

# An Energy-Efficient Virtual Channel Power-Gating Mechanism for On-Chip Networks

Amirhossein Mirhosseini\*, Mohammad Sadrosadati\*, Ali Fakhrzadehgan\*, Mehdi Modarressi<sup>†‡</sup> and Hamid Sarbazi-Azad\*<sup>‡</sup>

\*Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

<sup>†</sup>School of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran

<sup>‡</sup>Computer Science School, Institute for Researches in Fundamental Sciences, Tehran, Iran

Emails: {amirhosseini, sadrosadati}@ce.sharif.edu, alifakhrzadehgan@utexas.edu, modarressi@ece.ut.ac.ir, azad@{sharif.edu, ipm.ir}

**Abstract**—Power-gating is a promising method for reducing the leakage power of digital systems. In this paper, we propose a novel power-gating scheme for virtual channels in on-chip networks that uses an adaptive method to dynamically adjust the number of active VCs based on the on-chip traffic characteristics. Since virtual channels are used to provide higher throughput under high traffic loads, our method sets the number of virtual channel at each port selectively based on the workload demand, thereby do not negatively affect performance. Evaluation results show that by using this scheme, about 40% average reduction in static power consumption can be achieved with negligible performance overhead.

## I. INTRODUCTION

The dramatically diminishing feature sizes along with the limitation of increasing the working frequency of chips, have led to the design of complex Systems on Chip (SoCs) and Chip Multiprocessors (CMPs) with a large number of cores [2], [9], [18]. However, designing an efficient and scalable communication infrastructure among the cores in a CMP remains an important problem. Network-on-Chip (NoC) was proposed to overcome challenges of traditional bus-based systems and point-to-point connections [2], [9].

NoCs contribute to a significant portion of power consumption in a CMP system. Especially, the static power of NoCs, which accounts for a considerable percentage of CMP's total power consumption, is dramatically growing by the shrinking of technology size [19]. The performance of the NoC, on the other hand, plays an important role in the total performance of CMP systems as any increase in packet latency results in a decline in total program execution time [5]. Since NoC is considered to be one of the main performance bottlenecks in a CMP system, any method that attempts to reduce power consumption of the NoCs must contemplate the performance and power tradeoffs carefully in order to maintain the normal functionality and performance of the system [5], [17].

Power-gating (which is simply turning off an idle unit and turning it on back if needed) is a promising way for reducing the leakage power of digital systems that has been widely used in different areas [4], [5], [11], [13], [15], [16], [19], [20]. However, power-gating may impose a power overhead to the system. This is simply because of the power-line switching overhead which becomes more notable when the system has to be turned off and on frequently. Thus, efficient power-gating can be applied to a unit which has the following characteristics: first, it should have a low utilization in order to have a good potential of being turned off, and second, its idleness must

not be fragmented so that the power-line switching overhead remains negligible [11].

Routers in a 2D mesh NoC are very likely to be underutilized for a while and seem to be a proper candidate for power-gating. Nonetheless, recent research shows that the idleness of the routers is fragmented in a mesh network and does not provide enough time for power-gating [5]. Accordingly, a more fine-grained unit such as virtual channel (VC), which has a continuous idle period and also low utilization, would be a more appropriate candidate for power-gating method [15], [16], [20]. Since VCs play a very crucial role in satisfying the network throughput, the above-mentioned performance/power tradeoff becomes more important when electing them as the power-gating domain. The problem associated with most of the previous studies on power-gating of VCs is that is that they achieve power reduction at the price of some (often considerable) performance loss.

To efficiently address the aforementioned problem, in this paper, we propose a novel method for power-gating of VCs. This method adopts a dynamic scheme to adaptively adjust the virtual channel of a router to the current traffic load so that maximum power reduction is achieved while maintaining the performance in high traffic loads. Our proposed technique manages to gain up to 40% reduction in static power consumption compared to a baseline architecture while its performance overhead is negligible.

