U.D.C. 681.322.012-181.2-185.4

コンピュータアーキテクチャ技術の動向とスーパーコンピュータ

Computer Architecture and Supercomputer

堀越　彌*　*Hisashi Horikoshi*
浦城恒雄**　*Tsuneo Uraki*

近年の半導体技術の急激な進歩は，大規模かつ複雑なコンピュータの構築を可能にしつつある。スーパーコンピュータはその代表であり，先端技術を駆使して毎秒1億から10億に近い演算が可能である。本論文では現在の汎用コンピュータが抱えている一つの隘路，処理装置と記憶装置の間の命令語及びデータの多量な交換という問題が，スーパーコンピュータではアーキテクチャ技術の点からどのように解決され，その高速性の実現を可能にしているかを明らかにする。また近い将来に，その実現が予想されるアーキテクチャの一つ，多数の処理装置の複合によりスーパーコンピュータを構成する多重プロセッサ方式でも，より軽減された形ではあるが同様の問題が生じ，この解決が今後のアーキテクチャ技術の大きな課題であることも示す。

1　緒　　言

マイクロコンピュータから超大形コンピュータまで，各種のコンピュータが社会のあらゆる分野で重要な役割を演ずるようになってきている。一方コンピュータを構成する基幹ハードウェア技術，すなわち半導体技術の急激な進歩は素子コストの飛躍的低減を可能にし，従来不可能であったような大規模かつ複雑な装置を実現しつつある。その一例がスーパーコンピュータであり，数百MFLOPS※1)に及ぶその性能が科学技術計算分野の計算可能範囲を著しく拡大し，研究開発の速度をいっそう加速しつつある[1),2)]。

この論文では，汎用コンピュータからスーパーコンピュータに至るアーキテクチャ技術の背景とその展望について述べる。「コンピュータアーキテクチャ」という用語の厳密な意味は，「プログラムから見えるコンピュータの機能」であり，コンピュータ内部の実現方法とは独立した概念であるが，広義にはコンピュータの構成方法を表わす言葉にも用いられている。ここでは，コンピュータ，特に処理装置と記憶装置から成る系の機能と構成という視点から，汎用コンピュータ及びスーパーコンピュータのアーキテクチャ技術を展望する。

2　汎用コンピュータとベクトル命令方式

現在の汎用コンピュータアーキテクチャの基本的特徴の一つは，二つの数値間の演算を指示する命令(これをスカラー命令という。)を次々に実行して処理を進めていく点にある。一方，スーパーコンピュータは二組みの数値列間の演算を指示する命令(これをベクトル命令という。)を基本とする。本章ではスカラー命令を基本とする汎用コンピュータの性能決定要因を明らかにした後，科学技術計算分野では新しいアーキテクチャ，ベクトル命令方式の導入によって著しい性能向上が可能になることを述べる。

2.1　汎用コンピュータのアーキテクチャ

現在の汎用コンピュータは「フォンノイマン流のアーキテクチャ」であると言われることがある。これは現在のストアードプログラム形の計算機方式の創始者であるフォンノイマンの名を冠したものである。**図1**を用いて現在の汎用コンピュータの基本的動作原理を考察する。

この方式では，計算機の動作を指定する命令は記憶装置に格納される。命令 i の実行は，処理装置による命令語の読み出しとこれに続く命令解読によってその実行が開始される※2)。通常の命令は二つのオペランド(演算数)を指定し，一方のデータは記憶装置，他方は処理装置内の高速の内部レジスタから読み出される。二つのオペランドは演算装置で処理され内部レジスタへ戻される。これが代表的な命令の処理形態であり，このような命令の実行時間は二つの因子から決まる。第1は**図1**中の①，②，③などで示した基本的な処理単位の短縮である。これをマシンサイクルと呼んでいるが，主に素子や実装技術から決まる。第2は一つの命令に要する手順数，同図では①～⑤の数を減らすことである。これを命令実行サイクル数と呼び，主に方式論理技術から決まる。マシンサイクルを T_{MC}，命令実行サイクル数を N_{IE}，その平均を $\widetilde{N}_{IE}$ で表わすと，命令の平均実行時間 $\widetilde{T}_I$ は，$\widetilde{T}_I = \widetilde{N}_{IE} \times T_{MC}$ となる。現在の汎用超大形コンピュータでは，先端的半導体技術によって数十ナノ秒の T_{MC} が実現されており，N_{IE} も高度なパイ

※1)　MFLOPSとは，Million Floating Operations Per Secondの略で，毎秒100万回の浮動小数点演算を行なうスーパーコンピュータの性能を1MFLOPSという。1,000MFLOPSのことを1GFLOPSともいう。汎用超大形コンピュータでは数十MFLOPS，スーパーコンピュータでは数百MFLOPSの最大性能をもっているのが普通である。実効性能はスーパーコンピュータで1桁近く低下する。

※2)　この論文では，命令を含む情報単位を命令語，オペランド(演算数)を含む情報単位をデータと称するが，命令語，データを合わせて一般的にデータと呼ぶこともある。

* 日立製作所中央研究所 工学博士　** 日立製作所神奈川工場

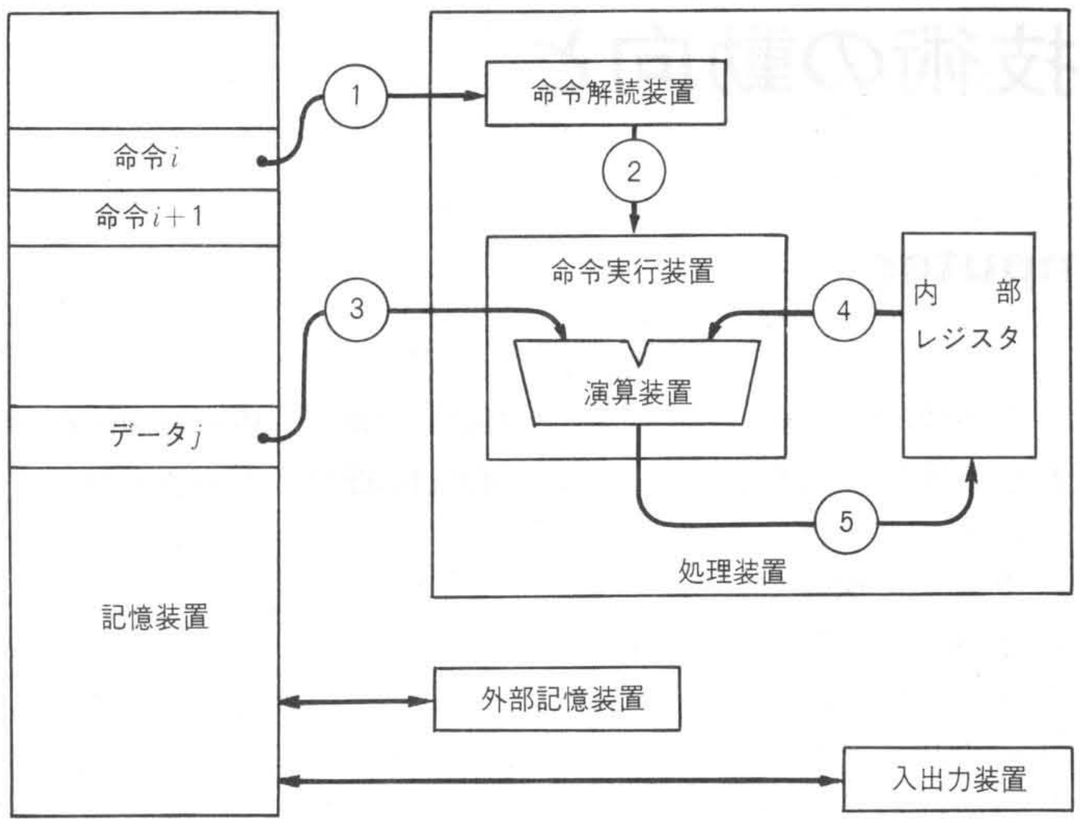


図1 汎用コンピュータの基本動作原理 処理装置と記憶装置との間で，命令語及びデータの交換が毎回発生する。

プライン制御技術によってほぼ1に近い値が実現されている。したがって，両者とも飛躍的な改善は困難になっており，ここに新しいアーキテクチャに対する期待が高まる背景がある。特に，科学技術計算分野は演算内容の規則性も高く，新しいアーキテクチャを実現する可能性も大きいため，従来既に数々の試みがなされており，最近はスーパーコンピュータという大きな独立分野に発展しつつあると言える。

