大容量データベース処理に適したソータ構成法†

佐藤哲司†† 武田英昭†† 津田伸生††

非数値処理分野で多用されるソーティングを，専用ハードウェアを用いて高速化するソータの研究が活発に行われている．特に，関係データベース処理の分野では，小形・高性能なハードウェアソータが実用に供せられてきている．本論文では，大容量データベース処理への適用を狙いとして，レコード数に依存しないハードウェア構成が採れる逐次多段マルチウェイマージソート法を提案する．k個のレコードを並列に比較するソートアレイと，マージするk本のレコード列の選択にデータ駆動型制御を適用することによって，十分に大きなウェイ数kのマージ処理を簡単な制御回路で高速に実現する．マージウェイ数を大きくしたこと，ソータへのレコード転送に重畳してマージ処理を行える構成とすることにより，ソート処理時間を短縮する．また，レコードを比較するソートアレイとレコードを格納するワークメモリを独立に実装できることから，ソート処理速度とソートできる最大レコード数の2つの独立した要求条件を同時に満足する構成が柔軟にとれる．シングルボードソータの試作とデータベースプロセッサ RINDA への適用により，その動作および大容量データベース処理への高い適用性を確認したので，その構成と性能について報告する．

1. はじめに

ソーティングは非数値計算分野で多用される基本的な演算であり，さまざまなソフトウェアのアルゴリズムが提案され，適用領域についても明らかにされてきた[1]．しかし，計算機システムによるデータ処理の普及にともない，大型計算機を用いても多大な処理時間を要する大量レコードの高速ソートが要求されてきた．特に，関係データベース処理分野では，アプリケーションによってソートするレコードの個数が大きく変動し，100万個を超えるレコードのソートも必要となってきた．また，レコード長が整数のような短いものから，文字列のような長いものまで効率よくソートできなければならない．一方，近年の VLSI 技術の進歩はめざましく，特定用途向けの専用ハードウェアを実現することが可能となってきた．本論文では，レコード数やレコード長に対して高い柔軟性が要求される関係データベース処理への適用を狙いとして，小形・高速なソータを実現できる逐次多段マルチウェイマージソート法を提案する．

これまでも，専用ハードウェアによるソータの研究は盛んに行われていたが，従来のソータは，並列比較による処理時間の短縮に主眼が置かれていたため，ソータのハード量が，ソート対象とするレコード数Nに対して$O(\log_2 N)$[2)~6)]，または$O(N)$[8)~11)]となる構成，すなわち，ハード量がレコード数に依存して増加する構成が一般的であった．100万個のレコードをソートするには，比較的ハードウェア量が少ないとされる$O(\log_2 N)$系の代表的なソータであるパイプラインマージソータ[4),5)]でも，20段のマージ回路が必要となり，各マージ回路ごとに容量の異なるワークメモリを分散して実装する必要もあって，小形化を阻む要因となっていた．マージ回路数で決まる個数を超えるレコードをソートするための複雑な制御を実現したソータ[4),5)]や，種々のレコード長に対応できるソータ[5),6)]の構成が報告されているが，レコード数やレコード長に柔軟に対応するには，制御の複雑化が避けられなかった．

本論文のマージソータは，ソート対象となるレコードを格納するワークメモリと，1次元アレイ構造の並列比較回路（ソートアレイ）で構成され，マージするレコード列の選択にデータ駆動型制御を適用したことにより，数十ウェイのマージ処理をウェイ数に依存しない速度で高速に行える．このマージ処理を逐次的に繰り返すことで大量なレコードをソートする．

ソートできる最大レコード数は，集中実装された1つのワークメモリの容量で決まり，比較回路や制御回路の構成に影響されない．ワークメモリの容量とレコード長で決まるレコード数をソートできることから，短いレコードはより多くソートできる．一方，ソート可能な最大レコード長は，実用上のデータベース処理を考慮して，ハードウェアの複雑化を避けるために現実的な長さとしている．1次元アレイ構造のソートアレイは，VLSI 技術との適合性が高く，複数の比較器を一括集積することで大幅な小形化がはかれる．

2章では，マルチウェイマージ法とハードウェア構

† A Multiway Merge Sorter for Sorting of Large Databases by TETSUJI SATOH, HIDEAKI TAKEDA and NOBUO TSUDA (NTT Communications and Information Processing Laboratories, Base Systems Architecture Laboratory).

