

高速画像処理向きセルラ・アレイ・プロセッサ

CELLULAR ARRAY PROCESSOR FOR HIGH-SPEED IMAGE PROCESSING

宮田裕行 磯西徹明 菅 隆志 岩瀬 正
Hiroyuki MIYATA Tetsuaki ISONISHI Takashi KAN Akira IWASE

三菱電機 情報電子研究所

Mitsubishi Electric Corporation Information Systems & Electronics Development Laboratory

1. まえがき

近年、画像処理に対する高速化の要求が急速に高まりつつある。特に衛星から送られてくる画像のように、その規模が大きく、また処理が複雑で多様なものに関しては、その傾向が強い[1]。そのため現在では CRAY-1 などのスーパーコンピュータと呼ばれる高速な計算機を用いてその処理を行う存じしている[2]が、これでもまだ十分にその条件を満たしているとは言えない。そもそも画像処理とは、大量な2次元データに対する処理であり、逐次型の計算機が不得手とするところがあるのであるが、その一因としてあげられる。そのため、近年、画像処理に向けたアーキテクチャに基づいた様々な専用プロセッサの研究開発が活発に行われている[3]。この画像処理プロセッサを分類すると次の4種になる[4]。

- (1) 完全並列型、連想処理型
- (2) 局所並列型、画像ディスプレイ型
- (3) パイプライン型
- (4) マルチプロセッサ型

(1) の完全並列型の代表的なものにセルラアレイ方式がある。この方式は上記4項目の中で、最も並列度が高く、高速化が期待できるアーキテクチャの1つである。しかしながらデータの入出力に時間がかかる、柔軟性に欠くなどの問題点も指摘されている。

本論文においては、これらの問題点を解決したセルラアレイ方式に基づく並列プロセッサ CAP (Cellular Array Processor) を提案す

る。また提案した CAP の性能評価を行い、画像処理への適合性を検討する。

以下、第2章では、セルラアレイ方式のアーキテクチャを紹介し、その特徴および問題点について述べる。第3章では、本論文で提案する CAP のアーキテクチャの概要を述べる。第4章以降では、セルラアレイ方式の問題点を解決するための CAP の様々な制御方式について説明する。まず第4章では、入出力の問題を解決するため、ブロックパイプラインと呼ぶ制御法について述べ、第5章では、MIMD 的な要素を持たせるための付加機能について、第6章では発火方式と呼ぶ、CAP 特有の制御方式を、第7章では、パケット転送を用いた並列データ転送方式を述べる。第8章で、各種画像処理に対する CAP の性能評価を行い、第9章を、セルラアレイ方式に対する考察に、本章をあてはきにあてる。

2. セルラアレイ方式

セルラアレイ方式とは、単純な演算機能を持ったセルと呼ばれるプロセッサを、多数、2次元格子状に配列したアーキテクチャであり、全セルが同時動作することにより、高速な処理を実現するものである。図1にその一例を示す。このアーキテクチャを用いて画像処理を行う場合、例えば、画像の1画素と1セルを対応させて行う。

さて画像処理とは、画素の2次元配列から成る画像に対する処理であり、次に示す特徴を持つ

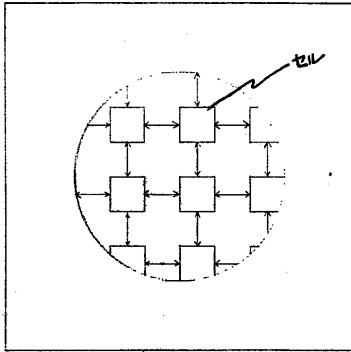


図1 セルラレイ方式の例

つ。

(1) 近傍処理：画像処理の中には、各画素とその周囲の画素を用いて行う近傍処理演算が多い。また、それらは各画素単位に独立して並列処理が可能である。

(2) 2次元データ処理：上記の近傍処理を初めとし、画像処理では、2次元状に存在するデータへのアクセスおよびその処理が複雑にある。

(3) 多様なビット画素に対する処理：対象画像により、画素を表現するビット数が異なる。通常の逐次実行型の計算機を使用して画像処理を行う場合の上記3点について考察する。まず近傍処理では、並列処理が可能であるにもかかわらず、順次、その処理を行わねばならず、またその画素数に比例した回数だけ実行しなければならぬ。2次元データ処理では、画像は通常、行単位あるいは列単位にメモリに格納されるため、その2次元アクセスには、多くのアドレス計算が必要となる。多様なビット画素データに対しては、常に同一の演算器を用いるため処理の効率が悪い。

セルラレイ方式を用いた画像処理では、一度、画像が取り込まれた後は、演算のためにメモリにアクセスする必要がなく、容易に近傍データを取得することができ、処理が行える。またそれらの処理がセル数だけ並列に実行できる。更に2次元データ処理に関しては、セル自体が2次元状に配置されているため、無駄なアドレス計算が不要となる。多様なビット画素に対しても各セル単位にビット単位の演算を行うため、

効率よく処理ができる。

以上の様に、セルラレイ方式は、本質的に画像処理に向いており、現在までのところ、

MPP〔5〕、DAP〔6〕、CLIP〔7〕、AAP〔8〕などのプロセッサが開発されている。各々、 64×64 から 256×256 個程度のセル数から構成されている。