それではスーパーコンピュータでは，**図1**に示すような汎用コンピュータ方式をどのように改善しようとしているのであろうか。一般にフォンノイマン流のアーキテクチャの大きな欠点は，記憶装置と処理装置の間のデータ交換が多いことだとされており，この部分を「フォンノイマンの隘路」と呼ぶことがある。**図1**の命令の場合①と③の2回の記憶装置参照が発生している。次節では科学技術計算の具体的例題を用いこの点を定量的に考察する。

2.2 汎用コンピュータによる科学技術計算

科学技術計算にはいろいろな種類があるが，**図2**に示す熱拡散問題を例題として，現在の汎用コンピュータのアーキテクチャが抱えている問題点を考察する。この例題は横1 m，縦1.2mの管で，左右が断熱壁，上下が200℃の境界条件で，内部の初期温度100℃から温度分布$T(x, y)$の時間変化を求めるものである。この現象は同図に示す熱拡散方程式によって

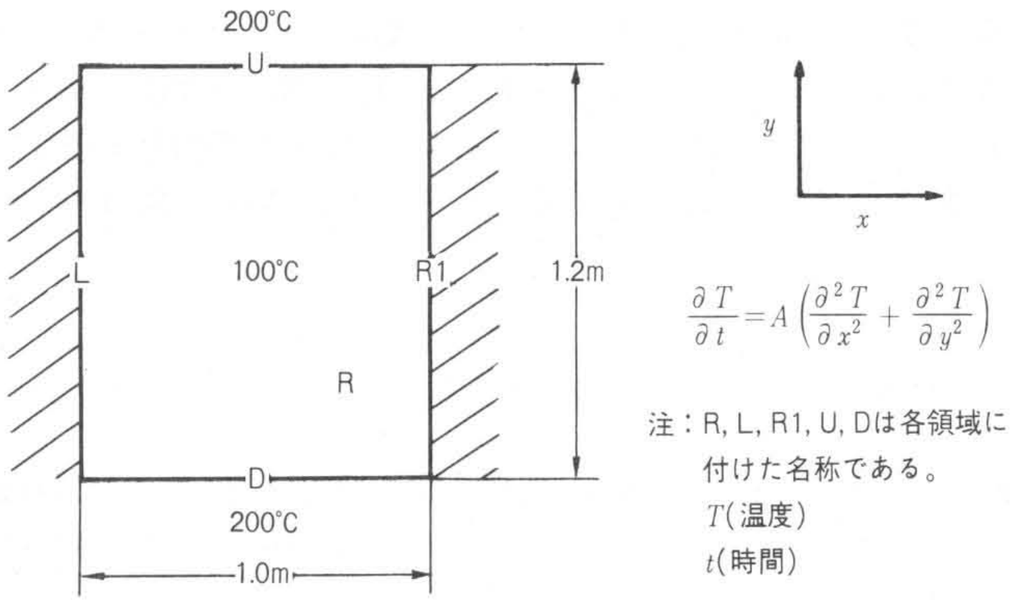


図2 科学技術分野の数値計算の一例(熱拡散問題) 科学技術計算の典型的な問題の一つであり，スーパーコンピュータが有効に働く例でもある。スーパーコンピュータのアーキテクチャを考察する際の例題として以下で用いる。

