FACOM230－75アレイプロセッサについて

上田徹　乾範男　内田啓一郎
(富士通)

はじめに

Computerを、専用processorと、汎用processorとに分けるきえが、はやっている。汎用processorとは、世の中の標準的なneedに合せて開発されるcomputerのことで、小型機から大型機まで、世の中の大抵のcomputerは、この分類に入る。それに対して、専用processorとは、ある特定の領域に対して、汎用computerとは比較にならないcost/performanceを狙って開発されるcomputerだとされている。その専用computerの代表的なものとして、Array Processorがある。Array Processorは、記憶装置上にあるArray Dataを、従来の汎用Computerとは比較にならないほどの非常な高速で実行するものである。

Array Processorが出現した社会的背景

従来のcomputerでは、cost/performanceの制約から、十分な処理能力を発揮しえず、その為、各界の科学技術者が、非常に長い計算機時間を使って問題を処理したり、場合によっては、理論的には解決がつき、その問題を解くことによって、著しい社会的効用が期待できるにもかかわらず、計算機時間の問題の為、解決を見送っているような問題等があります。このような社会的背景を考えますと、Array Processor等、超高速computerを世に送ることは、その台数に対する需要の大小にかかわらず、富士通にとって極めて大きな社会的義務であると考えます。

ハードウェアの構造

APの超高速性は、徹底したパイプライン制御によるノクロックノオペレーションの実現にあります。

その為には、

- 主記憶装置の高度なインタリーブを行ない、主記憶装置のデータ供給能力を充分高くすること。
- APの主記憶アクセス制御部は、高度の先行制御によりオペランドデータの先取りを行ない、充分バッファリングを行なうことによって、コンスタントにデータをアレイ処理部へ供給できること。
- APのアレイ処理部には、各種超高速演算回路を多数必要としますが、これらの演算回路を超高速のLSIによりコンパクトに実現すること。

等の裏付があってはじめて実現されるものです。

次の図は、主記憶装置とAPとの関係を概念的に表わしたものです。APのアレイ処理部で、

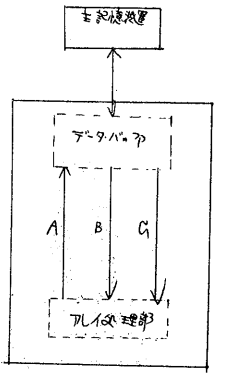
```
DO/  N=1,1000
/  A(N)=B(N)+C(N)
```

を実行するとしますと、/図に示しますように/サイクルに2つのオペランドを受取り(B、Cバス)、/つのオペランドを出力する(Aバス)必要があります。このデータの流れを主記憶装置とアレイ処理部の間で、定常的なスムーズな流れとする

為、APの主記憶アクセス制御部に大量のデータバッファがあり、データのバッファリングを行なっています。

主記憶装置は、1台(256KW)で8ウエイのインタリーブを行なうことができ、これが4台まで接続できますので、全体で32ウエイのインタリーブとなります。この場合APのノクロックノオペレーションに必要なデータの供給を行なうことができます。

