ソフトウェア基礎論　25－13
（1988.　5. 27）

ループプログラムに関する システリックアレー実現可能条件

Conditions for Loop Programs to be Implemented in Systolic Arrays

阿曽弘具
ASO, Hirotomo

（東北大学工学部情報工学科）
Faculty of Engineering, Tohoku University

あらまし　情報処理の並列化を実現する１つのアーキテクチャとしてシストリックアレーが知られている．今後，VLSI化されたある程度汎用的な使用に耐えるシストリックアレーが出現すると思われるが，それを効果的に利用するためには，システマティックなアルゴリズム設計法が望まれる．このために一般的な設計法がいくつか研究されてきており，それらをより一般化し適用範囲を広くする努力がなされている．本論文では，その設計法において残されていた発見的思考を必要とする部分について，詳細な検討を加え，よりシステマティックにするための基礎を与えるものである．特に，結線構造を与える近傍写像や時間調整写像について，その形を明らかにし，その設計法を明らかにした．また，最簡機能条件に関して，それが満たされるかどうかの判定ができるような形で必要条件を明らかにした．

Abstract: Systolic arrays are known as a highly parallel processing device architecture. Since the development of VLSI technology gives us programmable systolic arrays, systematic and automatic design method to use the array become significant and necessary matters. As one of the methods, transformation of a given program to a systolic array are now developed and improved. In this paper, to reduce the heuristic thinking in the design, some basic things are analyzed. Particularly, the necessary conditions of neighborhood mapping and time-delay elements are revealed. And also necessary conditions for the simplest implementation of cells are studyed.

1．はじめに

シストリックアレーは，演算の並列性・規則性，通信の局所性，構造の一様性・拡張性に優れ，ＶＬＳＩとして実現するのに適しており[1]，ハードウェア製造技術の発展によりその汎用性をもった具体化が現実のものとなっている[12]など．そのような汎用なシストリックアレーの利用技術としてシストリックアルゴリズムの実現がある程度システマティックにできることが望ましい．すなわち，解くべき問題の仕様を与えることで自動的にシストリックアルゴリズムのプログラムを生成することが望まれる．

この課題を解決するための１つの方法として，マッピングアルゴリズムとして知られている方法がある．漸化式を対象とするもの（若林[4]，Quinton[5]，Li，Wah[6]）と多重ループプログラムを対象とするもの（Moldovan[2,3,7]，堀池[9]，三浦[8]，飯国[11]，阿曽[13]）とがあり，その方法の適用対象を広げ，より組織的に設計する方法が開発されつつある．

本論文では，その設計法において残されていた発見的思考を必要とする部分について，詳細な検討を加え，よりシステマティックにするための基礎を与える．特に，結線構造を与える近傍写像や時間調整写像について，その形を明示し，その設計法を明らかにする．また，最簡機能条件に関して，それが満たされるかどうかの判定ができるような形で必要条件を明らかにする．

本論文の構成は次のとおりである．まず，第２章ではシストリックアレーの定式化[10,13]を説明し，ループプログラムからシストリックアレーへの変換法[13]の概要を説明する．第３章では，プロセッサセル間の結線構造を与える近傍写像およびセルの同期を与える時間調整写像について，その構成方法を詳細に分析している．また，セル機能が最簡であるために必要な条

件をいくつか明らかにしている．第４章では，３章で示したことが，かなり具体的であって，これ以上の詳細化を一般的には論ずることができそうにないことを示す例を説明している．それらは，更に，今後の解決の方向を示唆するようにも見える．

2．ループプログラムとシストリックアレーの基礎

2.1 シストリックアレー

整数の集合をＺと記す．集合 D_1から集合D_2 への写像の集合を $[D_1 \to D_2]$ と表す．

【定義１】シストリックアレー（セル構造情報処理系）Ａとは，次のセル機能Ｆ，セル空間Ｇ，セル間結線Ｗ，配置領域Ｒの４項組＜Ｆ，Ｇ，Ｗ，Ｒ＞である．

(1)　$F = \langle P_{in}, P_{out}, D, f \rangle$．$P_{in}$，$P_{out}$ は入力・出力ポート集合で，$|P_{in}| = m$，$|P_{out}| = r$ とおく．D はデータの集合で，$f = \langle f_1, f_2, \ldots, f_a, \ldots, f_r \rangle$ は $f_a : [P_{in} \to D] \to D$なる写像の組である．$[P_{in} \to D] = D^m$ は入力ポート上のデータの集合を表し，その元dを $\langle d(i) \mid i \in P_{in} \rangle$ とも表す．f_a はその入力データに対して出力ポートaの値を決める関数で，セル関数と呼ぶ．

(2)　Ｇはある可換群$G = \langle G ; +, 0 \rangle$ で，セル機能を配置する空間の座標の集合である（K次元格子空間を想定している）．

(3)　Ｗは，ポート間結線写像 $u: P_{in} \to P_{out}$ と近傍写像$N: P_{in} \to G$ との２項組 $\langle u, N \rangle$ である．各セルαの入力ポート i をセル $\alpha - N(i)$ の出力ポート $u(i)$ に接続することを意味する．

(4)　$R \subseteq G$は，実際にセル機能をおくセル座標の集合であり，アレー領域とも呼ぶ．

記憶を明示的には定式化していないが，$N(i)=0$ なる入力ポートi，出力ポート$u(i)$によって表現できる．

各セルの入出力ポートでその接続先がアレーの外に出てしまうものをアレーに対する入出力ポートとみなす．

シストリックアレーの各時点での動作はデータ転送，計算処理の２フェーズからなる．時刻ｔでの各セルの入出力ポート上のデータの組をセル入・出力様相といい，$y^t \in [P_{in} \times G \to D]$，$x^t \in [P_{out} \times G \to D]$ と定める．データ転送フェーズは，

$$y^t(i, \alpha) = x^{t-1}(u(i), \alpha - N(i))$$

であり，計算処理フェーズは，

$$x^t(a, \alpha) = f_a(\langle y^t(i, \alpha) \mid i \in P_{in} \rangle)$$

である．但し，$\alpha - N(i) \notin R$のとき，$x^{t-1}(u(i), \alpha - N(i))$は外部からの入力データである．

従って，シストリックアレーの動作は次のループプログラムで表される．

```
for t = 1 to M
  for α ∈ R
    for a ∈ Pout
x^t(a, α) = fa( <x^{t-1}(u(i), α-N(i)) | i∈Pin> )
```

