

セルラ計算機のための最適時間・並列アドレス 設定アルゴリズム†

梅 尾 博 司**

並列計算機におけるアドレス情報は、たとえば SIMD 型並列アルゴリズムを設計する際など重要な役割を果たす。本稿では、多数の同一プロセッサからなる並列セルラ計算機上でのアドレス設定問題について、オートマトン理論的な観点から考察し、VLSI 化に適した二つの最適時間・並列アドレス設定アルゴリズムを提案する。これらはパイプライン・アドレス設定法ならびに並列2分割・アドレス設定法と呼ばれ、両手法ともセルラロジックにより記述され、VLSI ベース・セルラ計算機・システムのアドレス設定に適している。 n 個のセルに対し、前者は $n-1+\lceil \log_2 n \rceil$ ステップ、後者は $2n-1$ ステップの時間計算量をもつ。本論文で提案したアドレス設定法は、 k 次元格子状結合をもつセルラ計算機上に容易に拡張可能である。

1. ま え が き

ハードウェア価格の低下とともに、種々のアーキテクチャをもった専用あるいは汎用の VLSI ベース・セルラ計算機に関する研究が数多くなされている^{1),3),5),8)}。

並列計算機におけるアドレス情報は、たとえば SIMD 型並列アルゴリズムを設計する際など重要な役割を果たす。さらに高速アドレス設定法は、データ・フロー・マシン、可変構造 (reconfigurable) 計算機など新しいアーキテクチャをもつ並列計算機に対して有用である可能性がある^{6),7)}。

セルラオートマトンは、

1. 規則正しいセル配列、
2. 各セルは隣接セルとのみ接続され、大局的通信網をもたない、
3. セルラアルゴリズムの記述は有限で、セル数に依存しない。したがって、アルゴリズムの変更なしに、セル数の増大に対処可能である、
4. 単純なデータフロー、

などの理由により、VLSI 向きであると指摘されている^{3),7)}。

本稿では、多数の同一プロセッサからなる並列セルラ計算機上でのアドレス設定問題についてセルラオートマトン理論的な観点から考察し、セルラロジックとして記述可能ないくつかの最適時間アドレス設定アル

ゴリズムを提案する^{9),10)}。本アルゴリズムは、上記の理由から VLSI 向きと考えられ、将来製造されるであろう VLSI ベース・セルラ計算機上でのアドレス設定に有用と思われる。本問題は、文献 1) において SIMD 型並列計算機のシストリック・アレイによる模倣法を考察した際、生じた問題である。

まず、2章ではアドレス設定問題を定義し、次いで 3.1 節ではパイプライン・アドレス設定法と呼ばれるアルゴリズムを提案する。 n 台のプロセッサに対し、 $n-1+\lceil \log_2 n \rceil$ ステップでアドレス設定を完了する。これは、各プロセッサにあらかじめプロセッサ総数に関する情報を与えた場合における最適時間アルゴリズムの一つである。

3.2 節では、並列2分割・アドレス設定法を提案する。本アルゴリズムは、セルラ・オートマトンにおける一斉射撃アルゴリズム^{2),4),6)}をアドレス設定という観点から見なおしたもので、 $2n-1$ ステップの時間計算量をもつ。これは、各プロセッサにあらかじめプロセッサ総数に関する情報をまったく与えない場合の最適時間アルゴリズムの一つである。

最後に、本稿で提案したアドレス設定アルゴリズムの2次元アレイ上への拡張について論議する。

2. アドレス設定問題

VLSI 技術の発展とともに、1チップあたり $10^3 \sim 10^5$ 個のプロセッサからなる1チップ・セルラ計算機の製造も近い将来可能となろう。それらのなかには、プロセッサ間の接続関係に比較的自由度をもたせた可変構造セルラ計算機も考えられ、解くべき問題に応じてプロセッサ・ネットワークの選択が可能になれば、

† Time-Optimum Parallel Binary Address Setting Algorithms for Cellular Computers 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

