NAL TM-325

UDC 681.31

航空宇宙技術研究所資料

TECHNICAL MEMORANDUM OF NATIONAL AEROSPACE LABORATORY

TM-325

FACOM-230-75アレイプロセッサーについて I

——ハードウェアの性能——

三好 甫・末松俊二

1977 年 3 月

航空宇宙技術研究所

NATIONAL AEROSPACE LABORATORY

昭和 52年 7月 8日

配布番号	171

所属 管理部

河崎俊夫 殿

企画課長

配布先限定文献の取扱いについての注意

この文献（航空宇宙技術研究所資料 TM-325）は、

配布先限定につき、取扱いには十分注意して下さい。

なお、貴殿の番号は上記番号です。また転退職の際には企画課

調査係あて、御連絡下さい。

FACOM-230-75アレイプロセッサーについて I*

——ハードウェアの性能——

三好　甫**・末松俊二**

1. 諸　言

1940年代ストアドプログラム制御方式の計算機が出現して以来今日に到る迄，その時代の計算機技術の先端を行く超大型機が現われ計算機工学および計算機利用技術の分野に大きな話題を提供したことがしばしばあった。例えば1950年代における IBM−STRETCH, UNIVAC−LARC，1960年代における IBM−360−85 および91，CDC−6600 等がそれである。STRECH，LARCは中央処理装置と入出力装置の同時動作とこれを制御する割り込み方式，先行制御方式等によりその時代の計算機技術の先端を行き，IBM−360−85 はバッファメモリを最初に使用した計算機として有名であり，IBM−360−91，CDC−6600 は演算装置における徹底的な演算の同時並行処理技術の採用により一時代を画した。これらの計算機で実現された技術の大部分はその後一般の計算機に取り入れられ，計算機技術の発展に多大の寄与をした。

1970年代において上記の超大型機に対応する計算機として1973年NASA−AMES−R．Cに導入されたILLIAC−Ⅳ，NASA−LANGLEY−R．C，ローレンス放射線研究所に導入されたCDC−STAR−100 を挙げることができる。しかしながら上記2機種はこれ迄の超大型機が汎用機であったのに対して Array-Processor（以下APと略称する。）と呼ばれる特徴をそなえている。1970年代の超大型機が汎用機ではなくAP という型式をとって現われた背景には，航空宇宙技術，原子力，気象科学等の大規模かつ高速な計算機の利用分野からの計算機の超高速性への要求と現在の計算機工学における計算機高速化技術の到達点とのギャップの存在という事実があり超大型AP の出現はその成功，不成功は別としても必然であるといえる。またこのギャップが将来にわたって存在しつづけるならば，今後の超高速大型機がAP の様な特異な型式をとって現われることも予測できる。我が国においても富士通(株)は航技研に対して現計算機システムの選考時にFACOM−230−75 をホストプロセッサとするAP（以下FACOM−APと略称する。）を提案してきた。航技研側はFACOM−APの仕様を検討した結果可成り大巾な仕様変更を要求したが，それは富士通(株)の受入れるところとなり，76年度中にAP ハードウェア，ソフトウェアの作成が一応完了し，77年度より実際問題による試運転が予定されている。しかしながらFACOM−APは同じAP といってもSTARやILLIAC−Ⅳ 程の巨人機ではないし（ハードウェアの規模でFACOM−230−75 CPUの1.5倍程度），またシステムにおける位置ずけもこれまで紙上で提案されたり，実際作成されたどのAP とも異なっているので，これを有効に使いこなすAP プログラムを作成し，効率よく運用するための運用体系を作成するためには「(1) FACOM−APのハードウェアの性能を詳細に検討すること。」「(2) 計算機ジョブをいくつかのパラメータで表現し，その様に表現されたジョブに対するFACOM−AP の実行速度を推定し，同時にそのパラメータを独立変数として航技研のジョブミックスを推定し，そのジョブミックスに対するFACOM−APの平均的実行速度を推定すること。」「(3) FACOM−230−75+AP システムを効率的に運用するため運用体系を作成し，効率的運用に対するシステム上のボトルネックがあればそれを発見し，オペレーティングシステムおよび言語プロセッサーの改良点を見い出すためのシミュレーションスタディを行うこと。」以上が必要となる。

本資料は上記3点のうち(1)についての検討結果を報告しAP の使用者が効率のよいAP プログラムを作成するための基礎を与え，同時に(2)，(3)の作業の基礎ずけを行うことを目的としている。

本論にはいる前にAP の出現の必然性と各種AP の中でのFACOM−APの位置ずけを行う。このことは航技研の計算機利用者がFACOM−APの概略を理解することに役立つであろう。

* 昭和52年1月5日　受付
** 計算センター

2. AP出現の背景とFACOM−AP

2.1 計算機高速化の技術とAP

2.1.1 演算装置に対するデータ供給能力向上の技術

計算機を実行速度という面からとらえた時，ソフトウェア技術とはハードウェアの速度を最大限に発揮させる技術といっても良いであろう。従って計算機の実行速度の上限はハードウェアの実行速度により定まるといえる。今，計算機の平均的な命令実行速度をP_tとすると

$$P_t = f(\max(T_\mu, C))$$

T_μはマシンクロックタイム，Cは主記憶サイクルタイム，$f(x)$はxの単調増大関数でfの形は機種毎に異なる。ところでT_μは現在の汎用大型機で100 ns*～30 ns程度であるがCは500 ns／1語～200 ns／1語程度でその間に5倍～10倍程度の差があり，上式から明かな様にT_μをいくら速くしてもP_tはCで押えられてしまう，CとT_μとのギャップを埋めるために考え出されたのがバッファメモリ方式とメモリインタリーブ方式である。

i) バッファメモリ方式

高価ではあるがT_μと同程度のサイクルタイムをもつ記憶装置（バッファメモリ）を小量演算装置と主記憶装置の間にはさみこむものであって演算装置はこの記憶装置と情報のやりとりを行う，演算装置の必要とする情報が全てバッファメモリ上にあれば計算機はあたかもT_μと同程度のサイクルタイムをもった主記憶をもっているかの様に動作できる。バッファメモリの技法は最初にIBM−360−85に採用され成功をおさめ，現在多くの大型機に採用されている。例えばIBM−370−195，168等，FACOM−230−75，M190，HITAC−8800，8700，M−180，ACOS−77−800，900等がある。バッファメモリの特徴を把握することはデータトロン205を使用した航技研の研究者にとっては容易である。データトロン205には通常のメモリよりも10倍速いアクセスタイムをもったメモリが80Wあり，これの使いこなしが計算速度を決定した。バッファメモリとデータトロンの高速メモリの決定的な差異はデータトロンにおいては高速メモリは使用者に見えていたのに対して，バッファメモリは使用者からは見えず，主記憶とバッファメモリの情報の交換は全てハードウェアが或る方式に従って自動的に行う所にある。従ってバッファメモリの成功の可否はできうる限り広い用途に関して以下の条件を満足する最適な方式を発見することである。

(1) バッファメモリの容量はせいぜい数K語であること。

(2) 演算装置の必要とする情報がバッファメモリ上にある確率をできるだけ大きくすること。

(3) 方式を実現するハードウェアが余り大きくならないこと。

(4) むだ時間が小さいこと。

以上の4つの条件は相互に矛盾するものであるため，主記憶とバッファメモリの対応ずけ，一度に行なうデータのやりとりの巾，バッファメモリから情報を追い出すアルゴリズムについて種々の方式が提案され，検討されている。

ii) インターリーブ方式

主記憶をそれぞれがメモリアクセスコントローラーをもつ多数のモジュール（通常バンクという）に分割し，各バンクから演算装置への情報の転送を1T_μずつづらして行う方式であって，この方式によれば1メモリサイクルタイムの間に多数のデータを演算装置に送り込むことができる。この方式に関するパラメータはバンク数と1バンクにおけるデータの巾である。

FACOM−230−75システムを例にとるとバンク数は4～32（これを4 way～32 wayのインターリーブと呼ぶ。），データの巾は2語である。メモリインターリーブ数を例えば4 WAYとするとFACOM−230−75システムのメモリの番地づけは下図の様になる。[4] 代表的な大型機のインターリーブ数とデータの巾を挙げるとIBM 370−195は

バンク 1		バンク 2		バンク 3		バンク 4	
⋮							
24							
16	17						
8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7

8～16 wayでデータの巾は64 bit，CDC−7600は16～32 way，60 bit巾，CDC−STAR−100は32 way，512 bit巾，FACOM−M190は4 way，64 bit巾である。

iii) バッファメモリ，インターリーブ方式の問題点

バッファメモリの技術は計算機プログラムのデータおよび命令シークェンスが適当な時間間隔内で見るならば局所化されていることに着目して開発された技術であるので，この局所性 ―これはバッファメモリの大きさと演算装置の実行速度に依存する― が破れる様なプログラムに対してはあまり効果がなく，最も極端な場合には依然として$P_t = f(\max(T_\mu, C))$となるが，現在バッファメモリを持つ計算機は科学技術計算に関しては平均的に95％程度のバッファメモリにおける情報存在確率を保証している様である。

インターリーブ方式はデータおよび命令シークェンス

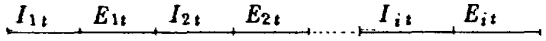
---

* nsは10^{-9}秒の略

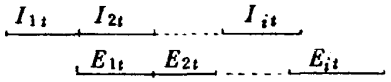
の順次性に着目して開発された技術であるのでメモリサイクルタイム以内に続いて同一バンクにメモリアクセスがかかれば，そのアクセスはメモリサイクルタイムが経過する迄は待たされることになりインタリーブの効果はなくなるのでこの様な方式をとる計算機の使用者はこのことを考慮してプログラムを作成しないならば，みすみす計算機の高速性を殺してしまうことになる。特にインターリーブ方式の場合，データの配列は完全に使用者に見えているので，これを無視してはならない。

2.1.2　計算機高速化のための技術

計算機が命令を実行する場合，その動作は基本的に2つに分かれる。即ち命令解読部（I部と略称する。）と解読された命令の実行部（E部と略称する。）である。今，I部の実行時間をI_t，E部のそれをE_tとするとこの命令の実行時間をP_tとすれば$P_t = I_t + E_t$となる。これをI部とE部を独立させ，E部で解読された命令を実行している時，I部は次の命令の解読を行う様にすれば$P_t = \max(I_t, E_t)$に短縮される。しかしながらI部もE部もそれぞれマシンクロックタイムT_uの数倍から+数倍かかるのでI部，E部の機能をそれぞれ更に細分して独立に動作できる様にして，それぞれの実行時間を$1T_u$にできるだけ近ずける様にすると理想的な条件のもとでは$P_t \cong 1T_u$となる。この3つの場合を図示すると第1の場合



となり，第2の場合



となり，第3の場合

となる。しかしながらこの様な並列化を徹底的に追求し実際に$P_t \cong 1T_u$を達成することは困難である。なぜならば各命令が全て$I_a \cdots E_y$を通過するわけではなくあるものはI部だけで終るものもあるし，又通過の順序が異なるものもある。また，各命令毎に各段階毎の通過時間が異なることがあるし，STORE；A，LOAD；Aの様な命令が続いた場合LOAD；Aは前の命令が終了する迄実行を待たされる。また判断分岐の処理が困難であり，割り込みもこの並列化の流れを止めてしまう等の困難がある。

これらの困難を克服するためには複雑な制御と膨大なハードウェアを必要とするが，これに挑戦した計算機としてIBM360－91，CDC－6600があり（両者の並列化の追求の方法は異なっている。）現在においてはIBM370(360)－195，CDC－7600があり，両者はともに$P_t \cong 1.5 \sim 2T_u$を実現し，汎用機としてはその実行速度において世界の最高を達成している。

2.1.3　計算機利用分野からの計算機高速化への要求とAPの必然性

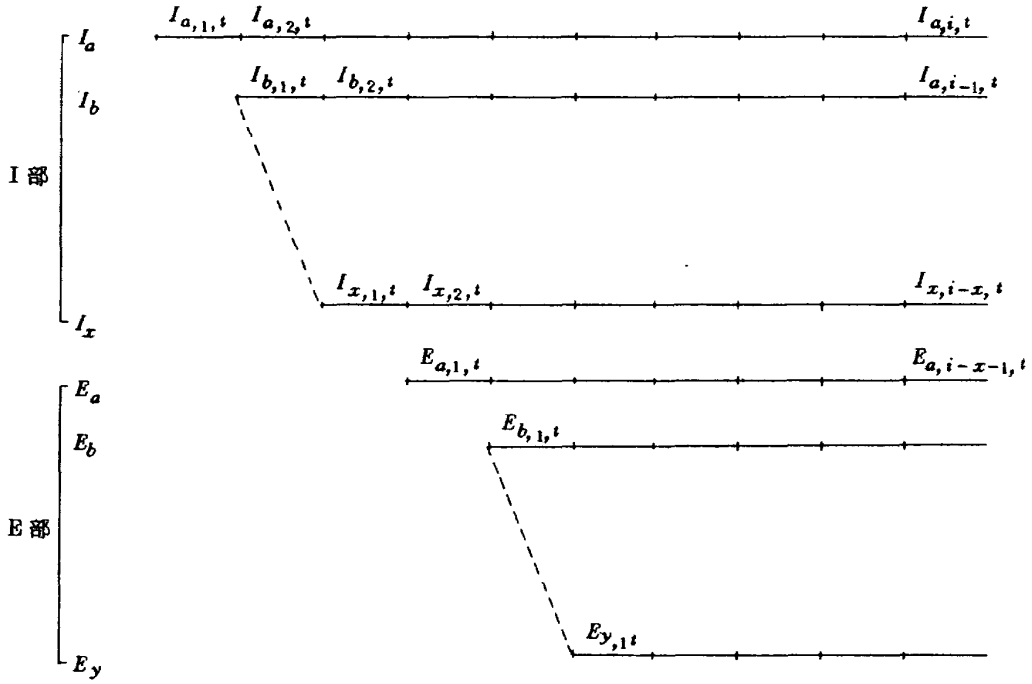
計算機の利用分野を航空宇宙技術の研究に限って考えることにする。この分野において計算機に高速性を要求するものとして

ⅰ）偏微分方程式の解法（差分法，有限要素法による空力，構造強度計算等）

ⅱ）多変数関数の極値問題（複雑な制御問題の最適解の計算等）

ⅲ）画像認識とその処理

等がある。[11] これ等の問題の内容について述べる余裕はここではないが，航空宇宙技術の開発，研究においてこれ



らの問題は計算機に数十内至数千MIPS(MEGA-INSTRUCTION PER SECOND, 即ち1MIPSとは1秒間に100万回計算機が命令を実行する能力をいう。)の高速性を要求している。ところが2.1.2でも述べた様に現在における計算機工学の粋をこらしたと考えられるIBM-370-195, CDC-7600といえどもその命令実行能力は10～15 MIPSといわれており, とうていこれらの要求を満たしえない。又FACOM-230-75は約3.5MIPSであり, 現在我が国最高速の計算機であると考えられるFACOM-M-190にしても高々7 MIPS前後である。この要求と計算機速度のギャップを埋めるものとして考えだされたのがAPである。

数十～数千MIPSを要求する問題のプログラム上の特徴は一言でいうならばプログラムの全ダイナミックステップ中におけるくり返し計算の比重の圧倒的な高さであり, FORTRAN言語で表現するならばDO-LOOPの比重の圧倒的な高さである。そしてこのDO-LOOPの中味はその大部分をアレイデータ(ベクトルデータ)またはビットストリングの計算として表現できるということである。しかも上に述べた様なプログラムの特徴は程度の差はあっても殆んど全ての科学技術計算についていえることである。アレイデータのDO-LOOP処理は一定の規則で並んだデータの列に一連の定まった命令列を適用することに帰着でき, この様な処理に対しては2.1.2におけるメモリインターリーブの技術, 2.1.3における並列化の徹底な追求の効果が100%実現される。APはこの点に着目して超高速化の技術をアレイデータの処理に集中, 徹底化させることにより大規模計算に関する超高速性の要求に答えようとするものである。

2.2 種々のAPとFACOM-AP

APという型式の計算機の提案は1950年代の後半から存在したが, 当時のハードウェア技術がこれを実現するに到っていなかったことおよび計算機の実行速度が急速に向上しつつあり, 使用者の側も或る程度の満足を感じていたこともあり実現にはいたらなかった。APの実現が具体的な問題としてとりあげられはじめたのは1960年代の後半以後のことである。

それ以後, 現在に到る迄いくつかのAPが実現した。その型式は汎用機に比較すれば千差万別といっても良いが大別すれば2つに型別することが可能である。

一つはILLIAC-IVにより代表されるプロセサーアレイ型という型であり演算装置を2次元的あるいはn次元的に数十台～数百台或はそれ以上配置し一つの計算機命令を同時に演算装置の数だけのデータに対して実行しようとするものである。今, 5MIPSの演算装置をn台並置すれば理想的にはアレイ演算に対しては$5 \times n$MIPSが実現できることになる。

今一つはCDC-STAR-100に代表されるパイプライン型と呼ばれるものであって, 2.1.2における並列技術をアレイ演算に対して徹底化し, 同時に代表的なベクトル演算, 例えば$A_i = B_i \circ C_i$($\circ$は例えば加減乗除, 内積等)をマクロ命令化することにより高速化をはかろうとするものである。

これら2つの型式にはそれぞれ長所, 短所があり現時点で一概にどちらが有利とはいえない。

表2.2.1および図2.2.1は各種APの概略の紹介と分類に関するものである。

図2.2.1におけるFACOM-APの位置はパイプライン型APの中で中央処理装置型とチャンネル型の中間に位置している。FACOM-APはホストプロセッサ(230-75)に入出力を依頼することに関しては通常の中央処理装置とは異なっているが, プログラムから見える命令体

表2.2.1 各種APの概略

機種＼項目	ホストプロセッサー	システム内の位置	型式	命令体系	速度	規格	
ILLIAC-IV	バローズ B6500	中央処理装置なみの演算装置	プロセッサアレイ型 メモリ個有	CPUと同じ	PEマシンサイクルタイム 240 ns	超大型機	
STAR-100	————	中央処理装置	パイプライン型 バーチャルメモリ	CPU+ベクトル命令	パイプライン3 速度 40 ns	同上	
F-AP	FACOM 230-75	中央処理装置なみの演算装置	パイプライン型	I/Oなし, 他はCPUなみ+ベクトル命令	パイプライン3 速度 90 ns	大型機	
IBM2938	IBM 360 44, 65, 75	チャネルと同格	同上	ベクトル命令のみ	パイプライン1 速度 200 ns	小型機	
IBM2937	同上	I/Oと同格	同上	同上	パイプライン1 速度 1 μs	同上	
UNIVAC-AP	UNIVAC 1106, 1108, 1110	チャネルと同格	同上	同上	パイプライン1 速度 230 ns	同上	
VECTOR-1	Sigma-2	中央処理装置と同格	同上	I/Oを除いてCPUなみ+ベクトル命令	パイプライン1 速度 1 μs	同上	
Raytheon ATP	Raytheon 703, 704, 706	チャネルと同格	同上	ベクトル命令のみ	パイプライン1 速度 不明	同上	
NOVA	————	中央処理装置	同上	CPU+ベクトル命令	ドラムメモリ	同上	

上の表の各種APを分類すると

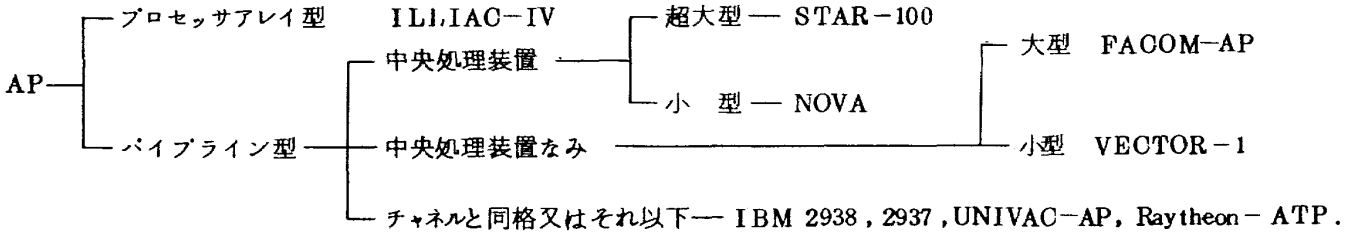


図2.2.1　各種APの分類

表2.2.2　（表中の数字の単位はMIPS）

機種	下限	上限	公称
超大型AP	15	700	100
大型汎用機A	10	25	15
大型汎用機B	2	5	3

系，レジスタを持ち，ユーザモード，モニタモードをもっている等を考えると通常の処理装置に近いといえる。

ここまでではAPについての一般的な概説を終えて以後FACOM−APについて述べることになるがその前にAPを使用する場合に一般的に注意すべき点を強調しておくことにする。

一般の汎用大型機についてもいえることであるが，現在の計算機はその高速性を実現するために，2.1で述べた様な技術を多かれ少なかれ取り入れており，使用者のプログラムがその技術的特徴に適合するしないにより実行速度に大きな差が生じてくる傾向にあり，計算機の実行速度の比較は以前の様に簡単ではなくなってきている。このことはFACOM−230−75とHITAC−8800の選考におけるベンチマークテストにおいても歴然と現われていたし，その他の機種間のベンチマークテストにおいても常に現われる事実である。そこで使用者の側もできる限り自己の使用する計算機のハードウェアの特徴を知ってプログラムを作成することが望ましい。この様な傾向はAPの様な計算機においては一層極端な形で現われる。そこでAPの使用者はAPハードウェアの特徴を押さえてプログラムを作成す可きであって，この資料の目的もそこにある。APおよび最近の大型汎用機の速度の上限と下限がどの程度に開いているかを一つの非公式資料から取りだして表2.2.2に示す。

文献解説

ILLIAC−Ⅳについては1)，2)，3)特に3)はILLIAC−Ⅳの主担当者の一人であった研究者の著作であって計画の発生から実現過程，システムのハードウェアソフトウェアにくわしい。

STAR−100については文献3)，4)に若干ふれられているがくわしくは5)，6)を参照のこと，CDC−7600については7)を参照のこと，360−195に関しては8)を参照のこと。

バッファメモリについては4)第3章と第3章に文献表がのっている。又7)を参照のこと。

メモリーインタリーブについては3)，7)，8)を参照のこと。

並列化技術に関しては3)にも若干ふれられているが文献9)に徹底的に論じられている。特に例をIBM−360−91，CDC−6600にとり，並列化を徹底化するためにハードウェアにどの様な機構が必要となるかについても論じられている。11)には空気力学の例についてどの様な問題がどれ位の高速性を計算機に要求するかについて詳細に論じられている。

3. FACOM−APのハードウェアについて

3.1 FACOM−APハードウェアの概略

くわしくは文献9)を参照のこと。ここではFACOM−APのハードウェアについて第4章以降の理解を助ける目的で概略について述べる。これはまたFACOM−APを使用する場合に知っておかなければならない最低限の知識でもある。

システム構成は図3.1.1の通りである。又FACOM−APの内部構成は図3.1.2の通りである。

図3.1.1および図3.1.2においてnMDW/Sは一秒間でのデータ転送能力を示す。DWはダブルワードの略，MDWはメガダブルワード（1MDWは100万ダブルワード）の略である。CPUは230−75をCHCはチャンネル制御装置，DIACは診断装置，P_1はベクトル加算パイプライン，P_2はベクトル乗算パイプライン，MBはビットストリングパイプラインをそれぞれ示す。

図3.1.2においてメモリA，B，C，Dはそれぞれ2モジュール，8バンクから構成され，0～7はそれぞれ128K語で構成される。航技研に予定されているシステムは6，7が欠けてけている。従って0～3迄512K語は32WAYインターリーブ512K～768K語は16WAYインタリーブとなる。したがって番地ずけは図3.1.3の様になる。そのため航技研システムでは0番地より524287番地迄は32MDW/Sのデータ供給能力をもつが524288番地から786431番地迄は16MDW/Sのデータ供給能力しかもたない。後で述べる様にベクトル演算$A=B+C$を単精度で行なった場合，FACOM−APはベクトルエレメントの処理に0.5T_M（$1T_M=90$ns）しかからないので45nsで3シングルワードのデータを必要とする。これはほぼ32MDW/Sのデータ供給能力に対応しているのでデータが524288番地より後にある場合にはデータ供給能力に不足をきたす結果となる。これをおぎなう意味でベクトルレジスタが1792語おかれている。ベクトルレジスタはデータレジスタとともにアクセスタイム60nsで動きパイプラインへのデータ供給能力は44MDW/Sである。バッファメモリはアクセスタイム60ns，データ供給能力は11MDW/Sである。バッファメモリは主としてプログラムの命令列のAPへの供給を行い。ベクトルレジスタはベクトル命令（以下V型命令と略称する）のオペランドデータの供給の補助を行ない，データレジスタは汎用レジスタとして使用される。なおバッファメモリはプログラマからは見えないがデータレジスタ，ベクトルレジスタ（以下V−Regと略称する）はプログラマブルであり，ベクトルレジスタはフォートランからも見えるレジスタである。

(1) データ形式

(i) ビットデータ（ビットストリングを除いて75

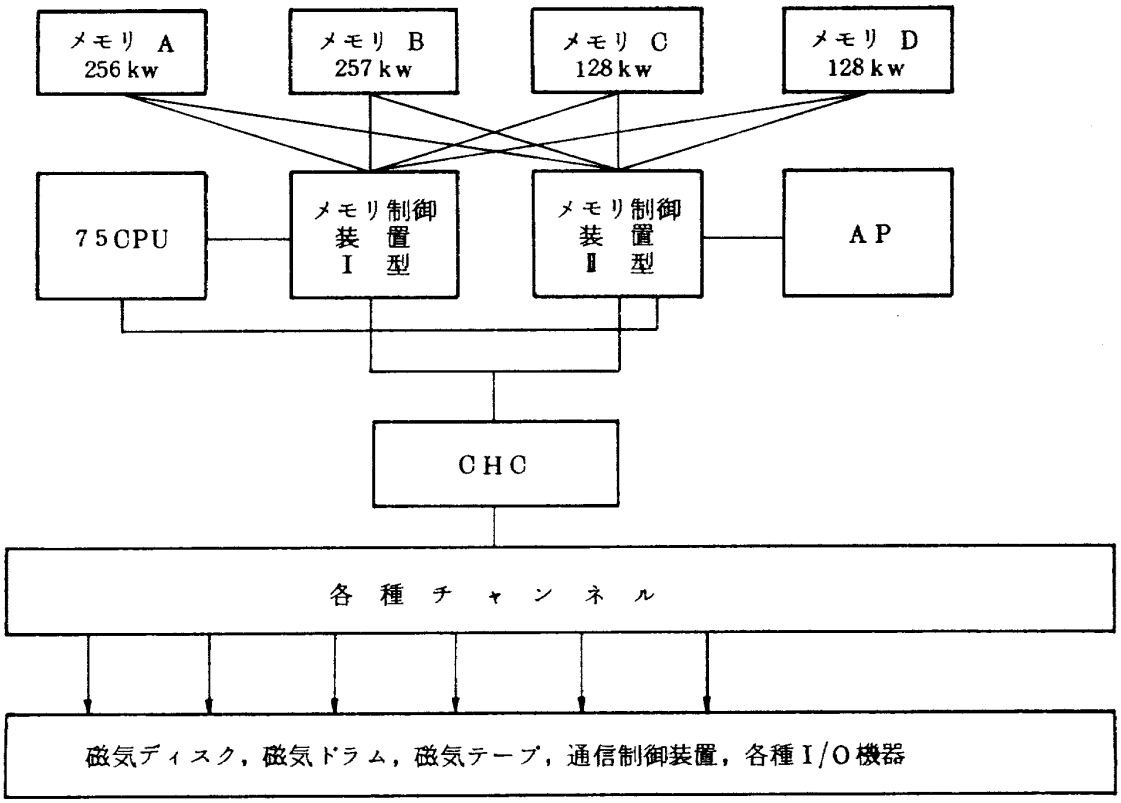


図3.1.1 75CPU+APシステム概略図

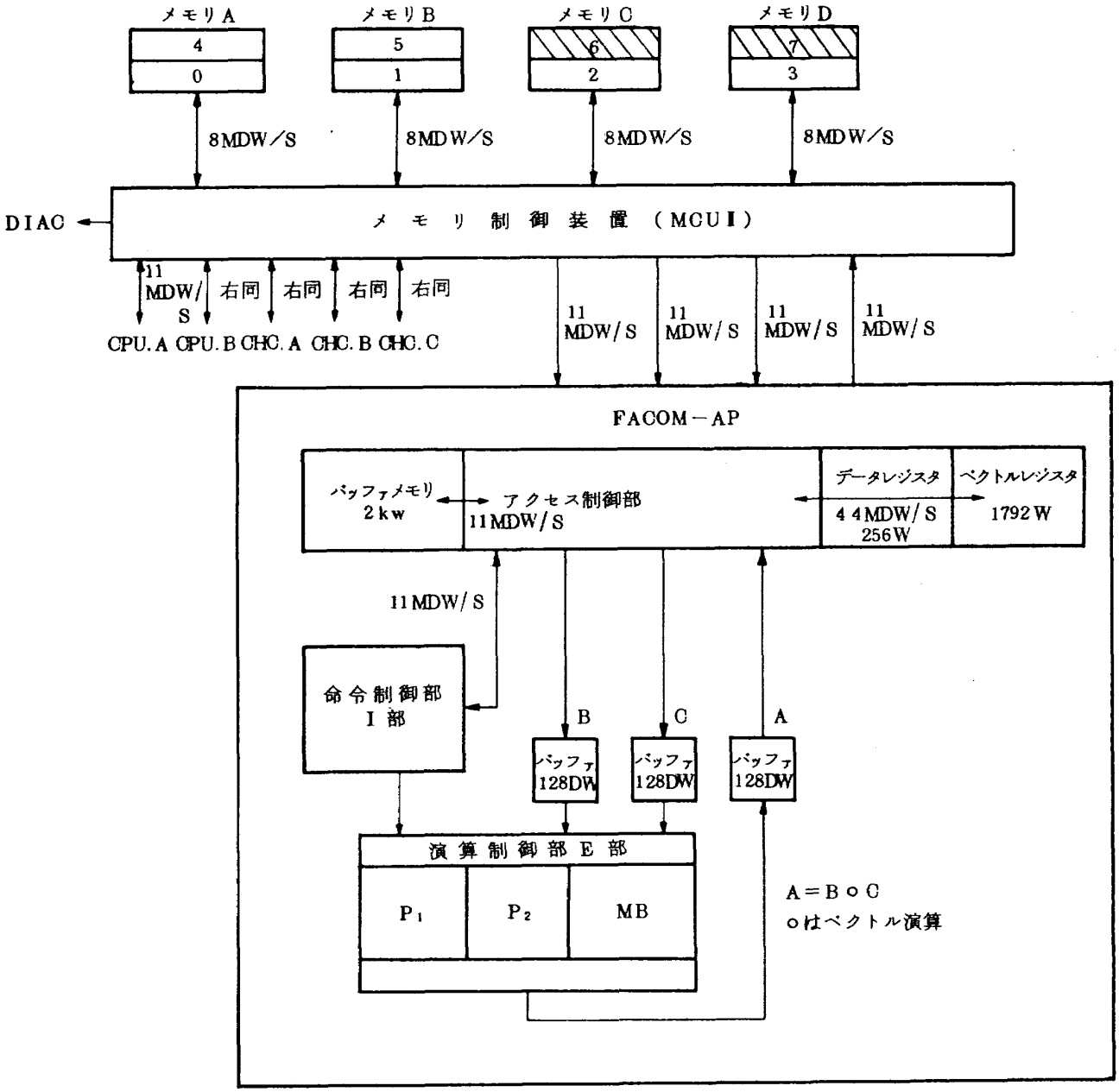


図3.1.2 FACOM-APハードウェア概略図

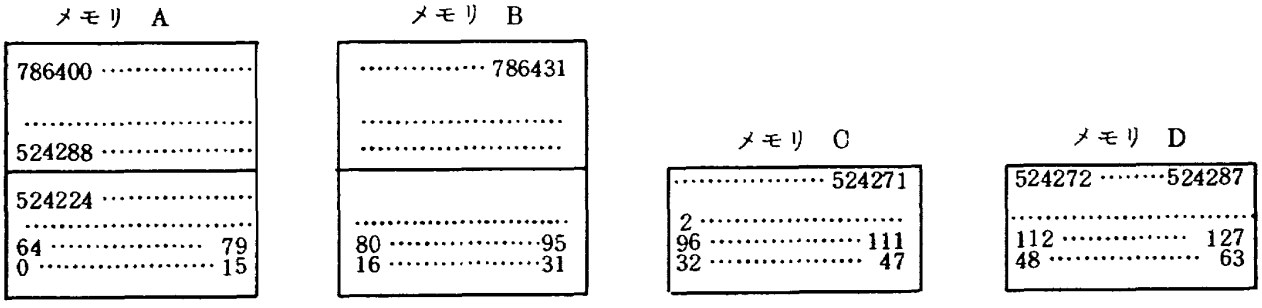


図3.1.3 航技研システムのメモリ番地ずけ

CPUと同じ）

(ii) 固定小数点（75 CPUと同じ）

(iii) 浮動小数点は単精度，倍精度，4倍精度（75と同じ）

(2) データの種類

(i) スカラ

(ii) ベクトル

(iii) リストベクトル

(iv) ビットスパースベクトル

(v) インデックススパースベクトル

(vi) ビットストリング

(2)の(ii)～(vi)迄はAPに固有のものであって75 CPUのアレイがスカラの集合として定義され取り扱かわれていたのと異なる。

3.2 FACOM−APの命令体系と実行速度

命令機能についてくわしくは文献9)を参照のこと，但し実行速度は掲載されていない。

表3.2.1においてM型命令とはメモリーデータレジスタ間の演算であり，R型命令とはデータレジスターデータレジスタ間の演算である。M型命令は2アドレス命令でR型命令は3アドレス命令である。これらはいずれもスカラーを取り扱かう。V型命令はベクトル演算を取りあつかい一般に3アドレス方式でメモリ（V−Reg）−メモリ（V−Reg）演算である。M型命令の実行速度はデータがバッファメモリ上にある場合についてであり，V型命令の実行速度はデータ供給能力，パイプラインの立上り時間，後処理時間を考慮しないいわば裸の実行時間である。又V型命令の実行時間中のNはベクトルオペランドの長さを示す。表3.2.1.aにおいてmはレジスタの数であり，表3.2.1.bにおいてCVA命令$\left(A_i=\sum_{j=1}^{L_B}(B_i+C_{i+j+1})\right)$

CVM命令$\left(A_i=\sum_{j=1}^{L_B}B_iC_{i+j+1}\right)$において$m=L_B$（Bオペランドの長さ）であり，PMM命令$\left(A_i=\sum_{j=1}^{L_B}B_jC_{(i-1)L_B+j}\right)$において$m=L_B$であり，POLY命令$\left(A_i=\sum_{j=1}^{L_B}B_j(C_i)^{L_B-j}\right)$において$m=L_C$（Cオペランドの長さ）である。これらの命令においてNはそれぞれ$N=L_C-L_B+1$，$N=L_C-L_B+1$，$N=L_B$．$N=L_B$となる。またT_Iは演算器の実行速度を示す。

