A Process-Variation-Adaptive Network-on-Chip with Variable-Cycle Routers and Variable-Cycle Pipeline Adaptive Routing

Nakata, Yohei

Kawaguchi, Hiroshi

Yoshimoto, Masahiko

(Citation)
IEICE Transactions on Electronics,95(4):523-533

(Issue Date)
2012-04-01

(Resource Type)
journal article

(Version)
Version of Record

PAPER *Special Section on Solid-State Circuit Design — Architecture, Circuit, Device and Design Methodology*

A Process-Variation-Adaptive Network-on-Chip with Variable-Cycle Routers and Variable-Cycle Pipeline Adaptive Routing*

Yohei NAKATA[†a)], ***Student Member***, **Hiroshi KAWAGUCHI**[†], ***and*** **Masahiko YOSHIMOTO**[†,††], ***Members***

SUMMARY As process technology is scaled down, a typical system on a chip (SoC) becomes denser. In scaled process technology, process variation becomes greater and increasingly affects the SoC circuits. Moreover, the process variation strongly affects network-on-chips (NoCs) that have a synchronous network across the chip. Therefore, its network frequency is degraded. We propose a process-variation-adaptive NoC with a variation-adaptive variable-cycle router (VAVCR). The proposed VAVCR can configure its cycle latency adaptively on a processor core basis, corresponding to the process variation. It can increase the network frequency, which is limited by the process variation in a conventional router. Furthermore, we propose a variable-cycle pipeline adaptive routing (VCPAR) method with VAVCR; the proposed VCPAR can reduce packet latency and has tolerance to network congestion. The total execution time reduction of the proposed VAVCR with VCPAR is 15.7%, on average, for five task graphs.

key words: *network-on-chip, process variation, adaptive circuits, routing algorithm*

1. Introduction

The minimum feature size of a CMOS process technology is scaled down, which enables higher density and lower chip fabrication cost. However, process variation is increased by technology scaling. Process variation strongly affects system-on-a-chip (SoC) circuit characteristics. A network-on-chip (NoC), which is one SoC that is emerging as a highly efficient network fabric for many-core processors [1], [12], [13], commonly adopts a synchronous design for a network across the chip. The NoC in a many-core processor has many network components, each of which is affected by process variation. The network component delays vary considerably as the network components become more numerous. Therefore, the frequency of a large-scale chip-wide synchronous network is degraded to the level of the slowest network component. Many studies have been undertaken to find means to mitigate the variations of many-core processors using dynamic voltage and frequency scaling (DVFS) [2], application scheduling [11], fine-grain body biasing (FGBB) [2], and dynamic voltage frequency-core scaling (DVFCS) [3]. However, no study has specifically addressed variation in a large-scale chip-wide synchronous network. In this paper, we examine process variation in an NoC.

The contribution of this paper is a proposal for a process-variation-adaptive NoC using a variation-adaptive variable-cycle router (VAVCR) and a novel routing scheme named variable-cycle pipeline adaptive routing (VCPAR) for the NoC with the VAVCR. The proposed VAVCR can configure the cycle latency of the router in adaptation to the spatial process variation. Thereby, the NoC with the proposed VAVCR can enhance the network frequency and the overall throughput. The proposed VCPAR is adaptive to the variable cycle latency of the proposed VAVCR. The VCPAR can reduce packet latency and can be tolerant of network congestion.

The paper is organized as follows. Section 2 describes the background of our work including the impact of process variation on NoC circuits. Section 3 presents the proposed VAVCR and VCPAR. In Sect. 4, we evaluate the proposed VCPAR method with the proposed VAVCR, and exhibit their effectiveness. Section 5 discusses the settings of the network frequency. In Sect. 6, we conclude this paper.

2. Background

2.1 Process Variation

Technology scaling increases the threshold-voltage (V_{th}) variation of MOS transistors composed of die-to-die (D2D) and within-die (WID) variations, of which the WID variation is divided into systematic and random variations. This paper considers the WID variation because it affects the characteristics of individual cores within a die, which turns out to be core-to-core (C2C) variation. Systematic variation results mainly from lens aberration and has a spatial correlation [14]. Therefore, neighboring transistors have similar characteristics. In contrast, random variation results mainly from random dopant fluctuation (RDF) and line-edge roughness (LER): random variations show no spatial correlation. For that reason, individual transistors have different characteristics from those of neighboring transistors.

2.2 Process Variation in NoC

Process variation in an NoC shows up as variation of op-

Manuscript received August 17, 2011.
Manuscript revised November 4, 2011.

[†] The authors are with Kobe University, Kobe-shi, 657-8501 Japan.
[††] The author is with JST CREST, Tokyo, 102-0076 Japan.

[*] This paper is the extended version of the 2011 EUROMICRO DSD publication [17].

a) E-mail: nkt@cs28.cs.kobe-u.ac.jp

DOI: 10.1587/transele.E95.C.523

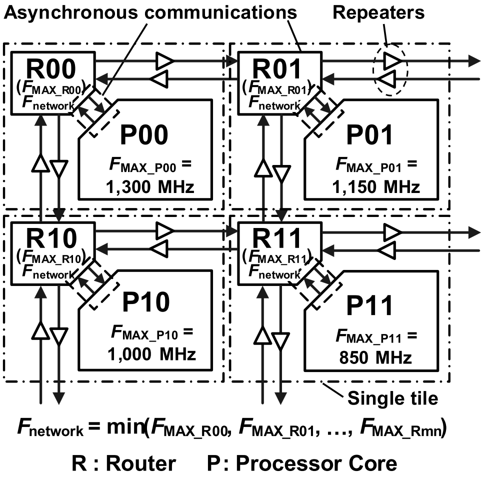
Asynchronous communications
Repeaters
R00
(F_{MAX_R00})
$F_{network}$
P00
F_{MAX_P00} = 1,300 MHz
R01
(F_{MAX_R01})
$F_{network}$
P01
F_{MAX_P01} = 1,150 MHz
R10
(F_{MAX_R10})
$F_{network}$
P10
F_{MAX_P10} = 1,000 MHz
R11
(F_{MAX_R11})
$F_{network}$
P11
F_{MAX_P11} = 850 MHz
Single tile
$F_{network} = \min(F_{MAX\_R00}, F_{MAX\_R01}, \ldots, F_{MAX\_Rmn})$
R : Router P: Processor Core

Fig. 1 Operating frequency variation in a GALS NoC. The operating frequencies of processor cores ($F_{\mathrm{MAX\_Pmn}}$) vary. The network frequency (F_{network}) is determined by the minimum operating frequency in routers.

erating frequencies of individual cores. Considering synchronous designs for entire NoC processor cores in situations where operating frequencies vary, each core in the NoC must synchronize with the slowest core. Therefore, the throughput of the entire NoC processor degrades with increasing impact of the process variation. Global-asynchronous local-synchronous (GALS) designs, in which the fabric with individual cores and network elements operate at their own maximum frequencies, are widely adopted in NoC design. The network portion composed of routers, wires, and buffers is designed frequently at a single frequency and in a single voltage domain [3], [15], [16] because the design of the network portion in an NoC is too complicated and overly costly when adopting multi-frequency and multi-voltage design. However, when the network portion is with a single frequency and a single voltage domain, its operating frequency is determined by the slowest component (such as a router and a buffer) because operating frequencies of routers and buffers distributed across the entire chip vary according to process variation. This issue is extremely important in a large NoC fabricated using scaled process technology.

