptive routing algorithm that computes the route using the information of *logical* neighbor routers to achieve best-effort minimum path and reduce congestion.

We evaluated FLOW using BookSim, a cycle-accurate interconnect simulator, for the detailed NoC evaluation, and gem5 for full-system evaluation [19], [20]. The effectiveness of FLOW is demonstrated by comparing with RP and NoRD [4], [12]. The full-system evaluations show that FLOW reduces power consumption by 31% and 20%, on average across several benchmarks, compared to Baseline and the state of the art, respectively, and it could keep the performance degradation to the minimum.

2 RELATED WORK

Fine-grained Interconnect Power-Gating. Significant research has applied power-gating techniques in NoCs [7],

[21]. Several fine-grained interconnect component power-gating techniques were proposed [10], [11], [22], [23]. Kim et al. introduced a dynamic link shutdown (DLS) technique together with dynamic voltage scaling to save link energy [24]. Soteriou et al. presented a power-aware network that reduces static power consumption by monitoring the link utilization and power-gating the underutilized links [22]. Matsutani et al. applied the power-gating technique to control the power supply of different components individually in an ultra fine-grained way [10]. Kim et al. proposed a buffer organization to adaptively adjust active buffer size by power-gating [23]. Parikh et al. introduced power-aware routing and topology reconfiguration to minimize detours while selected components in routers are power-gated [11]. These approaches work well to reduce the static power consumption, however, they only power-gate certain components of a router and require additional circuits.

Coarse-grained NoC Router Power-Gating. Coarse-grained router power-gating has been broadly studied as well. In [16], lookahead routing is utilized to wake up sleeping routers two hops in advance to hide the wakeup latency. However, as clock frequency increases, wakeup latency cannot be totally hidden. Chen et al. introduced Power Punch, a non-blocking power-gating scheme that wakes up powered-off routers along the path of a packet in advance, thereby preventing the packet from suffering router wakeup latency [13]. Zhan et al. presented a mechanism that can activate powered down cores for performance gains while considering thermal aware floor planning, and they also explored topological/routing support [25]. Catnap power-gates physical sub-networks based on the priority and predicted traffic load [26]. This work is orthogonal to FLOW, and FLOW can be applied to the powered-on subnetworks to achieve even more power savings. Recently, Samih et al. introduced Router Parking (RP) to power-gate routers when their attached cores are sleeping while some of the routers are kept on to ensure network connectivity [12]. RP dynamically reconfigures the network among aggressive, conservative and no power-gating modes to trade-off power saving and performance. Upon reconfiguration, it estimates the power saving and decides the power-gating mode by collecting stats from all routers and distributing the recomputed routing tables. This scheme requires centralized control and typically takes a long time to reconfigure the network that may suspend new injections into the network during this phase. Chen et al. proposed node-router decoupling (NoRD) approach to leverage the independence of power-gating a core and its attached router, which provides a decoupling route through network interface to bypass the power-gated router [4]. The decoupling bypass links ensure network connectivity by using an escape bypass ring network. However, a bypass ring is not scalable to large network sizes. Another issue with NoRD is that a bypass can be constructed in a $(k \times k)$ mesh, if and only if k is even.

Bypassing Mechanisms. Some studies have suggested bypassing for different purposes in NoCs [27], [28], [29], [30], [31]. Kumar et al. proposed express virtual channels that virtually bypass intermediate routers for packet transmission to achieve high performance [27]. In [28], dual functional physical channel buffers are used to bypass a router and to keep packets in the links along the path. Long-range

link [29], [30] and skip-link [31] were proposed to bypass routers for faster packet delivery. Unlike these studies, FLOV stands as a power saving perspective with performance awareness. EZ-Pass has latches in all the directions similar to FLOV and borrows the NoRD idea to bypass the router by going through the network interface, but it avoids the ring network [14]. EZ-Pass adds an extra routing computation unit in network interface for data bypassing, and unnecessarily going through network interface even when the packet has no need to make a turn. Furthermore, the unified VC state table increases the hardware complexity in order to support concurrent reads/writes for different ports and accessibility from both network interface and router. Muffin also incorporates similar ideas whereas it handles bypassing inside the router with extra control and arbitration [15]. In contrast, FLOV links in a router act as simple connections between the upstream and downstream routers. As a result, FLOV only takes direct bypass without arbitration to reduce the overhead. In concurrent with FLOV [32], another similar bypassing mechanism is TooT, which waits for the power-gated router to power on for turning purpose [33]. Later, Sponge presents a pivot router column, which is similar to the *always on* (AON) router column in FLOV, for making turns [34]. Sponge takes coarser-grained power-gating decision for the whole columns, while FLOV takes router-based policy that enables router power-gating independently. In addition, the voting approach of FLOV power-gating takes both local and global traffic information into account, which can be applied to other power-gating techniques.

Routing Algorithms for Power-Gating NoC. In most of NoC power-gating proposals, regular network routing is used by waking up the power-gated routers [10], [13], [33]. In contrast, Router Parking recomputes routing tables at every reconfiguration and NoRD introduces a bypass ring network to solve the routing problem in irregular networks. Other research has proposed mathematical models or tools for routing in reconfigurable network [35], [36]. However, they require more hardware complexity that increases power consumption, or their algorithm is too complex to perform in hardware to reflect rapid topology changes in the network. These tools are more suitable for application-specific multi-processor systems-on-chip (MP-SoC). They can also be used to help analyze the routing design. Although fault tolerance is not the scope of this paper, related routing algorithms are applicable in most cases [37]. But they are not easily extended to the cases where power-gated nodes may disconnect the network in the assumption of fault tolerance design, even the links are actually maintaining the connectivity in our setting. Therefore, we turn to exploiting the structure of mesh networks and develop algorithms for power-gating NoCs. We leverage the relative position information to achieve the best-effort minimal path in 2D mesh, in addition to adaptivity with a deadlock-free routing subfunction based on turn model [38] and Duato's protocol [39].

Earlier versions of FLOV research were presented in conference proceedings [18], [32]. In this paper, we further mature and improve FLOV. Better and effective algorithms are described in detail, and a router voting approach for adaptive power-gating is proposed. We also evaluated their performance and compared it with NoRD.

3 FLOV ROUTER MICROARCHITECTURE

Fig. 2 shows the FLOV router microarchitecture, which augments a baseline router with multiplexers (muxes) and demultiplexers (demuxes) added to input/output links as well as a FLOV output latch in each direction, as shown as shaded blocks and blue lines. When the power of the FLOV router is on, it works as a baseline 3-stage virtual-channel router, where muxes/demuxes are controlled to select the normal router datapath, and the latches are power-gated. If the router is power-gated, all components of the baseline router (white blocks) are power-gated and the muxes/demuxes are set as 1 to activate the FLOV links by the power-gating (PG) signal. For the routers at the edge of a 2D mesh, if they are power-gated, the FLOV links are activated only in the dimension X or Y where there are neighbors in both directions. A handshake controller (HSC) block is introduced to connect all neighboring routers for handshaking purposes. Power state registers (PSRs) are added to keep track of the power states of the physically adjacent neighbors and the logical neighbors, which are the nearest powered-on routers in each direction. We modified the Credit Control Logic (CCL) in virtual channel (VC) allocator to interact with HSC in order to always hold the buffer availability (credit) information of the logical neighbor routers.

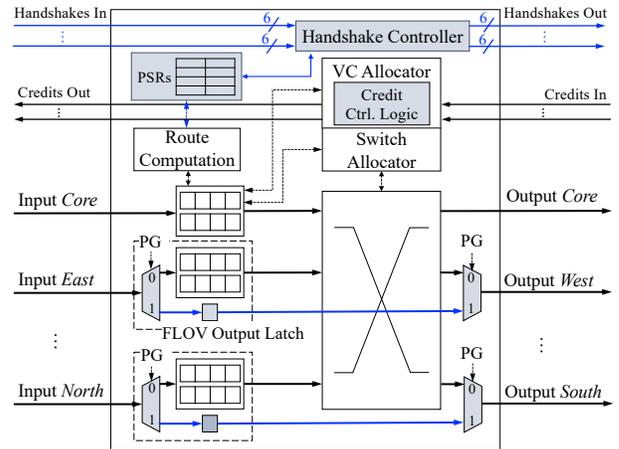


Fig. 2: The augmented FLOV router microarchitecture, with added blocks as shaded, datapath/control lines as blue, PG as power-gating signal.

4 ADAPTIVE FLOV POWER-GATING POLICY

Using the FLOV router microarchitecture in Section 3, we introduce two power-gating modes, restricted FLOV (R-FLOV) and generalized FLOV (G-FLOV), in addition to the baseline no power-gating mode (NO-FLOV). R-FLOV achieves better performance with limited power saving, while G-FLOV trades throughput for better power. Figs. 3a and 3b show the routing examples of R-FLOV and G-FLOV, respectively. In Fig. 3a, two routers at the right and left edges in the second row can exchange packets passing through the smallest number of the routers instead of detouring, although there is a power-gated router on the path. This path is possible owing to the FLOV link. In Fig. 3b, there are consecutive power-gated routers that are right next to each other. This placement is not allowed in R-FLOV but it is allowed in G-FLOV. In order to adapt to traffic loads for performance guarantee, we propose an adaptive power-gating policy to adjust router's power-gating modes through router voting,

where each router periodically decides its power-gating mode depending on the collected votes from routers in the same dimensions. If a router changes from R-FLOV to G-FLOV mode, it tries power-gating and drains packets if its attached core is power-gated. If a router regresses from G-FLOV to R-FLOV mode, it wakes up from *Sleep* state or remains *Active* if any of its neighbors are in *Sleep* state.

