

データベースマシンの一方式

ON A DATABASE MACHINE BY MEANS OF INFORMATION SPACE

大久保 明伸 打浪 清一 手塚 慶一

AKINOBU OHKUBO*, SEIICHI UCHINAMI⁺ AND YOSHIKAZU TEZUKA⁺

宇部 高专 大阪 大学 工学 部

UBE TECHNICAL COLLEGE* AND FACULTY OF ENGINEERING, OSAKA UNIVERSITY⁺

1. まえがき

現在広く利用されている電子計算機は基本的には数値計算向き構造をしていて、一度に多量のデータを処理するデータベースシステムに対して十分な性能、構造を有しているとはいえない。データ処理の特徴はある種の構造をもったデータ群に対してある操作をほどこし、それらのデータ群の部分集合を選択するところにある。これらの操作の一部は、並列処理化できるはずである。しかし現在ではこれらの操作のほとんどはソフトウェアで実現されている場合が多い。また最近ではデータベースの巨大化や高水準化の要求に対してソフトウェアのみでは十分にこれらの要求に対処することが困難になりつつある。また数値計算の分野でも並列処理化の要求がある。

他方、LSI技術の進歩により小型で安価なマイクロプロセッサ、磁気バブルメモリやCCDメモリ等の登場により多数のCPUと大容量メモリを結合したコンピュータシステムが実現可能になりつつある。このような現状の中で従来のソフトウェアの一部を直接ハードウェア化したデータベースマシンの提案がさかんになりつつある。これらのデータベースマシンは、多くのマイクロプロセッサを使用して二次メモリに格納されているデータに対して並列にアクセスをほどこし、二次記憶装置とホストコンピュータとのデータ伝送量を減少させようとしたものである。一般に二次記憶装置⁽¹⁾としては磁気ドラム、磁気ディスクが用いられているが、最近発

表されたものには二次記憶装置として磁気バブルメモリ、CCDメモリや通常のランダムアクセスメモリを想定した提案もある。またジャイロソート⁽²⁾によられる技術を使用したファイルプロセッサも提案されている。他方、内容呼び出しが直接できる連想メモリは古くから多くの形式が発表されている。一部の特殊な用途の計算機には利用されているが、これらはかなり高価である。実際にデータを貯える二次メモリ(ディスクやドラム)をこれにすべておきかえることは非常に困難であると考えられる。したがって連想動作をマイクロプロセッサを用いてシミュレートすることにより解決しようとしてされている。次にデータベースマシンの基本的な性能についてまとめると次のようになる。

- (1) 多量のデータ処理が並列にできること。
- (2) 記憶の内容呼び出しができること。
- (3) (1), (2)を効率的実現するための記憶構造をきつこと。
- (4) データベースシステムのDDL, DMLとデータベースマシンのきつ基本命令との翻訳が容易であること。
- (5) データのSecurityが十分であること。

筆が考えられるがこれらの性能のうちどこまでをハードウェアで実現するかを検討する必要がある。

今回の報告では(1), (2), (3)を中心にした考察を行う。

2. データモデル

データモデルとしては今までに種々のモデルが提案されているが分類すると大体次のようになる。

- (1) 階層データモデル
- (2) ネットワークデータモデル
- (3) リレーショナルデータモデル

一次元的に配列された従来の記憶装置にデータを(1), (2)の構造にして貯えるためにはポイントが必要になる。このポイントは従来の記憶装置のもつ固有な記憶構造に柔軟性ををもたせるために考えられたものである。データはポイントを使用して記憶装置に格納してあるために、目的のデータを検索するのにこのポイントを次々にたどらなければならない。

したがってポイントで複雑に結合されたデータ構造をもつモデルを利用したデータベースマシンにおいて、データ検索の並列処理は実現が困難であると考えられる。

現在提案されているデータベースマシンのためのデータモデルはこれらの点を考慮して(3)のリレーショナルモデルを直接ファイルにインプリメントされたものが多い。

リレーショナルモデルはE. F. Coddにより提案されたものであるが、データベースを構成するデータの属性の関数関係により定義される第3正規型等の表を作り、それらの表の集合として構成されるものである。

