JAIST Repository
https://dspace.jaist.ac.jp/

Title	超並列計算機向き相互結合網SRTのデッドロックフリー・ルーティング
Author(s)	川井, 雅之; 井口, 寧; 堀口, 進; KAWAI, MASAYUKI; INOGUCHI, YASUSHI; HORIGUCHI, SUSUMU
Citation	情報処理学会論文誌, 40(5): 1977-1984
Issue Date	1999-05
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/3976
Rights	社団法人 情報処理学会, 川井雅之/井口寧/堀口進, 情報処理学会論文誌, 40(5), 1999, 1977-1984. ここに掲載した著作物の利用に関する注意: 本著作物の著作権は（社）情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。 The copyright of this material is retained by the Information Processing Society of Japan (IPSJ). This material is published on this web site with the agreement of the author (s) and the IPSJ. Please be complied with Copyright Law of Japan and the Code of Ethics of the IPSJ if any users wish to reproduce, make derivative work, distribute or make available to the public any part or whole thereof. All Rights Reserved, Copyright (C) Information Processing Society of Japan.
Description	

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

超並列計算機向き相互結合網SRTのデッドロックフリー・ルーティング

川井雅之[†,☆] 井口寧[††] 堀口進[†]

超並列システムに適合する結合網には，科学技術計算によく用いられる2次元格子結合を含み，ノードあたりのリンク数が少数であるなどの実装性，耐故障性などの要件が求められている．SRT（Shifted Recursive Torus）はグリッドの大きさが異なるトーラス結合を再帰的にシフトして構成される，超並列計算機に適した結合網である．SRTは，トーラス結合網に遠距離ノード間通信のためのバイパスリンクを付加しノードあたりのリンク数を固定した階層構造を有する結合網であり，従来の相互結合網に比べて遜色ない次数や直径を有している．本論文では，仮想チャネルを導入したSRTのデッドロックフリーなルーティング方式を提案する．さらに，シミュレーションにより性能評価を行い，従来のネットワークと比較検討した．その結果，デッドロックフリーなSRTルーティングアルゴリズムは，ルーティングの経路が制限されているものの2次元トーラス結合と比べ非常に高い転送能力があり，ハイパーキューブ結合と同程度の性能を有することが分かった．

Deadlock-free Routing on the Interconnection Network SRT for Massively Parallel Computers

MASAYUKI KAWAI,[†,☆] YASUSHI INOGUCHI[††] and SUSUMU HORIGUCHI[†]

Interconnection networks for massively parallel computers require good network features such as small number of links, easy extendibility and fault-tolerance. Shifted Recursive Torus (SRT) consists of torus and shifted torus. SRT has advantages that number of links for a node is fixed, and diameter is relatively short. However, there is no guarantee that the recursive routing is deadlock free. In this paper, we propose a deadlock free wormhole routing for SRT by introducing virtual channels. The performance of the routing algorithm are evaluated in detail by computer simulation. It's seen the deadlock free routing of SRT achieves much better dynamic communication performance than tours network and the almost same performance of hypercube.

1. はじめに

自然科学におけるシミュレーションやVLSI設計など，先端科学技術分野における大規模科学技術計算の需要は増大しており，多数のマイクロプロセッサ（MPU）を用いた並列処理システムによる高速化が求められている．並列処理システムでは，プロセッサ要素（PE）間の通信性能が並列処理の効率に大きな影響を与えるために，様々な視点から数多くの相互結合網が提案されている[1]．

科学技術計算の多くは2次元または3次元構造のデータを対象とするため，格子型の結合網と適合しやすい．しかし，格子結合網は，システムの規模が大きくなると通信性能が急速に低下するという問題がある．

そこで，超並列計算機用の相互結合網として，格子結合網やトーラス網を基本として，遠距離ノード間の結合を付加した多くの相互結合網が提案されている[2)~5)]．楊ら[2),3)]は，基本トーラス網に対して，グリッド間隔を $\sqrt{2}n$ (n は整数) 倍し 45° 傾け再帰的に上位のトーラスを構成する相互結合網RDTを提案した．Kirkmanら[4)]，Linら[5)]は，2次元の基本格子網に対して，グリッド間隔を 2^n (n は整数) 倍した格子を，互いに重ならないように再帰的に重ね合わせて構成する相互結合網PECについて議論している．

Inoguchiら[6),7)]は，トーラス結合網を基に，ノー

† 北陸先端科学技術大学院大学情報科学研究科
School of Information Science, Japan Advanced Institute of Science and Technology
☆ 現在，株式会社日本総合研究所
Presently with The Japan Research Institute, Limited
†† 北陸先端科学技術大学院大学情報科学センター
Center for Information Science, Japan Advanced Institute of Science and Technology

ド間距離の異なるバイパスリンクを再帰的に付加したShifted Recursive Torus（SRT）を提案している．SRTは少ない次数でRDTやPECと同程度の直径を実現している．しかしながら，SRTにおける再帰ルーティングはデッドロックフリーが実現されていないという問題点がある．

