計算機アーキテクチャ 136－5
（2000．1．7）

並列デッドロック回復ルータRecover-xの性能評価

林 匡哉 堀田真貴 中村さゆり
吉永 努 大津金光 馬場敬信

宇都宮大学 工学部 情報工学科

並列デッドロック回復ルータ Recover-x は、デッドロックサイクルからの退避候補メッセージを限定し、少ないデッドロック退避用のハードウェア資源で並列デッドロック回復を行なう。そのため、デッドロック回復時のオーバヘッドが小さい。また、メッセージのバーチャルチャネル間の切り替えを簡素化し、動作速度の低下を抑える。

本稿では、Recover-x と幾つかのルータについて評価する。その結果、Recover-x はネットワークサイズ、転送パターンに関わらず、高バンド幅、低レイテンシ通信を達成することを確認した。特に、大規模なネットワークやネットワーク負荷の偏りがある転送パターンにおいて、高い通信性能を達成することがわかった。

Evaluation of a Concurrent Deadlock Recovery Router Recover-x

MASAYA HAYASHI , MAKI HORITA , SAYURI NAKAMURA ,
TSUTOMU YOSHINAGA , KANEMITSU OOTSU and TAKANOBU BABA

Department of Information Science, Faculty of Engineering,
Utsunomiya University

The key idea of Recover-x, the concurrent deadlock recovery router, is that it limits escape message candidates in a deadlocked configuration. It enables the router to recover from deadlocks with small hardware resources for deadlock recovery. Recover-x also keeps a fast operating speed due to its simple switching mechanism among VCs. Therefore, it does not incur the deadlock overhead.

In this paper, we have evaluated the Recover-x router and several traditional routers. The simulations results show that Recover-x attains high-bandwidth and low-latency communication for many network sizes and message sizes. Recover-x shows especially high performance under large networks or non-uniform communication patterns.

1. はじめに

我々は、並列デッドロック回復ルーティングを行なうRecover-xを提案している[7]。Recover-xでは、サイクリックなチャネル依存関係が存在しても、メッセージがサイクルから退避可能な経路(退避パス)を用意すればデッドロックを防止できることを利用している[5]。このことを利用した適応ルーティングとして、*-channel[3]やDISHA[1]等が提案されている。

-channelは、全ネットワークポートに適応ルーティング用のバーチャルチャネル(VC)と退避パス用のVCを装備し、適応ルーティング用VCを使用しているメッセージがブロックされると、退避パス用VCへメッセージを退避する。ただし、退避パス用VCを使用しているメッセージであっても、適応VCが利用可能になれば、適応VCへ戻り適応ルーティングを再開できる。このように、-channelでは退避用ハードウェア資源を多く用意し、一度退避したメッセージであっても適応ルーティングを再開できるようにすることで、デッドロック発生時の性能低下を小さくしている。しかし、この方式では、デッドロックサイクルから退避するためのハードウェア資源が増加する。そのため、十分な数のVCを実装できない場合には、適応ルーティング用のハードウェア資源が不足する。また、メッセージが適応VCと退避パス用VC間を複雑に切り替えて使用するため、複雑なVC切り替え機構が必要になり、動作速度が低下するといった問題がある。

一方、DISHAでは、退避パス用のハードウェアであるデッドロックバッファ(DB)を最小限にすることで、適応ルーティング用の資源を確保している。し

かし、デッドロックが頻発するような状況ではDBが不足し、デッドロック回復のオーバヘッドが大きくなる。そこで、VC数の増加等により、デッドロック発生確率を低下させる必要がある。しかし、VC数の増加は、動作速度の低下を招くことになる[4]。

以上のことより、適応ルーティングの自由度を確保しつつ、デッドロック回復のオーバヘッドを小さく抑えられるように、デッドロック退避用のハードウェア資源の量を見極めることが重要である。また、複雑なVC切り替えを排除し、動作速度の低下を防ぐことも重要なことである。そこで、Recover-xは、デッドロックサイクルから取り除くメッセージを限定し、デッドロック退避用の資源を適当な量に抑える。また、本方式では、並列デッドロック回復により、デッドロック回復のオーバヘッドが小さい。さらに、退避したメッセージのVC切り替えに制限を加えることで、VCの切り替えを簡素化している。

