ICOT Technical Report: TR-203

TR-203

関係データベース処理エンジンのソータの試作と評価

岩田和秀、神谷茂雄、酒井 浩、柴山茂樹(東芝)
村上国男(NTT)、伊藤英則(ICOT)

September, 1986

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

関係データベース処理エンジンのソータの試作と評価

岩田　和秀（東芝），神谷　茂雄（東芝），酒井　浩（東芝）
柴山　茂樹（東芝），伊藤　英則（ICOT），村上　国男（NTT）

Implementation and Evaluation of the Sorter in Relational Database Engine

Abstract

This paper describes the design considerations and the performance evaluation of a hardware sorter. Various kinds of sorters have been proposed. However, the sorters proposed so far aimed at the verification of hardware sorting algorithms and are not designed in detail. Our sorter is designed and implemented as a component of a relational database machine Delta developed by Japan's Fifth Generation Computer Project. The sorter is used not only for sorting but also as a preprocessor for relational database operations such as join. The sorter is based on the straight two-way merge-sort algorithm and implemented using currently available technology. The sorter consists of a linear array of 12 processing cells. The sorting operation is efficiently performed by pipeline processing synchronized with the data transfer rate of 3MB/sec. We discuss the functions of the sorter required for relational database processing and evaluate the performance theoretically and experimentally.

1．まえがき

第５世代コンピュータ・プロジェクトの前期（昭和57年度～昭和59年度）では、知識ベースマシン研究の第１歩として、Prologのファクトを格納する関係データマシンDelta の開発が行われた。筆者等は、このプロジェクトに参加し、Delta の特徴の１つである、関係代数演算等をパイプライン方式で実行する関係データベース処理エンジンの開発を行った[1][2]。本報告では、このエンジンの基本構成要素の１つであるソータの特徴と試作結果について述べる。

ソーティングは非数値処理分野における基本演算の１つであり、古くから数多くのアルゴリズムが提案されて、目的に応じた使い分けが行われてきた[3]。

1970年代に入ると、ホスト計算機のソート処理ルーチンを付加プロセッサのファームウェアで実行して、ホスト計算機の負荷を軽減する方式が提案された[4]。その後、VLSI技術の進歩と関係データベースマシンの研究の興隆により、ソータの提案が活発になった。

これまでに提案されたソータの主要なものには，磁気バブルメモリのループ構造を利用したソータ[5]、ｎ個のレコードをｎ個のセルを用いてソートするバイトニックソータ[6]や並列計数ソータ[7]、log ｎ個のセルを用いてｎ個のレコードをソートするパイプライン方式のソータ等がある。これらを現状の技術を用いて試作するという観点からみると、log ｎ個のセルを用いたパイプライン方式のソータが、ハードウェアが小形化できるので実用的と考えられる。

パイプライン方式のソータとしては、マージソータ[8]、ヒープソータ[9]、シストリックソータ[10]等が発表されている。ヒープソータはレコード列の入力を終了してから結果が出力され始めるまでの出力遅れ時間がなく、かつメモリの使用効率が良いという優れた特徴を持っている。しかし、データの入出力端が同一のため、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できない。シストリックソータは入出力端を独立にしてヒープソータの問題点を解決し、更に比較結果を移動歴として記憶することによりマルチウェイ・マージを可能とした大容量ソータである。しかし、マルチウェイ・マージを行う時のデータ転送制御方式等に未検討の課題が残されている。これらに対して、マージソータは若干の出力遅れ時間があること及びメモリの使用効率が悪い欠点を持つものの、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、汎用計算機の付加プロセッサとするのに適している。

マージソータについては、そのアルゴリズム[11]及びシミュレーション[8]による動作確認等がすでに発表されている。しかし、これらの研究はアルゴリズムの検証が中心であり、ソータを実際に応用する場合の諸条件が十分に検討されていない。即ち、ソータを関係データベースの処理に応用する場合には、レコード長、レコード数、キー長等のパラメータ、null値の扱い、重複レコードの検出等に関する柔軟な対応が必要であるが、現段階ではこれらの実現方式を盛り込んだソータの設計は行われていない。

そこで，筆者らは関係データベースの処理にソータを使用する場合にソータで具備すべき諸機能を考察し、それらの機能を盛り込んだソータの設計を行った。次に、3MByte/secのデータ転送速度に同期して4096個の同一形式のレコードをソートできるソータ（固定長ソータ）の試作を行い、処理時間の解析結果と実測値の比較評価を行ったので、その概要を報告する。

2．ハードウェア化の検討

2．1　アルゴリズム

アルゴリズムは2ウェイ・マージソート法を語（2バイト）単位で処理する方式にパイプライン化したものである。このアルゴリズムは、2つのソートされたレコード列（以下、ストリングと呼ぶ）のマージ操作を、繰返し実行してソート処理を行うものである。本ソータは、このマージ操作を、1次元に12個配置したセルにより連続的に行う。即ち、対象となる入力レコード列（以下、ストリームと呼ぶ）を1段目のセルから入力すると、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て、2^{i}個のレコードよりなるストリングを次段のセルに出力する動作を繰返す。従つて、ストリングの長さはセルで処理される毎に2倍され、最終段セルからは2^{12}個のレコードからなるストリングが得られる。

