

# 優先度逆転を低減させる 優先度付きオンチップネットワーク向け ルータアーキテクチャ

谷口 将一<sup>1,a)</sup> 山崎 大輝<sup>1</sup> 笹川 雄二郎<sup>1</sup> 松谷 宏紀<sup>1</sup> 山崎 信行<sup>1</sup>

受付日 2012年11月5日, 採録日 2013年4月5日

**概要:** 半導体集積技術の飛躍的な進歩にともない, Chip multiprocessor (CMP) に搭載されるコア数が増大しており, 従来のバス等の接続方式ではチップ内の通信要求を満たすのは困難となってきた。そのため, 大規模な CMP ではコア数の増加に対して高いスケーラビリティを持つ Network-on-Chip (NoC) が用いられる。CMP におけるリアルタイム性や Quality of Service (QoS) のサポートの要求に対しては, 優先度付きの packets を扱える NoC (優先度付き NoC) が用いられる。優先度付き NoC では, 低負荷時には優先度に基づいた制御が正確に行われるが, 高負荷時には低優先度 packets が高優先度 packets をブロックするという問題 (優先度逆転問題) が発生し, 低優先度 packets が高優先度 packets の転送遅延を増加させてしまう。本論文では, 優先度逆転を低減させる優先度継承手法と Virtual Channel (VC) 奪取手法の2種類の手法を提案し, ハードウェアコストとネットワーク性能の評価を行った。優先度継承手法はハードウェアコストは低く抑えられたが, 優先度逆転をほとんど低減されなかった。一方, VC 奪取手法ではオーバーヘッドは大きいものの, 優先度逆転を大きく低減され, 最高優先度 packets の転送性能を向上させた。

キーワード: オンチップネットワーク, 優先度逆転

## Router Microarchitectures for Reducing Priority Inversions in Priority-aware On-chip-networks

MASAKAZU TANIGUCHI<sup>1,a)</sup> DAIKI YAMAZAKI<sup>1</sup> YUJIRO SASAGAWA<sup>1</sup> HIROKI MATSUTANI<sup>1</sup>  
NOBUYUKI YAMASAKI<sup>1</sup>

Received: November 5, 2012, Accepted: April 5, 2013

**Abstract:** With advances in fabrication technology, the number of cores on chip multi-processors (CMPs) increases. And conventional interconnection fabrics, such as buses, cannot provide enough scalability for large-scale CMPs. Network-on-Chips (NoCs) are widely used as scalable interconnects for large-scale CMPs. Priority-aware NoCs that handle packets based on their priority are required for supporting Quality of Service (QoS) and guaranteeing real-time constraints on CMPs. Although packets are processed exactly with their priority for low workload, priority inversions that block high-priority packets with low-priority ones are introduced for high workload. In this case, the communication latency of higher-priority packets is increased by lower-priority ones. In this paper, Priority Inheritance (PI) method and Virtual Channel Stealing (VCS) method are proposed for reducing priority inversions and evaluated in terms of hardware cost and network performance. Although hardware overhead of the PI method is small, it does not avoid priority inversions in most cases. On the other hand, although the VCS method introduces larger hardware overhead, it improves the communication latency of highest-priority packets and reduces priority inversions.

**Keywords:** on-chip network, priority inversion

## 1. はじめに

動作周波数や実行効率の向上によるプロセッサの性能向上は限界を迎え、近年では増大したトランジスタ数を利用して、プロセッシングコアの複製によりスレッドレベル並列性を抽出する手法が有効なプロセッサ性能の向上手法とされている。単一チップ上に複数のコアを搭載する Chip multiprocessor (CMP) [1], [2] が近年における主要なプロセッサとなりつつある。利用可能なトランジスタ数の増大にともない、CMP の規模は拡大しており、近年では大規模な CMP の研究開発が活発に行われている [3], [4], [5], [6].

CMP の大規模化にともない、コア間の通信量が増大し、コアどうしの通信性能がプロセッサ性能に与える影響が大きくなってきている。特に数十以上のコアの相互接続が必要な CMP において、従来のチップ内通信に用いられてきたバスやクロスバ方式による接続では、バンド幅や転送遅延、面積の観点から CMP の要求を満たせなくなっている\*1。そこで、多数のノードを接続する効率的な接続方式であるパケット交換方式の Network-on-Chip (NoC) [8], [9], [10] が大規模な CMP におけるコア間の接続方式として有効とされている。NoC はチップ内に結合網を敷き、ルータを介して共有された伝送路を利用することで、効率的なノード間の接続を提供する方式であり、現在、その面積 [11] や電力 [12], [13], [14], [15], 転送遅延 [16], [17], [18], スループット [19], [20], [21] 等を向上させる様々な研究が行われている。

リアルタイム性や QoS の保証を行うシステムにおいても、CMP の利用が広まってきている。そのような CMP で NoC を利用するため [22], 優先度付きのパケットを扱う NoC (優先度付き NoC) が必要とされている。優先度付き NoC の重要性は高まっており、QoS をサポートする NoC アーキテクチャ [23], [24], [25], [26], [27] やシステム性能にクリティカルなパケットに高い優先度を与えることでシステム全体の性能を向上させる NoC アーキテクチャ [28], [29] 等の優先度付きのデータを扱う研究が数多く行われている。優先度付き NoC では、バッファやチャネル等の多くのリソースが共有して利用されているため、低優先度パケットがそれらの共有リソースを占有してしまい、高優先度のパケットがそれらのリソースを獲得できずにブロックされてしまうという状況が生じうる。高優先度パケットが低優先度パケットにブロックされてしまうという問題は優先度逆転問題と呼ばれる [30]。時間制約の保証と優先度制御が必須であるリアルタイムシステムにおいて、優先度に反した挙動は、アプリケーションにおいて、リソーススタベーションや意図しない動作 (ウォッチドッグタイマによるシ

ステム全体のリセット等) を引き起こす可能性がある。本研究の目的は、オンチップネットワークを用いたリアルタイム通信においてこのような優先度逆転問題を低減させることである。

本論文では、優先度継承手法と Virtual Channel (VC) 奪取手法というそれぞれ異なったアプローチの 2 種類の手法を提案し、優先度逆転問題を低減させ、高優先度パケットの転送性能の向上を図る。前者の手法は CPU のスケジューリングアルゴリズムに用いられる優先度継承手法をオンチップネットワーク向けに実装したものであり、後者の手法はあるパケットが使用中の VC の空きスロットを、他の高優先度パケットが間借りする形で利用する手法である。

本論文の構成は次のとおりである。2 章で優先度付き NoC の現状とその問題について述べ、3 章では優先度逆転を低減させる 2 手法を提案し、それらのオンチップルータへの実装について述べる。続いて、4 章で論理合成とネットワークシミュレーションによる評価および考察を行い、5 章で本論文の結論について述べる。

## 2. 優先度付き NoC

### 2.1 ルータアーキテクチャ

優先度付き NoC では、パケットの優先度制御を行うため、通常の NoC とは異なるルータアーキテクチャとなる。以降では、通常のルータアーキテクチャと優先度付き NoC のルータアーキテクチャに関しての説明を行う。

#### 2.1.1 通常の NoC のルータアーキテクチャ

標準的な NoC に用いられるワームホール方式のオンチップルータアーキテクチャは図 1 に示す構成をとり、入力ポート (図の Input) からパケットが到着し、パケットのヘッダ情報に基づいた制御が行われた後に出力ポート (図の Output) から出力される。以下に図 1 のルータの詳細な動作を説明する。