†† NTT 情報通信処理研究所基本アーキテクチャ研究部

成について述べる．3章では，大容量ソータを実現するための逐次多段マージソート法と繰り返し使用するワークメモリへのレコード列格納法，および並列比較を行うソートアレイの構成法を示す．4章では，試作ソータの構成と性能について評価結果を示し，リレーショナルデータベースマシン RINDA への適用事例について述べる．

2. マルチウェイマージアルゴリズム

データベース処理におけるソートは，ディスク等の2次記憶に格納されたデータベースに対して，選択等の処理を行った結果に対して行う．このため，ソートするレコードの個数をあらかじめ知ることができないばかりか，アプリケーションによっては膨大な個数のレコードをソートする必要もあり，ソートできる最大レコード数がマージ回路の段数に依存する従来のソータ[2]では，マージ回路の段数を最適化することが困難であった．

筆者らは，レコードの比較回路とレコードの格納回路を分離することで，ソートできる最大件数とソート速度の2つの独立した要求条件を同時に満足できるマルチウェイマージソート法を提案している[12),13)]．以下では，マルチウェイマージを行うハードウェアの構成とアルゴリズムを詳述する．

2.1 ハードウェア構成

レコードの並列比較を行うソートアレイと，レコードを格納するワークメモリ，およびマージ制御回路からなるソータの基本構成を**図1**に示す．ソートアレイは，2個のレコードを格納して大小関係を判定する比較エレメントを1次元アレイ状に配置してある．ワークメモリは，パイプラインマージソータ[6]と異なり全体を集中実装できることから，市販の大容量 RAM チップを用いて汎用計算機の主記憶装置と同様に極めて高密度に実装できる．マージ制御回路は，k（k は2以上の整数）本のレコード列をマージする k ウェイマージ制御と，最終的に1本のレコード列するための k ウェイマージ処理の逐次制御を行う回路である．ソートアレイの並列比較動作を**図2**に示す．図は，1桁の整数からなるレコードを降順にソートする例である．各比較エレメントは，同期して比較操作と比較に基づく転送操作を行う．ソートアレイの左端からレコードを入力する入力操作時は，各エレメントに格納された2個のレコードの小さい方（正確には大きくない方）を右方向に隣接しているエレメントに転送する．逆に，出力操作時には，大きい方のレコードを左方向に転送する．この結果，ソートアレイに入力されたレコードのうちで最大なレコードは，常に，ソートアレイの最左端のエレメントに保持され，レコードの入力/出力回数，あるいは，それらの順序関係に関わらず，待ち時間なしで取り出せる．$\lceil k/2 \rceil$ 個の比較エレメントからなるソートアレイでは，k 個のレコードのソートと k 本のレコード列のマージとを，レコードの入出力時間で行える．ここで $\lceil k/2 \rceil$ は，$k/2$ より小さくない最小の整数を示す．

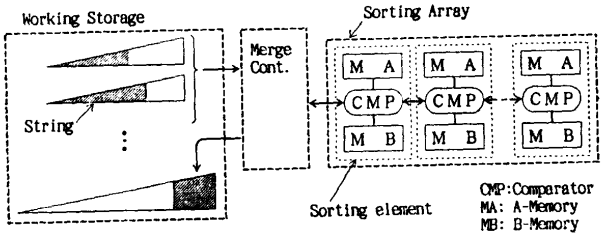

図1 マルチウェイマージソータの基本構成
Fig. 1 Block diagram of multiway merge sorter.

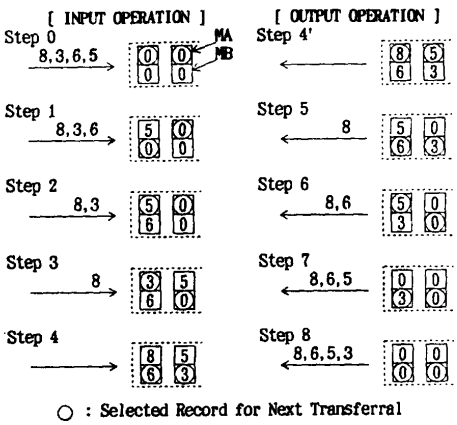

図2 ソートアレイの並列比較動作（降順ソート）
Fig. 2 Schematic diagram of input-output operation.