シストリックアルゴリズムが処理対象とする情報は空間的広がりをもつものである．この空間をG_D と記し，データ配置空間と呼ぶ．処理対象データはG_D からD への部分写像として定式化でき，これは配列と呼ばれるデータ構造である．

詳細は割愛するが，シストリックアルゴリズムは，上に定式化したシストリックアレーと，G_D 上の各点の値をどのアレー入力ポートにいつ入力するか，どの出力ポートのいつの時点での値をG_D 上のどの点の値とみなすかという入出力データのタイミングとで定められている[10]．

2.2 ループプログラム

対象問題の記述としてその本体がIF文と割当文とからなる多重ループプログラムを考える．また，IF文の条件判定や割当文の右辺の評価において副作用は生じないものとする．このとき，IF文に対して等価な if-then-else 関数を用いることにより，本体を，左辺に現れる配列変数（配列要素指定）には同じものがない割当文だけにすることができる．さらに，参照変数が入力として与えられ，ループ本体では値がセットされない場合には，ループの計算の意味を変えることなく次のように値をセットするようにする．すなわち，参照変数が $b[k_p{}^q(\mathbf{j})]$（$k_p{}^q(\mathbf{j})$は添え字式である）のとき，$b[k_p{}^q(\mathbf{j})] = b[k_p{}^q(\mathbf{j})]$という右辺と左辺が同一の項である割当文を付加する．このループプログラムの等価変換を流れ化変換と呼ぶ（シストリックアレー上で入力データをパイプライン的に流すことに対応する）．この結果はつぎのように表現できる．

```
  for j1 = b1 to e1
    ........
   for jn = bn to en
    for p = 1 to r
ap[kp(j1,…,jn)]
      = fp( <ap^q[kp^q(j1,…,jn)] | q∈Occur> )
```

但し，p は割当文の番号で，その集合を$ID = \{1, \cdots, r\}$とおく．このp に関するループは本体を略記したもので，p 番目の割当文で値が更新される配列名がa_p，その要素を指定する添え字式が$k_p(j_1, \cdots, j_n)$，その更新の計算式が f_pで，その q 番目の入力となる配列名が$a_p{}^q$，その添え字式が $k_p{}^q(j_1, \cdots, j_n)$である．$f_p$は，$(j_1, \cdots, j_n)$に直接依存してもよい．Occur は入力の出現番号の集合である．

ループインデックスの集合を，

$$Index = \{(j_1, \cdots, j_n) \mid b_1 \le j_1 \le e_1, \cdots, b_n \le j_n \le e_n\}$$

とおき，その元を $\mathbf{j}$ と記す．各b_t，e_tは$j_1, j_2, \cdots, j_{t-1}$に依存していてよい．Index は実行順序を反映する辞書式順序($>$)をもっており，それは Index×ID上に拡張されているものとする．添え字式k_p，$k_p{}^q$はIndexからデータ配置空間G_Dへの写像であるが，Z^n($\supseteq$Index)からの写像に自然に拡張されているとみなす．更に，

$Arg = ID \times Occur$，配列名の集合を Sort とおき，写像 $s: ID \to Sort$，$arg: Arg \to Sort$ をそれぞれ，$s(p) = a_p$，$arg(p,q) = a_p^q$ と定める．

ループ本体の各割当文で参照されている配列変数は，入力としてあらかじめ与えられているか，あるいは，どこかのループインデックスで計算されているはずである．その対応を示すため次の関数を導入する．

【定義２】次の関数 $B: Index \times Arg \to Z^n \times ID$ を，生成時点関数という．$B(\mathbf{j}, \langle p,q \rangle)$は，$arg(p,q) = s(p')$，$k_p^q(\mathbf{j}) = k_{p'}(\mathbf{j}')$，$(\mathbf{j}, p) > (\mathbf{j}', p')$ を満たす$(\mathbf{j}', p')$のうちで最大の順位にあるものを値とする（$\mathbf{j}' \notin Index$かも知れない）．特に，値の各々を示すため次の記法を用いる．$\mathbf{j} \cdot \langle p,q \rangle = \mathbf{j}'$，$st(\mathbf{j}, \langle p,q \rangle) = p'$．ドット・を逆行関数，stを文指定関数と呼ぶ．

流れ化変換をしているので生成時点関数の値は常に定義される．また，逆行関数は次のように表現できる．

$$\mathbf{j} \cdot \langle p,q \rangle = \max_{p' \in s^{-1}(arg(p,q))} I(p')$$

但し，$I(p') = \{ \mathbf{j}' \mid k_{p'}(\mathbf{j}') = k_p^q(\mathbf{j}),\ (\mathbf{j}', p') < (\mathbf{j}, p) \}$．

この定義から，$arg(p,q) = s(p') = a, k_p^q(\mathbf{j}) = k_{p'}(\mathbf{j}') = \mu$の場合，$\mathbf{j}$における文p の q番目の参照$a[\mu]$の値が，$\mathbf{j}'$において文$p'$で計算されていることを示す．また，このことは$\mathbf{j}$における$f_p$の計算より，$\mathbf{j} \cdot \langle p,q \rangle$における $f_{p'}$の計算が先に実行されなければならないことを意味する．これを計算の依存性という．

$IndexI = \{ \mathbf{j} \cdot \langle p,q \rangle \mid \mathbf{j} \in Index,\ \langle p,q \rangle \in Arg \}$，