ルータにフリット\*2が到着すると、フリットは Input Unit 内にある Virtual Channel (VC) が持つバッファキューに格納され、同時に Routing Unit による経路計算 (出力ポートの決定) が行われる。続いて、出力ポート先のルータの VC (出力 VC) を獲得するために VC Allocator へリクエストを出す。出力 VC が割り当てられた場合、続けて、クロスバを獲得するために Switch Allocator へリクエストを出す。Switch Allocator によりクロスバが割り当てられると、クロスバを経由して出力ポートへ転送され、リンクを通過して次のルータへ送信される。

\*1 数十コア程度の CMP ならばバス方式の改良により要求を満たすことも可能であるが [7], スケーラビリティの低さから、数百コアを超える CMP ではバス方式による接続は限界である。

\*2 オンチップルータでは、パケットは複数のフリットと呼ばれる単位で構成され、パケットの転送はフリット単位で行われる。

<sup>1</sup> 慶應義塾大学  
Keio University, Yokohama, Kanagawa 223-8522, Japan  
a) masakazu@ny.ics.keio.ac.jp

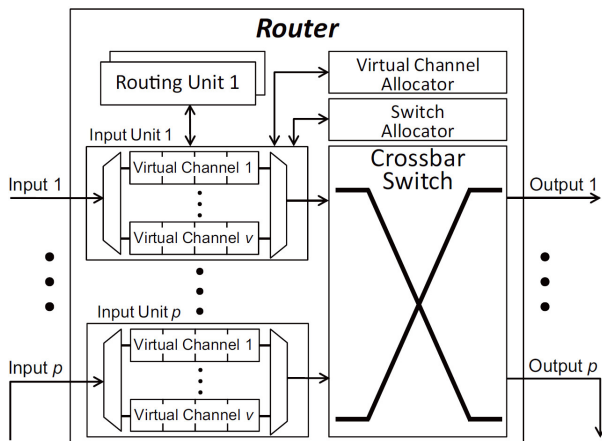


図 1 通常の NoC のルータアーキテクチャ  
Fig. 1 General router architecture.

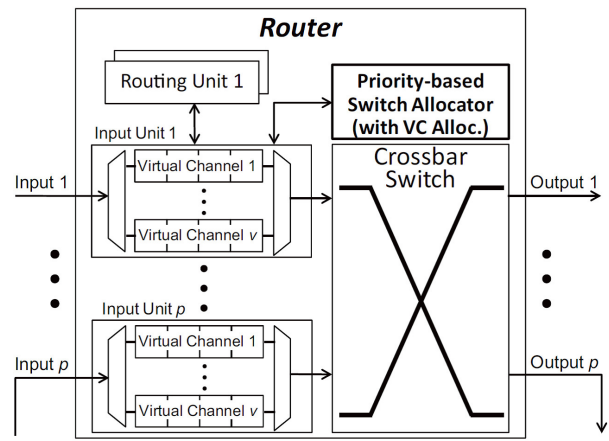


図 2 優先度付き NoC のルータアーキテクチャ  
Fig. 2 Router architecture for a priority-aware NoC.

### 2.1.2 優先度付き NoC のルータアーキテクチャ

優先度付き NoC では、パケットの優先度に基づいた資源割当てを行う必要があり、通常の NoC のルータアーキテクチャとは異なる設計が要求される。本研究では、ベースラインルータとして優先度付き NoC 向けに設計された VIX [31] ルータを用いる。VIX ルータのアーキテクチャを図 2 に示す。VIX ルータは、優先度アービタを用いて優先度をサポートするオンチップルータである。VIX ルータでは、投機的にクロスバの割当て、VC の割当てを同時実行するのではなく、同じパイプラインステージでクロスバの割当て、仮想チャネルの割当ての順で逐次的に割当てを行う。VC の獲得の有無にかかわらずすべてのパケットがクロスバ割当てのリクエストを出し、グラントを得たパケットのヘッダフリットがクロスバに進み、次のサイクルで隣接ルータに転送される。このとき、パケットが VC を獲得していなければ、クロスバ割当てのグラントをトリガに VC を獲得する。反対に、リクエストを出そうとする出力チャンネルに利用可能な VC がない場合は、リクエストを出さないことでクロスバのマッチング効率の低下を防ぐ。VIX ルータは各出力チャンネルに対して 1 サイクルで最大 1 つの VC しか割り当てないため、VC の割当ての論理はクロスバの割当ての結果を利用することができ、VC の割当ての論理を大幅に削減することができる。また、VIX ルータではクロスバが割り当てられたパケットのみが VC を獲得できるため、VC を浪費しにくくなり、バーチャルカッスルー方式よりもワームホール方式の方が広く使われるオンチップネットワークに適している。これらの理由から、本研究では VIX ルータをベースラインとして用いる。なお、同一入力ポート内での仮想チャネルの割り当て方法として、優先度ごとに VC を分離するという使い方も考えられるが、VIX および本論文では、本格的な優先度制御のため、8-bit (256 階層) の優先度をサポートすることを目標にしているため、256 階層の優先度ごとに VC を分離する

という使い方は現実的でない。VIX、および、本論文では、VC やクロスバの調停はパケットの優先度に従うものの、空いている VC があればパケットの優先度によらず利用できる。また、VIX ルータのフロー制御はオンチップルータで広く用いられている、クレジットベース [32] のフロー制御を用いている。

### 2.2 優先度逆転問題

優先度逆転は全パケット間で共有の資源を低優先度パケットが占有し、高優先度パケットの資源獲得をブロックすることにより発生する。優先度逆転の発生例を図 3 に示す。ルータ 0 に最高優先度パケットである優先度 3 のパケットがあり、ルータ 1 に進行しようとしているが、ルータ 1 のルータ 0 と接続されている入力ユニットは優先度 0 のパケットらにより VC が占有されてしまっている。結果として、優先度 0 のパケットが自身より優先度の高い優先度 3 のパケットをブロックしてしまい、優先度の逆転が生じてしまっている。さらに、優先度 0 のパケットらは、経路の競合により別の入力ユニット内の優先度 1 のパケットにブロックされており、間接的に優先度 1 のパケットが優先度 3 のパケットをブロックし、ここでも優先度の逆転が生じている。これらの優先度逆転の結果、最高優先度である優先度 3 のパケットは、設計者の意図に背き大きな遅延をともなって転送される。この優先度逆転の発生により、パケットの優先度に対して不適当な遅延が発生し、設計者の予想外の振舞いをシステムがしてしまう可能性がある。本研究では、このように下流ルータの低優先度パケットが隣接する上流ルータの高優先度パケットをブロックしている状態を優先度逆転と定義し、この状態が発生しているサイクル数を優先度逆転の発生回数と定義する。

### 2.3 優先度逆転の低減手法

スケジューリングにおける優先度逆転の一般的な解決策として、優先度継承プロトコルや優先度上限プロトコルが

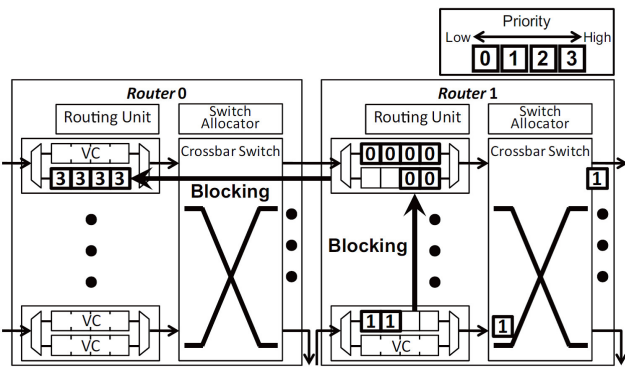


図 3 優先度逆転の発生

Fig. 3 Occurrence of priority inversion.