ところで以上示したセルラレイ方式は、その並列性を活かし、高速な演算が可能ではあるが、次に示す問題点がある。

(1) プロセッサへのデータの入力、あるいはデータの出力に時間がかかる。

(2) SIMD方式であるため、全セルが同時には同じ命令を実行することになり、柔軟性に欠ける。

(3) セル間の接続は、隣接するセルとの間に限られるため、離れたセルどうしのデータ転送が行いにくい。

本論文で提案するセルラレイ方式に基づくCAPは、これらの欠点を解決し、より高速な画像処理を目指している。

次章において、このCAPの構成について述べる。

3. CAPアーキテクチャ

3.1 CAPの全体構成

図2にCAPの全体構成を示す。破線が囲まれている部分がCAPのシステム構成であり、演算を行うCAP、記憶装置との入出力を行う3つのI/Oバッファ、入出力制御部、各々を制御するCAP制御部から成る。またシステム全体は、ホストコンピュータのホストで動作する。更に処理の対象となるデータを保存し、2次元アクセスが可能な記憶装置を周辺に接続する。

通常の処理では、記憶装置から処理対象であるデータが読み出され、CAPの右端にあるI/Oバッファを経て、CAP自身に転送される。CAPでの処理が終了した後、そのデータは、今度はCAPの左端にあるI/Oバッファを経て再び、記憶装置に格納される。CAPの上端にあるI/Oバッファについては後述する。

3.2 CAPの内部構成

一般的なセルラレイ方式では、各セルは隣

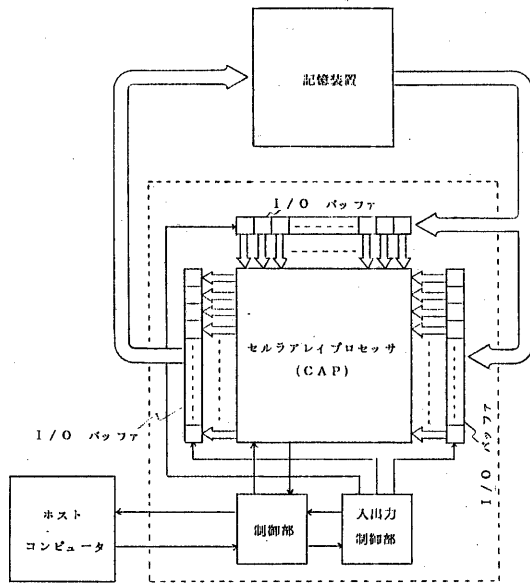


図2 CAPの全体構成

接する4個、あるいは8個のセルと接続されている。しかしながら隣接する4個のセルとの接続では、セル間のデータ転送に不都合な場合が無く、また隣接する8個のセルとの接続では、全体としてのセル間接続配線が複雑になる。

この欠点を回避し、更に次章で述べるブロック・パイプラインを可能とするため、CAPでは図3に示す構成をとる。図3において、Pで示すのが演算を行うプロセッシング・セル(Pセル)であり、Sで示すのがPセル間のデータ転送およびI/Oバッファとデータ転送を行うスイッチング・セル(Sセル)である。この様に、セルをその機能により明確に分離することによって、次に示す利点が生じる。

(1) Pセルは隣接する4個のSセルと接続されているだけであるが、他のPセルを経ることなく、周囲の8個のPセルとのデータ転送が可能である。

(2) I/Oバッファと各Pセル間のデータ入出力を、演算、セル間データ転送とは独立したSセルの一部を使用することにより行えるため、演算と入出力のオーバーラップが可能となる(第4章 ブロック・パイプライン参照)。

すなわちSセルはPセル間のデータ転送とI/O

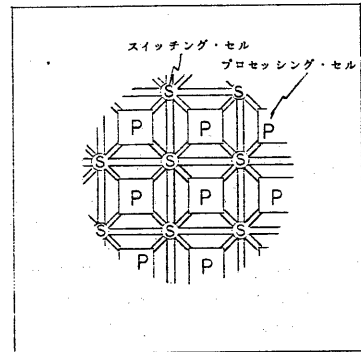


図3 CAPの内部構成

バッファとの入出力を独立して行うことができる。

3.3 プロセッシング・セル(Pセル)

Pセルにおけるデータ処理は、ビットシリアルに行う。これは各セルの構成を複雑にすると多数のセルから構成されるCAPにとって、その設計が困難になることと、任意ビット長に対する処理の効率化のためである。例えば9ビット長のデータを処理する時、ビットシリアルに行うか、8ビットALUを用いて行うかを考えれば明らかであろう。そのためPセルは、演算のための1ビット全加算器、レジスタファイル、RAMなどから構成される。

3.4 スwitching・セル(Sセル)

先に述べた様に、Sセルの機能には、Pセル間のデータ転送、I/Oバッファを介してのデータ入出力がある。Pセルがビットシリアル動作であったのと同様に、Sセルの動作もビットシリアルに行う。

更にSセルの付加機能として、ブロードキャスト機能がある。このブロードキャストとは、正方形格子状に配置されたPセルの行単位、列単位あるいは全セルに一定の値を外部から送ることを言う。例えば、列単位に一定値をブロードキャストする時は、先の3.1節で示したCAPの上端にあるI/Oバッファを使用し、列ごとのブロードキャストする値をI/Oバッファの対応

