社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

予測機構を持った低遅延オンチップルータアーキテクチャ

松谷　宏紀† 　鯉渕　道紘†† 　天野　英晴† 　吉永　努†††

† 慶應義塾大学大学院 理工学研究科　〒 223-8522 神奈川県横浜市港北区日吉 3-14-1
†† 国立情報学研究所　〒 101-8430 東京都千代田区一ッ橋 2-1-2
††† 電気通信大学 大学院情報システム学研究科　〒 182-8585 東京都調布市調布ヶ丘 1-5-1
E-mail: †{matutani,hunga}@am.ics.keio.ac.jp, ††koibuchi@nii.ac.jp, †††yosinaga@is.uec.ac.jp

あらまし　Network-on-Chip（NoC）において，コア間の通信遅延はアプリケーションの性能を左右する重要な要素である．ルータがパケットを転送するために要するサイクル数を減らすため，我々は予測機構を用いた低遅延ルータを提案してきた．本ルータでは，次に転送する出力チャネルを予測し，パケットの到着前に予めアービトレーションを完了させておく．この投機処理により各ルータにおける経路計算およびアービトレーションステージを省略し，低遅延な通信を実現する．本論文では，予測機構を持ったルータのデータパス構造，バッファ管理機構，アービタ，予測失敗時のリカバリ機構について検討する．これらの機能を持った予測ルータを NoC 向けに設計し，面積，フリット転送エネルギー，通信遅延について評価した結果，予測ルータは通常のルータと比べて，面積と転送エネルギーがそれぞれ 23.4% と 10.0% 増加したが，64〜256 コアのネットワークにおいて通信遅延が 14.2〜23.7% 減少した．

A Low-Latency On-Chip Router Architecture with Prediction Mechanism

Hiroki MATSUTANI†, Michihiro KOIBUCHI††, Hideharu AMANO†, and Tsutomu YOSHINAGA†††

† Graduate School of Science and Technology, Keio University　3-14-1, Hiyoshi, Yokohama, JAPAN 223-8522
†† National Institute of Informatics　2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, JAPAN 101-8430
††† Graduate School of Information Systems, The University of Electro-Communications
1-5-1, Chofugaoka, Chofu-shi, Tokyo, JAPAN 182-8585
E-mail: †{matutani,hunga}@am.ics.keio.ac.jp, ††koibuchi@nii.ac.jp, †††yosinaga@is.uec.ac.jp

Abstract　The communication latency between multi cores is one of the crucial factors that determine the application performance on Network-on-Chips (NoCs). In order to reduce the number of cycles required to forward packets on routers, we have proposed a low-latency router architecture that predicts an output channel being used by the next packet transfer and speculatively performs the switch arbitration. This predictable router achieves the low-latency communications, since packets can be transferred without the routing computation and arbitration stages if the prediction succeeds. In this paper, we developed architecture of the prediction router in terms of the datapath structure, buffer management, arbitration strategy, and recovery mechanism for miss predictions. We designed the prediction router for NoCs and evaluated it in terms of area, energy efficiency, and communication latency. The evaluation results showed that the area and energy were increases by 23.4% and 10.0% respectively, but the communication latency was reduced by 14.2-23.7% for the 64- and 256-core networks.

1. はじめに

半導体技術の進歩により，単一チップ上にプロセッサやメモリ，I/O など複数の設計モジュールをタイル状に実装できるようになった．このようなタイルアーキテクチャにおいてタイル同士の結合に Network-on-Chip（NoC）[1] が用いられている．

NoC はパケット処理を行うルータを多数用いることで，高スケーラビリティ，高スループットを実現している．しかし，ルータ構造は経路計算やアービトレーションなどの複雑な内部処理を行うため，リピータバッファを用いた従来のバス構造に

比べてデータの転送遅延が増大する．

コア間の通信遅延はアプリケーションの性能を左右する重要な要素であるため，我々は予測機構を用いた低遅延ルータを提案してきた [2]．予測ルータは，パケットを転送するために要するサイクル数を減らすため，次に転送する出力チャネルを予測し，予めアービトレーションを完了させておく．この投機処理により，各ルータにおける経路計算およびアービトレーション処理を省略し，低遅延な通信を実現する．つまり，予測が成功した場合，1) サーキットスイッチングのように低遅延な通信ができ，2) パケット毎にルーティングを行うパケットネットワークのように，通信前に end-to-end の経路上のネットワーク資源を予約，確保する必要がないという 2 つの利点を持つ．

