Vol. 24 No. 5 情報処理学会論文誌 Sep 1983

シストリック・アレイにより模倣可能なSIMD 並列計算機のあるクラス†

梅尾 博司††

大局的通信網をもったSIMD型並列計算機を局所的通信網のみを仮定したシストリック・アレイ上で模倣可能なことを示す．単純SIMDと呼ばれるSIMDのサブクラスを提案し，このクラスに属する任意の並列計算機M（時間計算量を$T(n)$とする）に対して，Mを$2T(n)+3n+1$ステップで模倣するシストリック・アレイが存在することを示す．単純SIMDは，(1)各プロセッサ(PE)はレジスタ以外のメモリをもたない，(2)一括伝播される命令は1ステップで実行される，(3)各PEは左右両隣接PEとのみ接続されている，の3点で従来のSIMD並列計算機の制限型のひとつと考えられる．しかしながら，これまでに提案された数多くのSIMD・ソーティング・アルゴリズム，SIMD画像処理アルゴリズム等は，単純型であるという理由から，このサブクラスはかなり広く有用なクラスと思われる．

1. まえがき

最近のVLSI設計ならびに製造技術の発展とともに，10^4〜10^5個のオーダの計算要素からなる1チップセルラ計算機の実現も可能となってきた．また，それらの上で動作する並列アルゴリズムに関しても，大いに注目を浴びている．

一般に，並列アルゴリズムの設計は，対象とする並列計算機のアーキテクチャに大きく依存する．これは，従来のノイマン型計算機との大きな相違と考えられる．

Flynnは命令およびデータの流れ(stream)の数に応じて，一般的な計算機を四つのクラスに分類した[1]．そのなかに，SIMD (single instruction stream/multiple data stream)，ならびにMISD (multiple instruction stream/single data stream)と呼ばれる並列計算機のふたつのクラスがある．前者に属するものとしてILLIAC-Ⅳが，後者にはパイプライン計算機，シストリック・アレイ等がある．

本稿では，これまでに開発，蓄積されてきたSIMD型の並列アルゴリズムを，VLSI向きと考えられているシストリック・アレイ上で動作させることを目的とし，シストリック・アレイによるSIMD型並列計算機の模倣法を与える．

SIMDとは，1台の制御装置(control unit, CU)，多数のプロセッサ(processing element, PE)ならびにそれらを結合するネットワーク(interconnection network)から構成される並列計算モデルで，CUから各PEへの大局通信網による命令の一括伝播(broadcast)を特徴とする[2]．

シストリック・アレイとは，プロセッサにおける入出力と内部処理を同時にパイプライン的に行うもので，従来から“I/O and computation imbalance”として知られている並列計算機のbottle-neckを克服する一つの手段と考えられている[3)〜5),8),9)]．

まず2章では，シストリック・アレイおよび単純SIMD並列計算機の説明を与える．単純SIMDとは，(1)各PEはレジスタ以外のメモリをもたない，(2)一括伝播される命令は1ステップ命令である，(3)各PEは左右両隣接PEとのみ接続されている，の3点で，従来のSIMDのサブクラスのひとつと考えられる．3章では，シストリック模倣定理を与える．任意の単純SIMD並列計算機M（時間計算量を$T(n)$とする）に対し，Mを$2T(n)+3n+1$ステップで模倣するシストリック・アレイが存在することを示す．本定理は，命令の一括伝播による大局的通信を局所的な通信で置換え可能なことを示している．4章は，結論および残された問題等について検討する．

2. 諸定義

2.1 単純SIMD並列計算機

単純SIMD並列計算機Mとは，1個の制御装置(control unit, CUと略す)，$N(=2^m, m$は非負整数$)$個のプロセッサ(processing element, PE_i $(0\leqq i\leqq N-$

† A Class of SIMD Parallel Computers Simulated by Systolic Arrays by HIROSHI UMEO (Department of Applied Electronic 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

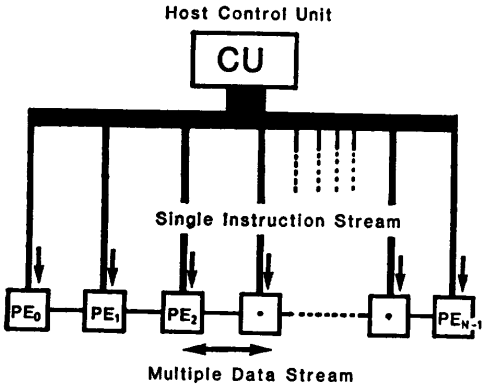

図 1 SIMD 並列計算機の概念図
Fig. 1 Organization of simple SIMD machine.