2.2 マルチウェイマージアルゴリズム

マージウェイ数を大きくすることによって，少ないマージ段数で大量のレコードをソートできる．マージウェイ数を拡大しても制御が複雑にならず，高いスループットを維持できるバンクタグを用いたデータ駆動型マージ制御法について述べる．

バンクタグを用いた k ウェイマージ制御を，**図3**に示す4ウェイマージを例として説明する．ワークメモリに格納された4本のレコード列には，#0 から #3 のバンク番号が付与してある．

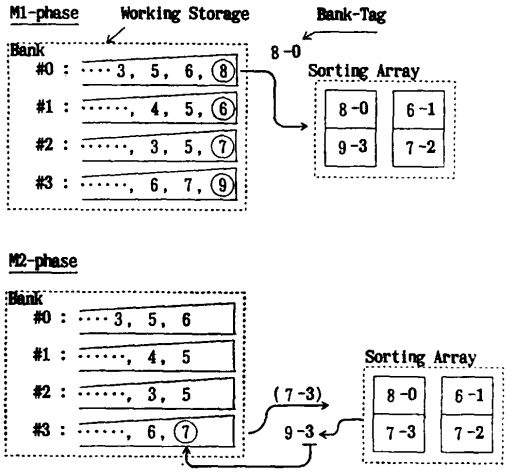

図 3 バンクタグを用いたマルチウェイマージ制御
Fig. 3 Continuous multiway merging diagram using bank-tag.

M1フェーズ：マージ対象とする各レコード列の最大レコード，すなわち各レコード列の先頭レコードに，そのレコードが属していたレコード列を識別するバンクタグを付与して，ソートアレイに入力する．

M2フェーズ：ソートアレイから1個のレコードを出力し，出力したレコードのバンクタグが指すレコード列から次のレコードを読み出してソートアレイに入力する．このマージ操作をすべての入力レコード列が空になるまで繰り返す．

最初の出力レコードは，M1フェーズの終了時点で最左端のエレメントに保持されているレコード，すなわちマージするレコード列内の最大レコードである．出力された最大レコードの次に大きいレコードの候補は，ソートアレイ中に残存する $(k-1)$ 個のレコード，あるいは，出力した最大レコードが属していたレコード列中の次のレコード（現時点での先頭レコード）であるから，上述のマージ操作を繰り返すことによって全体をマージできる．

k ウェイマージ処理の流れを制御するバンクタグは，レコードの大小比較に影響を与えないように，例えばレコードの最後部に付与することで，ソートアレイ内ではレコードと一体として扱うことができる．また，バンクタグとして必要な最小のビット数は $\lceil \log_2 k \rceil$ であり，$\lceil k/2 \rceil$ 個の比較エレメントを持つソートアレイを用いて，大きなウェイ数 k のマージ処理を少ない制御オーバヘッドで容易に実現できる．

N 個のレコードをソートするのに必要な全マージ段数（ステージ数）は $\lceil \log_k N \rceil$ である．マージウェイ数とレコード数の関係を，ステージ数をパラメータとして図4に示す．マージウェイ数を増やすとより少ないステージ数で大量のレコードをソートできることが判る．また，N 個のレコードのソート処理時間，すなわち，最初のレコードの入力開始から最後のソート結果出力までの時間は，各ステージの処理がソートアレイによる並列比較で $O(N)$ で実現されることから，この時間はステージ数に比例する時間 $O(N \cdot \log_k N)$ となる．

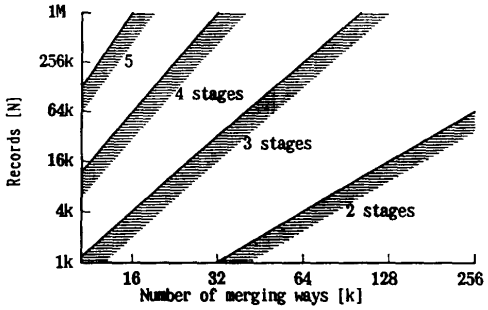

図 4 逐次多段マージによる大容量ソート
Fig. 4 Relationship of merging ways and record numbers.

