CA 25－3
(1977. 3. 18)

"リレーショナルデータ・ベースのハードウェア化について"

ON HARDWARE IMPLEMENTATION OF RELATIONAL DATA BASE

渥美 幸雄，佐藤 聖，小花 貞夫，南部 明
Yukio ATSUMI Satoshi SATO Sadao OBANA Akira NANBU
高田 健治，所 眞理雄，相磯 秀夫
Kenji TAKADA Mario TOKORO Hideo AISO
慶応義塾大学工学部
KEIO UNIVERSITY FACULTY OF ENGINEERING

SUMMARY

This paper is concerned with hardware implementation of data base system. Discussions are mainly put on i) system configuration, ii) high-speed data base operations, and iii)requirement of access control. A system is proposed which adopts relational data model. The system is composed of several functional modules, i.e., DIM (Data base Interface Module), COM(COmpiling Module), ACM(Access Control Module), and REAM(RElational Algebra execute Module). REAM carries out intra-domain parallel processing, consisting of 3.2MB CCD (or Magnetic Bubble) memory and 100 processors. Performance of REAM is estimated and the feasibility of the system is described.

1. 序論

近年，いろいろな方式のデータ・ベース・マシン（バック・エンド方式のXDMS[14]，ロジック・パー・トラック方式のRAP[10]，RARES[11]，CASSM[9]，及び電総研[16]，横浜国大[17]のシステム等）が研究されている。しかし，これらはデータ・ベース管理システムとして全体的に見ると，まだ数多くの問題点が残されている。我々はこれらの問題点のいくつかを解決すべく，ひとつのデータ・ベース・マシンを提案する。その特徴は

(1) マルチ・プロセッサによるデータ・ベース管理システムの機能分散
(2) イントラ・ドメインの並列処理
(3) アクセス・コントロールの強化

である。以下，2節はデータ・ベースのハードウェア化への必要性について述べ，3節は我々のシステム構成を概略する。4節ではアクセス・コントロールの実現方法について記述し，5節ではデータ集合演算[1]を基本オペレーションとする実行モジュールについて述べ，最後に評価と検討を行なう。

2. データ・ベースのハードウェア化

一般にデータ・ベース管理システムは，データ記述言語（DDL），データ操作言語（DML），及びアクセス法の3つの要素から構成される。特にcoddによって提唱されたリレーショナル・データ・モデルに基づく場合，データ・ベース管理システムのDDLは，ユーザあるいはデータ・ベース管理者によるリレーションの作成，及び削除，ドメインの追加，及び削除，ユーザ・ビューの定義，それにデータの保護に関する指定を行なう言語である。またDMLは実際にユーザがデータ・ベースにアクセスするための言語である。そしてアクセス法は論理データ構造と物理データ構造の橋渡しとなる。ところで，このデータ・ベース管理システムを汎用計算機で実現しようとすると，次のような問

題が起こる。

(1)汎用計算機のOSのもとでデータ・ベース管理システムが動作しており，CPUの負荷が大きい。

(2)データの検索，更新，削除，及び挿入の応答時間が長い。

そこで，これらの問題に対処するためにいくつかのシステムが研究されている。

まず問題(1)に対しては，ベル研究所のXDMSがあり，これはデータ・ベース管理機能を専用の計算機に委ねている。また問題(2)に対してはRAP，RARES，CASSM，電総研，及び横浜国大のシステム等があり，これらはデータを多数のプロセッサによって並列処理し，その上極力データ転送によるオーバ・ヘッドを少なくしている。しかし，RAP，RARES，CASSM等のシステムは，データの検索などの基本的な処理しか行なっておらず，その他のデータ・ベース管理機能は，まだ親計算機に依存しており，その負荷が大きいと考える。

そこで我々は，データ・ベース・システムを親計算機から完全に分離し，さらに，このデータ・ベース・システムをDMLのコンパイリング，アクセス・コントロール等に機能分散し，それぞれにプロセッサを割り当てた。リレーショナル・データ・ベースにおいて並列処理，及び連想処理を行なう場合，トゥプル単位とドメイン単位の処理形体がある。さらにドメインに関しては，インター・ドメインとイントラ・ドメインの方法がある。ここでハードウェア・レベルでの基本演算としてリレーショナル・アルジェブラ[2]を考えた時，イントラ・ドメインの場合に最大の並列性があると考えられるため，この処理形体を採用した。

3. システム構成

本システムは，リレーショナル・データ・モデルの管理機能を次の4つのモジュールに分散した。(Fig-1)

(1)データ・ベース・インターフェイス・モジュール (Data Base Interface Module: DIM)

