

# Fly-Over: A Light-Weight Distributed Power-Gating Mechanism for Energy-Efficient Networks-on-Chip

Rahul Boyapati\*, Jiayi Huang\*, Ningyuan Wang†, Kyung Hoon Kim, Ki Hwan Yum and Eun Jung Kim

Department of Computer Science and Engineering, Texas A&M University

Email: {rahulboyapati,jyhuang,khkim,yum,ejkim}@cse.tamu.edu

†Google Inc., Email: nywang@google.com

**Abstract**—Scalable Networks-on-Chip (NoCs) have become the *de facto* interconnection mechanism in large scale Chip Multiprocessors. Not only are NoCs devouring a large fraction of the on-chip power budget but static NoC power consumption is becoming the dominant component as technology scales down. Hence reducing static NoC power consumption is critical for energy-efficient computing. Previous research has proposed to power-gate routers attached to inactive cores so as to save static power, but requires centralized control and global network knowledge. In this paper, we propose *Fly-Over (FLOV)*, a light-weight distributed mechanism for power-gating routers, which encompasses FLOV router architecture, handshake protocols, and a partition-based dynamic routing algorithm to maintain network functionalities. With simple modifications to the baseline router architecture, FLOV can facilitate FLOV links over power-gated routers. Then we present two handshake protocols for FLOV routers, *restricted FLOV* that can power-gate routers under restricted conditions and *generalized FLOV* with more power saving capability. The proposed routing algorithm provides best-effort minimal path routing without the necessity for global network information. We evaluate our schemes using synthetic workloads as well as real workloads from PARSEC 2.1 benchmark suite. Our full system evaluations show that FLOV reduces the total and static energy consumption by 18% and 22% respectively, on average across several benchmarks, compared to state-of-the-art NoC power-gating mechanism while keeping the performance degradation minimal.

**Keywords**-Networks-on-Chip; Power-Gating; Routing Algorithms

## I. INTRODUCTION

Chip Multiprocessors (CMPs), scaled to 100s and 1000s of cores, are touted as the future solution for extracting huge performance gains using parallel programming paradigms. This is possible, as stated by Moore’s law [1], because of shrinking transistor sizes and allowing for denser on-chip packaging. However the failure of Dennard Scaling [2], supply voltage not scaling down with the transistor size, means that all the components on the chip cannot be run simultaneously without breaking the power and thermal constraints. Thus future CMP designs will have to work under stricter power envelopes. Scalable Networks-on-chip (NoCs), like 2D meshes, have become *de facto* interconnection mechanisms in these large CMPs. Recent studies [3]–[5] have shown that NoCs consume a significant portion, ranging from 10%

to 36%, of the total on-chip power budget. Hence power-efficient NoC designs are of the highest priority for power-constrained future CMPs.

Static power consumption of the on-chip circuitry is increasing at an alarming rate with the scaling down of feature sizes and chip operating voltages towards near-threshold levels. Previous studies [6]–[10] have shown 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. According to this trend, as we reach towards sub-10nm feature sizes, static power will become the major portion of the NoC power consumption.

Power-gating, cutting off supply current to idle chip components, is an effective circuit-level technique that can be used to mitigate the worsening impact of on-chip static power consumption. Due to low average core utilization in most modern workloads [11], [12], significant number of studies have proposed efficient mechanisms for power-gating cores with marginal impact on performance [13]–[15]. Some studies [10], [16] have proposed power-gating selected router components in a fine-grained fashion using topology reconfiguration. However limited research [8], [17], [18] has been done regarding mechanisms for power-gating routers, which will reduce NoC static power consumption.

Previous research has been proposed to power-gate routers, either by reacting to the network traffic [8] or based on the power state of the attached core [17]. Significant research at Operating System (OS) level has been proposed for achieving static power savings in CMPs by power-gating idle cores by consolidating the thread executions to fewer cores [13]–[15], [19]. Therefore, it is imperative to design router power-gating mechanisms that can work in synergy with OS level core power-gating mechanisms. Router Parking (RP) [17] 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. RP also creates a single point of failure if the centralized fabric manager goes down.

We propose Fly-Over (FLOV), a light-weight distributed power-gating mechanism that eliminates the need for centralized control to power-gate routers. FLOV tries to power-

\* authors contributed equally

gate routers as soon as the attached cores are powered down by the OS, in a distributed manner. Since such a distributed power-gating mechanism may create interconnect partitions without communication paths, FLOV links in power-gated routers are provided to enable incoming packets to travel straight through for network connectivity.

Specifically, FLOV comprises FLOV router architecture, handshake protocols, and its partition-based dynamic routing algorithm. We design FLOV router architecture by modifying the baseline router architecture to provide FLOV links over power-gated routers. Based on this FLOV architecture, we first present a handshake protocol working under restricted conditions, called *restricted FLOV (rFLOV)*, where no consecutive routers in a row/column can be power-gated at the same time. Then another handshake protocol, called *generalized FLOV (gFLOV)*, is presented, where two or more consecutive routers in a row/column can be power-gated simultaneously. Clearly, rFLOV is simpler than gFLOV, but gFLOV can provide more power saving capability. We propose a dynamic routing algorithm that ensures network routing functionality without the need for any global NoC information or needing to wakeup intermediate power-gated routers. The routing algorithm dynamically decides the output direction based on the destination and the power states of its neighboring routers.

