

解説

シストリック・アレイ†



梅 尾 博 司††

1. はじめに

最近の VLSI 技術の発展により、数万から数十万台のプロセッサからなる超並列計算機の実現も可能になりつつある。“蜂の巣六角アレイ” (IEEE Computer Magazine, Vol. 15, No. 1 の表紙) と “そしてよどみなき流れは、さらによどみなきリズムによって流れる” (And the smooth stream in smoother numbers flows: by Alexander Pope, 文献⁸⁾ (p. 299) の邦訳) の名文句で世界中に数多くの研究者をもつようになったシストリック・アレイ (Systolic Array, 以後 SA と略す) は, H. T. Kung and C. E. Leiserson^{8)~10)} の 2 人により提案された VLSI 向き並列計算モデルである。

シストリック (systolic) とは “心臓収縮の” を意味する形容詞である。SA は心臓 (IEEE Computer Magazine, Vol. 20, No. 7 の表紙) に見たてた単純なプロセッサを多数個規則的に結合してできる並列計算システムで、個々のプロセッサは、周辺のプロセッサへデータを送り出しかつ受け取る動作をくり返す。これは、心臓が収縮動作をくり返しながら、血液を全身に送りだしかつ汲み出す動作からのアナログである。

SA の特徴は、アレイへのデータの入出力を逐次的 (sequential) に、アレイ上での処理を並列 (parallel) に行い、しかもこれらの動作はパイプライン化され重畳的に進められるという点にある。最近のプログラマブル・シストリック・チップを用いた Warp マシンの試作・応用⁷⁰⁾、ノイマン型計算機との親和性の良さ、VLSI 化にともなう数々のメリットなどから、比較の実用性が高いパラレル・プロセッサの一つと考えられている。

Kung and Leiserson の 2 人が SA なる概念を提案して今年で 10 年を迎える。提案された当初は批判も

多かった (主に従来からのセルラ・オートマトンの研究者たちから) SA も、話題をシストリック・コンセプトに限定した国際会議が開かれたり¹⁹⁾、また数多くのシストリック・コンピュータが実際に稼働を始めるなど^{67)~71)}、並列処理システムにおける一つの研究分野を形成しつつある。

そこで本稿ではこれを機会に、シストリック・アレイ/アルゴリズム/アーキテクチャに関する基本的な事柄を整理するとともに、最近の話題として Turing 機械によるシストリック・アルゴリズムの設計方法を取り上げ、SA に関する近未来の研究動向を探る。

SA に関する文献は、1988 年 4 月現在約 450 編存在する。これらすべての文献を網羅することは事実上不可能である。また本学会誌でも SA に関連したすぐれた解説がいくつか書かれている^{16), 17), 35), 71)}。本稿ではこれらの解説内容とはできるだけ重複を避け、これまでにあまり扱われていないトピックスをとりあげる。まず 2. では SA 誕生時のシストリック・コンセプトの紹介、これまでに提案されているシストリック・アルゴリズムの一覧、シストリック辞書データ構造の紹介を行う。最後に SA を扱う基本定理を整理する。3. では、Ibarra, Palis and Kim らにより提案された逐次型 Turing 機械による SA の特徴付け定理を紹介する。これは SA に代表される並列計算モデルの諸側面、たとえば能力、性質、アルゴリズム設計などを、従来からよく知られている逐次型の Turing 機械上で考える手段を与えるもので、SA に関する数多くの研究のなかでも重要な進歩と思われる。

2. シストリック・アレイ

SA 誕生後、今年で 10 年を迎える。この間に多数のシストリック・アルゴリズムが考案され、それとともに数多くの SA モデルが提案されている。それでもなお (なおさら?) “シストリック・アレイの正確な定義は…?” とは、この分野の研究者のみならず VLSI 計算モデルに興味をもつ人たちがしばしば口にする質

† Systolic Arrays—A Survey by Hiroshi UMEMO (Dept. of Applied Electric Engineering, Faculty of Engineering, Osaka Electro-Communication University).

†† 大阪電気通信大学工学部応用電子工学科

間である。

本章では SA 誕生時にさかのぼり、Kung and Leiserson^{8)~10)} に従って、まずシストリック・コンセプトを紹介し (2.1)、次にこれまでに提案されているシストリック・アルゴリズムを整理する (2.2)、木マシン上におけるシストリック辞書データ構造についても言及する。最後に、SA を取り扱う基本定理をまとめる (2.3)。ここには、SA に関する最近の結果だけでなく、**Iterative array** (IA と略す)、**セルラオートマトン (Cellular Automaton, CA と略す)** など従来の並列計算モデルに関するもので SA 向きに再構成したものを含む。

2.1 シストリック・アレイ (SA) とは

従来のノイマン型計算機では、高速フーリエ変換 (FFT)、行列計算といった科学技術計算、画像処理、各種記号処理などのためのソフトウェアがサブルーチン・パッケージとして用意されていた。Kung and Leiserson^{8)~10)} らは、SA を次のような形態で使用されたと考えた。すなわち、上記のサブルーチン・パッケージにかわる特定用途向けのハードウェア付加装置としてである。図-1 にその利用形態を示す。

各 SA は図-1 に示すように、通常ノイマン型ホスト計算機のバスに直接接続され、その周辺機器の一つとして使用される。SA の内部は数千から数万個の簡単なプロセッシング・エレメントを規則正しく接続したものからなる。したがってホスト計算機とのデータの授受はバスを介して逐次的 (直列ともいう、**sequential**) に、アレイ上でのデータの処理は並列 (**parallel**) に行われる。このように、SA はノイマン型コンピュータとの親和性が非常に良く、概念の明解さ、スループットの良好性、高性能マシンへの期待、

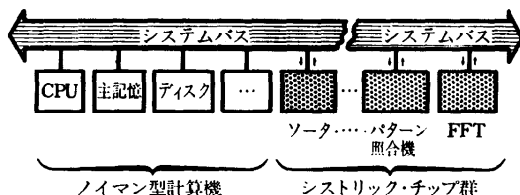


図-1 シストリック・アレイの利用形態^{(8)~(10)}

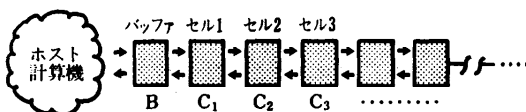


図-2 シストリック・アレイ⁽¹¹⁾

おりからの VLSI 技術の発展とともに世界中に急速に研究者数を増やしていった。

SA は従来から研究されてきた IA のサブクラスの一つと考えられる。多数の同一素子の規則正しい配列からなる並列計算モデルは、ノイマンの自己増殖オートマトン^{(4), (5)}に始まる。その後、CA, IA などの呼び名で数多くの研究^{(1)~(7)} がなされているが、SA の特徴である直列入出力動作を考慮に入れた並列計算モデルの研究は、Hennie⁽¹⁾ の IA に始まり、Atrubin⁽²⁾ の実時間 2 進数かけ算機、Fischer⁽³⁾ の素数生成、Cole⁽⁶⁾ の言語認識などがあげられる。

SA には多数のバリエーションが存在する。本稿では理解ならびに説明の容易さから、もっとも基本的なモデルと考えられる 1 次元 SA を取り上げ、シストリック・コンセプトを説明する。図-2 はアレイの左端で入出力動作を行う 1 次元 SA である。

SA は次の特徴をもつ。

2.1.1 局所結合ならびに局所通信

CA との関連から各プロセッサをセルとも呼ぶ。i 番目のセルを C_i と示す。各セル C_i ($i \geq 1$) は隣り同士のセル C_{i-1} , C_{i+1} とのみ接続される (局所結合)。 C_i の左右に隣接するセル C_{i-1} 及び C_{i+1} を C_i のノイマン近傍 (1 次元) と呼ぶ。各セルが 1 ステップ (単位時間) で直接通信できるのは、近傍内のセルだけである (局所通信)。したがって、お互いに離れたセル C_i , C_j 間でデータの交換/授受を行うには、そのセル間の距離に等しい $|i-j|$ ステップを要す。このようにアレイ上でのデータの送受信はすべて局所結合を通じて局所的に行われる。

他にすべてのセルに共通する線として、電源 (power line)、接地 (ground line)、ならびにクロック (clock line) がある。これらは局所的な信号線でない (global signal line) ことに注意されたい。

