

バッファの動的制御機構を有する 仮想出力キューイング方式オンチップルータ

松村 将嘉¹ 中野 秀洋¹ 宮内 新¹

概要: 仮想出力キューイング (VOQ) 方式オンチップルータは、仮想チャンネル (VC) を出力ポートごとに静的に対応付けることで、制御回路の簡素化を図ることができる。しかし、VOQ では、トラフィックに偏りがあるとき、バッファの利用効率が大きく低下する。本稿では、トラフィックに応じて VC のバッファサイズを動的に制御する手法を提案する。数値実験を行い、提案手法の有効性を示す。

A virtual output queuing on-chip router with dynamical control schemes for buffers

MASAYOSHI MATSUMURA¹ HIDEHIRO NAKANO¹ ARATA MIYAUCHI¹

Abstract: Virtual Output Queuing (VOQ) on-chip routers can simplify control circuits by statically associating virtual channels (VCs) and output ports. However, utilization rate can significantly decrease in the VOQ when the network traffic is biased. This paper proposed a method to dynamically control buffer size of VCs depending on the network traffic. The effectiveness of the proposed method is verified in the computer simulations.

1. はじめに

近年、マルチコアプロセッサや SoC におけるバスに代わる新たなコア間結合方式として Network-on-Chip(NoC)[1] が注目されている。NoC は、オンチップルータを介したネットワークにより各演算コアを結合する手法であり、コア間のデータ転送はパケットベースで行われる。本手法を用いることで、バスによる結合において必要であった長距離なグローバル配線はルータ間の短距離な配線に置き換えることが可能となり、配線遅延の影響が低減される。さらに、通信路が競合しない複数のデータ転送を同時に実行することが可能であるため、高いバンド幅が提供される。NoC において、オンチップルータはネットワーク全体の性能に影響する重要な構成要素である。NoC の性能を向上させるためにはパケット転送のレイテンシを削減し、またスループットを上昇させる必要がある。一方で、オンチップルータは LSI 上へ実装されるものであるため、そのハー

ドウェアコストを削減することも重要な課題である。本研究では、低遅延化とハードウェアコストの削減を同時に実現可能な手法として提案されている仮想出力キューイング (Virtual Output Queuing:VOQ) 方式オンチップルータに対してバッファの使用率を向上させる手法を提案する。これにより、ルータのハードウェアコストを増大させることなくさらなる NoC の性能向上を可能とする。提案手法の有効性を実証するために、提案するルータを用いた NoC におけるパケット転送のシミュレーション実験による性能評価を行った。また、FPGA 向けの実装を行うことにより、ルータの回路規模や最大動作速度の検証を行った。

2. 背景

本章では NoC において一般的なオンチップルータである仮想チャンネル (Virtual Channel:VC) ルータと、その改良手法である VOQ ルータについて述べる。なお、本研究においては、バッファサイズを削減可能であるという利点から、ワームホール方式によるパケット転送を行うことを想定する。これは、パケットをいくつかのフリット (Flow

¹ 東京都市大学
Tokyo City University

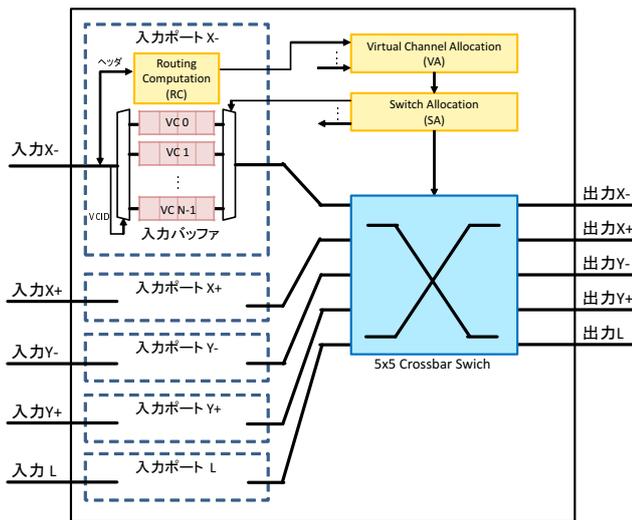


図 1 VC ルータ (5 入力/出力ポート, N VC)
Fig. 1 A VC Router (5 input/output ports, N VCs)

