ICOT Technical Report: TR-479

TR-479

可変長レコード用関係データベース処理エンジンの試作とソート処理性能の評価

伊藤　文英、島川　和典、
東郷　一生、松田　進(東芝)、
伊藤　英則(NTT)、大場　雅博(日立)

May, 1989

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

Vol. 30 No. 8 情報処理学会論文誌 Aug. 1989

可変長レコード用関係データベース処理エンジンの試作とソート処理性能の評価†

伊藤文英†† 島川和典††† 東郷一生††††
松田進†††† 伊藤英則††* 大場雅博†††††

ソーティングは計算機システムにおいて多用される基本的な操作であり，近年のVLSI技術の進歩により，ソータと呼ばれる専用ハードウェアの研究が活発に行われている．ソータの応用分野の1つにデータベース処理があり，データベース処理において可変長レコードの取扱いは不可欠である．そこで，筆者らは，関係データベースシステムに接続され，複数のフィールドからなる可変長レコード内の，任意の1つの固定長または可変長フィールドをキーとして，ソートを含む関係代数演算の実行を支援する関係データベース処理エンジンを試作した．エンジンはその基本構成要素として，12段のセルからなるパイプライン2ウェイマージソータをもつ．可変長レコードのソートにおいては，レコード長のばらつきにより，セルにおいて前のマージ処理の終了待ちが発生する．このため，総データ量が同じである固定長レコードソートよりも，一般に処理時間が長くなる．本エンジンでは，レコードからキーフィールドを切り出してソータに入力するキー切出し方式を採用し，前のマージ処理の終了待ちを低減した．また，可変長キーのマージ処理に必要な制御情報をタグとして付加することにより制御を簡単化し，処理速度を低下させることなくハードウェア量の削減を実現した．試作したエンジンのソート処理レートは2.67メガバイト/秒であり，ソート処理性能を，解析と実測により評価した．

1. まえがき

ソーティングは計算機システムにおいて多用される基本的な操作であり，古くから数多くの研究が行われてきた[1]．近年は，VLSI技術の進歩により，ソータと呼ばれる専用ハードウェアの提案，試作が活発に行われている．ソータの応用分野としては種々のものが考えられるが，その1つにデータベース処理がある．高度情報化社会の進展にともない，データベースの大容量，高速化の要求はますます高まっており，ソータ等の専用ハードウェアによるデータベース処理の高速化の研究も盛んに行われている[2)~5)]．

データベース処理において，可変長レコードの取扱いは不可欠である．しかし，これまでに試作されたソータは固定長レコードを処理するもののみであり，また，提案されている可変長レコードを処理するソータにおいても，データベースにおけるレコードの形式は十分に検討されていない．そこで，筆者らは，データベース処理において使われる形式の可変長レコードを対象として，ソートを含む関係代数演算の実行を支援する関係データベース処理エンジンを開発した[6]．

これまでに提案されている主なソータには，n個のレコードをn個のセルでソートするものとして磁気バブルメモリソータ[7]，バイトニックソータ[8]，並列計数ソータ[9]等，$\log n$個のセルでソートするものとしてパイプラインマージソータ[10]，パイプラインヒープソータ[11]，シストリックソータ[12]等がある．これらのソータを現状の技術を用いて実現するという点で比較すると，$\log n$個のセルを用いるソータの方が，大量データを処理する場合にハードウェアが小型化できるので有利である．このうちパイプラインマージソータは，若干の出力遅れ時間があることおよびメモリの使用効率が悪いことの欠点をもつが，ソータの容量を単位とした連続的なパイプライン処理が可能で制御が簡単なため実用化に適しており，固定長レコード用ソータの試作[13),14)]が報告されている．

可変長レコード用パイプラインマージソータについては，メモリ使用効率を最適化するアルゴリズム[15]とそれを実現するアーキテクチャ[16]が提案されている．しかし，これらの研究では，レコード全体をキーとしたソートを対象としており，データベース処理におけるレコード形式の考慮を含めた処理性能の検討は行わ

† Implementation of a Relational Database Engine for Variable Length Records and Performance Evaluation of Sorting by FUMIHIDE ITOH (Research Center, Institute for New Generation Computer Technology), KAZUNORI SHIMAKAWA (Information and Communication Systems Laboratory, Toshiba Corporatin), KAZUO TOGO, SUSUMU MATSUDA (Ome Works, Toshiba Corporation), HIDENORI ITOH (Research Center, Institute for New Generation Computer Technology) and MASAHIRO OBA (Systems Development Laboratory, Hitachi, Ltd.).

†† (財)新世代コンピュータ技術開発機構研究所
††† (株)東芝情報通信システム技術研究所
†††† (株)東芝青梅工場
††††† (株)日立製作所システム開発研究所
* 現在 日本電信電話(株)情報通信処理研究所

れていない．すなわち，一般にデータベースにおいてレコードは複数のフィールドからなり，ソートはそれらのフィールドをキーとして行われる．そこで，本稿では，データベース処理への応用を想定し，可変長レコード内の1つのフィールド（固定長または可変長）をキーとして処理する関係データベース処理エンジンの設計，試作，およびそのソート処理性能の評価について報告する．

2. ソート処理方式

本章では，ソートアルゴリズムの概要，可変長レコードをソートする場合の問題点と本エンジンでの解決策について述べる．

2.1 ソートアルゴリズムの概要

2つのソートされたレコード列（以下，ストリングと呼ぶ）のマージ処理を繰り返すことによりソート処理を行うことを，2ウェイマージソートという．パイプライン2ウェイマージソートでは，1次元に配置したセルにより，マージ処理を連続的に行う．i 段目のセルは前段セルからの 2^{i-1} 個のレコードからなる2つのストリングをマージし，2^i 個のレコードからなるストリングを次段セルに出力する動作を繰り返す．

可変長レコードに対するソート処理の様子を図1に示す．ただし，セルの処理レート（1語あたりの処理時間）を a，セル内をデータが流れるのに要する時間を b，最初のセルへの j 番目の入力レコードの語長を l_j とする．各セルにおいて，最初のマージ処理は2番目の入力ストリングの先頭データが到着したとき開始される．2番目以降のマージ処理は，偶数番目の入力ストリングの先頭データが到着したとき，前のマージ処理が終了している場合はただちに，未終了の場合はその終了を待って，開始される．

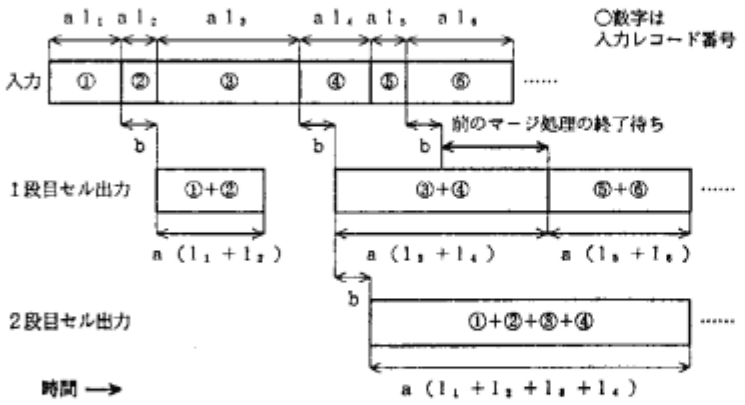


図1　可変長レコードソートの様子
Fig. 1　Overview of sorting variable length records.

