図形処理および連想処理のための高度並列演算装置*

棟 上 昭 男** 山 口 徹 郎**

Abstract

This paper describes basic construction of a highly parallel information processing system for two dimensional data analysis and associative processing which is now under development.

The system consists of AIPU (Associative Information Processing Unit), AIPU control, and a long word plated wire memory system; and the whole system is connected to a general purpose computer via a high speed data channel.

The basic operation in the system are 3×3 template matching, propagation, and parallel boolean operation.

Using the AIPU simulation program, it has been shown that most of the essential operations required for binary pattern analysis and procedures for converting patterns into their descriptions are successfully carried out with this system.

Keywords: Highly parallel information processing system. Graphical or two dimensional data processing. Conversion of graphical data into its description. Pattern analysis and recognition. Associative processing.

1. まえがき

文字認識あるいは図形認識の分野で，これまで試みられてきた方法の多くは，与えられた図形が，あらかじめ定められたクラスのいずれに属するかということを機械に判定させることを目的としたものであって，このような場合，機械から得られる答えは，与えられた図形の属するクラスの名前か，あるいは判定不能の応答のいずれかである．

しかしながら，このような方法のみでは，文字以外の種々の図形情報の処理を行なうことはほとんど不可能である．泡箱やスパークチェンバにおける粒子飛跡写真，指紋，染色体や神経細胞の写真，航空写真などのように自然界から得られる図形情報の処理，あるいは回路図，回路パターン，フローチャート，化学構造式などを含む構造の複雑な一般図形の処理などを考えれば，これは明らかであろう[1)~4)]．

このような場合要求されるシステムは，対象とする図形（たとえば二値図形，あるいは線図形といったような）に共通な，交点，端点，線分，あるいは濃度，色彩などの特徴を取り出し，それらの接続関係，位置関係，その他の属性といったものをもとに，与えられた図形を処理の容易な構造を有するデータあるいは記述に変換し必要に応じてこれらを記憶，伝送し，認識などの処理や再生を行なうことのできるものである．

ところで，現在一般に用いられている計算機は，必ずしもこのような処理に適した構造を有していない．ここで述べる演算装置を開発する目的の一つは，一般の計算機にこのような装置を付加することによって，上に述べたような処理が，容易に実行できるようにしようとするところにある†．

2. 問題の背景

図形処理，あるいは図形認識の問題を，言語処理の手法を用いて解決していこうという試みは，これまでにもいくつかみられる[5),6)]．とくに Kirsch および Narasimhan の仕事は，このような手法に先鞭をつけたもので高く評価されている[7),8)]．また最近では，図形記述言語による図形の形式的な取扱い，その図形認

* Highly Parallel Information Processing System for Graphical Data Processing and Associative Processing, by Akio Tojo and Tetsuro Yamaguchi (Electrotechnical Laboratory)

** 電気試験所・電子計算機部

† "図形処理" という言葉は，図形を介して人間と計算機の間で情報交換を行なう場合の処理に限定して使用される場合が多いが，これは "計算機図学" (Computer Graphics)，"図形を介する人間計算機系" (Man-Machine Graphical Communication System) などと呼ぶべきものである．"図形処理" (Graphical Data Processing, または Pictorial Data Processing) は，これらのほかに自然界から得られる図形の処理などを含め，より広い意味で用いられるべきであろう．

識システム，図形表示システムへの応用などに関する考察を行なった結果も発表されている[9]．著者の一人もこれと類似の図形表現方式と，図形をこのような記述に変換するためのアルゴリズムを考え，さらに，そのための装置の基本的な構成を提案した[10]．本論文はこのようなシステムを具体化するために，現在試作中の装置について，その構成，機能，シミュレーションを行なった結果などについて述べたものである．

言語処理・図形処理を初めとする種々の非数値問題は，計算機の応用分野の中でも，とくに重要なものであるが，このような問題の中心は多くの場合，各データ要素の間の関係をどのように計算機内部で表現し，取り扱うかというところにある．このような場合，情報を単純な配列の形で取り扱おうとすると，処理時間や記憶場所，システムの柔軟性などの点で困難な問題が生ずる．そこで，たとえば図形を介して，人間と計算機が緊密に連絡をとりながら，設計作業を行なうようなシステムにおいては，多重リスト構造によって，情報を表現するのが普通である[11]．

計算機の記憶装置内部に情報を格納するさいの物理的な位置と，情報の論理的構造をこのように切りはなし，データ要素相互間の関係の取扱いを容易にしたシステムは，一般に連想処理システム（Associative Processing System）と呼ばれているが，このような処理方式は，参照するデータの内容により直接情報を検索できる記憶装置（Content Addressable Memory または CAM）があれば非常に能率よく実現できる．しかしながら，実際にこのような装置を実現するには多くの問題があるため，これを待たずに Hash Coding などの手法を用い，プログラム技術によって，模擬的な CAM を実現し，これにより連想処理システムを作ろうとする試みが提案され，注目を集めている[12]．