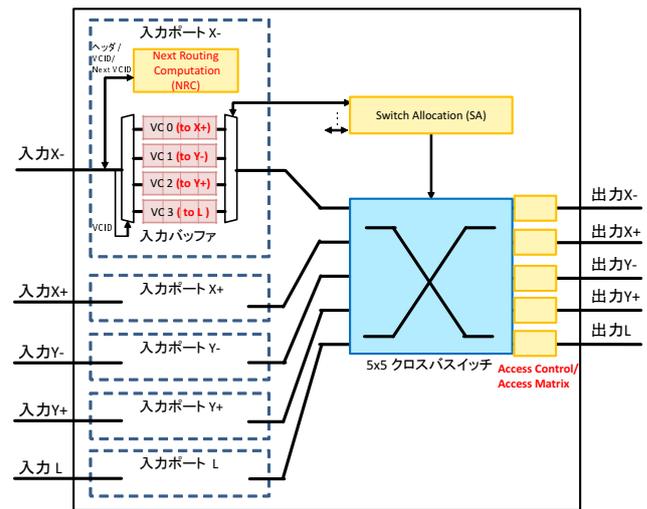


図 2 VOQ ルータ (5 入力/出力ポート)
Fig. 2 A VOQ Router (5 input/output ports)

Control Digit:Flit) と呼ばれる固定長の構造に分割し、それを最小単位として転送を行う方式である。フリットには Head, Body, Tail の 3 種類があり、それぞれパケットの先頭部、中間部、終端部に該当する。各フリットの先頭部には種類を識別するためのビット及び VC 番号 (VCID) を格納するビットが付加される。

2.1 仮想チャネルルータ (VC ルータ)

VC ルータ [2] は、一つの物理的なチャネル (伝送路) に対して仮想的な複数のチャネルを備えることで、ネットワークの混雑の抑制を図ったものである。VC ルータの構成を図 1 に示す。

VC は入力ポート部にいくつかのバッファ (一般的には FIFO) を並列に実装することにより実現される。各入力ポートのバッファが単一であった場合、バッファの先頭に格納されているパケットの転送がブロックされた際 (Head-Of-Line(HOL) ブロッキングと呼ばれる)、後続のパケットの転送もブロックされる。HOL ブロッキングは上流のルータへと拡散し、ネットワークの広範囲に混雑を発生させる。これはパケット転送のレイテンシを著しく増大させる原因となる。一方で、複数の VC が備えられていれば、一つのパケットの転送がブロックされた場合においても、他のパケットは残りの VC を経由して転送を継続することができる。

入力ポートよりパケットが入力された際、それをバッファに格納すると同時に、出力先を決定するための Routing Computation(RC) が実行される。続いて、出力先ルータで使用する VC を決定するために、VC Allocation(VA) が実行される。VC はパケット単位で割り当てられ、同時に同一の VC を使用可能なパケットは 1 つであるため、競合が発生した場合は調停が行われる。Switch Allocation(SA) では、VC の割当てが完了している全ての入力パケットに対して

調停を行い、クロスバスイッチのタイムスロットを割り当てる。タイムスロットはフリット単位で割り当てられる。タイムスロットを割り当てられたフリットは次のクロックサイクルでクロスバスイッチを通過し (Switch Traversal:ST) 隣接ルータへ転送される。上記のように、VC ルータにおいてパケット転送に要する処理の数は多いため、4 段のパイプラインにより実行される (RC/VA/SA/ST)。なお、RC および VA はパケット単位で実行されるため、Head フリットの到着時にのみ行われる。VC ルータの欠点として、以下の 2 点が挙げられる。

- 多段のパイプライン処理によるパケット転送レイテンシの増大
- VC Allocation 等の複雑な処理による回路規模の増大

2.2 仮想出力キューイングルータ (VOQ ルータ)

VC ルータの欠点を解決可能な手法として、仮想出力キューイング (VOQ) ルータ [3] が提案されている。VOQ ルータの構成を図 2 に示す。

VOQ ルータの入力ポート部には出力ポートと同数の VC が備えられ、それぞれが出力先と固定的に対応付けられる。VOQ ルータに Head フリットが入力されると、Next Routing Computation(NRC) が実行される。これは、一つ次 (下流) のルータにおける出力先を決定する処理である。次ルータにおいて用いる VC はその出力先により固定的に決定される。したがって、NRC の実行と同時にバッファに書き込まれるフリットの VCID を次ルータの VC 番号 (Next VCID) に書き換えることが可能となる。自身のルータにおける出力先は一つ前 (上流) のルータによって決定されているため、RC ステージはクリティカルパスから除外される。さらに、出力先の VC は NRC により決定されるため、VA ステージは不要となり、そのための回路も省略可能である。これにより、ルータのパイプラインステージ

