音響ディジタル信号処理を主目的とする高速演算装置
μKIDOCH†

安倍正人∥ 嶋明弘∥ 上田隆∥
金井浩∥ 牧野正∥ 城戸建一∥

筆者らは、ディジタル信号処理技術を応用して、音響および音声信号処理の研究を行っている。これにより、音響信号処理および機械系の故障診断、音源位置の推定を行い、音響信号処理の分野では不特定話者音声認識に関する研究を行っている。これらの処理はいずれも膨大な計算量および記憶容量が必要とし、かつ、最終的にはリアルタイムで行わなければならない。そのため、汎用計算機では演算速度、リアルタイム性および処理の連続性の点で問題があり、DSPでは記憶容量、演算精度およびソフトウェアの柔軟性の点で問題がある。このため、筆者らはホスト計算機のハードウェアプロセッサとして、ディジタル信号処理を主な目的とした以下に示すような特徴を持つ高速演算装置μKIDOCHを開発した。(1)ホスト計算機との間のデータ転送ネットワークを解消するため、パイプライン化メモリにより結合する。(2)マイクロプログラム方式によるパイプライン処理を行う。(3)データベースあるいはアドレスベースの制御をする。プログラムはネットワークベースとして使用できる。μKIDOCHの性能について報告する。

1. まえがき

近年のディジタル信号処理技術の応用分野は、電気通信のみならず、医学、生物医学等の自然科学、さらに化学科学など多岐にわたっており、これらを処理する計算機には、処理の多様性に伴う柔軟性、複雑化に伴う高速性が求められている。ディジタル信号処理で扱う演算を一般的な数値演算と比較すると、(1)演算、(2)データの内容、(3)データの内容に依存せず、常にアルゴリズムで、(3)多数回繰り返す、という特徴を持っている。このことにより、ハードウェア設計にあたっては、パイプライン化などの高速化の手法が適用しうる、ソフトウェアの作成における、最適化を施しやすいと言える。

以上のようないの観点が、LSI技術の発展によって、プログラムを数値演算器を内蔵したディジタル信号処理システム(DSP-LSI)が異彩を放つ。また、汎用計算機でも、従来のハードウェアに内蔵アナログプロセッサ(AP)を付加し、自動化、パラレル化の機能を持った最適化コンピュータを搭載することが求められている。

当研究室では、ディジタル信号処理技術を応用し、音響、音声信号処理の研究を行っている。具体的には、音響部門では、音源位置の推定システムの開発、機械系の故障診断システムの開発、音声部門では、不特定話者音声認識システムの構築等を行っている。これらのシステムはいずれも膨大な計算量が必要とするもので、実時間もしくはそれに準じる速度で演算させる必要がある。故に、音声認識を目的としてセンサを16個用いて、音源位置推定システムを構築する場合には、1024点FFTを18チャネル分、連続して実時間で実行できる必要があるため、20kHzサンプリングとして、FFT1回当たり約3ms以内で実行する必要がある。また、音声認識システムにおいては、LP分析等において、演算精度が問題になり、浮動小数点演算が必要になる。また、建築音響においては、部屋のインパルスレスポンスをパラメトリー法で推定するためには、FFTの長さを十分長く取る必要があります。少なくとも8192点FFTが必要になる。このような仕様を満たすためには、以下のような理由から専用の高速演算装置が不可欠である。すなわち、汎用計算機を用いたシステムでは、記憶容量、演算精度の点では満足できるものであるが、特定の目的のための専用機として使用するには演算速度あるいはリアルタイム性の点でまだ問題がある。DSPを用いたシステムでは、処理速度あるいはリアルタイム性の点では満足のいくものであるが、記憶容量が小さいため、例えば、FFTでは1024点まで
しか実行できないことのほかに、演算精度、およびソフトウェアの柔軟性という点で問題のある場合が多いく、ただし、演算精度に関しては、最近のDSPの中には浮動小数点演算ができるものもあるが、依然として、メモリ容量は小さく、音響信号処理の分野には使えない場合が多い。このような理由から、従来から科学計算や信号処理を目的として、様々な並列処理のシステムが提案されてきているが、SIMD型の計算機のILIACIVやBSPでは、任意のプログラムを書くのが難しい。また、VLIW型の計算機であるQA-IIは浮動小数点演算ができず、また同じ機能を持つ演算器が4台あるため、例えば、専用の集中化したFORTRANコンパイラ等を構築するのには難しい。また、同じVLIW型の計算機であるAP-120Bは、連続して同じ処理を行うような場合に、ホスト計算機との間の転送時間が問題となる場合が多い。このような理由から、ディジタル信号処理を主な目的とした高速処理装置の開発の必要が生じ、そのような特徴を持つVLIW方式のパッケージ型演算システムμKIDOCORを設計、試作した。}

