6H-2

メニーコアプロセッサのオンチップネットワークに関する研究

姜 軒 † 吉瀬 謙二 †
東京工業大学大学院 情報理工学研究科 †

1 はじめに

プロセッサの消費電力を抑え，性能を向上するため，近年のプロセッサアーキテクチャではコア数が増加の一途を辿っている．しかし，コア数の増加に伴い，コア間の通信遅延，スループットがアプリケーションに与える影響が益々大きくなってきている．コア間の通信にはNoC(Network-on-Chip)[2] が広く用いられるため，低通信遅延，ハイスループットオンチップルータの開発が急務である．

このようなハイスループットオンチップルータの一つに DSB(Distributed Shared-Buffer) オンチップルータ [1] が提案されている．DSB ルータはハイスループットであるが通信遅延の増加が深刻な問題である．そこで，本稿では，低遅延化にするため，DSB オンチップルータを対象とする，パイプラインをバイパスする制御方法を提案する．

2 DSB(Distributed Shared-Buffer) ルータ

2.1 アーキテクチャ[1]

図 1 に DSB ルータアーキテクチャを示す．このルータは入力バッファ，二つクロスバ，n 本の 1 出力ポート 1 入力ポートの中間メモリ，アビター回路から構成されており，各入力ポートに i 本の仮想チャンネルを持つ．**図** 2 に DSB ルータのパイプライン構造を示す．RC と TS を並列に処理し，VA と RC を並列に処理する．フリットが入力されてから次のホップのルータへ転送されるまで，最低 5 サイクル/ホップかかる．DSB ルータアーキテクチャは 5 段のパイプラインステージから構成される．入力されるヘッドフリットは RC(route computation) ステージで次のホップルータの出力ポートが計算され，TS(TimeStamping) ステージで，入出力のスケジューリングが行われる．また，ヘッダフリットは VA(Virtual channel Allocation) ステージで，出力仮想チャンネルを割り当てられ，CR(Conflict Resolution) ステージで，格納する中間メモリの割り当てが行われる．フリットは XB1+MM_WR(Crossbar+Memory Write) ステージで，クロスバ 1 を通過し，中間メモリに書き込まれ，XB2+MM_RD(Crossbar+Memory Read) ステージで，中間メモリから読み込まれ，クロスバ 2 を通過する．最後にフリットは LT(Link Traversal) ステージで，隣接するルータへ転送される．

フリットは TS ステージで，以後の中間メモリから読み込まれるタイミングが計算され，次のポップのルータへ転送される予定時刻が決定される．すなわち，各フリットが出力ポートを通過するタイミングをユニークにさせ，出力ポート競合を回避するため，クロスバ調停を行う．

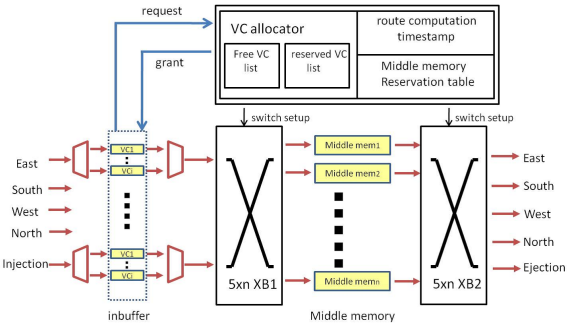

図 1: DSB ルータアーキテクチャ

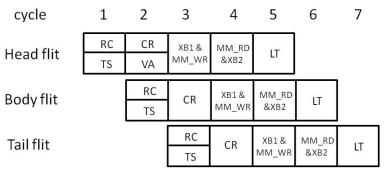

図 2: DSB のパイプライン構造

2.2 問題点

近年 NoC で主流となっているルータとして，シンプルな構成であるインプットバッファルータがある．インプットバッファルータは 3 段パイプラインステージ構成である [2]．DSB ルータはインバッファルータと比較し，重要な問題がある．DSB ルータはインプットバッファルータよりパイプライン段数が多く，通信遅延が大きい．たとえば，文献 [1] により，インプットバッファルータとの比較で，遅延のオーバーヘッドが大きい．この問題はコア数の増加に伴い，更に深刻になり，メニーコアプロセッサに適用できなくなる見通しである．

3 パイプラインをバイパスする DSB ルータ

ルータのパイプライン段数が通信遅延に大きく影響するため，ルータのパイプライン段数を減らすバイパス制御を提案する文献がいくつか挙げられる [3][4]．それらの研究では，パイプラインをバイパスする制御を用いて，コア間の通信遅延を削減することが明らかにされている．

そこで，本研究はパイプライン段数の多い DSB ルータを低遅延化にするため，パイプラインをバイパスする制御方法を提案する．このパイプラインをバイパスする BSD ルータは，従来の DSB の第 3 パイプラインステージ (XB1+MM_WR) をスキップする制御を加えることで，バイパス条件を満たす場合に，フリットを 4 サイクルで転送できる．