表3.2.1.a　1クロック＝1 T_u ＝90 ns

M 型 命 令		R 型 命 令	
命 令 名 称	実行速度 クロック数 (T_u)	命 令 名 称	実行速度 クロック数 (T_u)
Load	2	Add Register Fixed Point	1
Load Immediate	1	Add Register Floating Point Single	8
Store	2	Add Register Floating Point Double	8
Add	2	Add Register Floating Point Quadruple	13
Add Immediate	1	Subtract Register Fixed Point	1
Subtract	2	Subtract Register Floating Point Single	8
Subtract Immediate	1	Subtract Register Floating Point Double	8
Multiply	7	Subtract Register Floating Point Quadruple	13
Shift Right Logical	1	Multiply Register Fixed Point	7
Shift Left Logical	1	Multiply Register Floating Point Single	9
Shift Right Arithmetic	1	Multiply Register Floating Point Double	12
Shift Left Arithmetic	1	Multiply Register Floating Point Quadruple	20
Shift Right Circular	1	Divide Register Fixed Point	44
Shift Left Circular	1	Divide Register Floating Point Single	23
Branch on Condition	4 成 立 5 不成立	Divide Register Floating Point Double	30
Branch and Link	4	Divide Register Floating Point Quadruple	79
Branch on Count	4 成 立 5 不成立	Transfer Register	1
Set Bit	1	Masked Transfer Register	2
Reset Bit	1	Load Multiple Registers	2+2m
Set Bit of Memory	4	Store Multiple Registers	2+2m
Reset Bit of Memory	4	Supervisor Call	250
Test Bit of Memory	4	Supervisor Release	70
Load Condition Code and Set	16	Call CPU	8
		Load Data Base Register	5
		Load Vector Register	900
		Store Vector Register	900
		Load Program Base Register	8
		Buffer Memory Clear	130
		APU No Operation	1
		Halt APU	30
		Clear Watch dog Timer	5

表 3. 2. 1. b

V 型 命 令				単位 $1T_M = 90$ ns Nはオペランド長
命 令 名 称	省略型	単 精 度 (T_I)	倍 精 度 (T_I)	4 倍精度 (T_I)
Vector Move	VMV	$\frac{1}{2}N+8$	$N+8$	$4N+8$
Vector Add	VAD	$\frac{1}{2}N+8$	$N+8$	$4N+8$
Vector Multiply	VML	$N+9$	$2N+11$	$8N+10$
Vector Divide	VDV	$9N+10$	$16N+12$	$63N+20$
Average	AVG	$\frac{1}{2}N+8$	$N+8$	$4N+8$
Vector Sum	VSM	$\frac{1}{2}N+31$	$N+24$	$4N+20$
Vector Norm	VNM	$N+31$	$2N+27$	$8N+19$
Inner Product	IPD	$N+31$	$2N+26$	$8N+19$
Adjacent Mean	AJM	$\frac{3}{4}N+8$	$\frac{3}{2}N+8$	$4N+8$
Find Address MAX	FAX	$2N+48$	$2N+30$	$8N+25$
Find Address MIN	FAN	$2N+48$	$2N+30$	$8N+25$
Find Address MIN Positive	FAP	$2N+48$	$2N+30$	$8N+25$
Find Relation Equal	FRE	$2N+30$	$2N+30$	$4N+30$
Find Relation Not Equal	FRNE	$2N+30$	$2N+30$	$4N+30$
Find Relation Low	FRL	$2N+30$	$2N+30$	$4N+30$
Find Relation High or Equal	FRHE	$2N+30$	$2N+30$	$4N+30$
Compare Equal	CME	$\frac{3}{2}N+10$	$2N+10$	$4N+10$
Compare Low	CML	$\frac{3}{2}N+10$	$2N+10$	$4N+10$
Convolving Add	CVA	$(m+28)N+8$	$(2m+20)N+8$	$(8m+16)N+8$
Convolving Multiply	CVM	$(m+19)N+12$	$(2m+12)N+15$	$8mN+20$
Partial Matrix Multiply	PMM	$(m+19)N+12$	$(2m+12)N+15$	$8mN+20$
Polynomial	POLY	$3.5mN+13$	$8.5mN+20$	$35mN+34$
Scalar Multiply and Vector Add	SMVA	$3.5N+14$	$5N+17$	$8N+21$
Vector Move with Distance	VMD	$N+8$	$N+8$	$4N+8$
Vector Masked Move	VMM	$N+2P+33$ *	$N+2P+33$ *	$4N+33$
Logical Sum	LSM	$2\lceil N/2^5 \rceil+5$		
Logical Product	LPR	$2\lceil N/2^5 \rceil+5$		
Logical Difference	LDF	$2\lceil N/2^5 \rceil+5$		
One's Count	ONC	$2\lceil N/2^5 \rceil+5$	all "0" の word	
		$2P+5$ *	それ以外の word	

*) Pはビットストリング中の on bit の数である。

4. FACOM-APのベクトル演算速度

4.1 現実の条件のもとでのベクトル命令の算出法

3.2の表3.2.1におけるベクトル演算命令の実行速度はいわば理想的条件下における裸の実行速度であることは表3.2.1の説明でふれたが，この速度は現実には種々の環境条件により可成りな範囲で変化する。実行速度に影響を与えるものを列挙してみると，

(1) データ供給能力の不足によるおくれ時間
(2) ベクトルを操作するためのディスクリプタ制御の前処理時間
(3) 最初のオペランドをもってくるアクセス時間
(4) パイプラインの立上り時間
(5) 後処理時間

(1)をT_Mで，(2)~(5)をT_pで表わし，表3.2.1の裸の実行時間をT_Iで表わす。T_Iは$T_I=\lambda N+\mu$の形をしていた。T_Mは式(4.1.1)で表わされる。

$$T_M=\begin{cases}(1+\beta)\left\{\left\{(m-n)d+\frac{p}{11}\right\}/\left(\frac{w}{11}\right)\right\}\lambda N-\lambda N & ;\ 〔\ 〕>1\\ \beta\lambda N & ;\ 〔\ 〕\leq 1\end{cases} \qquad (4.1.1)$$

(4.1.1)式において，N，λは表3.2.1の通りであり，nはベクトルレジスタ上のオペランドの数(0，1，2，3)
mはベクトル演算に必要なベクトルオペランドの数，例えば$A_i=B_i\pm C_i$の場合は3，$s=\sum a_i\times b_i$の場合は2である。dは一つのベクトルに対する1T_u当りのデータ要求量，例えば$A_i=B_i+C_i$は単精度の場合$\lambda=0.5T_u$でありベクトルは単精度であるので0.5DWとなり$d=0.5\,\mathrm{DW}/0.5T_u=1\,\mathrm{DW}/T_u$だから$d=1$となる。$P$はCPUのデータ要求量で最大1.5DW，$w$はインターリーブ数，$\beta$は待ち時間係数で〔　〕>1の場合，最大で0.25，〔　〕≦1の場合，$N=5$，10，15，20，30で0.2，0.15，0.10，0.05，0.0程度である。ここでベクトル演算の総時間をT_eとすると

$$T_e=(T_I+T_M+T_p)\cdot T_u \qquad (4.1.2)$$

T_pはNに関係しないので$T_M=\gamma\lambda N$とおくと

$$\gamma=\begin{cases}(1+\beta)\left[\left\{(m-n)d+\frac{p}{11}\right\}/\left(\frac{w}{11}\right)\right]-1 & ;\ 〔\ 〕>1\\ \beta & 〔\ 〕\leq 1\end{cases} \qquad (4.1.3)$$

すると

$$T_e=\{(1+\gamma)\lambda N+\mu+T_p\}\cdot T_u$$

従ってベクトル1要素当りの演算時間は

$$\tilde{T}_e=T_e/N=\{(1+\gamma)\lambda+(\mu+T_p)/N\}T_u \qquad (4.1.4)$$

$N\to\infty$とすると$\tilde{T}_e$は$(1+\gamma)\lambda\cdot T_u$に収束する。即ち$N$が十分大きければ$\tilde{T}_e\fallingdotseq(1+\gamma)\lambda T_u$と考えてよい。$\tilde{T}_e$は32WAYの場合単調に減少して$(1+\gamma)\lambda T_u$に近ずくが16WAYの場合，ベクトルレジスタの使用によりNが有限な所で極小値をもつことに注意する必要がある。またデータ供給能力がベクトル演算のパイプライン速度に対して十分な場合，即ち式(4.1.1)において〔　〕<1であれば式(4.1.4)よりNがある程度大きければ$\gamma=0$，$(\mu+T_p)/N$が十分小となり$\tilde{T}_e\fallingdotseq\lambda\cdot T_u$となりAPの実行速度は現実に理想的な速度に達する。したがってλが小さい，即ち実行速度の早いベクトル命令に関しては〔　〕<1とするためにベクトルレジスタの有効使用が望ましい。このことはベクトル命令のオペランドが16WAYの領域に落ちる場合を考えて特に注意を要する所である。またNが小さい所ではT_pの値が$\tilde{T}_e$にきいてくる。T_pの値は命令の続きかたや即ちベクトル命令とM型およびR型命令のまざり方と，全オペランドがベクトルレジスタ上にあるかないかで定まる。これを表(4.1.1)に示す。

表4.1.1 T_pの値

命令の続き方	ベクトルレジスタ	T_p(単位T_u)
M，R-V	全オペランドがベクトルレジスタにある。	23
V-V	オペランドのうちベクトルレジスタ上にないものがある。	35
V-M，R	全オペランドがベクトルレジスタにある。	16
	オペランドのうちベクトルレジスタにないものがある。	16

上の表からNが小さい時には命令の続き方つまりベクトル命令はできるだけかためて行うこと，およびベクトルレジスタを有効に利用することが望ましいことがわかる。

4.2 V型命令の実行速度図の説明

V型命令の実行速度を，図4.2.1～図4.2.44に示し，各々のV型命令の性質と図における実行速度曲線の特徴を説明する。一つのV型命令について，命令が単精度，倍精度の場合にそれぞれ，インタリーブ数が32WAYと16WAYに分けて4つの図で示してある。それらの図において，縦軸はそのV型命令の実行速度をT_uを表示し，横軸はベクトルオペランドの長さNを10^3まで対数表示してある。以下それぞれのV型命令について説明するが，その方法として，まず最初にFACOM230-75，FORTRAN-H(以下FORTRAN-Hと略称する)とFAC

OM 230-75 APU AP－FORTRAN（以下AP－FORTRANと略称する）でベクトル演算命令を使用した場合としない場合のコーディングの相違を示す。但し，AP－FORTRANでベクトル演算命令を使用しない場合はFORTRAN－Hのコーディングと同様になるので，ここではFORTRAN－Hを代表させる。従ってAP－FORTRAN と略称した場合は，特に断りのない限りベクトル演算命令を使用したものとする。次に，今述べた2つのコーディングされたプログラムを翻訳時に相対プログラムのFASP形式として記述し*，各FASP命令とその命令の総実行時間（T_u）を比較したものを表にして，表4.2.1～表4.2.11に示す。そして最後に実行速度図における実行速度曲線の特徴とそのV型命令の性質を説明する。その実行速度曲線の特徴は箇条書きにして説明し，その実行速度曲線の区別記号は〔V－V〕，〔V－M・R〕は命令の続き方を示し，V-REG-0～3は，ベクトルレジスタ上のオペランド数を示す。従がって図中で記号の後に〔V－V〕V－REG－0と表示している場合は，その記号の実行速度曲線はV－V型命令でV－Reg上にオペランドが無い場合（以下V－V型V－Reg－0と略称する）を示す。以下同様に表示してあり，同様に略称する。

図4.2.1～図4.2.44の実行速度曲線において，共通している特徴を述べておくと，これらの実行速度曲線に示される，V型命令の実行速度は横軸に示してあるNが大きくなると相対的に速くなる。またV－V型命令とV－M・R型命令はNが非常に小さい時は，V－V型命令の方が実行速度は速い。しかしNが大きくなってくるとその実行速度は変化するので以下の図の説明に注意しなければならない。しかし，V－V型命令でV－Reg 上に全オペランドがある場合は，Nに関係なく，他の場合よりもその実行速度は速い。以下それぞれのV型命令の実行速度図について説明する。

4.2.1 VAD（Vector Add）

VADの実行速度図は，図4.2.1～図4.2.4に示してある。FORTRAN－Hでコーディングすると次の様になる。

```
      DO   10   I＝1,N
      A(I)＝B(I)＋C(I)
10    CONTINUE
```

またAP－FORTRAN でコーディングすると次の様になる。

```
      A(＊)＝B(＊)＋C(＊)
```

表4.2.1はVADの演算において，データ供給能力は十分でありその他の環境においても理想的条件を満足しているものとすると，その総実行速度の比較から，230－75は12.2T_uであり，FACOM－AP でベクトル演算命令を使用した場合は0.5T_uであるからFACOM－AP は現在の230－75に比べて，24.4倍の速度である。従がってFACOM－AP でベクトル演算命令を利用した方がコーディングは便利であるし，その実行速度は速い。またFACOM－AP でベクトル演算命令を使用しない場合は，現在の230－75よりも実行速度は遅くなるので以下に述べる他のV型命令においても実行速度表の比較に注意しなければならない。表4.2.1に示したFACOM－AP の実行速度は前述した様に理想的条件下であって，実際は種々の環境により実行速度は変わる，以下それについて，図の説明を入れながら述べる。図4.2.1は単精度でインタリーブ数が32WAYの場合，図4.2.3は倍精度で32WAY，図4.2.2は単精度で16WAY，図4.2.4は倍精度で16WAYの場合を示している。それら4つの図の実行速度曲線の特徴を箇条書きにして示す。

図4.2.1においては，

i）実行速度曲線において，V－V型V－Reg－1と2

表4.2.1

プログラム形式	230－75	FACOM－AP	
		ベクトル演算命令なし	ベクトル演算命令使用
FASP形式	FL	L	VAD
	FA	L	
	FST	ARS	
	IJXLE	ST	
		ADI	
		BRT	
総実行速度	12.2T_u	18T_u	0.5T_u

＊ FASP命令について，230－75は文献12），FACOM－APについては文献9）を参照のこと。

が重なっており，V−M・R型V−Reg−1と2も重なっている。

ii) Nがある程度大きくなると，実行速度は命令の続き方よりも，V−Reg 上のオペランド数に依存する。図中のV−V型V−Reg−0とV−M・R型V−Reg−1，2，3 との実行速度曲線の交差を指す。

iii) Nが非常に大きくなると，命令の続き方には関係なく，V−Reg 上のオペランドの有無で実行速度の収束値が異なってくる。そしてV−Reg 上にオペランドが有る場合は，理想的条件下における実行速度にほぼ近づいていく。

以上のことは図4.2.3についても同様のことがいえる。しかし図4.2.3は倍精度であるので，図4.2.1に比べて，その8本の実行速度曲線は相対的に遅くなり，また ii) で示した，V−Reg 上のオペランド数に依存して交差する実行速度曲線のNの値が異なる。

図4.2.3は単精度でインタリーブ数が16WAYである。この図においては，i) に示した様な実行速度曲線の重なりはない。そして ii) に示した様に，Nがある程度大きくなると，命令の続き方よりもV−Reg 上のオペランド数に依存してくる。その特徴としては，

iv) Nが非常に大きくなるとV−Reg−0の場合，V−Reg−1の場合，V−Reg−2と3の場合はそれぞれ実行速度の収束値が異なる。

このことは，図4.2.4の倍精度の場合についても同様のことがいえる。またその実行速度曲線は，インタリーブ数が32WAYの場合と同様に実行速度は相対的に遅くなっている。

以上VAD の4つの図についての実行速度曲線の特徴について述べた。このことから，VAD の演算においては先に示した様にNが非常に小さい時は，その実行速度は命令の続き方に依存するが，Nがある程度大きくなってくると，ii) に示した様にV−Reg 上のオペランド数に対する依存度が高くなり，Nが非常に大きくなると iii) iv) に示した様に完全にV−Reg 上のオペランド数に依存してしまう。このことは，データ供給能力に関係し，4.1における式（4.1.1）〜式（4.1.4）より説明される。また同インタリーブ数において，単精度と倍精度の実行速度曲線の性質は同じであるが，実行速度が違うのは表3.2.1.bのVAD の実行速度より同式にて説明される。このVAD において特に注意すべきことは，iv) で示した様にインタリーブ数が16WAYの場合はV−Reg 上のオペランド数によりその実行速度の収束値が異なっている。従がって4.1で述べた様にV−Reg の有効使用が望ましくなる。

4.2.2 VML（Vector Multiply）

VML の実行速度曲線図は，図4.2.5〜図4.2.8に示してある。

FORTRAN−Hでのコーディング

```
      DO   10   I＝1，N
      A(I)＝B(I)＊C(I)
10    CONTINUE
```

AP−FORTRANでのコーディング

```
      A(＊)＝B(＊)＊C(＊)
```

表4.2.2のVML の総実行速度をVADの場合と同様に比較すると，FACOM−APは1T_uであり，現在の230−75は13.2T_uであるので，VML の演算においては，現在の230−75に比べて13.2倍の速度になる。図4.2.5は単精度でインタリーブ数が32WAY，図4.2.7は倍精度で32WAY，図4.2.6は単精度で16WAY，図4.2.8は倍精度で16WAYを示している。以下それら4つの図の実行速度曲線について説明する。図4.2.5の図においては，

i) V−V型V−Reg−0，1，2の3本の実行速度曲線は重なっている。またV−M・R型V−Reg−0，1，2の3本の実行速度曲線も重なっている。

表4.2.2

プログラム形式	230−75	FACOM−AP	
		ベクトル演算命令なし	ベクトル演算命令使用
FASP形式	FL FM FST IJXLE	L L MRS ST ADI BRT	VML
総実行速度	13.2T_u	19T_u	1T_u

ii) V－V型命令の方がV-M・R型命令よりも実行速度は速い。そして同じ型の命令においてはV-Reg上にオペランドが全部有る方が実行速度は速い。

iii) Nが非常に大きくなると，命令の続き方やV-Regには依存することなく，理想的条件下での実行速度に近づいていく。

図4.2.7は倍精度の場合であり，図4.2.5と同じ性質の実行速度曲線である。これはVADの場合と同様に説明されるので省略する。

図4.2.6の単精度でインタリーブ数が16WAYの場合においては

iv) V－V型V-Reg－1，2の実行速度曲線は重なっており，V-M・R型V-Reg－1，2の曲線も重なっている。

v) Nがある程度大きくなると命令の続き方よりも，V-Reg上のオペランド数に依存してくる。図では，V－V型V-Reg－0の曲線とV-M・R型V-Reg－1，2，3の曲線の交差を指す。

vi) Nが非常に大きくなるとV-Reg上のオペランドの有無により実行速度の収束値が異なる。

図4.2.8は倍精度の場合であり，図4.2.6と同じ性質の実行速度曲線であり，これもVADの場合と同様に説明されるので省略する。

以上がVMLの実行速度曲線の特徴である。このVMLの演算においては，インタリーブ数が32WAYの場合はii）で示した様にNが小さいと命令の続き方やV-Reg上のオペランド数に注意する必要がある。Nが非常に大きい場合は，iii）に示した様にそれらは同じ実行速度に収束する。しかしインタリーブ数が16WAYの場合は，4.1で述べた様に，V-Regを上手に使用しないとv)，vi）に示した様にNが大きくなっても実行速度に差があり，その収束値の異なることに注意しなければならない。

4.2.3 VDV (Vector Divide)

VDVの実行速度図は，図4.2.9～図4.2.12に示してある。

FORTRANHでコーディングすると

```
      DO   10   I＝1，N
      A(I)＝B(I)／C(I)
10    CONTINUE
```

AP-FORTRANでコーディングすると

```
A(*)＝B(*)／C(*)
```

表4.2.3のVDVの総実行速度をVADの場合と同様に比較すると，FACOM-APは現在の230－75に比べて2.47倍の速度である。VDVの図4.2.9は単精度でインタリーブ数が32WAYの場合，図4.2.11は倍精度で32WAY，図4.2.10は単精度で16WAY，図4.2.12は倍精度で16WAYの場合である。それらの図中で使用している記号はVADの場合と同様である。VDVの4つの図の実行速度曲線は，精度やインタリーブ数に関係しないで同じ性質をしているので，図4.2.9を代表させて説明する。

i) V－V型V-Reg－0，1，2の3本の実行速度曲線は重なっている。またV-M・R型V-Reg－0，1，2の3本の曲線も重なっている。

ii) Nがある程度大きいところまでは，V－V型命令の方がV-M・R型命令よりも実行速度は速い。そして同じ型の命令の続き方においては，V-Reg上にオペランドが全部有る方が実行速度は速い。

iii) Nが非常に大きくなると，命令の続き方やV-Reg上のオペランド数に依存することなく，それらの収束値は一致する。

以上がVDVの実行速度曲線の特徴である。このVDVの演算においては，表3.2.1.6に示してある。VDVの実行速度を他のV型命令と比較してもわかる様に，非常

表4.2.3

プログラム形式	230－75	FACOM－AP	
		ベクトル演算命令なし	ベクトル演算命令使用
FASP形式	FL FD FST IJXLE	L L DRS ST ADI BRT	VDV
総実行速度	$22.2T_{\mu}$	$33T_{\mu}$	$9T_{\mu}$

に遅い。このことは，V型命令を実行する際に必要とするデータ供給能力に関して影響を及ぼすことがなくなる。したがってVDVにおける実行速度は，Nが大きくなっても iii) に示してある様に，V-Regの使用には依存しないが，このVDVそのものの実行速度が遅いため，プログラミングの際は，特別に必要の無い限り，この命令の使用は避けた方が望ましい。とくに（ベクトル）÷（スカラ）はこのVDVと同じになるので，1／（スカラ）＊（ベクトル）とすることは絶対に必要である。

4.2.4　VSM (Vector Sum)

VSMの実行速度図は，図4.2.13～図4.2.16に示してある。

FORTRANHでコーディングすると

```
      C＝0.0
      DO  10  I＝1,N
      C＝C＋A(I)
10    CONTINUE
```

AP-FORTRANでコーディングすると

```
      C＝VSUM(A(＊))
```

表4.2.4のVSMの総実行速度をVADの場合と同様に比較すると，FACOM-APは現在の230-75に比べて12倍の速度である。VSMの図において，図4.2.13は単精度でインタリーブ数が32WAYの場合，図4.2.15は倍精度で32WAY，図4.2.14は単精度で16WAY，図4.2.16は倍精度で16WAYの場合を示している。これら4つの図は同じ性質の実行速度曲線であるので，図4.2.13を代表させて説明する。

i) Nがある程度大きいところまでは，V－V型命令の方がV-M・R型命令の方が実行速度は速い。そして同じ型の命令の続き方では，V-Reg上にオペランドが有る場合の方が実行速度は速い。

ii) Nが非常に大きくなると，命令の続き方や，V-Reg上のオペランドの有無に依存することなくそれらの実行速度は，理想的条件下での実行速度に近づいていく。

以上がVSMの実行速度曲線の特徴である。但しこのVSMの演算においては，同インタリーブ数では，Nが非常に小さい時は，単精度よりも倍精度の方が実行速度は速い。しかしNが大きくなるとその値は，表3.2.1.bのVSMのV型命令の単精度，倍精度による実行速度に収束し，単精度の方が速くなる。このことは式（4.1.4）より説明される。このVSM命令の実行速度は，インタリーブ数には依存しない。

4.2.5　VNM (Vector Norm)

VNMの実行速度図は，図4.2.17～図4.2.20に示してある。FORTRANHでコーディングすると

```
      C＝0.0
      DO  10  I＝1,N
```

表4.2.4

プログラム形式	230－75	FACOM-AP	
		ベクトル演算命令なし	ベクトル演算命令使用
FASP形式	FA IJXLE	L ARS ADI BRT	VSM
総実行速度	$6T_u$	$14T_u$	$0.5T_u$

表4.2.5

プログラム形式	230－75	FACOM-AP	
		ベクトル演算命令なし	ベクトル演算命令使用
FASP形式	FL FMR FAWR IJXLE	L MRS ARS ADI BRT	VNM
総実行速度	$13.2T_u$	$23T_u$	$1T_u$

```
      C=C+A(I)** 2
   10 CONTINUE
```

AP-FORTRANでコーディングすると

```
      C=VNRM(A(*))
```

表4.2.5のVNMの総実行速度を，4.2.1のVADの場合と同様に比較をすると，FACOM-APは現在の230-75に比べて13.2倍の速度である。このVNMの4つの図において，図4.2.17は単精度でインタリーブ数が32WAYの場合，図4.2.19は倍精度で32WAY，図4.2.18は単精度で16WAY，図4.2.20は倍精度で16WAYの場合を示している。このVNMは，4.2.4のVSMの実行速度曲線と同じ性質であるので説明は省略する。但し，表3.2.1.bより，VNMとVSMの実行速度が異なっている。実行速度曲線もVNMの方がVSMよりも実行速度は遅くなっているがこれは4.1で示してある式(4.1.4)より説明される。

4.2.6 FAX (Find Address MAX)

FAXの実行速度図は，図4.2.21～図4.2.24に示してある。

FORTRANHでコーディングすると，

```
      C=A(1)
      DO  10  I=2,N
      B=A(I)
      IF(B.GT.C) C=B
   10 CONTINUE
```

AP-FORTRANでコーディングすると，

```
      C=FMXV(A(*))
```

表4.2.6のFAXの230-75とFACOM-APでベクトル演算を使用しない場合の総実行速度の欄に成立と不成立の2つに分けて示してあるが，これはプログラム実行の際に最大値がIF文によって変化する場合を成立とし，変化しない場合を不成立として，その総実行速度を算出したものである。従って成立した場合の総実行速度を，4.2.1のVADの場合と同様に比較すると，FACOM-APでFAXの演算を行うと現在の230-75に比べて成立の場合5.1倍不成立の場合4.2倍の速度である。

次にFAXの図について説明する。図4.2.21は単精度でインタリーブ数が32WAYの場合，図4.2.23は倍精度で32WAY，図4.2.22は単精度で16WAY，図4.2.24は倍精度で16WAYを示している。FAXの4つの図の実行速度曲線は，4.2.4のVSMと同じ性質の実行速度曲線であるので，説明は省略する。但し，実行速度曲線はFAXの方がVSMより遅くなっている。これは表3.2.1.bより4.1で示した式(4.1.4)より説明される。また，このFAXは，インタリーブ数が32WAY，16WAY供に，Nが小さい時は，単精度よりも倍精度の方が実行速度は速い。

4.2.7 FRE (Find Relation Equal)

FREの実行速度図は図4.2.25～図4.2.28に示してある。

FORTRANHでコーディングすると

```
      C=0.0
      DO  10  I=1,N
      G=A(I)
      H=B(I)
      IF(G.EQ.H) GO TO 20
   10 CONTINUE
      GO TO 30
   20 C=G
   30 CONTINUE
```

AP-FORTRANでコーディングすると

```
      C=FEQV(A(*),B(*))
```

表4.2.6

プログラム形式	230-75	FACOM-AP	
		ベクトル演算命令なし	ベクトル演算命令使用
FASP形式	FL FCWR JC FLWR NULL IJXLE	L SRS BRC TR NULL ADI BRC	FAX
総実行速度	成立 10.2T_u 不成立 8.4T_u	成立 18T_u 不成立 17T_u	2T_u

表4.2.7

プログラム形式	230−75	FACOM−AP	
		ベクトル演算命令なし	ベクトル演算命令使用
FASP形式	FL FL FCWR JC NULL IJXLE	L L SRS BRC NULL ADI SR BRC	FRE
総実行速度	成 立 9.4T_u 不成立 8.6T_u	成 立 19T_u 不成立 16T_u	2T_u

表4.2.7の230−75とFACOM−APでベクトル演算命令を使用しない場合の総実行速度の欄に，成立と不成立の2つに分けて示してあるが，4.2.6のFAXの場合と同様である。成立の場合の総実行速度をVADの場合と同じ条件で比較すると，FACOM−APは現在の230−75に比べて成立の場合，4.7倍，不成立の場合4.3倍の速度である。次にFREの図について説明する。図4.2.25は単精度でインタリーブ数が32WAYの場合，図4.2.27は倍精度で32WAY，図4.2.26は単精度で16WAY，図4.2.28は倍精度で16WAYの場合をそれぞれ示している。FREの4つの図はすべて同じ実行速度曲線であるので図4.2.25を代表させて説明する。

i) V−V型V−Reg−0，1の実行速度曲線は重なっており，V−M.R型V−Reg−0，1の曲線も重なっている。

ii) Nがある程度大きいところまでは，V−V型命令の方がV−M.R型命令よりも実行速度は速い。そして同じ型の命令の続き方の場合は，V−Reg上にオペランドが全部有る方が実行速度は速い。

iii) Nが非常に大きくなると，命令の続き方やV−Reg上のオペランド数に依存することなく，それらの実行速度は，理想的条件下での実行速度に近づいていく。

以上がFREの実行速度曲線の特徴である。このFREの4つの図が同じであるということは，表3.2.1.bよりFREの実行速度より，単精度，倍精度は供に同じ速度であり，その実行速度が比較的遅く，データ供給能力が満足される格好となり，インタリーブ数の関係も式(4.1.4)より説明され，それら4つの図は皆同じ実行速度曲線となる。このFREの演算においても，Nが大きい場合はiii)に示した様になるが，Nが小さい場合はii)に示した様に，命令の続き方や，V−Reg上のオペランド数に注意する必要がある。

4.2.8 IPD(Inner Prduct)

IPDの実行速度図は図4.2.29〜図4.2.32に示してある。

表4.2.8

プログラム形式	230−75	FACOM−AP	
		ベクトル演算命令なし	ベクトル演算命令使用
FASP形式	FL FM FAWR IJXLE	L L MRS ARS ADI BRT	IPD
総実行速度	13.2T_u	25T_u	1T_u

FORTRANHでコーディングする場合

```
      C=0.0
      DO  10  I=1,N
      C=C+A(I)*B(I)
   10 CONTINUE
```

AP-FORTRANでコーディングする場合

```
      C=AIPD(A(*),B(*))
```

表4.2.8のIPDの総実行速度を4.2.1のVADの場合と同様に比較すると，FACOM-APは，現在の230-75に比べて13.2倍の速度である。IPDの4つの図は，図4.2.29は単精度でインタリーブ数が32WAYの場合，図4.2.31は倍精度で32WAY，図4.2.30は単精度で16WAY，図4.2.31は倍精度で16WAYの場合を示している。このIPDの実行速度曲線は4.2.7のFREの場合と同じ性質の実行速度曲線であるので説明は省略する。但し，IPD命令は，単精度と倍精度の実行速度が異なっている。したがって，実行速度曲線も単精度と倍精度では実行速度が異なっている。これは表3.2.1.bより，4.1で示した式（4.1.4）より説明される。

4.2.9 VMV（Vector Move）

VMVの実行速度図は，図4.2.33～図4.2.36に示してある。

FORTRANHでコーディングすると

```
      DO  10  I=1,N
      A(I)=B(I)
   10 CONTINUE
```

AP-FORTRANでコーディングすると

```
      A(*)=B(*)
```

表4.2.9のVMVの総実行速度，VADの場合と同様に比較すると，FACOM-APは現在の230-75に比べて，16.4倍の速度である。図4.2.33は単精度でインタリーブ数が32WAY，図4.2.35は倍精度で32WAY，図4.2.34は単精度で16WAY，図4.2.36は倍精度で16WAYの場合を示している。まず図4.2.33においては

i) V－V型V-Reg-0と1の実行速度曲線は重なっており，V-M.R型V-Reg-0と1の曲線も重なっている。

ii) Nがある程度大きいところまでは，V－V型命令の方がV-M.R型命令よりも実行速度は速い。そして同じ型の命令の続き方においては，V-Reg上にオペランドが全部有る方が実行速度は速い。

iii) Nが非常に大きくなると，それらの実行曲線は，命令の続き方やV-Reg上のオペランド数に依存することなく，理想的条件下における実行速度に近づいていく。

図4.2.35は倍精度の場合であり，その実行速度曲線の性質は図4.2.34と同様であるので省略する。

図4.2.35においては

iv) Nが小さい時は，V－V型命令の方がV-M.R型命令よりも実行速度は速く，またV-Reg上のオペランド数に依存して，実行速度も異なる。

v) Nがある程度大きくなると，命令の続き方よりもV-Reg上のオペランド数に依存して実行速度が異なる。図ではV－V型V-Reg-0とV-M.R型V-Reg－1と2の実行速度曲線の交差を指す。

vi) Nが非常に大きくなると，命令の続き方には関係がなくなり，V-Reg上のオペランドの有無により実行速度の収束値が異なる。

図4.2.36は倍精度の場合であり，図4.2.34と同じ性質の実行速度曲線であるので説明は省略する。以上が，VMVにおける実行速度曲線の特徴である。このVMVの演算において，図4.2.33と図4.2.35は同インタリーブ数であり，実行速度曲線は同じ性質であるが，表3.2.1.bのVMVの実行速度は，式（4.1.4）より算出され，倍精度の方が単精度よりも遅くなっている。図4.2.34と図4.2.36についても同様である。インタリーブ数が32WAYの場合は，Nが非常に大きくなると，単精度，倍精度の場合の実行速度は共にiii）に示してある様になるが，16WAYの場合は，vi）に示してある様に，V-Reg上のオペランドの有無によりその実行速度の収束値が異なっ

表4.2.9

プログラム形式	230－75	FACOM-AP	
		ベクトル演算命令なし	ベクトル演算命令使用
FASP形式	FL FST IJXLE	L ST ADI BRT	VMV
総実行速度	$8.2T_u$	$8T_u$	$0.5T_u$

てくることに注意する必要がある。このことは4.1ですでに説明してある。

4.2.10 AJM (Adjacent Mean)

AJMの実行速度図は，図4.2.37～図4.2.40に示してある。

FORTRANHでコーディングすると

```
      DO   10   I＝1,NN
      A(I)=(B(I)+B(I+1))*0.5
10    CONTINUE
```

AP-FORTRANでコーディングすると

```
      A(*)=AJM(B(*))
```

表4.2.10のAJMの総実行速度を，VADの場合と同様に比較すると，FACOM-APで演算すると，現在の230-75に比べて，24.3倍の速度である。しかしながらこの命令は単精度の場合，同一メモリバンクへ連続的にアクセスがかかるので，メモリコンフリクトが生ずる可能性があり，これがどの程度，図3.1.2のデータバッファの存在と，フォーワーディングの技術で防止できるかは実測してみる必要がある。但し，このことはソースオペランドがV-Reg上にあるときには問題にならない。次に図について説明すると，図4.2.37は単精度でインタリーブ数が32WAYの場合，図4.2.39は倍精度で32WAY図4.2.38は単精度で16WAY，図4.2.40は倍精度で16WAYの場合を示している。この4つの図は，表3.2.1.bを参照すると，4.2.9のVMVと同じ実行速度であり，ベクトル演算に必要なベクトルオペランド数も同じであるので，実行速度曲線についての説明は省略する。

4.2.11 CME (Compare Equal)

CMEの実行速度図は，図4.2.41～図4.2.44に示してある。CMEをAP-FORTRANでコーディングすると次の様になる。

```
      C(*)＝A(*).EQ.B(*)*
```

このCMEの場合は，現在の230-75のFORTRANHでコーディングすると，機能上の差があり，これまで述べてきた様な比較ができない，機能上の相違とは，FACOM-APの場合は，CMEの演算においてはその対応がビット単位であるが，230-75の場合は語単位となる。したがって230-75でFACOM-APの機能のごとく実行させる場合は，コーディングに無駄が生じ，比較する対象にはならなくなる。故にこのCMEにおいては，実行速度図の表示だけにする。そしてその実行速度については，表3.2.1.bに示してある。

表4.2.10

プログラム形式	230-75	FACOM-AP	
		ベクトル演算命令なし	ベクトル演算命令使用
FASP形式	FL FA FMR FST IJXLE	L L ADR MRS ST ADI BRT	AJM
総実行速度	$18.2T_u$	$27T_u$	$0.75T_u$