（1）ホスト計算機との間のデータ転送ネットワークを簡略化するために、μKIDOCORとホスト計算機の間をパイプロライズメモリにより結合する。

（2）様々な応用に柔軟に対応させるために、128bit/wordのマイクロプログラム方式によるパイプロライン処理を行う。この方式を取ると、コンパイラを容易に構築できる。このデータパスあるいはアドレスバスとして使われる32bitのバスを用いることにより、複数の演算器が並列に動作する。

（4）データ形式として、A/D変換直後のデータ（12bit固定小数点）を効率的に扱うためのCI型（Complex Integer：MSB 16bit実数部、LSB 16bit虚数部）を設ける。

（5）信号処理で行われる演算はすべて複数データの積和演算（バタフライ演算）の組合せで表現できるが、CI型のバタフライ演算を高速に実行するためのバタフライ演算器を用意する。

（6）複雑な処理での演算精度の確保のために浮動小数点型も設ける。

（7）プログラム作成のためにクロスアセンブリを用意する。

本報では、試作した高速演算装置μKIDOCORの特徴とクロスアセンブリを用いて作成した種々のプログラムの全体構成図を示す。
図2 マイクロプログラムフィールド

Fig. 2 Micro-Program field of μKIDOC.

命令フィールドの31〜φbitはbit69のFMT切換により、FMTφとFMT1の2つに分けられる。
2. ハードウェア構成

図1に本システムの全体の構成を示す。ホスト計算機はモトローラの68000を使用した市販の製品で、OSは現在はCP/M-68Kを使用している。このホスト計算機はμKIDOCHはVMEバスを通じて結合されている。
μKIDOCHは独立に動作する14個の装置から構成されており、約A3判のプリント基板8枚からできている。
μKIDOCHはパイプレイン処理17により高速化を図っている。1つのパイプレインは、セグメントと呼ばれるさらに細かい種々の演算ユニットの連続体として構成され、これらは同期化されたタイミング信号の制御を受ける。2つのセグメント間にはラッチレジスタを入れられ、これがセグメントの入出力データを保持する。セグメントS_;での遅延時間τ,ラッチレジスタでの遅延時間をτ,とすると、同期クロックの最小小周期は

\[ \tau = \text{Max} [\tau_1] + \tau \]  

である必要がある。また、このパイプレインはn個のセグメントから構成されているものとすれば、滞在時間はnτ,処理時間はτとになる。
μKIDOCHの特徴の1つに、内部バスが3本あることがあげられる。これにより、例えば次のアドレスのアドレス移動、前に与えたアドレスのメモリ内容の転送および演算器からの出力の3つのデータの転送が同時に行える。さらに、バス入力の制御が2bitのマイクロプログラムで効率良く行うことができる特徴の1つである。すなわち、(0)ラッチしない、(1)バス1からデータをラッチする、(2)バス2からデータをラッチする、(3)バス3からデータをラッチするの4通りがわずか2bitで行える。もし、バス入力の制御にマイクロプログラムのbit数を増やして、例えば3bitにすると、ラッチしない場合を除いて、バスの数を7本に増やさないと効率が悪く、また増やすとコストがかかりすぎ、実現的でない。