The rest of this paper is organized as follows: In Section II, some previous works related to VCs leakage power reduction are briefly discussed. In Section III, the proposed method is described in details. Section IV evaluates our proposal by various metrics, and finally Section V concludes the paper.

## II. RELATED WORK

Several methods have been proposed for power reduction in NoCs. Some of them have focused on dynamic power [1], [3], [14], some on static power [4], [5], [11], [13], [15], [19], [20], and some others covered both power components [16]. Since our work attempts to reduce the static power via efficient VC power-gating, in this section, we target the previous works that deal with power-gated VCs.

Matsutani et al [16] proposed a novel method for reducing the power consumption of NoCs. This method includes techniques for alleviating both dynamic and static power consumption. However, since focus of our work is on static power, we explain part of their techniques targeting static power consumption. In their work, they have focused on VCs and utilized

two different policies to turn them off/on. These policies include:

- Turn off policy: every VC that is idle for more than  $T_{idle-detect}$  (4 cycles in their method) would be detected by the router and turned off immediately.
- Turn on policy: this step is slightly more complicated. In order to turn a VC on, a request from a flit in an adjacent router is needed. To be more specific, if any available VC cannot be found at the next-hop router of the packet, current router sends a wakeup request to that router. Based on  $T_{break-even}$ , which is the minimum required time to compensate the power-gating overhead on energy, the request can be granted or denied. On the grounds that a power-gated VC needs some time to become fully activated (3 cycles in their design)- which is denoted as  $T_{wakeup}$ - in order to compensate the performance overhead, the wakeup request is sent speculatively in the routing stage of the router's pipeline.

Furthermore, Matsutani et al [15] also proposed a more fine-grained method dividing the router architecture into several power domains; namely, VC buffer, Output latches, CB-MUX, and VCMUX and activates/deactivates them in different phases. They suggested several mechanisms for early wakeup control using lookahead policies. Basically, every flit looks two hops ahead to send the wakeup request signal. These two hops help the architecture to mitigate the wakeup latency. In a recent work by Yin et al [20], authors use the VC utilization factor for both power-gating and waking up the VCs. In their work, two thresholds are defined: low and high thresholds. When VC utilization exceeds the high threshold, a VC would be turned on and when it gets under the low threshold a VC would be turned off. These thresholds should be statically tuned based on the workload.

### III. PROPOSED METHOD

#### A. Motivation

Since the NoC is one of the most important performance bottlenecks in a NoC-based CMP [5], [17], any method that attempts to reduce its power consumption should consider the performance constraints precisely. Based on our observations, the previous methods that have applied power-gating technique to VCs, e.g. [16], [15], and [20], can impose significant performance deteriorations to the whole CMP system by increasing packet latency and reducing the NoC's maximum throughput. Moreover, this performance reduction may contribute to an increase in total energy consumption which is undesirable. The metric based on which most of the previous studies have power-gated the VCs is the number of cycles a VC has been idle. This metric could not be an efficient choice because: 1) By selecting the threshold for detecting an idle VC to be high, the efficiency of the power-gating method would be reduced since we have wasted a portion of the idleness period for detecting it. Thus, previous methods have set this threshold to be low (e.g. 4 cycles in [16]). This could cause an inadvertent power-gating of the VCs and impose performance reduction by turning off a VC which was really needed by the system. 2) This metric evaluates each VC individually while different VCs in a port are usually not different from each other and

it would be much more efficient to control each port rather than each VC. To further elaborate upon this problem, two example situations could be considered. one of them is where the idleness is distributed on different VCs of a port and thus, the idleness period of each VC is not long enough and cannot be detected. The other situation is that a VC of a port is idle while the other VCs are totally busy, hence this idleness does not necessarily mean that the idle VC should be turned off. Similarly, the VC utilization parameter used in another recent work [20] has somehow a similar problem. If the period based on which the utilization has been calculated is set to be short, the extracted numbers could be arbitrary and not be able to be decided upon. On the other hand, if the period is set to be long, the changes in the traffic could not be detected and the decisions would be based on obsolete data and statistics.