＊ AベクトルとBベクトルの対応する要素が等しいとき，Cビットストリングの対応するビットをオンにする。詳しくは文献9)を参照のこと。

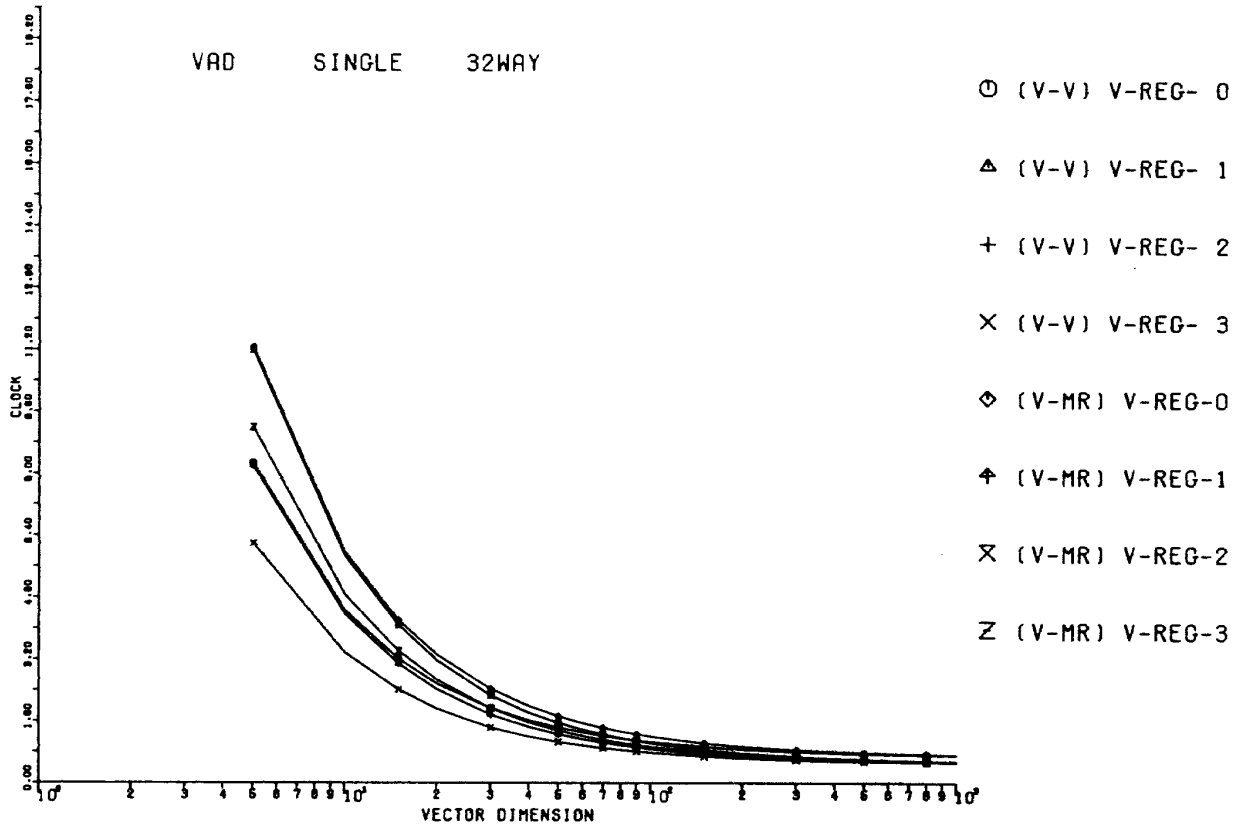


図 4.2.1

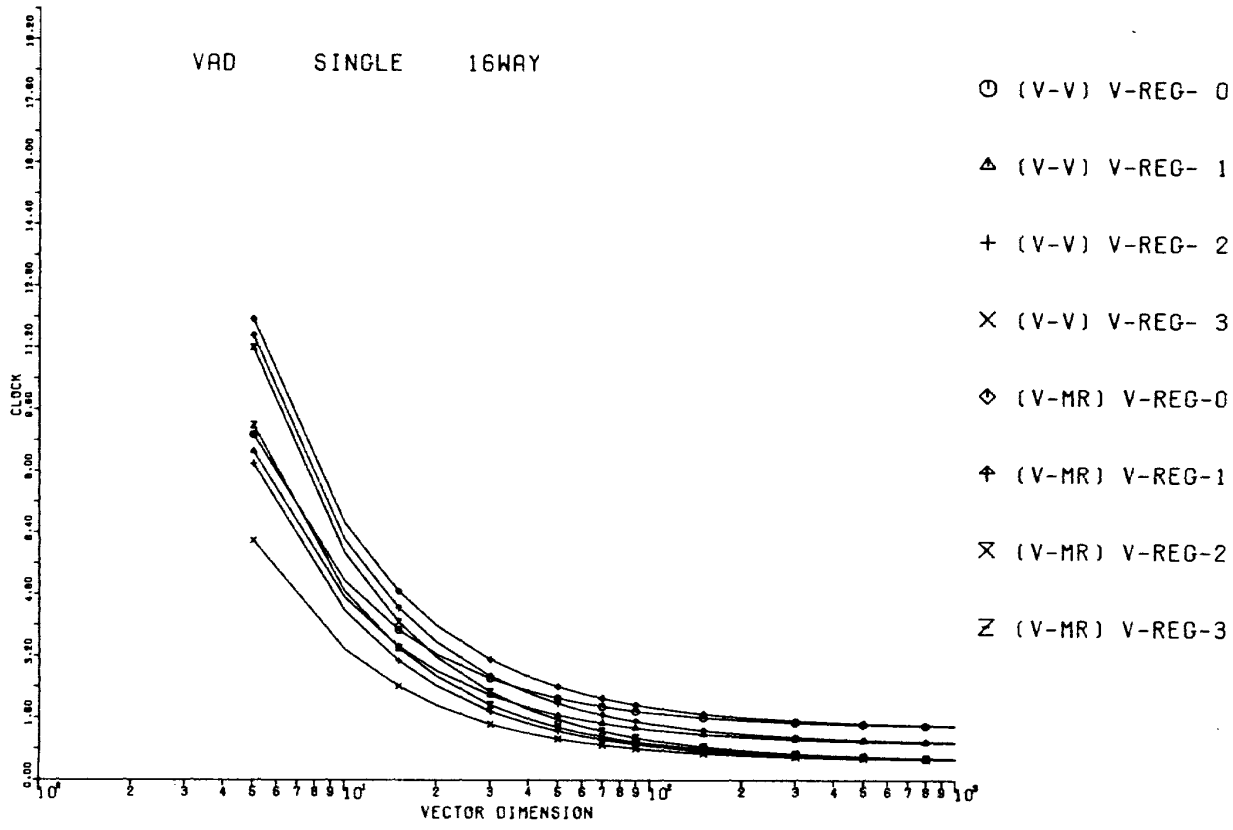


図 4.2.2

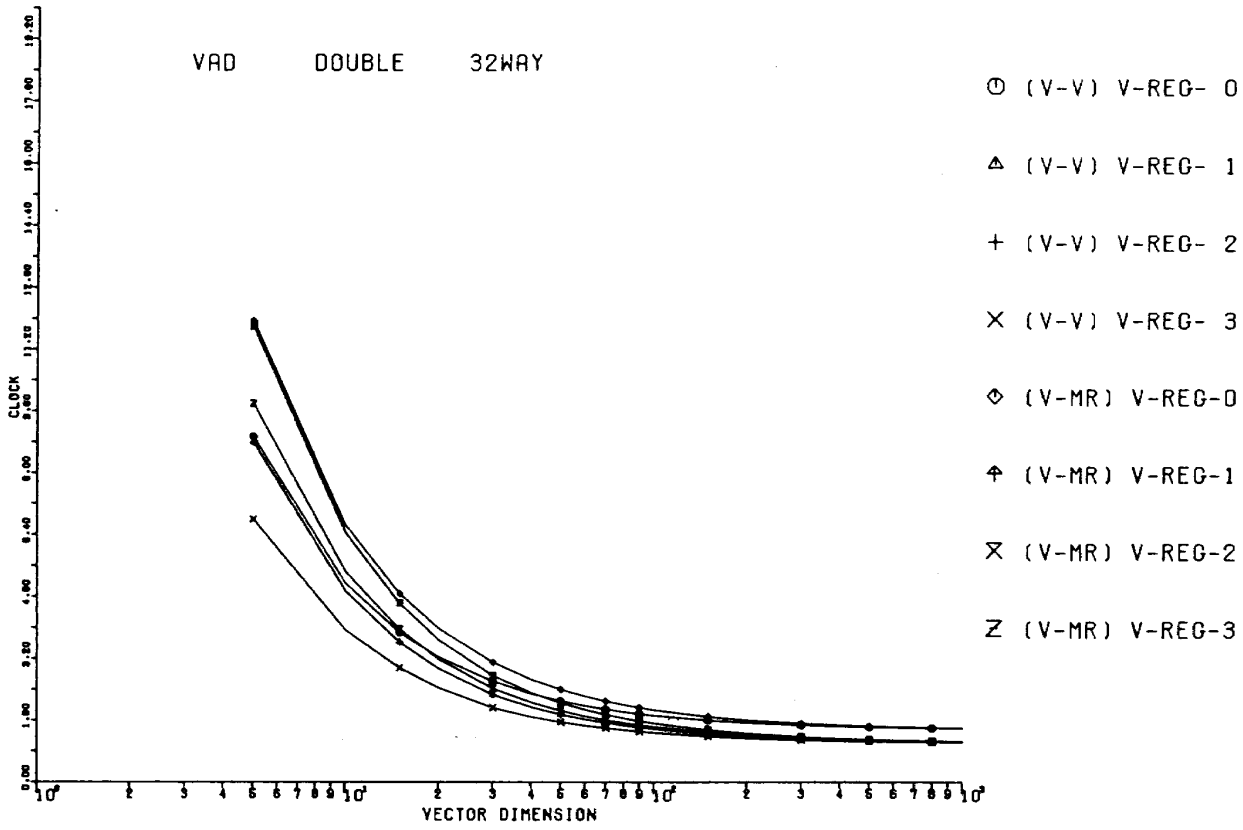


図4.2.3

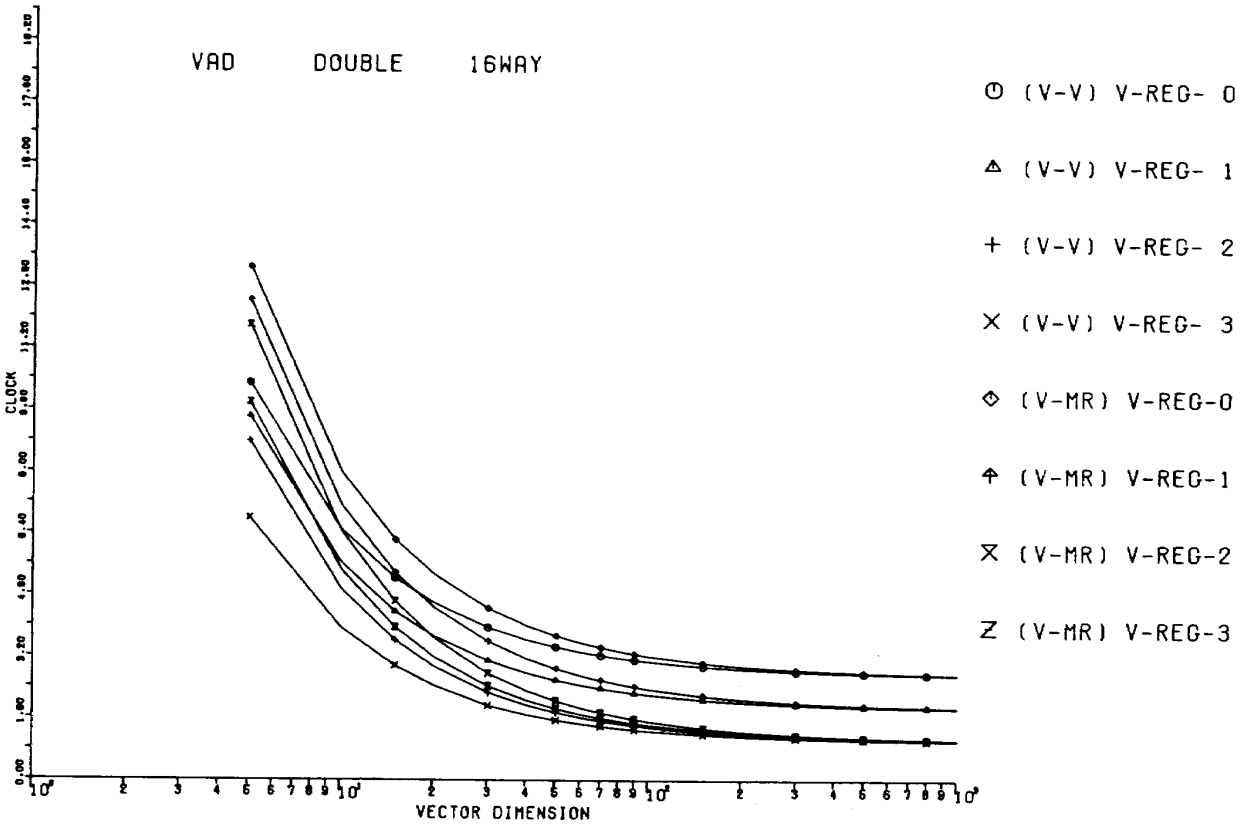


図4.2.4

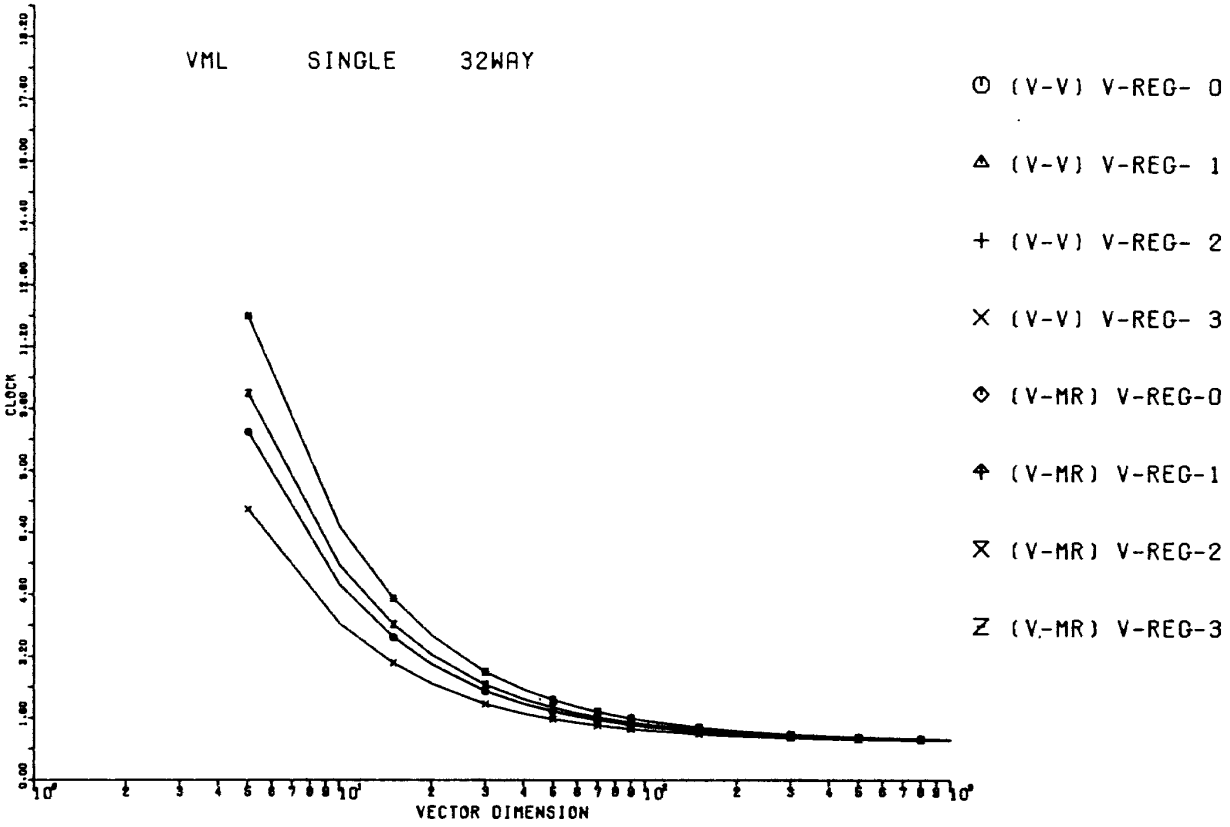


図 4.2.5

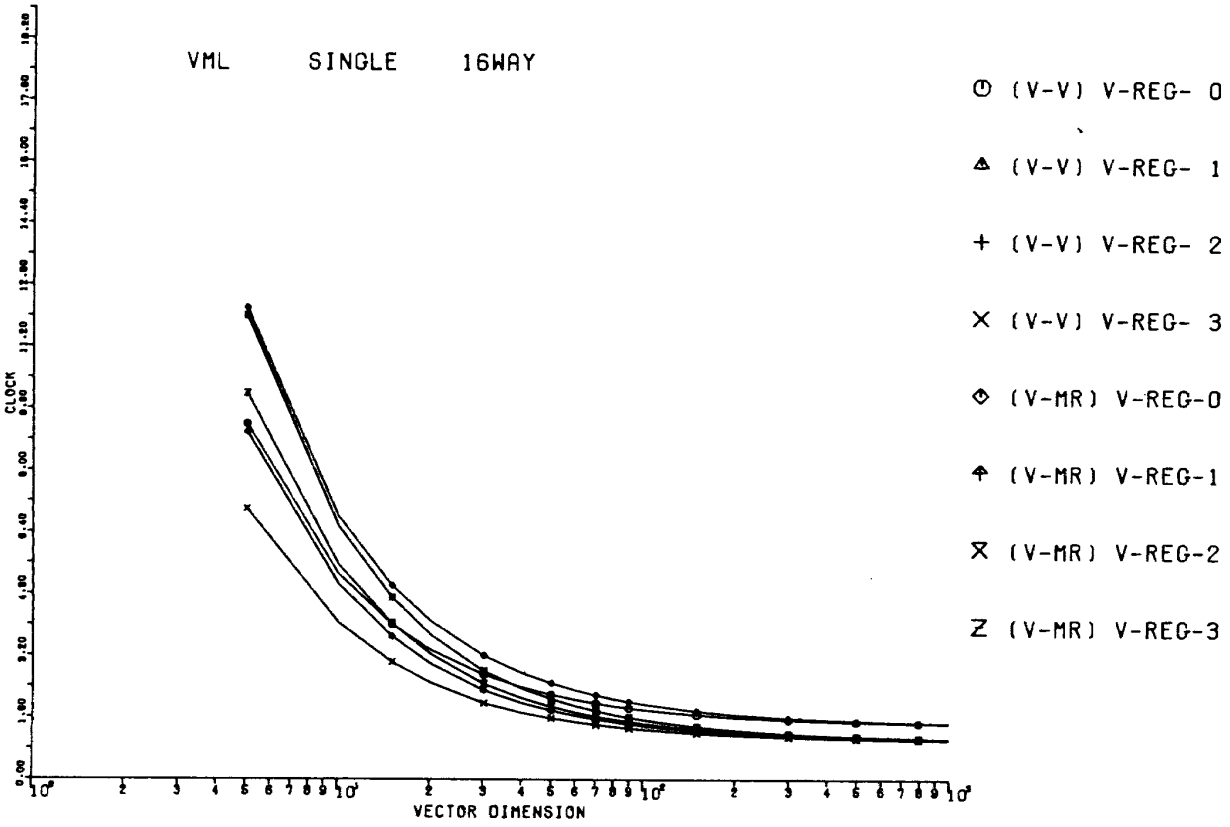


図 4.2.6

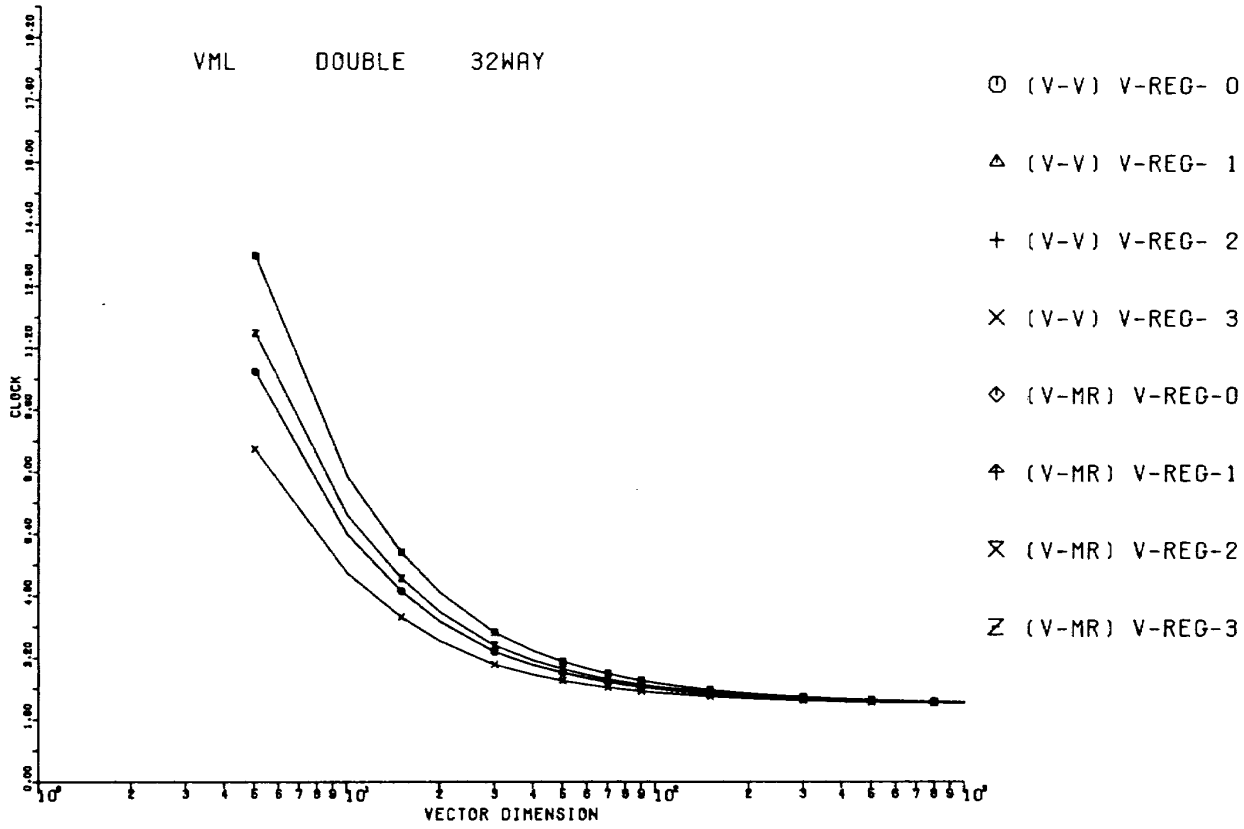


図 4.2.7

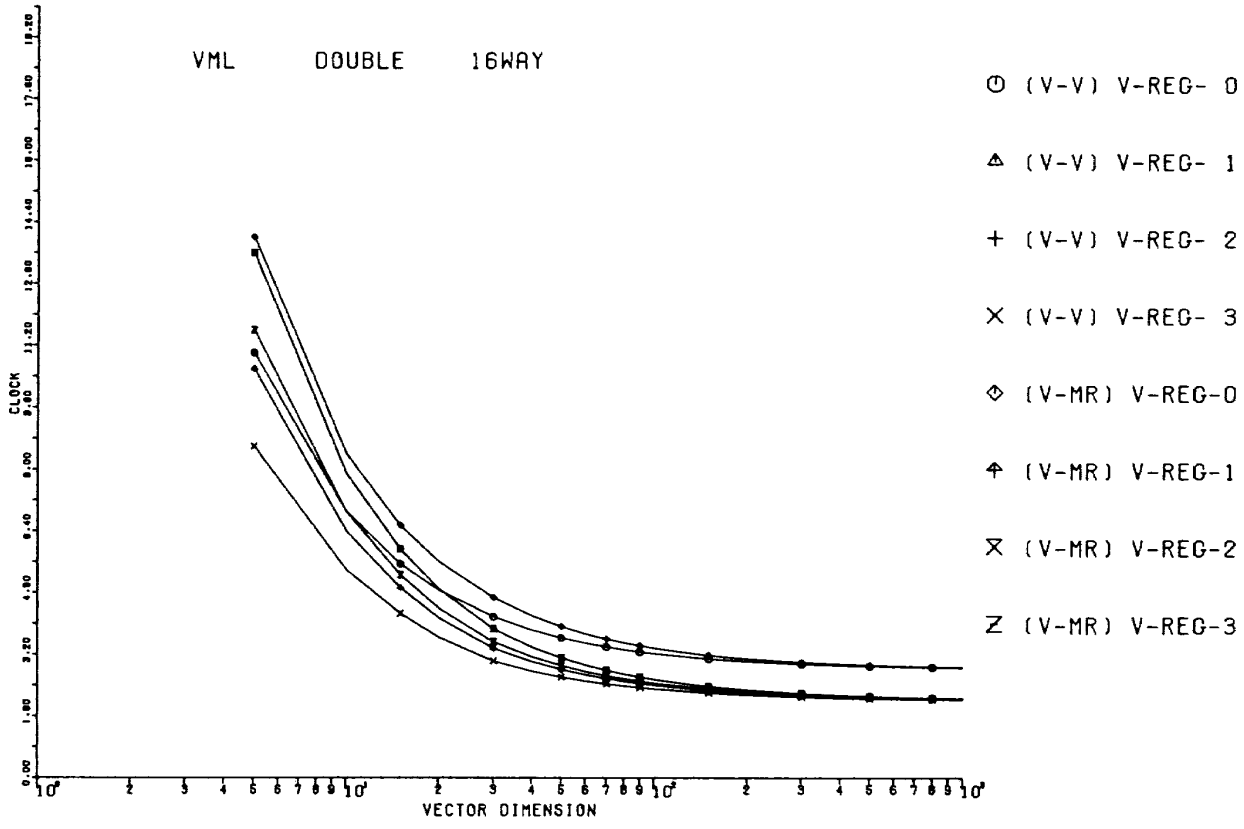


図 4.2.8

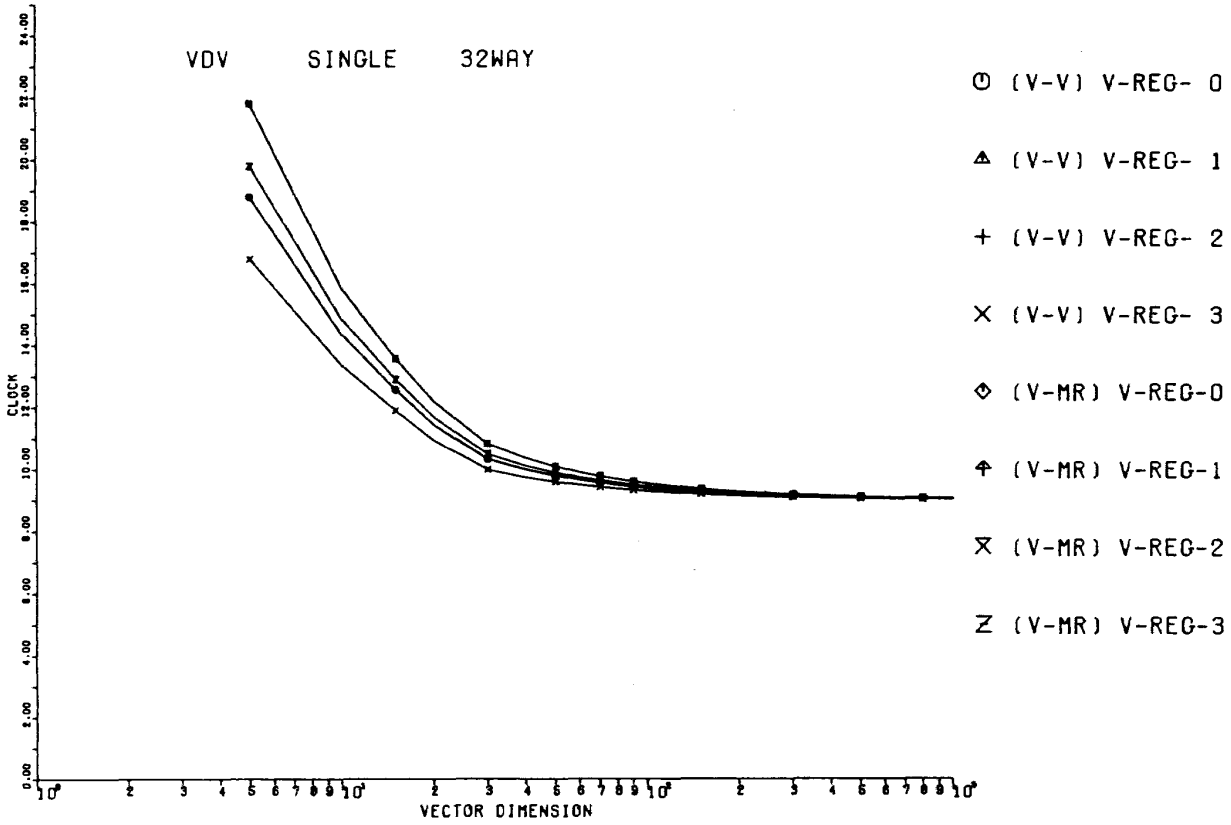


図 4.2.9

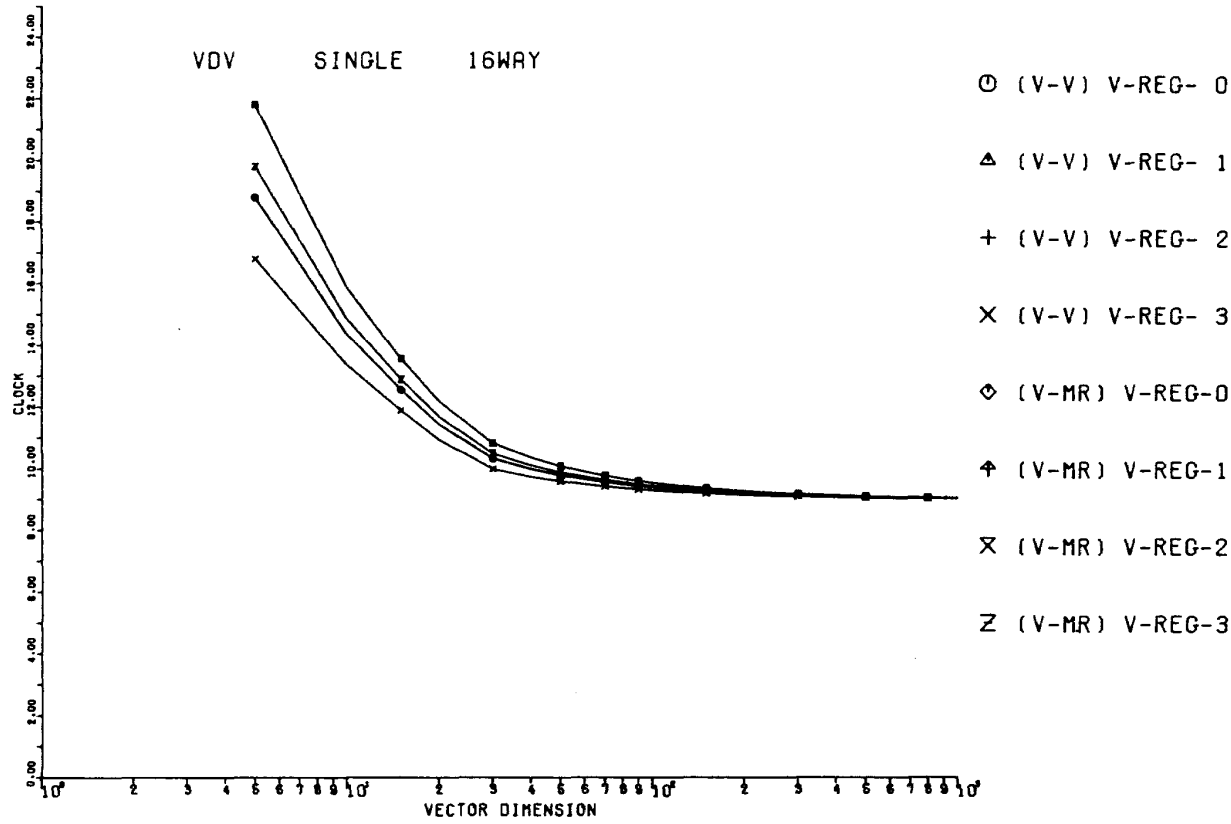


図 4.2.10

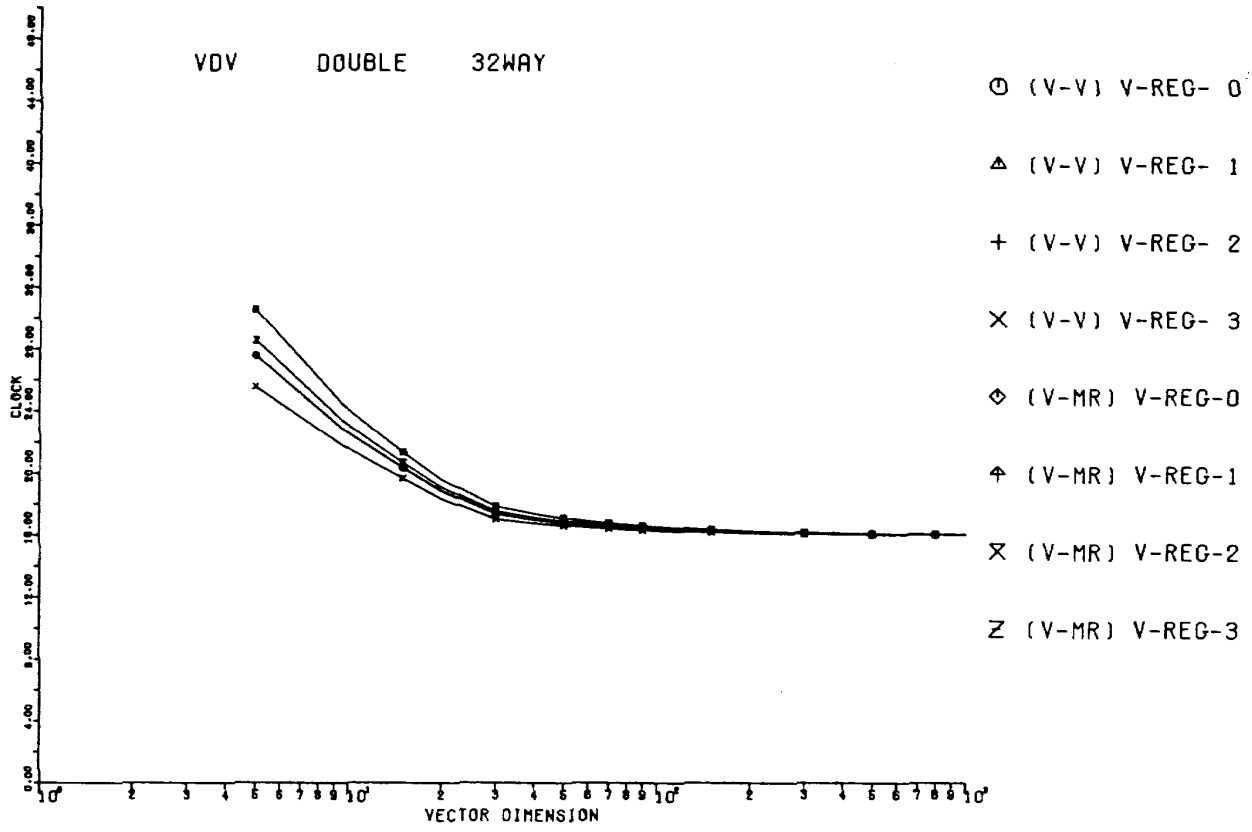


図 4.2.11

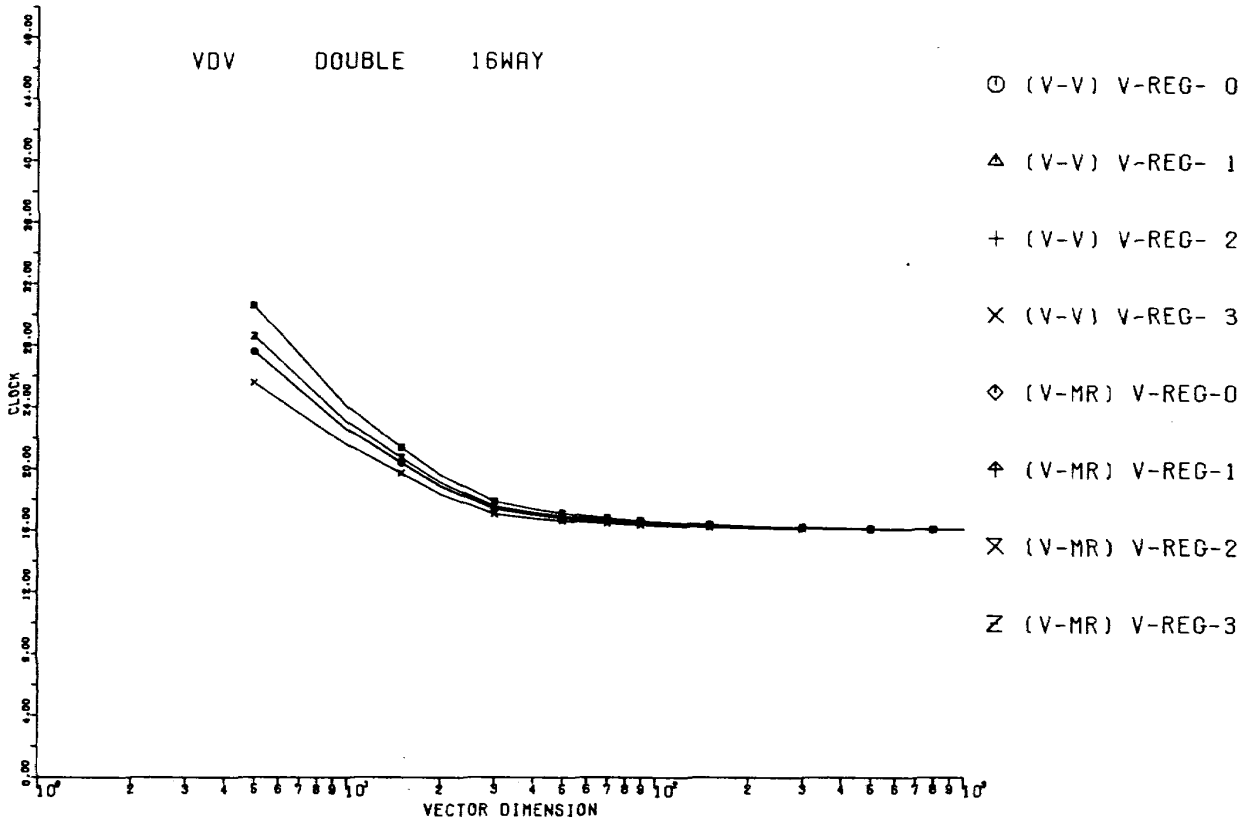


図 4.2.12