する部分に格納し、上端から各列単位にその値を送る。行単位のブロードキャストも同様に右端のI/Oバッファを使用して行う。

4. ブロック・パイプライン

4.1 CAPの入出力方式

CAPが記憶装置のデータをI/Oバッファを経て入力したり、逆に出すするにはSセルを使用した。以下、この方法について詳細に述べる。説明のために4×4個のPセルおよびSセルから成るCAPを取り上げる。図4にこの概略図を示す。破線内が示した部分が4×4個のセルから成るCAPである。CAPはトラスと言ひ、右端の列のSセルと左端の列のPセル、上端の行のSセルと下端の行のPセルが、各々、接続されている。

さて、記憶装置にある4×4個の画素から成る画像をCAPに入力する場合は、以下の手順で行う。

- (1) 記憶装置内の4×4画素のうち、左端の1列から順に列単位にCAPの右端のI/Oバッファに送る。
- (2) CAPの右端のI/Oバッファでは、順に記憶装置から送られてくる画素値の列を、CAPの右端の対応する行のSセルに送る。
- (3) CAP内のSセルでは、順に画素値を左シフトする。
- (4) 4×4個のSセルに対して、4×4個の画素値を送るため、最後の画素値の列が、

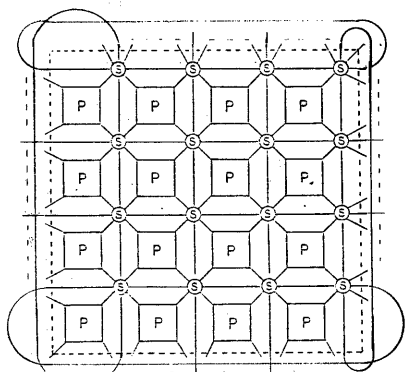


図4 4×4個のセルから成るCAPの例

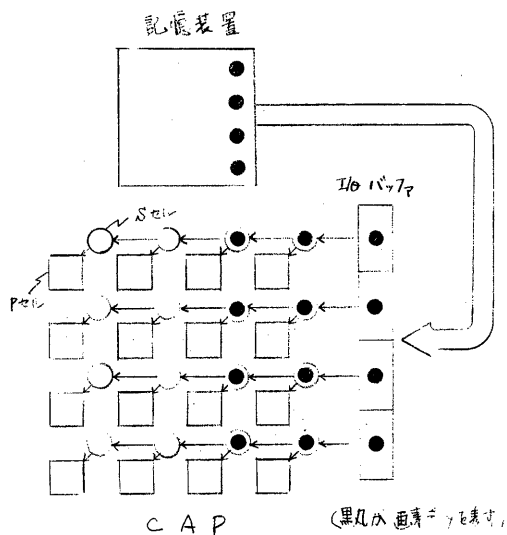


図5 CAPのデータ入力例

CAPの右端のSセルに転送されたところで各Sセルが対応した画素値を所有する。ここでSセル間のシフトは終了する。

(5) 各Pセルの右上のSセルに、対応する画素値が到達しているため、この画素値をPセルに送る。これにより画素値が各Pセルに格納され、演算が開始できる。

図5に画素値がCAPに入力されている途中の状態を示す。

出力に関しても、まったく同様であり、この逆の操作をCAPの左端のI/Oバッファを使用して行えばよい。

なお、CAPのSセル内では、ダブルバッファリングを用いて、この入出力をより高速に行っている。すなわち入出力用のバッファを各セル内で2つ用意し、常に記憶装置とのデータのやりとりが行える様になっている。

4.2 ブロック・パイプライン

CAPでの画像処理は1つのPセルに1つの画素値を対応させて処理を行うことを基本とする。ところが対象とする画像が必ずしもCAPのセル数以内の画素から構成されているとは限らないため、この大画像(画素数がCAPのセル数以上の画像)に対する処理が問題となる。1つの考え方としては、1つのPセルに複数の

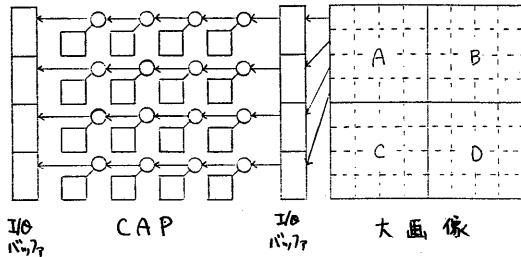


図6 CAPによる大画像の扱い

画素を格納することにし、画像全体を一度、CAPの中に取り込み、処理する方法がある。ところが、この方法には、次に示す欠点がある。

(1) 1セル内(Pセル)のデータ数が多くなるため、レジスタ等の使用に比べ、セル内のRAMへのアクセスの回数が増し、不要なメモリアクセス時間を要する。また余分なメモリ量を必要とする。

(2) 近傍データは、ほとんどセル内に存在することになり、セル間接続の効果が薄れる。

(3) 各セル内には1つの全加算器だけがあり、これにより相当数の画素を逐次処理することになるため、セルラアレイ方式の効果が薄れる。

(4) CAPと記憶装置とのデータの入出力に時間がかかる。

以上の欠点を解決するため、ここではブロックパイプラインと呼ぶ方式を提案する。説明のため、先に用いた4×4のセルから成るCAPで8×8の画素から成る画像を処理する場合を考察する。これを図6に示す。