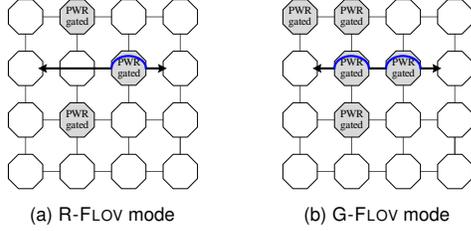


Fig. 3: Examples of R-FLOV (a) and G-FLOV (b) with power-gated routers. Each octagon is a router. 'PWR gated' indicates power-gated.

4.1 Restricted FLOV (R-FLOV)

Fig. 4 depicts the power state transition diagram of a router. If the core is powered-gated, the attached router sends a control signal to its neighbors using out-of-band control lines to indicate that it is in the *Draining* state. During this draining state, its neighbors cannot initiate any new packet transmissions to this router, while it is allowed to finish current packet deliveries.

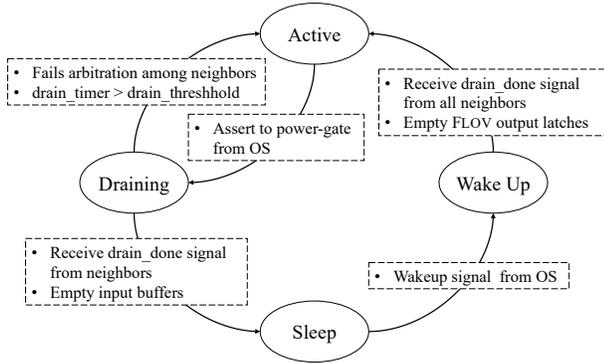


Fig. 4: A state transition diagram for the power status of a router.

In R-FLOV, a router is not allowed to power down if any of its neighboring routers is or to be power-gated. If a router in the *Draining* state receives the same signal from its neighboring router, only one of them with a smaller router ID is allowed to proceed, and the other router reverts back to normal *Active* state.

A router in *Draining* checks for any residing flits in its input buffers, and continues to forward them to downstream routers as normal. Note that remaining flits for the residing packets should also be drained in order to guarantee correct flow control. Once emptying all its input buffers and receiving *drain_done* signals from all its neighbors, the router is power-gated by shutting down the baseline router portion and entering *Sleep* state. At the same time, all the muxes/demuxes are controlled to select the FLOV datapath, and the router sends all its neighbors a signal to initiate new packet transmissions, and to update their immediate neighbor PSRs.

If a router is power-gated, a flit coming into the router is stored in the FLOV output latch without any rout-

ing/arbitration. Then, it is delivered to a designated virtual channel (VC) in the downstream router since the VC has been allocated by the upstream router. From the downstream router, the packet delivery becomes normal. When a router is in the *Sleep* state, the credit counts of its downstream router are copied to the upstream router so that the upstream router can obtain the correct credit information of the downstream router.

A power-gated router in R-FLOV mode wakes up again when its core becomes active or it is voted towards NO-FLOV mode for better performance. When a sleeping FLOV router wakes up due to the aforementioned conditions, it sends signals to its neighbors to stop new packet transmission and enters *Wakeup* state. When it completes current packet transmissions and emptying its output latches, the router powers on the baseline router portion and switches to select the normal router datapath. During the *Wakeup* process, the FLOV router may still relay credit signals from its downstream router to its upstream router. Once it becomes *Active*, it only processes credit information from downstream routers for its own, and its upstream router sets the corresponding credit to fully available.

Fig. 5 shows a set of snapshots of a working example to demonstrate R-FLOV mode in time sequence. For simplicity, draining of the packets and credit control are shown only for one direction, but a router has to perform these actions for all its neighbors before state transitions.

- In Fig. 5a, three routers are *Active*. Router A holds the body (B1) and tail (T1) flits of packet 1 as well as the head flit (H2) of packet 2. Router B holds the head flit (H1) of packet 1 and Router C is empty. The PSR entries of the routers show the power states of the immediate neighbors in the East (Routers A and B) or West (Router C). The current credit status of VC1 of the downstream routers is also shown. The shaded portion indicates the power-gated components that are the output latches.
- In Fig. 5b, both Routers B and C send *Drain* signals to their neighbors to indicate their willingness to go into the *Draining* state. Since Router B has the lower router ID, it wins the arbitration and Router C has to revert to *Active* state. The PSR entries in Routers A and C are updated to *Drain* due to Router B. Router A transfers flit B1 to Router B and B transfers flit H1 to Router C. The corresponding credit counters are also updated.
- In Fig. 5c, Router A sends the *drain_done* signal to Router B as it finishes transmitting packet 1 to B. Similarly, Router C sends the *drain_done* signal to B. But since Router B has not finished draining its buffers yet, it has to wait before going into the *Sleep* state.
- Fig. 5d depicts the scenario after Router B finishes draining packet 1 to Router C and goes into *Sleep* state. The shaded VC buffer indicates that the baseline router has been power-gated and the FLOV links (output latches) have been activated. Router B sends the *Sleep* signal to its neighbors so that they can update their corresponding PSR entries, and the credit counters are initialized as shown in Router A. Note that although Router A has a flit (H2) to send Router B, it still has to wait until B finishes its power state transition.
- Fig. 5e shows the credit control and maintenance between Routers A and C while Router B is power-gated.

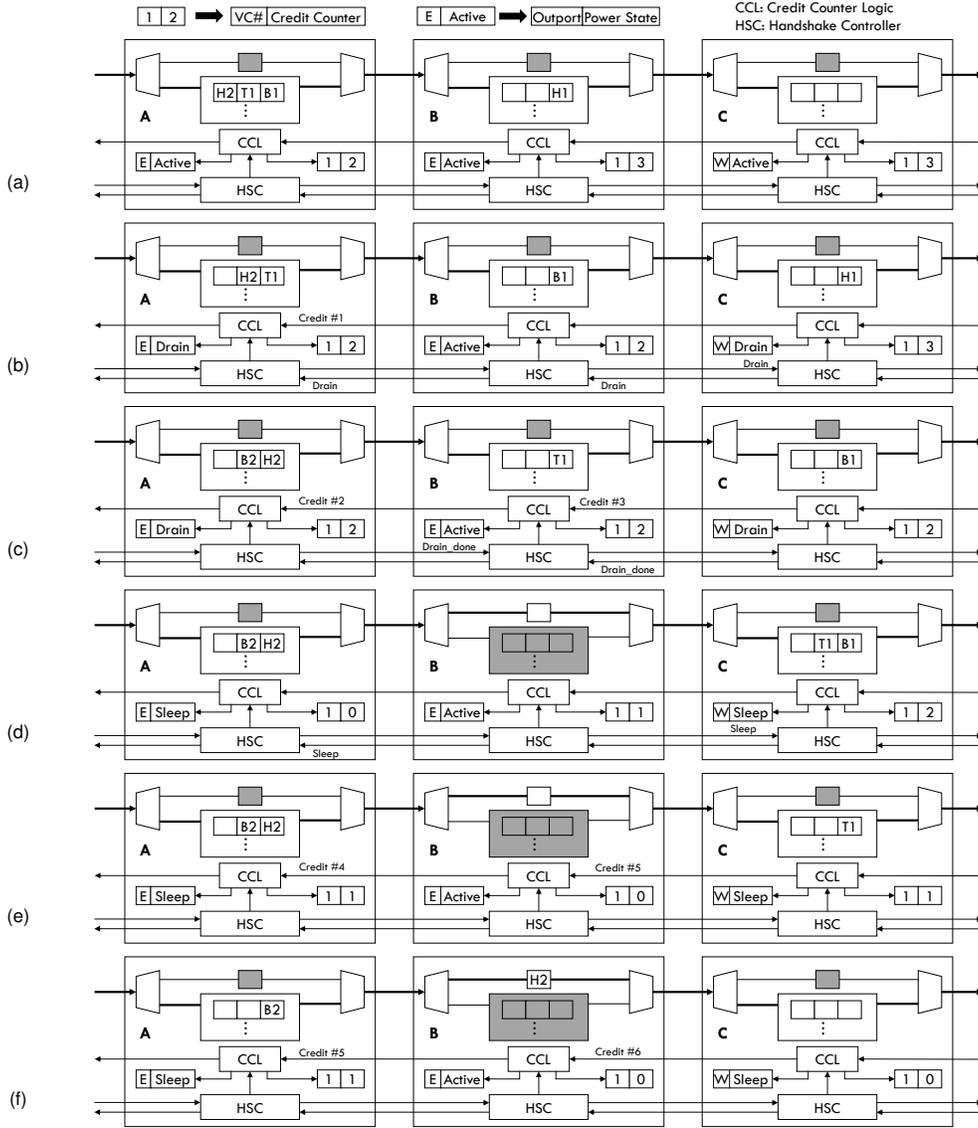


Fig. 5: An example of R-FLOW with snapshots in timeline from (a) to (f).

After Router B goes into the *Sleep* state, Router A initialize its credit counter entry and the credit information is copied from Router B to A (Credit #4). This is because Router C is the logical neighbor of Router A, so A has to keep track of the buffer availability (credits) in C. Credit #5 carries the newly available credit in Router C to Router B.

- (f) In Fig. 5f, Credit #5 is relayed by the power-gated Router B to Router A. Then, it updates its credit counters. This relaying scheme maintains the correct flow control between Router A and Router C.

The wake-up procedure is similar to the draining procedure, the *Wakeup* router sends the *wake-up* signals to its neighbors and starts to drain packets from its output latches. The router also waits for all its neighbors to finish any intermittent transmissions and sends *drain_done* signals. The router then receives the credit information from the downstream router and sends a signal to notify the upstream router to make its corresponding credit counter fully available. Once it happens, the router controls muxes/demuxes to resumes baseline router datapath.