2.1制御装置（MPM, PCU, ICU, VMCU）

MPMはマイクロプログラムメモリで、4k×128bitのRAMで構成されている。図2にマイクロ命令の構成図を示す。PCUはプログラムコントロールユニ

![Diagram](image_url)
2.2 記憶装置 (DMU1, DMU2)
DMU1 および DMU2 は、図 3 に示すように、32 ビット/ワードで、それぞれ 64
k-word・2 Bank、16 k-word・2 Bank のデータメモリユニットである。各メモリユニ
ーツの2つのバンクのうち、どちらか一方はμKIDOCH がアクセスし、他方はホ
スト計算機に接続されており、ホスト計算機から認識できるメモリである。バンク
の切り替えはマイクロプログラムによって
行われる。1 回のメモリアクセスには2ステップ必要
であるが、内部はパイプライン化されており、1 ステ
ップごとに連続したアクセスが可能である。
各メモリユニットが2つのバンクを持つと、例えば
A/D 変換と FFT を連続して行う場合には事実上データ
の転送に要する時間が 0 とみなせる。すなわち、
A/D 変換を要する時間、FFT に要する時間およびホ
スト計算機を後処理する時間のうち一番長い時間を1
フレームの長さとすると、図 4 において、フレーム 1
ではホスト計算機に接続されている A/D 変換器から
DMA（データメモリアクセス）コントローラに
より直接 DMU1 のバンク 1 にデータが転送される。
フレーム 1 とフレーム 2 の端末でバンクが切り替わ
れる。フレーム 2 では、μKIDOCH はフレーム 1 で
与えられたデータを FFT し、結果を DMU 2 に送る
する。その間、A/D 変換器に付属した DMA コン
トローラは次のデータを DMU1 のもう1 つのバンクに
転送する。次のフレームとの端末で再びバンクが切り
替えられる。フレーム 3 ではホスト計算機は DMU 2
にある FFT された結果を直接後処理することができる。
また、その間、μKIDOCH はフレーム 2 で与え
られた DMU1 にあるデータを FFT し、DMA コン
ストローラは次のデータを DMU1 の他のバンクに転送
する。このように、連続して同一の演算を行う場合に
は、DMA コントローラ、μKIDOCH およびホスト
計算機が完全に並列動作し、データの転送時間は実質
的に 0 とみなせる。
2.3 算術論理演算ユニット (ALU1, ALU2)
ALU1, ALU2 はそれぞれ AMD 社の 4 bit のビ
ットスタースマイクロプロセッサ Am 2901C を 8 個
用いて 32 bit 構成したものである。演算結果のコン
ディションは PCU から参照できる。条件 JUMP 等が
行える。ALU1 および ALU2 は固定小数点データ
の演算（加算および除算）のみでなく DMU 1 ある
いは DMU 2 のアドレス計算にも用いられる。
2.4 パタフライ演算ユニット (BMU)
信号処理演算は一般に積算演算の積算演算から
構成されている。特に 2 を基数とした FFT ではパタフライ
演算（積算の積算演算）の積算演算から構成され
ている。制御計算機あるいはマイクロプロセッサ等では、このパタフライ演算を逐次的に行っているために
時間がかかる。それに対して、BMU は図 5 に示すよ
うに 16 bit の乗算器を 4 個および 16 bit の ALU 7
個を並列に動作させて、(1) CI 型のパタフライ演算、
(2) INT 型の符号なし乗算を行うユニットである。
内部的には(1)入力データのシフトおよび乗算、(2)
各乗算器の出力の加算、2つのセグメントに分けられ
ており、異時時2ステップ、処理時間1ステップで
実行できるようになっている。
BMU を用いたパタフライ演算では、実数部、虚数
図5 BMU構成図
Fig. 5 Block diagram of Butterfly & Multiply Unit.

部はそれぞれ16bitの固定小数点の形式になっており、演算の途中でオーバーフローが起きる可能性がある。これを防ぐために、BMUにはオートスケーリング機能が付加されている。すなわち、このまま演算を続けた場合に次のステージでオーバーフローが生じる可能性があることを実数部および虚数部それぞれに対して上位2bitの排他的論理和を取ることにより検出し、次のステージではA入力とB入力を1bitダウンシフトさせるというものである。

2.5 浮動小数点演算ユニット (FADD, FMPY, FDIV, IF, FI)
FADD, FMPY, FDIV, IF, FI はそれぞれ浮動小数点加算演算ユニット、浮動小数点乗算ユニット、浮動小数点除算ユニット、固定小数点演算ユニット、固定小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットおよび浮動小数点演算ユニットを含むものとする。FADD 内部では (1)桁合せ、(2)加算演算の2つのセグメントに、FMPY 内部では (1)乗算、(2)正规化の2つのセグメントに分かれており、どちらも動的変更2ステップで演算を行う。また、FDIV は10ステップ、FI はそれぞれ1ステップで演算を行う。