これまでに，我々は予測ルータを用いたネットワークにおける予測成功率，パケットの転送遅延の削減，スループット向上について検討，評価を行ってきた [3] [2]．しかし，予測ルータのアーキテクチャとその実現性に関する詳細な議論，定量的な評価は行われていない．そこで，本論文では，予測機構を持ったルータのデータパス構造，バッファ管理機構，アービトレーション機構，予測失敗時のリカバリ機構について述べる．そして，これらの機能を持った予測ルータを NoC 向けに設計し，面積，フリット転送エネルギー，通信遅延について評価する．

本論文の構成は次のとおりである．まず 2. 章で一般的なオンチップルータの構造を説明し，これまでに提案された低遅延ルータのアーキテクチャを調査する．次に 3. 章で我々が提案している予測機構を持った低遅延ルータアーキテクチャについて述べる．最後に 4. 章で評価を示し，5. 章で本論文をまとめる．

2. 既存のルータアーキテクチャ

2.1 節で NoC において一般的な 3 サイクルルータ，2.2 節で低遅延な 2 サイクルルータを述べ，最後に関連研究を紹介する．

2.1 オリジナルルータ

通常，ルータは入力チャネルからパケットを受信し，パケットのヘッダに格納された宛先アドレスから出力チャネルを決定，クロスバスイッチのアービトレーションを行い，適切な出力チャネルからパケットを出力する．これらのパケット転送処理は以下の 4 ステップに分類できる [4].

- Routing computation（RC）：パケットのヘッダに格納された宛先アドレスから出力チャネルを決定する．
- Virtual channel allocation（VA）：出力仮想チャネルを決定する（仮想チャネルを使用するシステムの場合のみ必要）．
- Switch allocation（SA）：RC が選択した出力チャネルを使用するため，クロスバスイッチのアービトレーションを行う．
- Switch traversal（ST）：フリットがクロスバ上を移動．

仮想チャネルを持たないルータにおいて，上記の各ステップをそれぞれ 1 サイクルで実行すると，ルータのパイプライン構造は図 1 のような 3 段パイプラインになる．

仮想チャネルを持つルータの場合，各ステップを 1 サイクルで実行すると RC，VA，SA，ST の 4 段パイプラインとなる．NoC の場合，VA と SA を同時に実行する 3 サイクルルータも存在する．この場合，仮想チャネルの割り当て（VA）に成功し，かつ，クロスバスイッチの割り当て（SA）に成功した場合のみヘッダが VA/SA ステージを通過できる．そのため VA に失敗すれば SA が成功したとしても VA/SA ステージをやり直すことになる [4].

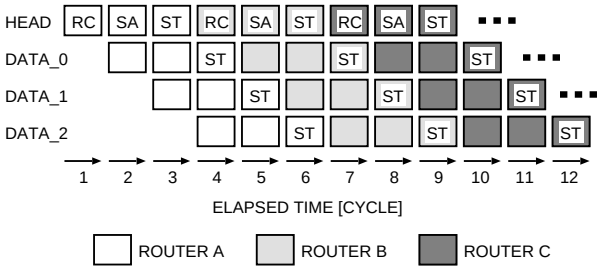


図 1　ルータのパイプライン構造（3 サイクルルータ）．

2.2 低遅延ルータ

低遅延ルータの例として，ルックアヘッドルーティングを用いた 2 サイクルルータを説明する．

図 1 の RC と SA をオーバラップさせることで，ルータのパイプラインステージを 1 ステージ分削減できる．ただし，SA は RC 完了後に実行する必要があるため，RC と SA を同一サイクルで実行するのは効率が悪い．そこで，1 ホップ手前のルータに次ホップのルータの RC を予め実行（Next routing computation，NRC）させる．NRC の結果は次ホップのルータで使われ，自ルータの SA に影響を与えないため，図 2 のように NRC と SA を並列に実行できる．このように次ホップの経路を計算するルーティングをルックアヘッドルーティングと呼ぶ．