2.1.2 均一構造

すべてのセルは単純で同一構造をもつ。各セルは有限個のレジスタをもつ。IA 及び CA では各セルは有限オートマトンと仮定される。SA 上では一般に上記のような仮定はなく、たとえば実数の四則演算なども 1 ステップで実行可能である。

2.1.3 静止状態

$t=0$ のとき、すべてのセル上のレジスタは空にセットされている。すべてのレジスタが空であるセルは静止状態にあるという。静止状態にあるセル C_i は、隣接するセル C_{i-1} 及び C_{i+1} がともに静止状態である

かぎりこの状態を保つ。

2.1.4 入出力は直列に

$t=0$ 時にはどのセルもあらかじめデータをもっていない (プレロードなしと呼ぶ)。処理に必要なデータはホスト計算機 (たとえばディスク) 上に貯えられている。 $t=0$ 時よりアレイへの入力開始される。入力はホスト計算機からあらかじめ定められた一定の割合で、たとえば、1ステップにつき1個のデータ、2ステップに1個のデータ、1ステップで2個のデータ、……というぐあいに与えられる。この割合をパイプライン・インタバルと呼ぶ。特に入力のそれは、入力パイプライン・インタバルと呼ぶ。上記の入力パイプライン・インタバルは、それぞれ 1, 2, $1/2$ であるという。 $a_1, a_2, \dots, a_n, \$$ を長さ n の入力とする。 $\$$ はアレイに入力の終わりを知らせる記号である。入力は $a_1, a_2, \dots, a_n, \$$ の順に先頭から入力パイプライン・インタバルに従って与えられる。このような入力方式を直列入力と呼ぶ。

出力 $b_1, b_2, \dots, b_k, \$$ も入力と同様に直列的にアレイから出力される。1個の出力 b_i ($1 \leq i \leq k$) を得るのに要する時間を出力パイプライン・インタバルと呼ぶ。

直列入出力 SA では、ホスト計算機 (外界) との単位時間あたりの通信量は $O(1)$ ビットである。Kung and Leiserson 以降、直列入出力 SA だけでなく、アレイ上の周辺プロセッサあるいは全プロセッサ上で入出力を行う SA も研究されている。これらのアレイ上では単位時間あたり $O(n)$ あるいは $O(n^2)$ ワードの通信が可能であり、並列入出力動作と呼ばれている。ここで n はアレイあるいは入出力データのサイズを意味する。 n が数千から数万であることを考えあわせると、並列入出力は実現が困難なように思われる。

2.1.5 同期モデル

各セルは同期して動作する。SA 上での意味のある計算は非静止状態のセル上でなされる。非静止状態のセル集合は、図-3 に示すように動作開始と同時に徐々に右方向に広がってゆく。そのスピードは 1 である。 $t=n$ 時には、高々 n 個のセルが非静止状態となっている。これは上記の (1) 及び直列入力の仮定から明らかである。したがって、SA 上での有効な計算とは図-3

に示すような状態を並べたものであり、静止状態のセルを考慮に入れる必要はない。ここで、 S_i 及び B_i' はそれぞれ、セル C_i 及びバッファ B の時刻 t におけるレジスタの内容を意味する。

2.1.6 計算量

最初の入力 a_1 を与えてから最後の出力 b_k が得られるまでのステップ数を SA の時間計算量と呼び $T(n)$ で表す。ここで n は入力サイズを意味する。一般に $T(n) \geq n$ 、 $T(n)$ には入力、処理、ならびに出力に要するすべての時間が含まれていることに注意されたい。最初の入力を与えてから最初の出力 b_1 が得られるまでの時間を応答時間と呼び、 $R(n)$ で表す。

次々とサイズの異なるデータ・セットを処理する

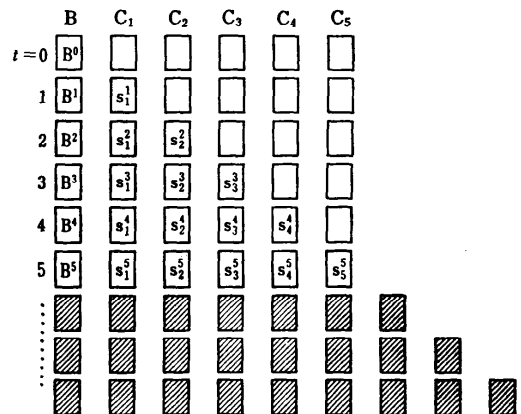


図-3 シストリック・アレイ上の有効な計算状況

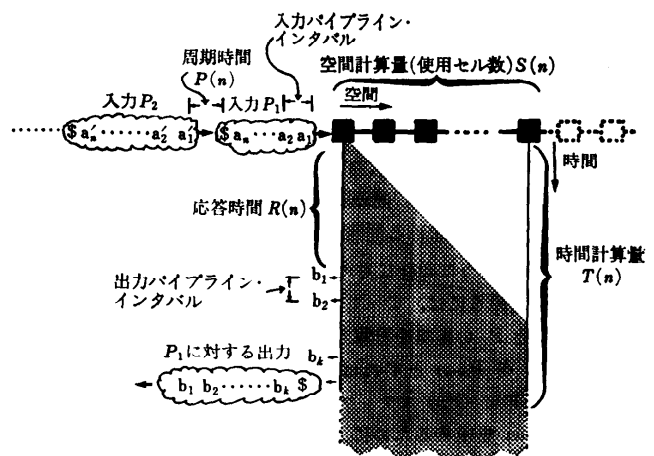


図-4 1次元 SA 上での時間-空間図式

ケースがある。このとき、一つのデータ・セット（サイズを n とする）を入力後、次のデータ・セットを入力するまでにホスト計算機が待機すべき時間を周期時間と呼び、 $P(n)$ で表す。 $T(n)$, $R(n)$ ならびに $P(n)$ は、解くべき問題、SA のアーキテクチャ/アルゴリズムに依存する。 $T(n)$ は線形時間、すなわち、 $T(n) = cn$ (c は正整数), $R(n)$, $P(n)$ は $O(1)$ が望ましい。特に $P(n)$ が $O(1)$ である場合、高いスループットが得られる。1次元 SA 上のアルゴリズムは、図-4 に示す時間-空間図式を使うと分かりやすい。ここで水平方向はセル空間を、垂直方向は時間軸を意味する。使用セル数を空間計算量と呼び、 $S(n)$ で表す。一般に $S(n) = n$ である。図-4 より、パイプライン・インタバル、 $S(n)$, $T(n)$, $P(n)$, $R(n)$ ならびにセル上の同期領域などの諸概念を容易に理解できよう。

2.2 シストリック・アルゴリズム

これまでに提案されているシストリック・アルゴリズムの一覧を表-1 に示す。これらのアルゴリズムは、1次元アレイ、2次元アレイ、木結合アレイ上で、また直列あるいは並列入出力動作をするアレイなど、いろいろなバリエーションをもつ SA 上で開発されている。実用的な観点からは、入出力ポートのサイズが $O(1)$ である直列入出力動作をする1次元アレイ及び木結合アレイが重要である。

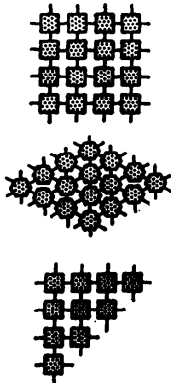
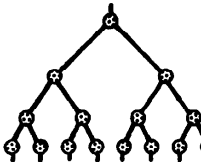
ここではまず1次元 SA 上に設計されたシストリック計算幾何アルゴリズムを次に木結合 SA 上に実現されたシストリック辞書データ構造を紹介する。

2.2.1 1次元 SA アルゴリズム

SA は、通常ノイマン型計算機上で $O(n^2)$ 時間を要する問題を $O(n)$ 時間で処理できる。次に述べる最近点対問題 (all neighbour pairs) を解くアルゴリズム⁷⁶⁾ は、1次元シストリック・アルゴリズムの典型的なものと考えられる。最近点対問題とは、 n 個の点 $P_1, P_2, \dots, P_n$ が与えられたとき、すべての $P_i (1 \leq i \leq n)$ について、自身を除く $n-1$ 個の点のなかで P_i にもっとも近い点 $P_{N(i)}$ を求める問題である。

