解　説

信号処理用プロセッサ*

内　田　俊　一**

1. はじめに

近年，汎用計算機により解かれてきた問題を，より高速に，より安価に解くために，それぞれの問題解決に適した機能，構成を持つ専用プロセッサが，開発されている．ここで述べる信号処理用プロセッサは，外界から取入れられた信号に対し，種々の演算処理をディジタル的に行って，有効な情報を抽出するために使用され，フーリエ変換やフィルタリングなどの演算を，高速，高能率に実行する機能を持つものである．これらは，もともと，機械振動，音響，通信，音声，画像などの信号処理の必要性から開発されたものが多く，その機能，構成，処理能力など，多種多様なものがある．小は，単に，入力信号の周波数スペクトルや，相関などを見るスペクトラム・アナライザやコリレータのような単機能のものから，大は，豊富な機能と数メガバイトの記憶，多数の演算器を持ち，汎用計算機をその中に含んでしまうものまで，各種あり，それぞれ，その使用分野における重要な演算処理，使用環境に適合するよう設計されている．このような，使用分野の影響は，そのアーキテクチャにも及び，非常にバラエティに富んだものとなっている．これらを，一まとめにして扱うのは無理があるが，近年，できる限り広範な信号処理の分野に適合することを意図した，汎用信号処理用プロセッサとも言うべきものが，多数，市販されるようになった．これらは，高速性と，それによる計算コストの低減を最優先したアーキテクチャを持っており，その中に，信号処理用プロセッサに共通する特徴を集約していると考えられる．そこで，本解説では，このような汎用信号処理を目ざすものを中心に述べる．現在では，これらの処理速度は，ミニコンのソフトウェアによる処理の数百倍，汎用大型機の数十倍以上であり，計算コストも，汎用大型機で数十時間を要する画像のフィルタリング処理等の場合，これを数十分に短縮することにより，十分の一以下にコストを低減するといわれている．以下，このようなプロセッサについて，発展過程，演算・制御方式，構成などについて述べる．

2. 信号処理用プロセッサの歴史的背景と現状

ディジタル信号処理用プロセッサの開発は，それ以前に築かれた汎用計算機を用いた信号処理という基盤の上で行われた．初期のディジタル信号処理は，軍用のレーダ装置等の例外を除けば，もっぱら汎用計算機を用いて行われた．しかし，これらは，乗算速度，語長，計算コスト等に不適当な点が多く使いにくいものであった．この解決に大きく貢献したものが，1960年中期以降のミニコンの登場と，FFT アルゴリズム[1]の考案であった．ミニコンは，語長も適当で，安価であり使い易いものであった．また，FFT は信号処理に欠くことのできないフーリエ変換の計算時間を大幅に短縮し，ディジタル信号処理の適用範囲を飛躍的に拡大した．この結果，ミニコンのソフトウェアによる FFT を用いた信号処理が広く普及し，新たな応用分野を開拓するとともに，その基盤を築いた．その後，実時間処理や，データ量の増大などの理由から，高速のプロセッサが求められた．1960年代も終りに近づくと，論理素子，記憶素子の IC 化が進み，高速，低価格のものが作られ，これらを用いた民生用の信号処理用プロセッサが，続々開発されるようになった[2),3)]．これらは，既に普及していたミニコン等を中心とする信号処理システムに追加接続するような形態をとることができた．このため，信号処理専用の設計が行え，高速，高パフォーマンス化が容易であったほか，ソフトウェアや使用に当ってのノウ・ハウ等をユーザに期待することができ，有利であった．

このようにして，汎用計算機をホストとし，その負荷を分担する従属型プロセッサが普及したが，同時に，

* Digital Signal Processors by Shunichi UCHIDA (Electrotechnical Laboratory)
** 電子技術総合研究所　情報システム研究室