一方，最近の大規模集積回路技術の発達に伴い，半導体集積回路によって，CAM を実現しようという試みも数多く見られる[13]．また，従来の記憶装置を利用するものとしては，$2\frac{1}{2}D$ の磁心記憶装置の周辺回路を変更して，n 語ごとにビット方向の読み書きを可能にし，これによって内容による検索機能を与えようという方式も提案されている（n は1語のビット数）[14]．

後述のように，ここで発表する装置も，$n\times n$ ビットの二次元情報を n 語 n ビットの語単位の情報とみなすとき，n 語ごとに与えられたビットパターンを有する語が存在するか否かを検出し，必要ならば，一致した語の所定のビットを，定められた値に設定する機能があり，これによって図形処理を初めとする種々の非数値問題に，有用な連想処理システムを実現することが可能である[15),16)]．

すでに，二重 Hashing により実現した連想処理システムを利用して，質問応答システムに演繹機能を持たせる試みがあるが[17]，このような手法は，図形処理の場合にも適用できる．図形認識を行なう場合に，たとえば，“より右”という関係と，“より左”という関係は，互いに逆であることを一般的に定義することによって，一方の関係から，他方の関係を自動的に導くことができるならば，個々の図形を定義する場合に，その構成要素の間の，すべての関係を網羅する必要は無くなるであろう．

以上のように，ここで述べる装置は，与えられた図形を，計算機で処理することの容易な構造を有するデータに変換するのに用いられるばかりでなく，その後の処理や，そのほかの非数値問題に，有用な連想処理システムを実現するうえでも利用できる機能を有しており，この装置を付加することによって現在の一般の計算機の機能を大幅に増大させることが可能である．

この論文で述べるような，高度に並列化された演算装置，あるいは計算機システムには，これまでにも種種の構成・機能・目的のものが提案され，試みられている[18]．本論文の装置と同様の目的のものとしては，イリノイ大学の ILLIAC-Ⅲ がもっとも著名であるが[19]，装置としての構成には問題があり，むしろ，これに関連した Narasimhan らの仕事が最大の成果であるといってよい．ここで述べる装置が，これまでのものと本質的に異なる点は，二次元情報に対する基本演算機能，語単位の一致検出機能などにあるわけであるが，装置的に見ても，集積回路の実装方式，制御信号の供電方式，演算装置を補助するための記憶装置に長語磁性線薄膜記憶装置を用いる点など，多くの特徴を有している[15),16),20)]．

3. 演算システムの構成と機能

3.1 演算装置の構成と実装方式

ここで述べる演算装置の中心となる部分は，比較的単純な論理機能を有する同一の演算回路（演算セル）を $n\times n$ のマトリクス状に配列し，それぞれの最近傍セルの間で情報の交換が行なわれるように接続を行なった並列構造の演算装置（AIPU——Associative Information Processing Unit）である．これらの演算セルに対しては，そのための制御回路から共通の制御

信号が供給される.

演算セルは4ビットのレジスタと，これに付随する論理回路で構成されているが，各セルの対応するビット位置を取り出して考えると，これは二次元的な情報をたくわえるための平面的なレジスタであると考えることができる．そこで，これをそれぞれ PLANE_0, PLANE_1, PLANE_2, PLANE_3 と呼ぶことにする．**Fig. 1** はこの演算装置内の情報の流れを，特定座標のセルについて，模型的に示したものである．

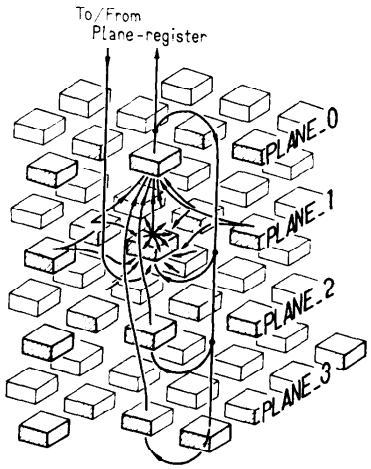


Fig. 1 Basic construction of Associative Information Processing Unit. Arrows show main information flow in a cell. Flow of control signals is not included.

この図では，制御信号の流れは示されていないが，情報のそれぞれの通路に対して，原則として演算装置全体に共通の制御信号が加えられるようになっており，これによって所定の演算が実行される.

PLANE_REGISTER は $n \times n$ ビットの情報を並列に読み出し，あるいは書き込むことのできる語長の非常に大きな記憶装置であって，AIPU の記憶容量を補助するために用いられる.

これに対しAIPUと通常の主記憶の間の情報交換は演算平面の周辺を通して n ビット単位で行なわれる.

このような構成の AIPU を制御するための制御装置は，最も基本的なレジスタとして R_REGISTER, D_REGISTER の2種類のレジスタを有している．R_REGISTER は2組の8ビットサイクリックシフトレジスタ，D_REGISTER は2組の n ビットレジスタで構成されており，それぞれ二次元局所的な一致検出演算（型行列演算），あるいは一次元データに対する一致検出演算を実行する場合の参照パターンを保持するのに使用される．R_REGISTER からの出力信号は，AIPU 全平面に対して共通に供給される．これに対して，D_REGISTER からの信号は，桁ごとに独立に AIPU の対応する列へ共通に加えられる.

