汎用計算機のための内蔵ベクトル演算方式†

堀越 彌†† 梅谷征雄††

より高い性能の汎用計算機に対する期待は根強いものがあるが，従来の命令体系（アーキテクチャ）の上での性能向上はしだいに困難になってきている．一方，科学技術計算の分野では，従来からベクトル演算を主体としたスーパーマシンまたはベクトルプロセッサが使われてきている．汎用機の特徴を失うことなく，ベクトルプロセッサの概念を導入し，高速の科学技術計算を可能とするような内蔵ベクトル演算方式を開発した．開発したベクトル演算方式では，計算機利用者から見て，その機能のある無しは高い性能という点を除いては見えないようになっている．すなわち，アドレッシング方式，割込み方式等は汎用機と同じ枠組みに収め，自動ベクトルコンパイラによってFORTRANコードからベクトルコードに自動的に変換されるため，計算機利用者はベクトル演算機能を利用するための特別の準備はいっさい不用である．また，汎用計算機という性格から，このためのハードウェアの増加を最小限とするため命令種も単・倍精度を含めて28に絞った．この方式は，HITAC M 180/200 H に内蔵アレイ・プロセッサ (IAP : Integrated Array Processor) 機能として実用化したところ，ベクトル演算能力評価用のループミクスで1.7～3.4倍，線形計算プログラムで1.6～3.4倍の性能向上を実現することができた．一般のバッチ計算センタでも40%程度の性能向上が可能と思われる．半導体技術の進展により回路コストの著しい低下が進んでいるので，近い将来には，現在のスーパーマシン並みのベクトル演算機能を汎用計算機に付与することが可能になると思われるので，このような方式の研究は，将来ますます重要になろう．

1. まえがき

より高い性能の計算機に対する要求はきわめて根強いものがあり，計算機技術のほうもそれに答えて着実な進歩を遂げてきた．最近の大型計算機では，10～20 MIPS (million instructions per second) の性能も可能になっているが，この高速性はおもに先端的半導体技術と，きわめて巧妙な先行制御論理方式によっており，これらをさらに改善していくことが，しだいに困難になってきているのも事実である．現在の汎用計算機では命令間の依存関係（たとえば前の命令の結果を次の命令のソース・オペランドとして用いるなど）は陽には示されないので，先行制御を進めていこうとすると並列処理の可能性を検出するための過度の負荷がかかってくる．これを軽減するために，ハードウェアとソフトウェアの境界（マシン・コード）をより上位へ移し，多数オペランドの処理を単位とするベクトル命令を導入し，より高い性能の計算機を実現しようと考えるのはごく自然であるといえる．

ベクトル命令の概念は，科学技術計算用の専用機，いわゆるスーパーコンピュータでは比較的以前から導入されていたが，汎用機では実現されていなかった[1),2)]．その理由は，

（1） ベクトル命令の論理仕様（アーキテクチャ）が，事務処理からトランザクション処理までの多様な用途を念頭においた汎用計算機の論理仕様になじまない．

（2） 一般にベクトル命令の高速性を生かすには，既存プログラムの改変が必要であり，この利用はその作業をいとわないごく少数の数値計算プログラマに限られる．

（3） ベクトル命令の高速性を実現するために処理装置の規模が増大し，高価になる．

などによっていると考えられる．

そこで筆者らは，これら従来の問題点を解決するために，

（1） 汎用計算機向きのベクトル演算の論理仕様を開発すると同時に，

（2） 既存のFORTRANプログラムを自動的にベクトル・コードに落とす自動ベクトル化コンパイラを開発し[9)~11)]，

（3） 汎用機に存在する高速演算機構とマイクロプログラム（ファームウェア）機構を生かし，付加ハードウェアを最小限に押さえた内蔵型ベクトル演算機構を開発した[5)~8)]．

これらのベクトル演算機能 (IAP 機能 : integrated array processor 機能とも略称される) は，HITAC

† Integrated Vector Arithmetic Facility for General-Purpose Computer by HISASHI HORIKOSHI and YUKIO UMETANI (Central Research Laboratory, Hitachi Ltd.).

†† (株)日立製作所中央研究所

M-180 (1978年), M-200 H (1980年), 用として実用化され[3),4)], 今日までに数多くのプログラムでその有効性が実証されるまでになった[13)~15)].

本稿では, 汎用計算機向きのベクトル演算アーキテクチャと, 得られた性能測定結果について報告する. このベクトル演算機能のもう一方の重要な柱である自動ベクトル・コンパイラについては別途報告の予定である[16)].