固定長レコードソートにおいては，2番目以降のマージ処理のための偶数番目の入力ストリングが到着する以前に，その前のマージ処理が終了する．特に，入力ストリングが連続して流れる場合，偶数番目の入力ストリングが到着すると同時に，その前のマージ処理が終了する．このため，固定長レコードソートにおいては，前のマージ処理の終了待ちは発生せず，さらに，入力ストリングが連続であれば出力ストリングも連続である．しかし，可変長レコードソートにおいては，次の2つの場合が起こりうる．

（1）　偶数番目の入力ストリングが到着したときにその前のマージ処理が終了していない場合，前のマージ処理の終了待ちが発生する．

（2）　偶数番目の入力ストリングが到着する前にその前のマージ処理が終了する場合，入力ストリングが連続であっても，出力ストリング間にデータの流れない間隔が発生する．

次に，2^n 個のレコードを n 段のセルによりソートする場合を考える．各段のセルにおける最後のマージ処理がその前のマージ処理の終了を待たないとすると，各セルは，最後の入力ストリングの入力開始から時間 b 後に，最後の出力ストリングの出力を開始する．最終段（n 段目）のセルにおけるマージ処理は1回であり，その出力ストリングがソート結果となる．すなわち，各段のセルにおける最後のマージ処理がその前のマージ処理の終了を待たない場合，最後の入力レコードの1段目のセルへの入力開始から時間 nb 後に，最終段のセルからのソート結果の出力が開始される．しかし，各段のセルにおける最後のマージ処理がその前のマージ処理の終了を待つ場合，その待ち時間はそのまま最終段のセルからのソート結果の出力開始の遅れとなり，そのぶんソート処理時間が増大する．よって，もし前のマージ処理の終了待ちを抑制できれば，可変長レコードソートの処理時間を向上できる．

2.2 前のマージ処理の終了待ち

j 番目のマージ処理の開始が前のマージ処理の終了を待たない場合と待つ場合のセルにおけるストリング入出力の様子を図2に示す．ただし，j 番

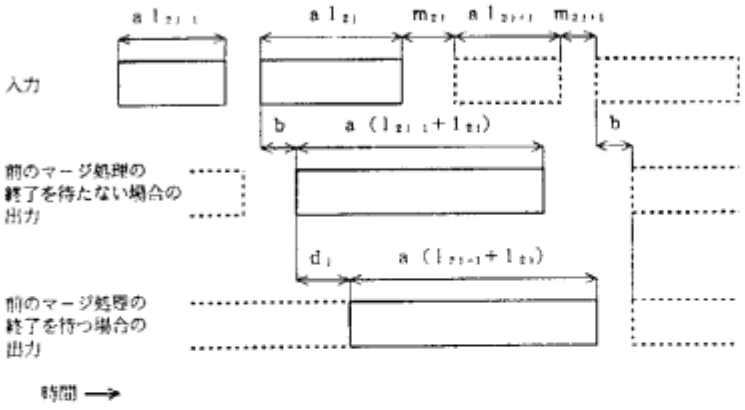


図 2 前のマージ処理の終了待ち
Fig. 2 Waiting of the previous merge processing.

目のマージ処理の開始が $(j-1)$ 番目のマージ処理の終了を待つときの待ち時間を d_j, k 番目の入力ストリングと $(k+1)$ 番目の入力ストリングの間のデータの流れない時間を m_k とする．j 番目のマージ処理の開始が $(j-1)$ 番目のマージ処理の終了を待たない場合，$(j+1)$ 番目のマージ処理の開始が j 番目のマージ処理の終了を待たないための条件は，次のようになる．

$$a(l_{2j+1}-l_{2j-1})+m_{2j}+m_{2j+1}\geqq 0 \quad (j\geqq 1) \qquad (1)$$

すなわち，$(2j+1)$ 番目の入力ストリングが $(2j-1)$ 番目の入力ストリングより短く，その差に対する処理時間が $2j$ 番目と $(2j+2)$ 番目の入力ストリングの間のデータの流れない時間の和より長いとき，待ちが発生する．一方，j 番目のマージ処理の開始が $(j-1)$ 番目のマージ処理の終了を待つ場合，$(j+1)$ 番目のマージ処理の開始が j 番目のマージ処理の終了を待たないための条件は，次のようになる．

$$a(l_{2j+1}-l_{2j-1})+m_{2j}+m_{2j+1}-d_j\geqq 0 \quad (j\geqq 2) \qquad (2)$$

この場合，j 番目のマージ処理の終了待ち時間のぶんだけ条件が厳しくなり，$(2j+1)$ 番目の入力ストリングが $(2j-1)$ 番目の入力ストリングより長いときでも，待ちが発生することがある．なお，(1)式は(2)式で $d_j=0$ の場合に相当するが，以降の考察のために別式とする．また，(1)式および(2)式の左辺は，正の場合，j 番目の出力ストリングと $(j+1)$ 番目の出力ストリングの間のデータの流れない時間を，負の場合，その絶対値は $(j+1)$ 番目のマージ処理の開始待ち時間 d_{j+1} をあらわす．

次に，いくつかのストリング長分布について考察する．

(1) 入力ストリング長が一定または単調に増加する場合：$l_{2j+1}\geqq l_{2j-1}$, $m_{2j}\geqq 0$, $m_{2j+1}\geqq 0$ であるから，(1)式は常に成立する．よって，常に(1)式が適用され，待ちは全く発生しない．また，出力ストリング長も一定または単調増加となるので，次段以降のセルにおいても待ちは発生しない．なお，固定長レコードソートはこの場合に相当する．

(2) 入力ストリングが連続して流れ，入力ストリング長が単調に減少する場合：$l_{2j+1}\leqq l_{2j-1}$, $m_{2j}=m_{2j+1}=0$, $d_j>0$ であるから，最初に $l_{2j+1}<l_{2j-1}$ となったところで(1)式は成立せず，待ちが発生する．以降，適用される(2)式は常に成立しないので，常に待ちが発生する．また，出力ストリングも連続で，ストリング長が単調に減少するので，次段以降のセルにおいても同様の動作をする．

(3) 入力ストリング間のデータの流れない時間が一定で，入力ストリング長が等差数列的に減少する場合：等差数列の公差の絶対値を c，データの流れない時間を m とすると，$l_{2j+1}-l_{2j-1}=-2c$, $m_{2j}=m_{2j+1}=m$ であるから，(1)および(2)式は次のようになる．

$$-2ac+2m\geqq 0 \qquad (3)$$

$$-2ac+2m-d_j\geqq 0 \quad (j\geqq 2) \qquad (4)$$

すなわち，$ac>m$ のときは，(3)，(4)式は成立せず，待ちが発生する．このとき，出力ストリングは連続となるので，次段以降のセルでは(2)の場合となり，待ちが発生する．$ac\leqq m$ のときは，(3)式が成立するので，待ちは発生しない．しかし，$ac=m$ のときは，出力ストリングが連続となるので，次段以降のセルで待ちが発生する．また，$ac<m$ のときは，出力ストリングも，データの流れない時間が一定の等差数列となる．しかし，公差の絶対値が $4c$ と4倍になるのに対し，データの流れない時間は $2(m-ac)$ と2倍未満であるので，次段以降のセルでは待ちの発生する可能性が高くなる．

(4) 入力ストリングの間隔が一定で，ストリング長が等差数列的に減少する（すなわち，入力ストリング間のデータの流れない時間が，入力ストリング長の公差に対する処理時間ずつ等差数列的に増加する）場合：等差数列の公差の絶対値を c，最初の入力ストリングと2番目の入力ストリングの間のデータの流れな

