Performance Degradation by Deactivated Cores in 2-D Mesh NoCs

Ikki Fujiwara* and Michihiro Koibuchi*†
*National Institute of Informatics
†The Graduate University for Advanced Studies (SOKENDAI)
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan 101-8430
Email: {ikki, koibuchi}@nii.ac.jp

Hiroki Matsutani
Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, Japan 223-8522
Email: matutani@arc.ics.keio.ac.jp

***Abstract*—Chip MultiProcessors (CMPs) will have darksilicon or frequently deactivated areas in a chip, as technology continues to scale down, due to power dissipation. In this work we estimate the influences of deactivated cores on performance of network-on-chips (NoCs). Even when a chip has a two-dimensional mesh topology, a deactivated core that includes an on-chip router makes topology irregular. We thus assume that a topology-agnostic deadlock-free routing is used with a moderate number of virtual channels in such CMPs. Thorough cycle-accurate network simulations of a 2-D mesh NoC, we found that (1) indeed a deactivated core degrades the performance to some extent in terms of throughput, but (2) latency is not increased or even reduced when a deactivated core is located in the corner of a mesh. Hence, we recommend choosing a corner core for deactivation to maintain the performance of NoCs.**

***Keywords*—Network-on-Chip, systems on chip, darksilicon.**

I. Introduction

Recently, Network-on-Chips (NoCs) have been used in chip multi-processors (CMPs) to connect a number of processors and cache memories on a single chip, instead of traditional bus-based on-chip interconnects that suffer the poor scalability. Figure 1 illustrates an example of CMP inspired by [1], in which eight processors (or CPUs) and 64 L2 cache banks are interconnected by sixteen on-chip routers. These cache banks are shared by all processors and thus a cache coherence protocol is running on the CMP. Each core includes an on-chip router.

NoCs can be evaluated from various aspects, such as the throughput, communication latency, hardware amount, and power consumption, but especially those used in CMPs are required to reduce the communication latency and power consumption. The communication latency is the primary performance factor, since it directly increases the cache access latency that affects the application performance on CMPs. As for the cost factor, the power consumption is becoming more and more important in almost all systems, since it affects the heat dissipation, packaging, and running costs of the system. The above is well discussed in previous works, such as [2].

The power consumption is classified into dynamic switching power and static leakage power. The switching power is consumed only when packets are transferred on a NoC, while the leakage power (or static power) is consumed without any packet transfers as long as the NoC is powered on.

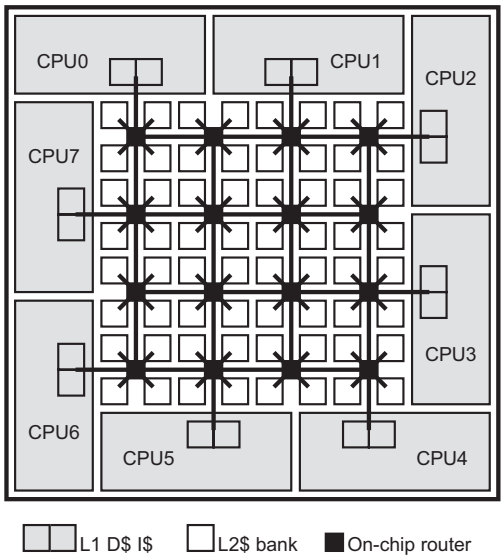


Fig. 1. Example of an 8-core CMP.

CMPs will have darksilicon or frequently deactivated areas in a chip, as technology continues to scale down, due to power dissipation [3]. Even when a chip has a two-dimensional mesh topology, a deactivated core that includes an on-chip router makes topology irregular. In most cases, custom deadlock-free routing for such a regular topology cannot be used. However, a topology-agnostic routing, such as up*/down* routing, usually reduces the throughput.

In this work we estimate the influences of deactivated cores on the performance of deadlock-free routing in network-on-chips (NoCs) through cycle-accurate network simulations. When a topology-agnostic deadlock-free routing is used [4], its routing performance is significantly degraded due to congested paths that avoid a deactivated core. By contrast, if it is located at the corner of the 2-D mesh, the performance is gracefully degraded.