$IndexA = Index \cup IndexI$

とおく．

２．３ システリックアレーの構成

ループプログラムに対してそれを実行するシストリックアレーは次のように構成される．

ループプログラムの本体の各計算式は，各セルのセル関数に対応させる．このとき，各セル関数の出力値が出力ポートに対応し，引数が入力ポートに対応するので，次のようになる．

$P_{out} = ID$，　$P_{in} = Arg = ID \times Occur$．

次に，各ループインデックス$\mathbf{j}$における計算を実行する時刻とセル位置を与える計算時刻関数Tc，計算セル位置関数Acを定める．

$Tc : IndexA \to T$，　$Ac : IndexA \to G$．

セルの配置領域Rについては次が成立する．

$Ac(Index) = R$．

各f_pの$\mathbf{j}$における計算は，セル$Ac(\mathbf{j})$で時刻$Tc(\mathbf{j})$に実行される．このf_pの q番目の参照変数（入力）は，$\mathbf{j} \cdot \langle p,q \rangle = \mathbf{j}'$ なるインデックスで $st(\mathbf{j}, \langle p,q \rangle) = p'$番目の計算式で計算される．従って，セル $Ac(\mathbf{j}')$で時刻$Tc(\mathbf{j}')$に計算され，時刻 $Tc(\mathbf{j})$までその計算結果を保存しておかなければならない．従って，セル$Ac(\mathbf{j})$の $i = \langle p,q \rangle$ という入力ポートはセル$Ac(\mathbf{j} \cdot i)$の出力ポート$st(\mathbf{j}, i)$に接続される．また，入力ポート i から計算素子 p の入力 q までに長さ $Tc(\mathbf{j}) - Tc(\mathbf{j} \cdot i) - 1$ のシフトレジスタが接続される．

セル機能の一様性，すなわち，どの$\mathbf{j}$に対する計算についても結線および関数機能が同一であるという要請から，次の条件を満たさなければならない．

(C0) $st(\mathbf{j}, i) = u(i)$ は$\mathbf{j}$に依存しない．

(C1) $Ac(\mathbf{j}) - Ac(\mathbf{j} \cdot i) = N(i)$ は$\mathbf{j}$に依存しない．

(C2) $Tc(\mathbf{j}) - Tc(\mathbf{j} \cdot i) = \delta(i)$ は$\mathbf{j}$に依存しない．

u はポート間結線写像であり，N は近傍写像である．δを時間調整写像という．計算の依存性，および，入出力ポート間のデータ転送に１単位時間を必要とすることから，次の必要条件が得られる．

(C3) $\delta(i) \geq 0$．特に，$\mathbf{j} \neq \mathbf{j} \cdot i$ ならば，$\delta(i) > 0$．

セル機能を単純にするために，同時刻，同一セルでは，高々１つのインデックスにおける計算だけ実行することを要請する．これを最簡機能条件といい，次のように記述される．

(C4) $Ac(\mathbf{j}) = Ac(\mathbf{j}')$ならば $Tc(\mathbf{j}) \neq Tc(\mathbf{j}')$．

この対偶は，“計算時刻が一致するならば計算セルは異なる”となる．

以上の条件(C1)～(C4)を満たすTc, Ac, N, δが存在すれば，それによりシストリックアレーが定まる．次にこのN, δに対する必要条件をより明示的な形に表現していく．

まず，Tc, Ac を組にして，写像 $TS: IndexA \to T \times G$ を $TS(\mathbf{j}) = \langle Tc(\mathbf{j}), Ac(\mathbf{j}) \rangle$ として導入する．時空写像と呼ぶ．更に，$\Delta(i) = \langle \delta(i), N(i) \rangle$ とおくと，(C1)，(C2) は，

$$TS(\mathbf{j}) - TS(\mathbf{j} \cdot i) = \Delta(i)$$

となる．これから次が成立する．

【命題１】[13] (1) $\mathbf{j} \cdot i = \mathbf{j}$ならば，$\Delta(i) = 0$．

(2) 各$i, i' \in Arg$ について，ある$\mathbf{j}$に対して，$\mathbf{j} \cdot i = \mathbf{j} \cdot i'$ ならば，$\Delta(i) = \Delta(i')$．

Arg を上の(2) に従って，同値類分割し，各同値類をe と表す．すなわち，$i, i' \in e \Leftrightarrow \exists \mathbf{j}; \mathbf{j} \cdot i = \mathbf{j} \cdot i'$．また，$\mathbf{j} \cdot i = \mathbf{j}$となるi が属する同値類を除いた同値類の集合をＥとおく．逆行関数は $IndexA \times E$ 上の関数に自然に拡張する．また，ΔもＥから$T \times G$への写像とみなす．

条件(C1)と(C2)は N, Ac, δ, Tc を未知関数とする方程式とみなすことができる．これを解くために，いくつかの記法を導入する．まず，$((\mathbf{j} \cdot e_1) \cdot e_2) \cdot e_3$ を $\mathbf{j} \cdot e_1 e_2 e_3$ と記す．Ｅの元からなる系列の集合をE^*と記し，その元を$e_1 e_2 \cdots e_n$あるいはσと記す．逆行関数は $IndexA \times E^*$ 上の関数に自然に拡張される．関数 $c: E^* \to [E \to Z]$ を，

$c(\sigma)(e) = $（$\sigma$中の eの生起回数）

と定める．更に，次の記法を導入する．

$$(c(\sigma), \Delta) = \Sigma_e\, c(\sigma)(e) \cdot \Delta(e)\ .$$

【命題2】[13]任意の $\mathbf{j}\in$ Index, $\sigma\in E^*$ に対して，

$$\mathrm{TS}(\mathbf{j})=\mathrm{TS}(\mathbf{j}\cdot\sigma)+(c(\sigma),\Delta).$$

