

BC プロセッサアレイと高並列マトリクス計算†

金田 悠紀夫^{††} 小畑 正貴^{†††} 前川 禎男^{††}

VLSI 技術の急速な進歩にともないシストリックアレイなど高並列計算向きアレイプロセッサが注目をあび多くの研究が行われている。本研究は隣接プロセッサとの通信用の端子の他に BC(ブロードキャスト)端子と呼ぶ特別な端子をもつ演算プロセッサ (BC プロセッサと呼ぶ) で構成される 1次元または 2次元のプロセッサアレイにより行列計算が高並列にしかも効率よく行えることを示す。ここで示す BC 端子は演算プロセッサを共通バスに接続するもので 1本の共通バスラインに多数の BC プロセッサを接続することができる。バスライン上の任意の一つがデータを BC 端子を介してバスラインに出力するとそのデータは他の全プロセッサに伝達 (つまり放送) されることになり、同一データの分配が実現できる。二つの BC 端子をもつ BC プロセッサ $p \times q$ 個を q 行 p 列のマトリクス状に配置して行方向に q 本、列方向に p 本張られた直交格子状の共通バスに接続したプロセッサアレイを想定する。このプロセッサアレイのもつデータの放送機能を有効に利用することにより行列とベクトルの乗算、行列と行列の乗算、三角線形方程式の計算、行列の LU 分解がシストリックアレイに比して $1/2 \sim 1/3$ のステップで実現できることを示す。また線形方程式のガウス消去や修正コレスキー法を用いた計算が効率よく実現できることも示す。

1. はじめに

Kungらによって提案されたシストリックアレイ^{1),2)} は 1次元または 2次元のプロセッサアレイで隣接プロセッサが相互に接続された形をしている。単純で規則的なデータの流れを受け入れそれにパイプライン制御されたシストリックアルゴリズムを適用していくことにより目的の計算を高並列に実行することができる。

われわれは隣接プロセッサとの結合端子に加えて新たに BC(ブロードキャスト) 端子と呼ぶ特別な端子を付加した演算プロセッサ (以後 BC プロセッサと呼ぶ) を提案し、BC プロセッサから構成される 1次元または 2次元のプロセッサアレイがシストリックアレイよりもすぐれた高並列計算能力を有していることを示す。

2. BC プロセッサ

BC プロセッサはシストリックアレイにおける演算プロセッサと同様に単純な構造をもったセルであるが異なる点は隣接プロセッサへの接続端子のほかに一つまたは二つの BC 端子をもつことである (図 1)。BC 端子は特別な端子で共通バスに接続されるように設計されており、1本の共通バスに複数個の BC プロセッサが BC 端子を介して接続されることになる。BC 端

子に出力されたデータは共通バスを介して他の BC プロセッサに伝送され各 BC プロセッサはそのデータを入力することができる。したがって l 台の BC プロセッサ ($P_1 \sim P_l$) が一本の共通バスに BC 端子を介して接続されている場合、任意の $P_i (1 \leq i \leq l)$ または外部端子から共通バスにデータが出力されると全 $P_j (1 \leq j \leq l)$ は同時にそのデータを入力することができる。したがって全 P_j に対するデータの放送が行われることになる。この場合、二つ以上の BC プロセッサや外部端子と一つ以上の BC プロセッサが同時に共通バス上にデータを出力すると競合が発生することになるので禁止されているとする。BC プロセッサに二つの BC 端子がある場合には 2本の共通バスにそれぞれ接続することができる (図 2)。

3. 行列・ベクトル乗算

行列 $A=(a_{ij})$ とベクトル $x=(x_1, x_2, \dots, x_n)^T$ の乗算問題 ($y=Ax$) を取り上げる。 A を $n \times n$ の帯行列としその帯幅を $w=p+q-1$ とする。積 $y=(y_1, \dots, y_n)^T$ の要素は

$$y_i = \sum_{k=\max(1, i-q+1)}^{\min(n, i+p-1)} a_{ik} x_k$$

となる。この場合 y_i の値は図 3 に示す構成の線状 BC プロセッサアレイ (w 個の BC プロセッサより構成) を用いて並行計算を行うことができる。ここで用いる BC プロセッサは 1 BC 端子、2 入力端子、1 出力端子、1 内部レジスタ R をもち BC 端子入力 (r_B)

† BC Processor Array and Highly Parallel Matrix Computation by YUKIO KANEDA, MASAKI KOHATA and SADAŌ MAEKAWA (Systems Engineering, Kobe University).