い時間 m_1 を m とすると，$l_{2j+1}-l_{2j-1}=-2c$，$m_{2j}=m+(2j-1)ac$，$m_{2j+1}=m+2jac$ であるから，(1)式は次のようになる．

$$2m+(4j-3)ac\geqq 0 \qquad (5)$$

(5)式は常に成り立つので，常に適用され，待ちは発生しない．また，出力ストリングも間隔が一定で，ストリング長が等差数列的に減少するので，次段以降のセルにおいても待ちは発生しない．

2.3 キー切出し方式

複数のフィールドからなるレコードをあるフィールドをキーとしてソートする場合，ソータにレコード全体を入力する方式（以下，レコード入力方式と呼ぶ）と，キーのみを入力する方式（以下，キー切出し方式と呼ぶ）が考えられる．

2つの方式を比較したとき，キー切出し方式では，1回のマージ処理で処理するデータ量がより少なく，キー以外のデータのぶんだけのデータの流れない時間が発生する．このため，前のマージ処理の終了待ちの発生する可能性がより低く，また，待ちが発生した場合でも，その待ち時間がより少ない．このことが，各段のセルにおける最後のマージ処理についていえるので，キー切出し処理方式のほうが，レコード入力方式よりソート処理時間が短くなることがある．

キー切出し方式の場合，キーとそれが切り出されたレコードをなんらかの方法で対応づける必要がある．そこで，レコード全体をソータとは別のメモリに格納し，ソータの入力側でレコードから切り出したキーの後にレコード識別番号を付加し，出力側でレコード識別番号をもとにレコードをメモリから読み出す方式を採用した．この方式では，キーのほかにレコード識別番号もソータを流れるが，一般に，レコード識別番号長はレコードにしめるキー以外のデータの長さより短いため，キー切出し方式は依然として有利である．なお，キー切出し方式には，セルのメモリ量を低減できる利点もある．

2.4 メモリ管理

セルは2つのストリングのマージ処理において，奇数番目のストリングをいったんメモリに格納する必要がある．また，マージ処理の開始が待たされるとき，あるいはマージの結果奇数番目のストリングのデータが1個でも出力されたあとは，偶数番目のストリングのデータもメモリに格納する必要がある．可変長レコードソートの場合，メモリ内に長さの異なる複数のストリングが同時に存在することがある．よって，メモリ管理では各ストリングを区別し，しかもストリング内のソートされたレコードの順序を乱さない制御が必要である．

セルは，2つのストリングのマージ処理を開始すると結果ストリングを連続して出力するため，メモリに格納されるデータの最大量は，そのセルが処理する奇数番目の入力ストリングの最大長となる．しかし，メモリ管理を実現するために，格納されるデータの最大量より大きい容量のメモリが必要となる．

2ウェイマージソートのメモリ管理方式には，連接リスト方式，ブロック分割方式，2重メモリ方式がある[16]．このうち，2重メモリ方式は，メモリを2つのエリアに分け，奇数番目のストリングと偶数番目のストリングを別のエリアに格納する．メモリは格納されるデータの最大量の2倍必要となるが，制御は最も簡単である．本エンジンでは，キー切出し方式によりソータを流れるデータ量が低減されていること，および制御の容易性から，2重メモリ方式を採用した．

2.5 制御情報

セルは2つのストリングのマージ処理において，キーの末尾，レコードの境界，ストリングの境界，ストリームの終了等を検出する必要がある．固定長レコードの場合，キー長，レコード長，ストリングのレコード数等を保持するレジスタとカウンタによる制御が可能である[13]．しかし，可変長レコードの場合，レコードごとにキー長やレコード長が異なるため，レジスタとカウンタによる制御は困難である．

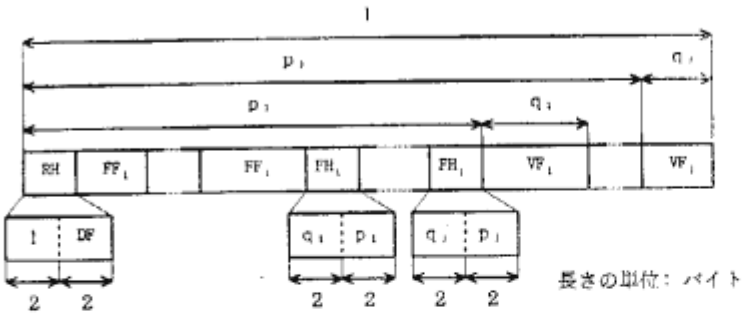
そこで，データの各語にキーの末尾，レコードの末尾，ストリングの末尾，およびストリームの終了を示す各1ビットのタグを付加し，これらのタグによりマージ処理を制御する方式を採用した．本方式は，タグをデータと一緒にメモリに格納するため，セルが必要とするメモリ容量が増大する欠点をもつ．しかし，タグとデータは同時に処理されるため，タグの付加による処理速度の低下はない．また，カウンタを必要としないため，セルのハードウェア量を削減できる．

3. 関係データベースエンジンの機能

本章では，エンジンが接続される関係データベースシステムのデータ形式，エンジンと汎用計算機の間の機能分担，およびエンジンにおけるソータの役割について述べる．

3.1 レコード形式

レコードの形式を図3に示す．1レコードは関係の



DF：論理削除フラグ，FF：固定長フィールド，FH：可変長フィールドヘッダ，RH：レコードヘッダ，VF：可変長フィールド，l：レコード長，p：可変長フィールド位置，q：可変長フィールド長

図 3 レコード形式
Fig. 3 Record format.

1組に相当する．レコードは，先頭から順にレコードヘッダ，固定長フィールド群，可変長フィールドヘッダ群，可変長フィールド群からなる．レコードヘッダにはレコード長と論理削除フラグが入る．固定長フィールドと可変長フィールドには，それぞれ固定長または可変長と定義された属性の具体値が入る．可変長フィールドヘッダは可変長フィールドと前から順に対応し，可変長フィールドの長さと位置（レコード先頭からフィールド先頭までの長さ）が入る．

エンジンにおける演算処理では，任意の1つの固定長または可変長フィールドがキーとして指定される．1つの関係の各レコードにおいて，対応する固定長フィールドおよび可変長フィールドヘッダのレコードの先頭からの位置は一定である．これは，ハードウェアによるキー切出し処理の実現に適する．固定長フィールドの位置と長さ，および可変長フィールドヘッダの位置は，関係のスキーマの一部として管理される．

3.2 エンジンと汎用計算機の機能分担

本エンジンは，汎用計算機の付加プロセッサとして，関係データベース演算を支援する．エンジンはすべての演算を実行するのではなく，ハードウェア向きでない複雑な処理や，エンジンとの間のデータ転送回数が増大する処理は，汎用計算機側で実行する．エンジンが実行する演算処理は次のとおりである．

（1） ソート演算およびユニーク演算：入力レコードをキー値の昇順または降順にソートする．ユニーク演算の場合，さらに，キー値の重複を取り除く．入力ストリームがエンジンのソート容量をこえる場合，ソート容量以下のサブストリームに分割し，サブストリームごとのソートおよび重複除去を繰り返す．ただし，ソートされたサブストリーム間のマージ処理は，汎用計算機で行う．