1) で示す) ならびに各隣接 PE を結合するプロセッサ・ネットワークからなる並列計算モデルである[1),2)]. M の概念図を**図 1** に示す.

一つの命令の流れ (single instruction stream), 複数個のデータの流れ (multiple data stream) を意味する SIMD は, 次の事実に由来する. すなわち,

(1) CU はすべての PE に同一時刻に同一命令を一括伝播 (broadcast) する.

(2) プロセッサ・ネットワークを通じて各 PE 間でデータ交換をする.

次に M の構成要素を説明する.

プロセッサ (PE)

(1) 各 PE は, いくつかのデータ・レジスタ, $R_1, R_2, \cdots, R_k$ および 1 個のアドレス・レジスタ, R_a からなる (**図 2** 参照).

(2) PE_i の R_a には, アドレス i が m 桁の 2 進表現で貯えられる.

(3) 各 PE は活性/不活性のいずれかの状態をとる.

(4) CU により一括伝播される命令は, 活性状態 PE でのみ実行される.

(5) PE の活性/不活性状態は, 命令に付加されるマスク (後述) により決定される.

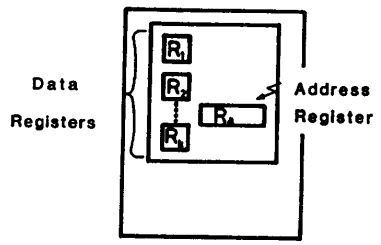

図 2 プロセッサ PE
Fig. 2 Processing element.

命令 (Instruction)

(1) CU はユーザ作成のプログラムを貯える.

(2) CU が一括伝播する命令は, 各 PE_i が自身を含めた左右両隣接 PE, すなわち, PE_{i-1}, PE_i, および PE_{i+1} の情報から PE_i のデータ・レジスタを 1 ステップで変更可能なもの (1 ステップ命令と呼ぶ) とする.

(3) すべての命令はマスクをもつ.

マスク (Mask)

(1) アドレス・マスクおよびデータ・コンディション・マスクの 2 種類のマスクを考える.

(2) アドレス・マスク $[x]$ は $I[x]$ の形で命令 I に付与される. ただし, x は $\{0, 1, X\}$ 上の長さ m のストリングである. m 個の各記号は, アドレス・レジスタのビット位置と対応し, x と R_a の内容が一致する PE のみが活性状態となる. ここで 0 は 0 に, 1 は 1 に, X は 0 あるいは 1 に一致し, m 桁すべてが一致したときに限り, x と R_a の内容が一致したと考える. たとえば $R_1 \leftarrow R_2 + R_3\,[X^{m-1}1]$ は奇数番地をもつすべての PE に対して, R_2 と R_3 の和を R_1 に貯えさせる命令を意味する.

(3) データ・コンディション・マスクは, "where 〈データ条件〉 do…elsewhere do…" の形で表現される. たとえば, "where $R_2 \geqq R_3$ do $R_1 \leftarrow R_2$ elsewhere do $R_1 \leftarrow R_3$" は, すべての PE において R_1 が R_2 と R_3 の内容のうち大きいほうを貯える命令を意味する.

I/O および時間計算量

(1) 初期データ $a_0, a_1, \cdots, a_{n-1}$ は a_i/PE_i として前もって各 PE_i に与えられている.

(2) 時間計算量 $T(n)$ は, CU が一括伝播する命令の総数により定義される.

(3) 出力は, $T(n)$ ステップ後の各 PE に分散的に残される.

本稿で定義した単純 SIMD は,

(1) 各PEはメモリをもたない,

(2) 各 PE は 1 ステップで両隣接 PE とのみ交信可能,

(3) 一括伝播される命令は 1 ステップ命令である,

という点で, 従来から研究されてきた SIMD のサブクラスの一つと考えられる. しかしながら, 画像処

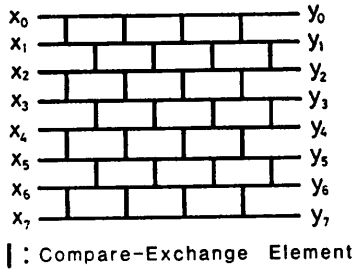

図 3 奇偶ソーティング・ネットワーク
Fig. 3 Odd-even transposition sorting network.