(2)コンパイリング・モジュール (Compiling Module: COM)

(3)アクセス・コントロール・モジュール (Access Control Module: ACM)

(4)リレーショナル・アルジェブラ・エグゼキュート・モジュール (Relational

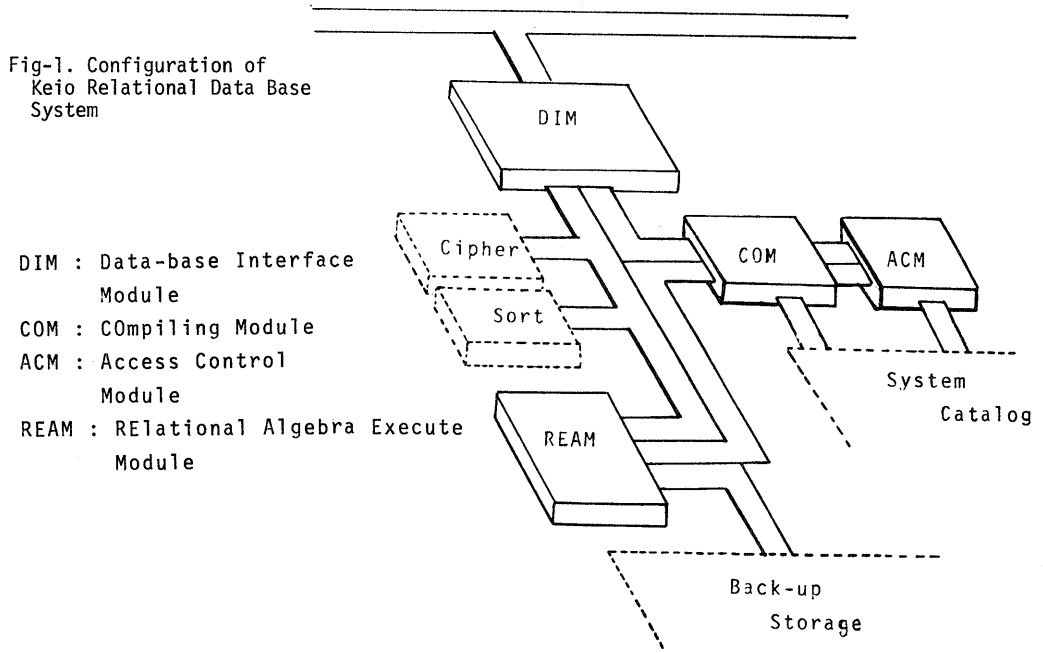


Fig-1. Configuration of Keio Relational Data Base System

DIM : Data-base Interface Module
COM : COmpiling Module
ACM : Access Control Module
REAM : RElational Algebra Execute Module

Algebra Execute Module: REAM)

3.1 データ・ベース・インターフェイス・モジュール（DIM）

DIMは汎用計算機（群）とデータ・ベース管理システムとのインターフェイスである。このモジュールの機能は，汎用計算機からユーザの問合せ（これには，データ記述言語，データ操作言語，及びデータ・ベース・システム・コマンドがある。）の受付けを行ない，問合せの処理の結果を該当ユーザに渡すことである。

3.2 コンパイリング・モジュール（COM）

COMの機能は，ユーザのデータ操作言語をREAMが実行可能な言語，すなわちリレーショナル・アルジェブラにコンパイルすることである。この機能を果たすために2つの手続きがある。ひとつはアクセス・チェックに関する情報を作成することで，もうひとつは問合せの最適化である。以下これらについて説明する。

(1) アクセス・チェック情報の作成

本システムのデータ操作言語においては，ユーザ・ビューとデータ・ベース・マクロ（これらに関する情報はCOMが管理する。）の定義及び引用が許されており，もしユーザの問合せの内部でデータ・ベース・マクロが引用されている場合には，それを展開する。次にACMでのアクセス・チェックのための情報として，その問合せ内部（ユーザ・ビューが引用されている場合には，その内部も含む）で使われているリレーション名，ドメイン名，及びそれらのドメインに対する操作（read,update,insert,delete）をリスト・アップする。

(2) 問合せの最適化[6]

ユーザの定義したビューとユーザの問合せ内の述語，さらにACMが作成するデータ保護のための追加情報（ドメイン，トゥプルに関する制限）とを結合し，リレーショナル・アルジェブラ・オペレータから成るオペレータ・ツリーを作成する。そして，このオペレータ・ツリーに対して次の2つの最適化が行なわれる。

