

ルータ回路のバッファ共有法における部分的共有法の性能評価 Examination of Pipeline Stage in Link-Sharing Method for NoC Router

深瀬 尚久[†] 三浦 康之[†] 渡辺 重佳[†]
Naohisa Fukase Yasuyuki Miura Sigeyosi Watanabe

1. はじめに

近年、チップ内の各コアの接続にネットワークを用いる NoC(Network on Chip)の研究が盛んに行われている。NoC で使用されるルータには、処理中に転送データを一時的に格納するバッファが取り付けられている。そして、このバッファの容量が大きいほど混雑時の性能が向上する。これは、ルータ一つ一つに搭載されているバッファの容量が一般的に転送データより小さいため、バッファの容量が大きいほど一つのデータがまたぐことになるルータが少なくなるためである。しかし NoC では、使用できるハードウェアコストに制限があるため、使用できるバッファの容量も限られている。

以前、我々は NoC ルータ内のバッファを物理リンク間で共有することにより、少量のバッファを有効利用する方法を提案した[1]-[5]。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。しかし提案手法を評価した結果、軽減は可能ではあるがハードウェアコストは 1.5~2 倍とやや大きくなることが確認できた。これはすべてのリンクを共有したために、制御に必要な情報が増大し、クロスバスイッチなどの回路量も増大したためである。

本稿では、提案手法のハードウェアコストをより削減するための実装形態を提案し、ハードウェアコストと通信性能についての評価を行う。

2. 提案手法

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。一般的なルータは、チャネルごとに等量のバッファを持つ構造となっている[6]-[9]。しかし、この構造では、未使用のチャネルに割り当てられたバッファが有効に活用されない場合がある。そこで、提案手法[1]-[5]では図 1 のように、一つのメモリを複数の物理リンクで共有することで、バッファを効率的に利用する。

物理リンクをまたぐバッファの共有法は、複数の物理リンクからの同時入出力の処理を行う必要があるため、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、各メモリセルをマルチポート化する通常のマルチポートメモリは、ハードウェアコストを大幅に増大させるという問題がある。そこで、提案手法ではハードウェアコストの増大を抑えるため、共有メモリにバンク型のマルチポートメモリ[10]-[14]を使用する。図 2 にバンク型マルチポ

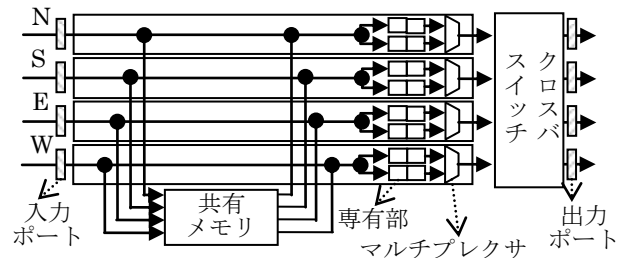


図 1. 提案手法の構成

トメモリの構成を示す。図 2 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。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、同一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、メモリ領域を必要分ずつ割り当てる手法では、同一バンクへの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の数がフリットサイズのメモリ領域数からブロック数に減少するため、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チャネルが発生する可能性があり、デッドロックを防ぐことが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ネルに通信に必要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、各チャネルは共有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。

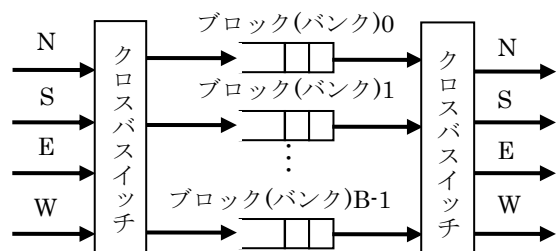


図 2. バンク型マルチポートメモリの構成

[†] 湘南工科大学 Shonan Institute of Technology

一般的に使用されているルータのパイプラインは 3 つのステージで構成され、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。

提案手法では、共有メモリを使用する場合としない場合で二つの経路が存在する。以降は使用しない場合を経路 1、使用する場合を経路 2 とする。経路 1 のパイプラインを図 3 に示す。経路 1 のパイプラインは、一般的なルータと同じく 3 段である。ただし、1 段目に共有メモリを使用するかを判断する In-judge(IJ)ステップを行う。経路 2 のパイプラインを図 4 に示す。経路 2 では、経路 1 の処理に加えて、以下の処理が必要になる。

1) SiA(Switch-i Allocation)

割り当て情報の更新と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。

2) SiT(Switch-i Traversal)