The method for waking a VC up used by most of the previous works [15], [16] is to generate a wake up signal from the neighboring router when the sender has a packet to send and the receiver does not have any idle VC. This could cause a temporary stallation until the requested VC gets activated which has a negative effect on system's performance. This effect could be mitigated by some techniques like generating the wakeup requests a few cycles sooner [16] or by using lookahead methods to generate the wakeup signals from the neighbors that are two hops away [15]. Nevertheless, based on our observations, these modifications are not much able to diminish the performance overhead since the pitfalls of this method are somehow intrinsic. In these methods, flits are basically in charge of sending the requests for waking up the VCs. In other words, these methods rely on the traffic flow of the network, or movement of packets, while the traffic flow moves more fluently when more VCs are active and the VCs being power-gated, on the other hand, contribute to the stoppage of the traffic flow which has a negative effect on wake up process. This issue forms a positive feedback loop which almost stops the network from turning on further VCs. Likewise, the utilization factor which is used in another work [20] which is the normalized number of VC accesses, does also depend on the flow or traffic movement, and forms the same feedback situation which again affects the wake up process negatively.

According to previous discussions, a proper metric for power-gating and waking up the VCs should be devised which at least has the following characteristics:

- It should not rely on the number of cycles that a VC has been idle.
- It must consider all VCs that are available in a port together (not look at each VC individually).
- It should use current parameters of the network and not rely on obsolete network status that does not reflect recent traffic changes.
- It should be independent of the traffic flow of the network.

A metric which has the aforementioned features will be introduced in the next section.

#### B. The Method

In this section, we propose a novel method for power-gating the VCs. To describe our method, we first introduce a new

metric for power-gating and waking up the VCs which does not have the drawbacks mentioned in the previous section. This metric is the ratio of the number of wins to the number of loses of VC allocation requests from the upstream router for each port. Obviously, this metric has nothing to do with the idle cycles of a VC and also it considers a port rather than a VC. Furthermore, this metric is not affected by the traffic flow since any decline in the traffic flow will cause a decrease in both numbers of wins and loses for the allocation requests, and hence their ratio can still function as an efficient metric. To use this metric for power-gating and waking up the VCs, we allocate two counters to each output port for holding the number of wins and loses for VC allocation (VCA) requests at that port. For evaluating the ratio value we use a comparator which compares the number of loses (multiplied by the assigned threshold value) and the number of wins. Moreover, we always set our threshold values to a power of 2 so that the multiplication can be implemented as a shift operation. Our simulation experiments show that the suggested threshold values can be 8 and 32 for turning on and turning off the VCs, respectively. This means that when the ratio goes over 32, a VC can be turned off and when the ratio falls under 8 a VC should be turned on. The request for turning off or turning on a VC is delivered from the current router to the downstream router. In case of turning off, the downstream router selects an idle VC which is not already allocated and power-gates it. Similarly, in case of turning on, the downstream router selects a VC which has been power-gated for more than  $T_{break-even}$  and wakes that VC up. In order to maintain a correct ratio value, the following points are applied:

- The overflow indicator of each counter is connected to the reset signal of the other one.
- Both the number of wins and number of loses would be reset after each change to the state of a port (i.e. when turning off or turning on a VC).
- When a change happens to the state of a port the results would be evaluated at least after 100 cycles.

Power-gating of all VCs in an input port does not affect the functionality and connectivity of the network since it is possible to wake them up whenever needed. However, our method does not generally allow the last operating VC of each port to be power-gated on the grounds that if an input port is totally power-gated, waking it up might be too costly and impose a huge performance overhead. Our method only permits power-gating of the last operating VC in a port providing that the port is not used by the routing algorithm. This could be detected when there is no VC allocation request for a port during a long period of time (e.g. 1000 cycles in our design). Moreover, some protocols require more than one VC in each input port or even the application may assign some special VCs to some particular data streams. In case more than one VC per port or some particular VCs that have been power gated are required for routing a VC remaining table defines virtualized VCs would solve the problem [8].