(i) 同一リレーションに対するrestrictionを複合ブール・オペレータで置き換える。（Fig-2） これにより，同一リレーションに対するアクセス回数を減らすことができる。

(ii) unaryオペレータ（projection,restriction）を可能な限りオペレータ・ツリーの下方へ納すことにより，上方のオペレータに渡されるリレーションの規模を小さくする。（Fig-3） これにより二次記憶とのアクセス回数及び主記憶内での処理量を減らすことができる。

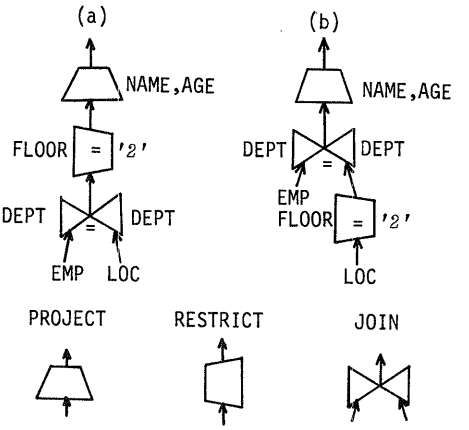


Fig-2. (a) The tree for a relational algebra query. (b) The tree for a transformation of the query in (a).

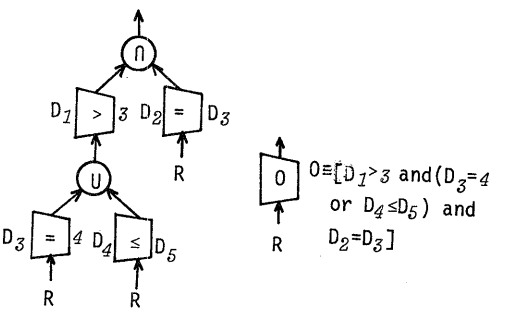


Fig-3. (a) Set operations on RESTRICTs of the same relation R. (b) Representation of the tree in (a) as a COMPOUND RESTRICT.

最後に，この最適化されたオペレータ・ツリーより，REAMが実行可能なリレーショナル・アルジェブラのシーケンスを発生する。

3.3 アクセス・コントロール・モジュール (ACM)

ACMは次の機能を持つ。

(1)ユーザのデータ操作に対するアクセス・チェック
(2)データ記述言語のコンパイリング
(3)ディレクトリの管理
(4)データ保護に関する情報の管理

COMによって作成されたユーザ識別子，パス・ワード，リレーション名，ドメイン名，及びドメインへのオペレーションを組にした情報をシステム内のアクセス・コントロール情報と照合し，アクセスの正当性をチェックする。

データ記述言語には，リレーションの作成，削除，ドメインの追加，削除，及びデータの保護記述がある。 前者はリレーション・ディレクトリ，ドメイン・ディレクトリを操作し，REAMのデータ記述フォーマットを作成する。 後者はプロテクション・プレディケイト・テーブル，ドメイン・アクセス・マトリクス，ドメイン・ディレクトリに必要事項を記述する。

尚ディレクトリは一般のユーザのリレーションとデータ・ベース管理者のリレーションに区別されている。 一般ユーザのリレーションを共有化する場合には，データ・ベース管理者が再登録する。

3.4 リレーショナル・アルジェブラ・エグゼキュート・モジュール (REAM)

REAMはデータの検索，削除，挿入，更新，及び述語レベルでの簡単な四則演算を行なうハードウェア・モジュールである。 基本オペレーションは，リレーショナル・アルジェブラである。 データの情報はイントラ・ドメインの並列及び連想処理方式で，ドメインのデータ・アイテムの長さは固定とする。 これはハードウェアの処理効率を向上させるためで，バリュ・コーディング・ハードウェアを使用する。 データを並列及び連想処理するためのワーク・エリアが限定されているため，バック・アップ・ストレージとのデータ転送が必要となる。 しかしダブル・バッファリング及びプリ・ページング方式を採用することで，データ転送によるオーバ・ヘッドが小さくなる。

データ定義に関しては，コード化されたリレーション名，ドメイン名，ドメイン・タイプ等の情報を基に，バック・アップ・ストレージに必要な処理を行なう。

4. アクセス・コントロールの強化

データ・ベース管理システムは，データの冗長を可能な限り少なくし，データの集中管理によってデータの共有を許している。 従って，データの安全，及び保全はデータ・ベース管理システムの重要な機能である。 しかし汎用計算機では，ユーザの問合せの応答時間を短かくしようとすることと，厳密なアクセス・コントロールを行ないたいという要求は，CPUの処理能力から考えて相容れないものである。 そこで我々は問合せのコンパイリングとアクセス・コントロールを並列化することにより，問合せ全体のスループットの低下に対処する。 ACMは1台のプロセッサとメモリを持ち，さらにディレクトリやデータの保護に関する情報を蓄えるための専用ディスクを持っている。