The rest of this paper is organized as follows. Section II surveys deadlock-free routing, and power gating as a technique to activate and deactivate cores in a chip. Section III evaluates influences of deactivated cores on NoC performance. Finally, Section IV concludes this paper.

978-0-7695-5086-2/13 $26.00 © 2013 IEEE
DOI 10.1109/MCSoC.2013.13

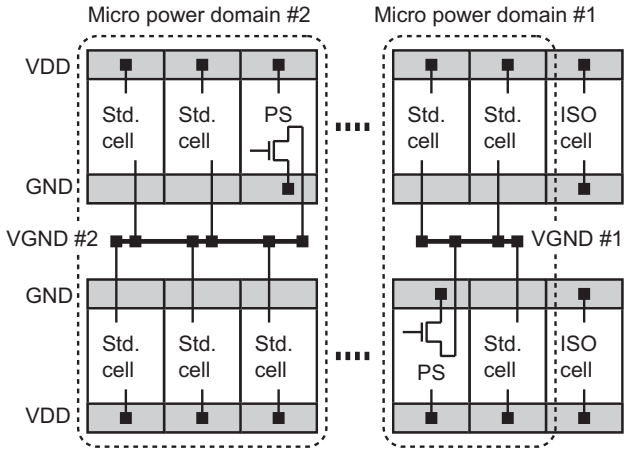


Fig. 2. Concept of the fine-grained power gating. PS and ISO refer to a power switch and an isolation cell, respectively [2].

II. Related Work

A. Deadlock-free Routing

Although there are a large number of deadlock-free routings, we focus on typical routings for 2-D mesh and irregular topologies.

1) Dimension-Order Routing: When using k-ary n-cube topology that includes 2-D mesh, a simple routing algorithm is dimension-order routing (DOR). It transfers packets along minimal path in the visiting policy of low dimension on first. For example, dimension-order routing uses y-dimension channels after using x-dimension channels in 2-D meshes. Dimension-order routing uniformly distributes minimal paths between all pairs of nodes.

2) Up/Down* Routing:* When cores that include on-chip routers are deactivated, topology becomes irregular. A topology-agnostic routing is usually needed for performing paths in such irregular topologies. A popular topology-agnostic routing is up*/down* routing. Up*/down* routing [5] avoids deadlocks in irregular topologies using neither virtual channels nor buffers. Up*/down* routing is based on the assignment of direction to network channels [5]. As the basis of the assignment, a spanning tree whose nodes correspond to routers in the network is built. The "up" end of each channel is then defined as follows: (1) the end whose node is closer to the root in the spanning tree; (2) the end whose node has the lower unique identifier (UID), if both ends are on nodes at the same tree level. A legal path must traverse zero or more channels in the up direction followed by zero or more channels in the down direction, and this rule guarantees deadlock-freedom while still allowing all hosts to be reached. However, an up*/down* routing algorithm tends to make imbalanced paths because it employs a one-dimensional directed graph. It has been reported that the dimension-order routing has higher throughput than up*/down* routing when attempting to k-ary n-cubes [6]. Although a work attempts to consider regularity of 2-D mesh that has a small fraction of deactivated cores to map the graph of routing, its procedure is not simple [7].

3) Subnetwork-based Routing: Up*/down* routing uses a number of non-minimal imbalanced paths so as not to create cycles among physical channels.

To reduce non-minimal imbalanced paths, the network is divided into layers of sub-networks with the same topology using virtual channels. A number of paths across multiple sub-networks are established to shorten the path hops. Enough restrictions on routing in each sub-network are applied to satisfy deadlock freedom by using an existing routing algorithm, such as up*/down* routing, as long as every packet is routed inside the sub-network. To avoid deadlocks across sub-networks, the packet transfer to a higher-numbered sub-network is prohibited. The subnetwork-based routing thus has shorter paths than those of up*/down* routing in most cases.