理[10]，ソーティング[7),11),12)]等の分野でこれまでに提案されてきた SIMD アルゴリズムは，本稿で定義した単純型であり，単純 SIMD のクラスはかなり広いものと思われる．

図 3 は n 個のデータを n ステップでソートする奇偶ソーティング・ネットワークである．比較-交換要素を命令と考えれば，**図 4** に示す単純 SIMD 型ソート・プログラムが容易に得られる．図 4 中，R(i) は PE_i のレジスタ R の内容を意味する．

2.2 シストリック・アレイ

シストリック・アーキテクチャとは，アレイ・プロセッサにおけるデータの入出力および内部処理をパイプライン的に行うもので，Kung らによって VLSI 向き行列計算専用プロセッサとして提案された[3)]．以来数多くのシストリック・アレイが提案されている[3)~6),8),9)]．

本稿では，**図 5** に示すような非常に単純なモデルを仮定する．**シストリック・アレイ**，A とは，1 個のバッファ（B で示す）および $n+1$ 個のシストリック・セル（以下ではたんにセルと呼び，$C_i, i=0,1,2,\cdots,n$ で示す）からなる．シストリック・アレイは，通常ノイマン型計算機のバス・ライン等に接続して使用される．B は内部にレジスタをもち，その内容はホスト計算機あるいは C_0 により変更される．各 C_i ($0 \leqq i \leqq n-1$) は C_{i-1} ($i=0$ のとき B) および C_{i+1} とのみ直接通信可能である．C_n は境界セルとして働く．各 C_i は，B を経由してデータが入ってくるまで静止状態でいる．

すでにデータを受け取ったセルあるいは，近傍セルにデータが揃っているすべてのセルは同期して動作をする．次の三つの項目についてシストリック・アレイの動作を記述する．

（1） 入力データ $a_0, a_1, \cdots, a_{n-1}$ の入力速度
（2） セルの遷移関数
（3） 出力を取り出すタイミングおよびその速度

シストリック・アレイの時間計算量は，入力，内部処理，および出力に要する時間の総和により定義する．

```
Program {Odd-Even Transposition Sorter for n( =2^m) Data}
    begin
        for i:=1 to n/2 do
            begin
                Compare-Exchange data between PE_i
                and PE_{i+1} such that R(i) ≤ R(i+1),
                where i = [x^{m-1}0].

                Compare-Exchange data between PE_i
                and PE_{i+1} such that R(i) ≤ R(i+1),
                where i = [x^{m-1}1].
            end;
end.
```

図 4 奇偶ソートを実行する単純 SIMD プログラム
Fig. 4 Odd-even transposition sorting program for simple SIMD machine.

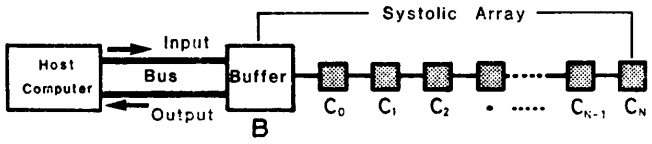

図 5 シストリック・アレイの概念図
Fig. 5 Systolic array.

3. 本 論

次のシストリック・アレイ上での並列アドレス設定法は，以下でのシストリック・模倣定理の証明に使用される．

シストリック・アレイ A 上でのアドレス設定問題とは，A の各セルの C_i ($0 \leqq i \leqq n-1$，ただし $n=2^m$) のアドレス i を計算し，その 2 進表現を C_i のアドレス・レジスタに貯えるセルの遷移関数（n に依存しない）を設計する問題である．

セルは有限個の有限レジスタおよび 1 個のアドレス・レジスタ R_a からなる．各セルの R_a は正確に m ($=\log_2 n$) 桁の駒に区切られているものとする．各駒を下位の桁から順に $d_0, d_1, \cdots, d_{m-2}, d_{m-1}$ で表す．また，C_i の時刻 t における d_j の内容を $d_j^t(i)$ で表す．任意の j ($0 \leqq j \leqq m-1$)，i ($0 \leqq i \leqq n-1$) に対し，$d_j^0(i)=$"v"（ブラン

ク記号）とする．

〔並列アドレス設定アルゴリズム〕

各セルとも最下位の桁から順次セットされる．C_0 は1個/1ステップの割合で，合計m個の波（w_j で示す．$j=0,1,2,\cdots,m-1$）を生成する．各波 w_j は，1セル/1ステップの速度で右方向に進みながら，各セルの R_a の d_j を0または1にセットする．決定の仕方は次の規則に従う．

まず，w_0 は C_0 から順に，0と1を交替的に各セルの d_0 の値とする．すなわち，

$$d_0^{i+1}(i)=\begin{cases}0, & i \text{ が偶数のとき}\\ 1, & \text{上記以外}\end{cases} \quad (0\leqq i\leqq n-1)$$

w_j は w_{j-1} の後を追いかけすでに設定済みである d_{j-1} の値を参照しながら，最初の $d_{j-1}=0$ である一連のセルおよび次の $d_{j-1}=1$ である一連のセルに対し，d_j の値を0に，また次の $d_{j-1}=0$ である一連のセルおよび次の $d_{j-1}=1$ である一連のセルに対しては，d_j の値を1にセットする．以上の操作を最後のセルまで交替的に繰り返す．正確には，