2.3 ソートアルゴリズムの特徴

大量レコードを k ウェイマージ処理の繰り返しでソートする基本アルゴリズムを示した．以下に，その特徴をまとめる．

（1） ソートアレイによる並列比較とバンクタグを用いたデータ駆動型制御とによって，数十ウェイのマージ回路を容易に実現できる．ウェイ数が大きいマージ処理を $O(N)$ の処理時間で実現できたことから，少ないステージ数で大量のレコードを高速にソートできる．現実的な応用では，数十ウェイのマージ回路を用いて $O(3N)$ あるいは $O(4N)$ の処理時間を達成できる．

（2） 最大ソート可能なレコード件数は，ワークメモリの容量で決まり，ソートアレイや制御回路の構成に依存しない．また，ソートアレイの規模を大きくしてマージウェイ数を拡大することで，マージ段数を削減してソート処理時間を短縮できる．この結果，ソート速度と最大レコード数の2つの独立した要求条件に応じた柔軟な構成のソータを実現できる．

（3） レコードの比較を行うソートアレイと，レコードを格納するワークメモリとを分離して実装できるため，ハードウェア化の範囲や規模を柔軟に設定できる．例えば，専用ハードウェア化したソートアレイを汎用計算機に付加して，従来ソフトウェアで行っていたマージソートを高速化する構成や，マージ制御回

路も含めて全体を専用ハードウェア化し，小形・高速なソータを実現することもできる．

3. 大容量ソータの構成法

データベース処理に適用するソータは，検索結果の一時表を入力とすることから，ソート対象となるレコードの個数および長さに柔軟に対応できなければならない．本章では，このような要求を満足できる逐次多段マージソータの構成法を示す．本ソータでは，ワークメモリに格納された複数のレコード列をマージして再度ワークメモリに格納する処理を段階的に繰り返して大量レコードをソートする．ワークメモリに効率よくレコード列を格納する方法，ソートするレコード長に柔軟に対処できるソートアレイの構成法についても詳述する．

3.1 逐次多段マージソート法

（1） 多段マージ制御法

マルチウェイマージ法は，3段あるいは4段のマージ処理で実用上十分な個数のレコードをソートできる．このため，従来のパイプラインマージソータのようなハードウェア的な多段化ではなく，逐次的なマージ処理で高速なソータを小形に実現できる．逐次多段マージソートは以下の3段階からなる．

マージ準備段階 (Pre-merge Stage)：入力される一連のレコードを，k 個ずつソートアレイに入力/出力して，大きさ k のソート済みレコード列としてワークメモリに格納する．入力が完了すると，ワークメモリには $\lceil N/k \rceil$ 本のレコード列が格納される．ここで，レコード列の大きさとは，レコード列に含まれるレコードの個数である．

中間マージ段階 (Intermediate Merge Stages)：ワークメモリに格納されたレコード列を k 本ずつマージして，再度ワークメモリに格納する k ウェイマージを段階的に繰り返す．1段の中間マージ処理で，レコード列の大きさは最大 k 倍に，レコード列の本数は最大 $1/k$ となる．レコード列数が k 以下となるまで中間マージ段階の処理を継続する．

出力マージ段階 (Output Merge Stage)：中間マージ段階までの処理で，k 本以下となったワークメモリ中のレコード列をマージして出力する．

マージ準備段階および出力マージ段階の処理は，データベースを管理しているホスト計算機等とソータとの間のレコード転送時間に重畳して処理できることから，レコードの入力完了から出力開始までのソータ保留時間は $N\times(\lceil \log_k N \rceil - 2)$ となる ($N \geqq k^2$)．この時間は，実用上は1段，最悪でも2段の中間マージ処理時間である．また，逐次マージ処理とすることで，ソートできる最大レコード数と無関係にソータのハードウェアを構成できる．

（2） レコード列の格納法

逐次多段マージ処理の中間マージ段階では，ワークメモリから読み出した複数のレコード列をマージして再度ワークメモリに格納する．このため，ワークメモリを有効に利用できるメモリ管理法が必要となる．以下では，ソートするレコードの個数を N，長さを L として，メモリ利用率 ($=LN$/メモリ容量) を比較評価する．