SA 上では次のような素朴な考え方をする。 P_i の最近点 $P_{N(i)}$ を求めるには、すべての $j (1 \leq j \leq n, j \neq i)$ に対して、 P_i と P_j の距離を計算しそのなかで最小距離を与える P_j を $P_{N(i)}$ とすればよい。したがって $P_{N(i)}$ を求めるのに $O(n)$ ステップを要す。すべての i について $P_{N(i)}$ を求めるために上記の計算をパイプライン化する。パイプライン・インタバルは1ステップで良い。したがって全体の計算も $O(n)$ 時間で済む。

表-1 シストリック・アルゴリズム

シストリック・アレイ	シストリック・アルゴリズム
 1次元 SA	- 信号処理: FIR, IIR フィルタ, コンボリューション計算 - パターンマッチング, 最大共通部分系列計算 - 多項式の乗除算, GCD - 多項式の評価 - ソーティング - データ構造: スタック, キュー, プライオリティキュー - リードソロモン符号の復号 - 計算幾何学問題 - グラフ問題 - 言語認識問題 - 行列ベクトル積計算 - 離散的フーリエ変換
 2次元 SA	- 行列計算 - 帯行列計算 - LU分解 - 線形方程式 - QR分解 - 三角化 - 逆行列 - 固有値計算 - 言語認識 - 分脈自由言語の認識 - 構文解析 - 計算幾何学問題 - 画像処理 - 動的計画法 - 関係データベース演算 - フーリエ変換 - 信号処理 - グラフ問題 - 推移的閉包 - 最短距離計算
 木結合 SA	- 辞書データ構造 - ソーティング - 探索問題 - 多項式, 再帰式の評価 - 言語認識 - 正規言語認識

まず n 個のセル $C_i (1 \leq i \leq n)$ を用意し、 C_i にはあらかじめ P_i が入力されているとする。次に左端から $P_1, P_2, \dots, P_n$ の順に1個/1ステップの割合で、すなわち、スピード1でポイント・データを流す。 P_n の次にエンドマークをつけておく。

各 C_i は P_i と流れてくるポイント・データとの距離を1ステップで計算する。常に P_i との距離（これまでに流れてきた点との距離）が最小になるように、その距離とその距離を与える点データを更新する。このために $O(1)$ 個の有限レジスタを備えている。各 C_i は

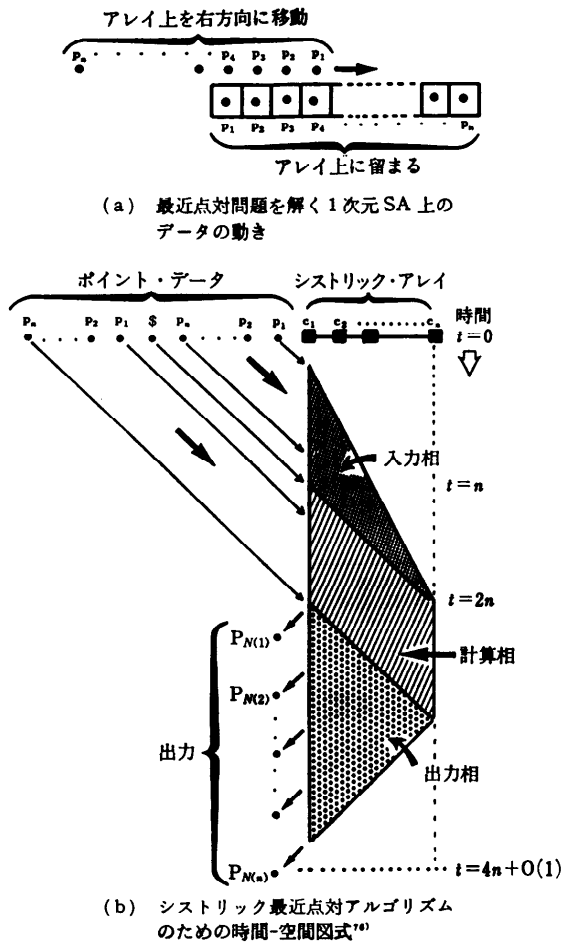


図-5

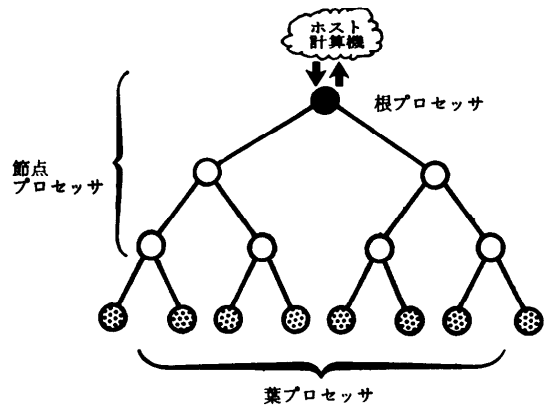
エンドマーク\$を読んだとき、各セル内に最近点 $P_{N(i)}$ を保持している。計算のようすを図-5(a)に示す。

実際には、ポイント・データのコピーを二つ用意し、図-5(b)に示す時間-空間図式で計算させる。これによると $4n+O(1)$ ステップ後には、すべての計算結果がホスト計算機に回収される。

上記アルゴリズムにおいて入出力パイプライン・インタバルはそれぞれ1及び2、各計算量は次の式であらわされる。すなわち、 $S(n)=n$, $T(n)=4n+O(1)$, $P(n)=O(1)$, $R(n)=2n$ 。読者は入力、計算、出力の各相がアレイ上でオーバーラップ化されているようすを理解できよう。

2.2.2 木結合 SA アルゴリズム

木結合 SA アルゴリズムとして、完全2進木結合のSA上で実現されるシストリック辞書データ構造を取

図-6 木結合シストリック・アレイ^(a)

り上げる。木結合SAでは、根プロセッサがホスト計算機とのI/Oプロセッサとして働く(図-6参照)。操作されるデータは (k, r) と対で表される。ここで k はキーを、 r はレコードを意味する。

ホスト計算機は次の6種類の問合せからなる命令列をSAに与える。

- Insert (k, r) ,
- Delete (k) ,
- Member (k) ,
- Extract-Min,
- Extract-Max,
- Near (k) .

である。各命令は次の機能をもつ。Insert (k, r) は辞書にデータ (k, r) を追加するものであり、Delete (k) は (k, r) なるデータを辞書から取り除き、Extract-MinならびにExtract-Maxは、それぞれ現在辞書に貯えられている最小あるいは最大の要素を取り除きそれらを出力する。Member (k) は、キーとして k をもつデータがあればそれを出力し、なければ“なし”を出力する。Near (k) は、 $|k-k'|$ が最小になる (k', r) を出力する。

表-2はこれまでに提案されているシストリック辞書データ構造の性能比較である。ここで、 N は全プロセッサ数、 n は辞書内のデータ数を意味する。辞書内にすでに存在するデータと同一データを冗長に挿入したり、存在しないデータを削除したりするときがある。このような命令を冗長挿入命令、冗長削除命令と呼ぶ。X-treeとは、完全2進木に同一レベル上のプロセッサを接続する結合を付加したモデルである。プロセッサがデータを保持しているか否かにより、活動/不活動プロセッサの二つに分けられる。読者は、

表-2 シストリック・ディクショナリ・マシンの性能比較

	Bently and Kung ⁽⁴¹⁾ (1979)	Leiserson ⁽⁴²⁾ (1979)	Leiserson ⁽⁴³⁾ (1980)	Ottmann, Rosenberg and Stockmeyer ⁽⁴⁴⁾ (1982)	Atallah and Kosaraju ⁽⁴⁵⁾ (1985)	Somani and Agarwal ^{(46),(47)} (1985)
命 令 セ ャ ッ ト	● Insert ● Delete ● Member	● Insert ● Delete ● Member ● X-Min		● Insert ● Delete ● Member ● X-Min ● Update	● Insert ● Delete ● Member ● X-Min ● X-Max ● Near	
冗 長 性	No			Yes		
アーキテクチャ	完全2進木	完全2進木 +葉プロセッサ間の鎖状結合	完全2進木	X-Tree	完全2進木	完全2進木
応 答 時 間	$O(\log N)$			$O(\log n)$ (最適時間)		
パイプライン・インタバル	Delete のみ $O(\log N)$ 他は $O(1)$		$O(1)$			
データの保持場所	葉プロセッサのみにデータを保持する			すべての節点プロセッサにデータを保持する		
Sorted Order の必要性	無	有				無
活動／不活動プロセッサの境界位置	葉プロセッサ上に固定			動的に移動 境界線は連続した2層にわたる		動的に移動 境界線は1層のみ