Figure 1 portrays the operating frequency variation in a GALS NoC. A processor core and a router communicate asynchronously with each other at a different frequency. An operating frequency in a processor core is determined by each maximum operating frequency ($F_{\mathrm{MAX\_Pmn}}$). The network frequency on the entire NoC (F_{network}) is determined by the minimum (= worst) operating frequency among all routers. Detailed discussion of the variations in a processor core and an NoC are presented respectively in Sects. 2.3 and 2.4.

2.3 Impact of Variation in Processor Core

This section presents a description of the impact of the process variation to the processor core. Assuming a 20 FO4 inverter chain delay as a single pipeline stage in the processor core, we conducted Monte Carlo simulations in a 65-nm process technology using a SPICE circuit simulator. The systematic variation in a threshold voltage (V_{th}) arises as C2C variation. In this simulation, the standard deviation of the systematic variation, σ_{system}, is calculated with [4], as 6.3% of the average V_{th}. Random variation is apparent at individual transistors. Consequently, it affects all circuits in the core. We use standard deviations of random variations in NMOSes and PMOSes from actual measurement [5]. We set the respective standard deviations, $\sigma_{\mathrm{rnd\_NMOS}}$ and $\sigma_{\mathrm{rnd\_PMOS}}$, to 43 mV and 28 mV (in sizing of L = 60 nm and W = 140 nm). The parameters used for estimation of the operating frequency are presented in Table 1.

Table 1 Parameters used for operating frequency estimation.

Technology	65-nm CMOS	σ_{system}	0.063 / μ_{Vth}
Process corner	TT	σ_{rnd_NMOS}	43 mV
Temperature	25℃	σ_{rnd_PMOS}	28 mV
# of Monte Carlo	10,000		

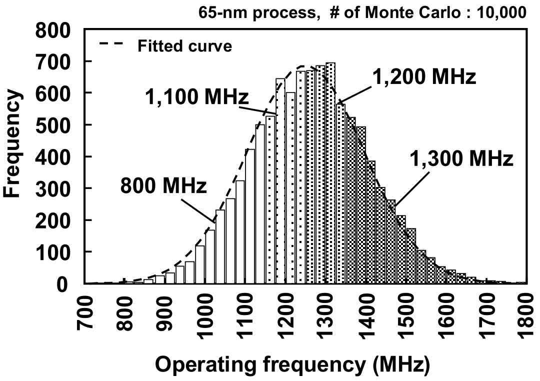

Fig. 2 Distribution of operating frequencies of 20 FO4 inverters. The dashed line signifies the fitted normal distribution curve.

Table 2 Operating frequency characteristics.

$\mu_{frequency}$	1,237.7 MHz	$\mu_{frequency} + 3\sigma_{frequency}$	1,653.6 MHz
$\sigma_{frequency}$	145.4 MHz	$\mu_{frequency} - 3\sigma_{frequency}$	801.5 MHz
Maximum	1,802.1 MHz	Minimum	775.3 MHz

Figure 2 shows the distribution of the operating frequencies obtained through simulations of 20 FO4 inverters. We set four frequency bins: 800 MHz, 1,100 MHz, 1,200 MHz, and 1,300 MHz. Details of the frequency bins are presented in Table 4. Table 2 shows summary statistics of the operating frequency distribution. From Fig. 2, the operating frequency variation derived from the V_{th} variation is apparent as a normal distribution. The standard deviation of the operating frequencies, $\sigma_{\mathrm{frequency}}$, in the 20 FO4 inverters is 145.4 MHz. Accordingly, the individual processor cores in an NoC under the V_{th} variation represent mutually differing operating frequency characteristics.

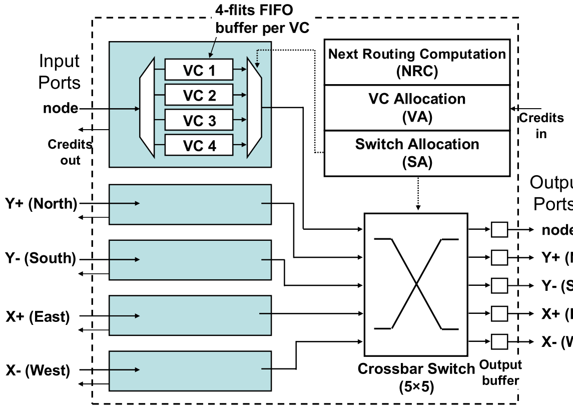

Fig. 3 Organization of the router. Parameters are the same as those of Table 3.

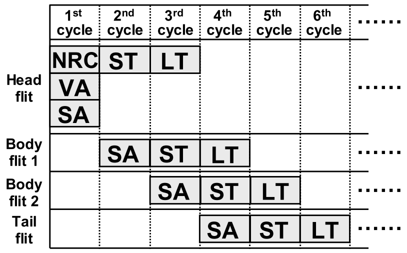

Fig. 4 Gantt chart of each pipeline stage of the router.

2.4 Impact of Variation in On-Chip Networks

Figure 3 depicts the organization of a router with virtual channels (VCs) [12]. Figure 4 presents a Gantt chart of each pipeline stage of the router. The pipeline stages are described as follows. The next routing computation stage (NRC) determines a hop direction for the next router, not for the current router. The virtual channel allocation stage (VA) allocates output VCs to the input packets. The switch allocation stage (SA) arbitrates the crossbar switch for the flit. The switch traversal stage (ST) delivers the packet across the crossbar to the output buffer. The link traversal stage (LT) traverses the packet from the output buffer to the next router. The SA, ST, and LT stages operate on every flit of the packet, differently from the NRC and VC stages, which compute once per packet.

As described in Sect. 2.2, F_{network} is degraded by a single frequency domain for the entire network portion because all components of the network portion must be synchronized with the slowest one. In this section, the delay variation in each pipeline stage of the router is evaluated. We used an open-source RTL of a router [6]. The router was synthesized using a 65-nm process technology with Synopsys Design Compiler. The configurations of the router synthesis are shown in Table 3. Then, the synthesized netlist was evaluated using a SPICE circuit simulator, and the delay variation was obtained. As parameters for the variation, the parameters shown in Table 1 were used as described in Sect. 2.1. We assumed the link length between nodes as 1 mm for the delay evaluation.

Table 3 Parameters for router delay estimation.

Topology	8 × 8 mesh
Flit size	64 bits
Routing	X-Y DOR [19]
Router type	Speculative, look ahead
# of VCs	4
VC buffer size	4 flits
# of input/output ports	5 (X+/-, Y+/- and own node)

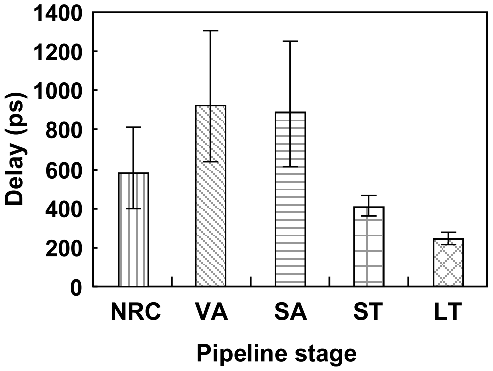

Fig. 5 Delay of each pipeline stage: NRC, next routing computation; VA, virtual channel allocation; SA, switch allocation; ST, switch traversal; and LT, link traversal.