これまで我々は，Recover-xの有効性を示すため、ハードウェア記述言語(HDL)によるLSI設計手法を活用した、並列計算機ルータのコストと性能に関する評価を行なってきた[9]。その結果、Recover-xは転送パターンに関わらず、高い通信性能を達成することがわかった。本稿では、様々なノイズ要因[1]の組合せにおけるRecover-xの通信性能について考察する[10]。ルータの通信性能評価にHDLを用いると、レジスタ転送レベル(RTL)シミュレーションによる緻密な評価を行なうことができるといった利点がある。しかし、実行時間が長いため、このことがネットワークサイズやメッセージ数を大きくする際の妨げとなる。そこで、汎用ネットワーク・シミュレータ生成系INSPIRE[8]を活用し、種々の実験を行なう。

2. Recover-x

2.1 基本アルゴリズム

Recover-xの基本的な考えは、デッドロックサイクルから退避するメッセージの候補(退避候補メッセージ)を、デッドロックサイクルから回復するために必要な集合に限定することにある。こうすることで、必要な退避用ハードウェア資源を適切な量に抑えることができる。これは、デッドロックサイクルが、サイクルを構成するメッセージ群から、一部のメッセージを取り除くことで解消されるからである。

図1に*-channelとRecover-xにおけるメッセージ処理の違いを示す。全メッセージをデッドロックサイクルからの退避候補とする*-channelでは、全ポートに退避パス用のVCを装備している。そのため、適応ルーティング用のVC数の割合が減少し、ルーティングの自由度が低くなる。また、DISHAでは、退避用ハードウェア資源を最小限に抑えるために、DBを複数のポートで共有している。そのため、デッドロック回復時にポート間の調停が必要になり、オーバヘッドが大きくなる。そこで、Recover-xでは、一部のネットワークポートにある退避パス用VCのみで、全ての退避候補メッセージを退避可能なように制限している。したがって、ルータ全体では、退避用のハードウェア資源の割合を小さく抑えることができる。そのため、多くのVCを装備できない場合でも、ルーティングの自由度が高い。また、デッドロック回復時のオーバヘッドを小さくするため、退避パス用のVCを各ネットワークポートで専用化している。そのため、デッドロック回復時のポート間調停が不要になる。なお、退避パス用VCでは、デッドロックを防止するために非適応ルーティングを行なう。そこで、適応ルーティングに使用するVCを適応VC、デッドロックサイクルからの退避パス用VCを非適応VCと呼ぶ。

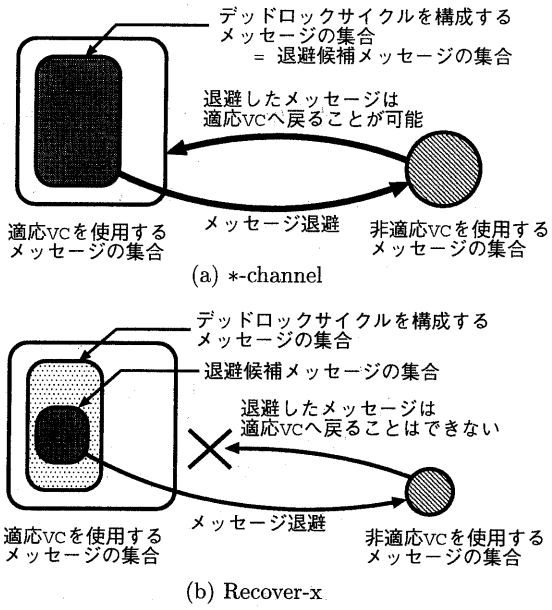


図 1: メッセージ処理の相違点

*-channelでは、非適応VCに退避されたメッセージでも、適応VCに戻り適応ルーティングを再開できる。しかし、我々の予備評価によると、VC切り替えの組み合わせ数は、ルーティングロジックの複雑さに影響し、ルータの動作速度を低下させる[7]。そこで、Recover-xでは非適応VCから適応VCへのメッセージ切り替えは行なわず、一度退避したメッセージは宛先ノードまで非適応VCを使用する。こうする

