

メモリ型並列計算モデルとその計算能力†

—スーパーコンピュータへの第3のアプローチ—

高木直史^{††} 武永康彦^{††} 矢島脩三^{††}

高い集積度の得られるメモリに若干の機能を付加した並列計算機構を考え、これを並列計算モデルとして形式化し、その計算能力を明らかにすることは、実用的にも、理論的にも興味深かつ重要であると考えられる。本論文では、スーパーコンピュータへの新しいアプローチとして、メモリ型並列計算モデル FRAM を提案し、その計算能力について考察する。FRAM は、通常の逐次計算機のモデルであるランダムアクセス機械に、アドレスの部分一致による複数のメモリセルへの並列アクセスが可能な機能メモリを付加したものである。FRAM において多項式時間で解くことのできる問題のクラスは、NP 問題に対する神託（オラクル）を持つ神託付きチューリング機械により多項式時間で解くことのできる問題のクラス A_1^P に等しい。したがって、工学の種々の分野においてしばしば現れ、逐次計算機で多項式時間で解くことは不可能であろうと考えられている組合せ問題や組合せ最適化問題の多くを、FRAM によって多項式時間で解くことができる。FRAM は実現性が高く、実用的に興味深いばかりでなく、新しい並列計算モデルとして、理論的にも興味深い。

1. はじめに

工学の種々の分野においてしばしば現れる組合せ問題や組合せ最適化問題は、その多くが NP（非決定性多項式時間）完全あるいは NP 困難な問題であり、計算量は問題の大きさ（サイズ）に対して指数関数的に増大すると考えられている。近年、増大する大規模・高速計算の要求に応えるべく、スーパーコンピュータの研究・開発が行われており、並列パイプライン方式のベクトル計算機が実用化され、また、並列プロセサ方式の並列計算機の研究も盛んである。しかし、ベクトル計算機はあくまでも逐次計算機であり、大きなサイズの組合せ問題を解くには膨大な計算時間を要する。また、並列計算機を用いても、膨大な数のプロセサがなければ、実用的な計算時間で解くことのできる問題のサイズはそれほど大きくならない。

本論文では、上記2種の方式とは異なる、スーパーコンピュータへの第3のアプローチとして、高集積化が可能なメモリ型の並列計算機構を考え、これを新たな並列計算モデルとして形式化し、その計算能力を明らかにする。

近年の半導体メモリの高集積化には目を見張るものがある。メモリはもちろんデータを記憶するためのものであるが、メモリに若干の機能を付加することにより、種々の計算を行える可能性がある。メモリはプロ

セサに比べ高集積化が可能であり、機能メモリを用いることにより、従来の並列プロセサ方式では実現が不可能と考えられるような大きな並列度を得ることができる可能性がある。

本論文で提案するメモリ型並列計算モデル FRAM (Random Access Machine with a Functional Memory) は、通常の逐次計算機のモデルであるランダムアクセス機械¹⁾に、新しい機能メモリを付加したものである。この機能メモリでは、アドレスの部分一致による複数のメモリセルへの並列アクセスが可能である。本論文では、この計算モデルにおいて多項式時間で解くことのできる問題のクラスが、NP 問題に対する神託（オラクル）を持つ神託付きチューリング機械²⁾により多項式時間で解くことのできる問題のクラス A_1^P に等しいことを示す。 A_1^P は、逐次計算機で多項式時間で解くことが不可能ではないかと考えられている NP 完全問題を1ステップで解くことができると仮定した場合に、多項式時間で解くことのできる問題のクラスであり、NP や co-NP を包含する³⁾。

メモリに若干の機能を付加した機能メモリの研究は30年以上前から行われており、連想メモリ (CAM: Content Addressable Memory) 等、種々の機能メモリが提案され、そのいくつかは実際に開発されている³⁾。近年、かなり大規模で高機能な CAM が実現されるようになってきており⁴⁾、CAM を用いた高速計算が注目されつつある^{5), 6)}。我々も、論理型言語の実行における基本的操作である単一化操作を CAM を用いて効率よく解く手法を示している⁷⁾。この中で、CAM において初期データとしてアドレスを記憶する

† A Memory-Type Parallel Computation Model and Its Computational Power—Yet Another Approach to Supercomputing— by NAOFUMI TAKAGI, YASUHIKO TAKENAGA and SHUZO YAJIMA (Department of Information Science, Faculty of Engineering, Kyoto University).

