

NoC ルータのためのリンク間共有法におけるパイプライン・ステージの検討

Examination of Pipeline Stage in Link-Sharing Method for NoC Router

深瀬 尚久†
Naohisa Fukase

三浦 康之†
Yasuyuki Miura

渡辺 重佳†
Sigeyosi Watanabe

1. まえがき

チップ内におけるコア間の接続方法として、接続にネットワークを使用する「ネットワークオンチップ(NoC)」という技術がある。NoC で使用されるルータには、フリットを一時的に格納するバッファが取り付けられており、このバッファの容量が大きいほど NoC 全体の性能は向上する。しかし NoC では、使用できるハードウェアコストに限りがあるため、少量のバッファを有効に利用することが重要である。

ルータ内のバッファを効率的に使用するため、各物理リンク内の仮想チャンネル間[1][2][3]や物理リンク間でバッファを共有する手法などが提案されてきた。しかし、後者は理想状態として評価に使用されることはあったが[3]、実装向きでないためこれまで用いられることはあまりなかった。

そこで以前我々は、物理リンク間でバッファを共有する手法の実装法を提案した[4]-[7]。この手法では、共有メモリをブロックという単位に分割し、ブロック単位で各チャンネルに割り振ることで、ハードウェアコストの増加を抑えている。その結果、提案手法は少ないハードウェアコストで実装でき、高い通信性能を持つことを確認した。

本稿では、この手法のパイプライン・ステージの検討を行い、ステージ数増加による遅延の増加などについての考察を行う。

2. 従来法

ここでは、提案手法について説明する前に一般的に使用されるルータと本手法と類似した手法である各リンク内の仮想チャンネル間でバッファを共有するルータについて説明する。

2.1 従来法 1 : 未共有

直接結合網を用いた NoC では、プロセッサコアとルータの組によって 1 つの PE が構成される。ルータ回路には、通信経路となる物理リンク同士を接続するためのクロスバスイッチが配置されている。そしてクロスバスイッチの入力側には通信の平滑化のために、バッファとして FIFO が取り付けられている。

一般的に使用されるルータは、各チャンネルに個別のバッファを持つ構造をしている[8][9][10][11]。しかしこのような構造の場合、未使用のチャンネルに割り当てられているバッファが有効に活用できない可能性があった。

2.2 従来法 2 : チャンネル間共有

各物理リンクには、デッドロックの回避やヘッドオブラインブロッキングによる影響の軽減を目的に複数の仮想チャンネルが設けられる場合がある[12]。

仮想チャンネルは、物理的に 1 本の物理リンクを共有するため、同時に複数の仮想チャンネルに対するパケットの入出力が生じない構造になっている。そのため、メモリを効率的に使用するため、1 本の物理リンクにおける複数の仮想チャンネルで 1 個のメモリを共有して使用する方法が用いられることがある[1][2][3]。従来法 2 の構成を図 1 に示す。図 1 のようにこの手法のルータは、各物理リンクに 1 個の Flit Payload Buffer という共有バッファを持ち、Buffer Management Unit 中の Free Pool や Header Control Block の Block Info でバッファを管理する。この手法では、フリットが入力されると、Flit Payload Buffer から 1 フリット分のメモリを動的に割り当てる。その際、割り当てられたメモリの制御を行うため、Free Pool と Block Info が使用される。Free Pool は、どのチャンネルにも取得されていないメモリのポインタを格納する。Block Info は、チャンネルが取得しているメモリのポインタを格納する。