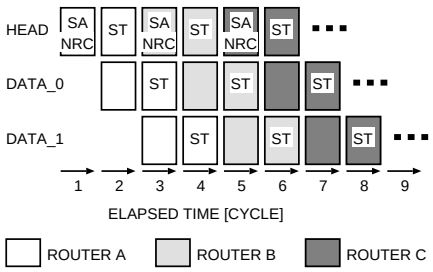


図 2　ルータのパイプライン構造（2 サイクルルータ）．

仮想チャネルを持つ場合は，NRC と VA をオーバラップさせて 3 サイクルルータとしたり，NRC と VA/SA をオーバラップさせて 2 サイクルルータとすることもできる．

さらに，SA と ST をオーバラップさせることで 1 サイクル転送も可能となるが，1 サイクルに多くの処理を詰め込むことになるので，動作周波数の大幅な低下を招きやすい [4].

隣接ルータ間で連携することでパケット処理の遅延を抑える方法も提案されている．Express VC は，仮想的に非隣接ルータ間でバイパス経路を構成することにより中継ルータにおける所要パイプライン段数を削減する [5]．しかし，局所性を持つ通信パターンに対しては低遅延化の効果が小さい．

Preferred Path [6] はデータパスを変更することで通信遅延を削減できるが，ソースルーティングが対象であり，さらにルータアーキテクチャが制限される．また，Preferred Path はクロスバを通過するデータパスと，それを迂回するデータパスの両

方が必要となる．一方，予測ルータでは予測の成否に関わらず，パケットは同一のクロスバを通過するデータパスによって転送される．

3. 予測ルータアーキテクチャ

我々が提案している予測ルータでは，ルータが次に転送する出力チャネルを予測し，次パケットが到着する前に SA を完了させておくことで，各ルータにおける RC および SA を省略し，最短 1 サイクルのフリット転送を実現する．本章では，2.1 節で述べた 3 サイクルルータを基準に，予測ルータのデータパス構造，バッファ管理，予測機構，アービタ回路，予測失敗時のリカバリ機構のために必要な変更点を述べる．

3.1 データパス構造

予測ルータの全体像を示すため，まず，予測ルータのデータパス構造を説明する．予測機構自体は 3.3 節で述べる．

予測ルータでは予測が失敗する可能性があるため，次の 2 つのデータパスが同時に動作する（図 3）．

- Backup datapath： 予測が失敗した場合に使用されるデータパス（3 サイクル転送）．
- Prediction datapath： 予測が成功した場合に使用されるデータパス（1 サイクル転送）．

backup datapath は通常のルータのデータパスと同じであるが，予測ルータでは新たに prediction datapath が必要になる．

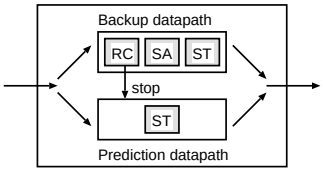


図 3 予測ルータのデータパス構造と stop 機構．

予測ルータでは，次パケットが到着する前に SA を完了させておくことで，各ルータにおける RC および SA を省略する．そのため，予測が成功すれば RC と SA 処理は不要であり，ヘッダフリットは prediction datapath を通って 1 サイクルで転送される．一方，予測が失敗すると，通常のルータと同じように，RC と SA を実行してからヘッダフリットを転送する．

この予測ルータアーキテクチャは次の 3 つの問題を抱えている．

（1） prediction datapath によって面積が増える．

（2） prediction datapath によって消費エネルギーが増える．

（3） 予測失敗時に prediction datapath から無効なフリットが間違った次ホップに対して送出される可能性がある．

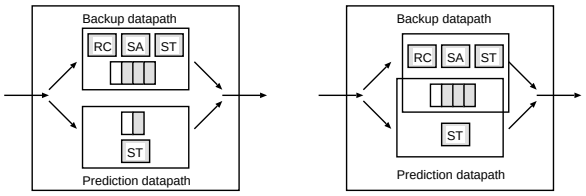
1 つ目の問題は，3.2 節で提案する共有バッファ方式で緩和する．3 つ目の問題は，3.5 節で提案する kill 機構で解決する．本節の残りの部分では，2 つ目の問題を緩和するための stop 機構について述べる．

仮想チャネルを持たないルータの場合，予測が成功したか失敗したかは，backup datapath の RC を実行した後に判明する．予測が成功すれば，backup datapath の出力結果は不要であり，backup datapath の動作を止めることで，回路の無駄なスイッチングを抑制して消費エネルギーの増加を抑えることができる．同様に，予測が失敗した場合は backup datapath の RC から prediction datapath に対し stop 信号を送り（図 3），prediction datapath の動作を止めることで消費エネルギーの増加を抑える．

