

シストリック・アーキテクチャとそのアルゴリズム

梅尾 博司

1. はじめに

“蜂の巣六角アレイ”と“そして淀みなき流れは、さらに淀みなきリズムにのって流れる”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 [1][2][3][4] の2人により提案されたVLSI向き並列計算機モデルである。

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は心臓に見立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、血液を全身に送り出し汲み出す動作からのアナロジーである。

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) に行ない、しかもこれらの動作はパイプライン (pipeline) 化され重畳的 (overlapping) に進められるという点にある。

本稿では、シストリック・アーキテクチャとそのアルゴリズムに焦点をあて、シストリック・コンセプトをわかりやすく解説する。まずシストリック・アーキテクチャの入門的説明を行ない、次にシストリック・アルゴリズムの典型例と考えられる全点対検査問題をとりあげる。

2. シストリック・アレイ

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

うめお ひろし 大阪電気通信大学

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 [3] らは、SAを上記のサブルーチン・パッケージにかわる特定用途向けハードウェア付加装置として使用されると考えた。図1にその利用形態を示す。

各SAは図1に示すように、通常ノイマン型計算機のシステム・バスに接続され、その周辺機器の1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サを規則正しく接続したもものからなる。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に行なわれる。

SAには多数のバリエーションが存在する。図2は代表的なSAの基本結合方式である。本稿では理解ならびに説明の容易さから、最も基本的なモデルと考えられる1次元SA、Mをとりあげ、シストリック・コンセプトを説明しよう。Mは図3に示すように、1個のバッファと1次元的に配置された多数の同一プロセッサからなる。SAは次の特徴をもつ。

局所結合ならびに局所通信

各プロセッサをセルと呼び、 i 番目のセルを C_i で示す。各セル $C_i (i \geq 1)$ は左右に隣接するセル C_{i-1} , C_{i+1} とのみ直接に接続される。このような結合を局所結合と呼ぶ。また C_i の左右に隣接するセル C_{i-1} および C_{i+1} を C_i の近傍と呼ぶ。各セルが1ステップ (単位時間) で直接通信できるのは、近傍内のセルに限定される。こ

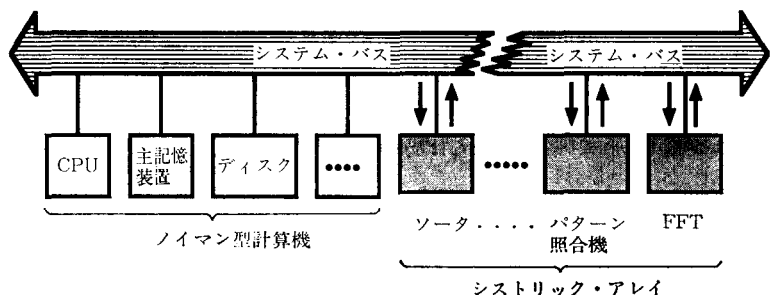


図1 シストリック・アレイの利用形態

れを局所通信と呼ぶ。したがって、お互いに離れたセル C_i, C_j 間でデータの授受を行なうには、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なわれる。

均一構造

すべてのセルは単純で同一構造を持ち、たとえば実数の四則演算など基本的な計算を1ステップで実行可能である。

逐次的な入出力動作

$t=0$ 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が始まる。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、1ステップで2個のデータ、……というぐあいにバッファを経由して与えられる。この割合を入力パイプライン・インタバル(input-pipeline-interval)と呼ぶ。上記の場合、それぞれ1, 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。このような方式を逐次入力と呼ぶ。

出力 $b_1, b_2, \dots, b_k, \$$ も入力と同様に逐次的にアレイからバッファを経由して出力される。出力 b_i と b_{i+1} ($1 \leq i \leq k$) の時間間隔を出力パイプライン・インタバルと呼ぶ。

同期モデル

各セルは同期(synchronous)して動作する。そのため、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)が備えられている。これらは局所的な信号線でない。