C_0 に対して，

$d_j^{j+1}(0)=0 \quad (1\leqq j\leqq m-1)$

C_i $(1\leqq i\leqq n-1)$ に対しては，

$$d_j^{j+1}(i)=\begin{cases}\overline{d_j^j(i-1)}\\ (d_j^j(i-1)=1 \text{ かつ } d_{j-1}^j(i)=0 \text{ のとき})\\ d_j^j(i-1) \quad (\text{上記以外})\end{cases}$$

と決定される．ただし，$1\leqq j\leqq m-1$，$t\geqq 2$．記号 "$\bar{x}$" は $x(\in\{0,1\})$ の否定，すなわち，$\bar{0}=1, \bar{1}=0$ を意味する．

w_j は最初の 2^j 個のセルの d_j を0に，次の 2^j 個のセルの d_j を1にセットし，以下の右端まで交替的にこれを繰り返す．

R_a は正確に $\log_2 n$ 桁に区切られているため，R_a のすべての桁がいっぱいになったときに限り C_0 は波の生成を中止する．

C_i のアドレス設定は $i-1+\log_2 n$ ステップ時に完了する．したがって $n-1+\log_2 n$ ステップで Aのアドレス設定は終了する．**図6**は，$n=8$ の場合のアドレス設定法を例示したものである．

シストリック・アレイによる単純 SIMD 並列計算機の模倣定理を与える．

〔**定理 1**〕（**シストリック・模倣定理**）

時間計算量 $T(n)$ の任意の単純 SIMD 計算機 M に対し，M を $2T(n)+3n+1$ ステップで模倣するシストリック・アレイ，A が存在する．

（証明） M は $n(n=2^m$ とする）個の PE_i $(0\leqq i\leqq n-1)$ からなるものと仮定．1個のデータ・レジスタ，およびアドレス・レジスタからなる PE を考える．k 個のデータ・レジスタをもつ PE については後述する．各 PE_i にあらかじめ与えられているデータを $a_i(0\leqq i\leqq n-1)$，時刻 t に一括伝播される命令を I_t $(1\leqq t\leqq T(n))$ で表す．

Mを模倣するシストリック・アレイ，Aは，1個のバッファBおよび $n+1$ 個のセル $C_i(0\leqq i\leqq n)$ からなる．Bは入力レジスタ B_{in} および出力レジスタ B_{out} をもち，それらの内容はそれぞれホスト計算機および C_0 により変更される．各 C_i は，m ビットからなるアドレス・レジスタ R_a の他に5個のレジスタ R_0, R_1, R_2 R_3, R_t をもつ．以下の説明中，次の記法を使用する．各 C_i の $R_j(1\leqq j\leqq 3)$ をまとめて第 j 層と呼び，L_j で表す．時刻 t におけるセル C_i の R_i の内容を $R_j^t(i)$ で表す．B_{in}^t, B_{out}^t, $R_a^t(i)$ に関しても同様である．レジスタRが空であること，空でないこと，Rを空にすることをそれぞれ，"$R=\phi$", "$R\neq\phi$", "$R\leftarrow\phi$" で表す．さらに，$R_i\leftarrow R_j$ は R_j の内容を R_i へコピーすることを意味する．

Aの1個のセルはMの1個の PE に対応し，C_i $(0\leqq i\leqq n-1)$ は PE_i を模倣する．C_n はAの境界セルとして働く．レジスタ $R_i(0\leqq i\leqq 3)$ は主として以下の目的で使用される．

R_0：アドレス設定用補助レジスタ．

R_1：初期データおよび命令を右方向に運搬する際

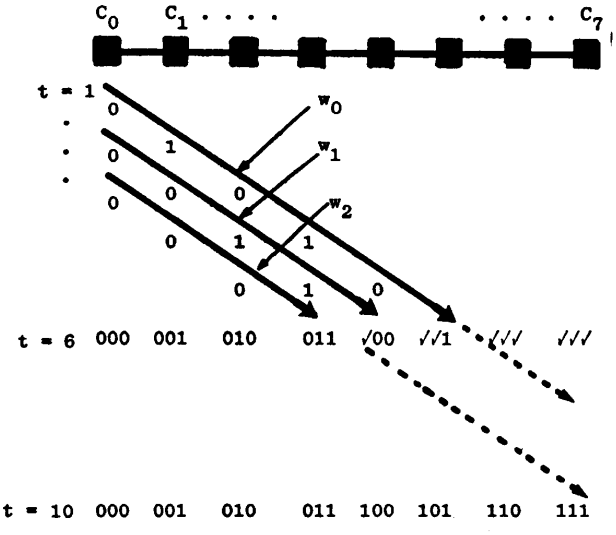

図 6 並列アドレス設定法 ($n=8$)

Fig. 6 Parallel binary address setting scheme.