Since load of routers that are located in various areas of the network might be different from one another, our method attempts to treat them differently. In order to distinguish between the different routers we use the link utilization metric. Our observations reveal that in most synthetic and real traffic

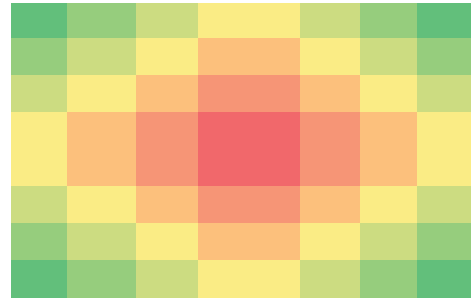


Fig. 1: Link utilization for 2D mesh topology under uniform traffic pattern.

workloads the distribution of link utilization parameter follows a specific pattern. In this pattern, link utilization in the regions close to the middle of the network is very high while the link utilization in the areas alongside the network edges is much lower. Figure 1 shows, as an example, the distribution of link utilization parameter in an  $8 \times 8$  mesh network under uniform traffic. Based on the mentioned distribution pattern, we categorize the routers into Hot, Warm, and Cold groups as depicted in Figure 2. In order to conduct different router treatment, our method attempts to power-gate the hot routers more conservatively by setting the threshold values for this routers as 16 and 64 (for turning on and turning off, respectively) which is twice the base numbers used for warm routers. Similarly, for cold routers the threshold values are chosen to be 4 and 16 in order for early power-gating.

As explained in the previous paragraph, our method uses a base pattern to distinguish among different routers and categorizes them into hot, warm, and cold groups. However, this categorization is meaningful only when the traffic load is high. This is due to the fact that in low traffic loads, all routers function as cold but they might be considered as hot or warm in the base pattern. Moreover, there are some traffics such as hotspot that might not match the base pattern; such a mismatch can impose power and performance overheads. To address this problem efficiently, we devise a dynamic method which is able to adaptively change the state of a router from cold, warm, and hot to another state when the traffic does not fit the base pattern. This method uses two counters for each port and functions based on the following rules:

- For VCs that have been idle for more than  $T_{break-even}$  consecutive cycles, *counter1* is incremented. This means that this VC could have been power-gated during this period.
- If a router has received a wake up signal for one of its ports and all power-gated VCs in that port are being turned off for less than  $T_{break-even}$ , *counter2* would be incremented. This means that power-gating has been done inefficiently in that particular port.
- If *counter1* has exceeded 31, both counters would be reset and router would be set to a colder state.
- If *counter2* has exceeded 7, both counters would be reset and router would be set to a hotter state.

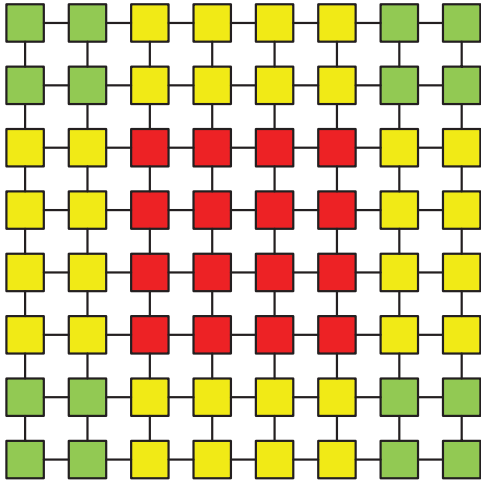


Fig. 2: cold, warm, and hot routers distribution in 2-D mesh NoC, green color depicts cold, yellow color depicts warm, and red color depicts hot routers.

#### IV. EVALUATION

In what follows, in Section IV-A, simulation environment is described which includes network configuration, traffic pattern, and comparison metrics. In Sections IV-B, IV-C, and IV-D simulation results are reported to evaluate power, performance, and overheads of our method. Finally, in Section IV-E the scalability of the proposed method is evaluated.