用いられるが、NoCでは共有資源の扱いがスケジューリングとは異なるため、これらの手法を適用できない。しかしながら、QoSをサポートするNoCでは優先度逆転に対処する必要があり、それらの研究ではパケットのドロップや専用のVCの使用等の手法で優先度逆転を解決している。

文献 [33] で提案されている優先度先送り方式では、高優先度パケットがブロックされているという情報を、ブロックが実際に発生しているところまで先送りすることで、網における優先度継承を実現している。優先度のサポートと高い絶対性能を実現する。しかし、このオフチップルータはスイッチング方式として、オンチップネットワークではあまり用いられないバーチャルカットスルー方式を前提としている。さらに、高優先度パケットがブロックされているという情報をブロックが実際に発生しているところまで先送りする機能をネットワーク全体でサポートする必要がある。

オンチップネットワーク向けのQoSをサポートする手法としてPreemptive Virtual Clock (PVC) [25] が提案されている。PVCはパケットのプリエンプション(パケットのドロップ)により優先度逆転を解消し、QoSのサポートを行う。高い公平性を持つバンド幅の保証を行うNoCアーキテクチャであるが、パケットのドロップをサポートするために専用のACKネットワークや再送バッファを要し、さらに各ルータの入力ポートでそのポートを経由するすべてのフローの状態レジスタを用意する必要があるため、きわめて高いハードウェアオーバーヘッドが生じる。

PVCとは異なるアプローチで優先度逆転を解消するNoCアーキテクチャとして、Globally-Synchronized Frames (GSF) [26] があげられる。GSFはフレームベースのQoSのサポート [34], [35] を行うNoCアーキテクチャであり、最高優先度のフレームに対して専用のVCを割り当てることで優先度逆転を解消している。厳しいバンド幅の保証を行う一方で、パケットの注入を制限するための巨大なバッファキューの実装によりハードウェアコストが増大する。

既存の研究で用いられる優先度逆転の対処手法は優先度

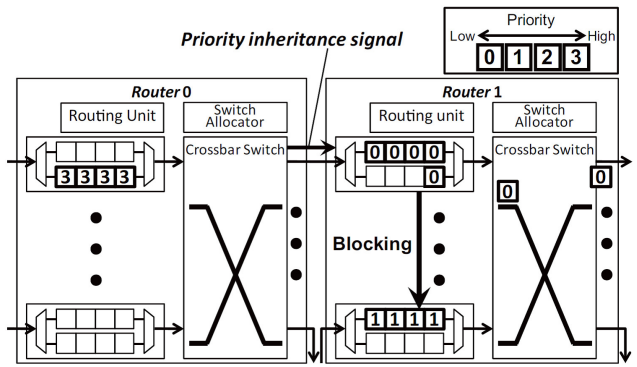


図 4 優先度継承ルータ

Fig. 4 Priority Inheritance router.

逆転の解消の代償として、きわめて高いハードウェアオーバーヘッドが生じる。本研究の提案手法は、現実的なハードウェアコストで、性能を犠牲にせずに優先度逆転を低減させる。さらに、既存の研究とは異なり、ルータレベルのアーキテクチャを提案することで、ネットワークトポロジに依存しない汎用性の高い手法となっている。

### 3. 優先度逆転を低減させるルータアーキテクチャ

本研究では2種類の優先度逆転の低減手法を提案する。1つ目の手法として、スケジューリングにおける優先度継承のアイデアを参考に優先度継承ルータを提案し、2つ目に、まったく異なるアプローチで優先度逆転を低減させるVC奪取ルータを提案する。

#### 3.1 優先度継承ルータ

優先度継承ルータは、優先度逆転によりブロックされている高優先度パケットの優先度を低優先度パケットに継承させることで、優先度逆転を低減させるルータである。以降では優先度継承ルータの詳細を述べる。

##### 3.1.1 優先度継承手法

優先度継承手法では、優先度逆転が生じた際に、高優先度パケットの優先度情報を低優先度パケットを持つVCに送り、優先度を低優先度パケットに一時的に継承させる。この作用により、高優先度パケットをブロックしている低優先度パケットは高優先度で転送され、VCを解放し、高優先度パケットを進行させる。優先度継承手法の動作例を図4に示す。優先度逆転の発生により、優先度継承シグナルが優先度0のパケットへ送られ、優先度0のパケットは優先度3を継承する。その結果、優先度1のパケットとの出力ポートの競合に勝利し、優先度1のパケットをブロックして優先度0のパケットが転送される。この結果、優先度0の占有していたVCが解放され、優先度3のパケットが進行可能となる。

### 3.1.2 実装

優先度継承手法を実装したオンチップルータである優先度継承ルータは、隣接ルータの優先度情報を基に優先度逆転を判定し、優先度逆転が検出された場合は優先度の継承を行う。以降に実装の詳細を述べる。

#### 3.1.2.1 継承させる優先度情報の転送

ブロックされている高優先度パケット側からブロックしている低優先度パケット側へ優先度情報を転送することで、優先度逆転の検出を行う。まず、出力 VC が獲得できないことでブロックされたパケットは優先度逆転の可能性があると判断して、優先度情報を経路先の出力ポートへ転送する。複数ルータにまたがる優先度逆転に対処するために、優先度継承を行ったパケットが転送する優先度は継承した優先度とする。ここで複数のパケットが同一の出力ポートへの優先度情報の転送を試みた場合は、最高優先度の情報が転送される。

この選択機構は物理チャネル数  $p^3$ 、VC 数  $v$ 、優先度の数  $Priority$ 、に対して、 $O(p^2v^2 \times Priority)$  で面積が増加する<sup>\*4</sup>ため、入力ポート内の選択と入力ポート間の選択の2段に分けて構成する。

#### 3.1.2.2 優先度逆転の検出と継承

各入力ポート内で、転送されてきた優先度が入力ポート内のいずれのパケットの優先度よりも高く、かつ入力ポート内に空き VC が存在しない場合は<sup>\*5</sup>、優先度逆転の発生を検出する。優先度逆転を検出した入力ポートは優先度継承を行い、パケットの優先度ではなく継承した優先度を用いて、クロスバへのリクエストを行う。

#### 3.1.2.3 オーバヘッド

優先度情報の転送のために、ルータ間に優先度ビット幅の信号線を専用に用意する必要があるため、優先度ビット幅分の信号線オーバヘッドが生じる。また、継承している優先度を保持するため、優先度ビット幅 × ポート数分のストレージオーバヘッドが生じる。さらに、継承させる優先度情報の選択、優先度逆転の検出および優先度継承の論理がハードウェアオーバヘッドとして生じる。

#### 3.1.3 優先度逆転の解消にかかるサイクル数

優先度継承手法では、優先度逆転が発生してから解消するまでに最小でも 4 サイクルの遅延を要する。まず、高優先度パケットの優先度情報が出力ポートへ転送され (0 サイクル目)、リンクを通過して隣接ルータへ送られる (1 サ

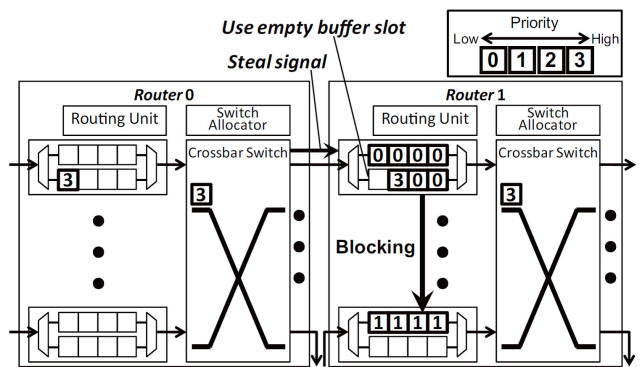


図 5 Virtual Channel (VC) 奪取ルータ  
Fig. 5 Virtual Channel (VC) Stealing router.