†† 京都大学工学部情報工学科

という考えを示している。さらに、この考えを発展させ、アドレスに対する部分一致検索により、代表的な NP 完全問題である CNF-SAT を多項式時間で解くアルゴリズムを示している⁸⁾。安浦らも、アドレスに対する部分一致検索と CAM 本来の機能により、種々の組合せ問題を効率よく解く、機能メモリアーキテクチャを提案している⁹⁾。しかし、本論文のように、機能メモリに基づく並列計算モデルを形式化し、その計算能力を明らかにした研究は見あたらない。

本論文で考える機能メモリは、通常のメモリのアドレスデコーダやメモリセルに若干の工夫を施すこと等により実現可能であり、実現性が高いと考えられる。現在の集積回路技術で数 M ビットの容量を持つ LSI が実現可能で、さらに、現在の実装技術で数 G ビットの容量を持つシステムが実現できると考えられる。また、本論文の機能メモリは、CAM においてあらかじめデータとしてアドレスを記憶した連想 ROM としても実現可能である。

本論文では、2章でメモリ型並列計算モデル FRAM を提案する。3章で、FRAM の計算能力について考察し、多項式時間で解くことのできる問題のクラスを明らかにする。4章では FRAM の実現性について述べる。

2. メモリ型並列計算モデル—FRAM—

本論文で提案するメモリ型並列計算モデル FRAM は、通常のランダムアクセス機械 (RAM: Random Access Machine)¹⁾ に、アドレスの部分一致による複数のデータへの並列アクセスが可能な機能メモリを付加したものである。

FRAM は、図 1 に示すように、RAM と、機能メモリ、検索結果レジスタから構成される。RAM は、一本の読み取り専用の入力テープ、無限個のレジスタ、プログラムカウンタ、有限のプログラムからなる。機能メモリは、図 2 に示すように、無限個のワー

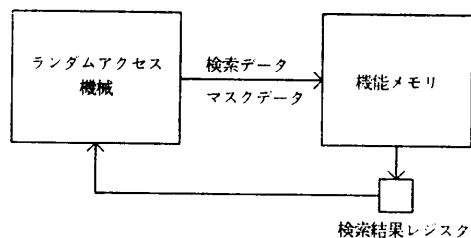


図 1 FRAM の構成
Fig. 1 The scheme of the FRAM.

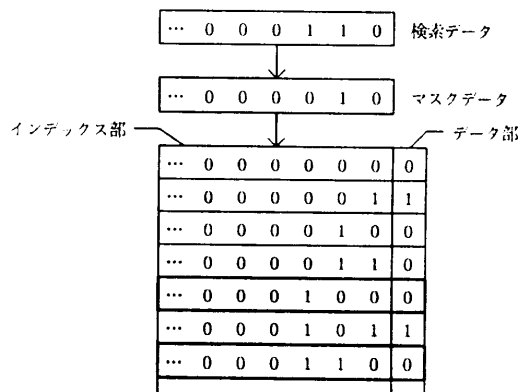


図 2 機能メモリによる検索
Fig. 2 Search by the functional memory.

表 1 FRAM の命令セット
Table 1 Instruction set of FRAM.