We evaluate the FLOV scheme using BookSim [20], a cycle-accurate interconnect simulator, for detailed NoC evaluation and using gem5 [21] for full system evaluation, and compare against RP [17]. Our full system evaluations show that FLOV reduces the total and static energy consumption by 18% and 22% respectively, on average across several benchmarks, compared to state-of-the-art NoC power-gating mechanism while keeping the performance degradation minimal.

## II. RELATED WORK

Recently significant research [13], [22] has been performed in applying power-gating techniques in NoCs for power savings. Kim et al. [23] and Bokhari et al. [?] proposed to save link and NoC dynamic power using voltage/frequency scaling. Soteriou et al. [24], Matsutani et al. [16], Kim et al. [25] and Parikh et al. [10] propose fine grained power-gating of components inside the NoC router. But such approaches require significant additional power gating circuitry. These approaches work well to reduce the static power consumption, however, they only power-gate certain components of a router.

In [18], Chen et al. introduced a performance-aware, 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. Catnap [26] proposed a mechanism where a light-weight subnetwork can be power-gated based on the priority and predicted traffic load. This work is orthogonal with FLOV,

since FLOV can be applied on top of the powered-on subnetworks to achieve even more power savings.

Chen et al. [8] proposed a node-router decoupling (NoRD) approach to leverage the independence of power-gating a core and its attached router. They provide a decoupling bypass route that connects the ejection and injection channels to form a bypass link to the router. The decoupling bypass links ensure network connectivity even for the extreme cases of all routers being turned off by using an escape 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.

Samih et al. [17] proposed Router Parking (RP) to power-gate as many routers as possible when their attached cores are sleeping while maintaining network connectivity. RP dynamically parks (or power-gates) routers to maintain a balanced trade-off between power saving and performance. However, this scheme requires centralized control using a Fabric Manager (FM) and typically takes a long time to reconfigure the network that may suspend new injections into the network during this phase. On the other hand, FLOV is a distributed power-gating mechanism that avoids the need for centralized control and keeps the network functionalities while routers are being power-gated.

Zhan et al. [27] propose a mechanism that can activate powered down cores for performance gains while considering thermal aware floor planning and to this order they also explore topological/routing support. Some studies have proposed bypass style mechanisms for different purposes in NoCs [28]–[32]. Unlike these studies, FLOV stands from a power saving perspective with performance-aware considerations. FLOV links in a router act as a simple connector between the upstream and downstream routers, thus making them logical neighbors for credit-based flow control. A flit entering a FLOV link already has a buffer slot allocated in the downstream router and does not take risk of creating protocol deadlocks.

## III. FLOV ROUTER ARCHITECTURE

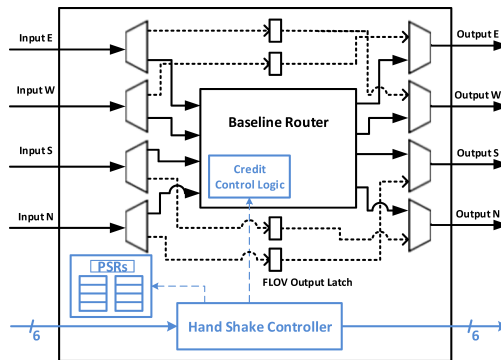


Figure 1. FLOV Router Architecture.

As shown in Figure 1, the FLOV router architecture has multiplexers and demultiplexers added to input/output links, in addition to a latch in each direction. When a FLOV router is powered-on, it functions like a baseline 3-stage virtual-channel router [33], and the muxes/demuxes are set to 0 as well as the latches are power-gated. When the router is power-gated, all the components of the baseline router are power-gated and the muxes/demuxes are set to 1 to activate the FLOV links. For the routers placed on the edges of the 2D mesh, the FLOV links are activated only in the dimension (X or Y) where there are neighbors in both directions. The Routers on the four corners of the 2D mesh do not have any FLOV links, since they can be isolated once they are power-gated. The HandShake Control logic (HSC) block is introduced, connecting to all the neighboring routers, which implements the handshake protocol between adjacent routers required before power-gating a router. Two sets of Power State Registers (PSRs) hold the power states of the immediate neighboring routers and the nearest powered-on routers (logical neighbors) in each direction, respectively. PSRs for logical neighbors are only used in the complex gFLOV power-gating mechanism described in Section IV-B. The Credit Control Logic (CCL) is modified to interact with HSC so as to always hold the buffer availability (credit) information of the nearest powered-on downstream router.

#### IV. RESTRICTED FLOV AND GENERALIZED FLOV HANDSHAKE PROTOCOLS

Using the FLOV router architecture in Section III, we propose two handshake protocols for FLOV routers: restricted FLOV (rFLOV) and generalized FLOV (gFLOV). rFLOV has a simpler protocol but its power saving is limited, while more complex gFLOV shows better power saving.

##### A. Restricted FLOV

In this scheme, when a core is powered down, its attached router waits for packets coming from the core or going to the core for a certain number of cycles. The state transition diagram in Figure 2 depicts the power states a router can be in. If there are no packets detected, the router sends a signal to its neighbors using out-of-band control lines to indicate that it is in the *Draining* state. During this state, its neighbors cannot initiate any new packet transmission to this router, while they are allowed to finish current packet deliveries.

In rFLOV, no two consecutive routers in a row/column are allowed to be powered down. Therefore, 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, while the other is back to normal (*Active* state). Hence, even though the attached core is powered-down, a router is not allowed to drain if one of its neighbors is in draining or sleeping.