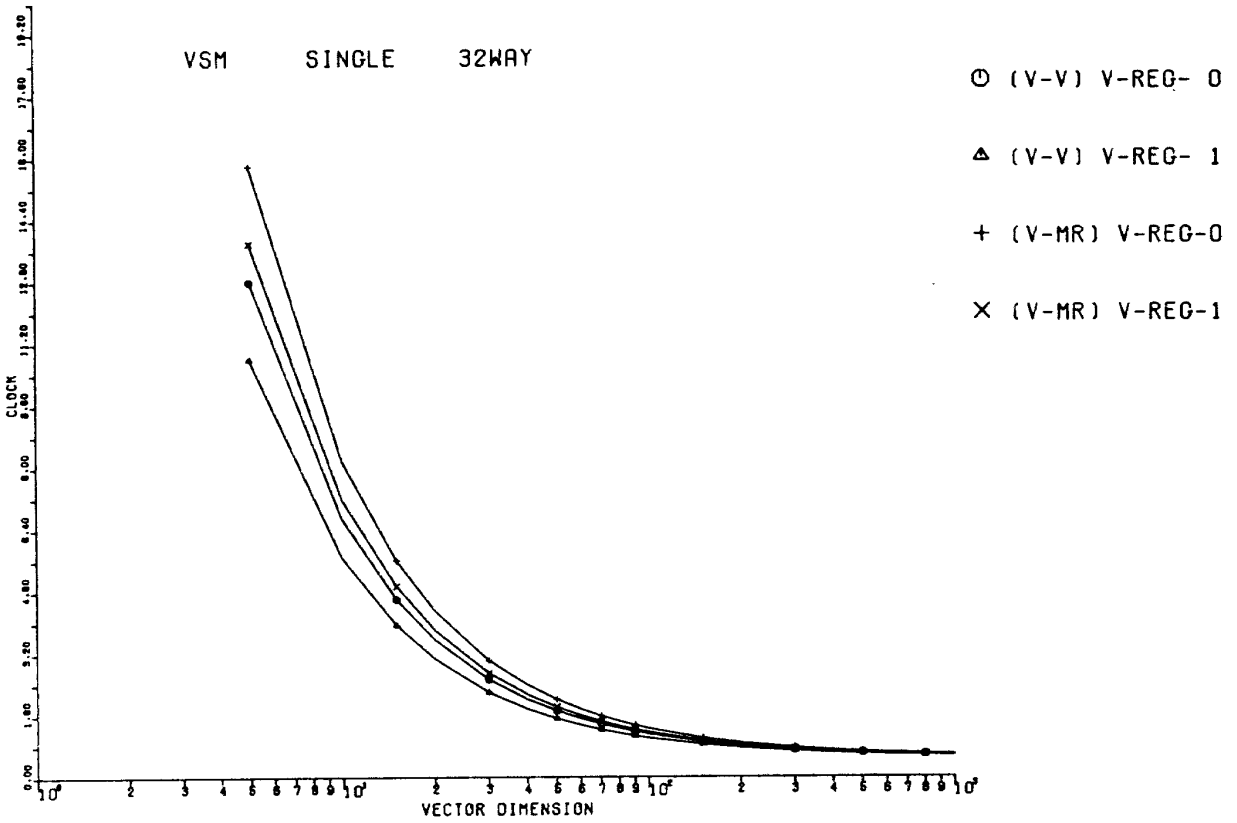


図 4.2.13

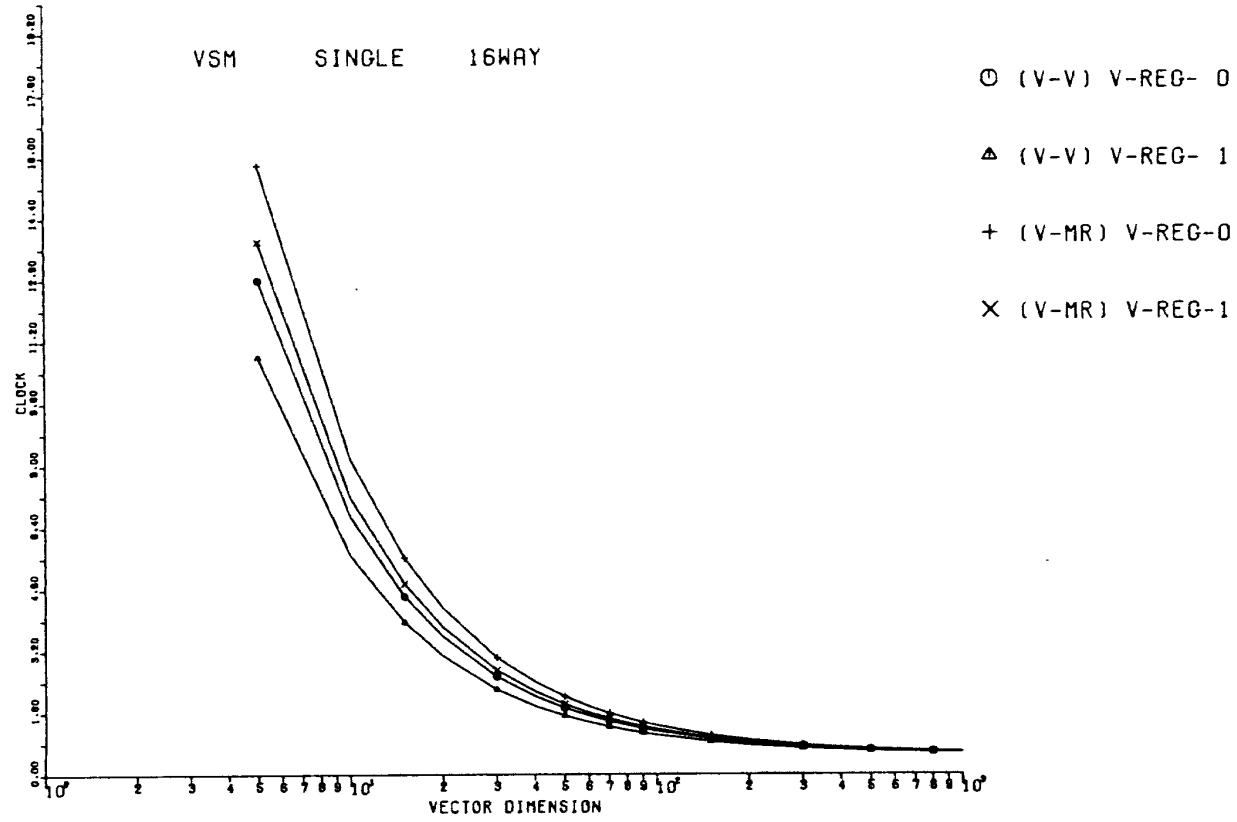


図 4.2.14

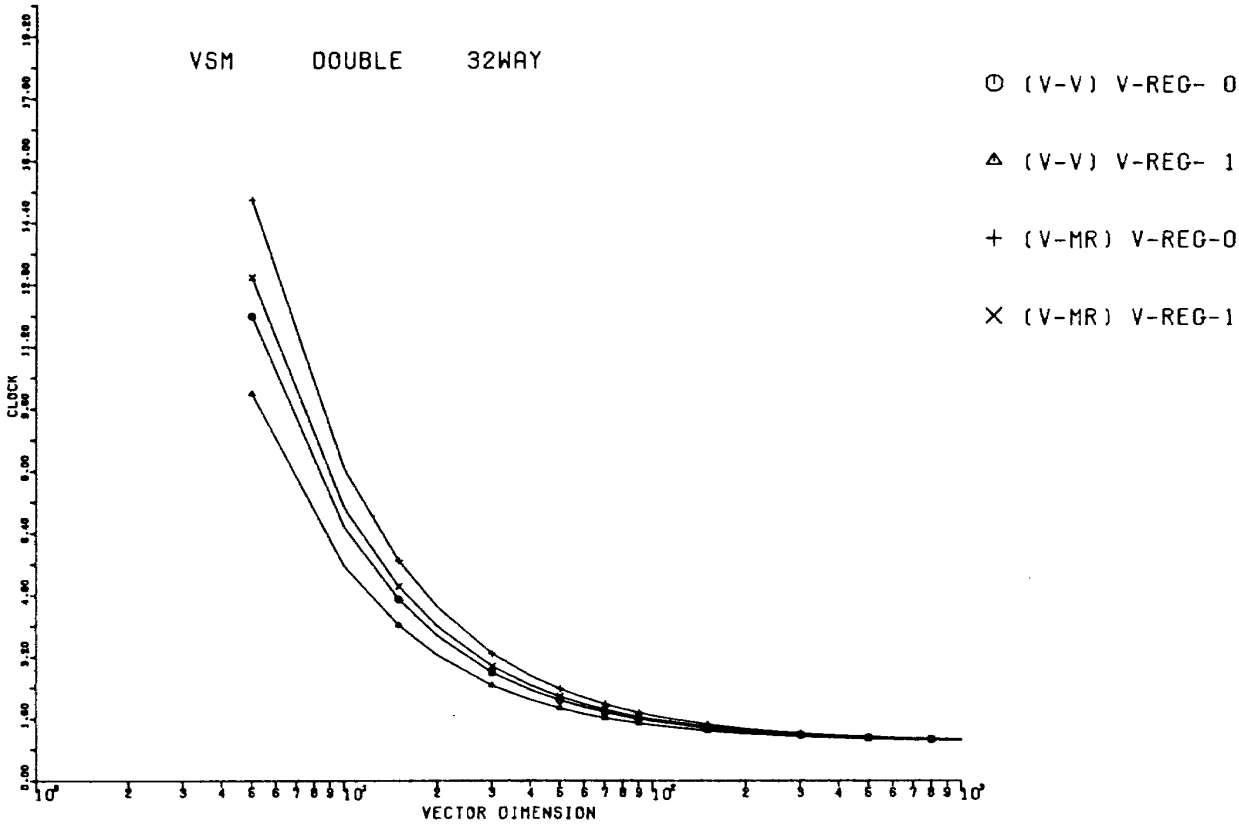


図 4.2.15

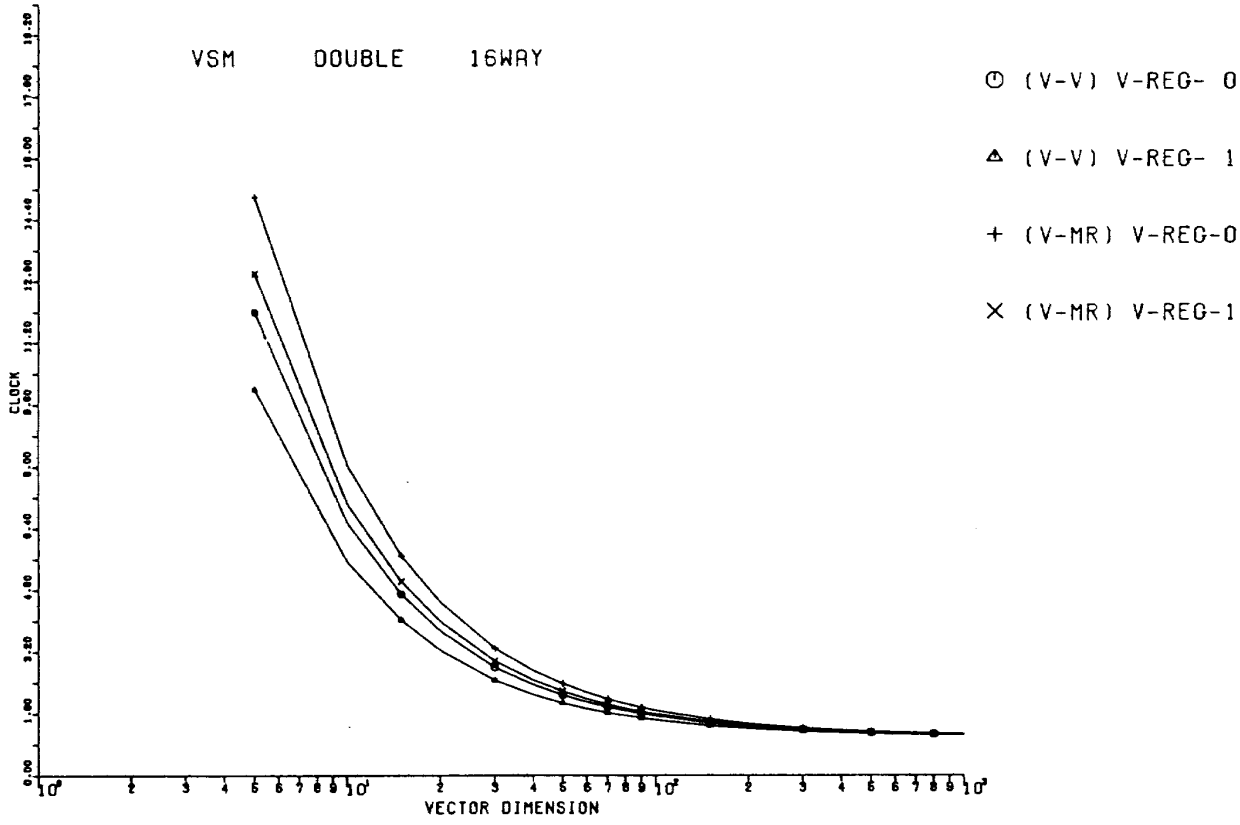


図 4.2.16

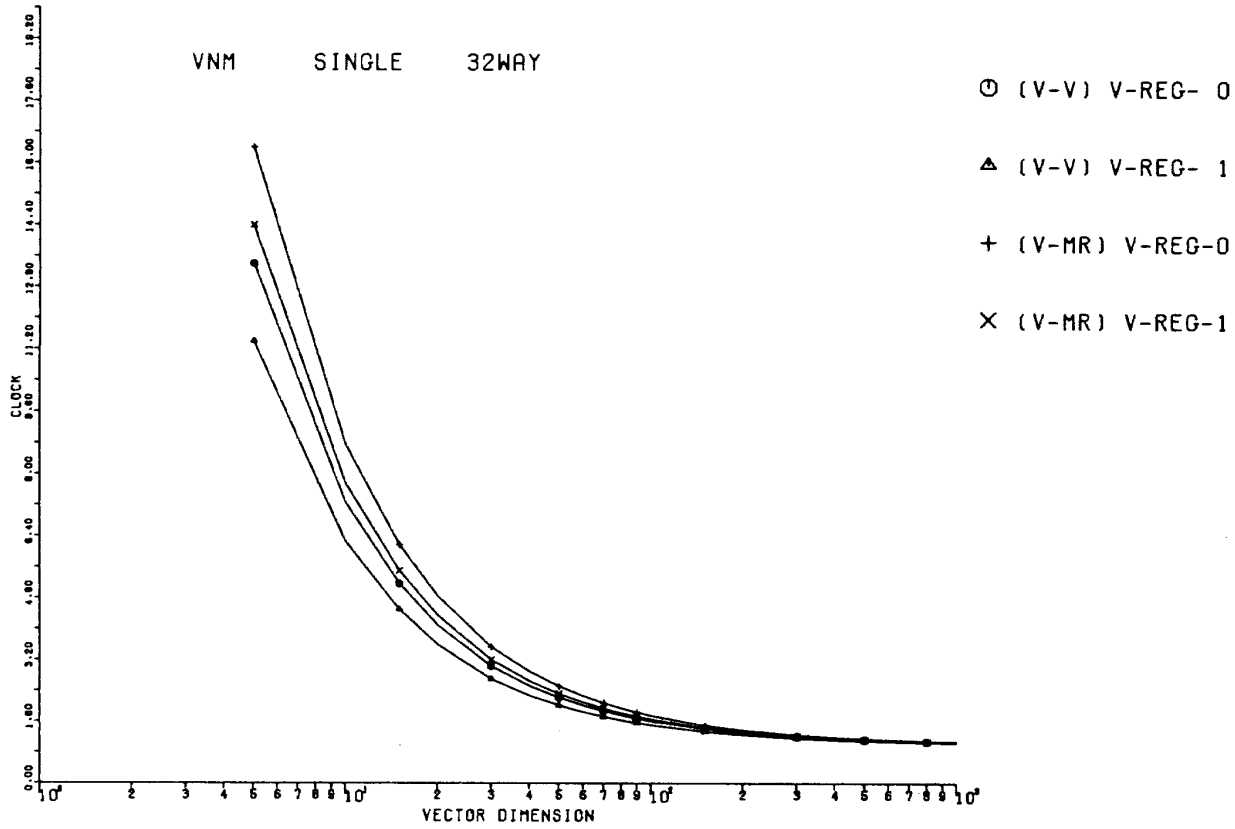


図 4.2.17

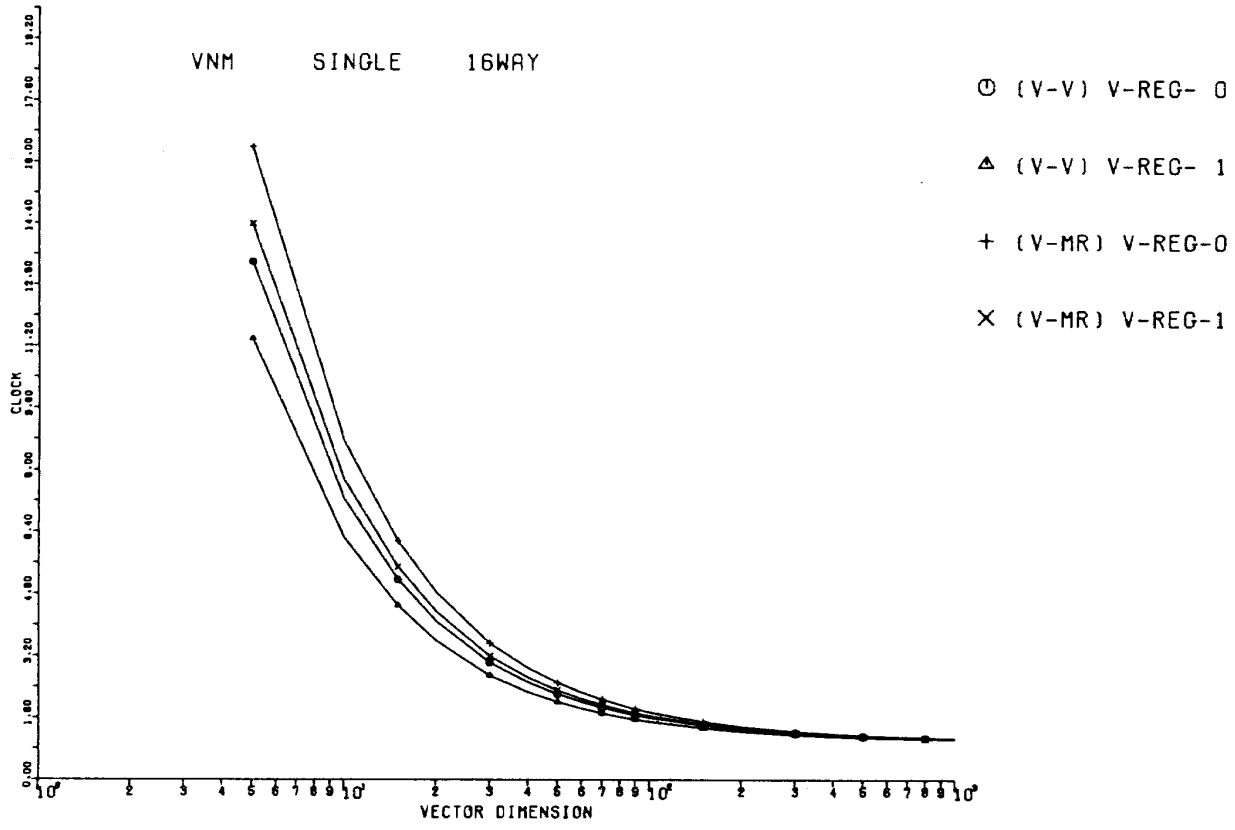


図 4.2.18

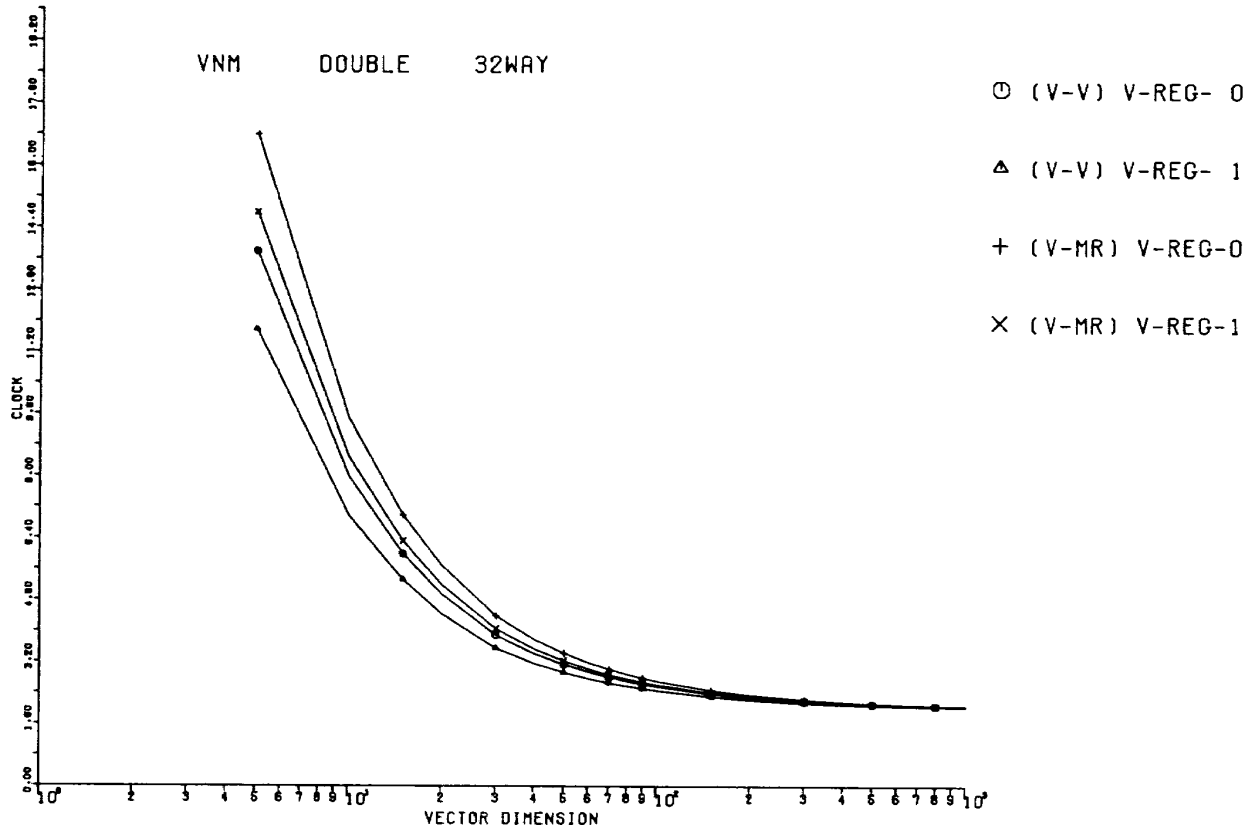


図 4.2.19

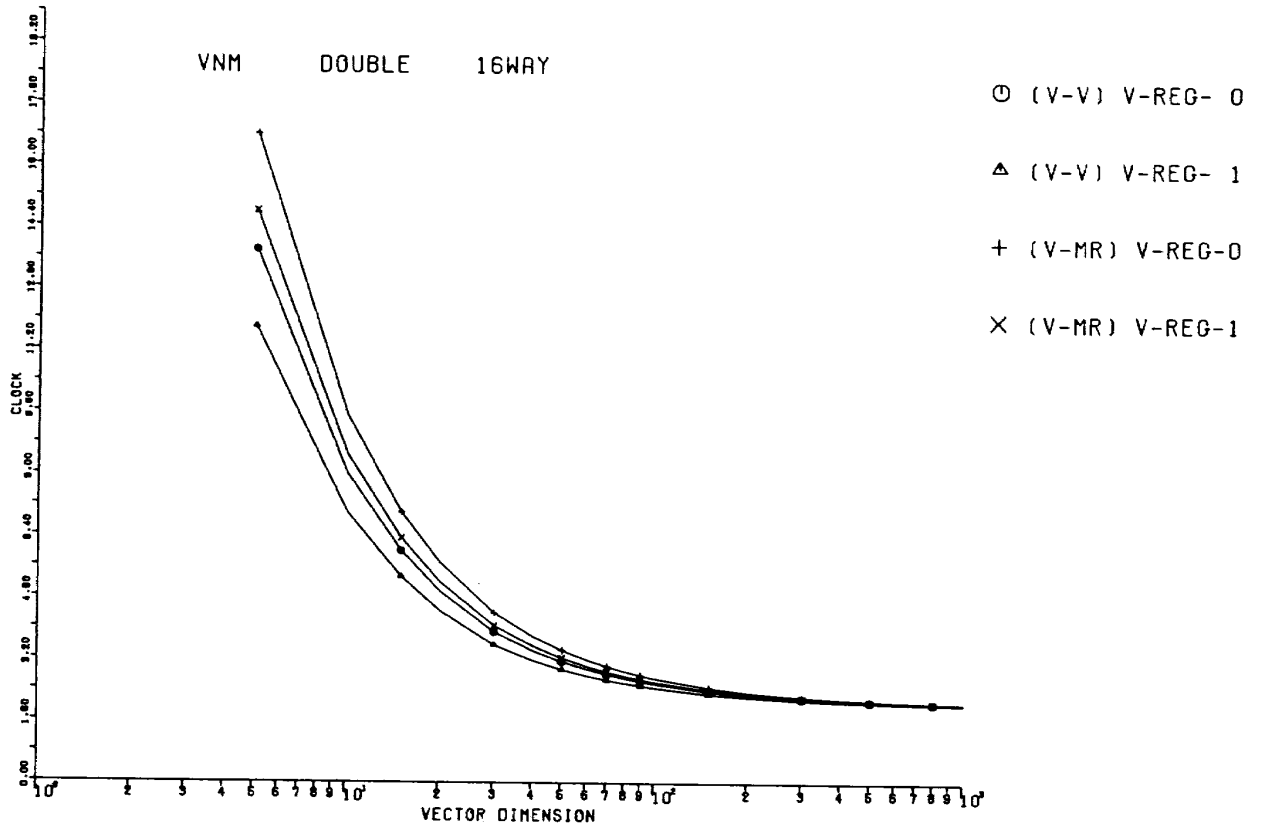


図 4.2.20

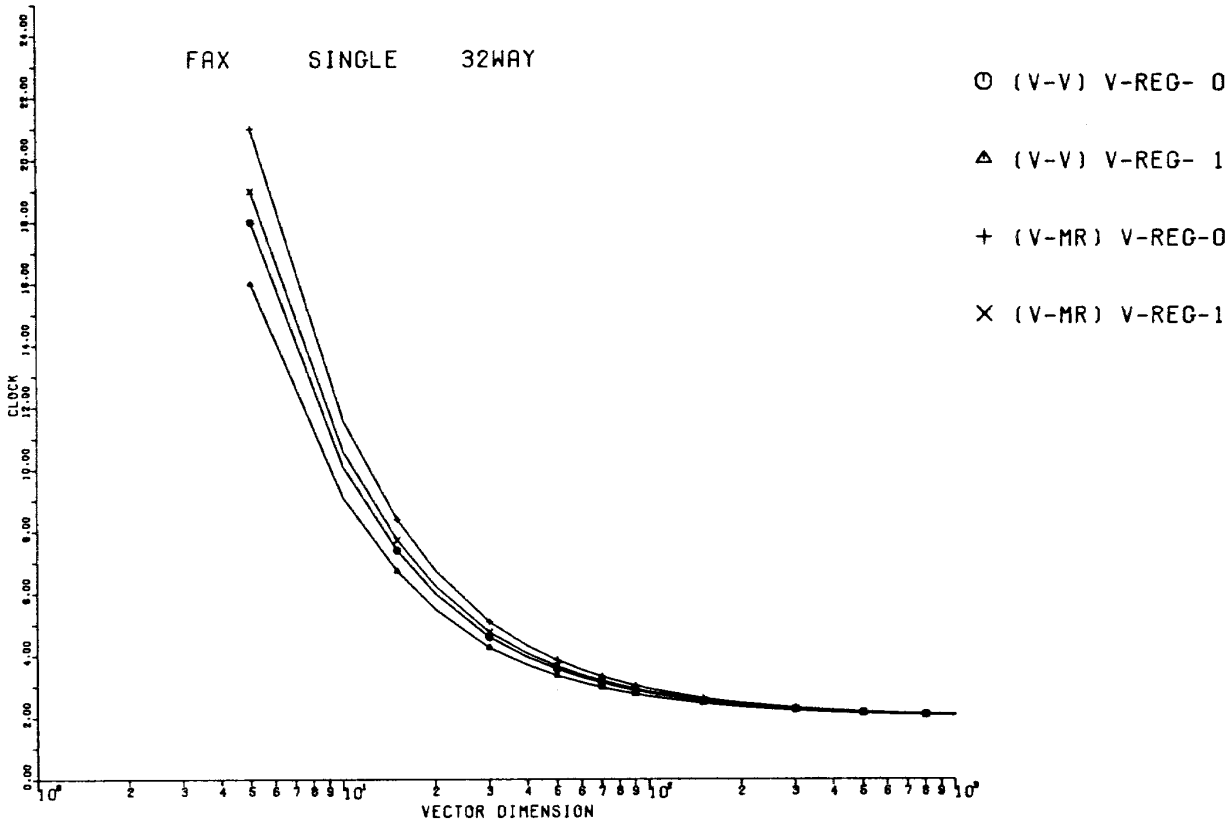


図 4.2.21

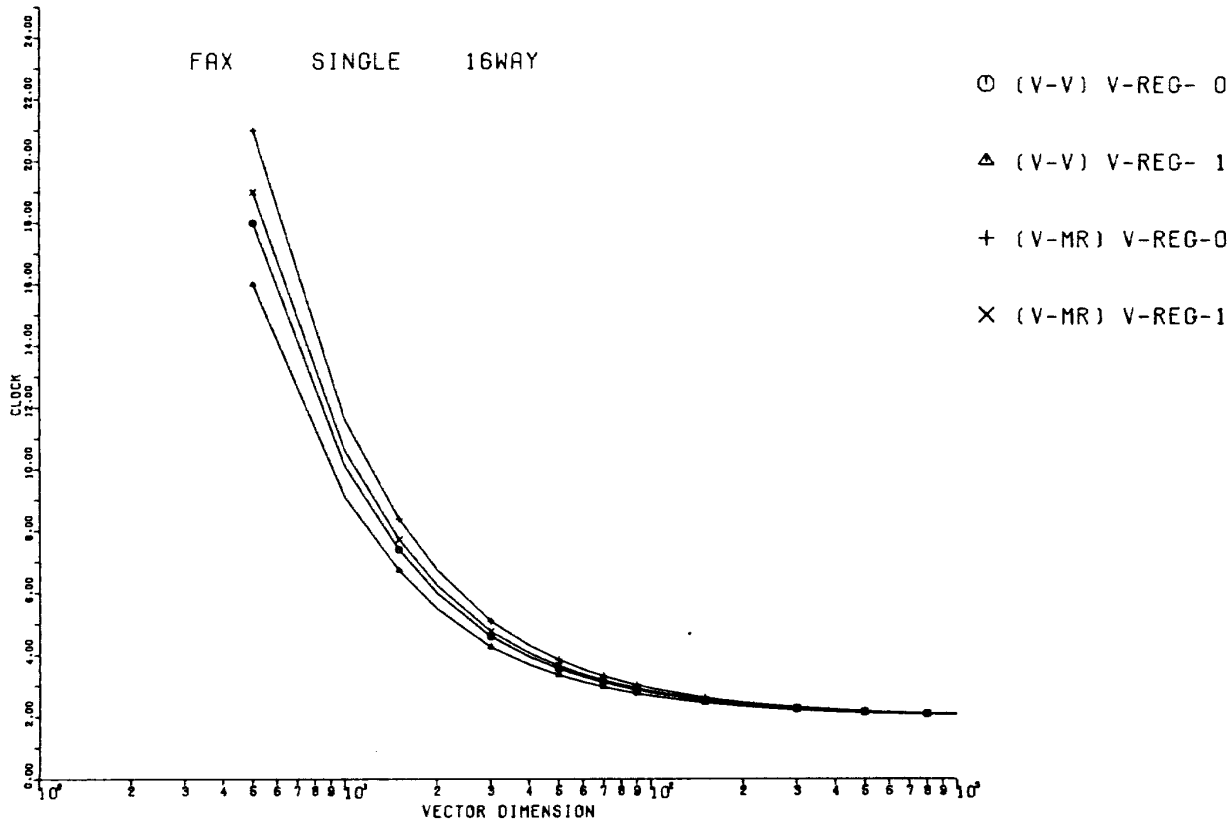


図 4.2.22

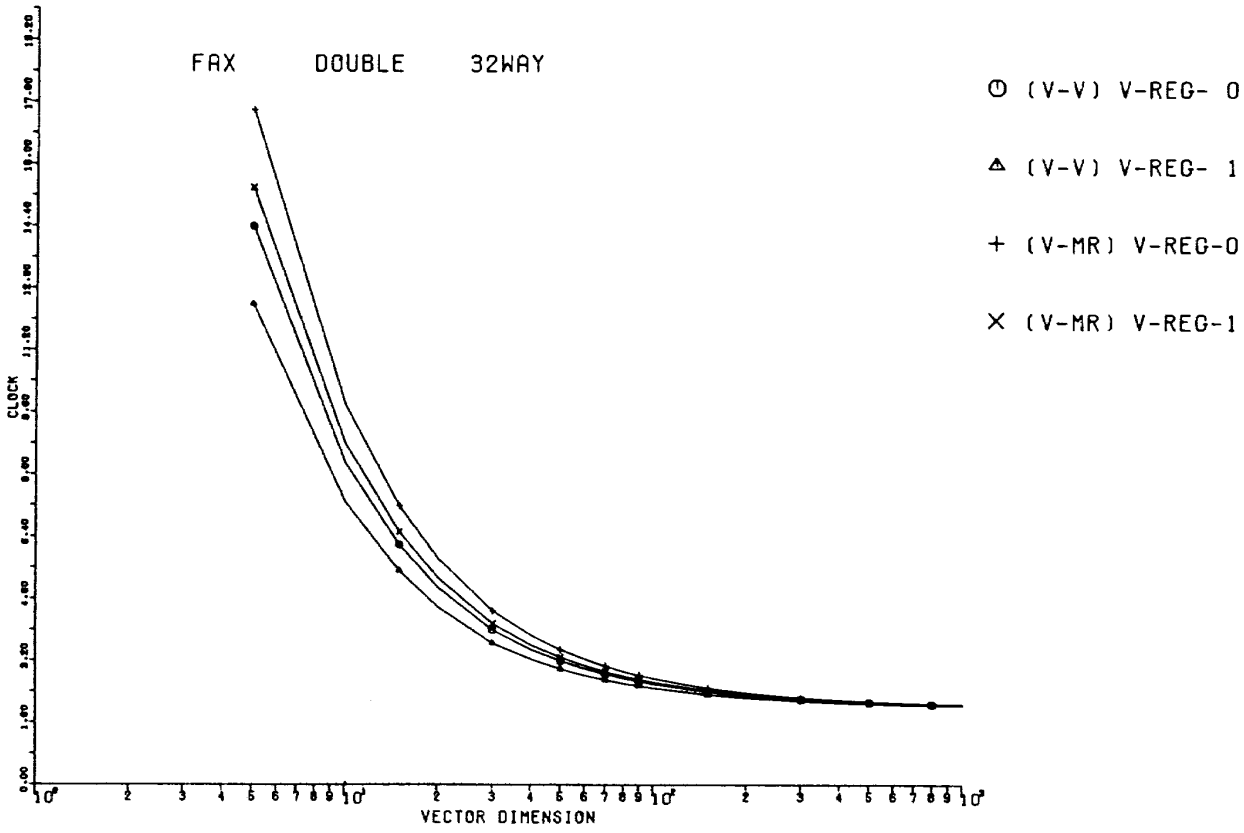


図 4.2.23

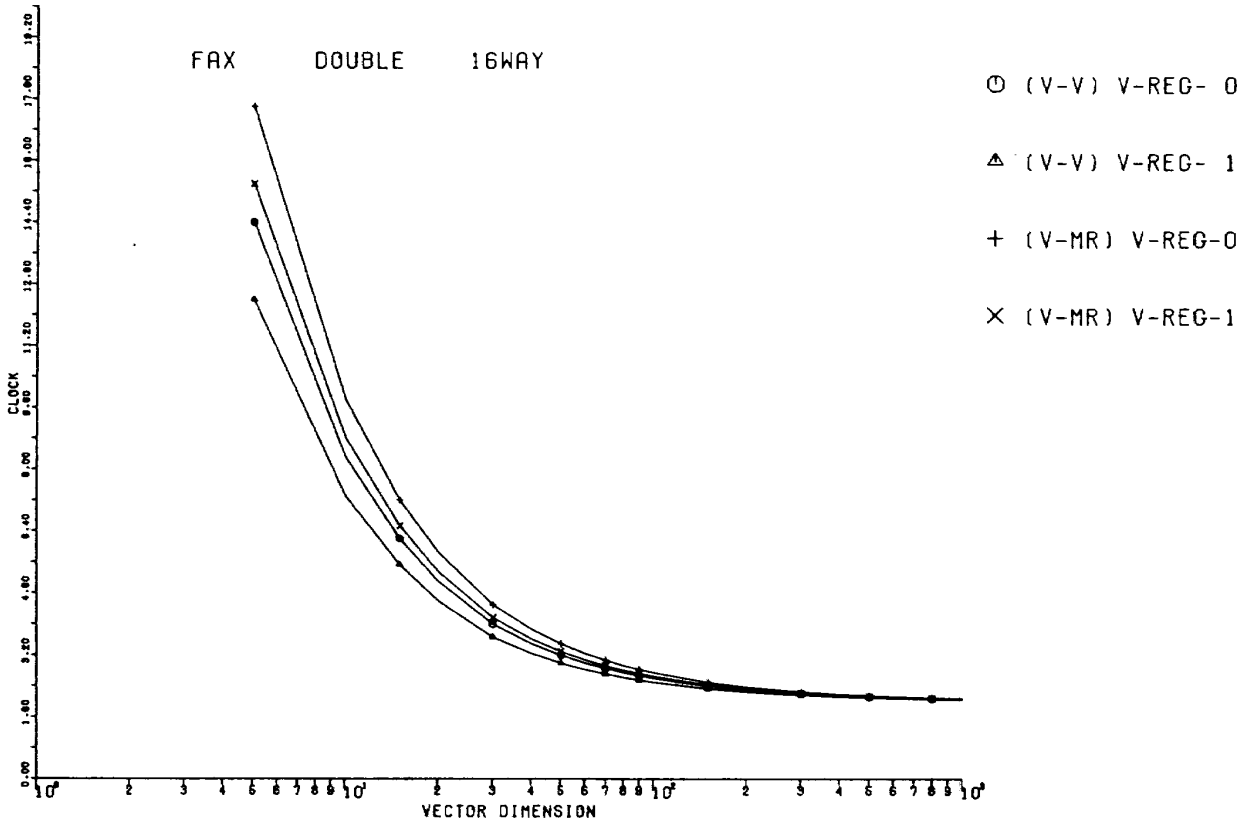


図 4.2.24

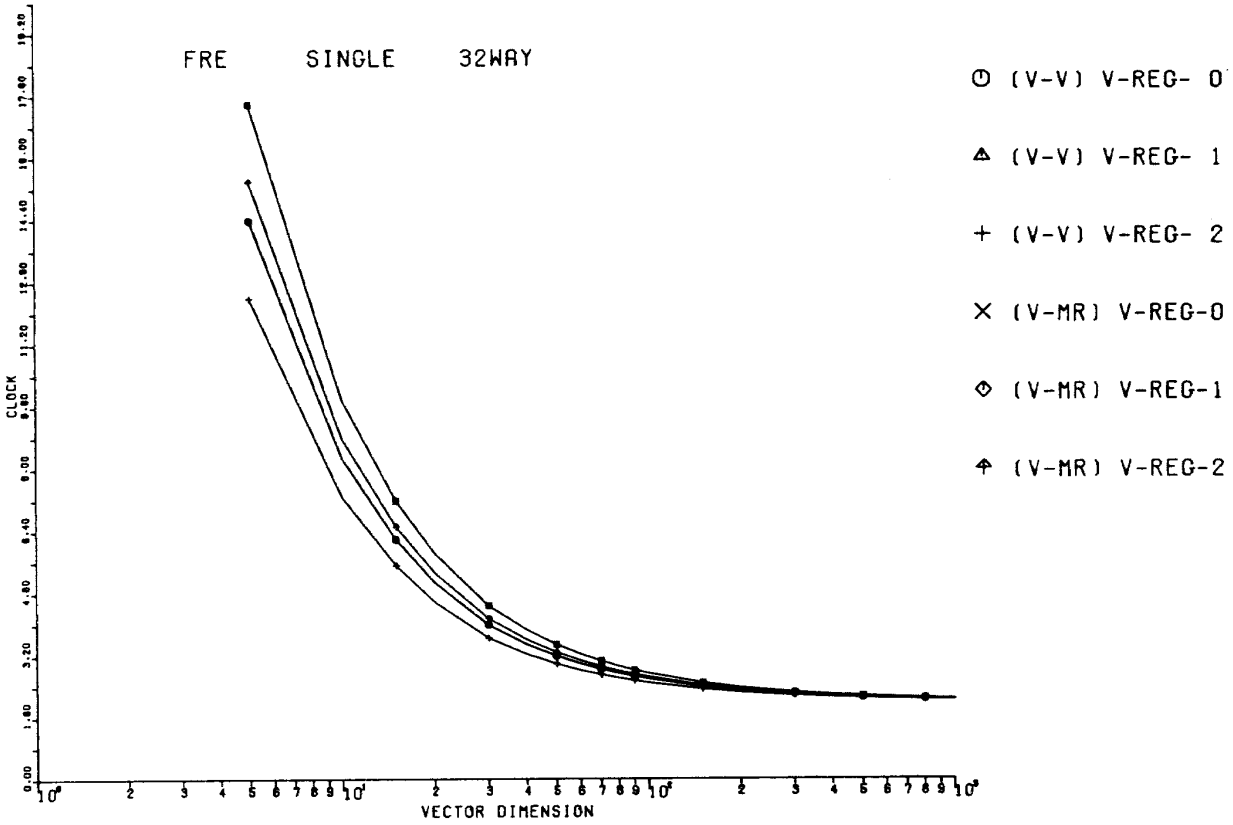


図 4.2.25

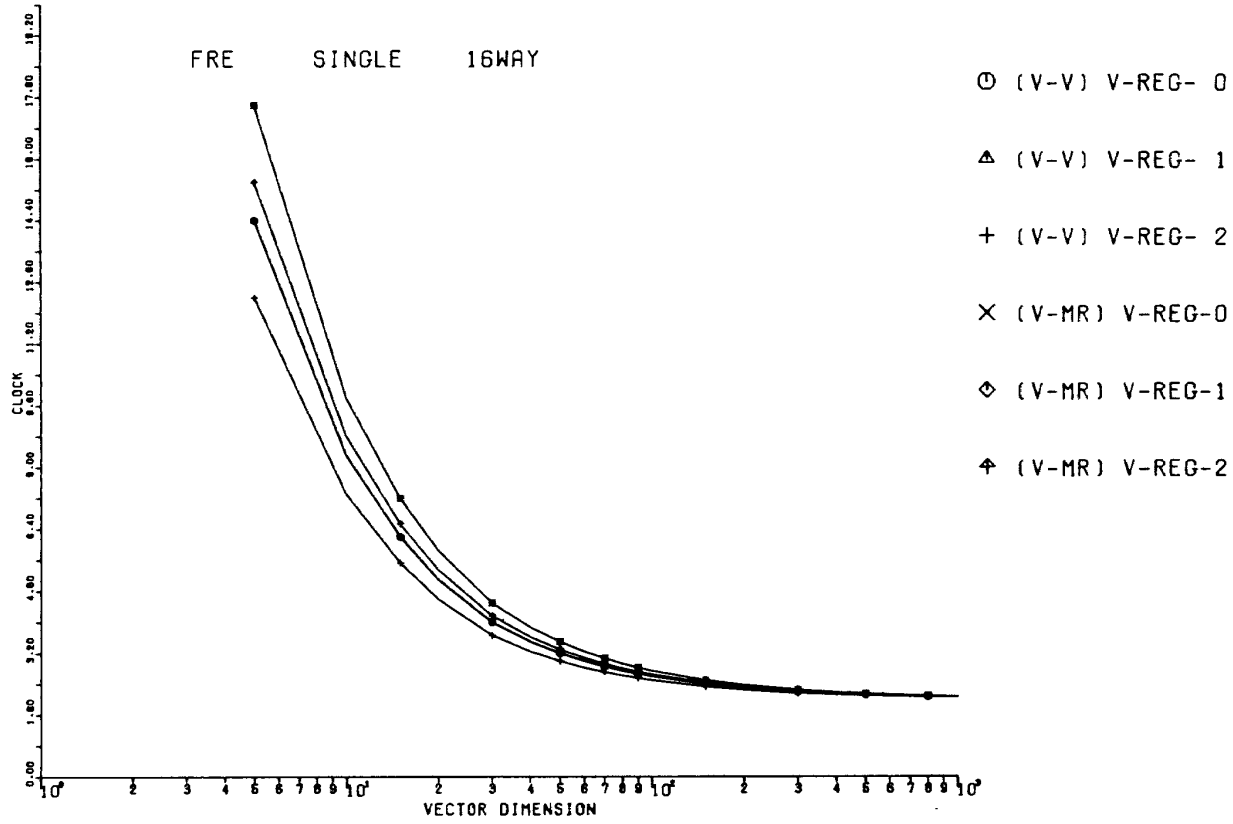


