

## 解 説

## 3. ハードウェアから見た命令セットアーキテクチャ



## 3.5 専用計算機の命令セットアーキテクチャ†

吉住 誠 一†† 長 島 重 夫††  
金子 憲 二†† 松 島 整††

## 1. はじめに

情報処理の形態は、数値、図形、記号、音声など処理の対象となるデータの種類の、処理の目的により異なり、応用範囲の拡大とともに多様化してきている。偏微分方程式を解く数値処理とデータベース検索処理とでは、処理内容、形態が大きく異なる。応用の特性、処理内容に計算機アーキテクチャの設計を合わせることににより、その分野の効率的処理を実現しようという狙いで、各種専用計算機の研究開発が進められてきた。

その中には、部分的には優れていても、システム全体の効率の点から疑問があり、いまだ単なるペーパーマシンや実験マシンで止まっているものも多い。本稿では専用マシンとしての成功例があり、今後発展性の大きいと思われる応用領域として、データベース、論理シミュレーション、信号処理、イメージ処理を取り上げ、専用計算機命令セットアーキテクチャについて解説する。

専用計算機といえども、ユーザインタフェースや、通常の汎用処理との結合は不可欠である。その実現形態として、汎用ホスト計算機とは独立の前置/後置形のもの、ホスト計算機に専用機構として組み込む形のものなどがあるが、本稿ではその形態よりも、専用機構としての特徴的な命令セットアーキテクチャを中心に述べる。

従来ソフトウェアで逐次的に行っていた処理をハードウェアでまとめて効率的に処理することが専用計算機のキーポイントである以上、応用分野に特有の処理が、なんらかの意味での並列性を多量に内在させてい

る必要がある。内在する並列性を引き出すことはソフトウェアの大きな課題であるが、今後次第に解決される分野が拡がると思われ、専用計算機の重要性が高まるであろう。一方、ユーザインタフェースの高度化、流通ソフトウェアの使用などによる通常処理の大幅増加の傾向は、システム全体からみた専用計算機の実在意義や効果を減価させることになる。将来の動向については、予断を許さない。

## 2. データベースマシンの命令セットアーキテクチャ

1970年代に入り、データベース分野の重要性の高まりや関係データベースの登場を背景として、データベースマシンの研究開発が盛んに進められてきた。今日までに数十件の提案が成され<sup>1)</sup>、いくつかの商用マシンも登場している。

初期のものは、磁気ディスク装置に代表される二次記憶装置側にヘッドやトラック単位にロジックをもたせる方式が多く、対象となるデータの絞り込みによるホスト、ディスク間での転送データ量の削減が目的である。フロリダ大 CASSM<sup>2)</sup>、オハイオ州立大 DBC<sup>3)</sup>などが代表的マシンである。しかし、これらの方式はデータベースの大規模化に対してロジック量が非現実的となるため、実用化は進んでいない。

一方、一次記憶側に必要な全データを持ち込み、それに対する高速処理方式の研究が盛んである。以下主なものをあげる。DIRECT<sup>4)</sup> はメモリ共用のマルチプロセッサ方式を取り、関係演算を並列に実行する。ストリックデータベースマシン<sup>5)</sup> では、タプル比較、集合積、重複排除、ジョインなどの関係演算専用ストリックアレイを構成要素としている。TOP<sup>6)</sup> はノードにプロセッサを配した木構造マシンであり、ノードにアトリビュートを対応させる形でタプルを格納する。ノードでタプルを処理しながら木構造の上を移動

† Instruction Set Processor Architecture of Special Purpose Processors by Seiichi YOSHIZUMI, Shigeo NAGASHIMA, Kenji KANEKO and Hitoshi MATSUSHIMA (Central Research Laboratory, Hitachi, Ltd.).