†† 神戸大学工学部システム工学科

††† 神戸大学大学院自然科学研究科システム科学専攻

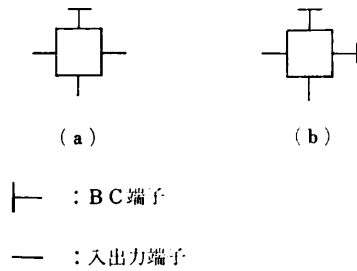


図1 BC プロセッサの例
Fig.1 Examples of BC processor.

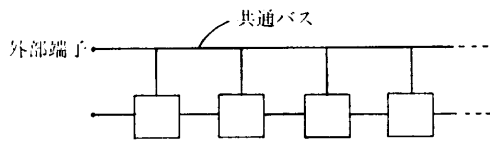


図2 1次元 BC プロセッサアレイの例
Fig.2 An example of a linear BC processor array.

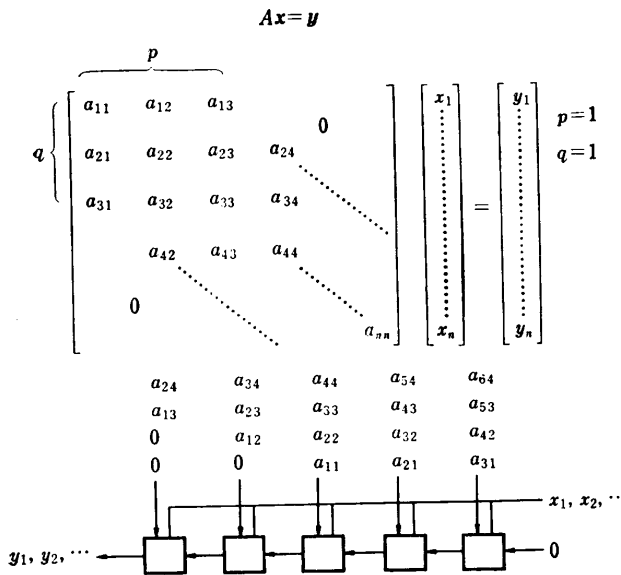


図3 行列とベクトルの乗算の場合
Fig.3 Multiplication of a vector and matrix.

と上方入力端子入力 (r_u) を用いて $R \leftarrow R + r_B \cdot r_u$ の計算を行う機能をもっている。

右端の端子には常に 0 が入力される。計算は $k=1 \sim n$ に関して以下の手順で進められる。

図3に示すように第1ステップから BC 端子には x_1, x_2, \dots, x_n の値を上入力端子群には帯内の列ベクトル a_1, a_2, \dots, a_n を列単位に入力していく。各 BC

プロセッサは1ステップごとに以下のサブステップの演算を行う。

1. BC 端子から x_k , 2入力端子から a_{ij} と右 BC プロセッサの演算結果を読み込み r_U, r_B, R とする。
2. $R \leftarrow R + r_B \cdot r_U$ の計算を行う。

p ステップ目から順に左端の端子に y_1, y_2, \dots, y_n が出力されてきて $n+p-1$ ステップ目に y_n が出力される。上述の1ステップの動作時間を基本単位時間とするとシストリックアレイにおいては3入力端子, 2出力端子をもつプロセッサ w 台で $2n+w$ の計算時間を必要とするが本 BC プロセッサアレイではほぼ半分の $n+p-1$ の計算時間で実現できる。

4. 2次元 BC プロセッサアレイによる行列乗算

二つの $n \times n$ 行列の乗算について考える。 $A=(a_{ij})$ と $B=(b_{ij})$ の積 $C=(c_{ij})$ は次式で計算される。

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} \quad (1 \leq i \leq n, 1 \leq j \leq n)$$

A と B がそれぞれ帯幅を $w_1=p_1+q_1-1$, $w_2=p_2+q_2-1$ とする帯行列とすると $w_1 \cdot w_2$ のマトリクス形に接続した BC プロセッサアレイで並列計算が可能となる。用いられる BC プロセッサは一つの内部レジスタ R と 2 BC 端子, 1入力端子, 1出力端子をもつもので(図4(a))各 BC プロセッサは図4(c)に示すように行方向と列方向の共通バスと左上方および右下方の隣接プロセッサに接続されている。アレイの左端の BC 端子群から a_i 列を1列ごとに列 BC 端子群から b_j 行を1行ごとに ($i=1 \sim n, j=1 \sim n$) 入力していく。計算ステップは以下のサブステップより実現される。

