2ウェイマージ機能を有するオメガネットワーク†

喜連川 優†† 小川泰嗣†††

近年データベース処理を高速実行するデータベースマシンの研究開発が進んでいるが，その多くは処理モジュールが共有メモリを持たない Shared nothing 型アーキテクチャをとる．このような並列マシン上では，ソートされるデータは複数の処理モジュールに跨って存在しているため，処理モジュールが協調してソートを行う必要がある．処理モジュール間ソートは並列2ウェイマージソートによって効率的に実行されるが，相互結合網にデータ列のマージ機能を埋め込むことができればさらに効率的処理が実現される．本論文では，相互結合網としてよく用いられるオメガネットワークにマージ機能を実装する方式を検討した．マージ木の節点をオメガネットワークのスイッチングユニットに対応させること（この対応の決定をマッピングと呼ぶ）ができれば，スイッチングユニットに2つのデータの比較を行うためのハードウェアを付加することにより，オメガメットワーク上でマージを実行できる．本論文では，データが2つの処理モジュール上にある場合についてマージ木をオメガネットワークにマッピング可能なことを証明し，そのアルゴリズムを示す．さらに，データが3つ以上の処理モジュール上にある場合についてマッピング可能なことを証明し，マッピングアルゴリズムを示す．

1. はじめに

ソート処理はさまざまなデータ処理の中でも，最も基本的でかつ重要な処理の1つである．ソート処理は負荷が大きいためその高速化の研究が続けられてきており，並列ソートのアルゴリズムの研究[1),2)]，ハードウェアソータ[3)~8),20)]の研究等が多数行われている．

ソート処理はデータベース処理においても重要な処理である．近年，データベース処理を高速実行するデータベースマシンの研究・開発が進んでおり[9)]，その多くがシステム拡張性の容易ないわゆる Shared nothing 型，すなわち，処理モジュールが共有メモリを持たずに相互結合網により結合したアーキテクチャを採用している[10)~13)]．このような疎結合並列マシン上でソートを行う場合，ソートされるデータはディスクからの読み出し・選択演算・結合演算等を経て用意されるため，複数の処理モジュールに分散して存在する．したがって，処理モジュールが協調してソートを行う必要がある[14),15)]．本論文では，このようなソートを処理モジュール間ソートと呼ぶ．処理モジュール間ソートは並列2ウェイマージソートにより簡単に実現できる．並列2ウェイマージソート[1),2)]では，まず各処理モジュールが自モジュール内のデータをソートし，つぎにそれらデータ列を2つずつマージすることで1つのソートされたデータ列を生成する．データ列のマージの際には，複数のデータ列が入力される場所を葉，出力される場所を根として二分木を構成し，マージソートを並列に実行する．疎結合並列マシンの相互結合網にデータ列のマージ機能を付加することで，ソートにおける処理モジュールの負荷を小さくできる．テラデータ社のデータベースマシン DBC/1012[13)] は Ynet と呼ばれる形状が二分木でマージ機能を埋め込んだ相互結合網を採用し，処理モジュールに負荷をかけることなく高速処理モジュール間ソートを実現している[14)]．

本論文では，相互結在網としてよく用いられるオメガネットワークに2ウェイマージ機能を実装する方式を検討する．オメガネットワーク[16)]は2入力2出力のクロスバスイッチを構成要素とし，複数のスイッチング装置を組み合わせた多段ネットワークの一種であり，結合能力等の比較において木ネットワークより優れている．本論文では，ソート支援を相互結合網に求める要望に答えるべく，木ネットワーク上に実現されるマージ機能をオメガネットワークに組み込む方式を検討する．我々はバケット平坦化機能を有するオメガネットワークというものを提案しており[17)]，オメガネットワークの機能拡張によりデータベース処理の高速化を目指している．本論文で提案するオメガネットワークの目的は，このような機能拡張の1つである．

オメガネットワーク上で並列2ウェイマージを実現するためには，スイッチング装置に2つのデータの比較を行うためのハードウェアを付加し，マージ木の節

† An Omega Network Embedding the 2-way Merge Function by MASARU KITSUREGAWA (Institute of Industrial Science, University of Tokyo) and YASUSHI OGAWA (Research and Development Center, RICOH Co., Ltd.).

†† 東京大学生産技術研究所
††† (株)リコー中央研究所

点とオメガネットワークのスイッチング装置の対応を決定できればよい．本論文では，節点とスイッチング装置の対応を決定することをマージ木のオメガネットワークへのマッピングと定義する．まず，データが2つの処理モジュール上にある場合についてマージ木がオメガネットワークにマッピングできることを証明し，マッピングアルゴリズムを示す．さらに，一般的な場合としてデータが3つ以上の処理モジュール上にある場合についてマッピングできることを証明し，マッピング決定の効率的アルゴリズムを示す．

以下，2章で疎結合並列マシン上でのソート法である処理モジュール間ソートについて説明する．3章で代表的相互結合網であるオメガネットワークについて簡単に説明する．ここでは，続く4章で必要となるオメガネットワークにおけるパスの設定法を簡単にまとめる．4章では，オメガネットワークにマージ木を埋め込むアルゴリズムを示す．まず，2つのデータ列をマージする場合のアルゴリズムを検討し，つぎにそのアルゴリズムを複数（3つ以上）のデータ列をマージする場合に拡張する．5章では，スイッチング装置の構成とソート時間について簡単に考察する．最後の6章で全体をまとめる．

2. 並列2ウェイマージソートによる処理モジュール間ソート

本論文では，処理モジュールが共有メモリを持たず相互結合網により結合した疎結合並列マシンを想定する．処理モジュール間ソートとは，疎結合並列マシンの処理モジュール上に分散しているデータを1つのソートされたデータ列とする処理を指す．