表-2 よりシストリック・辞書データ構造の性能向上を理解できよう。これらの命令の具体的な実現はここでは省略する。文献^{(40)~(46)}を参照されたい。

2.3 SA に関する基本性質

本節では SA の能力に関する基本性質を整理する。各定理の理解を容易にするため数学的な記述を避け、できるだけ図式的なものとする。

[定理1] パイプライン・インタバル縮小定理⁽⁶⁾

任意の自然数 k に対し、図-7(a) に示す SA, M 上で k パイプライン・インタバルを1に縮小できる。ただし、 M の応答時間は $R(n)=O(1)$ とする。

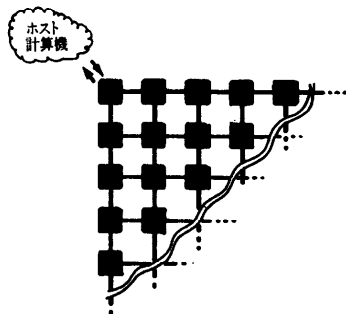
パイプライン・インタバルが1の最適時間シストリック・アルゴリズムの設計は比較的むずかしい。事実 Cole⁽⁶⁾ は実時間 (パイプライン・インタバルが1) 回文認識アルゴリズムを与えたが、そのアルゴリズムの正当性の理解は困難である。パイプライン・インタバルが2以上のアルゴリズムの設計は1のケースよりも容易であるので、設計後変換するという手段が考えられる。本定理は2次元 (図-7(b) 参照)、3次元、一般に n 次元 SA に対しても有効である。

[定理2] 加速定理⁽⁶⁰⁾

時間計算量 $T(n)$ の任意の SA (パイプライン・インタバルが1の図-7(a)のタイプ)、 M に対し、 M を $n + \lceil (T(n)-n)/k \rceil$ 時間で模倣する別の SA (同一アーキテクチャの) が存在する。ただし、 k は任意の自然



(a) 1次元 SA, 左端での直列入出力, パイプライン・インタバル: k , プレロードなし, 2方向の情報伝達



(b) 2次元 SA, 左上隅での直列入出力, パイプライン・インタバル: k , プレロードなし, 4方向の情報伝達

図-7 パイプライン・インタバルが縮小可能なシストリック・アレイ

数である。

線形時間シストリック・アルゴリズムのクラスは実用上重要である。定理2を使えば、線形時間で動作する SA を実時間に限りなく近い $(1+\epsilon) \cdot n$ 時間で動作するようにできる。ただし ϵ は任意の実数である。し

かしながら、 $(1+\varepsilon) \cdot n$ 時間は実時間ではないことに注意されたい。

この類の加速定理は従来からセルラオートマトン上で知られていた。Smith⁴⁷⁾ は、 $T(n)$ から $n + T(n)/k$ という少し性能の悪い加速定理を得ている。Ibarra⁶⁰⁾ らは、逐次型 Turing 機械による各種の IA の特徴付けを試み、Turing 機械上で知られていた加速定理をうまく適用することにより上記定理を得た。本定理は $T(n)$ がたとえば $T(n) = n + \log_2 n$ と書けるような動作時間が実時間に近いシストリック・アレイ上で有効である。

パイプライン・インタバルが2以上のアレイに対しては定理1を、パイプライン・インタバルが1のアレイに対しては定理2を適用することにより時間計算量の性能向上をみることが出来る。しかしながらこれらの定理は、“各セルの内部状態数が有限であるかぎり、いくら大きくても良い”という条件下での議論である。実際の実現にあたっては、その内部状態数が大きくセルが複雑になる可能性があるので注意が必要である。

【定理3】 ブロードキャスト/グローバル・コントロール除去定理^{54), 60)}

時間計算量の損失なしに、SA からブロードキャスト/グローバル・コントロール・バスを除去できる。

局所的通信網は SA の重要な特徴である。ブロードキャスト (BC)、グローバル・コントロール (GC) など、大局的通信網を SA に付加したモデルは、準 SA (semi SA) と呼ばれ、本来の純 SA (pure SA) とは区別されてきた¹⁴⁾。準 SA に関する研究は、Seiferas⁴⁹⁾ にさかのぼることができ、その能力についてこれまでに数多くの研究がなされている^{24), 29), 34), 49)}。

BC はホスト計算機から1ステップですべてのプロセッサに1種類の情報を伝達できる。GC 上では、情報は各セルを経由して伝達される。ただし遅延時間はゼロである。セルを通過するたびに情報は変更される可能性をもち、BC の拡張タイプと考えることができる。どちらも遠くはなれたプロセッサにいっせいにホスト計算機から情報を伝達できるので、BC/GC を備えたアレイはそれのないものよりも高速処理が可能であると考えられてきた。しかしながら言語認識問題に限定すれば、図-8 に示す二つのタイプの SA はまったく等価である。正確には、任意の $T(n)$ ステップで動作する図-8(a) の準 SA、 M に対し、 M を $T(n)$ ステップで模倣する図-8(b) の純 SA が存在する。

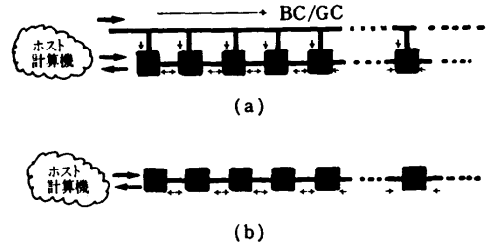


図-8 BC/GC を備えた SA (a) と通常の SA (b)

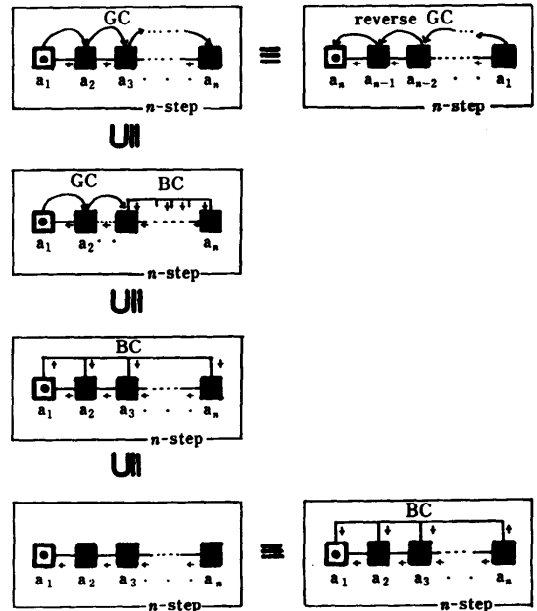


図-9 1方向実時間プレロード型 SA の能力関係

Ibarra らは上記定理を Turing 機械による特徴付けを用いて示した。Umeo⁵⁴⁾ らは、タイプ(a)の SA を(b)で直接模倣する技法を開発し、同様な定理が2次元、3次元、一般に n 次元アレイ上にも、さらに木、ピラミッド・マシン上でも成立することを示した⁵⁴⁾。タイプ(a)から(b)への変換によるオーバーヘッドは比較的小さい。BC 及び GC の利点は、シストリック・アルゴリズムの設計が非常に容易になることである。Ibarra⁶⁰⁾ らは、他に図-9 に示すような種々の BC/GC を備えた一方向アレイの能力的等価性、階層性も示している。すべてのモデルに対して、 $T(n) = n$ (実時間)、入力は一方向を仮定、●印は出力(受理)セルを意味する。詳細は文献⁶⁰⁾ を参照されたい。

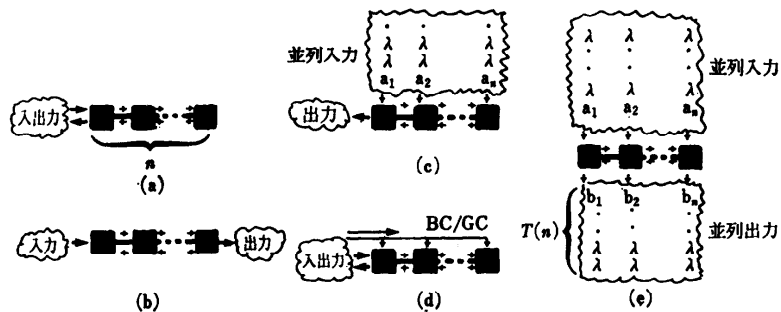


図-10 お互いに他を $c \cdot T(n) + O(n)$ ステップで模倣可能な1次元 SA のバリエーション
ただし、 c は定数、 $T(n)$ は模倣される側の時間計算量