AIPU 内における一致検出演算，各平面間の情報の転送，情報の消去，PLANE_REGISTER からの並列転送などを実行させるために，制御装置から各演算セルに共通に供給される制御信号は11種類であり，このほかに R_REGISTER からの参照パターン，転送するさいの肯定側・否定側の指定のための信号などを加えて，合計 35 種類の制御信号が各演算セルに共通に供給される[16),20)].

各演算セルに対する入力情報は周囲8個の最近傍セルからの8種類と PLANE_REGISTER からの1種類，出力情報は PLANE_0 と PLANE_1 からの2種類でデータに関するセルあたりの結線数は 21 である.

現在試作中の装置では，n の値は 16 で，各演算セルは市販の電流切換形集積化論理回路 15 個で構成されており，各セルが1枚のプリント基板におさめられている．**Fig. 2** はこの演算セルの基板を示したものである．この基板は多層基板を用いずに構成されており，また，平面全体の物理的な大きさもかなりのものとなるため，十分に高い演算速度を維持するためには，その実装状態には，とくに注意しなければならない.

Fig. 3 は AIPU の制御信号の給電方式を示したもので，演算平面内での制御信号の時間的なばらつきは 4〜5 ns 以内におさえられている．各演算セル間の情報交換のための相互結線に関しては，原則として整合をとることを行なっていない.

このような実装方式をとる場合，各演算セルに制御信号が加えられてから，その結果が最近傍セルに達するまでの時間は，10〜25 ns であることが確かめられ

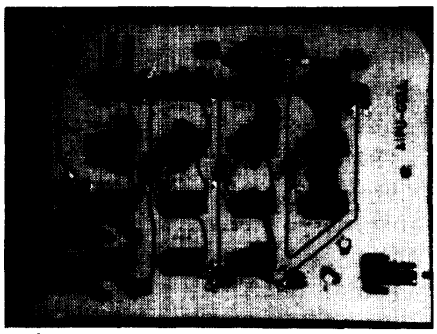


Fig. 2 AIPU unit cell package construction

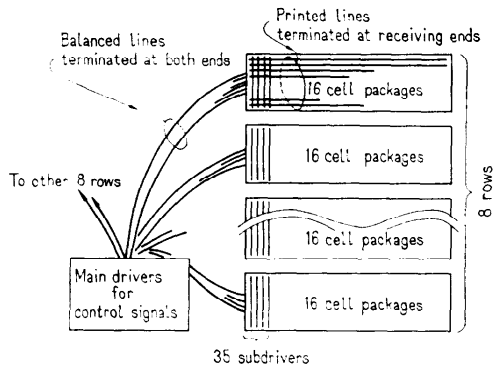


Fig. 3 Control signal distribution scheme for AIPU

ている[20]. 後述の基本演算は，これらの制御信号を 2～30 ステップ実行して実現されるようになっており，命令解読などの時間を見込んでも，現在一般に用いられている高速度の記憶装置に比較して，十分に速い動作を期待できる.

3.2 基本演算

演算平面内部における演算を実行するための基本命令として，ここで考えるものは 12 種類であって，このほかに AIPU 制御回路の各レジスタの内容を設定する命令，PLANE_REGISTER と AIPU 間で情報を転送するための命令などを含め，計 19 種類の命令が用意されている．各レジスタの内容の設定，格納は，本来は，主記憶を参照する命令になるはずであるが，現在の実験的なシステムは，AIPU とその制御回路が通常の計算機とチャネルを介して接続される構成になっており，AIPU の命令で直接主記憶のアドレスを指定できる構造にはなっていない．**Table 1** はこれらの基本命令を示したものである．命令を構成する 1 語は 16 ビットで構成されており，命令のコードはそのうちの 4 ビット，または 8 ビットを占める．条件判定などはすべて上計算機内で実行される．つぎに，これらのうちのおもなものについて説明する.

(1) 並列論理演算 (PLANE_AND/OR, BITWISE_AND/OR)

この演算は PLANE_1, 2, 3 の内容の肯定側，または否定側の間の論理積，または論理和を作り，PLANE_0 の内容に論理的に加える演算で，CL ビットにより，あらかじめ PLANE_0 の内容を消去するか否かが指定できる．PLANE_1, 2, 3 の肯定側をとるのか否定側をとるのか，あるいは無視するのかに関する

Table 1 AIPU instruction repertoire and format.