1. 右下端子から右下隣接 BC プロセッサの出力値行 BC 端子から a_i 行の要素値, 列 BC 端子から b_j 列の要素値を読み込み R_1, r_1, r_2 とする。
2. $R \leftarrow R + r_1 \cdot r_2$ の計算を行う。

第 $\min(p_1, q_2)$ ステップ目から図4のように左端および上端の出力端子群から c_{ij} が出力されてくる。

合計 $n + \min(p_1, q_2)$ 時間後にマトリクス C が得られる。シストリックアレイにおいては $3n + \min(w_1, w_2)$ 時間が必要となる。

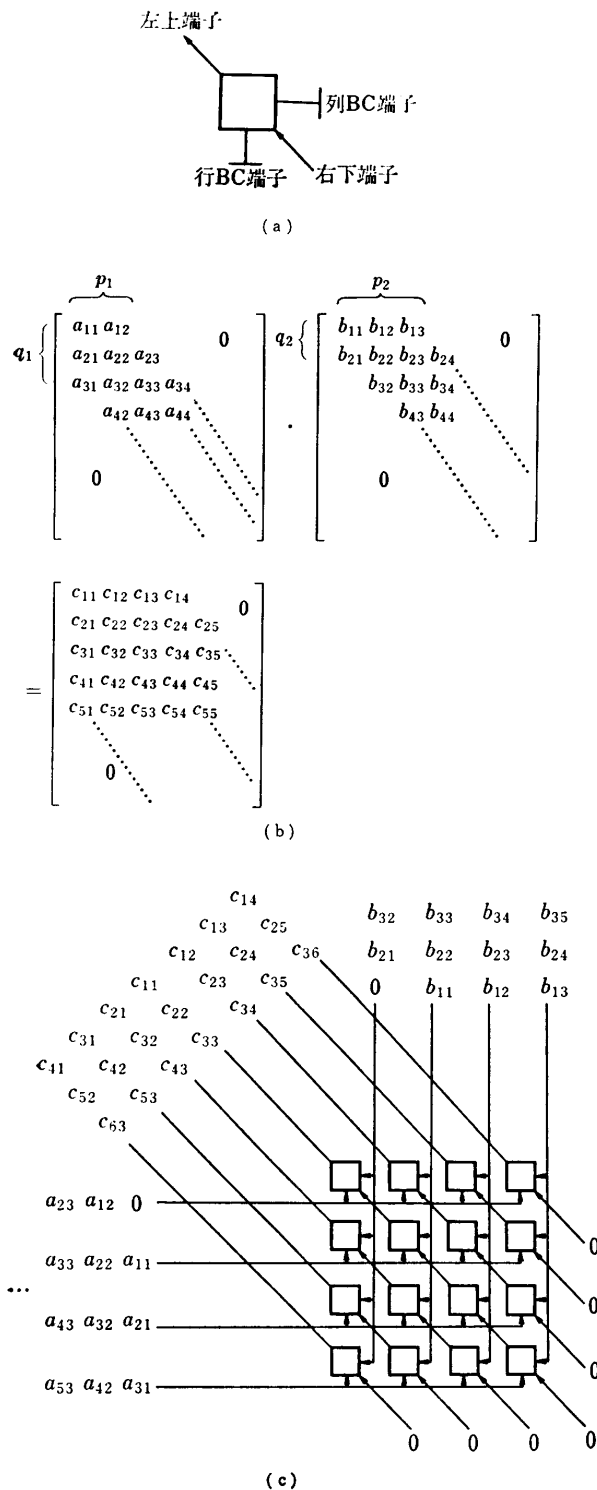


図4 行列乗算の場合
 (a) BC プロセッサ, (b) 行列の積, (c) BC プロセッサ・アレイ
 Fig. 4 Multiplication of matrices A and B.