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。

3) 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とブロック解放処理を同時に行う。

4) SoT(Switch-o Traversal)

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。

図のように、経路 2 のパイプラインは経路 1 より 2 段分大きくなる。しかし、専有部の容量が 2 以上であれば、経路 2 が使用されるのは通信のブロックが 2 回以上発生した場合なので、段数の差は吸収される。

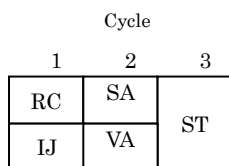


図 3. 経路 1 のパイプライン

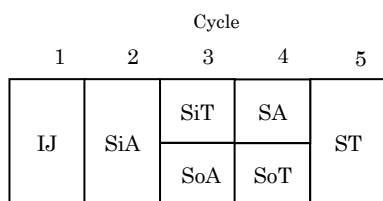


図 4. 経路 2 のパイプライン

3. ニリンク共有

以前、我々が提案した手法は、図 1 のようにすべてのリンクで一つのメモリを共有する構成であった。このような手法を以後は「全リンク共有」とする。全リンク共有のよ

うな構成は、ブロック割り当てのための回路や処理の複雑化が大きな問題となる。それを軽減するために本稿では、共有する範囲をニリンクずつとした実装について評価する。このような実装法を以後は「ニリンク共有」とし、その構成を図 5 に示す。また、図 5 にあるように今回使用するニリンク共有においては、E と W、N と S のリンクでそれぞれ一つのメモリを共有する構成とする。これは次元順ルーティングにおいて、共有したリンクにそれぞれ入力されたパケットが同じリンクを待つパターンが若干ながら少ないためである。

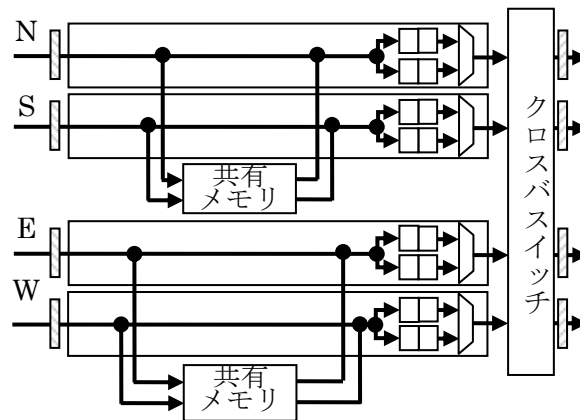


図 5. 2リンク共有のルータ構造

4. ハードウェアコストの評価

本節では、以前の研究で提案手法のハードウェアコストを見積もるために使用した式[3]を用いて各実装のトランジスタ数を求め、評価する。このときニリンク共有は、物理リンク数が 2、仮想チャネル数が 2 の場合の提案手法を 2 つ用意するものとして計算している。表 1 に結果を示す。表中の記号はそれぞれ以下の意味を表す。

- B: 全物理リンクのブロックの合計数
- C: 全物理リンクの仮想チャネルの合計数
- L: 物理リンク数
- F: 1 ブロックあたりに入るフリットの数
- W: 1 フリット辺りのデータのビット数

また表中の「バッファ本体」は実際にパケットを保存するバッファ、「制御用メモリ」は制御に必要な情報を保持するメモリ要素、「制御用論理」はブロック割り当て制御などに必要になる論理回路、「メモリ周辺」はマルチポートメモリ周辺の回路のトランジスタ数を表している。そして、最後の列の「増加率」に各実装形態の未共有と比較した場合のトランジスタ数の増加率を示している。また、ニリンク共有のブロック数(B)の項目のかっこ内の数字は、ルータ全体の合計ブロック数で、非かっこの数値はニリンク共有の各共有部のブロック数となっている。

全リンクの部分では、ブロック数が減るほどにハードウェアコストが大きく減少しており、ブロック数が 8 以下ならば 2 倍以下で実装可能であることが分かる。ニリンク共有では、全リンク共有にくらべてすべての項目でトランジスタ数が大きく減少しており、ブロック数が 4 つの時などでは、全リンク共有のブロック数が 2 の場合とほぼ同じハードウェアコストで実装できることが分かる。