A router in *Draining* checks its input buffers for any residing flits and continues to forward them to downstream

routers normally. After emptying all its input buffers and receiving *drain\_done* signals from all its neighbors, the router power-gates itself by shutting down the baseline router portion (*Sleep* state). Meanwhile, all the muxes/demuxes are switched to 1, and the router sends a signal to all neighbors so that new packet transmission can be initiated and the neighbors can update their immediate neighbor PSRs.

Once the FLOV router is power-gated, a flit coming into the router is stored in the FLOV output latch without any routing/arbitration. In the next cycle, it is delivered to a designated VC in the downstream router since the VC was already calculated in the upstream router. From the downstream router, the packet delivery becomes normal. When an FLOV 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 get the correct credit information of the downstream router.

A powered-down FLOV router wakes up when its core becomes active or its neighbor has a packet destined for its core (*Wakeup* state). When a currently sleeping FLOV router wakes up due to aforementioned conditions, it first signals its neighbors to stop new packet transmission. After finishing current packet deliveries and emptying its output latches, the FLOV router powers on the baseline router portion and switches the muxes/demuxes to 0. During *Wakeup*, the FLOV router still relays credit counts of its downstream router to its upstream router. However, once becoming *Active*, the router receives credit information from its downstream router, and its upstream router sets the corresponding credit to fully available.

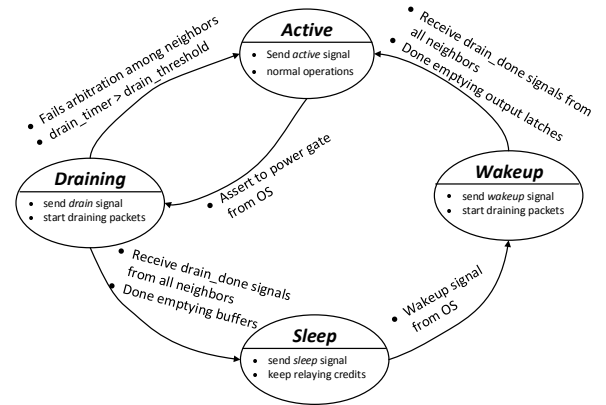


Figure 2. Router Power State Transition Diagram.

Figure 3 shows a working example of the rFLOV protocol. 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 Figure 3 (a), all three FLOV 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

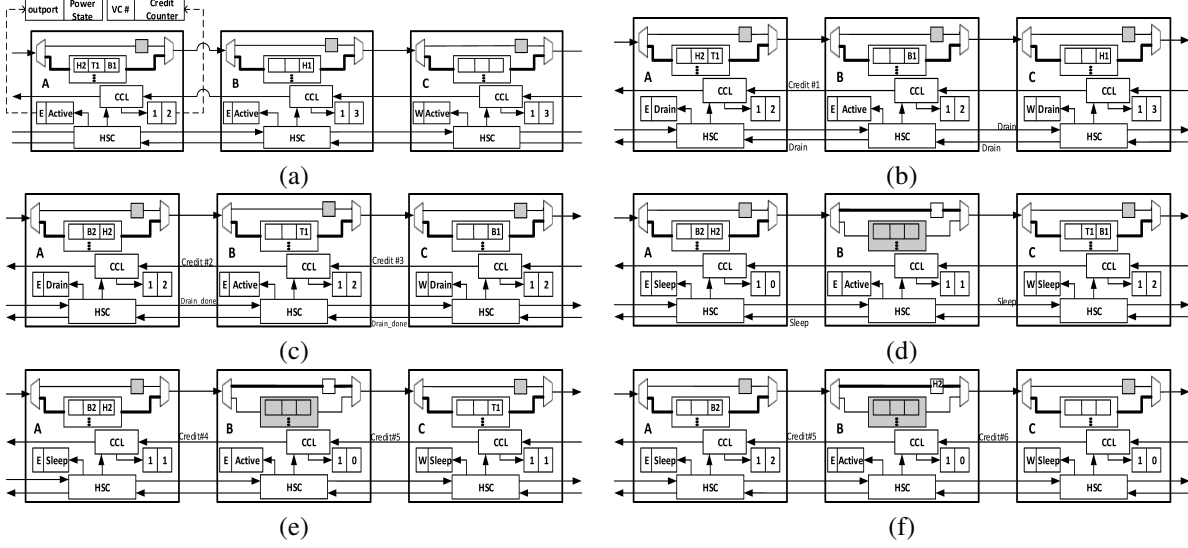


Figure 3. An Example of rFLOV in Timeline from (a) to (f).

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 here.

- In Figure 3 (b), 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 go back into the *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 updated as shown.
- In Figure 3 (c), Router A sends the *drain\_done* signal to Router B indicating that it finished 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.
- Figure 3 (d) depicts the situation after Router B finishes draining packet 1 to Router C and goes into *Sleep*. 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 also the credit counters are zeroed as shown in Router A. Note that even though Router A had a flit (H2) to send Router B, it has to wait until B finishes its power state transition.
- Figure 3 (e) shows the credit control and maintenance between Routers A and C while Router B is power-gated. After Router B goes into the *Sleep* state, Router

A zeroes its credit counter entry and the credit information is copied from Router B to A (Credit #4). This is because Router C is now logically the downstream router 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.

- In Figure 3 (f), we can see how the Credit #5 is relayed by the power-gated Router B to Router A, which then updates its credit counters. This is how Router A can keep track of the credit status of Router C via the relaying scheme in Router B.

The wakeup procedure is similar with the draining procedure, since a waking up router sends wakeup 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 destined to it 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 to fully available. Once this happens, the router switches the muxes/demuxes and resumes baseline operations.