命令	機能
LOAD operand	指示された操作を アキュムレータで実行
STORE operand	
ADD operand	
SUB operand	
JUMP label	プログラムカウンタに label をセット
JZERO label	
JGTZ label	
SEARCH 0 oper 1, oper 2	機能メモリを用いた命令 (本文参照)
WRITE 1 oper 1, oper 2	
READ operand	現在の入力記号を読み テープヘッドを1つ移動
ACCEPT	停止して受理
REJECT	停止して非受理

ドからなり、各ワードはインデックス部と1ビットのデータ部からなる。インデックス部には、先頭のワードから順に、0, 1, 2, ... の2進数が書き込まれており、書き換えはできない。インデックス部に書き込まれた数をアドレスと呼ぶ。検索データとマスクデータを与えることにより、全ワードのインデックス部に対する部分一致検索を行うことができる。マスクデータが0の部分のみに、検索データとの一致検索が行われる。データ部は検索結果により書き換えが可能である。図 2 に示した例では、太枠で囲まれたワードが一致している。

検索結果レジスタは、1ビットのフラグで、検索命令の結果により0または1が自動的に記入される。

FRAM の命令セットは、表 1 のとおりである。

SEARCH 0, WRITE 1 の2つを除いた命令は、通常の RAM の命令と同じものである¹⁾。ただし、オペ

ランドとして、検索結果レジスタの指定も可能である。プログラムはレジスタとは別の領域にあり、書き換えは認めない。

SEARCH 0, WRITE 1 が機能メモリを用いた命令である。第1オペランドが検索データ、第2オペランドがマスクデータである。それぞれのオペランドの内容を2進数とみなした列がデータとなり、インデックス部に対する部分一致検索を行う。ただし、データの上位には0が補われるものとする。SEARCH 0 命令は、検索の結果一致したワードの中に、データ部が0のものが存在するかどうかを調べる。存在すれば1、存在しなければ0が検索結果レジスタに自動的にセットされる。データ部の内容は変更しない。WRITE 1 命令は、一致したワードのデータ部に1を書き込む。一致しなかったワードのデータ部の内容は変更しない。検索データ、マスクデータとして n 桁の2進数を与えれば、下位から n ビット目より上位はすべて0になり、実質上アドレスが0から 2^n-1 までのワードのデータ部に対して操作が行われる。なお、初期状態において、データ部はすべて0であるとする。

FRAM 上での時間計算量を、通常の RAM と同じく実行した命令の数で定義する。

3. FRAM の能力

3.1 CNF-SAT を解く多項式時間アルゴリズム

FRAM では、問題のサイズの多項式オーダーの長さのデータを作れば、機能メモリ中の、問題のサイズの指数オーダー個のワードに対して、定数時間で操作を行うことが可能である。したがって、逐次計算機で指数時間を要すると考えられている問題が、FRAM により多項式時間で解ける場合がある。ここでは、NP 完全問題である CNF-SAT (和積形ブール式の充足可能性判定問題) を多項式時間で解くアルゴリズムを示す⁹⁾。

[k 変数 CNF-SAT を解く多項式時間アルゴリズム]

Step 1:

すべての和項について以下の処理を行う。

- (1) 和項に現れない変数に対応するビットのみが1で他は0であるようなマスクデータを作る。また、和項に正のリテラルが現れる変数に対応するビットが0、負のリテラルが現れる変数に対応するビットが1であるような検索データを作る。
- (2) (1)で作ったマスクデータと検索データによ

り、各ワードのインデックス部に対して部分一致検索を行い、一致した全ワードのデータ部に1を書き込む。

Step 2:

- (1) 下位から k ビット目までが1であるようなマスクデータを作る。
- (2) (1)で作ったマスクデータと任意の検索データにより、各ワードのインデックス部に対して部分一致検索を行い、一致したワード (アドレスが $0 \sim 2^k-1$ の全ワード) の中にデータ部が0のものがあるかどうか調べる。
- (3) (2)でデータ部が0のワードがあれば、CNF-SAT の答えは yes である。 □

このアルゴリズムでは、機能メモリの0番地から 2^k-1 番地までのワードを使用している。検索データ、マスクデータともに、下位から k ビット目より上位は常に0であると考え、Step 1 で与えるデータは、例えば5変数の CNF-SAT において $(x_5 + \overline{x_2} + x_1)$ という和項に対しては、検索データは 0-10、マスクデータは 01100 となる (-は0でも1でもよい)。このときアドレスが一致したワードは、和項を0にする割り当てに対応しているため、そのデータ部に1を書き込む。これは WRITE 1 命令により行える。また、データ部が0のワードの検索は SEARCH 0 命令により行える。

