

画像処理プロセッサの最近の動向

坂上勝彦(電総研) ・ 木戸出正継(東芝総研)

1. はじめに

デジタル画像処理技術は、ここ十数年のあいだ着実に進歩し、いろいろな分野に適用され実用化も進んでいる。しかし現在の逐次型コンピュータにおける画像処理能力不足は常に指摘されており、画像処理用高速プロセッサの開発が強く望まれている。実際に研究用、実用を問わず従来から種々のシステムが提案・試作されている。代表的なものについては文献(1,2,3)にサーベイがある。本稿では最近の話題について研究開発の動向を概観し、特に興味深い例を詳述する。

2. 最近の動向

画像処理プロセッサの研究の流れを図1のようにまとめた。アーキテクチャという観点から分類している。現在では次の4種類の大きな流れがあると考えられる。

① フレームメモリ型：

画像蓄積用フレームメモリに専用ハードウェアを付加し、ソフトウェアでは時間のかかる処理を高速化したシステム。

② 並列プロセッサ型：

画像の構造に合わせて演算装置を並列に配置することにより高速化をはかたシステム。

③ パイプライン型：

画像データがプロセッサ内をパイプライン的に流れるのにつれて処理が行なわれるシステム。

④ マルチプロセッサ型：

複数の並列動作可能な演算装置をタスクに合わせて柔軟性を持って配置することのできるシステム。

フレームメモリ型はコマンド型画像処理システムの処理速度向上のために、特に計算時間のかかるコマンドを選んでフレームメモリ自体に専用演算装置を組み込んだものである。一般的にコストパフォーマンスがよく、256×256, 512×512などよく使う大きさの画像が扱えるものが多い。代表的なシステムがIPU⁷⁾である。IPUのコンポリューションの部分はいわゆる局所並列型であり、パイプライン化による高速化が行なわれている。しかし領域番号付け、FFT、アソシエーションなど極めて有用なオペレーションもハードウェア化されていて、IPUを単なる局所並列型プロセッサと呼ぶことはできない。むしろ、ある目的

のために必要ないくつかの画像処理のコマンドとその目的に合った実行時間で実現できるようにした画像処理プロセッサと呼びてよいであろう。画像処理研究用という目的でEYなどのコマンドが1画素あたり/μsで実行できるように設計されたのがIPUである。この意味でIPUなどいくつかのプロセッサを図1では独立した流れとして描いたのである。研究用のものはDeAnza社やComtal社のような画像処理機能を持つアスキー装置の流れと平行して今後伸びて行くと考えられる。また、必要な機能だけを持たせて専用装置に組み込まれたものも多くなるであろう。

複数のプロセッサを並列動作させ処理を高速化する非ノイマン型並列プロセッサの研究は画像処理用とは限らず大きな流れとして存在している。画像処理への応用という立場からは、各演算モジュールがそれぞれ1つの画素に対応して互に同じ動作をするもの(STARAN¹⁾, CLIP4⁷⁾ etc)と一般に複数のプロセッサが並列に動作するもの(ZMOB⁴⁵⁾, SPARC⁵¹⁾ etc)に別れる。前者を並列プロセッサ型と呼ぶ。後者は、マルチプロセッサ型という1つの流れを形成している。

並列プロセッサの中でもCLIP4のように個々の基本演算モジュールを画素と同じ2次元構造に配列

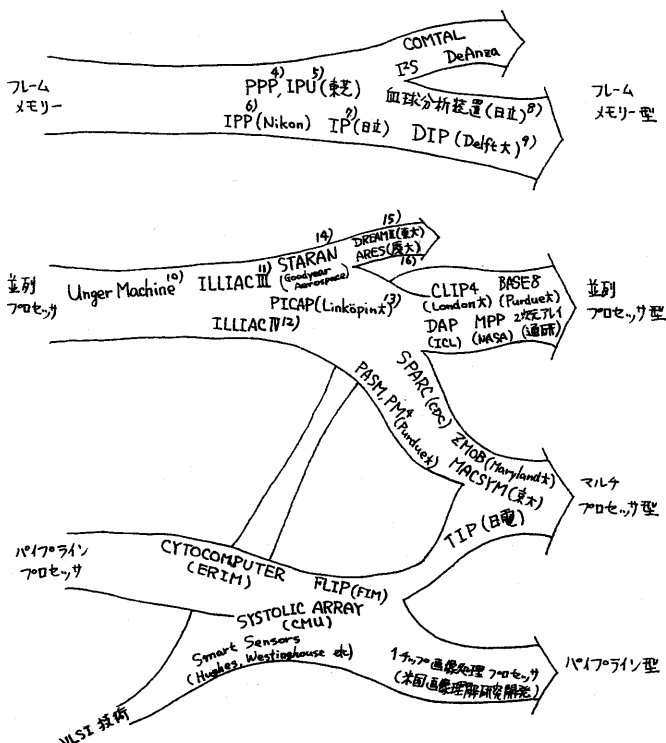


図1. 画像処理プロセッサ研究開発の動向。

したものを完全並列型と呼ぶ。各モジュールで並列動作させ、全画像のデータを同時に処理することによって高速処理を実現しようとしている。古くはUngerマシン⁽¹⁰⁾やILLIAC III⁽¹¹⁾によってその基本原理が提案されていた。しかし当時の技術背景では小規模なシステムによる実験やシミュレーションしかできなかったようである。ところが、完全並列型の規則性に富んだ構造はVLSI化に向いており、最近の高集積化技術の進歩により、かなりの数のモジュールを持つシステムが実際に作れるようになりつつある。それにともないDAP⁽⁸⁾やMPP⁽¹⁹⁾など、実用を考えた大規模なシステムの開発が進んでいるのが最近の動向である。