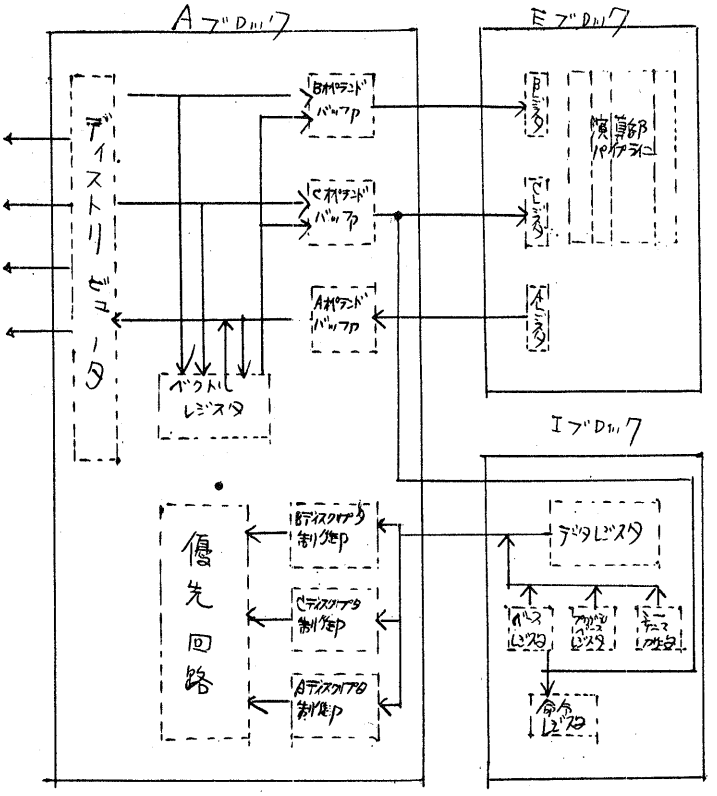
※FACOM230-75の主記憶装置は、2W単位のインタリーブを行なっています。



APのアレイ処理部は、ノクロック毎にオペランドを受け取り、高度にパイプライン化された演算器に流し込みます。演算結果は、演算パイプラインの最終段からノクロック毎に出力されます。演算器の主なものは、高速乗算器と高速加算器でありますが、これらを構成する基本素子はLSIで構成され、回路数はアレイ処理部のみでFACOM230-75 CPU相当ゲート数となりますが、このような膨大な回路を要する装置も、高密度なLSIがあって初めて実現するものです。

2図は、APのブロックダイアグラムであります。APはAブロック(Access Control Block)、Iブロック(Instruction Control Block)、Eブロック(Execution Control Block)の3ブロックで構成されています。

Aブロックは、主記憶アクセス制御、データプリフェッチ及び、バッファリングを制御します。Iブロックは、命令読出し制御とベクトル演算以外の一般命令の実行とを行ないます。Eブロックは、APのいわゆるアレイ処理部で、ベクトル演算命令を実行します。



2図　APのブロック・ダイアグラム

メモリアクセス制御とデータのバッファリング

メモリアクセス制御とデータのバッファリングは、Aブロックで行なわれています。

Aブロックメモリ制御部

Aブロックメモリ制御部は、APのデータの受けわたしに関する一切の仕事を行ないます。

APからのメモリアクセスの要求の頻度は極めて高く、メモリの競合状態の発生が無視しえなくなり、記憶装置の読み出し能力を100%利用する制御が必要となります。そこで、メモリアクセス要求元からくるメモリアクセス要求を、入ってきた順序に制御するだけでは不十分となり、後から入ってきたメモリアクセス要求でも、メモリバンクがあいている場合には、先に処理をするという制御を行ないます。このようなメモリアクセス順序の転倒化のことをフォワーディング(Forewarding)といいます。

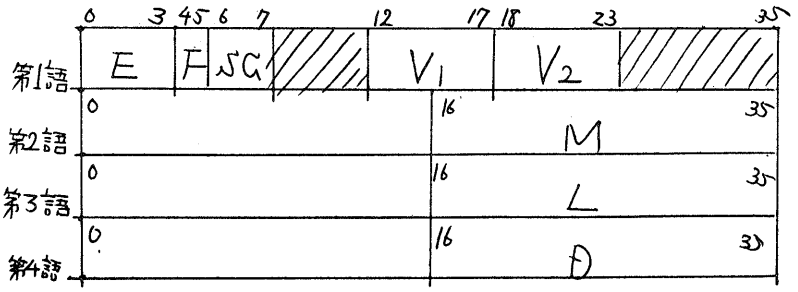
このようなフォワーディングを行なうことにより、記憶装置の能力を100%利用できます。

Aユニットディスクリプタ制御部

ベクトル演算命令におけるオペランドの指定は、ディスクリプタを通じて行なわれます。これらのディスクリプタの制御は、Aブロックディスクリプタ制御部で行ないます。すなわちディスクリプタは、ベクトルの記憶装置上の位置、ベク