処理モジュール間ソートのアルゴリズムとして並列2ウェイマージソートが効率的であることが知られている[18),19)]．並列2ウェイマージソートはソートフェーズおよびマージフェーズの2つのフェーズから構成される．ソートフェーズでは，各処理モジュールが自モジュール内にあるデータをソートする．本フェーズでは処理モジュール間でプロセスが並列に処理を進めることができる．ソートフェーズの終了時には各処理モジュールごとにソートされたデータ列が生成された状態になる．続くマージフェーズでは，各処理モジュールでソートされた複数のデータ列をマージして一本のソートされたデータ列にする．並列2ウェイマージソートでは，その名のとおりソートされたデータ列が2つずつマージされる．データ列がマージされる様子は二分木で表現され，この二分木はマージ木と呼ばれる．処理モジュール数がN個の場合マージ木の節点は $N-1$ 個である．マージフェーズを並列実行するためには，マージ木に対応して複数のプロセサが二分木状に結合されている必要がある．マージ木の葉には処理モジュールが対応し，各処理モジュールは内部でソートしたデータ列を自分の親の位置にあるプロセサに送る．節点に対応するプロセサは自分の子の位置にある2つのプロセサから出力されたデータを比較し，条件を満たすデータを自分の親の位置にあるプロセサに送る．**図1**にマージフェーズにおけるデータ列のマージの様子を示す．マージフェーズでは，木の同一レベルにあるプロセサが同時に処理を進める並行処理と，木のレベルの異なるプロセサが同時に処理を進めるパイプライン処理の2つの並列処理が実現されている．

各節点が行うマージ処理は2つのデータのソートキーを比較し条件を満たす方のデータを親の節点に転送するという単純な操作であるので，比較を整数に限る等の制約を設けることで各節点は簡単に実装可能である．例えば，テラデータ社のデータベースマシンDBC/1012では，Ynetと呼ばれる二分木と同じ形状をした相互結合網を採用している[13),14)]．Ynetの各スイッチング装置はマージ機能を実装しており，マー

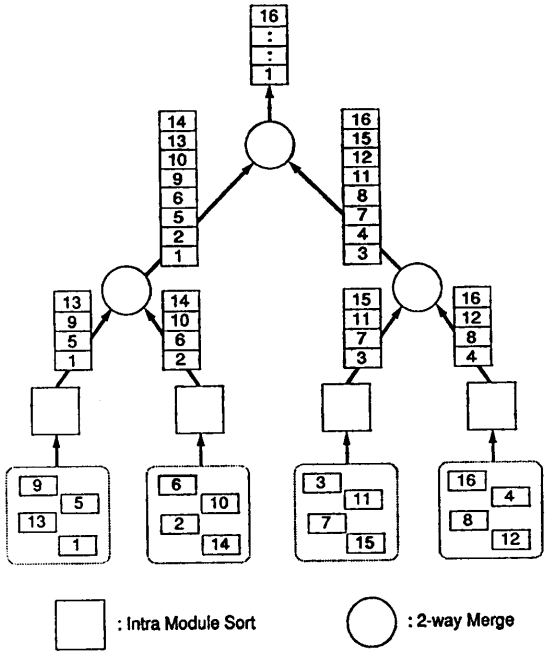


図1 並列2ウェイマージソート
Fig. 1 Parallel 2-way merge sort.

ジ処理が相互結合網によって実現されている．このようにネットワークにマージ機能を導入することにより，処理モジュールに負荷をかけることなく高速処理モジュール間ソートが実現できる．

3. オメガネットワーク

この章では，4章で述べる定理の証明に不可欠なため，相互結合網に用いるオメガネットワークの構成およびそのパスの設定法について簡単に説明する．

3.1 オメガネットワークの構成

オメガネットワークは多数の 2×2 のクロスバスイッチで構成される多段ネットワークの一種である[16]．入出力ポート数がNのとき，段数は $n=\log_2 N$ であり各段は $N/2$ 個のスイッチング装置から構成される．したがって，全スイッチング装置数は $N\log_2 N/2$ である．段間ではシャッフル交換 (Shuffle exchange) を行う．**図2**は $N=16$ の場合のオメガネットワークを示したものである．スイッチング装置の状態には**図3**に示す4つの状態がある．図3の(a)(b)の状態は通常 Straight, Crossed と呼ばれているが，本論文では特に 0-接続，1-接続と呼ぶこととする．図3の(c)(d)の状態はスイッチング装置の入力ポート番号に対応させて 0-ブロードキャスト，1-ブロードキャストと呼ばれ，1対多の入出力結合の実現するのに用いられる．オメガネットワークにより任意の入出力ポート間の結合が実現される．ただし，オメガネットワークは閉塞型ネットワークであるため，複数の入力ポートと出力ポート間の結合を同時には実現できない場合がある．

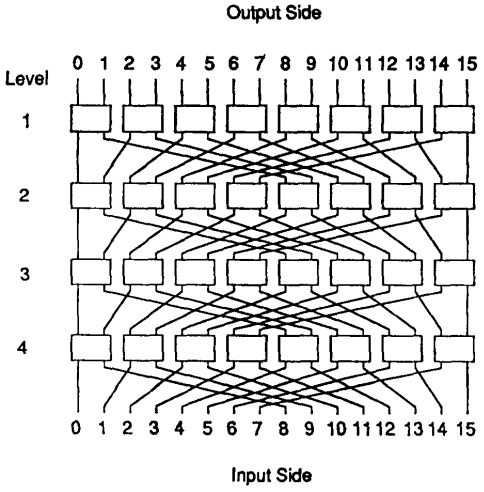


図2 オメガネットワーク ($N=16$)
Fig. 2 Omega network ($N=16$).

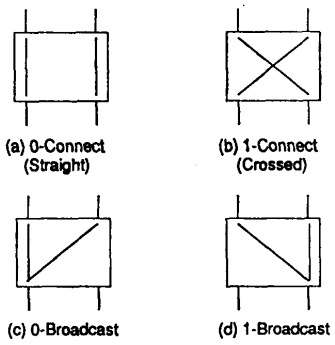


図3 スイッチングユニットの状態
Fig. 3 Four states of a switching unit.