リレーショナルモデルでは、データの検索はポイントをたどるかわりに「関係集合演算」とよばれる演算を表また表と表にほどこすことにより目的のデータを検索するものである。

多くのデータベースマシンはこの関係集合演算を多くのマイクロプロセッサを使用して演算の高速並列化処理をこころみている。

リレーショナルモデルでは第3正規型とよばれる表等に分解するため検索要求が表と表とにまたがる場合かなり複雑なアルゴリズムが必要になる。また表の大きさや各項目の長が一般に相違するために記憶使用量の増大や並列処理アルゴリズムを複雑にするに可能性がある。さらに第3正規型でも表現できない例も指摘さ

れている。

リレーショナルモデルではある属性の構成要素すなわち属性の具体値向の関係は全く同等のものとしている。しかし個体の属性値の組と個体向の関係が全く同列の関係としてとらえられているのは、意味を考えると構造が捉えられておらず処理が行いにくい。

我々の提案してきた情報空間モデルでは各個体に付属する属性値の組で一つのまとまった概念を表す。この各属性は空間の座標軸を構成し、具体値はその座標値を表現する。そこで各個体は一般に n 次元空間の点として表現される。また、個体向の属性向関係は空間の中に表現される。個体向につけられたよりハイレベルの関係は関係を表すハイレベルの空間に配置される。また個体をあらわす点は良く用いられるマトリックスになるので物理的にも格納しやすい。たとえば次のようになる。

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \cdots & \cdots & x_{ij} & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

x_{ij} は m 次元空間 j 番目の点で第 i 番目の座標である。

この空間内のデータの検索は要求のある属性軸への写像をほどこすことに得られる。情報は数値化されているため普通の行列演算により求めることができる。

この空間モデルを文献検索に使用して高い検索率をあげ得ることを確認している⁽⁴⁾⁽⁷⁾。

以下このデータモデルの定義を行う。

〔定義〕

- (1) データベースを構成するためのデータ集合 D_1, D_2, \dots, D_m があたえられたものとする。

D_i の構成要素をアトムとよぶ。

D_i は属性または定義域とよぶ。 D_i の構成要素は有限個であり順序付けられている。また D_i の構成要素には同じものはないものとする。但し D_i と D_j の構成要素は同じ要素からなってもかまわない。

(2) D_1, \dots, D_m を互に区別するために名前を付ける。これを A_1, \dots, A_m であらわし、 D_i の属性名 A_i という。

(3) 領域の構成要素すなわちアトムは名前や数値またはそれらの組合せである。(属性名自身も一つのアトムである。) アトムはそれぞれある座標軸の座標値をもっているが特別の場合として座標値 w とはそれぞれこの属性に対して未定義、無関係をあらわす。

(4) 集合 D_i を n 次元空間の第 i 番目の軸に対応付ける。この座標軸を属性という。

(5) 個体は n 次元空間の一員であらわされる。これを $P(x_1, x_2, \dots, x_n)$ であらわす。但し x_1, \dots, x_n は属性 A_1, A_2, \dots, A_n の座標値である。

(6) 点 $P(x_1, x_2, \dots, x_n)$ を他の点と区別するために少なくとも一個の属性軸があり、その属性を KEY 属性という。データの更新、削除はこの KEY を通して行なわれる。

以上のようにして構成された空間を情報空間とよぶ。またこの空間の個体の集合をファイルとよぶことがある。

実際にデータベースを設計する場合かなり高次元になるものと考えられるので属性群を適当なグループに分割し日部(無関係)をすてることにより次元の増加をおさえる必要がある。またある属性に対してくり返し属性がある場合はファイル中に同じ KEY がくり返しの回数だけあらわれることになる。ファイル中に同じ個体が幾度もあらわれるのを防ぐ手段としては次の二つの方法がある。第一の方法はくり返し属性自身どうしの直積を新しい軸として用いる。すなわちくり返し属性の構成要素の組み合わせたものを