4.2 Generalized FLOW (G-FLOW)

R-FLOW tends to achieve better throughput but limits power saving because none of a sleeping router's neighbors is allowed to sleep regardless of the power states of their attached cores. In this section, we introduce *generalized* FLOW (G-FLOW), where a router can be power-gated even if any of its neighbors is power-gated, thereby two or more consecutive routers in a row/column can be power-gated simultaneously. During handshaking, the power-gated routers in the middle should relay the handshake signals, in addition to update the corresponding logical and physical neighbor routers' power states in the PSRs. The flow control in G-FLOW is similar to R-FLOW, except that credit relaying may across several sleeping routers.

When a router is in R-FLOW mode, its neighbors are all *Active* if it is power-gated. In contrast, a power-gated router in G-FLOW mode may have neighbors in a chain that are also power-gated. Therefore, we introduce a few handshaking protocol modifications and additional functionalities to avoid protocol deadlock and to aid routing decision, described as follows:

Firstly, after a router enters *Sleep* state, it sends the corresponding power state and router ID of its logical neighbors in each direction to its upstream router, in addition to its new power state. Then, the logical neighbor of the power-gated router becomes the logical downstream router for its upstream router. Thus, the logical PSRs of all the routers can be kept up-to-date. Meanwhile, the *Sleep* power state in the handshake signal clears the credit counters for the just power-gated router. It also stops the credit copying procedure in the earlier power-gated router on the way. Then the newly power-gated router starts credit copying, one credit per VC at a time with one cycle per link traversal. Note that power-gating happens when the traffic is low, so the credit propagation latency has low impact on buffer utilization in most cases. The idle handshaking links can be used for credit copying if optimization is needed, which is not evaluated in this work. Moreover, the logical neighbor ID information helps design a better routing algorithm that is presented in Section 5.

Secondly, in wormhole switching, no two logical neighbor routers in the same row/column are allowed to stay in *Draining-Draining*, *Draining-Wakeup*, or *Wakeup-Wakeup* state combinations at the same time in order to avoid protocol deadlock or starvation. If one of the handshaking routers is trying to wake up and the other trying to drain, *Draining* has lower priority due to the fact that *Wakeup* is more crucial for performance. For the simplicity of handshaking, if a power-gated router has a downstream router in the *Draining* state, it cannot wake up until the draining router changes its state. When two handshaking routers are trying to drain or wake up at the same time, only the one with a smaller router ID can proceed. If virtual cut-through switching is applied, the above condition can be relaxed for the *Wakeup-Wakeup* case. Unlike the *Draining-Draining* combination, two waking up routers have no dependence on each other since they always bypass the flits. In addition, the buffer resource in a powered-on router between two *Wakeup* routers is sufficient to store a whole packet to finish intermittent transmissions for cut-through. In addition, *Wakeup* routers that are involving handshaking should relay the `drain_done` signal to the neighbor *Wakeup* router in the same direction if there is any.

4.3 Adaptive FLOV through Router Voting

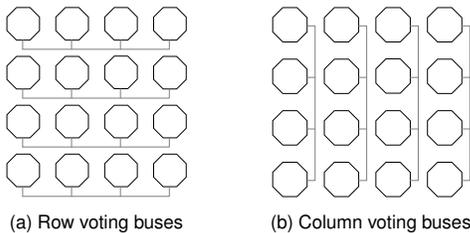


Fig. 6: Two-bit voting buses for rows (a) and columns (b) of adaptive FLOV power-gating policy.

Router power-gating is attractive in low network load since packets can be delivered with low latency. However, in medium to high loads, the network may become congested that can incur high latency overhead and sacrifice throughput. Therefore, it is important to dynamically adapt the trade-off of power saving and performance. Therefore, we propose *adaptive* FLOV (FLOV) policy to dynamically change

each router's FLOV power-gating mode among NO-FLOV, R-FLOV, and G-FLOV progressively through router voting. In a power-gating network, large packet latency is mostly due to either detours from the shortest path or long credit round-trip latency. Therefore, we adopt a voting approach that each router periodically votes for its row and column for more power saving or better performance. Then each router collects the votes of its row and column to decide its FLOV policy independently. We introduce a high and a low latency watermark *high-watermark* and *low-watermark* to compare with the average latency of received packets. If the average latency is lower than the low watermark, the router votes for more aggressive power-gating mode. If the average latency is higher than the high watermark, the router votes for more conservative power-gating mode towards performance consideration. Otherwise, it votes for no change. Given an empirical zero-load latency $latency_{zero-load}$, we define the watermarks as:

$$\begin{aligned} low-watermark &= 1.2 \times latency_{zero-load} \\ high-watermark &= 1.5 \times latency_{zero-load} \end{aligned}$$

Fig. 6 shows two-bit voting buses for rows and columns in the mesh network. These buses are time multiplexed for voting by each router in the corresponding row and column. Upon the voting time of a router, it compares the average latency of the packets received since last voting time to the watermarks. It votes -1 if the average latency is higher than *high-watermark* and votes 1 if the average latency is lower than *low-watermark*. Otherwise, it votes 0 . All the routers snoop the buses and collect the votes. If the accumulative vote is greater than zero, the router change itself to a more aggressive power-gating mode, from NO-FLOV to R-FLOV or from R-FLOV to G-FLOV; if the accumulative vote is less than zero, it regresses to a more conservative power-gating mode, from G-FLOV to R-FLOV or from R-FLOV to NO-FLOV. Otherwise, it remains unchanged. As a result, FLOV policy decides a router's power-gating mode jointly by the routers in the same row and column through voting.

5 DYNAMIC ROUTING ALGORITHMS

The FLOV NoC baseline architecture is a two dimensional mesh topology with one column or row of routers (on the edge) powered on all the time. In this paper, we assume the routers at last column are *always on* (AON routers) so as to ensure the network connectivity across the topology with the facility of FLOV links, forming an escape sub-network as shown in Fig. 7a. One VC of each powered-on router is reserved for deadlock-free routing subfunction, called an escape VC. The routing algorithms include routing for packets in the regular VCs and routing for packets in the escape sub-network. Previously, we proposed a FLOV routing algorithm that adopts a deadlock recovery mechanism, where a suspected deadlocked packet in a regular VC is sent to an escape VC to recover from deadlock [18]. In this paper, we propose a new algorithm, FLOV+, by introducing more adaptation to achieve better throughput. Instead of deadlock recovery, FLOV+ adopts Duato's Protocol to avoid deadlocks [39]. Note that routing computation is performed in *Active* and *Draining* routers, while *Sleep* and *WakeUp* routers only forward packets without changing the direc-

tion. In addition, routers in transitioning states (*Draining* or *WakeUp*) are not selected as output port candidates to avoid long transitioning time and deadlock risks.

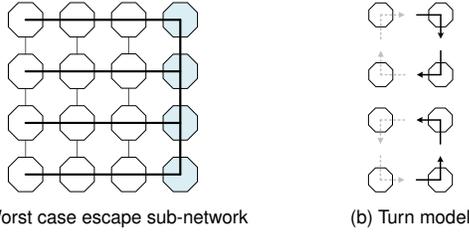


Fig. 7: Worst case of escape sub-network is shown in (a), which uses always-on routers (light blue octagons) at the last column; and turn model is shown in (b).

5.1 FLOV Routing Algorithm

The FLOV routing algorithm is based on dynamic YX/XY routing, with the consideration of power states of neighbor routers and deadlock-free escape routing subfunction.

For packets whose destinations are in the same dimension (X or Y) as the current router, the router sends them directly to the directions towards their destinations. Even in the case of power-gated downstream routers, the FLOV links ensure their delivery to the destinations. For packets whose destinations are at different dimensions (X and Y) from the current router, the paths incur turns towards their destination. If the Y neighboring router on the minimal route is powered-on, the packet is sent to the downstream router using YX routing. If it is power-gated, the power state of the X neighboring router on minimal route is checked, and if it is powered-on, the packet is forwarded to it.

When both the neighboring routers on the minimal path are power-gated, a viable path to the destination cannot be guaranteed since the current router may not be aware of the power states of the downstream routers in the further path. In this case, the packet is forwarded to the East neighbor towards AON routers and confined to the escape path and escape VCs. If the East router happens to be power-gated, the FLOV link is used for bypassing. The packet is not sent to the router in the Y direction because, in the worst scenario, if all the downstream routers in the Y direction are power-gated, the packet is not able to make a turn and hence cannot be routed to the destination. However, if the packet is directed to the East direction, we can guarantee that the packet is able to make a turn toward the destination using the AON router of the corresponding row. Note that no u-turn is allowed so as to avoid livelock situations, where a packet keeps bouncing between two neighbors.

Since the algorithm has both YX and XY decisions, it is not necessarily deadlock-free. We adopt a timeout mechanism for suspected deadlock recovery [40]. If a packet has been waiting in a buffer for an extended time, it may exceed a certain threshold and be directed to the escape VC in the downstream routers to reach the destination using the deadlock-free escape sub-network. In the escape subfunction, a packet with destination in the same dimension (X or Y) of the current router is directly sent to their destinations and use FLOV links if needed. If the above condition is not satisfied, it is forwarded to East to the AON router and make a turn toward another AON router that is located at the row as the destination. Then the packet is sent West

to its destination. Once a packet enters the escape path, it is confined to escape routing and escape VC. Based on the turn model [38], the escape subfunction is deadlock-free since it only allows four turns as shown in Fig. 7b.