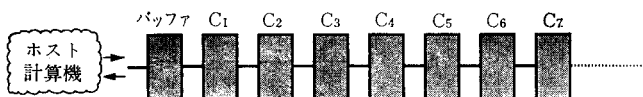


図3 1次元シストリックアレイ

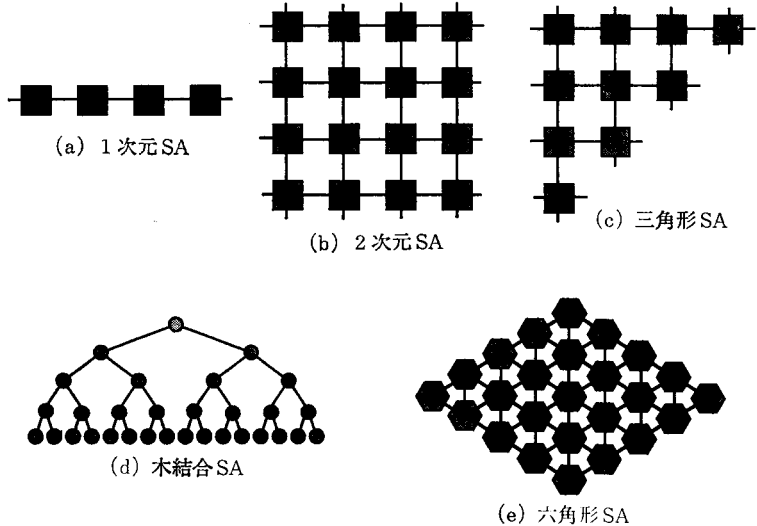


図2 シストリック・アレイの基本結合方式
アミかけの部分はプロセッサを、それらを接続する細線はローカル通信リンクを意味する。

計 算 量

最初の入力 a_1 を与えてから最後の出力 $\$$ が得られるまでのステップ数をSAの時間計算量(time complexity)と呼び $T(n)$ で表わす。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。 a^1 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(response time)と呼び、 $R(n)$ で表わす。

一般的に次々とサイズの異なるデータ・セットを処理するケースがある。このとき、1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(period time)と呼び、 $P(n)$ で表わす。 $T(n)$ 、 $R(n)$ ならびに $P(n)$ は、解くべき問題、SAのアーキテクチャおよびアルゴリズムに依存する。 $T(n)$ は線形時間、すなわち、 $T(n) = c \cdot n$ (c は正整数)、 $R(n)$ 、 $P(n)$ は $O(1)$ 、 $S(n) = n$ が望ましい。

次の2つの定理はSA上での最適時間アルゴリズムを設計する際に大変有用である。

[定理1][17] パイプライン・インタバル縮小定理

任意の自然数 k に対し、図4(a, b)に示すSA上で k パイプライン・インタバルを1に縮小できる。

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。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

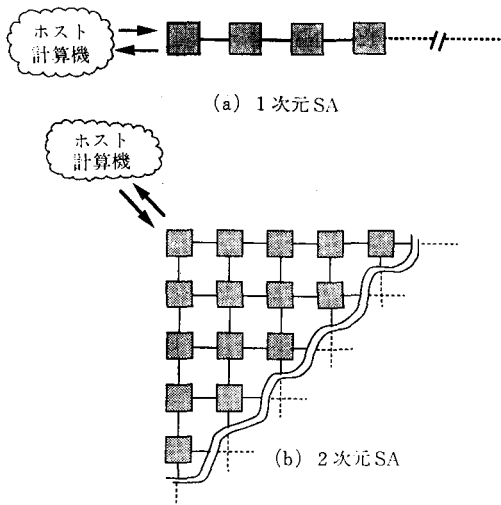


図 4 入力パイプライン・インタバルを1に短縮できるシストリック・アーキテクチャ

あるので、設計後変更するという手段が考えられる。

シストリック・アレイの最大の特徴は、すべてのプロセッサ間通信は局所結合のみを通じて行なわれるという点にある。このようなシストリック・アレイにもう1本の通信線、ブロードキャスト・バスを与えよう(図5(a))。ただし、任意のプロセッサがバスを通じて遠く離れた他のプロセッサに情報を送れるような一般的なバスではない。ホスト計算機のみが、バスを通じて全プロセッサに一齐に1方向的に情報をブロードキャストできるタイプである。1回のブロードキャストに要する時間は1ステップと考える。各プロセッサはホスト計算機から情報を受け取るのみであるので、この種のバスを1方向ブロードキャスト・バスと呼ぶ。

各プロセッサ $C_i (i \geq 1)$ は、 C_{i-1} 、 C_i 、 C_{i+1} のもつローカル情報、およびホスト計算機がブロードキャストするグローバル情報、合計4種類の情報にもとづいて次の時刻における動作を決定する。ホスト計算機から遠く離れたプロセッサは、いつも時間遅れなくホストの情報を知ることができる点が、従来のシストリック・アレイと違う点である。従来のアレイを純シストリック・アレイ (pure systolic array)、1方向ブロードキャスト・バスを備えたアレイを準シストリック・アレイ (semi systolic array) と呼ぶ。