3.2 オメガネットワークにおけるパスの設定法

この節では，オメガネットワークにおけるパスの設定法[16]を簡単に説明する．ここで，パスとは入力ポートから出力ポートにいたるオメガネットワークのスイッチング装置およびリンクで構成される道筋を指す．パスの設定とは入出力ポートを結合するためのスイッチング装置における状態を決定することである．

ネットワークの入出力ポートは左から右に向かって0から $N-1$，各段におけるスイッチング装置は左から右に向かって0から $N/2-1$ までの数字で識別できるので，ポート，スイッチング装置はそれぞれ n ビット，$n-1$ ビットの2進数で表現できる．またネットワークの各段は出力側から入力側に向かって1から n の数字で識別できる．シャッフル交換では，交換前のポート番号を $L=L_{n-1}L_{n-2}\cdots L_1L_0$ (L_i は0または1である）とした場合，交換後のポート番号UはLを用いて $U=L_{n-2}L_{n-3}\cdots L_1L_0L_{n-1}$ となる．

以下では，S番目の入力ポートとD番目の出力ポートを結合するパスの設定法を説明する．S，Dを2進数で表現すると $S_{n-1}\cdots S_0$，$D_{n-1}\cdots D_0$ となる．オメガネットワークの第n段（最も入力側に近い段）では，Sの下位 $n-1$ ビットである $S_{n-2}\cdots S_0$ 番目のスイッチング装置を S_{n-1} と D_{n-1} が一致するならば 0-接続，一致しなければ 1-接続とする．すなわち，⊗で排他的論理和を表せば $S_{n-1}\otimes D_{n-1}$-接続とすればよい．第 l 段 ($1<l<n$) では，S の下位 $l-1$ ビット $S_{l-2}\cdots S_0$ にDの上位 $n-l$ ビット $D_{n-1}\cdots D_l$ を連結した $n-1$ ビットの2進数 $S_{l-2}\cdots S_0D_{n-1}\cdots D_l$ 番目のスイッチング装置を $S_{l-1}\otimes D_{l-1}$-接続とする．第1段（最も出力側に近い段）では，Dの上位 $n-1$ ビット $D_{n-1}\cdots D_1$ 番目のスイッチング装置を $S_0\otimes D_0$-接続とする．

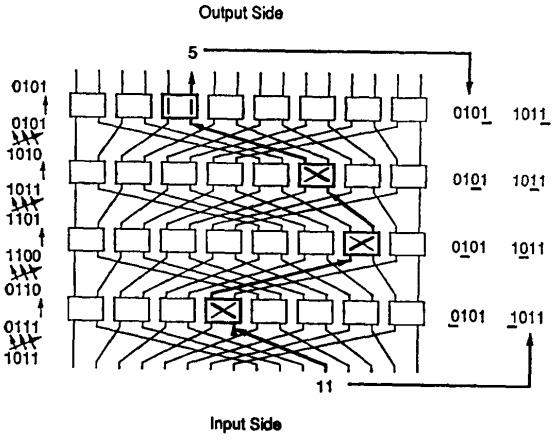


図 4　オメガネットワークにおけるパスの設定
Fig. 4　Pass determination in omega network.

図4は $N=16$ の場合のオメガネットワークにおいて，入力ポートを 11，出力ポートを5とした場合のパスを示したものである．この図においてネットワークの右側に並んだ数字は出力および入力ポート番号を2進数で表現したものである．各段ではこれらの数字のアンダーラインのついたビットが一致するか否かによってスイッチング装置の状態が決定される．ネットワーク左側の数字は各段におけるパスの位置を2進数で表現したものである．例えば，下から二番目の数字 0111_2 $(=7)$ は 1011_2 $(=11)$ の位置から入力されたデータのパスがシャッフル交換によって第4段（最も入力に近い段）では 0111_2 の位置に入力されることを表している．第4段のスイッチング装置は入力ポート番号 1011_2 と出力ポート番号 0101_2 $(=5)$ の第4ビット目は異なっているので，1-接続 (Crossed) となる．したがって，第4段の出力位置は入力位置 1011_2 の最下位ビットが反転した 0110_2 $(=6)$ となる．

4. マージ木のオメガネットワーク上のマッピング

この章では，並列2ウェイマージソートにおけるマージ木のオメガネットワーク上へのマッピングを検討する．

4.1 マージ木のマッピング

まず，マージ木のオメガネットワーク上へのマッピングを定義する．

定義：マージ木の節点をオメガネットワーク上の適切なスイッチング装置に割り当て，枝をオメガネットワーク上の適切な1つのリンクあるいはスイッチング装置によって接続された複数のリンクに割り当てることをマージ木のオメガネットワーク上へのマッピングと定義する．□

マージを行うためには，スイッチング装置の2つの入力ポートから入力されたデータを比較し条件を満たすデータを適切な出力ポートから出力する機能が必要となる．以下では，マージ機能を実現するスイッチング装置の状態を図5のように模式的に表し，(a)(b)の状態をスイッチング装置の出力ポート番号に対応させて 0-マージ，1-マージと呼ぶ．例えば，4本のデータ列を1本にマージする場合を考える．入力ポートが {0, 1, 2, 3} で出力ポートが5の場合，図6(a)のようなマージ木を構成すれば，16×16 のオメガネットワーク上に図6(b)のようにマッピングできる．

マージ木をオメガネットワーク上にマッピングする

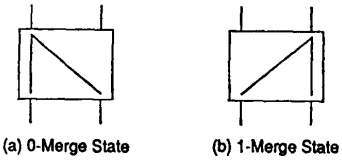
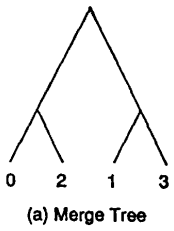


図 5　マージに必要なスイッチングユニットの状態
Fig. 5　Two additional states for merging.



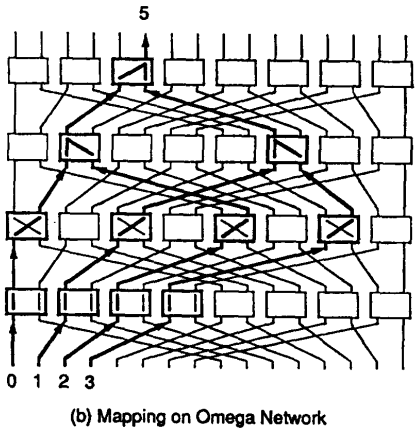


図 6　マージ木の構成とオメガネットワーク上へのマッピング
Fig. 6　Mapping merge tree onto omega network. Input ports: {0, 1, 2, 3}, output port: 5.