まず対象画像をセルの数に合わせて、4×4の画素から成る部分画像A、B、C、Dに分割する。そしてCAPへの入力、演算、出力を、この部分画像ごとに繰り返す。すると演算時には1セル1画素対応となるため、上記の(1)、(2)、(3)の欠点は解消できる。しかしながら(4)の入出力に必要となる時間は同じである。そこで各部分画像を処理する時に、既に演算が終了した部分画像の出力、現在の部分画像の演算、次に演算する部分画像の入力をすべて時間的にオーバーラップさせる様にす。

以下タイムチャートを用いて説明する。図7にこれを示す。図7(a)は、CAPに画像をすべて格納してから演算を行う場合を示す。全画

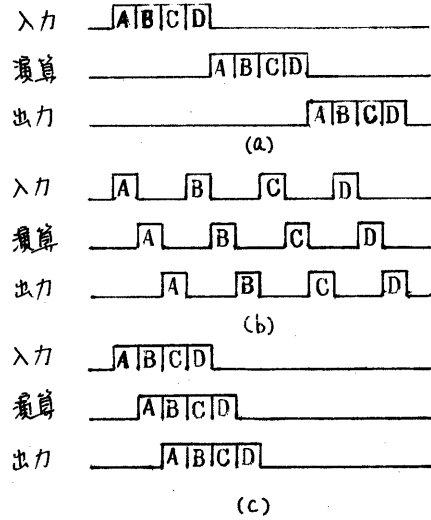


図7 入出力と演算のタイムチャート

像を一度、CAPに入力するため、入力、演算、出力が各々、別々に行われる。図7(b)は、部分画像A、B、C、Dごとに、入力、演算、出力を行った場合である。全体の処理に必要時間、図7(a)とまったく変わっていない。すなわち全体の処理時間中、演算は1/3であり、2/3は入出力に必要となる。

図7(c)が本論文で提案するブロック・パイプラインを採用した場合である。演算と入出力をオーバーラップさせているため、最初の部分画像Aの入力と、最後の部分画像Dの出力を除いて、常にCAPが演算を行っている。そのため全体の処理時間が、図7(a)、(b)に比べ、約半分になっている。

図8に図6の例を用いて、ブロック・パイプライン処理を行っている時のCAPの1状態を示す。以下に説明する。

- ・演算の終了した部分画像Aが、記憶装置に転送されるため、CAPのSセル間をシフトしている(出力中)

- ・部分画像Bが演算中である(演算)

- ・次に演算されるべき部分画像Cが、記憶装置から転送されるため、CAPのSセル間をシフトしている(入力中)

ここに示す様に、入出力機能と演算機能が独立に動作でき、一度にCAPで処理が可能で単位

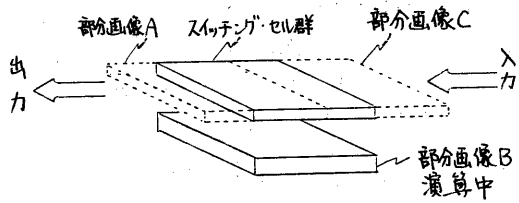


図8 ブロック・パイプラインの例

(これをブロックと呼ぶ、図6の例では1ブロックが4×4画素と作る)ごとに、パイプライン処理を行うことをブロック・パイプラインと呼ぶ。このブロック・パイプラインを採用することにより、本来、セルアレイ方式の入出力に時間がかかるという欠点を解消できる。

但し問題点として、ブロック当りの入出力時間と演算時間との兼ね合いがある。入力と出力は、同じ大きさのブロックを、同じ転送レートのクロックでSセル間シフトを行うため、まったく等しい時間を必要とするが、演算時間はその処理対象により異なる。例えば、図7(c)のタイムチャートにおいて、もし、1ブロックの入出力時間よりも、演算時間の方が短かかれば、各ブロックの演算後、入出力の終了を行つことになり、この間、CAPでの演算は行われぬ。これが、結局、性能低下につながる。この解決策としては、CAPは演算と入出力をPセルとBセルに分離して行うため、例えば、Sセルの入出力クロックレートを、Pセルの演算クロックレートよりも高速にすることが考えられる。この比較に関しては、第8章の性能評価の章で行う。

5. 命令修飾とアドレス修飾

セルアレイ方式に基づくプロセッサは、本来、SIMD型であるため、全セルで実行される命令は、同一のものであり、また各セル内にあるメモリに対するアドレスも同一である。ところが、各セル内の値により、次に実行すべき命令およびアドレスが異なってくることは数多くあり、これを単紀形SIMD型のプロセッサで実行することは困難である。

例えば、絶対値を求める演算を考えてみる。

今、各セルにおいて、レジスタAの値の絶対値を求め、その結果を、レジスタBに転送することとする。この場合は、レジスタAの値をチェックし、正の値ならばそのままレジスタBに転送し、負の値ならば正の値に変換して転送することにする。すなわち、レジスタAの値により次に実行すべき演算が異なってくる。

また浮動小数点加算を取り上げてみる。この場合、各セル内で、データの指数部の値により仮数部の桁合わせを行うが、この桁合わせは、各セル内のデータの値により異なる。すなわち加算命令のオペランドアドレスが、セル内の値により異なってくる。