（2） 選択演算：ある定数との大小関係を条件として，入力ストリームのレコードから，キー値が条件を満たすものを選択する．

（3） 共通部分集合演算および差集合演算：入力ストリーム1のキー値の集合を条件として，入力ストリーム2のレコードから，キー値が条件に含まれるもの（共通部分集合演算の場合），あるいは含まれないもの（差集合演算の場合）を選択する．

（4） 等結合演算：入力ストリーム1と入力ストリーム2の間で，等しいキー値をもつレコード同士をそのまま組み合わせる．ただし，組み合わされた2つのレコードを1つのレコードに再構成する処理は，汎用計算機で行う．

なお，キーのデータタイプは，文字列（可変長，固定長），および，絶対値数，整数，正規化された IBM 形式の浮動小数点数（以上固定長）である．

3.3 ソータの役割

エンジンが実行する演算処理のうち，ソート演算とユニーク演算では，ソート処理機能がそのまま使われる．このほか，共通部分集合，差集合，および等結合の各演算では，前処理として入力ストリームをソートすることにより，演算を効率的に実行できる．例えば，共通部分集合演算は，2つの入力ストリームがそのキー値で昇順にソートされている場合，次のようなマージアルゴリズムをどちらかの入力ストリームが尽きるまで繰り返すことにより実現される．ただし，R_1，R_2 はそれぞれ入力ストリーム1，入力ストリーム2のその時点での先頭レコード，k_1，k_2 はそれぞれ R_1，R_2 のキー値とする．

（1） $k_1=k_2$ のとき，R_2 を出力する．

（2） $k_1<k_2$ のとき，R_1 を捨てる．

（3） $k_1>k_2$ のとき，R_2 を捨てる．

なお，本エンジンでは，キー値の重複を考慮した，さらに改良したアルゴリズムを採用している．

4. 試　　作

本章では試作システム，エンジン，ソータの構成と機能，およびエンジン内部での処理データの形式と処理レートについて述べる．

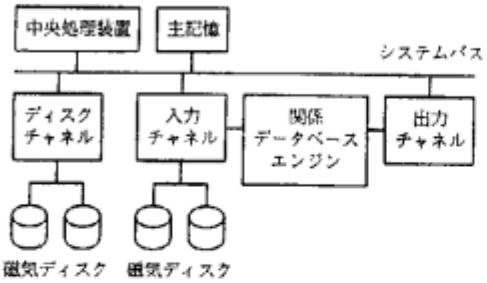


図 4　システム構成
Fig. 4　System configuration.

4.1　試作システムの構成

試作システムの構成を図4に示す．エンジンは入力用と出力用の2つのチャネルにより汎用計算機のシステムバスに接続され，データの同時入出力を可能とした．入力用チャネルには関係データ用磁気ディスクを接続し，磁気ディスクからエンジンへの直接データ入力を可能とした．

エンジンの起動は，汎用計算機のオペレーティングシステムに組み込んだドライバにより，チャネルコントロールコマンドで行う．入力用および出力用チャネルは，ディスクチャネルのハードウェアおよびファームウェアを改造した．

4.2　エンジンの構成と処理データの形式

エンジンの構成とエンジン内部でのデータの形式を図5に示す．エンジンは前処理部，レコードバッファ，ソータ，関係代数演算部，およびコントローラの各モジュールで構成される．それぞれの機能は次のとおりである．

（1）　前処理部：レコードからキーフィールドを切り出し，キーの後に2バイトのレコード識別番号を付加する（このキーとレコード識別番号の組をエレメントと呼ぶ）．キーのデータタイプが整数または浮動小数点数の場合，絶対値数に変換する．

（2）　レコードバッファ：入力レコード全体を1メガバイトのメモリに格納する．レコード格納番地とレコード識別番号の対応を，64キロ個のエントリをもつテーブルで管理する．関係代数演算部から送られるレコード識別番号に従って，出力レコードを読み出す．

（3）　ソータ：12段のセルにより，キーを最大4,096個ごとに連続してソートする．昇順および降順のソートが可能である．

（4）　関係代数演算部：関係代数演算の実行およびレコード識別番号の切出しを行う．また，切り出されたレコード識別番号のレコードバッファへの供給を緩衝するために，先入れ先出しメモリをもつ．

（5）　コントローラ：他のモジュールを初期化し，演算実行を制御する．前処理部に対しては，入力レコード数，キーが固定長か可変長か，キーフィールドの位置と長さ（固定長キーの場合），キーフィールドに対応するフィールドヘッダの位置（可変長キーの場合），およびキーのデータタイプを設定する．ソータに対しては，昇順ソートか降順ソートか，およびどのセルを使用するかを設定する．すなわち，入力レコード数が2,048以下の場合，あるいはメモリ容量の不足するセルがある場合，より後段のセルのみを使うように制御する．関係代数演算部に対しては，演算の種別を設定する．

エンジン内を流れるデータの形式には，次の3つがある．

（1）　レコード：前処理部およびレコードバッファへの入力，およびレコードバッファからの出力は，エンジンの入出力と同じレコードである．データは2バイトずつ流れる．

（2）　タグ付きエレメント：前処理部から関係代数演算部までは，キーとレコード識別番号，およびそれら2バイトにつき1バイトのタグである．データ2バイトとタグ1バイトが同時に流れる．

（3）　タグ付きレコード識別番号：関係代数演算部とレコードバッファの間は，レコード識別番号，および各レコード識別番号につき1バイトのタグである．レコード識別番号2バイトとタグ1バイトが同時に流れる．

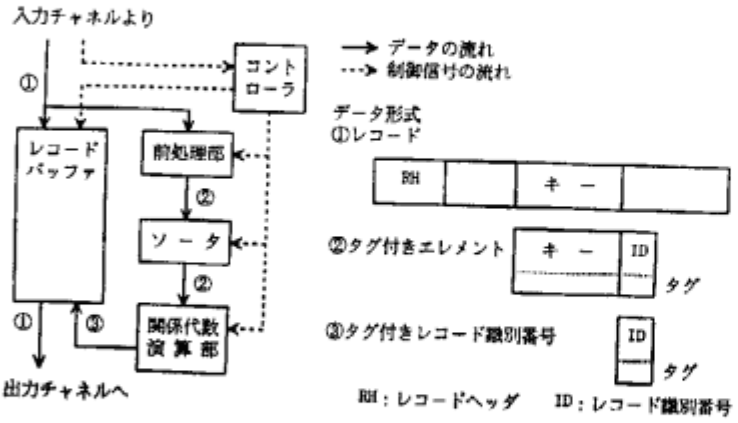


図 5　エンジンの構成とエンジン内のデータ形式
Fig. 5　Configuration and internal data format of the engine.

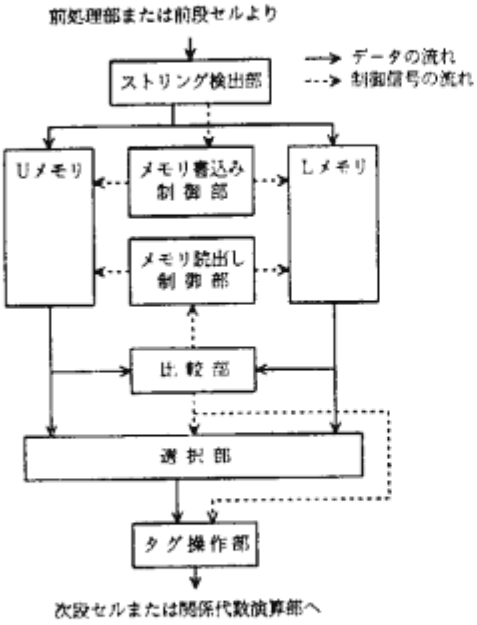


図 6 セルの構成
Fig. 6 Configuration of a cell.