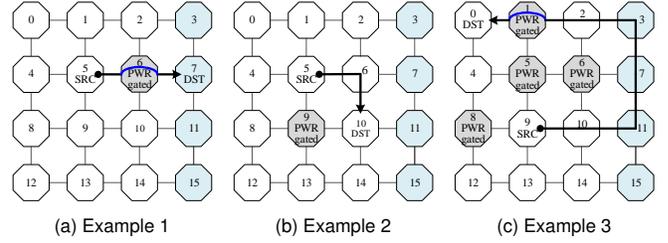


Fig. 8: FLOV routing algorithm examples 1 (a), 2 (b), and 3 (c). 'PWR gated' indicates power-gated. 'SRC' and 'DST' represent source and destination.

Fig. 8 shows three FLOV routing examples. In Fig. 8a, the destination is in the same dimension as the source router. Even though the router in between is power-gated, the packet can be forwarded to the East using the FLOV link. In Fig. 8b, the destination is in different dimensions from the source router. First, the routing algorithm checks the status of Router 9. Since Router 9 is power-gated, the packet can be sent to Router 6 that is powered-on. Then the packet makes a turn and reaches its destination. In Fig. 8c, both neighbor Routers 5 and 8 on the minimal path are power-gated, therefore, the packet is sent to Router 10 and confined to escape sub-network so that it can at least make a turn at the AON column. Router 10 computes the escape route to East toward Router 11 since the destination is not in the same dimension. Router 11 then routes the packet to Router 3 where it makes another turn toward the destination.

5.2 FLOV+ Routing Algorithm

The FLOV routing algorithm works well when only low to medium fraction of routers are power-gated. When the power-gated routers continue to increase, more packets are directed to the escape sub-network which can cause low regular VC utilization and high congestion in the escape sub-network, especially in the AON column. It also incurs detours and is not able to route packets through the shortest path in some cases. In addition, only one routing option is available for selection, which lacks adaptation and may block the packet for a long time when the only output port and VC set are busy.

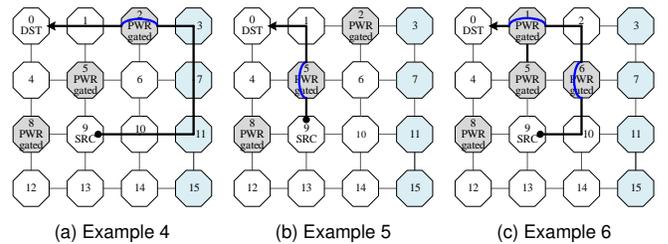


Fig. 9: Routing algorithm examples: example 4 (a) uses FLOV Routing while example 5 (b) and 6 (c) use FLOV+.

Figs. 9a and 9b show the sub-optimal and optimal routing examples, respectively. In Fig. 9a, a packet is sent from Router 9 to Router 0 using the FLOV routing algorithm. Since both physical neighbors of the source router are power-gated, the packet is directed to the East to use the escape

Algorithm 1: FLOW+ routing algorithm.

```

Input: cur, dest, in_port, in_vc
Output: routes

1 bool escape = false;
2 if IsEscape(in_vc) then
3   | escape = true;

4 (yx_port, xy_port) = GetMinimalPorts(cur, dest);
5 (yx_credits, xy_credits) = GetFreeCredits(cur, dest);
   // Add escape option with lowest priority
6 if yx_port == xy_port || at AON column then
7   | escape_port = yx_port;
8 else
9   | escape_port = East;
10 routes.Add(escape_port, escape_vc, lowest);
   // Add regular routing options
11 if escape == false then
   // Prioritize routing options
12 if yx_credits >= xy_credits then
13   | yx_pri = highest;
14   | xy_pri = high;
15 else
16   | yx_pri = high;
17   | xy_pri = highest;

   // Determine route availability
18 if logical_neighbor[yx_port] on minimal path &&
   NotUTurn(in_port, yx_port) then
19   | route_yx = true;
20 if logical_neighbor[xy_port] on minimal path &&
   NotUTurn(in_port, xy_port) then
21   | route_xy = true;

   // Add routing options
22 if route_yx == true then
23   | routes.Add(yx_port, regular_vcs, yx_pri);
24 if route_xy == true then
25   | routes.Add(xy_port, regular_vcs, xy_pri);
26 if route_yx == false && route_xy == false &&
   NotUTurn(in_port, escape_port) then
27   | routes.Add(escape_port, regular_vcs, low);

28 return routes;

```

network toward the AON column and makes turns to reach its destination, resulting in 7 hops in total. Note that there exists a shortest path from Router 9 to Router 0 by going through power-gated Router 5 to reach Router 1 to turn to the destination, traveling only 3 hops as shown in Fig. 9b. This path is not considered by FLOW routing because it ignores the relative position between the downstream *Active* router and the destination.

To tackle the aforementioned problem, we have improved the routing algorithm by leveraging the information of destination’s position relative to the downstream router’s position. Note that during handshaking, the router switching to *Sleep* state sends its corresponding logical downstream neighbor’s power state and ID in each direction to its upstream router. Therefore, the downstream routers’ relative positions to the destination can be calculated to make better routing decisions. Furthermore, more options are provided for routing selection to exploit the path di-

versity and adaptation. In this new algorithm, instead of deadlock recovery, we use deadlock avoidance by applying Duato’s Protocol where the escape route option is always provided for selection with lowest priority [39].

The new routing algorithm is described in Algorithm 1, called FLOW+ routing algorithm. With these optimizations, we can relax the burden of the escape sub-network, especially the AON column. For the FLOW escape routing sub-function, a packet is forwarded to the AON column to make turns as shown in Fig. 8c, which can lead to congestion. In contrast, when using FLOW+ routing, as shown in Fig. 9c, the routing option in line 27 of Algorithm 1 makes a detour by sending the packet from Router 9 to 10. Thus, the packet can make a turn at Router 10, then flies over Router 6 to reach Router 2, and finally turns to West toward the destination. This routing path mitigates the pressure of AON column and is the minimal path in the irregular power-gated network. Since a packet is always sending closer to the destination row and u-turn packets are directed to escape routing afterwards, the algorithm is also livelock-free. The adaptive routing options also provide higher throughput than FLOW in power-gated networks, translating to more power-gating opportunities, thereby more power savings.

6 EXPERIMENTAL EVALUATION

TABLE 1: Simulation Configuration Parameters

Parameter	Configuration
Network Topology	6×6, 8×8 (default), 10×10 and 20×20 Mesh
Input Buffer Depth	5 flits
Router	3-stage (3 cycles) router
Virtual Channel	3 regular VCs and 1 escape VC per virtual network 1 vnet for synthetic and 3 vnets for full system
Packet Size	5 flits/packet for synthetic workload 1-flit control and 5-flit data packet for full system
Memory Hierarchy	32 KB L1 I/D Cache, 8 MB L2 Cache MESI, 4 MCs at 4 corners
Technology	32 nm
Clock Frequency	2 GHz
Link	1 mm, 1 cycle, 16 B width
Power-Gating Parameters	Power-Gating overhead = 17.7 pJ wakeup latency = 10 cycles
Baseline Routing	Minimal Adaptive Routing

6.1 Experimental Methodology

We use BookSim [19] for synthetic workload experiments, and integrate it with gem5 [20] for full-system simulations. In addition, we use DSENT [5] to estimate power consumption of the interconnect components with 0.5 switching activity in 32-nm technology. We assume a 2-GHz clock frequency for the routers and links. Table 1 summarizes the simulation configuration parameters. Both synthetic and real workloads are evaluated for performance and power-saving comparisons of FLOW and FLOW+ against the baseline interconnects with no power-gating (Baseline), Router Parking (RP) and Node-Router Decoupling (NoRD). We compare both the original FLOW routing algorithm [18] and the proposed FLOW+ routing algorithm in adaptive power-gating setting. Otherwise stated, we assume 50% of the cores are power-gated that have no injections, where the power-gated cores are randomly decided. Uniform Random and Tornado traffic patterns (among powered-on cores) for synthetic workloads as well as eight benchmarks from PARSEC benchmark suite [6] are used for evaluation. For synthetic

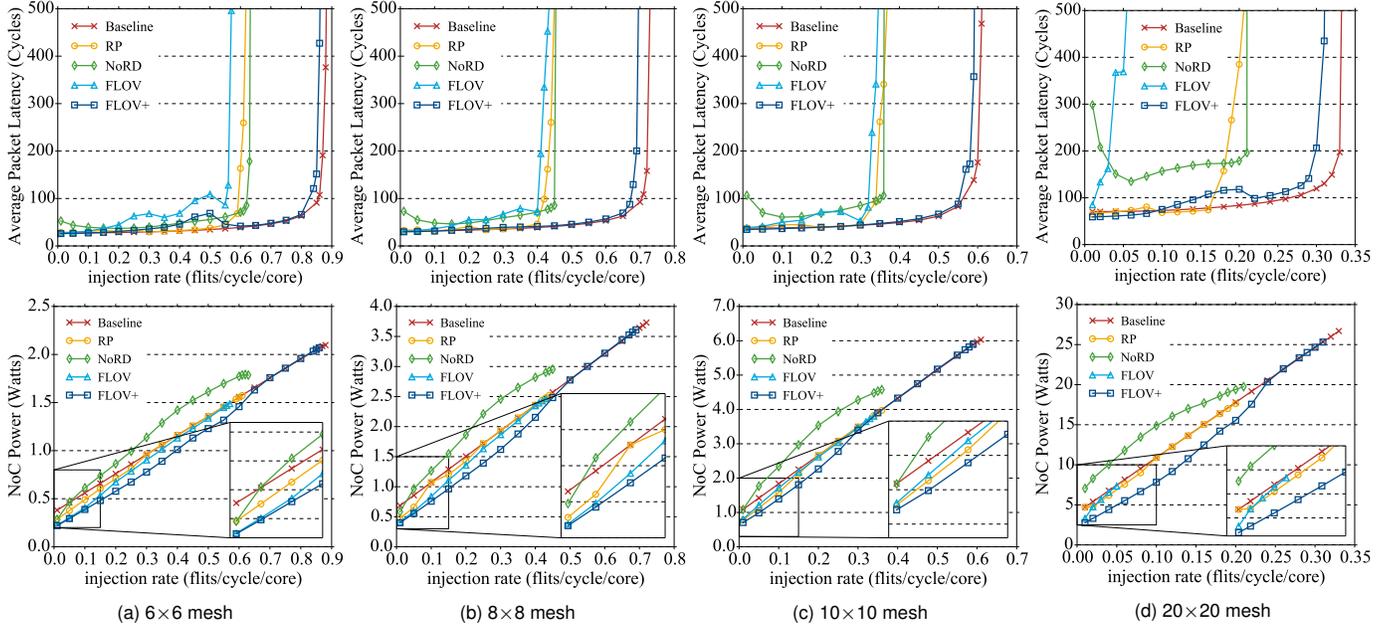


Fig. 10: Load-latency curves (top) and power consumption (bottom) under uniform random traffic with 50% cores power-gated for 6×6 (a), 8×8 (b), 10×10 (c) and 20×20 (d) mesh networks.