†† (株)日立製作所中央研究所

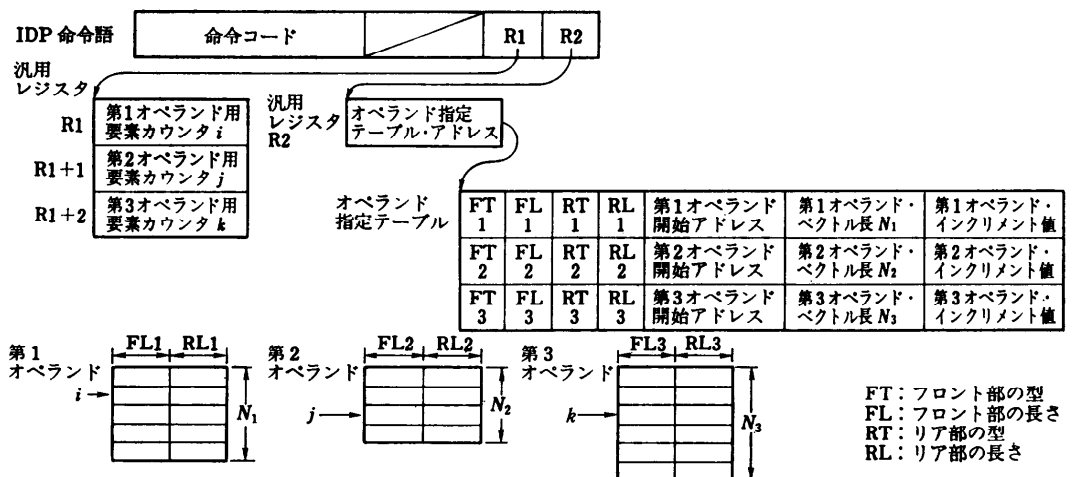
させることにより、関係演算を行う。IBM の並列パイプラインデータベースマシン<sup>7)</sup>では、ソート、ジョインアグリゲーションなどのパイプを構成要素とし、SQL 言語をサポートする。Teradata DBC/1012<sup>8)</sup>は、

1 ディスク、4 プロセッサからなる AMP を構成単位とし、1024 台までの AMP を、マージ機能に特徴のある樹状の Y ネットで結合する。IBM ホストの後置マシンとして商用化されている。日立の IDP<sup>9),9)</sup>も

表-1 デュアル・ベクトル操作命令の一覧<sup>9)</sup>

デュアル・ベクトル  $X, Y, Z$  の各要素は主に識別情報を格納する前半部 (表中は  $f$  で表示) とカラム値などを格納する後半部 (表中は  $r$  で表示) からなる。比較演算は後半部同士で実行される。  $i, j, k$  はそれぞれデュアル・ベクトル  $X, Y, Z$  へのインデックスであり、各要素の比較結果に応じて更新される。

命令	機能概要	主な適用処理
Merge Sort (1種)	$X(i).r : Y(j).r \begin{cases} \leq \\ > \end{cases}$ $Z(k) \leftarrow X(i); i \leftarrow i+1; k \leftarrow k+1$ $Z(k) \leftarrow Y(j); j \leftarrow j+1; k \leftarrow k+1$	ソート 集合和 アドレス変換
Merge Join (1種)	$X(i).r : Y(j).r \begin{cases} < \\ > \end{cases}$ $i \leftarrow i+1$ $Z(k).f \leftarrow X(i).f; Z(k).r \leftarrow Y(j).f; i \leftarrow i+1; j \leftarrow j+1; k \leftarrow k+1$ $j \leftarrow j+1$	結合 集合積 アドレス変換
Merge Difference (1種)	$X(i).r : Y(j).r \begin{cases} > \\ = \end{cases}$ $j \leftarrow j+1$ $Z(k) \leftarrow X(i); i \leftarrow i+1; k \leftarrow k+1$ $i \leftarrow i+1$	集合差 アドレス変換
Sequential Search (6種)	$X(i).r : Y(j).r \begin{cases} \text{yes} \\ \text{no} \end{cases}$ $Z(k).f \leftarrow X(i).f; Z(k).r \leftarrow Y(j).f; i \leftarrow i+1; j \leftarrow j+1; k \leftarrow k+1$ $i \leftarrow i+1; j \leftarrow j+1$	制限 選択 重複排除 統計演算