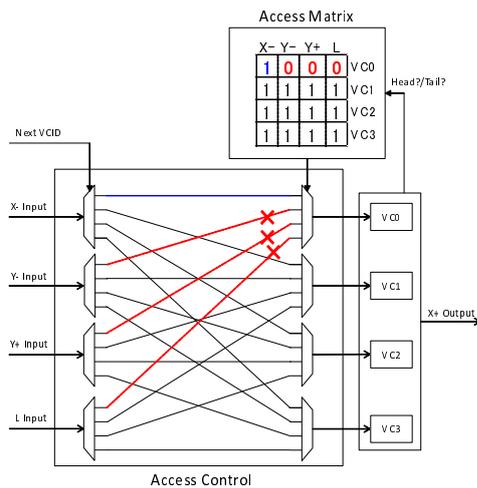


図 3 アクセス制御回路 (X+ 出力ポート)
Fig. 3 Access Control Logic (X+ Output Port)

は SA および ST のみとなる。これらを同時に実行することで、ルータのパイプライン段数を 1 段とすることが可能となる。さらに、同一の VC に格納されるパケットは全て同一の出力先であるため、出力先の異なるパケットの HOL ブロッキングの影響を受けることが無くなるという利点も持つ。ただし、これは個々のルータ内に限られており、パケットが経由する複数のルータでの HOL ブロッキングの影響を回避することはできない場合もある。

VA による出力 VC の調停が省略されているため、複数の入力パケットに対して同時に同一の出力 VC が割り当てられる状況が起こり得る。このような状況に対処するために、各出力ポート部にはアクセス制御回路 (Access Control Logic) とアクセスマトリクス (Access Matrix) が設けられる。アクセスマトリクスは各出力 VC を使用可能な入力ポートを示すテーブルであり、アクセス制御回路はそれを参照することで同時に複数の入力パケットが同一出力 VC へアクセスしないよう制御を行う。アクセス制御回路とアクセスマトリクスの例を図 3 に示す。アクセスマトリクスの 1 はアクセス可能、0 はアクセス不可を意味する。アクセスマトリクスの初期値は全ビットが 1 であり、パケットの Head フリットが通過する際、該当出力 VC のビットについて入力元ポート以外を 0 に更新する。また、Tail フリットが通過する際は該当出力 VC の全ビットを 1 に更新する。図 3 は X+出力ポートにおけるものであり、X-方向からの入力パケットが出力 VC0 へ転送中である場合の例である。この状態ではアクセスマトリクスの VC0 が "1000" となっているため、X-方向以外の入力パケットのアクセスが禁止される。そのパケットの Tail フリットが通過する際、アクセスマトリクスの VC0 は "1111" に更新され、他方向からの入力パケットのアクセスが許可される。

さらなる混雑発生を抑制を図るために、一つの出力先に対応する VC を 2 つに増加させた Multiple VOQ ルータ [3]

が提案されている。Multiple VOQ ルータにおいては、入力・出力 VC が増加しているため、アクセス制御回路がそれに合わせて変更されている。

VOQ ルータは VC ルータの欠点を解決可能であるが、一方でトラフィックに偏りがある場合にバッファの利用率が低くなるという欠点を持つ。一つの出力先に対応する VC が固定されているため、特定の出力先へのパケット転送が集中すると、他の出力先に対応する VC のバッファに十分な空き領域があるにもかかわらず、該当する VC のバッファのみが飽和しブロックを発生させる。

VOQ ルータのバッファ利用率を向上させる手法として、各 VC ごとの小サイズのバッファと全ての VC が共有するバッファを備える手法 [4] が提案されている。しかしながら、特に Multiple VOQ のように一出力方向に対応する VC が複数ある場合、トラフィックの偏りによっては使用されない非共有のバッファ領域が多くなる。

2.3 研究目的

本研究の目的は、VOQ(Multiple VOQ) ルータのバッファ利用率を上昇させることにより、ハードウェアコストを増大させずに NoC の性能を向上させることである。

3. 提案手法

前述のように、VOQ ルータはトラフィックの偏りによりバッファを有効に利用できないという問題を持つ。本研究では、トラフィックの状況に応じて各出力先に対応する VC 数やそのバッファサイズを動的に変化させることが可能な VOQ ルータを提案することによりその問題点を解決する。これを実現するために、Dynamic Virtual Channel Regulator (ViChaR) の概念を VOQ ルータに導入する。

3.1 Dynamic Virtual Channel Regulator (ViChaR)

Dynamic Virtual Channel Regulator (ViChaR)[5] はトラフィックに応じて仮想チャネル数を動的に変動させる手法である。入力ポート部には VC ごとに分けられたバッファではなく、Unified Buffer Structure(UBS) と呼ばれる単一系列のバッファが備えられる。n 個の slots を備えた UBS の例を図 4 に示す。制御回路に備えたテーブルによって各 VC のフリットが UBS のどの slot に格納されているのかを管理する。各 VC が使用可能なバッファの slot は任意であるため、最大でバッファの slot 数と同数の VC を同時に使用できる。これにより、バッファに空き slot がある限り、新しいパケットに対して常に新しい VC を割り当てることが可能となる。また、各 VC が使用する slot はフリットが到着するたびに 1 つずつ割り当てられるため、VC ごとのバッファサイズも変動する。