5. 行列の LU 分解

LU 分解は行列 A を $A=LU$ なる下三角行列 L と上三角行列 U に分解する計算で A の逆行列を求めたり線形方程式 $Ax=B$ を解く過程で用いる. ピボット選択なしで計算できるとすると, $L=(l_{ij}), U=(u_{ij})$ は $k=1 \sim n$ に対して次の漸化式で計算できる²⁾.

$$a_{ij}^{(1)} = a_{ij}$$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} \cdot u_{kj}$$

$$l_{ik} = \begin{cases} 0 & i < k \text{ の場合} \\ 1 & i = k \text{ " } \\ a_{ik}^{(k)} \cdot u_{kk}^{-1} & i > k \text{ " } \end{cases}$$

$$u_{kj} = \begin{cases} 0 & k > j \text{ " } \\ x_{kj}^{(k)} & k \leq j \text{ " } \end{cases}$$

A を帯幅 $w=p+q-1$ の行列であるとする計算は図5に示される 2 BC 端子, 1 入力, 1 出力端子をもつ BC プロセッサ $p \cdot q$ 台から構成されるプロセッサアレイで実現される.

行列 A の帯の部分には図5に示すように右斜下方から入力端子群を通して入力されてくる. $\min(p \cdot q)$ ステップ後にプロセッサ P_{ij} に $a_{ij} (1 \leq i \leq q, 1 \leq j \leq p)$ が伝搬してくる.

以後計算ステップは $k=1 \sim n$ に関して以下の一連のサブステップにより進められる.

1. プロセッサアレイの第1行が

$$u_{kj} = a_{kj} \quad (k \leq j \leq k+p-1)$$

とし u_{kj} をそれぞれの上出力端子に出力するとともに列方向の BC 端子に出力し, u_{kj} の列方向への放送を行う. この際 P_{11} は特別のプロセッサで u_{kk} の逆数 u_{kk}^{-1} を計算し列方向に放送する.

2. プロセッサアレイの第1列が u_{kk}^{-1} を列 BC 端子から読み込んで

$$l_{ik} = \begin{cases} 1 & (i=k) \\ a_{ik}^{(k)} \cdot u_{kk}^{-1} & (k \leq i \leq k+p-1) \end{cases}$$

の計算を行う. 求めた l_{ik} を左出力端子から出力するとともに行方向の BC 端子に出力し行方向に放送する. この際 P_{11} は 1 を左出力端子に出力する.

3. プロセッサアレイの第1行と第1列を除いた全プロセッサが

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} + l_{ik}(-u_{kj})$$

の計算を行う. l_{ik} は行 BC 端子から, u_{kj} は列 BC 端子から入力する.

4. $a_{ij}^{(k+1)}$ は左上方の出力端子に出力された左上方の隣接プロセッサへ伝搬される.

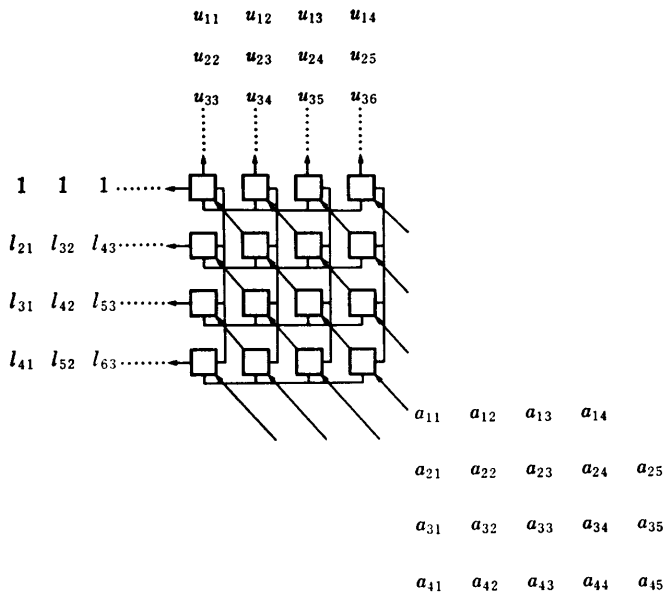


図5 行列のLU分解
Fig. 5 LU decomposition of a matrix A.