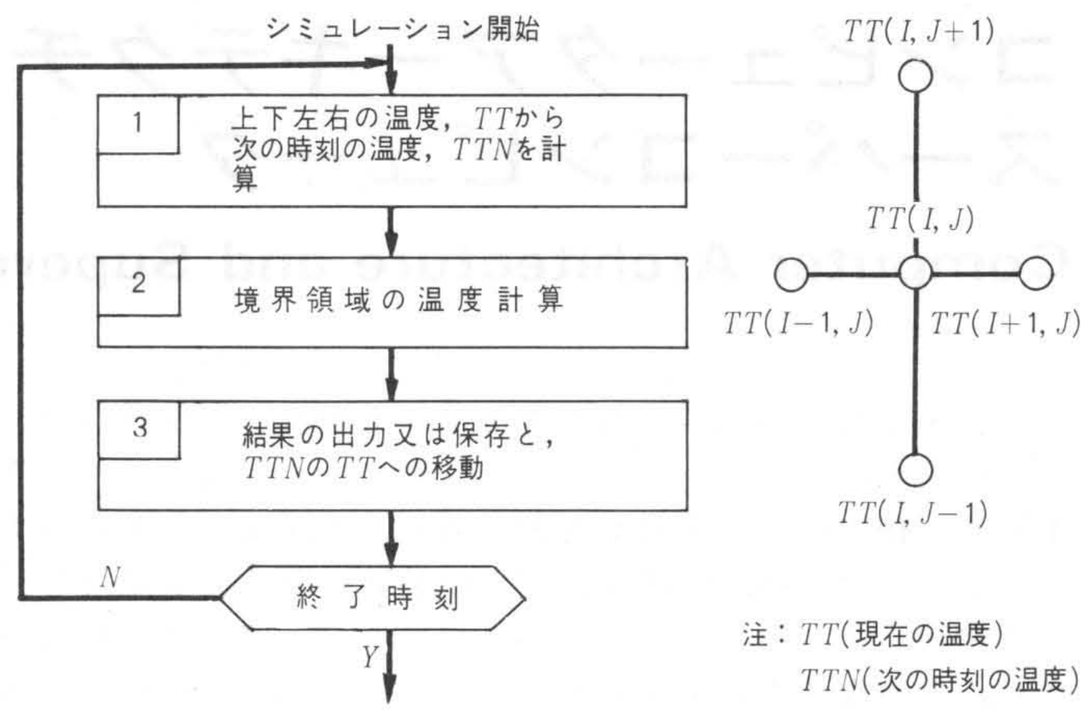


図3 熱拡散シミュレーションに必要な計算の処理の流れ 全格子点に対する計算を，時間刻みごとに行なう。

$$TTN(I,J)=TT(I,J)+DLT*A*(((TT(I+1,J)-2*TT(I,J)+TT(I-1,J))/DLX**2)+(TT(I,J+1)-2*TT(I,J)+TT(I,J-1))/DLY**2))$$

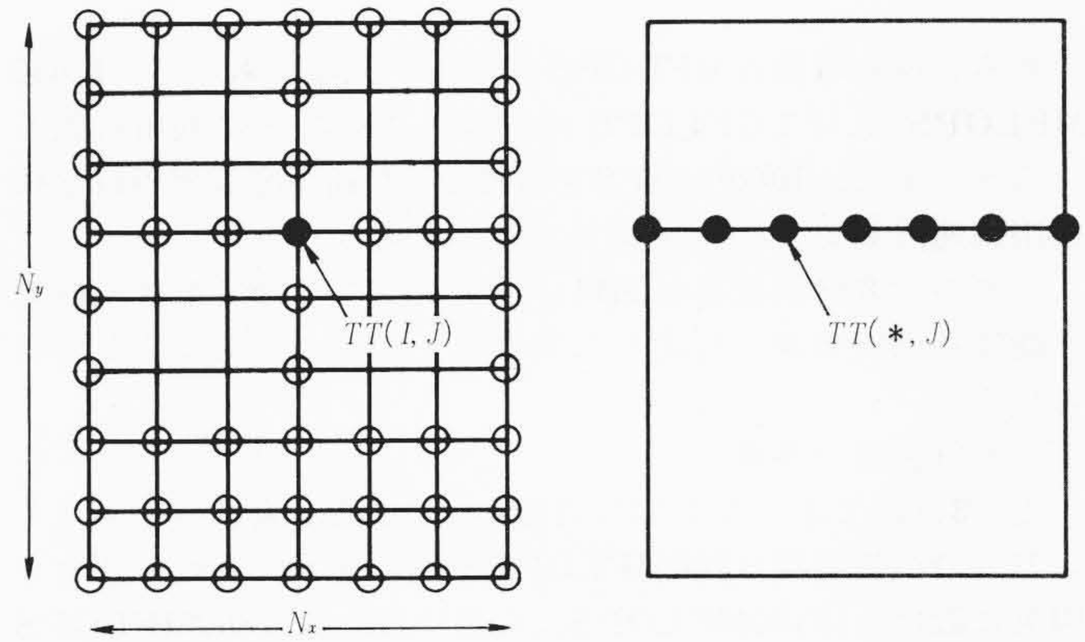
注：$TT(I, J)$は格子(I, J)の現在の温度を表わす。
$TTN(I, J)$は，格子(I, J)の次の時刻の温度を表わす。
DLTは，時間刻みを表わす。
DLXは，X軸の刻みを表わす。
DLYは，Y軸の刻みを表わす。
Aは，熱拡散定数を表わす。
見やすさのため，FORTRANの文法から外れた表記をしている。

図4 熱拡散シミュレーションで次の時間の温度を求めるプログラムの具体例 この問題の場合，コンピュータの大部分の処理時間はこの式の計算に費やされる。

表わされる。この方程式を解く数値計算の核部分を**図3**に示す。すなわち，管の断面を$N_x \times N_y$の格子に分割し，ある時刻での(I, J)番目の格子点の温度を$TT(I, J)$とし，次の時刻の温度$TTN(I, J)$を上下左右のTTから各格子点ごとに計算する。境界領域の格子点の計算法は内部と一般には異なるので処理2で行なうが，全格子点での計算の必要な処理1に比べてはるかに頻度が少なく，計算量としては無視できる。処理3では計算結果$TTN(I, J)$の出力又は後の出力用に別領域への移動保存が行なわれる。次の時刻の計算に先立つ$TTN(I, J)$の$TT(I, J)$への移動もここで行なわれる。

この計算の中心は処理1であり，この部分のFORTRANプログラムは**図4**のようになる。これを現在のFORTRANコンパイラを用いてスカラー命令に展開すると，1格子点の計算は約20命令の実行に対応する。したがって，時間刻み1回分の計算を行なうには$20 \cdot N_x \cdot N_y$のスカラー命令を実行する必要がある。$N_x = N_y \simeq 100$とすると2×10^5の命令実行になる。

このような分析からまずわいてくる疑問は，決まった手順の繰り返しになぜ2×10^5回もの命令を記憶装置から読み出し，かつその命令の解読を反復しなければならないかという点である。この反省からベクトル命令の概念が生まれる。ベクトル命令では数値列間の演算は一つの命令で済ますことができるので，**図5**に示すようにあるJに対応するすべての温度，$NTT(*, J)$の計算を20のベクトル命令で処理できる。したがって，全体では$20 \cdot N_y$の命令を実行すればよく，$N_x \simeq 100$



(a) スカラー命令実行数 $20\times N_x\times N_y$　　(b) ベクトル命令実行数 $20\times N_y$

図5　スカラー命令とベクトル命令の動作概念図　ベクトル命令の導入によって，同一の計算に必要な実行命令数が二桁前後減少する。

とすると二桁分だけ命令の読み出しと解読が軽減され，その分だけ処理装置と記憶装置のデータ交換が緩和される。この一例が次節の内蔵ベクトル演算方式のアーキテクチャである。

2.3　内蔵ベクトル演算方式

内蔵ベクトル演算方式とは科学技術計算能力を高めるために，日立製作所がHITAC M-180, 200H, 280H上にIAP(Integrated Array Processor)機能として実用化したものである[3]。**図6**に示すようなベクトル命令を28種備えることによって10 MFLOPS台の実効性能を汎用コンピュータで初めて実現した。また，**図7**に示す自動ベクトルコンパイラの働きによって，利用者は既存のFORTRANプログラムを用いることができる。この方式の一つの特徴は**図6**に示すように，すべてのベクトル命令のオペランドが記憶装置上に存在し，内部レジスタを用いていない点である。この結果，汎用コンピュータ並みのきめ細かい割込制御が可能になり既存のオペレーティングシステムとの共存を実現し，ベクトル命令方式の用途を拡大することができた。反面，記憶装置上のオペランドの参照が増大し，熱拡散問題の例では1時間刻みの計算当たり従来$10\cdot N_x\cdot N_y$回であった記憶装置参照が$30\cdot N_x\cdot N_y$と増大した。この値は命令語の読出し回数の$20\cdot N_x\cdot N_y$から$20\cdot N_y$

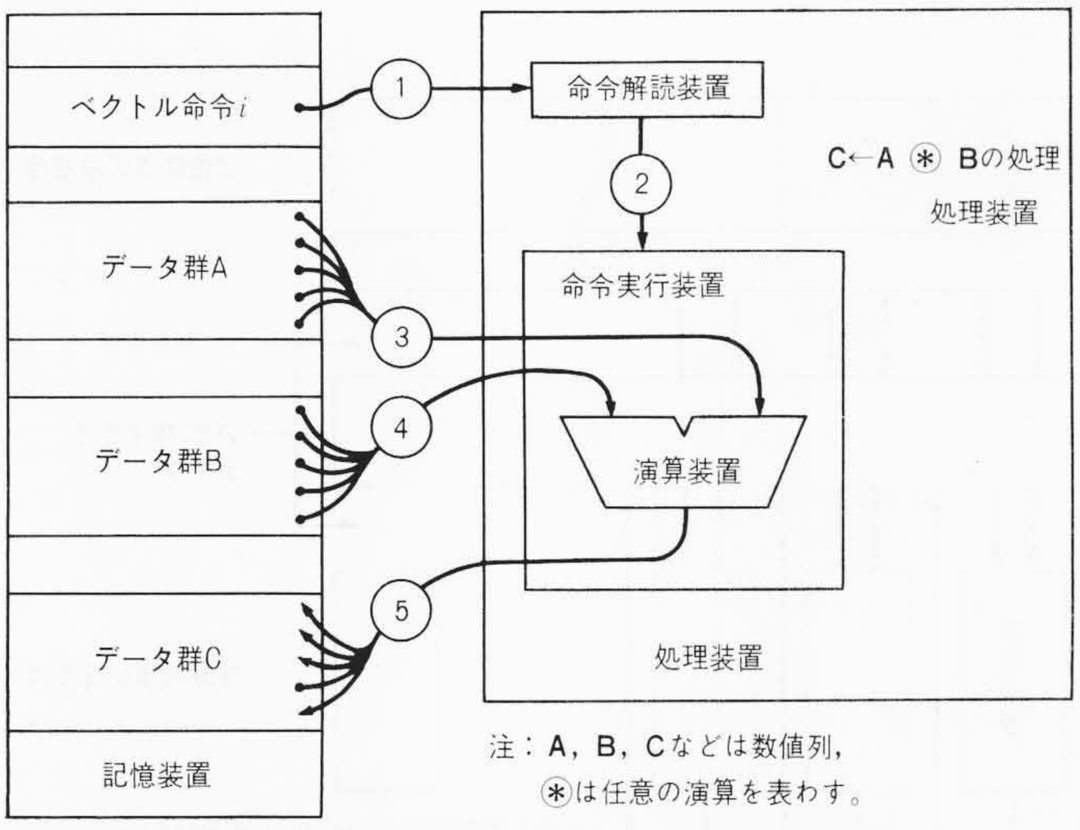


図6　内蔵ベクトル演算方式用のベクトル命令の概念図　汎用コンピュータにベクトル命令を付加する内蔵ベクトル演算方式は，日立製作所の大形汎用コンピュータHITAC M-180, 200H, 280Hに用いられ効果を挙げている。