純シストリック・アレイ上では、各プロセッサは1次元線上に結合され、プロセッサ間通信は局所通信しか許されていないので、アレイの後方 (I/O プロセッ

サから遠く離れた位置にあるプロセッサ群) は、先頭でどのような動作が起こっているか実時間で知ることができない。必ずそのプロセッサと I/O プロセッサとの間の距離に比例した時間遅れを生じる。これが純シストリック・アレイの特徴であり、高速アルゴリズムの設計を困難にしている点である。

次の定理2を用いることにより、時間計算量をまったく変えることなく、ブロードキャスト・バスを利用した準シストリック・アレイ上のすべての並列アルゴリズムを、バスを利用しないタイプに変換できることが知られている。

[定理2] [18, 19, 27, 29] 1方向ブロードキャスト・バスの除去定理

時間計算量 $T(n)$ の任意の準シストリック・アレイ (図5(a)) M に対し、 M の動作を $T(n)$ ステップで模倣する純シストリック・アレイ (図5(b)) N が存在し、 N は M より容易に構成できる。

3. シストリック全点对検査アルゴリズム

次の全点对検査問題 (all-pair examination problem) を効率よく解くシストリック・アルゴリズムを設計しよう。

全点对検査問題

n 個の点要素 a_1, a_2, \dots, a_n が与えられる。このとき、自分自身との点对を除くすべての点对 (a_1, a_2) , (a_1, a_3) , (a_1, a_4) , \dots , (a_1, a_n) , (a_2, a_1) , (a_2, a_3) , (a_2, a_4) , \dots , (a_2, a_n) , \dots , (a_n, a_{n-1}) を検査せよ。

点对の総数が $n(n-1)$ 個あることから、ノイマン型逐次計算機では $O(n^2)$ 時間を要することは明らかである。すべての点が、自分自身を除いて他のすべての点と少なくとも一度は出会う必要がある。この問題は、ソーティング、計算幾何学、信号・画像処理などの並列アル

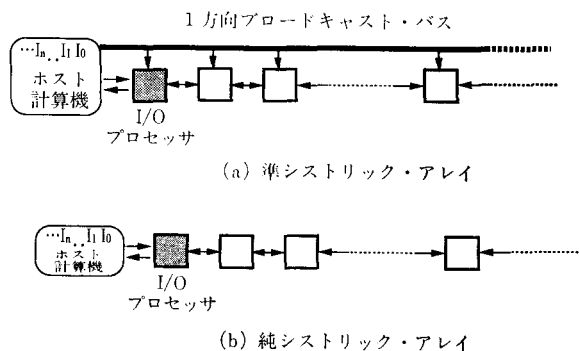


図 5 計算能力が等価なシストリック・アレイモデル

ゴリズムを1次元SA上で設計する場合、基本的な役割を果たすことが知られている。たとえばソートング問題では、 a_i は a_1, a_2, \dots, a_n とのすべて点対を検査する際、同時に大小比較を行ない、 a_i より小さいデータ数、すなわち n 個の要素中での a_i のランクを知ることができる。ランキングができれば、ソートングは容易である。

〔シストリック・アルゴリズム〕

次のような考え方をする。いま n 個のデータ a_1, a_2, \dots, a_n がプロセッサ P_1, P_2, \dots, P_n にあらかじめ与えられているとしよう。次に左端から a_1, a_2, \dots, a_n の順にデータがパイプライン化され、右方向に流れてくるとする。スピードは1、すなわち1プロセッサ/1単位時間とする。図6(a)参照。各プロセッサ P_i は単位時刻ごとに自分が蓄えているデータ a_i と流れてくるデータ a_j とを対にして検査することができる。図6(b)は、 $t=0$ のとき P_1 上でデータ (a_1, a_1) が検査されている状態から出発したデータ・フローを表現している。簡単化のため、数字1, 2, 3, ... はデータ a_1, a_2, a_3, \dots を意味する。各プロセッサは上下2つのレジスタをもつ。上のレジスタはアレイ上でデータをパイプライン的に流す役目をし、下のレジスタは初期データを保存する。 P_1 に注目すると、 $t=0$ より各時刻ごとに $(a_1, a_1), (a_1, a_2), (a_1, a_3), (a_1, a_4), \dots, (a_1, a_n)$ の各点対が1対/1ステップの割合で検査されている様子が明らかになろう。他のプロセッサ P_2, P_3, P_4, \dots においても同様である。自分自身どうしの点対検査を排除するために、次の事実を利用する。上記のデータ・フローで計算すると、点対 (a_i, a_i) は P_i 上で $t=2i-2$ のときに限り出会う。したがって、計算開始と同時に P_1 はスピード1/2で右方向に進むパルスを生成する。このパルスが P_i に伝えられているか否かにより、いま検査している点対 (a_i, a_j) において $i=j$ か否かを知ることができる。図6(b)の矢印はパルスの伝播を意味している。