3. μKIDOCH のアセンブラ言語
μKIDOCH は128bitのマイクロプログラムによって各演算器、メモリ等の機能とメモリの結合状態を制御している。このように長大なマイクロプログラムを直接2進数によって記述することは非人間的な作業である。単純な鎖の原因にもなる。そこで、ホスト計算機を用いた μKIDOCH 用のクロスアセンブラを製作した。

図6（a）に示したものは、128bitのマイクロプログラムの1命令に相当するものであり、アセンブラで
記述された一連のアセンブラプログラムは、その集めとして記述される。

＜ラベル＞は、マイクロプログラム上のこの命令のアドレスを保持する。

＜ユニット名＞には、[ALU1, ALU2, PCU, IUC, VMCU, DMU1, DMU2, BMU, FL, IF, FADD, FMPY, FDIV]のうち1つを指定することができる。機能については、それぞれのユニット名に応じて、種々のものが解釈されている。

＜疑似ユニット名＞には、[ORG, BSO1, BSO2, BSO3, IMM, FMT, IRT 68K]がある。ORG は、生成されるマイクロプログラムの先頭アドレスを指示するもので、アセンブラプログラムの第一行目に書く必要がある。BSO は、制御部に出力するユニットを指定するもので、＜疑似機能＞の一部には＜ユニット名＞を記述する。

IMM, FMT, IRT 68K はそれぞれ、マイクロプログラムに内包する 32bit のデータ、マイクロプログラムフィールドのマップ変換、ホスト計算機への割込みを指示するものである。図 6 (b) は、アセンブラで記述したマイクロプログラムの一例である。

4. ホスト計算機からの処理の依頼

ホスト計算機は、μKIDOCH のマイクロプログラムメモリ (MPM) にプログラムをロードしておき、起動をかける。その後、ホスト計算機は必要なデータをデータメモリ１ (DMU1) あるいはデータメモリ２ (DMU2) にストアし、μKIDOCH に割り込みをかける。μKIDOCH は必要な演算を実行した後、ホスト計算機に割り込みをかかって処理の終了を知らせ、ホスト計算機からの確認を受けたあとにメモリのバンクを切り替えることにより演算結果をホスト計算機に引き渡す。

△＜ラベル＞

△＜ユニット名＞△＜機能＞、＜機能＞、…、＜機能＞、△；＜コメント＞

△＜疑似ユニット名＞△＜疑似機能＞△/

ただし△は、任意の空白を示す。

図 6 (a) アセンブラプログラムの書式

Fig. 6 (a) Format of assembly program.

以下の計算を 1 で 1023 まで繰り返す

a(t) + a(t1) * b(t) + c(t1)
ただし a, b, c はいずれも
DMU1 の 0 から 4095 まで
DMU2 の 0 から 4095 まで
に記憶しているものとする。

図 6 (b) アセンブラで記述したマイクロプログラムの一例

Fig. 6 (b) An example of assembly program.

5. μKIDOCH の性能の評価

本章では、まず、μKIDOCH とはほぼ同じ目的で設計されたと思われる AP-120 B と μKIDOCH との比較を行い、続いて、当研究室で行っている研究を使用して種々のシステムを構成するために必要な 1024 点 FFT と 28 チャネル帯域フィルタ群を用いて、製作したμKIDOCH の性能評価を行う。
5.1 AP-120 B との比較検討
（1）AP-120 B のはやは（a）アドレス専用バス、（b）データメモリアリズムおよびレジスタファイルから演算器（FALU：浮動小数点 ALU と FMPY：浮動小数点演算器）への入力バスおよび、（c）演算器から演算器、データメモリアリズムおよびレジスタファイルへのリザルトバスという異なる機能を持つ 3 本のパスから構成されている。このため、AP-120 B では、FALU と FMPY とが同時に出力を得ることができない。また、2 以上の次元を持つ配列のアドレスリングには固定小数点の乗算が必要であるが、AP-120 B ではアドレスの計画に固定小数点の乗算を実行することができないため、2 次元以上の配列を扱うのは従来ではない。これに対して、μKIDOCH はどんな情報を流すことができが可能な汎用バス 3 本で構成されているので、プログラミングの自由度が大きい。また、μKIDOCH には固定小数点演算器があり、任意の次元の配列を扱うことができる。