本論文では，仮想チャネルを用いたSRTのデッドロックフリーなルーティングを提案する．また，シミュレーションにより動的通信性能について性能評価を行い，従来のネットワークの性能と比較検討する．

本論文の構成は次のとおりである．2章では1次元SRTと2次元SRTの基本構成を述べ，SRTの直径を理論的に導出する．3章では再帰ルーティングアルゴリズムについて説明し，デッドロックフリーWormholeルーティングを提案し，デッドロックフリーであることを証明する．4章でシミュレーションにより動的通信特性の評価を行う．5章は結言である．

2. SRTの構成

本章では，本論文で必要となるSRTの定義等について説明する．

2.1 1次元SRTの構成

1次元SRT（1D-SRT）は，ノード数N $(N=2^n)$から成る環状網を基に構成される．環状網のあるノードを番号0とし，ノードを昇順に番号付け，次の基本トーラスを定義する．

定義1（基本トーラス） Nノードから成る基本トーラスのノードx $(0 \leq x < N)$は，隣接する左右のノード$(x \pm 1)\ mod\ N$と結合される． □

トーラスの両端は互いに結合されるので，ノード0の隣接ノードは1と$N-1$である．基本トーラスを構成する結合リンクをレベル0のリンクと呼ぶ．

トーラス結合網の直径および平均距離を短縮するために，基本トーラスにバイパスリンクを付加する．バイパスリンクにより構成される上位トーラスは次の定義に従う．

定義2（1D-SRT） $N=2^n$ ノードから成る基本トーラスから，ノード番号が

$$x\ mod\ 2^l = 2^{l-1}, \tag{1}$$

を満たすノードを取り出し，環状に接続する．この手続きをレベルlが1から$l_{max}-1$まで繰り返す．l_{max}は基本型1D-SRTの最大のレベルであり，$l_{max}=\log_2 N=n$である． □

式(1)を満たすノードをレベルlのノードと呼び，x_lと表記する．また，レベルlノードの集合をV_lとする．$l \geq 1$のレベルを上位レベルと呼ぶ．ノード0は，どのような$l \geq 1$でも式(1)を満たさないので，上位レベルを持たない．このため，ノード0のレベルは0とする．V_lを環状に接続するリンクを上位リンクと呼び，E_lとする．x_lはレベル0のリンクとは別に，レベルlのリンクによって2^l離れた同じレベル数のノード（$(x_l \pm 2^l)\ mod\ N$）に接続される．したがって，x_lは

$$(x_l \pm 1)\ mod\ N, \quad (x_l \pm 2^l)\ mod\ N,$$

の4ノードに接続される．

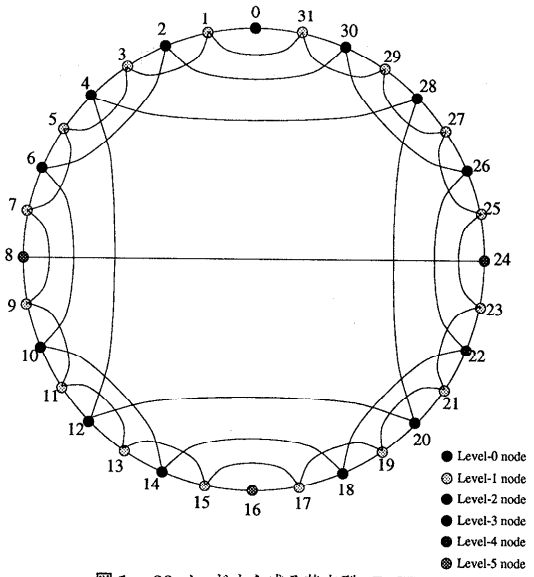


図1 32ノードから成る基本型1D-SRT
Fig. 1 Standard 1D-SRT consist of 32 nodes.

図1に32ノードから成る1D-SRTのリンク結合の様子を示す．ノードはレベルごとに2本のリンクを持つため，1D-SRTのどのノードも，基本トーラス（レベル0）を構成する2本と，上位リンクを構成する2本の合計4本のリンクを持つことが分かる．ただし，ノード$0, N/2$はどんな上位レベルも割り当てられないため，上位リンクを持つことはできない．また，ノード$\frac{1}{4}N$，$\frac{3}{4}N$に接続される上位リンクは1つであり，レベルは$l_{max}-1$である．

2.2 2次元SRTの構成

本節では，1D-SRTを2次元に拡張した2D-SRTについて述べる．$N \times N$ $(N=2^n)$のノードから成る2D-SRTは，x方向に直線状に配置したNノードから成る1D-SRTをy方向に積み重ねることにより構成される．2D-SRTは，x方向とともに，y方向にも1D-SRTが構成可能でなくてはならない．そこで，積み重ねる段ごとに，1D-SRTの原点ノード（ノード0）