INSTRUCTION	OP CODE (OCTAL)	FORMAT
PLANE_AND	07	OP CODE / CL / D1 / D2 / D3
PLANE_OR	06	
AND_TEMPLATE	05	OP CODE / STEP / RV / CL / D1 / D2 / D3
OR_TEMPLATE	04	
BITWISE_AND	03	OP CODE / CL / D2 / D3
BITWISE_OR	02	
PROPAGATE	01	OP CODE / D2 / D3
SHIFT_LOAD	016	OP CODE / SD / CH / CV / N
STORE_SHIFT	014	
SHIFT	010	OP CODE / CH / CV / N
PARALLEL_LOAD_FROM_PLANE_REGISTER	013	OP CODE / PR ADDRESS
PARALLEL_STORE_TO_PLANE_REGISTER	011	
LOAD	006	OP CODE / HV / UBR / LBR / CHR / RR
STORE	004	
PARALLEL_INTERNAL_TRANSFER	007	OP CODE / CLR / DEST
PARALLEL_INTERNAL_TRANSFER_NEGATED	005	
SET_RR_UPPER_HALF	003	OP CODE / RR1 / RR2 / RR3 / RR4
SET_RR_LOWER_HALF	001	
NO_OPERATION	000	OP CODE

指定は D1, D2, D3 の各 2 ビットずつで行なう.

たとえば，D1='10', D2='11', D3='01'†† とする積演算では

$$[\text{PLANE\_0}] \leftarrow \overline{\text{CL}} \cdot [\text{PLANE\_0}] + [\text{PLANE\_1}] \cdot [\overline{\text{PLANE\_3}}]$$

なる演算が実行される．ただし，ここで〔 〕は平面の内容を表わす.

これから推察されるように，この演算においては，'11' が無視条件で，指定された平面の内容は無視される．和演算の場合もこれは同様で，'11' に指定された平面の内容は，結果になんの影響も与えない.

桁ごとのブール演算は，このような演算を PLANE_1 に関しては桁ごとに指定できるようにしたもので，主として後述の語単位の一致検出演算に利用される．指定を桁ごとに行なうためには，このために用意された D_REGISTER が使用される.

(2) 型行列演算 (AND/OR_TEMPLATE)

この演算は PLANE_1 内の各セルと，その周囲の 8 個の最近傍セルにより作られるパターンが，**Fig. 4** のように与えられた参照パターン (3×3 型行列) と

†† Single quotation mark (') ではさまれた数字または文字は，ビットパターン，あるいは論理値を表わすものとする.

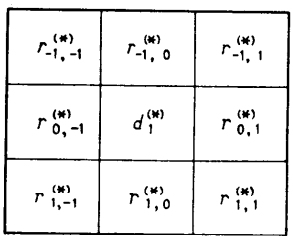


Fig. 4 3×3 template matrix. Asterisks denote 0 or 1

一致しているか否かを検出する二次元局所的な一致検出演算である.

参照パターンの中心以外の値, すなわち, Fig. 4 の $r_{k,t}^{(0)}, r_{k,t}^{(1)}$ の値としては, R_REGISTER の内容を用いる. R_REGISTER は前述のように8ビットのサイクリックシフトレジスタ2組で構成されており, 型行列命令中に与えられる符号を含め, 4ビットの STEP の値に応じて, その桁数だけ型行列を回転するごとに一致検出が実行され, 最終的にはそれらの結果の論理和が PLANE_0 中に求められる. 命令中の RV ビットが "1" ならば, 与えられた型行列の水平軸に対して対称なパターンについても同様の一致検出が行なわれる. したがって, この演算では最大 16 ステップの演算が行なわれることになる.

R_REGISTER が2ビットずつ計16ビット用意されているのは, 型行列の各位置について "1", "0", "Q" (無視条件) の指定を可能にするためで, たとえば, PLANE_t の座標 (i, j) におけるセルの内容を $p_{t,i,j}$ とすれば, 積型行列演算 (AND_TEMPLATE) は, 各ステップですべての (i, j), (k, l) の組に対し, つぎのような演算を実行することに相当する.

$$p_{0,i,j} \leftarrow p_{0,i,j} + \pi_{k,l}(r_{k,t}^{(0)} \cdot \overline{p_{1,i+k,j+l}} + r_{k,t}^{(1)} \cdot p_{1,i+k,j+l}) \cdot \cdot \pi_t(d_t^{(0)} \cdot p_{t,i,j} + d_t^{(1)} \cdot p_{t,i,j}) \quad (1)$$

$$\begin{pmatrix} k, l = 0, +1 (k = l = 0 \text{ を除く}) \\ t = 1, 2, 3. \quad i, j = 1, 2, 3, \cdots n. \end{pmatrix}$$

ただし, ここで π は論理積を求めることを表わす. また, $r_{k,t}^{(0)}, r_{k,t}^{(1)}$ は R_REGISTER の内容を表わし, $d_t^{(0)}, d_t^{(1)}$ は命令の Dt ビットで与えられる値である. CL ビットが "1" であれば, 第1ステップで PLANE_0 の内容はあらかじめ消去される.