##### A. Simulation Environment

We used Booksim 2.0 [12] to evaluate the performance (latency and throughput) of our design. 2D mesh was used as the topology of the evaluated NoCs; mesh is a desirable topology due to its regularity and suitability for manufacturing process with respect to other well-known topologies [7]. Other network parameters in our implementation are provided in Table I. As mentioned earlier,  $T_{break-even}$  is the number of cycles required to compensate the overhead of turning a new VC on. It is very crucial to adjust this value properly; otherwise, the power-gating method increases the dynamic power consumption. We set this parameter to 15 cycles according to [11] and calculated the static and dynamic power of VCs in 45nm technology size when the clock frequency is set to 1.2 GHz. Based on [15], the wakeup latency ( $T_{wakeup}$ ) was set to 4 cycles. To calculate the static power and evaluate the dynamic power and area overheads more accurately, we implemented a typical VC and added units in HSPICE simulation environment at transistor level using 45nm technology size.

A NoC based on 2-cycle virtual channel router [6] is considered as the baseline network architecture. Additionally, the method introduced by Matsutani et al. [16], known as SSSVC, was also used for comparison since the paper focuses exactly on power-gating of VCs while others [15], [20] had used a combination of techniques and had additional approaches. We compared the efficiency of our method in terms of power reduction and performance with the baseline architecture and SSSVC. Comparison metrics include static

TABLE I: Network Simulation Parameters

Parameter	for Synthetic	for Real Workload
Network Size	$8 \times 8$ or $10 \times 10$ or $12 \times 12$	$8 \times 8$
Number of VC per Input Port	4	2
Routing Function	XY	
Flow Control	Wormhole	
Allocation Policy	Separable Input First [6]	
VC Length (flits)	4	2
Packet Length (flits)	4	...
Flit Length(bits)	128	
$T_{break-even}$ (cycles)	15	
$T_{wakeup}$ (cycles)	4	

power and energy consumptions, average network latency, and maximum network throughput. To evaluate overheads, we calculated both area and dynamic power and compared them with the baseline architecture.

We performed our simulations using both synthetic traffic patterns and real workloads. The synthetic traffic patterns include *bit complement*, *bit reversal*, *shuffle*, *tornado*, *transpose*, and *uniform*, and the Netrace traces that model the Parsec suite traffic on an  $8 \times 8$  network [10].

##### B. Power Analysis

Leakage power reduction under various synthetic traffic patterns is reported in Figure 3 (leakage power overheads, discussed in Section IV-D, are also included in the results). Clearly, our method has successfully managed to use enough VCs for handling the traffic load efficiently under different traffic patterns. Our observations show that the number of operating VCs is usually close to the number of VCs required by the network to deliver generated packets efficiently. Thus, network is adapted to the traffic load in such a way that, in each traffic rate, the maximum possible VCs are being power-gated while the performance is maintained.

According to Figure 3, our improvement on leakage power, in comparison with the baseline architecture is between 15% to 40% from heavy to light traffic loads. In comparison with SSSVC method, in low traffic region, our method saves more static power while in high traffic region it consumes more static power. However, this is not a failure as performance deterioration of the SSSVC method in high traffic region has been diminished. In other words, in high traffic loads, our method only activate those VCs that are needed to maintain the NoC performance level.

##### C. Performance Analysis

To evaluate the impact of our method on performance, the maximum network throughput under different synthetic traffic loads and average network latency for Netrace workload are reported in Figure 4. From the maximum network throughput perspective, our method has a negligible effect on performance (0.3% decline on average) while SSSVC method causes up to 14.2% reduction in maximum network throughput. The evaluation based on average network latency under real workloads shows that our method increases network latency by 1.3% while the SSSVC method imposes 25.5% network latency overhead in comparison with the baseline architecture.