4.1 データ・ベースの保護レベル

データ・ベースの保護の対象は，データ構造，即ちデータ・モデルによって異なる。 CODASYLのデータ・モデルは保護のレベルが細部に渡り，保護手続きが複雑である。 またデータ独立が弱いため，構造の変更が行なわれると，その保護指定も変更しなければならず，好ましくない。 その点，リレーショ

ナル・データ・モデルは，データ構造が単純であり，その保護レベルは

(1) リレーション：R
(2) ドメイン：D
(3) ドメインの部分集合：D_{sub}

である。 (1)はリレーションRを単位とし，(2)はRの各ドメインDを単位として，ユーザのオペレーションを制限する。(3)はデータ・セマンティクスから考えて，さらにリレーションのトゥプルを制限する。 (2)，(3)はデータ・ベースに固有な保護レベルである。

4.2 データ・ベースの一貫性と保全

データ・ベースでは，データの更新，挿入，削除によってその信頼性が損なわれてはならない。 リレーショナル・データ・モデルにおいて，ドメインD_iが，リレーションRのキー・ドメインであり，さらにリレーションR'のノン・キー・ドメインである場合，キー・ドメインの値が更新，又は削除された時，ノン・キー・ドメインの値も更新，又は削除されないとデータ・ベースの一貫性が失なわれる。また同一リレーション内のドメインD_j D_kの間に強い相互関係がある場合も一貫性を指定する必要がある。

あるドメインDの更新されたデータや，Dに新たに挿入されたデータは，そのドメインの属性からある値の範囲に入らなくてはならない。 また，ドメインDがキー・ドメインなら，その値はユニークでなくてはならないといったデータの保全を指定する必要がある。

```
EX1
Group#:User#:Password IS PERMITTED READ OF EMPLOYEE.ENO,ENAME,DEPTNO
                      WHERE EMPLOYEE.DEPTNO=5
EX2
INTEGRITY EMPLOYEE.SAL>0
```