CPU の一部へ組込むモジュール型や，単独で動作できる独立型なども作られ，それぞれの適用分野へ広く普及していった．初期のプロセッサでは，その機能をFFT 等の少数のものに固定するものが多かったが，信号処理の適用分野が拡大するにつれ，その守備範囲拡大のため，豊富な機能を競って備えるようになった．最近では，多くのメーカが信号処理用プロセッサを市販しており，ますます多様化しているが，高速化，大型化，汎用化が顕著である．まず，従来の固定小数点演算にかわる浮動小数点演算の採用，演算ビット幅の拡大，記憶容量の拡大が見られるほか，多種類の演算機能を実現するためのアーキテクチャの柔軟性の強化が見られる．これらの実現にあたっては，最新の技術が駆使されている．まず，素子については，ECL，ショットキ TTL 等の高速論理素子，バイポーラや MOS 記憶等の MSI, LSI，高速のシフタ，バスを形成するマルチプレクサや3ステート・ゲートの多用が挙げられる．演算制御方式については，高速の乗算アルゴリズム，並列処理の採用，ダイナミックマイクロプログラミングの活用，マイクロプログラムのユーザへの解放などが挙げられる．このような最新技術の実用化という点では，汎用計算機の一歩先を進んでいるものも多い．これは，信号処理用プロセッサでは，高速性が生命であり，同時に，広範な用途のそれぞれに対し，高パフォーマンスを実現することが，市場確保の絶対条件であり，これらを両立させねばならないことによると思われる．

現在のプロセッサ開発の問題点についてみると，第一に挙げられるのは乗算器と記憶である．信号処理の演算は，並列化が効果的に適用でき，多数の演算器を用いることで高速化が行える．しかし，現状では，乗算器の価格が高く，ほとんどのものが，1～4個しか持たない．この解決策として，LSI 化がすすめられており，一部が市販され始めている．しかし，その機能，回路構成，消費電力等の点で，そのままプロセッサに組込むには，現在のところ，まだ問題が残されている．また，多数の演算器の使用は，当然，多数のデータを同時に供給する必要を生じ，いかにして安価で高速の記憶を得るか，また，いかにして，アクセスの競合を避けるか等の問題が生じる．このような問題の解決は，今後の研究を待たねばならないが，多くは価格面の問題であり，現在の広範な分野へのディジタル信号処理の浸透を見れば，その未来は，約束されているといえよう．

3. 信号処理演算とその特徴

信号処理に用いられる演算は，汎用計算機が通常扱う一般的な計算処理に比べ，はるかに均質であり，明確な特徴を持っている．信号処理用プロセッサは，この特徴をフルに利用して，高速化と，アーキテクチャの最適化をはかっている．

演算の対象となるデータは，一般に多数の標本からなる配列（ベクトルまたはアレイ）で，時間，周波数，空間座標等の関数である．この配列が一つの処理単位となり，配列どうしの加算，乗算等の簡単なものから，これらを複雑に組み合わせた，FFT，電力スペクトル，コーヒレンス関数，ケプストラムなど[4]のような，よりマクロな演算までが，組立てられる．これら演算処理を実行する際の特徴をまとめると次のようになる．

i）1回の処理の対象となるデータ数が多い．

ii）マクロな演算は，基本的演算の繰り返しで，実現される．

iii）基本的演算は，実数または複素数の加算，乗算の組み合わせで構成される．

iv）基本的演算の形は，演算処理の種類により，少しずつ異なる．

これらのうち，i），ii）は，高速化にあたって，並列処理が適していることを示し，その方法として，基本的演算をハードウェア化し，基本演算モジュールとし，このモジュール内部を最適化して高速化すること，また，このモジュールを多数用いて並列処理を行って高速化することが考えられる．また，iii），iv）は，基本演算モジュール中の構成を多少変更することで，多くの演算処理に使えることを示している．これらの，具体例として，信号処理において多用される FFT と，二次差分の計算について示す．

a）F F T

FFT は，離散フーリエ変換を高速に計算する方法で，いくつかの変形アルゴリズム[5),6)]があるが，最初に，Cooley と Tukey により発表されたアルゴリズム[1)]では，その基本的演算は，式(1)のように書け，その実現形は**図-1**(次頁参照)に示すものとなる．これは，その形状から，バタフライ乗算器と呼ばれる．

$$\left.\begin{aligned} \boldsymbol{A}&=\boldsymbol{X}+\boldsymbol{Y}\cdot\boldsymbol{W}^p \\ \boldsymbol{B}&=\boldsymbol{X}-\boldsymbol{Y}\cdot\boldsymbol{W}^p \end{aligned}\right\} \quad (1)$$