workloads, simulation is warmed up with the first 10,000 cycles and runs for 100,000 cycles in total. For PARSEC benchmarks, the ROI is evaluated with small input data size.

6.2 Throughput and Power Consumption

Fig. 10 shows the load-latency curves and power consumption under Uniform Random traffic with 50% cores power-gated, where Figs. 10a, 10b, 10c and 10d show for 6×6, 8×8, 10×10 and 20×20 mesh networks, respectively. Since static power is dominant in the evaluated technology, RP may stick to aggressive mode most of the time during medium to high traffic load and lead to high performance overhead. Therefore, we run aggressive, conservative and no power-gating modes, and switch to a more conservative mode for performance when the average latency increases more than 25%, unless it is already in no power-gating mode.

As shown in Fig. 10, NoRD tends to have higher latency compared to other techniques in low traffic load. While in low to medium load, both FLOV and NoRD have higher latency than the others, especially when the network is larger and more cores are power-gated. This is due to the limitation of the FLOV routing that directs a packet to escape path whenever its neighbor routers on the shortest path are power-gated, making escape network congested. NoRD may not wake up the power-gated routers early enough and suffers latency penalty, since power-gated routers are not woken up by its attached off core but by the packets that go through the ring network. In terms of throughput, the original FLOV saturates earliest due to lack of adaptation and heavy reliance on the escape network, especially when network scale is large (e.g. 20×20 mesh network in Fig. 10d). RP is better than FLOV and worse than NoRD since RP’s deterministic routing is not as flexible as NoRD’s adaptive routing. And FLOV+ outperforms these three and is the closest to Baseline. Since NoRD uses two VCs for escape and FLOV+ only has one escape VC, FLOV+ have more adaptive VC resources for better throughput. In addition, the escape routing subfunction of FLOV+ also reduces the hop counts

compared to NoRD. While comparing to Baseline, FLOV+ escape subfunction is worse than XY deterministic escape subfunction of Baseline, leading to an offset in throughput. All four network scales have similar trend in latency and throughput, showing FLOV+ scales well to different network sizes, improving throughput over the state of the art by 40% – 70%.

Fig. 10 also shows the power consumption of the corresponding injection rates. The low-load region is zoomed in for analysis, as shown in the lower right corner in the bottom figures. When traffic load is between 0 to 0.05 flits/cycle/core in 6×6 and 8×8 networks, NoRD saves static power since the network is mostly idle and it can power-gate more routers. However, as traffic load increases, NoRD starts to consume a bit more power than Baseline, mainly due to the long detour of the escape ring network. This is similar to the discovery in the literature [4]. This overhead tends to be higher for larger network size. In the 20×20 mesh network as shown in Fig. 10d, NoRD can hardly save power. Other techniques save more power when load is low and save less as load increases. Among them, FLOV+ saves the most static power under the same load and RP saves the least static power. When the load increases to the high region, FLOV+ and Baseline consume similar amounts of power.

6.3 Power-Gating Case Studies

In this subsection, we discuss cases of power-gating under different traffic patterns in low injection rates by varying the portion of power-gated cores. Fig. 11 summarizes the results in low load using Uniform Random traffic under different core power-gating scenarios. Similarly, Fig. 12 shows the results for Tornado traffic. The left subplots show average packet latency with breakdowns into router latency (number of hops × router pipeline latency), link latency (total link traversals), serialization latency (number of flits per packet), contention latency, and FLOV latency (number of FLOV routers/links traversed), while the right subplots

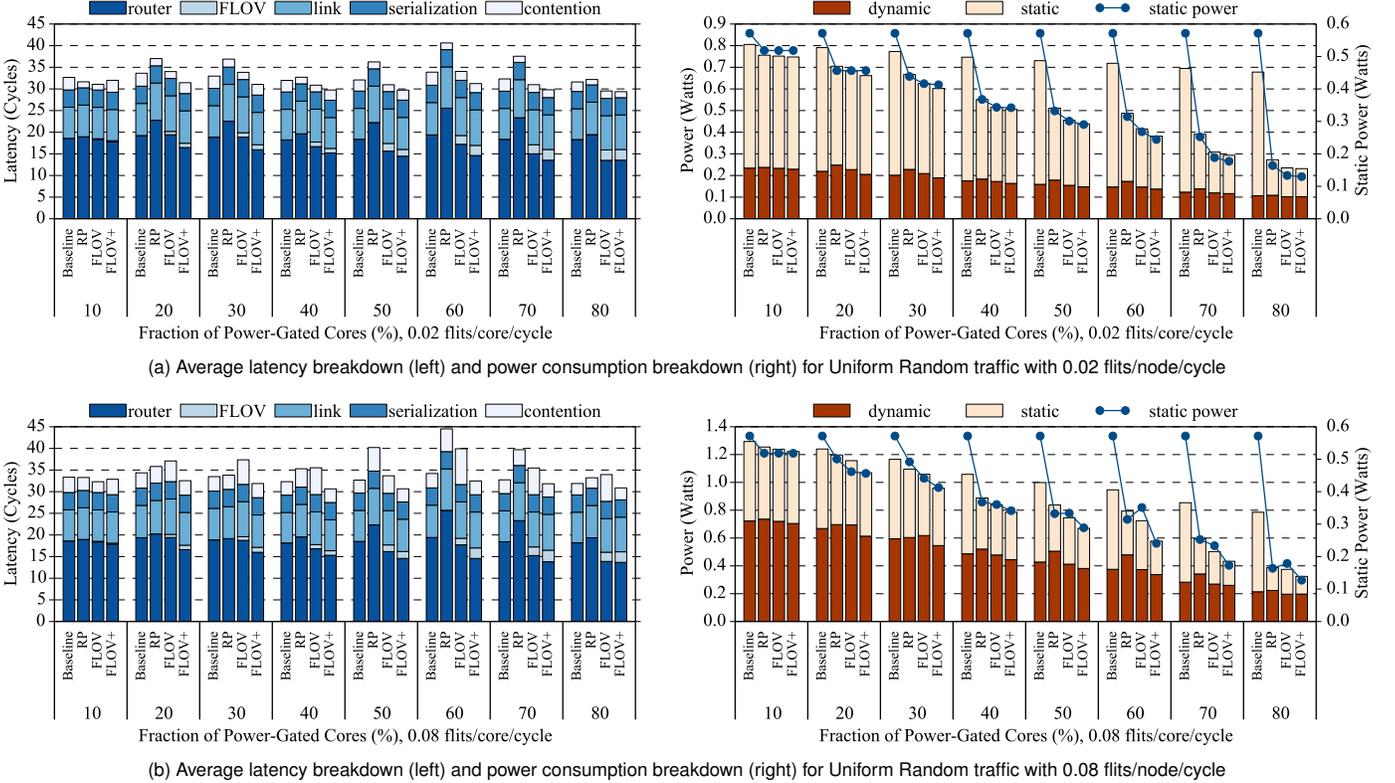


Fig. 11: Average latency breakdown (left) and power comparison breakdown (right) for injection rates of 0.02 (a) and 0.08 (b) flits/node/cycle with Uniform Random traffic pattern.

show power consumption with breakdown into dynamic and static components. As NoRD incurs high latency overhead in low load, we only compare to Baseline and RP to reveal details in this subsection. For RP, we have run both aggressive and conservative modes and selected the mode that consumes less power.

6.3.1 Performance

Average latency comparison of FLOV, FLOV+ with RP and Baseline under Uniform Random and Tornado traffic patterns are shown in the left subplots in Figs. 11 and 12, respectively. FLOV and FLOV+ outperform RP across different traffic patterns and injection rates except that FLOV is worse than RP in cases of 20% and 30% of power-gated cores under Uniform Random traffic in higher injection rate (0.08 flits/cycle/core in Fig. 11b). In RP, a packet always routes through powered-on routers, which may be non-minimal, thereby increasing the path length. In contrast, FLOV and FLOV+ take advantage of all the links and route a packet through a minimal path in the best effort using FLOV links. Even when minimal routing is impossible for some cases in the escape sub-network, the average packet latency can be reduced since the FLOV links do not incur the 3-cycle baseline router per-hop latency, as the flit is only temporarily held in the FLOV latch for one cycle. This can be observed clearly that the router latency for RP is larger than that of FLOV and FLOV+ due to detours. In Fig. 11, under Uniform Random traffic, the FLOV latency increases as more cores are power-gated for the FLOV and FLOV+, which show increased FLOV link utilization. Under Tornado traffic, the communication occurs between two power-on nodes in the same row/column, and the routers in the rightmost column

are always active. Therefore, less number of FLOV links are used, which leads to reduced FLOV latency as shown in Fig. 12. In Fig. 11b, when 20% and 30% of cores are power-gated, RP is slightly better than FLOV since it transitions to conservative mode as dynamic power penalty is higher than static power saving, which also improves performance. This can also be observed from right subplots of Fig. 11 in the configuration with 30% power-gated cores, where static power consumption for this injection rate (Fig. 11b) is higher than 0.02 flits/cycle/core injection rate (Fig. 11a). When 20% of cores are power-gated under Tornado, RP has lower latency in higher injection rate (Fig. 12b) compared to in lower injection rate (Fig. 12a). This is because it switches to conservative mode that improves latency. In the above cases, RP trades off static power savings for latency benefits. However, FLOV and FLOV+ achieve both power saving and performance guarantee in these cases.