(1)式からもわかるように, $r_{k,t}^{(0)} = r_{k,t}^{(1)} =$ "1" または $d_t^{(0)} = d_t^{(1)} =$ "1" の指定のあるビット位置または平面は無視して演算が実行される (無視条件), また $r_{k,t}^{(0)} =$ "0", $r_{k,t}^{(1)} =$ "1" あるいは $d_t^{(0)} =$ "0", $d_t^{(1)} =$ "1" であれば, PLANE_1 上の対応する位置の値が, "1" のときに一致の条件が満たされ, 逆の場合には "0" のときに条件が満たされる. 型行列中に $r_{k,t}^{(0)} = r_{k,t}^{(1)} =$ "0" あるいは $d_t^{(0)} = d_t^{(1)} =$ "0" の指定が1箇所でもあれば, PLANE_1 上の値が何であっても, 一致の条件が満たされることはなく, PLANE_0 上のパターンは変化しない. もちろん, この場合 CL ビットが "1" であれば, PLANE_0 上には, すべて "0" のパターンが得られる.

和型行列演算 (OR_TEMPLATE) では, 無視条件以外の部分で一致のとれる所が1箇所でもあれば, PLANE_0 上の対応する座標位置のセルの内容を "1" とする演算が実行される. すなわち

$$p_{0,i,j} \leftarrow p_{0,i,j} + \sum_{k,l}(\overline{r_{k,t}^{(0)}} + \overline{p_{1,i+k,j+l}}) \cdot (\overline{r_{k,t}^{(1)}} + p_{1,i+k,j+l}) + \sum_{t}(\overline{d_t^{(0)}} + \overline{p_{t,i,j}}) \cdot (\overline{d_t^{(1)}} + p_{t,i,j}) \quad (2)$$

ここで, Σ は論理和を求める演算を表わし, k, l, t などの条件は, (1)式の場合に等しいものとする.

これらの演算では, 演算平面の周辺では演算が定義されない. そこで, PLANE_1 の周囲には, 4個の n ビットレジスタと, 4ビットのレジスタ1個が用意されており, 周辺では, これらのレジスタの内容を参照して演算が実行される. これらのレジスタの内容は, レジスタの設定命令により, 独立に設定できるようになっており, 演算平面より広い広がりを有するパターンを, いくつかの部分平面に分割して演算を実行する場合などにはその境界情報がたくわえられる. またこれらのうちの二つは, 実際には桁ごとのブール演算のための D_REGISTER を構成するレジスタでもある.

(3) 伝播演算 (PROPAGATE)

これは PLANE_1 上の "1" の状態をリップル方式で, R_REGISTER 中に設定されたデータにより, 指定される方向に周辺まで伝播させる演算である.

この演算は語単位の一致検出演算, 演算平面の状態の判定, パターンの射影を作る演算, あるいはパターンの位置関係を調べる演算などに用いられる. また, 伝播の可能な領域は PLANE_2, および PLANE_3 上のパターンで制限できるので, 与えられたパターンの中から, ある点に接続する部分や, 線図形上の特定の点の間の線分を抽出する演算などにも利用される.

伝播可能領域の指定は, 命令中の D2, D3 で行なう. その方法は積演算, 積型行列演算などの場合と全

く同じで，たとえば，D2=D3='11' であれば，全平面上で伝播可能であり，またD2='10', D3='01' であれば，〔PLANE_2〕・〔PLANE_3〕で与えられるパターン上が伝播可能領域となる．

（4） 周辺の設定，格納と移動 (SHIFT_LOAD, STORE_SHIFT, SHIFT)

この演算は PLANE_0 の4辺のいずれかにチャネルからのデータを転送するか，その逆の転送を行ない，かつ1語の転送を行なうたびに，PLANE_0 の内容を1桁移動する演算である．4辺のいずれを選択するかは SD ビットで定められるが，これにより同時に移動方向も縦・横各2種類ずつのうちのいずれかに定められる．CH, CV ビットは平面の周辺同志を接続してサイクリックに移動を行なうか否かの指定を行ない，またNは移動の桁数(設定または格納の回数)を定める．

単純な移動演算の場合には，移動方向は R_REGISTER の内容で定められる．したがって，この場合の移動の基本方向は8種類であるが，この演算は和型行列演算の論理を利用して行なわれるので，R_REGISTER の内容と一致のとれる方向から移動してくることになり，2つ以上の方向へ同時に移動させるようなことも可能である．

（5） 演算平面間のデータ転送 (PARALLEL_INTERNAL_TRANSFER)

PLANE_0 の内容またはその否定側を，PLANE-1, 2, 3 の内容に論理的に加えるためのもので，転送先は DEST ビットで指定される．この場合2つ以上の平面に同時に転送することも可能である．CLR ビットはこれらの転送先をあらかじめ消去するか否かを指定するためのもので，DEST='000' であれば，指定された平面を消去する演算だけが実行される．

（6） その他の演算

指定された PLANE_EREGISTER の内容を PLANE_1 の内容に論理的に加える演算，および PLANE-0 の内容を指定された PLANE_REGISTER の内容とする演算がある，

