

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

向 後 卓 磨[†] 山 崎 信 行[†]

近年、メニーコア CMP の通信基盤である NoC において、ネットワーク・システムレベルの最適化や QoS 制御のためにパケットの転送遅延を優先度に基づいて制御することが有効であることが明らかにされ、優先度付き NoC の必要性が高まりつつある。しかし、従来の優先度付き NoC では優先度導入によるルータ面積の大幅な増加と高優先度パケットが低優先度パケットにブロックされる優先度逆転によるネットワーク性能の低下を考慮していない。本論文は、優先度付き NoC 向けのルータアーキテクチャを提案し、これらの問題を解決する。HDL による実装を行い、ルータ面積、クリティカルパス及び実システムに近い条件におけるネットワーク性能の評価を行った。ネットワーク性能の評価では、5 種類の case study 毎に転送遅延に関する詳細なデータを示す。

Design and Evaluations of VIX Router for Priority-Aware On-Chip Networks

TAKUMA KOGO[†] and NOBUYUKI YAMASAKI[†]

Since recent studies have unveiled that control packet transfer delay in NoC which is communication infrastructure of many-core CMPs is beneficial to network/system level optimization or QoS control, priority-aware NoCs become being highly required. But, previous studies on priority-aware NoC have not considered that naively introducing priority causes the two problems as follows: One hand is significant growth of router area, the other is performance degradation due to priority inversion that higher priority packets are blocked by lower priority packets. This paper proposes a router architecture for priority-aware NoC in order to mitigate those problems. We have implemented the proposal router with HDL and evaluated router area, critical path and network performance using practical system parameters. This paper presents detail data of network evaluations on the each five case study.

1. はじめに

半導体技術により数十億ものトランジスタが集積可能となることから、1 つのチップに複数のコアを搭載する Chip-Multiprocessors (CMPs) がプロセッサ性能を向上させるアーキテクチャとして有望である。既に多数のコアを搭載するメニーコア CMP^{(11),(12),(21)} が実現されており、今後も CMP のコア数は増加し続けると多くの研究者および開発者が予想している。メニーコア CMP ではコア同士の通信のためにスケラビリティ、転送遅延、バンド幅、ハードウェア量、消費電力などを考慮して、Networks-on-Chip (NoC)⁽⁴⁾ が用いられる。メニーコア CMP における NoC の性能はシステム全体に大きな影響を与える。特に NoC のパケットの転送遅延はアプリケーションの実行時間に大きく影響する。パケットの転送遅延は式 1 で与えられる。

$$Delay = H \times D_r + D_c + D_s \quad (1)$$

H は平均ホップ数、 D_r はルータ遅延、 D_c は衝突遅延、 D_s はシリアル化遅延である。コア数の増加に伴い第 1 項

の転送遅延の影響が大きくなるため、 D_r (ルータのパイプラインステージ数) を削減するルータアーキテクチャが数々提案されてきた^{(15),(17),(18),(20)}。また H を削減する効率の良いトポロジ^{(1),(6),(14)} も提案されており、これらの先行研究により第 1 項の遅延の影響は最小限に抑えられつつある。第 3 項の D_s はパケット長 (フリット数) で決まり固定である。さらなる性能向上のためには第 2 項の D_c を小さくする必要がある。

システムレベルで考えた場合、全てのパケットの D_c を削減する必要はなく転送遅延がクリティカルなパケットを優先することが有効であることが明らかにされている^{(2),(7),(8),(10),(16)}。クリティカルなパケットの転送遅延を制御するには、パケットに優先度を付加して、ネットワークを構成する各ルータでクロスバや Virtual Channels (VCs)⁽⁵⁾ などの資源を優先度に基づいて割り当てれば良い。しかし従来のオンチップルータに単純に優先度比較論理を追加した場合、ルータのハードウェア量が大幅に増大すると同時に、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高優先度パケットの転送遅延が増大してしまう優先度逆転問題の発生頻度が高いという問題がある。そこで本論文は、優先度制御を考慮したオンチップルータ VIX を提案

[†] 慶應義塾大学
Keio University

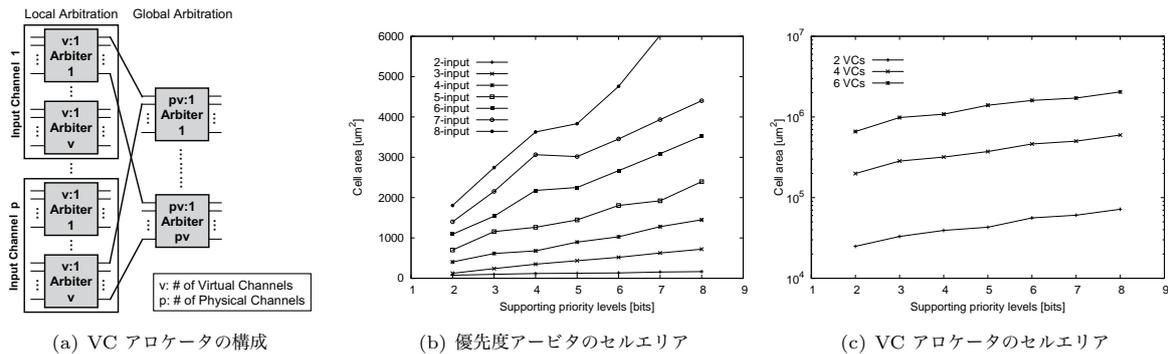


図1 ハードウェア量の見積り。論理合成ではツールに Synopsys Design Compiler, プロセッサライブラリに TSMC 90nm を用いた。優先度アービタ及び VC アロケータはそれぞれ 2.5GHz と 400MHz をターゲットに論理合成を行った。

し、これらの問題を解決する。

本論文の構成は次の通りである。第2章で優先度付き NoC のサーベイとその考察を述べ、第3章で提案する VIX ルータの設計及び実装について述べる。第4章では VIX ルータの評価を行う。第5章で本論文をまとめる。

2. 優先度付き NoC のサーベイとその考察

2.1 優先度付き NoC のサーベイ

NoC に優先度を導入する目的は以下の3つに分類できる。

- ネットワークの特性に基づいた資源割当制御²⁾
- アプリケーションの特性に基づいた資源割当制御^{7),8)}
- 転送遅延を保証する QoS 制御^{10),16)}

1つ目は、ネットワークレベルでの転送遅延削減を目的としている。 D_c の影響が大きいパケット長の短いパケットに高い優先度を与え、反対にパケット長の長いパケットには低い優先度を与える方式が提案されている²⁾。

2つ目は、ネットワークレベルの転送遅延削減が目的ではなく、アプリケーションの実行時間削減が目的である。通信遅延がクリティカルなアプリケーションのパケットに高い優先度を与える方式⁷⁾ や、アプリケーションの実行時間に大きな影響を与えるパケットに高い優先度を与える方式⁸⁾ が提案されている。