の位置をシフトする．シフトの幅を変化させることにより，様々な 2D-SRT が構成可能である．ここでは，2D-SRT の一般形について説明する．

定義 3 $N \times N$ $(N = 2^n)$ のノードから成るトーラス上で，アドレスを左下から順に与え，(x, y) と表記する．1 次元 SRT と同様に，トーラス上のノードに上位レベルを割り当てる．レベル l のノード (x_l, y_l) は次の式を満たすノードである．

$$(x_l + s_x \cdot y_l) \bmod 2^l = 2^{l-1} \qquad (2)$$

ここで s_x は x 方向のシフト幅である．ノード (x_l, y_l) は，隣接する 4 ノード

$((x_l \pm 1) \bmod N, y_l)$, $(x_l, (y_l \pm 1) \bmod N)$

と，2^l 離れた 4 ノード

$((x_l \pm 2^l) \bmod N, y_l)$, $(x_l, (y_l \pm 2^l) \bmod N)$

とに接続される． □

x 方向に 1D-SRT を構成すると同時に，y 方向にも 1D-SRT が構成されるためには，シフト幅 s_x が以下の条件を満たす必要がある．

(1) 原点ノードが各行・列に 1 つだけ存在する

(2) x と y が転置関係にある式が定義できること．つまり，式 (2) に対して，$(y_l + s_y \cdot x_l) \bmod 2^l = 2^{l-1}$ が得られる．

原点ノードとは，1D-SRT のノード番号が 0 のノードのことである．また，**図 2** にシフト幅が 1 の 1-shift 型 2D-SRT を示す．

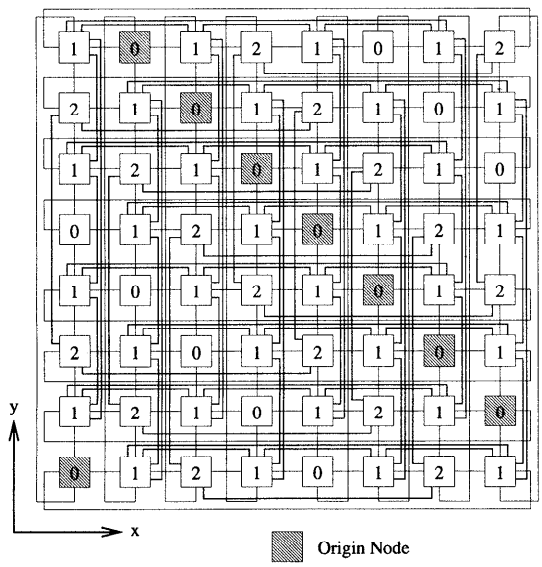


図 2 8×8 ノードから成る 1-shift 型 2D-SRT
Fig. 2 1-shift 2D-SRT consiste of 8×8 nodes.

2.3 1D-SRT の直径

本節では，再帰ルーティングの経路算出に必要である，レベル間距離について紹介する．レベル間距離とは，レベルが異なる最近隣ノード間の距離のことであり，下位ノードのレベルによって決定される．

また，上位の階層が下位の階層を含むことから，1D-SRT の直径はレベル間距離によって表され，再帰的にレベル間距離を求め，1D-SRT の直径を求めることができる．

上位レベルのノードに対する下位ノードの位置が，上位ノードの位置に依存しないことから，任意のノード $x_k(i) \in V_k$ から，$x_k(i)$ に最も近いノード $x_l(j) \in V_l$ までのノードレベルの並びを $s(k,l)$ $(k > l > 0)$ とすると，

$$s(k,l) = s(k+m,l) \qquad (m > 0)$$

と書くことができる．

任意のノード $x_k(i) \in V_k$ から，$x_k(i)$ に最も近いノード $x_l(j) \in V_l$ までの距離を $d(k,l)$ $(k > l > 0)$ とすると，

$$d(k,l) = d(k+m,l) \qquad (m > 0) \qquad (3)$$

となる．距離とは，レベル 0 リンクや上位リンクを用いた $x_k(i)$ から $x_l(j)$ までの経路のホップ数である．

$d(k,l)$ は，ルーティング時にノードのレベルを乗り換える（レベル k からレベル l のリンクへ乗り換える）ときの所要ホップ数に等しい．式 (3) よりすべての k, l $(k > l)$ について，$d(k,l)$ は $d(l_{max}, l)$ が求まれば十分である．そこで，$d(l_{max}, l)$ を $d_m(l)$ と表記し，これをレベル間の距離と定義する．

また，下位レベルのノード配置は上位レベルのノード配置に含まれるので，1D-SRT の直径は，上位のノードを持たないノード 0 とこれから最も離れたノード $N/2$ までの距離となる．また，ノード 0 はノード N と見なすことができ，ノード N はレベル $(l_{max}+1)$ ノードと見なせるので，次の定理が成り立つ．

定理 1 ノード数 N から成る 1D-SRT の直径 $D(N)$ は，

$$D(N) = d(l_{max}+1, l_{max}) = d_m(\log_2 N). \quad (4)$$

□

次に，レベル間の距離 $d_m(l)$ を具体的に計算する．レベル $(l-c)$ のリンクを通過するという条件のもとでの，ノード $N/2$ から（基本トーラス上で）最近隣のレベル l ノードまでの距離は，

$$d_m^c(l) = \left[2d_m(l-c) + (2^{c-1} - 1)\right] \qquad (5)$$
$$(1 \le c \le l)$$