2. 汎用計算機用ベクトル演算機能の満たすべき要件

2.1 ベクトル演算機能の透明性の実現

ベクトル演算機能の実現に際し, 最も重視したのは透明性の実現である. ここでいう透明性とは, 他のサブシステムがその存在, 非存在を意識しなくて済むということであるが, より具体的には,

(1) ユーザプログラム(FORTRAN プログラム)からみて, ベクトル演算機能の存在が見えないようにすることを目標とした. その理由は, (a)こうすることにより, 大多数のプログラムが大なり小なりベクトル演算機能による性能改善の恩恵にあずかれること, (b)また, ベクトル演算機能を生かすために作成したプログラムが, 他の計算機では処理できないというユーザの不安を除去できること, (c)さらに, ベクトル演算機能により向いたプログラム技法を提示することにより, 徐々にユーザがベクトル演算に慣れ, 近い将来のより大きな飛躍が可能と考えたことによる. また,

(2) 制御プログラムからの透明性の実現, にも配慮した. これは, 一つには, ベクトルコードを含むジョブまたはタスクを制御プログラムが特別にスケジュールしなくても済むことを意味しており, ベクトルコード実行中は割込みが禁止されるといった事態もなくなる. この意図は, ベクトル命令が長時間を要する科学技術計算専用のものではなく, 長期的には TSS のフォアグラウンドで走るジョブはもとより, さらには, システム・プログラムにまで適用範囲が拡大されるべきであるというところにある. 事実, 最近になって, ベクトル演算機能は最も TSS 向きの APL にも適応されるようになってきている[12)].

これら二つの基本的要請は**図1**に示すような論理仕様(アーキテクチャ), コンパイラ方式, 処理装置方式上の具体的要件として, 内蔵ベクトル演算機能に反映

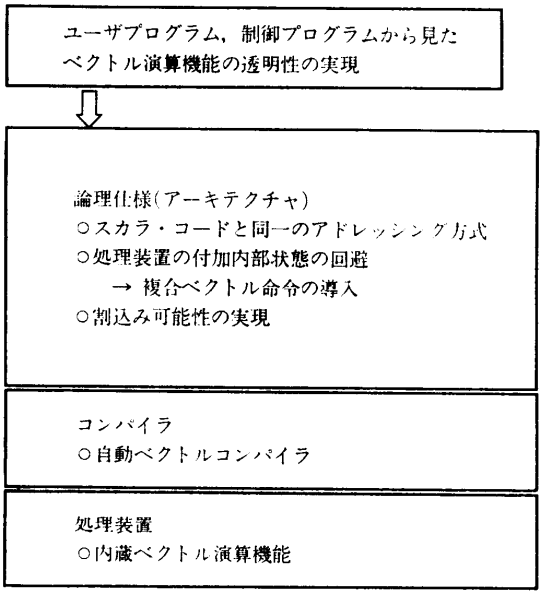

図1 内蔵ベクトル演算機能のアーキテクチャの概念図
Fig. 1 Architecture of integrated vector arithmetic facility.

される必要があったが, とくに論理仕様を取り上げ以下にその基本的考え方を示す.

2.2 アドレッシング方式

現在の汎用計算機は, 仮想記憶装置を可能とする論理アドレス方式を標準としているが, ベクトル処理を高速化するためには物理アドレス方式に魅力があり, そのような方式を採用するベクトルプロセッサもある[17)]. その理由は,

(1) 論理アドレス方式では, アドレス変換といった余分な処理が加わるため, 十分な速度でオペランドを読みだすのがむずかしくなること. ベクトルが連続領域に存在しないような場合にはアドレス変換の頻度が高まりとくに深刻になる.

(2) また, アドレス変換に伴うページ・フォルトなどの割込みは一般に抑止型で, 割込み処理が終わった段階でそこから再開始できる必要があるが, ベクトル処理装置は複雑な先行制御を行っているため, このような任意地点での停止, 再開がむずかしいこと,

などにある. しかし, 今回開発したベクトル演算機能では, これらの困難を覚悟の上で論理アドレス方式を採用することにした. これは, 2.1 節で述べた透明性を実現するためには必須と考えたからである.

2.3 適切な命令形式の選定

ベクトル命令の選定でまず問題になるのは, 基本的

な命令の形式を，

（1） 処理装置内のレジスタと記憶装置との間の演算を指定する RS 形式 (register to storage) にするか，

（2） 記憶装置と記憶装置の間の演算を指定する SS 形式 (storage to storage) にするか，

という問題がある．RS 形式は一方のオペランドをレジスタより読みだし，またはレジスタへ書き込める分だけ処理装置と記憶装置との間のデータ転送頻度が緩和され，一般にこの部分が隘路となるベクトル処理装置ではきわめて望ましいことといえる．しかし，RS 形式の欠点は，処理装置内にいわゆる大量のベクトルレジスタを内部レジスタとしてもつことになり，割込み時の待避回復が困難になることである．今回のベクトル演算機能では，2.1 節で述べた透明性を阻害する後者の欠点を重く見て，SS 形式 (storage to storage) を採用することとした．しかし，SS 形式によって増大した処理装置と記憶装置間の負荷を軽減するために複合演算命令を導入した．これについては次章で述べる．

2.4 割込み可能性の実現

2.1 節で述べた透明性を実現するには，十分な精度で割込みが可能でなければならない．割込みの精度に対する従来の汎用計算機の考え方は，基本的には，

（1） 命令境界での割込みと，

（2） 複合命令的な実行時間の長い命令においては，これを幾つかの操作単位 (unit of operation) に分け，その境界で許す割込み，

との二つのから成り立っている．ベクトル命令においても命令境界での割込みは，各種事象への応答性を保証するため当然許されなければならないが，今回開発した内蔵ベクトル演算方式では仮想記憶方式に伴うページ・フォルトの処理のために，さらに各操作単位境界での割込みも可能とした．後者はとくに他のベクトルプロセッサに例を見ない特徴的な方式である．