The evaluated result for the delays of each pipeline stage is depicted in Fig. 5. The upper bound (i.e. the worst delay) of each stage is assumed as 99.7% of the whole. The longest delay in the pipeline stage is the virtual channel allocation (VA) stage. The delay of the VA stage varies: 627–1319 ps.

3. Proposed Process-Variation-Adaptive Variable-Cycle Router and its Proposed Routing Algorithm

3.1 Process-Variation-Adaptive Variable-Cycle Router

In this section, a process-variation-adaptive variable-cycle router (VAVCR) is proposed. The proposed VAVCR can configure the cycle latency of the router corresponding to spatial process variation. The VAVCR can realize a variation-adaptive NoC configuration. Figure 6 presents timing diagrams of the conventional and proposed router pipelines. The values of the delays are brought from Fig. 5. Figures 6(a) and 6(c) show the worst delays (i.e. combination of the upper-bound delays in Fig. 5). Figures 6(b) and 6(d) show the best delays (i.e. the lower bounds in Fig. 5).

In the conventional router pipeline, the router frequency is determined by the worst delay (Fig. 6(a)). Accordingly, a great amount of slack emerges at the conventional router pipeline that operates in the best delay (Fig. 6(b)). Consequently, the larger the process variation, the greater is the slack at the conventional router pipeline.

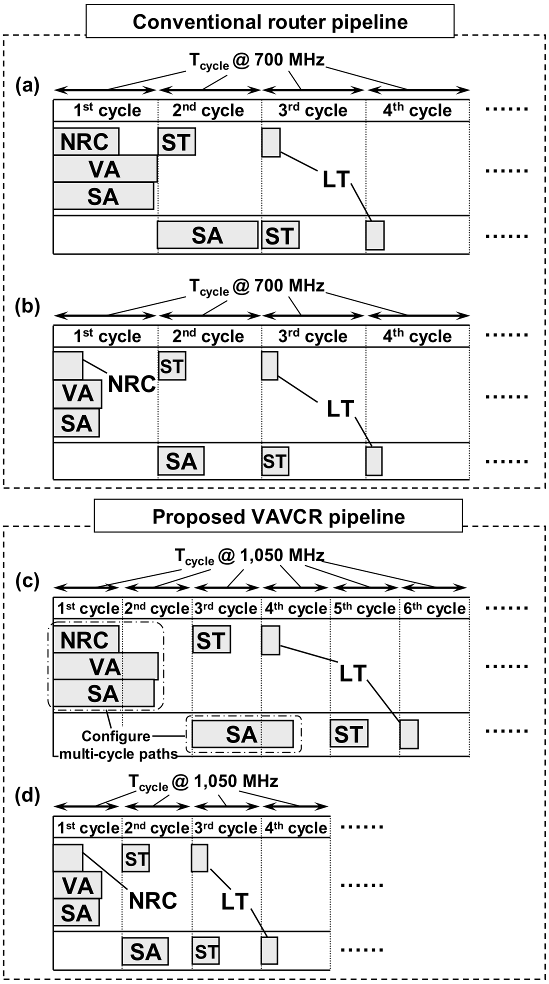

Fig. 6 Timing diagrams of the conventional and proposed router pipelines. Here, (a) and (b) respectively correspond to the worst and best delays in the conventional router pipeline; (c) and (d) respectively correspond to the worst and best delays in the proposed router pipeline.

Figures 6(c) and 6(d) portray timing diagrams of the proposed VAVCR pipeline. The VAVCR pipeline applies multi-cycle paths to NRC, VA, and SA stages (Fig. 6(c)) when a delay in the stages exceeds the predefined cycle time. The cycle time is set to 1/1,050 MHz in this example. In the case where no delay of the pipeline stage exceeds the predefined cycle time, the VAVCR pipeline does not apply the multi-cycle paths; it operates in the same way as the conventional pipeline, but it can do so at a higher frequency (compare Fig. 6(d) to Fig. 6(b)). Therefore, the proposed VAVCR pipeline can reduce the large slack at the conventional router pipeline, and can realize greater network throughput.

Figure 7 portrays a distribution of the router cycle latency for an 8 × 8 mesh network. The number in the circle is the latency of the router. Figures 7(a) and 7(b) respectively depict the networks of the conventional router and proposed VAVCR. In the proposed VAVCR, the routers are configured as a three-cycle latency router or a four-cycle latency router corresponding to the spatial process variation on the chip. The pipeline delay of each router can be measured in a burn-in test, and can configure the cycle latency in a testing process. F_{network} can be increased to 1,050 MHz from 700 MHz by applying the proposed VAVCR.

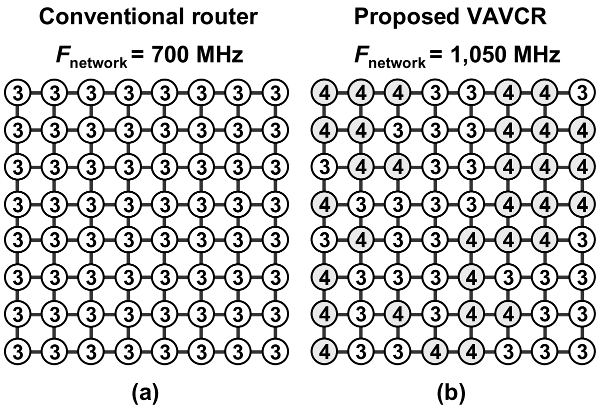

Fig. 7 Distributions of the router latencies for 8 × 8 mesh networks: (a) conventional router and the (b) proposed VAVCR.

3.2 Variable-Cycle Pipeline Adaptive Routing

In the proposed VAVCR, the packet latency is increased by the variable-cycle router pipeline. In this section, we propose a specific routing algorithm, considering the spatial distribution of the router latency.

The proposed variable-cycle pipeline adaptive routing (VCPAR) employs the odd-even turn model [18] to avoid deadlocks; it can select a hop direction adaptively considering their pipeline latencies with neighboring VAVCRs. VCPAR aims for low-latency routing in an NoC with the VAVCRs. The detailed procedure in the VCPAR algorithm is described as follows:

1 Each VAVCR has a distribution of the router latency similar to that shown in Fig. 7(b). The distribution information on the mutual router latencies is stored in a testing process.
2 Five ports of a VAVCR have five transmission counters storing the number of packet transmissions.
3 On the way to the destination router, if a next router position (NRP) is at the same row or at the same column, the hop direction is set to a straight-ahead direction (the row address and the column address are increased or decreased monotonically).
4 In a false case of Procedure 3 and if the destination is toward east:

4.1 If the NRP is at an even column, then the available direction can be set to east.
4.2 If the NRP is at an odd column, then the available direction can be set to east or either north or south according to the destination direction.

5 In a false case of Procedure 2 and if the destination is toward the west:

5.1 If the NRP is at an odd column, then the available

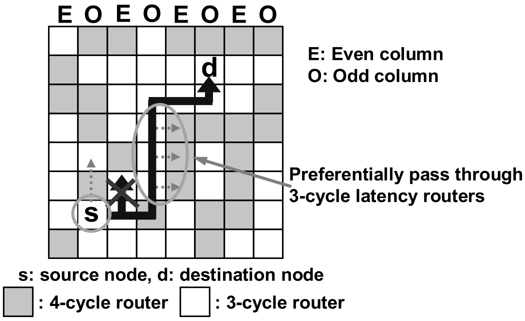

Fig. 8 Overview of the proposed VCPAR method.

direction can be set to west.