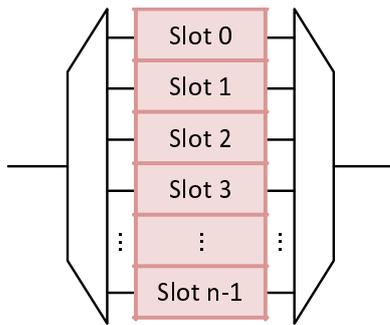


図 4 UBS (n スロット)
Fig. 4 UBS (n Slots)

- Avail. : 各スロットが使用可能であることを示す。
- LHead : 各 VC に対応する連結リスト構造の先頭部のフリットが格納されているスロット番号を示す。
- Lnext : 各スロットに格納されているフリットの後続フリットがどのスロットに格納されているのかを示す。
- LTail : 各 VC に対応する連結リスト構造の終端部のフリットが格納されているスロット番号を示す。

UBS に対してフリットを読み書きする際は、これらのレジスタに対して適切に参照・書き込みが行われる。また、図 6 において、WP は到着フリットを書き込むスロット番号であり、RP は読み込みスロット番号である。WP は Avail. を優先順位エンコーダに入力したものであり、RP は SA よりスイッチが割り当てられた VC の LHead となる。

ViChaR において、各ポートの VC 数の上限値はスロット数と同数である。しかし、VOQ ルータは VC と出力先を固定的に対応づけることで複雑な VC 割り当て処理を削減したものである。よって、一つの出力先に対応する VC 数が不定となった場合、その利点が失われてしまう。これを解決するために、本提案手法では一つの出力先に対応する VC 数に上限を設ける。ここではその上限を 2VC とする。したがって、一つの出力先に対応する VC 数は 0 から 2 の間で変動する。この上限と同数の VC が割り当てられた場合、前述の Multiple VOQ の VC 数と同数となる。また、VC 番号と出力先の対応もあらかじめ決めておく。これにより、スイッチ割り当てやアクセス制御を行う回路は Multiple VOQ と同様のものを用いることが可能となる。しかし、本提案手法は Multiple VOQ と異なり、転送されるパケットが存在しない出力先に対応する VC (およびその VC になが占有するバッファ領域) は存在しない。よって、出力先の VC 数不足によりブロックが発生した場合でも、そのパケットが格納されている VC に十分なバッファ領域を割り当てることが可能となる可能性が高くなる。これにより、上流ルータのバッファに格納されている後続フリットを多く受信することが可能となり、上流ルータにおけるブロックの発生を抑制することができる。出力先の VC 数が上限に足しているか否かは、Head フリットが転送される際に出力ポート部のアクセスマトリクスを参照することにより判定する。上限に達していた場合はそのパケットの転送がブロックされる。

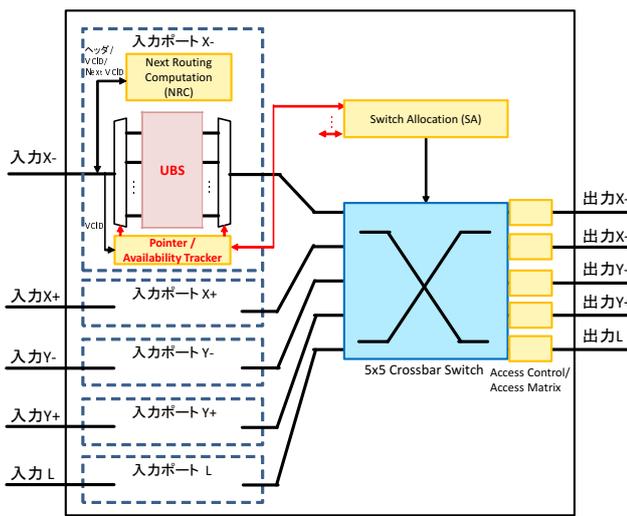


図 5 提案するオンチップルータ
Fig. 5 A Proposal On-Chip Router

3.2 提案手法の実装

本研究では、VOQ ルータに対して、トラフィックに応じて動的に VC 数と各 VC のバッファサイズを変動させるという ViChaR の概念を適用する。これにより、転送されるパケットが多い出力先に対応する VC 数とその VC が使用するバッファサイズを増加させることが可能となる。言い換えれば、転送されるパケットが存在しない出力先に対応する VC を省略し、その分のバッファ領域を必要とされる VC に割り当てることが可能となる。本手法により、バッファの利用率を上昇させることが可能となる。