[定理4] 線形時間等価性定理

図-10 に示した SA はお互いに他を定数時間係数で模倣できる。

図-10 は使用セル数を n に限定した1次元 SA である。同図(b)は左端で直列入力、右端で直列出力する SA、(c)は $t=0$ 時のみ並列入力(プレロードに相当)、左端直列出力をする SA、(d)は BC/GC を備えた SA、(e)は $t=0$ 時及び $T(n)$ 時のみ、各1回ずつ並列入出力動作をする SA である。ここで $T(n)$ は時間計算量を意味し、一般に $T(n) \geq n$ である。これまでシストリック・アルゴリズムは、これら一連のバリエーション上で開発されており、時間計算量は線型時間である。大部分の SA は時間計算量 $T(n)$ に対し、 $c \cdot T(n) + O(n)$ の定数係数と線型時間の和程度のオーバヘッドでお互いに他を模倣可能である。しかも良いことには、定数係数 c は、2 とか 3 などの比較的小さい整数である。したがって線形時間計算量の範囲内では、どのモデル上でアルゴリズム開発を行っても大きな違いはない。

図-10(e)はこれまでに数多くの研究がなされている CA である。計算開始前にすべてのデータが各セルに並列入力されており、出力は計算終了後 ($T(n)$ ステップ後) 並列出力される。Umeo^{36), 37)} は、この CA モデルを最適時間で模倣する(a)タイプの SA を提案している。

SA に関しては、他に次の性質が明らかになっている。

[定理5] 言語受理能力⁴⁷⁾

1次元 SA の受理する言語クラスは決定性文脈依存言語のクラスと一致する。

[定理6] 能力的階層性⁴⁸⁾

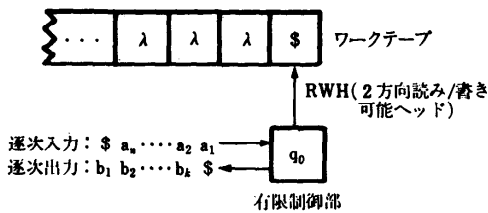
実時間 SA においては、任意の自然数 k に対し、

($k+1$) 次元 SA は k 次元 SA より真に能力が高い。

3. Turing 順序変換機によるシストリック・アレイの特徴付け

最適時間シストリック・アルゴリズムの設計は、時として非常に難しい場合がある。なぜなら設計者はいつも多数のプロセッサ間の同期とそれらの上でなされる仕事の並列性を考慮しなければならず、ノイマン的思考に慣れたわれわれの頭脳には多量の並列性はとっつきにくいからである。また同じ理由で、いったん設計したアルゴリズムの正当性証明ならびに解析も一般に難しいものとみなされている。

シストリック・アルゴリズムの設計を簡素化したり、統一的かつ形式的に取り扱う手法はこれまでに数多く提案されているが^{21)~39)}、それらは使用するのが難しかったり、特定の問題にのみ有効であるケースが多かった。このような状況のなかで最近 Ibarra, Palis and Kim ら^{56)~66)}により得られた逐次型 Turing 変換機を用いるシストリック・アルゴリズムの特徴付け定理は、逐次型及び並列型計算モデルという二つのまったく異なった概念を結びつけるものである。われわれはこれを使ってシストリック・アルゴリズムの設計を逐次的な考え方で行うことができる。これまでに集積した Turing 機械に関する膨大な知識を利用でき、ノイマン的思考に慣れ親しんでいるわれわれには、非常に好都合と考えられる。Ibarra らは、他にセルラ・オートマトン、トレリス・オートマトンなどの並列計算システムについても同様な特徴付けを試みたが^{56)~58), 60), 63)}、本稿では読者の興味の多さから判断して、図-2 に示すタイプのシストリック・アレイに議論を限定する。本章では彼らのアイデアを分かりやすく説明し、その有効性を示す。

図-11 Turing 順序変換機概念図⁽⁴⁾

3.1 Turing 順序変換機

Ibarra⁵⁵⁾⁻⁶⁶⁾が導入した Turing 機械Mは、従来からよく知られている順序変換機(逐次型)のサブクラスと考えることができる。Mは有限制御部と半無限テープならびに1個の2方向読み書き可能ヘッド(Read Write Head, RWHと略す)から構成される。有限制御部は入力端子と出力端子を1個ずつ備えている(図-11参照)。

動作開始前、テープ上の右端を除くすべてのマス目にはλ記号(ブランク記号)が記入されている。最初 RWH はテープの\$記号上にあり、有限制御部は初期状態 q_0 にある。次の一連の動作(i)~(iii)をMの完全走査と呼ぶ。

(i) 有限制御部が q_0 の状態で RWH が\$を読んでいる状態から動作を始め、テープ上の非λ記号を書き換えながら、最初にλ記号を読むまで左方向に進む。この間、有限制御部は真の入力記号を読まないし出力もしない(この動作をε動作と呼ぶ)。

(ii) RWH がλ記号を読んだとき、有限制御部は1個真の入力記号を読み、RWH が読んでいるλを非λ記号に書き換え、さらに1個の出力記号を出力する。

(iii) その後有限制御部は q_0 状態になり、RWH が\$を読むまで右方向に進む。その間テープ記号をまったく変更しないし、有限制御部の入出力もε動作である。

完全走査は RWH による記憶テープの右端から左方向への1回の走査、テープの左端での入出力動作、ならびに左端から右方向への1回の走査からなる。

任意の長さ n の入力 $a_1 a_2 \dots a_n \$$ に対し、有限制御部がエンド・マーク\$を出力するまで高々 $T(n)$ 回の完全走査をするとき、Mの走査時間計算量は $T(n)$ であるという。

次の定理が Ibarra⁵⁵⁾⁻⁶⁶⁾の主要結果である。

[定理7] SA の特徴付け定理⁵⁵⁾⁻⁶⁶⁾

時間計算量 $T(n)$ の SA と走査時間計算量 $T(n)$ の Turing 順序変換機とは等価である。

3.2 実時間言語認識アルゴリズム

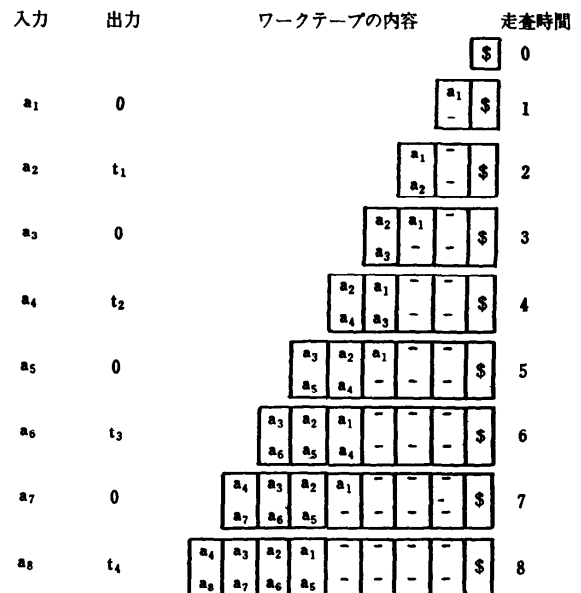
それでは定理7の応用例として言語認識アルゴリズムを示そう。時間計算量 $T(n)$ のシストリック・アルゴリズムの設計が困難である場合、かわりに走査時間計算量 $T(n)$ の Turing 順序変換機を構成すればよい。

次の言語 L の認識問題を考える。

$$L = \{x \cdot x \mid x \in \Sigma^+\}$$

ここで Σ は有限アルファベット集合 ($|\Sigma| \geq 2$) である。 L は前半分と後半分がまったく等しい記号列からなる。入力 $a_i (i=1, 2, \dots)$ は1個/1ステップの割合で与えられる。これまでに読み込んだ入力 $a_1 a_2 \dots a_{2m}$ は、すべての $i, 1 \leq i \leq m$, について $a_i = a_{i+m}$ を満足するときに限り L に属す。 $a_1 a_2 \dots a_{2m}$ が L に属するかどうかの判定を時刻 $t=2m+1$ (あるいは $t=2m+O(1)$) 時に下せるならば、 L を実時間で認識したという。Coleによる L の実時間認識アルゴリズムの構成は非常に巧みで、アルゴリズムの設計方針は Cole⁶⁾ から伺い知れない。