（2）AP-120 B の演算器 FALU および FMPY はデータメモリとレジスタファイルに対して対等であり、信号処理で最も良く用いられる積和演算を特に意識して作られていない。これに対して、μKIDOCH では、16 bit 固定小数点用ではあるが、BMU と呼ばれるバタフライ演算器が用意してある。ただし、16 bit 固定小数点においても、ブロックフローティング FFT ができるので、ダイナミックレンジとしては 60 dB 以上が得られ、音響信号処理としては十分な演算精度が得られる。

（3）AP-120 B も μKIDOCH もともにバックエンドプロセッサであり、使用するためにはホスト計算機が必要であるが、ホスト計算機との間のデータ転送が必要となる。このデータ転送の効率化に関しては、AP-120 B はそんな対策も用意していないが、μKIDOCH ではパイプライン化メモリを用意して処理を連続して行う場合にはデータ転送の時間を実効的に 0 にすることができる。

5.2 1024 点 FFT
図 7 に μKIDOCH で FFT を実行する際のアルゴリズムを示す。FFT のアルゴリズムは、時間間引き型とし、データのアドレス計算およびループの制御はすべて加算、シフト演算のみで処理できるようにしている。また、2 つのメモリユニットを持っているという μKIDOCH のハードウェアの特徴を生かすために、一方のメモリユニットを読み出し専用、他方を書き込専用として、ステージが変わるたびにこれを反転させるという方法を用いる。

プログラムは Stage Counter, Block Counter, Iteration Counter を制御変数とする 3 重ループからなる。このうち、最も内側のループ（メモリアクセスも含めた 1 回のバタフライ演算に相当）は最も多く実行されるものであり、この部分の処理速度をいかに速くするかが FFT 全体の実行速度を決定する。図 8 に FLOAT 型のバタフライ演算のタイミングチャートを示す。バタフライ演算を連続的に行う場合には、図中で 0～2 ステップで行う演算を 10 ～14 ステップのところで行わせることが可能である。そのため、実効的には 1 バタフライ演算当たり 12 ステップで処理できる。ただし、タイミングチャート中の &An は変数 An のメモリユーニット上でのアドレスを表している。このときの各バスおよび演算器の稼働率は表 1 に示す。なお、CI 型は 3 ステップでバタフライ演算が実行でき
表1 FTT 実行時の計算装置各部の稼働率  
Table 1 Working ratio of each processing unit at execution of FFT.

<table>
<thead>
<tr>
<th>CI型の場合</th>
<th>Bus</th>
<th>ALU</th>
<th>DMU1</th>
<th>DMU2</th>
<th>BMU</th>
</tr>
</thead>
<tbody>
<tr>
<td>バタフライ</td>
<td>8/9</td>
<td>86%</td>
<td>66%</td>
<td>66%</td>
<td>66%</td>
</tr>
<tr>
<td>FLOAT型の場合</td>
<td>28/36</td>
<td>81%</td>
<td>78%</td>
<td>53%</td>
<td>50%</td>
</tr>
</tbody>
</table>

20倍程度の時間が必要であるが、μKIDOCHを用いると実時間処理が可能となる。ここでは、29チャネル帯域フィルタ群をFLOAT型で構成したときの結果を示す。この場合、1チャネル当たり滞在時間14ステップで実行できるので、十分に実時間処理が可能となる。また、そのときの各パースおよび演算器の稼働率を表3に示す。
表 3 フィルタリング実行時の演算装置各部の稼働率
Table 3 Working ratio of each processing unit at execution of filtering.