図 4.2.26

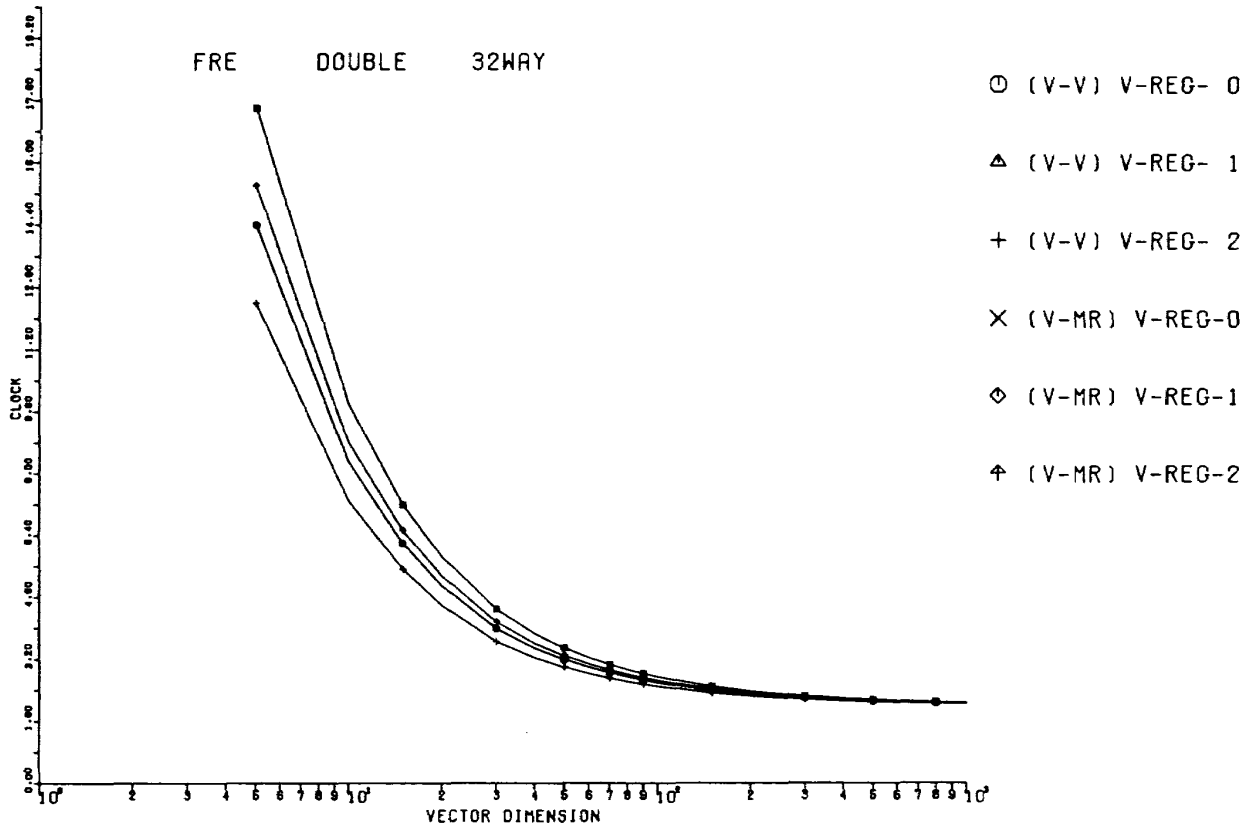


図 4.2.27

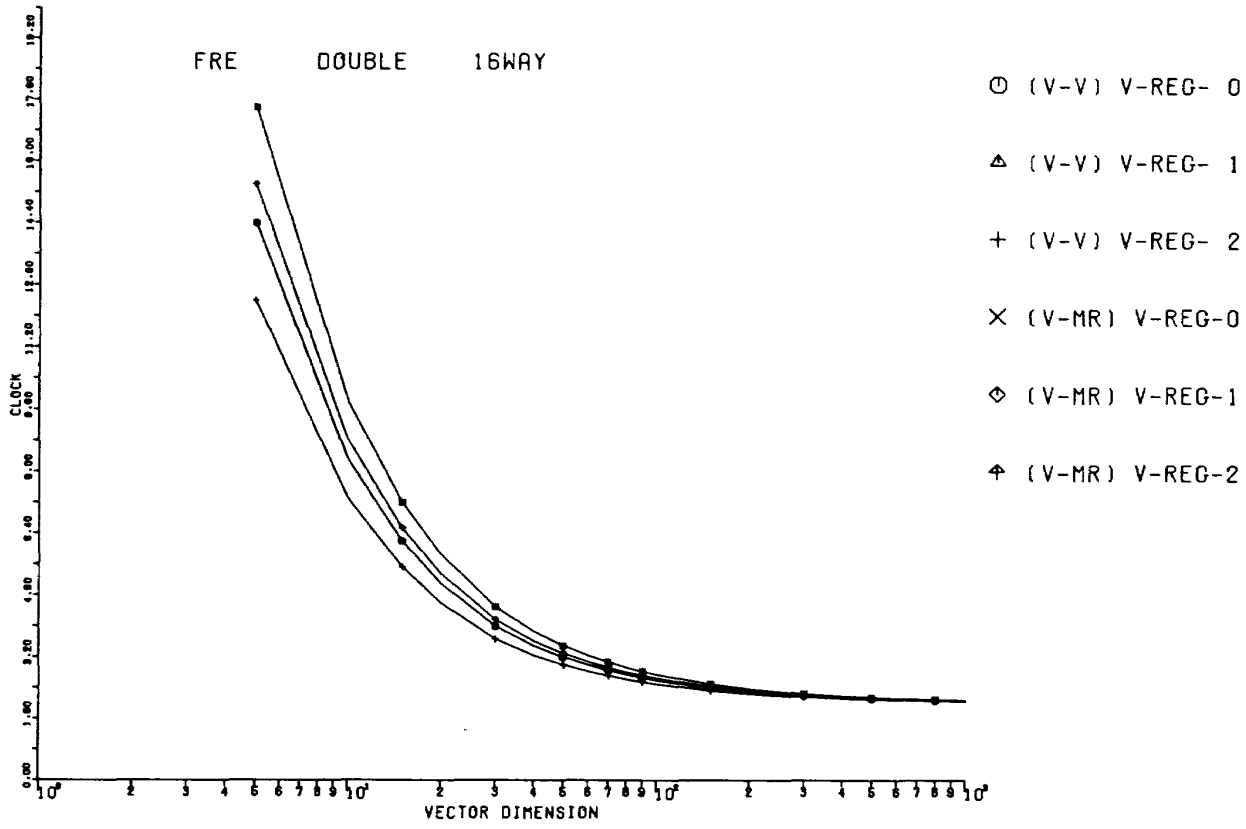


図 4.2.28

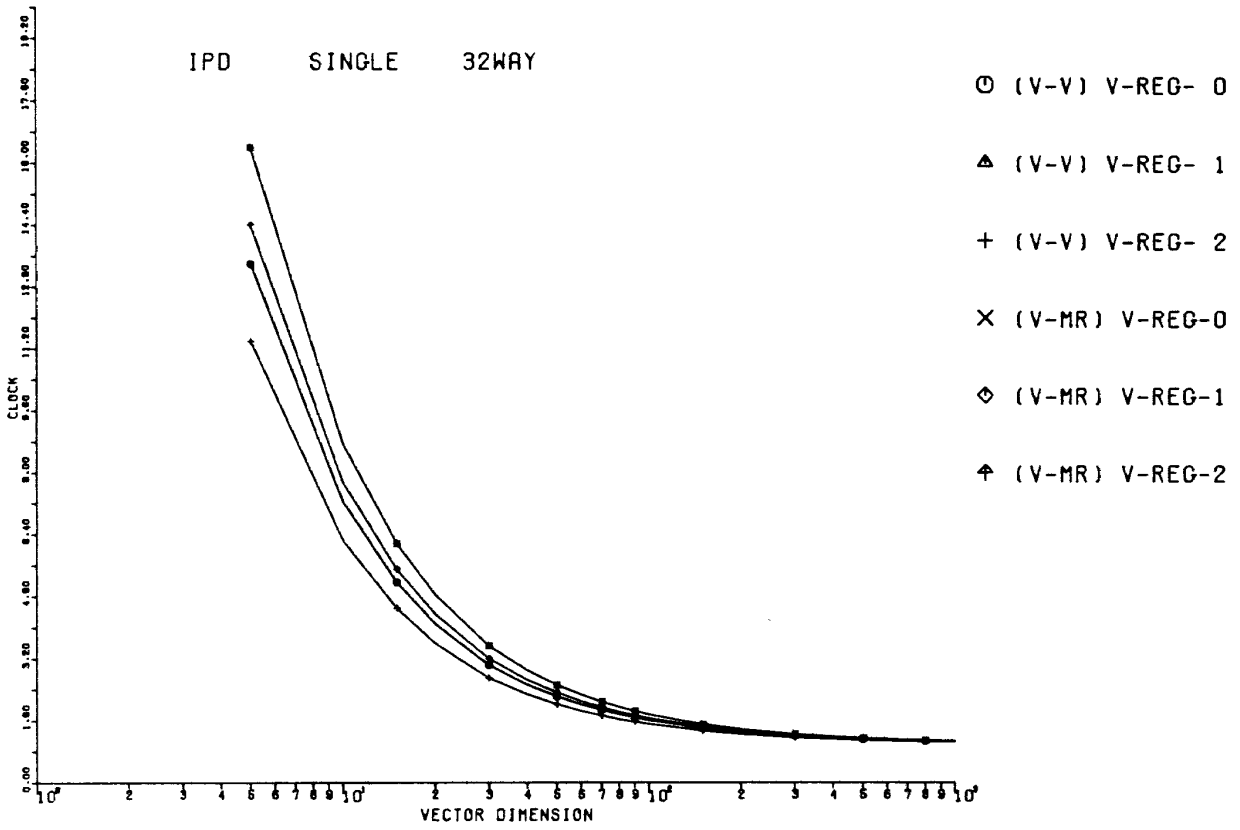


図 4.2.29

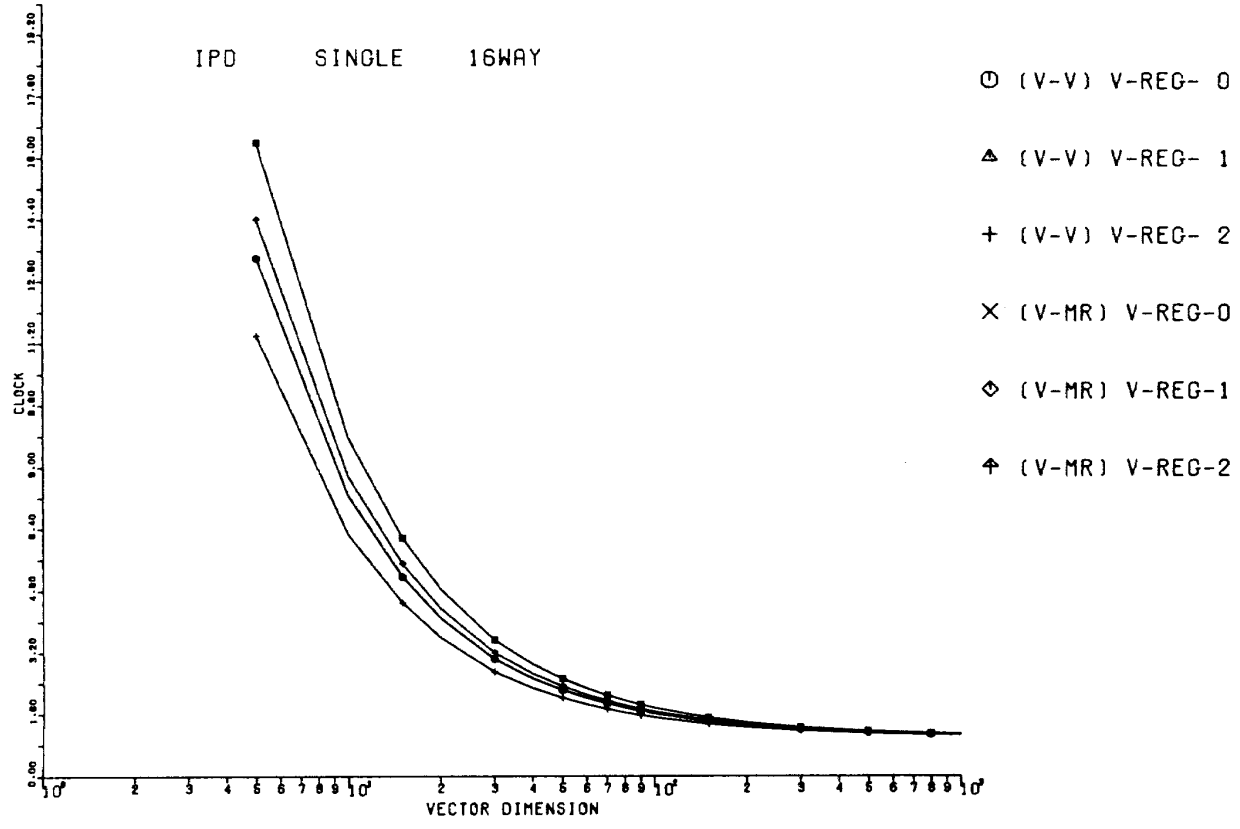


図 4.2.30

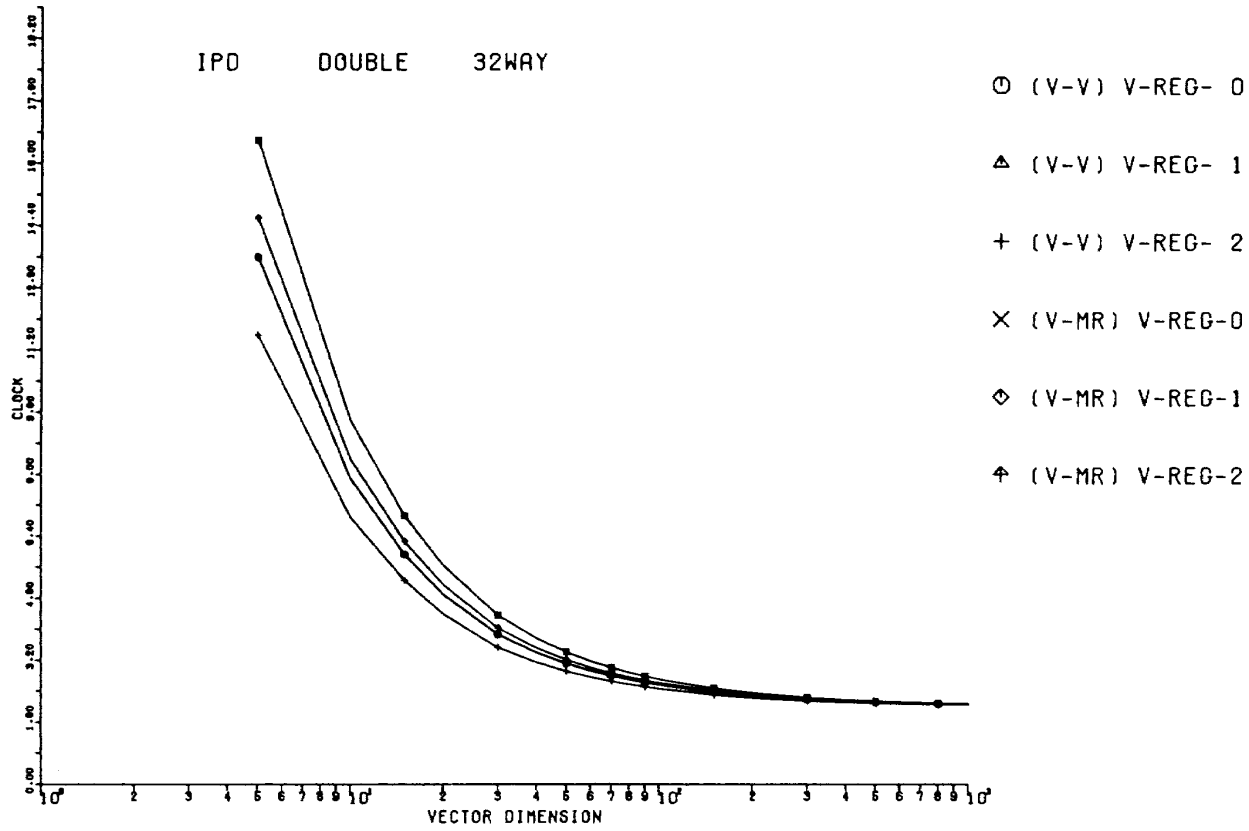


図 4.2.31

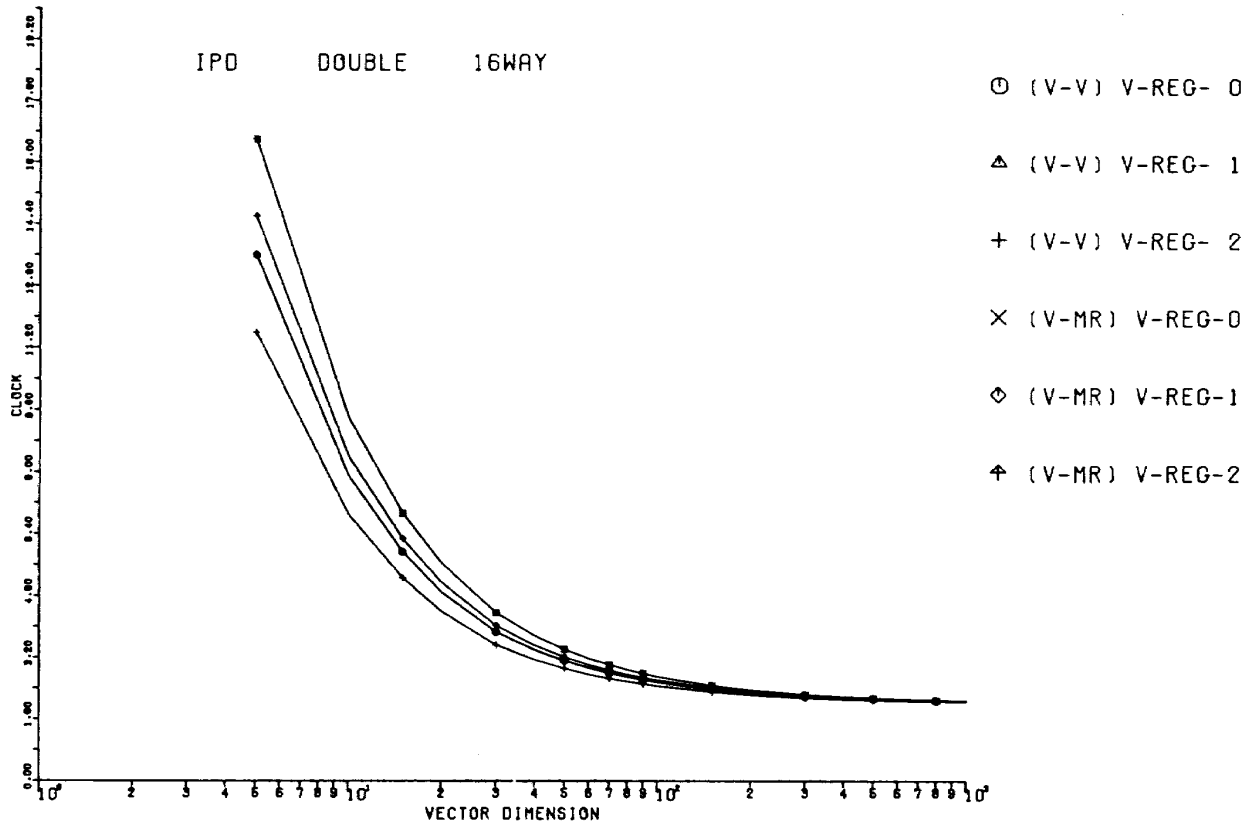


図 4.2.32

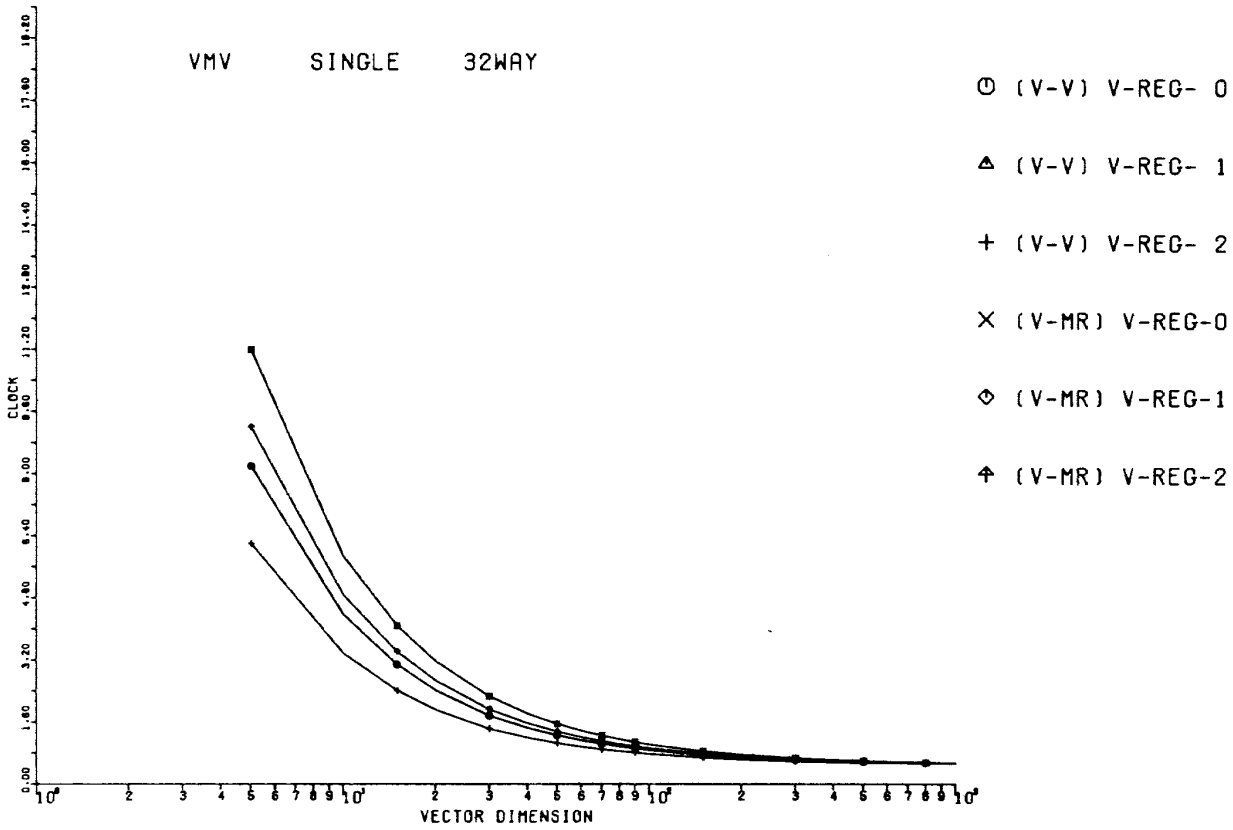


図 4.2.33

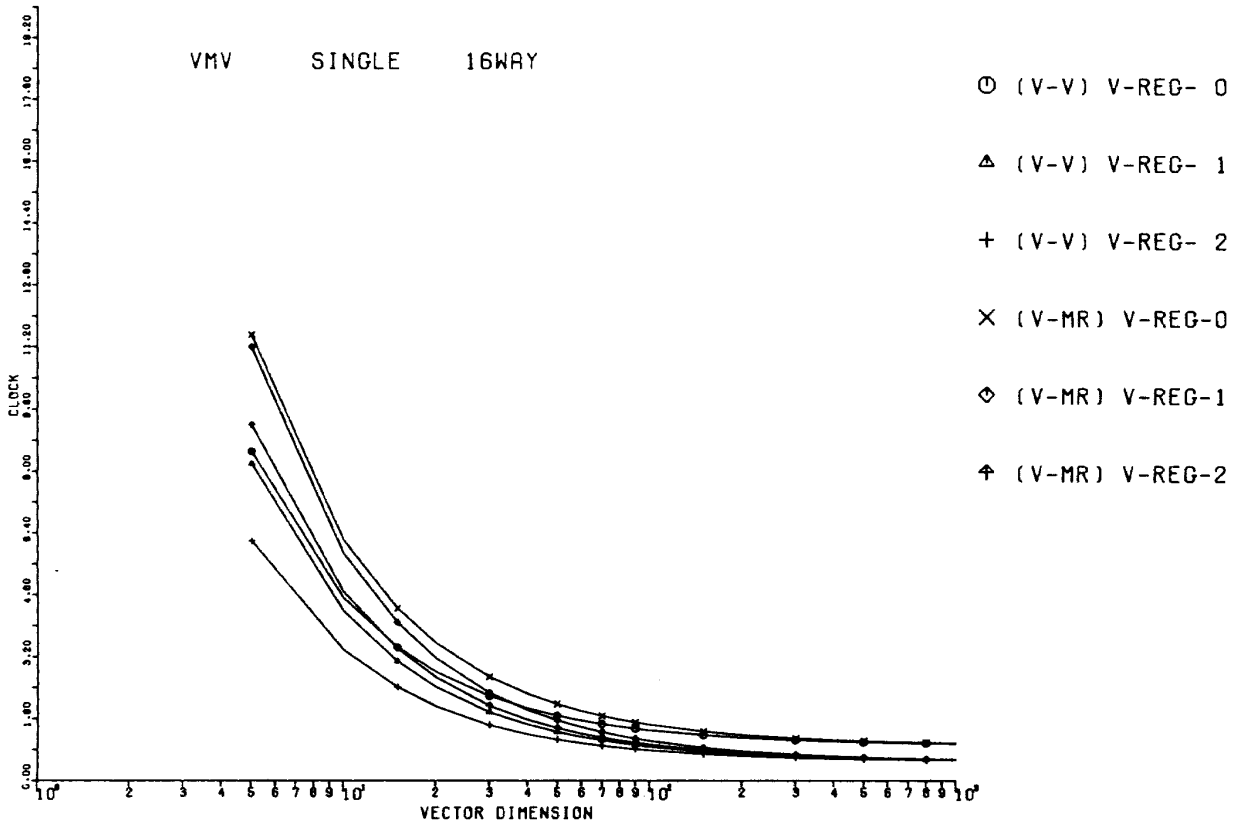


図 4.2.34

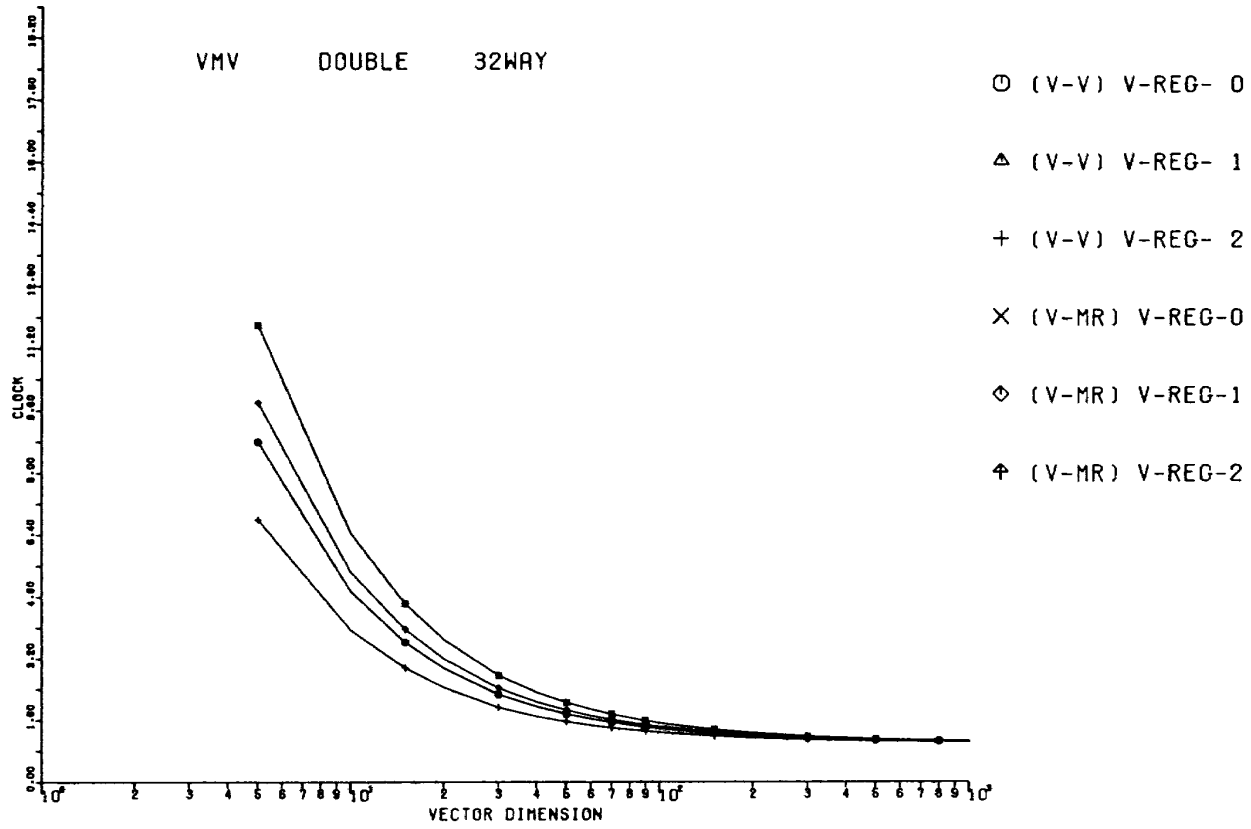


図 4. 2. 35

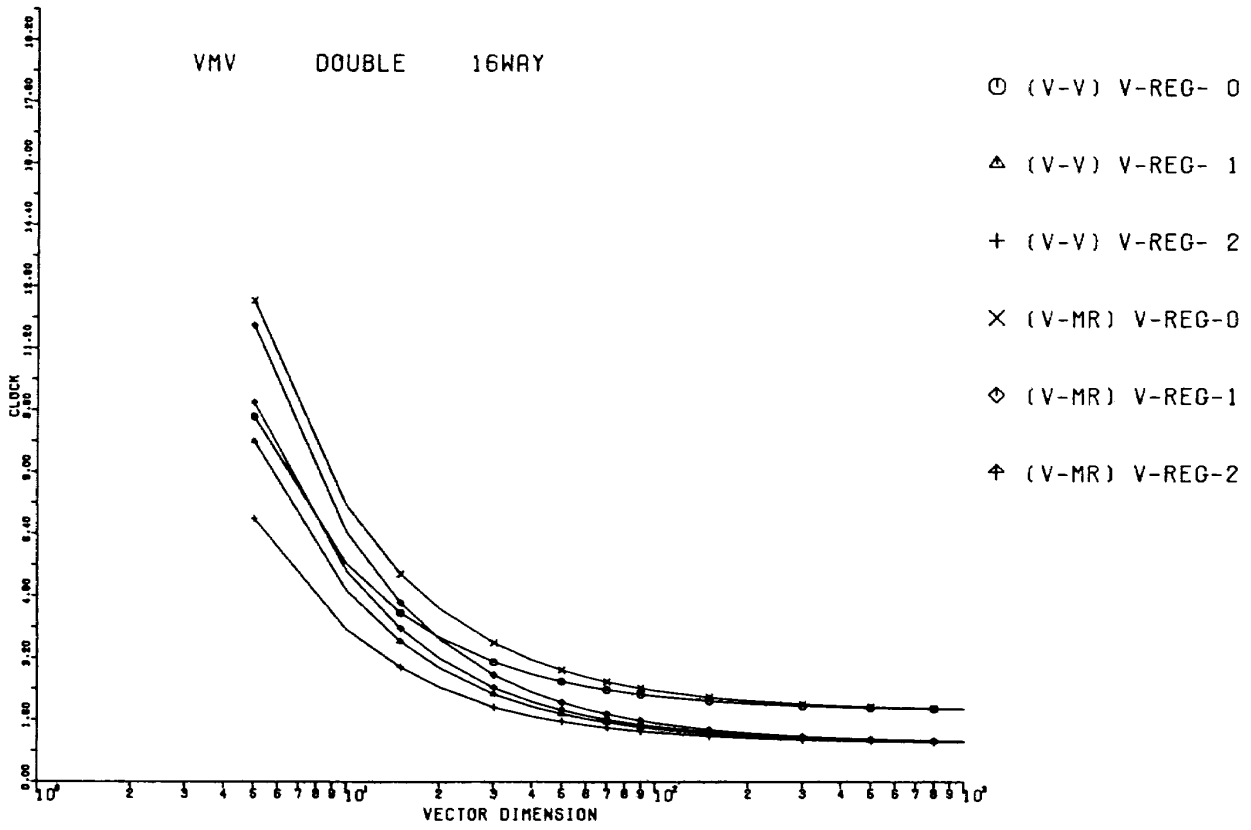


図 4. 2. 36

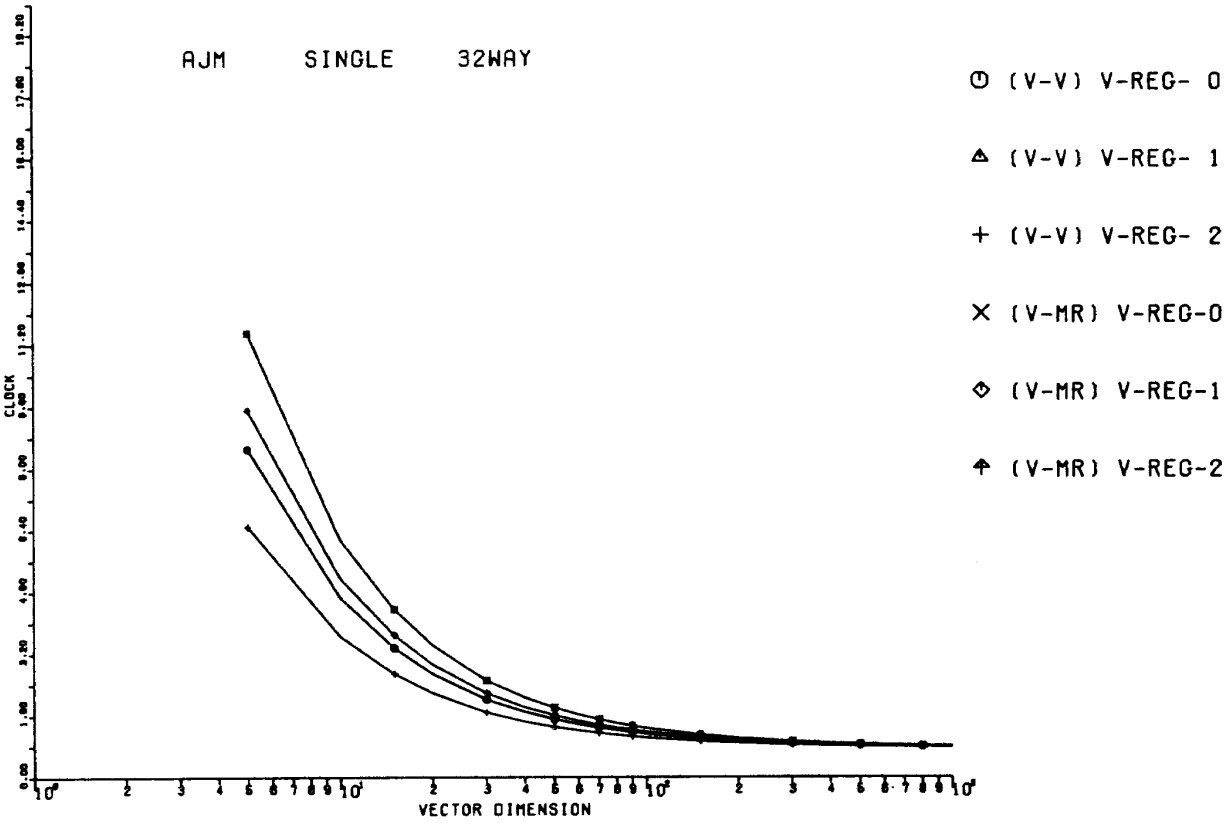


図 4.2.37

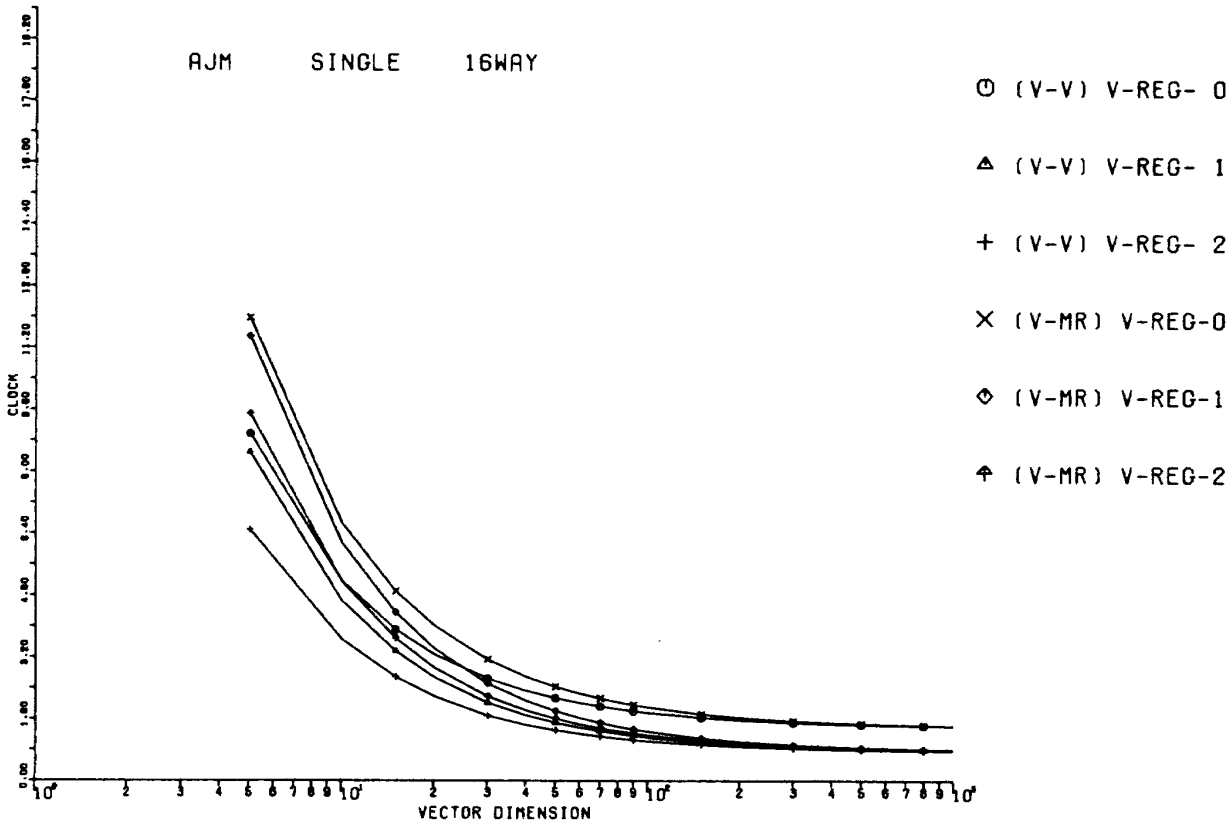


図 4.2.38

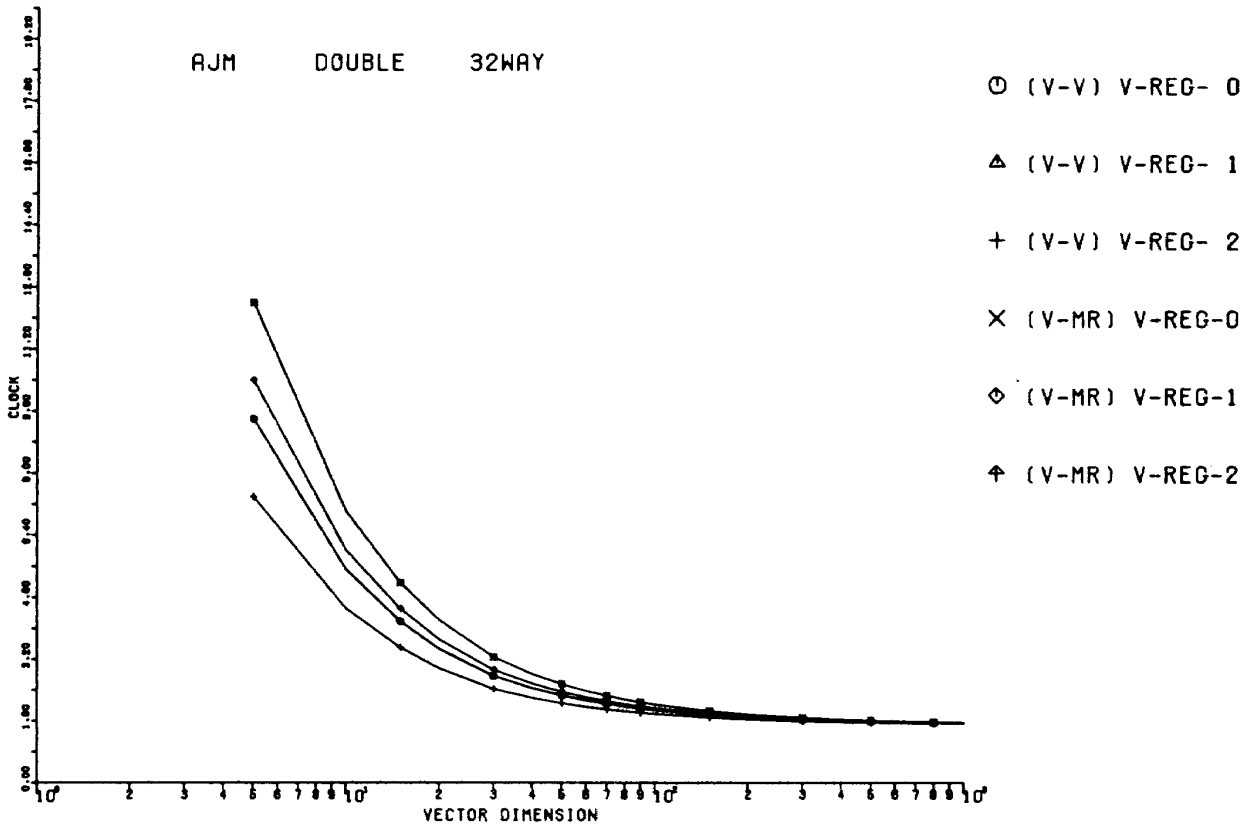


図 4.2.39