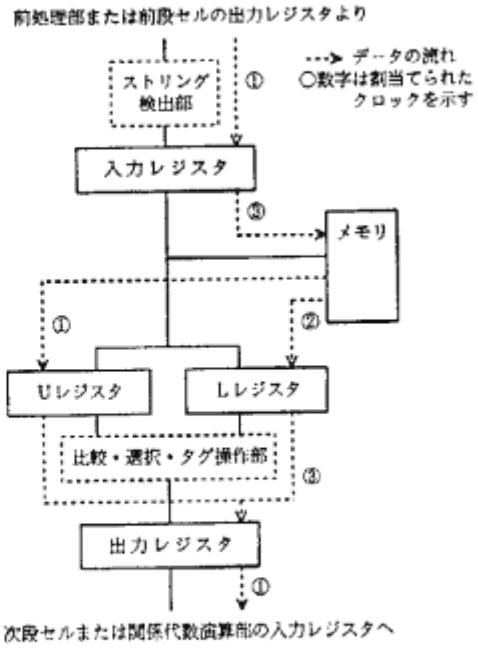


図 7 セル内のデータの流れ
Fig. 7 Data flow timing in a cell.

タグの内容は，キーの末尾，レコードの末尾，ストリングの末尾，ストリームの終了，キー値の重複を示す各1ビットの制御情報，および3ビットのパリティ検査ビットである．

4.3 セルの構成

ソータは12段のセルで構成される．1つのセルの論理的構成を図6に示す．マージされる2つのストリングは，UメモリとLメモリに格納される．メモリの容量は，1〜11段目がUメモリ，Lメモリ各48キロバイト（タグ用16キロバイトを含む），12段目がUメモリ，Lメモリ各96キロバイト（タグ用32キロバイトを含む）である．ただし，1〜11段目のUメモリとLメモリは，物理的には1つのものを分けて使っている．ストリング検出部はタグによりストリングの境界を検出し，メモリ書き込み制御部は入力ストリングを交互に2つのメモリに振り分ける．メモリ読み出し制御部は，キー値の大小決定前は比較すべき2つのデータ，決定後は出力すべき一方のデータの読み出しを制御する．比較部はデータとタグにより，キー値の大小の決定，および次の比較の開始を制御する．選択部は出力すべきデータを選択する．タグ操作部は出力ストリングの境界，およびキー値の重複をタグに設定する．

4.4 処理レート

セルにおけるデータ経路の物理的構成と，データの流れるタイミングを図7に示す．比較処理を行うハードウェアはメモリアクセスより高速であるから，セルの処理レートはメモリアクセスの回数により定められる．アクセスが最も多くなるのは，メモリ内の2つのストリングのマージ操作を行いつつ，前段セルからのストリングをメモリに格納している場合で，メモリ内の2つのストリングから各々1語のデータを読み出して比較器へ供給するために2回，前段セルからのストリングの1語をメモリに書き込むために1回の計3回のメモリアクセスを連続して行う必要がある．制御の容易性から，各クロック（クロックはメモリアクセス時間の単位）にこれら3種類のメモリアクセスを（固定的に）順に割り当てたので，セルの処理レートは1語/3クロックとなった．また，比較処理などを含めて，セルをデータが流れるのに要する時間は，6クロックとなる．なお，UメモリとLメモリを別の物理メモリとすれば，2回のデータ読み出しを同時に行い，処理レートを1語/2クロックにすることが可能である．しかし，本エンジンでは，実装上の制約，エンジンと外部との間のデータ転送レート，および一部の関係代数演算実行時の処理レートを考慮し，採用しなかった．

一方，レコードバッファのデータ入出力は，エンジン外部との転送レート，ソータの処理レート，およびデータの同時入出力の必要性を考慮し，入出力それぞれ原則として1語/2クロックとした．すなわち，レコード

バッファへの入力は前処理部への入力と同期しているので，キー部の入力は1語/3クロックとなる．また，レコードバッファからの出力は，メモリの書き込みと読み出しを交互のクロックで行うことにより，入力と同時に実行できる．ただし，メモリの書き込みと読み出しが全く同じクロックになるような場合は，書き込みが優先される．

なお，1語は2バイト，1クロックは250ナノ秒である．

5. 性能評価

本章では，レジスタトランスファレベルの解析と実測により，エンジンのソート処理性能を評価する．

5.1 固定長レコードソート

レコード長，キー長とも一定の場合のソート処理性能を評価する．ただし，レコード数を n（n は4,096以下の2の冪乗），レコード長を $2r$ バイト（r 語），キー長を $2k$ バイト（k 語）とし，キーがレコードの末尾にあるとする．

はじめに，レジスタ転送レベルの解析により，計算値を求める．レコード数が4のときのデータの流れる様子を図8に示す．各モジュールの動作概要は次のとおりである．なお，レコードバッファへの入力は，前処理部への入力と同期している．

（1） 前処理部：各レコードに対して，キーフィールドに達するまでのデータを2バイト/2クロックで読み飛ばし，キーとレコード識別番号を2バイト（タグを含めて3バイト）/3クロックでソータに出力する．レコード識別番号の出力と次のレコードのデータの読み飛ばしは同時に行う．よって，レコードの入力からエレメントの出力までの時間 d_{p0}，およびエレメント出力の間隔 i_p のクロック数は次のようになる．

$$d_{p0}=2(r-k)+4 \qquad (6)$$

$$i_p=3k+2(r-k)\pm1 \qquad (7)$$

ここで，(6)式の右辺の第1項はキー以外を読み飛ばす時間，第2項は前処理部をデータが流れるのに要する時間の平均を示す．また，(7)式の右辺の第1項はキーを出力する時間，第2項はキー以外を読み飛ばす時間を示す．ただし，ソータへの出力は3クロックごとの特定のクロックが割り当てられているため，i_p は常に3の倍数となる．

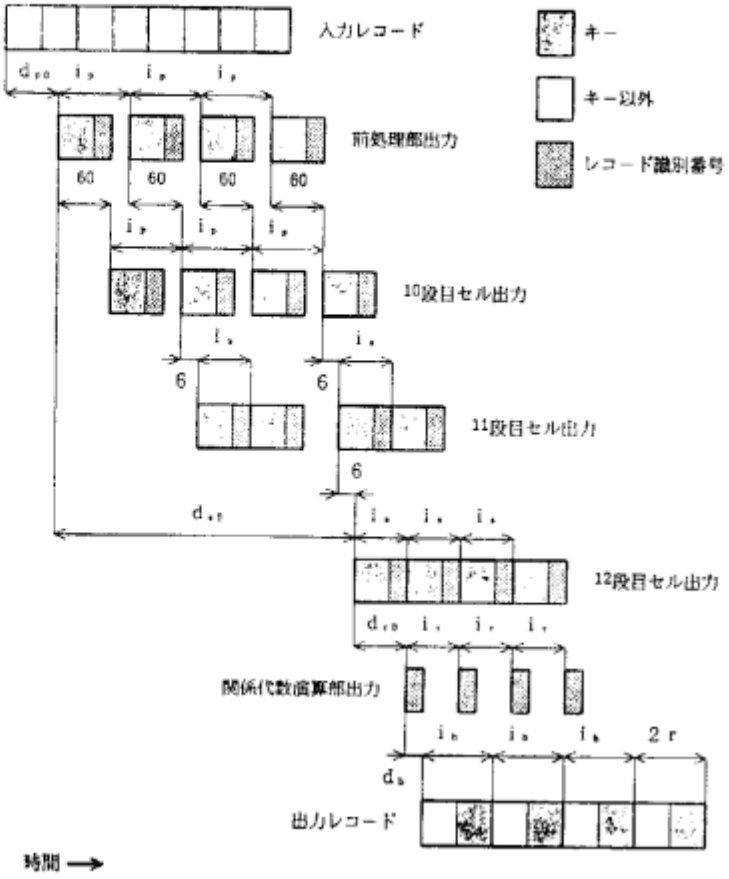


図8 エンジンにおけるデータの流れ
Fig. 8 Data flow in the engine.