3. 内蔵ベクトル演算機能のアーキテクチャ

3.1 機能の概要

この章では，筆者らが HITAC M-180/M-200H 用に開発した内蔵ベクトル演算機能のうち，ソフトウェアとハードウエアのインタフェースを規定するいわゆるアーキテクチャ（論理仕様）のうち，特徴的部分について詳細に記す．内蔵ベクトル演算機能は，すでに図1に示したように，

（1） アーキテクチャ（論理仕様）

と，

（2） この論理仕様に規定されたベクトル命令列を生成する自動ベクトルコンパイラ，

```
00001          DIMENSION A(100,100),B(100,100),C(100,100)
00002          DO 10 I=1,N
00003          DO 10 J=1,N
00004          DO 10 K=1,N
00005       10 C(I,J)=C(I,J)+A(K,I)*B(K,J)
00006          STOP
00007          END

010666 5800 C040  00003  100003  L    0,=F'0'
01066A 5000 C018                 ST   0,J0002
01066E 1200                      LTR  0, 0
010670 4700 C110                 BC   13,=X'00004'
010674 5830 C048         100005  L    3,=F'0'
010678 5A30 D08C                 A    3,=F'4'
01067C 5820 C030                 L    2,=F'0'
010680 5A20 C02C                 A    2,=F'0'
010684 5840 C024                 L    4,=F'0'
010688 5A80 C018                 L    8,J0002
01068C 1892                      LR   9, 2
01068E 5AA0 C044                 L    10,=F'0'
010692 1244              00004   LTR  4, 4
010694 4700 C108                 BC   13,=X'00007'
010698 41E3 6000         100008  LA   14,A
01069C 50E0 D058                 ST   14,V0001
0106A0 41EA 7000                 LA   14,B
0106A4 50E0 D060                 ST   14,V0002
0106A8 1B00              00005   SR   0, 0
0106AA 5810 D0A4                 L    1,N
0106AE 2800                      SDR  0, 0
0106B0 7809 B000                 LE   0,C
0106B4 41F0 D068                 LA   15,=A(001
0106B8 A40F 0040                 VIPE 0,15
0106BC 7009 B000                 STE  0,C
0106C0 1A95              100007  AR   9, 5
0106C2 1AA5                      AR   10, 5
0106C4 4680 C00A                 BCT  8,=X'00006'
0106C8 5800 C02C         100004  L    0,=F'0'
0106CC 5A00 D08C                 A    0,=F'4'
0106D0 5000 C02C                 ST   0,=F'0'
0106D4 5800 C048                 L    0,=F'0'
0106D8 1A05                      AR   0, 5
0106DA 5000 C048                 ST   0,=F'0'
0106DE 5820 C014                 L    2,=F'0'
0106E2 5820 C010                 S    2,=F'1'
0106E6 5020 C014                 ST   2,=F'0'
0106EA 4760 C0AE                 BC   6,=X'00003'

(001)  ISN OF DO : 00000004  LABEL : 10
       JMK8001I VECTORIZABLE DO LOOP.
       (1) TEMPORARY VECTOR SIZE = 00000 BYTES
       (2) INDEX VARIABLES    : K
       (3) RELATIVE CONSTANTS : I    ,J
```

図 2 ベクトル内積計算のソース・プログラムとベクトル命令を含むオブジェクト・プログラム

Fig. 2 Source code and vectorized object code of vector inner product calculation.

それに，

（3） 論理仕様に規定されたベクトル命令を実行するハードウェアとしての内蔵ベクトル演算機構

から成り立っている．

図2にはベクトル内積計算を例として，ソースプログラムとベクトルコンパイラによって自動的に生成されたベクトル命令を含むオブジェクトプログラムが示してある．ベクトル命令は，従来のスカラ命令列内に，図2に示すように区別なく埋め込まれており，処理装置内の命令解読部がベクトル命令を認識すると，内蔵ベクトル演算機構内の必要な論理回路を用いてその命令を処理する．後続命令がスカラ命令の場合には，ごく自然な形で従来の動作に戻ることになる．

3.2 基本命令形式

ベクトル命令は**図3**に示すように32ビット（4バイト）からなり，最初の1バイトはベクトル命令であることを示し，ベクトル命令の詳しい種別はうしろ2バイトの OP (operation) 拡張コードで与えられる．2バイト目は R1，R3の二つの汎用レジスタを指定するために用いられている．R1は R1+1と組で用いられ，R1はベクトル内の開始要素番号を，R1+1はベクトル長を指定する．R3には，OAV(operand address vector) と呼ばれポインタ・テーブルのアドレスがセットされており，OAV には各ベクトル対応に存在する VDT (vector descriptor table) のアドレスが格納されている．VDT はそのベクトルの要素の先頭アドレスと要素間の増分値を与える．通常，ベクトル命令の実行開始時は R1の内容は零で，VDT 内の要素先頭

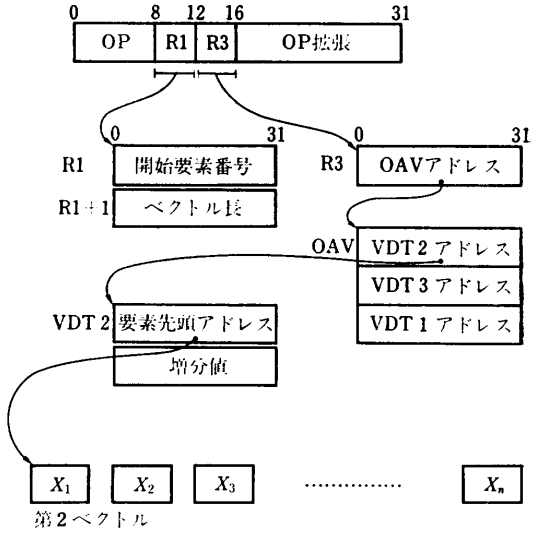

図 3 ベクトル命令の形式
Fig. 3 Format of vector instruction.