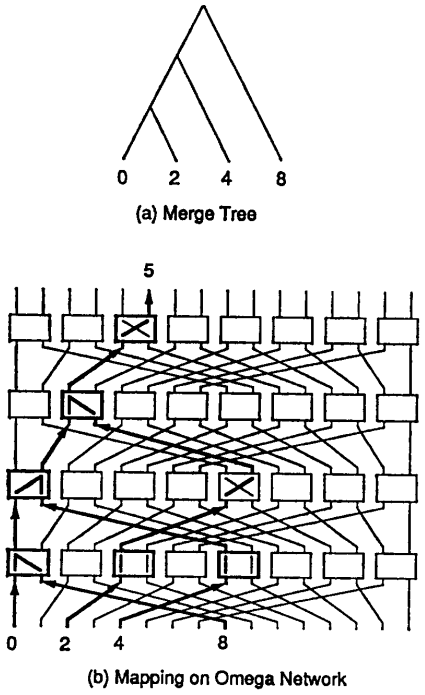


図 7 マージ木の構成とオメガネットワーク上へのマッピング
Fig. 7 Mapping merge tree onto omega network. Input ports: {0, 2, 4, 8}, output port: 5.

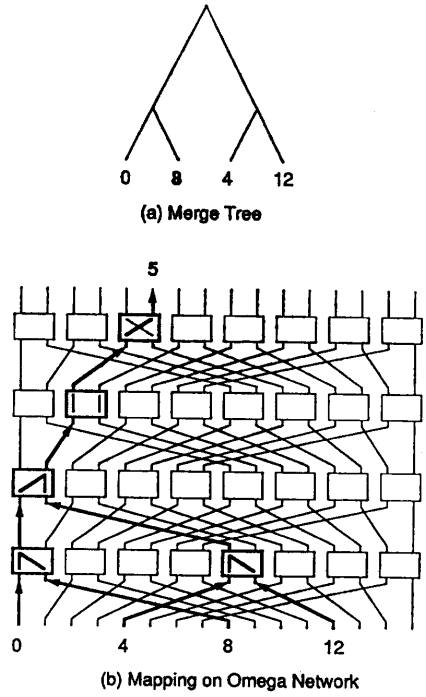


図 8 マージ木の構成とオメガネットワーク上へのマッピング
Fig. 8 Mapping merge tree onto omega network. Input ports: {0, 4, 8, 12}, output port: 5.

際，次の2つの点について検討する必要がある．

- マージ木の形状の決定
- マージ木の節点とオメガネットワークのスイッチング装置との対応の決定

マージ木の形状の決定とは，複数のデータ列をマージする場合様々な形状のマージ木が構成可能であるが，そのなかからオメガネットワークにマッピング可能なものを決定することを指す．入力ポートの集合が {0, 1, 2, 3} である先ほどの例の場合オメガネットワークにマッピングできるマージ木は図6（a）の形状に限られ，入力ポートが {0, 2, 4, 8} の場合マージ木の形状は**図7**（a）となる（この場合，図7（b）のようにオメガネットワーク上にマッピングされる）．一方，節点とスイッチング装置の対応の決定とは，形状が決定したマージ木の節点をオメガネットワークのスイッチング装置に割り当てることを指す．マージ木の形状が同じでもオメガネットワークへのマッピングが異なる場合があるので，注意が必要である．例えば，入力ポートが {0, 4, 8, 12} の場合，マージ木の形状は図6の例と同じく**図8**（a）のようになるが，オメガネットワーク上へのマッピングは異なり図8（b）のようになる．

入力ポートが2つの場合はマージ木の形状が一意に定まるので，マージ木の節点とオメガネットワークのスイッチング装置との対応関係だけを検討すれば良い．そこで，まず4.2節でデータ列が2つである場合のマージ木のマッピングについて検討し，続く4.3節でデータ列が3つ以上である場合を扱う．

4.2 2つのデータ列に対するマージ木のマッピング法

この節では，2つのデータ列をオメガネットワーク内の適切なスイッチング装置でマージし1つのソートされたデータ列とする場合を検討する．マージが行われるオメガネットワーク上のスイッチング装置についてつぎの定理が成り立つ．

定理1（マージが行われるスイッチング装置）：マージされる2つのデータ列が入力されるポート番号をSとT，マージされたデータ列が出力されるポート番号をDとする．このとき，オメガネットワークの第l段でマージが行われることはつぎの条件と同値である．

（a） $l=1$ の場合つぎの関係式を満たす．

$$S_0 \neq T_0$$

（b） $1<l<n$ の場合つぎの関係式を満たす．

$$S_{n-2}S_{n-3}\cdots S_0 \neq T_{n-2}T_{n-3}\cdots T_0$$

$$\left.\begin{array}{c} S_0 = T_0 \\ \vdots \\ S_{l-2}\cdots S_0 = T_{l-2}\cdots T_0 \\ S_{l-1}S_{l-2}\cdots S_0 \neq T_{l-1}T_{l-2}\cdots T_0 \\ \vdots \\ S_{n-2}S_{n-3}\cdots S_0 \neq T_{n-2}T_{n-3}\cdots T_0 \end{array}\right\} \quad (1)$$

(c) $l=n$ の場合つぎの関係式を満たす.

$$\begin{array}{c} S_0 = T_0 \\ \vdots \\ S_{n-2}S_{n-3}\cdots S_0 = T_{n-2}T_{n-3}\cdots T_0 \end{array}$$

上の条件を満たす l に対して, $l=1$ の場合 $D_{n-1}\cdots D_1$ 番目, $1<l<n$ の場合 $S_{l-2}\cdots S_0 D_{n-1}\cdots D_l$ 番目, $l=n$ の場合 $S_{n-2}\cdots S_0$ 番目のスイッチング装置の状態をそれぞれ D_0-マージ, D_{l-1}-マージ, D_{n-1}-マージとすればよい. □

証明: $1<l<n$ の場合についてのみ証明する. $l=1$ および $l=n$ の場合も同様に証明できる. 2つのデータ列はその段より入力側では2つのパスが異なるスイッチング装置を通り, その段から出力側では2つのパスが同じスイッチング装置を通るようになる段でマージされる. 2.2 節で示したパスの設定法から S と D, T と D を結合するスイッチング装置が決定できるので, マージされる条件を l ($1<l<n$) の場合について S, T, D を用いて表現するとつぎのようになる.