並列計算機の効率向上に役立つと思われる。ネットワークが変化するとそれに伴ってプロセッサ・アドレスも変化し、そのネットワークに応じたアドレスの高速設定が望まれる。

本稿では、図1に示す最も単純なネットワークを仮定したセルラ計算機Mを考え、ホスト計算機からの指令によりM上の各プロセッサのアドレスを高速に設定するアルゴリズムを提案する。Mを構成するプロセッサは両端を除きすべて同一構造をもつ。プロセッサ数を n とする。各プロセッサを以下ではセルと呼び、 $C_i(0 \leq i \leq n-1)$ で表す。

Mは C_0 を介してホスト計算機と接続されている。両端のセル C_0 および C_{n-1} はエンド・セルと呼ばれ、それぞれ C_1 , C_{n-2} と接続されている。他の $C_i(1 \leq i \leq n-2)$ は、すべて一様に C_{i-1} , および C_{i+1} と直接つながれている。

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。セル上のすべての有限レジスタの内容の組合せを、そのセルの内部状態と呼ぶ。 $C_i(0 \leq i \leq n-1)$ の時刻 $t(\geq 0)$ における内部状態を s_i^t で表す。

最初Mのすべてのセルは、静止状態にいる。 $t=0$ のときにホスト計算機から C_0 に与えられる信号により、Mは左端から順に動作を開始する。任意の $t(\geq 0)$ に対し、 $s_i^{t+1}(1 \leq i \leq n-2)$ は、 $s_{i-1}^t, s_i^t, s_{i+1}^t$ の局所的な情報に依存して局所状態遷移関数 δ により次のように決定される。

〔2方向関数〕

$$\begin{aligned} s_i^{t+1} &= \delta(s_{i-1}^t, s_i^t, s_{i+1}^t) & t \geq 0, 1 \leq i \leq n-2 \\ s_0^{t+1} &= \delta_1(s_0^t, s_1^t) & t \geq 1 \\ s_{n-1}^{t+1} &= \delta_2(s_{n-2}^t, s_{n-1}^t) & t \geq 0 \end{aligned}$$

と決定される時、 $(\delta, \delta_1, \delta_2)$ を2方向関数と呼ぶ。ここに、 $\delta, \delta_1, \delta_2$ はそれぞれ、 $\delta: Q^3 \rightarrow Q, \delta_1, \delta_2: Q^2 \rightarrow Q$ なる関数である。ただし、 Q はセルの内部状態集合を意味する。

〔1方向関数〕

$$\begin{aligned} s_i^{t+1} &= \delta(s_{i-1}^t, s_i^t) & t \geq 0, 1 \leq i \leq n-1 \\ s_0^{t+1} &= \delta_1(s_0^t) & t \geq 1 \end{aligned}$$

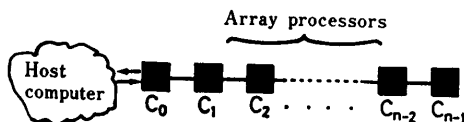


図1 ホスト計算機に接続されるセルラ計算機
Fig. 1 An attached cellular computer.

と決定される時、 (δ, δ_1) を1方向関数と呼ぶ。ここに $\delta: Q^2 \rightarrow Q, \delta_1: Q \rightarrow Q$ なる関数である。1方向、2方向はそれぞれアレイ上における情報伝達方向の自由度を意味する。Mの状態遷移の仕方は、大局的な通信網(global communication)をもたないストリック・アレイ³⁾のそれとまったく同じである。

M上のアドレス設定問題とは、Mの各セル $C_i(0 \leq i \leq n-1)$ のアドレス i を計算し、 i の2進表現を C_i の R_a に貯えるセルの遷移関数 $\delta^*, \delta_1^*, \delta_2^*$ ならびに δ^* (n に依存しない)を設計する問題である。ここに δ^* は、本問題のため δ の定義域を $Q \times Q \times Q \times \Gamma \rightarrow Q \times \Gamma$ (あるいは $Q \times Q \times \Gamma \rightarrow Q \times \Gamma$)に拡張した局所状態遷移関数である。 Γ は R_a に記入されるアルファベット集合で、 $\Gamma = \{0, 1, \vee\}$ とする。 δ_1^*, δ_2^* も同様である。