IndexA上に関係←を次のように定める．$\mathbf{j}\leftarrow\mathbf{j}' \Leftrightarrow \exists\sigma;\ \mathbf{j}\cdot\sigma=\mathbf{j}'$．この関係は意味的には，$\mathbf{j}$ における計算のために $\mathbf{j}'$ における計算結果を必要とすることを示し，依存関係と呼ぶ．この依存関係の反射的交換的推移的閉包をとるとIndexA上の同値関係となり，⇔と表す．異なる同値類の元の間には依存関係が存在しないから，それらの計算は独立に実行可能であり，各々を実現することにより全体が実現できる．そこで，一般性を失うことなく，⇔による同値類は1つであるとする．すなわち，ある $\mathbf{j}_0\in$ IndexAが存在して，任意の $\mathbf{j}$ に対して $\mathbf{j}\Leftrightarrow\mathbf{j}_0$ であるとする．

この関係 $\mathbf{j}\Leftrightarrow\mathbf{j}_0$ を成立させる E^* の元の組を π とおく．すなわち，$\pi=(\sigma_0,\sigma_1,\sigma_2,\cdots,\sigma_{2n})$ に対して，$\mathbf{j}_{1,0}=\mathbf{j}$, $\mathbf{j}_{0,n}=\mathbf{j}_0$, $\mathbf{j}_{1,s}\cdot\sigma_{2s}=\mathbf{j}_{1,s+1}\cdot\sigma_{2s+1}=\mathbf{j}_{0,s}$ $(s=0\sim n)$ であり，σ_0,σ_{2n} 以外は空系列ではない．この π について，

$$c(\pi)=\Sigma_{s=0}{}^{n}\ c(\sigma_{2s})-\Sigma_{s=0}{}^{n-1}\ c(\sigma_{2s+1})$$

と定め，各 $\mathbf{j}$ について次の集合を定める．

$$V(\mathbf{j})=\{c(\pi)\mid \pi\text{は上で定めた}\ \mathbf{j}\Leftrightarrow\mathbf{j}_0\ \text{を成立させる系列の組}\}$$

【命題3】任意の $\nu\in V(\mathbf{j})$ について，

$$\mathrm{TS}(\mathbf{j})=\mathrm{TS}(\mathbf{j}_0)+(\nu,\Delta).$$

(証明) $\nu=c(\pi)$ である．$\pi=(\sigma_0,\sigma_1,\sigma_2,\cdots,\sigma_{2n})$ の定義および命題2から，各 $s=0\sim n$ に対して，

$$\mathrm{TS}(\mathbf{j}_{1,s+1})=\mathrm{TS}(\mathbf{j}_{0,s})+(c(\sigma_{2s+1}),\Delta)$$

$$\mathrm{TS}(\mathbf{j}_{1,s})=\mathrm{TS}(\mathbf{j}_{0,s})+(c(\sigma_{2s}),\Delta)$$

従って，

$$\mathrm{TS}(\mathbf{j}_{1,s})=\mathrm{TS}(\mathbf{j}_{1,s+1})-(c(\sigma_{2s+1}),\Delta)+(c(\sigma_{2s}),\Delta)$$

これを s について加えていって，命題の式を得る．□

そこで，$V(\mathbf{j})$ の代表元を1つ選んで，それを値とする写像 $v(\mathbf{j})$ を定義できる．次が成立する．

【系1】写像 v:Index→[E→Z]が存在して，

$$\mathrm{TS}(\mathbf{j})=\mathrm{TS}(\mathbf{j}_0)+(v(\mathbf{j}),\Delta).$$

$v(\mathbf{j})$ は，$\mathbf{j}_0$ から $\mathbf{j}$ までの計算の依存関係の状況を縮約し代表したものとなっており，$\mathbf{j}$ の状況ベクトルという．系1は，$\mathrm{TS}(\mathbf{j})$ が定数 $\mathrm{TS}(\mathbf{j}_0)$ と Δ の線形結合とから定まることを示している．Δ を決めることでTSが定まることが知られた．

Δ に関する必要条件は次のように与えられる．

【命題4】(1)ある $\mathbf{j}\in$ Index において $\mathbf{j}\cdot\sigma=\mathbf{j}\cdot\sigma'$ ならば，$(c(\sigma)-c(\sigma'),\Delta)=0$.

(2)任意の $\nu_1,\nu_2\in V(\mathbf{j})$ について，$(\nu_1-\nu_2,\Delta)=0$.

(証明)それぞれ命題2，3から明らかである．□

3. 実現可能条件

3.1 近傍写像・時間調整写像の形

Δ の形を明らかにする．そのために，以下では，$G=Z^K$ の場合を考える．Δ は，EからT×Gへの写像であるが，$T\times G=T\times Z^K$ より，各 $e\in E$ について，$\Delta(e)$ は $(1+K)$ 次元行ベクトルとみなすことができる．また，$|E|=m$ として，Δ 自身は $m\times(1+K)$ 行列とみなす．更に，$c(\sigma)$ はEからZへの写像であったが，m 次元行ベクトルとみなす．すなわち，$c(\sigma)\in Z^m$ $(=[E\to Z])$．また，$(c(\sigma),\Delta)=c(\sigma)\cdot\Delta$ と行ベクトルと行列との積として表す．ベクトル $\mu^1,\mu^2,\cdots,\mu^s$ に対して，それらから生成される線形空間を $\mathrm{Span}(\mu^1,\mu^2,\cdots,\mu^s)=\{\Sigma_i\ a_i\mu^i \mid a_i$ は有理数$\}$ と定める（スカラー倍は有理数に限っていることに注意する）．

次に，E^* 上に同値関係≡を導入する．

$$\sigma\equiv\sigma' \Leftrightarrow \exists\mathbf{j};\ \mathbf{j}\cdot\sigma=\mathbf{j}\cdot\sigma'\ \text{かつ}\ c(\sigma)\neq c(\sigma').$$