レコード数Cが3、レコード長Lが6バイトの時のパイプライン処理の様子を図1に示す。この図より、ストリームが入力され始めた時点から結果が出力され終るまでのソート処理時間T(S)を求めると、次のようになる。

$$T(S) = 2CLT + (2^{a} - C)LT - (L - 2a)T$$

ただし、$a = < \log_2 C >$で$< >$は小数点以下を切上げた整数値、Tは1バイト当りの処理時間を示す。上式において、第2項はCの2の累乗からのずれによる遅れ時間、第3項はレコードの最初の2バイトが入力された時点から処理が開始されるためのレコードの終りからみた先行時間を示す。

次に、セルiは原理的には$(2^{i-1}+1)\times L$のメモリ容量を持てばよいが、メモリは2つのストリングを区別してしかもキュー機能を持つ必要があるので、本ソータでは$2^{i}\times L$の容量のメモリをセルiに実装してセルの制御回路を簡単にする方式を採用した。

2．2　レコード形式とデータタイプ

関係データベース処理では、レコードをキーとして扱う場合とレコードのあるフィールドをキーとして扱う場合がある。本ソータでは、セルに後述するパス・モード機能を持たせて、ストリーム内のレコードのレコード長とキー長をともに2バイト単位で最大4Kバイトの長さまで指定できるようにした。

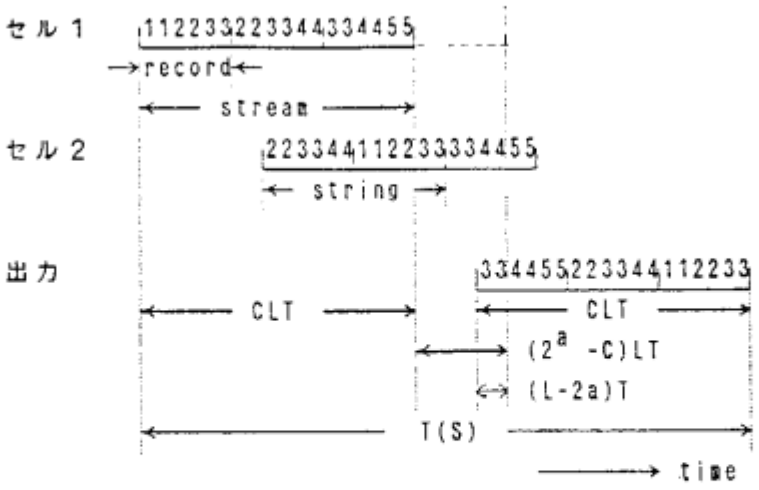

図１　パイプライン処理の様子

フィールドをキーとした演算を効率よくパイプライン処理するには、キーがレコードの先頭にある状態で入力される必要がある。そこで、ソータの入力側に１レコード分のメモリを用意して、レコードのフィールドを回転させ、出力側で元に戻す方式を採用した。

データベースの処理では様々なデータタイプが扱われるが、それをそのままソータで扱うことは困難である。そこで、ソータの入力側でキーを絶対値数に変換し、出力側で逆変換する方式を採用した。本ソータで扱う入力データのタイプは、絶対値数、整数及び正規化されたIBM フォーマットの浮動小数点数である。

２．３　Null値と重複キーの取り扱い

データベースでは、フィールドの値が不明または定義されていない時、その値はnull値と呼ばれ、特別な扱いがされる。本ソータでは、入力ストリームにnull値が含まれている場合、まず正常なキー値を持つレコードを出力し、その後にnull値のキーを持つレコードを入力順に出力するようにした。実現方式としては、セルの制御回路を簡単にするため、ソータの入力側で後述するnull信号を発生させる方式を考案した。

等しいキーを持つレコードを検出することは、レコードをキーとして扱う場合はunique演算等で、フィールドをキーとして扱う場合はjoin演算等で重要となる。等しいキーの検出はセルで容易にできるが、タグの操作がセルの制御回路を複雑にする。本ソータでは、最終セルの出力にチェッカと呼ぶ専用回路を付加して、ソート結果のチェックと重複信号の発生を行うようにした。なお、重複したキーを持つレコードは、入力された順に出力してマルチ・キー・ソートに対応できるようにした。