各セルの R_a は、整数 $i(0 \leq i \leq n-1)$ の2進表現を貯えるのに十分なメモリ、すなわち少なくとも $m(= \lceil \log_2 n \rceil)$ 個の桁をもつものとする。下位の桁から順に $d_0, d_1, \dots, d_{m-2}, d_{m-1}, \dots$ とする。 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 $j(0 \leq j \leq m-1), i(0 \leq i \leq n-1)$ に対し、 $d_j^t(i) = \vee$ (ブランク記号)とする。

上記アドレス設定問題を考えるに際し、オートマトン理論的な観点から次の仮定をおく。

A1: 1方向状態遷移関数

A2: 2方向状態遷移関数

A3: 各セルは、1ステップにつき R_a 上の少なくとも $\lceil \log_2 n \rceil$ 桁ある桁のうちただかかひと桁の内容を“ \vee ”から“0”に、あるいは“ \vee ”から“1”に変更できる。

A4: 左端セル C_0 はプロセッサ総数に関する情報を $d_{\lceil \log_2 n \rceil}^0(0) = \$$ なる形で R_a にもっている。ただし、このとき $\Gamma = \{0, 1, \vee, \$\}$ とする。

3. アドレス設定アルゴリズム

2種類の最適時間アドレス設定アルゴリズムを提案する。アルゴリズムは、波頭記法(wave-front notation)により記述される。波頭記法は、従来からセルラ・オートマトン・アルゴリズムを簡潔に表現する有用な道具として知られ、最近では文献1)においてVLSIベース・セルラ計算機上でのアルゴリズム記述に使われている。波頭記法で記述されたアルゴリズムから状態遷移関数への変換は容易である。

3.1 パイプライン・アドレス設定アルゴリズム

まずA1, A3, A4を仮定したパイプライン・ア

ドレス設定法 (Ver. 1) を提案する. 本アルゴリズムは文献9)においてすでに報告済みであるが, 以下の議論で重要な役割を果たすので, ここで再考察する.

[パイプライン・アドレス設定法 (Ver. 1)]

(アルゴリズムと時間計算量)

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. $j=0, 1, 2, \dots, m-1$) 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,

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

w_j は w_{j-1} の後を追いかけて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,

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

$C_i (1 \leq i \leq n-1)$ に対しては,

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

と決定される. ただし, $1 \leq j \leq m-1, t \leq 2$. 記号 " \bar{x} " は $x(x \in \{0, 1\})$ 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, 以下右端まで交替的にこれを繰り返す.

R_a は正確に $\lceil \log_2 n \rceil$ 桁に区切られているため, R_a のすべての桁が一杯になったときに限り C_0 は波の生成を中止する. 図2は, $n=8$ の場合のパイプライン・アドレス設定法を例示したものである.

C_i のアドレス設定は $i-1 + \lceil \log_2 n \rceil$ ステップ時に完了する. したがって $n-1 + \lceil \log_2 n \rceil$ ステップで M のアドレス設定は終了する. 本設定法は, 次の意味で最適時間アルゴ

リズムである. すなわち, C_{n-1} の活性化(同時に d_0 の値が決定される)に n ステップを要し, さらに R_a の残りの桁のセットに $\lceil \log_2 n \rceil - 1$ ステップを要する. ■

次の定理を得る.

[定理1] (仮定: A1, A3, A4)

n 台のプロセッサに対し, パイプライン・アドレス設定法 (Ver. 1) は, $n-1 + \lceil \log_2 n \rceil$ ステップの最適時間でアドレス設定を完了する.

次に A1, A3 を仮定したパイプライン・アドレス設定法 (Ver. 2) を示す.

[パイプライン・アドレス設定法 (Ver. 2)]

(アルゴリズムと時間計算量)

アドレス設定の仕方は, 基本的には前定理と同様で, $\lceil \log_2 n \rceil$ 個の波により最下位の桁から順次セットされる. 本定理では A4 を仮定しないため, C_0 は $\lceil \log_2 n \rceil$ 個の波数を計数できない.