3つ目は、転送遅延削減でもアプリケーションの実行時間削減でもなく、通信遅延の保証が目的である。これはチップ内通信にリアルタイム性が必要なシステムやアプリケーションに限らない。ある通信パターンに対してパケットの転送遅延が何十倍以上のオーダーで増大するようではシステムを構成する上で支障となりかねない。この通信時間予測性の低下はコア数の増加に伴うため、今後は QoS 制御の重要性が増していくと考えられる^{10),16),19)}。

2.2 優先度付き NoC における問題点

優先度付き NoC の先行研究^{2),7),8),10),16)} では先読みルーティング⁹⁾ と投機的クロスバ割当²⁰⁾ を組み合わせた従来の

2ステージパイプラインルータ⁵⁾の VC アロケータ及びクロスバアロケータに単純に優先度比較論理を追加している²⁾。本節ではこの単純な優先度比較論理の追加によるハードウェア量が大幅に増加する問題と優先度逆転問題が生じることを指摘する。

2.2.1 ハードウェア量

優先度付き NoC では VC アロケータとクロスバアロケータを構成する各アービタ (Matrix Arbiter²⁰⁾) に優先度比較論理が追加される。優先度アービタの面積は入力数及びサポートする優先度のビット数に対してそれぞれおよそ線形である (図1(b) 参照)。また従来の VC アロケータは多数の入力数の大きなアービタで構成されているため、優先度比較論理の追加によってハードウェア量が大幅に増大する。具体的には、物理チャネル数を p 、VC 数を v とすると VC アロケータはそれぞれ pv 個の $v:1$ ローカルアービタと $pv:1$ グローバルアービタで構成される (図1(a) 参照)。そのため、VC 数が多いときは優先度のビット数がルータ面積に与える影響²³⁾が大きくなる。文献⁸⁾ では優先度8ビットと VC 数6を仮定しているが、このとき VC アロケータの面積は約 2.0mm^2 であり (図1(c))、これは VC で使用するバッファ (約 2KB) の面積 0.5mm^2 を超える非常に大きなサイズである²⁴⁾

²¹⁾ 2ステージパイプラインルータはインテルのメニーコア CMP¹²⁾ でベースとして採用されているだけでなく、多くの NoC の研究^{15),17),18)} においてもベースにされている事実上の標準ルータであるため、本論文では2ステージパイプラインルータを従来ルータと呼ぶことにする。

²²⁾ 優先度毎に専用の VC を持つアーキテクチャ³⁾ も提案されているが、VC のコスト (ハードウェア量) が高い NoC においては、コストパフォーマンスの観点から不向きであり、また、複数の優先度付加方式を組み合わせるといった優先度のレベル数が増える要求を満たせないという観点からも不向きである。

²³⁾ 優先度アービタの面積を $O(\text{Input} \times \text{Priority})$ と仮定すると、VC アロケータの面積は $O(p^2 v^2 \times \text{Priority})$ と考えることができる。

²⁴⁾ バッファ、クロスバアロケータ及びクロスバの面積はそれぞれ $O(pv)$ 、 $O(\max\{v, p\} \times p \times \text{Priority})$ 及び $O(p^2)$ であるので、VC 数の増加による VC アロケータの面積の増加率はルータ内の他のユニッ

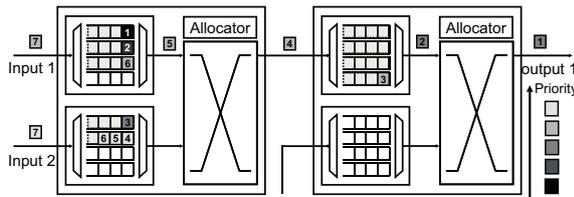


図 2 優先度逆転問題の発生例。色つきの四角はフリットを表し、各パケット（フリット）の優先度は色の輝度と対応している（明るいほど優先度は高い）。四角の中の数字は左側ルータに到着したときの時刻（サイクル）を表す。優先度が互いに異なる 5 つのパケットが左側ルータに到着し、それぞれ右側ルータに転送される状況を表している。また割当済の VC は灰色で表している。説明のため、物理チャンネルを 2 本とし、優先度は 5 段階で値が大きい程高いものとする。1 サイクル目では入力 1VC1 に優先度 1 のヘッダフリット（以降 HF）が、入力 2VC1 に優先度 3 の HF が到着し、両 HF はそれぞれ VC1, VC2 を獲得するが、高優先度入力 2VC1 の HF のみクロスバ割当のグラントを獲得する。2 サイクル目では入力 1VC2 にプリエンブションによって優先度 2 の HF が到着し、入力 2VC1 に優先度 3 のボディフリット（以降 BF）が HF に続けて到着する。入力 1VC2 の HF は VC3 を獲得するが、クロスバ割当のグラントは入力 2VC1 の BF が獲得する。3 サイクル目では入力 1VC3 にプリエンブションによって優先度 4 の HF が到着し、入力 2VC1 に優先度 3 の BF が続けて到着する。入力 1VC3 の HF は VC4 を獲得すると同時に優先度が高いためグラントを獲得する。4 サイクル目では入力 1VC3 に HF に続けて優先度 4 の BF が到着し、入力 2VC2 にプリエンブションによって優先度 5 の HF が到着する。ここで、低優先度パケットによって全て VC が使用されているために入力 2VC2 の HF（最高優先度）は VC 割当を受けることができずストールする。そして入力 1VC3 の BF（低優先度）がグラントを獲得する。5, 6 サイクル目では入力 1VC3 に続けて BF が到着しグラントを獲得する一方、入力 2VC2 に到着した優先度 5 の BF は VC を獲得することなくストールする。

2.2.2 優先度逆転問題

優先度を考慮しない従来ルータは、多数の VC を用いてフリットレベルのインタリーブを行うことでクロスバのマッチング効率を上げ、高スループットと低転送遅延を実現しているが、単純に優先度比較論理を追加した場合、インタリーブよりも多く発生するフリットレベルのプリエンブションによって VC を効率良く用いることができずスループットが低下する。これは従来ルータの積極的な VC^{*}割当^{**}による VC の浪費が原因である。図 17 に VC の浪費による高優先度パケットの転送遅延が増大する例を示す。

従来ルータでは VC 割当とクロスバ割当が独立で行われており、クロスバが割り当てられることのない低優先度パ

ケットに対しても VC は割り当てられる。しかし低優先度パケットは他の高優先度パケットによってブロックされている間、獲得した VC を使用できない。この VC の浪費はルータの実質的な VC 数を減らし、ネットワークの性能を低下させる。VC の浪費は高負荷時に発生しやすく、低優先度パケットにクロスバが割り当てられるまでの長い時間解消しない。特に全ての VC が低優先度パケットに獲得され使用されずにいる場合、VC の枯渇によりプリエンブションが不能となる優先度逆転問題^{***}が発生する。VC の浪費と枯渇は高優先度パケットの転送遅延を増大させるだけでなく、高優先度パケットにブロックされている低優先度パケットの転送遅延も連鎖的に増大させる。