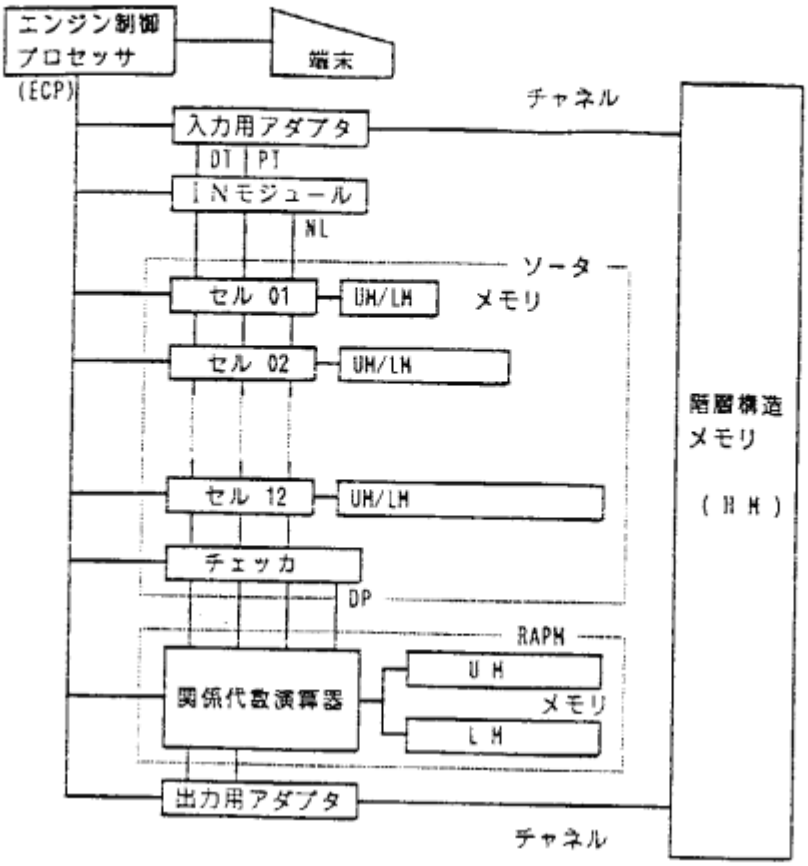

図2　RDBEのシステム構成

3．システム構成の概要

3．1　システム構成

本ソータは関係データベース処理エンジン（以下、RDBEと略記する）の基本構成要素の1つとして開発されたので、ソータの性能評価は図2に示すRDBEのシステム構成で行った。RDBEは入力用アダプタ(HMA-IN)、INモジュール、ソータ、関係代数演算モジュール(RAPM)、出力用アダプタ(HMA-OUT) 及び制御用マイクロプロセッサ(ECP) で構成され、チャネルを介して階層構造メモリ（HM）に接続されている。

RDBEはECP の制御により、HMからのデータをHMA-IN、INモジュール、ソータ、RAPM、HMA-OUT を介してパイプライン処理し、結果をHMに転送する動作を行う。

3．2　各モジュールの機能

(1) HMA-INとHMA-OUT ：RDBEとHM間のインタフェースの制御を行う。

(2) INモジュール：フィールドの回転操作、データ・タイプの変換操作及びnull信号を発

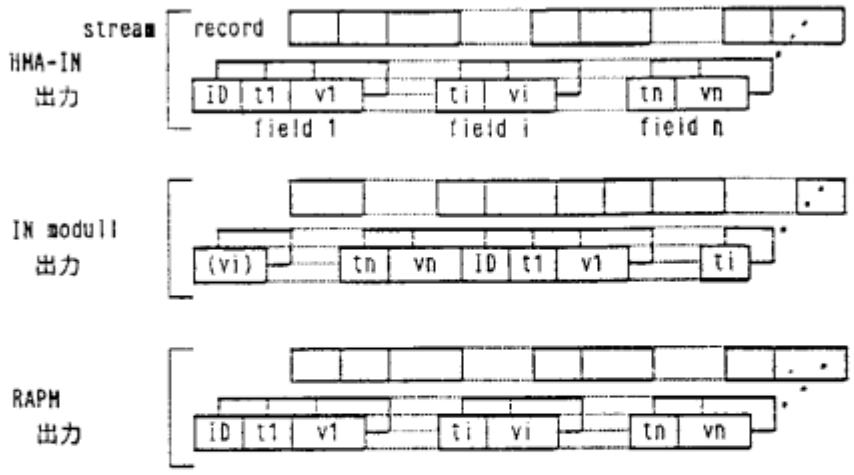

図３　レコード形式とフィールドの回転操作

生する操作等を行う。Null信号はタグを調べて１語毎に生成するもので、パリティ・ビットと同様に専用線で転送される。

(3) ソータ：12個のセルと１個のチェッカにより構成され、入力ストリームをソートした後、その結果のチェックとキーの重複を検出して重複信号を生成する。

(4) RAPM：２つの64Kバイトメモリ(UM,LM) と関係代数演算器で構成され、関係代数演算、マージ演算及びINモジュールで行った各種変換の逆変換等を行う。

(5) ECP ：上記のモジュールの制御及びRAPMでは処理できない算術演算等を実行する。

(6) HM: 汎用コンピュータで実現された大容量メモリで、RDBEとは3MByte/secのデータ転送速度を持つ２つのチャネルで結合されている。

図３に、RDBEに入力されるストリームのレコード形式とINモジュール及びRAPMにおけるキー・フィールドｉの回転操作の様子を示す。同図において、IDはレコード識別子、ＴｉとＶｉ はそれぞれｉ番目のフィールドのタグ及び値、(Vi)は絶対値を示す。