アレイ上にデータがあらかじめロードされていない場合、点データのコピーを2つ用意し、図7(a)に示す時間・空間図式で計算させる。水平方向はアレイ空間を、垂直方向は時間軸を意味する。アレイ上での動作は、次の3相からなる。すなわち、

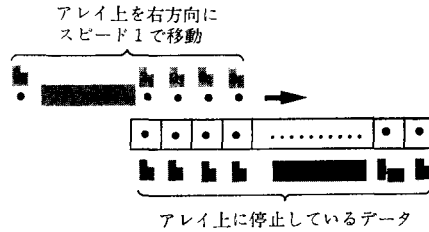


図 6(a) アレイ上を移動しているデータ(上側)と停止しているデータ(下側)とのマッチング

| | P_1 | P_2 | P_3 | P_4 | P_5 | P_6 |
|-------|--------|--------|--------|--------|--------|--------|
| $t=0$ | ■ | 2 | 3 | 4 | 5 | 6 |
| $t=1$ | 2 1 | 1 2 | 3 | 4 | 5 | 6 |
| $t=2$ | 3 1 | ■ | 1 3 | 4 | 5 | 6 |
| $t=3$ | 4 1 | 3 2 | 2 3 | 1 4 | 5 | 6 |
| $t=4$ | 5 1 | 4 2 | ■ | 2 4 | 1 5 | 6 |
| $t=5$ | 6 1 | 5 2 | 4 3 | 3 4 | 2 5 | 1 6 |
| $t=6$ | 7 1 | 6 2 | 5 3 | ■ | 3 5 | 2 6 |

図 6(b) 全点対検査のためのデータ・フロー

- (1) データを各プロセッサにロードする相、
- (2) 点対検査相、
- (3) 出力相。

の3相に分けられる。これらの3相が時間が時間、空間両軸上でオーバーラップ化され、効率よく処理されている様子が明らかであろう。

データのロードにはフォールディング(folding)法を使用する。フォールディングはシストリック・アレイ上でよく使用されるデータ・ルーティング法のひとつで、データ列 a_1, a_2, \dots, a_n の一端 a_1 を固定したロープとみため、ロープが2つ折りになるようにデータを移動させてゆく手法である。折点は時間とともにダイナミックに右方向に移動する(図7(b)参照)。ロープの固定端を C_1 と考える。

出力は1個/2ステップの割合で得られる。全出力が得られるのは入力開始後 $4n+O(1)$ ステップ後である。 $O(1)$ の項は、データの初期配置などに依存する定数ステップを意味する。

ノイマン型逐次計算機上では $O(n^2)$ 時間かかる問題が、1次元SAを利用すると $O(n)$ 時間で解けることが明らかになった。

ここに示した線形時間・点対検査アルゴリズムは、計算幾何学問題の1つとして知られる最近点対問題の解法でもある。

最近点対問題とは、 n 個の点 a_1, a_2, \dots, a_n が与えられたとき、すべての $a_i(1 \leq i \leq n)$ について、自身を除く