Another observation is that as the injection rate increases from 0.02 to 0.08, the performance impact on RP is higher than on FLOV+. Figs. 11 and 12 show higher contention latency for RP than FLOV+ when injection rate increases from 0.02 to 0.08. This is because certain routers, connecting different network partitions to ensure network connectivity, become network hotspots in RP. In contrast, The proposed FLOV+ routing algorithm avoids such network hotspots.

In Fig. 12, both FLOV and FLOV+ outperform Baseline with Tornado traffic. This is because in Tornado, the traffic injected to each router is destined to a router in the same row/column. Thus, FLOV and FLOV+ can use FLOV links with minimal paths and avoid the 3-cycle router latency. FLOV+ also shows better performance than Baseline even in Uniform Random traffic as shown in Fig. 11. This is due

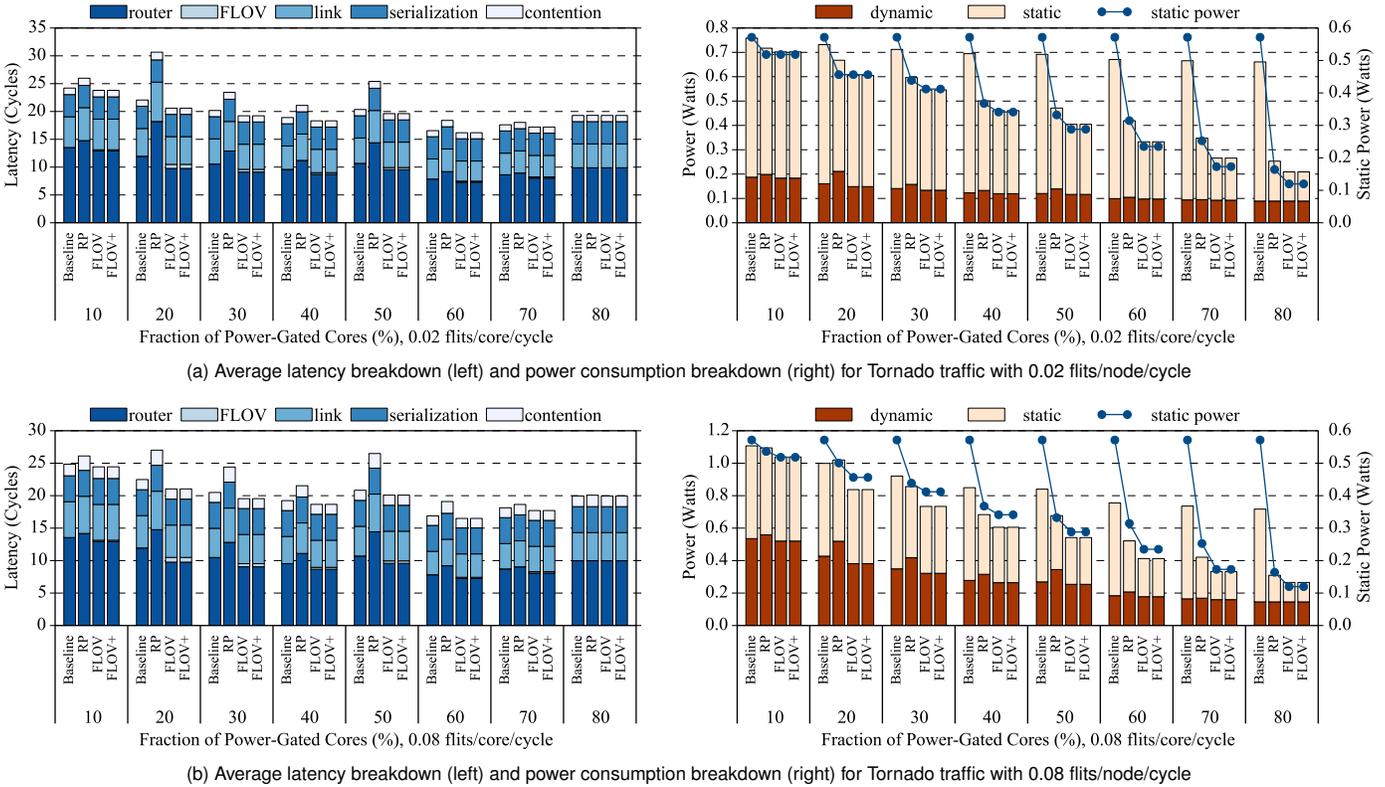


Fig. 12: Average latency breakdown (left) and power comparison breakdown (right) for injection rates of 0.02 (a) and 0.08 (b) flits/node/cycle with Tornado traffic pattern.

to the fast FLOV link and the proposed routing algorithm, which considers logical neighbor positions to provide best-effort shortest path.

In Figs. 11 and 12, both FLOV and FLOV+ have relatively higher contention latency at higher injection rate. One reason is that packets have a higher probability of being routed to the AON router column for guaranteed paths to the destinations, which may create congestion in the AON router column. Also, when packets are routed through consecutive FLOV links in a row/column, packet transmission may be delayed due to the longer credit round-trip latency across consecutive gated routers. However, the higher utilization of FLOV links compensates for the contention latency, which can be explained by the router and FLOV latency. Note that RP also tends to have higher contention latency compared to the FLOV and FLOV+ because of the high probability of hotspot creation.

As the number of power-gated cores increases, FLOV, FLOV+ and RP all power gate more routers. However, only FLOV+ has stable latency while RP is the worst. From our study, we have observed that NoRD can be even worse in such low loads, since the attached core of the power-gated router is also off, thereby it may take longer detour in order to wake up routers when more cores are power-gated.

6.3.2 FLOV Routing versus FLOV+ Routing

FLOV+ routing algorithm has similar trend with FLOV routing algorithm as compared to RP. In Fig. 11, FLOV+ can achieve lower latency than FLOV. This is mainly due to the fact that FLOV+ has a higher chance to route packets through the shortest path instead of sending to the AON column to make turns, reducing the number of hops traversed with respect to FLOV routing. Such benefit is confirmed by

the lower router latency in FLOV+ than that in FLOV. Interestingly, FLOV+ routing achieves better latency performance than Baseline. This is owing to the fast FLOV links. Since FLOV+ can take advantage of the one-cycle FLOV links on the minimal path, it can avoid the router pipeline stages as in Baseline for faster packet delivery.

6.3.3 Power Consumption

The subplots on the right in Figs. 11 and 12 show the power consumption breakdowns into dynamic and static power. The static power is also depicted as secondary results. For both injection rates, the dynamic power consumption of FLOV and FLOV+ are lower than RP, since in RP every hop in the rerouted packet traversal requires the full router pipeline execution, whereas in FLOV and FLOV+ the intermediate power-gated routers use FLOV links that consume significantly lower power. RP also consumes more dynamic power than Baseline due to its non-minimal path rerouting of packets as the number of power-gated cores increases. At higher fractions of power-gated cores, FLOV and FLOV+ consume less dynamic power than Baseline by avoiding router pipelines.

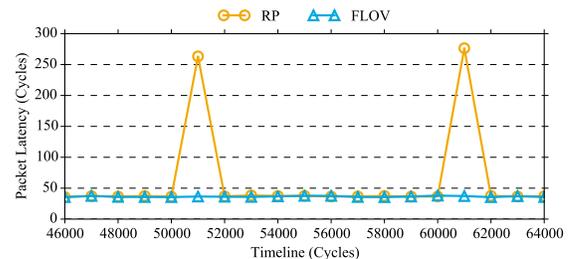


Fig. 13: Reconfiguration overhead of RP and comparison with FLOV.

For static power consumption, under Uniform Random traffic, FLOV and FLOV+ saves more than RP by power-gating more routers, which are required to be powered-on in RP to maintain network connectivity. FLOV+ consumes even less static power than FLOV, since FLOV needs to keep more routers in R-FLOV mode to guarantee performance due to routing limitations. While under Tornado traffic pattern, the only different trend compared to Uniform Random is that both FLOV and FLOV+ save the same amounts of power. This is because the communications only happen in the same row/column, which are handled in the same way in both routing algorithms.

6.4 Reconfiguration Overhead Analysis

We analyzed the impact of the network reconfiguration on packet latency by the comparison of RP and FLOV. Fig. 13 shows average packet latency of FLOV and RP across the timeline of execution under Uniform Random traffic with 0.02 flits/cycle/node injection rate when 10% of the cores are power-gated. In RP, whenever the configuration of power-gated cores changes (at 50,000 and 60,000 cycles), the network has to be reconfigured by the Fabric Manager who recomputes and distributes the routing tables to the routers that are active in the next epoch (Phase I of reconfiguration protocol in RP). During reconfiguration, the network has to stall and no new injections are allowed except reconfiguration packets, which incurs additional queuing delays in packet latency. Our evaluations show that the reconfiguration time in RP Phase I is more than 700 cycles. We observe that the newly injected packets during this time experience significant queuing delays in RP. In FLOV, there is negligible handshaking overhead since the routers are power-gated in a distributed manner. So new packet transmissions can be initiated while some routers either power-gate or wake up independently.