トルの大きさ、ベクトルの種類、データの種類を指定します。

㊍ディスクリプタは、オペランドベクトルについて、ベクトルの種類、データの種類、記憶装置上での位置、大きさ等を詳細に記述するもので、一般形は次の通りです。



E：SCALAR、VECTOR、LIST等ベクトルの種類を指定する。

F：固定小数点、浮動小数点、浮動小数点等データの種類を指定する。

SC：ソース　オペランドの符号を制御する。

V_1、V_2：インデックス　レジスタ指定する。

M：ベクトルのアドレス部を指定する。

L：ベクトルのエレメント部を指定する。

D：ベクトルのエレメント間の距離を指定する。

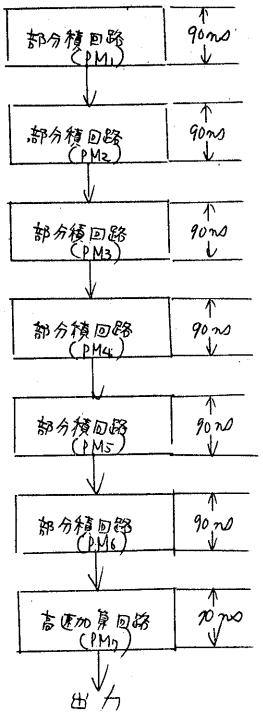
アレイ演算部のパイプライン構成と演算方式

APのアレイ演算部は、Eブロックと呼ばれ、高速乗算器と高速加算器から構成されます。この演算器は、いずれも浮動小数点を扱うもので、各々は幾つかのステージを持つパイプライン構成になっていて、1クロック毎に2倍精度演算のリザルト（Result）を出力することができます。

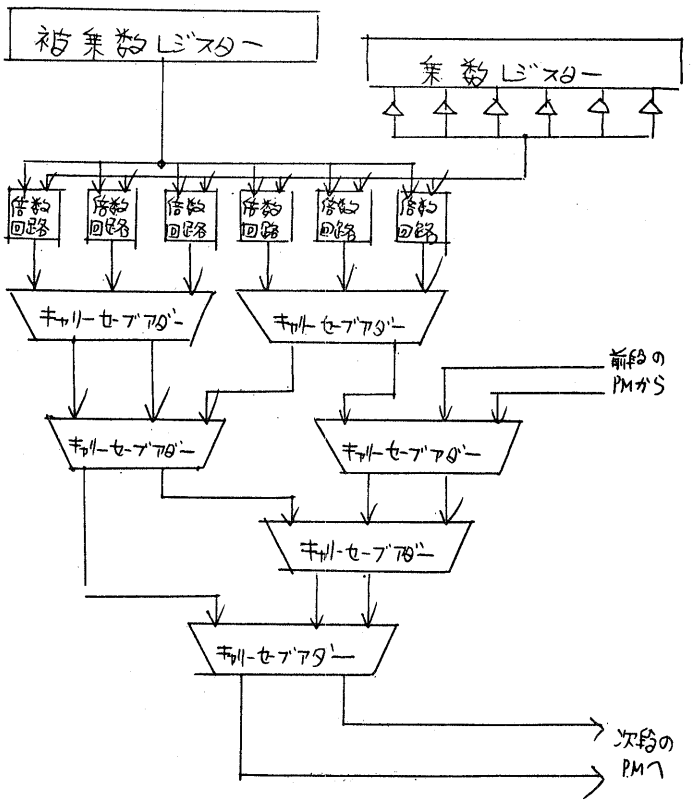
高速乗算器

高速乗算器は、3図に示すように、6個の部分積回路（PM：Partial Multiplicator）と高速加算回路（CPA：Carry Propagation Adder）とから構成されています。PMは72ビットのデータと12ビットのデータの乗算を、1クロックタイムで行なうことができ、6個のPMを合せると2つの72ビットのデータの積の中間結果が得られます。この中間結果をCPAで加算することにより積が求まります。CPAは72ビットの高速加算回路で、1クロックタイムで加算を実行することができます。