並列処理とならんで処理の高速化に効果があるのがパイプライン方式である。この方式を画像データの処理に対して積極的に応用したのがCYTOCOMPUTER⁽²³⁾などのパイプライン型プロセッサである。最大の特徴は画像データの転送と処理が同じモードで行なわれるという点にあり、スループットの高速化が比較的容易に実現できる。また、構造の規則性、モジュール間の接続の次元性などVLSI化にも適している。よって実時間処理が要求される専用装置に組み込む画像処理プロセッサのアーキテクチャとしてパイプライン型は今後広く採用されると考えられる。スマートセンサーをはじめとする1チップ画像処理プロセッサの開発^(25~35)や、VLSI向きのアーキテクチャとして注目されているシストリックアレイ^(36~42)など、活発な研究が進められている。

パイプライン型は一般に融通性が欠け、定型的な処理しかできないという欠点を持つ。最近発表されたTIP⁽⁴⁹⁾はデータ駆動の概念を導入することによってパイプライン路をデータ別に任意に設定できるようにしたプロセッサである。これはパイプライン型からマルチプロセッサ型に合流する新しい流れとみなすことができる。

マルチプロセッサ型では再構成可能な(reconfigurable)複数の演算モジュールが画像データを並列に処理する。これまで開発されてきた数の画像処理プロセッサは一般にある特定の用途にはよいが、用途が少しでも異なると融通性が欠け効率的でないという問題をかかえている。また、原画像から得られた特徴量を使ってさらに高レベルの処理を行なおうとする場合には複雑な条件分岐をプログラムする必要がある。全画面を一緒にしか処理できないような画像処理プロセッサではこのようなことは難しい。再構成可能なマルチプロセッサの考え方はこういった問題に対する答の1つになり得ると考えられている。ただし提案されているマルチプロセッサ型システム多くは画像処理への応用の可能性を論じただけのものが大部分で、実際に処理例を示したものは極めて少ない。今後の研究に期待したい分野である。

以上最近の研究開発の動向を概観した。次章では特に興味深い例についていくつか詳述する。

3. 画像処理プロセッサ—最近の話題—

3-1. 完全並列型プロセッサの新しい動き

現在、実際に使われているか、あるいは製作中の完全並列型プロセッサの中で、比較的大規模なものとして広く知られているのがCLIP4⁽¹⁷⁾、DAP(Distributed Array Processor)⁽⁸⁾、MPP(Massively Parallel Processor)⁽¹⁹⁾である。(表1) このほかPurdue大のBASE8^(20,21)、通研の

	モジュール数	モジュール間の接続	RAMの深さ	基本命令サイクル	モジュール数/チップ
CLIP4 (ロンドン大;英)	96×96	8近傍	32 bit	10μs	8
DAP (ICL;英)	64×64	4近傍	4 Kbit	200 ns	4
MPP (NASA;米)	128×128	4近傍	1 Kbit	100 ns	8

表1. 代表的な完全並列型プロセッサ。

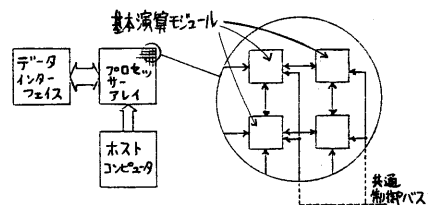


図2. 完全並列型プロセッサの典型的なアーキテクチャ。

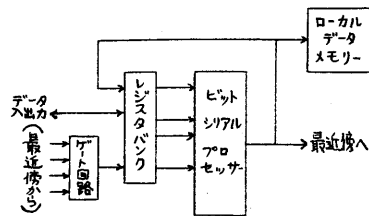


図3. 基本演算モジュールの典型的な構成。

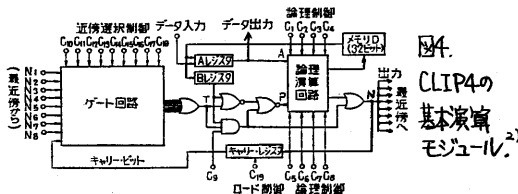
2次元アレイ⁽²²⁾などがある。いずれも基本的には図2、図3のような共通のアーキテクチャを持つ。図2はアレイのアーキテクチャ、図3は基本演算モジュールのアーキテクチャである。各モジュールはビットシリアル型論理演算プロセッサと、ビット単位のアクセスが可能なローカルRAMから構成されており、RAMの深さに相当する枚数の2値画像を貯えることができる。濃淡画像は複数枚のビットプレーンとして扱う。DAPとMPPはCLIP4と比べ桁違いに多いビットプレーンを扱え、完全並列型プロセッサの新しい動きとして注目される。

完全並列型プロセッサではモジュールは莫大な数になるため、基本演算モジュールはできるだけ単純なものでない実現性がない。しかもある程度の柔軟性を持たせる必要がある。基本演算モジュールとして最新のMPPでもビットシリアル型プロセッサが採用されている最大の理由はこの点にある。ビットパラレル型では固定語長のデータしか扱えないし、また、たとえば8ビットのプロセッサで2値画像を処理しようするとハードウェアが無駄になる。これに対しビットシリアル型だと単純な構成のモジュールであらゆるビット数のデータ(浮動小数点データ等可)