The above concept has been utilized in various subnetwork-based routing techniques [8] [9], and a similar concept is used [10]. Their comparison is well-discussed in [4].

B. Power Gating Techniques

To deactivate the cores, power gating technique has been widely attempted. Power gating is a typical leakage-power reduction technique. It shuts off the power supply of idle circuit blocks by turning off (or on) the power switches which are inserted between the GND line and the blocks or between the VDD line and the blocks. This concept has been applied to circuit blocks with various granularities [2]. Depending on the granularity of target circuit blocks (i.e., power domains), the power gating is classified into coarse-grained and fine-grained approaches. We briefly explain both as follows.

1) Coarse-Grained Power Gating: Each target circuit block is surrounded by a power/ground ring. Power switches are inserted between the core ring and power/ground IO cells. The power supply to the circuit block can be controlled by the power switches. Since the power supply to all cells inside the core ring is controlled at one time, this approach is well suited to the IP- or module-level power management. The coarse-grained approach has been popularly used, since its IP- or module-level power management is straightforward and easy to control. However, it typically imposes a microsecond order wakeup latency.

2) Fine-Grained Power Gating: This approach has received a lot of attention in recent years because of its flexibility and short wakeup latency [11] [12]. Although various types of fine-grained power gating techniques have been proposed, we focus on the method proposed in [12]. In this method, customized standard cells, each of which has a virtual ground (VGND) port by expanding the original cell, are used. These standard cells that share the same active signal form a single micro power domain, by connecting their VGND ports to a shared local VGND line, as shown in Figure 2. Power switches are inserted between the VGND line and GND line to control the power supply to the micro power domain. Figure 2 illustrates two micro power domains, each of which has its own local VGND line and power switch.

In this work we assume that several cores are deactivated due to severe upper bound of the power consumption in a chip using the power gating. Thus, to avoid such deactivated cores we will use up*/down* routing as a default in the evaluation.

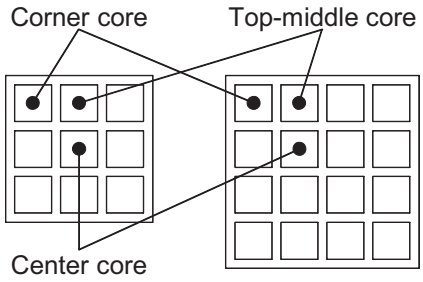


Fig. 3. Patterns of deactivated cores.

III. Evaluation

A. Cycle-accurate Network Simulation

The NoCs that have deactivated cores were evaluated in terms of the network latency and throughput using a flit-level network simulator written in C++. We assume 2-D mesh topology on a chip. Every router thus has three, four or five ports, and a single processing element (PE) connected to every router. A "core" thus consists of PE and its local router. Wormhole switching was used as the switching technique of the router. Three clock cycles are required for a flit to pass through a router, i.e. one clock for routing, one for transferring the flit from input channel to output channel through a crossbar, and one for transferring the flit to the next node. The PEs inject packets independently of each other. We set the packet length at 8 flits, including one header flit. Three traffic patterns are used assuming that 3- or 4-bit coordinates are assigned to each core.

- *Uniform traffic*: All destination nodes are selected randomly, and so the traffic is distributed uniformly.
- *Bit reversal traffic*: A node with the identifier $(a_0, a_1, \ldots, a_{n-1})$ sends a packet to the node whose identifier is the bit reversal $(a_{n-1}, \ldots, a_1, a_0)$ of the source node.
- *Matrix transpose traffic*: A node (x, y) sends a packet to the node $(k-y-1, k-x-1)$ where k is the number of nodes in each dimension, or $(k-x-1, k-y-1)$ when $x+y = k-1$.

To take minimal paths, we use up*/down* routing with escape paths for all the cases except that no deactivated cores are used. Only the case for no deactivated cores uses custom deadlock-free routing, i.e. dimension-order routing in 2-D mesh.