[1]回路設計者が値を決定できないパラメータ。転送パターンやネットワーク負荷、ネットワークサイズ等がこれにあたる。

ことで、非適応から適応へのVCの切り替えが不要になり、VC切り替えの組み合わせ数が減少する。ただし、デッドロック回復を効率的に行なうため、並列デッドロック回復はサポートする[2]。

2.2 2次元トーラス用アルゴリズム

2次元トーラス用Recover-xルータでは、退避候補メッセージを以下のように制限している。すなわち、Yアドレスのルーティングが完了するか、Yアドレスのルーティングが必要なく、X次元方向へのみ進む必要のあるメッセージである。こうすることで、Y次元ポートに非適応VCが不要になる。メッセージは、各ポートの適応VCを用いて最短経路で完全適応ルーティングを行なう。そして、退避候補メッセージが、一定時間以上ブロックされた場合にデッドロックと判断し、これを非適応VCへ退避する。一度非適応VCへ退避したメッセージは、宛先ノードまで非適応VCを使用する。

本方式では、X方向へ進むメッセージのみをデッドロック回復するので、Y方向には非適応VCが不要になり、ルーティング自由度を高くすることが可能である。さらに、一度退避したメッセージは、宛先ノードまで非適応VCを使用するため、VC切り替えの組み合わせ数が少なく動作速度が高い。また、非適応VCをネットワークポートごとに装備し、非適応VCをトーラスサイクルが発生しないように使い分けることで、並列デッドロック回復を可能にしている。そのため、デッドロック回復のオーバヘッドが比較的小さい。

3. ルータの仕様

3.1 ルータの基本構成とデータパス

図2にルータの基本構成を示す。全てのルータが、4つのネットワーク(N,E,W,S)ポートとPE I/Fを持つ。各ポートとも、入出力チャネルをそれぞれ持ち、それらのデータパスの幅は4バイトとする。本稿で考察するルータにおけるVCの最少構成はそれぞれ異なるが、各ネットワークポートごとに3本のVCを用意すれば全てのルータが構成可能である。したがって、本実験ではネットワークポートのVC数を3本に統一する。なお、PE I/Fが装備するVCは2本とする。したがって、1ノード当たりのVC数は、$3\times4+2=14$となる。なお、各VCのバッファ容量は、4バイト×8とする。

次に、図2にしたがってデータパスについて説明する。(1)ネットワークから入力されたメッセージはVC内のバッファに格納される。(2)そして、受信したメッセージのアドレスをデコードし、出力ポートに出力要求を行なう。(3)次に出力ポートのOutput Channel Arbiter(OCA)が各ポートからの出力要求を調停し、選択したVCへ出力許可を返す。(4)出力許可を受け取ったVCは、メッセージを出力する。

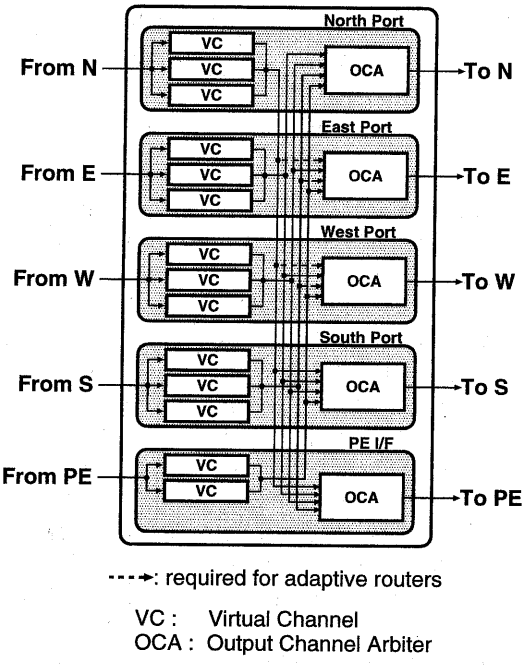


図 2: ルータの基本構成

3.2 動作速度

一般に、ルーティングの複雑さは、ルータの動作速度に影響を与える[4]。そのため、ルータの転送性能を評価する際に、ルータの動作速度を見積もることは重要なことである。本研究ではVerilog-HDLによってルータを記述し、論理合成を行なうことで動作速度の見積もりを行なった。論理合成は、全てのルータにおいて以下のような同一の条件で行なった。