と表すことができる．ルートは，ホップ数 $d_m(l-c)$ でノード $N/2$ から，ノード $N/2$ に最近隣のレベル

$(l-c)$ ノードに到達し，そこからレベル $(l-c)$ のリンクを $2^{c-1}-1$ ホップ伝わり，最後に $d_m(l-c)$ のホップ数でレベル l ノードに到達する．ノード間の距離 $d_m(l)$ は，c を変化させたときの $d_m^c(l)$ の最小値である．式 (5) を解き，$d_m^c(l)$ を最小とする c を求めると，レベル間距離は次のようにまとめられる．

補題 1（レベル間の距離） レベル間の距離 $d_m(l)$ は，

$$d_m(l) = 2^{c_f-1}(c_f - 1 + s_f) + 1. \quad (6)$$

ここで

$$c_f = \left\lfloor \frac{1}{2}\left\{\sqrt{8l+1}-1\right\} \right\rfloor \quad (7)$$

$$s_f = l - \frac{1}{2}c_f(c_f+1) \quad (0 \le s_f \le c_f). \quad (8)$$

このとき，経路はレベル $(l-c)$ のリンクを通過する．

□

なお，補題 1 では c を切り捨てて整数にしたが，同様に切り上げて整数化し，$d_m(l)$ を求めることもできる．

3. ルーティングアルゴリズム

3.1 再帰ルーティング

1D-SRT のルーティングアルゴリズムを図 **3** に示す．ルーティングの始点ノードを x_s，終点ノードを x_d とする．関数 *FindMLevel* によって，ルーティングに使用するリンクの最大レベル l_r を求め，関数 *FindNearestNodes* で，このレベルのノード x_s^r，x_d^r を探す．始点ノード x_s と x_s^r，x_d^r と終点ノード x_d との間で，手続きを再帰的に呼び出し，経路を求める．この方針に基づくルーティングを再帰ルーティングと呼ぶ．

```
RecursiveRouting(x_s, x_d){
  if( x_s = x_d )        return( φ );
  if( x_s ≤ x_d )        dir = +1;
  else                   dir = −1;
  l_r = FindMedLevel(x_s, x_d);
  (x_s^r, x_d^r) = FindNearestNodes(l_r, x_s, x_d);
  l_u = FindUprLevel(x_s, x_d);
  (x_s^u, x_d^u) = FindNearestNodes(l_u, x_s, x_d);
  if( |x_s − x_s^u| + |x_d − x_d^u| < |x_s − x_s^r| + |x_d − x_d^r| ){
    l_r = l_u;
    x_s^r = x_s^u;      x_d^r = x_d^u;
  }
  RoutingList = RecursiveRouting(x_s, x_s^r);
  while( x_s^r ≠ x_d^r ){
    addlist(RoutingList, x_s^r);
    x_s^r = x_s^r + dir * 2^{l_r};
  }
  addlist(RoutingList, RecursiveRouting(x_d^r, x_d));
  return( RoutingList );
}
```

図 3 1 次元 SRT のルーティングアルゴリズム
Fig. 3 The routing algorithm for 1D-SRT.

32 ノードから成る 1D-SRT における，ノード 0 $(x_s = 0)$ からノード 15 $(x_d = 15)$ までのルーティング例を図 **4** に示す．以下，図 4 を例にして，ルーティングについて詳細に述べる．

手順 1．最大レベルのリンクの決定 ルーティングで使用するリンクのうち，最大のレベル l_r を決定する．*FindMLevel* で，補題 1 に基づき，レベル l_r を計算する．つまり，$l = \log_2|x_d - x_s|$ から，式 (7) により c_f を計算し，$l_r = l - c_f$ を求める．図 4 のノード 0 から 15 への経路では，$\log_2|15-0|$ を切り上げ，$l = 5$ を得る．式 (7) より，$c_f = 2$ を計算して，$l_r = l - c_f = 3$ を得る．

手順 2．バイパスリンクの最近隣端点の探索 *FindNearestNodes* で，レベル l_r のノードのうち，x_s，x_d に最も近いノードを探索する．ノード番号が x より大きい，x に最近隣のレベル l のノード番号は次式で与えられる．

$$nlnear\_f(x,l) = \left\lceil \frac{x-2^{l-1}}{2^l} \right\rceil \cdot 2^l + 2^{l-1} \quad (9)$$

同様に，ノード番号が x より小さい，x に最近隣のレベル l のノード番号は次式で求まる．

$$nlnear\_b(x,l) = \left\lfloor \frac{x-2^{l-1}}{2^l} \right\rfloor \cdot 2^l + 2^{l-1} \quad (10)$$