6.5 Real Workload Evaluation

We also ran PARSEC 2.1 using gem5 [20] with BookSim integrated for full-system evaluation. For RP, we ran both aggressive and conservative mode, only the one with lower energy-delay product is selected for better efficiency. The system parameters are described in Table 1.

Fig. 14a shows the runtime speedup of RP, NoRD, FLOV and FLOV+ over Baseline. It shows that FLOV+ has the least negative impact on performance, and even has a small improvement (3.9%) compared to Baseline. FLOV and RP have negligible difference compared to Baseline while NoRD degrades performance by around 10%. Note that PARSEC applications have low network traffic loads, making them beneficial for power-gating. Since FLOV+ has a better routing algorithm compared to FLOV, and the one-cycle FLOV link compensates for the detour and round-trip credit loop latency, FLOV+ helps reduce network latency and improve performance. On the other hand, NoRD incurs high detour at low traffic load, leading to high latency. This phenomenon is more severe in larger network scales such as the evaluated 8×8 mesh network. Since the network traffic is low, RP has little negative effect on latency, thereby has similar performance as Baseline.

Fig. 14b shows the dynamic and static NoC power breakdown normalized to Baseline. It shows that RP, NoRD,

FLOV, FLOV+ reduces the total power consumption by 25%, 15%, 25%, and 31%, respectively. For static power consumption, RP, NoRD, FLOV, and FLOV+ achieve power reductions of 34%, 32%, 34% and 41%, respectively. Although NoRD saves the second most static power, the detour incurred by the ring introduces extra dynamic power consumption, offsetting the benefit and ending up saving the least total power. RP and FLOV have similar savings for both dynamic and static power, while FLOV+ saves the most power for both components. FLOV+ reduces power consumption by 8% and 20% compared to RP and NoRD, respectively. Note that in this work, routers in FLOV and FLOV+ adaptively change their power-gating modes independently through router voting. In the prior version [18], if every router is configured to G-FLOV mode, it can save much more power than RP but may not adapt to applications that have higher traffic as the adaptive one in this paper.

6.6 Area Overhead Analysis

In the proposed router microarchitecture, the modifications include 4 muxes and 4 demuxes in addition to the four output latches. The mux and demux selection signals are only toggled when the router wakes up or is power-gated, so the logic needed for the select signals is minimal. Every router has two sets of PSRs, where each entry incurs a 2 bit overhead (for power state). Hence the total overhead for the PSRs accounts to 16 bits (2 sets of 4-entry registers). The credit control logic is modified to be connected to the HSC so that the credit counters can be reset or zeroed based on the signals from the HSC. The additional overhead incurred due to this change is mainly the connecting wires and minor modifications to the CCL logic for decoding the two HSC signals. The HSC requires 6-bit wires to connect the adjacent neighboring routers (4 bits for current and logical neighbor router power state change notifications, 1 bit for draining notification and 1 bit for physical neighbor assertion). In addition, the voting mechanism needs a 2-bit bus for voting snoop. This is approximately 0.1% of the baseline router area according to DSENT [5]. The overall area overhead for the above components for a single router in 32-nm technology is estimated about $2.8 \times 10^{-3} \text{ mm}^2$ which is 3% of the baseline router area. The power consumption of the HSC is also minimal due to the handshaking occurring only after long intervals of time (reconfiguration times) as shown in Section 6. None of the modifications incur any significant critical path delay and do not affect the frequency of the NoC operations.

7 CONCLUSIONS

In this paper, FLOV is enhanced with a voting approach and a better routing algorithm to enable efficient and adaptive NoC power-gating using Fly-Over links. By router voting, each router collects votes from both dimensions of the network and adapts the power-gating mode among G-FLOV, R-FLOV and NO-FLOV with performance awareness, making it flexible for all different traffic loads. In addition, the proposed routing algorithm introduces more adaptivity as well as logical neighbor information for routing decisions, achieving the best-effort minimal route. As a result, FLOV improves the throughput by more than 40% compared to

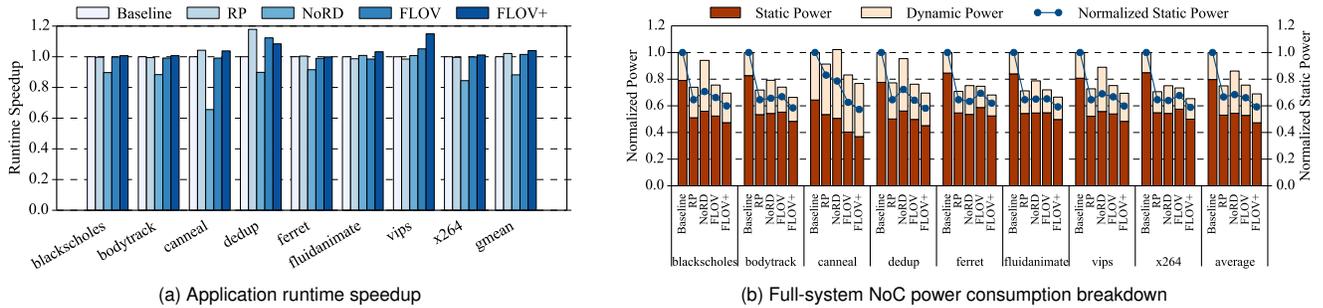


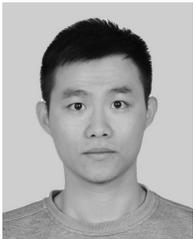
Fig. 14: Full-system simulation results: application runtime speedup (a), and normalized NoC power consumption (b) over Baseline.

the state of the art, similar to the Baseline, while saving more power. Additionally, it scales well for different network sizes and outperforms other approaches. Our full system evaluation shows even performance improvement over Baseline by 3.9%, and power reduction by 31% and 20% compared to Baseline and the state of the art, respectively. In summary, FLOV not only reduces power consumption but also adapts to maintain performance, improving both power and energy efficiency.

REFERENCES

- [1] M. B. Taylor, J. Kim, J. Miller, D. Wentzclaff, F. Ghodrati, B. Greenwald, H. Hoffman, P. Johnson, J.-W. Lee, W. Lee, A. Ma, A. Saraf, M. Seneski, N. Shnidman, V. Strumpfen, M. Frank, S. Amarasinghe, and A. Agarwal, "The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs," *IEEE Micro*, vol. 22, no. 2, pp. 25–35, 2002.
- [2] J. Howard, S. Dighe, S. R. Vangal, G. Ruhl, N. Borkar, S. Jain, V. Erraguntla, M. Konow, M. Riepen, M. Gries, G. Droege, T. Lund-Larsen, S. Steibl, S. Borkar, V. K. De, and R. Van Der Wijngaart, "A 48-Core IA-32 Processor in 45nm CMOS Using On-Die Message-Passing and DVFS for Performance and Power Scaling," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 173–183, 2011.
- [3] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-GHz Mesh Interconnect for a Teraflops Processor," *IEEE Micro*, vol. 27, no. 5, pp. 51–61, 2007.
- [4] L. Chen and T. M. Pinkston, "NoRD: Node-Router Decoupling for Effective Power-Gating of On-Chip Routers," in *International Symposium on Microarchitecture (MICRO)*. IEEE Computer Society, 2012, pp. 270–281.
- [5] C. Sun, C.-H. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L.-S. Peh, and V. Stojanovic, "DSENT – A Tool Connecting Emerging Photonics with Electronics for Opto-Electronic Networks-on-Chip Modeling," in *International Symposium on Networks on Chip (NoCS)*. IEEE, 2012, pp. 201–210.
- [6] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC Benchmark Suite: Characterization and Architectural Implications," in *International Conference on Parallel Architectures and Compilation Techniques (PACT)*. ACM, 2008, pp. 72–81.
- [7] N. Madan, A. Buyuktosunoglu, P. Bose, and M. Annavaram, "A Case for Guarded Power Gating for Multi-Core Processors," in *International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2011, pp. 291–300.
- [8] J. Lee and N. S. Kim, "Optimizing Throughput of Power- and Thermal-Constrained Multicore Processors Using DVFS and Per-Core Power-Gating," in *Design Automation Conference (DAC)*. IEEE, 2009, pp. 47–50.
- [9] J. Leverich, M. Monchiero, V. Talwar, P. Ranganathan, and C. Kozyrakis, "Power Management of Datacenter Workloads Using Per-Core Power Gating," *Computer Architecture Letters*, vol. 8, no. 2, pp. 48–51, 2009.
- [10] H. Matsutani, M. Koibuchi, D. Ikebuchi, K. Usami, H. Nakamura, and H. Amano, "Ultra Fine-Grained Run-Time Power Gating of On-Chip Routers for CMPs," in *International Symposium on Networks-on-Chip (NOCS)*. IEEE, 2010, pp. 61–68.
- [11] R. Parikh, R. Das, and V. Bertacco, "Power-Aware NoCs through Routing and Topology Reconfiguration," in *Design Automation Conference (DAC)*. IEEE, 2014, pp. 1–6.
- [12] A. Samih, R. Wang, A. Krishna, C. Maciocco, C. Tai, and Y. Solihin, "Energy-Efficient Interconnect via Router Parking," in *International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2013, pp. 508–519.
- [13] L. Chen, D. Zhu, M. Pedram, and T. M. Pinkston, "Power Punch: Towards Non-Blocking Power-Gating of NoC Routers," in *International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2015, pp. 1–12.
- [14] H. Zheng and A. Louri, "EZ-Pass: An Energy & Performance-Efficient Power-Gating Router Architecture for Scalable Nocs," *IEEE Computer Architecture Letters*, vol. 17, no. 1, pp. 88–91, 2017.
- [15] H. Farrokhbakht, H. M. Kamali, and N. E. Jerger, "Muffin: Minimally-Buffered Zero-Delay Power-Gating Technique in On-Chip Routers," in *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2019, pp. 1–6.
- [16] H. Matsutani, M. Koibuchi, D. Wang, and H. Amano, "Run-Time Power Gating of On-Chip Routers Using Look-Ahead Routing," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE Computer Society Press, 2008, pp. 55–60.
- [17] A. Vega, A. Buyuktosunoglu, and P. Bose, "SMT-Centric Power-Aware Thread Placement in Chip Multiprocessors," in *International Conference on Parallel Architectures and Compilation Techniques (PACT)*. IEEE, 2013, pp. 167–176.
- [18] R. Boyapati, J. Huang, N. Wang, K. H. Kim, K. H. Yum, and E. J. Kim, "Fly-Over: A Light-Weight Distributed Power-Gating Mechanism for Energy-Efficient Networks-on-Chip," in *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2017, pp. 708–717.
- [19] N. Jiang, D. U. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. E. Shaw, J.-H. Kim, and W. J. Dally, "A Detailed and Flexible Cycle-Accurate Network-on-Chip Simulator," in *International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2013, pp. 86–96.
- [20] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The Gem5 Simulator," *SIGARCH Comput. Archit. News*, vol. 39, pp. 1–7, 2011.
- [21] R. Kumar, A. Martínez, and A. González, "Dynamic Selective Devectorization for Efficient Power Gating of SIMD Units in a HW/SW Co-Designed Environment," in *International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. IEEE, 2013, pp. 81–88.
- [22] V. Soteriou and L.-S. Peh, "Design-Space Exploration of Power-Aware On/Off Interconnection Networks," in *International Conference on Computer Design (ICCD)*. IEEE, 2004, pp. 510–517.
- [23] G. Kim, J. Kim, and S. Yoo, "Flexibuffer: Reducing Leakage Power in On-Chip Network Routers," in *Design Automation Conference (DAC)*. IEEE, 2011, pp. 936–941.
- [24] 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," in *International Symposium on Low Power Electronics and Design (ISLPED)*. ACM, 2003, pp. 459–464.
- [25] J. Zhan, Y. Xie, and G. Sun, "NoC-Sprinting: Interconnect for Fine-Grained Sprinting in the Dark Silicon Era," in *Proceedings of the 51st Annual Design Automation Conference (DAC)*. ACM, 2014, pp. 1–6.
- [26] R. Das, S. Narayanasamy, S. K. Satpathy, and R. G. Dreslinski, "Catnap: Energy Proportional Multiple Network-on-Chip," in *ACM SIGARCH Computer Architecture News*, vol. 41, no. 3. ACM, 2013, pp. 320–331.