以上、示した既存処理を、単純なSIMD型のプロセッサで行うことは、非常に困難であるため、CAPにおいては次に示す既存修飾機能を各セル内に設けた。

5.1 命令修飾

各セル内に数ビットの状態フラグを設け、この値により、与えられた命令を修飾する。例えば次に示す様な機能がある。

- (1) 与えられた命令を、そのまま実行するか、修飾して実行するか、あるいは無効命令として処理するかを定める。
- (2) 修飾して実行する時は、ある与えられた命令がセルにより異なった命令となり実行される。例えば、1つの命令がセルにより、加算命令、減算命令、転送命令となる。
- (3) ある命令を実行している時に、何らかの条件を満たした場合(例えばゼロチェック、オーバーフローチェックなど)、その情報を選別して、そのセルだけが以後の演算を中止する。これらの機能を使用することにより、例えば、先に示した絶対値演算などを効率よく、高速に処理することが可能となる。

5.2 アドレス修飾

各Pセル内にあるレジスタに、アドレス修飾機能を持たせた。すなわちレジスタに対するアドレスが全セルに共通のアドレスでもよいし、各セル単位に作成されたアドレスを使用してもよい既存修飾機能を付加した。

この機能を使用することにより、先に示した浮動小数点加算などを高速に行える。

6. 発火制御方式

前章までに述べてきたCAPの制御方式を用いることにより、大半の画像処理は高速に処理することができる。しかしながら、例えば、ある2点が指定されて、その間を最短経路を選択する、という処理は、これまでの手法では不可能である。そのため、ここでは発火制御方式と呼ぶCAP特有の制御方式を提案する。

これは、あるセルで発火信号が起り、その信号が順次、隣接するセルに伝播していき、最終的に求めるセルに達したところで発火信号が沈火し、これにより2点間の経路などを検索するものである。以下にこの制御方法について説明する。

各セルには、次の3つのフラグが存在する。

(1) S: 状態フラグ

各セルは、次に示すどれかの状態をとる。

(a) 活性状態: 発火信号を受け取った時、周囲の隣接セルに、その信号を伝播する($S=00$)。

(b) 一時不活性状態: 発火信号の伝播を終了し、再び信号を受け取っても伝播は行わない($S=01$)。

(c) 永久不活性状態: 発火信号を受け取っても伝播を行わない($S=10$)。

(2) E: 種類フラグ

最終的に目的とするセルであれば'1'、そうでない通常のセルは'0'となる。

(3) F: 発火信号伝播フラグ

発火信号を受け取れば'1'、通常は'0'である。この動作を示す前に、CAPでのSum-OR-Tree [5]、Sum-AND-Treeについて説明する。各々はあるセルのある指定した1ビットの論理和、論理積を全セルについて求めるものである。

図9に各セルでの動作を示す。但し、最初に発火が起きてから、最終目的セルに達するまでの動作のみである。図中、Sum-AND-Treeが'1'となった時、すべてのセルが発火信号を受け取っていないこととなるため、発火信号は消失している。すなわち、最終目的セルに発火信号が達することなく処理が終了したこととなる。またSum-OR-Treeが'1'となった時には、逆に目的セルに信号が到達したためであり、いわゆる探索が成功したことを示す。〔処理〕は、応用例により定まる処理を示す。

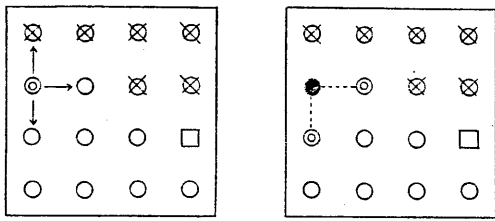
```

if F=0 (発火信号を受け取らない)
then '1'をSum-AND-Treeに送る
else if F=1 (発火信号を受け取る)
then begin
if E=1 (最終目的セルである)
then '1'をSum-OR-Treeに送る
else if S=00 (活性状態である)
then begin
S ← 01 (一時不活性状態とする)
〔処理〕
隣接する4近傍セルのF ← 1
end
end
end
    
```

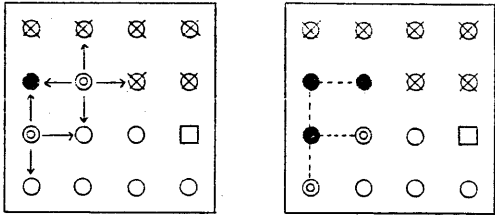
図9 発火制御方式によるセルの動作

以下に経路探索の場合を例にとり説明する。図10に4×4のPセルから構成されるCAPの例を示す。説明のため、Sセルは省略する。各セルを区別するため(i, j) (i: 行を示す, j: 列を示す i, j=1~4)の記法を用いる。例えば、左上のセルは(1, 1)、右下のセルは(4, 4)、右上のセルは(1, 4)となる。各セルの状態フラグ、種類フラグは事前に設定しておく。