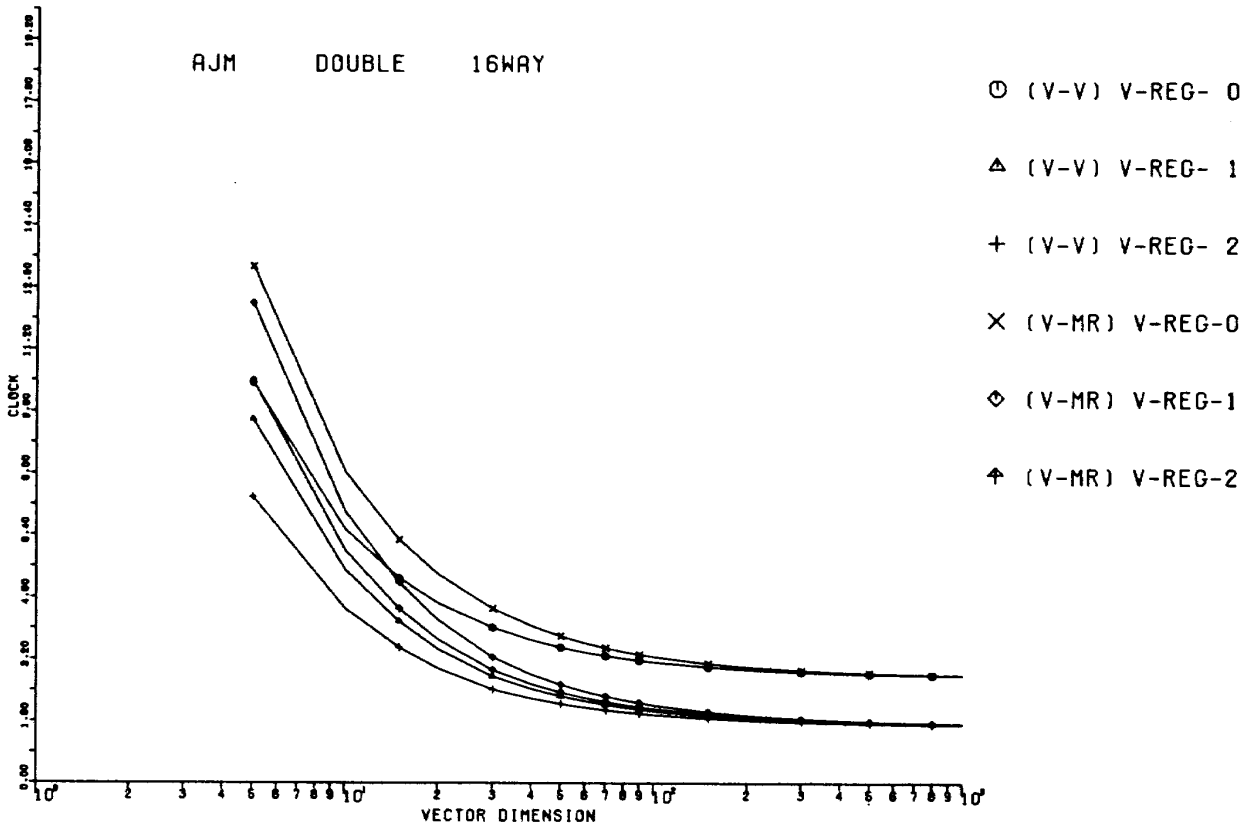


図 4.2.40

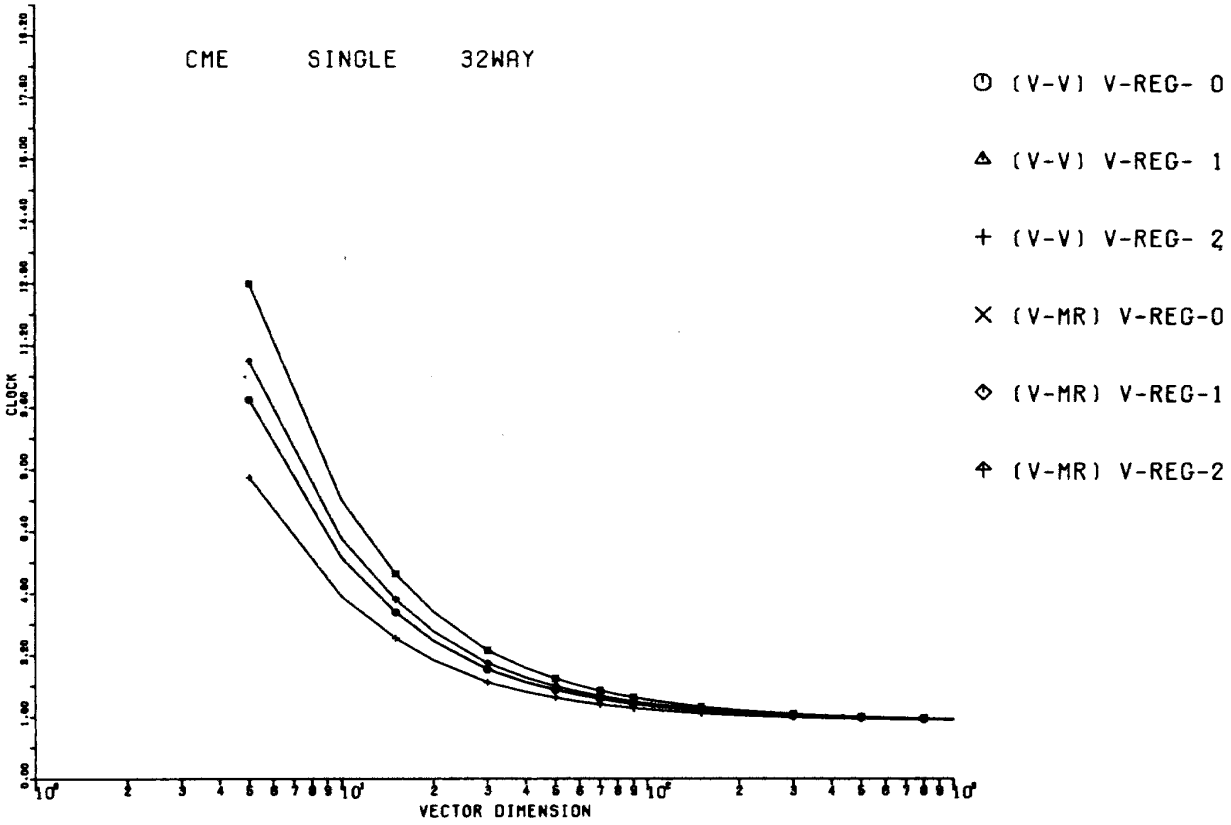


図 4.2.41

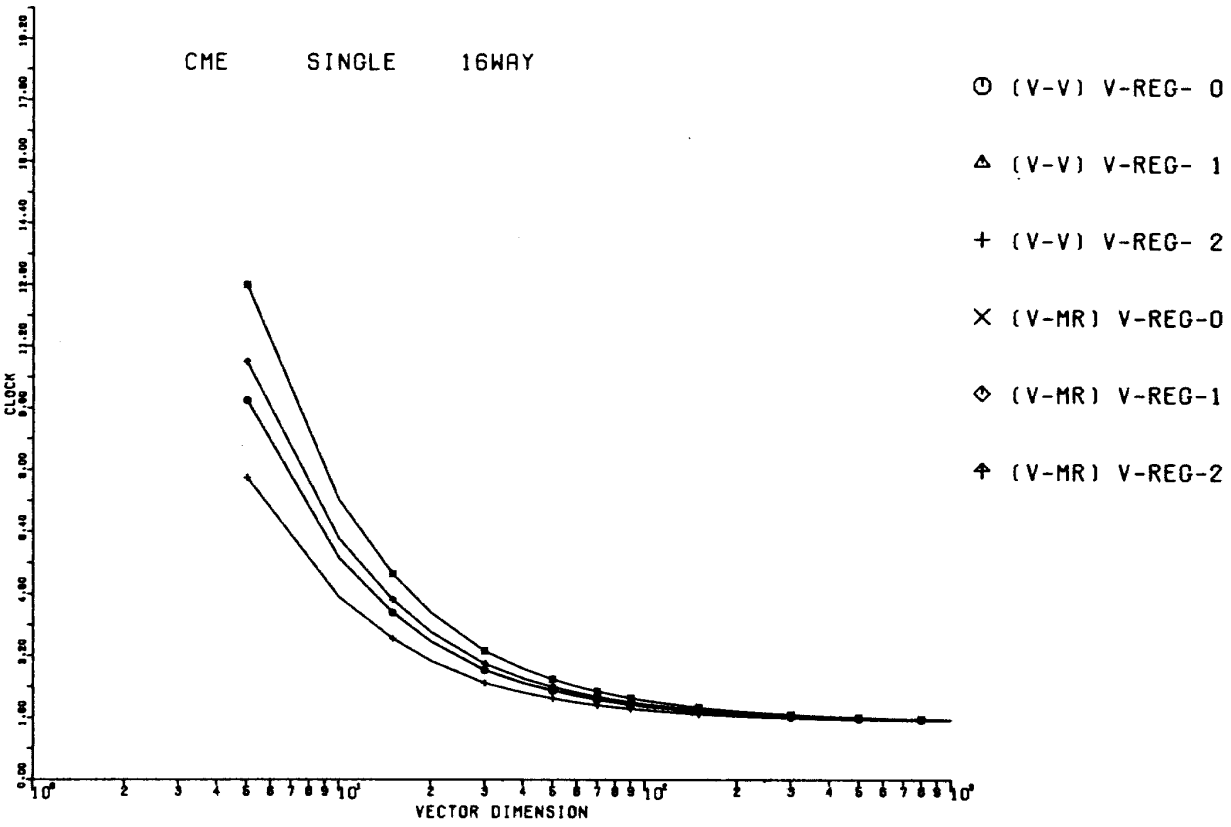


図 4.2.42

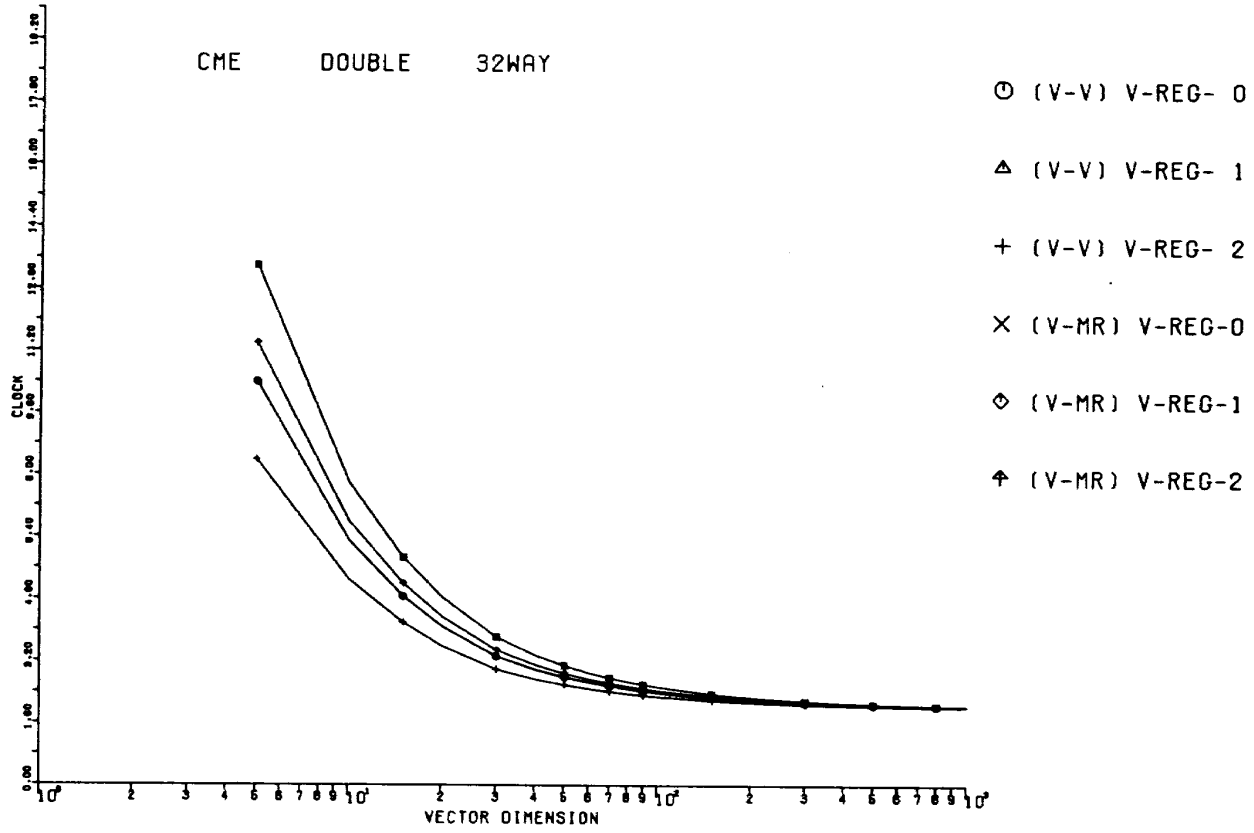


図 4.2.43

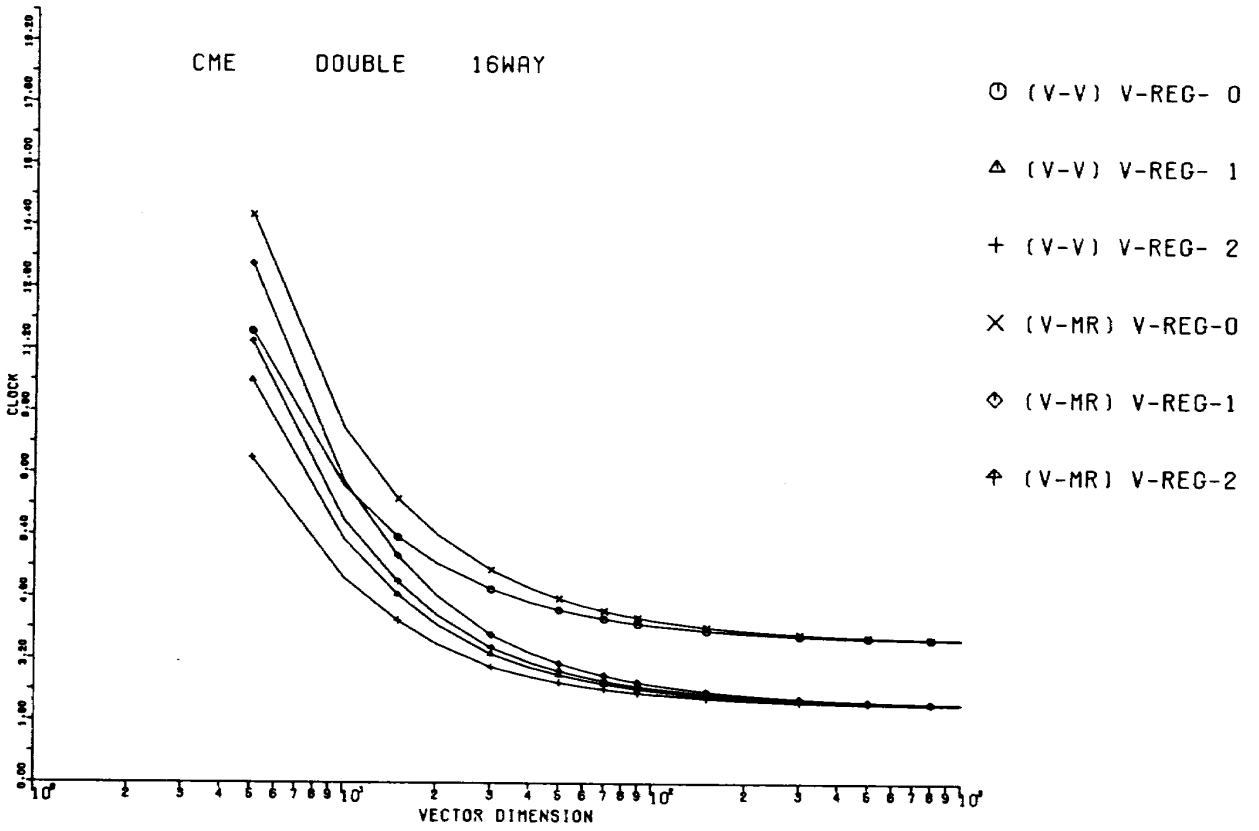


図 4.2.44

4.3 代表的なV型命令の実行速度図の説明

4.2で説明したV型命令の中から代表的な6つの命令を選び，その実行速度図を図4.3.1～図4.3.12に示しその実行速度曲線の特徴について説明する。それらの図は，インタリーブ数が32WAYと16WAYの場合において，単精度，単精度ディスタンス付き（以下SDと略称する），倍精度，4倍精度のとき，それぞれV-Reg上にオペランドが全部有か否の8本の実行速度曲線で示してある。それぞれの図で使用している記号は単精度，SD，倍精度，4倍精度についてのV-Reg上のオペランドの関係を示している。例えば記号○は単精度でV-Reg上にオペランドが全く無い場合であり，以下，（単精度・V-Reg無）と略称する。同様に単精度でV-Reg上にオペランドが全部有る場合は，（単精度・V-Reg有）と略称して記号△を指す。以下，SD，倍精度，4倍精度についても同様に略称する。また，それらの図における縦軸と横軸の表示方法は4.2の場合と同じである。以下それぞれのV型命令の実行速度図について説明する。