イクル目)。続いて、低優先度パケットが到着した優先度を継承してクロスバを獲得し (2 サイクル目)、クロスバが割り当てられたことにより VC の解放がリンクを通過して隣接ルータへ通知される (3 サイクル目)。VC の解放が通知され、優先度逆転が解消される (4 サイクル目)。これは優先度を継承したパケットが、入力ポートにテイルフリットしか残されていない最良のケースであり、実際の解消までの遅延は入力ポートに残されているフリット数に応じてさらに増加する。

### 3.2 Virtual Channel (VC) 奪取ルータ

Virtual Channel (VC) 奪取ルータは、低優先度パケットが使用中の VC を一時的に間借りすることで、VC が解放されるのを待たずに高優先度パケットを進行させる。なお、本手法は、ネットワークポロジにかかわらず適用可能である。以降では、VC 奪取ルータの詳細を述べる。

#### 3.2.1 VC 奪取手法

VC 奪取手法では、優先度逆転が生じた際に、空き VC がない出力ポートへ高優先度パケットを VC 奪取シグナルとともに送り、低優先度パケットの VC の空きバッファスロットを使用して転送を行う。VC 奪取手法の動作例を図 5 に示す。優先度逆転が発生し、かつ出力ポート先に空きバッファが発見されると、高優先度パケット (優先度 3) は VC 奪取シグナルとともに転送される。VC 奪取シグナルを受信した VC は低優先度パケット (優先度 0) を一時的にサスペンドし、高優先度パケットを優先して転送する。この結果、優先度 3 の高優先度パケットは VC が占有されている状況下においても転送することが可能となる。

#### 3.2.2 実装

VC 奪取手法を実装したオンチップルータである VC 奪取ルータは、優先度逆転を検出し、低優先度パケットが使用中の VC の空きバッファを利用して転送を行う。以降に実装の詳細を述べる。

#### 3.2.2.1 優先度逆転の検出

隣接ルータの優先度情報を保持することで、優先度逆転

\*3 パケットがルーティングされるチャネルの数である。たとえば、2D-mesh では東西南北の 4 方向とローカル方向の 1 方向を足して 5、3D-mesh では東西南北上下の 6 方向とローカル方向の 1 方向を足して 7 となる。

\*4 優先度付き選択機構は  $pv$  個の  $v:1$  ローカルアービタと、 $p$  個の  $pv:1$  グローバルアービタで構成される。1 つのアービタの面積が  $O(Entry \times Priority)$  で増加するため、優先度付き選択機構の面積は  $O(p^2v^2 \times Priority)$  で増加する。

\*5 隣接ルータに VC の利用情報が通知されるまでに数サイクルの遅延があるため、VC が空いているにもかかわらず、優先度情報が転送されてくる場合がある。

の検出を行う。隣接ルータの優先度情報の管理は、パケットを出力ポートから送信するたびに、優先度情報を記憶することで実現する。この優先度情報を用い、出力 VC を獲得できないパケットは自身の優先度と出力ポート先のパケットの中で最も高い優先度を比較し、自身の優先度の方が高い場合は優先度逆転の発生を検出する。

なお、優先度継承ルータでは優先度を継承する下流側のルータで優先度が逆転したという情報を使用していたが、VC 奪取ルータでは優先度を継承させる上流側のルータで優先度が逆転したという情報を使用する。そのため、これら 2 つのルータでは異なる優先度検出の方法を用いた。

### 3.2.2.2 VC の奪取

優先度逆転を検出した際に、1 つ以上のバッファに空きがあり、かつ他のパケットにまだ奪取されていない VC が存在する場合、VC の奪取を実施する。VC の奪取を行うパケット（奪取パケット）のクレジットラウンドトリップ遅延 [32] を最小限に抑えるために、複数の VC が奪取候補である場合は最も空きバッファ数が多い VC が選択される。奪取パケットは奪取シグナルとともに転送され、パケットとともに奪取シグナルを受信した入力ユニットは VC に奪取を通知する。奪取された VC は現在使用中のパケット（被奪取パケット）を疑似的なスリープ状態にし、奪取パケットの転送を優先して行う。なお、1 つ以上のバッファに空きがあることを VC の奪取を実施する条件に含めたのは、奪取パケットと被奪取パケットの間でのデッドロックを回避するためである。

ここで、経路先ルータの高優先度パケットがブロックされている場合を考慮し、低優先度パケットが高優先度パケットを奪取する手法も考えるが、その場合、奪取を行ったことによる優先度逆転が発生する。これは、高優先度パケットが転送可能であるにもかかわらず、上流の低優先度パケットにサスペンドされてしまう可能性があるためである。また、あるパケットが同一優先度の他のパケットを奪取する手法も考えるが、奪取パケットが使えるバッファは被奪取パケットが使用している分だけ小さなサイズとなる。そのため、奪取パケットは複数のルータにまたがる可能性が高く、他のパケットをブロックしかねない。そもそも本研究の目的は優先度逆転の低減であり、低優先度パケットのスループットの向上ではないため、本研究では高優先度パケットが低優先度パケットを奪取するのみとする。

### 3.2.2.3 バッファの共有

通常のルータで用いられるバッファはパラレル FIFO 構造 [36] になっており、読み出し先のバッファや書き込み先のバッファを指すポインタが読み出しや書き込みのたびにインクリメントされることで FIFO を実現しているが、VC 奪取手法では 1 つの VC に 2 つのパケットが混在するため、FIFO 構造を持つ機構は適さない。そのため、VC 奪

取ルータでは共有バッファ方式を採用する。パケットごとの読み出し用と書き込み用のポインタテーブルを用意することで、バッファ上に混在したフリットにアクセスが可能となる。

### 3.2.2.4 オーバヘッド

各出力ポート先の優先度情報を管理するために、各出力ポートごとに優先度ビット幅  $\times$  出力 VC 数  $\times 2$ （奪取パケットと被奪取パケット）ビットのストレージオーバヘッドとそれらの優先度から最高優先度を選択するマルチプレクサおよび優先度アービタが必要となる。また、すべての VC 内のバッファの共有バッファへの置き換えや VC の管理機構の変更により、VC の複雑化が生じる。

### 3.2.3 優先度逆転の低減効果

奪取パケットの転送性能は奪取先の VC の空きバッファ数に依存し、バッファ数が少ないほどクレジットラウンドトリップ遅延の影響を大きく受け、実質的に優先度逆転による転送遅延の低下を被る。また、空きバッファがまったくない場合には VC の奪取は行えない。空きバッファ数への依存度が高い一方で、優先度逆転の発生から、その解消までに遅延が生じず、高い優先度逆転の低減効果が得られる。

## 4. 評価

### 4.1 評価環境および評価指標

本研究では、VerilogHDL を用いてベースラインルータ、優先度継承ルータおよび VC 奪取ルータを実装し、Cadence 社の NC-Verilog を用いてネットワークシミュレーションを行い、Synopsys 社の Design Compiler を用いて論理合成を行った。論理合成では、面積と動作周波数を評価し、ネットワークシミュレーションでは、優先度逆転の発生数、平均転送遅延、ジッタおよび最大転送遅延を評価した。転送遅延と無負荷時遅延の差の標準偏差をジッタとし、シミュレーション期間中で最も遅延が大きかったパケットの転送遅延を最大転送遅延とした。ネットワークシミュレーションは 10,000 サイクルのウォームアップ期間を用いて 100,000 サイクルの期間で行った。