また，PLANE_1 の周辺，あるいは R_REGISTER の内容を設定したり格納したりする命令が用意されている．この命令 (LOAD, STORE) において H, V は PLANE_1 上の4辺を構成するレジスタのうちの特定の2つを表わし，UER, DER, LER, RER, CNR は PLANE_1 の外側に用意された型行列演算における境界情報を保持するためのレジスタを指定する．RR は R_REGISTER を指定する．また LER, RER は桁ごとのブール演算のための D_REGISTER を指定することにもなる．この命令ではいずれの場合でも，指定されたレジスタとチャネルの間でデータの転送が行なわれる．このほか，型行列を指定するために R_REGISTER の内容を，命令のオペランドで設定する演算が用意されている．

3.3 基本演算を利用する重要な二次的演算

以上述べた基本演算を用いて実行される，二次元情報処理および連想処理において一般性の高い重要な二次的演算としては

（1） 3×3 より大きな任意の型行列を用いる型行列演算
（2） 任意の広がり（大きさ）を有するパターンの処理
（3） 語単位の一致検出演算
（4） 演算平面上の任意の座標の指定
（5） 演算平面の状態の検出

などが考えられる．これらのうち(1)の問題は，任意の型行列を用いる型行列演算が，3×3型行列演算に展開できることがわかっているので，これにより演算を実行することができる[21]．

(2)の問題は前節でもふれたように，与えられたパターンを，演算平面に等しい大きさの部分パターンに分割して演算を行なうことにより解決される．型行列演算の型行列が 3×3 より大きな場合，あるいは型行列演算の結果に，再び型行列演算をほどこす場合などでは，各部分パターンに対して 3×3 以内の型行列演算を実行し，境界情報を新しいデータに置き換えてから，つぎの型行列演算を実行するという方法をとることになる．また，伝播演算の場合，与えられた伝播方向が2つ以上で，それらが互いに逆の成分を含む場合には，境界情報を管理すると同時に，一般には，安定状態に達するまで，各部分パターンに対し，繰返し演算を行なうことが必要とされる．

(3)の語単位の一致検出演算の機能は，この演算装置を特徴づける最も重要な機能の一つである．

この演算は，演算平面上の二次元的な情報を，1語 n ビットの n 語の情報とみなし，外部から与えられた参照データに一致するビットパターンを有する語を検出する演算である．すなわち，これは1ブロック n 語のブロックごとに，ブロック内に所定のビットパターンを有する語が含まれているか否かを検査する演算であると考えることができる．

この演算は，桁ごとのブール演算と伝播演算により

実現される，すなわち，D_REGISTER に参照パターンを設定し，桁ごとの論理積演算を実行すれば，PLANE_1 の各桁の値のうち D_REGISTER に一致した値を有する部分に対応する PLANE_0 上の値が "1" に設定される．もちろん，D_REGISTER 上で "Q" に設定された桁に対応する PLANE_0 の桁は，すべて "1" となる（マスクビット）．したがって，与えられた参照パターンに一致した内容を有している語に対応する部分では，PLANE_0 上の値が行方向（語方向）にみた場合，すべて "1" となるはずである．これを周辺で検出するには，PLANE-0 の内容の否定側を行方向に伝播すればよい．一致のとれない語では，LSD または MSD の桁の値が "1" となるから，これにより検出を行なうことができる．また，一致のとれた語の特定の桁を，"1" に設定する演算も簡単に実現することができる[16]．

（4）の座標の選択は，周辺レジスタの設定と伝播演算，および並列論理演算の組合せで実現できることは明らかであろう．

（5）の問題は，平面上に与えられた条件を満たすセルが，存在するか否かを調べるためのもので，終極的には PLANE_1 上に "1" 状態のセルが存在するか否かを調べる問題に帰着される．そして，これは伝播演算を行なった後，周辺セルの内容を検査することにより，簡単に実行することができる．

4. 演算装置のシミュレーション

これまでに述べたような構成の演算装置を実現するにあたり，これを模擬するためのプログラムを作製し，これによって二値の二次元図形を解析する場合の基本的な処理を実行し，その有効性を確かめた．そのうちの二，三の例をつぎに示す．

Fig. 5(b) は同図 (a) に示す図形の骨格抽出を行なった例である．骨格抽出の方法は種々考えられるが，その方式とここで考える交点，端点の抽出方式とは密接な関係を有する．ここに示すのは，そのうちの最も簡単な方式による抽出例である．

Fig. 5 (c) は (a) 図の交点および端点を抽出した例である．もちろん，ここで交点と端点を別々に抽出することも可能である．これらはすべて型行列演算と並列論理演算で実現される．

Fig. 5 (d) は抽出された各端点と，これに接続する交点の間の各線分をすべて抽出した例である．この演算では並列論理演算と伝播演算が用いられる．

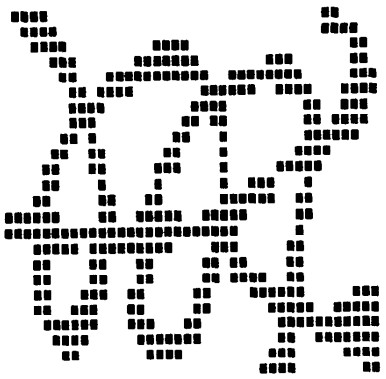
Fig. 6(b) は同図 (a) のような太さが均一でない図形から太さの急激に変化する部分，あるいは突起の付根の部分などを抽出した例である．