ループプログラムの逆行関数から一意に次の集合が定まる．

$$\mathrm{Diffc}=\{\mu\mid \mu=c(\sigma)-c(\sigma'),\ \sigma\equiv\sigma'\}\subseteq Z^m.$$

命題4(1)は，任意の $\mu\in$ Diffc について，$\mu\Delta=0$ が成立しなければならないことを示している．

$$\mathrm{Orthc}=\{h\mid \forall\mu\in\mathrm{Diffc};\ \mu h=0\}\subseteq Z^m$$

とおく．Diffc の元に直交する列ベクトルの集合である．これは，Δ の列，すなわち，δ，N の列の候補の集合である．

Diffc から1次独立なベクトルを選出する．すなわち，$\mu^1\in$ Diffc を1つ選ぶ．次に，$\mu^2\in$ Diffc-Span(μ^1) を選ぶ．同様にして，$\mu^1,\mu^2,\cdots,\mu^i$ が選ばれているとき，$\mu^{i+1}\in$ Diffc-Span$(\mu^1,\mu^2,\cdots,\mu^i)$ を選ぶ．Diffc-Span$(\mu^1,\mu^2,\cdots,\mu^d)=\phi$（空集合）と初めてなった d を Diffc の次元という．明らかに，$d\leq m$ である（なぜなら，$m+1$ 個の m 次元ベクトルは1次従属となる）．この選ばれた $\mu^1,\mu^2,\cdots,\mu^d$ を Diffc の基底といい，$d\times m$ 行列 $B=\begin{bmatrix}\mu^1\\ \vdots\\ \mu^d\end{bmatrix}$ に表現する．また，これらにより生成される線形空間をSpan(B)と記す．定義から，$0\notin$ Diffc, $0\in$ Span(B) である．

【補題1】$\forall\mu\in$ Diffc, $\exists\xi\in Z^d$, $\exists b\in Z$;

$$\xi B=b\mu.$$

(証明) $\mu\in$ Span(B) より，有理数ベクトル ξ_R が存在して，$\xi_R B=\mu$．ξ_R の各要素の規約分数表現からその分母の最小公倍数を b とおいて，$\xi=b\xi_R$．□

【補題2】列ベクトル $h\in Z^m$ について，すべての $\mu\in$ Diffcについて $\mu h=0$ である必要十分条件は $Bh=0$ であることである．

(証明)必要性はBの各行がDiffc の元であるから明らかである．十分性は補題1から成立する．□

Bの列を入れ換えて，$B=[D\ \ S]$，$d\times d$ 行列Sは逆行列が存在するというふうにできる．Sが整数行列だから S^{-1} の各要素は有理数であり，それを規約分数表現し，その分母の最小公倍数を b とおいて，$bS^{-1}=S^\dagger$ とおく．$S^\dagger$ は整数行列である．また，$S^\dagger S$

$=SS^{\dagger}=bI$ である．Iは単位行列である．

【補題3】任意の$h\in$ Orthcに対して，m-d 次元整数ベクトルh_Fが存在して，$h=\begin{bmatrix} I \\ -(1/b)S^{\dagger}D \end{bmatrix}h_F$ である．

逆に，$S^{\dagger}Dh_F$ の各要素が bの倍数であるm-d次元整数列ベクトルh_F を選び，上式により h を定めると h はOrthcの元である．

(証明)$h=(h_F{}^T,h_D{}^T)^T$ とおくと，$Bh=0$より，$Dh_F+Sh_D=0$．従って，$h_D=-S^{-1}Dh_F$． □

これは，Orthc の次元が m-dであることを明示的に示している．d=mならば，Orthc={0} となり，$\Delta=0$ である．この場合，$\delta>0$ となるΔを選ぶことができず，この方法ではシストリックアレーを構成できない．以下では，d<m と仮定する．

Δはm×(1+K)行列であった．これを m次元の列ベクトルの集合とみて1次独立なものの個数を1+q とする（q≦K,m-1）．δを含めて，1次独立なベクトルを選び，それらをm×(1+q)行列Δ_B で表現する．Δの列を入れ換えることによって，$\Delta=[\Delta_B,\Delta_D]$ とできる．Δ_B の定義から，$\Delta_D=\Delta_B C$ となる(1+q)×(K-q)行列Cが存在する．すなわち，

$$\Delta=\Delta_B[I\ \ C]$$

と表現できる．従って，セル空間Gは K次元空間としたが，Ac(Index) として定まる実際に利用される空間は q次元空間である．従って，K=qとしても一般性を失わない．すなわち，構成すべきシストリックアレーのセル空間の次元はΔの列ベクトルの次元から決まるとしてよい．以下では，特に断わらない限り，K=q，$\Delta=\Delta_B$とする．

さて，Δの各列は Orthcの元であるから，補題3より，(m-d)×(1+q)行列Q_F が存在して，

$$\Delta=\begin{bmatrix} I \\ -(1/b)S^{\dagger}D \end{bmatrix}Q_F$$

と表される．s=m-d-1 とおくと，Q_F の各列は1次独立だから，q≦s である． そこで，Q_F の行を入れ換えて（Δの対応する行も入れ換える），

$Q_F=\begin{bmatrix} Q \\ RQ \end{bmatrix}$とできる．但し，Qは(1+q)×(1+q)行列で，Q^{-1}が存在するものである．$Q^{-1}=(1/w)Q^{\dagger}$ と表す．w∈Z，$Q^{\dagger}$ は整数行列である．Q_FのQ以外の行はQの行の1次結合で表されるから，(s-q)×(1+q)行列RによりRQと表される．従って，次が成立する．

【命題5】Δは次の形をもつ．

$$\Delta=\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & -(1/b)S^{\dagger} \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & I \\ D^{(1)} & D^{(2)} \end{bmatrix}\begin{bmatrix} I \\ R \end{bmatrix}Q[I\ C]$$