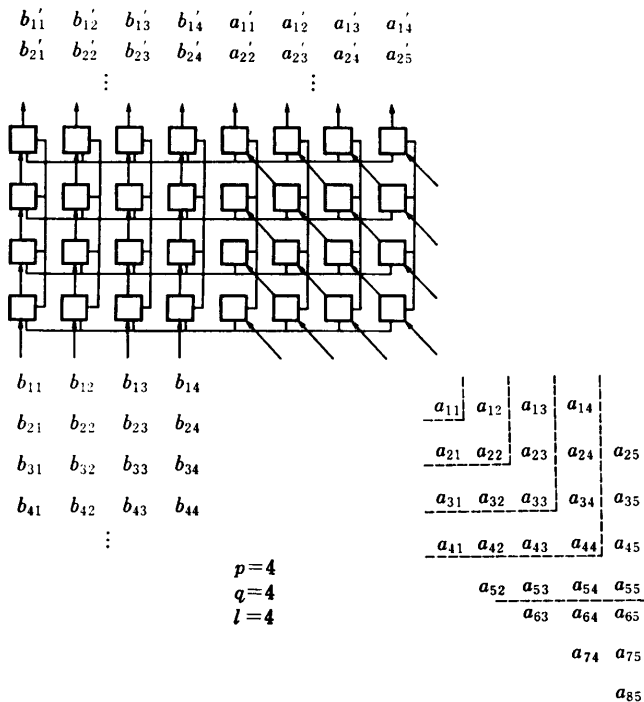


図6 ガウス消去における前進消去過程
Fig. 6 Forward elimination of a Gaussian elimination.

第1, 第2行および第1, 第2列のプロセッサの能力を強化して他のプロセッサがサブステップ 3, 4 の演算を行う時間内にサブステップ 3, 4, 1, 2を行うようにすれば1ステップの実行時間を短縮することが可能となる。

全体として $n + \min(p, q)$ 時間で LU 分解が完了する。シストリックアレイにおいては $3n + \min(p, q)$ 時間を要する。

6. ガウス消去法と修正コレスキー分解

n 元連立一次方程式 $Ax=B$ のガウス消去法および修正コレスキー分解法による計算をとりあげる。

6.1 ガウス消去法

ガウス消去法においては前進消去と後退代入の二つの計算過程が存在する。後退代入は LU 分解や後述する修正コレスキー法による計算の場合にも共通となる計算過程であるから7章で論じることとしここでは前進消去についてのみ論じる。前進消去は $Ax=B$ を変形して $A'x=B'$ の形にするもので A' は右上三角行列の形となる。 A を $n \times n$ のサイズで帯幅 $w = p + q - 1$ の帯行列とし B を $n \times l$ の密行列とする。

図6にBCプロセッサアレイによるガウス消去実行の例が示してある。計算は図6に示すように B に関しては $P_{p1} \sim P_{p1}$ の入力端子に下方から入力された上方に向けて各ステップ1行ずつ上方に伝搬させいく。行列 A のほうは帯部分が $P_{p1+1} \sim P_{p1+q}$ および $P_{1+q} \sim P_{p-1+q}$ の入力端子に図6に示すように1ステップ1区画(点線で区切られた区画)ずつ入力され斜上方に伝搬されていく。計算は q ステップまで伝搬を行ってから $k=1 \sim n$ に対して以下の手順で実行される。

1. P_{1+1} が a_{kk}^{-1} の値を行 BC 端子に出力し、行方向に放送する。
2. プロセッサアレイの第1行が
 $a_{kj} = a_{kj} / a_{kk} \quad (k+1 \leq j \leq k+q)$
 $b_{kj} = b_{kj} / a_{kk} \quad (1 \leq j \leq l)$
 の計算を行い a_{kj}, b_{kj} の値を列 BC 端子に出力して列方向に放送する。
3. 各プロセッサが a_{ij} および b_{ij} の更進

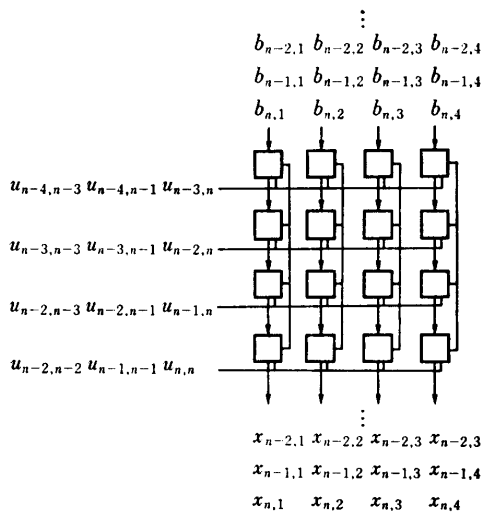


図8 三角線形方程式の計算

Fig.8 Computation of triangular linear equations.