最近隣ノードの探索時，x_s に最近隣のレベル l_r のノードは，x_d 側だけに限られない．そこで，通信相手と反対側の逆方向のレベル l_r のノードも探索し，より近いノードを決定する．その判断は次式で与えられる．

$$|nlnear\_f(x,l) - x| < |nlnear\_b(x,l) - x| \quad (11)$$

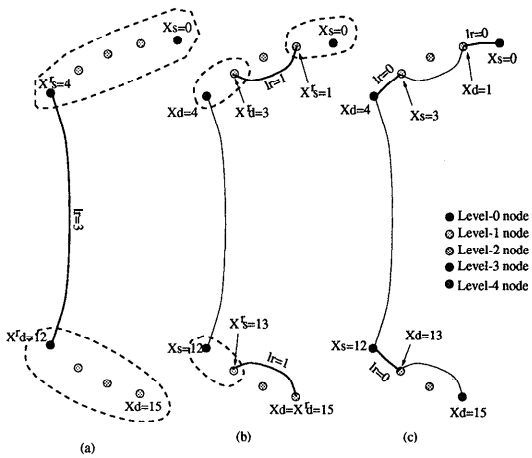


図 4 1 次元 SRT のルーティングの例
Fig. 4 Example of 1D-SRT routing.

式 (11) が真のときは，$nlnear\_f(x,l)$ を採用し，偽ならば $nlnear\_b(x,l)$ を採用する．決定された x_s 側の最近隣ノードを x_s^r，x_d 側の最近隣ノードを x_d^r とする．図 4 (a) で，$l_r=3$ なので，$x_s^r=4$, $x_d^r=12$ となる．

手順 3. l_r の補正

(手順 3-1) 手順 1 によるレベルの決定は，ノードの位置を考慮していないため，上位レベルのリンクを使用した経路の方が明らかに短い場合でも，上位レベルのリンクを使用しないという問題点がある．たとえば，図 1 でノード 8 から 24 へのルーティングの場合，式 (7) を用いたレベル決定では $l_r=3$ となるが，実際には $l_r=4$ のリンクを使用した方が良い．そこで，*FindUprLevel* で x_s から x_d 間のリンク中の最大のレベル l_u を取り出す．l_u は，

$$l_u=\log_2(|x_d-x_s|)$$

と計算できる．例では $l_u=3$ である．

(手順 3-2) 手順 2 と同様に，レベル l_u のノードのうち，x_s，x_d に最も近いノードを探索する．例では，$l_u=3$ なので，$x_s^u=4$, $x_d^u=12$ となる．

(手順 3-3) l_r をそのまま使用するか，l_u を使用した経路にするかを決定する．x_s^r，x_d^r よりも x_s^u，x_d^u の方が x_s，x_d に近い場合は，レベル l_r の代わりに l_u を使用する．例では，$l_r=l_u$ なので，$l_r=3$ を使用する．

手順 4. 低位リンクに対する再帰ルーティング

x_s と x_s^r 間で，$x_s=x_s^r$ となるまで再帰的に *RecursiveRouting* を呼び出す．

3.2 デッドロックフリールーティング

3.2.1 1D-SRT

1D-SRT における再帰ルーティングでは，上位レベルノードを算出する際，より x_s に近いノードを探索するため，逆方向へルーティングする可能性があり，チャネルに循環を生じさせてしまう原因となる．また，再帰ルーティングを行うために必要な仮想チャネル数が明確にされていないため，ラウンドトリップループを用いると，チャネルに循環が生じてしまい，デッドロックを引き起こす可能性がある．

本論文では再帰ルーティングに次の制約条件を付加することでデッドロックを回避する．

(C1) ノード探索は 1 方向にのみ行う

(C2) 物理リンク 1 本につき 2 つの仮想チャネルを設ける

ここで，両条件はともにチャネルの循環を防ぐための条件であるが，特に，(C1) は逆方向へのルーティングを回避する条件であり，(C2) はラウンドトリップループを用いた場合のチャネルの循環を直接的に回避するための条件である．したがって，ノードあたり 8 つの仮想チャネルが必要となる．また，ノードの 1 方向のみへの探索は，再帰ルーティングアルゴリズムおいて，式 (11) による条件を取り除くことで得ることができる．つまり，$x_s<x_d$ の場合，*FindNearestNodes* で得られるレベル l ノードは $x_s^r=nlnear_f(x_s,l)$, $x_d^r=nlnear_b(x_d,l)$ となる．