但し，右辺の行列はそれぞれ，m×m，m×(1+s)，(1+s)×(1+q)，(1+q)×(1+q)，(1+q)×(1+K)行列である（m-1≧m-d-1=s≧q，q≦K）．R，Cは有理数行列であるが，他は整数行列である．$D=[D^{(1)}D^{(2)}]$であり，$D^{(1)}$，$D^{(2)}$ はd×(1+s)，d×(s-q)行列である．

従って，Δは与えられたBに対して，Q，R，(C)を定めることで定まる．命題5で最初の3つを計算すると，

$$\Delta=[\delta,N]=\begin{bmatrix} I \\ R \\ -(1/b)S^{\dagger}(D^{(1)}+D^{(2)}R) \end{bmatrix}Q[IC]$$

とも表される．

以上は，命題4(1)の必要条件を基礎に導かれた．次に，命題4(2)の必要条件について考える．

各 j∈IndexA について，

$$\mathrm{Diffv}(\mathbf{j})=\{\mu \mid \mu=\nu_1-\nu_2,\ \nu_1,\nu_2\in V(\mathbf{j})\}$$

とおき，$\mathrm{Diffv}=\bigcup_{\mathbf{j}\in \mathrm{IndexA}}\mathrm{Diffv}(\mathbf{j})$ とおく．

【補題4】 Diffc ⊆ Diffv．

(証明) ∀μ∈Diffc，∃j，$\mathbf{j}\cdot\sigma=\mathbf{j}\cdot\sigma'$ かつ $\mu=c(\sigma)-c(\sigma')$．$\mathbf{j}\cdot\sigma=\mathbf{j}_1$ とおき，$\nu\in V(\mathbf{j}_1)$をとる．明らかに，$c(\sigma)+\nu\in V(\mathbf{j})$．$\mathbf{j}\cdot\sigma'=\mathbf{j}_1$であるから，$c(\sigma')+\nu\in V(\mathbf{j})$．故に，

$\mu=(c(\sigma)+\nu)-(c(\sigma')+\nu)\in\mathrm{Diffv}(\mathbf{j})$． □

命題4(2)より，すべてのμ∈Diffvについて，$\mu\Delta=0$．そこで，Diffc に対するのと同様にDiffv の基底を選ぶ．補題4より，そのとき，Diffc の基底Bを含むようにでき，その結果をB_v とおく．補題2と同様に，∀μ∈Diffv，$\mu\Delta=0 \Leftrightarrow B_v\Delta=0$．

Diffv の次元d_vは $m\geq d_v\geq d$ を満たす．$d_v=m$ の場合はやはり$\Delta=0$ となり，$\delta>0$ となるδを選ぶことができない．

以下，B_v を用いて，同様の議論でΔの形を求めることができるが，ここでは省略する．

既存のループプログラムにおいては，ある$\mathbf{j}_0$ が存在して，任意のjに対してあるσが存在して $\mathbf{j}\cdot\sigma=\mathbf{j}_0$ となっている．この$\mathbf{j}_0$ はループインデックスの初期値である．この場合は次が成立し，Diffc に基づく議論で十分であることが分かる．

【命題6】 $\exists\mathbf{j}_0,\forall\mathbf{j},\exists\sigma;\ \mathbf{j}\cdot\sigma=\mathbf{j}_0$ ならば，

$$\mathrm{Span}(\mathrm{Diffc})=\mathrm{Span}(\mathrm{Diffv}).$$

従って，$B_v=B$とできる．

V(j)の代表元v(j)を選んで，TS(j) が表現できることを示したが，これらとDiffv との関係は次のようである．＋は元どうしの和を表す．

【命題7】 $V(\mathbf{j})=v(\mathbf{j})+\mathrm{Diffv}$．

(証明) $\forall\nu\in V(\mathbf{j})$，$\mu=\nu-v(\mathbf{j})\in\mathrm{Diffv}(\mathbf{j})$ であるから，$\nu=v(\mathbf{j})+\mu\in v(\mathbf{j})+\mathrm{Diffv}$．一方，$\forall\mu\in\mathrm{Diffv}$，$\exists\mathbf{j}'$，$\exists\nu_1,\nu_2\in V(\mathbf{j}')$;$\mu=\nu_1-\nu_2$．故に，明らかに，$v(\mathbf{j})+\nu_1-\nu_2=c(\pi)$ を満たす j から$\mathbf{j}_0$ へのE^* の元の組πが存在する．従って，$v(\mathbf{j})+\mu=c(\pi)\in V(\mathbf{j})$．よって，証明された．□

3.2 最簡機能条件

次の命題は最簡機能条件の定義から明らかである．

【命題8】最簡機能条件は，時空写像 TS が1対1で

あること，すなわち，$\mathbf{j} \neq \mathbf{j}'$ならば $TS(\mathbf{j}) \neq TS(\mathbf{j}')$ であることに等価である．

このことは，$TS^{-1}(t, \alpha)$が存在すれば一意であることを意味し，各セルでインデックス計算可能であること，すなわち，セルαで時刻tにおいてそのとき実行する計算のループインデックス$\mathbf{j}$がαとtとから一意に計算できることを意味している．

以下では，TSが最簡機能条件を満たすものとして，その必要条件を示す．

【命題９】(1) $\{V(\mathbf{j}) \mid \mathbf{j} \in IndexA\}$は，互いに排反な$Z^m$の部分集合の族である．すなわち，$\mathbf{j} \neq \mathbf{j}'$ならば$V(\mathbf{j}) \cap V(\mathbf{j}') = \phi$．

(2) $\mathbf{j} \neq \mathbf{j}'$ならば，任意の$\nu \in V(\mathbf{j})$，$\nu' \in V(\mathbf{j}')$について，$\nu\Delta \neq \nu'\Delta$．

(証明) $\nu \in V(\mathbf{j}) \cap V(\mathbf{j}')$ならば，命題３より，

$$TS(\mathbf{j}) = TS(\mathbf{j}_0) + \nu\Delta = TS(\mathbf{j}').$$

これはTSの１対１に反する．(2)も同様である． □

【命題１０】(1) $\sigma \equiv \sigma'$ならば，$\forall \mathbf{j};\ \mathbf{j}\cdot\sigma = \mathbf{j}\cdot\sigma'$．