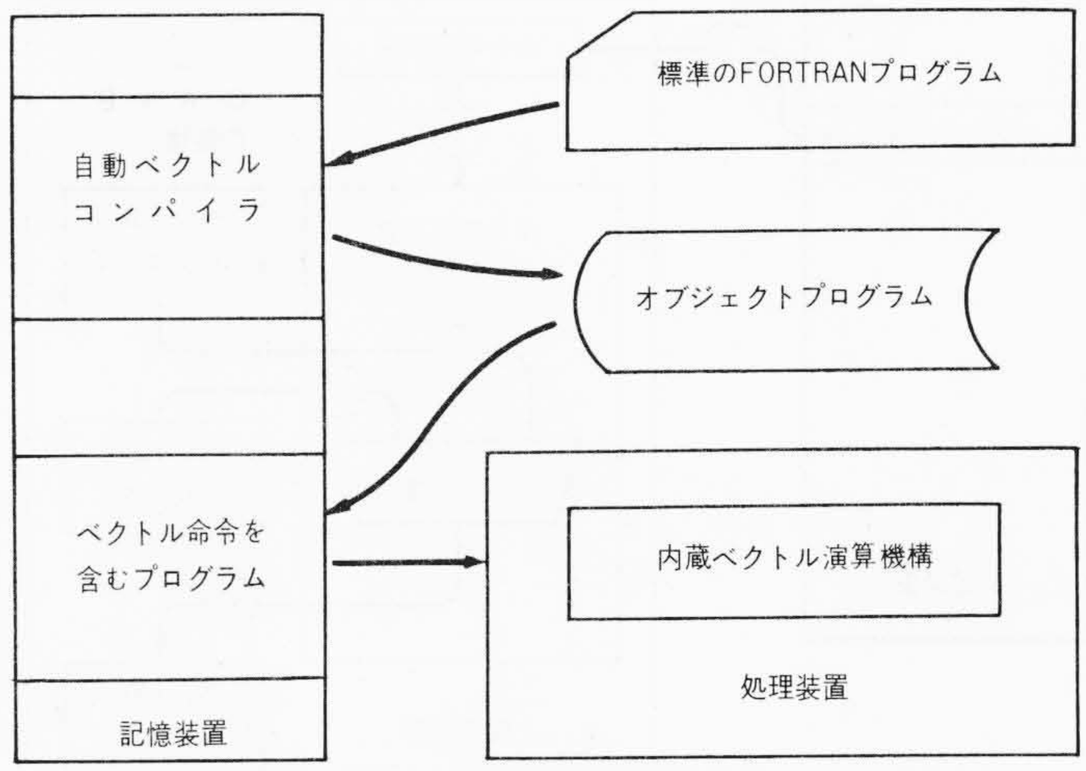


図7　内蔵ベクトル演算方式の使い方　内蔵ベクトル演算方式の利用者は，従来と全く同じのFORTRAN言語を用いることができる。ベクトル命令への展開は，自動ベクトルコンパイラが行なう。

への減少を相殺する値のものであるが，全体では大きな性能向上が得られている。その理由は命令解読回数が$20\cdot N_x\cdot N_y$から$20\cdot N_y$に減少したこと，及びベクトルデータの構造と規則性をあらかじめ処理装置に伝えることによって高度のパイプライン処理が可能になったことにある。一方，フォンノイマンの隘路を緩和するには，次章で述べるベクトルレジスタを導入して記憶装置参照回数を軽減する必要がある。しかし，この方式では処理装置内部に多量のプログラム固有のレジスタを保持するため，きめ細かい割込制御は不可能である。ここにスーパーコンピュータという専用機の登場する背景がある。

3　スーパーコンピュータのアーキテクチャ技術

本章では日立製作所が最近開発したHITAC S-810に代表されるスーパーコンピュータのアーキテクチャ技術の特徴について述べた後，近い将来の発展の可能性についても触れる[2]。

3.1　アーキテクチャ技術の特徴

現在のスーパーコンピュータの高速性は，次の三つの特徴的論理方式から実現されていると考えられる。

(1)　ベクトルレジスタを使ったベクトル命令方式

前章で述べたように，ベクトル命令の導入によって1演算当たりの命令数が減り，命令の読み出し及び解読に要する処理装置，記憶装置の負荷が軽くなる。現在のスーパーコンピュータでは，**図8**に示すように更に多量のベクトルレジスタを処理装置内部に設け，記憶装置とのデータ交換量を内蔵ベクトル演算方式よりも大幅に低減している。熱拡散プログラムの例では，時間刻み当たり命令語とデータを含め汎用コンピュータでは$30\cdot N_x\cdot N_y$であったものが，スーパーコンピュータでは$10\cdot N_x\cdot N_y+20\cdot N_y$程度に減少する。現在のスーパーコンピュータはすべてこの方式の土台の上に構築されている。

(2)　演算パイプライン方式

スーパーコンピュータでは対象とする科学技術計算分野の要請から，数値表現範囲の広い浮動小数点形式のデータを扱う。浮動小数点演算用演算装置の実現方法は一般に複雑で，1演算の実行に複数のマシンサイクルを必要とする。仮に1マシンサイクルを10ns，演算の実行に必要なサイクル数を10とすると，この演算器では10MFLOPSの性能しか実現できない。この演算器を**図9**に示すように10ns単位のステージに分