アドレスを用いて最初のベクトル要素にアクセスする．1操作単位が完了すると，R1の内容に +1 され，次の要素は

(要素先頭アドレス)＋(R1)＊(増分値)

でアクセスされる．いずれかの操作単位境界で割込みが発生した場合には (R1) の内容は汎用レジスタの一部として待避されるので，再度，そのベクトル命令に制御が渡った場合には，後続の操作単位から動作が再開されることになる．このようにして本内蔵ベクトル演算機能では命令境界，操作単位境界での割込みを保証している．

表 1 ベクトル命令一覧
Table 1 Summary table of vector instructions.

命令名称	略称 短精度	略称 倍精度	演算語の S/V 1	2	3	動作概要
Vector Move	VME	VMD	V	S/V		$Z_i \leftarrow X_i$
Vector Elementwise Complement	VECE	VECD	V	S/V		$Z_i \leftarrow -X_i$
Vector Elementwise Add	VEAE	VEAD	V	S/V	S/V	$Z_i \leftarrow X_i + Y_i$
Vector Elementwise Subtract	VESE	VESD	V	S/V	S/V	$Z_i \leftarrow X_i - Y_i$
Vector Elementwise Multiply	VEME	VEMD	V	S/V	S/V	$Z_i \leftarrow X_i * Y_i$
Vector Elementwise Divide	VEDE	VEDD	V	S/V	S/V	$Z_i \leftarrow X_i / Y_i$
Vector Element Sum	VSME	VSMD	S	S/V		FPR $\leftarrow$ FPR $+\sum X_i$
Vector Element Sum with Complement	VSMCE	VSMCD	S	S/V		FPR $\leftarrow$ FPR $-\sum X_i$
Vector Inner Product	VIPE	VIPD	S	S/V	S/V	FPR $\leftarrow$ FPR $+\sum X_i * Y_i$
Vector Inner Product with Complement	VIPCE	VIPCD	S	S/V	S/V	FPR $\leftarrow$ FPR $-\sum X_i * Y_i$
Scalar Multiply and Add	VSMAE	VSMAD	V	S	S/V	$Z_i \leftarrow Z_i + X * Y_i$
Scalar Multiply and Subtract	VSMSE	VSMSD	V	S	S/V	$Z_i \leftarrow Z_i - X * Y_i$
First Order Iteration	VITRE	VITRD	V	S/V	S/V	$Z_{i+1} \leftarrow X_i + Y_i * Z_i$
Convert Double to Single	VCVDE		V	S/V		Z_i(Single) $\leftarrow X_i$(Double)
Convert Single to Double		VCVED	V	S/V		Z_i(Double) $\leftarrow X_i$(Single)

表1は28種の命令の一覧表を示したものである．演算語（オペランド）がスカラかベクトルかはOP拡張部で指定されるがこれは命令種に含まれていない．28の命令は16種の一般ベクトル命令，10種の複合演算ベクトル命令，2種の変換ベクトル命令からなる．複合演算ベクトル命令については3.3節で述べる．

3.3 複合演算ベクトル命令

2.3節で述べたように，SS形式のベクトル演算機能を採用したために，処理装置と記憶装置の間のデータ転送負荷が重くなる可能性があった．このため，表1に示すように，VIPE/D, VIPCE/D, VSMAE/D, VSMSE/D, VITRE/D, の10命令の複合演算ベクトル命令を付加した．基本演算ベクトル命令がベクトル要素間の1演算の集まりとして定義されているのに対し，これらの命令はいずれも2演算を含んでおり，VSMAE を例にとると，

（1） 複合命令を用いた1操作では，スカラ・オペランドXの読みだしを除くと，Z_i, Y_i の読みだし，Z_i の書き込みの3回の記憶装置参照が発生するのに対し，

（2） 二つの基本命令に分解した場合には，中間結果 $X*Y_i$ の書き込みと読みだしが加わり，3回が5回に増えることになる．

したがって，複合演算ベクトル命令の導入により，この種の演算では記憶装置の負荷が5回/1操作から3回/1操作に40%削減されることになり，きわめて大きな効果があるといえる．**表2**は，複合演算命令の効果を予測するために，各20題の軽科学計算と重科学計算を選び，全実行 FORTRAN ステートメントの何%が対応するベクトル命令に変換されるかを動的に調べたものである．一般計算センタで普通に流される軽科学計算では5.4%にすぎないが，実行時間の長い重科学計算では24.4%と，その効果の大きいことがわかる．表2には条件付き命令，その他の各欄があるが前者はDOループ内にIF文を含むような場合であり，後者は整数演算，等を含んでいる．さらに，高いベクトル化率を実現するためにはこれらに対応する命令が必要になるが汎用機の付加機構という制約を考えて M 180/200 H では採用しなかった．

表2 複合演算ベクトル命令の効果
Table 2 Effectiveness of compound arithmetic vector instruction.