図２において、DT、PT、NL、DPはそれぞれデータ線（16ビット）、パリティ・ビット線（２ビット）、nullビット線（１ビット）、重複ビット線（１ビット）を示す。Nullビット線と重複ビット線は、データ線上のデータの属するレコードのキー値が、null値であること及び前のレコードのキー値と等しいことをそれぞれ示す。

３．３　ソフトウェア構成

RDBEのソフトウェアは、ハードウェアを制御する機能とハードウェアでは実行できない算術演算等を行う機能を持つが、本稿では前者の機能を実現する制御プログラムについて述べる。

制御プログラムの構成を図４に示す。各部の機能の概要は次の通りである。カーネル部

は割り込み処理や特権命令の処理、入出力制御部はRDBEの各モジュールの制御、関係代数演算処理部はRDBEコマンドの解釈と実行、トップ・レベルはRDBEコマンド処理のスケジューリングをそれぞれ行う。なお、テスト・評価プログラム部には、データベースのサイズの指定と作成機能及びRDBEコマンドの種類をメニューから選択できる機能を持たせ、端末よりソータの性能測定が行えるようにした。

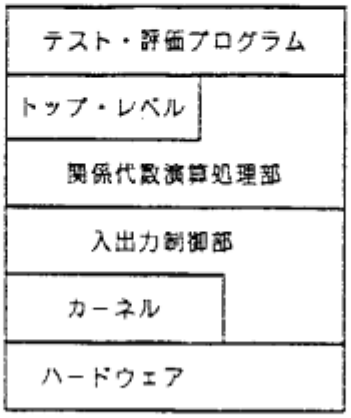

図4　RDBE制御プログラムの構成

4．ソータのハードウェア構成

4．1　ソータの仕様

試作したソータの概略仕様は次の通りである。

ソート・レコード数　：4096個
レコード長L　：　$2 \leq L \leq 4096$バイト
キー長K　：　$2 \leq K \leq 4096$バイト
セル数N　：$N = 12$
メモリ素子　：　8×8kビットSRAM
セルiのメモリ容量　：　2^{i+4} バイト
セル12のメモリ容量M：$M = 64k$ バイト
処理速度T　：$T = 330ns$ ／バイト

本ソータはレコード長が16バイト以下の時、4096個のレコードをソートする能力を持つが、セルのメモリ容量が固定されているので、この値はレコード長により変化する。そこで、セルには2つの入力ストリングをマージするソート・モードの他に、入力されたデータをメモリを介さずにそのまま出力するパス・モードを用意し、レコード長が大きい時はそれを格納できるメモリ容量を持つセルまでパス・モードを用いることにした。従って、本ソータの実効ソート容量Eは、次式のようになる。

$$E = L \times 2^{e}$$

ただし、eは実効動作セル数で$e = \min([\log_2 M/L], N)$、Nは実装セル数、Mは最終段セルの実装メモリ容量、[] はガウス記号を示す。

4．2　セルの制御方式

セルの2つの動作モードは、入力ストリームのレコード数Cとレコード長Lにより、次式に従って使い分ける。

$$J = \min([\log_2 M/L], N, [\log_2 2(C-1)])$$

即ち、1から(N-J) 番目のセルがパス・モードで、(N-J+1) から12番目のセルがソート・モードで動作するよう制御する。例えば、L＝ 100バイトでC＝ 200の時、J＝8となるので、1から4番目のセルがパス・モードで、5から12番目のセルがソート・モードで動作する。入力データを直接RAPMに転送する時は、全セルをパス・モードで動作させる。

4．3　セルの構造

試作したセルの構造を図5に示す。セルは演算部、アドレス生成部、制御部及びインタフェース部より構成され、メモリに接続される。

演算部は入力データ用レジスタ(INR)、データ保持用の2つのレジスタ(UR,LR)、比較器(CMP)、出力データを選択するセレクタ(SEL) 及び出力データ保持用レジスタ(OUTR)より構成され、2つのデータを比較して条件に合致したデータを出力する動作を行う。

アドレス生成部は書込みアドレスカウンタ(WAC)、2つの読み出しアドレスカウンタ(RUAC,RLAC)、2つのアドレス保持用レジスタ(UAHR,LAHR) より構成され、入力データの書込みアドレスと比較データの読み出しアドレスを生成する。UAHR(LAHR)は比較中のレコードの先頭アドレスを保持するもので、選択されなかったレコードを次のサイクルで読み出す時に用いる。

制御部は入力ストリームのレコードを計数するカウンタ(STMC)、レコード長を保持するレジスタ(RLHR)、レコード長制御カウンタ(RLCC)、キー長保持レジスタ(KLHR)、キー長制御カウンタ(KLCC)、ストリングのレコード数保持レジスタ(STRR)、入力ストリングのレコードを計数するカウンタ(WSTC)、メモリに格納されたストリングの未処理レコードを計数するカウンタ(USTC,LSTC)、状態遷移を制御する15個のフリップ・フロップ及び外部インタフェースより構成され、セル全体の制御を行う。