IDP 命令は、オペランド対応に独立な要素カウンタを有する点と、オペランドがデュアル・ベクトル形式である点とが特徴である。

図-1 IDP 命令の形式<sup>9)</sup>

商用データベースマシンの一つであり、汎用大型機 M-68 XH に内蔵する形をとっている。以下その命令セットアーキテクチャについて述べる。

IDP ではベクトル処理方式を採用し、デュアル・ベクトル（ベクトル要素の前半、後半におのおの独立な値を格納可）を基本データ形式として導入することにより、リレーションを二項関係の組合せで表現する。24 種の 新設命令の中で特徴的なものは、デュアル・ベクトルを直接操作する命令である（表-1）。命令仕様は以下の考慮に基づいて図-1 に示す形をとっている。まず演算中に命令を中断できるように、演算される要素を示す要素カウンタを各オペランドごとに独立に 3 個設け、汎用レジスタを充てる。また、ジョインやサーチなど命令実行前に出力要素数が予測できない演算に対処するため、ベクトル長を各オペランドごとに独立に指定可とし、どれか一つの要素カウンタが、指定されたベクトル長を超えた場合に命令を終了させる。その残りについては、終了時の要素カウンタに従って命令の再実行を繰り返す。これにより、ソートやジョイン演算に関し、20 倍以上の高速化を可能としている。

以上、いくつかのデータベース専用マシンについて発展の経緯と現状を掻い摘んで述べた。半導体技術の目覚ましい進展は、一次記憶装置を大量に利用したデータベースマシン、アーキテクチャの研究開発を、今後一層進めさせることになる。それにともなって二次記憶装置側の隘路解消の必要性が増大し、大量のロジック付加が要求されるが、以前に比べ実現可能性が高まっているといえよう。

### 3. 論理シミュレーションマシンの命令セットアーキテクチャ

論理シミュレーションマシンは、論理動作を高速にシミュレートすることができる専用の処理装置である。マイクロプロセッサはもちろん、最近の計算機の論理はすべて LSI で実現されている。論理設計の段階で論理不良が混入したままで LSI を作成し、検査の段階でそれが発見されると、論理を修正し再度 LSI を作成する必要がある。このための期間は一般には週～月のオーダーの時間を要し、不良件数によっては検査段階の期間が膨大となる。したがって、設計段階でいかに論理不良を除去するかが開発期間短縮の鍵であり、この一手段として論理シミュレーションが利用されている。以前はこの処理はソフトウェアで実現して

いたが、たとえば汎用大型計算機の論理全体をシミュレートするためには、本来のハードウェアの 1 秒分の動作のシミュレーションに年のオーダーの計算機時間がかかる。論理シミュレーションマシンはこの課題を少しでも解決するために開発されたものである。

シミュレーションマシンには 1 LSI の論理の検証を目的としたワークステーションレベルのものから百万ゲート規模の計算機全体の論理シミュレーションが可能なものまである。特に後者はメインフレームメーカーが自社の計算機の開発を目的にそれぞれが独自に開発している<sup>10)~13)</sup>。この中で IBM、富士通、日本電気は並列処理方式を採用して高速化を図っている。一方、日立はスーパーコンピュータに論理シミュレーション専用のアーキテクチャを追加し、ベクトル処理方式により高速シミュレーションを実現している。

論理シミュレーションではゲート単位（もしくはゲートを集合したブロック単位）で入力値を評価して出力値を求める。このとき、評価の方式として入力値が変化したゲートのみを評価する方式（イベントドリブン方式）とすべてのゲートを評価する方式（全ゲートシミュレーション方式）がある。全ゲート中で実際に動作しているゲート数は数%程度であるのでイベントドリブン方式のほうが処理効率が高いが、動作すべきゲートのみを常に把握しなければならないというペナルティがある。一方、全ゲート方式は処理効率は低い動作が単純なのでマシンのハードウェア規模の拡大が容易であり、これによって高速処理を実現することができる。

並列処理方式で全ゲートシミュレーションを実現しているマシンとして IBM の EVE がある<sup>10)</sup>。EVE は 220 個の論理プロセッサ (LP) と 4 個のアレイプロセッサ (AP) から構成され、前者は論理のシミュレーションを、後者は記憶のシミュレーションを分担する。LP には命令メモリとデータメモリがある。命令メモリには命令、すなわち評価すべきゲートごとのゲート機能、4 オペランドアドレス（入力に接続されたゲートアドレス）が格納されている。またデータメモリにはゲートの出力値が格納されている。命令を実行すると、まず 4 つのオペランド（ゲートの入力値）がデータメモリから読み出される。この値とゲート機能からテーブルを引いて出力値に変換し、結果を再びデータメモリへ格納する。テーブル索引の前後で入力値を反転させるなどの指示も可能となっている。またテーブルの内容も 32 種類指定できる。大規模の論