また，仮想チャネルの使用法は以下のルール従う．

(1) 仮想チャネルに HIGH，LOW，2 つのクラスを割り当てる．

(2) 最初は LOW の仮想チャネルを使用しルーティングを開始する．

(3) ラウンドトリップループを用いる際，仮想チャネルを HIGH に切り替える

定理 2 1D-SRT における再帰ルーティングに (C1), (C2) の条件を付加したアルゴリズムはデッドロックフリーである．

証明 拡張した再帰ルーティングアルゴリズムよりチャネルに循環が生じないことを示す．

各ノードのチャネルに対して，次のようにチャネル番号 $vi\lambda$ を割り当てる．ここで，$vi\lambda$ は次のような値である．

- v：仮想チャネル番号（1：HIGH，0：LOW）
- i：
  - チャネルの方向が反時計回りのとき：ノード番号 i
  - チャネルの方向が時計回りとき：ノード番号のサイズに対する補数 $(N-1-i)$
- λ：
  - 0：方向が反時計回りのレベル 0 のチャネル
  - 1：方向が時計回りのレベル 0 のチャネル
  - 2：方向が反時計回りの上位レベルチャネル
  - 3：方向が時計回りの上位レベルチャネル

制約条件 (C1) より，メッセージは決められた 1 方向にのみルーティングが行われるため，チャネル番号は単調に増加する．また，制約条件 (C2) により仮想チャネルが付加されたため，ラウンドトリップループを通過する場合は，仮想チャネルを HIGH に切り替えることができ，この場合もチャネル番号は単調に増加する．つまり，拡張した再帰ルーティングアルゴリズムよりチャネル番号は単調増加する．したがって，チャネルには循環が生じることはない． □

3.2.2 2D-SRT

次に 2D-SRT におけるデッドロックフリー再帰ルー

ティングを考える．従来の2D-SRTにおける再帰ルーティングは，1D-SRTにおける再帰ルーティングと同様にルーティングに使用するリンクの最大レベルを求め，そのレベルのノードと出発・目的ノードとの間で再帰的に手続きを繰り返すことで経路を求める．

しかしながら，2D-SRTのルーティングに使用するリンクの最大レベルが出発ノードと同じ方向（行あるいは列）に存在するとは限らない．したがって，目的ノードに達するまでにメッセージの進む方向が何度も変わってしまい，デッドロックフリーを保証することはできない．これを克服する方法としては，仮想チャネルを十分に設けることで方向転換が起こるごとにチャネルを切り替えるといった方法[9]も提案されている．しかしながら，何回の方向転換でルーティングが終了するかは分からないため，膨大な仮想チャネルが必要となり現実的ではない．

そこで2D-SRTにおいては，まず，1D-SRTにおける再帰ルーティングを x 次元に，目的ノードの x 次元のアドレスと一致するまで施し，その後 y 次元に対し1D-SRTにおける再帰ルーティングを施す．つまり，1D-SRTにおける再帰ルーティングを次元オーダで使用することでデッドロックフリーを保証する．したがって，付加する条件は

(C3) ルーティングは次元オーダで行われる

と記述でき，基本的な1D-SRT再帰ルーティングと条件(C1)，(C2)，(C3)を用いることで，2D-SRTに対しデッドロックフリーなルーティングアルゴリズムが実現できる．

定理3 2D-SRTにおいて1D-SRTにおける再帰ルーティングに(C1)，(C2)，(C3)の条件を付加したアルゴリズムはデッドロックフリーである．

証明 ルーティングは，(I) ルーティングが同一方向で閉じている場合，(II) x 次元のある方向から y 次元のある方向へルーティングが必要な場合の2つに大きく分けられる．

(I) の場合： メッセージは1D-SRT上をルーティングしているのと等価である．したがって，定理2よりデッドロックは生じない．

(II) の場合： (i) y 次元の転送先に他のメッセージが存在しない場合と，(ii) y 次元の転送先に他のメッセージが存在する場合の2つのケースが考えられる．

(i) x 次元に存在するメッセージは，ブロックされることなく y 次元へ転送することができる．したがって，デッドロックフリーである．

(ii) y 次元にすでに存在しているメッセージは，(C3) より x 次元へ転送されることはない．つまり y 次元のみで転送が行われている．したがって，定理2より，すでに存在するメッセージはデッドロックを引き起こなさい．よって，x 次元にあるメッセージは，ブロックされることはあっても，デッドロックを引き起こすことはない．

よって (i)，(ii) より x 次元から y 次元へと転送される場合もデッドロックは生じない．

以上より，1D-SRTにおける再帰ルーティングに条件(C1)，(C2)，(C3)を付加したアルゴリズムはデッドロックフリーである． □

4. 性能評価

4.1 1D-SRTの直径・平均距離

表1 に1D-SRTの直径および平均距離を示す．表中のOPTは最適ルーティング，RECは再帰ルーティング，DFRはデッドロックフリー再帰ルーティングである．最適ルーティングは可能な経路を全探索し，最短経路を使用するルーティング方式である．これに対し，1D-SRTの再帰ルーティングでは10%前後平均距離が長くなる．この平均距離の増加は，ノード数が増えると大きくなる．ノード数が増えると，距離を短縮する上位レベル l_u と補題1で計算されるレベル l_r の差が大きくなる．レベル l のノード数は，$N/2^l$ なので，距離の短縮に有効な l_u のノードの数が l_r のノードの数に比べて少なくなる．このため，レベル l_u のリンクを使用しにくくなり，距離の短縮が有効に行われない．再帰ルーティングによる直径は，1D-SRTの理論直径である $D(N) = d_m(\log_2 N)$ に一致する．

4.2 2D-SRTの直径・平均距離

2D-SRTの平均距離は，1D-SRTと同様の理由で，最適ルーティングに比べ，再帰ルーティングの平均距離が長くなる．2次元では，“1次元方向の平均距離の増加”の2乗となるため，最適ルーティングとの

表1 SRTの直径（平均距離）
Table 1 Diameter (degree) of SRT.