メモリは入力ストリングを区別するためUメモリ(UM)とLメモリ(LM)に論理的に分割されており、入力ストリングをUM、LMの順に交互に格納する。

4．4　セルの外部インタフェース

セルは、セル間のデータ転送と ECPの入出力バスに関するインタフェース機能を持つ。

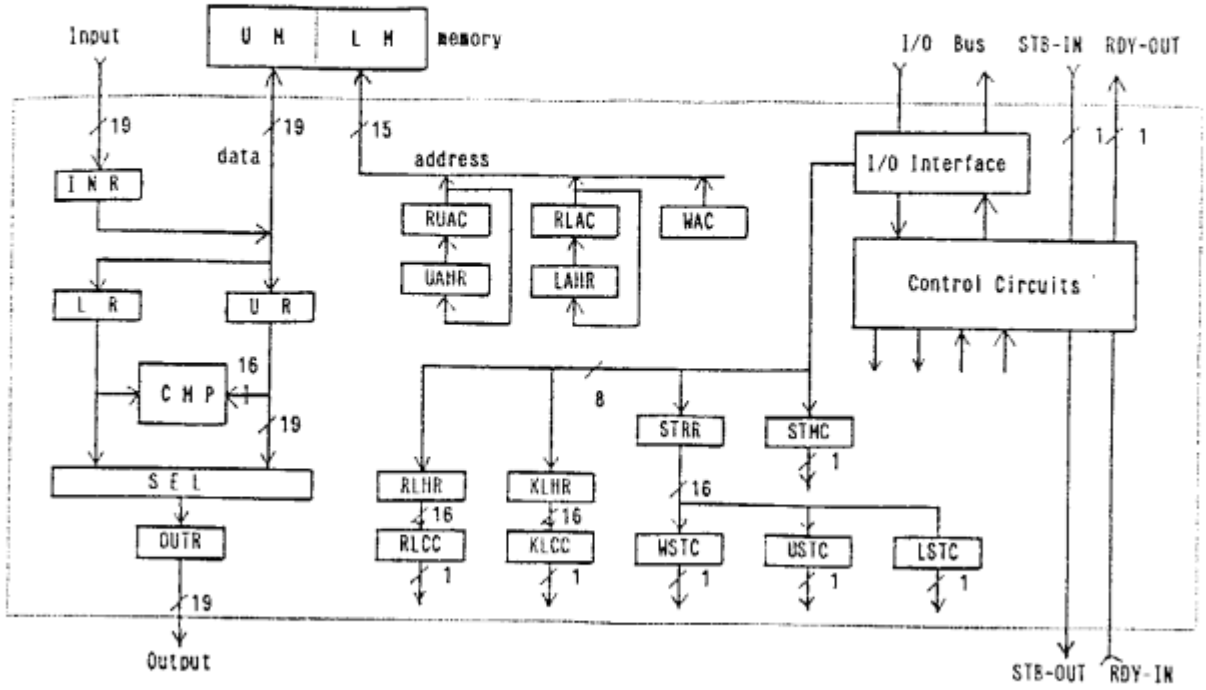

図５　セルの構造

セル間のデータ転送は、転送準備完了信号STB-IN(OUT) と転送要求信号RDY-IN(OUT) により前後のセルの準備が完了した時に起動され、一方でも準備が完了していなければ待ち状態に入るよう制御される。このため、ソータの入力側及び出力側でデータ転送が一時的に中断されてもパイプライン動作は乱れない。

ECP からセルにセットされる制御情報には、レコード長、キー長、ストリームとストリングのレコード数、セルの動作モード指定がある。

４．５　セルの処理タイミング

本セルは３サイクルで１語の処理を行う。処理タイミング・チャートを図６に示す。サイクルタイムは220 nsである。

第１サイクルは、セル間の制御信号STB-IN(OUT) とRDU-IN(OUT) の４つが共にONの時に開始され、入力データのINR へのセット、UMのデータをURに読み出す動作、及びOUTRデータの出力動作を行う。第２サイクルはLMまたは INRのデータをLRに読み出す動作を、第３サイクルはURとLRの比較及びINR データのメモリへの書き込み動作を行う。