ベクトル命令種別	ベクトル命令の適用率		備考
	軽科学計算	重科学計算	
基本演算命令	25.1%	30.6%	M180/200H の命令レパートリ
複合演算命令	5.4%	24.4%	
条件付き命令	16.9	10.1	
その他	24.9	20.7	
以上，最内側ループの合計	72.3	85.8	残りは一般にはベクトル化不能

このような複合ベクトル演算命令を導入した場合の一つの課題は，高度な自動ベクトルコンパイラが必要になることである．とくに，一般の FORTRAN プログラムから，VIPE/D 命令に対応する内積演算を検出するのはかなり困難である．これについては文献[16]を参照されたい．

3.4 割込み方式

2.4節でも述べたように汎用計算機でベクトル命令を実現するためには，各種の割込み要因に対してその応答性を保証し，また，後の対応を可能にするような十分な精度の割込み方式が必要になる．M 180/200 H では，各ベクトル命令を複数の操作単位 (unit of operation) に分割し操作単位境界での割込みを可能にすることにより，必要な応答性と解像度を実現した．より，具体的には VEAE, VIPE の各命令では，ある i に対する $Z_i \leftarrow X_i + Y_i$, $FPR \leftarrow FPR + X_i * Y_i$ が一つの操作単位になっておりM 200 H 程度の大型機では 1 μs 前後で完了する単位であるから，応答性を保証するに十分小さなものといえる．

このような操作単位で割込みが発生した場合には，**表3**に示すように再開に必要な情報が退避される．表に示したものは通常発生する割込みに対応したもののみであるが，たとえば，あるベクトル要素の読みだし処理中にページフォルト（対応するデータが主記憶に存在しない）を発生した場合には，表3に示すような取消し型の割込み（その操作単位の実行がなかったような形の割込み）となり，退避される PSW に現在実行中のベクトル命令のアドレスが格納され，ベクトル命令の R1 部で指定される開始要素番号には，ページフォルトを発生した要素番号が格納される．割込みにより通報を受けた制御プログラムでは，必要ページを

表3 割込み時の退避情報
Table 3 Status saving when an interrupt occurs.

割込みの種類	PSW の命令アドレス	(RI)の開始要素番号
取消(ページフォルト等)	CIA	CEC
完了(オーバフロー，入出力割込み等)	CIA+4	CEC+1

(注) PSW : program status word
CIA : current instruction address, 実行中のベクトル命令のアドレス
CEC : current element count, 割込みを発生したベクトル要素番号

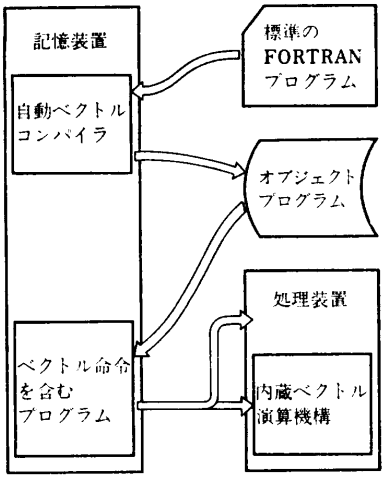

図 4 内蔵ベクトル演算機能の動作概念図
Fig. 4 Operational flow of integrated vector arithmetic facility.

2次記憶装置からページインした後，同じベクトル命令に制御を戻せば，同時に退避回復されたR1レジスタ内の開始要素番号の指定によって，先に取り消された操作単位の処理が再開されることになる．

3.5 自動ベクトルコンパイラと内蔵ベクトル演算機構

図4に示すように，M 180/200 H のベクトル演算機構は，一般の FORTRAN プログラムから自動ベクトルコンパイラの働きにより，ベクトル化可能部分が自動的に表1のベクトル命令に落とされる．展開されたベクトル命令は，処理中の内蔵ベクトル演算機構により実行される．自動ベクトルコンパイラについては文献[16]によるとして，この節では，汎用計算機に付加された内蔵ベクトル演算機構の特徴を200H を例とし述べることにする．

図5 に示すように，200H 処理装置の内部構成は，最近の多くの大型機に見られるように，記憶装置，記憶制御ユニット，命令制御ユニット，命令実行ユニットの4ユニットからなるが，内蔵ベクトル演算機構を実現するために，ベクトル演算制御ユニットが付加されている．

（1） ベクトル演算制御ユニットは，おもに演算部の制御を行い，また，内部にデータバッファを有し，オペランドの円滑な格納と供給を図っている．当ユニットは演算パイプラインを制御して高速処理を実現すると同時に操作単位での割り込みを保証するための緊急停止回路を内蔵しており，IAP を特徴づける重要な部分となっている．ベクトル演算機能を実現するために各ユニットに新たに付け加えられた機能には次のものがある．