このアルゴリズムを実行するために必要な時間を評価する。和積形ブール式中の和項の数を m とする。Step 1 では、マスクデータと検索データを作るのにそれぞれ $O(k)$ の時間がかかり、部分一致検索を行う回数は和項の数に等しい。Step 2 でも、マスクデータを作るのに $O(k)$ の時間がかかる。したがって、 k 変数の CNF-SAT は計算時間 $O(mk)$ で解くことができる。

Step 2 でデータ部が0のワードが存在しなければ、ブール式が充足不能であることがわかる。

以上より、次の補題が得られる。

[補題1]

- (1) NP に属する問題はすべて FRAM により多項式時間で解くことができる。
- (2) co-NP に属する問題はすべて FRAM により多項式時間で解くことができる。

<証明>

(1)は、NP 問題はすべて、RAM で多項式時間で CNF-SAT に変換でき、CNF-SAT は上記のアルゴ

リズムで多項式時間で解くことができることから、明らかである。(2)は、与えられた co-NP 問題の補問題である NP 問題を解けばよい。FRAM では、その解が yes か no かがわかるので、補問題である NP 問題の解が no であれば元の co-NP 問題の解は yes である。□

3.2 多項式時間で解くことのできる問題のクラス

本節では、FRAM によって多項式時間で解くことのできる問題のクラスが、NP 問題に対する神託 (オラクル) を持つ神託付きチューリング機械 (OTM: Oracle Turing Machine) により多項式時間で解くことのできるクラス Δ_2^P に等しいことを示す。

[定理 1]

FRAM によって多項式時間で解くことのできる問題のクラスは、 Δ_2^P に等しい。

<証明>

NP 問題に対する神託を持つ OTM と FRAM を相互に模倣することにより証明する。基本的なアイデアは、機能メモリによる一連の検索を、CNF-SAT に対する神託に対応させることである。

まず、NP 問題に対する神託を持つ多項式時間限定 OTM を、FRAM によって模倣する。

FRAM の入力テープは、模倣する OTM の入力テープと同じにする。OTM が入力テープの内容を読んでヘッドを動かすときに、FRAM では READ 命令を実行すればよい。FRAM の最初の 5 つのレジスタには、OTM の状態、ワークテープのヘッドの位置、神託テープのヘッドの位置、テープの模倣に使っている FRAM のレジスタの番号の最大値、それまでに模倣に用いた機能メモリのアドレスのビット数を記憶しておく。

OTM の 2 本のテープ (ワークテープと神託テープ) の内容は、6 つ目以降のレジスタで記憶する。テープは各々両無限であるが、各々のテープを折り畳み、テープ中の番号の絶対値の等しいマスに FRAM の 1 つのレジスタに記憶する。すなわち、1 つのレジスタで OTM の 2 本のテープの各々 2 つのマス、計 4 つのマスの内容を記憶する。OTM のテープ記号の数を p とすると、 $\lceil \log_2 p \rceil$ ビットで 1 つのマスの内容を表すことができるので、レジスタの下位 4 $\lceil \log_2 p \rceil$ ビットで 4 つのマスの内容を表す。模倣の際、ヘッド位置はマスの番号の絶対値で表し、正負は状態に組み込むものとする。テープを模倣するレジスタの内容は p^4 (定数) 通りの可能性しかない。レジスタから 1 つ

のマスを取り出すには、 p^4 個のどれにあたるかを定数回の比較 (減算と条件付き分岐) で調べればよい。テープ記号の書き換えは、そのレジスタの内容に適当な加減算を行えばよい。

レジスタ上で各種の演算を行う場合には、模倣している OTM のテープの内容を破壊しないように、テープの模倣に使っているレジスタの番号の最大値より大きな番号のレジスタを用いる。逆に、マスの内容をレジスタに書き込むときには、マスの番号を模倣に使っているレジスタ番号の最大値と比較して、それよりも大きければ、そのレジスタにブランク記号 4 つに相当する数をロードしてから演算を行う。このとき、レジスタ番号の最大値には 1 を加えておく。