$$\left.\begin{array}{c} D_{n-1}\cdots D_1 = D_{n-1}\cdots D_1 \\ \vdots \\ S_{l-2}\cdots S_0 D_{n-1}\cdots D_l = T_{l-2}\cdots T_0 D_{n-1}\cdots D_l \\ S_{l-1}\cdots S_0 D_{n-1}\cdots D_{l+1} \neq T_{l-2}\cdots T_0 D_{n-1}\cdots D_{l+1} \\ \vdots \\ S_{n-2}S_{n-3}\cdots S_0 \neq T_{n-2}T_{n-3}\cdots T_0 \end{array}\right\} \quad (2)$$

(2)式の関係から左辺と右辺で共通な D に関する項を除くと(1)式の関係が得られる. 逆に(1)式の関係より(2)式の関係が成り立つことは明らかなので, (1)式が成立すれば第 l 段でマージが行われる. 以上より, l ($1<l<n$) 段でマージが行われることと(1)式は同値である. Q. E. D.

この定理からマージする段の決定を与えるつぎの定理が導かれる.

定理2 (マージする段の決定法): マージされる2つのデータ列が入力されるポート番号を S と T とする. このとき, 出力側の第1段から入力側の第 $n-1$ 段まで順に S と T の下位 l ビットがはじめて異なる l を捜す. その条件を満たす l が見つかれば, 2つのデータ列はその段でマージされる. もし見つからなければ, 2つのデータ列は第 n 段でマージされる. (証明省略) □

さらにこの定理2から, マージ段の簡単な決定法を与えるつぎの定理が導かれる.

定理3 (マージする段の決定法): マージされる2つのデータ列が入力されるポート番号を S と T とする. このとき, 出力側の第1段から入力側の第 $n-1$ 段まで順に S と T の l ビット目がはじめて異なる l を捜す. その条件を満たす l が見つかれば, 2つのデータ列はその段でマージされる. もし見つからなければ, 2つのデータ列は第 n 段でマージされる. □

証明: 定理2より第 l 段では $S_{l-1}\cdots S_0 \neq T_{l-1}\cdots T_0$ ならばマージされる. しかし, 第 l 段でマージされるか否か調べるのは第 $l-1$ 段まででマージされていない場合に限られるので, $S_{l-2}\cdots S_0 = T_{l-2}\cdots T_0$ である. したがって, 第 l 段では第 $l-1$ ビットを調べればよい. Q. E. D.

図9 は $N=16$ の場合のオメガネットワークにおいて, 入力ポートを {11, 15}, 出力ポートを5とした場合, どのようにマージ段が決定されるかを示したものである. この図において右側の数字は 1011_2 (=11) から入力されたデータ列のパス, 左側の数字は 1111_2 (=15) から入力されたデータ列のパスを表す. 各段ではアンダーラインのついたビットが異なるか否かが, 出力側から入力側に向かって調べられる. この場合, 第3段ではじめて異なるのでこの段でマージが行われる.

以下では, マージ木をオメガネットワーク上にマッピングする, すなわち, オメガネットワーク上のスイッチング装置の状態を適切に設定するアルゴリズムを検討する. **図 10** に示したアルゴリズムでは, 定理3に基づいてデータ列がマージされる段が決定される. ここでは, 出力側からマージされる段を捜しながらスイッチング装置の状態を決定していく. まず, 第1段

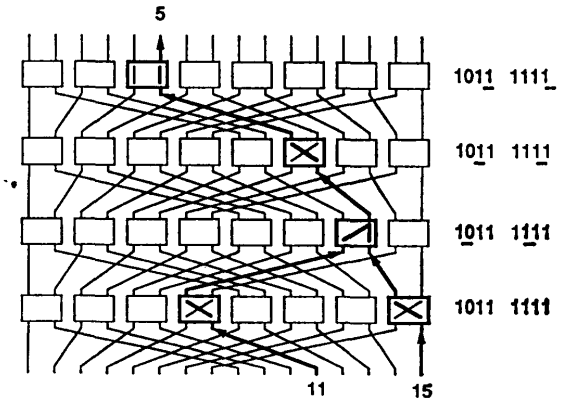


図9 2つのデータ列マージの場合のマージ段の決定
Fig. 9 Determination of switching unit's states.

アルゴリズム1：2つのデータ列のマージ木のマッピング
入力：データストリームの入力ポート番号S,Tと出力ポート番号D

```
if S0 = T0 then
  第1段 Dn-1…D1のSUをS0⊗D0p-接続
else
  第1段 Dn-1…D1のSUをD0-マージ
endif
l = 2
merged = TRUE
while l < n begin
  if merged = TRUE then
    if Sl = Tl then
      第l段 Sl-2…S0Dn-1…Dlの SUを Dl-1-マージ
      merged = FALSE
    else
      第l段 Sl-2…S0Dn-1…Dlの SUを Sl-1⊗Dl-1-接続
    endif
  else
    第l段 Sl-2…S0Dn-1…Dlの SUを Sl-1⊗Dl-1-接続
    第l段 Tl-2…T0Dn-1…Dlの SUを Tl-1⊗Dl-1-接続
  endif
  l = l + 1
end
if merged = TRUE then
  第n段 Sn-2…S0の SUを Dn-1-マージ
else
  第n段 Sn-2…S0の SUを S0⊗Dn-1-接続
  第n段 Sn-2…S0の SUを T0⊗Dn-1-接続
endif
```

図 10 アルゴリズム1（2つのデータ列マージ）
Fig. 10 A mapping algorithm for two data streams.