従来のメモリ管理法として，2重メモリ法，ポインタ法，ブロック分割法がある[7]．ワークメモリを2つの領域に分けて使用する2重メモリ法は，制御は簡単であるがメモリ利用率は 1/2 である．各レコードにポインタを持たせて空き領域を管理するポインタ法は，固定長レコードであれば制御は極めて簡単であるが，メモリ利用率は $L/(L+$ポインタ長$)$ であり，レコード長が短いと利用率が低下する．ブロック管理法は，固定サイズのブロック単位で空き領域を回収・再利用し，ブロック内はポインタチェインで管理する方法であり，2重メモリ法では2分割であったワークメモリの分割を細分化してメモリ利用率を向上した方法といえる．この方法は，マージウェイ数 k に等しい補助ブロックを必要とし，ウェイ数が小さい領域で有効な手法であった．

そこで，逐次多段マージソータのメモリ管理法として，メモリ利用率を $k/(k+1)$ にできるエリア格納法を検討した．この方法は，マージウェイ数を大きくすることによって利用率を向上できる．エリア格納法を用いた逐次多段マージ処理の手順を**図5**に示す．マー

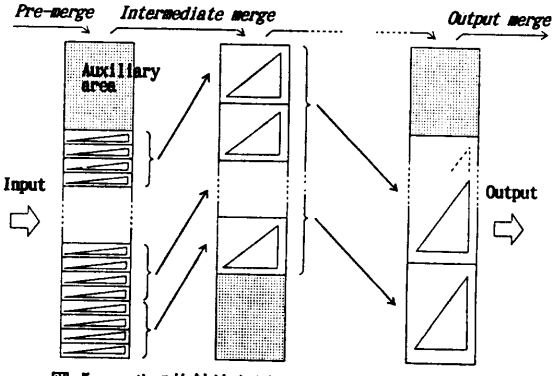

図5 エリア格納法を用いた逐次多段マージ処理
Fig. 5 Merge process using area store method.

ジ準備段階では，入力レコードをワークメモリの下位方向から，大きさ k のレコード列として順次格納していく．1つのレコード列を連続領域に格納することで，メモリ管理を容易にしている．マージ準備段階の終了時には，レコード列を格納した上方に補助領域がある．中間マージ段階では，補助領域が分断されないように奇数段と偶数段とで交互にワークメモリの上端と下端から連続的にレコード列を格納する．

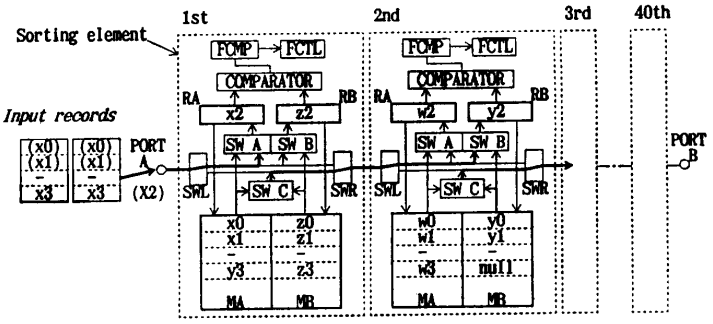

図 6 ソートアレイ LSI の基本構成図
Fig. 6 Architecture of the prototype sorting array.

補助領域は，k 本の入力レコード列とは別の領域に，出力レコード列を格納するために使用する．その大きさは，ワークメモリに格納するレコード列の最大の大きさに等しい．すなわち，最終の中間マージ段階の出力レコード列で最大なものの大きさに等しい．

補助領域の大きさを最小にする条件は，最後の中間マージ段階で生成する出力レコード列を大きさが均一な k 本にすることである．中間マージの最終段で k 本の大きさが均一なレコード列を生成するには，その1つ前のマージ段階で，大きさが均一な k^2 本のレコード列を生成すればよいから，結局，初段の中間マージ処理で k のべき乗本の大きさが等しいレコード列をワークメモリ上に生成することに帰着する．マージ準備段階の終了時点で，入力レコードの個数は確定しているから，初段の中間マージ処理で，最悪でも大きさが k だけ異なる k のべき乗本のレコード列を生成できる．一般に $N \gg k$ であること，レコード列の大きさの差 k を保ったままで中間マージが行えるように最初の中間マージを制御できることから，これ以上の均一化は不要といえる．