さて図10(a)では、(2, 1)のセルで発火信号が発生した様子を示す。この信号は、上下、左右の4隣接セルに伝播する(CAPは8近傍のセルに同時にデータを送れるが、ここでは4近傍としE)。(2, 2), (3, 1)のセルは、活性状態であるため、処理が行われる。ところが(1, 1)のセルは、永久不活性状態であり、発火信号が伝播しても、以後にその信号は伝わらない。図10(b)では、(2, 1)のセルから発火信号が伝播した後、そのセル自身は一時不活性状態となり、既に発火信号が伝播したという情報をもつ。そのため以後にこのセルに発火信号が伝わっても、伝播は生じない。またセル(2, 1)ーセル(2, 2)、セル(2, 1)ーセル(3, 1)間に伝播信号が流れたという記録が残される(図10(b)の破線)。図10(c)では、発火信号が伝わり、処理が行われた(2, 2), (3, 1)のセルから、信号の伝播が生じる。先と同様に、活性状態のセルに対してのみ行われる。図10(d)では、図10



(a) (b)



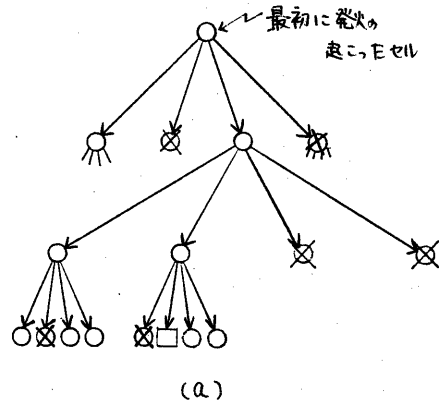
(c) (d)

- 活性状態 ● 一時不活性状態
- ⊗ 永久不活性状態
- ◎ 発火セル □ 最終目的セル

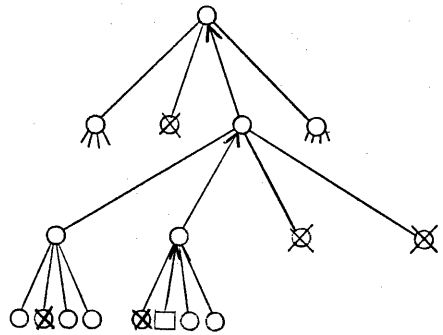
図10 発火方式制御の例

(c)の発火信号伝播により、それまでに生じた一時不活性セルおよび各セル間の発火信号伝播経路の記録を示す。但しセル(3,2)は、セル(2,2)および(3,1)のセルから信号が伝播するが、ここでは、どちらか一方のみを残すことにする(図10(d)では、セル(3,1)→セル(3,2)の経路のみを残す)。これは、発火信号の伝播経路をトリ-状にすることによる。詳しくは、後述する。

以下、同様の信号伝播を繰り返していき、最終目的セル(図10(d)ではセル(3,4))に達するまで行う。もし最終目的セルに達するまでにこの発火信号が消滅した場合、求めべき経路が存在しないことになる。さて、最終目的セルにまで信号が伝播した場合は、それまでに得られた経路の中から、最初に発火の起こったセルと最終目的セルを結ぶ経路を定める必要がある。これは、最初に発火を起こしたセルから経路がトリ-状に広がっていることによる。これを図11(a)に示す。すなわちトリ-の根に当たる最初の発火を起こしたセルから、信号は最大隣接する4個のセルに順次伝播する。これが先に示した様にループを形成することなく、順



(a)



(b)

図11 発火制御方式による経路

に伝播していくため、各ノードが最大4個の枝を持つたトリ-となる。発火信号が最終目的セルに達した時点では、図11(a)に示すトリ-内の枝がすべて存在するため、この中から、トリ-の根(最初に発火したセル)と最終目的セルを結ぶ経路を見つける。これを図11(b)に示す。すなわち最終目的セルから逆に経路をたどり、最初のセルまで戻す。その際の経路が求めべき経路である。

以上示した発火制御方式は、いわゆるトリ-の探索問題を解いていることになる。通常は、この探索問題の解法に、複雑なアルゴリズムとかなりの処理時間を必要とするが、CAPを使用することにより、単純なアルゴリズムで、かつトリ-の探索が並列に行えるため、高速に処理が可能となる。この発火制御方式を用いることにより、2点間を結ぶエッジの検出、線分の

検出などの画像処理が実現可能となる。

7. 並列データ転送方式

本章では、セルラアレイ方式の欠点である離れたセルどうしのデータ転送の困難さを、CAPでいかに解決するかについて述べる。特に問題となるのは、各セルからの送り先が一定でなく、各セルごとに異なる場合である。

CAPでは、この問題を解決するために、パケット転送による並列データ転送も可能である。但し各セルは、自身のアドレスを所有しているものとする。以下にアルゴリズムを示す。

(1) 各セルにおいて、送出するデータに対してその送り先のセルの位置情報を付加する。

(2) 以下、各セル内部のデータの位置情報により

(a) 隣接する東のセルへデータを送る。

(b) 隣接する北のセルへデータを送る。

を繰り返す。但し各セルには、待ち行列のバッファを数個設け、送られてきたデータを、あいている限り、このバッファに取り込む。(あるいはルーティングにより、この経路を迂回する) 各データ転送ごとに、該当するデータを受け取ったセルでは、内部にその値を取り込む。

(3) 上記(2)の処理を、全データが目的とするセルに到達するまで行う。

なお、ここでは4.1節で述べたトーラスの構成を使用する。

通常の幾何補正を別にシミュレーションを行ったところ、待ち行列のバッファは2個あれば十分なことがわかった。

以上述べた並列データ転送方式を用いることにより、SIMD型のセルラアレイ方式では困難であった遠方のセルとのセル単位に送り先の異なるデータ転送が可能となる。

なお上記(1)では、各セル内において、既に行き先アドレスが決まっていると仮定したが、通常は(1)各セルのアドレスのIP-シミュレーション(2)各セルの機能により定められる値などが用いられる。