### B. Generalized FLOV

Power saving is limited in rFLOV since, when a router goes to sleep, none of its neighbors are allowed to sleep regardless of the power states of their attached cores. In this section, we propose *generalized FLOV (gFLOV)* where two or more consecutive routers in a row/column can be power-gated simultaneously.

The main challenge of gFLOV in comparison with rFLOV is the added complexity of handshaking between routers so as to keep consistent PSRs and maintain the credit information of downstream routers. This is because, unlike in rFLOV, consecutive routers can be power-gated, the handshake signaling between two active routers (logical

neighbors) may need to cross several power-gated routers. In rFLOV, there is no need for handshake relaying because the handshaking occurs always between two immediate (physical) neighbors, whereas when a router wants to drain/wake up in gFLOV, it has to handshake with the nearest powered-on router in each direction (if there is one), which is its logical neighbor. The power-gated routers in the middle should forward the handshake signals, in addition to updating their corresponding logical and physical neighbor routers' power states in the PSRs.

The credit control is similar with rFLOV, where the power-gated router is responsible for copying its credit counters to its upstream router. Since there might be multiple consecutive power-gated routers in the middle, the credit information is relayed across these sleeping routers until it reaches a powered-on upstream router. Like rFLOV, a router that wakes up will receive credit information from its downstream router and the upstream router sets its credits to full availability.

The handshake protocol of gFLOV requires some protocol level restrictions and additional functionalities, when compared with rFLOV, which are described as follows.

- In gFLOV, after a router finishes power-gating (goes into the *Sleep* state), it should send its corresponding logical downstream neighbor's power state in each direction to its upstream router, in addition to its current power state. This is because the logical downstream router of the power-gating router will now become the logical downstream router for its upstream router. This way the logical PSRs of all the routers are kept up-to-date.
- In gFLOV, no two logical neighbor routers in the same row/column are allowed to stay in *Draining-Draining* or *Draining-Wakeup* state combinations at the same time in order to avoid protocol deadlock. Since *Wakeup* is more crucial for performance, *Draining* has lower priority if one of the handshaking routers is trying to wake up and the other trying to drain. However, for 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. Similar with rFLOV, if the handshaking routers are trying to drain at the same time, only the one with a smaller router id can proceed.
- Two routers in the same row/column can wake up at the same time in gFLOV. Unlike the *Draining-Draining* combination, two waking up routers have no dependence on each other. Any of the handshaking *Wakeup* routers should relay the `drain_done` handshake signal to the other *Wakeup* router.

## V. DYNAMIC ROUTING ALGORITHM

The FLOV NoC baseline architecture is a two dimensional mesh topology with one column or row of routers (on the

edge) which are always powered on. This is to facilitate connectivity across the topology using our routing algorithm which is explained below. In the following section, for ease of explanation, we assume that the last column of routers are *always on* (AON routers). One VC of each powered-on router is reserved for deadlock recovery, called an escape VC. The proposed routing algorithm consists of routing for packets in the regular VCs and routing for packets in the escape sub-network. A packet in a regular VC can be sent to an escape VC when required by the deadlock recovery mechanism. Note that routing computation is performed in powered-on routers, while power-gated routers only forward packets without changing the direction.

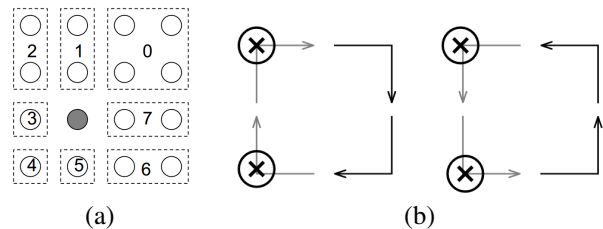


Figure 4. Destination Partitioning in a 2D Mesh Network (a), Turns Allowed/Not Allowed in the Escape Sub-Network (b).

We propose a partitioned-based dynamic routing algorithm based on YX routing for packets in regular VCs. Each router divides the network into partitions as shown in Figure 4 (a). The routing decision is made based on two variables, the partition which the destination falls into and the power states of neighboring routers. For packets with destinations in partitions 1, 3, 5, and 7, the router will send them directly to North(Y+), West(X-), South(Y-), and East(X+) downstream routers, respectively. This is because even in case of power-gated downstream routers, FLOV links will ensure the connectivity to the destinations.

For packets with destinations in partitions 0, 2, 4, and 6, the route will include a turn towards the destination. In the proposed dynamic routing algorithm, if the neighboring router in the Y direction is powered-on, the packet will be sent to this router using YX routing. If this neighboring router is power-gated, the router will check the state of the neighboring router in the X direction, and if this router is powered-on, the source router will send the packet to it.

In case both the routers in the X and Y directions are power-gated, a viable route to the destination cannot be guaranteed since the current router is not aware of the power states of the farther downstream routers. Then the packet will be forwarded to the neighbor, in the East direction, toward AON routers using the FLOV link of the neighboring power-gated router. The packet is not sent to the router in the Y direction because, in the worst case, if all the downstream routers in the Y direction are powered off, the packet will not be able to make a turn and hence cannot be routed to the destination. In contrast, once the packets are directed to the East direction, we can guarantee that the packet will be

able to make a turn toward the destination in the *AON* router of the corresponding row. Noted that a router cannot send a packet back to the direction from which it arrived so as to avoid livelock situations, where a packet keeps bouncing between two neighbors.