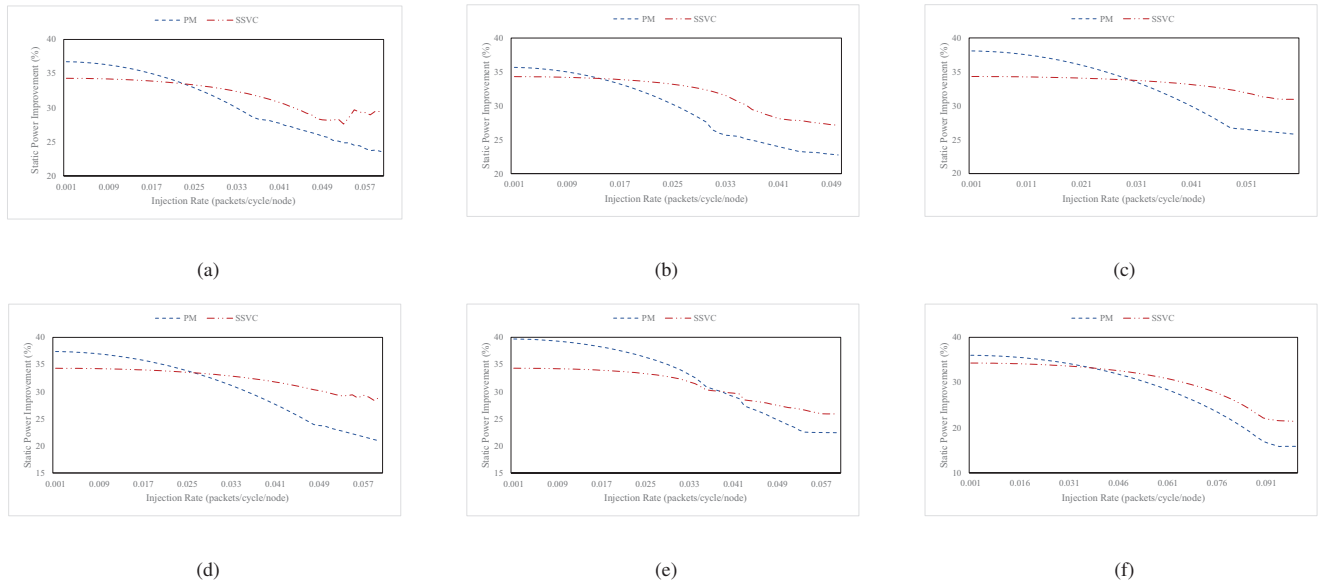


Fig. 3: Leakage Power Consumption for Network on Chip using PM (proposed method), SSVC, and baseline, (a) bit complement, (b) bit reversal, (c) shuffle, (d) tornado, (e) transpose, and (f) uniform.

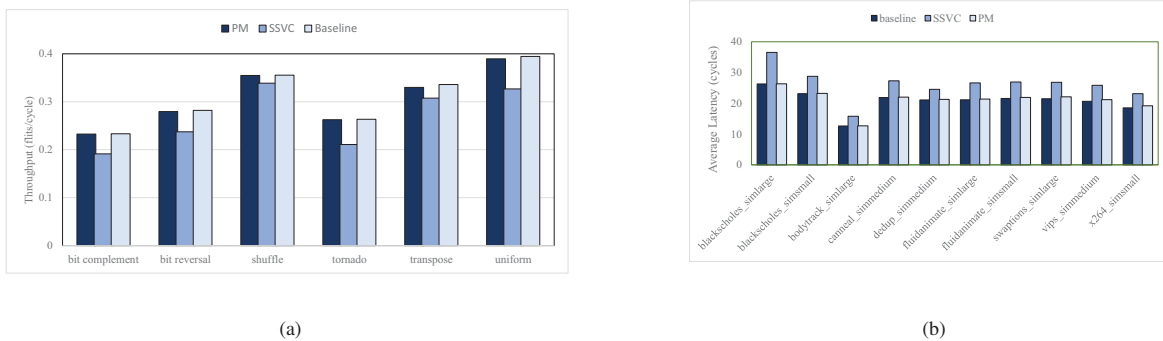


Fig. 4: Network performance analysis using PM (proposed method), SSVC, and baseline, (a) Throughput for different synthetic traffic patterns, (b) Average latency for real workloads.