理は分割され、多数の LP で並列に処理される。

このほかのシミュレーションマシンはイベントドリブン方式を採用しているが、中でも日立の VELVET はスーパーコンピュータを利用して高速化している点に特徴がある<sup>11)</sup>。本来スーパーコンピュータは高速科学技術計算を目的とし、ベクトル処理方式により高速処理を実現している。ベクトル処理では一連の数値データに対し同種の演算を施すが、これは論理値データに対しても可能である。論理シミュレーションでは多数のゲートについて入力値とゲート機能から出力値を求める処理が中心であるが、これをベクトル処理で実行することができることに着目したわけである。

イベントドリブン方式でのゲート評価をベクトル処理するためのフローを図-2 に示す。すべてのゲートやフリップフロップとその接続関係は LET (Logic

Expression Table) にて表現されている。またある時点で評価すべきゲートのアドレスは GEVT (Gate Event Table) に格納されている。シミュレーションは読み出し、評価、格納の3ステップからなる。これらが完了すると時間ポインタが更新され、新しい GEVT のもとで次のシミュレーションが開始される。

HITAC S-810 は83種のベクトル命令を有し、この中には5種の論理演算命令がある。シミュレーション処理は従来のベクトル命令を用いても実現不可能ではないが、次の6種のベクトル命令を追加し、一層の高速化を図っている。各命令の適用先は図-2を参照されたい。