波 $w_j (0 \leq j \leq \lceil \log_2 n \rceil - 1)$ は, 各セルの d_j をセットするとともに, 次の波 w_{j+1} をある決められたセル上に生成する機能をもつ. 決められたセルとは, w_j が生成後最初に d_j の値を0にセットするセルである. w_j は各セル上ですでに設定されている d_{j-1} の値を参考にしながら d_j の値を決定する. すなわち, 最初の 2^j 個のセルに対して $d_j=1$, 次の 2^j 個のセルに対して $d_j=0$ とセットし, 以後これを右端のセルまで繰り返す.

w_0 は次のように生成される. まず C_0 は $d_0^0(0)=0$

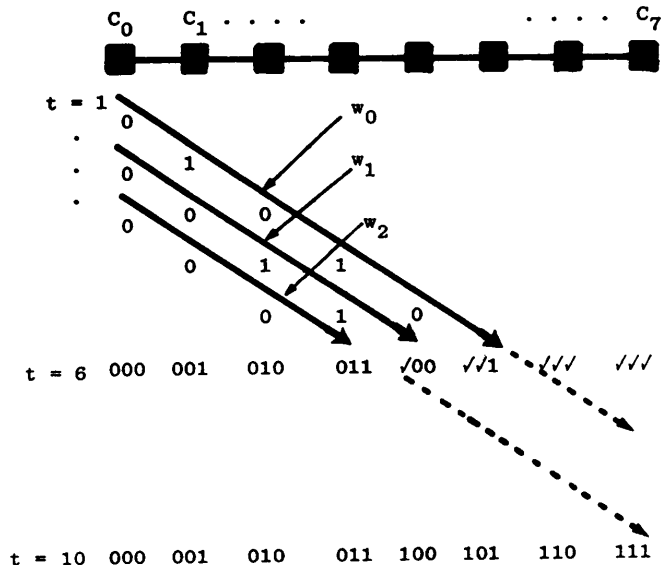


図2 パイプライン・アドレス設定法 (Ver. 1)
Fig. 2 Pipelined addressing scheme (Ver. 1).

とする。 $t=2$ のとき、 C_1 において w_0 が生成される。
 w_0 は以後右方向にスピード1で伝えられ、各セルの d_0 の値を $1, 0, 1, 0, \dots$ と交替的にセットする。 w_0 の生成と同時に、 C_1 において $d_1(1)=1$ とセットされる。

帰納法により次の事実は容易に示せる。すなわち、波 $w_j (0 \leq j \leq \lceil \log_2 n \rceil - 1)$ は、時刻 $t=2^j+j+1$ において C_j 上に生成される。

$n-1+\lceil \log_2 n \rceil$ ステップでアドレス設定は完了し、前定時と同じ理由で最適時間である。図3は、 $n=8$ の場合におけるパイプライン・アドレス設定法 (Ver. 2) を示したものである。

次の定理を得る。

[定理2] (仮定: A1, A3)

n 台のプロセッサに対し、パイプライン・アドレス設定法 (Ver. 2) は、 $n-1+\lceil \log_2 n \rceil$ ステップの最適時間でアドレス設定を完了する。

パイプライン・アドレス設定法 (Ver. 2) では、Ver. 1 と比較して、

	Ver. 1	Ver. 2
C_0 :	$\cdot \checkmark \checkmark 0 \cdot \cdot 0 0 0$	$\cdot \checkmark \checkmark \checkmark \checkmark \checkmark 0$
C_1 :	$\cdot \cdot \checkmark 0 \cdot \cdot 0 0 1$	$\cdot \cdot \checkmark \checkmark \checkmark \checkmark 1$
C_2 :	$\cdot \cdot \checkmark 0 \cdot \cdot 0 1 0$	$\cdot \cdot \checkmark \checkmark \checkmark 1 0$
C_3 :	$\cdot \cdot \checkmark 0 \cdot \cdot 0 1 1$	$\cdot \cdot \checkmark \checkmark \checkmark 1 1$
\vdots	\vdots	\vdots
	$\underbrace{\quad \quad \quad}_{\lceil \log_2 n \rceil}$	\vdots
\cdot		\cdot