(2) $\sigma, \sigma' \in E^*$について，$c(\sigma) = c(\sigma')$ならば，

$$\forall \mathbf{j};\ \mathbf{j}\cdot\sigma = \mathbf{j}\cdot\sigma'.$$

(3) $\forall \sigma_1, \sigma_2 \in E^*;\ \mathbf{j}\cdot\sigma_1\sigma_2 = \mathbf{j}\cdot\sigma_2\sigma_1$

(証明)命題２から容易に示される． □

セル位置α，計算時刻tが与えられたとき，ループインデックス$\mathbf{j}$を求めることを考える．$1+q$ 個の１次独立な列ベクトルからなるΔを定めたとする．

$$TS(\mathbf{j}) = (t, \alpha) = \beta \in Z^{1+q},$$

$$TS(\mathbf{j}_0) = (t\ \ \alpha_0) = \beta_0$$

とおく．$\beta = \beta_0 + v(\mathbf{j})\Delta$ である．そこで，

$$Cand(\beta) = \{\nu \mid \beta = \beta_0 + \nu\Delta\}$$

とおく．

【補題５】$\forall \nu \in Cand(\beta)$;

$$\begin{aligned} \nu = (\beta - \beta_0)\,&[Q^{\dagger}(1/w), 0, 0] \\ &+ \xi^{(2)}[-R,\ \ I, 0] \\ &+ \xi^{(1)}[D^{(1)},\ D^{(2)}, S] \end{aligned}$$

但し，$\xi^{(2)} \in Z^{s-q}$，$\xi^{(1)} \in Z^{d}$．

(証明)命題５から容易に示せる． □

従って，$Cand(\beta)$は $s-q+d\ (=m-q-1)$次元空間である．

【命題１１】$Cand(\beta) \cap V(\mathbf{j}) \neq \phi$ となる$\mathbf{j}$は一意である．また，このとき，$V(\mathbf{j}) \subseteq Cand(\beta)$ かつ $TS(\mathbf{j}) = \beta$．

(証明)一意でなければ，$\mathbf{j} \neq \mathbf{j}'$ かつ $Cand(\beta) \cap V(\mathbf{j}) \neq \phi$ かつ $Cand(\beta) \cap V(\mathbf{j}') \neq \phi$ となる$\mathbf{j}$，$\mathbf{j}'$が存在する．共通集合の各々の元をν，ν'と選ぶと，$\beta_0 + \nu\Delta = \beta = \beta_0 + \nu'\Delta$．従って，命題３より，$TS(\mathbf{j}) = TS(\mathbf{j}')$が成立する． □

$V(\mathbf{j})$はd_v次元，B_v からΔを定めたとき$Cand(\beta)$は$s_v-q+d_v\ (=m-q-1)$次元である．$s_v - q \geqq 0$より$Cand(\beta)$の方が次元が高い．

４．　適用例

$V(\mathbf{j})$と$Cand(\beta)$の次元が一致する必要はないのか．また，$q=s_v$ とする必要はないのか．$v: Z^n \to Z^m$ について，v^{-1} を求めるためには vが線形であると議論しやすくなる．しかし，これらの予想あるいは期待が成立しないことが次の例からわかる．

４．１　行列積

行列積を計算する次のループプログラムを考える．

```
for i=1 to I
 for j=1 to J
  for k=1 to K
     c(i,j)=c(i,j)+a(i,k)*b(k,j)
```

これに対して，流れ化変換などをしたのち逆行関数を求めると次のようになる．

$E = \{b, a, c\} = Pin$，$\mathbf{j} = (i, j, k)$．

$\mathbf{j}\cdot b = (i-1, j, k) = \mathbf{j} - (1, 0, 0)$

$\mathbf{j}\cdot a = (i, j-1, k) = \mathbf{j} - (0, 1, 0)$

$\mathbf{j}\cdot c = (i, j, k-1) = \mathbf{j} - (0, 0, 1)$

３次元３方向で，$Diffc = \phi$．従って，

$m = 3, d = 0, s = 2, q \leqq s$，$n$(多重度)$=3$．

$\mathbf{j}\cdot a^j b^i c^k = O = \mathbf{j}_0$．

$$\begin{aligned} TS(\mathbf{j}) &= TS(\mathbf{j}\cdot a^j b^i c^k) + c(a^j b^i c^k)\Delta \\ &= TS(O) + (i, j, k)\Delta \end{aligned}$$

但し，Δの行の順番はb, a, cの順である．

$V(\mathbf{j}) = \{\mathbf{j}\}$，$v(\mathbf{j}) = \mathbf{j}$．

① $q=0$ の場合：$\Delta = [\delta, 0]$，$N = 0$．

$\delta = [JK, K, 1]^T$ とすれば，考えているIndex の範囲では TS は１対１になる．(逐次処理になっている)．

② $q=2=s$ の場合：Δは3×3行列で$1+2$個の１次独立なベクトルをもつようにする．すなわち，$\det \Delta \neq 0$ となるようにする．TSは１対１である．

４．２　コンボリューション

データ依存ベクトルが定数とならず，ループインデックスが動的に変わる次のループプログラム[13]を考える．(従来の方法[2]など を適用できない例である)．

```
for i=0 to n+m
{ i_o = ⌊i/2⌋;
  if i is even then c_i = a_io b_io
              else c_i = a_i-io b_io + a_io b_i-io
  for j=i_o-1 to 0 by -1
    c_i = c_i + a_i-j b_j + a_j b_i-j
}
```

これを次のように等価変換する．

```
for i=0 to n+m
 for k=0 to ⌊i/2⌋ { i_o = ⌊i/2⌋;
  c_i = if k==0 and i is even then a_io b_io
  else if k==0 and i is odd then
                               a_i-io b_io + a_io b_i-io
  else  c_i + a_i-io+k b_io-k + a_io-k b_i-io+k
}
```