のパイプラインとして使用される．

R_2, R_3：PE のデータ・レジスタの内容を一時的に記憶する．さらに，R_3 は最終結果を出力する際のパイプラインとしても使用される．

ホスト計算機はAへの入力を，

$$\overbrace{\blacksquare\llcorner, I_{T(n)}, \llcorner, I_{T(n)-1}, \llcorner, \cdots, \llcorner, I_2, \llcorner, I_1,}^{2T(n)+1}$$
$$\overbrace{a_{n-1}, a_{n-2}, \cdots, a_2, a_1, a_0}^{n} \qquad (1)$$

の形で準備し，(1)の右端から1個/1ステップの割合で B_{in} を経由して入力する．ただし，$B_{in}^0=a_0$ とする．"⌴" は空白を意味する．初期データは1個/1ステップ，命令は1個/2ステップの割合で入力される．"■" は命令の終了を意味する記号である．

各セルは，初期データ a_i，命令 I_i，■，ならびに⌴の区別が可能で，さらにMが PE に一括伝播するすべての命令を解釈し実行できるものとする．各セルにおける模倣は，三つの相 (phase) に分けられる．すなわち，初期データの入力およびアドレス設定をする相，命令を解釈し実行する相，ならびに最終結果を出力する相である．R_s(状態レジスタ) は，これらの相を区別する目的で使用される．

アレイ全体では，データおよび命令の入力，処理ならびに結果の出力過程をオーバラップさせながらパイプライン的に実行する(図7参照)．

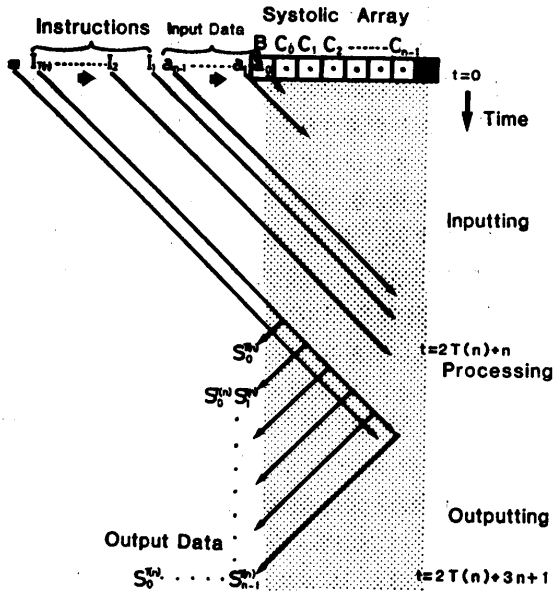

図 7 単純 SIMD 並列計算機を模倣するシストリック・アレイの時間-空間図式
Fig. 7 Time-space diagram for systolic simulation of simple SIMD machine.

次に各相におけるレジスタの動作を説明する．ただし，$R_j^0(i)=\phi, R_s^0(i)=\phi\ (0\leqq i\leqq n-1, 0\leqq j\leqq 3)$ とする．

相の決定 まず R_1 は，

$$\begin{cases} R_1^{t+1}(0)\leftarrow B_{in}^t \\ R_1^{t+1}(i)\leftarrow R_1^t(i-1) \quad (1\leqq i\leqq n-1) \end{cases}$$

なる動作をして，B_{in} を経由して与えられる入力すべてを1セル/1ステップの速度で L_1 上を右方向に伝える．■記号が C_{n-1} に伝えられると，次の時刻に C_{n-1} は←記号を R_1 上に生成する．"←" は各セルに出力相の終了を知らせる記号で，以後左方向に L_1 上を1セル/1ステップの速度で伝えられる．

$C_i(0\leqq i\leqq n-1)$ は，L_1 上を運ばれてくる記号により相を決定し，$R_s(i)$ をそれらに応じた状態にセットする．C_i における各相は，次の $\alpha+1, \beta+1$，ならびに $\gamma+1$ ステップからなる区間である．

入力相：$R_1^t(i)=a_0$ かつ $R_1^{t+\alpha}(i)=a_{n-1}$ を満足する $(t, t+1, \cdots, t+\alpha)$ の区間

実行相：$R_1^t(i)=I_1$ かつ $R_1^{t+\beta}(i)=$"$I_{T(n)}$ の次の空白記号" を満足する $(t, t+1, \cdots, t+\beta)$ の区間

出力相：$R_1^t(i)=$■ かつ $R_1^{t+\gamma}(i)=$"←" を満足する $(t, t+1, \cdots, t+\gamma)$ の区間