Fig-4. Example of Protection Language

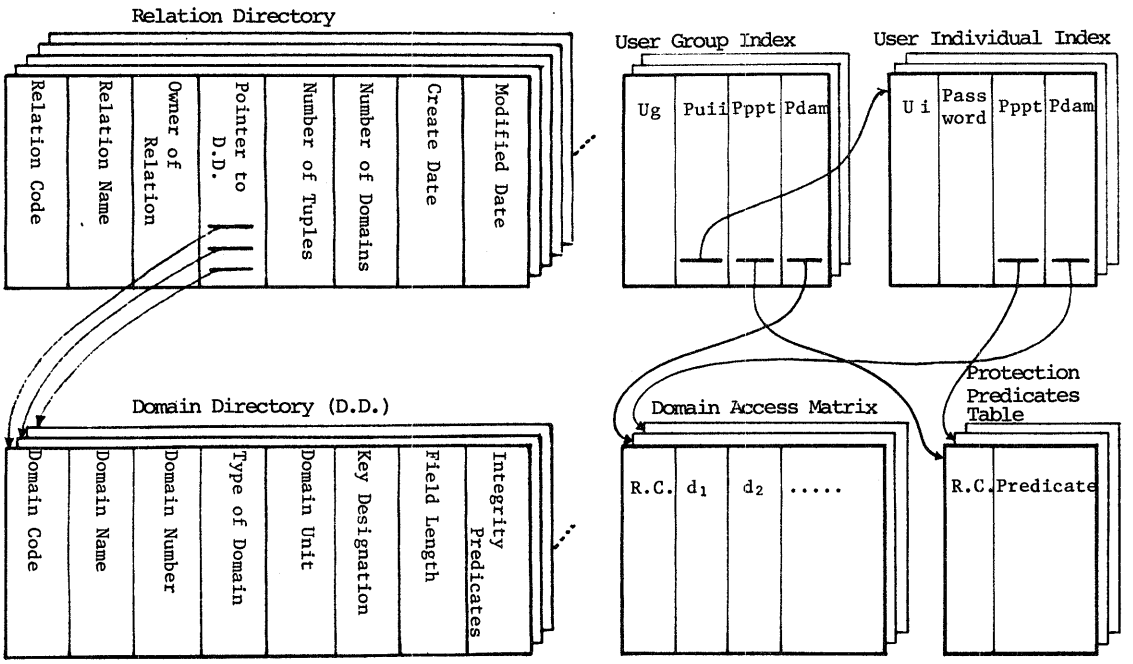


Fig-5. System Catalogue

4.3 保護言語

データ・ベースの保護は，データ・ベースの作成者，あるいはデータ・ベース管理者が保護言語を使って指定する。 保護言語は次の機能が要求される。

(1)データ作成者を認識
（データ作成後，リレーションの修正が，そのリレーションの作成者自身によって行なわれているかどうかを認識する。）

(2)対象となるユーザの指定
（ユーザは，所属するグループ識別子と個人識別子に分けられ，複数のグループに属することも可能。）

(3)保護レベルの指定
（4.1節の各保護レベルを指定でき，Dsubは述語で記述できる。）

(4)各保護レベルに対するユーザのオペレーションを指定
（オペレーションとしては，read,update,insert,deleteが指定可能。）

4.4 システム・カタログ

データ・ベースの保護等に必要な情報として，Fig-5の情報を考えた。 保護述語は保護レベル(3)の指定を記述した述語が，また保全述語はドメインに関する保全指定を記述した述語が，いずれもそのままの形式で蓄えられている。 ドメイン・アクセス・マトリクスは，アクセス・マトリクスとケイパビリティ・リストの混合した構造を持ち，ビット・マップ方式でオペレーションを記述している。 専用ディスクへのI/Oを減ずるため，ACMのメモリ内に最近使用された何人かのアクセス情報をLRUアルゴリズムで保持する。

5. イントラ・ドメインの並列処理

REAMではCOMとのインターフェイスであるリレーショナルアルジェブラが，直接的に，しかも効果的に処理されるように，並列化及び記憶構造の単純化がなされている。 それは，

(1)タグアーキテクチャ連想プロセッサによるイントラ・ドメインの並列処理

(2)二次記憶上でデータ・ベースは，n-aryリレーションの各ドメインごとにブロック化されている。（トゥプルはドメインの物理的位置で認識。）

5.1 REAMの構成

Fig-6に示すように，REAMのハードウェアの主要構成要素は，

(1)REAM制御ユニット（RC）
(2)I/O制御ユニット（IOC）
(3)ループ制御ユニット（LC）
(4)セル（100ケ）

である。 セルはメモリと処理ロジックから成っている。 メモリはメジャ/マイナ・ループ方式のCCDあるいは磁気バブルを，8ビット並列に読み書きできるように構成したものである。 処理ロジックは，8ビット・マイクロプロセッサ及び付加ロジック（RAMを含む）から成る。 REAM制御ユニットはREAM全体の制御を行なうもので，他のモジュール（COM，DIM）とのインターフェイスとなる。 I/O制御ユニットはREAMの処理能力拡大のための仮想記憶システムを実現する。 ループ制御ユニットは，セル内のマイナ・ループの管理を行なう。 アクティブ用とバッファ用のループの制御は，別々のコントローラが行なう。 以下，主要構成要素について説明する。

(1) REAM制御ユニット（RC）

RCは，COMから送られてくるリレーショナル・アルジェブラを解釈し，制御信号を各セルへ送る。 尚リレーショナル・アルジェブラのファームウェアは，RCにあり，セルは集中管理されている。 またCOMから送られてくる問合せを蓄えるバッファがあり，次の問合せの処理に必要なリレーション，及びドメインを先読みして，I/O制御ユニットに知らせる。 リンク・テーブルの作成，保持もRCが行なう。

(2) I/O制御ユニット(IOC)

IOCはREAMと二次記憶とのデータの転送を制御するものであり，RCから送られてくるプリ・ページングのための情報を基に，次に必要となるリレーションおよびドメインを二次記憶からバッファ・ループへ転送する。

(3) ループ制御ユニット(LC)

LCはセル内のマイナ・ループを一括して制御するユニットである。各セルの同じアドレスのマイナ・ループには，同一のドメインに属するデータがはいっており，LCは128ケのループ・アドレスを制御する。尚LCはアクティブ・ループ用とバッファ・ループ用の2つがある。