5.2 If the NRP is at an even column, the available direction can be set to west or either north or south according to the destination direction.

6 If only one direction is available, then the packet is transmitted to that direction.

7 If two directions are available, then the VAVCR checks the transmission counters of the two ports.

7.1 If the two transmission counters have equal values, then the packet is transmitted to the direction which has the least pipeline latency.

7.2 If the two transmission counters have different values, then the packet is transmitted to the direction which has a lower value.

8 The transmission counter in the transmitted direction is incremented by a size of the packet. All transmission counters are decremented by one in each cycle.

The proposed VCPAR reduces packet latency with preferential selection of three-cycle latency routers unless the routers are congested (Fig. 8). The VCPAR uses only two-hop-ahead routers' latencies and makes less complexity routing than other routing methods that compute global-variation-adaptive routing paths on an entire NoC. In addition, the transmission counter avoids congestion through specific paths by preferential routing and enhances the communication efficiency.

4. Evaluation

In this section, we present an evaluation of the proposed VAVCR and the proposed VCPAR. First, we present the evaluation of routing methods for the NoC with the VAVCR including the conventional routing and the proposed VCPAR in Sects. 4.1 and 4.2. From this evaluation, the routing method suitable for the NoC with the proposed VAVCR and the effectiveness of the proposed VCPAR can be obtained. Second, we evaluate the proposed VAVCR with VCPAR using task graphs in Sects. 4.3 and 4.4. Lastly, we estimate the area overhead of the proposed VAVCR with the VCPAR in Sect. 4.5.

Table 4 Frequencies and ratios in the frequency bins and router latencies in the proposed VAVCR.

Frequency bin	Frequency 0	Frequency 1	Frequency 2	Frequency 3
Frequency	800 MHz	1,100 MHz	1,200 MHz	1,300 MHz
Ratio	27.6%	25.8%	24.7%	23.2%
Router latency of the proposed VAVCR	4 cycles	4 cycles	3 cycles	3 cycles

4.1 Evaluation Methodology of Routing Methods

We used a BookSim simulator [7] to evaluate the entire NoC implemented with the proposed VAVCR. The BookSim simulator was modified to evaluate the proposed VAVCR and proposed VCPAR. The router configuration is identical to that shown in Table 3, except for the routing method.

The spatial process variation is modeled using a simplified VARIUS model [4]. The spatial correlation parameter is assumed as $\Phi = 0.5$ We used a simple spatial process variation model that has the same V_{th} value within a single tile, which includes a processor core, router, and repeater buffers, as shown in Fig. 1. The variation parameters in Table 1 are used as explained in Sect. 2. The processor core frequencies are determined by the V_{th} variation map and the frequency bins in Table 4. The router latencies of the proposed VAVCR are four cycles for Frequency 0 and 1 bins, and three cycles for Frequency 2 and 3 bins. Ten variation maps (chips) are taken in this evaluation. We evaluate the conventional routing method including X-Y DOR [19], ROMM [20], Toggle X-Y (TXY) [21], Odd-Even Random, and the proposed VCPAR method. Odd-Even Random routing uses the Odd-Even turn model in which the next hop direction is determined randomly if it has two available directions. All evaluation in this section, the NoC with the proposed VAVCR is used.

Traffic patterns used for the routing evaluation are uniform random, transpose, bit reverse, hot spot with one hot spot, and hot spot with four hot spots (the hot spot percentage is 6%) [22]. The packet size is four flits and 16 flits.

4.2 Evaluation of the Routing Method

Figures 9 and 10 present the evaluation results of routing methods (respective packet size are four flits and 16 flits). The X-axis shows the injected traffic (packets/ cycle/ node); the Y-axis specifies the average packet latency (= cycles) for the ten variation maps.

In the case of uniform random traffic (Fig. 9(a) and Fig. 10(a)), X-Y DOR outperforms the other routing methods. This is reasonable because the uniform random traffic is uniform and suitable for X-Y DOR [18]. In the transpose traffic (Fig. 9(b) and Fig. 10(b)) and the bit reverse traffic (Fig. 9(c) and Fig. 10(c)), the proposed VCPAR yields the lowest latency and exhibits the best tolerance to network congestion (except the transpose for the 16 flit packet

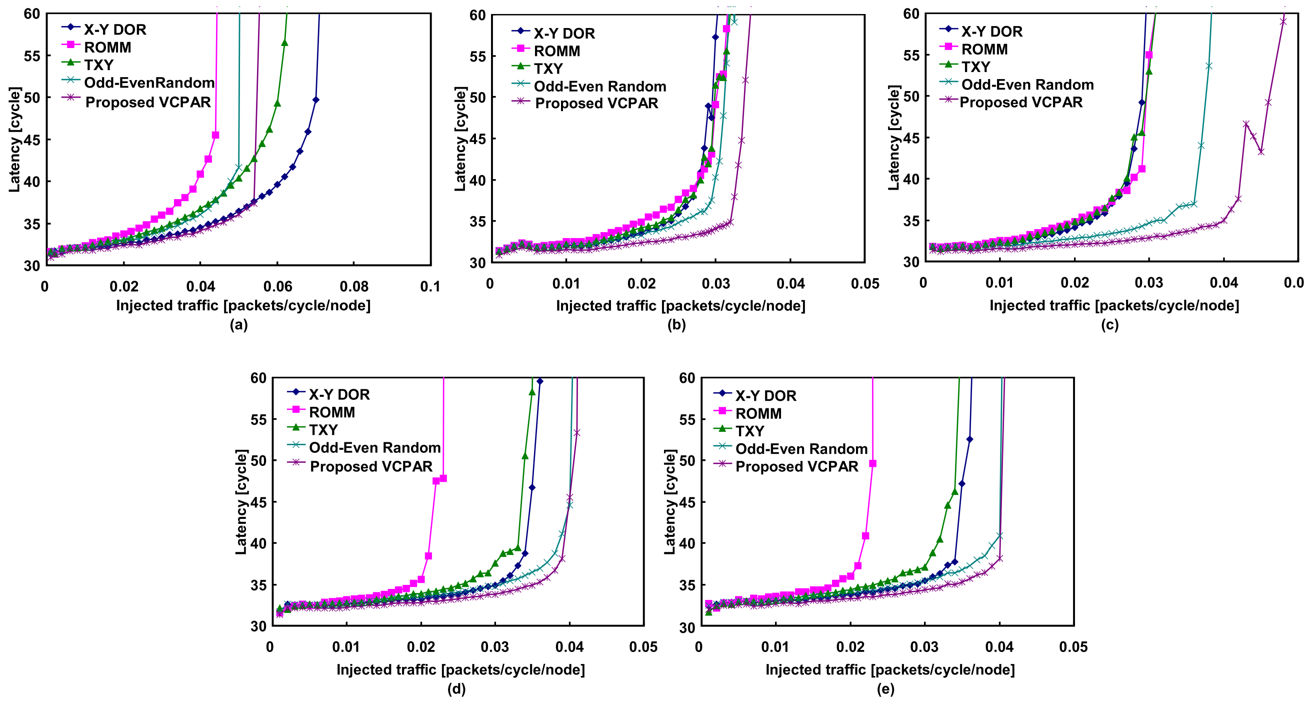

Fig. 9 Evaluation results of routing methods (packet size is four flits): (a) uniform random traffic, (b) transpose traffic, (c) bit reverse traffic, (d) hot spot traffic with one hot spot node (the hot spot percentage is 6%), and (e) hot spot traffic with four hot spot nodes (the hot spot percentage is 6%).