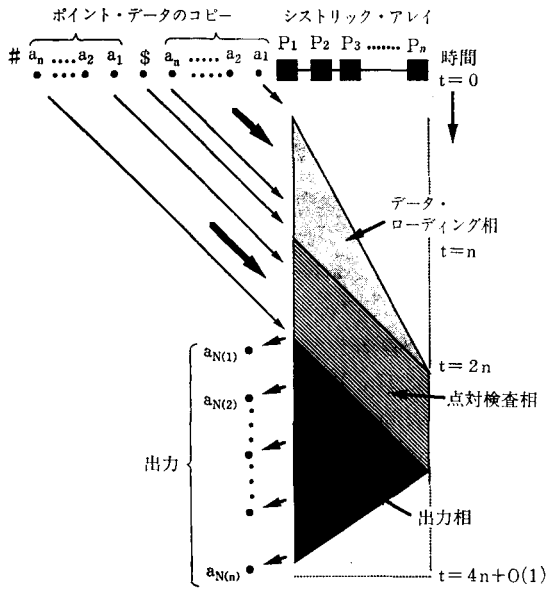


図 7(a) 全点对検査アルゴリズムのための時間・空間図式

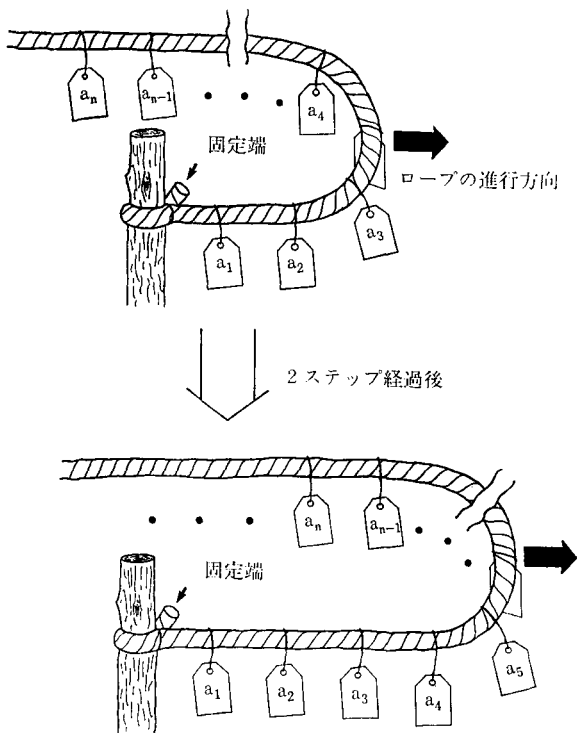


図 7(b) フォールディングによるデータ・ルーティング

$n-1$ 個の点の中で a_i に最も近い点 $a_{N(i)}$ を求める問題である。逐次計算機では Voronoi データ構造を利用す

ることにより、 $O(n \log n)$ 時間で n 点の最近点対を見い出せることが知られている。

a_i の最近点 $a_{N(i)}$ を求めるには、すべての $j (1 \leq j \leq n, j \neq i)$ に対して、 a_i と a_j の距離を計算し、その中で最小距離を与える点 a_j を $a_{N(i)}$ とすればよい。各 P_i は a_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 a_i と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。このために図 6 (b) に示した 2 個のレジスタのほかに 1 個のレジスタを備えているものとする。各 P_i はエンドマーク # を読んだとき、プロセッサ内に最近点 $a_{N(i)}$ を保持している。その後、これらのデータは左方向にシフトされ出力される。

本節でとりあげたシストリック点对検査アルゴリズムの計算幾何学問題への応用は、文献 [36] を参照されたい。数多くの計算幾何学問題が、 n 台のプロセッサからなる 1 次元 SA 上で $O(n)$ 時間で解決可能なことが示されている。

4. おわりに

シストリック・アレイはプロセッサ粒度が非常に細かい (超細粒度) SIMD 型並列計算モデルである。同様なものにセルラ・アレイ、イテラティブ・アレイ等がある。これらはメッシュ結合を基本結合方式とする並列計算モデルで古くから数多くの研究がなされている [37]~[44]。DAP, Mas Par, Zephyr, MAPLE など、この分野の研究をベースとして生まれた並列計算機が市販されている。これらは、AP-1000, CM-5, NCUBE, ADENART など MIMD 型中粒度並列計算機と並び商用並列計算機の一翼を形成している。

本稿では、シストリック・アーキテクチャとそのアルゴリズムの入門的解説を行なった。紙面の都合もあり、シストリック・アルゴリズムの設計手法など割愛せざるを得ないテーマも数多くあった。最後に文献ガイドを示す。シストリック・アレイの入門的解説として、文献 [1]~[16] が、またシストリック・アレイが提案される以前の関連研究として、文献 [17], [37]~[44] があげられる。シストリック・アルゴリズムの設計手法をとりあげたものとして、文献 [18]~[35] があげられる。