3. VIX ルータアーキテクチャ

3.1 VIX オンチップルータの設計

VC アロケータの多大なハードウェア量と VC の浪費による高優先度パケットの転送遅延増大を同時に解決するために、VC アロケータの構成を変更することが望ましい。優先度付き NoC では優先度を考慮しない NoC と比較すると、低優先度パケットのクロスバが割り当てが成功する可能性は低いいため、低優先度パケットに VC を割り当てたとしてもスループットは向上せず、反対に割り当てることによって VC を浪費する可能性がある。この洞察に基づき VC 割当を従来の積極的な方式から簡素な非積極的な方式にすることによってハードウェア量と高優先度パケットの転送遅延の問題を同時に解決する。

本論文では VC 割当とクロスバ割当のパイプラインステージを統合する VIX (Virtual channel allocation integrated with crossbar allocation) ルータを提案する。VIX ルータでは、投機的にクロスバ割当を行うことで VC 割当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割当、VC 割当の順で逐次的に行う。VC 獲得の有無に関わらず全てのパケット（フリット）がクロスバ割当のリクエストを出す。このときパケット（フリット）が VC を獲得していなければクロスバ割当のグラントをトリガに VC を獲得する。反対にリクエストしようとする出力チャネルに利用可能な 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 割当の論理はクロスバ割当のグラントを利用することができる。従来の面積の

トよりも大きい。そのため、優先度数ビットかつ VC 数 4 以上のとき、VC アロケータがルータ面積の多くを占めることになる。

^{*}以降、特に断りがない限り VC といえば出力チャネルつまり隣接ルータの入力チャネル内の VC を指すものとする。

^{**}従来の VC アロケータは 1 サイクルで最大 p 本の VC を割当可能である。しかし、パケット（ヘッダフリット）の入力タイミングやラフィックパターンなどにより、実際に同時に割り当てられる VC は p 本以下となることが殆どのため過剰である。さらに VC 割当は VC が空になるまで可能な限り行われるため、本論文では従来の VC 割当を積極的な VC 割当と呼ぶ。

^{***}ルータ内の各出力チャネルに対して優先度逆転問題を定義する。具体的には、当該出力チャネルの VC が全て占有されており、当該出力チャネルの VC に空きがあると仮定したときにクロスバが割り当てられるパケットの優先度が当該出力チャネルの VC を占有するパケットの優先度の最大値より大きいときに優先度逆転問題が発生していると定義する。

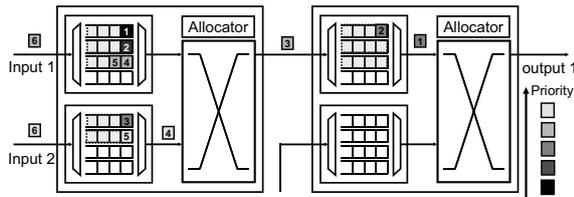


図 3 VIX ルータが優先度逆転問題の発生を抑制する例。シナリオ及び表記は図 17 と同じである。1 サイクル目では入力 1VC1 に到着する HF はクロスバが割り当てられないことがないので、VC を獲得しない。同様に、2 サイクル目では入力 1VC2 に到着する HF は VC を獲得しない。その結果、VC が 2 本空き、4 サイクル目では入力 2VC2 に到着した最高優先度パケット (HF) は VC を獲得しプリエンブションを実行する。

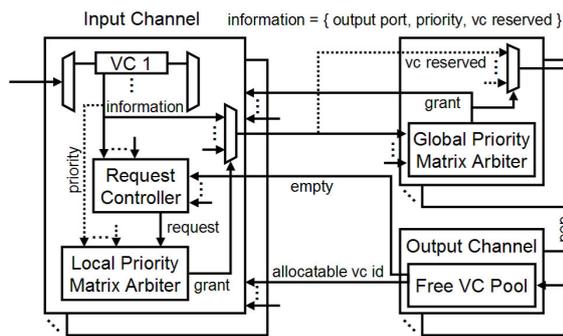


図 4 VIX ルータのアーキテクチャ

大きな VC アロケータが不要となるため、VC 割当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る高優先度のパケットのみが VC を獲得できるので VC の浪費問題も同時に解決できる (図 3 参照)。先読みルーティング⁹⁾を併用することが可能なため、VIX ルータのパイプラインは従来ルータと同じ 2 ステージにすることができる。

3.2 VIX オンチップルータの実装

図 4 に VIX ルータのアーキテクチャを示す。VIX ルータでは VC アロケータが除外され、代わりに同等の機能となる VC プールを各出力チャネルに 1 つずつ追加している。VC プールは利用可能な VC を管理し、そのうち 1 つの VC 識別子を出力する。VC の管理は従来ルータでも既に実装されているため、実質的な追加論理は各出力チャネル毎に利用可能な VC を 1 つ選択する論理、つまり p 個の $v:1$ 固定アービタのみである。VC プールは従来通りクレジットと VC 割当の結果によって更新される。VIX では各 VC プールが VC 割当の結果を得るために、各リクエストの VC 割当済を意味する情報 **vc reserved** をローカルアービタ及びグローバルアービタのグラントによって選択することで伝搬させている。VC プールからは利用可能な VC が存在しないことを意味する情報 **empty** 及び利用可能な VC の識別子を意味する情報 **allocatable vc id** を出力させている。これら 2 つの情報は各入力チャネルに追

Abbreviation: *Win* means a winning request.
OC(request) means an output channel of the request.
MC(request) means a message class of the *OC(request)*.

Operation of the each Free VC Pool

- 1: if *Win* exists then
- 2: if *Win* has not get VC yet then
- 3: forward VC id from Free VC Pool of *MC(Win)*
- 4: update Free VC Pool of *MC(Win)*
- 5: else
- 6: no operation
- 7: end if
- 8: else
- 9: no operation
- 10: end if

Operation of the each Request Controller

- 11: if request has already get VC then
- 12: through request
- 13: else
- 14: if Free VC Pool of *MC(request)* is not empty then
- 15: through request
- 16: else
- 17: invalidate request
- 18: end if
- 19: end if

図 5 メッセージクラスサポート時の各出力チャネルの VC プール及び各入力チャネルのリクエストコントローラの動作

加したリクエストコントローラがハンドルの。リクエストコントローラは各リクエストのうち不要なリクエストをフィルタリングする。リクエストコントローラは VC 獲得済のリクエストはそのまま通過させ、VC 未獲得のリクエストは利用可能な VC が存在すれば通過させ、存在しなければ無効にする。