4.2 節の評価では，通常のルータ，stop 機構を持たない予測ルータ，stop 機構を持つ予測ルータの電力効率を比較する．

3.2 バッファ管理

入力チャネルには，パイプライン段数に応じて入力フリットを蓄えるためのバッファが必要となる．backup datapath と prediction datapath の 2 つのデータパスを持つ予測ルータでは，次の 2 つのバッファ管理方法が考えられる．

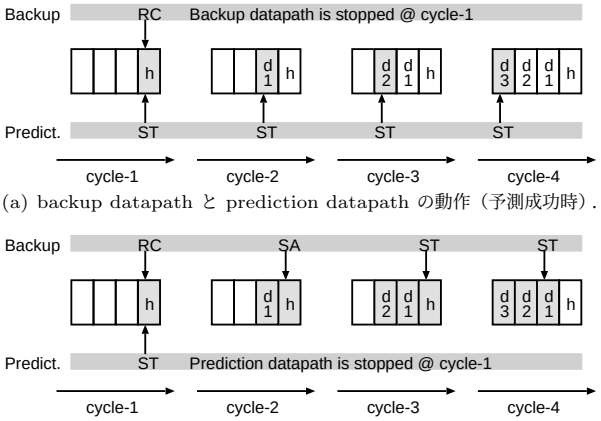


(a) 独立バッファ方式． (b) 共有バッファ方式．

図 4 2 種類のバッファ管理方法．

- 独立バッファ方式：backup datapath と prediction datapath が別々のバッファを持つ（図 4(a)）．
- 共有バッファ方式：backup datapath と prediction datapath が同一バッファを共有する（図 4(b)）．

当然，共有バッファ方式のほうが面積コストで有利である．本節の残りの部分では，共有バッファ方式において 2 つのデータパスが正しくバッファの読み書きができることを示す．



(a) backup datapath と prediction datapath の動作（予測成功時）．

(b) backup datapath と prediction datapath の動作（予測失敗時）．

図 5 共有バッファ方式のバッファ管理（図中の h はヘッダフリット，di は i 個目のデータフリットを示す）．

予測が成功した場合（図 5(a)），無駄な回路のスイッチングを抑えるため backup datapath の動作は 1 サイクル目で止まる．その間，prediction datapath は毎サイクルごとに 1 フリット分のデータを読み込み次ホップに転送していく．

一方，予測が失敗した場合（図 5(b)），prediction datapath の動作は 1 サイクル目で止まる(注1)．代わりに，backup datapath によって RC と SA 処理をした後，3 サイクル目から，

(注1)：この場合，1 サイクル目で最初のフリット（ヘッダ）が間違った次ホップに送出されてしまう．この問題は 3.5 節で提案する kill 機構で解決する．

パケットは 1 フリットずつ正しい次ホップに転送される．なお，出力チャネルの競合が起きると SA に複数サイクルかかるが，このような場合でも 1 サイクル目で prediction datapath の動作は止まるので，間違った prediction datapath によって backupdata path のデータが上書きされることはない．仮想チャネルがある場合については 3.6 節で議論する．

4.1 節の評価では，通常のルータ，独立バッファ方式の予測ルータ，共有バッファ方式の予測ルータの面積を比較する．

3.3 予測機構

予測機構には，次に使われる出力チャネルを予測する predictor と，予測の成否を検査する checker が必要となる．

predictor として，我々は様々な予測アルゴリズムを提案，検討してきた [3]．最も単純な予測アルゴリズムは静的に出力チャネルを予測する方法である．静的直進予測アルゴリズム（SS）は入力パケットが常に同一次元を直進すると予測する．すなわち，2 次元トーラスにおいて北の入力チャネルに到着したパケットは南の出力チャネルへ，東からは西へといった具合である．他にも，ランダム予測アルゴリズム（Rand）はランダムに予測出力チャネルを選択する．