いく。 p ステップ後に $b_{n-i+1,j}$ ($1 \leq i \leq p$, $1 \leq j \leq l$) が $P_{p-i+1,j}$ に到達する。以後計算は以下のように $k=1 \sim n$ に対して実行されていく。

1. 行 BC 端子群に U_{n-k+1} 列を入力する。
2. 第 p 行のプロセッサが

$$x_{n-k+1,j} = b_{n-k+1,j} / u_{n-k+1,n-k+1} \quad (1 \leq j \leq l)$$

の計算を行い $x_{n-k+1,j}$ を下方端子に出力するとともに列 BC 端子を介して列方向に放送する。

3. $P_{i,j}$ ($1 \leq i \leq p-1$, $1 \leq j \leq l$) が

$$b_{n-k-i+1,j} = b_{n-k-i+1,j} - u_{n-k-i+1,n-k+1} \cdot x_{n-k+1,j}$$

の計算を行う。

4. 新しく計算した $b_{n-k-i+1,j}$ を下方に 1 行伝搬させる。

初期化に p, x 群計算に n の合計 $n+p$ 時間の計算で解が得られる。シストリックアレイでは $2n+p$ 時間要する。

この場合も第 1 列、第 p 行、第 $p-1$ 行のプロセッサの能力を強化して他のプロセッサが 3, 4. のサブステップを実行する時間内に 3, 4, 1, 2 のサブステップを実行するようにすれば計算時間の短縮が可能となる。

8. 結 論

チップ当たり 10 万ゲートを越える VLSI が実現してくるとその応用として超高並列計算の分野が重要な目標となってくる。超高並列計算は $10^2 \sim 10^5$ オーダの数の演算プロセッサによってベクトルや行列など 1 次元または 2 次元に配置されているデータに並列に演算を加えていくことにより目的の計算を行うもので並列

計算アルゴリズムと演算プロセッサの結合方式がきわめて重要で広く研究が行われている。Kung らが提案しているシストリックアレイはその代表的なものできわめて VLSI に適した性質をもっており高い評価を受けている。本論文で提案する BC プロセッサアレイはシストリックアレイと同様に VLSI 向きの構成をしているのみならずシストリックアレイに比して以下の点ですぐれている。

1. シストリックアレイではベクトルと行列の積、三角線形計算に約 $2n$ ステップ、行列の乗算、LU 分解に約 $3n$ ステップ要するのに比して、BC プロセッサアレイではいずれの場合も約 n ステップで実現でき、また演算プロセッサの稼働率が高い。
2. 放送機能を有するためガウス消去法や修正コレスキー法のアルゴリズムが効率よく実現できる。
3. シストリックアレイでは伝搬してきたデータを保持しておくための内部レジスタが必要となるが、BC 端子から放送されるデータは共通バスに保持されているので、BC プロセッサ内にデータを保持しておくための内部レジスタを省くことが可能となる。

今後の問題としてはより多くの計算アルゴリズムに対して BC プロセッサアレイの有効性を調べることが上げられる。

参 考 文 献

- 1) 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).
- 2) Kung, H. T.: *The Structure of Parallel Algorithms, Advances in Computers*, Vol. 19, pp. 65-112, Academic Press, New York (1980).
- 3) 高橋: 並列処理のためのプロセッサ結合方式, 情報処理, Vol. 23, No. 3, pp. 201-209 (1982).
- 4) 戸川: マトリクスの数値計算, オーム社, 東京 (1971).
- 5) 金田, 小畑, 前川: マトリクスブロードキャストメモリ結合形並列計算機システムによる連立一次方程式の並列計算法, 情報処理全国大会 (昭 57 前期), pp. 895-896 (1982).
- 6) 金田: 並列処理システムによる連立一次方程式と楕円形偏微分方程式の数値計算法, 情報処理, Vol. 16, No. 2, pp. 122-129 (1975).
- 7) 金田: 並列処理システムによる線形計画計算と実対称行列の三重対角化計算, 情報処理, Vol. 16, No. 1, pp. 39-45 (1978).
- 8) 金田: 環状結合型超多重プロセッサシステムに

- よる大次元連立一次方程式の並列計算, 情報処理学会論文誌, Vol. 21, No. 5, pp. 402-406 (1980).
- 9) Kaneda, Y. and Kohata, M.: Highly Parallel Computing of Linear Equations on the Matrix-Broadcast-Memory Connected Array Processor System, 10 IMACS (1982).
- 10) 金田, 小畑, 前川: マトリクスブロードキャス

トメモリ結合形並列計算機による n 元連立一次方程式の $O(n)$ 時間計算, 情報処理学会論文誌, Vol. 23, No. 5, pp. 570-575 (1982).

(昭和 57 年 6 月 7 日受付)

(昭和 57 年 9 月 6 日採録)