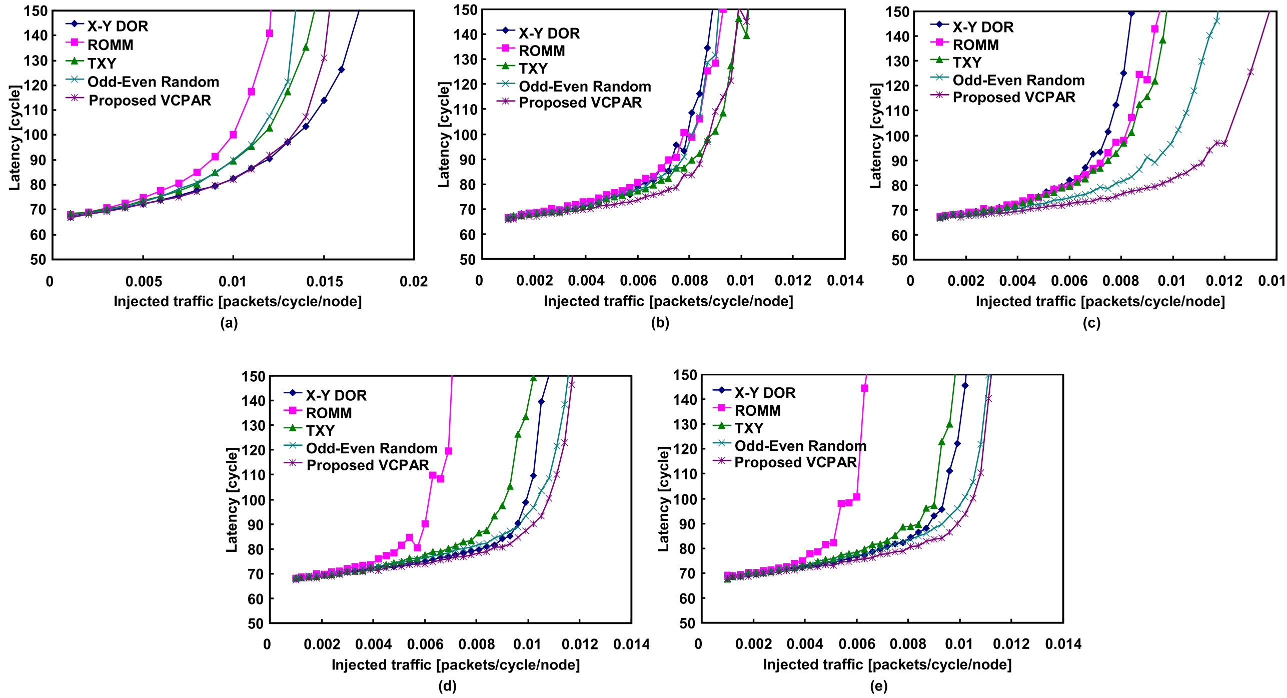

Fig. 10 Evaluation results of routing methods (packet size is 16 flits): (a) uniform random traffic, (b) transpose traffic, (c) bit reverse traffic, (d) hot spot traffic with one hot spot node (the hot spot percentage is 6%), and (e) hot spot traffic with four hot spot nodes (the hot spot percentage is 6%).

size (Fig. 10(b)). TXY can avoid congestion on the specific paths in the transpose because it can select the next hop direction randomly from two available directions). This fact demonstrates that preferentially selecting three-cycle latency routers can reduce the packet latency; adaptability based on direction selection with the transmission counter

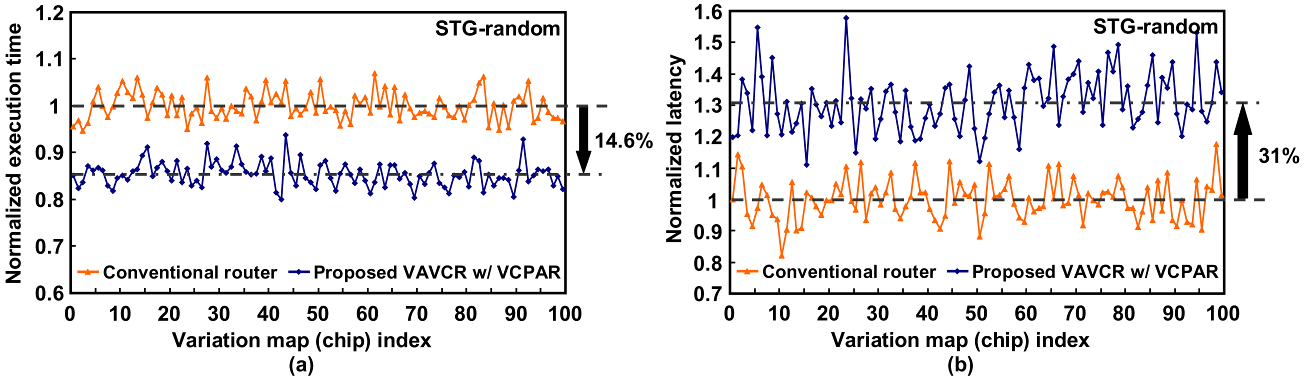

Fig. 11 Evaluation results of STG-random: (a) normalized execution time and (b) normalized latency.

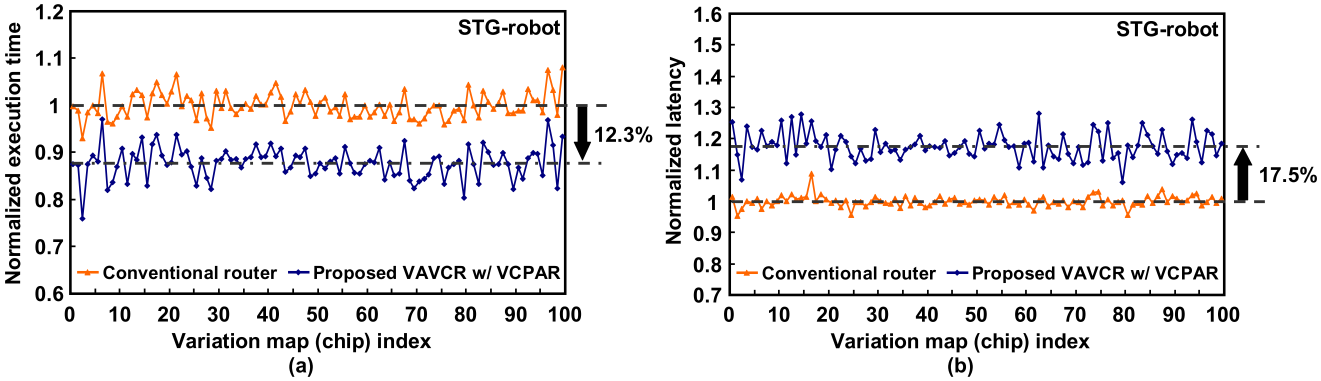

Fig. 12 Evaluation results of STG-robot: (a) normalized execution time and (b) normalized latency.

(described in Sect. 4.1) alleviates network congestion. The same tendency is observed for hot spot traffic (Figs. 9(d) and 9(e), Figs. 10(d) and 10(e)). The proposed VCPAR outperforms the other routing methods in the hot spot traffic. The proposed VCPAR with the VAVCR makes use of process variation and has a low-latency feature even in the network congestion.