The network size is set to 9 cores or 16 cores. For ease of understanding, a single core is deactivated in each evaluation. The coordinates of the deactivated core influence the path sets of a topology-agnostic routing that affect the network performance. Three patterns of coordinates of a deactivated core are thus picked up and evaluated as shown in Figure 3. They are compared to 2-D mesh that fully activates all the cores.

B. Throughput and Latency

Figures 4 and 5 show the accepted traffic vs. network latency for 2-D mesh of 9 cores and 16 cores, respectively. The throughput is the maximum amount of accepted traffic [13]. "No deact. cores" refers to the baseline case in which all the cores are activated.

With uniform traffic, the deactivated *corner* core decreases the throughput by 25.5% and 19.0% in 9- and 16-core networks, respectively, while the latency is even reduced by 1.0% and 2.1% on average in 9- and 16-core networks, respectively, when compared to the baseline. The deactivated *center* core decreases the throughput by around 24.6% in both 9- and 16-core networks, as well as increasing the latency by 6.9% and 2.2% on average in 9- and 16-core networks, respectively. The deactivated *top-middle* core leads to an intermediate performance between the baseline and the case of deactivating the *center* core.

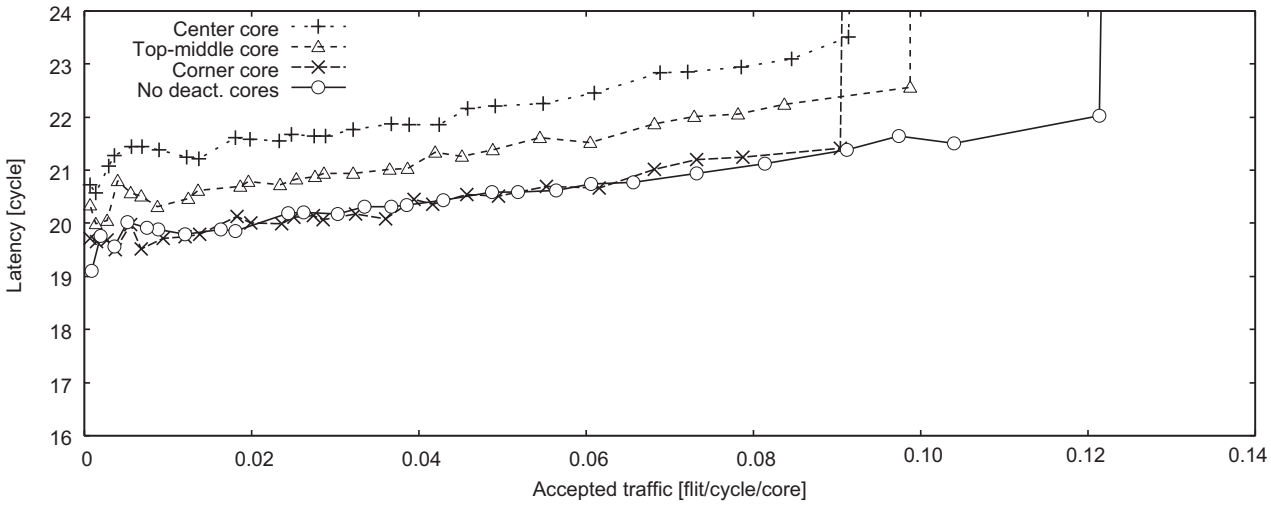
With matrix transpose traffic, the advantage of deactivating a *corner* core rather than a *center* or a *top-middle* core becomes clearer. The deactivated *corner* core reduces the latency by 5.4% and 6.9% on average while decreasing the throughput by 31.4% and 5.5% in 9- and 16-core networks, respectively. The deactivated *center* core further decreases the throughput by 45.4% in 9-core network while increasing the latency by 3.9% in 16-core network. The deactivated *top-middle* core leads to a similar throughput to the case of deactivating a *corner* core and the similar latency to the baseline in 9-core network, while slightly improving the throughput (by 4.3%) and the latency (by 0.9%) in 16-core network.