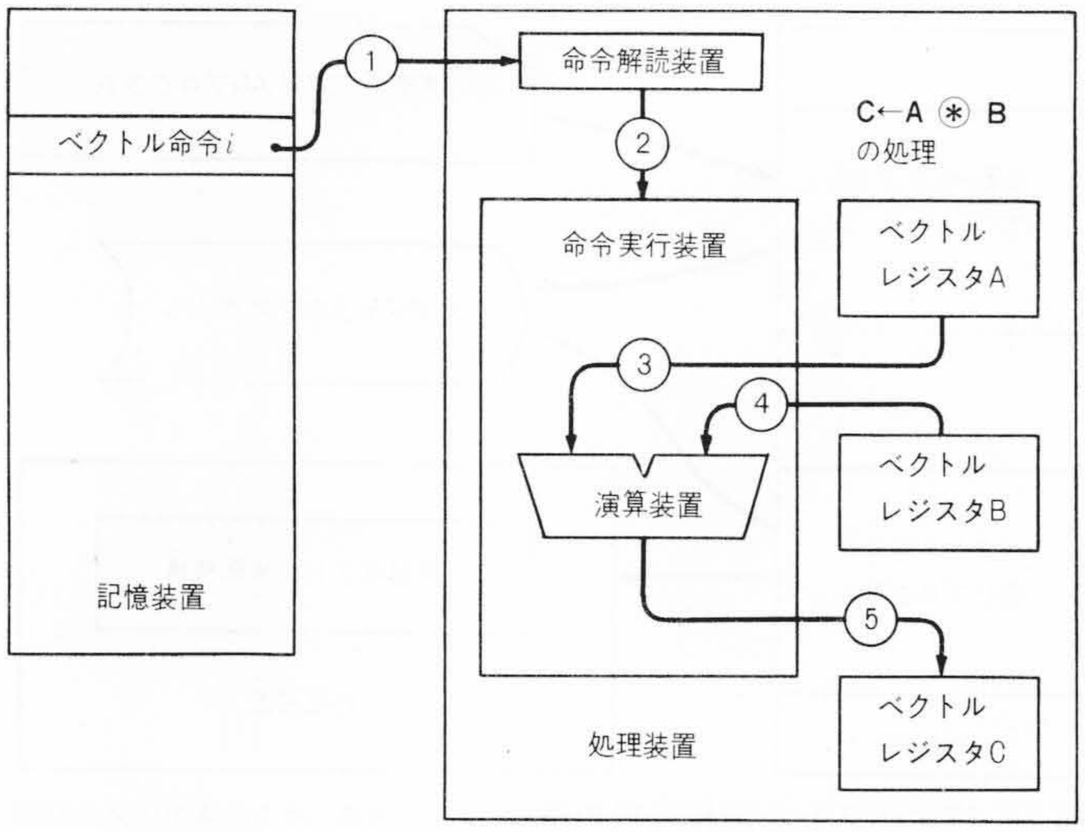


図8 ベクトル命令方式とベクトルレジスタ方式 現在のスーパーコンピュータは，すべてこの方式を採用している。ベクトルレジスタへのロード，ベクトルレジスタからのストアには別の命令を用いる。

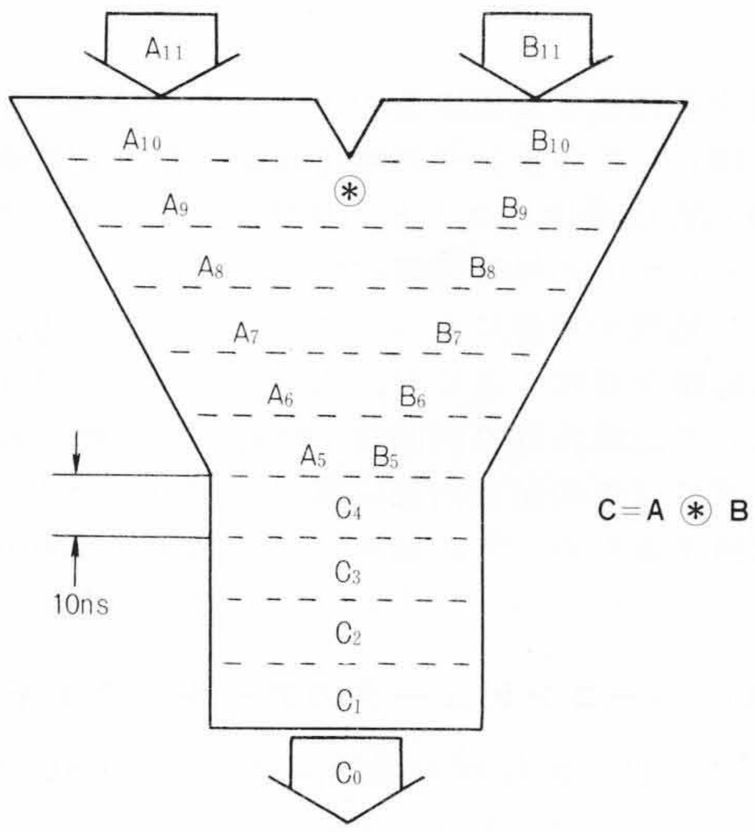


図9 演算パイプライン方式の概念図 演算装置の内部を多段に構成し，同時に処理できる演算の数を増し，演算装置全体のスループットを向上させる。

割し，先行演算の完了する前に次々と後続演算を開始させるのが演算パイプライン方式であり，この例では演算性能を100MFLOPSまで向上することができる。代案として演算器を並列に置く方法も考えられるが，論理素子の利用効率の点から演算パイプライン方式を採用するのが普通である。

(3) 命令チェイニング方式

後続の複数のベクトル命令を同時に実行するのが命令チェイニング方式である。パイプライン方式を採用した汎用計算機でも同時に複数命令の処理が進行するが，後続命令の処理は通常準備段階を完了すると先行命令が完了するまで保留される。これに対し命令チェイニング方式では，実行も並行して進む。したがって，**図10**に示すように先行命令の演算結果を後続命令の演算に接続する機構が必要になる。この方式のもう一つの重要な効果は，複数演算器を自然な形で同時に働かせることができることである。すなわち，4命令のチェイニングによって100MFLOPSの演算器を四つ動かせば400MFLOPSのスーパーコンピュータが実現できる。

3.2 最大性能を決める因子

スーパーコンピュータの一つの重要な評価項目は最大性能である。現在は数百MFLOPSの性能が普通であるが，1,000MFLOPS(又は1GFLOPS)の性能も近い将来に利用可能となろう。また，10GFLOPSを目標とした通商産業省の科学技術用高速計算システムプロジェクトもある。スーパーコンピュータの主要構成要素は**図11**に示す三つの部分であり，最大性能は各構成要素に対応した次の因子によって決まると考えられる。

(1) 演算装置の多重度

先の**3.1**節で述べたように，10ns刻みで動く演算パイプライン方式の演算装置は100MFLOPSの能力をもつため，これを四重に設ければ400MFLOPS，八重に設ければ800MFLOPSとなる。スーパーコンピュータに用いられる演算装置は一般に最も複雑かつ高度な論理方式を採用しているため，一つだけでも大形計算機並みの規模になり，現在の技術で32，64といった演算装置を組み込むのは容易でない。しかし，VLSI，ULSI技術の進展によって論理規模面からの制約は徐々に取り除かれていくことになろう。

(2) 記憶装置の多重度

演算には命令語とデータが必要になる。現在のスーパーコンピュータでは先の**3.1**節で述べたように，記憶装置とのデータ交換は汎用コンピュータよりも軽減されている。熱拡散問題を例にとると，平均して1演算に1回の記憶装置参照が発生する。しかし極めて高速なスーパーコンピュータではこれ