シンセサイザ: Synopsys HDL Compiler Ver.1999.05

ライブラリ: LSI Logic 0.6μm Gate Array

回路の動作条件: 民生用最悪条件

マッピング最適化: Medium effort

配線負荷: セル面積による自動選択

上記条件の中で動作条件とは、ライブラリで指定されている製造プロセス、動作温度、供給電圧、および、配置配線の相互接続モデル等をあらわしている。今回、民生用最悪条件としてそれらの値が定義されているものを指定した。また、我々が実験を行なった範囲においては、デフォルトのMedium effortとより綿密に最適化を行なうBest effortでは、合成結果に顕著な差が見られなかった。しかし、Best effortでは合成時間が大幅に増加した。したがって、今回は合成

表 1: ルータの動作速度

Router	動作速度 (MHz)
Dimension-order	161.2
*-channel	120.4
Recover-x	142.8

時間を考慮し、デフォルトのMedium effortとした。なお、合成においては全てのサブ回路の境界最適化を行なった。

表1に各ルータの動作速度を示す。なお、ルータの動作速度とは、論理合成結果がタイミング条件を満たす範囲で最高の値とする。なお、全てのルータは、クロックの立上りエッジ駆動としている。

本実験では、従来のルータとして、非適応ルータにDimension-order[6]を、適応ルータに*-channel[3]ルータを採用する。Dimension-orderは、経路選択が最も簡単であり最速となる。*-channelは、適応/非適応VC間の切り替えが多く、動作速度が低下する。Recover-xは、退避候補メッセージを制限し、非適応VCから適応VCへの切り替えが不要なため、必要なVC切り替え数が少ない。したがって、*-channelよりも動作速度が高い。なお、これらの詳細については文献[9]で述べている。

4. 性能評価

4.1 シミュレーション条件

Dimension-order、*-channel、Recover-xルータに対しシミュレーションを行なった。シミュレーション条件を以下に示す。

シミュレータ: INSPIRE[8]により生成

ネットワークトポロジ: 2次元トーラス

ネットワークサイズ: 8×8, 16×16

チャネルのバンド幅: 4バイト/クロック

転送パターン: 以下の説明では、ネットワークサイズを$N \times N$とする。また、全ての転送パターンにおいて、全ノードがメッセージを送信する。

(1) Hot-spot: 各ノードが送信するメッセージの10%を、ノードアドレス$(\frac{N}{2}, j)$のN個のノードのうち、任意のノードに等確率で送信する。ただし、$0 \leq j \leq N-1$とする。残りのメッセージを、自ノード以外の任意のノードに等確率で送信する。

(2) All-to-all: ノードnは$n+1 \rightarrow n+2 \rightarrow \cdots \rightarrow N \times N - 1 \rightarrow 0 \rightarrow \cdots n-1$の順にメッセージを送信する。$n-1$まで送信したら、再び$n+1$から上記の順序で送信を繰り返す。