シミュレーションに用いたパラメータを表 1 に示す。

プロセスは TSMC の 90nm のテクノロジーを用い、トポロジは  $8 \times 8$  の 2 次元メッシュトポロジを採用した。トラフィックパターンは Uniform random<sup>\*6</sup>、Transpose<sup>\*7</sup>、Bit complement<sup>\*8</sup> の 3 通りを用い、VC 数を 2, 4 の 2 通りに分けてシミュレーションを行った。パケットは 5 フリット構成として、128 ビットのフリットサイズとした<sup>\*9</sup>。

\*6 各ソースノードがディスティネーションノードを完全にランダムに選択。

\*7 行列の転置と同様、X 座標と Y 座標を入れ替えた座標のノードにパケットを送信。

\*8 ソースノードの座標をビット反転させた座標のノードにパケットを送信。

\*9 一般的なキャッシュブロックのサイズを想定した。

表 1 シミュレーションパラメータ  
Table 1 Simulation parameter.

Technology	TSMC 90 nm
Topology	8-ary 2-mesh
Traffic pattern	Uniform random / Transpose / Bit complement
Routing algorithm	Dimension-ordered
Pipeline latency	3 cycles
Packet priority	8 bits (4 bits are used)
Packet size	5 flits
Flit size	128 bits
Number of VCs per port	2, 4
Number of buffers per VC	4 flits

また、オンチップネットワークではバッファ面積の制限からバーチャルカットスルー方式よりもワームホール方式の方が広く使われているため、バッファサイズは4フリットとした。ネットワークシミュレーションには4ビット（16レベル）の優先度を用い、論理合成には8ビットの優先度を用いた。ネットワークシミュレーション時の優先度を4ビットに制限したのは、各優先度のパケットの数を相対的に増加させることで、それぞれのレイテンシの平均値を安定させるためである。また、各パケットへは一様ランダムに優先度を割り振り、最高優先度パケット（優先度15）と最低優先度パケット（優先度0）の2種類のパケットに関して評価した。フロー制御はワームホール方式で行った。また、VCの割当てはすべてのルータで、VIXルータ[31]の割当てアルゴリズムを用い、同一優先度のアービトレーションにはmatrixアービタ\*10を用いた。

#### 4.2 論理合成結果

論理合成後の面積を図6(a)に示す。各グラフにおいて、Baselineはベースラインルータを、PI(Priority Inheritance)は優先度継承ルータを、VCS(Virtual Channel Stealing)はVC奪取ルータをそれぞれ示す。

優先度継承ルータの面積はVC=2, VC=4の場合に対してそれぞれ5.1%, 5.3%の増加となり、VC奪取ルータの面積はVC=2, VC=4の場合に対してそれぞれ17.7%, 20.5%の増加となった。いずれのルータもVC数を2倍にしても同程度の面積増加率となっており、追加の論理は複数のVC数において小さな増加率となった。ルータ間の比較では、優先度継承ルータは優先度の継承に関する優先度情報の比較論理の増加にとどまるのに対し、VC奪取ルータはVCの奪取等の論理に加え、優先度情報を格納するバッファやそれを扱う優先度アービタやマルチプレクサが面積を圧迫したため、優先度継承ルータと比べて大きな面積オーバーヘッドとなった。

続いて、論理合成後の動作周波数を図6(b)に示す。

\*10 LRUポリシーに基づいたアービトレーションを行う。

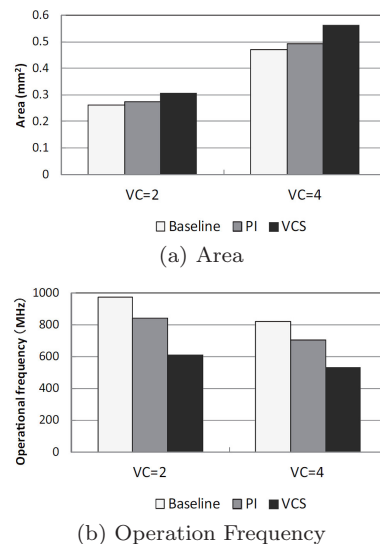


図 6 ルータの論理合成結果  
Fig. 6 Results from logic synthesis of routers.

ベースラインルータと比較した場合、優先度継承ルータの動作周波数はVC=2, VC=4の場合に対してそれぞれ13.4%, 14.1%の低下を示し、VC奪取ルータの動作周波数はVC=2, VC=4の場合に対してそれぞれ36.8%, 34.8%の低下を示した。面積と同様に、いずれのルータも、ベースラインルータと比較した場合、VCの本数によらずに一定の動作周波数の低下率であり、複数のVC数において小さな低下率となった。優先度継承ルータでは優先度比較論理の追加やアービタサイズの拡大により、平均で13.8%の低下であったが、VC奪取ルータは優先度逆転の検出のために、従来のクリティカルパスに直列に優先度比較論理が追加されたため、平均で35.8%と優先度継承ルータの2.6倍の動作周波数の低下を示した。一般的に、信頼性とハードウェア量の間にはトレードオフの関係がある。35.8%の動作周波数の低下にともなうスループット低下は、フリット幅を大きくする等、物量を増やすことで十分に補償可能であり、ウォッチドッグタイマによるシステム全体のリセット等の致命的なエラーを考えると十分に影響が少ない。本提案は実用的なハードウェアコストで優先度逆転を低減させることができ、優先度逆転にともなうリスクを低減させることができる。

#### 4.3 ネットワークシミュレーション結果

以下ではネットワークシミュレーションの結果について述べる。

##### 4.3.1 優先度逆転の発生量

Uniform random, Transpose, Bit complement トラフィックにおけるVCが2本の場合の優先度逆転の発生回数を図7(a), 図7(b), 図7(c)に示す。横軸はパケットのネットワークへの注入率(トラフィック)を示し、縦軸は優先度逆転の発生回数を示している。

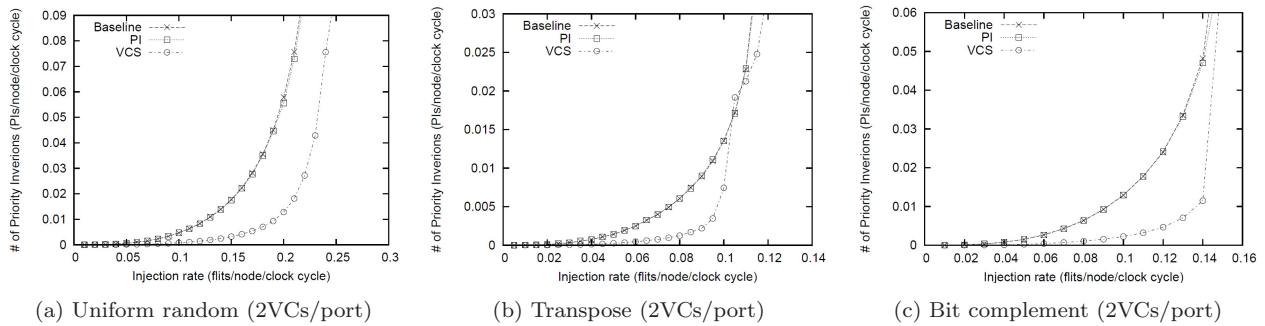


図 7 優先度逆転の発生回数

Fig. 7 Number of priority inversions.