事実 Leiserson and Saxe²⁴⁾ においても、そのアルゴリズムの理解の困難さが指摘されている。しかし、

図-12 $L = \{xx \mid x \in \Sigma^+\}$ を $T(n)=n$ の走査時間計算量で認識する Turing 変換機のワーク・テープ系列⁽⁴⁾

L の認識を前述の Turing 順序変換機で行うことを考えると事情は一変する。定理 7 によれば、長さ n の入力 x に対し n 回の完全走査で x を認識する Turing 順序変換機を構成すれば良いわけである。図-12 は $a_1 a_2 \dots a_8$ を 8 回の完全走査で認識する Turing 順序変換機 M のワーク・テープ系列である。 M は偶数時刻における左方向への走査において、テープ上の二つのトラックのます目（実際にはテープの左半分）に記入された記号が一致するか否かを逐次的に検査する。たとえば $t=6$ の出力 t_3 は

$$t_3 \leftarrow (a_1 = a_4) \text{ and } (a_2 = a_5) \text{ and} \\ (a_3 = a_6)$$

を計算すれば良い。図-11 に示した Turing 順序変換機による認識は、 L の定義にもとづいて入力ストリングを逐次的に検査するものであり、アルゴリズムの設計方針は明確である。図-12 に示したワーク・テープ系列を生成する Turing 順序変換機の構成は困難ではないが煩雑である。ここでは詳細を省略する。

3.3 Ibarra⁽⁵⁵⁾⁻⁽⁶⁰⁾ らによる特徴付け定理の有効性

Turing 順序変換機による SA の特徴付け定理は、並列計算につきものの多量の並列性と同期問題を逐次的な考え方で取り扱えるという新しい見方をわれわれに提供した。ここでは定理 7 を応用することによって新たに得られた、あるいは改良された結果などを整理することにより、彼らの定理の有効性を示そう。

3.3.1 他の並列計算モデルに対する特徴付け

Ibarra らは、Turing 順序変換機による特徴付けが SA のみならず他の並列計算モデルに対しても可能なことを明らかにした。文献⁽⁵⁶⁾ では、シストリック 2 進木オートマトン (SBTA) の特徴付けを行い、SBTA により認識できない言語族の存在を証明する手法を確立している。特徴付けで用いられた Turing 機械は、付属装置として 2 進カウンタを備えている。これは SBTA の木構造に由来する。文献^{(57), (58)} では、トレリス・オートマトン (TA) に対しても同様な特徴付けを行った。TA は Culik, Gruska and Salomaa⁽⁵²⁾ により導入された並列計算モデルで、CA の一種と考えることができる。

CA に関する研究は、Neuman⁽¹⁾, Hennie⁽¹⁾, Codd⁽²⁾ らにより古くからなされている。CA と SA との違いは、データの入出力とその計算の進め方にある。前者は上記三つの動作を時間的に分離して実行するのに対し、後者はパイプライン的にオーバーラップさせて行う。後者のほうがより現実的な計算モデルとみなせよ

う。セルラ・アルゴリズムは歴史が古いだけに多数のアルゴリズムが累積されている。SA 上でセルラ・アルゴリズムを使用するために、Umeo^{(36), (37)} は最適時間で CA を模倣する方法を与えている。

CA, SA に代表される IA 族は、セル間の通信方向、入出力方式、使用セル数、入力受理方式、時間計算量などのパラメータにより多数のバリエーションに分類される。

Ibarra⁽⁵⁵⁾⁻⁽⁶⁶⁾ らは、上記の諸性質の組合せとして得られる各種の IA が、前節で示した Turing 変換機あるいはその変形タイプにより特徴付けが可能であることを示した。文献⁽⁶⁴⁾ は 1 方向 IA のこれまでに知られていない能力を明らかにした点で興味をもたれる。文献⁽⁵⁹⁾ では、CA の言語受理モードとその能力について考察している。各タイプの IA とそれに対応する Turing 変換機に関する詳細は文献⁽⁶⁰⁾ を参考にされたい。

3.3.2 シストリック・アルゴリズムの設計、ならびに正当性の証明が容易になる

最適時間で動作するシストリック・アルゴリズムの設計は難しい場合が多い。またアルゴリズムの正当性を証明することも一般に困難である。しかし Turing 機械を用いて考えれば、多数のセル間の並列性を考慮に入れる必要はなく、設計ならびに証明が非常に容易になる。Turing 機械を使用した最適時間（あるいは最適に近い）シストリック・アルゴリズムの設計例が文献^{(55), (61)} に与えられている。ここでは次の二つの例をあげる。

[例 1] 実時間 2 進かけ算機

2 進表現された 2 数（それぞれ、 n bit）のかけ算を $2n+O(1)$ ステップで実現する。このアルゴリズムは Atrubin⁽²⁾ により設計されたものであるが、従来からトリッキンな IA アルゴリズムの一つとして知られている。かけ算はコンボリューション計算の一種とみなされ、梅尾⁽³⁰⁾ は最適時間シストリック・コンボルバの設計及びその正当性の証明を試みているが、Turing 機械によるものよりは複雑である。

[例 2] 言語認識アルゴリズム

次の言語を考える。

$$L_1 = \{xx^R \mid x \in \Sigma^+\} \text{ (回文集合)}$$

$$L_2 = \{xx \mid x \in \Sigma^+\}$$

$$L_3 = \{x \mid x \text{ は非決定性カウンタ・マシンにより受理される}\}$$

$$L_4: \text{線形文脈自由言語}$$

L_5 : 文脈自由言語

$L_1 \sim L_5$ を最適時間あるいはそれに近い時間で認識する SA を直接構成することは単純な仕事ではないが^{61), 48)}, Turing 機械を用いれば比較的簡単であり^{55), 60), 66)}, 何よりも良いことにはアルゴリズムの正当性は Turing 機械の動作からほとんど明白で, その証明が不要なことである。

3.3.3 並列計算モデル間の能力比較

並列計算モデル間の能力比較を行うことは一般に困難であり, たとえば, ある仕事は線形時間 SA で解決不可能なことを証明する手法は知られていなかった。並列計算モデル上では対角線論法が使えないあるいは使いにくいというのが主な理由である。

一方 Turing 機械に関しては, 複雑さの理論の進歩とともに 1965 年ごろから 1970 年代にかけて膨大な知識が集積されており, Turing 機械間の能力比較を行う種々の技法が開発されている。

Ibarra らの特徴付け定理は従来困難であるとされていた並列計算モデル間の能力比較を Turing 機械上で行うことを可能とするもので, 数多くの結果が明らかにされている。たとえば, CA の言語受理能力関係に関しては, 文献⁶⁰⁾を参照されたい。

以上 Ibarra らの特徴付け定理の有効性を示す例として, これまでに提案されている IA 族の統一的特徴付け, それらの能力関係, IA アルゴリズムの設計例などについて述べてきたが, 先にあげた定理 2, 3 も同様な手法により得られた新しい結果である。他にも多数興味ある結果が得られているが, 本稿では省略する。文献^{55)~66)}を参照されたい。

4. おわりに

並列アルゴリズムに対する興味は近年急速に高まりつつある。Kung and Leiserson らによる提案以来, SA に関する研究は世界中で精力的に進められている。本稿ではシストリック・アレイ/アルゴリズム/アーキテクチャに関する基本的な事柄を整理し, 数多くのバリエーションをもつ SA に統一的な見方を与えた。これらのなかには, シストリック・コンセプト以前の Iterative array やセルラオートマトンに関する結果も含まれる。

最近の話題として, Ibarra らの Turing 機械によるシストリック・アレイの特徴付け定理を紹介した。SA の挙動は, 逐次的な見方が可能となり今後ますます研究が進むものと期待される。他にとりあげるべき

重要な課題として, 最近急速に開発されつつあるシストリック・コンピュータ, 数多くの方法が提案されているシストリック・アルゴリズムの設計手法, ならびに SA におけるフォールト・トレランス問題などが考えられるが, 紙面の都合で省略する。詳細は各文献^{21)~39), 67)~71), 73), 74), 77), 78)}を参照されたい。