（2） ソータ：レコード数が4のときは，11段目と12段目のセルのみがソートに使用され，1～10段目のセルはデータがそのまま流れる．11段目のセルは，2番目のエレメントが入力されたとき最初のエレメントとの比較を開始し，ソートされた2個のエレメントを連続して出力する．また，最後のエレメントが入力されたとき，(初めの2個のエレメントの出力は終了しているので) 3番目のエレメントとの比較を開始し，ソートされた2個のエレメントを連続して出力する．12段目のセルは，11段目からの2番目のストリングの最初のエレメントが入力されたとき最初のストリングとのマージ処理を開始し，ソートされた4個のエレメントを連続して出力する．固定長レコードソートにおいては前のマージ処理の終了待ちは発生しないので，12段目のセルからの出力開始は，1段目のセルへの最後のエレメントの入力から，ソータをデータが流れるのに要する時間後となる．このこと

は，一般にレコード数が n（ただし，n は 4,096 以下の2の冪乗）のときに成り立つ．よって，1段目のセルへの最初のエレメントの入力から 12 段目のセルからの最初のエレメントの出力までの時間 d_{s0}，およびエレメント出力の間隔 i_s のクロック数は次のようになる．

$$d_{s0}=i_p(n-1)+72 \quad (8)$$

$$i_s=3(k+1) \quad (9)$$

ここで，(8)式の右辺の第1項は最後のエレメントが入力されるまでの時間，第2項はソータをデータが流れるのに要する時間（セル1段あたり6）を示す．また，(9)式の右辺は，キーとレコード識別番号が2バイト/3クロックで流れるのに要する時間を示す．

（3） 関係代数演算部：各エレメントのキーを読み飛ばし，レコード識別番号をレコードバッファに出力する．よって，エレメントの入力からレコード識別番号の出力までの時間 d_{r0}，およびレコード識別番号出力の間隔 i_r のクロック数は次のようになる．

$$d_{r0}=3k+12 \quad (10)$$

$$i_r=i_s=3(k+1) \quad (11)$$

ここで，(10)式の右辺の第1項はキーを読み飛ばす時間，第2項は関係代数演算部をデータが流れるのに要する時間を示す．

（4） レコードバッファからの出力：関係代数演算部から送られるレコード識別番号に対応するレコードを，2バイト/2クロックで出力する．このとき，レコードバッファへのすべてのレコードの入力は終了しているので，メモリ書き込みによる読み出しの待ち合わせはない．レコード出力の間隔は，1個のレコードの出力に要する時間と，レコード識別番号が入力される間隔の大きいほうとなる．よって，レコード識別番号の入力からレコードの出力までの時間 d_b，およびレコード出力の間隔 i_b のクロック数は次のようになる．

$$d_b=4 \quad (12)$$

$$i_b=\max[2r,\ i_r] \quad (13)$$

以上より，全処理時間のクロック数 u は次のようになる．

$$u=d_{p0}+d_{s0}+d_{r0}+d_b+i_b(n-1)+2r \quad (14)$$

ここで，右辺の第1～4項は，それぞれ前処理部，ソータ，関係代数演算部，およびレコードバッファにおけるデータの入力開始から出力開始までの時間，第5項は最初のレコードの出力開始から最後のレコードの出力開始までの時間，第6項は最後のレコードの出力に要する時間を示す．また，(14)式の右辺を計算すると次のようになる．

$$u=\{i_p(n-1)+2r+k\}+\{i_b(n-1)+2r\}+92 \quad (15)$$

すなわち，全処理時間は，レコードの入力および出力時間（第1，2項）と，データがエンジンを流れるのに要する時間（第3項）の和となる．

5.2 可変長レコードソート

レコードが可変長キーフィールドのみからなり，入力レコードのキー長が等差数列的に変化する場合のソート処理性能を評価する．ただし，レコード数を n（n は 4,096 以下の2の冪乗），j 番目の入力レコードのキー長を $2k_j$ バイト（k_j 語），キー長が一定または増加する場合の等差数列の公差を $4e$ バイト（$2e$ 語），キー長が減少する場合の等差数列の公差の絶対値を $4f$ バイト（$2f$ 語），ソート後の j 番目の出力レコードのキー長を h_j とする．さらに，キー長は最低でも10バイトとし，$4f \geqq 4$ とする．

キー長の平均を $2k$ バイト（k 語）とすると，次の関係が成立する．

$$\sum_{j=1}^{n} k_j=\sum_{j=1}^{n} h_j-kn \quad (16)$$

また，キー長が一定または増加する場合，次の関係が成り立つ．

$$k_j=k_1+2e(j-1) \quad (17)$$

$$k_1=k-e(n-1) \quad (18)$$

一方，キー長が減少する場合，次の関係が成り立つ．

$$k_j=k_1-2f(j-1) \quad (19)$$

$$k_1=k+f(n-1) \quad (20)$$

はじめに，計算値を求める．各モジュールの動作概要は次のとおりである．

（1） 前処理部：j 番目の入力レコードのキー長を $2k_j$ バイトとすると，レコード長は，レコードヘッダ4バイトおよび可変長フィールドヘッダ4バイトを加えて，$(2k_j+8)$ バイトとなる．最初のレコードの入力からエレメントの出力までの時間 d_{p1}，および j 番目のエレメントが出力されてから $(j+1)$ 番目のエレメントが出力されるまでの時間 x_j のクロック数は次のようになる．

$$d_{p1}=12 \quad (21)$$

$$x_j=3k_j+9 \quad (22)$$

ここで，(21)式はヘッダ部を読み飛ばす時間8クロックと前処理部をデータが流れるのに要する時間の平均4クロックの和である．また，(22)式の右辺の第1項

は j 番目のレコードのキーを出力する時間，第2項は $(j+1)$ 番目のレコードのヘッダ部を読み飛ばす時間8クロックと x_j が3の倍数であるための遅れ1クロックの和である．なお，j 番目のエレメントの出力に要する時間は，キーと識別番号を合わせて $(3k_j+3)$ クロックである．よって，各出力エレメント間に，6クロックのデータが流れない時間がある．

（2） ソータ：キー長が一定または増加する場合と，減少する場合にわけて考える．

（A） キー長が一定または増加する場合：2.2節で述べたように，前のマージ処理の終了待ちは発生しないので，12段目のセルからの出力開始は，1段目のセルへの最後のエレメントの入力から，ソータをデータが流れるのに要する時間後となる．よって，1段目のセルへの最初のエレメントの入力から，12段目のセルからのソート後の最初のエレメントの出力までの時間 d_{s1} のクロック数は，次のようになる．

$$\begin{aligned} d_{s1} &= \sum_{j=1}^{n-1} x_j + 72 \\ &= (3k+9)n - 3k_n + 63 \\ &= (3k+9)n - 3k - 3e(n-1) + 63 \end{aligned} \tag{23}$$