優先度継承ルータは、いずれのトラフィック、いずれの負荷においても優先度逆転の低減効果が見受けられない。優先度継承ルータは VC を獲得するまでに、VC を追い出す遅延が生じるため、低負荷時には低減効果が見込みにくく、また、高負荷時には継承先のパケットの進行先の VC も満杯になっているケースが頻発するためである。また、継承先パケットのテイルフリットが到着していない場合、優先度継承してもテイルフリットは他のルータにあるため、VC が空かず、継承元パケットはいつまでも VC を獲得できなくなる。これを回避するために、継承先パケットのテイルフリットに優先度を継承させることも考えられる。しかし、その実現にはパケットの流れとは逆方向への信号線が必要であったり、パケットの流れに対して順方向と逆方向、両方からの優先度を継承して比較するためのきわめて複雑な制御論理が必要であったりする等、面積、消費電力および動作周波数の観点から現実的ではない。

優先度継承ルータの低減効果が上がっていない一方で、VC 奪取ルータは優先度逆転を低減させており、Uniform random トラフィック、Bit complement トラフィックにおいては負荷の大きさに関係なく優先度逆転の低減効果が見受けられる。Transpose トラフィックにおいてのみ高負荷時に低減効果が見受けられなくなる。これは、任意の時刻を見たときに、同時に奪取している、または、されているパケットの数が多くなり、これらのパケットの間ではさらなる奪取は行えないため、奪取しているパケット間での優先度逆転が頻発したものと考えられる。Transpose トラフィックはパケットの経路の競合が激しいため、VC の奪取が頻発し、同時に奪取しているパケットの数を増加させる。

また、いずれのトラフィックにおいても負荷が大きくなるにつれて、VC 奪取ルータの優先度逆転問題の低減の効果が大きくなっている。これは、負荷が大きいくほど優先度逆転の発生頻度が増加するため、効果的に優先度逆転を低減させることができる奪取ルータの効果が相対的に大きく現れたものである。

#### 4.3.2 ネットワーク性能

Uniform random トラフィックにおける VC が 2 本の場合の最高優先度パケット（優先度 15）と最低優先度パケッ

ト（優先度 0）の 2 種類のパケットの平均転送遅延、ジッタおよび最大転送遅延の評価を図 8 (a), 図 8 (b), 図 8 (c) に示す。横軸はパケットのネットワークへの注入率（トラフィック）を示し、縦軸はそれぞれの評価項目を示している。転送遅延と無負荷時遅延の差の標準偏差をジッタとし、シミュレーション期間中で最も遅延が大きかったパケットの転送遅延を最大転送遅延とした。

平均転送遅延に関して、優先度継承ルータではベースラインルータとの差があまり見受けられないのに対し、VC 奪取ルータでは最高優先度パケットの転送遅延が低下し、最低優先度パケットの転送遅延が増加した。ジッタの評価では、最高優先度パケットのジッタが減少し、最低優先度パケットのジッタが増大した。そして、最大転送遅延の評価では、すべてのパケット注入レートで最高優先度パケットの最大転送遅延が低下している。これらの結果は、優先度逆転を低減した効果による。

続いて、VC の本数を 4 本に増加させた場合のネットワーク評価を図 8 (d), 図 8 (e), 図 8 (f) に示す。いずれの評価においても、VC が 4 本の場合では高優先度パケットの転送性能の向上率が大幅に減少している。これは、VC の本数が増加したことにより、高優先度パケットが VC を獲得しやすくなったため、優先度逆転の発生頻度が減少し、優先度逆転の低減機会が減少したためである。

VC の本数が 2 本の場合の Transpose トラフィックと Bit complement トラフィックの評価を図 9 に示す。Uniform random トラフィックと同様に、優先度継承ルータは低減効果を発揮していない。一方で、VC 奪取ルータは、転送遅延、ジッタの評価において、飽和するまですべての Injection rate で低優先度パケットの性能が低下し、高優先度パケットの性能が向上している。最大転送遅延については Transpose では Injection rate が 0.07 以下で、Bit Complement では Injection rate が増大するにつれて高優先度パケットの性能が向上している。トラフィックによらず、VC 奪取ルータの性能が向上していることが分かる。

#### 4.4 既存の研究とのストレージオーバーヘッドの比較

既存のオンチップネットワーク向けの優先度逆転を低減



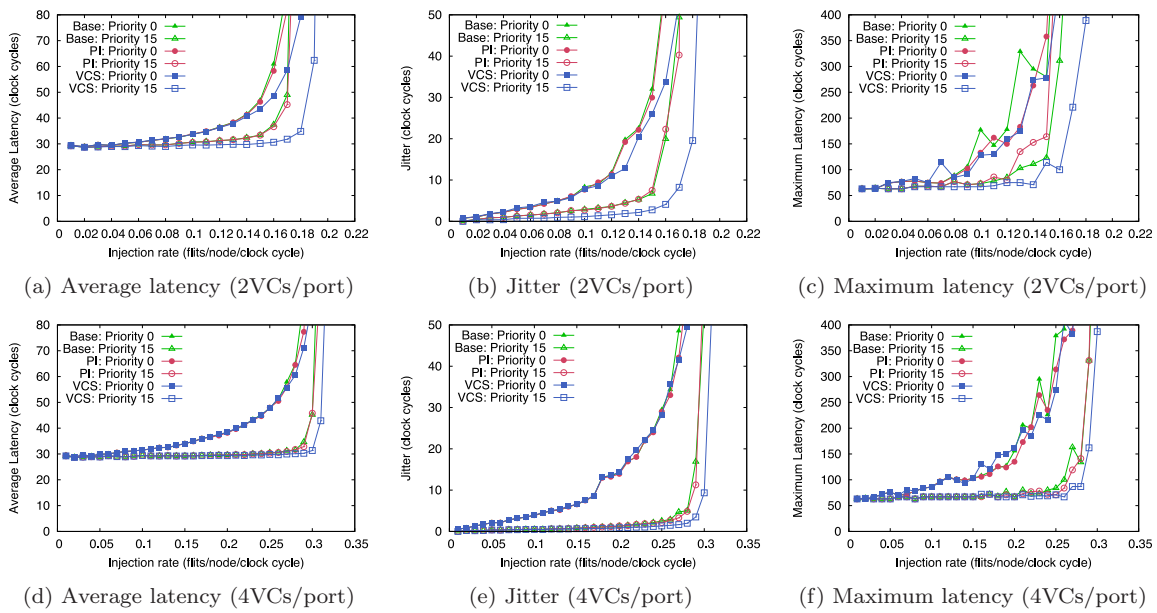


図 8 Uniform random トラフィック

Fig. 8 Uniform random traffic.