一方，直前ポート予測アルゴリズム（LP）は，入力パケットが 1 つ前のパケットと同一の出力チャネルを選択すると動的に予測する．したがって，通信履歴は入力ポート毎に 1 つ分が必要となる．より洗練された方法としては，パターンマッチ予測アルゴリズム（SPM）がある．SPM はパターンマッチングに基づくユニバーサル予測アルゴリズムに，系列の長さ制限等の制約条件を付けたものである [3]．過去の通信履歴から繰り返しパターンを検索することによって，並列プログラムが持つ通信の規則性を抽出しやすい利点を持つ．さらには，反辞書法を用いて分岐予測に用いる方法についても検討を行った．

checker は backup datapath の RC ステージに実装される．1) 予測アルゴリズムによって予測した出力チャネルと backup datapath の RC が計算した出力チャネルが一致し，かつ，2) 予測した出力チャネルへのクロスバの仮予約が済んでいるとき，checker は予測が成功したと判断する．予測が成功したら checker は backup datapath の動作を止め，予測が間違っていたら prediction datapath の動作を止める．

4. 章では，predictor として LP を採用した予測ルータを用いて，ハードウェア量とフリット転送エネルギーを評価する．

3.4 アービタ回路

ルータ内のある出力チャネルに対し，複数の入力チャネルが SA を実行しようとしたとき，出力チャネルへのアクセス権をどれか 1 つの入力チャネルに付与するのがアービタの仕事である．通常は，ラウンドロビン法など公平な方法を用いて，出力チャネルと入力チャネルのマッチングを実行する．

予測ルータの入力チャネルでは，prediction datapath が SA を実行せずにパケットを仮予約中の出力チャネルに送出する．そのため，ある出力チャネルに対し，複数の入力チャネルが SA を実行しようとしたとき，その出力チャネルを仮予約している入力チャネルがあれば優先的に利用権を与える必要がある．

例えば，図 6 では，入力チャネル p2 が出力チャネル p0 を

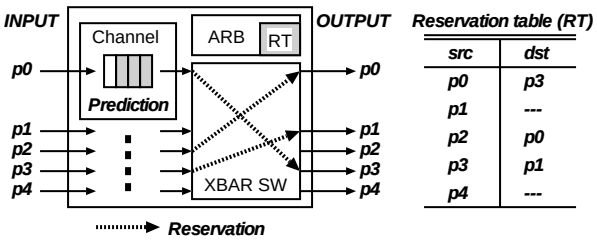


図 6　予測ルータ向けアービタと reservation table.

仮予約している．ここで，入力チャネル p2 と他の入力チャネル（例えば，入力チャネル p4）が，出力チャネル p0 に対して同時にリクエストを出したとする．この場合，出力チャネル p0 の利用権を入力チャネル p2 に付与しないと，入力チャネル p2 の prediction datapath から来たパケットが失われてしまう．そこで,「現在，どの出力チャネルをどの入力チャネルが仮予約しているか」という情報を図 6 の reservation table としてアービタ回路に持たせ，アービトレーションの際に参照する．

3.5 予測失敗時のリカバリ機構

3.1 節で述べたとおり，予測失敗時に prediction datapath から無効なフリットが間違った次ホップに対して送出される可能性がある．このような無効フリットはネットワーク資源を浪費してしまうので，予測に失敗したら 1 ホップ先のルータに kill 信号を送り，無効フリットの伝搬を止める．

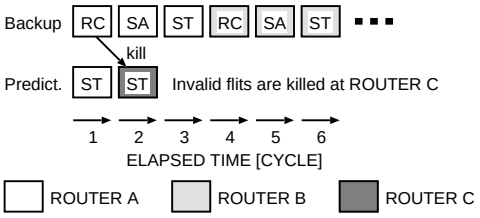


図 7　予測失敗で発生する無効フリットの除去（仮想チャネル無し）.

図 7 では，ルータ a から b へのパケット転送の際，間違った予測によってルータ a から c へ無効フリットが流れてしまう様子を表している．この場合，1 サイクル目の RC（backup datapath）で予測の成否が判明するが，同時に prediction datapath によって最初のフリットは間違った次ホップに転送されてしまう．そこで，1 サイクル目の RC で無効フリットが伝搬してしまったルータ c の入力チャネルに対し kill 信号を送る．ルータ c は 2 サイクル目で，無効フリットと kill 信号を同時に受信するため，無効フリットはルータ c で廃棄され，無効フリットのこれ以上の伝搬を防ぐことができる．

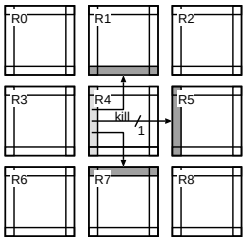


図 8　無効フリットを除去するための kill 信号.