Number of node	2^8	2^{10}	2^{12}
1D-SRT(OPT)	17 (7.0)	25 (11.5)	41 (17.7)
1D-SRT(REC)	17 (7.4)	25 (12.4)	41 (20.0)
1D-SRT(DFR)	20 (9.6)	32 (16.9)	54 (27.9)
2D-SRT(OPT)	6 (3.6)	8 (4.8)	11 (6.3)
2D-SRT(REC)	6 (4.2)	8 (5.7)	11 (7.8)
2D-SRT(DFR)	10 (4.3)	14 (6.4)	20 (9.5)
2D Torus	16 (4.0)	32 (8.0)	64 (16.0)
RDT(2,4,1)/α	–	7 (5.6)	8 (6.7)
HyperCube	8 (4)	10 (5)	12 (6)

差が 1D-SRT よりも大きく，10～30%程度増加する．RDT(2,4,1)/α の平均距離と比べると，ほとんど劣らない性能を持つことが分かる．

2D-SRT の直径と次数を他の結合網と比較すると，2D-SRT は少ない次数でハイパーキューブに近い性能を持ち，他の超並列計算機向きの結合網と比べ，遜色ない性能を持つことが分かる．2D-SRT は，トーラスだけを階層的に用いて構成されるので配線が容易であり，故障回避アーキテクチャなどのインプリメントが可能[8]という特徴を有している．

4.3 デッドロックフリー再帰ルーティング（DFR）を用いた場合の直径・平均距離

デッドロックフリー再帰ルーティングでは，デッドロックを回避するために各リンクに 2 つ仮想チャネルが必要となる．この条件は，バイパスを持たない通常のトーラス網と同じ条件であるため，実装上さほど問題にならない．したがって，デッドロックフリーな再帰ルーティングを使用した場合に問題となる点は，上位レベルノードを探索する際，探索方向を 1 方向に制限したことによるホップ数の増加にある．

表 1 より，再帰ルーティングに比べ直径，平均距離ともに増加していることが分かる．これは，デッドロックフリー再帰ルーティングでは，上位レベルノードの探索を行うとき，探索方向が制限されるためである．この静的性能の悪化は，Dally ら[9]の方法で，仮想チャネルを付加し逆方向へのルーティングを制限付きで許容してやることで改善されると考えられるが，それに関する考察は今後の検討課題とする．

4.4 動的通信性能評価のシミュレータ概要

デッドロックフリーな再帰ルーティングの性能を評価するために，C++によりシミュレータを作成し，性能評価を行った．シミュレータはフリットレベルでのシミュレートが可能であり，トポロジ，ルーティングアルゴリズム，パケットサイズ，メッセージの発火確率などが変更可能である．

4.5 デッドロックフリー再帰ルーティングの動的性能評価

ここでは，一様なランダム転送を対象とした，デッドロックフリー再帰ルーティングの動的通信性能評価を行う．1D-SRT，2D-SRT，2 次元トーラス，ハイパーキューブの動的通信性能の比較を行う．

シミュレーションは，ネットワークサイズを 256 PE，パケット長を 16 フリット，仮想チャネル数を 2，フロー制御方式を Wormhole とした．

なお，ハイパーキューブは仮想チャネル数が 1 でデッドロックフリーが実現できるため，メッセージがネットワークに投入される際，空いている方を使用する（ただし，いったんそのチャネルを使用したらその後はチャネルを変更することはできない）．

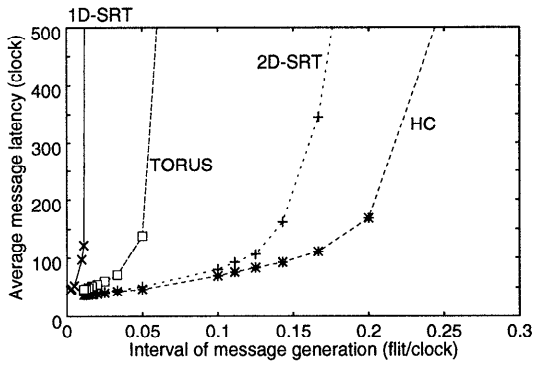


図 5 ランダム転送時の平均通信時間

Fig. 5 Average message latency as a function of interval of message generation.