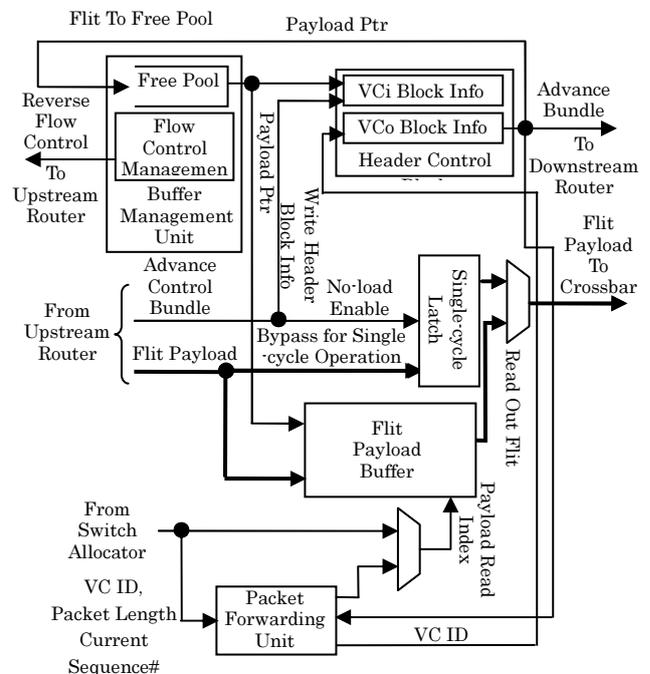


図 1. 従来法 2 の構成

† 湘南工科大学, Shonan Institute of Technology

3. 提案手法：リンク間共有法

物理リンクをまたぐバッファの共有法がこれまで使用されてこなかった主な理由としては、ハードウェアコストの大幅な増加がある。リンク間の共有では複数の物理リンクからの同時アクセスに対応するため、共有メモリにマルチポートメモリを使用する必要がある。しかし、一般的なマルチポートメモリでは、ハードウェアコストが大幅に増大するという問題がある。

提案手法と使用するバンク型マルチポートメモリの構造をそれぞれ図 2, 3 に示す。図 2 のように提案手法のルータでは、各チャンネルが専用のバッファである専有部を持ち、入力ポートと専有部の間に全ての物理リンクで 1 個の共有メモリを持つ構造となっている。提案手法ではハードウェアコストの問題を解決するため、共有メモリにバンク型マルチポートメモリ [13]-[17] を使用する。図 3 のように、バンク型マルチポートメモリはバンクと呼ばれる少数のポートを持つメモリをクロスバスイッチなどのスイッチによって入出力ポートと接続することで構成される。この構造は通常のマルチポートメモリのように、各メモリセルのマルチポート化を行う必要がないため、ハードウェアコストの増加を抑えることができるという利点がある。しかしこのマルチポートメモリでは、同バンク内のメモリ領域に対して同時にアクセスすることができないため、従来法のようなフリットサイズのメモリ領域を一つずつ割り当てる手法では、同時アクセスの問題が発生する。

バンク型マルチポートメモリの問題を解決するため、提案手法ではブロック単位共有を提案している。この手法は、バンク型マルチポートメモリのバンク一つをブロックという単位とし、メモリの割り当てと解放をこの単位で行う手法である。こうすることにより、各バンクへの入出力がそのバンク(ブロック)を取得したチャンネルからのみ行われるようになり、同時アクセスの問題が解決される。また、これにより管理対象がフリットサイズのメモリ数からブロック数になるため、制御用回路のハードウェアコストも大幅に削減できる。

また、リンク間の共有ではバッファの共有により、メモリを取得することができない仮想チャンネルや物理リンクが発生する可能性があり、デッドロックを防ぐことが

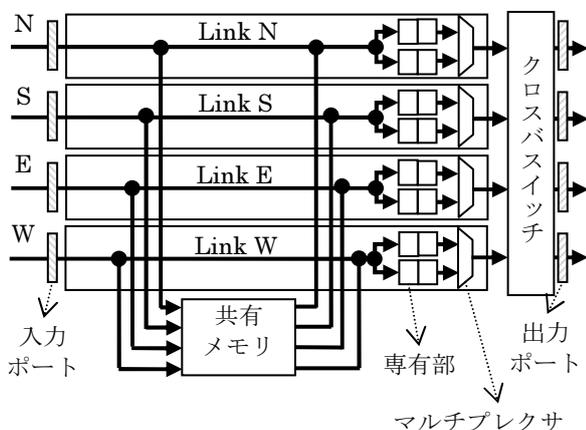


図 2. 提案手法のルータ構造

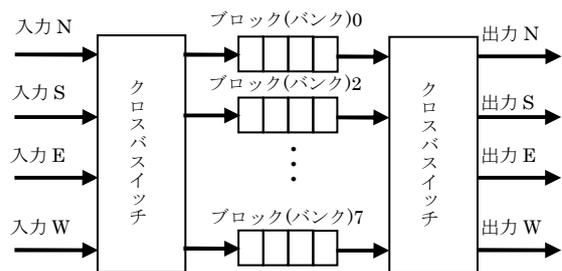


図 3. バンク型マルチポートメモリの構成

できなくなるという問題もある。この問題を解決するため、本提案手法では各チャンネルに通信を続けることが可能な最低限の容量の専有部というバッファを設けている。こうすることにより、各チャンネルはメモリの取得が不可能な場合であっても通信を続けることができ、デッドロックを回避することができる。