なお，無効フリットを 1 ホップ先で消去するためには，各入力チャネルはパケットを転送する可能性のある 1 ホップ先のすべての入力チャネルに対し，kill 信号を張っておく必要がある．

図 8 に，ルータ 4（R4）の西側の入力チャネルから 1 ホップ先の入力チャネルへの kill 信号を示す．各入力チャネルごとに 3 本程度の kill 信号を配送する必要があるが，データ幅は 1-bit であり，kill 信号の配線長も通常のルータ-ルータ間リンクと同じであるため，kill 信号の配線量は小規模であり，配線遅延も問題にならないと考えられる．

3.6 仮想チャネルルータの場合

ここでは仮想チャネルを持つルータの予測機構について検討する．仮想チャネルルータの場合，予測成功の条件は，1) 予測アルゴリズムが予測した出力チャネルと出力仮想チャネルのペアが，backup datapath の RC と VA の計算結果と一致し，かつ，2) 予測した出力チャネルへのクロスバの仮予約が済んでいるときである．つまり，backup datapath の VA までが完了して初めて，予測の成否が判明する．

次に無効フリットの kill 機構について検討する．単純な仮想チャネルルータではパイプライン構成は RC，VA，SA，ST となるが，backup datapath における VA の完了が 2 サイクル目になると，予測の成否が判明する際には 2 個のフリットが prediction datapath から出力されてしまう．そのため，予測が外れた場合は kill 信号を 2 ホップ先のすべての入力チャネルに配送しなければならず，配線量や配線遅延の増加を招く．

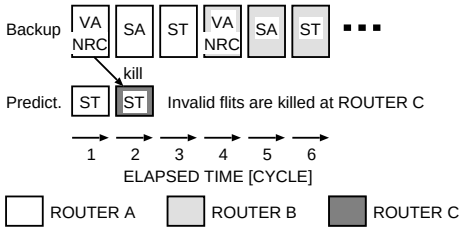


図 9　予測失敗で発生する無効フリットの除去（仮想チャネル有り）．

そこで，VA を backup datapath の 1 サイクル目で完了させるために 2.2 節で紹介したように RC を 1 ホップ手前のルータで実行する（NRC）．そうすることで，図 9 のように NRC と VA をオーバラップさせることができ，VA を 1 サイクル目で実行できる．VA が 1 サイクル目で完了できれば，kill 機構は 3.5 節で述べたものと同じとなる．

4. 評　　価

本章では，予測ルータをハードウェア量，フリット転送エネルギー，無負荷時の通信遅延について評価する．

4.1 ハードウェア量

本節では，通常の 3 サイクルルータと予測ルータをハードウェア量で比較する．

3. 章で述べた予測ルータを 90nm スタンダードセルライブラリを用いて実装した．予測アルゴリズムとして，LP（1 つ前のパケットが使用した出力チャネルを再び選択する）を採用し，予測が当たれば 1 サイクル転送，予測が外れれば 3 サイクル転送するようにした．バッファ管理方式として，独立バッファ方式と共有バッファ方式の場合を別々に実装した．さらに，それぞれのバッファ管理方式ごとに，仮想チャネルを 1 本持つ場合（1-VC）と 2 本持つ場合（2-VC）を実装した．物理チャネル数は 5 本とし，ルータのデータ（フリット）幅は 64-bit とした．ルーティングには次元順ルーティングを用い，スイッチング方式にはワームホール方式を採用した．各仮想チャネルごとに 4-flit 分の FIFO バッファを持たせた．なお，バッファの深さが浅いため，入力バッファはメモリマクロではなく，フリップフロップを用いて実装した．

通常の 3 サイクルルータとして，ここでは文献 [7] のワームホールルータを用いる．このルータは RC，VA/SA，ST ステージから成り，1 本または 2 本の仮想チャネルを持つ．それ以外の実装パラメータは予測ルータと同じである．以上のルータを Synopsys Design Compiler で合成してゲート数を見積もった．