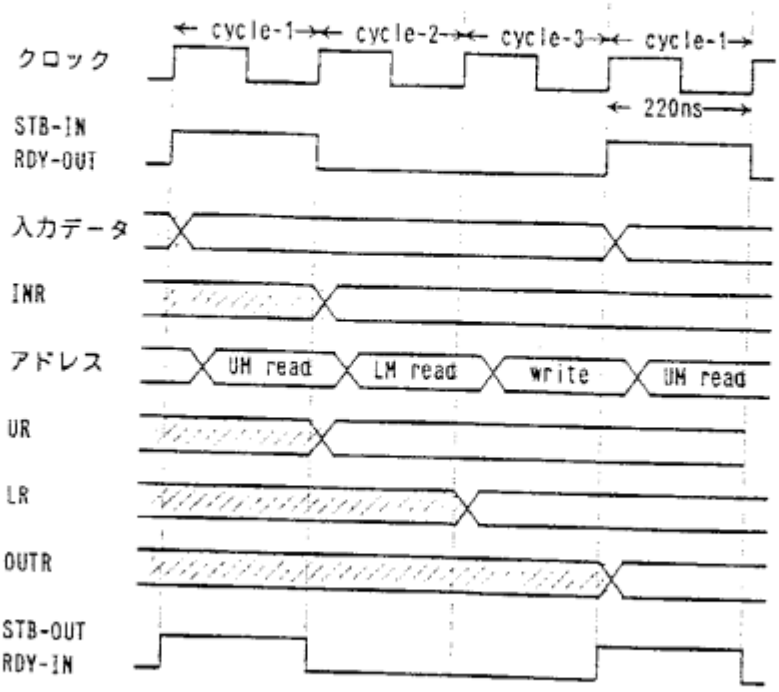

図6　処理タイミング・チャート

5．セルの状態遷移

5．1　ソート・モードの状態遷移

このモードの動作は、次に示すような3階層構造で制御される。各階層での状態遷移を図7に示す。

(1) 下位階層：1語単位の繰返し処理を制御する。URD、LRD、WT/Cの3つの状態が、それぞれ図6の第1、第2、第3サイクルに対応した動作を行う。

(2) 中位階層：レコード・レベルの繰返し処理を制御する。CMP 状態は等号が成立して比較結果がまだ決定していない状態で、URのデータをOUTRに転送する。比較結果が決まると、UM(LM)のレコードが選択された場合はSELU(SELL)状態へ遷移する。なお、この状態の始めにnull信号がチェックされ、一方がnull値であれば正常なキーのレコードが、両方ともnull値であればUMのレコードが選択される。SELU(SELL)状態はUM(LM)のレコードの残りの語を出力する。

(3) 上位階層：ストリング／ストリーム・レベルの繰返し処理を制御する。S01 状態は第1ストリングをUMに格納する。S02.S03 状態は第2ストリングをLMに格納しつつ、両メモリのレコードの比較を行い、LMへの格納が終ると、両メモリにレコードが有る時は S04へ、UM(LM)だけに有る時はS05(S06)へ遷移する。S04 状態は第3ストリングをUMに格納しつつ、

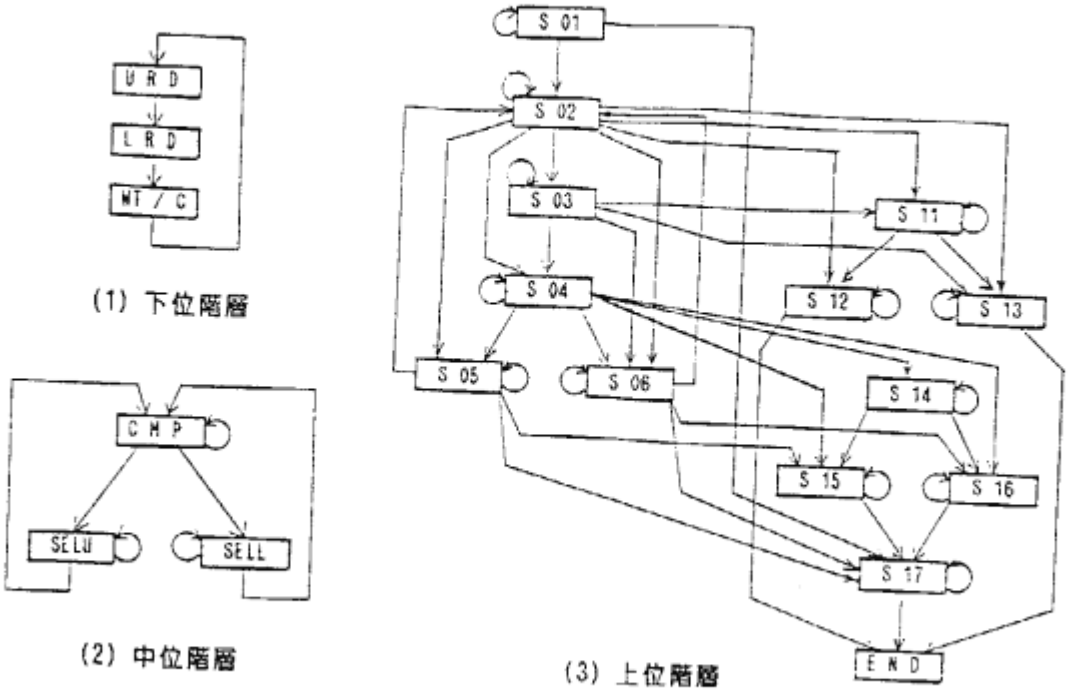

図7　状態遷移

両メモリのレコードの比較を行い、UM(LM)の比較中のストリングがすべて出力されるとS06(S05)へ遷移する。S05(S06)状態はUM(LM)のレコードの出力と入力レコードのUMへの書込みを行い、S04 状態へ遷移する。

以下のS11 ～S17 状態は、上記S02 ～S06 の状態でストリームが入力され終った時の処理を行う。

S11 状態は両メモリのレコードの比較を行い、LM(UM)にレコードが無くなるとS12(S13)へ遷移する。S12(S13)状態はUM(LM)のストリングを出力してEND へ遷移する。S14 状態は両メモリのレコードの比較を行い、LM(UM)にレコードが無くなるとS15(S16)へ遷移する。S15(S16)状態はUM(LM)の残ったストリングを出力してS17 へ遷移する。S17 状態はUMの最後のストリングを出力してEND へ遷移する。