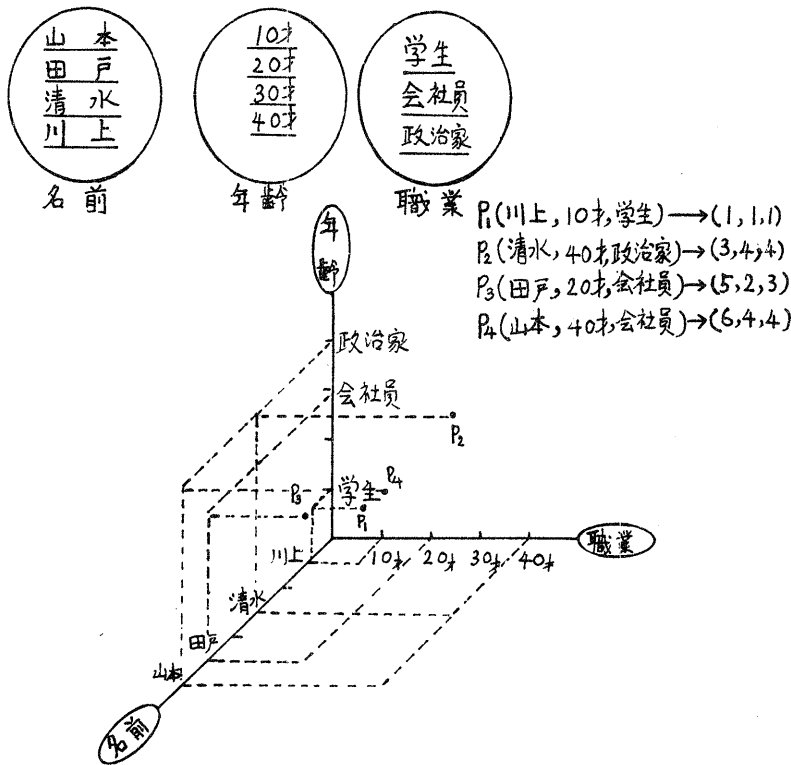


図-1 情報空間の例

新しいアトムとする属性軸と考える。この方法はくり返し属性が現われても次元の増加にはつながらぬ。ただし属性値を管理する部分には大きな負担がかかる。

第2の方法はくり返しの数だけ属性軸を新しくする方法である。この方法は各個体に対してくり返し回数が極端に相違する場合日数の増加につながらぬ。この二つの方法による検索効率の優劣は現在検討中である。

次に、空間の圧縮法には二通りの方法がある。前述したように属性A, Bの直積を一つの新しい軸としてA, B軸と置換する方法がある。あと一つはこの直積による圧縮法の変形として属性A, Bが作る作る平面の斜交軸への写像による空間圧縮法がかんがえられる。この例を図-2に示す。

次に空間に対するデータ検索の基本操作について述べる。ある検索要求に対して一つの空間の奥が決まる。この奥向奥の属性A_iの座標をa_iとすると、この検索質問は(a₁, a₂, ..., a_n)なるベクトルと考える。これに直交するベクトルを考える。このベクトル面に各奥を写像すると適合奥の集合はこの空間のある一奥に写像されるはずである。実際には写像された奥は検索要求以外のデータを写像されてしまうのでこの奥に写像された奥の中から必要な情報を撰択する必要がある。この操作をくり返せば必要なデータは基本的に求めることができる。