8. 性能評価

本章では、第4章で示したCAPにおける入出力時間と演算時間について、各種の画像処理を対象として考察する。また論理シミュレーションによる同様の結果をもとに、両者の整合性を検討する。更にそれらの結果を踏まえ、画像処理におけるCAPの総合評価を行う。

まず本章で処理対象とする画像処理について述べる。

(1) 2値化

与えられたしきい値により、濃淡画像を2値画像に変換する。

(2) 画素間演算

2つの画像の対応する画素間で演算(ここでは差の絶対値を求める)を行う。

(3) 一様フィルタ

画像の各画素ごとに、周囲の3×3画素の平均を求め、その値を更新値とする。

(4) 任意フィルタ

画像の各画素に対し、3×3の任意の値とその周囲の3×3画素との積和を求め、その値を更新値とする。

(5) 幾何変換

各画素の変換座標が既知の画像の幾何変換を行う。但し相対変換距離は8以内とする。

(6) 濃度値変換

与えられた濃度値変換テーブルに従って、各画素の濃度値を変換する。

(7) ヒストグラム

画像のヒストグラムを求める。

(8) 相関法

与えられた画像と16×16画素から成る参照画像との相関を求める。

(9) 最尤法

5バンドから成る画像に最尤法を用いて、10カテゴリに分類する。

次に評価上の各種パラメータを設定する。

(1) 対象画像サイズ: $N \times N$ (N は2の累乗)

(2) CAPのセル数: $VP \times VP$ (VP は2の累乗)

(3) 対象画像の画素値レベル: M (より各画素は $\log_2 M$ ビットで表現される)

以上の仮定のもとに、上記の9種類の画像処理について、CAP上での入出力に必要な時間と演算に必要な時間を求めた。これを、

表1に示す。

ここで言う**演算時間**とは

- (1) Pセル内部での計算に要する時間
- (2) Sセルを経由したPセル間でのデータ搬送に要する時間
- (3) ブロードキャストに要する時間

以上、3つの時間を合計したものを指す。また入出力時間とは

- (1) 入力だけ、あるいは出力だけを行う時間
- (2) 入力と出力をオーバーラップさせて行う時間

以上、2つの時間を合計したものを指す。

次にCAPの論理シミュレーションに基づく画像処理の評価を図12に示す。但しここで扱った画像処理は、次の6項目についてである。

- ・ 2値化
- ・ 一樣フィルタ
- ・ 任意フィルタ
- ・ 画素間演算
- ・ 濃度値変換
- ・ ヒストグラム

また対象画像は、画素値が8ビットから成り、 2048×2048 の大きさである。CAPのセル数としては 128×128 、1クロックの同期を入と仮定した。

第4章でのブロック・パイプラインの入出力時間と演算時間との比率を見積もるため、個々の処理に関して、各々を別々に示している。また全体の処理時間は、ブロック・パイプラインを採用しているため、演算時間と入出力時間のうち、より大きい方が該当する。これは、例えば、演算時間の方が入出力時間に比べて、大きい場合には、ブロック・パイプラインの効果により、入出力時間が打ち消されることによる。逆も同様である。

表1は総称に各処理に必要な時間だけを見積もっているのに対し、図12では、実際に論理シミュレーションした結果であるため、オーバーフローチェックなどの余分な処理が含まれている。そのため、若干の値の変動はあるが、これにより表1の評価結果が検証される。

次にブロック・パイプラインの観点から考察を加える。この方式を採用したことにより、演算時間と入出力時間のより大きい方が全体の処理時間となる。もし演算時間の方が大きい場合には、常にCAPで演算が行われていることに

表1 CAPによる各種画像処理の性能評価

時間 画像処理	入出力	演算
二値化	$(N^2 \log M) / \sqrt{P} + \sqrt{P} \log M$	$\log M + (\log M + 1) N^2 / P + (N^2 \log M) / P$
画素間演算	$2(N^2 \log M) / \sqrt{P} + \sqrt{P} \log M$	$2(\log M + 1) N^2 / P + 3(N^2 \log M) / (2P)$
一樣フィルタ	$(N^2 \log M) / \sqrt{P} + \sqrt{P} \log M$	$4(\log M + 2) N^2 / P + (5 \log M + 4) N^2 / P$
任意フィルタ	$(N^2 \log M) / \sqrt{P} + \sqrt{P} \log M$	$36 + (45 \log M + 54) N^2 / P + 9(N^2 \log M) / P$
濃度値変換	$(N^2 \log M) / \sqrt{P} + \sqrt{P} \log M$	$M \log M + 2(\log M + 1) M N^2 / P + (N^2 \log M) / P$
幾何変換*	$\sqrt{P} \log M + \sqrt{P} \log M$	$2(\log N + 16) + 36 \log N + 19 \log M + 1154 + 401 \log M$
ヒストグラム	$(N^2 \log M) / \sqrt{P} + \sqrt{P} \log M$	$(M \log M) / \sqrt{P} + M((N^2 / P + 1)(3 \log M + 3) + 2 \log N + 3) + ((N^2 / \sqrt{P} + \sqrt{P}) \log M + 2 M \log N)$
相関法*	$\sqrt{P} \log M + 4N^2 / \sqrt{P}$	$256 \log M + 2 \log \sqrt{P} + 256 \log^2 M + 512 \log M + 2650 + 55(N^2 \log M) / P$
最尤法	$55(N^2 \log M) / \sqrt{P} + 4N^2 / \sqrt{P}$	$310 \log M + 10(40 \log^2 M + 95 \log M + 61) N^2 / P + 55(N^2 \log M) / P$