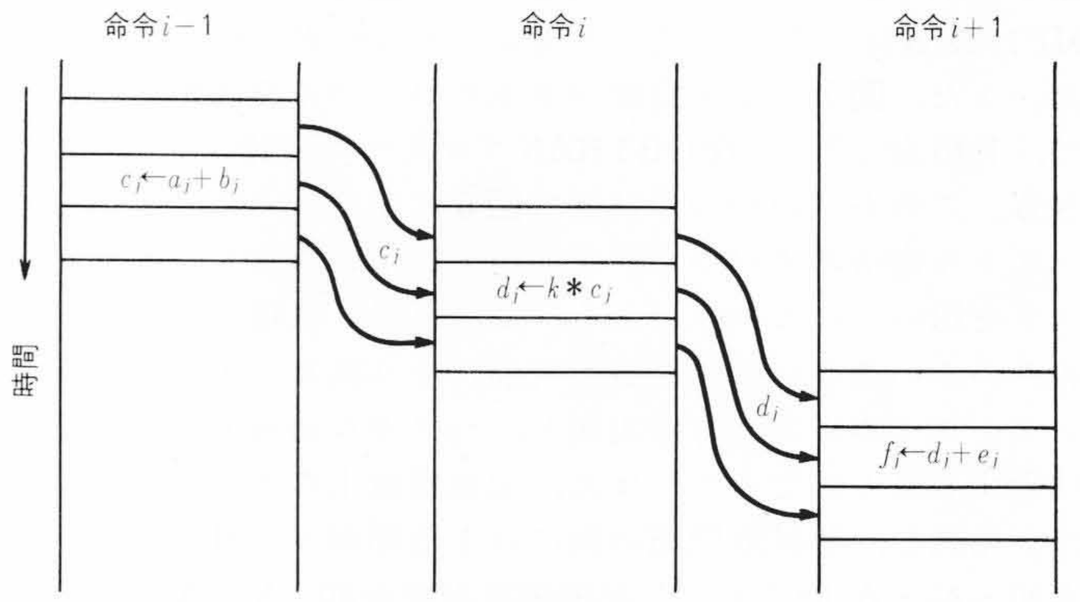


図10 命令チェイニング方式の概念図 命令チェイニング方式は，ベクトル命令のパイプライン制御である。一つのベクトル命令の実行時間が長大なため，先行ベクトル命令の完了以前に演算結果を後続命令に渡す，チェインする必要が生じる。

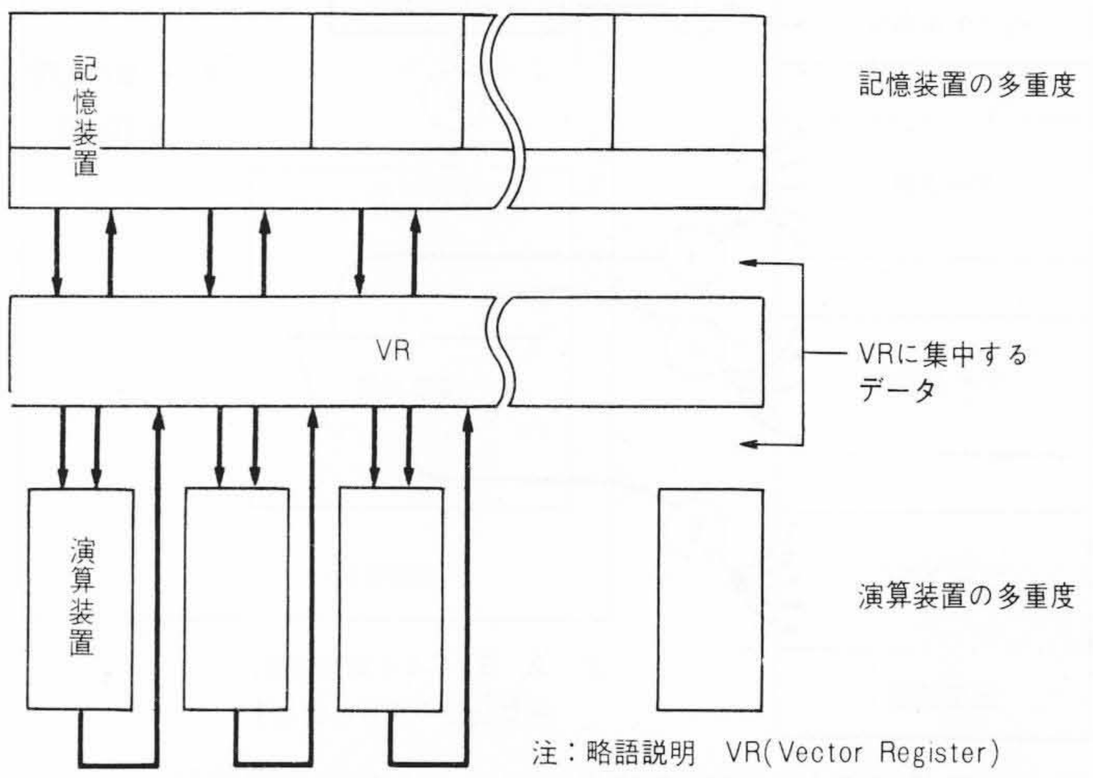


図11 スーパーコンピュータの性能を決める基本構成要素 複雑なスーパーコンピュータも，その最大性能は三つの構成要素の能力で決まる。

でも膨大な量になる。オペランドは8バイトのデータ量をもつので800MFLOPSの機械では毎秒6,400Mバイト(6.4Gバイト)のデータ供給能力を記憶装置はもつ必要がある。高速の記憶素子を採用したとしても，実現できるメモリサイクルは100ns前後のため，仮に80nsとすると8バイト幅のこの記憶装置1台で可能なデータ供給量は毎秒100Mバイトである。したがって，6,400Mバイトを供給するにはこの記憶装置が64台必要である。記憶素子技術の分野では性能に劣らず容量の確保が重要であり，装置性能の飛躍的向上はあまり期待できない。したがって，記憶装置の多重度を今後も増大させて必要能力を確保することになろう。その場合の最も困難な課題の一つは，接続ケーブルをいかに処理するかである。多重度64の場合でも64×72(本)×2(往復分)=9,216本の信号線の処理が必要であり，将来はもっと増大する。アーキテクチャ上の工夫によって1演算当たり必要な記憶装置参照をいかに減らすことができるかという点が将来の課題である。

(3)　ベクトルレジスタの能力

現在のスーパーコンピュータの特長の一つは，演算装置と記憶装置の間にVR(Vector Register：ベクトルレジスタ)を介在させてデータの流れを円滑化し，合わせて演算当たりの記憶装置参照回数を減らしていることである。しかし反面，すべてのデータの流れがここに集中しやすい。単純化すると**図11**に示すように1演算当たりVRと演算装置の間に3回，VRと記憶装置の間は読み出し，書き込み合わせて平均的に1回のデータ交換が発生し，全体で1演算当たり4回となる。800MFLOPSの機械では毎秒4×8×800=2万5,600Mバイト(25.6Gバイト)のデータ供給能力をVRがもたねばならないことになる。これは10nsごとに32の入口，出口から8バイトのデータを供給又は受け取ることに相当する。アーキテクチャ上の革新が期待される一つの分野である。

3.3　実効性能を決める因子

前節ではスーパーコンピュータの最大性能を決める因子について述べたが，実際のプログラムをスーパーコンピュータで処理する場合には必ずしも最大性能値から期待されるような高い性能は実現できない。実効性能は概略以下の二つの因子から決まる。

(1)　ベクトル化率：r

一般の利用者が作成したプログラムは自動ベクトルコンパイラの働きにより**図12**に示すようなスカラー命令とベクトル命令から成るオブジェクトプログラムに展開される。一方，スーパーコンピュータの処理装置は既存のスカラー命令を実行する機構と，スーパーコンピュータ用のベクトル命令を実行する機構を並置した形でもっており，前者をスカラープロセッサ，後者をベクトルプロセッサと呼んでいる。オブジェクトプログラムのうち，ベクトル命令はベクトルプロセッサで実行されるが，スカラー命令はスカラープロセッサで実行されるため，その性能は汎用コンピュータと大差のないものになる。したがって，実効性能を最大性能に近付けるためには，ベクトル命令に展開され実行されるプログラムの部分，これをベクトル化率，rと呼ぶ，を増やすことが重要になる。