- [27] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, "Express Virtual Channels: Towards the Ideal Interconnection Fabric," in *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2. ACM, 2007, pp. 150–161.
- [28] A. Kodi, A. Louri, and J. Wang, "Design of Energy-Efficient Channel Buffers with Router Bypassing for Network-on-Chips (NoCs)," in *International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2009, pp. 826–832.
- [29] U. Y. Ogras and R. Marculescu, "Application-Specific Network-on-Chip Architecture Customization via Long-Range Link Insertion," in *International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2005, pp. 246–253.
- [30] —, "It's a Small World After All": NoC Performance Optimization via Long-Range Link Insertion," *IEEE Transaction on Very Large Scale Integration Systems*, vol. 14, no. 7, pp. 693–706, 2006.
- [31] S. J. Hollis, C. Jackson, P. Bogdan, and R. Marculescu, "Exploiting Emergence in On-Chip Interconnects," *IEEE Transactions on Computers*, vol. 63, no. 3, pp. 570–582, 2014.
- [32] R. Boyapati, J. Huang, N. Wang, K. H. Kim, K. H. Yum, and E. J. Kim, "Poster: Fly-Over: A Light-Weight Distributed Power-Gating Mechanism for Energy-Efficient Networks-on-Chip," in *2016 International Conference on Parallel Architecture and Compilation Techniques (PACT)*. IEEE, 2016, pp. 413–414.
- [33] H. Farrokhbakht, M. Taram, B. Khaleghi, and S. Hessabi, "Toot: an efficient and scalable power-gating method for noc routers," in *2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*. IEEE, 2016, pp. 1–8.
- [34] H. Farrokhbakht, H. M. Kamali, N. E. Jerger, and S. Hessabi, "Sponge: A scalable pivot-based on/off gating engine for reducing static power in noc routers," in *Proceedings of the International Symposium on Low Power Electronics and Design*, 2018, pp. 1–6.
- [35] Y. Xue and P. Bogdan, "Improving NoC Performance under Spatio-Temporal Variability by Runtime Reconfiguration: A General Mathematical Framework," in *2016 tenth IEEE/ACM international symposium on networks-on-chip (NOCS)*. IEEE, 2016, pp. 1–8.
- [36] Z. Qian, P. Bogdan, G. Wei, C.-Y. Tsui, and R. Marculescu, "A Traffic-Aware Adaptive Routing Algorithm on A Highly Reconfigurable Network-on-Chip Architecture," in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, 2012, pp. 161–170.
- [37] M. Radetzki, C. Feng, X. Zhao, and A. Jantsch, "Methods for Fault Tolerance in Networks-on-Chip," *ACM Computing Surveys (CSUR)*, vol. 46, no. 1, pp. 1–38, 2013.
- [38] C. J. Glass and L. M. Ni, "The Turn Model for Adaptive Routing," in *International Symposium on Computer Architecture (ISCA)*, 1992, p. 278–287.
- [39] J. Duato, "A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 12, pp. 1320–1331, 1993.
- [40] K. Anjan and T. M. Pinkston, "An Efficient, Fully Adaptive Deadlock Recovery Scheme: DISHA," in *proceedings of the 22nd annual international symposium on computer Architecture*, 1995, pp. 201–210.



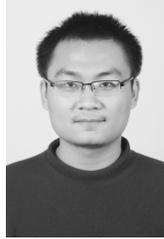
Jiayi Huang received the B.Eng degree in information and communication engineering from Zhejiang University in 2014, and the Ph.D. degree in computer engineering from Texas A&M University in 2020. He is a postdoctoral researcher in the Department of Electrical and Computer Engineering at The University of California, Santa Barbara. His research interests include computer architecture, networks-on-chip and interconnection networks. He is a member of the IEEE Computer Society and the ACM.



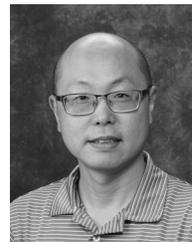
Shilpa Bhosekar received her B.E degree in Electronics and Communication from Birla Institute of Technology and Science, Pilani, India in 2014 and MS degree in Computer Engineering from Texas A&M University in 2018. She currently works as an Architecture Engineer with Ambarella. From 2014-16, she worked as an ASIC Engineer with Nvidia for their Tegra line of SoCs. Her research interests include computer architecture, machine learning and low-power design.



Rahul Boyapati received the B.Tech degree from National Institute of Technology Warangal, India, the MS and Ph.D. degrees from Texas A&M University. He is currently working as a design engineer in Intel Corp., Hillsboro, Oregon. His research interests include computer architecture, exascale multicore systems, interconnection networks, power-efficient designs and approximate computing.



Ningyuan Wang received the BS degree in Applied Physics and BE degree in Computer Science and Technology from University of Science and Technology of China, China, in 2013, the MS degree in Computer Engineering from Texas A&M University, USA, in 2015. He is a Software Engineer in Google LLC, Mountain View, California. His research interests include computer architecture, low-power design, and performance evaluation.



Byul Hur received his B.S. degree in Electronics Engineering from Yonsei University, in Seoul, Korea, in 2000, and his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Florida, Gainesville, FL, USA, in 2007 and 2011, respectively. In 2017, he joined the faculty of Texas A&M University, College Station, TX, USA, where he is currently an Assistant Professor. He worked as a postdoctoral associate from 2011 to 2016 at the University Florida previously. His research interests include Mixed-signal/RF circuit design and testing, measurement automation, environmental & biomedical data measurement, and educational robotics development.



Ki Hwan Yum received the BS degree in mathematics from Seoul National University, Korea, in 1989, the MS degree in computer science and engineering from Pohang University of Science and Technology, Korea, in 1994, and the PhD degree in computer science and engineering from the Pennsylvania State University in 2002. From 1994 to 1997, he was a member of Technical Staff in Korea Telecom Research and Development Group. He is currently an research assistant professor in the Department of Computer Science and Engineering, Texas A&M University. His research interests include Computer Architecture, Parallel/Distributed Systems, Cluster Computing, and Performance Evaluation. He is a member of the IEEE, the IEEE Computer Society, and the ACM.



Eung Jung Kim received the BS degree in computer science from KAIST, Korea, the MS degree in computer science from Pohang University of Science and Technology, Korea, and the PhD degree from the Department of Computer Science and Engineering, Pennsylvania State University. She is an associate professor in the Department of Computer Science and Engineering, Texas A&M University. Her research interests include computer architecture, power efficient systems, parallel/distributed systems, cluster computing, security and sensor network. She worked as a member of technical staff in Korea Telecom for three years. She is a member of the IEEE Computer Society. More information about her research is available at <http://faculty.cse.tamu.edu/ejkim>.