図 3 に，提案する DSB ルータのアーキテクチャ図を示す．この DSB ルータは各入力バッファの右のディマルチプレクサの出力配線から相対する中間メモリの出力

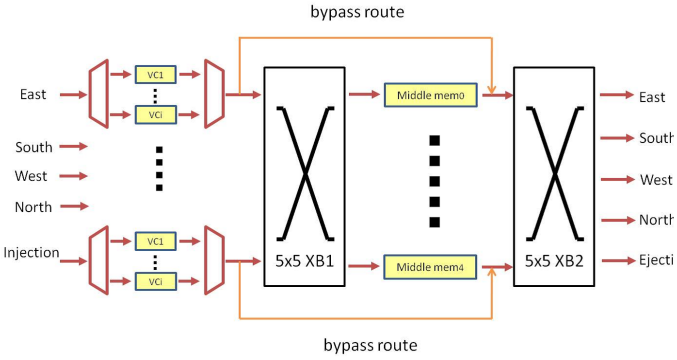

図 3: バイパスする DSB ルータアーキテクチャ

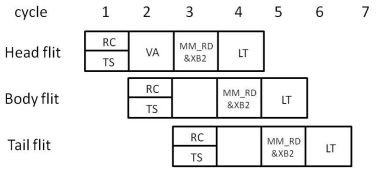

図 4: バイパスが成功した場合のパイプライン構造

配線にバイパスルート配線が接続されている．

フリットがバイパスされる際の処理について説明する．フリットは入力バッファからディマルチプレクサを通過し，中間メモリの右のクロスバを経由し，隣接ルータへ転送される．この場合，クロスバ 1 と中間メモリを使わず，フリットを転送するため，従来の第 3 パイプラインステージ (XB1+MM_WR) をスキップできるようになる．TS 処理では，バイパスされるフリットは通常フリットより 1 サイクル早い隣接ルータへ転送される予定時刻が決定される．

図 4 にバイパスが成功した場合のパイプライン構造図を示す．フリットが入力されてから次のホップのルータへ転送されるまで，4 サイクル/ホップを要する．

バイパスルート配線で転送する際，フリットの転送のタイミングによっては競合が起きる可能性がある．そのため，以下の三つの場合において，バイパスを行わない．

1. バイパスされるフリットと中間メモリに格納されているフリットの出力ポートとその出力のタイミングが同じである場合
2. バイパスされるフリットと第二パイプラインステージに入っているフリットの出力ポートとその出力のタイミングが同じである場合
3. バイパスされるフリットと中間メモリに格納されているフリットのクロスバ 2 の入力ポートとその入力のタイミングが同じである場合

バイパス条件を満たさない場合に従来手法と同様にフリットを 5 サイクルで転送し，バイパス制御に関するペナルティがない．また，同じタイミングで同じ出力にバイパスできるフリットが複数存在する場合は，優先度が一番高いフリットのみをバイパスルート配線で転送する．

4 評価

フリット・サイクルレベルのネットワークシミュレータを用いて，評価を行う．ネットワークトポロジは 2D

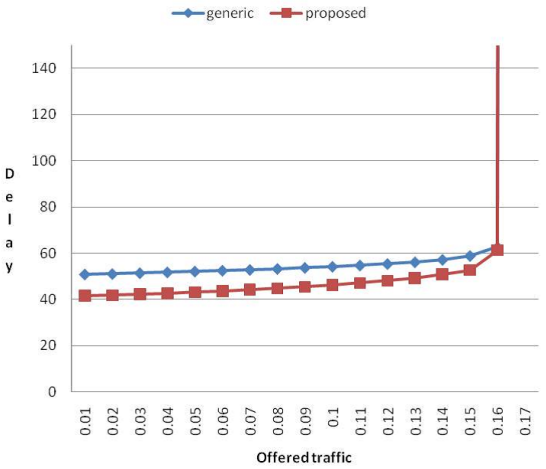

図 5: bit-complement traffic

メッシュ，ネットワークサイズは 8×8 とする．ウォームアップサイクル数を 5,000 に設定し，測定サイクル数を 100,000 に設定する．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る．シミュレーションに用いる通信トラフィックパターンとして，ビットコンプリメントを用いる．

図 5 に結果を示す．縦軸はパケットの遅延，横軸はオーファドトラフィックである．ビットコンプリメントにおいて，遅延を最大 18.3%削減し，平均 14.3%の削減を達成している．オーファドトラフィックが低いところにバイパスの成功率が高く，遅延の削減率が一番高い．オーファドトラフィックが高いところにほとんどバイパスできず，パイプラインをバイパスする DSB ルータが BSD と同じ通信遅延である．

5 まとめ

本稿では，通信遅延の削減を目指し，DSB ルータを対象にするバイパス制御方法を提案した．フリット・サイクルのネットワークシミュレータを用いて，性能評価をし，通信遅延の削減を確認した．今後の課題として，ハードウェア量と消費エネルギーの評価を行うことが挙げられる．

謝辞

本研究の一部は, 科学技術振興機構・戦略的創造研究推進事業 (CREST) 「アーキテクチャと形式的検証の協調による超ディペンダブル VLSI」の支援による．

参考文献

[1] R. S. Ramanujam, V. Soteriou, B. Lin, L. Peh Design of a High-Throughput Distributed Shared-Buffer NoC Router NOCS '10

[2] W. J. Dally ,B. Towles Route Packets, NotWires: OnChip Interconnection Networks Morgan Kaufmann 2004

[3] Amit Kumar, Li-Shiuan Peh, Partha Kundu, Niraj K. Jha Express virtual channels: towards the ideal interconnection fabric ISCA'07

[4] Amit Kumar, Li-Shiuan Peh, Niraj K. Jha Token Flow Control MICRO 41