4. パイプライン

4.1 一般的ルータのパイプライン

一般的に使用されているルータのパイプラインを図 4 に示す [18]。図 4 のように一般的なパイプラインは次の 4 つの処理を 3 段のパイプラインにより実行する。

1) Routing Computation (RC)

ヘッダフリットの情報から出力リンクを決定する。

2) Virtual Channel Allocation (VA)

出力する仮想チャンネルを割り当てる。

3) Switch Allocation (SA)

クロスバスイッチのアービトレーションと設定を行う。

4) Switch Traversal (ST)

フリットがクロスバスイッチを通過する。

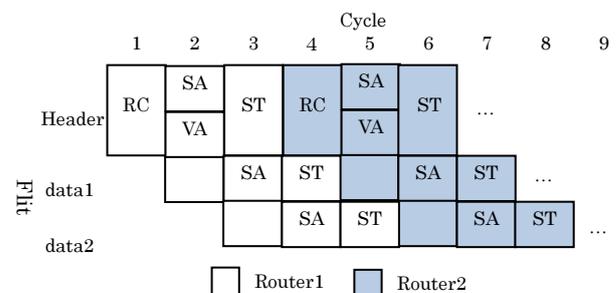


図 4. 一般的なルータのパイプライン

4.2 提案手法のパイプライン

一般的なルータと異なり、提案手法に入力されたフリットは状況に応じて 2 種類の経路を通る。

一つは、入力ポート到着後すぐに専有部に送られる経路である。この経路は、通信のブロックが発生せず、専有部のみで通信が可能である場合に使用される。以後この経路を「経路 1」とする。

二つ目は、入力ポートに入力後、一度共有メモリに入力され、その後で専有部に入力される経路である。この経路は、混雑により通信のブロックが発生し、専有部の

みでは後続のフリットを受け入れられない場合に使用される。以後この経路を「経路 2」とする。

経路 1 のパイプラインを図 5 に示す。図 5 のように経路 1 は、一般的なルータと同様に 3 ステージのパイプラインから構成される。ただし、図 5 にあるように 1 ステージ目で In-judge(IJ)ステップを行う。IJ は、「共有メモリを使用するか否か」と「新たにブロックを取得するか否か」の判定を行うステップである。パケットの出力リンクは、共有メモリを使用するか否かとは無関係に決まるので、RC と IJ は並列に処理できる。

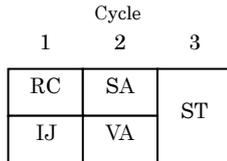


図 5. 経路 1 のパイプライン

経路 2 のパイプラインを図 6 に示す。提案手法において共有メモリを使用する場合に、必要になるステップは次のようになる。

1) IJ(In-Judge)

入力フリットが共有メモリを使用する必要があるかを判定する。同時に、共有メモリを使用する場合、当該チャネルが新たにブロックを取得する必要があるか否かの判定を行う。

2) SiA(Switch-i Allocation)

制御情報の更新を行うと同時に、バンク型マルチポートメモリの入力用クロスバスイッチの設定を行う。

3) SiT(Switch-i Traversal)

SiA ステップに成功した場合、フリットをバンク型マルチポートメモリの入力用クロスバスイッチを通過させて、共有メモリに格納する。

4) SoA(Switch-o Allocation)

バンク型マルチポートメモリの出力用クロスバスイッチの設定、およびブロック解放処理を同時に行う。

5) SoT(Switch-o Traversal)

SoA ステップに成功した場合、バンク型マルチポートメモリの出力用クロスバスイッチにフリットを通過させ、専有部に格納する。

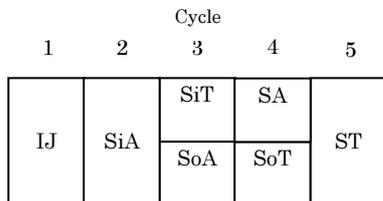


図 6. 経路 2 のパイプライン

図 6 のように、経路 2 は経路 1 に比べてパイプラインが 2 ステージ増加する。しかし、以下の理由により、経路 2 のパイプラインによる遅延は隠ぺいされる。

・ネットワークが混雑しておらず、専有部に空きがある場合、3 段パイプラインに従って処理されるため、従来法と同じ量の遅延となる。

・ネットワークが混雑するに従って、専有部に空きが少なくなると共有部が使用される。専有部からクロスバスイッチに送られるパケットがブロッキングされることにより、専有部中のフリットが増えることになるが、そのようなブロッキングを 1 回まで許容できるように専有部を設計すれば(専有部のフリット数を 2 以上にすれば)、共有部を使用したフリットのパイプラインがスムーズに動作する。