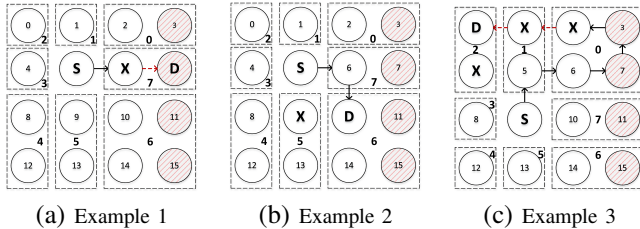


Figure 5. Routing Algorithm Examples: X indicates a power-gated router.

The proposed adaptive routing algorithm is not necessarily deadlock-free. We use Duato’s algorithm and a timeout mechanism to ensure deadlock recovery in our scheme [34]. If a packet has been waiting in a buffer for a long time, it will 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.

The routing algorithm in the escape sub-network is also based on the partitioning from Figure 4 (a). Packets with destinations in partitions 1, 3, 5, and 7, will be sent directly to North, West, South, and East, respectively. Packets whose destinations are in partitions 0, 2, 4, and 6, should be sent to East where the *AON* routers are located for the same reason mentioned above. Figure 4 (b) shows the turns that are allowed and not allowed in our escape routing algorithm which ensure deadlock freedom.

The proposed dynamic routing algorithm is explained in details using examples in Figure 5.

- In Figure 5 (a), the destination is in partition 7 of the source router’s partitions, so even though the next router is power-gated, the packet is forwarded to the East using the FLOV link.
- In Figure 5 (b), the destination is in partition 6, so the routing algorithm first checks for Router 9’s state. Since Router 9 is power-gated, the packet is sent to Router 6 that is powered-on, which will then in turn route the packet to the destination.
- In Figure 5 (c), the destination is in partition 2, so Router 5’s state is checked. Since it is powered-on, the packet is forwarded to Router 5. Router 5 then executes the same logic and since Routers 1 and 4 are both power-gated, the packet has to be sent to Router 6 so that it can at least make a turn at the *AON* router. Router 6 computes that the destination is in partition 2 and checks Routers 2 and 5. Since Router 2 is power-gated and it cannot send the packet back to Router 5, the packet is forwarded to Router 7. Router 7 then routes the packet to Router 3 where it makes another turn toward the destination. If the packet wait time in any

router exceeds the threshold, it is routed to the escape VC. Once the packet enters the escape VC, it has to remain in the escape sub-network until it reaches the destination.

#### A. Overhead Analysis

In this section we discuss the area and power overhead incurred by the proposed scheme. The modifications proposed to the router microarchitecture include 4 multiplexers and 4 demultiplexers in addition to the four output latches. The mux and demux selection signals are only toggled when the router powers on or off, 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 signals from the HSC. The additional overhead incurred due to this 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 neighbor 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). This accounts to approximately 0.1% of baseline router area according to our modeling using DSENT [9]. The HSC also includes the power state transition FSM implementation (4 states), which incurs minimal area overhead. The overall area overhead for the above components for a single router in 32nm technology is quantized at  $2.8 \times 10^{-3} \text{ mm}^2$  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 VI. The power consumption overhead for the handshaking and the credit relaying is accounted for in the DSENT model and is included in the power consumption evaluation results in the next section. None of the modifications incur significant critical path delay and do not impact the frequency of operation of the NoC.

## VI. EXPERIMENTAL EVALUATION

In this section we evaluate the FLOV mechanism by comparing static, dynamic and total power consumptions in addition to NoC latency with Router Parking [17].

#### A. Experimental Methodology

We use Booksim [20] for synthetic workload experiment, and integrate it with gem5 [21] for full system simulation. DSENT [9] is used to estimate static and dynamic power consumptions of the interconnect components with a switching activity of 50% in 32nm technology. A 2GHz clock frequency is assumed for the routers and links. Table I summarizes the simulation configuration parameters. We use both



synthetic and real workloads to evaluate the performance and power-savings of rFLOV and gFLOV against the Baseline interconnects with no router power-gating (Baseline) and Router Parking (RP). We use Uniform Random and Tornado traffic for synthetic workloads and nine benchmarks from PARSEC benchmark suite [11] for our evaluation.

Table I  
SIMULATION TESTBED PARAMETERS

Network Topology	8×8 Mesh
Input Buffer Depth	6 flits
Router	3-stage (3 cycles) router
Virtual Channel	3 regular VCs and 1 escape VC per vent, 3 vnets
Packet Size	4 flits/packet for synthetic workload
Memory Hierarchy	32KB L1 I/D \$, 8MB L2 \$ MESI, 4 MCs at 4 corners
Technology	32nm
Clock Frequency	2GHz
Link	1mm, 1 cycle, 16B width
Power-Gating Parameters	Power-Gating overhead = 17.7pJ wakeup latency = 10 cycles
Baseline Routing	YX Routing

### B. Synthetic Workload Evaluation

For synthetic workloads, we use first 10,000 cycles to warm up the simulation and run for 100,000 cycles in total. Figure 6 summarizes the simulation results using Uniform Random traffic. Similarly, Figure 7 shows the results for Tornado traffic. In the figures the top row is for the injection rate of 0.02 flits/cycle/router and the bottom row is for the injection rate of 0.08 flits/cycle/router. Each column shows average latency, dynamic, and total power consumptions for a given injection rate, respectively. Figures 8(a) and (b) break down average packet latencies of the different mechanisms into accumulated router latency (number of hops  $\times$  router pipeline latency), link latency (total link traversals), serialization latency (number of flits per packet) contention latency, and FLOV latency (number of FLOV links traversed). The static power consumption analysis for Uniform Random and Tornado traffic is shown in Figure 9.