(4) セルの構成

REAMには，同一構造のセルが100ケある。各セルのメモリ部分は，2つのメジャ・ループ(32kB×2)から成り，交互にアクティブ・メモリとして動作する。また各メジャ・ループは，256バイトのマイナ・ループを128ケ持つ。セル内におけるデータは，セルC_i ($1 \le i \le 100$)，各セルのマイナ・ループm_j ($1 \le j \le 128$)，ドメインD_kのnケのデータ・アイテムd_{kl} (リレーションのk番目のドメインのl番目のデータ・アイテム，ただし$n \ge 100$)とすると，$(C_1, m_1) \leftarrow d_{1,1}$，$(C_2, m_1) \leftarrow d_{1,2}$ ---- $(C_{100}, m_1) \leftarrow d_{1,100}$，$(C_1, m_1) \leftarrow d_{1,101}$ ---- $(C_i, m_j) \leftarrow d_{1,n}$，次に$(C_1, m_{j+1}) \leftarrow d_{2,1}$，$(C_2, m_{j+1}) \leftarrow d_{2,2}$ ----のように記憶される。従って，一つのドメインについて，100台のプロセッサによる並列処理が可能となる。また一つのセル内にトウプルが構成される。

REAMでは，検索結果の保存，多重条件処理および出力のため，一種のタグ情報としてマーク・ビットを個々のデータに持たせる。マーク・ビットはセルのマーク・ビット用RAMにある。

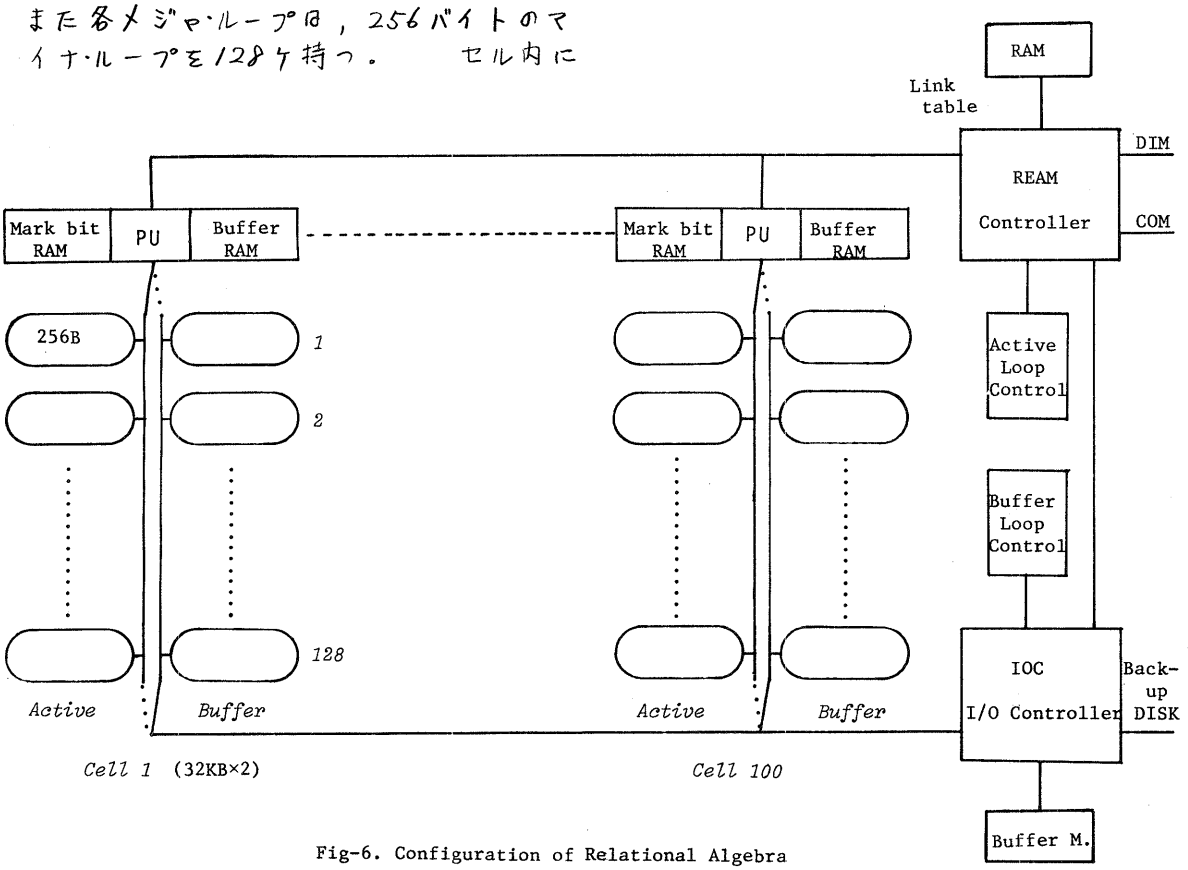


Fig-6. Configuration of Relational Algebra Execute Module

5.2 リレーショナル・アルジェブラの実行

REAMでは，リレーショナル・アルジェブラ・オペレーションのうち restriction, projection をマーク・ビットの操作で行なう。 join については，リンク・テーブル方式により処理の効率を上げている。その他のオペレーション division,…… についても join と同様，リンク・テーブルを用いて処理する。 ここでは，リンク・テーブル方式による join 操作について説明する。