提案するパイプラインの動作例を図 7 に示す。図 7 では、先頭のフリットがブロックされたことにより、3 つ目に入力されたフリットが経路 2 を使用する場合である。

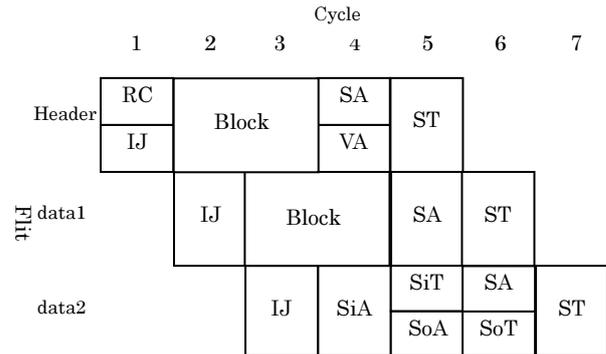


図 7. パイプラインの動作例

5. 性能評価

提案手法の動的通信性能をソフトウェアシミュレータによって評価する。シミュレータでは、各 PE で 1 サイクルごとに指定された確率でパケットを生成し、ランダムに選択した他の PE に送信する。これらの処理をパケットの発生確率を変化させて実行し、グラフを作成する。

平均スループットに対する、共有メモリの利用率を図 8 に示す。図 8 の横軸は、PE 数が 64、合計バッファ容量が 64、パケット長が 64 の場合の平均スループットである。縦軸は、それぞれの条件において、ルータに滞在中のパケットが共有メモリを使用する比率を示している。平均スループットが低い場合(ネットワークが混雑していない場合)、専有部のみを使用した転送が多いため、一般的な手法と提案手法のふるまいに大きな差が見られない。平均スループットが高くなる(ネットワークが混雑する)に従って、共有部を用いた転送の割合がゆるやかに増加し、スループットが 0.2 flit/Cycle・PE を超えて従来法と提案手法の違いが現われるあたりで、共有部の割合の増加が激しくなる。ただしその場合も、4.2 節において述べたように、共有部の通過に伴う遅延はほぼ隠蔽される(前方のフリットのブロッキングによって隠される)ため、遅延時間の増加による通信性能のロスが目立たない。

図 9 に PE 数が 64、合計バッファ容量が 64、パケット長が 64 の場合の各実装法のシミュレーション結果を示す。この図 9 のグラフは、横軸が平均スループット、縦軸が平均転送時間となっている。またグラフには、提案手法の性能向上幅を確認するために、従来法 1(no-sharing)と従来法 2(By-flit and channel sharing)を、ブロック単位共有による性能の低下幅を確認するために、リンク間共有でフリットサイズのメモリ割り当てを行う手法(By-flit and Link-

sharing)の結果も同時に載せている。結果より、提案手法のようなリンク間の共有は、そのほかの手法と比較して十分に性能が向上することが確認できる。また、フリットサイズでメモリの割り当てを行う手法と比較した結果、ブロック単位共有による性能低下はほとんどないことが確認出来る。

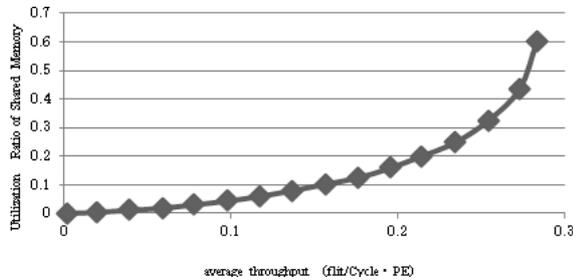


図8. 平均スループットに対する共有メモリの利用率
(PE数:64, 合計バッファ:64, パケット長:64)

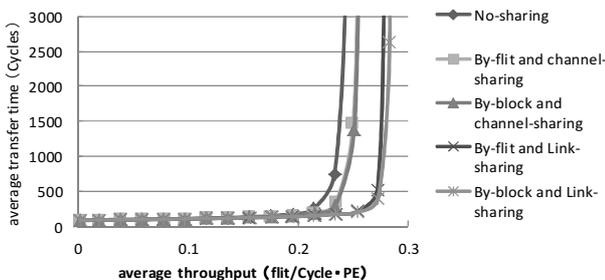


図9. 動的通信性能の評価結果
(PE数:64, 合計バッファ:64, パケット長:64)