さらに流れ化変換などをする（詳細は[13]参照）.

```
  for i=0 to n+m
   for k=0 to ⌊i/2⌋ { i_o = ⌊i/2⌋ ;
 a_{i-io+k} = a_{i-io+k} ;   b_{i-io+k} = b_{i-io+k} ;
 A_{io-k}= if k=0 and i is even  then  a_{i-io+k}
                                 else  A_{io-k}  ;
 B_{io-k}= if k=0 and i is even  then  b_{i-io+k}
                                 else  B_{io-k}  ;
 c_i     = if k=0 and i is even  then  A_{io-k}b_{i-io+k}
   else if k=0 and i is odd then
                         a_{i-io+k}B_{io-k}+A_{io-k}b_{i-io+k}
   else  c_i+a_{i-io+k}B_{io-k}+A_{io-k}b_{i-io+k}  ;
 }
```

$E=\{0,1,2\}$.　$\mathbf{j}=(i,k)$.

0:<cc>, 1:<aa>,<bb>, 2:<AA>,<BB>. ここで<cc>などは<pq>に対応し，文c の参照変数c を意味する.(<Aa>などは $\mathbf{j}\cdot$<Aa>$=\mathbf{j}$ となり，セル内直結でEの元にならない).

$(i,k)\cdot 1=(i-1,k+(i \bmod 2))$

$(i,k)\cdot 2=(i-1,k-1+(i \bmod 2))$

$(i,k)\cdot 0=(i,k-1)$

２次元３方向のためサイクルができる.

$(i,k)\cdot 01=(i,k)\cdot 2$ より,

$\mathrm{Diffc}=\{\ell(1,1,-1)\mid \ell\in Z\}$

$B=(1,1,-1)$.

iが偶数のとき，$(i,k)\cdot(12)^{i/2}0^k=(0,0)$.

iが奇数のとき，$(i,k)\cdot 2(12)^{i/2}0^k=(0,0)$.

従って，$\mathbf{j}_0=(0,0)$のとき,

$(k,\lfloor i/2\rfloor,\lfloor i/2\rfloor+i \bmod 2)\in V(i,k)$

$v(i,k)=$(上式)$+\ell(1,1,-1)$

で適当な ℓ を選べば良い. $\ell=\lfloor i/2\rfloor+i \bmod 2$ とすれば,

$v(i,k)=(k+\lfloor i/2\rfloor+i \bmod 2,\ i,\ 0)$

$v:Z^2\to Z^3$ は非線形である.

$V(i,k)=v(i,k)+\mathrm{Diffc}$.

$m=3, d=1, s=3-1-1=1, q\leqq s$ より $q=0$ or 1. $n=2$.

○ $q=1$ の場合： $\Delta=\begin{bmatrix}1 & -1\\ 1 & 1\\ 2 & 0\end{bmatrix}$ とすればTSは１対１である.

5.　むすび

シストリックアレーの定式化を基礎にしてループプログラムからシストリックアレーを構成する一般的方法について，詳細な検討を加えた．特に，近傍写像，時間調整写像の形を明示的に与え，自動設計が可能な形にした．また，最簡機能条件について，ループプログラムに要請される必要な性質を示した.

この構成法に並列度や計算時間の評価を導入してより実用的なものにすることおよびループプログラムの多重度とシストリックアレーの次元との関係をより明確にすることは，今後の課題である.

最後に，本研究を進めるうえで御討論頂いた東北大学木村正行教授に感謝する．また，本研究の一部は文部省科学研究費補助金（総合研究(A)課題番号62302032）の援助を受けている．ここに記して感謝する.

文　献

[1] H.T.Kung;"Let's design algorithms for VLSI", Proc.Caltech Conf.VLSI,pp65-90.(1979)

[2] D.I.Moldovan;"On the analysis and synthesis of VLSI algorithms",IEEE.Trans.Comp.C-31,11, pp.1121-1126.(1982)

[3] D.I.Moldovan;"On the design of algorithm for VLSI systolic arrays",Proc.IEEE,71,1,pp.113-120. (1983)

[4] 若林,菊野,吉田;"漸化式を用いたハードウェアアルゴリズムの設計について"，信学論Ｄ,J67-D,8, pp.861-868.(1984)

[5] P.Quinton ; "Automatic synthesis of systolic arrays from uniform recurrent equations", Proc.11th Ann.Int.Symp.Comp.Arch.,pp.208-214. (1984)

[6] G.J.Li and B.W.Wah; "The design of optimal systolic arrays",IEEE Trans.Comp.C-34,1, pp.66-77.(1985)

[7] D.I.Moldovan, I.A.B.Fortes : "Partitioning and mapping algorithms into fixed size systolic arrays",TEEE Trans.Comp.C-35,1, pp.1-12.(1986)

[8] 三浦,阿曽,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",信学論Ｄ,J70-D,3,pp.515-524.(1987)

[9] 堀池,西田,坂口；"プロセッサ数の制限下におけるシストリックアレーの設計法",信学論Ｄ, J70-D,5,pp.880-888.(1987)

[10] 阿曽，稲垣；"シストリックアルゴリズムの定式化と情報の流れ",信学論Ｄ，J70-D,6,pp.1074-1082（1987)

[11] 飯国,酒井,得丸；"データフローグラフを用いたシストリックアレイの構成法",信全大(総合), 1676.(1987)

[12] J.A.Fortes and B.W.Wah(Eds.);"Systolic Arrays",Computer,20,7,IEEE.(1987)

[13] 阿曽；"シストリックアレーの自動設計法", 信学論Ｄ，J71-D,8.（1988）（掲載予定）（"セル構造情報処理系の自動設計"，『セル構造に基づく高度並列情報処理システムに関する総合的研究』報告書，名古屋大学.(1988).）

（以上）