一般に上述したようにして作られた空間はある個体に関する空間である。

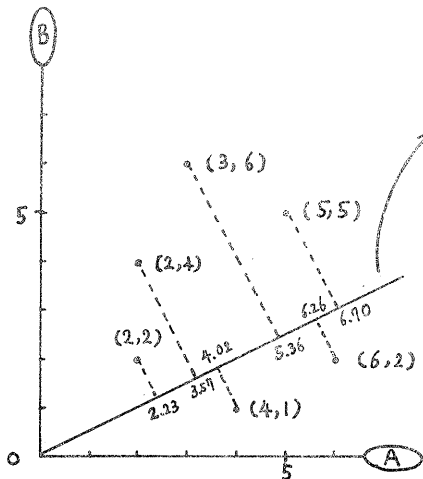
異なる空間の関係を表現するために個体関係空間を定義する必要がある。

ここで2つの個体空間A, Bを次のように関係づける。空間Aをこの関係の親とし空間Bを子とする。空間Aの奥を一意的に識別できる属性(個体識別名)をえらびかつ空間Bの奥を一意的に識別できる属性をえらぶ。この属性を座標軸にえらび空間を構成する。この空間に新しい名前をつけ、これを空間の関係名という。このようにして構成すれば個体を表す空間ができ、この個体空間の関係を表す空間ができる。これをくり返すことにより空間を階層的に構成することができる。

3. データベースマシンの構成

図-3にシステム全体の構成を示す。ユーザにより発せられた検索要求はホストコンピュータ(HP)によりデータベースマシンの命令へ翻訳され必要な情報とともにデータベースマシン内のコントロールプロセッサ(CP)に送られる。CPは磁気ディスク(MDR)の各ヘッドにヒリッかけてあるプロセッサエレメント(PE)に対して必要な指令を出す。ユーザによる検索要求はこのデータベースマシン内である程度処理されてホストコンピュータのメモリや外部メモリにおくられる。

図-3の奥線で囲まれた部分をデータベースマシンという。



$$\left(\frac{2}{\sqrt{5}}, \frac{1}{\sqrt{5}}\right) \begin{pmatrix} 2 & 2 & 4 & 3 & 6 & 5 \\ 2 & 4 & 1 & 6 & 2 & 5 \end{pmatrix} = (2.33, 3.57, 5.36, 6.26, 6.10)$$

図-2 空間の圧縮例

データベースマシンは次のブロックよりなり
たっている。

- (1) コントロールプロセッサ ---- CP
- (2) プロセッサエレメント ----- PE
- (3) 磁気ディスク装置 ----- MDU α
- (4) コントロール/データバス --- C/DB

以下各部の機能についてのべる。

(1) CP (コントロールプロセッサ)
CPは大きく3つの機能にわけられる。まずホストコンピュータ (HP) により翻訳されされたデータ操作命令を各PEに対する命令に分解し各PEのメモリへ転送する。第2にPEの応答を観察しながら各PEの起動、停止の制御を行う。第3に各PEのモードの設定を行う。CPはデータ操作が完了するとデータをHPへ伝送する。またCPのメモリにはデータベースを構成する属性やファイルの配置、Dロム、DM Lコンパイラ等が格納されている。

(2) PE (プロセッサエレメント)
PEは同一のものが何個ありCPのコントロール/データバス (C/DB) に並列に

接続されている。PEは個々の属性の管理を行う。またMDUの各ヘッドに接続されている。MDUの各トラックには個々の属性の座標がはいつている。

PEのROMにはデータ操作に必要なサブルーチン群が格納されている。

CPはPEのどのサブルーチンを起動するかをデータバスから指令を送る。このサブルーチン群は実際にはハード化する必要があるが現在のところマイクロプログラムで実現することを考慮中である。一まとまりのデータ操作が終了するとCPへ終了信号を送りCPは各プロセッサ (PE) の終了信号がすべて伝送された時点で次のデータ操作に移行する。さらにPEは相互連絡モードをもちPEと隣りのPEと直接データの伝送することができる。

(3) MDU (磁気ディスク)

MDUはHead-per-track方式の磁気ディスクである。個体又は関係の組はトラックに直角の方向に格納されている。

(4) C/DB (コントロールとデータバス)