表 1. 各実装形態のトランジスタ数

W	手法	L	C	B	F	バッファ 本体	制御用 メモリ	制御用論理	メモリ周辺	合計	未共有	増加率
64	二リンク	4	8	8(16)	4	24576	2340	5368	25280	57564	29832	1.93
64		4	8	4(8)	8	24576	1116	3432	12544	41668	29832	1.40
64		4	8	2(4)	16	24576	516	2368	6224	33684	29832	1.13
64	全リンク	4	8	16	4	24576	4422	10856	50944	90798	29832	3.04
64		4	8	8	8	24576	2010	6456	25280	58322	29832	1.96
64		4	8	4	16	24576	918	4080	12544	42118	29832	1.41
64		4	8	2	32	24576	402	2800	6224	34002	29832	1.14
128	二リンク	4	8	8(16)	4	49152	2340	5368	49856	106716	54408	1.96
128		4	8	4(8)	8	49152	1116	3432	24832	78532	54408	1.44
128		4	8	2(4)	16	49152	516	2368	12368	64404	54408	1.18
128	全リンク	4	8	16	4	49152	4422	10856	100096	164526	54408	3.02
128		4	8	8	8	49152	2010	6456	49856	107474	54408	1.98
128		4	8	4	16	49152	918	4080	24832	78982	54408	1.45
128		4	8	2	32	49152	402	2800	12368	64722	54408	1.19

5. 通信性能の評価

評価にはソフトウェアシミュレータを使用する。評価では、通信パターンにユニフォームトラフィック、ルーティングアルゴリズムが次元順ルーティングである 2 次元トラス網を使用する。

図 6 に全リンク共有におけるブロック数ごとの結果を示す。図 6 は PE 数が 64、バッファ総量が 32、パケット長が 16 ビットの場合の結果である。図中の「未共有」は共有を行わなかった結果、「フリット単位」は、全リンク共有は行うがブロック単位の共有を行わずに、メモリの割り当てをフリット 1 つ分のメモリ領域で行った場合の結果、B2、4、8 はそれぞれブロック数が 2、4、8 である提案手法の結果となっている。この結果より、ブロック数が 8 の提案手法は、ブロック単位共有を行わなかった場合と同等の性能を持つことが分かる。たいして、ブロック数が 2、4 のものは性能が若干落ちることが分かる。そのことから全リンクを共有した提案手法における性能上の最適なブロック数は 8 個であることが分かる。

続いて、二リンク共有におけるシミュレーション結果を図 7-9 および表 2 に示す。図と表中の「2 リンク B2、4、8」

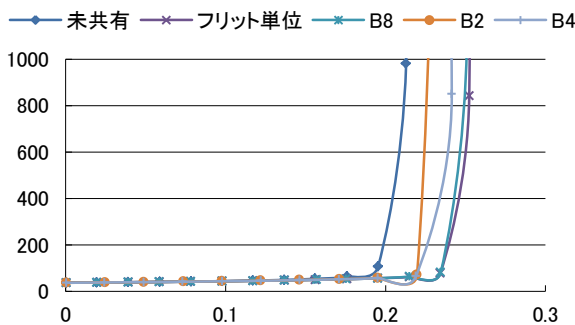


図 6. 全リンク共有のシミュレーション結果

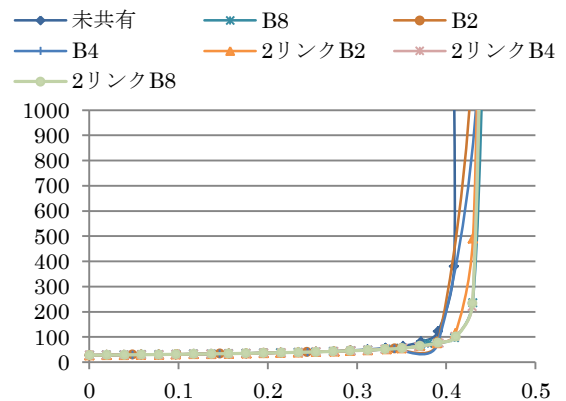


図 7. 通信性能のシミュレーション結果 (PE-No:16, Buffer:64, Packet:16)

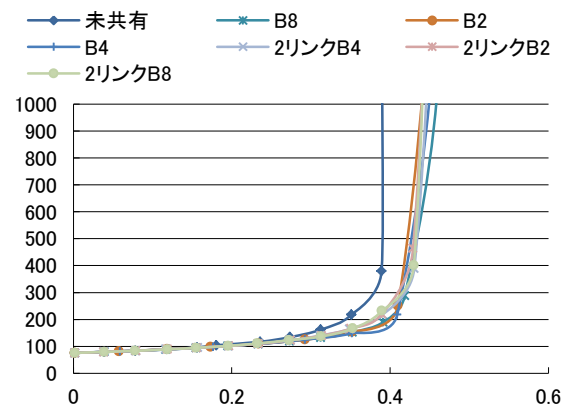


図 8. 通信性能のシミュレーション結果 (PE-No:16, Buffer:64, Packet:64)