提案するルータの構成を図 5 に示す。

入力ポート部には ViChaR と同様の UBS を備え、Pointer/Availability Tracker により管理を行う。なお、ViChaR ではバッファの管理にテーブルを用いているが、回路規模を削減するために本提案手法では連結リスト方式 [4] を用いた。この方式では、VC を 1 つの連結リスト構造と見なすことにより管理を行う。UBS と Pointer/Availability Tracker の構造を図 6 に示す。図 6 は、UBS のスロット数 (および最大 VC 数) が 8 である場合の例である。Pointer/Availability Tracker における各レジスタの用途を以下に示す。

4. 実験

提案手法をハードウェア記述言語により実装し、HDL シミュレーションによる NoC の性能評価及び FPGA への実装による回路規模・最大動作速度の評価を行った。表 1 に提案手法及び比較対象の従来手法のルータの仕様を示す。表 1 において、n は各入力ポート当たりのバッファサイズ [Flits] を示している。

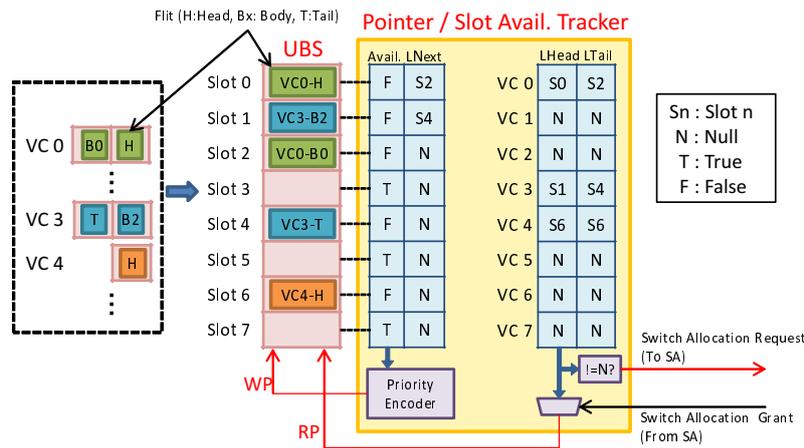


図 6 UBS と Pointer/Availability Tracker
Fig. 6 UBS and Pointer/Availability Tracker

表 1 オンチップルータの仕様
Table 1 Specification of On-Chip Routers

	提案手法-n	4VC-n	VOQ-n	Multiple VOQ-n
入出力ポート数		5 (X-, X+, Y-, Y+, L)		
フリットサイズ [bits]		64		
ルータ間リンクバンド幅 [bits/cycle]		64(1Flit) × 2 (双方向同時転送可)		
パケット転送方式		ワームホール		
ルーティングアルゴリズム		次元順ルーティング (X → Y)		
仮想チャネル数 [VC/Port]	最大 8 (可変)	4	4	8
1 出力方向に対応する VC 数	0 ~ 2 (可変)	-	1	2
入力バッファサイズ [Flits/Port (Flits/VC)]	n (1 ~ n (可変))	n (n/4)	n (n/4)	n (n/8)
パイプライン段数	1	4	1	1
仮想チャネル割り当てアルゴリズム	-	ラウンドロビン	-	-
スイッチ割り当てアルゴリズム		iSLIP [6]		
フロー制御方式		On/Off		

4.1 NoC の性能評価

各ルータを用いたネットワークを構築し、HDL シミュレーションにより平均パケット転送レイテンシと平均バッファ利用率を計測した。シミュレーションソフトウェアには Icarus Verilog 0.9.6 を用いた。シミュレーション環境を表 2 に示す。

ネットワーク上の各ノードには演算コアの代わりにテスト用モジュールを配置する。このモジュールはパケットの生成部と受信部により構成されている。生成部では、予め設定されたトラフィックパターンとパケット生成レートに基づいてパケットを生成し、その時刻を記録する。生成されたパケットはキューに格納され、接続されたルータが受信可能な状態であればフリット単位で順次送信される。一方、受信部ではルータよりフリットを受信する。Tail フリットを受信した際 (すなわち、1つのパケットの受信を

表 2 シミュレーション環境

Table 2 Simulation Environment

ネットワークトポロジ	8 × 8 2 次元 Mesh 型
ノード数	64
パケットサイズ	5[Flits/Packet] (固定)
トラフィックパターン	Uniform-Random
オンチップルータの種類	提案手法 (n=8,16), 4VC (n=8,16), VOQ (n=8,16), Multiple VOQ(n=8,16,24,32)