でSとTの最下位ビット S_0 と T_0 が一致するか調べる．一致すれば第1段 $D_{n-1}\cdots D_1$ のスイッチング装置を D_0-マージとする．一致しなければ，この段より入力側の段ですでにマージされているので，$D_{n-1}\cdots D_1$ のスイッチング装置を $S_0\otimes D_0$-接続する．第1段でマージされなければ，入力側へ向かって第l段ではSとTの下位第lビットが一致するか調べる．一致すればその段でマージされるので，$S_{l-2}\cdots S_0D_{n-1}\cdots D_l$ のスイッチング装置を D_{l-1}-マージとする．一致しなければ $S_{l-2}\cdots S_0D_{n-1}\cdots D_l$ のスイッチング装置を $S_{l-1}\otimes D_{l-1}$-接続する．マージされた後は，第l段では $S_{l-2}\cdots S_0D_{n-1}\cdots D_l$ のスイッチング装置を $S_{l-1}\otimes D_{l-1}$-接続，$T_{l-2}\cdots T_0D_{n-1}\cdots D_l$ のスイッチング装置を $T_{l-1}\otimes D_{l-1}$-接続とする．第$n-1$段までにマージされていないならば第n段でマージされ，$D_{n-2}\cdots D_0$ のスイッチング装置を D_{n-1} マージとする．

4.3 複数のデータ列に対するマージ木のマッピング法

前節で扱った2つのデータ列をマージする場合マージ木の形状は節点が常にただ1つなので問題とならなかったが，複数（3つ以上）のデータ列を扱う場合マージ木をどのように構成するか検討する必要がある．

複数のデータ列の場合でも，定理3に基づいてマージ木を構成することが可能である．すなわち，複数のデータ列を2つのデータ列の組の集合と捉えれば，各2つのデータ列の組について定理3によりどの段でマージされるか決定できるので，それをもとにマージ木の形状を決定できる．データ列の組合せにより同一スイッチング装置をマージ状態と接続状態というように設定しようとする状態の衝突が発生した場合，マージを優先してスイッチング装置の状態を決定すればよい．例えば，A，B，Cの3つのデータ列をマージする場合，AとBが先に（より入力側に近い所で）マージされるとする．このとき，AとC，BとCの組はそのスイッチング装置をマージ状態に設定するが，AとBの組はそのスイッチング装置を接続状態に設定しようとするので衝突が起こる．この場合，そのスイッチング装置をマージ状態に設定する．

しかし，単純に2つのデータ列の組合せごとにスイッチング装置の状態決定を行うと，M個のデータ列に対するマージ木をオメガネットワークにマッピングするのに $O(M^2)$ の計算量が必要となる．マージされたデータ列の管理をマージ後は1つのデータ列として行えば，マージが進むにつれマージされるか否かを調べなければならない組合せが減らすことが可能となり，スイッチング装置の状態決定に必要な計算量を減少させることができる．しかし，そのためにはマージが行われるか否かを入力側から調べる必要がある．先ほどの定理3はマージする段が出力側から調べる場合の決定法を与えたが，つぎの定理はマージする段を入力側から調べる場合の決定法を与える．

定理4（マージする段の決定法）：マージされる2つのデータ列が入力されるポート番号をSとTとする．このとき，入力側の第n段から出力側の第2段まで順にSとTの下位 $l-1$ ビットがはじめて一致するようなlを捜す．その条件を満たすlが見つかれば，2つのデータ列はその段でマージされる．もし見つからなければ，2つのデータ列は第1段でマージされる．（証明省略） □

この定理に基づいてマージ木を構成することができる．

定理5（マージ木の構成法）：入力側の第n段から出力側の第2段まで順に，全入力ポート番号のなかからはじめて下位 $l-1$ ビットが一致するような入力ポート番号の組合せを探す．その条件を満たす組合せが見つかれば，その組合せはその段でマージされる．その際，マージされた組合せのどちらか一方を入力ポート

番号の集合から取り除く．もし見つからなければ，その段でマージされる組合せはない．第1段で入力ポートが2つ残っていれば，その組合せは第1段でマージされる．このようにして，オメガネットワークにマッピング可能なマージ木が構成される．(証明省略)□

例えば，データ列を入力する処理モジュールが {0, 1, 4, 8, 11, 15} の場合，マージ木は**図 11** のよう構成される．この図で“X”はそのビットが比較において0，1のいずれでもよいことを示している．例えば，第4段では下位3ビットが一致するか否かを調べればよく，{0, 1, 4, 8, 11, 15} の組合せの中では0と8の組のみが一致しているので，その組のみが第4段でマージされる．

定理5から，**図 12** に示したような複数のデータ列に対するマッピングアルゴリズムが導かれる．このアルゴリズムでは，入力ポート番号の集合をAとして管理している．出力側から入力側に向かって各段ごとにAのなかの2つのポート番号の組合せがマージされるか否か調べる．まず，Aからポート番号の一番小さいものを取りだす（Sとする）．つぎにSに対してAからSより大きいポート番号のものを順に取りだし（Tとする），定理4に従って第 l 段では下位 $l-1$ ビットが一致するか調べる．一致する場合はその組合せは第 l 段でマージされるので，$S_{l-2}\cdots S_0D_{n-1}\cdots D_l$ のスイッチング装置を D_{l-1}-マージとしポート番号の大きい方をAから取り除く．Sに対して下位 $l-1$ ビットが一致する組合せが見つからない場合，Sはその段でマージされないので，$S_{l-2}\cdots S_0D_{n-1}\cdots D_l$ のスイッチング装置を $S_{l-1}\otimes D_{l-1}$-接続する．この操作をAの組合せで調べていないものがなくなるまで繰り返すことで，その段のスイッチング装置の状態がすべて設定される．第1段（最も入力に近い段）では，それまでにすべての組合せがマージされていれば（このときAに残っているポート番号をSとする）$D_{n-1}\cdots D_1$ のスイッチング装置を S_0-$\otimes D_0$-接続とし，マージされていなければ $D_{n-1}\cdots D_1$ のスイッチング装置を D_0-マージとする．先ほどの図 11 に示したマージ木は出

<u>アルゴリズム2：複数のデータ列のマージ木のマッピング</u>
入力：データストリームの入力ポート番号の集合Aと出力ポート番号D

```
merged = FALSE
l = n
while l > 1 begin
  while Aから小さい順に要素を出せる(Sとする) begin
    while AのSより大きい要素を小さい順に出せる(Tとする) begin
      if S_{l-1}…S_0 = T_{l-1}…T_0 then
        第l段 S_{l-2}…S_0D_{n-1}…D_l のSUを D_{l-1}-マージ
        (第n段の場合 S_{l-2}…S_0のSUを D_{n-1}-マージ)
        AからTを取り除く
        merged = TRUE
        break
      endif
    end
    if merged = FALSE then
      第l段 S_{l-2}…S_0D_{n-1}…D_lのSUを S_{l-1} ⊗ D_{l-1}-接続
      (第n段の場合 S_{n-2}…S_0のSUを S_{n-1} ⊗ D_{n-1}-接続)
    else
      merged = FALSE
    endif
  end
  l = l + 1
end
if Aの要素の数が2 then
  第1段 D_{n-1}…D_1のSUを D_0-マージ
else
  第1段 D_{n-1}…D_1のSUを S_0 ⊗ D_0-接続
endif
```

図 12 アルゴリズム2（複数のデータ列マージ）
Fig. 12 A mapping algorithm for any number of data streams.