4.3 Evaluation Methodology of VAVCR with VCPAR

To evaluate the proposed VAVCR with the VCPAR, we use the same methodologies and parameters described in Sect. 4.1. In this evaluation, 100 different variation maps (chips) are assessed. The network frequency for the conventional router and proposed VAVCR are 700 MHz and 1,050 MHz, respectively. The conventional router adopts X-Y DOR as a routing method.

In reality, the optimal network frequency for the proposed VAVCR, which maximizes throughput, depends on characteristics of the traffic pattern. In this evaluation, we took 1,050 MHz as F_{network}. The detailed discussion on the optimal network frequency will follows in Sect. 5.

As the traffic pattern used in this evaluation, we used the standard task graph set (STG) [8] and task graphs for free (TGFF) [9]. For the STG, we used random (500 tasks, the task graph number is 0000), robot[†], sparse[††], and fpppp[†††]. We set the packet size of each edge as 16 ± 8 flits. For TGFF, we set parameters as follows: number of tasks = 500, processing cycle of tasks = 3,000 ± 1,500; and packet size = 32 ± 16 flits [10]. Each task in the task graph is assigned to the processor core based on the critical path method [23].

4.4 Evaluation of VAVCR with VCPAR

Figures 11–15 present the evaluation results for the conventional router and the proposed VAVCR with the VCPAR. They signify the total execution times of the task graphs. Each result includes the evaluation of 100 variation maps (chips): The index number is from 0 to 99 in the figures. The execution times are normalized by the average of the conventional router. The dashed and chained lines respectively represent the average of the conventional router and the proposed VAVCR with the VCPAR.

In Fig. 11, the proposed VAVCR is shown to reduce the total execution time of the STG-random by 14.6% on average. The packet latency of the STG-random is increased by 31% on average. The execution time is reduced because of the increase in the network frequency.

[†] STG-robot: is a task graph for Newton–Euler dynamic control calculation.

[††] STG-sparse is a task graph for a random sparse matrix solver of an electronic circuit simulation.

[†††] STG-fpppp is a task graph for subroutine of SPEC95fp fpppp.

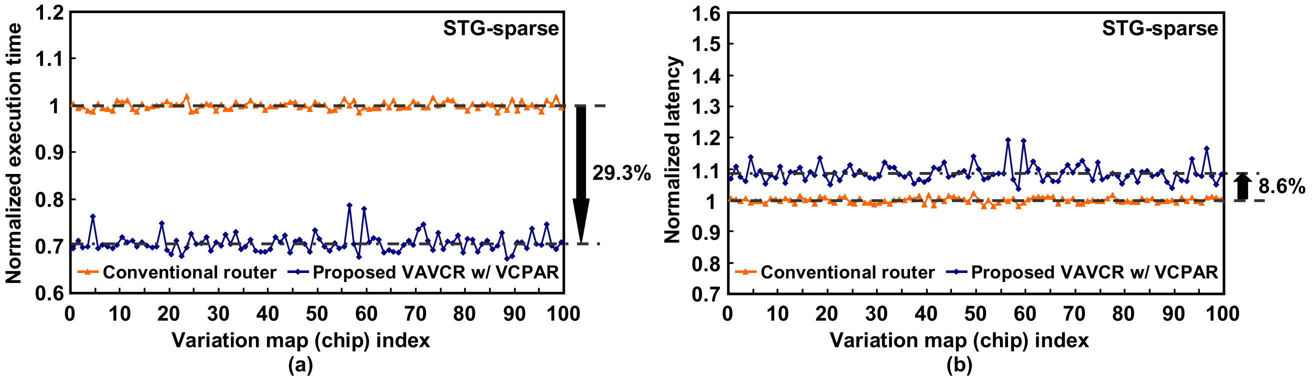

Fig. 13 Evaluation results of STG-sparse: (a) normalized execution time and (b) normalized latency.

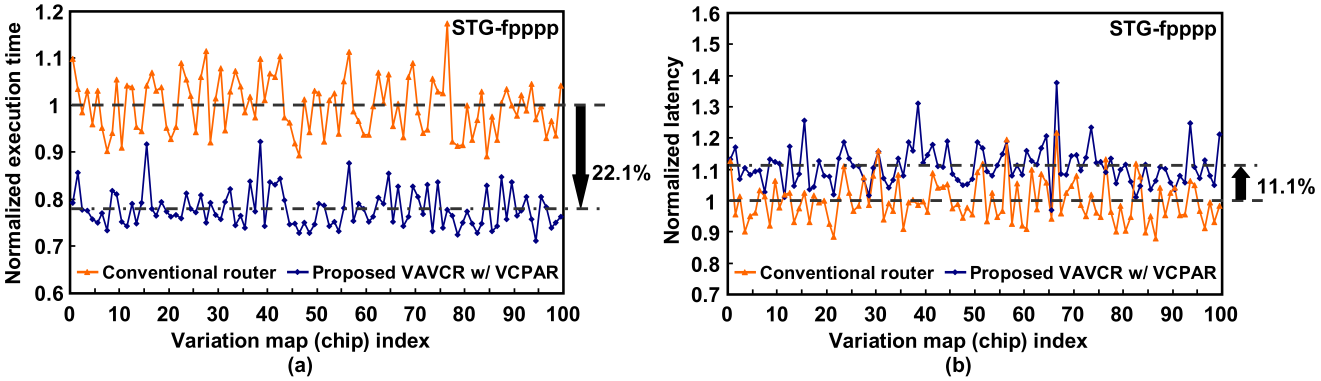

Fig. 14 Evaluation results of STG-fpppp: (a) normalized execution time and (b) normalized latency.

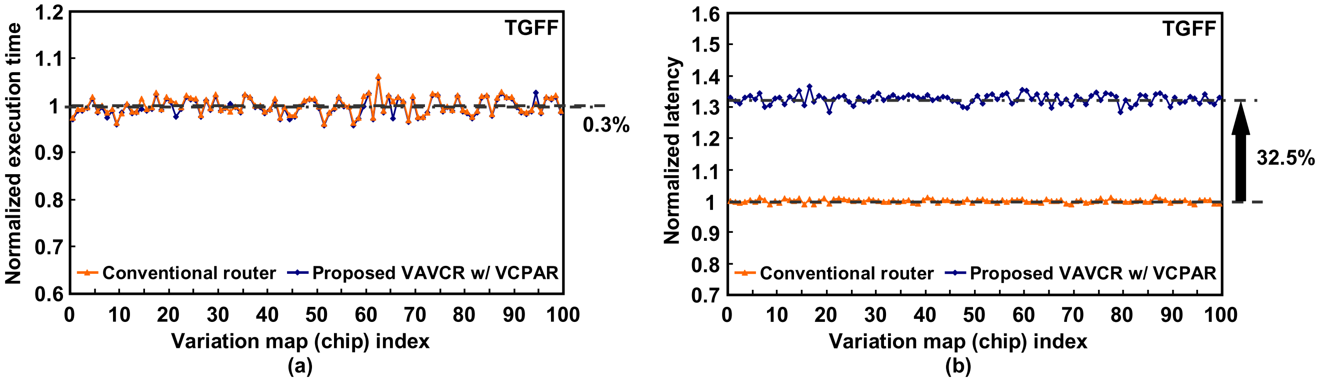

Fig. 15 Evaluation results of TGFF: (a) normalized execution time and (b) normalized latency.