入力相 セルが入力相にある場合は，次の規則に従って初期データを R_1 より R_3 へ取り込む．

C_0：$R_3^2(0)\leftarrow R_1^1(0)$

$C_i(1\leqq i\leqq n-1)$：**if** (C_{i-1} が時刻 $t-1$ に，$R_3^{t-1}(i-1)\leftarrow R_1^{t-2}(i-1)$ なる動作 (a_{i-1} の取込み) をした)

then $R_3^{t+1}(i)\leftarrow R_1^t(i)$

else R_3 へのデータの取込みはしない．

これにより，$C_i(0\leqq i\leqq n-1)$ は $t=2i+2$ 時に a_i を得る．すなわち，$R_3^{2i+2}(i)=a_i(0\leqq i\leqq n-1)$．すでに R_3 にデータを得たセルは，実行相になるまで何もしない．

初期データの入力と同時に，左端のセルから順に R_a のアドレス設定をさきに示した方法により開始する (R_0 を使用)．これにより，C_i のアドレスは $t=i-1+\log_2 n$ 時に設定され，最初の命令が C_i に到着する $t=n+i+1$ 時(後述)までに終了している ($\because$ すべての n に対し，$n+i+1>i-1+\log_2 n$)．

実行相 $R_1^{n+i+1}(i)=I_1(0\leqq i\leqq n-1)$ となり，実行相が開始される．いま $R_1^t(i)=I$ とする．C_i では次の動作をする．

if (C_i の R_a の内容と I のアドレス・マスクが一致する)