これらの処理は，原パターンを分割せずに処理できるものとすれば，骨格抽出以外は数ステップないし数十ステップのプログラムで実行できる．骨格抽出の場合には，原図形の太さにより演算時間が異なることになるが，1ループのプログラムステップ数は，やはり同程度である．詳細に関しては文献 10)，16) を参照していただきたい．

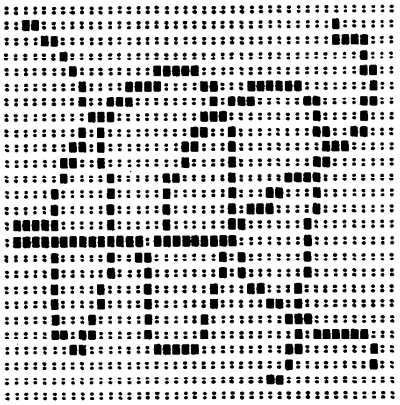
5. むすび

現在試作中の高度並列演算装置の演算平面の大きさは 16×16 で，これに PLANE_REGISTER のための長語磁性線薄膜記憶装置と，命令を解読し演算を実行するための AIPU 制御装置が組み合わされる．この実験的なシステム全体は，Honeywell DDP-516 を基本とする NEAC 3200（1語 16 ビット，8K 語．サイクル時間 960 ns）と高速度チャネルを介して接続され，いわば，ハードウェア化されたサブルーチンとして動作する．したがって，現在の実験的な段階ではこれは一種のハードウェアシミュレーションであるということもできよう．しかしながら，主計算機の方が外部装置に対して，その主記憶に直接アクセスすることを許す完全なサイクルスティーリングが可能となれば，AIPU 制御装置に条件判定，制御関係の機能を附加することによって，完成されたシステムとして動作させることが可能なはずである．もちろん，この場合レジスタの設定・格納などの命令は，2語で構成されることになる．

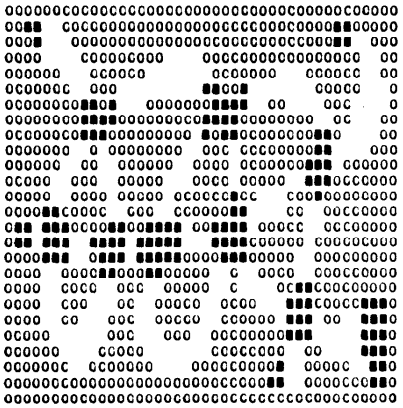
演算セルは，すでに述べたように，集積回路 15 個で構成されているが，これは現在の市販集積回路に用いられている中規模集積回路の技術をそのまま用いたとしても，3～4 個以内におさめるようにすることは容易なはずである．大規模集積回路が非常に近い将来，完全に実用化されることは疑いのない事実であるが，それとともに，これまで受け入れられてきた装置設計の基盤を見直す必要が生じてきた．また，これまで実際的ではないと思われてきたような方式にも，工学的な実現性の生じてきたものがある．ここで述べたシステムも，そのような部品技術の進歩と，計算機の応用分野の拡大からくる要請を考慮に入れた，より高度の機能を有する情報処理システムを生み出していくための一つの試みであると考えることができよう．



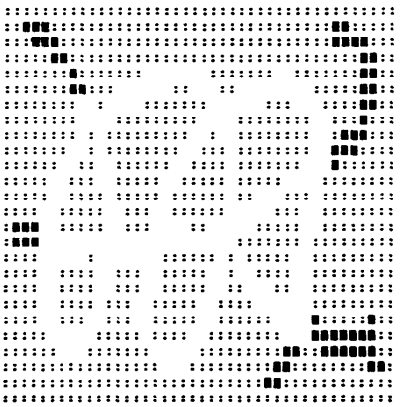
(a) Original pattern



(b) Skeleton obtained with the most simplified algorithm

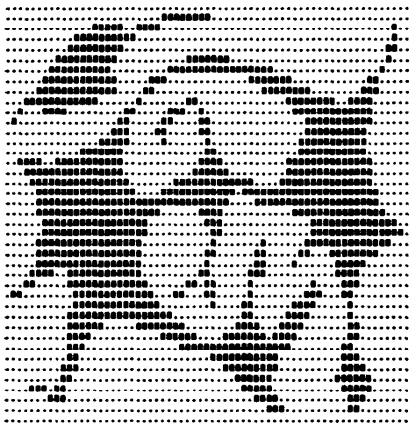


(c) Cross points and end points detected



(d) Segments connecting the end points and the nearest cross points including the end points

Fig. 5 Results of excuting some of the fundamental operations required for binary pattern analysis using AIPU simulation program