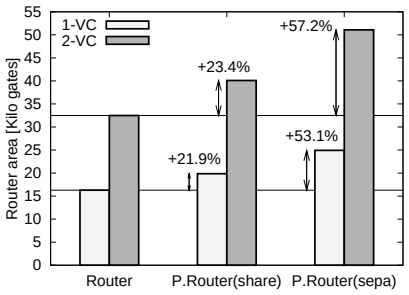


図 10　3 サイクルルータと予測ルータのハードウェア量．

図 10 に，通常の 3 サイクルルータ（Router），共有バッファ方式の予測ルータ（P.Router(share)），独立バッファ方式の予測ルータ（P.Router(sepa)）のハードウェア量を示す．通常のルータと比べ，独立バッファ方式の予測ルータはハードウェア量が 50% 以上増えてしまったが，バッファを共有することで，予測ルータの面積オーバヘッドを 23.4% 以下に抑えることができた．

4.2 フリット転送エネルギー

ルータが 1-flit（64-bit）のデータを転送するのに要する消費エネルギーを**フリット転送エネルギー**とする．本節では，通常のルータと予測ルータをフリット転送エネルギーで比較する．

評価に使用する予測ルータ，および，通常の 3 サイクルルータのアーキテクチャは 4.1 節と同じである．ただし，予測ルータについては面積効率の高い共有バッファ方式のみの評価とし，代わりに，3.1 節で提案した stop 機構（予測の成否に応じて使われない方のデータパスの動作を止める）を持つものと持たないものをそれぞれ評価した．stop 機構によって，回路の無駄なスイッチングを抑えることができ，消費エネルギーの削減が期待される．

これらのルータのフリット転送エネルギーを求めるために，1) Design Compiler でルータ回路を合成，2) Astro で配置配線（CTS でバッファを挿入），3) Verilog-XL で配置配線後シミュレーションを行い switching activity interchange format（SAIF）を生成し，4) Power Compiler でこの SAIF をもとに消費電力を解析した．なお，クロックゲーティングやオペランドアイソレーション等の省電力対策は十分に施してある．

図 11 に，通常の 3 サイクルルータ（Router），stop 機構を持つ予測ルータ（P.Router(stop)），stop 機構を持たない予測ルータ（P.Router）のフリット転送エネルギーを示す．参考の

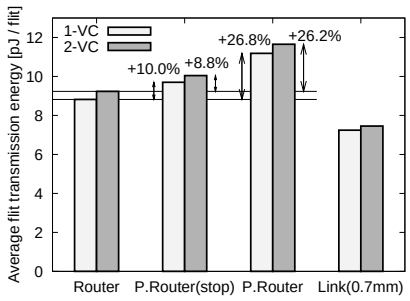


図 11　3 サイクルルータと予測ルータのフリット転送エネルギー.

までに 1-flit が 0.7mm のリンクを移動する際に消費するエネルギーも示した．通常のルータと比べ，stop 機構を持たない予測ルータはフリット転送エネルギーが 26% 以上増えてしまったが，stop 機構を導入することで，予測ルータのフリット転送エネルギーのオーバヘッドを 10.0% 以下に抑えることができた．

4.3　無負荷ネットワークにおけるパケット転送遅延

理想的な低遅延通信はネットワークが空いている，つまり，パケットの衝突が発生しない場合におけるパケット転送である．本節ではその最小の通信遅延について評価を行う．

無負荷状態のワームホールネットワークにおけるパケット転送遅延は次式で計算できる [2].

$$Latency = T_{Link}(d+1) + (T_r + T_a + T_s)d + \frac{PktSize + hd}{\mathrm{LinkBW}}$$

ただし，T_{Link}，T_r，T_a，T_s，d，h はそれぞれリンク遅延，ルータにおけるルーティング遅延，アービトレーション遅延，スイッチング遅延，ホップ数，ヘッダサイズを表す．通常の 3 サイクルルータでは $T_r = T_a = T_s = 1$ となる．一方，予測ルータにおいて予測が成功した場合は $T_r = 0$ かつ $T_a = 0$ となるが，予測が外れた場合は 3 サイクルルータと一緒となる．予測成功率は文献 [2] で解析した結果を用いた．

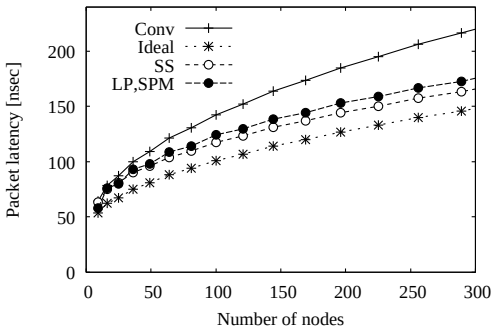


図 12　無負荷状態のネットワークにおけるパケット転送遅延.