謝辞 本研究の一部分は Alexander von Humboldt 財団により援助されている。本稿をまとめる機会を与えていただいた守屋悦朗教授 (東京女子大) に, 数多くの参考文献を送っていただいた Prof. H. T. Kung (Carnegie-Mellon Univ.), Prof. K. Culik II (South Carolina Univ.), Prof. O. H. Ibarra (Minnesota Univ.) らに, 最後に筆者が西ドイツに滞在中に (1987 年 3 月~1988 年 3 月)にご討論いただいた Prof. R. Vollmar (Braunschweig Univ.) に謝意を表す。

参 考 文 献

参考文献は,

- シストリック・コンセプト以前のもの,
 - SA を紹介した初期の文献ならびに SA に関する全般的な解説文献,
 - シストリック・アルゴリズムの設計ならびに検証手法に関するもの,
 - シストリック辞書構造に関するもの,
 - シストリック・オートマトンに関するもの,
 - Ibarra, Palis and Kim らの特徴付け定理に関するもの, ならびに,
 - シストリック・コンピュータに関するもの
- の 7 つの項目別に分類されている。個々のシストリック・アルゴリズムを扱った文献は紙面の都合もあり省略した。SA に関する文献リストとして, 文献²⁰⁾などがある。
- シストリック・コンセプト以前のもの
 - 1) Hennie III, F. C.: Iterative Arrays of Logical Circuits, MIT Press, John Wiley Sons Inc., New York (1961).
 - 2) Atrubin, A. J.: A One-Dimensional Real-Time Iterative Multiplier, IEEE Trans. on Electronic Computers, Vol. EC-14, pp. 394-399 (1965).
 - 3) Fischer, P. C.: Generations of Primes by a One-Dimensional Real-Time Iterative Array, JACM, Vol. 12, No. 3, pp. 388-394 (1965).
 - 4) Von Neumann, J.: Theory of Self-Reproducing Automata (edited and completed by Burks, A. W.), Univ. of Illinois Press (1966).
 - 5) Codd, E. F.: Cellular Automata, Academic Press, New York (1968).
 - 6) Cole, S. N.: Real-Time Computation by n-Dimensional Iterative Arrays of Finite-State-Machines, IEEE Trans. on Computers, Vol. C-

- 18, No. 4, pp. 349-365 (1969).
- 7) Beyer, W. T.: Recognition of Topological Invariants by Iterative Arrays, Project Mac. Tech-Rep, TR-66, MIT, Cambridge, MA(1969).
- SAを紹介した初期の文献ならびにSAに関する全般的な解説文献
- 8) Kung, H. T. and Leiserson, C. E.: Systolic Arrays (for VLSI), Sparse Matrix Proc. 1978 (eds. Duff, I. S. and Stewart, G. W.), Academic Press, Orlando, Fla., pp. 256-282 (1979) also in Algorithms for VLSI Processor Arrays, which is Section 8.3 of Introduction to VLSI Systems (by Mead, C. and Conway, L.) Addison-Wesley, Reading Mass., pp. 271-292 (1980). 菅野, 榊監訳: 超 LSI システム入門, 培風館 (1981).
- 9) Kung, H. T.: Let's Design Algorithms for VLSI Systems, Proc. of the Caltech Conf. on VLSI (ed. Deitz, C. E.), pp. 55-90 (1979).
- 10) Leiserson, C. E.: Systolic Priority Queues, Proc. of the Caltech Conf. on VLSI (ed. Deitz, C. E.), pp. 199-214 (1979).
- 11) Foster, M. J. and Kung, H. T.: The Design of Special-Purpose VLSI Chips, Computer, Vol. 13, No. 1, pp. 26-40 (1980).
- 12) Kung, H. T.: The Structure of Parallel Algorithms, Advances in Computers, Vol. 19, pp. 65-112, Academic Press (1980).
- 13) Kung, H. T., Sproull, R. F. and Steel Jr., G. L. (eds.): VLSI Systems and Computations, Computer Science Press (1981).
- 14) Kung, H. T.: Why Systolic Architectures?, IEEE Computer, Vol. 15, No. 1, pp. 37-46 (1982).
- 15) 都倉: VLSI アルゴリズムおよび面積時間複雑度, 情報処理, Vol. 23, No. 3, pp. 176-186 (1982).
- 16) Leiserson, C. E.: Area-Efficient VLSI Computation, MIT Press (1983).
- 17) 坂村, 石川: VLSI コンピュータ・アーキテクチャ, 情報処理, Vol. 24, No. 2, pp. 156-175 (1983).
- 18) Ullman, J. D.: Computational Aspect of VLSI, Computer Science Press (1984).
- 19) Moore, W., McCabe, A. and Urquhart, R. (eds.): Systolic Arrays, Proc. of Intern. Workshop on Systolic Arrays, Oxford, 1986, July, Adam Hilger, Bristol and Boston, Mass., p. 334 (1987).
- 20) Kung, H. T.: A Listing of Systolic Papers, Dept. of Computer Sci., Carnegie-Mellon Univ. (Oct. 1987).
- シストリック・アルゴリズムの設計ならびに検証手法に関するもの
- 21) Guibas, L. J., Kung, H. T. and Thompson, C. D.: Direct VLSI Implementation of Combinatorial Algorithms, Proc. of the Caltech Conf. on VLSI (ed. Deitz, C. E.), pp. 509-525 (1979).
- 22) Weiser, U. and Davis, A.: A Wavefront Notation Tool for VLSI Array Design, VLSI Systems and Computations, pp. 226-234, Computer Science Press (1981).
- 23) Choffrut, C. and Culik II, K.: Folding of the Plane and the Design of Systolic Arrays, Information Processing Letters, Vol. 17, pp. 149-153 (1983).
- 24) Leiserson, C. E. and Saxe, J. B.: Optimizing Synchronous Systems, J. of VLSI and Computer Systems, Vol. 1, No. 1, pp. 41-67 (1983).
- 25) Capello, P. R. and Steiglitz, K.: Unifying VLSI Array Designs with Geometric Transformations, Proc. of IEEE Intern. Conf. on Parallel Processing, pp. 448-457 (1983).
- 26) Moldovan, D. I.: On the Design of Algorithms for VLSI Systolic Arrays, Proc. of the IEEE, Vol. 71, No. 1, pp. 113-120 (1983).
- 27) Melhem, R. G. and Rheinboldt, W. C.: A Mathematical Model for the Verification of Systolic Networks, SIAM J. Comput., Vol. 13, No. 3, pp. 541-565 (1984).
- 28) Miranker, W. L. and Winkler, A.: Space-Time Representations of Systolic Computational Structures, Computing, Vol. 32, pp. 93-114 (1984).
- 29) Culik II, K. and Fris, I.: Topological Transformations as a Tool in the Design of Systolic Networks, Theoretical Computer Sci., Vol. 37, pp. 183-216 (1985).
- 30) 梅尾: 実時間シストリック・コンボルバの設計について, 電子通信学会論文誌, Vol. J68-D, No. 11, pp. 1838-1845 (1985).
- 31) Lam, M. S. and Mostow, J.: A Transformational Model of VLSI Systolic Design, IEEE Computer, Vol. 18, No. 2, pp. 45-52 (1985).
- 32) Fortes, J. A. B., Fu, K. S. and Wah, B. W.: Systematic Approaches to the Design of Algorithmically Specified Systolic Arrays, Proc. of 1985 Intern. Conf. on Acoustics, Speech, and Signal Processing, IEEE, pp. 8.9.1-8.9.5 (1985).
- 33) Li, G.-J. and Wah, B.: The Design of Optimum Systolic Arrays, IEEE Trans. on Computers, Vol. C-34, pp. 66-77 (1985).
- 34) Umeo, H.: A Class of SIMD Machines Simulated by Systolic Arrays, J. of Parallel and Distributed Computing, Vol. 2, No. 4, pp. 391-403 (1985).
- 35) 萩原, 和田: ハードウェア・アルゴリズムの記述と検証, 情報処理, Vol. 25, No. 6, pp. 603-612