もそのビット数に応じて有効に扱うことができるのである。もちろんデータビット数が多くなると必要な命令ステップ数が多くなり、それだけ処理に時間はかかることになる。

表1のそれぞれのプロセッサの基本演算モジュールを図4~6に示す。DAP, MPPはローカルRAMの容量が大きいことから先もかのように浮動小数点演算まで含めた算術演算向きのプロセッサである。そのためDAP, MPPでは各モジュールは4近傍としか接続されておらず、しかもそのどれかをマルチプレクサで選択することしかできないようになっている。MPPは部分積の蓄積用にシフトレジスタまで持っており、完全に算術演算向きに設計されていると言える。さらに、128×128のモジュールを一次元につなぐことも可能で、画像にみかわらず一般の算術演算の並列処理にも利用できるように工夫されている。

DAPのAレジスタ(図5)、MPPのGレジスタ(図6)は、どちらもモジュールのActivityを制御する。(CLIP4, BASE8にはない)



いかに組み合わせ、汎用性のあるシステムを作ることができるかが今後の課題であろう。

3-2. パイプライン型画像処理プロセッサ

画像処理オペレーション自体がパイプライン的に連結された本格的なパイプライン画像処理プロセッサとして代表的なのがCYTOCOMPUTER²³⁾である。すでに文献2)で紹介されているので詳細は省略する。FLIP²⁴⁾もパイプライン型とみなすことができる。これについては文献3)で紹介がある。本節では1チップ画像処理プロセッサとシストリクアレイについて解説する。

1チップ画像処理プロセッサ

米画像理解研究開発の中でCCD(Charge-Coupled Device)やVLSIによる1チッププロセッサが登場している。実際の応用例としては軍用画像処理を想定している。たとえばミサイルの先端にTVカメラを取り付け目標を認識したり、ヘリコプターに搭載したFLIR(Forward Looking Infrared)センサーの信号により地上の単気車やトラックを 실시간で発見、認識するシステムなどがある。このようなシステムでは小型軽量性と 실시간処理が不可欠な条件となり、VLSI化、パイプライン化の研究が進められているのである。

ヒューズ研究所のNudd²⁵⁾は画像理解システムにおける階層性を図8のように表現した。ピラミッドの横幅は各レベルにおけるデータ量を示している。底辺の部分は画像データの信号レベルの処理に相当する。例としては微分処理などがある。上層に上がるにつれて条件分岐などが多くなり処理は複雑になり計算量の多いものになる。しかし逆に、扱うデータ量は上層ほど減る。たとえば最下層では全画素を扱う必要はないが、次のレベルでは微分値の大きい画素、さらに次はラインセグメントにほると言うようにデータ数は指数関数的に減少するのである。従って処理量(スループット)は各レベルで一定にすることができる。図8の最下層の部分(低レベル)の処理は条件分岐がほとんどなく、高精度は要求されない。また手法に

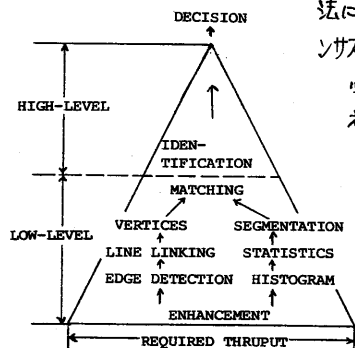


図8. 画像理解システムの階層性²⁵⁾

法にあってある程度のコンセンサスがあり専用LSIプロセッサ化に適していると言える。高いレベルの部分はデータレートが低いため汎用プロセッサでも間に合う。このようにして原画像データのレベルから最終的な判断まで

を 실시간のスループットで実現しようとするのが米画像理解研究における 실시간処理の原理である。

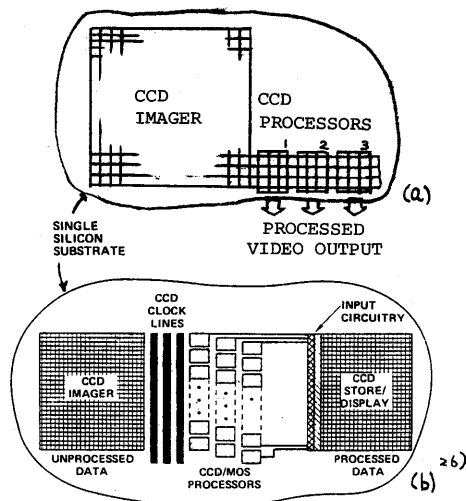
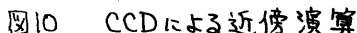


図9. スマートセンサー

LSI化の試みはスマートセンサー開発から始められた。次世代の軍用撮像素子としては小型軽量高感度、低消費電力、残像がないなどの特長を持つCCDイメージセンサーが中心になるであろう。CCDセンサと同じチップ上にやはりCCDで作った画像処理プロセッサを組み込み、ビデオ出力として処理済の信号を得ようとするのがスマートセンサーである。CCDでは画像データを電荷の量で記憶する。その精度は6bit程度しかないが、低いレベル処理の場合にはこれで十分なのである。図9が概念図である。(a)はビデオ出力の部分にCCDプロセッサを付加したものである。²⁵⁾ 図の例は3x3近傍演算プロセッサの例である。この図のようなプロセッサを並べると1つのチップからたとえば「ラプラス変換、平滑化など」いろいろなビデオ信号を得ることができるのである。さらに高速に処理を行ないたいときは(b)のように受光部の電荷を転送するときに並列処理を行なえばよい。²⁶⁾ しかしどちらのタイプも概念として提案されているだけで、現状は(a)の撮像部を除いたCCDプロセッサの部分だけをLSI化して実験している段階である。