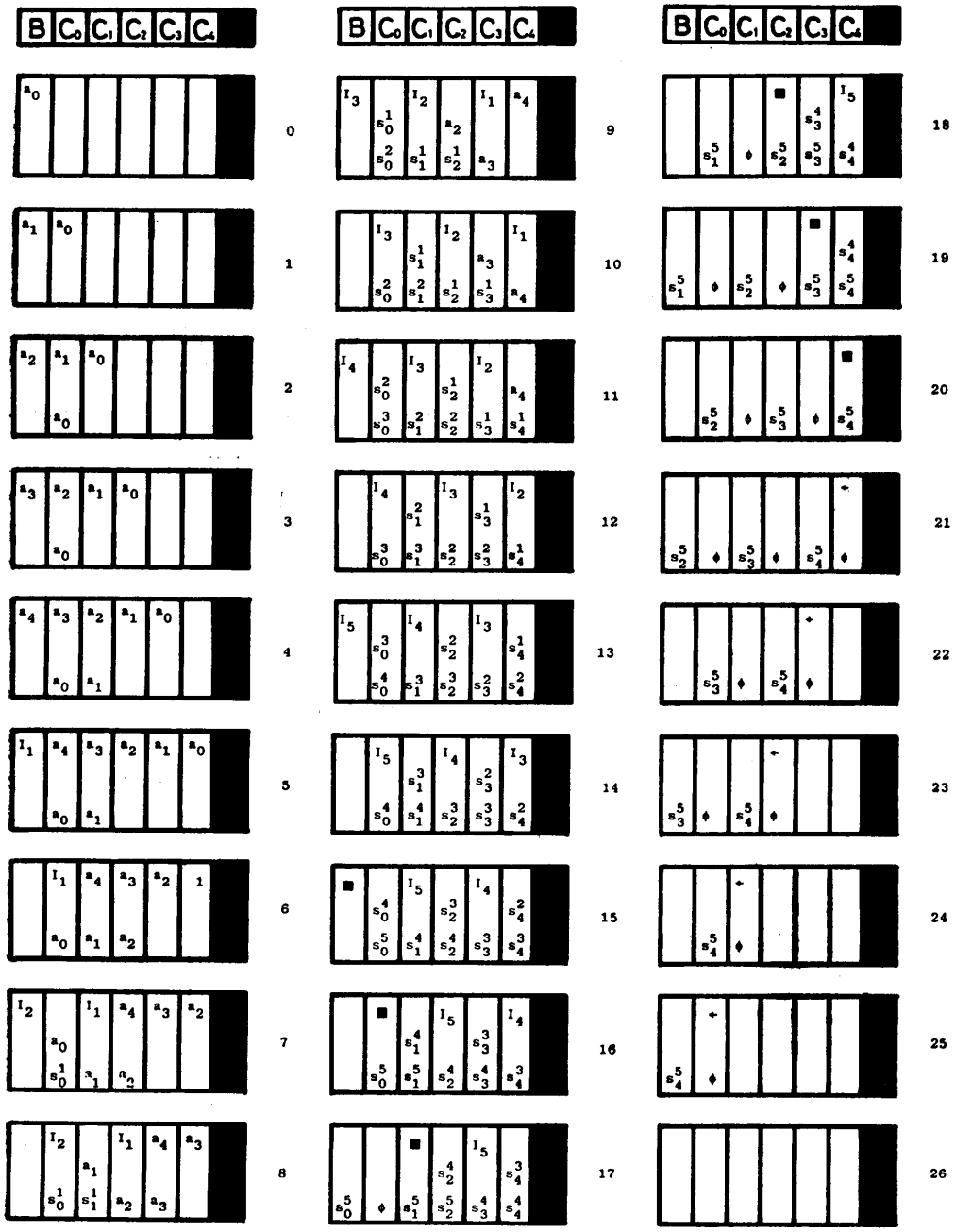

図 8 単純 SIMD 並列計算機を模倣するシストリック・アレイの計算状況
Fig. 8 Configurations of the systolic array which simulates a simple SIMD machine in the case $T(n)=n$ and $n=5$.

then $\begin{pmatrix} R_2^{t+1}(i)\leftarrow R_3^t(i), \\ R_3^{t+1}(i)\leftarrow I[R_2^t(i-1), R_3^t(i), R_3^t(i+1)]\star \end{pmatrix}$

else $\begin{pmatrix} R_2^{t+1}(i)\leftarrow R_3^t(i) \\ R_3^{t+1}(i)\leftarrow R_3^t(i) \end{pmatrix}$

* C_0 に対応している PE_0 は左隣接 PE をもたない．したがって，C_0 は，$R_3^{t+1}(0)\leftarrow I[R_3^t(0), R_3^t(1)]$ を実行する．C_{n-1} に対しても同じ理由で，$R_3^{t+1}(n-1)\leftarrow I[R_3^t(n-2), R_3^t(n-1)]$ となる．

ここで，$I[R_3^t(i-1), R_3^t(i), R_3^t(i+1)]$ は，$R_2^t(i-1)$, $R_3^t(i)$, $R_3^t(i+1)$ および I により定まるデータを意味する．

さらに，$R_i^t(i)=\sqcup$ なる C_i では，

$R_2^{t+1}(i)\leftarrow\phi$, $R_3^{t+1}(i)\leftarrow R_3^t(i)$

を実行する．

出力相 $R_1^i(i)=$■, ただし $t=n+2T(n)+i+1$, となり, C_i において出力相が開始される. 出力相では各セルは次の動作をする. まず C_0 では, 次の2ステップからなる.

$B_{out}^{t+1} \leftarrow R_3^t(0), R_3^{t+1}(0) \leftarrow \phi$ (時刻 $t+1$)

$R_3^{t+2} \leftarrow R_3^{t+1}(1)$ (時刻 $t+2$)

の動作を繰り返す.

$C_i(1 \leqq i \leqq n-2)$ では,

$R_3^{t+1}(i) \leftarrow \phi$ (時刻 $t+1$)

$R_3^{t+2}(i) \leftarrow R_3^{t+1}(i+1)$ (時刻 $t+2$)

の動作を,

C_{n-1} は, $R_1^{t+1}(n-1) \leftarrow$ "$\leftarrow$", $R_3^{t+1}(n-1) \leftarrow \phi$ を実行する. B_{out} には2ステップに1個の割合で出力が得られ, ただちにホスト計算機に取り込まれる.

(1)の入力に $2T(n)+n$ ステップ, さらに結果の出力に $2n+1$ ステップ, 全体でMの模倣に $2T(n)+3n+1$ ステップを要す.

図8は, $T(n)=n, n=5$ の場合における単純 SIMD 計算機のシストリック・アレイによる模倣の様子を示したものである. 図の中で s_i^t は時刻 t における PE_i のデータ・レジスタの内容を示す.

各 PE が k 個のデータ・レジスタをもつ場合は, R_2, R_3 に対応するレジスタを $2k$ 個用意することにより模倣可能である. ■

次に本手法の実用上の問題点を考察する. 定理1の証明において, Aの境界セルは最初から固定していた. したがって, データ数とシストリック・セル数が一致する場合にのみ有効であった. セルの遷移関数を少し変更することにより, $m(<n)$ 個のデータが与えられた場合, C_m を境界セルとして働かせることができる. これによりシストリック・アレイのサイズを変えることなく, 少数データを扱うことが可能である.

さらに, わずかの変更で定理1の証明に使用した"$\leftarrow$" 信号 (出力相の終了を意味する) に, 各セルのすべてのレジスタをリセットする機能を付与することができる. これにより, 同一のシストリック・アレイを再使用することができ, 多数のM上のタスクの処理が可能となる. 次の定理を得る.