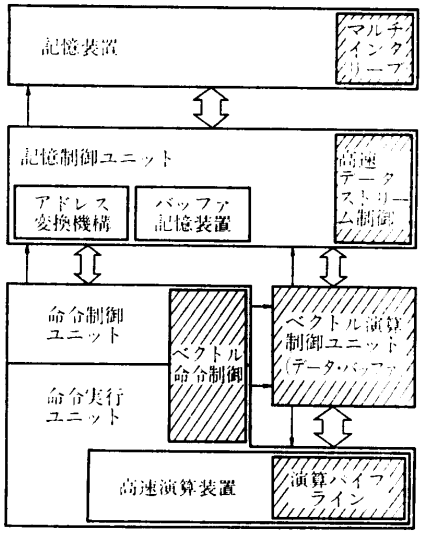

図 5 内蔵ベクトル演算機構を含む HITAC M-200 H の内部構造
Fig. 5 Machine organization of HITAC M-200 H with integrated vector arithmetic feature.

（2） 命令制御ユニット：命令実行ユニットと協力して，ベクトル命令の解読から始まり，初期アドレスの決定と第1ベクトル要素の読みだしを含む前処理，ベクトル演算実行中のアドレス制御，終了後の後処理等を行う．

（3） 命令実行ユニット：当ユニット内の高速演算装置を使ってベクトル演算を行う．演算の制御はベクトル演算制御ユニットから発せられる．高速演算装置は1サイクルごとに新しい演算が始められるよう，演算パイプライン構成にして，ベクトル演算に備えている

（4） 記憶制御ユニット：ベクトル演算に必要なオペランドを間断なく供給できる構成になっているが，とくに非連続領域のデータが毎サイクル読みだせる高速データストリーム制御を行っている．また，毎サイクルアドレス変換機構が使用できるように工夫を行っている．

（5） 記憶装置：通常の8ウェイ・インタリーブに対し，ベクトル演算機構を付加した場合には，16ウェイ・インタリーブが可能な構造とし，メモリスループットの強化に努めている．

表 4 ループミクスの定義
Table 4 Definition of the loop mix.

ループ名	重み (%)	FORTRAN	通常命令列	ベクトル・命令列
2項	20	DO $I=1, N$ $Z(I)=X(I)+Y(I)$	→LD AD STD —BXLE	VEA(Z, X, Y)
内積	35	DO $I=1, N$ $S=S+X(I)*Y(I)$	→LD MD ADR —BXLE	VIP(X, Y)
3項	15	DO $I=1, N$ $Z(I)=Z(I)$ $+X(I)*Y(I)$	→LD MD AD STD —BXLE	VEM(T, X, Y) VEA(Z, Z, T)
4項	30	DO $I=1, N$ $Z(I)=Z(I)$ $-(E*X(I)$ $+F*Y(I))$	→LD MDR LD MDR ADR LCDR AD STD —BXLE	VEM(T_1, E, X) VEM(T_2, F, Y) VEA(T_1, T_1, T_2) VES(Z, Z, T_1)

4. 性能の評価

4.1 ループミクスによる評価

内蔵ベクトル演算機能がおもにFORTRANのDOループ部のベクトル命令展開による高速化を目標としていることから，性能評価のための一つの指標としてループミクス (loop mix) を導入した．これは，汎用計算機で広く用いられている各種の命令ミクスにならったものである．ループミクスは**表4**に示すように，4種類のループの1ループ当りの平均処理時間を求め，これを図に示すような重みで平均し，1ループ当りの平均実行時間を求めるものである．ループの種類と重みは，初期に実行した各種の実プログラムの解析から経験的に設定したものである．この種の1次評価尺度は，(a)簡単に計算できること，(b)評価結果がハードウェアまたはソフトウェアの方式の優劣と容易に関係付け可能なこと，等が必須であり，この点簡明なループミクスの重要性は，きわめて高い．

表5は HITAC M-180, 200 H について，ループミクスを測定した結果の1例を示したものである．各ループのスカラ命令による実行時間とベクトル命令による実行時間を対比して示してある．ループミクスは一般には取り扱うベクトル長によって大きく異なってくるので，上半分にループ回数またはベクトル長64の場合，下半分にループ回数またはベクトル長512の場合を示してある．もちろん，ループミクスの基本となるループ当りの実行時間は，ループ回数が多くても1ループ当りに換算したものを用いている．

表 5 ループミクスの測定値
Table 5 Performance evaluation by the loop mix.

ループ名	重み (%)	M180			200H		
		μs/loop		相対比	μs/loop		相対比
		Scalar	Vector	S/V	Scalar	Vector	S/V
2項	20	1.54	0.72	2.1	0.44	0.17	2.6
内積	35	1.67	0.60	2.8	0.55	0.16	3.4
3項	15	2.24	1.27	1.8	0.71	0.17	4.2
4項	30	2.86	2.32	1.2	1.08	0.52	2.1
合計	100	2.08 μs	1.24 μs	1.7	0.71 μs	0.27 μs	2.6
2項	20	1.52	0.55	2.8	0.40	0.11	3.6
内積	35	1.65	0.41	4.0	0.54	0.11	4.9
3項	15	2.21	0.96	2.3	0.66	0.11	6.0
4項	30	2.84	1.85	1.5	1.03	0.40	2.6
合計	100	2.07 μs	0.95 μs	2.2	0.68 μs	0.20 μs	3.4