完了した際) は、その時刻を記録する。ここで、時間の単位はクロックサイクルとし、時刻はシミュレーション開始時より経過したクロックサイクル数とする。

4.1.1 平均パケット転送レイテンシ

NoC の性能評価において、パケット転送レイテンシは特

に重要な指標である。本実験では、送信元ノードにおいてパケットが生成されてから、送信先ノードでそのパケットのTailフリットを受信するまでの時間を計測し、その平均値を求めた。なお、ここでは時刻1000[cycle]から10000[cycle]までの間に生成されたパケットのみを計測の対象とした。図7、図8、図9にパケット生成レートに対する平均パケット転送レイテンシを示す。ネットワークが飽和状態となるとパケット転送のレイテンシは急速に増大する。ここで、平均パケット転送レイテンシが100[cycles]を超過するパケット生成レートを飽和点と仮定する。図7より、 $n=8$ の場合において、Multiple VOQの飽和点は0.305[flits/cycle/node]であるのに対し、提案手法は0.355[flits/cycle/node]となっており、提案手法は飽和点を16.7%上昇させることができていくことがわかる。また、図8より、 $n=16$ の場合においてはMultiple VOQの0.330[flits/cycle/node]に対して提案手法は0.380[flits/cycle/node]となっており、提案手法は飽和点を15.2%上昇させることができていくことがわかる。さらに、図9より、提案手法-8の飽和点はMultiple VOQ-24に対してほぼ同等(+1.4%)であり、提案手法-16はMultiple VOQ-32に対して5.7%高いことがわかる。

4.1.2 平均バッファ利用率

本提案手法はバッファ利用率の向上を図ったものである。それを検証するために、平均バッファ利用率を求めた。平均バッファ利用率の導出にあたっては、まず各時刻における全オンチップルータ内のバッファの利用率について平均値を求め、さらにそれについて一定時間の平均値を求めた。本実験においては、計測時間を時刻1000[cycle]から10000[cycle]までとした。図10、図11、図12にパケット生成レートに対する平均バッファ利用率を示す。これらの結果より、提案手法のバッファ利用率はVOQやMultiple VOQと比較して大きく上昇していることがわかる。前項で示した飽和点におけるバッファ利用率に着目すると、提案手法-8はMultiple VOQ-8の2.89倍、提案手法-16はMultiple VOQ-16の3.38倍となった。

4.1.3 考察

Uniform-Randomトラフィックは送信元-宛先の組み合わせに偏りの少ないトラフィックパターンである。しかし、本実験においてはルーティングアルゴリズムとして次元順ルーティングを用いたため、各ルータにおける出力先として直進方向が選択される割合が高い。そのため、VOQルータやMultiple VOQルータにおいては直進方向以外に対応するVCのバッファに使用されない領域が多く、平均バッファ利用率が低くなったと考えられる。一方で、提案手法は直進方向に対応するVC数やそのバッファサイズを増加させることが可能なため、バッファ利用率を上昇させることが可能となった。

前述の通り、提案手法-8とMultiple VOQ-24の性能が同等である。したがって、提案手法はMultiple VOQに

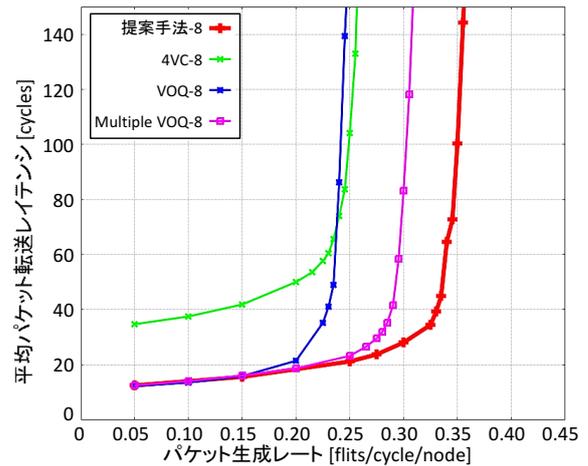


図7 平均パケット転送レイテンシ (n=8)

Fig. 7 Average Packet Transfer Latency (n=8)

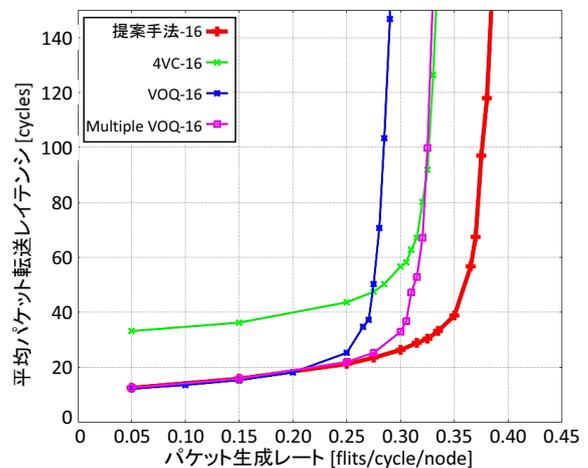


図8 平均パケット転送レイテンシ (n=16)