(1) VSF: Vector Simulate Function

入オペランドの一部で指定されるゲート機能に従って入力データを評価する命令である。ゲート機能

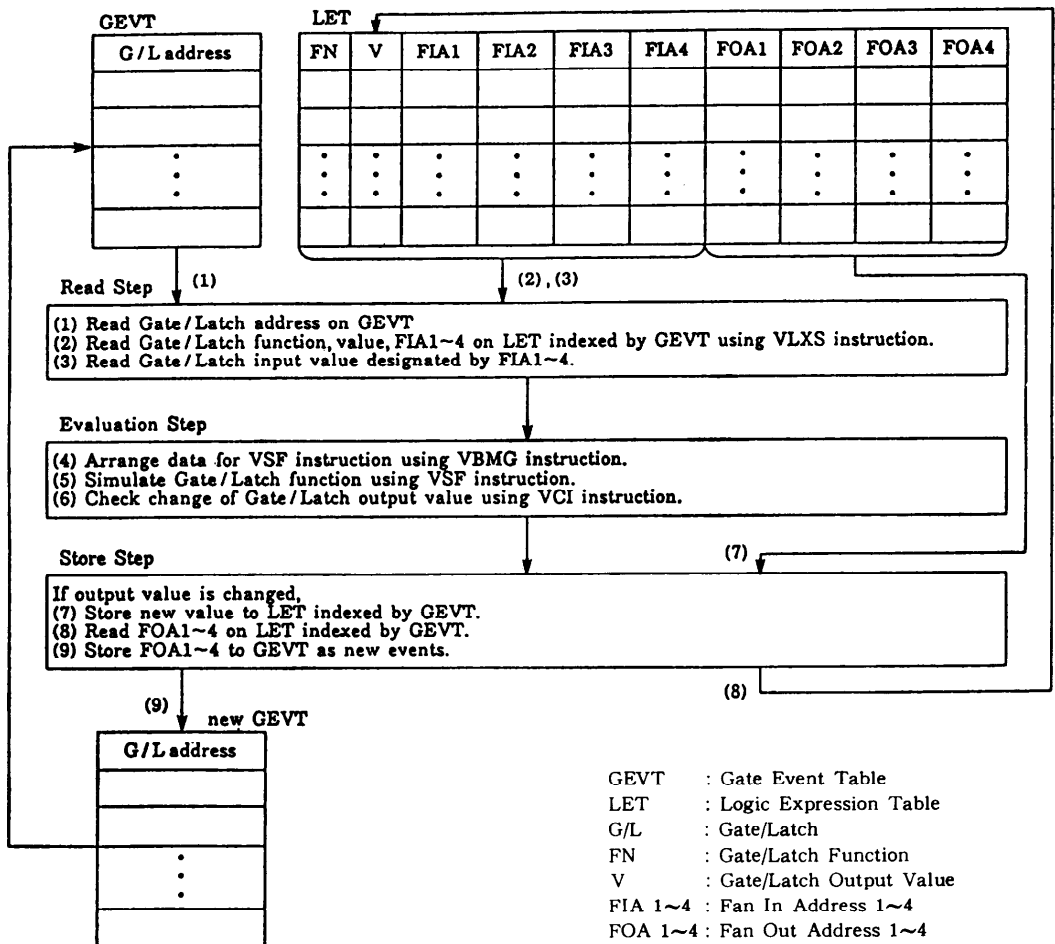


図-2 ベクトル処理による論理シミュレーションの流れ

には AND, OR などがあり, またフリップフロップも 1 機能として処理できる。

(2) VCI: Vector Indirect Function

入力オペランドの一部で指定される比較機能に従って入力データ同士を比較する命令である。

(3) VITRL: Vector First Order Iteration  
Logical

FORTRAN 表現で次の機能を実現する命令である。

DO 10 I=1, L

10 A(I)=(A(I-1). AND. C(I)). OR. B(I)

本命令は論理シミュレーションと機能レベルシミュレーションの結合処理に利用される。

(4) VBMG: Vector Bitwise Merge

指定された制御情報に従って, 2 オペランドを混合する命令である。

(5) VLS: Vector Load Separate

連続する二つの 4 バイトのオペランドを読み出し, 二つのベクトルレジスタにロードする命令である。本命令は各種テーブルの読み出しに利用される。

(6) VXLS: Vector Load Indexing Separate

リストベクトル形式で(5)の命令機能を実行する命令である。

この中で(1), (2)の命令はオペランドの一部に動作を指定するフィールドがあり, これに従って残り部分のデータを操作する。ベクトル処理は本来一連の

データに対し同種の演算を施すものであるが, ここではこの概念が拡張されているとみることができる。

論理シミュレーションマシンによる処理速度の向上は実効的には数十〜百倍程度である。したがってさきに述べた実マシンとの速度比を解消するにはほど遠い。しかしこの高速化でも論理設計者に与えたインパクトは大きく, より高性能の計算機開発に寄与している。今後もさらに高速のマシンの研究開発が期待されている。

#### 4. 信号処理マシン命令セット

##### アーキテクチャ

ディジタル信号処理プロセッサ (DSP) は, 音声, 通信, 画像などの信号処理を高速に行うことを目的としたマイクロプロセッサの一種である。信号処理における最も基本的な要素は, 積和演算の高速化と演算精度の確保である<sup>14)</sup>。これらの要素の実現のために以下に述べるアーキテクチャと命令セットの工夫がされている。

##### (1) 基本アーキテクチャ

これまでに発表された代表的な DSP の諸元を表-2 に示す。これらの DSP は, 一般的には図-3 に示すように並列乗算器を内蔵した積和演算主体の構成となっている。DSP では処理の高速化のために, データアクセス・乗算・加算のパイプライン化を図り, 演算と

表-2 各社 DSP の演算精度と動作の並列度

品 種 名	デ ー タ フ ォ ー マ ッ ト	積 和 演 算		動作の並列度	文 献
		演算データ長	サイクル		
$\mu$ PD 7720 (日本電気)	固定 16 b	$16 \times 16 + 16$ (2 データバス)	1	演算, 2 アドレス演算 レジスタ転送	14)
TMS 32010 (Texas Inst.)	固定 16 b	$16 \times 16 + 32$ (1 データバス)	2	演算, 1 アドレス演算	15)
HD 61810 (日 立)	浮動 12E4	$12E4 \times 12E4 + 16E4$ (2 データバス)	1	演算, 2 アドレス演算	16)
MB 8764 (富士通)	固定 16 b	$16 \times 16 + 26$ (2 データバス)	1	演算, 2 アドレス演算	17)
DSP 32 (AT & T)	浮動 24E8	$24E8 \times 24E8 + 32E8$ (1 データバス)	1	演算, 3 アドレス演算 データ I/O	18)
$\mu$ PD 77230 (日本電気)	浮動 24E8	$24E8 \times 24E8 + 47E8$ (2 データバス)	1	演算, 2 アドレス演算 レジスタ転送	19)
TMS 32020 (Texas Inst.)	固定 16 b	$16 \times 16 + 32$ (2 データバス)	1	演算, 2 アドレス演算	20)
HD 81831 (日 立)	固定 16 b	$16 \times 16 + 32$ (4 データバス)	1	演算, 3 アドレス演算 データ I/O (ピコ命令)	21)
DSP 56000 (Motorola)	固定 24 b	$24 \times 24 + 56$ (4 データバス)	1	演算, 2 アドレス演算	22)