$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{W}$ は複素数．$\boldsymbol{W}$ は回転因子．

$$\boldsymbol{W}^p=\exp\left(-\boldsymbol{j}\frac{2\pi}{N}p\right)$$

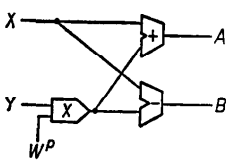

図-1　2を法とするクーリ・チューキ法によるバタフライ乗算

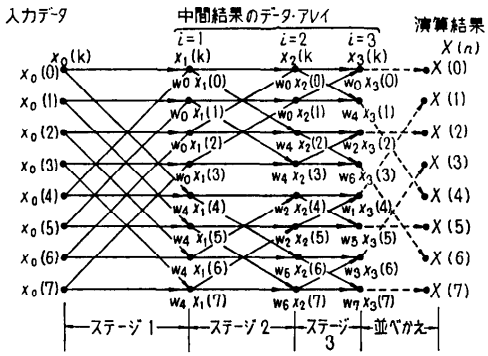

図-2　2を法としたクーリ・チューキ法によるFFT実行流れ図

$$=\cos\left(\frac{2\pi}{N}p\right)-j\sin\left(\frac{2\pi}{N}p\right)$$

これを，**図-2** に示すように繰り返し用いて計算を行う．図-2 では，$N=8$ 個の標本からなる時間関数 $x_0(k)$ が，周波数関数 $X(n)$ へ変換されている．結果は，そのままでは，添字の順とはならず，並べかえが行われている．並べかえは，n を2進表現し，$(011)_2$ なら $(110)_2$ のようにビットの上下を反転した添字を持つものとの入替によりなされ，この操作をビット反転 (bit reversion) と呼ぶ．W^p は回転因子と呼び，式(1)に示すような三角関数で，前もって準備されたテーブル中から供給される．

以上のように，FFT では，バタフライ乗算を基本演算モジュールとし，これを繰り返し用いて計算が行われる．その入力は複素数であるから，そのまま実現するには，実数用の4個の乗算器と6個の加算器が必要である．また，ビット反転操作を含む図-2 の演算を行うための番地発生も，面倒なものとなっている．

b) 二次差分方程式

二次差分の計算は，ディジタル・フィルタの実現に重要な演算で，その基本的演算は，式(2)のように書ける．

$$y_n=A_1\cdot y_{n-1}+A_2\cdot y_{n-2}+B_2\cdot x_{n-2}+B_1\cdot x_{n-1}+x_n \qquad (2)$$

この場合は，1個の加算器と乗算器を繰り返し用いて計算できるほか，4個の加算器と乗算器を用いて，基本演算モジュールを構成すれば，高速化が可能である．これらのほかにも，いろいろな形の基本演算があり，それを一つのモジュールと考え，その内部での最適化がいろいろ考えられている．a), b) は，ともに，通信機器等の中へ部品として組込まれ，実時間処理を行うような用途も多く，ROM を用いた乗算方式等が，盛んに研究されている[7)~9)]．

上記のもの以外のいろいろな演算についても，いろいろなレベルでの並列化が可能であり，そのプロセッサの適用分野を考慮し，どのレベルで，いかにして効果的な並列化を実現するかが，高速化の鍵となる．

4. 演算方式と演算器の構成

4.1 演算処理の並列化

処理の並列化については，汎用計算機に適用するための方法が，種々考案され実装されている．信号処理用プロセッサにも，これらの方法が用いられているが，その方式と実装のレベルには，次のものがある．

a．並列処理方式
- i) 複数の演算器を用いるもの（単一命令複数データ・ストリーム）
- ii) パイプライン化によるもの（複数命令単一データ・ストリーム）

b．並列化のレベル
- i) バタフライ乗算器のような基本演算モジュールのレベル
- ii) 基本演算モジュール内部での加算器，乗算器のレベル
- iii) 乗算器等の各演算器内部のレベル

これらのどの方式をどのレベルに用いるかによってその処理速度，柔軟性，価格が大きく変わる．FFT の場合を考えれば，多数のバタフライ乗算器を用いることで大幅な高速化が可能である．しかし，その価格は高価となり，また，FFT 以外の演算処理には不適当なものとなり柔軟性が失われる．価格の問題については，LSI 化等により改善が期待できるが，柔軟性の欠除は大きな問題であり，これらの理由から，i)のレベルの並列化は，ほとんど行われていない．また，ビット幅の広い浮動小数点乗算器等も高価であり，これを多数用いることも少ないのが現状である．