（B） キー長が減少する場合：ソータの処理レート3クロック/語，等差数列の公差の絶対値 $2f \geqq 2$ 語，入力エレメント間のデータが流れない時間6クロックであるから，$6f \geqq 6$ となり，2.2節で述べたように，前のマージ処理の終了待ちが発生する．レコード数 n を 2^N とおくと，初めの $(12-N)$ 段のセルはデータがそのまま流れるので，これらの各セルにエレメントが入力されてから出力されるまでの時間 g_0 のクロック数は，次のようになる．

$$g_0 = 6 \tag{24}$$

その次の段のセルでは，2番目のエレメントが入力されたときマージ処理が開始され，出力ストリングは連続となる．よって，セルに最初のエレメントが入力されてから，最初のエレメントが出力されるまでの時間 g_1 は，次のようになる．

$$g_1 = x_1 + 6 - 3k_1 + 15 \tag{25}$$

以降のマージ処理が行われる i 段目 $(2 \leq i \leq N)$ のセルでは，$(2^{i-1}+1)$ 番目のエレメントが入力されたときマージ処理が開始され，出力ストリングも連続となる．よって，各セルに最初のエレメントが入力されてから，最初のエレメントが出力されるまでの時間 g_i は，次のようになる．

$$\begin{aligned} g_i &= \sum_{j=1}^{2^{i-1}} (3k_j+3) + 6 \\ &= 3 \cdot 2^{i-1}(k_1 + f + 1) - 3 \cdot 4^{i-1} f + 6 \end{aligned} \tag{26}$$

以上より，1段目のセルへの最初のエレメントの入力から，12段目のセルからのソート後の最初のエレメントの出力までの時間 d_{s2} のクロック数は次のようになる．

$$\begin{aligned} d_{s2} &= (12-N)g_0 + g_1 + \sum_{i=2}^{N} g_i \\ &= 3(k_1+f+1)n - fn^2 + 3k_1 - 2f + 75 \\ &= (3k+3)n - 3k + f(n-1)(2n-1) + 75 \end{aligned} \tag{27}$$

一方，j 番目のエレメントが出力されてから $(j+1)$ 番目のエレメントが出力されるまでの時間 y_j のクロック数は，次のようになる．

$$y_j = 3h_j + 3 \tag{28}$$

（3） 関係代数演算部：各エレメントのキーを読み飛ばし，レコード識別番号をレコードバッファに出力する．よって，最初のエレメントの入力から最初レコード識別番号の出力までの時間 d_{r1} のクロック数は，次のようになる．

$$d_{r1} = 3h_1 + 12 \tag{29}$$

また，j 番目の識別番号が出力されてから $(j+1)$ 番目の識別番号が出力されるまでの時間は，ソータからの出力と同じ y_j となる．

（4） レコードバッファからの出力：最初のレコード識別番号の入力からレコードの出力までの時間 d_b のクロック数は，固定長レコードソートの場合と同じく，次のようになる．

$$d_b = 4 \tag{12}$$

また，j 番目の識別番号が入力されてから $(j+1)$ 番目の識別番号が入力されるまでの時間は $y_j = (3h_j+3)$ クロックであり，一方，j 番目のレコードの出力に要する時間は $(2h_j+8)$ クロックである．全レコードの出力に要する時間はこれらのうちの大きいほうに依存するが，いま，キー長 $2h_j \geqq 10$ バイトであるから，常に，$(3h_j+3) \geqq (2h_j+8)$ である．よって，ソートされたレコードの出力に要する時間のクロック数 z は，次のようになる．

$$\begin{aligned} z &= \sum_{j=1}^{n-1} (3h_j+3) + (2h_n+8) \\ &= (3k+3)n - h_n + 5 \end{aligned} \tag{30}$$

以上より，全処理時間のクロック数を求める．

(A) キー長が一定または増加する場合：全処理時間のクロック数 v は次のようになる．

$$v=d_{p1}+d_{s1}+d_{r1}+d_b+z$$
$$=(6k+12)n-3k+3h_1-h_n-3e(n-1)+96 \quad (31)$$

(31)式の大部分を占めるのは，レコードの入出力時間に相当する $(6k+12)n$ である．いま，キー長の平均 k を一定として，等差数列の公差 e を変化させる場合，e, h_1, h_n による変化はわずかであり，v はほぼ一定となる．このことは，ソータにおいて前のマージ処理の終了待ちが発生しないことからも明らかである．

(B) キー長が減少する場合：全処理時間のクロック数 w は次のようになる．

$$w=d_{p1}+d_{s2}+d_{r1}+d_b+z$$
$$=(6k+6)n-3k+3h_1-h_n$$
$$+f(n-1)(2n-1)+96 \quad (32)$$

すなわち，w は f の増加とともに増加し，その増加量は f に比例する．次に，f を最大にする場合について考える．キー長が減少する場合のキー長の最小値は，最後のレコードのキー長 $2k_n$ である．(19)式と(20)式より，k_n は次のようになる．

$$k_n=k-f(n-1) \quad (33)$$

キー長は最低でも 10 バイトという条件から，f を最大にしたとき，次の関係が成り立つ．

$$f(n-1)=k-k_n=k-5 \quad (34)$$

(32)式に(34)式を代入すると，次のようになる

$$w=(8k-4)n-4k+3h_1-h_n+101 \quad (35)$$

(31)式と(35)式の kn の項の係数を比較することより，キー長が減少する場合，処理時間が最大 33% 弱増加することがわかる．

5.3 測　　定

試作システムにおいてソート処理を実行し，エンジンのレコード入力開始から出力終了までの時間を測定した．計算値と測定値の比較を図 9 および図 10 に示す．ただし，図 10 の入力レコードにおいて，キー値は，$k_1=h_n$, $k_n=h_1$ となるように与えている．計算値と測定値の差は入力チャネルのファームウェアのオーバヘッドであることがわかっており，エンジンは設計どおりの性能をもつことが示された．また，試作システムでは，演算要求の解析，エンジンの初期化，演算終了の検出等のために，チャネルおよびコントローラのファームウェアのオーバヘッドが約 10 ミリ秒あるが，これは測定値に含めていない．

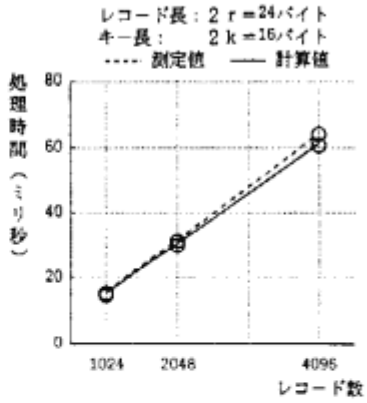


図 9 固定長レコードソート処理時間
Fig. 9 Fixed length record sorting.

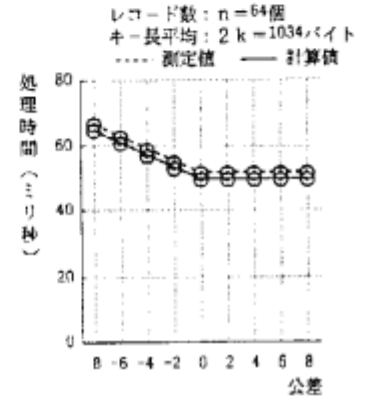


図 10 可変長レコードソート処理時間
Fig. 10 Variable length record sorting.