5．2　バス・モードの状態遷移

このモードは、上記下位階層の3つの状態遷移を次のように簡単化して1階層で制御される。URD 状態で入力データのINR へのセットとOUTRデータの出力動作を、LRD 状態でINR のデータのLRへの転送を、WT/C状態でLRのデータのOUTRへの転送を行う。

従って、1語は660ns でセルを通過する。

6．性能評価

6．1　処理時間の解析

入力ストリームは、実効ソート容量E以下のサブストリームに分割して処理される。この時、サブストリームの個数により処理方式が異なる。サブストリームの個数FをF＝<CL/E>と表わすと、F＝1の時はソータを、F＝2の時はソータとRAPMを、F＞2の時はF＝2の処理とRAPMを用いた処理方式となる。

4.1節で述べたように、本ソータはストリームの大きさによりセルの動作個数が変化するので、処理時間の一般式を導くのは複雑になる。そこで、以下の解析では、ストリームの大きさが64k バイトの整数倍の場合を考え、E＝64k バイトとする。

(1) F＝1の場合の処理時間T(1)：ストリームはHMA-IN、INモジュール、ソータ、RAPM、HMA-OUT を介してHMに送られる。従って、T(1)は次式で表わされる。

$$T(1) = t0 + t(S) + 3LT + t2$$

ここで、t0 はコマンドの解釈とエンジンを起動するための時間、t(S)は 2.1節で求めたT(S)でCL=Eとしたソータでの処理時間、3LT はINモジュールとチェッカとRAPMにおける1レコード分のバッファリング遅れ時間の和、t2 はHMへの終了報告時間を示す。

(2) F＝2の場合の処理時間T(2)：ストリームSは2つのサブストリーム(S1,S2) に分割されてソートされ、RAPMでマージされる。この場合の各モジュールにおける処理時間を図8に示す。同図において、t1はS1からS2への切替えとエンジンにパラメータをセットする時間を示す。従つて、T(2)は次のようになる。

$$T(2) = t0 + t2 + (t1 + 2t(s) + ET + 5LT)$$

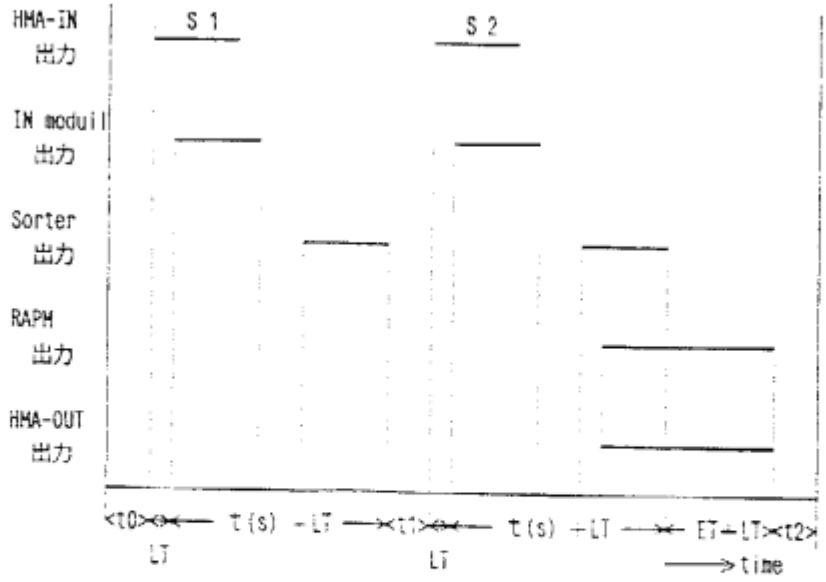

図8　各モジュールにおける処理時間

(3) F > 2の場合の処理時間T(2^i) ：F = 4の例を検討してF = 2^i の場合に拡張する。F = 4の時、まずストリームSを2つのサブストリーム(S1,S2) に分割し、これをF = 2の方式で処理する。次に、S1, S2をEの単位に分割、即ち(S11,S12) と(S21,S22) にして、これら4つのサブストリームのマージ操作をRAPMで行う。従って、T(4)は次のようになる。

$$T(4) = t0 + t2 + 2A + 2(2t1 + 3LT + 2NT + 2T + 3ET)$$

だだし、A = (t1 + 2t(s) + ET + 5LT)である。

上式の右辺において、2AはF = 2の処理を2回行う時間を、括弧内は S11と S21をマージする場合の最悪処理時間を示す。即ち、 S11をソータをパス（遅れ時間(N+1)T)してRAPMのUMに入力（入力時間ET）し、次に S21を同じくソータをパスしてRAPMのLMに入力しつつ、UMとLMのレコードのマージ（処理時間 2ET）を行う時間を示す。次に、F = 2^i の場合の処理は、まずストリームを2E単位でソーティングするためF = 2の処理を 2^{i-1} 回繰返し、その後上記のマージ操作を (2^i E/2E) log(2^i E/2E) 回繰返すことにより実行する。従つて、T(2^i) は次のようになる。

$$T(2^i) = t0 + t2 + A\,2^{i-1} + B(i-1)\,2^{i-1}$$

ただし、B=(2t1 + 3LT + 2NT + 2T + 3ET)である。

6．2　測定結果

測定は図2のハードウェア構成で、端末よりRDBEコマンドを入力し、ECP 内の評価プログラムにより、コマンドの解釈からレスポンスの作成までの時間を計測した。なお、評価の対象としたストリームは、ECP で乱数を発生させ、それをすべてHMのバッファ上に格納して行った。レコードの長さはすべて16バイトで、レコード全体をキーとして扱った。