謝 辞

本稿執筆の機会を与えていただいた都倉信樹教授 (大阪大学)、ならびに中森眞理雄教授 (東京農工大学) に謝意を表す。

参 考 文 献

- [1] Kung, H. T.: "Why systolic architectures?", *IEEE Computer*, Vol.15, No.1, pp.37-46, 1982.
- [2] Kung, H. T. & Leiserson, C. E.: "Systolic arrays (for VLSI)", *Sparse Matrix Proc. 1987* (Duff, I. S. & Stewart, G. W. eds.), pp.256-282, Academic Press, 1979, also in "Algorithms for VLSI processor arrays", which is Section 8.3 of *Introduction to VLSI Systems* (by Mead, C. & Conway, L.), pp.271-292, Addison-Wesley, Reading Mass., 1980. 菅野卓雄, 榊裕之 (監訳): 『超L S I システム入門』, 培風館, 1981.
- [3] Kung, H. T.: "Let's design algorithms for VLSI systems", *Proc. of the Caltech Conf. on VLSI* (Deitz, C. E. ed.), pp.55-90, 1979.
- [4] Leiserson, C. E.: "Systolic priority queues", *Proc. of the Caltech Conf. on VLSI* (Deitz, C. E. ed.), pp.199-214, 1979.
- [5] Fortes, J. A. B. & Wah, W. (eds.): *IEEE Computer*, Special issue on Systolic Arrays, July, Vol.20, No.7, 1987.
- [6] Bromley, K., Kung, S. Y., & Swartzlander, E. (eds.): 1988 *IEEE Proc. of Intern. Workshop on Systolic Arrays*, 1988.
- [7] 梅尾博司: 「シストリック・アレイ」, *情報処理*, 第30巻, 1号, pp.15-28, 1989.
- [8] McCanny, J., McWhirter, J., and Swartzlander Jr. E. (eds.): "Systolic Array Processors", Prentice Hall (1989).
- [9] 梅尾博司: 「超並列計算機アーキテクチャとそのアルゴリズム」, p.216, 共立出版 (1991).
- [10] Quinton, P. and Robert, Y.: "Systolic Algorithms and Architectures", Prentice Hall, p.364 (1989).
- [11] Tchente, M.: *Parallel computation on regular arrays*, p.315, Halsted Press (1991).
- [12] Leighton, F. T.: *Introduction to parallel algorithms and architectures: Arrays-Trees-Hypercubes*, p.831, Morgan Kaufmann (1992).
- [13] 都倉信樹, 萩原兼一: "VLSI 用の行列計算", *情報処理*, Vol.24, No.4, pp.558-562 (1983).
- [14] 吉田典可, 菊野享, 藤井隆志: "シストリック配列とバス付きセル配列上のハードウェアアルゴリズム", *情報処理*, Vol.26, No.6, pp.613-621 (1985).
- [15] 安浦寛人: "シストリックアルゴリズム", *電子情報通信学会誌*, Vol.71, No.8, pp.843-845 (1988).
- [16] 梅尾博司: 並列処理入門メッシュ・アーキテクチャとそのアルゴリズム (1)~(4), *Computer Design*, Vol.8, No.11 (1991), No.1~3 (1992).
- [17] Cole, S. N.: "Real-time computation by n-dimensional iterative arrays of finitestate machines", *IEEE Trans. on Computers*, Vol.C-18, No.4, pp.349-365, 1969.
- [18] Umeo, H., Worsch, T., & Vollmar, R.: "Broadcast-bus elimination without any loss of time efficiency in iterative (cellular or systolic) arrays", *Trans. of IEICE*, Vol.E-72, No.5, pp.647-655, 1989.
- [19] Umeo, H., Worsch, T., & Vollmar, R.: "On the power of global-bus in mesh-connected architectures", *Future Generation Computer Systems*, Vol.7, pp.1-8 (1992).
- [20] Choffrut, C. & Culik II, K.: "Folding of the plane and the design of systolic arrays", *Information Processing Letters*, 17, pp.149-153, 1983.
- [21] Culik II, K. & Fris, I.: "Topological transformations as a tool in the design of systolic networks", *Theoretical Computer Sci.*, 37, pp.183-216, 1985.
- [22] Johnson, L., Weiser, U., Cohen, D., & Davis, A.: "Toward a formal treatment of VLSI arrays", *Proc. 2nd Caltech Conf. on VLSI*, Jan., pp.378-398, 1981.
- [23] Leiserson, C. E. & Saxe, J. B.: "Optimizing synchronous systems", *J. of VLSI and Computer Systems*, Vol.1, No.1, pp.41-67, 1983.