CPとPEの間データ・制御信号の伝送に用いる。各ラインの動作は次章でのべる。

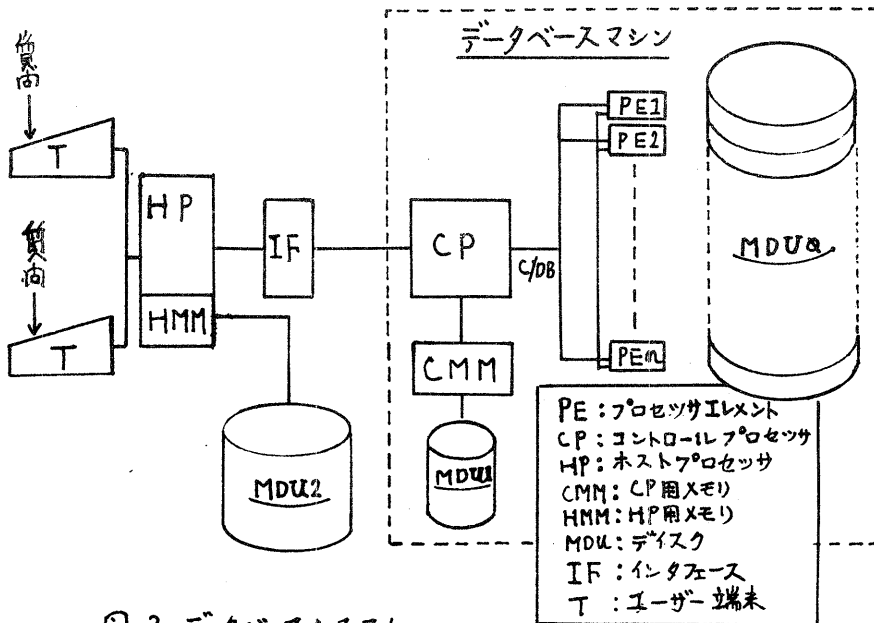
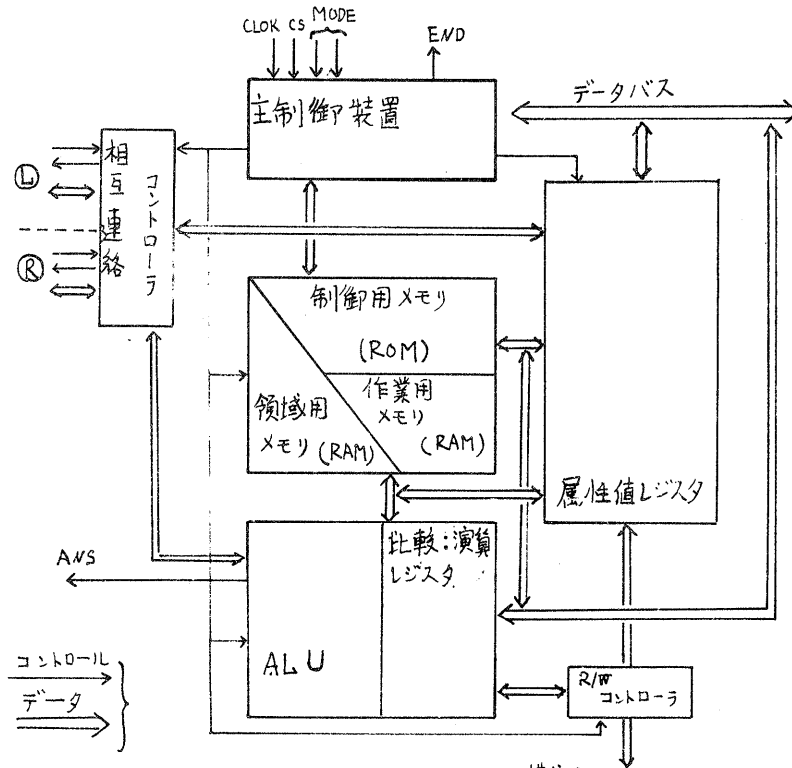