図13はこの関係を更に定式化したもので，スカラー処理機能しかもたない汎用コンピュータ又はスカラープロセッサを使った場合のあるプログラムの実行時間をT_sとし，ベクトルプロセッサを使った場合の実行時間をT_vで表わすと，T_vとT_sの間には同図中に示すような関係が成立する。ベクトル化部分の性能向上率，aはスーパーコンピュータでは10以上の値を示すのが普通であるから，rの向上がT_vの短縮に必須であることが分かる。すなわち，rが0.9以上の値を実現しないと実効性能は最大性能と著しい乖離を示すことになる。

rは通常次の三つの要因からその上限が決まる。

(a)　問題から決まる上限

問題そのものがベクトル命令による並列性を不可能にしている部分がある。すなわち，問題の内のベクトル化可能な部分の集まりからrの一つの上限が決まる。

(b)　アーキテクチャから決まる上限

上記(a)の範囲では原理的にベクトル化が可能であるが，ベクトルプロセッサが対応するベクトル命令を備えていないためにベクトル化できない場合がある。スーパーコンピュータのアーキテクチャ技術の課題は(b)をできるだけ(a)に近付けることにある。

(c)　コンパイラ技術から決まる上限

上記(b)の点からはベクトル化できても，コンパイラが対応するオブジェクトプログラムを生成できない場合に発生する上限であり，現実のベクトル化率はこの値になる。

(2)　性能の改善率：a

ベクトル化部分がスカラー処理時に比べ何倍高速化したかを示す値で，基本的には最大性能の値が反映すると考えられる。しかし，次の理由からこの値も下方修正を受ける。

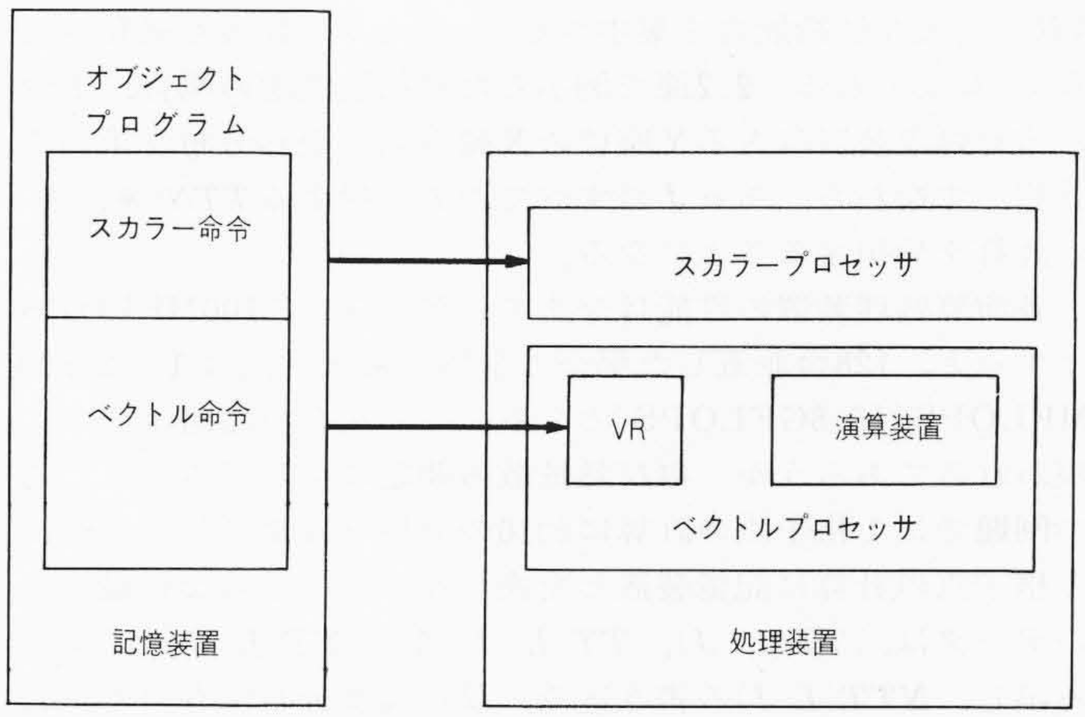
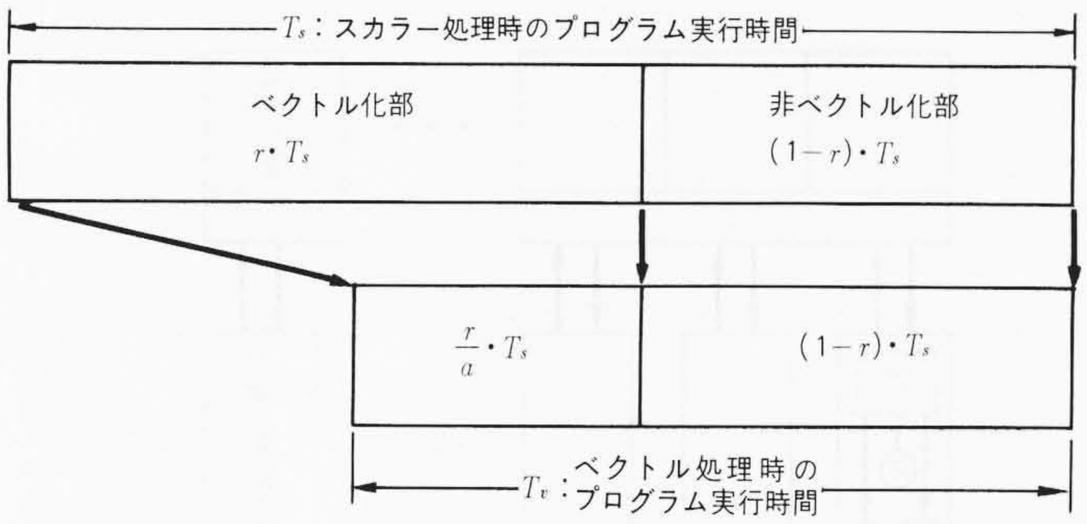


図12　スーパーコンピュータによるプログラムの実行　利用者の書いたプログラムのすべてが，ベクトルレジスタや多数の演算装置から成るベクトルプロセッサで実行されるわけではない。スカラープロセッサも重要な役割を演じる。



$$T_v = \frac{r}{a} \cdot T_s + (1-r) \cdot T_s$$

r：ベクトル化率

a：性能の改善率　aは更にベクトル長，オペランドの連続性の影響を受ける。

図13　スーパーコンピュータによるプログラムの実行時間　スーパーコンピュータの実効性能は，ベクトル化率rや性能の改善率の影響を大きく受ける。

(a) ベクトル長の影響

最大性能はベクトル長が無限の場合に実現する。現実のプログラムでは100以下のベクトル長の場合が多く，ベクトル命令の前後処理の影響が外に出て a が低下する。この改善もアーキテクチャ技術の課題である。

(b) データの非連続性の影響

必要なベクトルデータが記憶装置上に連続して配置されている場合に最も能率の良いデータの読み出し，書き込みが可能で，最大性能が実現できる。実際のデータは非連続データも多く a を低下させる。これもアーキテクチャ技術の課題の一つである。

以上の理由から実プログラム実行時のスーパーコンピュータの性能予測は極めて困難なものになっており，安定したスーパーコンピュータの性能の実現，これが今後のアーキテクチャ技術の最大の課題といえる。

3.4 多重処理装置方式への展開

3.2節で述べたように，今後の素子技術の進歩は大規模論理装置，大容量記憶装置の実現を容易にするが，素子間の接続技術に飛躍的進歩は期待できないとすると近い将来のスーパーコンピュータとしてどんな形のアーキテクチャが考えられるであろうか。図11に示すように，VRと演算装置を一体にしてベクトル演算処理装置とし，これを128台あるいは256台並べる構成が一つの例である。このような分割が可能ならばVRに過大な供給能力を集中させているという弊害も緩和されるように思われる。2.2節で例示した熱拡散問題の場合，1台の演算処理装置はあるY座標のX軸方向の全熱分布を求める計算，すなわち，あるJのすべてのIに対する$TTN(*, J)$の計算を分担することになる。