〔定理 2〕 命令集合が同一の l 個の単純 SIMD 並列計算機を $M_k(k=1\sim l)$, それぞれの時間計算量を $T_k(n)$ とする. $M_k(k=1\sim l)$ を $2\sum_{k=1}^{l} T_k(n)+3ln+l$ ステップで模倣するシストリック・アレイAが存在する.

(略証) 定理1の証明とまったく同じ形式でホスト計算機から M_k の初期データおよび命令列を入力する. M_k の出力に要する時間, すなわち $2n$ ステップ経過後, 再び M_{k+1} の初期データおよび命令列を入力する. 以後これを繰り返す.

定理1の証明では, R_3 をデータの一時記憶レジスタならびに最終結果の出力用パイプラインとして併用していた. 出力用のパイプラインを新たに設け, さらにレジスタのリセット機能を■記号にもたせることにより, 定理2の l 個のタスクをオーバラップさせて処理できる. 次の定理を得る.

〔定理 3〕 命令集合が同一の l 個の単純 SIMD 並列計算機 M_k (時間計算量を $T_k(n)$ とする, $k=1\sim l$) に対し, $M_k(k=1\sim l)$ を $2\sum_{k=1}^{l} T_k(n)+ln+l+2n$ ステップで模倣するシストリック・アレイ A が存在する.

(略証) ホスト計算機は, M_k と $M_{k+1}(k=1\sim l-1)$ に対応したデータおよび命令列を入力する際, $2n$ ステップ待機する必要がなくなる. したがって $(l-1)2n$ ステップだけ高速になる.

4. むすび

単純 SIMD と呼ばれる SIMD 並列計算機のサブクラスを定義した. このサブクラスに属し, 時間計算量 $T(n)$ をもつ任意の並列計算機Mに対し, Mを $2T(n)+3n+1$ ステップで模倣するシストリック・アレイが存在することを示した.

単純 SIMD は, メモリ部をもたない, 結合が局所的, 一括伝播命令は1ステップで終了する等の性質を特徴とし, 従来の SIMD の制限型のひとつと考えられる. しかしながら, これまでに提案されてきた数多くの SIMD・ソーティング・アルゴリズムおよび SIMD 画像処理アルゴリズム等は単純型であるか, あるいは容易にその型に変換しうる. このような理由から, このサブクラスはかなり広くかつ有用なものと思われ, 今後の研究が期待される.

謝辞 論文の改善に有益なご意見をいただいた査読者に謝意を表す. 本研究の一部は文部省科研費 (No. 58780050) によった.

参考文献

1) Flynn, M. J.: Some Computer Organization and Their Effectiveness, *IEEE Trans. Comput.*, Vol. C-21, No. 9, pp. 948-960 (1972).
2) Siegel, H. J.: A Model of SIMD Machines and a Comparison of Various Interconnection Networks, *IEEE Trans. Comput.*, Vol. C-28, No. 12, pp. 907-917 (1979).

3) Kung, H. T. and Leiserson, C. E.: Systolic Arrays (for VLSI), *Proc. Symp. Sparse Matrix Computations and Their Applications*, pp. 256-282 (1978).
4) Kung, H. T.: Let's Design Algorithms for VLSI Systems, *Proc. Conf. Very Large Scale Integration, California Institute of Technology*, pp. 65-90 (1979).
5) Kung, H. T., Sproull, B. and Steele, G. (Eds.): *VLSI Systems and Computations*, p. 415, Springer-Verlag, New York (1981).
6) Leiserson, C. E. and Saxe, J. B.,: Optimizing Synchronous Systems, *22nd, Annual Symposium on Foundations of Computer Science*, IEEE, October, pp. 23-36 (1981).
7) Knuth, D. E.: The Art of Computer Programming, Vol. 3, p. 723, *Sorting and Searching*, Addison-Wesley, Reading (1973).
8) 梅尾, 菅田：並列アルゴリズムの SIMD 型よりシストリック型への変換法, 電子通信学会技術研究報告 (1982. 7).
9) 梅尾：同期型並列アルゴリズムの MIMD, SIMD 型よりシストリック型への変換法について, 夏のLA シンポジウム(於京都大) (1982. 7).
10) Siegel, L. J.: Image Processing on a Partitionable SIMD Machine, in B. J. B. Duff and S. Levialdi, ed. *Languages and Architecture for Image Processing*, Academic Press, New York (1981).
11) Thompson, C. D. and Kung, H. T.: Sorting on a Mesh-Connected Parallel Computers, *CACM*, Vol. 20, No. 4, pp. 263-271 (1977).
12) Baudet, G. and Stevenson, D.: Optimal Sorting Algorithms for Parallel Computers, *IEEE Trans. Comput.*, Vol. C-27, No. 1, pp. 84-87 (1978).

(昭和57年10月18日受付)
(昭和58年 3 月11日採録)