これにかわるものとして，a の ii)のパイプライン化がひんぱんに用いられている．そのレベルとしては，

bのⅱ), ⅲ)が多く，演算器数も1～4個程度が用いられ，これらによる並列演算とパイプライン化が，高速，低価格を目ざして，いろいろ組み合わされる．

バタフライ乗算器の例として，著者らが設計したKSP[10),11)]で用いたFFT演算器を**図-3**に示す．これでは，4個の16ビット乗算器（M1～M4），6個の16ビット加算器（A1～A6）を用いるとともに，4段のパイプラインとしている．すなわち，2つの記憶から送られてくるデータの順序を調整するデータ編成ユニット（DOU）で1段，複素演算ユニット（CAU）で2段，データの順序調整とシフト回路からなるデータ再編成ユニット（DRU）で，1段となっている．1段あたりの遅れは，乗算を2段に分割すること，素子としてTTLを用いることから決まり，600 nsとなっている．この演算器は，バタフライ乗算のほか，複素数の加減乗算や絶対値が計算できるほか，32ビットの実数加減算，ハニング加重等が行えるような柔軟な構成を持っている．図-3からもわかるように，バタフライ乗算器1個でも，その構成はかなり複雑で大きなものとなり，LSI化するにしても，どのような機能を持たせ，どのような構成をとるかの決定は，困難な問題を多く含んでいる．

次に，バタフライ乗算器等の構成で，最も問題となる乗算器について述べる．高速乗算器を構成する方法としては，並列型，並列繰り返し型，ROMを用いたもの等がある[4)]．並列型は，原理的には，$(m\times n)$ビットの乗算を行う場合，加算器を$m\times n$個アレイ状に配置する方法である．実際には，IC化された(4×2)や(4×4)ビット乗算器と加算器を組み合わせて構成する．こうして構成される乗算器は，組み合わせ回路となり，素子の時間遅れのみで実行できるが，ビット幅が増えるとIC数は，非常に多くなる．このため，繰り返しを含めた並列繰り返し型とし，パイプライン化する方法が多く採られる．これは，(16×16)ビットの乗算なら，(16×4)ビットは，並列に行い，これを4回繰り返して，(16×16)ビット乗算を行うような方法で，IC数も少なくすみ，パイプライン化により乗算時間も，短くすることができる．このため，ビット幅の大きな浮動小数点乗算器等でよく用いられる[12)]．このほか，最近注目されているものに，ROMを用いた方法がある．これは，原理的には，乗算結果を前もってROMに格納しておき，乗数，被乗数が与えられるとそれによって番地を決定し，積を読出すものである．乗算時間は，ROMのアクセス時間だけとなり高速化でき，LSI化に有利であるが，ROMの容量が大きくなる．このため，1ビットずつ乗算を行う等，この容量を減らす方法が，種々研究されている[8)]．

以上，いろいろなレベルでの並列化について述べたが，現状では，パイプライン化が主流を占めている．その速度は，素子により変わり，TTLの場合，1段あたりの遅れが，200～600 ns，ECLの場合，50～200 nsとなっており，非常に高速なものとなっている．

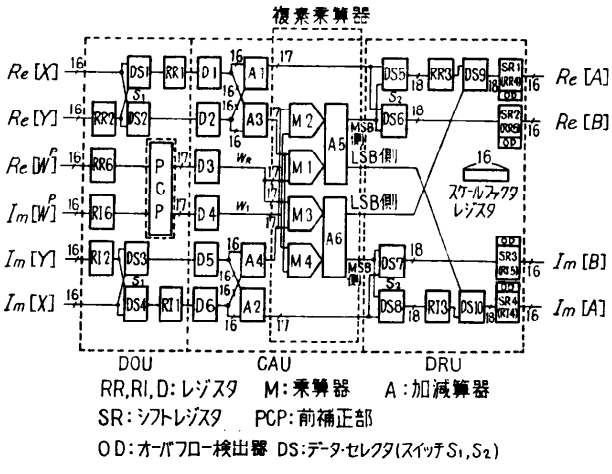

図-3 FFT演算器の構成

4.2 数値表現とその影響

数値表現は，そのプロセッサの演算精度を決定するほか，演算器や記憶の構成にも，大きな影響を及ぼす．現在のプロセッサでは，小型・中型では，固定小数点表現，大型では浮動小数点表現のものが多い．

a）固定小数点表現

固定小数点表現は，そのダイナミック・レンジが小さいが，演算器が簡単で，速度も早く低価格でできるうえ，ホストとなる汎用計算機との整合性も良い．このため，小型・中型のものでよく使われ，ビット幅は，11～18ビットのものが多い．また，演算器も，図-3に示すように複数個用いられることが多く，パイプライン化も乗算器について，2段程度に分割される．ダイナミック・レンジの拡大については，アレイ・スケーリングまたはブロック・スケーリングと呼ばれる方法が用いられることがある[11)]．これは，一つの配列の処理の途中で，桁あふれ，桁落ちが生じた場合，その配列全体を右または左へシフトし，これらを防止する方法である．何ビット・シフトしたかは，そのためのレジス