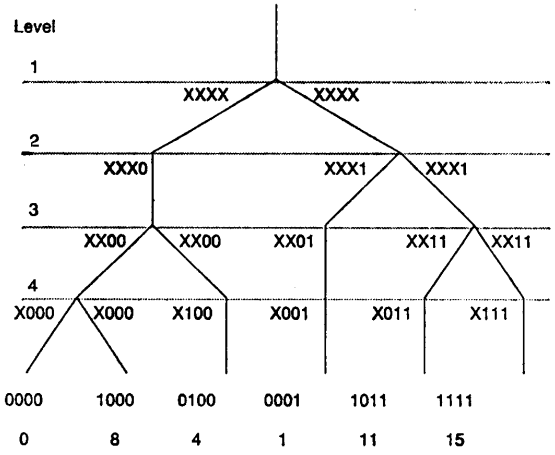


図 11 マージ木の生成
Fig. 11 Generation of merge tree. Input ports: {0, 1, 4, 8, 11, 15}.

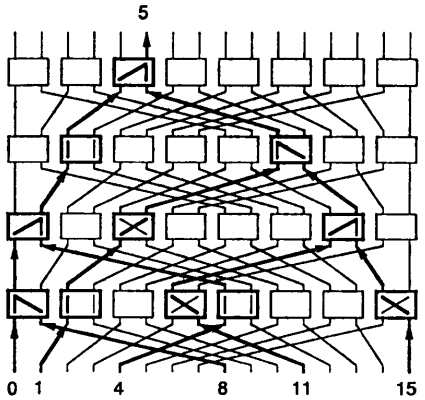


図 13 マージ木のオメガネットワーク上へのマッピング
Fig. 13 Mapping merge tree on omega network. Input ports: {0, 1, 4, 8, 11, 15}, output port: 5.

力ポートを5とした場合オメガネットワーク上に図13に示すようにマッピングされる.

5. 考　　察

ここでは,本オメガネットワークにおけるスイッチング装置の構成とソート時間とについて考察する.

5.1 スイッチング装置の構成

並列2ウェイマージソートでは,2章で述べたようにマージ木の節点にあるプロセサは子の位置にあるプロセサから出力されたデータを比較し条件を満たすデータを親の位置にあるプロセサに送る.したがって,各プロセサにはこの位置にあるプロセサから出力されたデータを保持するための2データ分のメモリがあればよい*.同様にして,本論文で提案したオメガネットワークの各スイッチング装置はその位置によらず2データ分のメモリを用意すれば良い.さらに,スイッチング装置にはデータの大小を決定するための比較器が必要である.

我々がすでに提案しているバケット平坦化機能を有するオメガネットワーク[17]においては,バケット平坦化を実現するためのカウンタ用メモリおよびカウンタ値からスイッチング装置の状態を決定するための比較器が用意されている.したがって,バケット平坦化機能を有するオメガネットワークにマージ機能を付加するには制御部分を改良するだけで対応できる.

5.2 ソート時間

並列2ウェイマージソートを木ネットワーク上で実行する際のソート時間は次のように与えられる[2].各処理モジュールが自モジュール内でデータ列をソートするのに要する時間は無視し,ソートされたデータ列のマージに要する時間のみを考える.ここでは,P (Pは2の累乗とする)本の長さLのデータ列をマージする基本的な場合を扱う.また,木の各接点のプロセサが子プロセサから出力されたデータの読み込みおよび比較に要する時間を単位時間とする.木の段数が$\log_2(P)$なので,最初のデータが根から出力されるには$\log_2(P)+1$単位時間がかかる.それ以降データは2単位時間間隔で根から出力されるので,残りのデータが出力されるのに$1+2(PL-1)$単位時間かかる**.したがって,ソート時間は$\log_2(P)+2(PL-1)$単位時間となる.

本オメガネットワークでは,木ネットワークの場合と最初のデータが出力段から出力されるまでの時間が異なる.最初のデータが出力されるにはオメガネットワークの段数が$\log_2(N)$なので,データ列の数Pに依存せず,$\log_2(N)+1$単位時間かかる.ここで,Nはネットワークの入出力ポート数である.最初のデータが出力されてから最後のデータが出力されるまでに要する時間は木ネットワークの場合と等しいので,全体では$\log_2(N)+2(PL-1)$の時間がかかる.木ネットワークと本オメガネットワークを比較するとソートに要する時間の差は$\log_2(N)-\log_2(P)$だけであり,この値はソートするデータ列の長さに依存せずソート時間と比較して非常に小さい値である.また,$P=N$(すべての処理モジュールからソートすべき部分列が生成される)場合はソート時間は両者で等しくなる.

6. おわりに

共有メモリを持たない疎結合並列マシン上の処理モジュール間に分散しているデータのソートは並列2ウェイマージソートにより効率的に実現される.その際,相互結合網にマージ機能を埋め込むことができれば,処理モジュールに対するソートの処理負荷を小さくし,処理モジュール間ソートを高速実行できる.マージ木の形状と同じ木ネットワークの節点にデータの比較機能を持たせることで,2ウェイマージを行う木ネットワークは既に実用化されている.本論文では,結合能力等の点でネットワークより優れており相互結合網として一般的に用いられている多段結合網の1つであるオメガネットワークに2ウェイマージ機能を組み込む方式を提案した.