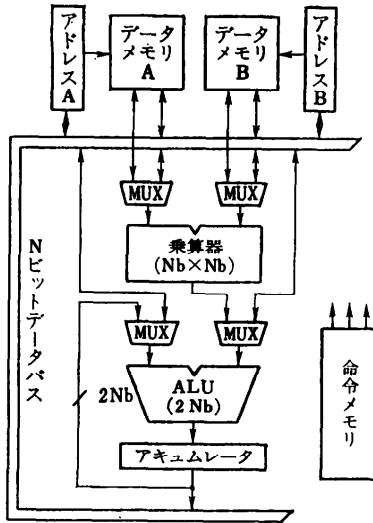


図-3 DSPの基本構成

演算種	アドレス演算 A   B	データ転送
-----	-----------------	-------

図-4 DSPの命令フォーマット

複数データの同時転送の並列化、信号処理特有の高速アドレッシング機能（後述）を備えている。

演算精度の確保のためには、積和の加算においてデータ長を2倍以上に拡張したり（TMS 32010）、オーバフロー防止用のシフト機能（μPD 7720）を備えているものが多い。最近では浮動小数点のデータ形式を採用するDSPも多くなっている（HD 61810, DSP 32, μPD 77230 など）。

(2) 命令セットの特徴

DSPでは並列度の高い動作を1命令サイクルで実行するために、多ビット長の水平型マイクロ命令を採用している。ただし、イミディエイト命令やブランチ命令では、複数ワード命令を採用しているものもある。図-4に示すように、データ演算種、複数データのアドレス演算、データ転送などを同時に指定可能である。HD 81831のように3データのアドレッシングが可能で同時に4データまでアクセス可能なDSPもある。言語としてはアセンブリ言語を用いているものがほとんどであるが、最近ではC言語を採用する動きもある（DSP 32）。

DSPの命令セットは数値演算中心の命令構成となっており、I/O関連の命令は少ない。数値演算では積和演算命令を有しているのが特徴である。データ演算

と並列に実行するアドレス演算では、ビットリバース演算やモジュロ演算などをサポートするDSPも増えてきている（MB 8764, μPD 77230, HD 81831）。

この他、信号処理では短いステップ数の処理を多数繰り返す演算が多いため、高速の繰り返しを可能とする専用命令を有するDSPも多い。通常のパイプライン動作では、繰り返しなどの分岐動作が入るとパイプラインが乱れるために処理フローに余分な動作が必要となる。DSPでは、この余分な動作を回避するためのハードウェアを備えている。

これまでのDSPについてその概要を述べたが、近年のDSPは信号処理に適した基本的な形態を保持しつつ、高機能化、高速化の方向にある。とくに、浮動小数点形式の採用、I/O命令の強化、マルチプロセッシング機能などが強化されている。応用対象も画像処理や映像処理に広がりつつある。

5. イメージ処理

イメージ処理はオフィス、工場などで頻繁に利用されてきており、情報処理システムにおいてその重要さがますます増している。その代表的な応用分野と処理内容を表-3に示す。

これらのイメージ処理は、いずれもデータ処理量が多い。これを高速に処理する必要があるために、大半のイメージ専用計算機は個別応用向けの専用アーキテクチャ<sup>24)~26)</sup>をもっており、この分野全体として一般的な命令セットアーキテクチャが確立されているとは言い難い。本稿では比較的多くのものが採用している命令セットを概観する形でまとめる<sup>27)</sup>。

イメージ処理は大略次のような形で指定できる。

結果=ファンクション（入力イメージ（群）、参照イメージ、処理領域、走査モード、パラメータほか）

表-3 イメージ処理の応用分野

分類	応用分野	主なイメージ処理 (1)
1 生成・復元	グラフィックス、CT（医用、検査）	図形生成、陰面消去、フーリエ変換、など
2 伝送・蓄積	ファクシミリ、放送、記憶装置	圧縮・伸長、多重化、写真・文字領域分離
3 変換・強調	座標変換、惑星画像処理	平行移動、回転、極座標変換、濃度変換
4 認識・理解	OCR、ロボット、図面読取、衛星画像処理など	前処理（雑音除去など）、2値化、パターンマッチング、特徴抽出・計測、統計処理、ベクトル化など