- (1985).
- 36) Umeo, H.: A Time-Optimum Systolic Simulation of One-Way Cellular Automata, *J. of Information Processing*, Vol. 8, No. 3, pp. 222-224 (1985).
 - 37) Umeo, H.: A Time-Optimum Systolic Simulation of Two-Way Linear-Time Cellular Automata, *The Trans. of IECE of Japan*, Vol. E 69, No. 3, pp. 196-199 (1986).
 - 38) Moldvan, D. I. and Fortes, J. A. B.: Partitioning and Mapping Algorithms into Fixed Size Systolic Arrays, *IEEE Trans. on Computers*, Vol. C-35, No. 1, pp. 1-12 (1986).
 - 39) 三浦, 阿曾, 稲垣: 多重ループプログラムを処理するシストリックアルゴリズムの構成法, *電子情報通信学会論文誌*, Vol. J 70-D, No. 3, pp. 515-524 (1987).
 - シストリック辞書構造に関するもの
 - 40) Leiserson, C. E.: Systolic Priority Queues, *Caltech Conf. on VLSI*, pp. 199-214 (1979).
 - 41) Bentley, J. L. and Kung, H. T.: A Tree Machine for Searching Problems, *IEEE International Conf. on Parallel Processing*, pp. 257-266 (1979).
 - 42) Leiserson, C. E.: *Area-Efficient VLSI Computation*, Cambridge, MA: M. I. T. Press (1983).
 - 43) Ottmann, T. A., Rosenberg, A. L. and Stockmeyer, L. J.: A Dictionary Machine (for VLSI), *IEEE Trans. on Computers*, Vol. C-31, No. 9, pp. 892-897 (1982).
 - 44) Somani, A. K. and Agawal, V. K.: An Efficient VLSI Dictionary Machine, *The 11th Annual Symp. on Computer Architecture*, pp. 142-150 (1984).
 - 45) Somani, A. K. and Agawal, V. K.: An Efficient Unsorted VLSI Dictionary Machine, *IEEE Trans. on Computers*, Vol. C-34, No. 9, pp. 841-852 (1985).
 - 46) Atallah, M. J. and Kosaraju, S. R.: A Generalized Dictionary Machine for VLSI, *IEEE Trans. on Computers*, Vol. C-34, No. 2, pp. 151-155 (1985).
 - シストリック・オートマトンに関するもの
 - 47) Smith, A. R. III: Real-Time Language Recognition by One-Dimensional Cellular Automata, *J. of Comput. System Sci.*, Vol. 6, pp. 233-253 (1972).
 - 48) Kosaraju, S. R.: Speed of Recognition of Context-Free Languages by Array Automata, *SIAM J. Comput.*, Vol. 4, pp. 331-340 (1975).
 - 49) Seiferas, J. I.: Iterative Arrays with Direct Central Control, *Acta Informatica*, Vol. 8, pp. 177-192 (1977).
 - 50) Dyer, C. R.: One-Way Bounded Cellular Automata, *Inf. and Contrl.*, Vol. 44, pp. 261-281 (1980).
 - 51) Dyer, C. R. and Rosenfeld, A.: Triangle Cellular Automata, *Inf. and Contrl.*, Vol. 48, pp. 54-69 (1981).
 - 52) Culik II, K., Gruska, J. and Salomaa, A.: Systolic Automata for VLSI Balanced Trees, *Acta Informatica*, Vol. 18, pp. 335-344 (1983).
 - 53) Culik II, K., Salomaa, A. and Wood, D.: Systolic Tree Acceptors, *R. A. I. R. O., Informatique Theorique*, Vol. 18, No. 1, pp. 53-69 (1984).
 - 54) Umeo, H., Worsch, T. and Vollmar, R.: Broadcast Elimination without Any Loss of Time Efficiency in Iterative (Cellular or Systolic) Arrays, *Proc. of Cellular Meeting in Braunschweig* (1988).
 - Ibarra, Palis and Kim らの特徴付け定理に関するもの
 - 55) Ibarra, O. H., Palis, M. A. and Kim, S. M.: Designing Systolic Algorithms Using Sequential Machines, *Proc. of FOCS*, pp. 46-55 (1984).
 - 56) Ibarra, O. H. and Kim, S. M.: A Characterization of Systolic Binary Tree Automata and Applications, *Acta Informatica*, Vol. 21, pp. 193-207 (1984).
 - 57) Ibarra, O. H. and Kim, S. M.: Characterizations and Computational Complexity of Systolic Trellis Automata, *Theoret. Comput. Sci.*, Vol. 29, pp. 123-153 (1984).
 - 58) Ibarra, O. H., Kim, S. M. and Moran, S.: Sequential Machine Characterizations of Trellis and Cellular Automata and Applications, *SIAM J. of Computing*, Vol. 14, pp. 426-447 (1985).
 - 59) Ibarra, O. H., Palis, M. A. and Kim, S. M.: Fast Parallel Language Recognition by Cellular Automata, *Theoretical Computer Sci.*, Vol. 41, pp. 231-246 (1985).
 - 60) Ibarra, O. H., Palis, M. A. and Kim, S. M.: Some Results Concerning Linear Iterative (Systolic) Arrays, *J. of Parallel and Distributed Computing*, Vol. 2, pp. 182-218 (1985).
 - 61) Ibarra, O. H., Kim, S. M. and Palis, M. A.: Designing Systolic Algorithms Using Sequential Machines, *IEEE Trans. on Computers*, Vol. C-35, No. 6, pp. 531-542 (1986).
 - 62) Ibarra, O. H.: Systolic Arrays: Characterization and Complexity, *Proc. of MFCS '86*, pp. 140-153, *Lecture Notes on Computer Science*, Vol. 233, Springer-Verlag, New York/Berlin (1986).
 - 63) Ibarra, O. H. and Palis, M. A.: Two-Dimensional Iterative Arrays: Characterization and Application, *Tech. Rep. of University Minnesota* (1986).
 - 64) Chang, J. H., Ibarra, O. H. and Vergis, A.: On the Power of One-Way Communication,

- Proc. of FOCS, pp. 455-464 (1986).
- 65) Ibarra, O. H. and Palis, M. A.: On Efficient Simulations of Systolic Arrays by Random-Access Machines, SIAM J. on Comput., Vol. 16, No. 2, pp. 367-377 (1987).
- 66) Chang, J. H., Ibarra, O. H. and Palis, M. A.: Parallel Parsing on a One-Way Array of Finite State Machines, IEEE Trans. on Computers, C-36, No. 1, pp. 64-75 (1987).
- シストリック・コンピュータに関するもの
- 67) Bromly, K., Symanski, J. J., Speiser, J. M. and Whitehouse, H. J.: Systolic Array Processor Developments, VLSI Systems and Computations, pp. 273-284, Computer Science Press (1981).
- 68) Fisher, A. L., Kung, H. T., Monier, L. M. and Dohi, Y.: Architecture of the PSC: A Programmable Systolic Chip, Proc. of the 10th Ann. Symp. on Comp. Architecture, pp. 48-53 (1983).
- 69) Fortes, J. A. B. and Wah, W. (eds.): IEEE Computer Special Issue on Systolic Array (July 1987).
- 70) Annaratone, M., Arnould, E., Gross, T., Kung, H. T., Lam, M., Menzilcioglu, O. and Webb, J. A.: The Warp Computer: Architecture, Implementation and Performance, IEEE Trans. on Computers, Vol. C-36, No. 12, pp. 1523-1538 (1987).
- 71) 土肥: シストリック・マシン, 情報処理, Vol. 28, No. 1, pp. 68-73 (1987).
- 補遺
- 72) Bentley, J. L. and Kung, H. T.: An Introduction to Systolic Algorithm and Architectures, Nav. Res. Rev. (USA), Vol. 35, No. 2, pp. 3-16 (1983).
- 73) Kung, H. T. and Lam, M. S.: Fault-Tolerance and Two-Level Pipelining in VLSI Systolic Arrays, Proc. of the Conf. on Advanced Research in VLSI, MIT (1984).
- 74) Leighton, T. and Leiserson, C. E.: Wafer-Scale Integration of Systolic Arrays, IEEE Trans. on Comput., Vol. C-35, No. 5, pp. 448-461 (1985).
- 75) Umeo, H.: Parallel Address Setting Schemes for Complete-Binary-Tree-Connected Machines, J. of Information Processing, Vol. 9, No. 4, pp. 237-240 (1986).
- 76) 梅尾, 浅野: シストリック計算幾何アルゴリズムに関する最近の研究, 情報処理, Vol. 27, No. 11, pp. 1270-1281 (1986).
- 77) Culik, II, K. and Yu, S.: Fault-Tolerant Schemes for Some Systolic Systems, Intern. J. Comput. Math., Vol. 22, pp. 13-42 (1987).
- 78) Abraham, J. A.: Fault Tolerance Technique for Systolic Arrays, Computer, Vol. 20, No. 7, pp. 65-75 (1987).

(昭和 63 年 5 月 13 日受付)