リンク・テーブル方式は，各トゥプルの記憶されている順番，つまり位置をtid（トゥプル識別子）として，joinすべきリレーションを，tid を使ってリンクづけするものである。 Fig-7 におけるリレーションR とSの join を考え，ジョイニング・ドメインを d_3 とする時，あらかじめ RとSのジョイニング・ドメイン d_3 について積集合をとり，tidのリンク・テーブルを作成する。 これは，RC によって管理される。 join 結果を出力する時は，必要なドメインをRAMバッファに取り出し，リンク・テーブルの tidを使って出力する。 以上の方法により，ループへのアクセスが最小限になるため，処理効率が上がる。

REAMでの join処理時間の簡易的な評価式を示す。

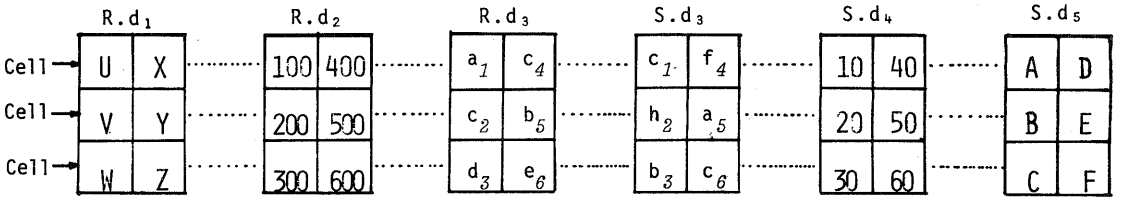
M_R : リレーション R の全トゥプル数
M_R' : リレーション R の選択されたトゥプル数

R

R.d_1	R.d_2	R.d_3
U	100	a
V	200	c
W	300	d
X	400	c
Y	500	b
Z	600	e

S

S.d_3	S.d_4	S.d_5
c	10	A
h	20	B
b	30	C
f	40	D
a	50	E
c	60	F



italic は tid

R [d_3 = d_3] S Link table

R tid	S tid
1	*5*
2	*1*, *6*
3	*1*, *6*
4	*3*

d_1, d_2, d_3 は Rのtid を，また d_4, d_5 は Sのtid を Link table の段から順に取出す

d_1	d_2	d_3	d_4	d_5
U	100	a	50	E
V	200	c	10	A
V	200	c	60	F
X	400	c	10	A
X	400	c	60	F
Y	500	e	30	C

Fig-7. Link table 方式

M_S：リレーションSの全トゥプル数
M_S'：リレーションSの選択されたトゥプル数
T_r：メジャ・ループの一回転時間（32ms）

メジャ・ループは，128ケのマイナ・ループから構成されているから，マイナ・ループ一回転の時間は，$T_r/128$（250μs）である。

リレーションR，Sのjoinの全処理時間T_jは，次の式で表わされる。

T_j = （Link table の作成時間）
+（リレーション Rの出力時間）
+（リレーション Sの出力時間）

（第一項）
マイナ・ループ当り，Nケのドメイン・アイテムが記憶されているものとすると，並列に動作するマイナ・ループの数は，リレーション RとSではそれぞれ

$\lceil M_R/(N*100) \rceil$ ，$\lceil M_S/(N*100) \rceil$

となる。マイナ・ループのアクセス時間と tidの転送時間をaで考慮すると，第一項は，

$M_R'*(T_r/128)*(1+a)*(\lceil M_R/(N*100) \rceil + \lceil M_S/(N*100) \rceil)$

（第二項）
出力されるドメインの数をN_R，ドメインのひとつのデータをセル・バッファから読み出すのに b(ms)，マイナ・ループのアクセス時間をcで考慮すると，

$N_R*((T_r/128)*\lceil M_R/(N*100) \rceil *(1+c) + b*M_R')$

（第三項）
出力されるドメインの数をN_Sとすると，第二項と同様に，

$N_S*((T_r/128)*\lceil M_S/(N*100) \rceil *(1+c) + b*M_S')$

6. 評価及び検討

データ・ベース管理システムを汎用計算機のOSから完全に分離することにより，汎用計算機の負荷を軽減することができる。さらにその機能を分散することで，次のような利点が得られた。
(1)外部のシステムとのインターフェイスをDIMが受け持つので，いろいろなユーザのデータ・ベースの共有に対処できる。(2)問合せのコンパイリングとアクセス・コントロールにそれぞれプロセッサを割当て，並列化することで，アクセス・コントロールを強化できる。
(3)REAMにおけるデータ・ベースの並列及び連想処理は，汎用計算機の場合の転置ファイルやインデックス，そして数種類のアクセス法などを必要とせず，効率の良い処理が行なえる。
(4)REAMとCOMのインターフェイスをリレーショナル・アルジェブラに設定することで，SEQUELのコンパイリングが容易である。また，その他の高水準な問合せ言語も，その構文解析部分を交換することで実現できる。