With bit reversal traffic, the observations are almost the same as those with other two traffic patterns. The deactivated *corner* core reduces the latency by 11.3% and 5.3% on average while it decreases the throughput by 31.3% in 9-core network and even improves the throughput by 2.9% in 16-core network. The deactivated *center* core decreases the throughput by 24.9% and 12.4% in 9- and 16-core networks, respectively, as well as increasing the latency by around 4.3% on average in both 9- and 16-core networks. The deactivated *top-middle* core leads to almost an intermediate performance between the baseline and the case of deactivating the *center* core.

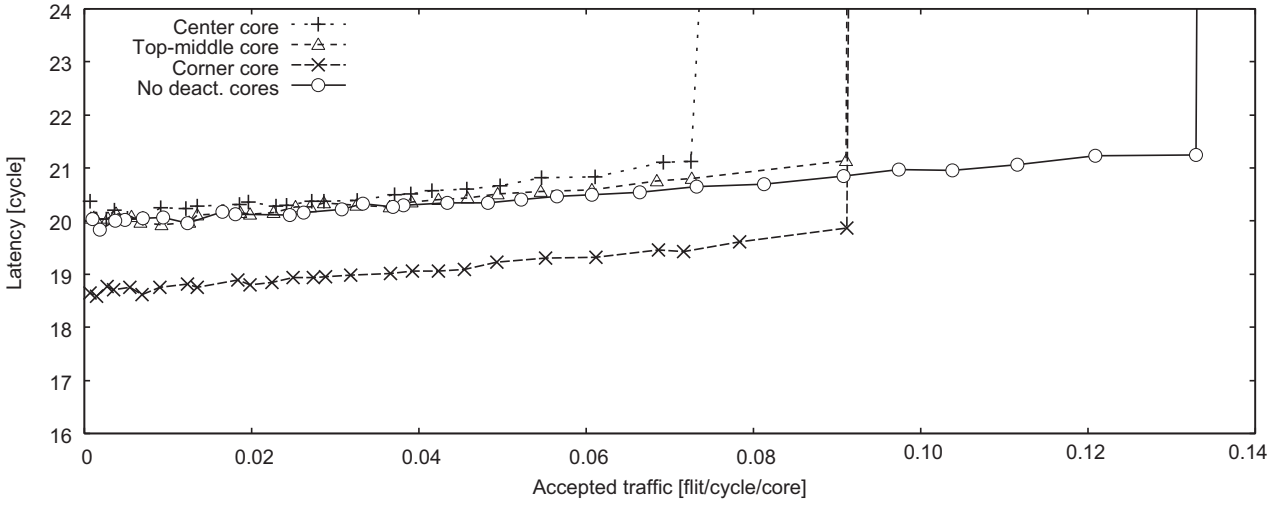
In an overall conclusion, the location of the deactivated core strongly affects the network performance, and a *corner* core should clearly be the first choice of deactivation. The reason is that a deactivated *corner* core does not make the shortest paths between any pair of cores longer, whereas a deactivated *center* or *top-middle* core may cause a detour between cores located on the opposite sides of it.