Furthermore, in order to investigate the tradeoff between leakage power reduction and performance overhead, the static energy factor or the leakage power delay product, is reported in Figure 5, which is normalized to the baseline system. As shown in this figure, our method overcomes the SSVC method in terms of static energy, for all workloads.

#### D. Area overhead and Power Consumption

Since VC power-gating requires using some switches between VDD and each VC, the VCs used in our design consume more static and dynamic power compared to normal VCs without power-gating capability. These values are 5.8% and 3.6% for static and dynamic powers, respectively. Moreover, the added units in our design to implement our power reduction method also impose an overhead on dynamic power which

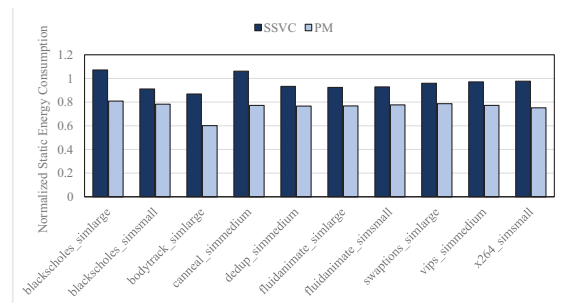


Fig. 5: Normalized static energy consumption for PM and SSVC under real workloads.

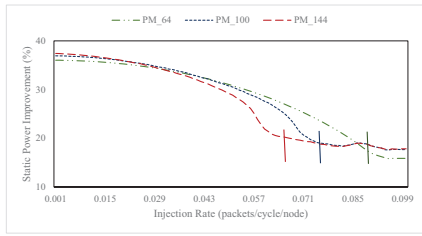


Fig. 6: Static power improvement for different network sizes under uniform traffic pattern (Vertical lines show the saturation points).

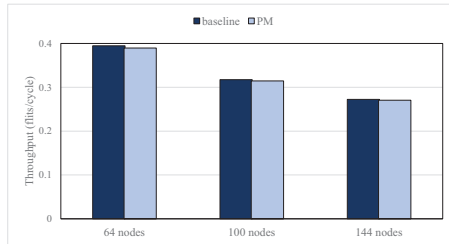


Fig. 7: Throughput for different network sizes under uniform traffic pattern.

is 2.4%. The modified VCs and added units impose an area overhead of 3.1% and 0.9%, respectively.

### E. Scalability

In order to discuss the scalability of our method, the leakage power improvement and maximum network throughput, compared to the baseline architecture, are reported in Figures 6 and 7 respectively, for different network sizes. As shown in Figure 6, the static power reduction is independent of the network size and for all network sizes it behaves similarly before saturation region. In each size, around saturation point and beyond, the static power improvement deteriorates in order to maintain performance. Accordingly, as shown in Figure 7, the performance decline is negligible for all network sizes.

## V. CONCLUSION

In deep nanometer technology, static power is considerably high. Power-gating as a promising solution for leakage power reduction is gaining more eminence in large chip-multiprocessors that employ on-chip networks for inter-core communications. Since virtual channels in on-chip networks consume much leakage power and they are not fully utilized during chip activity most of the time, in this paper, we focused on them and proposed a novel method for turning off/on the VCs on demand. An advantage of this method was its adaptivity which allowed it to be dynamically adjusted to the traffic load. Our evaluation showed that the proposed method could reduce the leakage power considerably with a negligible effect on performance. We reported the static power reduction, the maximum network throughput, and average network latency in comparison with a baseline architecture; totally, the proposed method achieved up to 40% leakage power saving while the average performance overhead was 0.3%.