図10にCCDによる近傍演算プロセッサの例を示す。²⁶⁾ (a)は平滑化フィルタである。画素の濃淡レベルに比例した電荷によってゲート電極には電圧が発生する。各電極を(a)のように接続すれば3x3の各電荷量の和に比例した電圧がセンスできるのである。ソベレルフィルタのようにウェイトを与えたい場合は電極の面積を図10(b)のように変えればよい。ソベレルフィルタを実現するためには、さらに、X方向の微分値、Y方向の微分値の絶対値をそれぞれ計算し加算する回路が必要となる。ヒューズ研究所ではこれらの回路もCCD技術によって実現している。²⁶⁾

ヒューズ研究所では表2のような機能を持つ3種のテストチップ



スマートセンサーで行なう低レベル処理よりはやや上位

表2. CCDスマートセンサーによる画像処理²⁵⁾

シストリックアレイ



$$C_{out} \leftarrow C_{in} + A_{in} \times B_{in}$$

$$A_{out} \leftarrow A_{in}$$

$$B_{out} \leftarrow B_{in}$$

図11. シストリットアレの基本演算モジュール.

図12. 帯行列($p=2, q=3$)とベクトルの乗算³⁸⁾

~5~

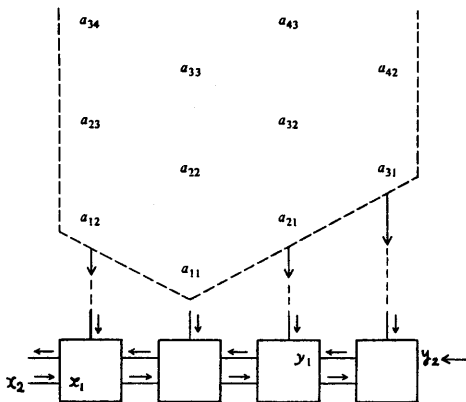


図13. 行列×ベクトルのためのシストリックアレイ.³⁸⁾

外部との接続するための端子数が最小化される³⁶⁾など、VLSI化にとって極めて有利な特徴を持っており最近特に注目されているアーキテクチャである。

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 & \dots \\ 0 & a_{11} & a_{12} & a_{13} & a_{14} & \dots \\ 0 & 0 & a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & 0 & 0 & a_{11} & a_{12} & a_{13} \\ 0 & 0 & 0 & 0 & a_{11} & a_{12} \\ 0 & 0 & 0 & 0 & 0 & a_{11} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix}$$

A X Y

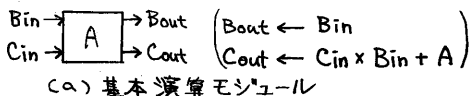
図14. 有限インパルス応答コンボリューション.

1次元データについては図14のような計算を図13のシストリックアレイで行なえば有限インパルス応答のコンボリューションが計算できる。³⁷⁾ 2次元画像データの場合も原画像と重み係数をそれぞれ1次元データに展開すれば、1次元コンボリューションとして扱うことができる。³⁹⁾

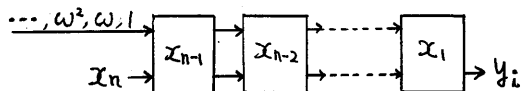
フーリエ変換(DFT)は、図15のような行列・ベクトル乗算を行なえばよい。しかし係数行列は帯行列ではないため図13のシストリックアレイでは実現がむずかしい。文

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^5 \\ 1 & \omega^3 & \omega^6 & \omega^7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

図15. フーリエ変換(DFT).



(a) 基本演算モジュール



(b) フーリエ変換用リニアシストリックアレイ

図16. シストリックアレイによるDFT.⁴⁰⁾

献40)では図16のような917°のシストリックアレイによるDFTを提案している。図15の式をデータ数=5として展開すると次式が得られる。

$$\begin{aligned} y_1 &= (((x_5 - 1 + x_4) \cdot 1 + x_3) \cdot 1 + x_2) \cdot 1 + x_1 \\ y_2 &= (((x_5 \cdot \omega + x_4) \cdot \omega + x_3) \cdot \omega + x_2) \cdot \omega + x_1 \\ y_3 &= (((x_5 \cdot \omega^2 + x_4) \cdot \omega^2 + x_3) \cdot \omega^2 + x_2) \cdot \omega^2 + x_1 \\ y_4 &= (((x_5 \cdot \omega^3 + x_4) \cdot \omega^3 + x_3) \cdot \omega^3 + x_2) \cdot \omega^3 + x_1 \\ y_5 &= (((x_5 \cdot \omega^4 + x_4) \cdot \omega^4 + x_3) \cdot \omega^4 + x_2) \cdot \omega^4 + x_1 \end{aligned}$$

図16(b)はこの式を計算するシストリックアレイと言うことができる。

一般にn点のDFTを直接計算すると計算量は n^2 に比例する($O(n^2)$)。FFTアルゴリズムでは $O(n \log n)$ となる。これに対し、シストリックアレイを使うと $O(n)$ となり理想的なスピードでDFTを実現できるのである。

ESL社では実際にシストリックアレイを組み込んだシストリックプロセッサを開発した⁴¹⁾(図17)。これはホスト計算機に対する付加プロセッサとして働くように設計されている。シストリックアレイの部分は図18のような構成になっている。各モジュールはLSIの乗算器、アキュムレータから構成されており、16ビットの積和演算が200nsで実行される。