IV. Conclusions

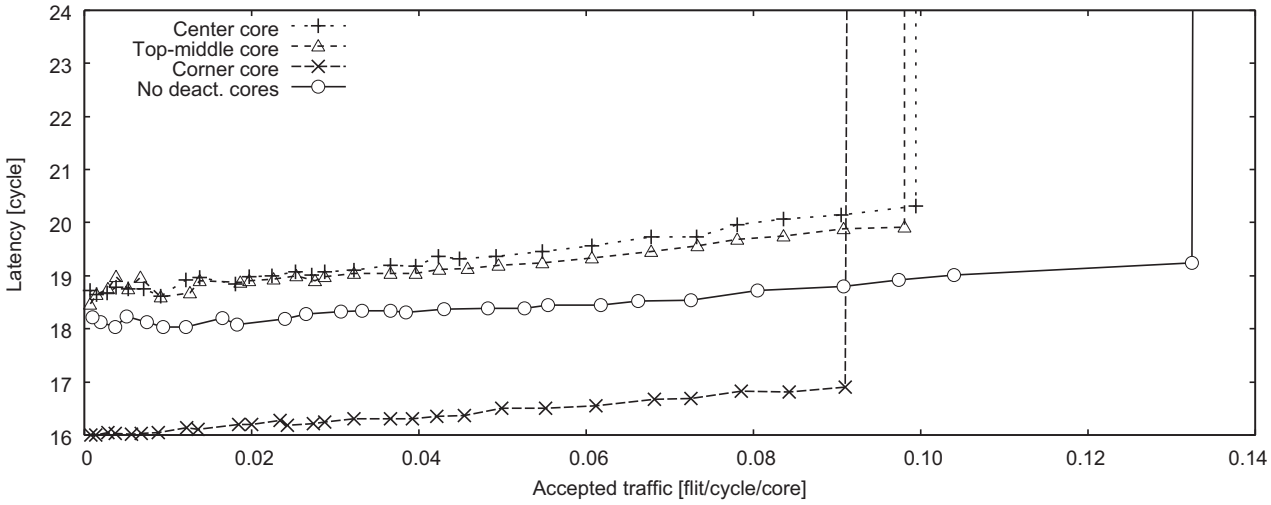
In this work we estimated the influences of deactivated cores on the performance of the network-on-chips (NoCs). Thorough cycle-accurate network simulations, it is clarified that the location of the deactivated core strongly affects the latency and the throughput. Our main finding includes that the deactivated core located in a corner of 2-D mesh network can reduce the latency, whereas the deactivated core in other locations leads to a degraded performance in terms of both latency and throughput. Therefore, when a chip needs to deactivate its core, we recommend choosing a corner core for deactivation to maintain the performance of NoC.