注：上半分ループ長 64 の場合，下半分ループ長 512 の場合

科学技術計算用計算機の評価によく用いられるMFLOPS (million floating operations per second) に換算するためには，2項，内積，3項，4項の各ループ当りの浮動小数点演算数がそれぞれ 1, 2, 3, 4であることより，ループミクスの平均では2.4浮動小数点演算となるので，MFLOPS=2.4／(ループミクス値) となる．たとえば，ループ回数512の場合のM-180, 200 H での 2.07 μs (スカラ命令)，0.95 μs (ベクトル命令)，0.68 μs (スカラ命令)，0.20 μs (ベクトル命令) という実測値は，それぞれ1.2 MFLOPS, 2.5 MFLOPS, 3.5 MFLOPS, 12 MFLOPS に対応し，ベクトル命令の採用により2.2倍(M180)，3.4倍(200 H) の性能向上が実現できたことを示す．

一方，汎用計算機に用いられる MIPS (million instructions per second) という視点からは，2項，内積，3項，4項の各ループが，それぞれ4, 4, 5, 9のスカラ命令を含むことから，ループミクスの平均では5.65命令となり，MIPS=5.65/(ループミクス値) となる．したがって，先程の 2.07 μs, 0.95 μs, 0.68 μs, 0.20 μs という値は，2.7 MIPS, 5.9 MIPS, 8.3 MIPS, 28 MIPS に対応することになる．以上に示してきたように，ループミクス評価で1.7〜3.4倍という性能改善値は，汎用計算機の付加機構が実現したものとしてはきわめて大きな値といえる．

表 6 線形計算プログラムによる評価
Table 6 Performance evaluation by linear arithmetic programs.

プログラム	200H		
	Scalar (秒)	Vector (秒)	相対比 S/V
帯コレスキ分解 帯幅 81 元数 480	1.01	0.57	1.8
連立一次方程式(一次元コレスキ) 300×300	2.52	0.71	3.6
固有値解(HH バイセクション) 300×300	13.05	4.51	2.9
固有値解(HH QR 逆反復) 300×300	9.81	4.66	2.1

表 7 ベンチマーク・ジョブによる評価
Table 7 Performance analysis by some benchmark jobs.

BJ群の名称	群当りのBJ数	1 Job 当りの平均実行時間	BJ群の実行時間 Scalar	BJ群の実行時間 Vector	改善率 (S/V)	Vectorization Ratio
I	8	0.96秒	7.70秒	7.51秒	1.03倍	13.6%
II	8	2.89	23.09	17.99	1.28	22.5
III	8	6.02	48.14	39.27	1.23	18.1
IV	8	14.62	116.92	89.19	1.31	23.2
V	8	46.54	372.35	251.86	1.48	31.9
合計	40	14.21秒	568.20秒	405.82秒	1.40倍	28.3%

4.2 線形計算プログラムによる評価

表 6 は線形計算プログラムによる HITAC M-200 H内蔵ベクトル演算機能の性能評価結果の1例を示したものである．このような具体的プログラムにおいても，200H で1.8～3.6倍の改善となっている．

4.3 システム性能の評価

ループミクスや線形計算プログラムのような科学技術計算の典型的プログラムでは，大きな効果のあることがわかったが，それでは多種多様なジョブを処理する一般の計算センタとしては，内蔵ベクトル演算機能からどの程度の性能改善を期待できるのだろうか．**表 7** はおもに科学技術計算を業務とするセンタのジョブから40本のプログラムを選んで，HITAC M-200H で実測したものである．表では，スカラ命令での実行時間の小さいものから，8ジョブずつに区切って，五つのジョブ群に別けて示してある．たとえば，第1群は最も実行時間の小さい八つのジョブからなり，1ジョブ当りの平均実行時間は0.96秒である．この8本のジョブのスカラ命令での実行時間は7.70秒であり，ベクトル命令を導入すると7.51秒に短縮され，1.03倍に，すなわち3％の性能向上が実現することを示している．表には，プログラムの何％がベクトルコードに落とされたかを示すベクトル化率も示してあるが，第1群の場合のベクトル化率は13.6％であることを示している．ベクトル化率は，スカラ命令での実行時の機械語の全行数を求め，そのうちの何％がベクトル命令に置換されたかを逐一調査して求めたものである．この種のデータは選択されるジョブに大きく依存するが，このジョブが無作為に選ばれており，数も40とかなり多いことを考えると，ある程度平均的な値を示していると考えられる．この表からは次のことがわかる．

(1) ベクトル化率は，実行時間の短いジョブでは概して低下するがそれほど大きな差はない．40本のプログラム全体では平均28.3％のベクトル化率を示す．

(2) ベクトル化による性能改善度は，第1群のジョブでは非常に小さい．課金時間0.96秒といったようなジョブでは，FORTRAN の実行部分以外の基本的処理にかなり食われるだろうから，ある意味では当然といえる．

全体では1.40倍，すなわち40％の性能向上となっており，一つの付加機能の実現する値としてはかなり大きい値といえる．

5. むすび

汎用計算機向きのベクトル演算機能を開発し，HITAC M-180/200 H で実用化した．ベクトル命令の導入により，ループミクスで1.7～3.4倍，線形計算プログラムで1.6～4.2倍の性能向上を実現することができた．