タへ貯えられており，スケール・ファクタと呼ばれる．

b）浮動小数点表現

最近の大型のプロセッサで採用され，ビット幅も，32〜38 ビット（指数部 8〜10 ビット，仮数部 24〜28 ビット）と大きなものとし，画像の処理や特徴抽出等の非常に繰り返しの多い処理でも，桁あふれ，桁落ちのおこらないよう配慮されている．このため，演算器は複雑となるが，徹底的なパイプライン化により，固定小数点演算に優るとも劣らぬ高速処理を実現している．パイプライン化は，乗算器はもちろんのこと，加算器についても行われ，乗算器で 3〜6 段，加算器で 2〜3 段程度に分割されている．乗算器，加算器数は，一般に少なく，基本システムでは，1 個ずつとし，オプションとして，4 個程度まで拡張するものが多い．ほとんどのものが，ECL を用いており，パイプライン 1 段あたりの遅れが，50 ns 以下のものもある．また，記憶も，バイポーラ素子を使用し，45〜150 ns のサイクル時間を持つ，高速のものを使用しているものが多い．

5. 制御方式とプログラミング

信号処理演算の高速化は，いろいろなレベルで，処理を並列化することにより実現されるが，いろいろな演算処理のおのおのについて，高能率の処理を行うためには，演算器や記憶，バスなどの構成を実行すべき演算に適合させる「アーキテクチャの柔軟性」が不可欠である．このため，ほとんどのプロセッサでは，プログラム制御を採用しており，プログラムによる制御は，演算のきわめて低いレベルにまで及んでいる．このために，最新のマイクロプログラム制御技術が，フルに活用されている．その中には，アーキテクチャを動的に再構成して，実行すべき演算に適合させるダイナミック・マイクロプログラミング[7]や，プロセッサ内部の算術演算ユニットや，番地発生ユニットをそれぞれ独立したマイクロプログラム制御のプロセッサとするような機能分化[13]など，汎用計算機で十分実用化されていないものも見られる．この理由としては，信号処理の演算は，同一の演算が数多く，繰り返されるため，演算の種類が変わるごとに，マイクロプログラムを書替えたりしても，そのオーバ・ヘッドは十分無視できるものになること，実行すべき演算処理の性質がよくわかっており，実装されている演算器等をフルに使うような最適化が行え，さらに，その効果も十分あがることなど，信号処理演算の特徴に起因するものが挙げられる．このほか，ユーザの望む非標準的な演算の実現が可能なこと，システムの拡張が容易で，システム寿命も伸びることなども挙げられる．また，この背景には，高速の ROM, RAM の価格低下により，長いビット幅のマイクロ命令の実装が容易となったこと，高速の ALUや PLA (Programmed Logic Array) の出現により高速の実行回路が容易に実現できるようになったことなどがある．

実際に用いられているマイクロ命令のビット幅は，16〜128 ビットと広範囲にわたり，演算器が増設されるごとに 32 ビットずつ増えるもの*などもある．その形式も垂直型から水平型まで各種あり，水平型のものでは，加算器，乗算器，記憶のアクセス，制御用ALUなどを並列動作させるために，それぞれ独立のフィールドを持っている．一例として，AP-120 B[14] のものを図-4 に示す．マイクロ命令の実行サイクル時間は，TTL のものでは，200 ns 以上，ECL では，50〜150 ns のものが多く，ほとんどのものでは，パイプライン演算器 1 段あたりの遅れと同一のサイクル時間を持っている．

これらのプログラミングについて見ると，ほとんどのものが，そのホストに，プログラムの作成，管理，実行時のローディングをまかせている．これらプロセッサの使用は，各ホスト上で使用できるサブルーチン・ライブラリを用い，FORTRAN プログラムからCALL