1) *Performance*: Figure 6 (a) and Figure 7 (a) show average latency comparison of rFLOV and gFLOV with RP and Baseline. Both rFLOV and gFLOV perform better than RP across different traffic and injection rates. This is because, in RP, a packet will always need to route through powered-on routers and links connecting them, which may be non-minimal, thereby increasing the path length. In the FLOV mechanism, we take advantage of all the links, thus trying to route a packet through a minimal path using FLOV links. Even when minimal routing is not possible due to the proposed routing algorithm in Section V, the average packet latency can be reduced since the FLOV links do not incur the 3-cycle baseline router per-hop latency, since the flit is only temporarily held in the FLOV latch for one cycle. This can be observed clearly in Figure 8(a) and (b), where the accumulated router latency for RP is larger than that of the FLOV mechanism, due to non-minimal detours. In Figure 8 (a), under Uniform Random traffic, the FLOV latency increases as more cores are power-gated for

the FLOV mechanism, which shows the increased FLOV link utilization. For Tornado traffic in Figure 8 (b), 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 the number of power-gated cores increases, rFLOV power-gates as many routers as possible under the aforementioned restrictions, and gFLOV power-gates all the routers attached to the power-gated cores, whereas RP makes a dynamic decision based on maintaining network connectivity. When the fraction of power-gated cores is low, rFLOV and gFLOV perform significantly better than RP in terms of average latency due to less detour and fast FLOV links. Also average latencies of rFLOV and gFLOV are similar due to the numbers of power-gated routers being similar at lower fractions of power-gated cores. However, when the fraction of power-gated cores is high, rFLOV can only power-gate at most half the routers, while gFLOV can do more.

Figure 6 (a), at the fraction of 70% power-gated cores, shows a case where gFLOV slightly outperforms rFLOV. This is counterintuitive since lesser number of power-gated routers in rFLOV should generally incur more minimal routing paths and higher network performance. This is due to the reduced per hop latency of FLOV links showing more impact on average latency than minimal routing capability. Figure 8 (a) shows that the accumulated router latency for rFLOV is significantly larger compared to gFLOV at 70%, since gFLOV utilizes the FLOV links more. Figure 6 (a) shows that the performance of RP becomes closer to the FLOV mechanism as the fraction of power-gated cores becomes larger since the traffic injected into the network becomes very low due to lesser number of active cores. This can be also observed in Figure 8(a) and (b), where the contention latency and accumulated router latency for RP decrease as the fraction of power-gated cores goes from 60% to 80%.

Another observation is that as the injection rate increases from 0.02 to 0.08, the performance impact on RP is higher than on rFLOV and gFLOV. This is because certain routers, connecting different network partitions to ensure network connectivity, become network hotspots in RP. Such routers become congested especially at high injection rates, thus creating communication bottlenecks. The proposed dynamic routing algorithm in FLOV avoids such network hotspots.

In Figure 7 (a), rFLOV and gFLOV outperform Baseline with Tornado traffic. This is because in Tornado, a significant portion of the traffic injected from each router is destined to a router in the same row/column. Thus rFLOV and gFLOV can use FLOV links with minimal paths and avoid the 3-cycle router latency.

One interesting observation is that, under Uniform Random traffic with an injection rate of 0.08 flits/cycle/router in Figure 6 (a), RP shows similar latency as both rFLOV

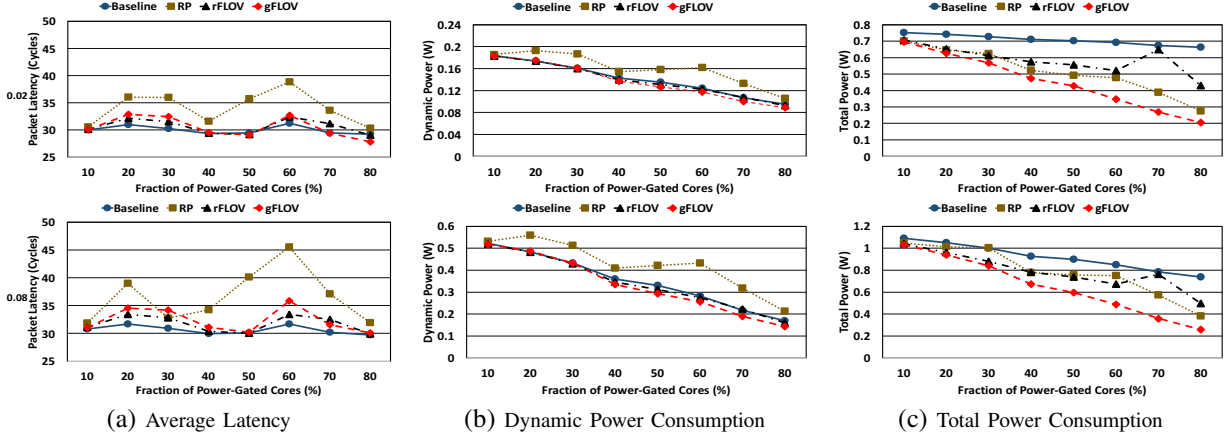


Figure 6. Average Latency and Power Comparison for Injection Rates of 0.02 (top) and 0.08 (bottom) flits/node/cycle with Uniform Random Traffic.