<table>
<thead>
<tr>
<th>Bus</th>
<th>ALU</th>
<th>DMU1</th>
<th>DMU2</th>
<th>FADD</th>
<th>FMYP</th>
<th>IF</th>
</tr>
</thead>
<tbody>
<tr>
<td>≪66%</td>
<td>≪61%</td>
<td>≪56%</td>
<td>≪44%</td>
<td>≪33%</td>
<td>≪44%</td>
<td>≪11%</td>
</tr>
</tbody>
</table>

5.4 パイプライン化メモリの効用
図4に示した処理を連続して行う場合について、パイプライン化メモリの有効性を調べた。
データ転送を含めたA/D変換に要する時間、FFTの時間、データ転送を含めたホスト計算機による後処理の時間をそれぞれ、TAD, TFFT, THOST とするとき、パイプライン化メモリを用いない場合には、1組の処理時間 トN は

\[ T_N = T_{AD} + T_{FFT} + T_{HOST} \]  

(2)

となる。また、データの抜けを無くして処理を連続して行う場合には、A/D変換器には大容量のバッファメモリを付ける必要がある。一方、パイプライン化メモリを用いて連続的に処理する場合には、A/D変換とホスト計算機による後処理をともにVMEバスをアクセスするのでパースが競合する。そのため、1組当たりの処理時間 トP は

\[ T_P = \max(T_{AD} + T_{HOST} + T_{FFT}) \]  

(3)

となり、連続して処理する場合にても、A/D変換器には小容量のバッファメモリを付けるだけで、比較的容易に構成できる。図9はA/D変換器のサンプリング周波数を40kHz、VMEバスのデータ転送速度を500ns/サンプル、TFFT=16ms（1024点複素数浮動小数点FFT、固定小数点から浮動小数点へのFORMAT変換、および時間帯の掛け算を含む）としたときの

THOST をパラメータとした トN と トP の値をプロットしたものである。この図から、パイプライン化メモリが有効であることがわかる。すなわち、サンプリング周波数40kHzで1024点FFTを連続的に行うためには、1組の処理（図4におけるフレームの長さ）を約25ms以内に行わなければならない。その場合、なんらの対策を取らないときには、ホスト計算機に与えられる後処理の時間 トN は8msしかないが、パイプライン化メモリを用いると、A/D変換のデータ転送の時間約1msを除いて後処理の時間 トP として約24msを使うことができる。

5.5 FFT演算の高速化のためのμKIDOCHのアーキテクチャの改善
μKIDOCH のように各装置間のデータ転送をす
1316 情報処理学会論文誌 Dec. 1987

ければならないので、スリップ数は増え、図8のタイ
ミングチャートでは、12スリップかかっている。この
原因は、（1）メモリリクエストにアドレス記憶領域が
1つしかなく、また、FMPY およびFADD には入力デー
タの記憶領域が1つずつしかないため、後で同じ値を
使うとわかっていても、それを取っておけば、再転送
しなければならないこと、（2）FMPY およびFADD
は入力データを与えてから3スリップまではじめて結
果が利用できるため、他の算術とタイミングがうまく
合わず、演算器が利用できないスリップが生じるため
である。もしそ、（1）での問題である記憶領域がそれ
ぞれ2つになると、スリップ数は7になる。さらに、
（2）が解決できれば6スリップとなり、理論上の最高
速度で1パ Cra が実現できることがある。

6. む す び

当研究室で行っている音響信号処理や音声認識等の
研究を効率的に行うために信号処理用高速演算装置
μKIODOH を開発した。本演算装置はホスト計算機
との間のデータ転送の仕方で特徴を持つ、すなわち、
パイプライン化メモリにより処理を連続的に行う場合
には実質的にデータ転送の時間が0とみなせるため、
本演算装置はホスト計算機は高度の並列処理が可能と
なっている。現在のところ、ソフトウェアとしては、
FFT とディジタルフィルタのみであるが、行列演算
等を順次増やしていく予定である。さらに、現在、
μKIODOH 用のμC 言語の高級言語を開発中であり、
出来上がり次第発売する予定である。

謝辞
本演算装置を製作する際にいろいろ御協力を
頂いた（株）小野製作所の入野信彦氏、前田利重氏および
佐々木徹氏に感謝いたします。また、有益な討論をし
ていたい東北大学城戸研究室の皆様に感謝いたします。