に示すように、実際にセットされたアドレス・レジスタの内容の長さが、すべてのセルにおいて $\lceil \log_2 n \rceil$ に正規化されていない。 R_a の正規化は、次に示す2方向の情報伝達を仮定したパイプライン・アドレス設定法 (Ver. 3) でも可能である。

[パイプライン・アドレス設定法 (Ver. 3)]

(アルゴリズムと時間計算量)

波 $w_j (0 \leq j \leq \lceil \log_2 n \rceil - 1)$ の生成の仕方ならびに各波の機能は Ver. 2 と同様である。違うのは、 w_j が生成されると同時に左方向に進むスピード1の波が生成される点である。これらの波は、未設定最下位桁を0にセットする。時間計算量は $n-1+\lceil \log_2 n \rceil$ ステップで最適で

ある。図4に本設定法を示す。

次の定理を得る。

[定理3] (仮定: A2, A3)

n 台のプロセッサに対し、パイプライン・アドレス設定法 (Ver. 3) は、 $n-1+\lceil \log_2 n \rceil$ ステップの最適時間でアドレス設定を完了する。

Ver. 3 の方法では、各セルはアドレス設定の終了

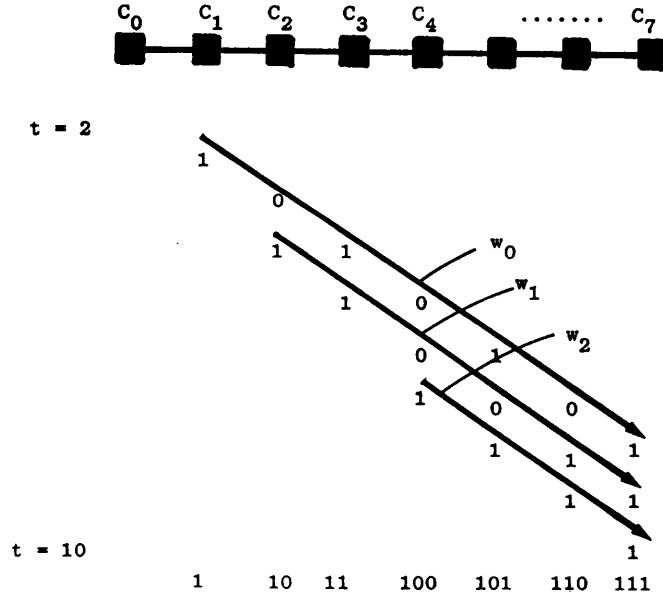


図3 パイプライン・アドレス設定法 (Ver. 2)
 Fig. 3 Pipelined addressing scheme (Ver. 2).

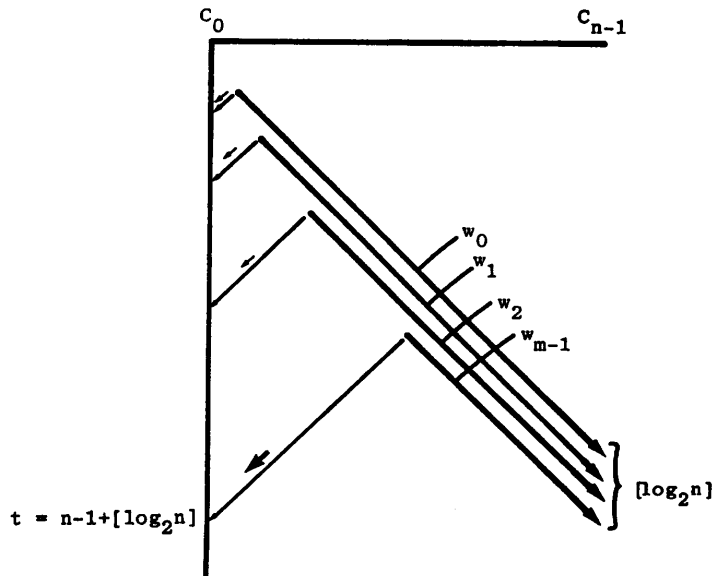


図4 パイプライン・アドレス設定法 (Ver. 3)
 Fig. 4 Pipelined addressing scheme (Ver. 3).