各演算モジュールにはセルメモリと呼ばれるRAMが内蔵されている。パイプライン処理の際には、データ系列と平行してセルメモリを読み出すためのアドレスがアレイに流し込まれる。(図18(b)) 各演算モジュールでは、読み出されたセルメモリの内容と入力データとの間で積和演算が行なわれるのである。各セルメモリに重み係数をプリセットしておけばコンボリューション(1次元、2次元)が計算できる。また各モジュールのセルメモリに行列の列成分をセットしておけば行列・行列の乗算を実行できる。シストリックプロセッサではDFT、コサイン変換をこの方法で実現している。

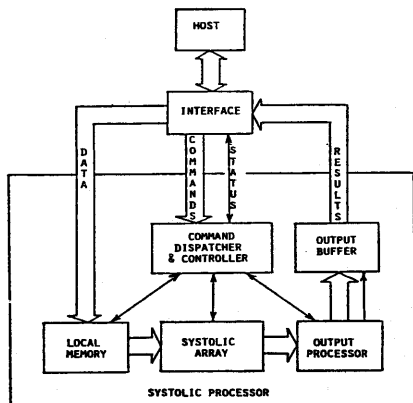


図17. ESLシストリックプロセッサ⁴¹⁾

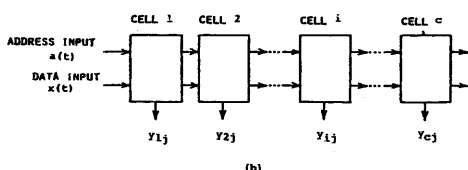
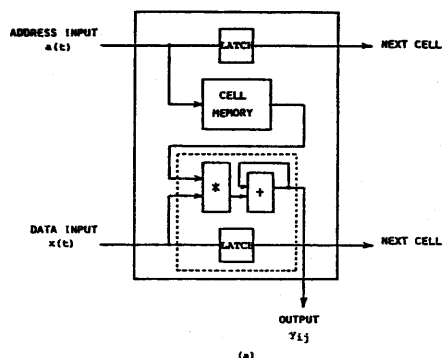


図18. シストリックプロセッサの基本モジュールと直線状アレイ⁽⁴⁾

シストリックアレイの概念を生かす汎用性を失わないようにしたのがシストリックプロセッサの特徴であろう。計算能力は、 128×128 画素、濃淡8ビットの画像に対し、コンボリューションがサイズ 5×5 では10.6ms、 19×19 の場合は66.5msで実行される。後者の計算レートは131 MOPSに達する。モジュール数(現在は、20個)を減らし、パイプラインをきめ細かくすればさらに高スループットが期待できる。

シストリックアレイには文献(42)で述べられているように様々なバリエーションが存在する。画像や信号処理に実際に応用する場合の構成法についても他にいろいろなものが提案されている。少しでも汎用性のあるアーキテクチャが今後伸びると思われる。

3-3. マルチプロセッサ型画像処理プロセッサ

まず提案されているいくつかのマルチプロセッサ型画像処理プロセッサの分類を試みる。FLIP⁽²⁴⁾とDIP⁽⁹⁾はそれぞれパイプライン型、フレームメモリ型のプロセッサであるが、細かく見れば再構成可能な構造を持つ。そこで、これらも含めて考えてみた。

● 基本演算モジュールの種類

汎用Mプロセッサ:

PASM⁽⁴³⁾, PM⁽⁴⁴⁾, FLIP(16個)⁽²⁴⁾

ZMOB(280×256 個)^(45,46)

MACSYM(2800×16 個)⁽⁴⁷⁾

PX-1(280×32 個)⁽⁴⁸⁾

乗算器, 加算器 等:

TIP^(49,50), SPARC⁽⁵¹⁾, DIP⁽⁹⁾

(注) アンダーライソも引いたものは開発中, 稼動中

● モジュール間のデータ交換法

バス : FLIP, DIP, MACSYM

リング : TIP, ZMOB

バス+リング : PX-1

クロスバー : SPARC

ネットワーク : PASM, PM⁽⁴⁾

● 画像データへのアクセス方法

ラスタ-走査のみ : FLIP, DIP

共有画像メモリ-ランダムアクセス:

TIP, SPARC

分割記憶+共有画像メモリ:

PASM, PM⁽⁴⁾, MACSYM, PX-1

分割記憶のみ : ZMOB

こういったプロセッサに関する提案は他にもいろいろある。しかしアイデアだけの提案も多く、実際に稼動ししかも画像処理プロセッサとして有効に使われているものとなるとその数は極めて少ないと思われる。この主な理由はハードウェア的な問題よりむしろソフトウェア面にあると考えられる。データの流れを制御し、各演算モジュールを効率よく並列動作させるためのある種の高級言語を開発しなくてはとても使いこなすことはできないであろう。この点がこの種のプロセッサの最大の課題であると考えられる。1つのアプローチがTIP(後述)に見られる。TIPは、データフロー型のアーキテクチャを持ち、関数型のアセンブラでプログラムすることができるようになっている。これによって機械語レベルのプログラミングよりはるかに開発効率が上がったことが報告されている。⁽⁵⁰⁾

また、マルチプロセッサ型は高レベル画像処理向けのアーキテクチャとは言われているが、実験例として紹介されているのはほとんどコンボリューションやFFTのように低レベルのものはかりである。単に複数個のプロセッサの接続法に関するアイデアや試作だけでなく本当に高レベル処理の高速化が可能かどうかを示す研究が進むことを期待したい。