各転送パターンにおいて、ネットワークがある程度混雑した状態での評価を行なうため、予備実験を行なった。それを以下で述べる。

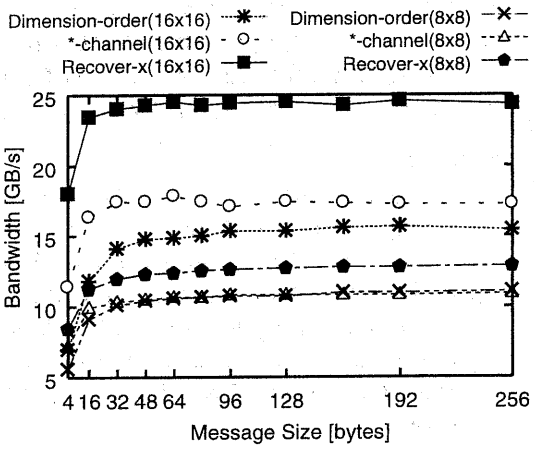
種々の転送パターン、ネットワークサイズ、メッセージ生成確率において、ネットワークが空の状態から到着順に、一定数ごとのメッセージ平均レイテンシを測定した。その結果、シミュレーション初期に到着したメッセージは、ネットワークがあまり混雑していないため、レイテンシが短いことがわかった。しかし、我々の行なった全ての実験において、3,000メッセージ以降に到着したメッセージについては、ほぼ一定のレイテンシを示した。また、All-to-allにおいては、初めは近傍のノードへメッセージを送信し、徐々に遠方のノードへメッセージを送信する。そして、遠方のノードへの転送後は、再び近傍のノードへの転送を行なう。したがって、遠方のノードへ送信するメッセージを評価対象に含めるためには、十分な数のメッセージを評価しなくてはならない。以上のことより、本実験では、到着順で3,000〜63,000メッセージを評価対象とする。

図3, 4に各転送パターンのバンド幅とレイテンシを示す。バンド幅は、単位時間当たりのネットワーク全体のメッセージの転送量とし、グラフの横軸はメッセージサイズである。その際、各PEは連続してメッセージを生成する。平均レイテンシのグラフでは、縦軸が平均レイテンシであり、横軸はバンド幅である。レイテンシは、メッセージサイズを128バイトとし、メッセージがPE内で生成されてから、宛先ノードに到着するまでの時間とする。なお、他のメッセージサイズにおけるレイテンシ評価も行なったが、一般に各々のサイズで高バンド幅なルータほど低レイテンシとなる傾向が見られた。なお、PEは100メッセージ(= 128バイト ×100)の容量のインジェクションバッファを持つ。平均レイテンシのグラフで、最もレイテンシが長い時のメッセージ生成確率は1であり、この時のバンド幅が、バンド幅のグラフにおいて128バイト時のバンド幅に等しい。

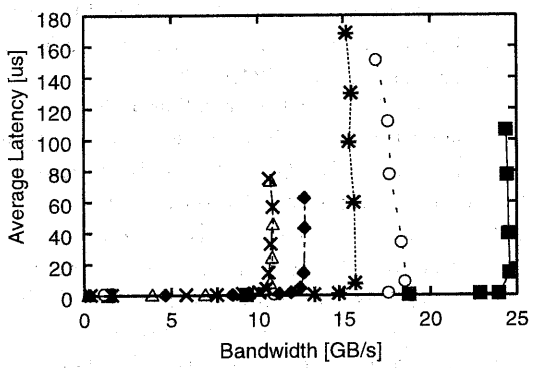
4.2 Hot-spot

一般にHot-spotのように局所的にネットワーク負荷の高い転送パターンでは、適応ルーティングは高バンド幅を示す。Recover-xは、*-channelよりも動作速度が高く、かつルーティングの自由度が高い。したがって、全てのネットワークサイズ、および、メッセージサイズにおいて、最高のバンド幅を達成する。ネットワークサイズが8×8の場合、*-channelとDimension-orderはほぼ同程度のバンド幅を示す。これは、*-channelの適応ルーティングによるバンド幅の向上が、低速な動作速度によって相殺されるためである。

同一ルータで16×16と8×8のネットワークにおけるバンド幅を比較すると、256バイトの場



(a) バンド幅



(b) 平均レイテンシ

図 3: Hot-spot

合、Dimension-orderで4.4GB/s (1.40倍) の向上がみられる。同様に*-channelで6.3GB/s (1.58倍)、Recover-xで11.5GB/s (1.89倍) の向上がみられる。このことより、ネットワークが大きくなると、適応ルーティングのバンド幅が大きく向上していることがわかる。これは以下のような理由による。ネットワークサイズが大きくなると、送信ノードと宛先ノードの平均ネットワーク距離が伸びる。すると、選択可能な経路が増え、メッセージの衝突をより効率的に回避できるためである。また、Recover-xは、*-channelよりも適応VCが多く、ルーティングの自由度が高い。したがって、Recover-xのバンド幅が最も大きく増加している。