(a) Original pattern

(b) Result

Fig. 6 Detection of abrupt change in thickness

このシステムは，二値図形の解析とその記述への変換，および語単位の一致検索をおもな目的とするが，このほかにも，二点間の接続路を見い出す問題など，ある種の図形合成問題にも利用できることがわかっている[22]．また，伝播阻止形の文字認識なども，このシステムで実行することができる[23]．

このシステムに関し現在残された問題は，二値以上の図形の処理を，このシステムで実現するための有効なアルゴリズムを見い出すこと，あるいはそのための機能を附加するとすれば，いかなるセル構成にすればよいかを明らかにすることなどである．

最後に，この研究を可能にしていただいた野田克彦電子計算機部長，ご指導・ご討議いただいた西野博二情報システム研究室長，石井治計算機回路研究室長，および渡辺定久主任研究官に厚く感謝します．また，シミュレーションプログラムの作成，演算セルの設計製作などに関連し，協力していただいた青山 宏，秋葉裕次両技官にもあわせて感謝します．

参考文献

1) B. F. Wadsworth: "PEPR——A Hardware Description", Proc. Emerging Concepts in Computer Graphics, pp. 41～70, 1967.
2) J. H. Wegstein: "A Computer Oriented Single-Fingerprint Identification System", NBS Technical Note 443, March 1968.
3) R. S. Ledley, L. S. Rotolo, et al.: "Pattern Recognition Studies in the Biomedical Sciences", Proc. SJCC, pp. 411～430, 1966.
4) J. K. Hawkins, G. T. Elerding, et al.: "Automatic Shape Detection for Programmed Terrain Classification", Proc. Filmed Data and Computer Seminar, pp. XVI-1～XVI-9, June 1966.
5) J. Feder: "The Linguistic Approach to Pattern Analysis——A Literature Survey", Technical Report 400～133, New York Univ., Feb. 1966.
6) W. F. Miller, A. C. Shaw: "Linguistic Methods in Picture Processing——A Survey", Proc. FJCC, pp. 279～290, 1968.
7) R. A. Kirsch: "Computer Interpretation of English Text and Picture Patterns", IEEE Trans., EC-13, No. 4, pp. 363～376, 1964.
8) R. Narasimhan: "Labeling Schemata and Syntactic Descriptions of Pictures", Information and Control, Vol. 7, pp. 151～179, 1964.
9) A. C. Shaw: "The Formal Description and Parsing of Pictures", Technical Report SLAC-84, Stanford Linear Accelerator Center, Stanford Univ., March 1968.
10) 棟上昭男: "高度並列情報処理装置による図形の記述(I)", 電気試験所彙報, Vol. 31, No. 8, pp. 930～946, 1967.
11) 棟上昭男: "電子計算機による自動設計", エレクトロニクス, Vol. 13, No. 13, pp. 1423～1437, 1968.
12) J. A. Feldman: "Aspects of Associative Processing", Technical Note 1965-13, Lincoln Lab., MIT, Apr. 1965.
13) J. A. Narud, W. C. Seelbach: "High Speed Integrated Circuit Memories", Proc. Conf. on Integratd Circuits, pp. 29～38, May 1967.
14) H. S. Stones: "Associative Processing for General Purpose Computers through the Use of Modified Memories", Proc. FJCC, pp. 949～955, 1968.
15) 棟上昭男, 山口徹郎, 他: "高度並列演算装置による図形の記述(II)——並列演算システムの構成", 昭和43年度電気四学会連合大会, No. 2588.
16) 棟上昭男, 山口徹郎, 他: "高度並列演算装置による図形の記述(VI)——演算システムの構成とシミュレーション", 電気試験所彙報, Vol. 33, No. 5, pp. 478～505, 1969.
17) W. L. Ash, E. H. Sibley: "TRAMP; An Interpretive Associative Processor with Deductive Capabilities", Proc. ACM National Conf., pp. 143～156, 1968.
18) J. C. Murtha: "Highly Parallel Information Processing Systems", in Advances in Computers, Vol. 7, pp. 1～116, Academic Pr., 1966.
19) B. H. McCormick, "The Illinois Pattern Recognition Computer——ILLIAC III", IEEE Trans., EC-12, No. 6, pp. 791～813, 1963.
20) 山口徹郎, 棟上昭男: "高度並列演算装置のセル構造と動作", 昭和44年度電気四学会連合大合, No. 3168.
21) 棟上昭男, 山口徹郎: "高度並列演算装置による図形の記述(IV)——型行列一致演算の一般性について", 電気試験所彙報, Vol. 31, No. 12, pp. 1294～1301, 1967.
22) 棟上昭男: "高度並列演算装置による図形の記述(V)——距離関数と最短接続路の合成", 電気試験所彙報, Vol. 32, No. 9, pp. 930～942, 1968.
23) H. A. Glucksman: "A Parapropagation Pattern Classifier", IEEE Trans., EC-14, No. 3, pp. 434～443, 1965.

(昭和44年8月7日受付)