本節では最近の実例としてZMOB, TIPを取り上げ簡単に解説する。

ZMOB

ZMOB^(45,46)はメリーランド大学で現在設計および

シミュレーションを進めているマルチプロセッサシステムである。基本モジュールは Z80A マイクロッサ(命令サイクル = 2.5 μ sec) と高速乗算器から成り、それぞれ 64 Kbyte のメモリを持つ。この基本モジュール 256 個をホスト計算機 (VAX 11/780) が Conveyor Belt と呼ばれるシフトレジスタ・リンクによって環状に接続されるのである。リンクのスピードは 12.5 MHz、幅は 48 ビット (control: 8, data: 16, source: 12, destination: 12 ビット) である⁴⁶⁾。よって全システムでの記憶容量は 16 Mbyte、処理速度は 100 MOPS となる。たとえば 512×512 画素の画像を処理する場合はそれぞれのモジュールが 2 ラインの処理を担当するのである。近傍演算の場合は 32×32 画像 256 枚に分割する方が能率がよいであろう。アクセスしたいデータが自分のモジュールにない場合は Conveyor Belt を通じてデータを他のモジュールから受け取ることになる。画像毎演算なら話は簡単であるが、近傍演算や幾何学的変換などの処理ではどの程度の効率で動くか、モジュール間のデータのやりとりをどう管理するのか、など解決すべき問題も多い。

ΣMOBと比較的似た考え方で構成されているのが京都大学のMACSYM (Micro-processor Asynchronous Complex SYstem)⁽⁴⁷⁾である。これはマスタープロセッサ1台とスレーフプロセッサ16台(いずれもΣ8001)を非同同期共通バスで接続している。各プロセッサのメモリーは128 Kbyteである。さらに、バスには可変構造共有メモリー(8Mビット)が接続されている。

TIP

データ駆動の概念を導入してパイプライン路を可変に

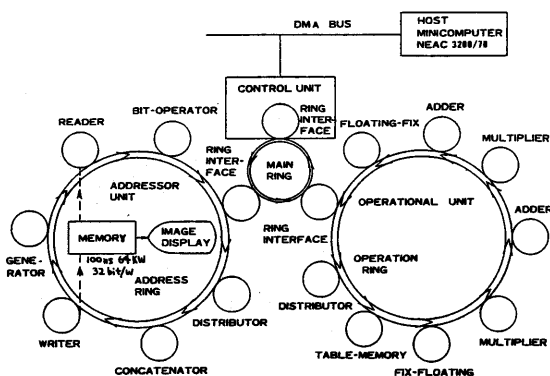


図19. TIPの構成⁽⁴⁹⁾

し、汎用性を持たせたのが TIP (Template-controlled Image Processor ; 日電.)^{49, 50)} である。パイプライン処理が行なわれるのと同時に、複数演算モジュールの並列動作によって処理の高速化が行なわれる。図 19 に構成を示す。処理データは単方向性リソグバスを経由しながら各モジュールで流れ作業的に処理を受ける。Addressor Unit は主にメモリからのデータ読み出し、メモリへのデータの書き込みを行なう。Operational Unit は浮動小数点演算によるデータ処理を行なう。Control Unit はホスト計算機とのデータ授受、パイプライン路の初期設定などを担当する。

リンクバス上を移動するデータは42ビットであり、CMF(初期設定フラグ):1ビット、MSF(モジュール選択フラグ):10ビット、ID(データ名):5ビット、ERR(エラー状態):2ビット、DAT(データ値):24ビット、から構成される。画像処理動作中のデータの行き先はMSFで決定される。各モジュールは異なるIDを持つデータに異なる処理を与えることができるようにIDごとの初期設定値を保存し、IDごとにデータ駆動制御を行なう。行き先モジュールに到着したデータはID別に処理を受け、その処理結果(ERR, DAT)はID別に保存されている新しいMSF, IDと共にリンクバスへ送出されるのである。このようにテンプレート(各モジュールで、各IDに対して必要な制御情報)によって各モジュールがリンクされ、リンクバスを転送路としたパイプライン路が形成されるのである。

テンプレート を各モジュールに割り当てるために関数型のアセ
 ソブラ言語が開発されている。⁵⁰⁾ データフローグラフにおける
 ノードを関数に、枝を変数に割り当てればよい。たとえば、
 図20のようなFFT処理を

プログラミングしたのである。

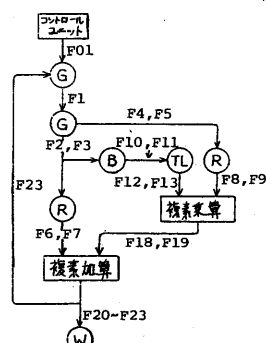


図20. FFTのデ-タフロー-グラフ⁵⁰⁾

TIPの速度の目標値

は、パイプラインサイクル
100ns(乗算器は150ns),
リングバスの転送速度50ns
で、FFTを16MFLOPS
で実行する。ただし実
際にはバスビジーなどの
現象によりやゝ速度は落

ちるようである。図19の
Operational Unitを見てもわかるように各演算モジュールの並
べ方はかなりFFTのバタフライ演算向きにできているように
思える。FFT以外のオペレーションも実行の際、果たしてこの

並べ方で良いのかどうかが検討の余地がある。応用例としては SAR データ処理などが挙げられているが、文献 50) でも最後に触れられているように、初期化終了と同時に切り離されてしまう Control Unit を積極的に処理の流れの中に組み込むことにより、単に全画面一様な処理だけでなく複雑な処理も高速に行なう可能性も出てくる。今後の研究に期待したい。