OTM が神託要求状態に入ると、FRAM は神託テープの内容を取り出し、その内容を CNF-SAT に帰着させる。模倣している OTM は多項式時間に限定されているので、記号が記入されるマスの数は問題のサイズの多項式で抑えられる。また、神託は NP に属する問題であるから、通常の RAM で多項式時間で CNF-SAT に変換することが可能であり、したがって、FRAM でも (機能メモリを使わずに) 多項式時間で変換することが可能である。CNF-SAT に帰着された後は、前節に示したアルゴリズムにより多項式時間で解くことができる。ただし、このとき、機能メモリの未使用の領域を使用する。

以上より、NP 問題に対する神託を持つ多項式時間限定 OTM の各動作は、FRAM で多項式時間で模倣することができる。したがって、この OTM を、FRAM により多項式時間で模倣することができる。

次に、逆に、多項式時間限定 FRAM を NP 問題に対する神託を持つ OTM によって模倣する。

OTM のワークテープに FRAM の各レジスタの内容とレジスタ番号を対にして並べて記憶させる。FRAM の機能メモリを用いた命令以外の命令は、通常の RAM の命令であり、これらは、通常のチューリング機械により多項式時間で模倣できることが知られている¹⁾。機能メモリを用いた命令は次のように模倣する。

機能メモリによる計算の結果を参照するのは、検索結果レジスタを通してのみ可能である。したがって、FRAM においてオペランドとして検索結果レジスタが指定されたとき、そのときのみ、OTM は神託要求状態に入り、神託テープに書かれた問題に対する解を与える。WRITE1, SEARCH0 命令が実行される

たびに、OTM はその検索データとマスクデータを神託テープに記入する。前節の CNF-SAT を解くアルゴリズムからもわかるように、1つの検索データとマスクデータの対はブール式の和項に対応することから、このような形で記入された一連の系列は容易に和積形ブール式に対応させられる。したがって、CNF-SAT を解く神託があれば、検索結果レジスタの内容を与えることができる。

OTM が神託を呼び出す回数は、FRAM が検索結果レジスタを参照する回数に等しく、これは問題のサイズの多項式で抑えられる。また、神託テープに書き込まれるデータの長さも多項式のオーダーで抑えられる。以上より、多項式時間限定 FRAM を、CNF-SAT に対する神託を持つ OTM により多項式時間で模倣することができる。□

上記の証明から、任意の問題を解く際に、NP 問題に対する神託を持つ OTM で必要な神託の回数は、FRAM で検索結果レジスタの参照を必要とする回数に等しいことがいえる。

[定理 1] より、 d_2^2 に属する問題はすべて、FRAM により多項式時間で解くことができることが示された。しかし、これらの問題の中には膨大な容量の機能メモリを要する問題も含まれている。FRAM が CNF-SAT に対する神託を持つ OTM と等価であることが示されたので、このような OTM において神託のサイズを制限した場合に解くことのできる問題のクラスについての研究が、現実的な意義を持つようになると考えられる。

3.3 FRAM に対する機能の追加と計算能力

本節では、FRAM において、SEARCH0 命令に代えて、検索結果に基づいてアドレスを読み出す、次の命令を加えても、多項式時間で解くことのできる問題のクラスが変化しないことを示す。

S0GET oper 1, oper 2

S0GET 命令は、検索の結果一致したワードのうち、データ部の内容が 0 のものがあれば、そのうち最もアドレスの小さなワードのアドレスを検索結果レジスタに取り出す。検索結果レジスタには 1 ビットではなく任意の大きさの整数を記憶できるものとする。データ部の内容は書き換ええない。データ部が 0 のワードが存在しない場合には、検索結果レジスタに -1 を返すものとする。S0GET 命令は、SEARCH0 命令にアドレスを読み出す機能を加えたものと考えられる。これにより和積形ブール式が充足可能かどうかのみな