6. まとめ

本稿では、以前我々が提案した NoC ルータにおけるバッファのリンク間共有法のパイプラインについて検討を行った。その結果共有メモリを使用する場合、パイプラインが一般的なルータのパイプラインよりも2ステージ増加することが分かった。ただしこのパイプラインの増加による遅延は、専有部の容量を確保しておけば防げる。

今後は、制御部の論理回路を実装に伴うパイプラインステージの見直しと、詳しいハードウェアコストの見積もりを行う。

参考文献

- [1] A.Kumary, P.Kunduz, A.P.Singh, L.-S.Pehy, N.K.Jhay, A 4.6Tbits/s 3.6GHz single-cycle NoC router with a novel switch allocator in 65nm CMOS, 25th International Conference on Computer Design(ICCD 2007), pp.63-70, Oct. 2007.
- [2] Gregory L. Frazier, Yuval Tamir, The design and implementation of a multiqueue buffer for VLSI communication switches, Proceedings of the International Conference on Computer Design Cambridge, Massachusetts, pp.466-471, Oct.1989.
- [3] Yuval Tamir, Gregory L. Frazier, Dynamically-Allocated Multi-Queue Buffers for VLSI Communication Switches, IEEE Trans. Computers, Vol.41, No.6, pp.725-737, 1992.
- [4] 深瀬尚久, 三浦康之, 直接結合ネットワークのルータ回路におけるバッファの有効利用, 情報処理学会創立 50周年記念(第72回)全国大会, 2M-2, 2010.03.
- [5] 深瀬尚久, 三浦康之, 渡辺重佳, 直接結合ネットワークにおけるバッファのリンク単位共有法, 情報処理学会第73回全国大会, 6H-1, 2011.03.
- [6] Naohisa Fukase, Yasuyuki Miura, Shigeyoshi Watanabe, Link-Sharing Method of Buffer in Direct-Connection Network, The 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp.208-213, 2011.08.
- [7] 深瀬尚久, 三浦康之, 渡辺重佳, NoC ルータにおけるリンク間共有法の通信性能の評価, 情報処理学会第74回全国大会, 5K-6, 2012. 03.
- [8] M. Ni and P. K. McKinley, A Survey of Wormhole Routing Techniques in Direct Networks, Proc of the IEEE, Vol. 81, No. 2, pp. 62-76, 1993.
- [9] Yasuyuki Miura, Masahiro Kaneko, Shigeyoshi Watanabe, Adaptive Routing Algorithms and Implementation for Interconnection Network TESH for Parallel Processing, The 35th IEEE Conference on Local Computer Networks (LCN), 2010.
- [10] T.C.So, S.Oyanagi, K.Yamazeki, Speculative Selection in Adaptive Routing on Interconnection Networks, 情報処理学会論文誌. コンピューティングシステム, Vol.44, pp.147-156, 2003.
- [11] M.Koibuchi, K.Anjo, Y.Yamada, A.Jouraku and H.Amano, "A Simple Data Transfer Technique Using Local Address for Networks-on-Chips", IEEE Transaction on Parallel and Distributed Systems, vol.17, No.12, pp.1425-1437, 2006.
- [12] W.J.Dally, Virtual-Channel Flow Control, IEEE Trans on Parallel and Distributed Systems, vol.3, No.2, pp.194-205, 1992.
- [13] Y.Tatsumi et al., "Fast quadratic increase of multiport-strage-cell area with port number," Electronics Letters, Vol.35, No.25, pp.2185-2187, 1999.
- [14] Michael Golden et al., "A 500MHz write-bypassed, 88-entry, 90bit register file," Proc. of Symposium on VLSI Technology, Session C11-1, 1999.
- [15] H.J Mattausch, K.Kishi and T.Gyohten, "Area-efficient multi-port SRAMs for on-chip data-storage with high random-access bandwidth and large storage capacity," IEICE Trans. Electron., Vol.E84-C, No.3, p410, 2001.
- [16] 井上他, K 出力可能な閉そく網と非閉そく網を階層的に用いたバンク型マルチポートメモリの構成と評価, 電子情報通信学会論文誌 A, Vol.J89-A, No.10, pp.774-789, 2006.
- [17] 佐々木他, オンチップマルチプロセッサ用共有キャッシュの実現方式の検討とその性能面積評価, 電子情報通信学会論文誌 D-I, Vol.J87-D-I, No.3, pp.350-363, 2004.
- [18] W. J. Dally and B. Towles. Principles and Practices of Interconnection Networks. Morgan Kaufmann Publishers, 2004