The packet latency of the STG-random is increased because of the existence of the four-cycle routers. Irrespective of the amount of the increase in the packet latency, the proposed VAVCR reduces the total execution time. Similarly, the proposed VAVCR reduces the total execution times of the STG-robot (Fig. 12(a)), STG-sparse (Fig. 13(a)), STG-fpppp (Fig. 14(a)), and TGFF (Fig. 15(a)) by 12.3%, 29.3%, 22.1%, and 0.3% on average, respectively. The packet latencies of the STG-robot (Fig. 12(b)), STG-sparse (Fig. 13(b)), STG-fpppp (Fig. 14(b)), and TGFF (Fig. 15(b)) were increased by 17.5%, 8.6%, 11.1%, and 32.5% on average, respectively.

Table 5 Evaluation results of 100 variation maps. (F_{network} = 1,050 MHz)

	Reduction of the total execution time	Increase in the packet latency	$\sigma_{\mathrm{exec\_time}}$ of conv.	$\sigma_{\mathrm{exec\_time}}$ of prop.	$\sigma_{\mathrm{latency}}$ of conv.	$\sigma_{\mathrm{latency}}$ of prop.
STG-random	14.6%	31%	2.97%	2.68%	6.42%	9.28%
STG-robot	12.3%	17.5%	2.79%	3.4%	1.75%	4.45%
STG-sparse	29.3%	8.6%	0.78%	1.95%	0.86%	2.75%
STG-fpppp	22.1%	11.1%	5.91%	4.04%	6.9%	6.34%
TGFF [17]	0.3%	32.5%	1.81%	1.86%	0.53%	1.54%
Avg. of 5 TGs	15.7%	20.2%	2.85%	2.79%	3.29%	4.87%

The proposed VAVCR can efficiently reduce the total execution time necessary for executing network-bound tasks such as STG-random, STG-sparse, and STG-fpppp. In contrast, the proposed VAVCR reduces it inefficiently when executing computation-bound tasks such as TGFF.

Table 5 presents a summary of the reductions of the total execution time, the increases in the packet latency, the standard deviations of the total execution times in the conventional router, and the proposed VAVCR with the VC-PAR, the standard deviations of the packet latencies in the conventional router, and the proposed VAVCR with the VC-PAR. They are 15.7%, 20.2%, 2.85%, 2.79%, 3.29%, and 4.87% on average of the five task graphs (TGs), respectively.

4.5 Area Overhead of the VAVCR w/ VCPAR

To estimate the area overhead of the proposed VAVCR with the VCPAR, its transistor count is to be compared with that of the conventional router. The transistor count of the conventional router and the proposed VAVCR with the VCPAR are 618.1 k and 629.1 k, respectively. The area overhead of the proposed VAVCR with the VCPAR is 1.78% as a single router, which infers that the total area overhead will turn out almost negligible because a router portion is much smaller than a processor portion in an NoC.

5. Discussion on the Network Frequency Optimization

In this section, the optimization of the network frequency is discussed. The execution time and packet latency depends on F_{network}. Figures 16(a) and 16(b) show the reduction of the total execution time and increase in the packet latency when F_{network} is varied. 100 variation maps are again utilized as well as in Sect. 4.4. "Static" in the figure means "full use of the network", in which a single packet has 16 flits. We utilize "Static" as a reference to be compared with the other five task graphs.

The reductions in the total execution times of the STG-random and STG-fpppp monotonically increase with F_{network} because they do not incur network congestion; a faster F_{network} is better in these cases. The STG-random presents degradation of the execution time at a network frequency of 850 MHz or less because its traffic is uniform and thus X-Y DOR is eligible for it (we have already discussed this in Sect. 4.2). In contrast, the STG-sparse and STG-robot that incur network congestion have local maximums at 1,100 MHz and 950 MHz, respectively, in terms of reduction in the execution time. Note that they have similar shapes to "Static" that fully uses the network. The TGFF is not affected by F_{network} at all because it is a compute-bound task; the computation occupies over 99% of the total execution cycles.

From this discussion, an appropriate F_{network} should be set by designers based on characteristics of a traffic pattern.

6. Conclusion

As described in this paper, we proposed a process-variation-adaptive NoC with a variation-adaptive variable-cycle router (VAVCR) and a variable-cycle pipeline adaptive routing method (VCPAR). The proposed VAVCR can configure its cycle latency adaptively corresponding to the spatial process variation. It increases the network frequency, which is limited by the slowest network component in the conventional router. The proposed VAVCR can reduce the total execution time by 15.7% based on an average of the five task graphs at a network frequency of 1,050 MHz. The proposed VC-PAR can reduce packet latencies in the NoC adaptively with variable cycle router and can efficiently suppress network congestion.

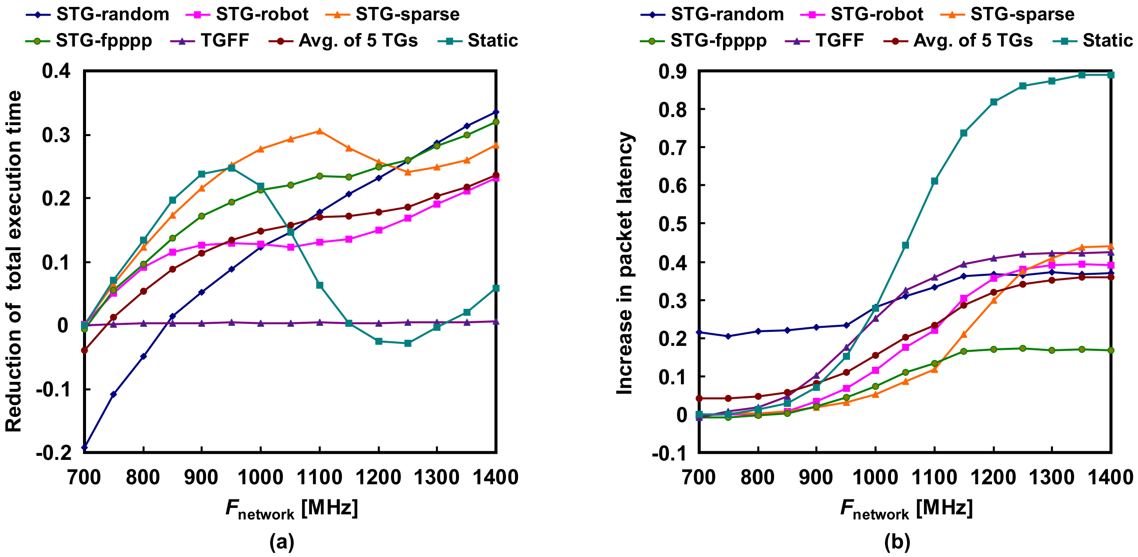

Fig. 16 (a) reduction of the total execution time versus F_{network} (averaged by 100 variation maps) and (b) increase in the packet latency versus F_{network} (averaged by 100 variation maps).

Acknowledgments

This work was supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Cadence Design Systems, Mentor Graphics, and Synopsys, Inc.

References

[1] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," Comput., vol.35, no.1, pp.70–78, Jan. 2002.

[2] R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "Mitigating parameter variation with dynamic fine-grain body biasing," International Symposium on Microarchitecture (MICRO), pp.27–42, 2007.

[3] S. Dighe, S.R. Vangal, P. Aseron, S. Kumar, T. Jacob, K.A. Bowman, J. Howard, J. Tschanz, V. Erraguntla, N. Borkar, V.K. De, and S. Borkar, "Within-die variation-aware dynamic-voltage-frequency-scaling with optimal core allocation and thread hopping for the 80-Core TeraFLOPS processor," IEEE J. Solid-State Circuits, vol.46, no.1, pp.184–193, Jan. 2011.