(a) uniform traffic

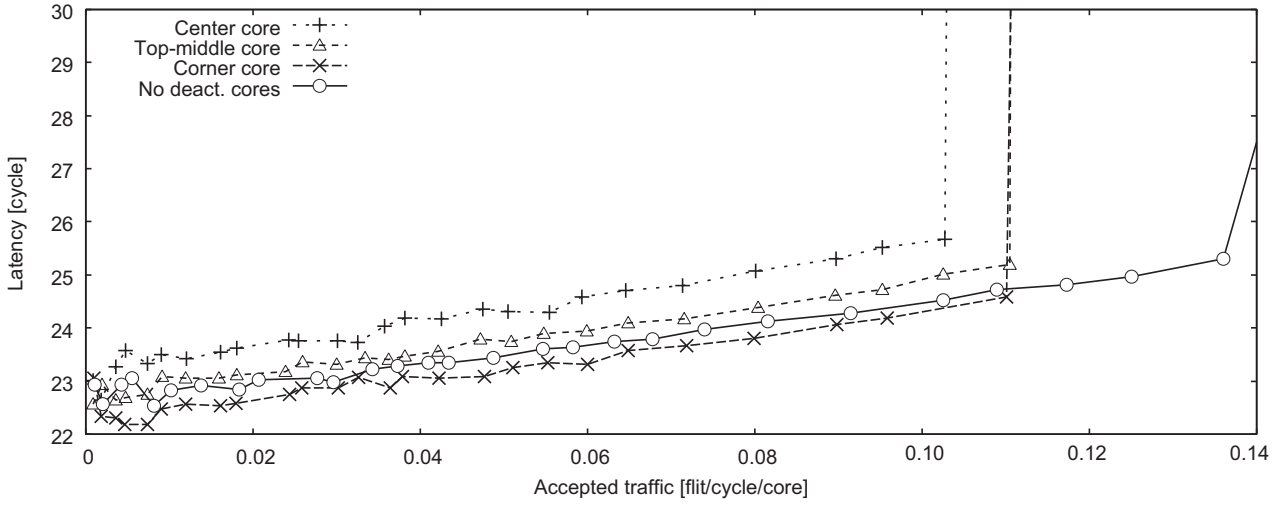


(b) matrix transpose traffic

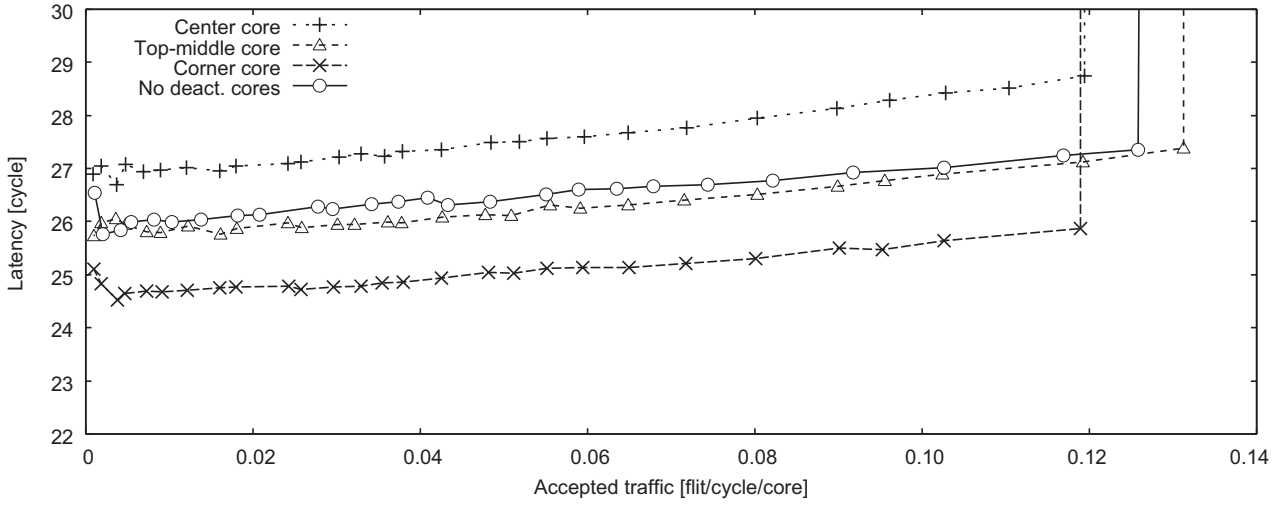


(c) bit reversal traffic

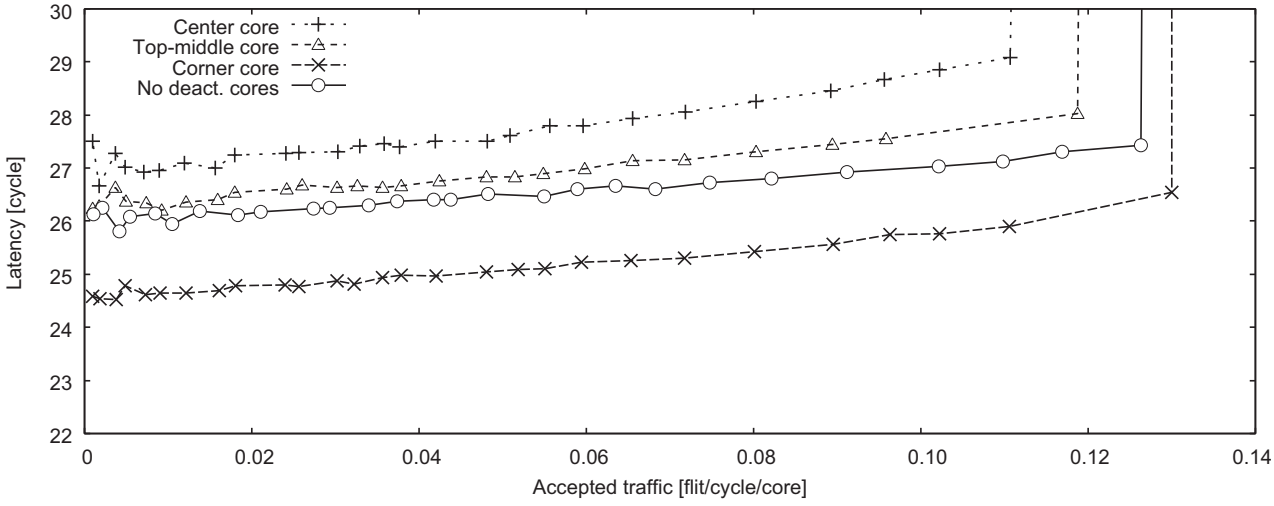
Fig. 4. Accepted traffic vs. latency for 2-D Mesh of 9 cores