を知ることができない。 w_0 が C_{n-1} に達した後、その反射波を左方向に伝えることにより $2n-1$ ステップですべてのセルにアドレス設定の終了を知らせることが可能である。

3.2 並列2分割・アドレス設定アルゴリズム

A2, A3 を仮定する。次の並列2分割アドレス設定法は、セルラ・オートマトンの同期化手法として知られている一斉射撃アルゴリズム^{2),4),6)} を利用している。

[並列2分割・アドレス設定法]

(アルゴリズムと時間計算量)

本設定法は、Waksman⁴⁾, Balzer⁶⁾ らによる最適時間一斉射撃アルゴリズムに基づく、“將軍”状態セルから発射される速度 $1/(2^j-1)$, $j=1, 2, \dots$ の無限種類のパルスを利用。これらの発生機構については文献^{2),4),6)} を参照。アレイ上における波の伝播の様子を図5に示す。

アドレスは最上位の桁から順次、次の規則に従いセットされる：最初の右方向に進む速度1のパルスを除き、右(左)方向に進むパルスは各セルの未セット最上位桁を $0(1)$ にセットする。一斉射撃と同時に、各セルはアドレス設定を終了する。時間計算量は $2n-1$ ステップ。本設定法の仮定より、 C_0 はセル総数を知る必要がある。それには、まず C_{n-1} の活性化に n ステップ、さらに C_{n-1} から信号が戻ってくるのに $n-1$ ステップ、計 $2n-1$ ステップを要する。したがって本アルゴリズムは最適時間アルゴリズムである。図6は、 $n=8$ の場合における最適時間並列2分割・アドレス設定法を例示したものである。

本設定法は、 $n=2^k$ (k は任意の非負整数) なる n に対してのみ有効であることに注意されたい。

以上より次の定理を得る。

[定理4] (仮定: A2, A3)

n 台のプロセッサに対し、並列2分割・アドレス設定法は、 $2n-1$ ステップの最適時間でアドレス設定を完了する。ただし、 $n=2^k$ (k は任意の非負整数)。

定理3と定理4はともに最適時間アルゴリズムである。前者では、アドレス設定動作の終了事実を各セルが知ることが不能であるが、後者では可能であるという点で相違する。

定理1, 2, 3, 4 に示したアドレス設定法は次の特長をもつ。すなわち、アドレ

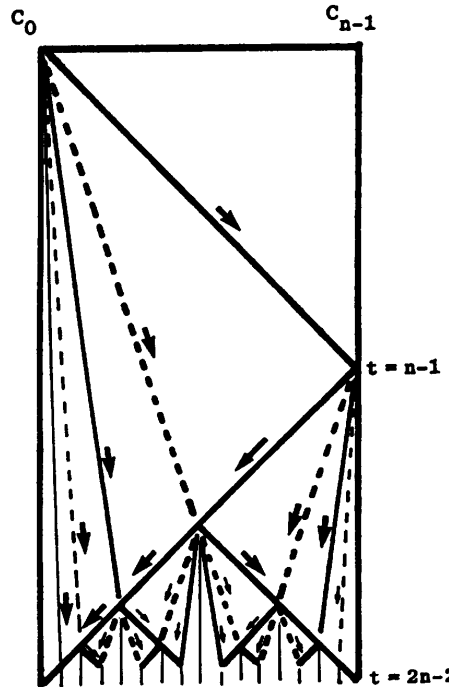


図5 最適時間一斉射撃アルゴリズムにおける波の伝播
Fig. 5 Wave propagation scheme in optimum firing squad synchronization.