* 幾何変換、相関法の項には、すべて $N / (\sqrt{P} - 15)^2$ を乗ずる。

また「 \log 」は ガウス記号である。logの底は2とする。

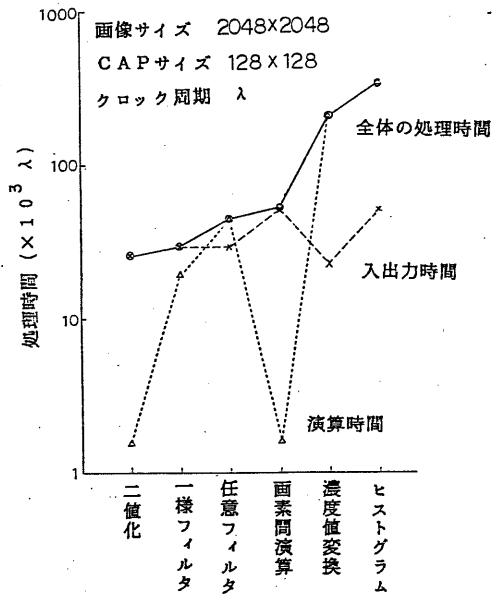


図12 CAPによる各種画像処理の実行時間

なり、ピーク性能に近い性能が得られる。すなわちセルラアレイ方式の欠点であるデータの出入力時間に時間が掛かるという点を解消し、高速演算が行える。

以上の検討結果より、既存の汎用計算機との性能比較を行った場合、CAPのクロック周期を100 ns、セル数を128x128と仮定すると、処理対象にもよるが約数百倍から千倍程度の高速性が得られることがわかった。特に、並列性の高い近傍演算では、その傾向が強い。

9. 考察

前章までで、セルラアレイ方式によるCAPのアーキテクチャを提案し、画像処理に対する検討、評価を行った。

CAPは従来のセルラアレイ方式の欠点とされていた(1)入出力問題 (2)SIMD型による柔軟性の欠如 (3)遠隔セル間のデータ転送の困難さを解消し、更にセルラアレイ方式の利点を踏襲している。そのため従来のパイプライン型の汎用計算機では効率よく実行できなかった各種の画像処理を高速に実行できる。更に衛星画像処理のように大規模な画像処理に対しても十分に対応できると考えられる。

CAPの並列性は次の2項目に集約される。

(1) 空間的並列処理

演算機能を持つセルを、多数、二次元格子状に配置し、それらが同時動作することにより得られる並列性。また対象となる画像データは、本来、二次元構成であるため、通常はこの二次元アクセスに多くの時間を必要とする。しかしCAPでは、対象画像を各セル内のメモリに格納し、各セルからのアクセスを可能にすることにより、容易に二次元アクセスが行える。これもまた空間的並列処理と言える。

(2) 時間的並列処理

本論文におけるブロック・パイプライン方式を採用することにより、CAPへのデータの入力、CAP内での演算、CAPからのデータの出力をすべて同時に行える様にした。

10. あとがき

本論文では記さなかったが、

- (1) CAPに適したアルゴリズムの開発
 - (2) CAPシステムをサポートする並列記述言語、ソフトウェアの開発 など
- を現在、進めている。今後は、これらに対する詳細な検討が重要になってくる。

なお、本研究は通産省工技院大型プロジェクト「科学技術用高速計算システム」の研究開発の一環として行ったものである。

参考文献

- [1] "衛星搭載リモートセンサデータの高速処理"
映像情報, vol. 13, no. 9, pp. 53-55, (1981).
- [2] Ono et al.: "Fast Digital Processing SEASAT
SAR Data", 15 International Symp. on Remote
Sensing of Environment, D-19.
- [3] 木戸出、坂上; "パイプライン方式と完全並列
処理方式が増えた最近の画像処理装置", 日経エ
レクトロニクス, 1982年7月19日号(1982).
- [4] 坂上、木戸出; "イメージプロセッサの最近の
動向", 信学会誌, vol. 67, no. 1, pp. 90-98(1984).
- [5] K. E. Batcher: "Design of A Massively Parallel
Processor", IEEE Trans. on Comput. C-29, no. 9,
Sept., (1980).
- [6] R. W. Hockney and C. R. Jesshope: "Parallel Com-
puters architecture, programming and algorithms",
pp. 178-192. Adam Hilger Ltd. Bristol (1981).
- [7] M. J. B. Duff: "Review of the CLIP image process-
ing system", NCC, pp. 1055-1060 (1978).
- [8] "2次元処理に向けたアレイ・プロセッサを開発"
日経エレクトロニクス, 1980年12月8日号,
(1980).
- [9] 菅, 宮田, 上田, 田中; "高速画像処理向き並列
処理アーキテクチャの検討", 情処第26回全国大会,
4P-11, pp. 261-262 (1983).