各演算処理装置の性能は今までの例に従って100MFLOPSとすると，128台並置した場合の全体の最大性能は1万2,800MFLOPS(12.8GFLOPS)となる。この場合の隘路はどこに現われるであろうか。再び熱拡散の問題に戻ってみよう。この例題では1格子点の計算に約10の演算を必要とする。一方，1格子点の計算に記憶装置と交換しなければならない最低限のデータは，$TT(I, J)$，$TT(I, J-1)$，$TT(I, J+1)$の読み出し，$NTT(I, J)$の書き込み，及び温度分布保存のための$NTT(I, J)$の読み出しと書き込み，$NTT(I, J)$の$TT(I, J)$への移動のための参照と，全部で8回の記憶装置参照になる。1格子点での演算量は10程度であるから1演算当たりの記憶装置参照は0.8回，すなわち1演算当たり約1回となり，現在のアーキテクチャとあまり差のない結果になる。すなわち，処理装置群と記憶装置群のデータ交換量は毎秒8×12.8＝102.4Gバイトになる。これは毎秒100Mバイトの能力をもつ記憶装置を1,024台並べねばならぬ量に相当する。これらの考察は共用記憶装置のアーキテクチャの選択が今後とも重要な意味をもつ一つの例であり，これを打開する各種アーキテクチャの研究が行なわれている。

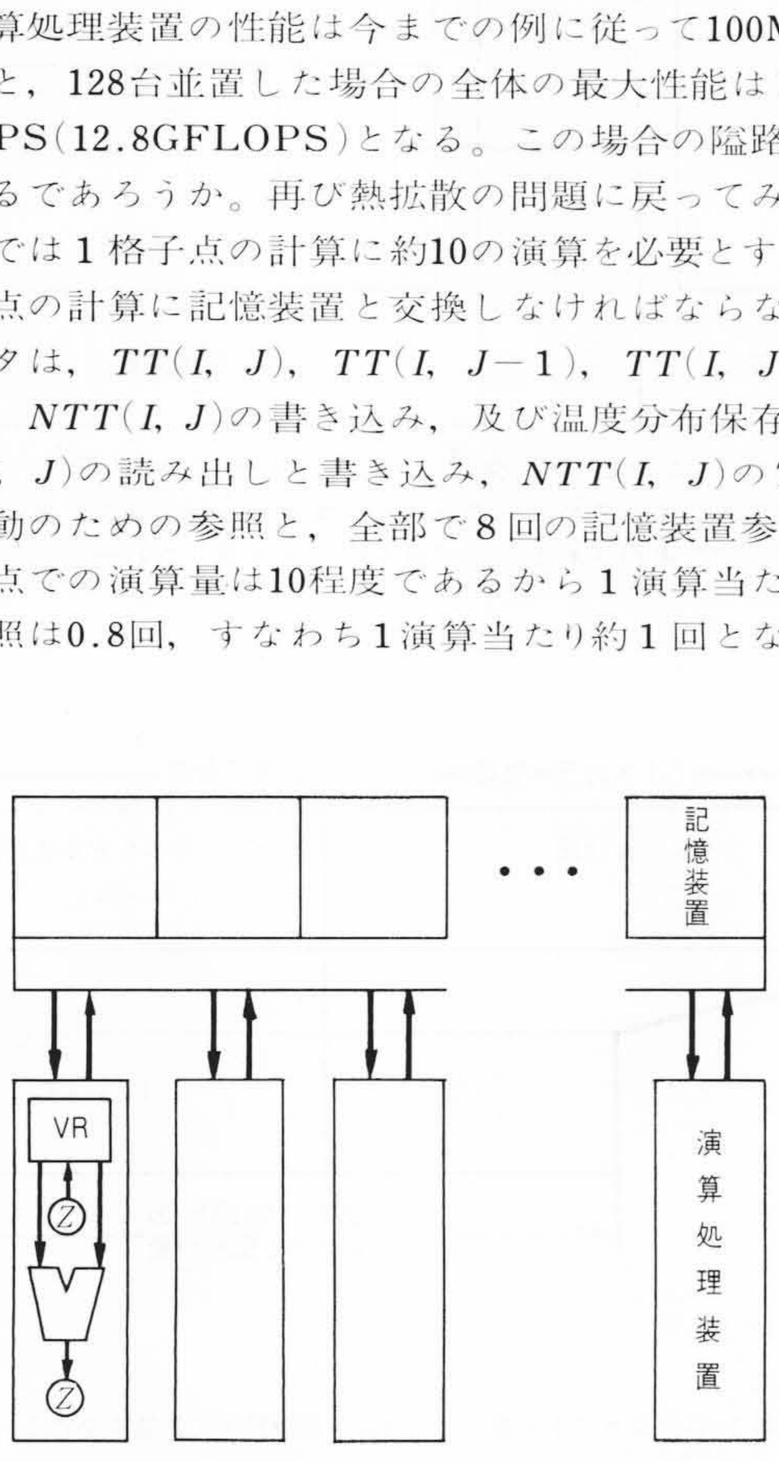


図14 多重プロセッサ方式によるスーパーコンピュータの構成法の一例　より速いスーパーコンピュータを実現するための一つの方法は，VRと演算装置とを組み合わせた演算処理装置を多数配するアーキテクチャである。

4 結　言

以上，汎用コンピュータアーキテクチャの抱えている幾つかの問題点を科学技術計算への利用という視点から分析し，それがスーパーコンピュータでどのように解決されたか，あるいはされつつあるかについて述べた。本稿では直接触れなかったが，スーパーコンピュータアーキテクチャを考える上では次の課題も重要である。

(1) ハードウェア技術

スーパーコンピュータには常に最先端の素子実装技術が使われる。それは主にマシンサイクルの短縮，利用可能な論理規模及び記憶容量の拡大という形でアーキテクチャに反映してくる。

(2) ソフトウェア技術

スーパーコンピュータが処理するプログラムは，複雑かつ大規模なものが多い。したがって，利用者のプログラムに要する労力をより軽減するようなソフトウェアが必要である。利用者にとってより簡潔な記述が可能なシステムでは，その分だけアーキテクチャの自由度が増し，より高い性能を実現する可能性が開けると考えられる。

(3) システム性能

スーパーコンピュータの性能は，問題により大きく変動する。最大性能と代表的プログラムによる実効性能とはかなり差が生じるし，更にベクトル命令の利用できない場合には事実上汎用コンピュータ並みの性能まで低下する。この変動をいかに少なくするかがアーキテクチャ技術の最大関心事である。同時に，高度なコンパイラ技術がこれを支える必要がある。

(4) 入出力機器

スーパーコンピュータは膨大な数値を扱うので，記憶装置に収容できないデータの管理方法を誤るとその部分が性能上の隘路となる。これを解決する一つの方法として，半導体素子を使った高速の固体ディスクが注目を浴びつつある。また，膨大な数値計算結果をいかに利用者に提示するかも大きな課題であり，画像表示装置が重要な役割を演じるようになってきた。

以上の課題を解決しながら，GFLOPSオーダの性能が1990年代には一般化し，スーパーコンピュータの社会的役割はいっそう増すものと思われる。

参考文献

1) 小高，外：超高速演算の動向，情報処理，21，9，927～937(1980)
2) 特集“スーパーコンピュータ”，日経エレクトロニクス，314，105～184(1983-4-11)
3) 小高，外：HITAC M-180内蔵アレイプロセッサ，日立評論，60，6，451～456(昭53-6)