図3(b)に、メッセージ平均レイテンシを示す。このグラフより、図3(a)のメッセージサイズ128バイト時のバンド幅が高い順にネットワークの飽和点が高くなり、レイテンシが短いことがわかる。例えば、ネットワークサイズが8×8の場合、Dimension-orderと*-channelの飽和点とレイテンシはほぼ等しいが、16×16では*-channelの飽和点が高く、レイテンシが短い。

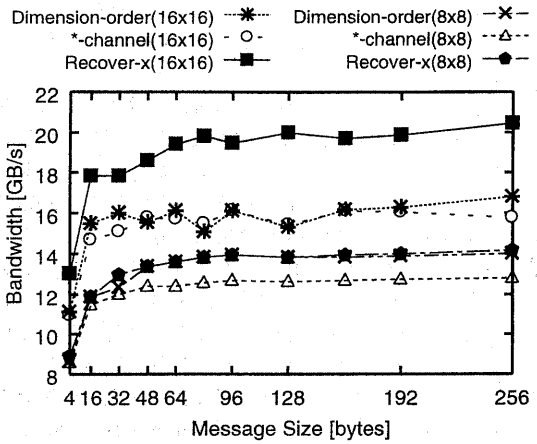
同一ルータで16×16と8×8におけるメッセージ生成確率1の場合(グラフ中の最もレイテンシが長い場合)の平均レイテンシを比較すると、Dimension-orderで93.7ns (2.26倍)、*-channelで77.5ns(2.06倍)、Recover-xで44.3ns (1.72倍)の増加となる。したがって、この転送パターンでは以下のようなことが言える。適応ルーティングは大規模ネットワークにおいて、効率的にメッセージの経路を決定することができる。そのため、低レイテンシかつ高バンド幅を達成する。しかし、*-channelは動作速度が低速なため、小規模なネットワークではDimension-orderと同程度のレイテンシとなる。一方、Recover-xは、ルーティングの自由度が高く動作速度が高いため、ネットワークサイズに関わらず、低レイテンシを達成する。

4.3 All-to-all

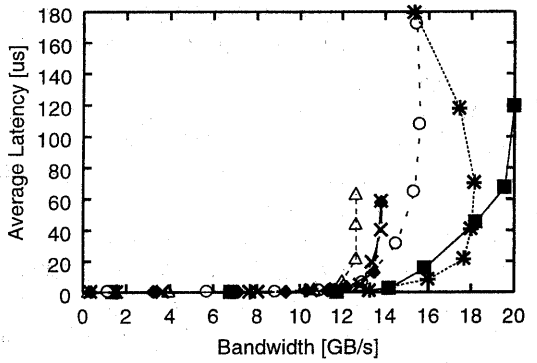
All-to-all通信は、ネットワーク全体にメッセージが均等に分散するユニフォーム通信である。そのため、メッセージの代替経路を探しても、空いているチャネルを発見することが困難である。したがって、Hot-spot等のノンユニフォームな転送パターンに比べ、適応ルーティングの効果が小さく、適応/非適応間のバンド幅の差は小さくなる。特に、8×8のネットワークにおいて、Dimension-orderが*-channelよりも高バンド幅を達成する。一方、Recover-xは*-channelよりも動作速度が高いため、ユニフォームな転送パターンであってもDimension-orderと同程度のバンド幅を達成する。これは、適応ルータの動作速度がDimension-orderのそれよりも低速なためである。また、16×16のネットワークでは、選択可能な経路が増加するため適応ルータのバンド幅が向上し、*-channelはDimension-orderと同程度のバンド幅となる。さらに、Recover-xは最高バンド幅を達成する。