シミュレーションはある発生確率でランダムな宛先にメッセージを発生させ，その平均通信時間を測定した．発生確率はフリットが毎クロック投入される確率を 1 とし，ほとんど投入されなくなる確率（$=10^{-4}$）まで，36 ポイントで測定を行い，各ポイントでの 10000 メッセージの平均通信時間を求める．

図 5 にメッセージ発生確率と平均通信時間の関係を示す．図 5 から，1D-SRT の遅延が非常に大きいことが分かる．一方，2D-SRT はルーティングする方向に制限を加えたにもかかわらず，2D-TORUS に比べ大幅に通信遅延が減少している．また，ハイパーキューブに比べても，若干性能は低下するが，ほとんど同じ性能が得られていることが分かる．

したがって，本論文で提案したデッドロックフリー再帰ルーティングは，従来の再帰ルーティングに比べ静的性能は少し悪くなるが，十分高い動的性能が得られることが分かる．

5. ま と め

本論文では，SRT におけるデッドロックフリーな Wormhole ルーティング提案を行った．SRT におけるデッドロックフリーな Wormhole のルーティングは従来の再帰ルーティングに簡単な制約条件を付加されるだけで得られる．また，シミュレーションによる動的通信性能評価を行った．その結果，1D-SRT では通信性能の低下が見られたが，2D-SRT では，十分高い通信性能が得られることが分かった．

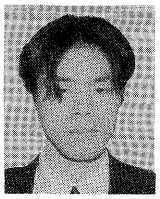
今後の課題として，適応型ルーティングの提案を行う予定である．

謝辞 本論文の一部は，文部省科学研究助成金を用いて行われた．関係各位に感謝する．

参考文献

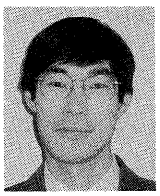
1) Duncan, R.: A Survey of Parallel Computer Architechturcs, *IEEE Computer*, Vol.23, No.2, pp.5–16 (1990).
2) 楊 愚魯，天野英晴，柴村英智，末吉敏則：超並列計算機に向き結合網：RDT，信学論（D-I），Vol.J78-D-I, No.2, pp.118–128 (1995).
3) 天野英晴，楊 愚魯：超並列計算機に向き結合網RDT上でのデッドロックフリールーティング，信学技報，CPSY94-56，pp.109–116 (Sep. 1994).
4) Kirkman, W.W. and Quammen, D.: Packed Expornential Connections – A Hierarchy of 2D-Meshes, *Proc. 5th International Parallel Processing Symposium*, pp.464–470, (Apr. 1991).
5) Lin, C.-C. and Prasanna, V.K.: Bounds on the diameter of one-dimenstional PEC networks, *Journal of Parallel and Distributed Computing*, Vol.29, No.1, pp.1–16 (1995).
6) Inoguchi, Y. and Horiguchi, S.: Shifted Recursive Torus Network for Mesh-Oriented Interconnections, *Proc. 31st Conference on Information Sciences and Systems*, Baltimore, pp.351–356 (Mar. 1997).
7) Inoguchi, Y. and Horiguchi, S.: Shifted Recursive Torus Interconnection for High Performance Computing, *IEEE High Performance Computing in Asia Conference*, Seoul, pp.61–66 (Apr. 1997).
8) 井口 寧，堀口 進：超並列計算機用プロセッサ結合網SRT – ネットワーク特性と故障回避アーキテクチャ，信学技報，CPSY96-45，pp.37–43 (May 1996).
9) Dally, W.J. and Aoki, H.: Deadlock-Free Adaptive Routing in Multicomputer Networks Using Virtual Channels, *IEEE Trans. Parallel Distrib. Syst.*, Vol.4, No.4, pp.466–475 (Apr. 1993).

(平成10年8月31日受付)
(平成11年3月 5 日採録)



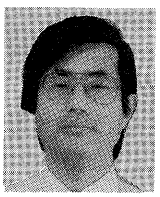
川井 雅之

昭和49年生．平成9年法政大学工学部システム制御工学科卒業．平成11年北陸先端科学技術大学院大学情報科学研究科博士前期課程修了．



井口 寧（正会員）

昭和42年生．平成3年東北大学工学部機械工学科卒業．平成9年北陸先端科学技術大学院大学情報科学研究科博士後期課程修了．現在，同大学院情報科学センター助手．また，平成6～9年日本学術振興会特別研究員として研究に従事．この間並列システムに関する研究を行う．電気情報通信学会，IEEE各会員．



堀口 進（正会員）

昭和51年東北大学工学部通信工学科卒業．昭和56年同大大学院博士課程修了．昭和57年同情報工学科助手．昭和64年同助教授．現在，北陸先端科学技術大学院大学情報科学研究科教授．この間，並列処理，超並列システム，ウェーハ規模集積システム，並列アルゴリズム，マルチメディア統合システムに関する研究を行う．昭和61年6月～62年7月米国IBMワトソン研究所・客員研究員として並列計算アルゴリズムの研究に従事．平成6年7～8月ルイジアナ州立大学（ラファエット）客員教授，平成9年7月～8月テキサスA&M大学客員教授．昭和64年～平成7年IEEE学会WSI国際会議組織委員会委員．平成4～7年同国際会議アジア地域議長．IEEEシニア会員，電子情報通信学会会員．