らず、充足させる変数の値の組合せが 1 つ得られることになる。

[定理 2]

FRAM の SEARCH0 命令を S0GET 命令に代えても多項式時間で解ける問題のクラスは変化しない。

<証明>

S0GET 命令を FRAM で多項式時間で模倣することによって証明する。

S0GET 命令を持つモデルにおいて検索結果レジスタに得られる値を記憶するレジスタを用意し、検索結果レジスタの代わりにこのレジスタを参照する。S0GET 命令が現れると、以下のサブルーチンを実行する。

- (1) S0GET 命令と同じ検索データおよびマスクデータにより、SEARCH0 命令を実行する。もし、データ部が 0 のワードが存在しなければ、検索結果を記憶するレジスタに -1 を書き込み終了する。
- (2) マスクデータで 1 になっているビットがなくなるまで以下の操作を繰り返す。
マスクデータで 1 になっているビットのうち、最上位のものを 0 とし、検索データの対応するビットを 0 とする。これらのデータにより、SEARCH0 命令を実行する。もし、データ部が 0 のワードが存在しなければ、今 0 にした検索データのビットを 1 にする。
- (3) (2) が終了した時点での検索データの値を、検索結果を記憶するレジスタに書き込み、終了する。

||

1つの S0GET 命令を模倣するのに、マスクデータ中の 1 の個数に比例する計算時間が必要である。マスクデータの桁数は問題のサイズの多項式で抑えられるので、S0GET 命令を FRAM により多項式時間で模倣することができる。□

機能メモリにさらに機能を追加した場合の計算能力に関する研究も興味深い。

4. FRAM の実現性

FRAM の機能メモリでは、アドレスの部分一致による複数のメモリセルへの並列アクセスが可能であり、アクセスされたセルに並列に 1 を書き込むこと、および、アクセスされたセルの中に 0 を記憶しているものがあるかどうかを並列に調べることができる。

このような機能メモリは、通常のメモリにおいてアドレスデコーダやメモリセルに若干の工夫を施すこと

等により実現可能であると考えられる¹⁰⁾。現在、4 M ビット DRAM LSI が実用されており、16M ビット DRAM LSI の試作が行われるようになっている。FRAM の機能メモリも、現在の集積回路技術で数M ビットの容量を持つ LSI が実現可能であると考えられる。この機能メモリ LSI を用いて、通常のメモリ LSI で主記憶装置等を構成するのと同様に、大規模な機能メモリシステムを構成することができる。ただし、アドレスの部分デコードと検索結果の収集のための機構が必要である。現在、数百Mバイトもの主記憶装置が実現されており、FRAM の機能メモリも数G ビットの容量を持つシステムが実現できると考えられる。

FRAM の機能メモリの機能は、既存の CAM によっても実現することができる。CAM では、全ワードの記憶データに対する並列部分一致検索、および、全一致ワードに対する並列部分書き込みが可能である。したがって、CAM に予めデータとしてアドレスを書き込んでおけば、FRAM の機能メモリの機能を実現できる。数年前に 20 K ビットの CAMLSI が開発されており⁴⁾、現在はより大容量のものが実現可能と考えられる。アドレスの部分を ROM 化した連想 ROM として実現すれば、さらに大容量のものが実現可能と考えられる。

FRAM では、NP 完全問題などを解く場合には、入力サイズの指数オーダーの容量の機能メモリが必要である。例えば、30 変数の CNF-SAT 問題を解くには 1 G ビット必要であるが、この程度のメモリシステムは現在の技術で十分実現可能であると考えられる。よりサイズの大きな問題を解くには、実用的には、ソフトウェアとの組合せを考える必要がある⁹⁾。

5. おわりに

本論文では、スーパーコンピュータへの第3のアプローチとして、高集積化が可能なメモリ型の並列計算機構を提案し、これを新しい並列計算モデルとして形式化し、その計算能力を明らかにした。