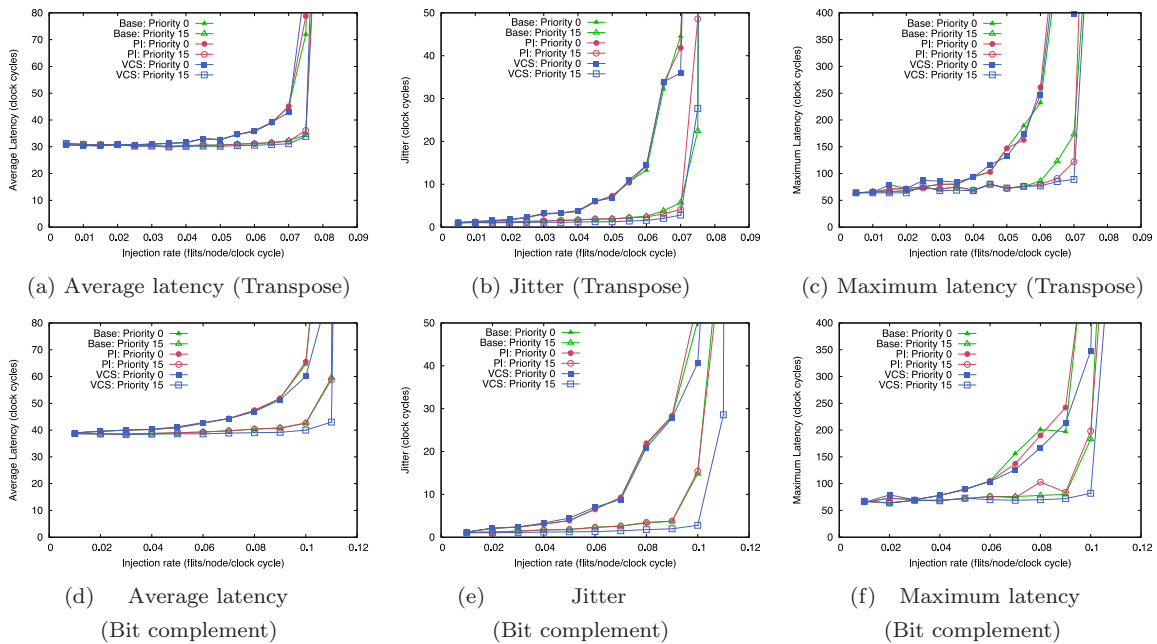


図 9 Transpose トラフィックと Bit Complement トラフィック (2VCs/port)

Fig. 9 Transpose and Bit complement traffic (2VCs/port).

させる手法と、優先度継承ルータ、VC 奪取ルータのハードウェアオーバーヘッドを比較するため、その近似値として、それぞれのストレージ容量を表 2 に示す。なお、No QoS は文献 [25] でベースラインとして使われている QoS をサポートしないルータ [37] である。文献 [25] 同様、簡単のため、バッファを制御するための論理と、各ノードのネットワークインタフェースのバッファは除いている。既存の手法のオーバーヘッドは文献 [25] を参考にした。優先度継承ルータのオーバーヘッドは、継承させる優先度を隣接ルータに伝えるために各出力ポートに取り付けたバッファオーバーヘッドと、継承中の優先度を保持するために各入力ポ

ートに付けたバッファオーバーヘッドを含んでいる。VC 奪取ルータのオーバーヘッドは、FIFO を共有バッファに置き換えたオーバーヘッドと、各出力ポート先の優先度情報を管理するためのバッファオーバーヘッドを含んでいる。

既存の手法である GSF、PVC のストレージオーバーヘッドはそれぞれ 16,700%, 80% と大変大きく、どれだけ優先度の逆転が抑制できたとしても、実際に使用するのは現実的でない。一方、本研究の提案手法である優先度継承ルータ、VC 奪取ルータのストレージオーバーヘッドはそれぞれ 0.4%, 9.3% と非常に小さく、提案手法は、面積効率の良い実用的な手法であることが分かる。

表 2 ストレージオーバーヘッドの比較  
Table 2 Comparison of storage overhead

手法	ストレージ容量 [bytes]	No QoS との差 [bytes]	相対オーバーヘッド [%]
No QoS	1,920	0	0
GSF	33,920	32,000	16,700
PVC	3,376	1,456	80
優先度継承ルータ	1,928	8	0.4
VC 奪取ルータ	2,100	180	9.3

## 5. 結論

優先度逆転問題は、時間制約の保証と優先度制御が必須であるリアルタイムシステムにおいて、リソーススタベーション等、意図しない動作を引き起こす可能性があり、対処が必要である。本論文では、優先度逆転問題を軽減する優先度継承手法と VC 奪取手法の 2 種類の手法を提案し、それらの手法を優先度付き NoC 向けのオンチップルータへ実装した。そして、3 通りのトラフィックパターンを用い、VC 数を 2 通りに分け、ネットワークシミュレーションによる評価を行った。評価の結果、優先度継承手法では優先度逆転軽減の効果は現れなかったものの、VC 奪取手法ではトラフィックによらずに、最高優先度パケットの平均転送遅延、ジッタおよび最大転送遅延の性能が向上しており、優先度逆転の軽減が確認された。

**謝辞** 本研究の一部は科学技術振興機構 CREST の支援によるものであることを記し、謝意を表す。本研究の提案に対し大変貴重なご意見をいただいた向後琢磨氏に深謝の意を表す。

## 参考文献

- [1] Hammond, L., Nayfeh, B.A. and Olukotun, K.: A Single-Chip Multiprocessor, *Computer*, Vol.30, No.9, pp.79–85 (1997).
- [2] Olukotun, K., Nayfeh, B.A., Hammond, L., Wilson, K. and Chang, K.: The Case for a Single-Chip Multiprocessor, *Proc. 7th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'09)*, pp.2–11 (1996).
- [3] Hazra, R.: MIC Architecture and the Path to Exascale Computing, Presented at *International Conference for High Performance Computing, Networking, Storage and Analysis* (2011).
- [4] Salihundam, P., Jain, S., Jacob, T., Kumar, S., Erraguntla, V., Hoskote, Y., Vangal, S., Ruhl, G. and Borkar, N.: A 2 Tb/s 6x4 Mesh Network for a Single-Chip Cloud Computer With DVFS in 45 nm CMOS, *IEEE Journal of Solid-State Circuits*, Vol.46, pp.757–766 (2011).
- [5] Shin, J.L., Tam, K., Huang, D., Petrick, B., Pham, H., Hwang, C., Li, H., Smith, A., Johnson, T., Schumacher, F., Greenhill, D., Leon, A.S. and Strong, A.: A 40 nm 16-core 128-thread CMT SPARC SoC Processor, *IEEE International Solid-State Circuits Conference (ISSCC'10)*, San Francisco, CA, pp.98–99 (2010).
- [6] Wentzlaff, D., Griffin, P., Hoffmann, H., Bao, L., Edwards, B., Ramey, C., Mattina, M., Miao, C.-C., Brown, J.F. and Agarwal, A.: On-Chip Interconnection Architecture of the Tile Processor, *IEEE Micro*, Vol.27, pp.15–31 (2007).
- [7] Udipi, A., Muralimanohar, N. and Balasubramonian, R.: Towards Scalable, Energy-Efficient, Bus-Based On-Chip Networks, *Proc. 16th IEEE International Symposium on High-Performance Computer Architecture (HPCA'10)*, pp.1–12 (2010).
- [8] Dally, W.J. and Towles, B.: Route Packets, Not Wires: On-Chip Interconnection Networks, *Proc. Design Automation Conference (DAC'01)*, Las Vegas, NV, pp.684–689 (2001).
- [9] Guerrier, P. and Greiner, A.: A Generic Architecture for On-Chip Packet-Switched Interconnections, *Proc. Design, Automation and Test in Europe Conference and Exhibition 2000 (DATE'00)*, Paris, pp.250–256 (2000).
- [10] Benini, L. and Micheli, G.D.: Networks on chips: A new SoC paradigm, *IEEE Computer*, Vol.35, pp.70–78 (2002).
- [11] Kim, J.: Low-Cost Router Microarchitecture for On-Chip Networks, *Proc. 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'09)*, pp.255–266 (2009).
- [12] Fallin, C., Craik, C. and Mutlu, O.: CHIPPER: A Low-complexity Bufferless Deflection Router, *Proc. 17th IEEE International Symposium on High-Performance Computer Architecture (HPCA'11)*, pp.144–155 (2011).
- [13] Moscibroda, T. and Mutlu, O.: A Case for Bufferless Routing in On-Chip Networks, *Proc. 36th Annual International Symposium on Computer Architecture (ISCA'09)*, pp.196–207 (2009).
- [14] Hayenga, M., Jerger, N.E. and Lipasti, M.: SCARAB: A Single Cycle Adaptive Routing and Bufferless Network, *Proc. 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'09)*, pp.244–254 (2009).
- [15] Tota, S., Casu, M.R. and Macchiarulo, L.: Implementation Analysis of NoC: A MPSoC Trace-Driven Approach, *Proc. 16th Great Lakes Symposium on VLSI (GLSVLSI'06)*, pp.204–209 (2006).
- [16] Matsutani, H., Koibuchi, M., Amano, H. and Yoshinaga, T.: Prediction Router: Yet Another Low Latency On-Chip Router Architecture, *Proc. 15th IEEE International Symposium on High-Performance Computer Architecture (HPCA'09)*, pp.367–378 (2009).
- [17] Kumar, A., Peh, L.-S., Kundu, P. and Jha, N.K.: Express Virtual Channels: Towards the Ideal Interconnection Fabric, *Proc. 34th Annual International Symposium on Computer Architecture (ISCA'07)*, pp.150–161 (2007).
- [18] Michelogiannakis, G., Pnevmatikatos, D. and Katevenis, M.: Approaching Ideal NoC Latency with Pre-Configured Routes, *Proc. ACM/IEEE International*