4. おわりに

以上、画像処理プロセッサの最近の動向について解説した。フレームメモリ型の実例は割愛したが、最近では画像表示装置でかなり高度な処理能力を持ったものが製品化されており、注目値する。

単純に全画面一様な処理を施すプロセッサは今後も着実に進歩するであろう。問題は現在のところ汎用計算機で実行するしか方法のない高レベルで時間のめぐる処理をいかに効率よく計算するかであろう。また、ソフトウェアの問題も忘れてはいけない。このままでは画像処理プロセッサの数だけ画像処理言語が作られてしまう。

最後に簡単な 性能比較を試み た表3に示す。	プロセッサ名	計算時間 msec	画像サイズ
	PPP	262	512x512
	DIP	180	256x256
	PICAP	14.7	64x64 (増設4bit)
	CLIP4	20	96x96
	MPP	5.4	512x512 (25x128 分割処理)
	(Smart Sensor Test Chip III)	33.3	512x512
	(ESL Systolic Processor)	10.6	128x128
	FLIP	250	512x512

表3. 濃淡画像(8bit) 3x3コンボリューション計算時間。

【参考文献】

- 1) 木戸出, 篠田, "デジタル画像高速処理装置の開発の流れを辿る", 日経エレクトロニクス, No. 191, pp. 110-140, 1979.
- 2) 木戸出, "画像処理ハードウェア", 情報処理, Vol. 21, No. 6, pp. 620-625, June 1980.
- 3) 木戸出, "画像処理ハードウェアの動向", 情報学会, コンピュータビジョン研究, CV 8-6, September 1980.
- 4) 木戸出他, "並列画像演算装置PPP", 東芝レビュー, vol. 34, No. 6, pp. 511-514, 1979.

INPUT F01, F02, F03, F04, F05
 $F1 = G(F01, F23/7F, 7F, 0/2, 0/1, W, 3)$
 $F2, F3, F4, F5 = G(F1/3, 0, 0/0, 0/1, W, 3)$
 $F6 = R(F2/0, 0, W, 0)$
 $F7 = R(F3/1, 0, W, 0)$
 $F8 = R(F4/40, 0, W, 0)$
 $F9 = R(F5/41, 0, W, 0)$
 $F10 = BO(F2/REV, 0)$
 $F11 = BO(F3/REV, 0)$
 $F12 = TL(F10/0, 3F)$
 $F13 = TL(F11/0, 3F)$
 $F14 = MO(F8, F12/Q)$
 $F15 = MO(F9, F13/Q)$
 $F16 = M1(F9, F12/Q)$
 $F17 = M1(F8, F13/Q)$
 $F18 = AO(F14, F15/Q/SUB)$
 $F19 = AL(F16, F17/Q/ADD)$
 $F20 = AO(F6, F18/Q/SUB)$
 $F21 = AL(F7, F19/Q/ADD)$
 $F22 = AL(F6, F18/Q/SUB)$
 $F23 = AO(F7, F19/Q/ADD)$
 $W(F20, F02/V/4, FE2FFF, 20/B, 0)$
 $W(F21, F03/V/4, FE2FFF, 20/B, 0)$
 $W(F22, F04/V/4, FE2FFF, 20/B, 0)$
 $W(F23, F05/V/4, FE2FFF, 20/B, 0)$
 END

図21. FFT テンプレートプログラム⁵⁰⁾

- 5) 麻田他, "二次元フーリエ変換機能などを持った並列画像処理装置の開発", 信学研資, IE78-12, May 1978.
- 6) 河原他, "マイコン・制御の事前処理装置を持つ画像処理システム", 信学大全 959, March 1976.
- 7) 松島他, "A1方式を用いた映像処理装置の開発", 信学研資, IE78-11, May 1978.
- 8) 鈴木他, "顕微鏡画像処理用高速二次元プロセッサ", TV学会全国大会資料, 14-7, 1980.
- 9) R.P.W. Duin, et al., "The Delft image processing system, design and use," Proc. 5th ICPR, pp. 768-774, 1980.
- 10) S.H. Unger, "A computer oriented toward spatial problems," Proc. IRE, vol. 46, no. 10, pp. 1744-1750, 1958.
- 11) B.H. McCormick, "The Illinois pattern recognition computer ILLIAC III," IEEE Trans. vol. EC12, pp. 771-813, 1963.
- 12) W.J. Bouknight, et al., "The ILLIAC IV system," Proc. IEEE, vol. 60, no. 4, pp. 369-388, April 1972.
- 13) B. Kruse, "The PICAP picture processing laboratory," Proc. 3rd IJCPR, pp. 875-881, November 1976.
- 14) J.L. Potter, "The STARAN architecture and its application to image processing and pattern recognition algorithms," Proc. '78 NCC, vol. 47, AFIPS, pp. 1041-1047, 1978.
- 15) 後藤他, "多値ビット構成連想プロセッサ DREAM-II のハードウェア", 信学研資, EC 79-64, pp. 41-50, 1980.
- 16) 市川他, "連想プロセッサARES", 信学研資, EC77-66, 1977.
- 17) M.J.B. Duff, "CLIP4: a large scale integrated circuit array parallel processor," Proc. 3rd IJCPR pp. 728-733, November 1976.
- 18) D.J. Hunt, "The ICL DAP and its application to image processing," in Languages and Architectures for Image Processing, M.J.B. Duff and S. Levialdi, eds., pp. 275-282, Academic Press, New York, 1981.
- 19) K.E. Batcher, "Design of a massively parallel processor," IEEE Trans. C-29, No. 9, pp. 836-840, Sept. 1980.
- 20) A.P. Reeves, "A systematically designed binary array processor," IE3 Trans. vol. C-29, No. 4, pp. 278-287, April 1980.
- 21) A.P. Reeves and R. Rindfuss, "The BASE8 binary array processor," Proc. Pattern Recognition and Image Processing '79, pp. 250-255, August 1979.
- 22) 日経エレクトロニクス 1980.12.9, pp. 74-80.
- 23) S.R. Sternberg, "Parallel architectures for image processing," in M. Onoe et al (eds), Real-time parallel computing - Image Analysis, pp. 347-359, Plenum Press, 1981.