参考文献
2) Hagiwara, Y., Kita, Y., Miyamoto, T., Toba, Y., Hara, H. and Akazawa, T.: A Single Chip
3) μPD 77230, 日本電気（株）
4) COSMO 700 S システム説明書, 三菱電機（株）.
5) Abe, M., Nagata, Y. and Kido, K.: A New Method to Locate Sound Sources by Searching
the Minimum Value of Error Function, Proceedings of IEEE International Conference
6) 上田 陸, 安倍正人, 城戸健一: 直線配列マイ
クロホンを用いた CCSM による音波放射方向推
7) Kanai, H., Abe, M. and Kido, K.: Detection of
Slight Defects in Ball Bearings by Non-periodic
277 (1986).
10) 高橋義義: 並列処理マシン開発の現状, 情報処
11) 富田真治: 並列計算機構成論, 昭晃堂, 東京
(1986).
12) Bouknight, W. J. et al.: The ILLIAC IV
13) Kuck, D. J. and Stokes, R. A.: The Burroughs
Scientific Processor (BSP), IEEE Trans., Vol.
14) Hagiwara, H. et al.: A Dynamically Micro-
programmable Computer with Low Level
15) Charlesworth, A.E.: An Approach to Scientific
Array Processing : The Architectural Design of
the AP-120 B/FFS-164 Family, IEEE Comput.,
16) 上田 陸, 安倍正人, 城戸健一: 信号処理用高
速演算装置のソフトウェア作成言語と目的プログラ
マの最適化について, 第 32 回情報処理学会全
国大会論文集, No. 5-5-S-10, pp. 271-272 (1986).
17) 地域: コンピュータの高速演算方式, 近代科学
社, 東京 (1980).
18) MMI PAL FAMILY, MMI ジャパン.
19) 三輪賢二, 城戸健一: 音声認識による分析フィル
タの特性の検討, 第 8 回東北大学応用情報学研究

(昭和 62 年 6 月 18 日受付)
(昭和 62 年 11 月 11 日採録)
安信 正人（正会員）
昭和27年生。昭和56年東北大学大学院工学研究科修士。現在同大学応用情報学研究センター助手。工学博士。音響ディジタル信号処理、計算機アーキテクチャなどの研究に従事。IEEE、電子情報通信学会、日本音響学会、日本音響制御工学会各会員。

崎 明弘
昭和34年生。昭和58年東北大学工学部電気工学科卒業。昭和60年同大学院修士課程（情報工学専攻）修了。現在同大学院博士課程在籍。音響信号の計算機処理および信号処理用音響装置の研究に従事。日本音響学会会員。

上田 隆（正会員）
昭和35年生。昭和59年東北大学工学部機械工学科卒業。昭和61年同大学院情報工学専攻修士課程修了。現在NTT無線システム研究所勤務。ディジタル移動通信の研究に従事。電子情報通信学会会員。

金井 浩（正会員）
昭和33年生。昭和56年東北大学工学部通信工学科卒業。昭和61年同大学院博士課程修了。工学博士。同年東北大学情報処理教育センター助手。現在に至る。音響・振動信号等のディジタル信号処理と機械系診断への応用に関する研究に従事。昭和62年度石川賞受賞。日本音響学会、電子情報通信学会、日本機械学会各会員。

牧野 正三
昭和22年生。昭和44年東北大学工学部電子工学科卒業。昭和49年東北大学大学院工学研究科修了。同年東北大学電気通信研究所助手。昭和55年東北大学応用情報学研究センター助手。昭和62年同助教授。音声の自動認識に関する研究に従事。日本音響学会、電子情報通信学会各会員。

城戸 健一（正会員）
1926年4月15日生。1948年3月東北大学工学部電気工学科卒業。東北大学電気通信研究所助手。同大学工学部電気工学専攻修士課程修了。現在に至る。音声自動認識、ディジタル信号処理とその応用に関する研究に従事。著書：音響工学（電子通信学会編、コロナ社）、ディジタル信号処理入門（丸善）、電子計算機原論 上下（丸善）、過渡現象論（朝倉書店）等。工学博士。日本音響学会、電子情報通信学会、電気学会、計算自動制御学会、IEEE、AESなどの会員。アメリカ音響学会フェロー。