NoC を構成する上でメッセージクラスのサポートは重要である。以降では VIX ルータのメッセージクラスのサポートのための変更点について述べる。VIX ルータは各出力チャネルにある VC プールから割当可能な VC 識別子を 1 つだけ出力しているが、メッセージクラス 1 つ毎に割当可能な VC を管理し、それぞれから識別子を 1 つ出力するように変更する。またリクエストのタプル (information) にメッセージクラス識別子の情報 **mc id** を追加し、リクエストコントローラのリクエスト無効化の論理は当該出力チャネルかつ当該メッセージクラスの割当可能な VC の存在をチェックするように変更する。VC の獲得は当該出力チャネルかつ当該メッセージクラスの VC 識別子を受ける取るように変更する。VC プールの更新はどのメッセージクラスに対してポップするか決められるように **vc reserved** と同様にリクエストのタプルの 1 つとして **mc id** を伝搬させ、VC プールが受け取ることができるように変更する。これらを図 5 にまとめた。

表 1 評価のためのパラメータ

| Characteristic | Baseline | Variations |
|------------------------|---|---|
| Topology | 8-ary 2-mesh | Case study 5: 4-ary 2-mesh |
| Traffic | Uniform | Case study 1: Uniform, Transpose ⁵⁾ , Local ^{☆1} |
| Link delay | 1 cycle | |
| Packet size | 1 or 5 flits (50-50 chance) | Case study 4: 1 or 9 flits (50-50 chance) |
| Priority ^{☆2} | 4 bits = {batch ^{☆3} , packet size ²⁾ } | Case study 2: 6 bits = {batch, packet size, hop count ^{☆4} } |
| Switching | Wormhole + VC (credit base) | |
| Routing | Dimension-order | |
| Allocation | Priority + least recently served ^{☆5} | |
| # of Ports | 5 /router | |
| # of VCs | 4 /port | Case study 3: 2/port |
| VC depth | 4 flits/VC | |
| Flit width | 128 bits | Case study 4: 64 bits |
| Others | Req/Ack constraint ^{☆6} , 2 MCs ^{☆7} | Synthesis: Not support |

4. 評価

Verilog HDL を用いてオンチップルータを実装し、評価を行った。実装したルータは提案ルータの VIX 及び 2.2 章で説明を行った従来の 2 ステージパイプラインルータの SPC である。評価では各ルータの論理合成の結果と 5 種類の case study におけるネットワーク性能を示す。論理合成にはプロセスライブラリに TSMC 90nm, ツールに Synopsis Design Compiler を、ネットワークシミュレーションには Cadence NC-Verilog を用いた。評価のためのパラメータを表 1 に示す。論理合成及びネットワークシ

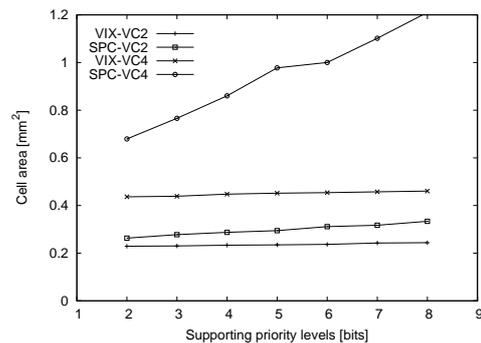


図 6 ルータのセルエリア (500MHz)

☆1 空間的局所性を考慮した Local トラフィックを定義する。Local トラフィックはデイスティネーションが 75 % の確率で隣接ノードの中からランダムに、25 % の確率で全ノードからランダムに決まる。
 ☆2 エイジングを行わない固定優先度付加方式は低優先度パケットがスタベーションを起こしネットワークの挙動が不安定になる。バッチ方式は先行研究^{(7),(8),(10),(16)} で採用されているため、バッチ方式をデフォルトの優先度付加方式に組み込むこととした。評価では 2 種類の優先度付加方式を適用するが、VIX ルータは優先度付加方式には非依存であり、様々な優先度付加方式と組み合わせることができる。
 ☆3 バッチ方式は優先度をエイジングする方法でありバッチインターバルと呼ぶ一定サイクル毎にパケットの優先度を 1 上げる方法である。バッチ方式は低優先度パケットのスタベーションの発生を防ぐだけでなく、パケットの公平性も向上させる。詳細は文献を参照されたい。評価では、バッチインターバルを 64 サイクル、バッチ数を 8 としている。
 ☆4 D_c の影響が大きいホップ数の多いパケットに高い優先度を付加する方式⁽¹³⁾ である。本論文の評価では優先度を $[hop\ counts/4]$ として算出した。
 ☆5 各アロケータを構成するアービタは優先度比較論理を導入した Matrix Arbiter⁽²⁰⁾ である。そのため、最高優先度リクエストがグラントを得ることになる。同一優先度リクエストが存在するときは least recently served のポリシーでグラントが与えられる。
 ☆6 リクエストパケット (パケット長はランダムに決定される) は送信ノードで生成され、リクエストパケットが宛先ノードに到着すると送信ノード宛てにアクノレッジパケットが生成される。またアクノレッジパケットのパケット長はリクエストパケットがショートならばロング、ロングならばショートと対応付けられている。
 ☆7 リクエスト・アクノレッジパケットは共用 VC の VC0 と VC1 を優先的に使い、共用 VC が無い場合はリクエストパケットなら VC2 を、アクノレッジパケットなら VC3 を使用するものとした。

ミュレーションにおける 5 種類の case study ではデフォルトパラメータを基に表 1 右のパラメータに変更して評価を行った。各評価パラメータは実 CMP における NoC のパラメータ (表 2) と優先度付き NoC の先行研究^{(2),(7),(8),(10),(16)} で用いたパラメータを基に定めた。トポロジはレイアウトが容易な 2 次元メッシュとした。ネットワークサイズは 64 ノードを標準とし、文献⁽¹²⁾ のようにクラスタを NoC で接続する構成については 16 クラスタを想定した。軽量から中量ルータを想定し、VC 数を 2 から 4、フリット幅を 64 から 128 とした。パケットのペイロードが 64 Bytes であることを想定しパケット長を定めた。また、制御用のショートパケットとデータ用のロングパケット想定し、発生確率を同率とした。優先度は優先度付き NoC の先行研究^{(2),(7),(8),(10),(16)} で用いられた 1 ~ 8 bits を想定した。

4.1 論理合成結果

図 6 より VIX はルータ面積を削減していることがわかる。VC が 2 本の場合、優先度 2 ビット時に 12.9%、優先度 8 ビット時に 26.9%削減し、特に VC が 4 本の場合、優先度 2 ビット時に 35.7%、優先度 8 ビット時に 61.9%と大幅に面積を削減している。VC の本数が 2 本から 4 本と 2 倍になると、SPC のルータ面積は優先度 2 ビット時に 2.58 倍、優先度 8 ビット時に 3.62 倍となる一方で、VIX