*Symposium on Networks-on-Chip (NOCS'07)*, pp.153-162 (2007).

[19] Jiang, N., Becker, D.U., Michelogiannakis, G. and Dally, W.J.: Network Congestion Avoidance Through Speculative Reservation, *Proc. 18th International Symposium on High-Performance Computer Architecture (HPCA'12)* (2012).

[20] Michelogiannakis, G., Jiang, N., Becker, D.U. and Dally, W.J.: Packet Chaining: Efficient Single-Cycle Allocation for On-Chip, *Proc. 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'11)*, pp.33-36 (2011).

[21] Kinsy, M., Cho, M.H., Wen, T., Suh, E., van Dijk, M. and Devadas, S.: Application-Aware Deadlock-Free Oblivious Routing, *Proc. 36th Annual International Symposium on Computer Architecture (ISCA'09)*, pp.208-219 (2009).

[22] Owens, J.D., Dally, W.J., Ho, R., Jayashima, D.N., Keckler, S.W. and Peh, L.-S.: Research Challenges for On-Chip Interconnection Networks, *IEEE Micro*, Vol.27, pp.96-108 (2007).

[23] Grot, B., Hesness, J., Keckler, S.W. and Mutlu, O.: Kilo-NOC: A Heterogeneous Network-on-Chip Architecture for Scalability and Service Guarantees, *Proc. 38th Annual International Symposium on Computer Architecture (ISCA'11)*, San Jose, CA, pp.401-412 (2011).

[24] Ouyang, J. and Xie, Y.: LOFT: A High Performance Network-on-Chip Providing Quality-of-Service Support, *Proc. 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'10)*, pp.409-420 (2010).

[25] Grot, B., Keckler, S.W. and Mutlu, O.: Preemptive Virtual Clock: A Flexible, Efficient, and Cost-effective QoS Scheme for Networks-on-Chip, *Proc. 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'09)*, New York, NY, pp.268-279 (2009).

[26] Lee, J.W., Ng, M.C. and Asanovic, K.: Globally-Synchronized Frames for Guaranteed Quality-of-Service in On-Chip Networks, *Proc. 35th Annual International Symposium on Computer Architecture (ISCA'08)*, Beijing, pp.89-100 (2008).

[27] Goossens, K., Dielissen, J. and Radulescu, A.: AEthereal network on chip: Concepts, Architectures and Implementations, *IEEE Design & Test of Computers*, Vol.22, pp.414-421 (2005).

[28] Das, R., Mutlu, O., Moscibroda, T. and Das, C.R.: Aéria: Exploiting Packet Latency Slack in On-Chip Networks, *Proc. 37th Annual International Symposium on Computer Architecture (ISCA'10)*, pp.106-116 (2010).

[29] Das, R., Mutlu, O., Moscibroda, T. and Das, C.: Application-Aware Prioritization Mechanisms for On-Chip Networks, *Proc. 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'09)*, New York, NY, pp.280-291 (2009).

[30] Lampson, B.W. and Redell, D.D.: Experience with Processes and Monitors in Mesa, *Comm. ACM*, Vol.23, pp.105-117 (1980).

[31] 向後卓磨, 山崎信行: 優先度付きオンチップネットワークのための VIX ルータとその評価, 先進的計算基盤システムシンポジウム論文集, 東京, pp.381-390 (2011).

[32] Dally, W.J. and Towles, B.: *Principles and Practices of Interconnection Networks*, Morgan Kaufmann (2004).

[33] 戸田賢二, 西田健次, 高橋栄一, Michell, N., 山口喜教:

優先度先送り方式による実時間相互結合網用ルータチップの実現と性能, 情報処理学会論文誌, Vol.36, No.7, pp.1619-1629 (1995).

[34] Kim, J.H. and Chien, A.A.: Rotating Combined Queueing (RCQ): Bandwidth and Latency Guarantees in Low-Cost, High-Performance Networks, *Proc. 23rd Annual International Symposium on Computer Architecture (ISCA'96)*, Philadelphia, PA, pp.226-236 (1996).

[35] Golestani, S.J.: Congestion-Free Communication in High-Speed Packet Networks, *IEEE Trans. Comm.*, Vol.39, pp.1802-1812 (1991).

[36] Yakovlev, A.V., Koelmans, A.M. and Lavagno, L.: High-level Modeling and Design of Asynchronous Interface Logic, *IEEE Design & Test of Computers*, Vol.12, pp.32-40 (1995).

[37] Peh, L.-S. and Dally, W.J.: A Delay Model and Speculative Architecture for Pipelined Routers, *Proc. 7th International Symposium on High-Performance Computer Architecture (HPCA'01)*, Monterrey, Mexico, pp.255-266 (2001).



山口 将一

2011 年慶應義塾大学理工学部情報工学科卒業。2013 年慶應義塾大学大学院理工学研究科開放環境科学専攻修士課程修了。



山崎 大輝

2012 年慶應義塾大学理工学部情報工学科卒業。現在、慶應義塾大学大学院修士課程に在籍。オンチップネットワークの研究に従事。



笹川 雄二郎

2010 年慶應義塾大学理工学部情報工学科卒業。2012 年慶應義塾大学大学院理工学研究科開放環境科学専攻修士課程修了。



松谷 宏紀 (正会員)

2004年慶應義塾大学環境情報学部卒業。2008年慶應義塾大学大学院理工学研究科開放環境科学専攻博士課程修了。博士(工学)。現在、慶應義塾大学理工学部情報工学科専任講師。2009年度より2010年度まで日本学術振興会特別研究員SPD。計算機アーキテクチャ、オンチップネットワークの研究に従事。



山崎 信行 (正会員)

1991年慶應義塾大学理工学部物理学科卒業。1996年慶應義塾大学大学院理工学研究科計算機科学専攻博士課程修了。博士(工学)。同年電子技術総合研究所入所。1998年10月慶應義塾大学理工学部情報工学科助手。同専任講師を経て、2004年4月より同助教授(現、教授)。リアルタイムシステム、プロセッサアーキテクチャ、並列分散処理、システムLSI、ロボティクス等の研究に従事。日本ロボット学会、IEEE各会員。