6. おわりに

可変長レコード用関係データベース処理エンジンの設計と試作，およびそのソート処理性能の評価について述べた．エンジンは汎用計算機の付加プロセッサとして，主記憶または磁気ディスク上のデータに対して，ソートや関係代数演算をストリーム処理する．ソート処理速度は 2.67 メガバイト/秒（2 バイト/750 ナノ秒），エンジンのデータ入出力は最大 4 メガバイト/秒（2 バイト/500 ナノ秒）である．

可変長レコードソートにおいては，レコード長のばらつきによる前のマージ処理の終了待ちが発生する．このため，総データ量が同じである固定長レコードソートよりも，一般に処理時間が長くなる．本エンジンでは，キー切出し方式により，セルにおける前のマージ処理の終了待ちを低減し，処理速度の向上をはかった．また，可変長キーのマージ処理に必要な制御情報をタグとして付加することにより制御を簡単化

し，処理速度を低下させることなくハードウェア量の削減を実現した．

ソート処理性能の解析と実測による評価により，エンジンの性能が設計どおりであることを確認した．また，入力レコードが等差数列的にキー長の減少するキーフィールドのみからなる場合，総データ量が同量の固定長レコードに対するよりも，ソート処理時間が約33%長いことが示された．

謝辞 本研究は(財)新世代コンピュータ技術開発機構における研究開発の一環として実施したものであり，研究開発に参加された同機構および(株)東芝の関係者に深く感謝いたします．

参考文献

1) Akl, S. G.: *Parallel Sorting Algorithms*, p 229, Academic Press, Orlando (1985).
2) Teich, W. and Zeidler, H. Ch.: Data Handling and Dedicated Hardware for the Sort Problem, in Leilich, H.-O. and Missikoff, M. (eds.), *Database Machines*, pp. 205-226, Springer-Verlag, Berlin (1983).
3) Kitsuregawa, M. et al.: Architecture and Performance of Relational Algebra Machine GRACE, *Proc. International Conference on Parallel Processing 1984*, pp. 241-250 (1984).
4) Sakai, H., Iwata, K., Shibayama, S., Abe, M. and Itoh, H.: Development of Delta as a First Step to a Knowledge Base Machine, in Sood, A. K. and Qureshi, A. H. (eds.), *Database Machines Modern Trends and Applications*, pp. 159-181, Springer-Verlag, Berlin (1986).
5) Torii, S. et al.: A Relational Database System Architecture Based on Vector Processing Method, *Proc. International Conference on Data Engineering*, pp. 182-189 (1987).
6) Itoh, F., Shimakawa, K., Togo, K., Matsuda, S., Itoh, H. and Oba, M.: Design, Implementation, and Evaluation of a Relational Database Engine for Variable Length Records, in Kitsuregawa, M. and Tanaka, H. (eds.), *Database Machines and Knowledge Base Machines*, pp. 269-282, Kluwer Academic Publishers, Boston (1988).
7) Chung, K. M., Luccio, F. and Wong, C. K.: On the Complexity of Sorting in Magnetic Bubble Memory Systems, *IEEE Trans. Comput.*, Vol. C-29, No. 7, pp. 553-563 (1980).
8) Nassini, D. and Sahni, S.: Bitonic Sort on a Mesh Connected Parallel Computer, *IEEE Trans. Comput.*, Vol. C-27, No. 1, pp. 2-7 (1979).
9) 安浦，高木：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. J65-D, No. 2, pp. 179-186 (1982).
10) Todd, S.: Algorithm and Hardware for a Merge Sort Using Multiple Processors, *IBM J. Res. Dev.*, Vol. 22, No. 5, pp. 509-517 (1978).
11) Tanaka, Y., Nozaka, Y. and Masuyama, A.: Pipeline Searching and Sorting Modules as Components of a Data Flow Database Computer, in Lavington, S. (ed.), *Information Processing 80*, pp. 427-432, North-Holland Publishing Company, Amsterdam (1980).
12) 土肥：大容量ファイルを整列するシストリック・ソータ，電子通信学会論文誌，Vol. J67-D, No. 3, pp. 281-288 (1984).
13) 岩田，神谷，酒井，柴山，伊藤，村上：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).
14) Kitsuregawa, M., Yang, W., Suzuki, T. and Takagi, M.: Design and Implementation of High Speed Pipeline Merge Sorter with Run Length Tuning Mechanism, in Kitsuregawa, M. and Tanaka, H. (eds.), *Database Machines and Knowledge Base Machines*, pp. 89-102, Kluwer Academic Publishers, Boston (1988).
15) 楊，伏見，喜連川，田中：パイプラインマージソータに於ける可変長レコード用 String Length Tuning アルゴリズム，電子通信学会技術研究報告，CPSY 86-9 (1986).
16) 楊，喜連川，高木：可変長レコードを支援するパイプラインマージソータの構成，情報処理学会研究報告，Vol. 86, No. 77, 86-CA-63, pp. 109-117 (1986).

(昭和63年8月8日受付)
(平成元年5月9日採録)

伊藤 文英（正会員）

昭和33年生．昭和56年大阪大学基礎工学部制御工学科卒業．昭和58年同大学院基礎工学研究科物理系専攻修士課程修了．同年東京芝浦電気(株)（現(株)東芝）入社．同社青梅工場，情報通信システム技術研究所において，データベースマシン，データベース処理エンジンの研究・開発に従事．昭和62年(財)新世代コンピュータ技術開発機構に出向．同機構研究所において，知識ベースマシンの研究・開発に従事．

島川 和典（正会員）

昭和26年生．昭和45年島根県立浜田高校普通科卒業．同年東京芝浦電気(株)（現(株)東芝）入社．同社青梅工場，情報通信システム技術研究所において，データベース管理システム，データベースマシンの研究・開発に従事．

東郷 一生（正会員）

昭和37年生．昭和59年鹿児島大学工学部電子工学科卒業．同年(株)東芝入社．同社青梅工場において小型計算機の開発に従事．

松田 進（正会員）

昭和23年生．昭和46年九州大学工学部通信工学科卒業．同年東京芝浦電気(株)（現(株)東芝）入社．同社青梅工場において小型計算機の開発に従事．

伊藤 英則（正会員）

昭和21年生．昭和44年福井大学工学部卒業．昭和49年名古屋大学大学院工学研究科博士課程修了．工学博士．昭和49年4月，日本電信電話公社横須賀電気通信研究所入社．昭和60年4月，(財)新世代コンピュータ技術開発機構に出向．平成元年2月，日本電信電話(株)情報通信処理研究所に復帰．これまでに，オートマトンと数理言語の研究，計算機ネットワークシステムの研究開発，知識ベースシステムの研究開発に従事．電子情報通信学会，人工知能学会各会員．

大場 雅博（正会員）

昭和50年，東京大学工学部精密機械工学科卒業．昭和52年，同大学院修士課程修了．同年，(株)日立製作所に入社．昭和60年より3年間，(財)新世代コンピュータ技術開発機構に勤務．(株)日立製作所システム開発研究所にて，物流システムや水道システムの計画・管理・運用技法の研究に従事．IEEE，電気学会，計測自動制御学会などの会員．