図1-3 データベースシステム

4. プロセッサエレメントの構成

プロセッサエレメントの概念図を図4に示す。PEは一種のマイクロコンピュータであり、算術論理、比較機能をもった演算装置がある。PEは原則としてCPの制御信号によって動作する。データベースマシンはSELECTION、PROJECTIION等の基本命令をモッテいる。この基本命令をCPは翻訳し各PEへ翻訳結果を伝える。PEはすでにROMに基本命令の翻訳結果を記憶してあるのでCPは各PEの担当する属性とデータ操作に必要な情報とPEが実行すべきプログラムの番地情報と伝送すればよい。これらの情報を送るときはPEをMODE 0(コマンドモード)にしておく。実行にうつるにはMODE 1(実行モード)にしてCPがCS信号(セレクト信号)を發すればPEは実行にうつる。基本命令の実行が終了するとPEはEND信号をCPに送るのでCPは次のコマ

ンドを送ればよい。図4で領域用メモリは属性の具体値が格納されている。属性値レジスタはMDUからのデータが格納されるレジスタで属性との関係はCPが管理する。CPの属性管理情報はCPからのコマンドにより伝えられるのでPEは作業用メモリにその情報を格納しておき実行のときに参照すればよい。MODE 2はPE間の相互間のデータのやりとりを行うデータの相互の交換、計算の実行はやはりあらかじめ伝送されたコマンド情報に基づいて行なわれる。PEがCPよりあるデータの判定を指示された場合、結果が真か偽かをANS端子より直接CPに伝送する。この信号によりCPは次に送るコマンドを選択することができる。またR/WコントローラはPEにより制御されている。このコントローラはトラック上のデータの読み書きばかりでなく前述したフラッグビットを各データに書き込み、消去する機能をもつ。これを利用してファイル全体に対する操作が可能になる。このフラッグはMODE 3によりトラク



MODE	意味
0	CPからコマンドを受け。 (コマンドモード)
1	PEはCS信号が受信されると命令を実行する
2	PE相互間に連絡ができてデータの交換ができる
3	MDU中のトラック上のフラッグビットエリアにビットをたてる

表1 モードの種類

図-4 PE(プロセッサエレメント)の構成図 MDU

上のフラッグビットエリアに書きこまれる。

信号をCPに送る。(HALT)

5. 基本演算

以下4.にのべたデータベースマシンのデータ操作に関する基本演算についてのべる。基本演算は種々考えられるがここではSELECT命令, PROJECTION命令を例に説明する。

一般に各PEが実行にはいるまえに, PEはCPより初期設定が行なわれる。基本演算は1つのファイルに対して行なわれる。実行が終了するとCPに終了信号を送る。PEはCPのCS信号により次の実行にはいる。以下SELECT, PROJECTION命令についてのべる。

(1) [SELECTION φ]

この基本演算は1つのレコードのある特定の属性に関して条件を設定し, その結果にたいして論理演算を行う。形式は次のようなものである。

SELECT φ [(A₁) op a₁ θ (A₂) op a₂ θ ... θ A_m op a_m]

ただし (A_i) は属性名 A_i の属性値, a_i は定数, op は等号または不等号, θ は論理演算子である。

この命令のアルゴリズムは次のようになる。

STEP φ: (初期設定) CPは属性マップを参照して各PEの属性を知り, 各PEに具体値 a_i と op を送る。(MODE φ)

STEP 1: MODE 1 と CS 信号により各PEは a_i と MDU φ のデータと比較する。(A_i) op a_i の結果を CP ANSWER LINE を通して送る。

STEP 2: CPで論理判定を行い, 該当するPEにMODE 3を指示する。(ディスクのトラック上のフラッグエリアにビットを立てる処理済)

STEP 3: Fileが終了するまでSTEP 1にJUMPする。
Fileが終了するとEND

(2) [PROJECTION]

この演算はある面 (a₁, a₂, ..., a_n) に対して, 各点 (B₁, B₂, ..., B_n) の射影をとるものである。

PROJECTION [(B₁, B₂, ..., B_n) / (a₁, ..., a_n)]

これは行列演算

$$(a_1 a_2 \dots a_n) \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & & \vdots \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & & b_{nm} \end{pmatrix}$$