オメガネットワーク上でマージソートを行うためには,マージ木の節点とネットワークのスイッチング装置の対応を決定できなければならない.データが2つの処理モジュールに分散している場合について,マージ木がオメガネットワークにマッピング可能であることを示すとともに,オメガネットワークの状態決定アルゴリズムについて述べた.さらに,データが3つ以上の処理モジュールに分散している一般的な場合について,マージ木の形状を検討し,決定されたマージ木をオメガネットワークのスイッチング装置にマッピングするアルゴリズムを提案した.つぎに,スイッチング装置の構成を検討した.各スイッチング装置は2データ分のメモリと比較器が必要であるが,そのために要求されるハードウェアはわずかであり簡単に実装

---

* この点で,最終段のプロセサにストリーム長の半分のメモリが必要となるパイプラインマージソート[3]と異なる.

** スイッチング装置のデータ用のメモリを二重バッファ構成とすれば入出力は重畳可能なので,データは1単位時間間隔で出力され,残りのデータを出力するのに$PL-1$単位時間かかる.

可能である．ソート時間についても検討した結果，木ネットワークの場合と比較してソートに要する時間の差は非常に小さく，さらに，ソートすべきデータ列が全処理モジュールから生成される場合には木ネットワークとソート時間が一致することを示した．最後に本論文で述べたマッピングアルゴリズムは，オメガネットワーク以外の $\log_2 N$ 段からなる多段結合網，さらには，$K \geq 3$ であるK入力K出力クロスバスイッチから構成される多段結合網に対しても容易に拡張可能であることを指摘しておく．

参考文献

1) Bitton, D., DeWitt, D. J., Hsiao, D. K. and Menon, J.: A Taxonomy of Parallel Sorting, *ACM Comput. Surv.*, Vol. 16, No. 3, pp. 287-318 (1984).
2) Akl, S. G.: *Parallel Sorting Algorithms*, Academic Press (1985).
3) Tanaka, Y., Nozaka, Y. and Masuyama, A.: Pipeline Searching and Sorting Modules as Components of a Data Flow Database Computer, *Proc. of IFIP*, pp. 427-432 (1980).
4) Kung, H. K.: The Structure of Parallel Algorithms, *Advances in Computer*, Vol. 19, pp. 65-112 (1980).
5) 安浦寛人，高木直史：並列計数法による高速ソーティング回路，電子通信学会論文誌，Vol. J 65-D, No. 2, pp. 179-186 (1982).
6) 喜連川優，伏見信也，桑原和弘，田中英彦，元岡 達：パイプラインマージソータの構成，電子通信学会論文誌，Vol. J 66-D, No. 3, pp. 332-339 (1983).
7) Satoh, T., Takeda, H. and Tsuda, N.: A Compact Multiway Merge Sorter Using VLSI Linear-Array Comparators, *Proc. Int. Conf. on Foundations of Data Organization*, pp. 223-227 (1989).
8) Kitsuregawa, M., Yang, W., Fushimi, S., Kimura, H., Shinano, J. and Kasahara, Y.: Implementation of LSI Sort Chip for Bimodal Sort Memory, *Proc. of the Int. Conf. on VLSI*, pp. 285-294 (1989).
9) 喜連川優，伏見信也：データベースマシン，情報処理，Vol. 28, No. 1, pp. 223-234 (1987).
10) DeWitt, D. J., Gerber, R. H., Graefe, G., Heytens, M. L., Kumar, K. B. and Muralikrishna, M.: GAMMA, A High Performance Dataflow Database Machine, *Proc. of the 12th Int. Conf. on VLDB*, pp. 228-237 (1986).
11) 楊 維康，平野 聡，瀬川芳久，喜連川優，高木幹雄：スーパーデータベースコンピュータSDC のアーキテクチャ，第39回情報処理学会全国大会論文集，pp. 1544-1545 (1989).
12) The Tandem Performance Group: A Benchmark of Nonstop SQL on the Debit Credit Transaction, *ACM SIGMOD '88*, pp. 337-341 (1988).
13) Teradata Corp.: DBC/1012 Data Base Computer Concepts and Facilities, Technical Report C02-0001-05, Teradata Corp. (1988).
14) Neches, P. M.: The Ynet: An Interconnect Structure for a Highly Concurrent Data Base Computer System, *Proc. of 2nd Symp. on the Frontiers of Massively Parallel Computation*, pp. 429-435 (1988).
15) Lorie, R. A. and Young, H. C.: A Low Communication Sort Algorithm for a Parallel Database Machine, *Proc. of the 15th Int. Conf. on VLDB*, pp. 125-134 (1989).
16) Lawrie, D. H.: Access and Alignment of Data in an Array Processor, *IEEE Trans. Comput.*, Vol. C-24, No. 12, pp. 1145-1155 (1975).
17) 喜連川優，小川泰嗣：バケット平坦化機能を有するオメガネットワーク，情報処理学会論文誌，Vol. 30, No. 11, pp. 1494-1503 (1989).
18) Bitton, D. and DeWitt, D. J.: Duplicate Record Elimination in Large Data Files, *ACM Trans. Database Syst.*, Vol. 8, No. 6, pp. 255-265 (1983).
19) Beck, M., Bitton, D. and Wilkinson, W. K.: Sorting Large Files on a Backend Multiprocessor, *IEEE Trans. Comput.*, Vol. C-37, No. 7, pp. 769-778 (1988).
20) 喜連川優，楊 維康，鈴木慎司：VLSI ソートプロセッサ，情報処理，Vol. 31, No. 4, pp. 457-465 (1990).

(平成元年11月15日受付)
(平成2年11月13日採録)

喜連川 優（正会員）

昭和30年生．昭和53年東京大学工学部電子工学科卒業．昭和58年同大学院情報工学専門課程博士課程修了．工学博士．同年，東京大学生産技術研究所講師．現在，同研究所助教授．並列コンピュータアーキテクチャ，データベースマシン，データ工学などの研究に従事．電子情報通信学会，電気学会，IEEE, ACM 各会員．

小川 泰嗣（正会員）

1962年生．1985年東京大学工学部計数工学科卒業，1987年同大学院工学系研究科情報工学専攻修士課程修了．同年，(株)リコー入社．中央研究所にて情報検索およびデータベースシステムの研究開発に従事．IEEE, ACM 各会員．