3.2 ソートアレイの構成法

（1） ソートアレイの基本構成

1次元アレイ構造のソートアレイは，比較エレメントの繰り返し構造であるから，複数のエレメントを大規模一括集積できる．40個の比較エレメントを一括集積した試作ソートアレイ[16),17)] の構成を**図 6** に示す．図は，4バイトのレコード（W から Z までの5レコード）を，バイト単位で比較転送して降順ソートする例である．レコード W, Y, Z は既にソートアレイに入力され，レコード X の第3バイト $x2$ を入力している．左端（1番目）のエレメントでは，入力データ $x2$ と自エレメント内のメモリ〔MB1〕から読み出した $z2$ とを比較する．同時に，2番目のエレメントでは，1番目のエレメントから転送される $y2$ と自エレメントの $w2$ とを比較する．レコードは，4回の比較転送動作で大小関係を判定し，比較結果に基づいてデータ転送路（SW A, SW B, SW C）を切り替える．ソートアレイの入力操作時と出力操作時の違いによるデータ転送方向の切替は，SW L と SW R とで行う．

（2） レコード長の拡張

ソートできるレコード長の上限は，基本的に，比較エレメントを構成するメモリユニットの容量に依存する．この問題を解決するために，各エレメントのメモリユニットの容量を大きくしたのでは，レコード長が短い場合にメモリを有効に利用できない．このため，メモリユニットの容量は変えずに，レコード長に応じて複数エレメントを連結して動作させる方法を採用した．本方法では，メモリユニットの容量を超えるレコードは，複数の連結されたエレメントに分割格納される．分割されたレコードは，それぞれを格納しているエレメントで独立に比較し，比較結果をエレメント間の制御回路で合成して，レコード全体の大小関係を決定する．エレメントを連結させた場合には，ソートアレイの実効エレメント数は，1/(連結数) になる．

エレメントの連結には，エレメント間で比較結果を合成する回路と，エレメント間にデータ転送をバイパスする回路が必要となる．この連結回路を複数のチップに跨って実現することは，チップの端子数増加と，連結回路による比較速度の低下要因となることから，エレメントの連結はチップ内に限定する．この方法は，ソートアレイをメモリユニットも含めて一括集積する際に，同一のハードウェア量でより多くのエレメントを実現できる方式といえる．

（3） 往復ソート機能

マージ準備段階の処理は，k 個のレコードを連続し

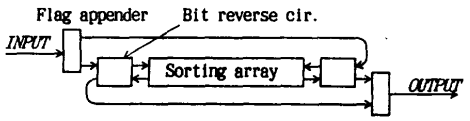

図 7 往復ソート機能を実現するハードウェア構成
Fig. 7 Block diagram for bi-directional sorting.

てソートアレイに入力した後に，k 個のレコードを連続に出力する操作の繰り返しである．このため，レコードの入力と出力が間欠的になる．この問題を解決して，マージ準備段階の処理時間を半減する往復ソート機能について示す．

往復ソートとは，ソートアレイ（図6参照）のポートA（左端)/ポートB（右端）からソート結果を出力する操作と，ポートB/ポートAに新たなレコードを入力する操作とを，交互にかつ重畳して行うことである．この機能を実現するには，ソートアレイ内でポートAから入力したレコードとポートBから入力したレコードとを分離する必要がある．このための従来技術として，入力レコードにポートA，Bのいずれから入力したかを示すタグを付与し，1次元アレイ状に配置した各比較回路で判定する手法がある[9]が，比較回路の制御が複雑になりスループットの低下が避けられなかった．このため，ソートアレイの入出力端に，往復ソート機能を実現するハードウェアを付加する構成とした．

上記機能を実現するハードウェアの構成を**図7**に示す．レコードは，その先頭位置（MSB 側）で制御フラグを付与した後に，ビット反転回路を介してソートアレイに入力する．ビット反転回路は，ソートアレイの入力/出力レコードを制御フラグを含めてビット反転する機能を持つ．制御フラグとビット反転回路によって，ポートAから入力したレコードを，ポートBから入力したレコードより，常にその値が大きくなるようにできる．その結果，両者のレコードは，ソートアレイ内で分離され，往復ソート機能を実現できる．往復ソート機能を実現するためのレコード修飾法を**表1**に示す．本方法では，往復ソート機能と同時に，昇順/降順のソート順を制御できる．このため，ソートアレイ内の比較転送機能をソート順によらず一定とでき，比較エレメントの制御回路を簡略にできる．また，レコードの先頭位置に付与したフラグと，前述したマージ制御のためのバンクタグの長さを同一とすることによって，マージ準備段階，中間マージ段階および出力マージ段階のすべての処理段階で，ソートアレイで扱うレコード長を同一にできる．

表 1 往復ソート機能を実現するレコード修飾法
Table 1 Record modification rules for bi-directional sorting.