本論文で提案したメモリ型並列計算モデル FRAM は、通常のランダムアクセス機械に、アドレスの部分一致による複数のデータへの並列アクセスが可能な機能メモリを付加したものである。FRAM において多項式時間で解くことのできる問題のクラスは、逐次計算機で多項式時間で解くことは不可能であろうと考えられている A_2^P に等しい。また、FRAM は実現性が

高く、実用的にも、理論的にも興味深い並列計算モデルである。

FRAM において使用する機能メモリの容量に制限を加えた場合の能力や、異なるタイプの機能メモリを用いた計算モデル等についての研究が今後の課題である。

参 考 文 献

- 1) Cook, S. A. and Reckhow, R. A.: Time Bounded Random Access Machines, *J. Comput. Syst. Sci.*, Vol. 7, No. 4, pp. 354-375 (1973).
- 2) Stockmeyer, L. J.: The Polynomial Time Hierarchy, *Theor. Comput. Sci.*, Vol. 3, No. 1, pp. 1-22 (1977).
- 3) Kohonen, T.: *Content Addressable Memories*, 2nd ed., Springer-Verlag (1987).
- 4) 小倉 武, 山田慎一郎, 山田順三: 20Kb CMOS 連想メモリ LSI, 昭和 61 年度電子通信学会全国大会予稿集, 分冊 2, p. 477 (1986).
- 5) Chisvin, L. and Duckworth, R. J.: Content-addressable and Associative Memory: Alternatives to the Ubiquitous RAM, *IEEE Comput.*, Vol. 22, No. 7, pp. 51-64 (1989).
- 6) 国分明男, 樋口哲也, 古谷立美: 意味ネットワークマシン IXM における並列連想記憶, 情処研報, ARC 80-9, pp. 65-72 (1990).
- 7) 大久保雅且, 安浦寛人, 高木直史, 矢島脩三: UNION-FIND メモリを利用した単一化操作について, 信学技報, AL 85-49, pp. 31-40 (1985).
- 8) 高木直史, 矢島脩三: 機能メモリを用いた高速並列アルゴリズムについて, 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, 昭和 62 年度科学研究費補助金 (総合研究 A) 研究成果 (第 1 年次) 報告書 (1988).
- 9) 安浦寛人, 辻本泰造, 田丸啓吉: 組合せ問題に対する機能メモリ形並列プロセッサアーキテクチャ, 電子情報通信学会論文誌, Vol. J 72-A, No. 2, pp. 222-230 (1989).
- 10) 高木直史, 武永康彦, 矢島脩三: メモリ型並列計算機構をもつ計算システムについて, 情処研報, ARC 80-13, pp. 97-103 (1990).

(平成元年 8 月 30 日受付)

(平成 2 年 7 月 10 日採録)

**高木 直史 (正会員)**

昭和 34 年生. 昭和 63 年京都大学工学部情報工学科卒業. 昭和 58 年同大学院修士課程修了. 昭和 59 年 4 月より京都大学工学部助手. 工学博士. VLSI 向きハードウェアアルゴリズム, 算術演算回路, 論理設計用 CAD/DA 等の研究に従事. IEEE, 電子情報通信学会各会員.

**武永 康彦**

昭和 41 年生. 平成元年京都大学工学部情報工学科卒業. 同大学院修士課程在学中. 並列計算, 計算量理論の研究に従事. 電子情報通信学会会員.

**矢島 脩三 (正会員)**

昭和 8 年生. 昭和 31 年京都大学工学部電気工学科卒業. 同大学院博士課程修了. 工学博士. 昭和 36 年より京都大学工学部に勤務. 昭和 46 年情報工学科教授. 昭和 35 年京都大学第一号計算機 KDC-1 を設計稼動. 以来, 計算機, 論理設計, オートマトン等の研究教育に従事. 著書は「電子計算機の機能と構造」(岩波, 57 年) 等. 本学会元常務理事, 元会誌編集委員 (地方), 元 JIP 編集委員. 電子情報通信学会元評議員およびオートマトンと言語研専元委員長, North-Holland 出版元 IPL 編集委員, IEEE Senior Member.