PMは、4図のように被乗数の為の72ビットのレジスタMLCD、乗数の為の12ビットのレジスタMLR、倍数回路Mキャリーセーブアダー（Carry Save Adder）CSAとから構成されます。



3図　高速乗算の構成



4図　部分積回路(PM)のブロックダイアグラム

高速加算器

ここで言う加算器とは、浮動小数点数値の加算器であり、2倍精度の加算をノクロックノオペレーションで実現するものです。5図に示すように、加算器は、5つのステージより構成されています。各ステージは、ノクロックタイムで部分演算を実行することができます。

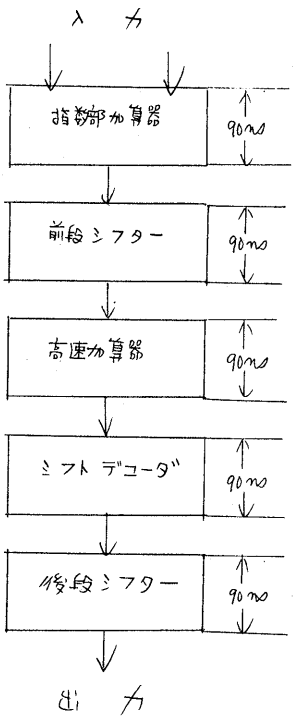
(1) 指数部加算器(Exponent Adder:EXA)
EXAは、9ビットの加算回路が主たる構成要素です。ここでは2つの入力オペランドの指数値を計算します。

(2) 前段シフター(Pre-shifter:PRS)
PRSは、72ビットの高速右シフト回路で、ノクロックタイムで72ビットの右シフトを行ないます。

(3) 高速加算器(High Speed Adder:HSA)
HSAは、72ビットの高速加算回路で高度のキャリールックアヘッド(Carry Look Ahead)を行ない、ノクロックタイムで加算するようになっています。ここでは仮数部の加算を行なっています。

(4) シフトデコーダ(Shift Decoder:SHD)
SHDは、正規化前の仮数部を左端から走査して、正規化に必要なシフト数を検出するシフトデコーダであります。

(5) 後段シフター(Post-shifter:POS)
POSは、72ビットの高速シフト回路で、左シフトは72ビットまで、右シフトはノビットまで、各々ノクロックタイムでシフトすることができます。



5図 高速加算器の構成

ベクトル演算の制御方式

三プロックには2倍精度浮動小数点数値を取扱う前述の乗算器と加算器がありますが、これらの演算器を利用して各種ベクトル演算が実行されます。

(1) ベクトル加算(Ai ← Bi + Ci)
加算器にベクトルB、Cの対応する要素の組(B1、C1)、(B2、C2)、……を連続的に流し込むことにより、和のベクトルAの各要素A1、A2、……がノクロックタイム毎に求められます。

(2) ベクトル乗算(Ai ← Bi × Ci)
乗算器にベクトルB、Cの対応する要素の組(B1、C1)、(B2、C2)、……を連続的に流し込むことにより、積のベクトルAの各要素A1、A2……がノクロックタイム毎に求められます。

(3) ベクトル除算(Ai ← Bi ÷ Ci)
除算は乗算器を用い、複数回の乗算に置き替えて行ないます。
いま、 $Q = \frac{N}{D}$ (D ≠ 0)

において、簡単の為、N、Dを正の正規化された浮動小数点の仮数としますと、N、Dは次の範囲に入っています。

$$\frac{1}{2} \leqq N < 1 \qquad \frac{1}{2} \leqq D < 1$$

ここでD = 1 − xと置きますと、xについて次式が成立します。

$$0 < x \leqq \frac{1}{2}$$

次にRi, Diについて下記のように定義しますと、

$$R_{i+1} = 2 - D_i \quad (D_0 = D)$$
$$D_{i+1} = D_i \cdot R_{i+1}$$
$$(i = 0, 1, \cdots\cdots)$$