しかし，このシステムはまだいくつかの検討すべき点がある。まず，REAMは一度にただひとつの問合せしか処理できず，データ・ベースのコンカレント処理の問題を残している。また，リレーショナル・アルジェブラを並列及び連想処理によって実現しているが，データをソーティングしてから処理する方式，即ち，特別にソーティング・ハードウェアを付加する方式も現在検討中である。そして，問合せのコンパイリングとアクセス・コントロールの負荷配分が適当であるかどうか，実際にソフトウェアを作成し，COM，ACM内のバッファ容量を検討しなければならない。ACMはシステム・カタログを専用ディスク（これは通常の4MB程度のディスクを想定）に蓄え，必要に応じI/Oを出すのであるが，REAMの問合せ処理が極端に速くなっ

てしまった場合には，専用ディスクをCCD等を用いた連想処理デバイスに変更することが考えられる。データの保護に関して，リレーションのドメイン単位，さらにドメインの部分集合を保護できるが，データ・ベースのaggregate function(average,sum,max,min,count)は，ドメインあるいは，ドメインの部分集合に対して作用される以外に，実際にはユーザに許されていない部分のデータを含めてaggregate functionを作用できる必要があり，検討しなくてはならない。

7. おわりに

以上，我々のデータ・ベース・システムについて，そのシステム構成を述べてきたが，さらに検討すべき点を残している。

最後に，本研究を進めるにあたって，日本ユニバック総合研究所(株)小林功武取締役，千葉恭弘主任，電子技術総合研究所 植村俊亮主任研究官，横浜国大 有澤博氏，そして情報処理学会データ・ベース研究委員会の諸氏には，貴重な御意見を頂き深謝します。

REFERENCE

[1] E.F.Codd "A relational model of data for large shared data banks" Comm. ACM 13,6,June,1970

[2] E.F.Codd "Relational completeness of data base sublanguages" In current Computer Science Symposia,vol.6 Data Base Systems

[3] M.M.Astrahan,et al. "System-R:Relational approach to data base" ACM-TODS 1,2, June,1976

[4] G.D.Held,M.R.Stonebraker,E.Wong "INGRES:A relational data base System" NCC,1975

[5] D.D.Chamberlin,R.F.Boyce SEQUEL:A structured english query language" Workshop on data description, access and control,ACM-SIGFIDET,May,1974

[6] J.M.Smith,P.Y.Chang "Optimizing the performance of a relational algebra data base inteface" Comm.ACM 18,10,October,1975

[7] M.Stonebraker "Implementation of integrity constraints and views by query modification" Proc.ACM-SIGMOD International conference on management of data,May,1975

[8] B.W.Lampson "Dynamic protection structures" FJCC,1969

[9] S.Y.W.Su,et al. "CASSM:A Cellular System for Very Large Data Bases" International Conference on Very Large Data Bases,1975

[10] E.A.Ozkarahan,et al. "RAP:An Associative Processor for Data Base Management", NCC,1975

[11] C.S.Lin,et al. "The Design of a Rotating Associative Memory for Relational Database Applications" ACM TODS 1,1,1976

[12] H.R.Hartson,D.K.Hsiao "A semantic model for data base protection languages" Systems for large data bases North-Holland, September,1976

[13] D.K.Hsiao "Access control mechanisms for data security systems" Tutorial for DB protection and security, August, 1975

[14] R.H.Canaday,et al. "A Back-end Computer for Data Base Management" Comm. ACM 17,10, October, 1974

[15] 植村俊亮（電子技術総合研究所）〝データ・ベース・マシンのアーキテクチャ〟 昭51年 電気四学会連合大会

[16] 植村俊亮（電子技術総合研究所）〝磁気バブルによるデータ・ベース・マシンの構想〟 1977年1月 情報処理学会，CA研究会，資料24

[17] 有澤博（横浜国大）〝ハードウェア・ソートによるデータ・ベース基本演算の高速化について〟 1977年1月 情報処理学会 CA研究会，資料24