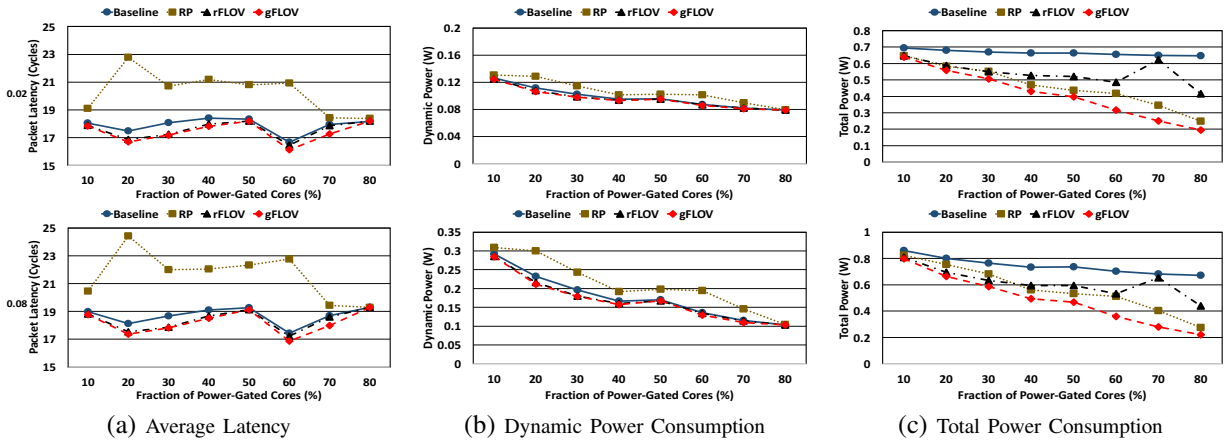


Figure 7. Average Latency and Power Comparison for Injection Rates of 0.02 (top) and 0.08 (bottom) flits/node/cycle with Tornado Traffic.

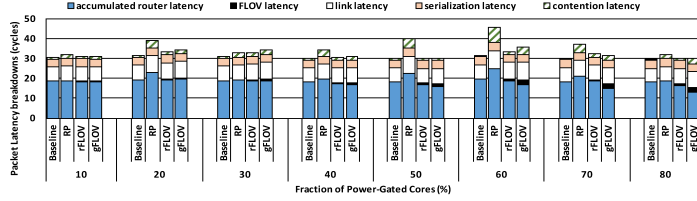
and gFLOV when 30% of cores are power-gated. This is due to the fact that RP dynamically turns on additional routers attached to power-gated cores to negate the impact of higher traffic in the network. This can also be observed from Figure 6 (c), where total power consumption is increased when the fraction of power-gated cores goes from 20% to 30%. From these results, it is clear that RP trades off static power savings for latency benefits. This is also shown in Figure 8 (a), where the router latency of RP significantly decreases as the fraction of power-gated cores goes from 20% to 30% due to RP powering on additional routers to reduce the non-minimal detour paths.

In Figure 8(a) and (b), both rFLOV and gFLOV have relatively higher contention latency at high fractions of power-gated cores. One reason is that packets have 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 round-trip latency of credit information. However, the higher utilization of FLOV links compensates for the contention latency, which can be explained by the router and FLOV latencies.

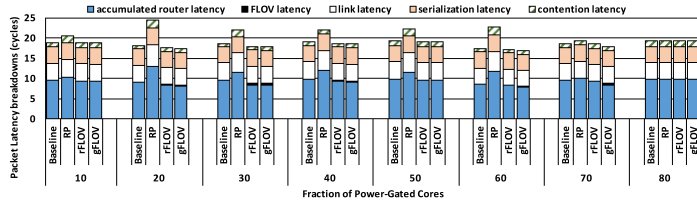
Note that RP also tends to have higher contention latency compared to the FLOV mechanism because of the high probability of hot spot creation.

2) *Power Consumption:* Figures 6 (b), 6 (c), 7 (b), and 7 (c) show dynamic and total power consumptions of the FLOV mechanism compared with RP and Baseline for multiple injection rates. In Figures 6 (b) and 7 (b), for multiple injection rates the dynamic power consumptions of rFLOV and gFLOV are lower than RP, since in RP every hop in the rerouted packet traversal requires the total router pipeline execution, whereas in 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, the FLOV mechanism consumes less dynamic power than Baseline due to avoiding the router pipeline execution. Figures 6 (c) and 7 (c) show total power consumptions of rFLOV and gFLOV compared with RP. It is clear that gFLOV unanimously has lower power consumption, since the dynamic and static power consumptions in gFLOV are lower than RP. Note that total power consumption of rFLOV is higher than RP at higher fractions of power-gated cores, mainly due to static power





(a) Uniform Random Traffic Pattern (0.08 flits/cycle/node)



(b) Tornado Traffic Pattern (0.08 flits/cycle/node)

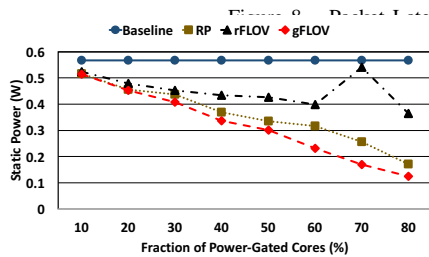


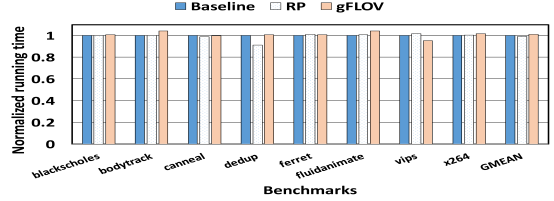
Figure 9. Static Power Comparison of FLOV with RP and Baseline.

consumption explained below.