を行なえばよい。

(a₁, ..., a_n) を検索射影面という。

アルゴリズムは次のようになる。

STEP α: CPは属性マップを考慮して各PEの属性を知り, 各PEに送る。

STEP 1: MODE 1 と CS 信号によりPEはMDU αより演算レジスタに送り a_i × b_i を行う

STEP 2: PEをMODE 2にしてPEの合計をとる。

STEP 3: ファイルが終了するまでSTEP 1にJUMPする。
終了したらEND信号をCPに伝送する。

次にこの2つの演算の所用時間を計算するために次のように各定数を定める。

N... PEの総数, t₀: PEを初期化する回数,

t⁺... PEで加算に必要な時間, T: ディスクの回転時間

t^x... 積の計算に必要な時間, B: バッファの容量,

t_{DB}... DISK から バッファへの転送時間, n: 処理するレコード数,

t_e... バッファから演算器への転送時間, R: 1レコード中の属性の数,

t(k) ... k個のデータを並列に合計をとるために必要な時間, M: トラックのプロック数,

以上の記号を用いて大体の計算時間は次の表2に示す。

	SELECT	PROJECTION
$k \leq N$	$t_0 + \frac{T}{2}$	$(S + \frac{T}{2})M$
$k > N$	$\left[\frac{R}{N}\right](t_0 + \frac{T}{2})$	$\left[\frac{R}{N}\right](S + \frac{T}{2}) + S'$

表2 基本命令処理時間

但し S, S' は

$$S = B(t_{0B} + t_e + t^x + t(k) + t^t \log_2 k)$$

$$S' = \left[\frac{R}{N}\right] \times t_{cPT} + \left(\left[\frac{R}{N}\right] - 1\right) t_{cPA}$$

t_{cPT} は PE より CP への伝送時間。

t_{cPA} は PE での加算の時間である。

以上、二つの基本演算を例にあげたが、基本演算として必要なものは他にも考えられる。ある演算をほどこして、目的のレコードが選択された場合、すべての属性値が重要な場合よりもある特定の属性値だけを選択できる命令が必要である。またディスクとバッファまたはホストコンピュータのメインメモリとバッファ間の転送命令等が必要になる。またデータの検索が相異なるファイルにまたがるような場合の操作命令も必要になる。

6. おまじい

データベースを構成するデータを多次元空間の中に配列し、各属性のファムに順位を付した構造のデータベースマシンを提案しその空間の構成法、物理的な実現可能性および基本的な操作命令について考察した。

データを空間に写像することによりデータ構造は単なるマトリックスとして表現でき基本演算として写像の計算が非常に簡単に実行できることをしめした。また座標にデータをわりふる場合ある属性の中でデータにならうかの意味に関して順位づけを行うことにより情報の意味的なとりあつかいができる。

本論文でのべたデータ構造は本質的には階層的なファイルをもつことになるが他方リレーショナルデータベースを実現することも可能であると考えられる。

また PE は並列処理の高速化のためにはハードウェアで実現することがのぞましいがマイクロプログラム可能なプロセッサでも十分可能であるう。

文 献

- (1) 田中, 田畑, 津田: マルチマイクロプロセッサシステム HARPS. 信学会 EC 研資 (1977-1)
- (2) 植村, 弓場他: 磁気バブルによるデータベースマシンの構想. 信学会 EC 研資 (1977-1)
- (3) 角澤, 土肥: データベースマシン向きのデータ処理方式. 信学論 D 52-613
- (4) M. EDELBERG, L.R. SCHISSLER: Intelligent memory, Proc. of NCC, 1976
- (5) 嶋本, 富永: ルーフ構造としたファイルメモリの分布制御方式について. 昭52年信学会 大 1407
- (6) 嶋本, 打志, 千塚: 意味空間を用いた信学論 D 昭47-5
- (7) 嶋本, 打志, 千塚: 意味空間を用いた主題分析法. 情報処理 1973-5