ソート順	制御フラグ	ビット反転回路	
		ポートA	ポートB
昇 順	0	反 転	非反転
降 順	1	非反転	反 転

4. ソータの実現と性能評価

図6の試作ソートアレイチップは，1チップで 16 バイトまでのレコード 80 個を 3 M バイト/秒のスループットで並列に比較できる．本ソートアレイチップと，8 MB のワークメモリ，および汎用マイクロプロセッサからなるシングルボードソータを試作した．本試作ソータでは，500 k レコード（レコード長 15 バイト：1バイトは制御用に使用）を，3段マージによって約 13.4 秒でソートできる．試作ソータを用いて，3.2 節に示したレコード長の拡張と往復ソート機能とを確認した．さらに，エリア格納法を適用し，大容量ソータを実現した．

リレーショナルデータベースプロセッサ RINDA[14),15)]では，本アルゴリズムに基づくハードウェアソータを，マージ制御部を含めて専用 LSI で実現した．RINDA ハードウェアは，データベースの非定型検索処理，ソート処理，結合処理をオンザフライで高速に処理する装置で，チャネルインタフェースによってホスト計算機とディスク装置との間に接続される．RINDA に適用したソータの特徴を示す．

(1) マージ準備段階と出力マージ段階の処理は，本体装置とのデータ転送に追従して処理する．転送するデータは，ナルカラムや可変長タプルを含むデータベースの一時表であり，専用ハードウェアを用いてカラム抽出，コード変換等を行って固定長のレコードに変換する．このため，ソートするレコード長は，一般にタプル長より短くなる．ソータは，カラムの個数や属性を意識することなく，レコードの先頭から比較する．測定の結果，中間マージ段階の処理時間はデータ転送時間の1割以下[18)]であり，ほぼデータ転送時間でソートできた．

(2) 比較エレメントの連結機能により，ソートできるレコード長の上限を 250 バイト程度まで拡張した．また，大容量の DRAM チップを用いて 64 MB のワークメモリを高密度に実装した．

5. おわりに

データベース処理の高速化を狙いとした大容量ソータの基本アルゴリズムとその実現法を示した.

(1) データ駆動型マージ制御によるレコード列の選択と,並列比較を行うソートアレイとによって,数十ウェイで高速にマージできるマルチウェイマージ法を実現した.本マージ処理を繰り返すことで,大量のレコードをソートできる.

(2) レコードの比較回路とレコードの格納回路とを独立して高密度に実装できるため,要求条件に応じた最適構成のソータを小形に実現できる.

(3) 比較エレメントを連結してレコード長を拡張する.本方法を適用したハードウェアソータは,レコード長に対して柔軟な構成がとれる.

(4) ワークメモリを有効に利用して大容量ソートを実現するエリア格納法を実現した.数十ウェイのマージソータでは,メモリ利用率を十分に大きくできる.

謝辞 本研究を進めるにあたりご指導いただいたNTT 電子応用研究所川田忠通主席研究員,NTT 技術情報センタ篠岡信部長をはじめとして,熱心に議論していただいた研究グループの諸氏に感謝いたします.