ス・レジスタの内容はひとたび記入されると、以後決して消されたり、書き換えられたりすることはない。したがって、アドレス・レジスタの各桁に消去不能という制限を付与しても、これらの設定法は最適時間アルゴリズムである。

表1はこれまでのアドレス設定アルゴリズムをまとめたものである。

本稿でのアドレス設定法は、図7に示す2次元アレイ上に容易に拡張できる。各セルは境界を除き、上、下、左、右の4セルと接続されているものとする。 $C_0, 0$ か

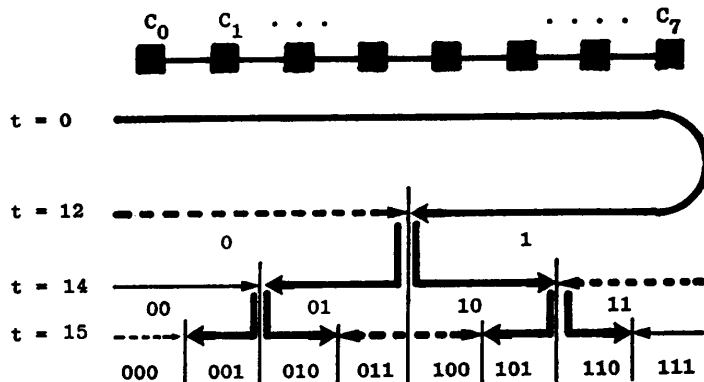


図6 並列2分割・アドレス設定法
Fig. 6 Parallel binary division addressing scheme.

表 1 本稿で考察したアドレス設定アルゴリズムとその並列時間計算量

Table 1 Address setting algorithms and their time complexities.

Addressing scheme	Restrictions	Time complexity for n processors
Pipelined addressing (Ver. 1)	A1, A3, A4 A5 (yes), A6 (yes)	$n-1+\lceil\log_2 n\rceil$ (optimum)
Ver. 2	A1, A3 A5 (no), A6 (no)	$n-1+\lceil\log_2 n\rceil$ (optimum)
Ver. 3	A2, A3 A5 (no), A6 (yes)	$n-1+\lceil\log_2 n\rceil$ (optimum)
Parallel binary division addressing	A2, A3 A5 (yes), A6 (yes)	$2n-1$ (optimum)

Restrictions

A1: One-way information, A2: Two-way information, A3: 1-bit/1-step information transfer, A4: C_0 can count $\lceil\log_2 n\rceil$, A5: Can all cells detect the completion of address setting operation?, A6: Are all addresses normalized in length $\lceil\log_2 n\rceil$?

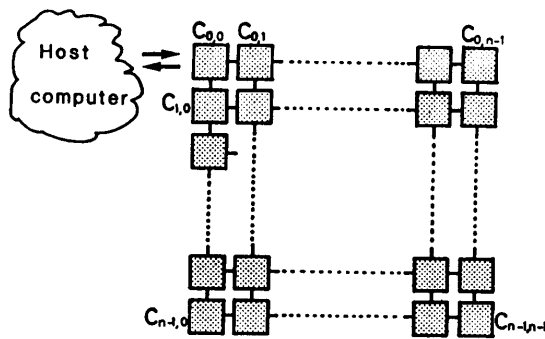


図 7 2次元セルラ計算機

Fig. 7 Two-dimensional cellular computer.

ら最も遠く離れた $C_{n-1,n-1}$ を活性化するのに $2n-1$ ステップ, さらに $C_{n-1,n-1}$ のアドレスを設定するのに $\lceil\log_2 n\rceil$ ステップを要する。したがって, どのような方法を採用しようとも, 2次元アレイのアドレス設定には少なくとも $2n-1+\lceil\log_2 n\rceil$ ステップを要する。ここで各セルは1ステップにつき, 水平, 垂直両方向の二つのアドレスを各1ビット変更可能と仮定する。

まず2次元パイプライン・アドレス設定法を説明する。 $C_{0,0}$ は右ならびに下方向に活性化信号を伝え, 第1行, 第1列上に定理1における C_0 に相当するセルを生成する。 $C_{0,i}$ および $C_{i,0}$ は, $t=i$ 時に活性化される。 $C_{0,i}$ および $C_{i,0}$ の活性化後, それぞれのセルは各列, 各行上でパイプライン・アドレス設定法を実行する。これにより, $2n-1+\lceil\log_2 n\rceil$ ステップの最適時間でアドレス設定を完了する。このときの情報の伝