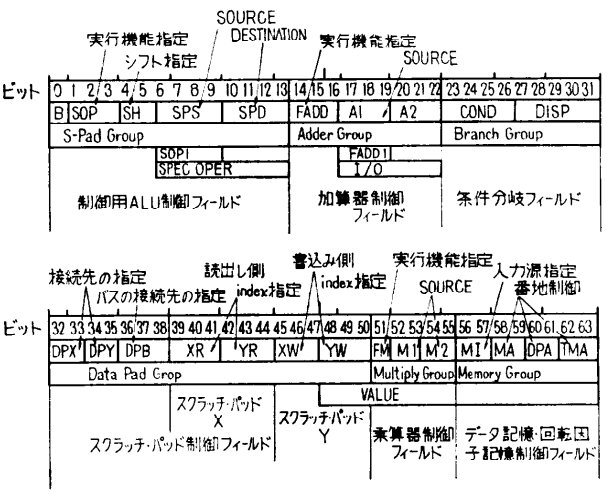

図-4 AP-120 B のマイクロ命令

* DATAWEST 社の REAL TIME III などがある．

することにより行える．このサブルーチン中に，準備されていない処理や，より能率のよい処理を必要とする場合には，ユーザ自身にマイクロプログラムを作成させるものが多く，このためにアセンブラ，シミュレータ，ディバッグなどが準備されている．しかし，マイクロプログラムのコーディングには，アーキテクチャの細部までの知識が不可欠であり，能率のよいプログラムの作成は難しく，より高級な言語の開発が望まれている．

6. 実際の信号処理用プロセッサ

信号処理用プロセッサの内部構成は，それぞれのプロセッサの適用分野や設計方針の違いから，多種多様なものがあり，汎用計算機のバス構成のような一般性のある構成は見あたらない．まず，その構成要素については，次のようなものが挙げられる．

構成要素

i）算術演算器（加算器，乗算器からなる）
ii）データ記憶（いくつかのバンクに分れている）
iii）回転因子記憶（データ記憶で兼用するものもある）
iv）番地発生器（制御用 ALU で発生するものもある）
v）制御用 ALU（入出力制御やホストとの交信制御のほか，演算処理にも使用される）
vi）制御装置と制御記憶（制御が分散しているものもある）
vii）ホスト・インタフェース
viii）入出力機器（AD 変換器，ディスプレイなど）

以下，これらが，どのように構成されているかを実際のプロセッサについて示す．

a）FDP[15]

FDP は，音声研究用に開発され，UNIVAC-1219 へ接続されている．その構成は，**図-5** に示すように，2 バンクからなるデータ記憶 (M^a, M^b)，4 つの算術演算器 (AE 1～AE 4)，制御記憶 (M^c)，番地発生器 (F) からなる．各 AE は，18 ビットの固定小数点演算を行う加算器，乗算器を 1 個ずつ含むほか，パイプライン処理を行えるよう 3 つの中間結果保持用レジスタが入っている．演算の制御は，M^c 中のプログラムにより行われ，1 命令は，18 ビット×2 の構成からなり，左半分は，データ記憶の番地を指定し，右半分は，4 つの AE 中の演算と AE 間のデータ転送を制御する．AE の接続は，バタフライ乗算と二次差分の計算に適したものとなっている．

b）KSP[10),11)]

KSP は，汎用信号処理を目ざし，著者らが設計したもので，FFT を中心とした処理を行う．その構成は，**図-6** に示すように，バスを中心とした構成をとり，これに，3 つのバンクからなるデータ記憶，**4.** で述べた FFT 演算器，制御用演算器 (CALU)，番地発生器 (AGU)，制御装置 (CU) などが接続されている．その制御は，CU 中の垂直型のマイクロ命令によって行われ，データの経路，演算の種類やビット幅，番地パタンなどを，演算処理が変わるごとに再構成し，処理効率をあげるよう工夫されている．

c）AP-120 B[14]

AP-120 B は，高速度とともに高精度を目ざし，38 ビットの浮動小数点演算を行う．その構成は，**図-7** (次頁参照)に示すように，3 段にパイプライン化された乗算器と 2 段にパイプライン化された加算器を 1 個ずつ持つほかに，高速小容量のスクラッチ・パッドを 2 組持っている．これらは，2 ポートのレジスタ・メ