- [24] Melhem, R. G. & Rheinboldt, W. C.: "A mathematical model for the verification of systolic networks", *SIAM J. Comput.* Vol.13, No.3, pp.541-565, 1984.
- [25] Miranker, W. L. & Winkler, A.: "Space-time representations of systolic computational structures", *Computing*, 32, pp.93-114, 1984.
- [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] Ibarra, O. H., Kim, S. M., & Palis, M. A. : "Designing systolic algorithms using sequential machines", *IEEE Trans. on Computers*, Vol. C-35, No.6, pp.531-542.
- [28] Ibarra, O. H., Palis, M. A., & Kim, S. M. : "Some results concerning linear iterative (systolic) arrays", *J. of Parallel and Distributed Computing*, 2, pp.182-218, 1985.
- [29] 阿曾弘具:「シストリックアレーの自動設計法」, 電子情報通信学会論文誌D, Vol. J71-D, No.8, pp.1487-1495, 1988.
- [30] Cappello, P. R. & Steiglitz, K. : "Unifying VLSI array designs with geometric transformations" *Proc. Intern. Conf. on Parallel Processing*, pp.448-457, 1983.
- [31] Hennessy, M. : "Proving systolic systems correct", *ACM Trans. on Programming Languages and Systems*, Vol.8, No.3, pp.344-387, 1986.
- [32] Moldovan, D. I. & 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.
- [33] Guerra, C. & Melhem, R. : "Synthesis of systolic algorithm design", *Parallel Computing*, 12, pp.195-207, 1989.
- [34] Fortes, J. A. B. & Wah, B. W. : "Systematic design approaches for algorithmically specified systolic arrays", Tech. Rep. of Univ. of Illinois, pp.32, 1985.
- [35] Quinton, P. : "Automatic synthesis of systolic arrays from uniform recurrent equations", *Proc. of 11th Annu. Symp. on Computer Architecture*, pp.208-214, 1984.
- [36] 梅尾博司, 浅野哲夫:「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, 第27巻, 11号, pp.1270-1281, 1986.
- [37] Neumann, J. von : *Theory of self-reproducing automata* (edited and completed by Burks, A. W.), Univ. of Illinois Press, 1966.
- [38] Codd, E. F. : *Cellular automata*, Academic Press, 1968.
- [39] Beyer, W. T. : "Recognition of topological invariants by iterative arrays", Ph. D. thesis, Massachusetts Institute of Technology, 1969.
- [40] Smith III, A. R. : "Real-time language recognition by one-dimensional cellular automata", *JCSS*, 6, pp.233-253, 1972.
- [41] Kosaraju, S. R. : "Speed of recognition of context-free languages by array automata", *SIAM J. Comput*, Vol.4, No.3, pp.331-340, 1975.
- [42] Hennie III, F. C. : *Iterative arrays of logical circuits*, M. I. T. Press, 1961.
- [43] Atrubin, A. J. : "A one-dimensional real-time iterative multiplier", *IEEE Trans. on Electronic Computers*, EC-14, pp.394-399, 1965.
- [44] Seiferas, J. I. : "Iterative arrays with direct central control", *Acta Informatica*, 8, pp.177-192, 1977.

表紙のデザインを公募します

このたび、オペレーションズ・リサーチ学会では、「オペレーションズ・リサーチ」誌の表紙デザインを公募することになりました。現在の表紙は、高井英造氏によるもので、1987年より毎年色を変えながら皆様に親しまれてまいりましたが、すでに6年目を迎えました。そこで、このあたりで表紙のデザインを変えて気分を一新したいと思います。ORの未来を切り開くような斬新なイメージの作品を募集します。皆様ふるってご応募ください。

<記>

応募資格：本学会正会員および学生会員、賛助会員の社員

締切：1992年8月31日

発表：本誌紙上

賞金：10万円

大きさ：B5判

その他：①現在の表紙に記載されている文言を入れられるようにして下さい。

②3色刷り（1色はスミ色）で主たる色（2色）を変更できるようにして下さい。

③使用にあたっては、技術上の修正をすることがあります。

④著作権は学会に帰属し、応募作品は返却いたしません。（編集委員会）