- 24) K. Luetjen, P. Gemmar and H. Ischen, "FLIP: a flexible multiprocessor system for image processing," *Proc. 5th ICPR*, pp. 326-328, Dec. 1980.
- 25) G.R. Nudd, "Image understanding architectures," *Proc. of National Computer Conf.*, pp. 377-390, 1980.
- 26) G.R. Nudd, et al., "A charge-coupled device image processor for smart sensor applications," *SPIE*, vol. 155, pp. 15-22, 1978.
- 27) G.R. Nudd, et al., "Charge coupled device technology for smart sensors," *Proc. of Image Understanding Workshop*, pp. 16-21, May 1978.
- 28) G.R. Nudd, et al., "Implementation of advanced real-time image understanding algorithms," *ibid.*, pp. 151-157, April 1979.
- 29) G.R. Nudd et al., "Development of custom-designed integrated circuits for image understanding," *ibid.*, pp. 1-9, November 1979.
- 30) S.D. Fouse et al., "Application of LSI and VLSI to image understanding architectures," *ibid.*, pp. 190-194, April 1980.
- 31) T.J. Willett and N. Bluzer, "CCD implementation of an image segmentation algorithm," *ibid.*, pp. 9-11, October 1977.
- 32) N. Bluzer et al., "A CCD histogram sorter: feasibility chip," *ibid.*, pp. 7-8, October 1977.
- 33) T.J. Willett et al., "Relaxation, systolic arrays and universal arrays," *ibid.*, pp. 164-170, April 1979.
- 34) W.L. Eversole et al., "Investigation of VLSI technologies for image processing," *ibid.*, pp. 159-163, April 1979.
- 35) W.L. Eversole et al., "Investigation of VLSI technologies for image processing," *ibid.*, pp. 10-14, November 1979.
- 36) H.T. Kung and C.E. Leiserson, "Systolic arrays for <VLSI>," *TR. CMU-CS-79-103*, CMU, Department of Computer Science, April 1978.
- 37) H.T. Kung, "Let's design algorithms for VLSI systems," *ibid.*, *TR. CMU-CS-79-151*, January 1979.
- 38) C. ミード, L. コンウェル 共著, 菅野卓雄, 神谷之監訳, "超LSI システム入門," 培風館, pp. 299-318, 1981.
- 39) H.T. Kung et al., "A two-level pipelined systolic array for convolutions," in H.T. Kung et al. (eds.), *VLSI Systems and computations*, pp. 255-264, Computer Science Press, 1981.
- 40) H.T. Kung, "Special purpose devices for signal and image processing: an opportunity in VLSI," *TR. CMU-CS-80-132*, CMU, Department of Computer Science, July 1980.
- 41) D.W.L. Yen and A.V. Kulkarni, "The ESL systolic processor for signal and image processing," *Proc. Computer Architectures for Pattern Analysis and Image Database Management*, pp. 265-272, November 1981.
- 42) H.T. Kung, "Why systolic architectures?," *IEEE Computer*, Vol. 15, No. 1, pp. 37-46, January 1982.
- 43) H.J. Siegel, "PASM: a reconfigurable multi-microcomputer system for image processing," in *Languages and Architectures for Image Processing*, M.J.B. Duff and S. Levialdi (eds.), pp. 257-265, Academic Press, New York, 1981.
- 44) F. Briggs et al., "PM⁴— a reconfigurable multi-processor system for pattern recognition and image processing," *AFIPS Conf. Proc. vol. 48, NCC*, pp. 127-137, 1979.
- 45) C. Rieger, "ZMOB: a mob of 256 cooperative Z80A - based microcomputers," *Proc. Image Understanding Workshop*, pp. 25-30, Nov. 1979.
- 46) C. Rieger, "ZMOB: hardware from a user's viewpoint," *Proc. Pattern Recognition and Image Processing*, pp. 399-407, August 1981.
- 47) 坂井, 稲垣, 加藤, "フクシマ端末をもつパソコン理解並列処理装置," *信学研資*, PRL 80-98, 1980.
- 48) 佐藤, 松浦, 小川, 飯島, "パターン情報処理用マルチマイクログロッサシステム PX-1," *信学論(D)*, vol. J 64-D, No. 11, pp. 1021-1028, November 1981.
- 49) 天満, 溝口, 花木, "画像処理プロセッサ TIP の構成とシミュレーション評価," *信学研資 IE 81-6*, April 1981.
- 50) 天満, 溝口, 花木, "データ駆動型プロセッサ TIP-1 での画像処理プロセッサ," *信学研資 PRL 81-62*, December 1981.
- 51) G.R. Allen et al., "SPARC— symbolic processing algorithm research computer," *Proc. Image Understanding Workshop*, pp. 182-190, November 1978.