ストリームの	ソート処理時間(msec)	
レコード数	計算値	実測値
1	15	16
4,096	57	65
8,192	123	141
16,384	369	460
32,768	999	1,340
65,536	2,535	3,330
131,072	6,159	8,160
262,144	14,511	18,390
524,288	33,423	45,860

表1　ソート処理時間

6．3　考察

本ソータの処理時間の計算値と測定値を表1に示す。t0 ～ t2 の値としては、それぞれ実測平均値10ms、 3ms、5ms を用いたが、測定値は計算値よりデータ量が増加するにつれて12%から35%ほど大きくなつている。これは、図2の評価システムがRDBEからHMにデータ要求を出して処理する方式のため、上記のt0 ～ t2 以外にRDBEとHM間で多数の内部コマンドが発行されるためのオーバーヘッドである。従って、このオーバーヘッドを減すためには、RDBEをHMの入出力機器としてHMから制御することが有効と考えられる。

7．むすび

関係データベースの処理を行う場合に必要となるレコード数、レコード長、キー長等のパラメータ、null値の取り扱い、重複レコードの取り扱い等の諸機能を考慮したソータの設計と試作結果について述べた。本セルの回路規模は、約5000ゲートであり、8000ゲートのゲートアレイで実現することができた。

本ソータは、単体で64K バイト、RAPMを連動させた時は128Kバイトのストリームを3MByte/secの処理速度でソートする。この処理速度は、現在のチャネルの最高転送速度に相当する。従って、本ソータは汎用計算機の付加プロセッサとして、磁気ディスクから主記憶へのデータ転送時に、ソート処理や関係代数演算を行う等の応用に適している。後者への応用については別途報告する予定である。

なお、本ソータは関係代数演算の前処理に用いることを主目的に開発したので、

(1) 大量レコードのソートでは性能が低下する。

(2) セルのメモリ使用効率が悪い。

等の欠点を持つ。(1) については、RAPMにマルチウェイ・マージ機能を付加する、汎用計算機の付加プロセッサとして用いる場合は汎用計算機でマルチウェイ・マージを行う等の方法が考えられる。しかし、マルチウェイ・マージを効率よく行うためには磁気ディスクを含めた検討が必要であり、今後の課題である。(2) については、原理的にはセルiは$(2^{i-1}+1)\times L$のメモリ容量を持てばよいので、ポインタを用いて削減することも可能である[10]。しかし、セルの制御回路が複雑になること及びセルのメモリ容量が2倍づつ増えていく特殊構成のため、本ソータの規模では $2^{i}\times L$のメモリ容量を実装する方がハードウェアを小型化できる。現在、本ソータの開発経験を基に，可変長キーを処理できるソータの設計を進めている。

参考文献

(1) Sakai,H.,Iwata,K.,Kamiya,S., Abe,M.,Tanaka,A.,Shibayama,S. and Murakami,K.: Design and Implementation of the Relational Database Engine,FGCS'84,pp.419-426 (1984).

(2) Kamiya,S.,Iwata,K.,Sakai,H., Matsuda,S.,Shibayama,S. and Murakami,K.: A Hardware Pipeline Algorithm for Relational Database Operation and Its Implementation Using Dedicated Hardware,IEEE 12th ISCA,pp.250-257(1985).

(3) Knuth,D.E.:The Art of Computer Programming,Vol.3,Sorting and Searching, Reading,Addison Wesley,pp. 11-388(1973).

(4) Barsamian,H.:Firmware Sort Processor System,U.S.Patent,3.713.107(1973).

(5) 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,pp553-563(1980).

(6) 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).

(7) 安浦，高木：並列計数法による高速ソーティング回路，信学論(D),Vol.J65-D,No.2,pp.179-186(1982).

(8) 喜連川，伏見，桑原，田中，元岡：パイプラインマージソータの構成，信学論(D),Vol.J66-D,No.8,pp.332-339(1983).

(9) Tanaka,Y.,Kozaka,Y. and Masuyama,A.:Pipeline Searching and Sorting Modules as Components of a Data Flow Database Computer,IFIP 80,pp.427-432(1980).

(10)上肥：大容量ファイルを整列するシストリック・ソータ，信学論(D),Vol.J67-D,No.3,pp.281-288(1984).

(11)Todd,S.:Algorithm and Hardware for a Merge Sort Using Multiple Processors,IBM J.RES.DEVELOP.,Vol.22,No.5,pp.509-517(1978)