Fig. 8 Average Packet Transfer Latency (n=16)

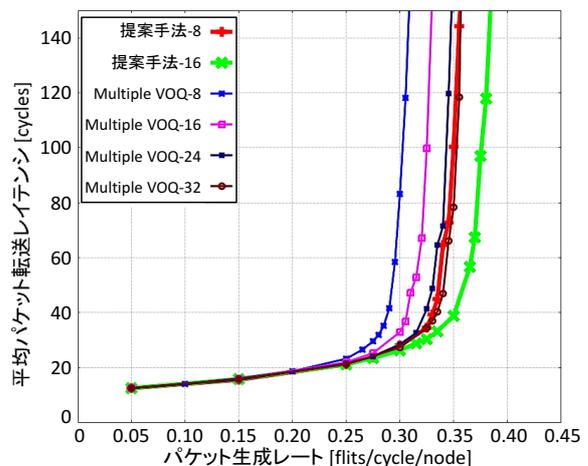


図9 平均パケット転送レイテンシ

(提案手法とMultiple VOQの比較)

Fig. 9 Average Packet Transfer Latency
(Comparing Proposal Method with Multiple VOQ)

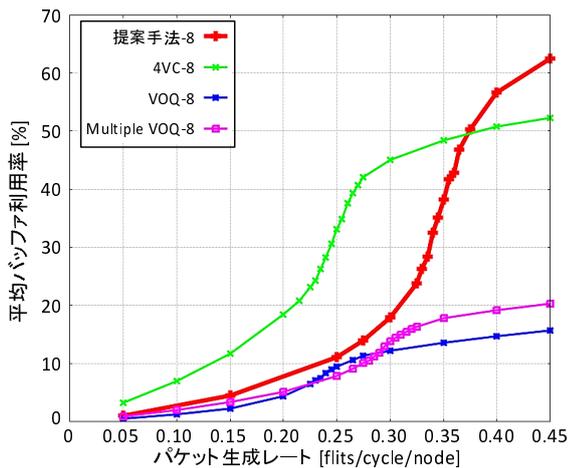


図 10 平均バッファ利用率 (n=8)

Fig. 10 Average Buffer Utilization (n=8)

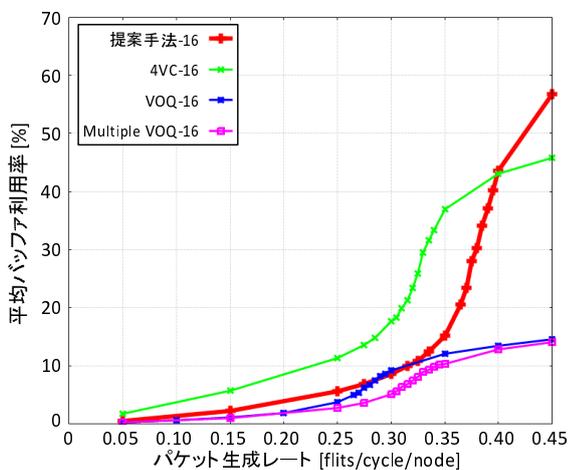


図 11 平均バッファ利用率 (n=16)

Fig. 11 Average Buffer Utilization (n=16)

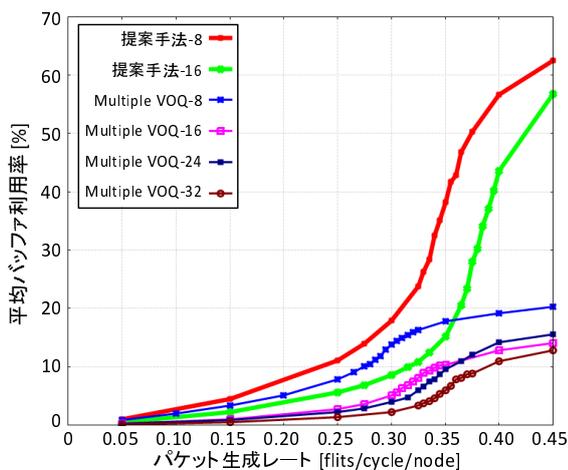


図 12 平均バッファ利用率

(提案手法と Multiple VOQ の比較)

Fig. 12 Average Buffer Utilization

(Comparing Proposal Method with Multiple VOQ)

対して 66.7%小さいバッファサイズで同等の性能を達成できるということが言える。また、提案手法-16 の性能は Multiple VOQ-32 よりも優れている。よって、提案手法は Multiple VOQ に対して 50.0%小さいバッファサイズで性能を向上させられるということが言える。

4.2 回路規模・最大動作速度の評価

各ルータを FPGA へ実装し、回路規模と最大動作速度の評価を行った。ターゲットデバイスは Altera Cyclone IV GX(EP4CGX150DF31I7) とし、FPGA 実装ソフトウェアには Altera Quartus II 11.0 を用いた。