(a) uniform traffic



(b) matrix transpose traffic



(c) bit reversal traffic

Fig. 5. Accepted traffic vs. latency for 2-D Mesh of 16 cores

ACKNOWLEDGMENTS

This work was partially supported by JSPS KAKENHI Grant Number 25280043, and by NII Joint Research Fund.

REFERENCES

[1] B. M. Beckmann and D. A. Wood, "Managing Wire Delay in Large Chip-Multiprocessor Caches," in *Proceedings of the International Symposium on Microarchitecture (MICRO'04)*, Dec. 2004, pp. 319–330.

[2] 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 *Proceedings of the International Symposium on Networks-on-Chip (NOCS'10)* , May 2010, pp. 61–68.

[3] Ahmad Samih and Ren Wang and Anil Krishna and Christian Maciocco and Yan Solihin, "Energy-Efficient Interconnect via Router Parking," in *Proceddings of the International Symposium on High Performance Computer Architecture (HPCA)*, Feb. 2013.

[4] J. Flich, T. Skeie, A. Mejia, O. Lysne, P. Lopez, A. Robles, J. Duato, M. Koibuchi, T. Rokicki, and J. C. Sancho, "A Survey and Evaluation of Topology Agnostic Deterministic Routing Algorithms," *IEEE Trans. on Parallel Distrib. Syst.*, vol. 23, no. 3, pp. 405–425, 2012.

[5] M. D. Schroeder, A. D. Birrell, M. Burrows, H. Murray, R. M. Needham, and T. L. Rodeheffer, "Autonet: a High-speed, Self-configuring Local Area Network Using Point-to-point Links," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1318–1335, Oct. 1991.

[6] F. Silla and J. Duato, "Is it Worth the Flexibility Provided by Irregular Topologies in Networks of Workstations?" in *Proceedings of Workshop Comm. and Architectural Support for Network-Based Parallel Computing*, Jan. 1999, pp. 47–61.

[7] M. Koibuchi, "A Partially Network Reconfiguration Mechanism on Two-dimensional Mesh and Torus with Faults," in *The IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN)*, Feb. 2009, pp. 91–96.

[8] J.C.Sancho, A.Robles, J.Flich, P.Lopez, and J.Duato, "Effective Methodology for Deadlock-Free Minimal Routing in InfiniBand," in *Proceedings of the International Conference on Parallel Processing*, Aug. 2002, pp. 409–418.

[9] M. Koibuchi, A. Jouraku, and H. Amano, "Descending Layers Routing: A Deadlock-Free Deterministic Routing using Virtual Channels in System Area Networks with Irregular Topologies," in *Proceedings of the International Conference on Parallel Processing*, Oct. 2003, pp. 527–536.

[10] O. Lysne, T. Skeie, S. A. Reinemo, , and I. Theiss, "Layered Routing in Irregular Networks," *IEEE Transactions on parallel and distributed systems*, vol. 17, no. 1, pp. 51–65, 2006.

[11] D. Ikebuchi *et al.*, "Geyser-1: A MIPS R3000 CPU Core with Fine Grain Runtime Power Gating," in *Proceedings of the IEEE Asian Solid-State Circuits Conference (A-SSCC'09)*, Nov. 2009, pp. 281–284.

[12] K. Usami and N. Ohkubo, "A Design Approach for Fine-grained Run-Time Power Gating using Locally Extracted Sleep Signals," in *Proceedings of the International Conference on Computer Design (ICCD'06)*, Oct. 2006.

[13] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks: an engineering approach*. Morgan Kaufmann, 2002.