図4(b)に、メッセージ平均レイテンシを示す。この転送パターンでもHot-spotと同様に、図4(a)のメッセージサイズ128バイト時のバンド幅が高いルータほど、ネットワークの飽和点が高く、レイテンシが短い。Dimension-orderは動作速度が高いため、16×16のネットワークの場合に、一部Recover-xよりも低レイテンシを達成する。しかし、Dimension-orderは、ネットワーク負荷が高くなると、レイテンシが急激に増加し、バンド幅が低下する。一方、Recover-xは、ネットワーク負荷が高くなってもバンド幅は低下せず、レイテンシの増加も比較的小さい。

以上のことより、適応の効果が得られにくいユニフォーム転送であっても、Recover-xはDimension-



(a) バンド幅



(b) 平均レイテンシ

図 4: All-to-all

orderと同程度、あるいはそれ以上の通信性能を示すことがわかる。特に、大規模ネットワークや高負荷なネットワークの場合には、Recover-xの効果が大きい。

5. まとめ

本稿では、我々の提案する並列デッドロック回復方式を実現するRecover-xルータの通信性能について考察した。その結果、Recover-xはネットワークサイズ、転送パターンに関わらず、高バンド幅、低レイテンシ通信を達成することを確認した。特に大規模ネットワークやネットワーク負荷に偏りのある転送パターンで、Recover-xは高い通信性能を達成する。

今後の課題として、プロセッサモデルを考慮した実アプリケーションでの評価が挙げられる。

謝辞 本研究を遂行するにあたり、INSPIREを快く提供して頂いた筑波大学の朴 泰祐助教授ならびに、ご指導頂いた板倉 憲一氏に深く感謝する。

本研究の一部は東京大学 大規模集積システム設計教育研究センターより提供して頂いたCADツールを使用している。本研究は、一部文部省科学研究費 基盤研究(B)課題番号10558039、奨励研究(A)課題番号11780190、および、実吉奨学会の援助による。

参考文献

[1] K. V. Anjan and T. M. Pinkston: "An Efficient, Fully Adaptive Deadlock Recovery Scheme: DISHA", *Proc. 22nd ISCA*, pp.201–210 (1995).

[2] K. V. Anjan, T.M. Pinkston and J. Duato: "Generalized Theory for Deadlock-Free Adaptive Wormhole Routing and its Application to *Disha* Concurrent", *Proc. IPPS*, pp.815–821 (1996).

[3] P.E. Berman, L. Gravano, G.D. Pifarré and J.L.C. Sanz: "Adaptive Deadlock and Livelock Free Routing with all Minimal Paths in Torus Networks", *Proc. SPAA* (1992).

[4] A. A. Chien: "A Cost and Speed Model for k-ary n-Cube Wormhole Routers", *IEEE Transactions on Parallel and Distributed Systems*, vol.9, no.2, pp.150–162(Frbruary. 1998)

[5] J. Duato: "A New Theory of Deadlock-Free Adaptive Routing in Wormhole Network", *IEEE Trans. on Parallel and Distributed Systems*, vol.4, no.12, pp.1320–1331(1993)

[6] W.J. Dally and C.L. Seiz: "Deadlock-Free Message Routing in Multiprocessor Interconnection Network", *IEEE Trans. Comut.*, vol.C-36, no.5, pp.547–553 (1987).

[7] 林匡哉, 堀田真貴, 吉永努, 大津金光, 馬場敬信: "適応ルータの効率的な並列デッドロックリカバリ方式の提案", 並列処理シンポジウム JSPP'99 論文集, pp.55-62(1999).

[8] 原田智紀, 曽根 猛, 朴泰祐, 中村宏, 中澤喜三郎: "並列処理ネットワークのための性能評価用シミュレータ生成系INSPIRE", 情報処理学会研究報告 Vol.95, No.80, pp.65-72(1995).

[9] 堀田真貴, 林匡哉, 中村さゆり, 吉永努, 大津金光, 馬場敬信: "RTL設計による並列計算機ルータの評価", 情報処理学会研報, vol.99, No.67, pp.67–72 (1999).

[10] P. López, R. Alcover, J. Duato and L. Zúnica: "Optimizing Network Throughput: Optimal versus Robust Design", *Seventh Euromicro Workshop on Parallel and Distributed Processing* (1999).