表 2 実 CMP における NoC のパラメータ

| CMP | Network Topology | # of VCs | Flit width | Routing |
|--------------------------|------------------|----------|------------|-----------------|
| TeraFlops ¹¹⁾ | 8 × 10 Mesh | 2 | 32 bits | Source routing |
| TILE64 ²¹⁾ | 8 × 8 Mesh | 1 | 32 bits | Dimension-order |
| SCC ¹²⁾ | 4 × 6 Mesh | 8 | 128 bits | Dimension-order |

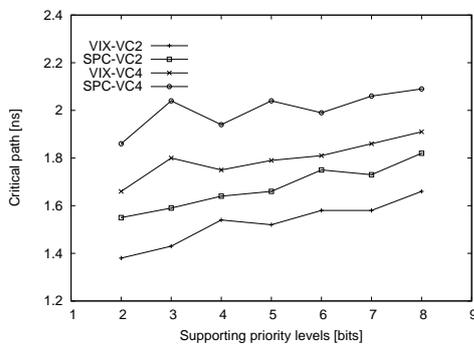


図 7 ルータのクリティカルパス

のルータ面積は優先度 2 ビット時に 1.90 倍、優先度 8 ビット時は 1.89 倍となった^{*}。VIX は優先度のビット数が増加しても面積の増加率が小さく、優先度のビット数に対してスケラビリティがあるといえる。

図 7 より VIX はクリティカルパスを増加させていないことがわかる。VIX のクリティカルパスは VC が 2 本の場合、優先度 2 ビット時に 11.0%、優先度 8 ビット時に 8.8%、VC が 4 本の場合、優先度 2 ビット時に 10.8%、優先度 8 ビット時に 8.6% 短いという結果になった。SPC のクリティカルパスは VC アロケータと投機論理それぞれのクリティカルパスの和である。VIX ルータのクリティカルパスはリクエストコントローラとクロスバアロケータと VC プールそれぞれのクリティカルパスの総和である。ただし、VC プールはリクエストコントローラ及びクロスバアロケータと並列化できるため、VIX ルータの実質的なクリティカルパスはリクエストコントローラとクロスバアロケータそれぞれの和である。クリティカルパスの値は昇順に VIX のクロスバアロケータのクリティカルパス、SPC の VC アロケータのクリティカルパス、VIX のクリティカルパス (リクエストコントローラ+クロスバアロケータ)、SPC のクリティカルパス (VC アロケータ+投機論理) となった。つまり、クリティカルパスの長い SPC の VC アロケータを除去し、クリティカルパスはクロスバアロケータとなり短くなるが、リクエストコントローラを直列で追加することで長くなる。ただし、SPC は VC アロケータと投機論

^{*} SPC の面積の多く占める VC アロケータの面積が $O(p^2v^2 \times Priority)$ であるのに対し、VIX の VC 割当機能を持つクロスバアロケータの面積は $O(p^2v \times Priority)$ である。ルータの物理チャンネル数が大きいとき、SPC の面積はさらに増大すると考えられる

理が直列になっているため、投機論理を含めた全体で考えるとクリティカルパスが短くなるということである。

4.2 Case study 1: トラフィックパターン

ショートパケット及びロングパケットの転送遅延についてトラフィックパターン毎の評価を示す^{**}。

図 8(a), 9(a), 10(a) より、各トラフィックに共通してネットワークの飽和直前の負荷において、高優先度が与えられているショートパケットの平均転送遅延が削減されていることがわかる。また低優先度が与えられているロングパケットの平均転送遅延は増加していない。結果、全パケットの平均転送遅延が削減されている。高優先度パケットの転送遅延の削減は VIX のクロスバ割当に連動した VC 割当により、VC 利用効率が向上したことで、優先度逆転問題の発生の頻度が低下したためである。VC 利用効率が向上したことでスループットも向上している。

図 8(b), 9(b), 10(b) より、SPC は優先度逆転問題の頻発により、低い負荷であっても最大転送遅延が大幅に増大し、場合によっては高優先度パケットの最大転送遅延が低優先度パケットの最大転送遅延を上回ることが観測されているのがわかる。VIX は優先度逆転問題の頻発を軽減しているため、負荷の増大に従って最大転送遅延が増加するという傾向となった。また VIX では高優先度パケットの最大転送遅延が低優先度パケットの最大転送遅延を上回ることがなかった。さらに、VIX で低優先度パケットの転送は高優先度パケットの転送完了に依存しているため、VIX の高優先度パケットの転送遅延削減により低優先度パケットの転送遅延も連鎖的に削減されているのである。

図 8(c), 9(c), 10(c) より、VIX は低い負荷からジッタ (転送遅延の標準偏差) の増大を抑制していることがわかる。これは、SPC では負荷が低いときにおいても VC の浪費及び枯渇が発生し、高優先度パケットの転送遅延の急激な増加が発生しているといえる。最悪転送遅延についての考察と同様に高優先度パケットの転送遅延の急激な増加を抑制したことにより連鎖的に低優先度パケットの急激な増加も抑制したといえる。