内蔵ベクトル演算機能は徐々に一般化する傾向にあるが，長期的には次の方向で今後の研究開発を進めていく必要がある．

(1) 第1には内蔵ベクトル演算機能および性能の拡張である．半導体技術の進展に伴って，妥当な費用の範囲でかなりの複雑な論理回路が汎用計算機に組み込めるようになっているので，現在の科学技術専用計算機(スーパーコンピュータ)程度の機能と性能は，近い将来汎用計算機において提供されるようになるであろう．その変化は，当初，科学技術計算用の付加機能であった浮動小数点演算機能が，汎用計算機用の標準機能となったのと似た経緯をたどることになろう．この場合，スーパーコンピュータは多数の処理装置からなる並列処理などを含むより高度な分野に発展していくものと思われる．

(2) 第2は，ユーザの作成するプログラムとより整合性のとれたものにすることである．現在でも，ベクトル演算機能の効果のほとんどがあがらないプログラムは多数存在するし，汎用計算センタでの実績も十分とはいいがたい．ベクトル演算機能のもつ潜在力をさらに発揮させるには次の二つの方向の努力が必要である．すなわち，

(a) ハードウェアのもつベクトル演算機能の充実と，より高度のコンパイラ技術の開発により，より多くのプログラムがベクトル演算の高速性の恩恵に浴せるようにすること，

(b) ベクトル化に適したソース・プログラムをユーザが作成するように先導すること．

(b)項は，なんらかの高級言語の先導なしには一般ユーザを引き込むことは困難であろう．過去において，仮想記憶管理向きのプログラミングといったことがほとんど成功しなかったことを十分反省する必要がある．すなわち，歴史の流れはより制約なしにユーザが計算機を利用できること，の1点を目指して進んでいるというべきであろう．

本研究は着手以来すでに8年の歳月が流れており，非常に多くの方々のお世話になった．とりわけ実用化の機会を与えて下さった日立製作所小田原工場 曽我副工場長，同神奈川工場 浦城部長，井上部長，小高主任技師，泉主任技師，同ソフトウェア工場 高須部長，また，ベクトル演算機能の発展のために終始ご指導いただいた同神奈川工場 中沢工場長，古厩副部長，森田主任技師，同小田原工場 堤主幹技師長，同中央研究所 村田技師長，また，直接研究の推進にご協力いただいた同神奈川工場 河辺技師に感謝の意を表します．

参 考 文 献

1) 小高俊彦，他：超高速演算の動向，情報処理，Vol. 21, No. 9, pp. 927-937 (1980).
2) IBM：IBM System/370 Principles of Operation, IBM マニュアル，GA22-7000.
3) 中沢喜三郎，他：HITAC M-200H 汎用高速処理装置，日立評論，Vol. 61, No. 12, pp. 7-12 (1979).
4) Horikoshi, H. et al.: An Example of LSI-Oriented Logic Implementation in a Large-Scale Computer, the HITAC M-200H, Compcon, pp. 62-65 (1980).
5) 小高俊彦，他：HITAC M-180 内蔵アレイプロセッサ，日立評論，Vol. 60, No. 6, pp. 53-58 (1978).
6) 小高俊彦，他：HITAC M-180 内蔵アレイプロセッサ，電子通信学会専門委員会，EC 78-17, pp. 23-32 (1978).
7) 河辺 峻，他：HITAC M-200 H 内蔵アレイプロセッサ (IAP)，電子通信学会電算機研究会，EC 80-79, pp. 67-75 (1980).
8) 日立製作所：HITAC M-180/200H, 内蔵アレイプロセッサ HITAC マニュアル，8080-2-041.
9) Takanuki, R. et al.: Some Compiling Algorithms for an Array Processor, 3rd USA-JAPAN Computer Conference, pp. 273-279 (1978).
10) Umetani, Y. et al.: An Analysis of Applicability of the Vector Operations to Scientific Programs and the Determination of an Effective Instruction Repertoire, 3rd USA-JAPAN Computer Conference, pp. 331-335 (1978).
11) 梅谷征雄，他：ベクトルマシンとベクトル化コンパイラの方式，電子通信学会専門委員会，EC 80-15, pp. 49-56 (1980).
12) 松尾 洋：VOS 3 APL の開発，情報処理学会第22回 (昭和56年前期) 全国大会，pp. 175-176 (1981).
13) 唐木幸比古：M-200 H IAP の威力，東京大学大型計算センターニュース，Vol. 12, No. 8, pp. 78-83 (1980).
14) 唐木幸比古：FORTRAN 素子間の IAP 相関効果，東京大学大型計算センターニュース，Vol. 13, No. 3, pp. 39-48 (1981).
15) 唐木幸比古，他：FORTRAN プログラムから見たアレイプロセッサの性能評価，情報処理学会アーキテクチャ研究会，41-4, pp. 1-8 (1981).
16) 梅谷征雄，他：内蔵ベクトル演算機構のための自動ベクトルコンパイラ方式，情報処理学会論文誌，Vol. 24, No. 2, pp. 238-248(1983).
17) Cray Research Inc.: CRAY-1 Computer System Hardware Manual, Pub, No. 2240004 (1979).

(昭和57年7月13日受付)
(昭和57年9月 6 日採録)