イメージ専用計算機の命令セットは、この式の各項目を指定する命令群、すなわちファンクションを指定するファンクション（制御）命令、結果・入力・参照イメージの格納場所を指定するイメージメモリ制御命令、処理領域や走査モードを指定するアドレス制御命令、などに分類することができる。

以下これらを個別に説明する。

#### (1) ファンクション命令

ファンクション命令は処理の内容を指定するために用いる。代表的なファンクションを以下に示す。

- 凶形発生（面，線，円など）
- 交点算出（陰面消去）
- 直交変換（フーリエ変換など）
- 符号化・復号化（圧縮・伸長）
- 座標変換（平行移動，回転，拡大，縮小など）
- 画素単位処理（2値化，濃度変換など）
- 画像間演算（算術演算，論理演算，色抽出など）
- フィルタリング演算（算術，論理フィルタ）
- パターンマッチング
- 総和（頻度分布，周長，面積）
- 伝播処理・ラベル付け（領域分離）
- 最大値・最小値（及びその位置）の検出
- 追跡（輪郭，中心線，ベクトル化）
- パック，アンパック

#### (2) イメージメモリ制御命令

イメージメモリ制御命令で使用するイメージを指定する。処理開始前に使用メモリを指定しておくことによって、各データごとに指定する面倒をなくしている。

#### (3) アドレス制御命令

アドレス制御命令はイメージメモリのアドレスレジスタを制御する命令である。これによって処理領域とイメージの走査方法を指定する。走査方法には、テレビ画面の走査のように画面を左から右へラインずつ、それを上から下へラインごと走査する（順）ラスタ走査，その逆の逆ラスタ走査型，ランダムアクセス，垂直水平走査型などがある。

#### (4) そ の 他

入出力，表示制御用の命令も用意されている。具体的には、ビデオ，テレビカメラなどの入力機器の選択，複数台のカメラの指定，照明の強さなどの設定，表示方法の設定，カーソルの表示の指定などの命令がある。

また，グラフィック関係の命令として，複数のイメージの重畳表示，点・線・面の描画など，表示画面の制御用の命令も用意されている。

以上，イメージ処理専用計算機における命令セットを示したが，これは確立されているわけではない。イメージ処理高速化にむけて，超並列処理などの独自アーキテクチャも研究されており，今後もそれらに対応した新しい命令セットが生まれてくるものと考えられる。

### 6. その他の専用マシン

上記以外の分野に関しては，本特集の他の解説に譲るべきものも多いので，ここではごく簡単に述べるに止める。

数値計算や各種シミュレーションの分野ではCRAYを始めとするスーパーコンピュータが一種の専用計算機として活用されている。現行のベクトル処理方式だけでなく，今後は並列処理方式が積極的に導入されると考えられる。

通信分野では，以前から通信制御装置が開発され，ホストの負荷低減に寄与している。現状では特筆すべきアーキテクチャ項目は見当たらないが，将来の大容量高速通信に向けて大きな変革の必要性が高まっている。

オンライン・トランザクション処理分野ではノンストップコンピュータが注目される。信頼性を向上させるためのハードウェアの多重化方式がアーキテクチャ上の特徴となっている。

### 7. おわりに

専用計算機の効用と限界は，応用分野や先端技術の開発状況によって大きく変わる。両者を十分考慮した研究開発努力によって，今後，より大きな効用が生み出されることが期待される。

### 参 考 文 献

- 1) Ozkarahan, E.: Database Machines and Database Management, Prentice Hall (1986).
- 2) Copeland, G. P. et al.: The Architecture of CASSM: A Cellular System for Nonnumeric Processing, Proc. of First Annual Symposium on Computer Architecture, pp. 121-128 (1973).
- 3) Banerjee, J. et al.: Concepts and Capabilities of a Database Computer, ACM TODS, Vol. 3, No. 4 (1978).
- 4) DeWitt, D. J.: DIRECT-A Multiprocessor