4.3 Case study 2: 優先度付加方式

Case study 2 では、case study 1 より優先度が 2 ビット多いため、優先度付きトラフィック特有のフリットレベ

^{**} バッチによる優先度はネットワーク上で刻々と変化するため、ネットワーク注入時から排出時まで固定であるパケット長による優先度のみを取り扱うことにした。

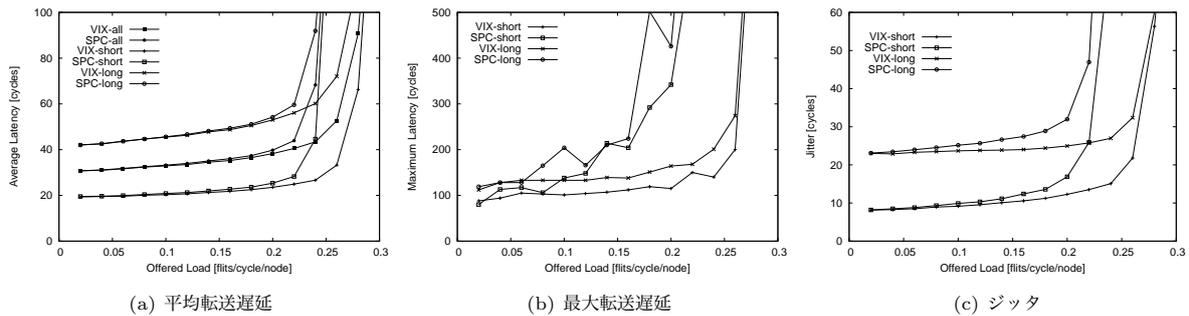


図 8 Case study 1 : Uniform トラフィック

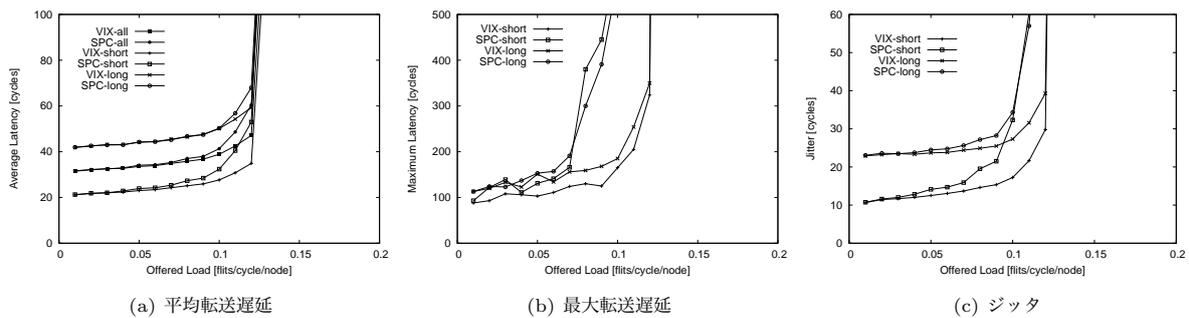


図 9 Case study 1 : Transpose トラフィック

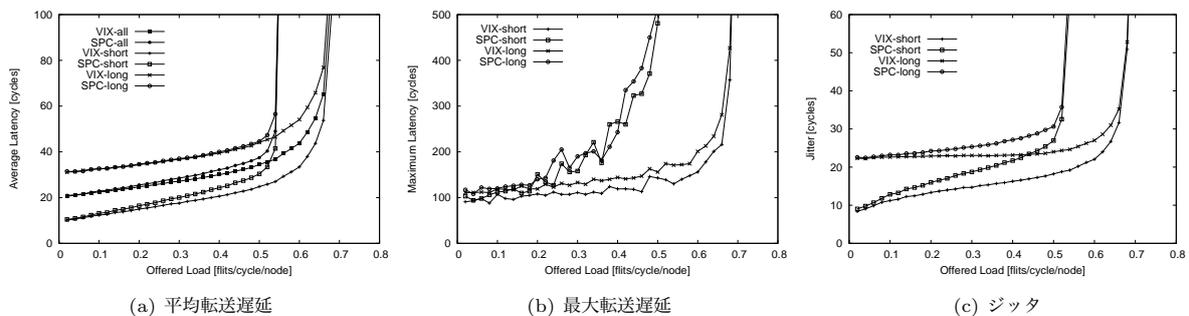


図 10 Case study 1 : Local トラフィック

ルのプリエンブションが発生する頻度が高くなる。つまりルータの優先度に基づいたフロー制御がトラフィックに与える影響が大きくなる。

図 11 より, case study 1 と同様に高優先度パケットに限らず, 低優先度パケットも含めた全パケットの平均転送遅延が削減されていることがわかる。スループットの向上についても case study 1 と同様である。VIX では, 負荷が無負荷状態から飽和状態になる手前の間, 負荷に対する平均転送遅延の増分が僅かであり, ホップ数を基に与えた優先度が転送遅延を正確に制御している。

Case study 1 と同様に, 図 12 より最大転送遅延を削減し, 図 13 より, 高優先度パケットのジッタを削減していることがわかる。一方で, VIX 及び SPC の両者共に優先度 0 のホップ数 1~3 (1-3H) のロングパケットの平均転送

遅延が優先度 1 のホップ数 4~7 (4-7H) のロングパケットの平均転送遅延を超えている。このようなホップ数と平均転送遅延の逆転は高優先度であるショートパケットでは発生していない。このホップ数と平均転送遅延の逆転は低優先度パケットが高優先度パケットにブロックされ続けるために生じると考えらる。これは各手法とも最低優先度パケットのジッタが他の優先度のパケットよりも大きい, つまり D_c が大きくなっていることからいえる。これに加えて VIX は優先度逆転問題の頻発を回避する代わりに低優先度パケットの転送機会を奪っているため, VIX では最低優先度パケットの転送遅延が増大する懸念がある。しかし, 高優先度パケットの転送遅延削減に付随して低優先度パケットも削減される。これらを差し引いた結果, 図 11 のとおり下位 2 レベルの優先度のパケットの平均転送遅延