はそれぞれ二リンク共有の各共有部分のブロック数が2、4、8の場合の結果である。すなわち、ルータ全体でブロック数が4、8、16となる。また、表2の各数値は、未共有と比較した場合の増加率で、単位はパーセントとなっている。結果より、二リンク共有ではシミュレーションに使用した範囲ではブロック数による性能の変化が小さいことが分かる。また、2リンクB2、4はB8と比べると若干ながら性能が低下するが、B2、4と比べると同等か若干高い性能を持つことが分かる。

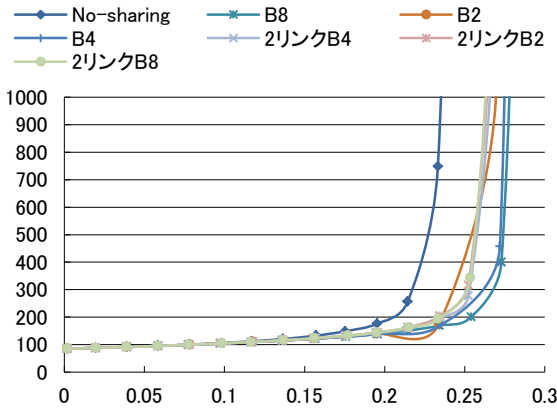


図9. 通信性能のシミュレーション結果(PE-No:64,Buffer:64,Packet:64)

表2. 各実装形態における通信性能の増加率

P	E	Buf	Pac	B2	B4	B8	2リンク	2リンク	2リンク
		fer	ket				リンク	リンク	リンク
							B2	B4	B8
1	6	32	16	3.5	10.4	11.5	9.9	9.8	10.2
			32	8.9	10.4	9.5	8.7	8.2	7.9
			64	6.2	5.4	8.7	8.9	8.6	7.5
	64	16	5.8	6.7	9.7	7.2	7.5	7.9	
		32	3.8	11.1	12.4	8.1	8.2	8.6	
		64	17	18.1	18.6	16.2	17	16.2	
6	4	32	16	7.8	13.4	17.9	15.1	15.2	15.0
			32	8.8	9	16.4	14.4	14.1	13.7
			64	8.4	8.5	9	8.5	8.6	8.7
	64	16	3.9	7.2	7.5	6	7.9	8.2	
		32	7.8	9.9	16.4	11.8	12.1	12.2	
		64	15.8	16.6	21.5	15.3	15.4	15.0	

6. まとめ

本稿において、我々は以前提案した提案手法について、ハードウェアコストの増加を軽減する方法として二リンク共有を紹介し、評価を行った。その結果、共有範囲の2リンク化には、ブロック数の合計とバッファ総量を全リンク共有の場合とあわせた場合に、若干の性能低下と引き換えに、ハードウェアコストを大きく削減できることを確認した。

以後は、今までシミュレーションで行ってきた通信性能の数学的な評価と提案手法の実装を行う。

参考文献

- [1] 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.
- [2] 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.
- [3] 深瀬尚久, 三浦康之, 渡辺重佳, 直接結合網のルータ回路におけるバッファのリンク間共有法の提案, 電気学会論文誌C(電子・情報・システム部門誌) Vol.132(2012) No.10, P1675-1688, 2012.10.
- [4] Naohisa Fukase, Yasuyuki Miura, Shigeyoshi Watanabe, The Hardware Cost Reduction Method of Control Circuit for Link-Sharing Method of Buffer in NoC Router, 2013 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing, pp.480-483, March 2013.
- [5] Naohisa Fukase, Yasuyuki Miura, Shigeyoshi Watanabe, The Proposal of Link-Sharing Method of Buffer in NoC Router: Implementation and Communication Performance, Journal of Basic and Applied Physics, (In printing).
- [6] 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.
- [7] 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.
- [8] T.C.So, S.Oyanagi, K.Yamazeki, Speculative Selection in Adaptive Routing on Interconnection Networks, 情報処理学会論文誌.コンピュータシステム, Vol.44, pp.147-156, 2003.
- [9] 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.
- [10] 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.
- [11] Michael Golden et al., "A 500MHz write-bypassed, 88-entry, 90bit register file," Proc. of Symposium on VLSI Technology, Session C11-1, 1999.
- [12] H.J.Mattausch, K.Kishi and T.Gyohten, "Area-efficient multiport 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.
- [13] 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.
- [14] 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.