4.3.1 VAD

図4.3.1と図4.3.2にVADについて，インタリーブ数が32WAYと16WAYの場合の実行速度図を示してある。まず図4.3.1においては

i) SDと倍精度のV-Reg無の実行速度曲線が重なっている。またSDと倍精度のV-Reg有の曲線も重なっている。

ii) Nが小さいときは，単精度，倍精度，SDは供にV-Reg有の方が実行速度は速い。しかしNがある程度以上大きくなると，4.2.1で述べた様に単精度の方が，V-Reg無の場合でも実行速度は速くなる。

iii) 4倍精度の実行速度曲線は，Nが小さい時はV-Reg有の方が実行速度は速いが，大きくなるとその実行速度の収束値は一致する。

iv) それらの実行速度曲線は，Nが非常に大きくなると，単精度，倍精度，4倍精度の順に（但しSDは倍精度と同じになっている）しかもV-Reg上のオペランドの有無に依存して実行速度の収束値が異なっている。

次に図4.3.2の実行速度曲線においては，

v) SDと倍精度の実行速度曲線の重なりは図4.3.1の場合と同じである。そして単精度でV-Reg無の場合より，倍精度とSDのV-Reg有の方が実行速度は速い。

vi) 4倍精度においては，V-Reg上のオペランドの有無により実行速度の収束値が異なる。

以上がVADの実行速度曲線の大まかな特徴である。

VADの演算速度は，インタリーブ数が32WAYの場合は，Nが小さい時は，4倍精度を除き，命令の精度より，V-Reg上のオペランドの有無に依存するが，Nが大きくなると，V-Reg上のオペランドの有無より，命令の精度に依存してくる。このことは4.2.1で説明した様に図4.2.1と図4.2.2の比較による。そして4倍精度においては，表3.2.1.bより，その実行速度は速く，データ供給能力が満足される格好となりその実行速度の収束値は一致してくる。インタリーブ数が16WAYになると32WAYの場合とはv)，vi)に示した様に実行速度曲線の性質が異なってくる。4倍精度を除き，その実行速度は命令の精度より，V-Reg上のオペランドの有無に依存している。そして4倍精度においてもV-Reg上のオペランドの有無によりその実行速度は異なっている。以上のことからインタリーブ数が32WAY，16WAYでもV-Regの有効使用が望ましいことがわかる。そして4.1で述べた様に，16WAYの場合は特にV-Regの有効使用が望ましくなる。また図の実行速度曲線からもわかる様に4倍精度は実行速度がかなり遅いことに注意する必要がある。

4.3.2 VML

図4.3.3と図4.3.4にインタリーブ数が32WAYと16WAYの場合の実行速度図を示してある。

図4.3.3の実行速度曲線は，

i) 単精度とSDのV-Reg無の実行速度曲線は重なっている。また単精度とSDのV-Reg有の曲線も重なっている。

ii) 4倍精度を除き，Nが小さい時は，命令精度よりV-Reg上のオペランド数に依存しているが，Nが大きくなると，命令精度に依存している。そしてNが非常に大きくなると，それらの実行速度は，各々の命令精度にて（4倍精度を含む）収束値が異なる。

図4.3.4の実行速度曲線は

iii) 単精度とSDの実行速度曲線の重なりは，i）と同じである。そして，Nが小さい時はii）と同じであるが，Nが大きくなると，同じ命令精度でもV-Reg上のオペランドの有無により実行速度が異なりNが非常に大きくなってもその実行速度の収束値は4倍精度を除き，V-Reg上のオペランドの有無により異なる。

以上がVMLの実行速度曲線の特徴であるが，このVMLの演算において，インタリーブ数が32WAYの場合はii）に示している様に，Nが非常に大きくなると，V-Reg上のオペランドの有無よりも命令精度により実行速度の収束値が異なっている。しかしインタリーブ数が16WAY

の場合は，Nがどんなに大きくなっても，32WAYの場合と違い同じ命令精度でも，V−Reg 上のオペランドの有無で，実行速度の収束値は異なっていることに注意しなければならない。但し4倍精度の場合は，表3.2.1.bに示してある様に実行速度が遅く，データ供給能力が満足されるので，実行速度の収束値は一致する。この4倍精度も図の実行速度曲線からわかる様に実行速度が遅いので使用する際は注意する必要がある。

4.3.3 VDV

図4.3.5と図4.3.6にインタリーブ数が32WAYと16WAY の場合の実行速度図を示してある。この2つの図は同じ実行速度曲線をしている。これは，表3.2.1.bの実行速度に示す様にVDV の実行速度は，他のV型命令に比べて非常に遅い。それ故データ供給能力はどの命令精度においても，十分満足される格好となり，V−Reg 上のオペランドにもあまり依存することはなくなってくる。したがってインタリーブ数が32WAY，16WAYにおいても影響はなくなる。

4.3.4 VSM

図4.3.7と図4.3.8にインタリーブ数が32WAYと16WAY の場合の実行速度図を示してある。このVSMにおいて，2つの図は同じ実行速度曲線であるので，図4.3.7を代表させる。

i) Nが小さい時は，倍精度でV−Reg 有の場合が単精度でV−Reg 有よりも実行速度は速い。これは，4.2.4で述べた。また単精度とSDのV−Reg 無は，4倍精度でV−Reg 有よりも実行速度は遅い。

ii) Nがある程度大きくなっても，実行速度は命令精度よりも，V−Reg 上のオペランドの有無に依存している。

iii) Nが非常に大きくなると，V−Reg 上のオペランドの有無よりも，命令精度に依存して実行速度の収束値が異なる。この場合精度とSDの実行速度は同じ収束値となる。

以上がVSMの実行速度曲線の特徴である。このVSMの演算において，Nが非常に小さい時は，V−Reg 上にソースオペランドが有か否かにかなり依存しており，ある程度Nが大きくなっても，かなりの影響がある。したがってNの値が小さい時に注意する必要がある。また4倍精度の演算においても実行速度は，他の命令精度に比べて遅いことにも注意する必要がある。

4.3.5 VMV

図4.3.9と図4.3.10にインタリーブ数が32WAYと16WAY の場合の実行速度図を示してある。

図4.3.9の実行速度曲線においては

i) SDと倍精度のV−Reg 無の実行速度曲線は重なっている。またV−Reg 有に関しても重なっている。

ii) 4倍精度を除き，Nが小さい時は，命令精度よりV−Reg 上のオペランドの有無に依存して実行速度は異なっている。しかしNがある程度大きくなると，命令精度に依存してくる。

iii) Nが非常に大きくなると，各々の命令精度の実行速度は，V−Reg 上のオペランドに対しての依存度はなくなり，各々の命令精度で収束値をもつ。

図4.3.10の実行速度曲線においては

iv) SDと倍精度の実行速度曲線の重なりは，i) で示したのと同様である。そして命令精度とV−Regの関係はii) とほぼ同様である。

v) Nが非常に大きくなると，4倍精度を除き，同じ命令精度でもV−Reg に依存して実行速度の収束値が異なる。

以上がVMV の実行速度曲線の特徴である。インタリーブ数が32WAYにおいては，ii) に示してある様に実行速度はNの値によりV−Reg に依存している。そしてNが非常に大きくなるとiii) に示す様に各々の命令精度での実行速度に収束するので，V−Reg にあまり依存はしなくなるが，V−Reg の有効使用は必要である。これが16WAYになると，v) に示してある様に，同じ命令精度でもV−Reg の有効使用が絶対に必要となる。4倍精度に関しては，これまでのV型命令の説明と同様に，その演算においては，実行速度が遅いことに注意してもらいたい。

4.3.6 FAX

図4.3.11と図4.3.12にインタリーブ数が32WAYの場合の実行速度図を示してある。FAX においては，2つの図は同じ実行速度曲線であるので，図4.3.11を代表させる。

i) 単精度とSDのV−Reg 無の実行速度曲線は重なっている。また同じく，V−Reg 有の曲線も重なっている。

ii) Nがある程度大きいところまでは，単精度よりも倍精度の方が実行速度は速い。そして同じ命令精度においてはV−Reg 有の方が速い。

iii) Nが非常に大きくなると，4倍精度を除き，それらの実行速度は同じ値に収束する。

以上がFAX の実行速度曲線の特徴である。この場合，ii) に示してある様に倍精度の方が単精度よりも実行速度が速いことに注意する必要がある。そしてV−Reg の有効使用がやはり要求される。4倍精度においては今までのV型命令の説明と同じく，その実行速度が遅いこと