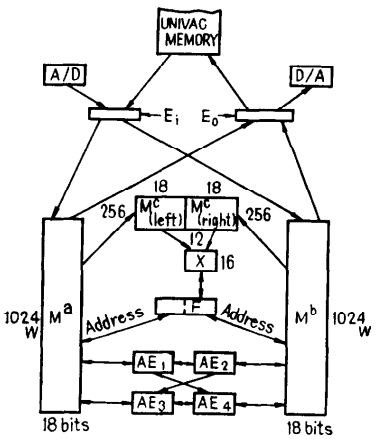

図-5 FDP の構成

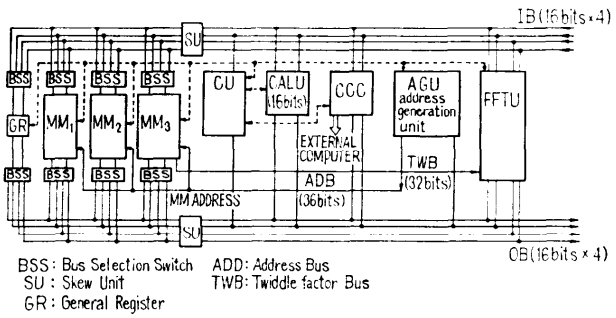

図-6 KSP の構成

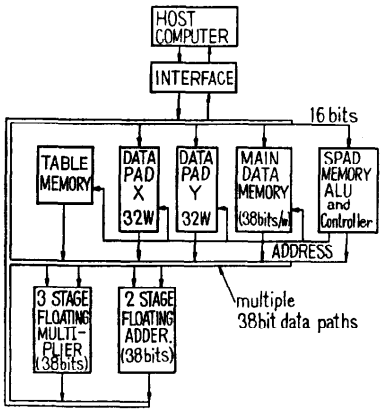

図-7 AP-120 B の構成

モリで，同時に読み書きが可能である．これを用いて，バッファリングを行うことにより，データ記憶に，500 ns のサイクル時間のものを使用しているにもかかわらず，高速の処理が実現されている．ホストとのデータ形式が異なる場合は，インタフェース中の変換回路により，転送時に変換され，固定小数点および異なる形式の浮動小数点表現でも使用できるようになっている．

a) MAP-300[13]

MAP は，大量のデータの高速処理を目ざす大型のプロセッサで，32 ビットの浮動小数点演算を行い，内部は，図-8 に示すように，2 つの算術演算プロセッサ (AP)，中央処理プロセッサ (CSPU)，3 つのバンクからなるデータ記憶，入出力制御プロセッサ (I/O SCROLL) などからなる．データ記憶の各バンクは，256 kB まで付加でき，8, 16, 32 ビットごとにアクセスできる．各 AP には，加算器，乗算器が 1 個ずつ含まれ，16 ビット幅の AP 制御マイクロプログラムで制御され，2 回の加算と 1 回の乗算を 415 ns で実行する．また，CSPU 中には番地発生器が含まれ，これ

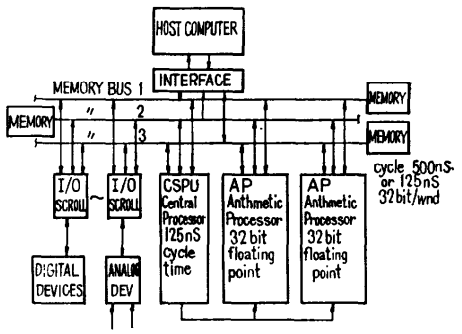

図-8 MAP-300 の構成

は，25 ビット幅の番地発生制御マイクロプログラムで制御される．全体の制御は，CSPU の 16 ビット幅のミニコン程度の命令セットにより書かれたプログラムで制御される．以上のように，内部は各プロセッサに分かれ機能分化している．また I/O SCROLL は，データ入出力のほか，ビット反転による並べ換えも実行する．AP 中の乗算器，加算器はパイプライン化されており，それぞれ 6 段，3 段に分かれている．

以上，4 種のプロセッサについて述べた．これらの性能を表-1 にまとめた．このほかにも，いろいろなプロセッサがあるが，最近のプロセッサでは，使用する論理素子，記憶素子の高速化が目立つ．また，MAP をはじめとして，演算器の数は，モデルにより異なり，少ないものから多いものへと段階的に拡張できるよう配慮されている．表-1 では，この種のものについては，最大構成のものについて示した．

7. おわりに

汎用信号処理を目ざすプロセッサを中心に，その演算，制御手法，構成などについて概説し，その中で行われている高速化，汎用化への努力について触れてきた．この中でも述べたように，現在のプロセッサ開発

表-1 プロセッサの性能

システム名	乗算時間 (ns)	演算ビット幅	メモリサイクル (ns)	命令実行時間 (ns)	命令ビット幅	FFT 実行時間 (ms)	素子	製造者
FDP	450	18 FIX	150	150	36	5.5	ECL	MIT
KSP	600/2	16 FIX	300	300	32	3.7	TTL	KEIO
AP-120 B	167/3	38 FLO	500	167	64	7.4	ECL	FLOATING POINT SYSTEMS
MAP-300	70/6	32 FLO	125	70	16, 25	4.5	ECL	CSPI
REAL TIME III	120/?	32 FLO	45	100	128	2.3	ショットキ TTL	DATAWEST