トラフィックパターンはユニフォーム，パケット長は 16-flit とし，k-ary 2-cube トーラスにおいてノード数を 9〜289 に変化させたときのパケットの平均転送遅延を見積もった．

評価結果を図 12 に示す．グラフ中には，予測を行わないネットワーク（Conv），予測が 100% 成功すると仮定した理想的なネットワーク（Ideal），予測アルゴリズムとして SS，LP，SPM を採用したネットワークのパケットの平均転送遅延がそれぞれ示されている．4.2 節で行ったルータの配置配線結果より，予測ルータの動作周波数は 464.8MHz，予測を行わない通常の 3 サイクルルータは 470.0MHz とした．

図 12 において，横軸はノード数，縦軸は無負荷状態のネットワークにおけるパケット遅延を表しており，負荷が小さいほど，低遅延な通信を実現できていることが分かる．予測ルータのネットワークは予測成功時にパケット転送サイクル数が減るので，すべてのノード数，予測アルゴリズムにおいて，パケット遅延が Conv より小さくなっている．実際，SS は Conv と比べ，64 コアのネットワークにおいて 14.2%，256 コアのネットワークにおいて 23.7% 通信遅延を削減できている．

5.　ま と め

本論文では NoC の通信遅延を減らすため，予測機構を持ったオンチップルータの構造について検討した．予測ルータにおいて，予測が成功すれば RC と SA 処理は不要であり，ヘッダフリットは prediction datapath を通って 1 サイクルで転送される．一方，予測が失敗するとヘッダフリットは backup datapath を通って 3 サイクルで転送される．ただし，このルータアーキテクチャには次の 3 つの問題点がある．1) prediction datapath によって面積が増える．2) prediction datapath によって消費エネルギーが増える．3) 予測失敗時に prediction datapath から無効なフリットが間違った次ホップに対して送出される．まず，1) の問題を緩和するために，prediction datapath と backup datapath が同じバッファを共有するようにした．次に 2) の問題を緩和するために，予測の成否を 1 サイクルで判断し，予測が当たれば backup datapath の動作を止め，予測が外れれば prediction datapath の動作を止めるようにした．3) の問題を回避するために，各入力チャネルから 1 ホップ先のチャネルに kill 信号を張り，無効フリットの伝搬を止めるようにした．評価の結果，予測ルータは通常のルータと比べて，面積と転送エネルギーがそれぞれ最大で 23.4% と 10.0% 増加したものの，64〜256 コアのネットワークにおいて通信遅延が 14.2〜23.7% 減少した．

文　献

[1] W. J. Dally and B. Towles: “Route Packets, Not Wires: On-Chip Interconnection Networks”, Proceedings of the Design Automation Conference (DAC’01), pp. 684–689 (2001).

[2] 鯉渕, 吉永, 村上, 松谷, 天野：“予測機構を持つルータを用いた低遅延チップ内ネットワークに関する研究”, 先進的計算基盤システムシンポジウム (SACSIS’08) 論文集 (2008). (to appear).

[3] 吉永, 村上, 鯉渕：“2-D トーラスネットワークにおける動的通信予測の効果”, 先進的計算基盤システムシンポジウム (SACSIS’07) 論文集, pp. 219–226 (2007).

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

[5] A. Kumar, L.-S. Peh, P. Kundu and N. K. Jha: “Express Virtual Channels: Towards the Ideal Interconnection Fabric”, Proceedings of the International Symposium on Computer Architecture (ISCA’07), pp. 150–161 (2007).

[6] G. Michelogiannakis, D. N. Pnevmatikatos and M. Katevenis: “Approaching Ideal NoC Latency with Pre-Configured Routes”, Proceedings of the International Symposium on Networks-on-Chip (NOCS’07), pp. 153–162 (2007).

[7] H. Matsutani, M. Koibuchi, D. Wang and H. Amano: “Adding Slow-Silent Virtual Channels for Low-Power On-Chip Networks”, Proceedings of the International Symposium on Networks-on-Chip (NOCS’08) , pp. 23–32 (2008).