Ri、Diは次のような数列をなします。

$$R_1 = 1 + x \qquad D_1 = 1 - x^2$$
$$R_2 = 1 + x^2 \qquad D_2 = 1 - x^4$$
$$\vdots \qquad\qquad \vdots$$
$$R_i = 1 + x^{2^{i-1}} \qquad D_i = 1 - x^{2^i}$$

iを充分大きくとりますと、$0 < x < \frac{1}{2}$でありDiは1に近づきます。そこで$\frac{N}{D}$の分子、分母にR1、R2 …… Riを乗しますと次式からわかる通り、分母はDiとなります。

$$\frac{N}{D} = \frac{N \cdot R_1 \cdot R_2 \cdots\cdots R_i}{D \cdot R_1 \cdot R_2 \cdots\cdots R_i} = \frac{N \cdot R_1 \cdot R_2 \cdots\cdots R_i}{D_i}$$

ので、iを充分大きく取りますと、$D_i \fallingdotseq 1$となりN、R1、R2 …… Riが$\frac{N}{D}$の商となります。

i = 7とすれば、$1 - D_7 = x^{128} < 2^{-128}$となり2倍精度浮動小数点数の精度を保つことができます。この場合R2、R3 …… R7を求めるための6回の乗算と、N・R1・R2 …… R7を求めるための7回の乗算により、$\frac{N}{D}$の商が求まります。

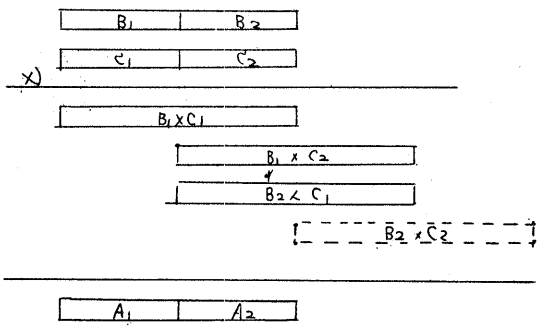
実際には、R1はDの上位8ビットをデコードして求め、Di = D・Riが2^{-}程度の誤差範囲で1に近づくようにする為、乗算の回数はもっと少なくなっています。

(4) 4倍精度ベクトル

4倍精度のオペランドは、2つの2倍精度のオペランドに直して演算します。

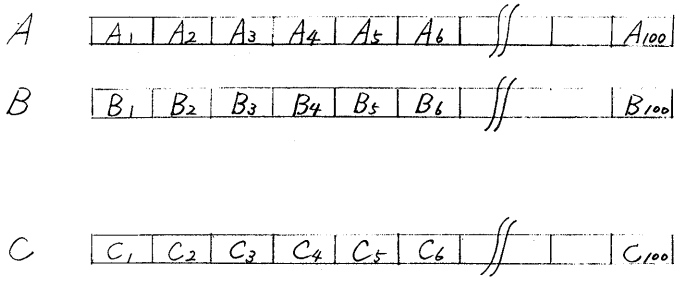
4倍精度の加算は、2倍精度の2回の加算に置きかえることができます。

4倍精度乗算は、3回の2倍精度乗算に置きかえることができます。次図で$B_2 \times C_2$の部分の乗算は省略します。



6図 4倍精度ベクトルの乗算方法

Array Processor の Speed の秘密



上図で示しますように、2つの入力 operand vector A、Bの各要素毎の加算を行ない、結果を destination operand vector Cに格納していくような operation を考えてみたいと思います。
この操作を、FACOM230-75 Array Processor の機械語で conding いたしますと、1命令になります。アレイプロセッサは、1クロック1データのパイプライン設計がなされていますので、ほぼ100クロック900nsで実行することができます。これを汎用 computer の命令セットで実行いたしますと、大体1データ3~4命令必要となり、全体で300~400命令が必要となります。FACOM230-75の平均命令実行時間を260nsとしますとほぼ、全命令を実行するのに1μsかかります。
以上から Array Processor が、汎用 computer の10倍の処理能力があることが分ると思います。