*1: 乗算時間は，パイプ 1 段のおくれ／分割段数で示す．？は未公表であることを示す．
*2: 演算ビット幅において，FIX は固定，FLO は浮動を示す．
*3: 命令ビット幅の 16, 25 は，2 種類以上あることを示す．
*4: FFT 実行時間は，1024 点の複素 FFT の場合を示し，並べかえは含んでいない．

の問題点は，乗算器を含む演算器であり，この解決策として，乗算器やバタフライ乗算器などの LSI 化に，大きな期待がかけられている．LSI 化は，演算器の小型化，低価格化のみならず，これを多数用いることによる高速化を可能とし，現在のプロセッサの速度をさらに1桁以上向上させることが期待できる．また，これとは別に，従来，その速度と価格のために用いられなかった，レーダやソーナ，多重通信などの実時間処理の分野での実用化に向って，一歩前進し，各種通信機器へ部品として組み込まれることが期待できる．このためには，なお一層の高速化，小型化，低価格化が必要であり，新しい並列処理の方法や，乗算アルゴリズム，広範な用途に使用できる演算器構成法等の研究がさらに盛んになり，ディジタル信号処理の分野をますます活気あるものにしていくと思われる．

参考文献

1) J. W. Cooley, J. W. Tukey: An Algorithm for the Machine Calculation of Complex Fourier Series, Math. of Comput., Vol. 19, pp. 297～301 (Apr. 1965).
2) G. D. Bergland: Fast Fourier Transform Hardware Implementations-A Survey, IEEE Trans., AU-17, 2, pp. 109～119 (June 1969).
3) G. D. Bergland: Fast Fourier Transform Hardware Implementations-An Overview, IEEE Trans., AU-17, 2, pp. 104～108 (June 1969).
4) 電子通信学会編：ディジタル信号処理，電子通信学会，昭和50年11月.
5) W. M. Gentleman, G. Sande: Fast Fourier Transforms-For Fun and Profit, Proc. of FJCC 1966, pp. 563～578.
6) E. O. Brigham: The Fast Fourier Transform, Prentice-Hall Inc., (1974).
7) J. Allen: Computer Architecture for Signal Processing, Proc. IEEE. Vol. 63, 4, pp. 624～633 (Apr. 1975).
8) B. Liu, A. Peled: A New Hardware Realization of High-Speed Fast Fourier Transformers, IEEE Trans. Vol. ASSP-23, 6, pp. 543～547 (Dec. 1975).
9) S. L. Freenly: Special Purpose Hardware for Digital Filtering, Proc. IEEE, Vol. 63, 4, pp. 633～648 (Apr. 1975).
10) 内田他：高速パイプライン信号処理装置のマイクロプログラム制御，信学論 D, 58-D, 6, pp. 328～335 (June 1975).
11) 所 他：ハードウェアによる高速フーリエ変換，信学論 D, Vol. 58-D, 9, pp. 578～585 (Sept. 1975).
12) S. F. Anderson, et al.: The IBM System/360 Model 91: Floating-Point Execution Unit, IBM Journal, pp. 34～53 (Jan. 1967).
13) An Introduction to the MAP Series, Models 100, 200 & 300, Computer Singnal Processor Inc. (USA), (Feb. 1975).
14) AP-120B Processor Handbook, Floating Point Systems, Inc. (USA), (May 1976).
15) B. Gold, et al.: The FDP, a Fast Programmable Signal Processor, IEEE Trans., C-20, 1, pp. 33～38 (Jan. 1971).
16) A. S. Zukin and S. Y. Wong: Architecture of a realtime fast fourier radar signal processor, Proc. of AFIPS SJCC 1970, pp. 417～435 (1970).
17) M. J. Corinthios, et al.: A Parallel Radix-4 Fast Fourier Transform Computer, IEEE Trans., C-24, 1, pp. 80～92 (Jan. 1975).
18) 高速フーリエ変換とチャープZ変換を巡って，日経エレクトロニクス，1976年9月20日号，pp. 58～83.
19) 通信の分野に浸透するディジタル信号処理技術，日経エレクトロニクス，1976年11月29日号，pp. 42～68.

(昭和51年12月13日受付)
(昭和52年2月8日再受付)