達は, 2方向であることに注意されたい。仮定 A4 の有無, アドレス設定終了時の検知などの条件を考慮すれば, 2次元アレイ上でも定理 1, 2, 3 に対応した設定法が得られるがここでは省略する。

次に2次元並列2分割アドレス設定法について説明する。1行, 1列上に将軍状態を順次生成し, 各行, 各列上で1次元の並列2分割・アドレス設定法を実行する。これにより $3n-3$ ステップですべてのアドレスが設定される。正方形アレイに対して, $2n-2$ ステップの一斉射撃アルゴリズムが知られているが, アドレス設定には使えない。 $2n-1+\lceil\log_2 n\rceil$ ステップで完了する2次元並列2分割アドレス設定法の存在は明らかでなく興味ある問題である。

4. むすび

最適時間・並列アドレス設定アルゴリズムを2種類, すなわち, パイプライン・アドレス設定法ならびに並列2分割・アドレス設定法を提案した。 n 台のプロセッサに対し, 前者は $n-1+\lceil\log_2 n\rceil$ ステップ, 後者は $2n-1$ ステップの時間計算量をもつ。両手法ともセルラ・ロジックにより記述され, VLSI ベースのマルチ・プロセッサ・システムにおけるアドレス設定に適している。

本稿で提案したアドレス設定法は, 2次元格子状結合をもつアレイプロセッサ上に容易に拡張可能である。しかしながら, 一般的なグラフ結合を仮定したマルチプロセッサの場合には, 本稿での設定法は適用できない。

謝辞 日頃ご指導いただく菅田一博助教授(大阪大学基礎工学部)に, また有用なコメントをいただいた森田憲一博士(大阪大学基礎工学部)ならびに阿曾弘具助教授(名古屋大学工学部)に謝意を表す。本研究の一部は, 文部省科研費総合研究(A)「多オートマトン系の情報処理機能とその応用に関する研究」(代表: 本多波雄), ならびに奨励研究(A) (No. 58780050) より補助をうけた。

参考文献

- 1) 梅尾, 菅田: Systolic Simulation of Synchronous SIMD Parallel Computers, 電子通信学会技術研究報告, Vol. 82, No. 130, AL82-21, pp. 51-60 (1982).
- 2) 小林: オートマトン理論とパズル, 一斉射撃の問題, 数理科学増刊パズル I (1976. 11).
- 3) Kung, H. T.: Why Systolic Architecture?, *IEEE Comput.*, Vol. 15, No. 1, pp. 37-46(1982).

- 4) Waksman, A.: An Optimal Solution to the Firing Squad Synchronization Problem, *Inf. Control.*, Vol. 9, No. 1, pp. 66-78 (1966).
- 5) Rosenfeld, A. and Wu, A. Y.: Reconfigurable Cellular Computers, *Inf. Control*, Vol. 50, No. 1, pp. 64-84 (1981).
- 6) Balzer, R.: An 8-State Minimal Time Solution to the Firing Squad Synchronization Problem, *Inf. Control*, Vol. 10, No. 1, pp. 22-42 (1967).
- 7) Kung, S.-Y., Arun, K. S., Gal-Ezer, R. J. and Bhaskar Rao, D. V.: Wavefront Array Processor; Language, Architecture and Application, *IEEE Trans. Comput.*, Vol. C-31, No. 11, pp.1054-1066 (1982).
- 8) Mead, C. and Conway, L.: *Introduction to VLSI Systems*, Addison Wesley, Reading (1980).
- 9) 梅尾: パイプライン・アドレス設定法, 情報処理学会論文誌, Vol. 24, No. 5, pp. 719-720 (1983).
- 10) 梅尾: シストリック・アレイにおける並列アドレス設定法, 昭和57年度科学研究費補助金総合研究(A)「多オートマトン系の情報処理機能とその応用に関する研究」(代表: 本多波雄) 研究会資料 (1982).

(昭和58年3月11日受付)

(昭和58年7月19日採録)