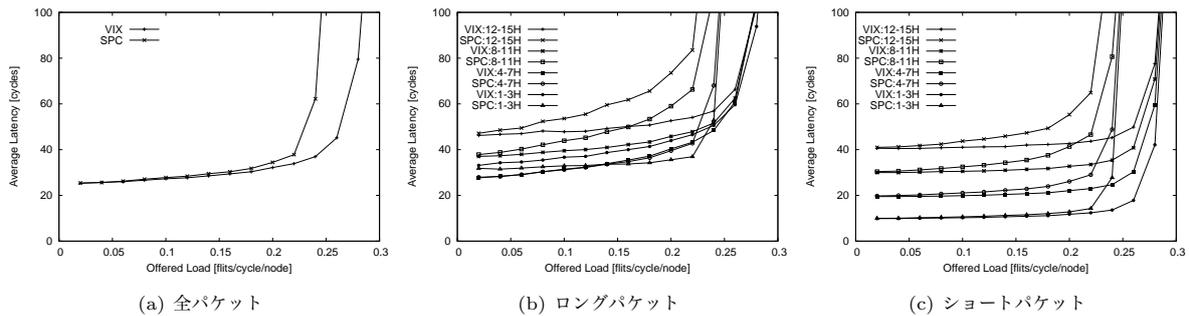


図 11 Case study 2 : 平均転送遅延

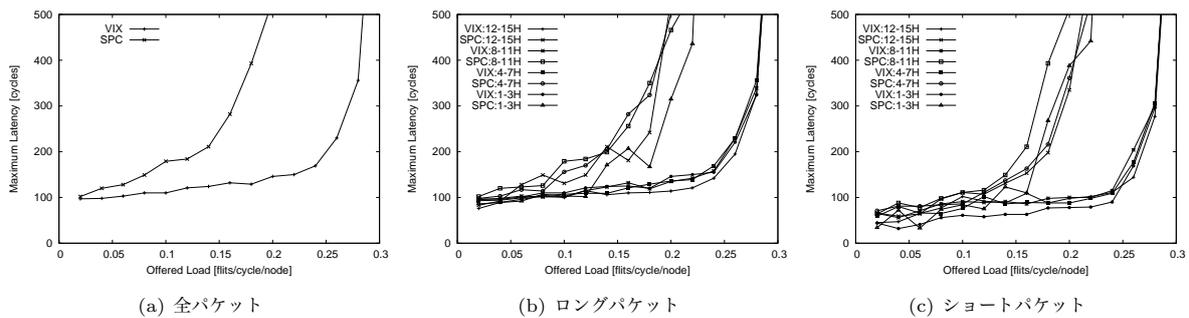


図 12 Case study 2 : 最大転送遅延

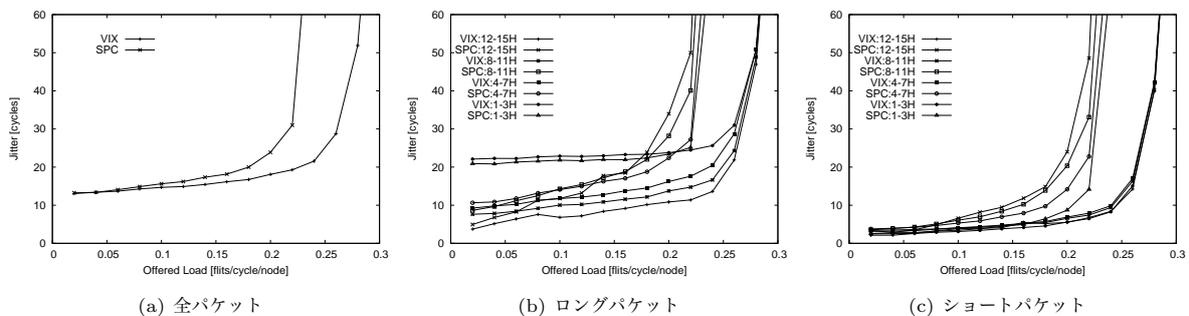


図 13 Case study 2 : ジッタ

は VIX は SPC より大きくなっている (VIX の方が最大転送遅延が小さいことに注意されたい)。サポートする優先度のビット数が増加するに従って、提案方式における低優先度パケットの転送遅延が増加するトレードオフが顕著に現れると考えられる。

4.4 Case study 3, 4, 5 :

VC 数, パケット長, ネットワークサイズ

VC 数, パケット長及びネットワークサイズのパラメータを変更しても, VIX は図 14 より平均転送遅延, 図 15 より最大転送遅延, 図 16 よりジッタを削減することがわかる。

Case study 3 では VC 数を case study 1 の 0.5 倍としている。図 14(a) と図 8(a) を比較すると VC が 2 倍になると VIX ではスループットが 1.93 倍になったのに対し、

SPC では 1.85 倍となっている。VIX は VC 数が多くなるほどスループットで有利であることがいえる。

Case study 4 ではパケット長が長くしているため、ネットワークを跨がってロングパケットが VC を占有することになり、case study 3 と同様に VC 枯渇が発生しやすい構成である。図 14(b) と図 8(a) を比較すると VIX ではスループットは低下しないのに対し、SPC では約 0.9 倍となった。VIX は VC 枯渇が発生しやすいとき、スループットで有利であることがいえる。

Case study 5 ではネットワークサイズが小さくなっているため、パケット同士の衝突が少なく、VC 枯渇が発生しにくい構成である。図 14(c) と図 8(a) を比較すると VIX ではスループットが 1.72 倍になったのに対し、SPC では 1.83 倍となった。VC 枯渇が発生しにくいとき、VIX は

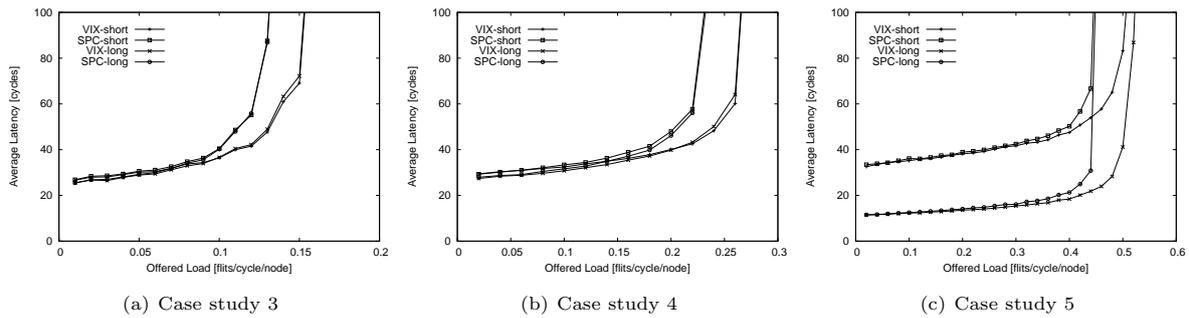


図 14 Case study 3, 4, 5 : 平均転送遅延

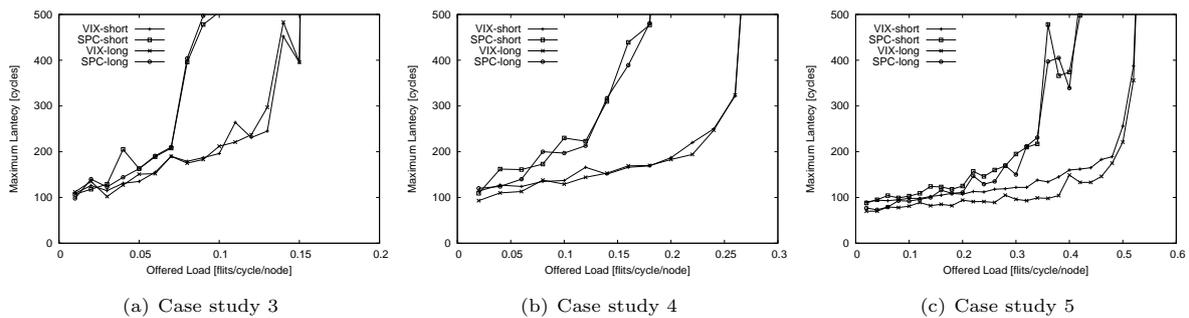


図 15 Case study 3, 4, 5 : 最大転送遅延

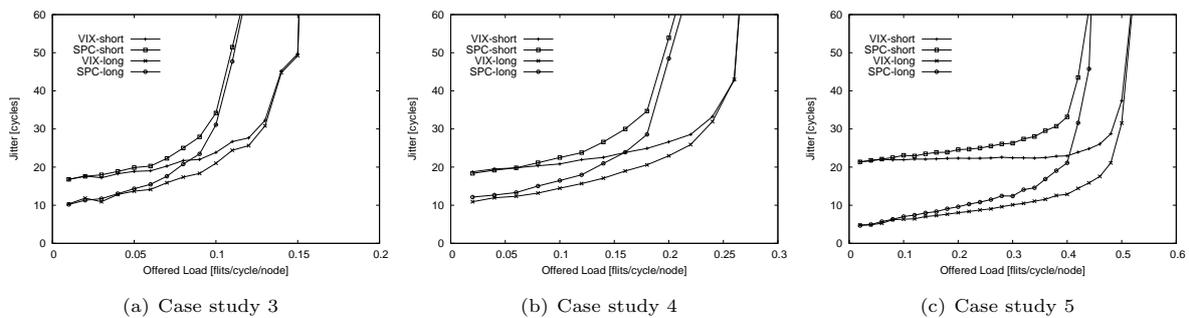


図 16 Case study 3, 4, 5 : ジッタ

SPC とスループットの差が小さくなることがいえる。

4.5 優先度逆転問題発生率

図 17 は、Uniform トラフィックにおける各ルータの各出力チャンネルにおける優先度逆転問題の発生率を示している。Case study 1 (case1) において VIX は優先度逆転問題発生率を負荷が 0.10 のとき 21%削減し、負荷が 0.22 のとき 54%削減した。優先度逆転問題発生率に差が出る負荷 0.10 以上のとき、図 8(a), 8(c), 8(b) からわかるように平均転送遅延、ジッタ及び最大転送遅延においても差が出ている。同様に、case study 2 (case2) において、負荷が 0.18 のとき 22%削減し、負荷が 0.26 のとき 56%削減し、優先度逆転問題発生率に差が出る負荷 0.12 以上のとき、図 11(a) 及び 13(a) からわかるように全パケットの平均転送遅延及びジッタにおいても差が出ている。以上より、VIX