- Organization for Supporting Relational Database Management Systems, IEEE Transactions on Computers, Vol. C-28, No. 6 (1979).
- 5) Kung, H. T. et al.: Systolic (VLSI) Arrays for Relational Database Operations, Proc. of ACM-SIGMOD, Int. Conf. of Management of Data, pp. 105-116 (1980).
  - 6) Borwccelli, M. A. et al.: A VLSI Tree Machine for Relational Data Base, Proc. of Comp. Arch. (1983).
  - 7) Gajski, D. et al.: A Parallel Pipelined Relational Query Processor: An Architectural Overview, Proc. of Int. Conf. on Computer Architecture, pp. 134-140 (1984).
  - 8) Torii, S. et al.: A Relational Database System Architecture Based on a Vector Processing Method, Proc. of the Third Int. Conf. on Data Engineering, pp. 182-189 (1987).
  - 9) 鳥居他: リレーショナル・データベースの処理速度向上を図る CPU 内蔵型データベース・プロセッサ, 日経エレクトロニクス, No. 414, pp. 185-206 (1987).
  - 10) Beece, D. K. et al.: The IBM Engineering Verification Engine, ACM IEEE 25th Design Automation Conference Proceedings, pp. 218-224 (1988).
  - 11) Nagashima, S. et al.: HARDWARE IMPLEMENTATION OF VELVET ON THE HITACHI S-810 SUPERCOMPUTER, ICCAD '86 Proceedings, pp. 390-393 (1986).
  - 12) 山田 博他: シミュレーションプロセッサ"SP", 電子情報通信学会論文誌, Vol. J71-D, No. 4, pp. 644-651 (1988).
  - 13) 小池誠彦他: 論理シミュレーションマシンのアーキテクチャ, 情報処理学会論文誌, Vol. 25, No. 5, pp. 864-872 (1984).
  - 14) Nishitani, T. et al.: A Single-Chip Digital Signal Processor for Telecommunication Applications, IEEE J. of SSC, Vol. sc-16, No. 4, pp. 372-376 (Aug. 1981).
  - 15) Magar, S. S. et al.: A Microcomputer with Digital Processing Capability, ISSCC 82 (Feb. 1982).
  - 16) Hagiwara, Y. et al.: A Single Chip Digital Signal Processor and its Application to Real-Time Speech Analysis, IEEE J. of SSC, Vol. SC-18, No. 1, pp. 91-99 (Feb. 1983).
  - 17) Kikuchi, H. et al.: A 23 K Gate CMOS DSP with 100 ns Multiplication, ISSCC 83 (Feb. 1983).
  - 18) Hays, W. P. et al.: A Programmable Digital Signal Processor with 32 b Floating Point Arithmetic, ISSCC 85 (Feb. 1985).
  - 19) Kawakami, Y. et al.: A 32 b Floating Point CMOS Digital Signal Processor, ISSCC 86, (Feb. 1986).
  - 20) Kneib, K. et al.: An NMOS Digital Signal Processor with Multiprocessing Capability, ISSCC 85 (Feb. 1985).
  - 21) Kaneko, K. et al.: A 50 ns DSP with Parallel Processing Architecture, ISSCC 87, (Feb. 1987).
  - 22) 56-Bit General Purpose Digital Signal Processor, Motorola Semiconductor Technical Data (1986).
  - 23) 田村, 坂根, 富田, 横矢, 金子, 坂上: ポータブル画像処理ソフトウェアパッケージ SPIDER の開発, 情報処理学会論文誌, Vol. 23, No. 3, pp. 321-328 (May 1982).
  - 24) Hwang, K. et al.: Computer Architectures for Image Processing (special issue), Computer, pp. 10-80 (Jan. 1983).
  - 25) 坂上, 水戸出: イメージプロセッサの最近の動向, 信学誌, Vol. 67, No. 1, pp. 90-98 (Jan. 1984).
  - 26) 「画像処理特集」信学論, Vol. J68-D, No. 4, pp. 853-932 (1985. 4).
  - 27) 小林, 奥山, 折田, 浅田, 藤原: 汎用画像認識解析装置 "HIDIC-IP", 日立評論, Vol. 67, No. 9, pp. 63-66 (1985. 9).

(昭和63年10月3日受付)