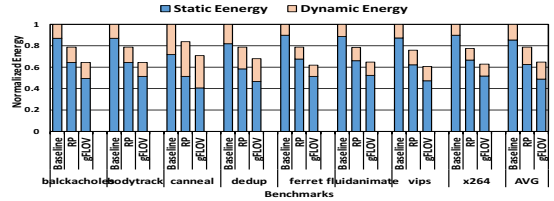
Figure 9 shows static power consumption comparison, which is injection rate and workload independent for rFLOV and gFLOV, since all routers attached to power-gated cores are power-gated in gFLOV, while rFLOV power-gates a limited number of routers to preserve the restriction. RP dynamically decides whether to conservatively or aggressively power-gate routers, using power saving versus latency tradeoff prediction based on the interconnect workload. To reduce redundancy of using the same results of the FLOV mechanism for multiple injection rates and workloads, we compare against the aggressive RP power-gating scheme that power-gates as many routers as possible, which will make the RP power results also workload independent. This allows a fair comparison with RP and lets us depict the static power evaluation in Figure 9.

In Figure 9 the static power consumption of gFLOV is lower than RP and the disparity increases as the number of power-gated cores increases. This is mainly due to the fact that gFLOV power-gates more routers than RP. rFLOV consumes more static power compared to RP, especially as the fraction of power-gated cores increases, since the number of routers that can be power-gated starts to saturate.

3) *Real Workload Evaluation*: To evaluate the behavior of the power-gating mechanisms under real workloads and show the impact on the full system environment, we run PARSEC 2.1 in gem5 [21] integrated with Booksim. The system parameters are described in Table I. Figures 8 (c) and



(c) Application execution time



(d) Full system energy consumption

(d) show the execution time and the energy consumption,

with the power-gating mechanisms compared to Baseline and RP. They show that FLOV achieves 43% reduction in static energy consumption on average compared to Baseline with only a 1% degradation in performance. This is due to the best-effort shortest-path routing and the low-latency FLOV link compensate for the detouring and round-trip credit loop latency. Compared to RP, FLOV reduces static energy by 22% as a net effect from the distributed power-gating control and the dynamic routing algorithm.

### C. Reconfiguration Overhead Analysis

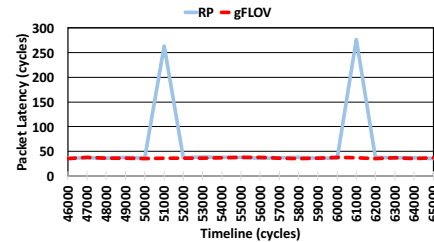


Figure 10. Reconfiguration Overhead of RP and Comparison with gFLOV.

In this section we analyze the impact of the network reconfiguration on packet latency in RP by comparing with gFLOV. Figure 10 shows average packet latency of gFLOV and RP across the timeline of execution using Uniform Random traffic with an injection rate of 0.02 flits/cycle/node 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 FM and then the corresponding routing tables have to be distributed to the routers that will be active in the next epoch (Phase I of reconfiguration protocol in RP). While this reconfiguration is performed, 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. The performance overhead due to this is shown in Figure 10, where we can

clearly observe that the newly injected packets during this time experience significant queuing delays in RP. In gFLOV, there is no such network reconfiguration 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.

## VII. CONCLUSIONS

In this paper, we proposed Fly-Over (FLOV), a lightweight distributed router power-gating mechanism for NoCs. After constructing the FLOV router enabling FLOV links by modifying the baseline router microarchitecture, we presented two different handshake protocols for FLOV routers, called rFLOV and gFLOV, and explained the dynamic routing algorithm in details. FLOV power-gates routers attached to powered-down cores without global network information, but still ensures network connectivity.

Performance evaluations using synthetic and real workloads show that FLOV not only achieves better NoC power savings due to power-gating more routers but avoids aggregated traffic rerouting in the network unlike Router Parking. Also, average latency is reduced compared with Router Parking. We show that FLOV reduces the total and static energy consumption by 18% and 22% respectively, on average across several benchmarks, compared to state-of-the-art NoC power-gating mechanism while keeping the performance degradation within 1%.

## REFERENCES

- [1] G. Moore, "Cramming More Components onto Integrated Circuits," *Electronics*, vol. 38, no. 8, p. 56, 1965.
- [2] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design of Ion-Implanted MOSFET's with Very Small Physical Dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974.
- [3] M. B. Taylor, J. Kim, J. Miller, D. Wentzlaff, 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.
- [4] 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 Wijngaert, "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.
- [5] 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.
- [6] X. Chen and L.-S. Peh, "Leakage Power Modeling and Optimization in Interconnection Networks," in *International Symposium on Low Power Electronics and Design (ISLPED)*. ACM, 2003, pp. 90–95.
- [7] A. Banerjee, R. Mullins, and S. Moore, "A Power and Energy Exploration of Network-on-Chip Architectures," in *International Symposium on Networks-on-Chip (NoCS)*. IEEE Computer Society, 2007, pp. 163–172.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] J. L. Henning, "SPEC CPU2006 Benchmark Descriptions," *ACM SIGARCH Computer Architecture News*, vol. 34, no. 4, pp. 1–17, 2006.
- [13] M. Annamaram, "A Case for Guarded Power Gating for Multi-Core Processors," in *International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2011, pp. 291–300.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [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] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] —, "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.
- [32] 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.
- [33] L.-S. Peh and W. J. Dally, "A Delay Model and Speculative Architecture for Pipelined Routers," in *International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2001, pp. 255–266.
- [34] 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.