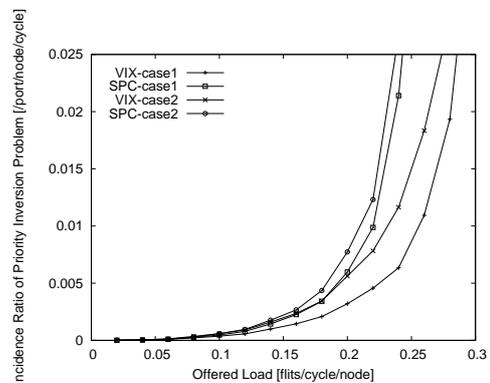


図 17 優先度逆転問題発生率

は優先度逆転問題の発生率を抑えることでネットワーク性能を向上させているといえる。

5. 結 論

本論文では優先度付き NoC 向けのルータアーキテクチャ VIX を提案し、その詳細な評価を示した。VIX は VC 割当をクロスバ割当の結果を用いるアーキテクチャであり、これによりルータ面積を大幅に削減すると同時に VC 利用効率を向上させ、高優先度パケットの転送遅延削減、転送遅延のバラつき抑制及びスループット向上をクリティカルパスを増大させることなく達成した。5 種類の case study によってトラフィックパターン、優先度付加方式、VC 数、パケット長及びネットワークサイズに依らず、VIX はネットワーク性能を向上できることを示した。

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。また本論文執筆に際し貴重なご意見を頂きました慶應義塾大学の松谷宏紀先生に感謝致します。

参 考 文 献

- 1) J. Balfour, W. Dally: Design and Tradeoffs for Tiled CMP On-Chip Networks, *Proceedings of the International Conference on Supercomputing (ICS'06)*, pp. 187–198 (2006).
- 2) E. Bolotin *et al.*: The Power of Priority: NoC Based Distributed Cache Coherency, *Proceedings of the 1st ACM/IEEE International Symposium on Networks-on-Chip (NOCS'07)*, pp. 117–126 (2007).
- 3) E. Bolotin *et al.*: QNoC: QoS architecture and design process for Network on Chip, *Journal of Systems Architecture: the EUROMICRO Journal*, pp. 105–128 (2004).
- 4) W. J. Dally, B. Towles: Route Packets, Not Wires: On-Chip Interconnection Networks, *Proceedings of the Design Automation Conference (DAC'01)*, pp. 684–689 (2001).
- 5) W. J. Dally, B. Towles: *Principles and Practices of Interconnection Networks*, Morgan Kaufmann (2004).
- 6) R. Das *et al.*: Design and Evaluation of a Hierarchical On-Chip Interconnect for Next-Generation CMPs, *Proceedings of the Symposium on High-Performance Computer Architecture (HPCA'09)*, pp. 175–186 (2009).
- 7) R. Das *et al.*: Application-Aware Prioritization Mechanisms for On-Chip Networks, *Proceedings of the International Symposium on Microarchitecture (MICRO'09)*, pp. 280–290 (2009).
- 8) R. Das *et al.*: Aergia: Exploiting Packet Latency Slack in On-Chip Networks, *Proceedings of the International Symposium on Computer Architecture (ISCA'10)*, pp. 106–116 (2010).
- 9) M. Galles: Scalable pipelined interconnect for distributed endpoint routing: The SGI SPIDER chip, *Proceedings of the International Symposium on High-Performance Interconnects (HOTI'96)*, pp. 141–146 (1996).
- 10) B. Grot, S. W. Keckler, O. Mutlu: Preemptive Virtual Clock: A Flexible, Efficient, and Cost-effective QoS Scheme for Networks-on-Chip, *Proceedings of the International Symposium on Microarchitecture (MICRO'09)*, pp. 89–100 (2009).
- 11) Y. Hoskote *et al.*: A 5-GHz Mesh Interconnect for a Teraflops Processor, *IEEE Micro*, Vol. 27, pp. 51–61 (2007).
- 12) J. Howard *et al.*: A 48-Core IA-32 Message-Passing Processor with DVFS in 45nm CMOS, *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC'10)*, pp. 108–109 (2010).
- 13) S. Kato, Y. Fujita, N. Yamasaki: Periodic and Aperiodic Communication Techniques for Responsive Link, *Proceedings of IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'09)*, pp. 135–142 (2009).
- 14) J. Kim, J. Balfour, W. J. Dally: Flattened Butterfly Topology for On-Chip Networks, *Proceedings of the International Symposium on Microarchitecture (MICRO'07)*, pp. 172–182 (2007).
- 15) A. Kumar *et al.*: Express Virtual Channels: Towards the Ideal Interconnection Fabric, *Proceedings of the International Symposium on Computer Architecture (ISCA'07)*, pp. 150–261 (2007).
- 16) J. W. Lee, M. C. Ng, K. Asanovic: Globally-Synchronized Frames for Guaranteed Quality-of-Service in On-Chip Networks, *Proceedings of the International Symposium on Computer Architecture (ISCA'08)*, pp. 89–100 (2008).
- 17) H. Matsutani *et al.*: Prediction Router: Yet Another Low Latency On-Chip Router Architecture, *Proceedings of the Symposium on High-Performance Computer Architecture (HPCA'09)*, pp. 367–378 (2009).
- 18) R. Mullins, A. West, S. Moore: Low-latency virtual-channel routers for on-chip networks, *Proceedings of the International Symposium on Computer Architecture (ISCA'04)*, pp. 188–197 (2004).
- 19) J. D. Owens *et al.*: Rereach challenges for on-chip interconnection networks, *IEEE Micro*, pp. 96–108 (2007).
- 20) L.-S. Peh, W. J. Dally: A delay model and speculative architecture for pipelined routers, *Proceedings of the Symposium on High-Performance Computer Architecture (HPCA'01)*, pp. 255–266 (2001).
- 21) D. Wentzlaff *et al.*: On-Chip Interconnection Architecture of the Tile Processor, *IEEE Micro*, Vol. 27, pp. 15–31 (2007).