[4] S.R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "VARIUS: A model of process variation and resulting timing errors for microarchitects," IEEE Trans. Semicond. Manuf., vol.21, no.1, pp.3–13, Feb. 2008.

[5] T. Tsunomura, A. Nishida, and T. Hiramoto, "Analysis of NMOS and PMOS difference in VT variation with large-scale DMA-TEG," IEEE Trans. Electron Devices, vol.56, no.9, pp.2073–2080, Sept. 2009.

[6] Stanford Univ. – Open Source Network-on-Chip Router RTL https://nocs.stanford.edu/

[7] Booksim http://sourceforge.net/projects/booksim/

[8] T. Tobita and H. Kasahara, "A standard task graph set for fair evaluation of multiprocessor scheduling algorithms," J. Scheduling, vol.5, pp.379–394, Sept. 2002.

[9] R.P. Dick, D.L. Rhodes, and W. Wolf, "TGFF: Task graphs for free," International Workshop on Hardware/Software Codesign, pp.97–101, March 1998.

[10] J. Chan and S. Parameswaran, "NoCEE: Energy macro-model extraction methodology for network on chip routers," International Conference on Computer-Aided Design (ICCAD), pp.254–259, Nov. 2005.

[11] R. Teodorescu and J. Torrellas, "Variation-aware application scheduling and power management for chip multiprocessors," 35th International Symposium on Computer Architecture (ISCA), pp.363–374, 2008.

[12] W. Dally and B. Towles, Principles and Practices of Interconnection Networks, Morgan Kaufmann, 2004.

[13] A. Kumary, P. Kunduz, A.P. Singhx, L.-S. Pehy, and N.K. Jhay, "A 4.6 Tbits/s 3.6 GHz single-cycle NoC router with a novel switch allocator in 65 nm CMOS," 25th International Conference on Computer Design (ICCD), pp.63–70, 2007.

[14] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos, "Modeling within-die spatial correlation effects for process-design co-optimization," 6th International Symp. on Quality of Electronic Design (ISQED), pp.516–521, 2005.

[15] J. Howard, S. Dighe, S.R. Vangal, G. Ruhl, N. Borkar, S. Jain, V. Erraguntla, M. Konow, M. Riepen, M. Gries, G. Droege, T. Lund-Larsen, S. Steibl, S. Borkar, V.K. De, and R. Van Der Wijngaart, "A 48-Core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFS for performance and power scaling," IEEE J. Solid-State Circuits, vol.46, no.1, pp.173–183, Jan. 2011.

[16] D.N. Truong, W.H. Cheng, T. Mohsenin, Yu Zhiyi, A.T. Jacobson, G. Landge, M.J. Meeuwsen, C. Watnik, A.T. Tran, Xiao Zhibin, E.W. Work, J.W. Webb, P.V. Mejia, and B.M. Baas, "A 167-processor computational platform in 65 nm CMOS," IEEE J. Solid-State Circuits, vol.44, no.4, pp.1130–1144, 2009.

[17] Y. Nakata, Y. Takeuchi, H. Kawaguchi, and M. Yoshimoto, "A process-variation-adaptive network-on-chip with variable-cycle routers," 14th Euromicro Conference on Digital System Design (DSD), Aug. 2011.

[18] G.-M. Chiu, "The odd-even turn model for adaptive routing," IEEE Trans. Parallel Distrib. Syst., vol.11, pp.729–738, July 2000.

[19] W.J. Dally and C.L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," IEEE Trans. Comput., vol.C-36, no.5, pp.547–553, May 1987.

[20] T. Nesson and S.L. Johnsson, "ROMM routing: A class of efficient minimal routing algorithms," Proc. First International Workshop on Parallel Computer Routing and Communication, pp.185–199, 1994.

[21] D. Seo, A. Ali, W.T. Lim, and N. Rafique, "Near-optimal worst-case throughput routing for two-dimensional mesh networks," International Symposium on Computer Architecture (ISCA), pp.432–443, June 2005.

[22] M.L. Fulgham and L. Snyder, "Performance of Chaos and Oblivious Routers under Non-Uniform Traffic," Technical Report UW-CSE-93-06-01, Univ. of Washington, July 1993.

[23] E.G. Coffman, Computer and Job-shop Scheduling Theory, John Willey & Sons, 1976.

Yohei Nakata received B.E. and M.E. degrees in Computer and Systems Engineering from Kobe University, Hyogo, Japan in 2008 and 2010, respectively, where he is currently pursuing a Ph.D. degree in engineering. His current research interests include dependable and variation-aware processor designs and multi-core processor architecture. He is a student member of IPSJ and IEEE.

Hiroshi Kawaguchi received the B.E. and M.E. degrees in electronic engineering from Chiba University, Chiba, Japan, in 1991 and 1993, respectively, and the Ph.D. degree in engineering from the University of Tokyo, Tokyo, Japan, in 2006. He joined Konami Corporation, Kobe, Japan, in 1993, where he developed arcade entertainment systems. He moved to the Institute of Industrial Science, the University of Tokyo, as a Technical Associate in 1996, and was appointed a Research Associate in 2003. In 2005, he move to the Department of Computer and Systems Engineering, Kobe University, Kobe, Japan, as a Research Associate. Since 2007, he has been an Associate Professor with the Department of Computer Science and Systems Engineering, Kobe University. He is also a Collaborative Researcher with the Institute of Industrial Science, the University of Tokyo. His current research interests include low-power VLSI design, hardware design for wireless sensor network, and recognition processor. Dr. Kawaguchi was a recipient of the IEEE ISSCC 2004 Takuo Sugano Outstanding Paper Award and the IEEE Kansai Section 2006 Gold Award. He has served as a Program Committee Member for IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips), and as a Guest Associate Editor of IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. He is a member of the IEEE and ACM.

Masahiko Yoshimoto received the B.S. degree in electronic engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1975, and the M.S. degree in electronic engineering from Nagoya University, Nagoya, Japan, in 1977. He received a Ph.D. degree in Electrical Engineering from Nagoya University, Nagoya, Japan in 1998. He joined the LSI Laboratory, Mitsubishi Electric Corp., Itami, Japan, in April 1977. From 1978 to 1983 he was engaged in the design of NMOS and CMOS static RAM including a 64 K full CMOS RAM with the world's first divided-word-line structure. From 1984, he was involved in research and development of multimedia ULSI systems for digital broadcasting and digital communication systems based on MPEG2 and MPEG4 Codec LSI core technology. Since 2000, he has been a Professor of the Dept. of Electrical and Electronic Systems Engineering at Kanazawa University, Japan. Since 2004, he has been a Professor of the Dept. of Computer and Systems Engineering at Kobe University, Japan. His current activity is focused on research and development of multimedia and ubiquitous media VLSI systems including an ultra-low-power image compression processor and a low power wireless interface circuit. He holds 70 registered patents. He served on the Program Committee of the IEEE International Solid State Circuit Conference from 1991 to 1993. In addition, he has served as a Guest Editor for special issues on Low-Power System LSI, IP, and Related Technologies of IEICE Transactions in 2004. He received the R&D100 awards from R&D Magazine for development of the DISP and development of a realtime MPEG2 video encoder chipset in 1990 and 1996, respectively.