4.2.1 回路規模

回路規模は FPGA へ実装した際の使用ロジックエレメント (LE) 数により評価した。表 3 に各オンチップルータの合計 LE 数 (LEs) と LC Combinationals および LC Registers の値を示す。

提案手法の回路規模は、それぞれバッファサイズが同一の各従来手法に対して増大している。Multiple VOQ と比較すると、LE について 46.7%(n=8), 51.8%(n=16) の増加となった。これは、提案手法の入力ポートに従来手法よりも複雑な制御回路が追加されたことによるものである。しかしながら、前節において提案手法-8 が Multiple VOQ-24 と同等以上の性能を達成できることを示した。提案手法-8 は Multiple VOQ-24 に対して、LE, LC Combinationals, LC Registers がそれぞれ 22.2%, 14.1%, 52.8%減少している。また、提案手法-16 が Multiple VOQ-32 と同等以上の性能を達成できることも確認された。この 2 つを比較すると LE, LC Registers がそれぞれ 11.3%, 35.7%小さく、LC Combinationals が 4.4%大きい。バッファサイズがそれぞれ 66.7%, 50.0%小さいため、特に LC Registers(レジスタ数) が削減されている。この結果より、同等の性能を達成する場合、提案手法で追加された制御回路による回路規模の増加量はバッファサイズの減少による回路規模の削減量を下回っているということが言える。したがって、提案手法において追加された制御回路のオーバーヘッドはルータの性能向上に対して大きな欠点にならない。

4.2.2 最大動作速度

最大動作速度は FPGA 実装ソフトウェアのタイミングアナライザを用いて見積もられた最大クロック周波数 (f_{max}) により評価した。表 4 に各オンチップルータの最大クロック周波数を示す。提案手法の最大クロック周波数は Multiple VOQ と比較して 23.9%(n=8), 15.3%(n=16) 低下している。これらは、各入力ポートに備えられた制御回路によるものと考えられる。VOQ ルータのパイプライン段数は 1 段であり、本制御回路も同一のサイクル内で動作する。そのため、本制御回路の遅延時間の影響により最大クロック周波数が低下したと考えられる。

表 3 回路規模
 Table 3 Logic Area

	提案手法-n		4VC-n		VOQ-n		Multiple VOQ-n			
	n=8	n=16	n=8	n=16	n=8	n=16	n=8	n=16	n=24	n=32
LEs	9539	13980	7172	10368	4690	8002	6501	9207	12254	15787
LC Combinationals	7888	11051	5830	7630	3346	5188	5228	7099	9183	10590
LC Registers	3860	6825	3065	5585	2800	5400	3096	5656	8176	10616

表 4 最大クロック周波数 (f_{max}) [MHz]
 Table 4 Maximum Clock Frequency (f_{max}) [MHz]

提案手法-n		4VC-n		VOQ-n		Multiple VOQ-n			
n=8	n=16	n=8	n=16	n=8	n=16	n=8	n=16	n=24	n=32
36.66	36.29	76.66	74.62	59.30	62.46	48.20	42.84	37.78	45.74

5. 結論

本研究では、VOQ ルータのバッファ使用効率を上昇させることにより、NoC の性能向上と回路規模増大の抑制を両立させることを可能とするオンチップルータを提案した。検証実験により、従来手法である Multiple VOQ ルータと比較してバッファサイズを 66.7%削減した場合においても同等以上の性能を達成でき、ルータの回路規模を 22.2%削減可能であることを確認した。また、バッファサイズを 50.0%削減した場合においてより高い性能を達成でき、ルータの回路規模を 11.3%削減可能であることも確認した。

参考文献

- [1] W. J. Dally and B. Towles: *Route packets, not wires: on-chip interconnection networks*, 2001. Proceedings of Design Automation Conference (2001).
- [2] W. J. Dally and B. Towles: *Principles and Practices of Interconnection Networks*, Morgan Kaufmann (2004).
- [3] S. T. Nguyen and S. Oyanagi: *A Low Cost Single-Cycle Router Based on Virtual Output Queuing for On-Chip Networks*, 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD) (2010).
- [4] C. V. Thiem and S. Oyanagi: *An Input Buffer Architecture for On-chip Routers*, Second International Conference on Networking and Computing (ICNC) (2011).
- [5] C. A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M. S. Yousif and C. R. Das: *ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers*, 39th Annual IEEE/ACM International Symposium on Microarchitecture (2006).
- [6] N. McKeown: *The iSLIP Scheduling Algorithm for Input-Queued Switches*, IEEE/ACM Transactions on Networking, Volume 7, Issue 2 (1999).