## REFERENCES

- [1] e. a. Ansari, Amin, "Tangle: Route-oriented dynamic voltage minimization for variation-afflicted, energy-efficient on-chip networks," *HPCA2014*, 2014.
- [2] L. Benini and G. De Micheli, "Networks on chips: a new soc paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, Jan 2002.
- [3] C.-H. O. Chen, S. Park, T. Krishna, and L.-S. Peh, "A low-swing crossbar and link generator for low-power networks-on-chip," in *Proceedings of the International Conference on Computer-Aided Design*, ser. ICCAD '11, 2011, pp. 779–786.
- [4] L. Chen and T. M. Pinkston, "Nord: Node-router decoupling for effective power-gating of on-chip routers," in *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-45, 2012, pp. 270–281.
- [5] L. Chen, L. Zhao, R. Wang, and T. M. Pinkston, "Mp3: Minimizing performance penalty for power-gating of cros network-on-chip," 2014.
- [6] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers Inc., 2003.
- [7] W. Dally, "Performance analysis of k-ary n-cube interconnection networks," *Computers, IEEE Transactions on*, vol. 39, no. 6, pp. 775–785, Jun 1990.
- [8] M. Evripidou, C. Nicopoulos, V. Soteriou, and J. Kim, "Virtualizing virtual channels for increased network-on-chip robustness and upgradeability," in *VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on*. IEEE, 2012, pp. 21–26.
- [9] P. Guerrier and A. Greiner, "A generic architecture for on-chip packet-switched interconnections," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '00, 2000, pp. 250–256.
- [10] J. Hestness, B. Grot, and S. W. Keckler, "Netrace: dependency-driven trace-based network-on-chip simulation," in *Proceedings of the Third International Workshop on Network on Chip Architectures*. ACM, 2010, pp. 31–36.
- [11] Z. Hu, A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose, "Microarchitectural techniques for power gating of execution units," in *Proceedings of the 2004 International Symposium on Low Power Electronics and Design*, ser. ISLPED '04, 2004, pp. 32–37.
- [12] N. Jiang, D. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. Shaw, J. Kim, and W. Dally, "A detailed and flexible cycle-accurate network-on-chip simulator," in *ISPASS2013*, April 2013, pp. 86–96.
- [13] G. Kim, J. Kim, and S. Yoo, "Flexibuffer: Reducing leakage power in on-chip network routers," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, June 2011, pp. 936–941.
- [14] H. Matsutani, Y. Hirata, M. Koibuchi, K. Usami, H. Nakamura, and H. Amano, "A multi-vdd dynamic variable-pipeline on-chip router for cmps," in *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, Jan 2012, pp. 407–412.
- [15] 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 *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*, May 2010, pp. 61–68.
- [16] H. Matsutani, M. Koibuchi, D. Wang, and H. Amano, "Adding slow-silent virtual channels for low-power on-chip networks," in *Proceedings of the Second ACM/IEEE International Symposium on Networks-on-Chip*, ser. NOCS '08. IEEE Computer Society, 2008, pp. 23–32.
- [17] A. Mishra, R. Das, S. Eachempati, R. Iyer, N. Vijaykrishnan, and C. Das, "A case for dynamic frequency tuning in on-chip networks," in *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*, Dec 2009, pp. 292–303.
- [18] J. D. Owens, W. J. Dally, R. Ho, D. Jayasimha, S. W. Keckler, and L.-S. Peh, "Research challenges for on-chip interconnection networks," *IEEE micro*, vol. 27, no. 5, p. 96, 2007.
- [19] A. Samih, R. Wang, A. Krishna, C. Maciocco, C. Tai, and Y. Solihin, "Energy-efficient interconnect via router parking," in *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, Feb 2013, pp. 508–519.
- [20] J. Yin, P. Zhou, S. S. Sapatnekar, and A. Zhai, "Energy-efficient time-division multiplexed hybrid-switched noc for heterogeneous multicore systems," 2014.