参考文献

1) Knuth, D.E.: *The Art of Computer Programming, Vol. 3, Sorting and Searching*, Addison-Wesley (1973).
2) Todd, S.: Algorithm and Hardware for a Merge Sort Using Multiple Processors, *IBM J. Res. Dev.*, Vol. 22, No. 5, pp. 509-517 (Sept. 1978).
3) Tanaka, Y., Nozaka, Y. and Masuyama, A.: Pipelined Searching and Sorting Modules as Components of a Data Flow Database Computer, *Proc. of IFIP '80*, pp. 427-432 (1980).
4) 岩田和秀,神谷茂雄,酒井 浩,柴山茂樹,伊藤英則,村上国男:関係データベース処理エンジンのソータの試作と評価,情報処理学会論文誌, Vol. 28, No. 7, pp. 748-757 (1987).
5) 喜連川優,伏見信也,桑原和宏,田中英彦,元岡 達:パイプラインマージソータの構成,信学論, Vol. J66-D, No. 3, pp. 332-339 (1983).
6) 伏見信也,科野順蔵,鈴木 孝,笠原康則,太刀掛伸一,鍋田芳則,木村廣隆,沢井善彦,喜連川優,楊 維康:LSI ソートプロセッサ,信学技報, DE 88-2, pp. 9-17 (1988).
7) Kitsuregawa, M., Fushimi, S., Tanaka, H. and Moto-oka, T.: Memory Management Algorithms in Pipeline Merge Sorter, *Int. Work. on Database Machine*, Florida, pp. 208-232 (1985).
8) Kumar, M. and Mirschberg, D.S.: An Efficient Implementation of Batcher's Odd-Even Merge Algorithm and Its Application in Parallel Sorting Schemes, *IEEE Trans. Comput.*, Vol. C-32, No. 3, pp. 254-264 (Mar. 1983).
9) Miranker, G., Tang, L. and Wong, C.K.: A Zero-Time VLSI Sorter, *IBM J. Res. Dev.*, Vol. 27, No. 2, pp. 140-148 (Mar. 1983).
10) 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,信学論, Vol. J 65-D, No. 2, pp. 179-186 (1982).
11) 菊野 享,吉田典可,若林真一,石川裕次:VLSI のための高速ソーティング回路,信学技報, AL 81-50, pp. 23-30 (1981).
12) 佐藤哲司,津田伸生:一次元アレイを用いたソート処理装置の構成法,第29回情報処理学会全国大会論文集, 4F-1 (1984).
13) Satoh, T., Takeda, H. and Tsuda, N.: A Compact Multiway Merge Sorter Using VLSI Linear-array Comparators, *Int. Conf. on Foundation of Data Organization and Algorithms* (June 1989).
14) 速水治夫,井上 潮,福岡秀樹,鈴木健司:リレーショナルデータベースプロセッサ RINDA のアーキテクチャ,情処研報, Vol. 88, No. 79, 88-ARC-73-12, pp. 85-92 (1988).
15) 井上 潮,速水治夫,福岡秀樹,鈴木健司,松永俊雄:データベースプロセッサ RINDA の設計と実現,情報処理学会論文誌, Vol. 31, No. 3, pp. 373-379 (1990).
16) 佐藤哲司,津田伸生:階層化冗長構成による1次元アレイ論理 LSI の欠陥救済,信学技報, FTS 87-4, pp. 27-36 (1987).
17) Tsuda, N., Satoh, T. and Kawada, T.: A Pipeline Sorting Chip, *IEEE ISSCC Digest of Technical Papers*, pp. 270-271 (Feb. 1987).
18) 武田英昭,佐藤哲司,中村敏夫,速水治夫:関係演算高速化プロセッサ,情報処理学会論文誌, Vol. 31, No. 8, pp. 1230-1241 (1990).

(平成元年10月26日受付)
(平成2年9月11日採録)

佐藤 哲司 (正会員)

昭和32年生.昭和55年山梨大学工学部電子工学科卒業.同年,日本電信電話公社入社.主に論理回路の大規模集積技術,ハードウェアソータ,データベースマシンの研究に従事.現在,NTT 情報通信処理研究所主任研究員.電子情報通信学会, IEEE 各会員.

武田　英昭（正会員）

昭和30年生．昭和54年北海道大学工学部電気工学科卒業．昭和56年同大学院工学研究科電気工学専攻修士課程修了．同年，日本電信電話公社入社．以来，データベースマシンの研究開発に従事．現在，NTT 情報通信処理研究所主任研究員．IEEE 会員．

津田　伸生（正会員）

昭和 46 年金沢大学工学部電子卒業．昭和 49 年同大学院修士課程修了．同年電電公社武蔵野通研入所．以来，集積回路の設計・製造，ハードウェアアルゴリズム，文字認識装置，パターン情報処理の研究に従事．現在，NTT 情報通信処理研究所主幹研究員．電子情報通信学会，応用物理学会，IEEE 各会員．