に注意をする必要がある。

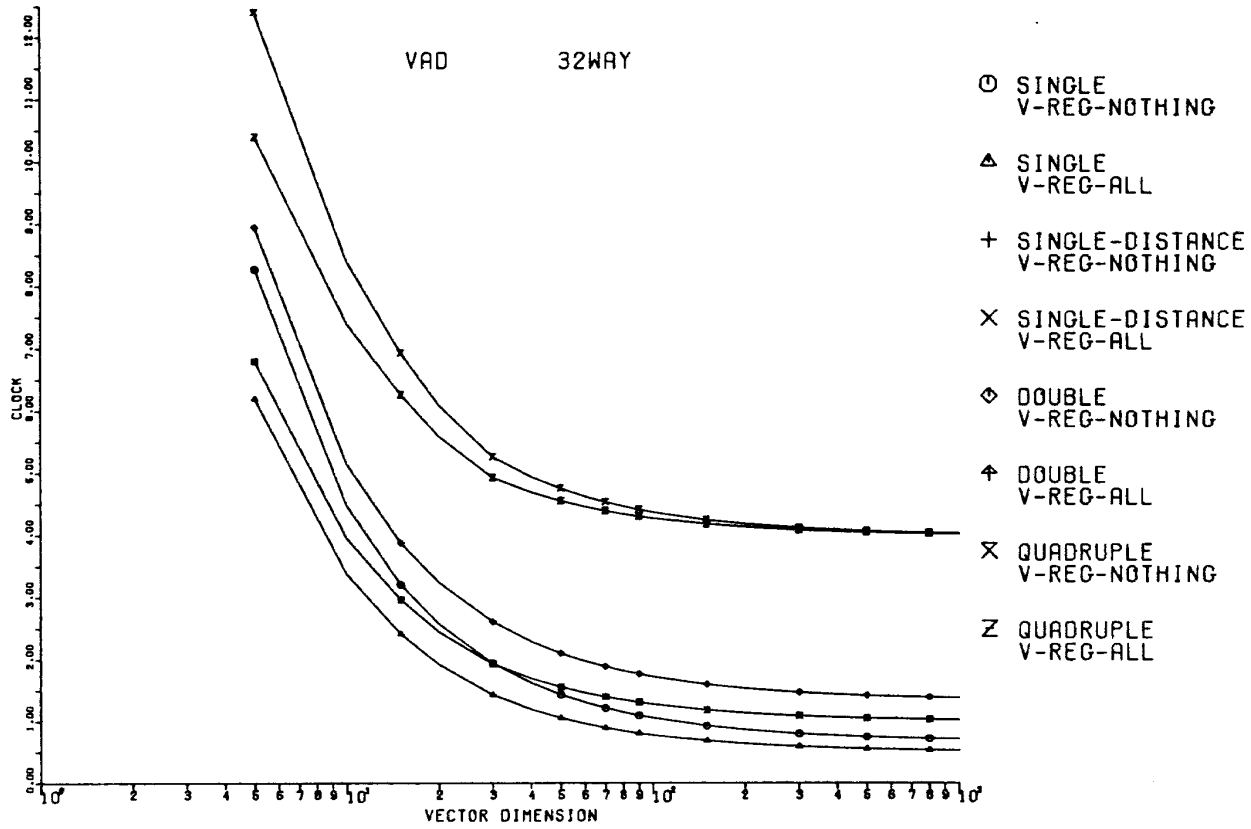


図 4.3.1

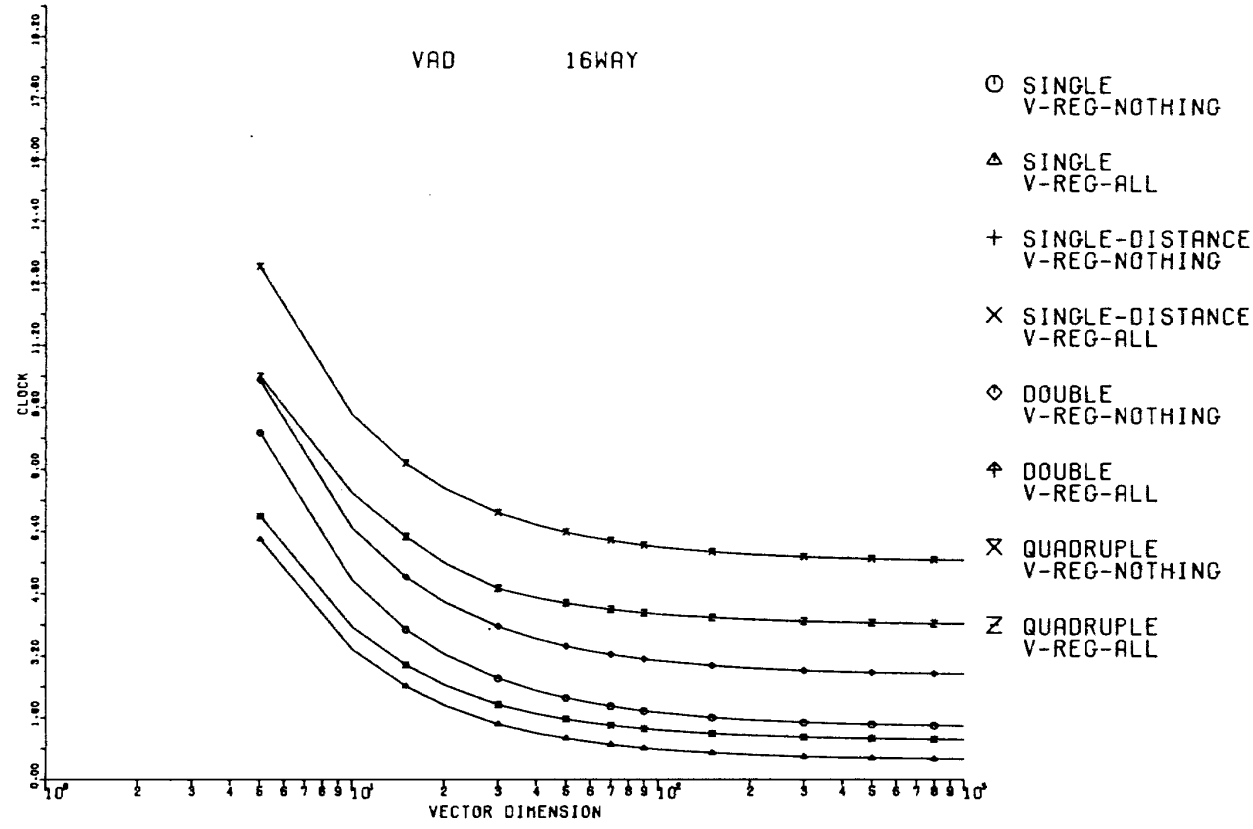


図 4.3.2

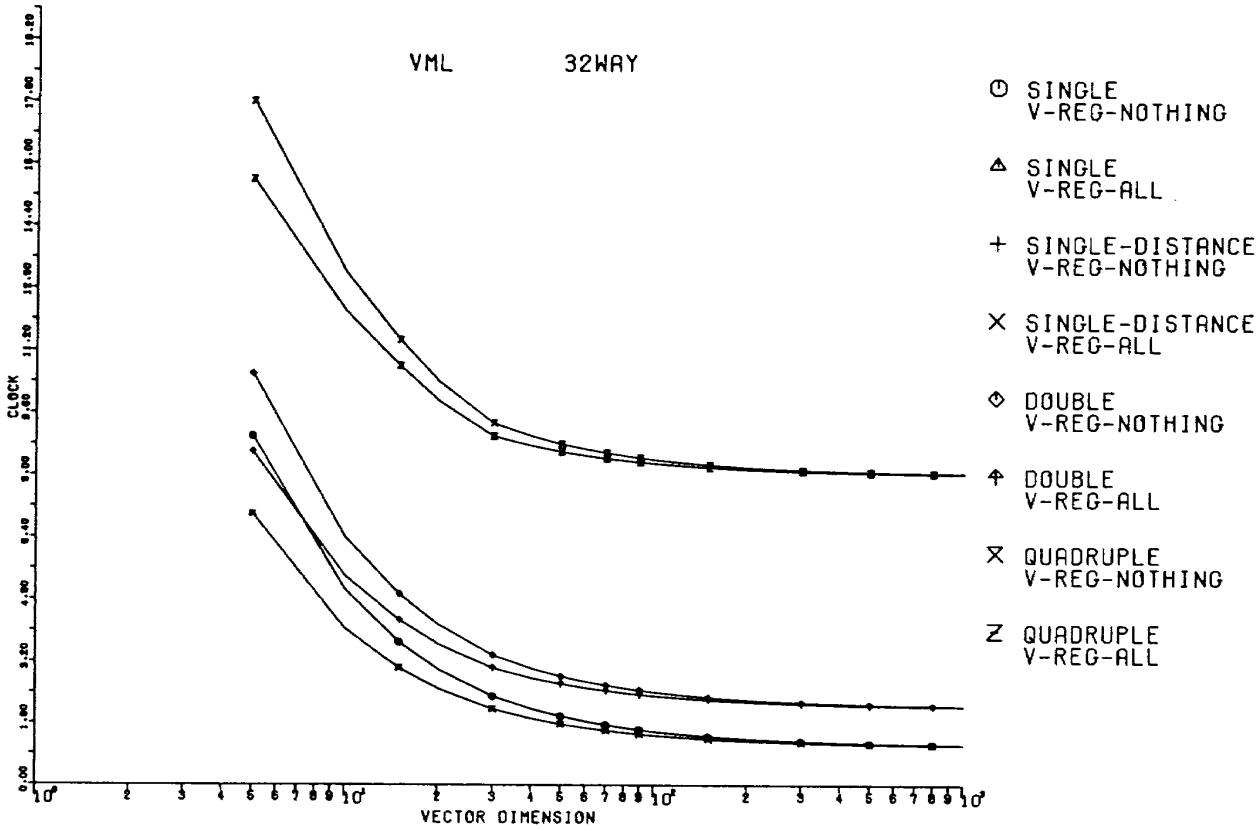


図 4.3.3

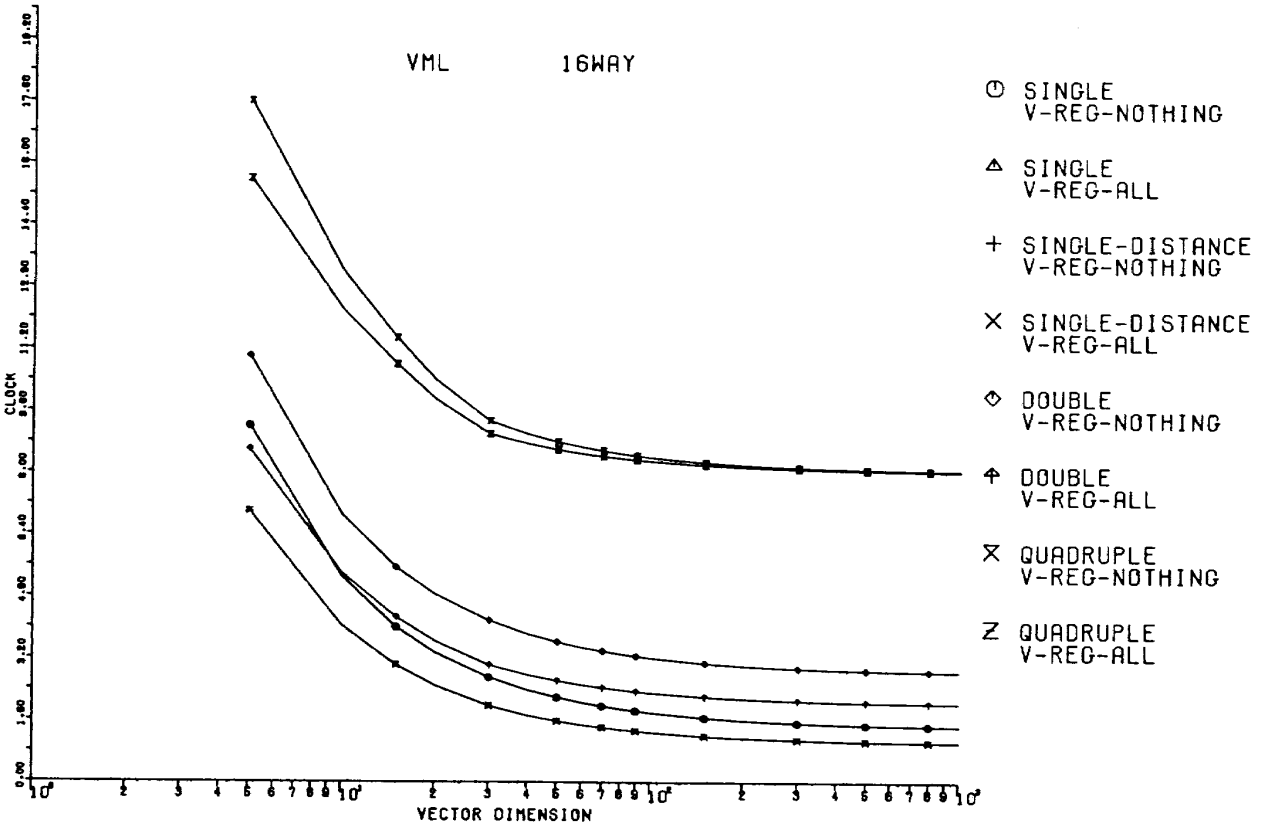


図 4.3.4

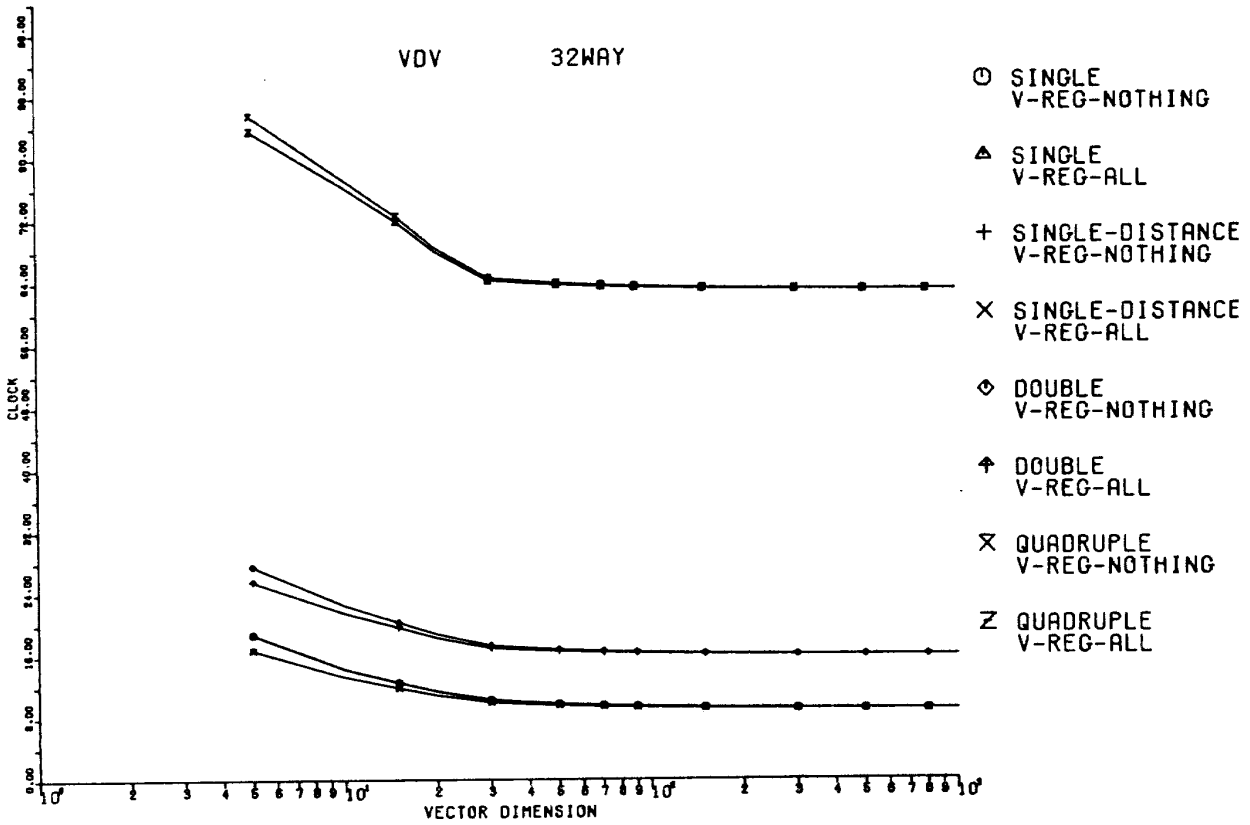


図 4.3.5

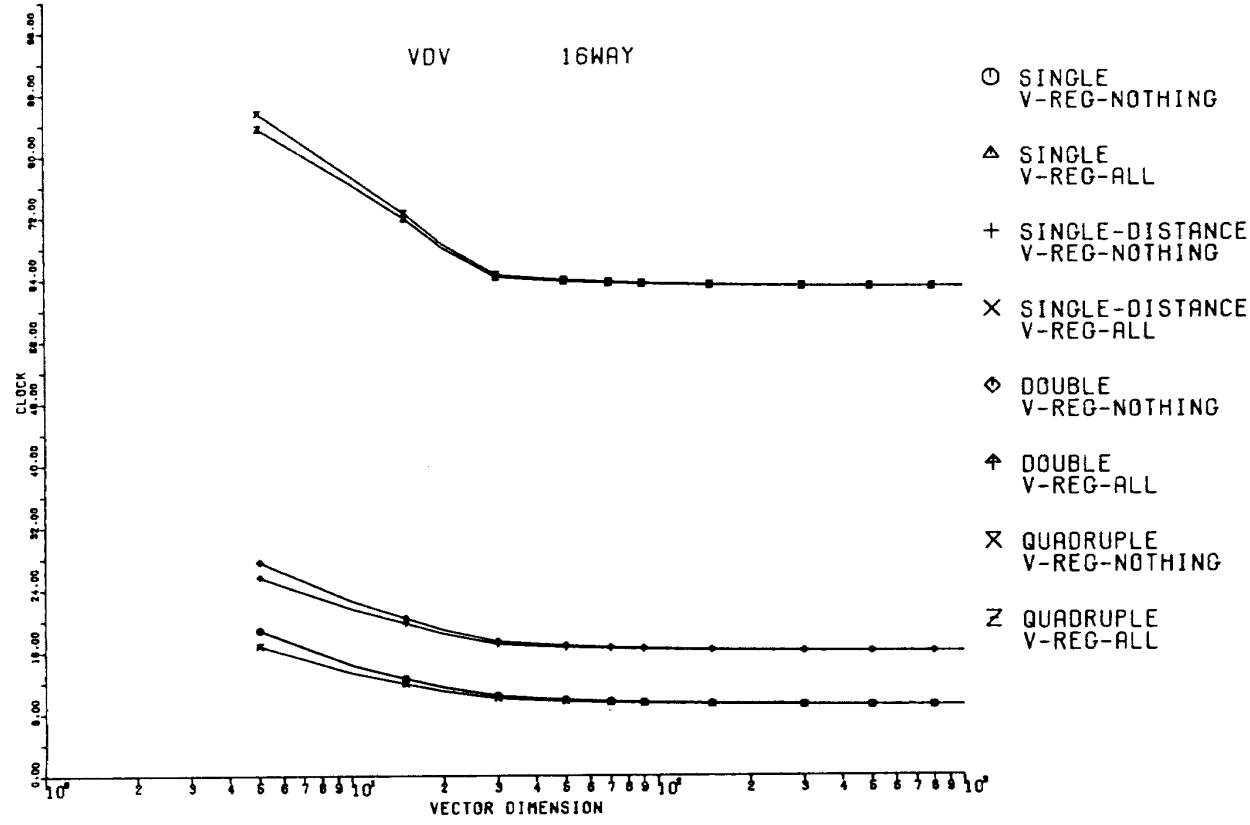


図 4.3.6

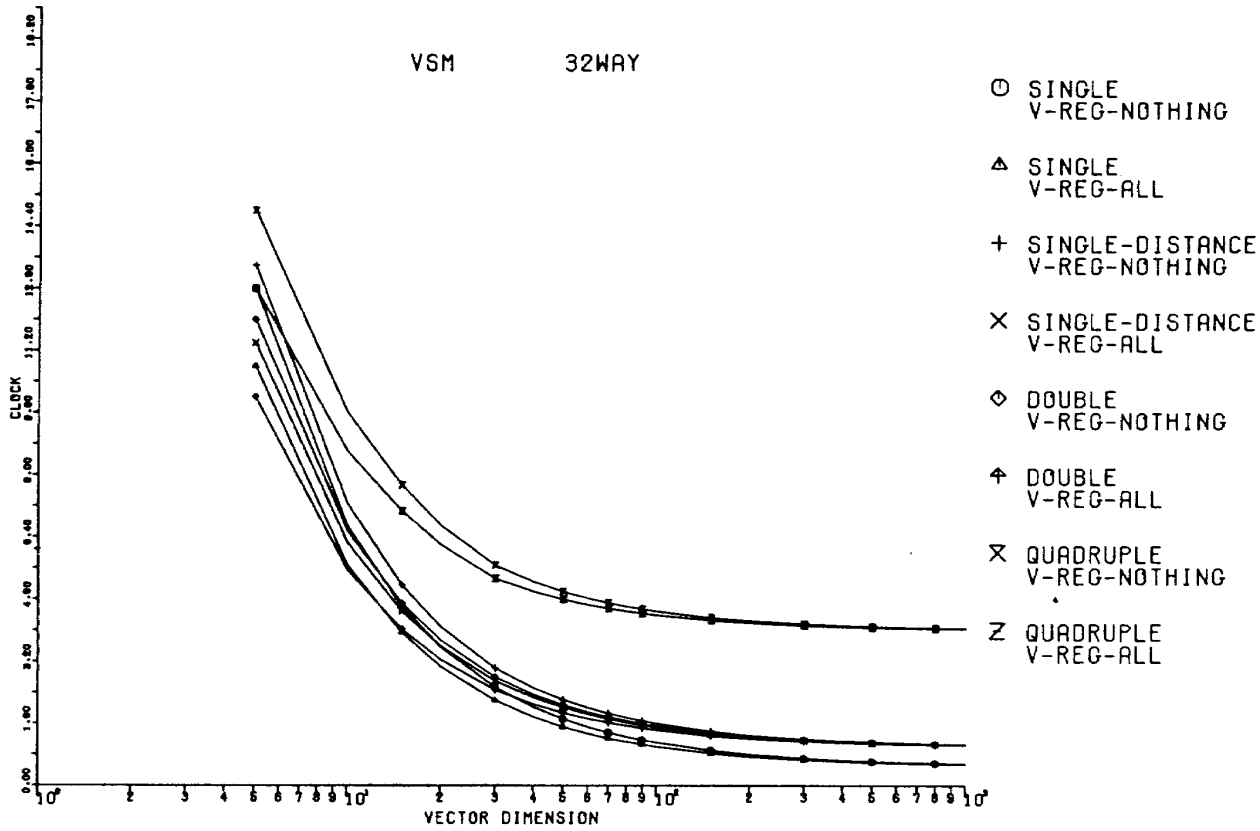


図4.3.7

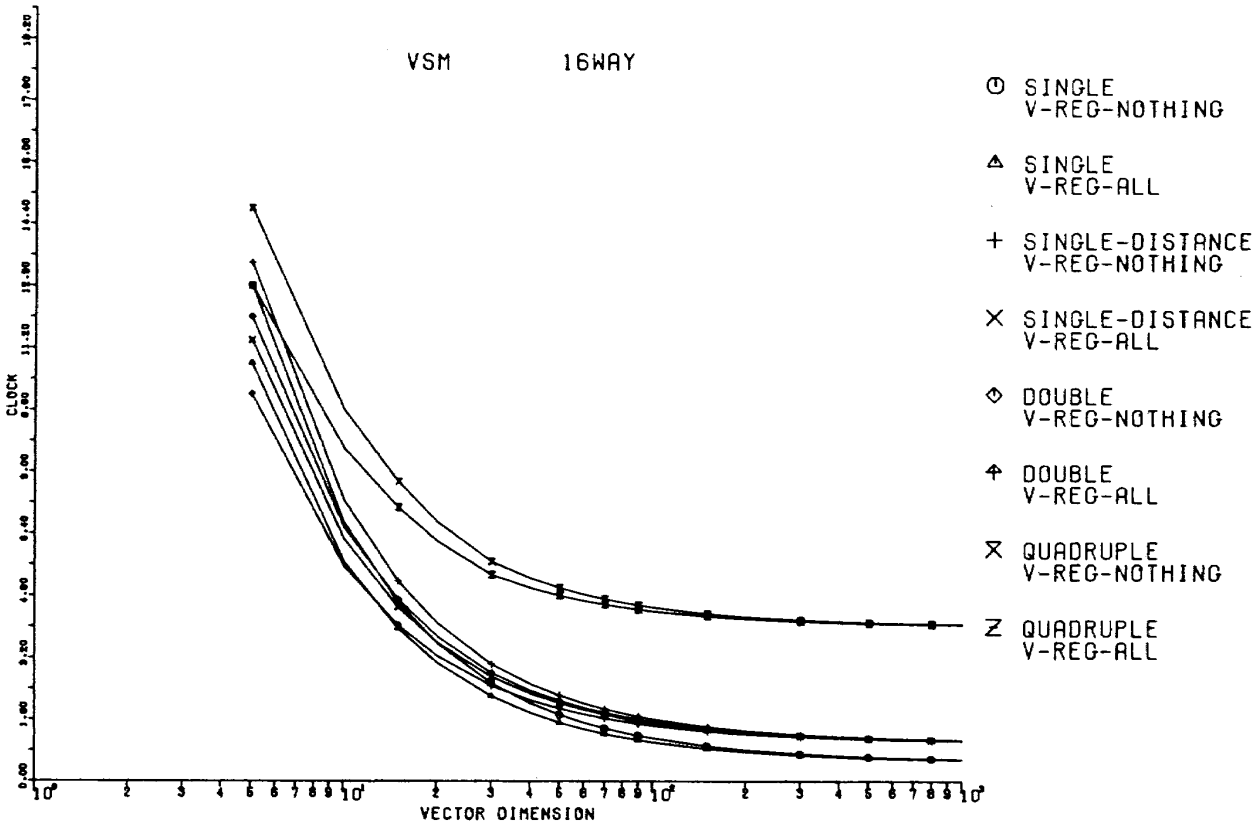


図4.3.8

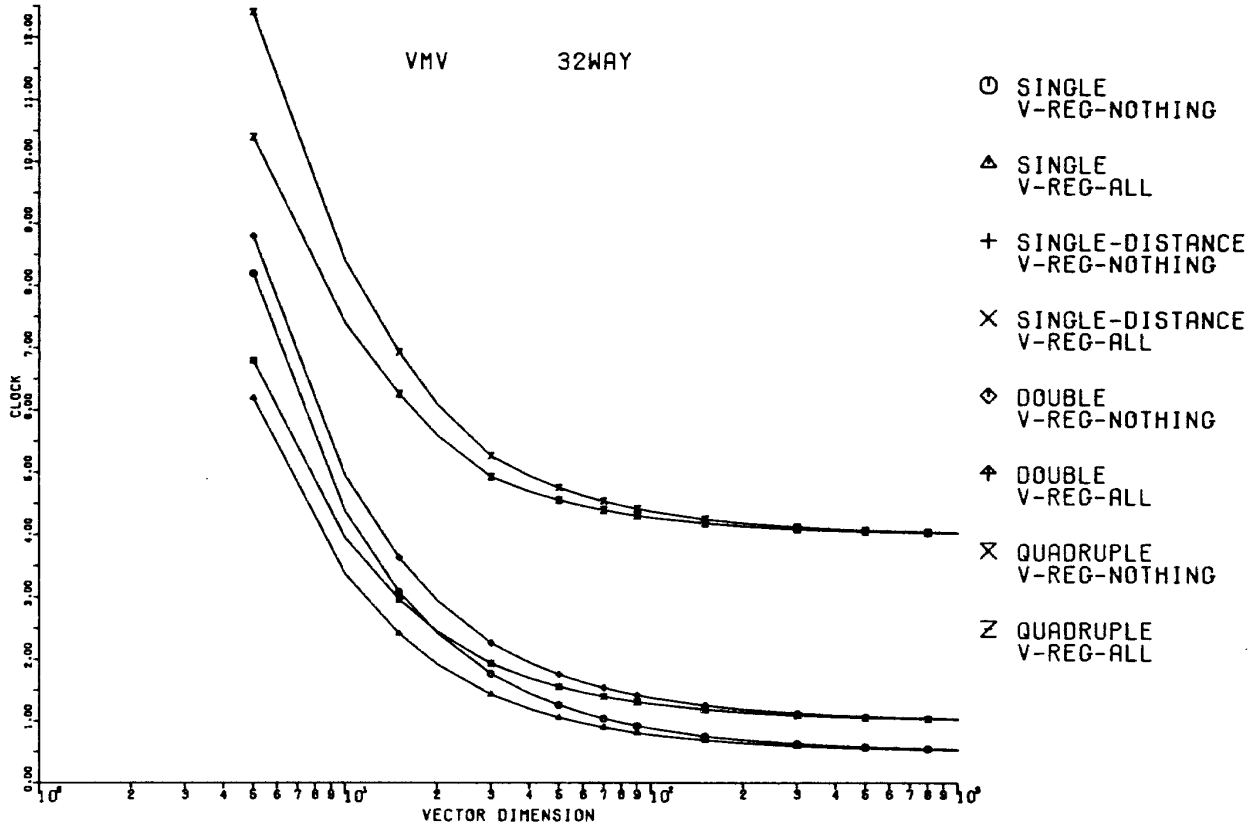


図 4. 3. 9

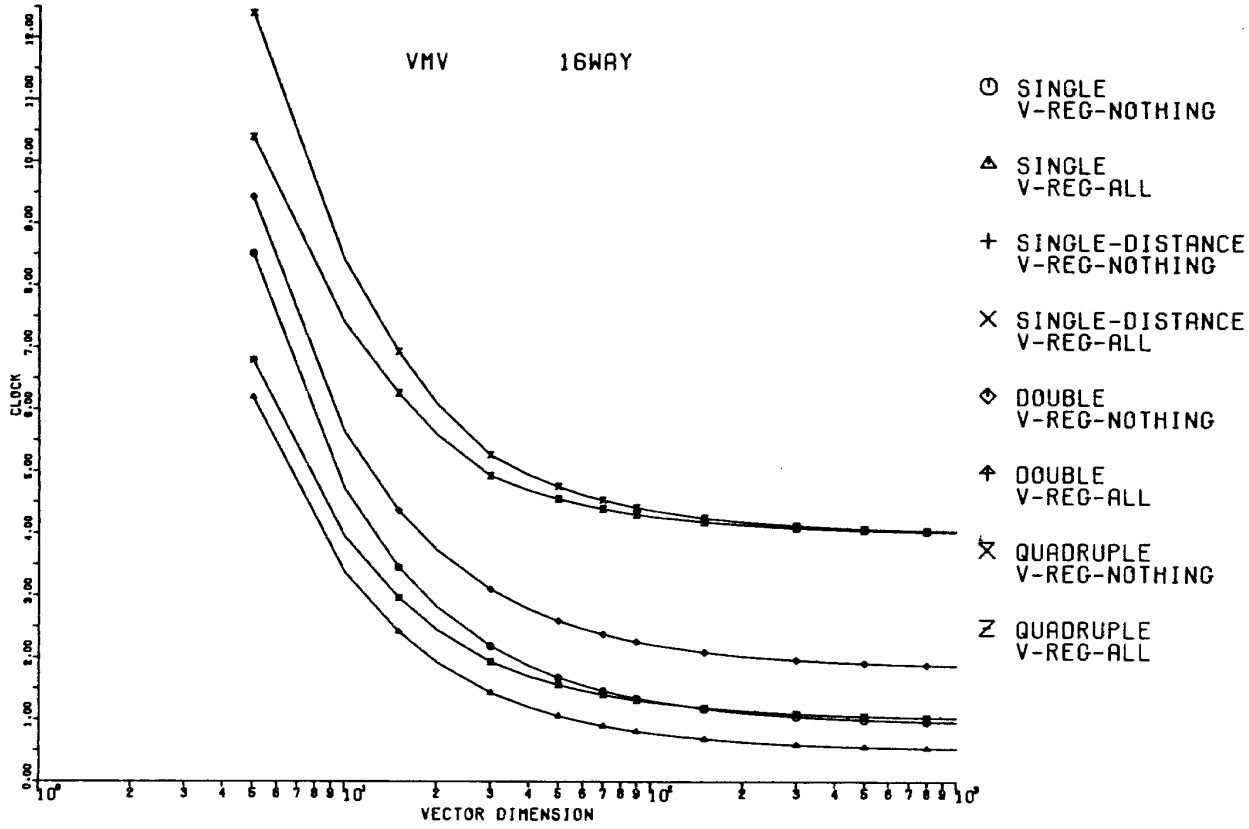


図 4. 3. 10

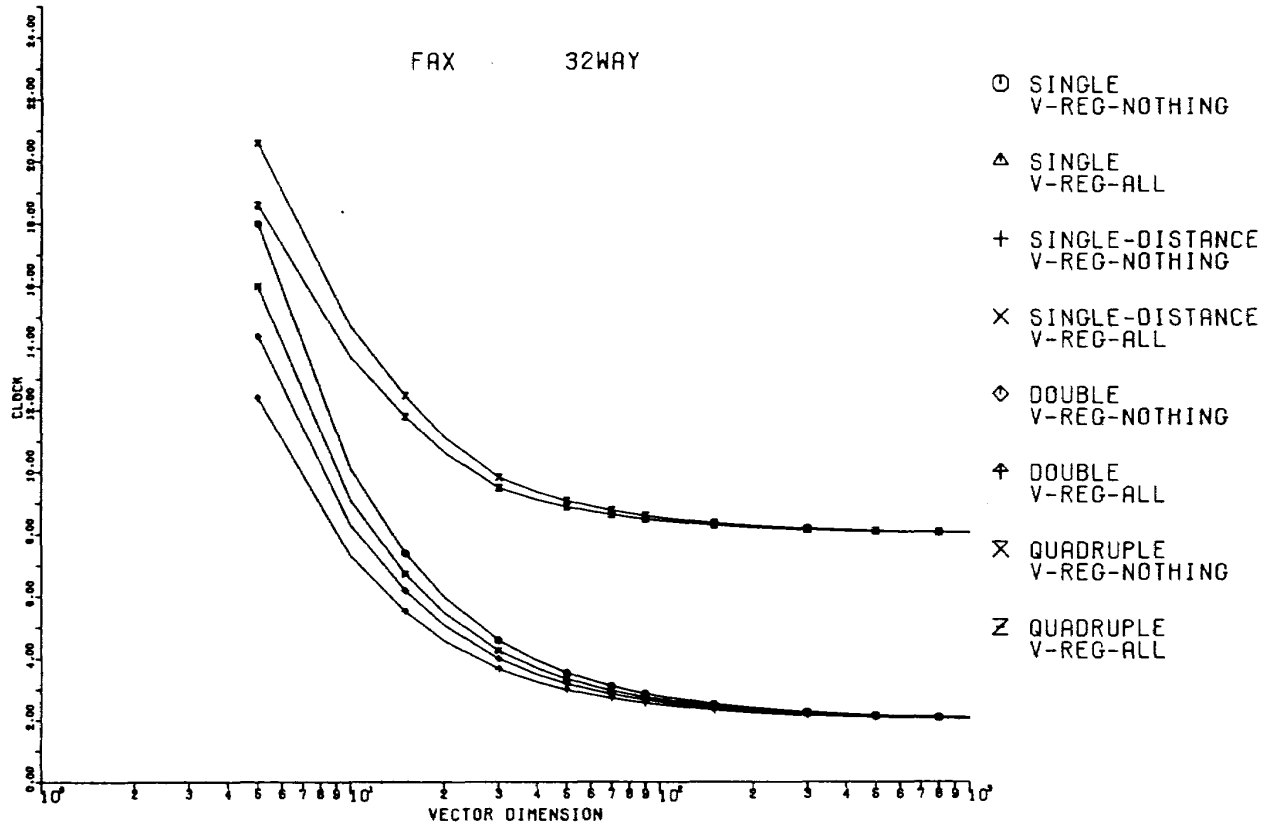


図 4.3.11

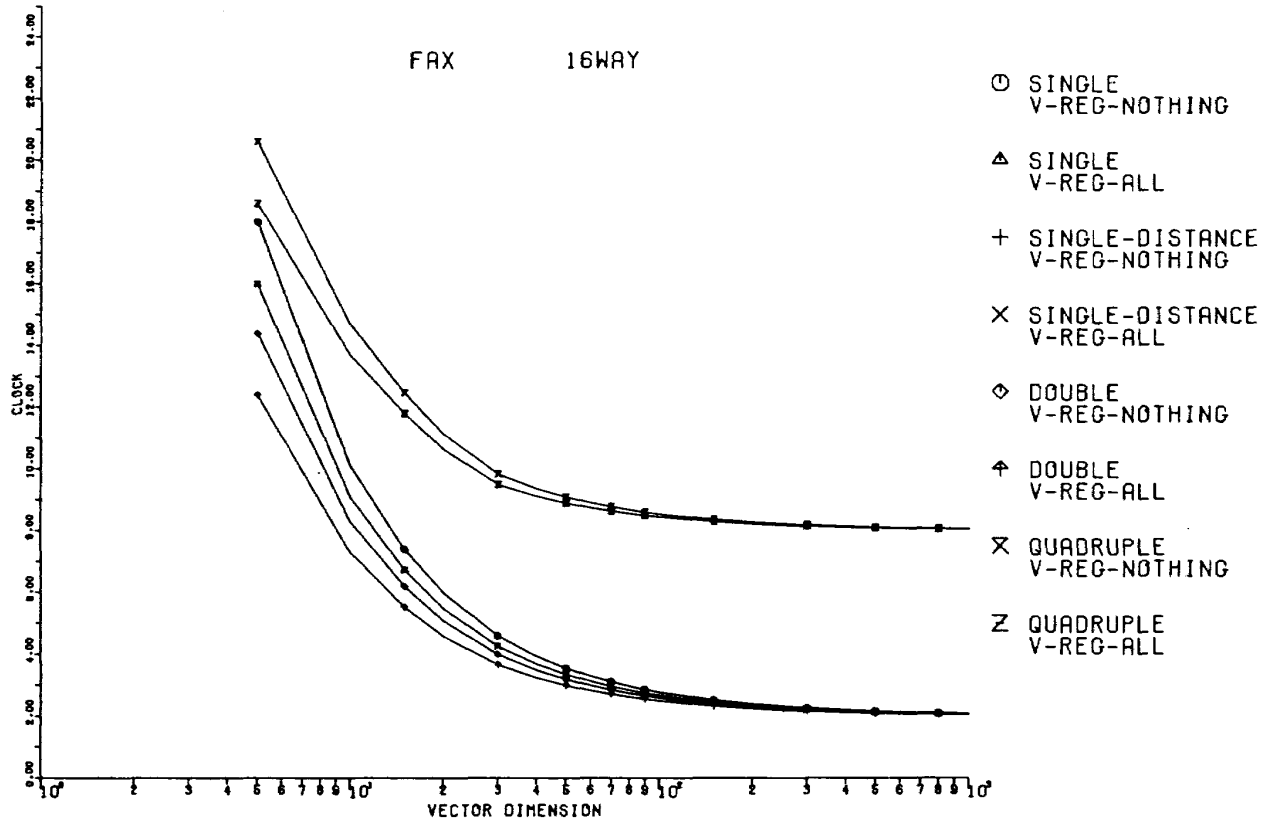


図 4.3.12

5. FACOM−APのハードウェア実行速度の総合性能

第4章ではFACOM−AP のV型命令の種々の環境条件のもとでの実行速度の算出法を示し，各V型命令の実行速度曲線を求め，それについて論じ，さらに230−75 CPUおよびAPでV型命令を使用しないで計算を行なった場合との比較を行なった。この章ではFACOM−APのハードウェアの総合的な実行速度の算出を行う。総合的な実行速度の算出とは実行速度が種々の条件に左右される場合，種々の条件の出現確率を定めそれに基ずいた確率的期待値を求めることに他ならない。汎用機の場合この期待値は個々の命令の出現確率を算出することにより得られ，種々の命令の出現確率が提案されているが，もっとも通常なのはギブソン・ミックスといわれるものであろう。FACOM−AP の場合，V型命令に関しては命令自体の実行速度が環境条件により大きく変化するため汎用機の場合の様に簡単でなく，最終的にはジョブミックス迄考慮しなければならなくなるがそれでは余りにも一般性を欠くという理由と第1章で述べたこの資料の目的からはずれることにもなるので実行速度の期待値はベクトルオペランドの長さNと語長およびインターリーブ数（16WAYと32WAY）の関数として求めることにする。V型命令とM.R 型命令は別に論ずる。

5.1 V型命令の平均命令実行速度

（4.1.1）式～（4.1.4）式により

$$\tilde{T}_e = g(N, P, w, SE, VR, INS) \qquad (5.1.1)$$

と表現されることがわかった。ここで

N ；ベクトルオペランドの長さ

P ；語長（単精度．倍精度，4倍精度の別）

w ；インターリーブ数

VR ；ベクトルレジスタにあるオペランドの数

SE ；命令の続き方

INS；命令の種類

である。この節の目的はINS, VR, SEを確率変数とする期待値を求め

$$T_e = \tilde{g}(N, P, w) \qquad (5.1.2)$$

を求めることである。

(1) 各種V型命令の出現確率の決定

命令の出現確率に関してはギブソン・ミックス（以下G.M と略称する。）を基礎において考える。またAPの実行速度は通常の汎用機を基準において考えることにする。その根拠は

(a) V型命令は汎用機でこれを実行する場合には複合命令であってAPの実行速度は最終的には汎用機換算で求められるべきである。

(b) APの実行速度を汎用機と比較する必要がある。

以上2点である。

i) 先ず基本演算である加減乗除について考える。加減算：乗算：除算のG.M における出現比率は73：40：16である。ベクトル演算でA_i/B_iの出現する確率は極く極く小さいと考える。A_i/a（aはスカラ）もベクトル除算であるが，これはベクトル除算の極端に遅いことを考えれば$\frac{1}{a}\cdot A_i$と乗算にすべきであって本質的なベクトル除算の出現確率はとるに足らないものであってその殆んどは乗算に廻ると考えて良い。そこでこの比率を73：54：2とする。加算のパイプラインは乗算のパイプラインの2倍の速度をもっているので以下この2つのパイプライン速度の使用頻度確率という考え方に基礎をおくことにする。

ii) AVG，AJM，VNM，IPD，CVA，CVM，PMM，SMVA，PLY の9つのV型命令はそれぞれVAD，VML，VSM 命令に分解できる。また汎用機でこれらの命令と同じ内容を実行させる場合もその様に分解する。VADとVSM は加算パイプラインを使用しており，λの値もそれに等しい。上記9つの命令を分解したときVAD，VSMとVML の出現の比率は1：1であるがAJM，AVG は加算パイプラインの速度で実行される。CVA，CVM，PMMは複合命令の複合の形をしており，加算，乗算両方のパイプラインを使用しているが，1ベクトル当り（表3.2.1では長さNのベクトルをm回計算する様になっている。）の演算時間は乗算パイプラインのそれにほぼ等しいので単純V型命令に分解した場合のそれぞれのパイプライン速度は乗算パイプラインの速度の半分，即ち，加算パイプラインの速度に等しいと考えて良い。又IPD，VNMはVML，VSM に分解されるがこの実行速度も乗算パイプラインの速度に等しく分解して実行した時の速度は加算パイプライン速度に等しいと考えてよい。以上の考察から上記9つの複合命令を基本命令に分解した時のそれぞれのパイプライン速度の出現比率を73：54にとってもAP の速度の過大評価にはならない。また複合命令を基本命令に分解して考える事は複合命令で実行する方が一般にずっと速いのでこれもAP 速度の過大評価をさける意味から許される。

iii) FIND−ADRESS，FIND−RELATION 関係のV型命令は7つあり，その出現確率は小さいと考えられるが不明である。しかし汎用機の速度との比較という観点からは第4章第2節，第3節に示した様にこれらの命令に関してはAP の実行速度は汎用機に比較して速いのでこの出現確率を無視することは汎用機との速度比較

という立場だけで考えるならば安全サイドで評価するという観点からさしつかえない。

iv) VMV 命令は汎用機の STORE, TRANSFER 命令に当り汎用機でのこの命令の出現確率は，8％強である。又COMPRE 関係の命令は2つあり，この命令に対応する汎用機の命令の出現確率は4％である。* V型命令としてこれ等の命令の出現確率はこれよりづっと小さくなると考えられる。VMV の速度は加算パイプラインの速度であり，COMPRE 命令の速度は乗算パイプラインの速度である。これら命令の出現確率が2：1の割り合から極端にはずれることなく小さくなると仮定すれば加算パイプラインと乗算パイプラインの使用頻度比率73：54 を変える必要はないであろう。

v) VMD, VMM, LSM, LSR, LDF, ONCの出現確率は無視しうると考える。

以上 i)～v)の議論よりV型命令ミックスとして加算：乗算：除算＝73：54：2を使用することにする。iii), iv) の議論の根拠が若干弱い様に思はれるが大すじでは安全な比率であると考える。また汎用機との比較という点で考えるならばiii), iv) の議論の成立する根拠は十分であると考える。

(2) 命令シークエンスに関する比率

問題となるのは命令の続きがV型－V型およびM型，R型－V型：V型－M型，R型の比率である。この比率は使用者のプログラムの書き方に依存するが，AP を意識しないで書かれたプログラムでも平均5：1位にとって良いと考える。これはかなり安全な比率であってV型命令は出来るだけまとめて使用する習慣が使用者につくこと，およびAP－フォートランコンパイラーの最適化がこのことに留意していることを考慮すれば10：1程度にとっても良い様に思はれる。**

(3) V－Reg にV型命令のオペランドが乗る確率

V－Reg は使用者から見えるレジスタであるので上手な使い方をすればこの確率は可成り高く出来ると考えられる。またV－Reg の使用をコンパイラーにまかせた場合，その割り付アルゴリズムはLOCAL－割り付けを主体としたものとなり，この効率はこれから考えるGLOVAL－割りつけより良くなると考えられるが，初めの2つはともに問題と使用者のプログラム技術に依存しすぎて一般的な考察の対象にならないこと，および安全サイドの原則からGLOVAL－割りつけの基礎にたって議論を展開することになる。

V－Reg は1792単精度語ある。これにV型命令のオペランドが0，1，2，3個乗る確率を求める。そのために

$\tilde{N}=N\times l$; $l=1$　単精度

$l=2$　倍精度

$l=4$　4倍精度

Nはそのプログラムで使用されるベクトルの平均長

$P=1700/\tilde{N}$

92語の残りは出現頻度の高い定数に割り付けるとする。するとP個のレジスタに対するオペランドの割り付けの問題となる。議論を進めるためにいくつかの仮定をおく。

(i) プログラムで使用されるデータは局所的である。

(ii) 局所的に使用されるベクトルの本数は50本程度である。

(iii) 50本のベクトルは出現頻度から見て10本づつにグループわけできる。

(iv) グループ毎の出現頻度は小さい方から大きい方へ指数的に増加する。

すると50本のベクトルが出現する局所的なプログラムでV型命令のオペランドがV－Reg に乗る確率VRPは

$$VRP=\int_{(50-m)/10}^{5} a^x dx \Big/ \int_0^5 a^x dx \quad (5.1.1)$$

$$m=\mathrm{MIN}(50, P) \quad (5.1.2)$$

となり一つ一つのV型命令のオペランドがV－Reg に乗る事象を独立事象と考えると（これを仮定(V)とする。）

(a) V－Reg にオペランドが1つも乗らない確率；$(1-VRP)^3$

(b) V－Reg にオペランドが1つ乗る確率；$3(1-VRP)^2 VRP$

(c) V－Reg にオペランドが2つ乗る確率；$3(1-VRP)VRP^2$

(d) V－Reg にオペランドが3つ乗る確率；VRP^3

$a=1.5$ とすると

表 5.1.1

事象 \ $\tilde{N}$	30	50	70	100	200	300	500
(a)	0	0.00	0.02	0.08	0.29	0.45	0.62
(b)	0	0.05	0.17	0.31	0.44	0.41	0.32
(c)	0	0.31	0.43	0.42	0.23	0.13	0.06
(d)	1.0	0.64	0.37	0.19	0.04	0.01	0.00

$a=2.0$ とすると

表 5.1.2

事象 \ $\tilde{N}$	30	50	70	100	200	300	500
(a)	0	0	0	0.02	0.08	0.16	0.29
(b)	0	0.01	0.06	0.18	0.32	0.40	0.44
(c)	0	0.17	0.34	0.44	0.42	0.34	0.22
(d)	1.0	0.82	0.59	0.37	0.18	0.10	0.04

---

* この出現頻度はG.Mによる。

** 本資料では5：1とした。

となる。$a=2$ とすると最出現頻度グループは最低グループの32倍出現することになり，$a=1.5$ とするとこの値は7倍程度となりこちらの方が実際に近いと思われる。実際問題での例と比較すると（通常の紙上コーディングしたものをAP フォートランで紙上コンパイルしたとして）

表 5.1.3

問 題	$\tilde{N}$	ベクトル本数	VRP
イ	200	28	0.6
ロ	50	25	0.5
ハ	300	72	0.6
ニ	300	16	0.6

表 5.1.4

事象＼問題	(イ) $\tilde{N}=50$	(ロ) $\tilde{N}=200$	(ハ) $\tilde{N}=300$	(ニ) $\tilde{N}=300$
(a)	0.125	0.06	0.06	0.06
(b)	0.375	0.29	0.29	0.29
(c)	0.375	0.43	0.43	0.43
(d)	0.125	0.22	0.22	0.22

表 5.1.1，5.1.2，5.1.4 を比較すると問題(ロ)，(ハ)，(ニ)は $a=2$ の場合にむしろ近い。(イ)は $a=1.5$，$a=2.0$ のいずれともはずれているが $\tilde{N}=50$，ベクトル本数 25 で $50\times25=1250$ で全部V−Reg に乗るはずであってコーディングの手法とコンパイルの原理が合わなかったと考えるべきである。$a=1.5$ と安全サイドで VRP を取ることにすればこの節の目的であるV型命令のオペランドがV−Reg に乗る確率はプログラムのベクトルの平均長と語長が定まれば決まることになる。この様な VRP の定め方にはいくつかの批判が考えられる。それを列挙してみる。

(i) V−Reg にオペランドが0，1，2，3と乗る事象は独立事象ではなく，通常，それ以前に実行されたV型命令に依存する。

(ii) スカラ○ベクトルはV型命令であって，この型のV型演算は可成り多いので $P=1700/\tilde{N}$ とするのは不当であり P はずっと大きくなる傾向にある。

(iii) VRP の算出をGLOVAL−割り付けの基礎に行なっているが実際は局所的な割り付が加味されており，これを併用すれば VRP はもっと大きくなる。

(iv) 局所的なプログラムエレメントにおけるベクトルの本数を50本としたのでは少なすぎる場合があるしエレメント毎にV−Reg をロードするのは全V−Reg のロードに 990 T_u かかることを考えると損である。

(v) 出現頻度に指数法則を適用するのはおかしい場合が実際問題でも沢山ある。

批判の(i)〜(iii)はAP にとって有利な批判であって，実際にこの批判はあたっているが安全サイドの原則からあえて無視する。(iv)の批判について50本という数はディスクリプタ（普通のベクトルでデータレジスタ4語を使用する。）の数とデータレジスタが256個ということからきめたものである。また局所的なプログラムで50本という数は決して少ない数ではないし，50本より多い場合でも殆んどの場合さらにエレメントを細分すれば良く，その際V−Reg を全部クリアする必要はなく除々に移しかえて行けば良い。その様なプログラムこそ使用者は局所割り付の原理を問題の全体的観点から見て行うべきであってコンパイラにV−Reg の使用をまかせることはこの様な場合行うべきではない。(v)の批判は批判が指数法則に向けられたものであればそれ程本質的なものではない。問題になるのはベクトルの出現頻度を低い順に並べて一番高いものが1番低いものの7〜8倍程度現れ，中間的な順位のベクトルは2−3倍出現するということであって，これを満足するならば a^x の代りに ax でも ax^2 でもかまわないのである。指数法則を仮定したのはAP にに適した解法と考えられるディスクリート法では局所的にはベクトルの出現頻度は指数法則的になるものが多いというだけの理由である。この法則にもっとも反すると思われる行列の消去法では一度に 100〜1000本あるいはそれ以上の本数のベクトルが局所的に現われるがその本数を N とすると局所的には主役となるベクトルは唯の一本であってピボッティングを行なわない場合にはその出現頻度は他のN−1本が1に対して 2N−2 回現われる。この様な極端な場合でも本節の VRP に基ずいて行なった推定実行速度は実際コーディングを行なった実行速度よりも遅い値を与える。

V型命令のオペランドがV−Reg に乗る確率を求めるためのもっとリファインされた数学モデルはいくらでも作れるし，そこから VRP を求めることはそのモデルのシミュレーションにより可能であろうがシミュレーションのための統計データが存在しない現在では多小精度は悪く，安全サイドにかたむきすぎるきらいはあるが本節の議論はおおむね妥当なものと考える。

以上(1)，(2)，(3)の議論により式（5.1.2）の T_e を計算したのが図 5.1.1 であって N を横軸に対数座標でとり，縦軸を単位 T_u として普通の座標でとった。語長 P は単

精度，倍精度，4倍精度．インターリーブ数は16と32で計6本の曲線が画かれている。これらの曲線群はインターリーブ数と語長を定めた時の各ベクトル長に対するFACOM-APのV型命令のG.Mに対応するものを与える。図5.1.1において単精度の場合$N=50$迄，倍精度の場合$N=20$迄，4倍精度の場合も$N=20$迄32WAYの場合と16WAYの場合の実行速度が同じであるのはデータ供給能力の問題に原因があり単精度，倍精度に関してはV-Regによる効果，4倍精度の場合は実行速度が遅いことによるデータ供給に対する要求量の少なさとV-Regによる効果がきいている事を示している。16WAYの場合，最高の実行速度は$N<\infty$で達成されておりその値は単精度で$N=150$の時$T_e=1.33T_u$であり，倍精度では$N=80$の時$T_e=2.65T_u$であり，4倍精度では$N=100$の時$T_e=7.51T_u$である。32WAYの場合，APの実行速度はNの単調減少関数であり単精度，倍精度，4倍精度に対して$N=1000$でそれぞれ$T_e=0.96T_u$，$1.87T_u$，$6.63T_u$である。16WAYの場合Nが有限な所でAPの実行速度が最高に達するのはデータ供給能力に対するV-Regの寄与によるものである。この図からNが比較的小さい所ではV-Regの利用に関してプログラム技術上の注意をはらうことの必要性が明らかとなる。また不必要な長精度計算は主記憶の有効利用の面からも実行速度の面からも行うべきでない。このことは特に4倍精度計算についていえることである。

5.2 M型およびR型命令の平均命令実行速度

M型およびR型命令の平均命令実行時間をT_eとおくと

$$T_e = G_0 + G_R + G_{NFP} \qquad (5.2.1)$$

により定まる。ここでG_0は全てのデータがバッファメモリ上にあると考えた時のG.M値，G_Rは命令シークエンスにおいて前後の命令でのデータレジスタのぶつかり合による遅れ時間，G_{NFP}はデータがバッファメモリ上にない場合の遅れ時間である。G_0の値を計算するために各命令の出現確率はLoad，Add，Subtruct；25%，Store；8%，Branch（無条件）；17.5%，Branch（条件付成功）；3.9%，Branch（条件付不成功）；2.6%，Compre；4%，Multiply；0.6%，Divide；0.2%，Shift；4.6%，Logical 1.7%，Index（メモリ）；3.8%，Index（Immediate）；15.2%，Float Add；7.3%，Float Multiply；4.0%，Float Divide；1.6%である。従って表3.2.1.aより

$$\begin{aligned} G_0 &= 303\ \text{ns} \quad \text{単精度} \\ &= 324\ \text{ns} \quad \text{倍精度} \\ &= 456\ \text{ns} \quad \text{4倍精度} \end{aligned} \qquad (5.2.2)$$

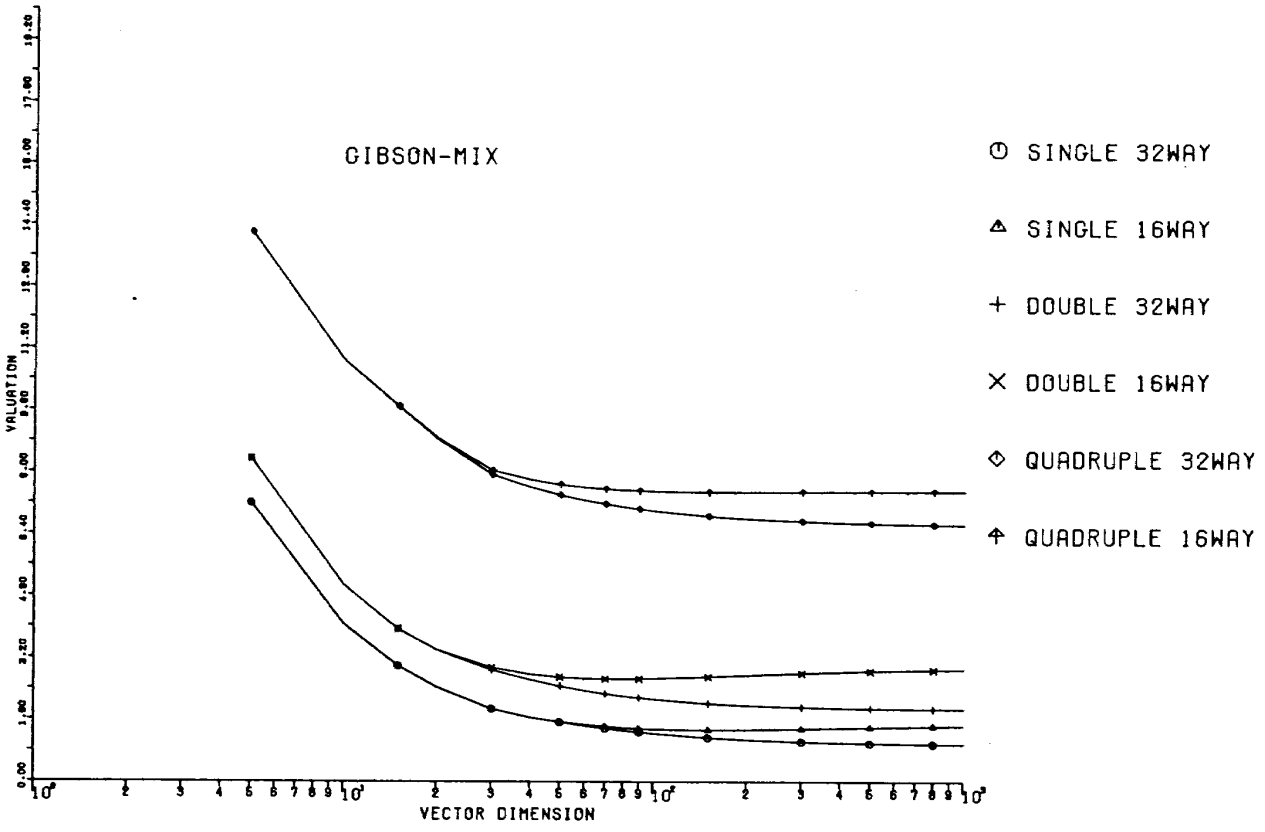


図 5.1.1

G_R＝12 ns（富士通(株)内部資料），G_{NFP}は情報がバッファメモリにない確率を5％とすると85 ns（富士通(株)内部資料）従って式（5.2.1）より

$$T_e = 400\,\text{ns} = 4.44\,T_\mu \quad \text{単精度}$$
$$= 421\,\text{ns} = 4.68\,T_\mu \quad \text{倍精度} \qquad (5.2.3)$$
$$= 553\,\text{ns} = 6.14\,T_\mu \quad \text{4倍精度}$$

これでFACOM−APのM型およびR型命令の平均命令実行時間は定まったわけであるが，この決め方にはいくつかの問題がある。

(1) FACOM−APには汎用（演算）レジスタが256個ありその効果が考慮されていない。

(2) 浮動小数点演算は大部分がV型演算に廻ると考えられるのでG.M中の浮動小数点演算の出現確率をけずる必要がある。

(3) APのR型命令は3−アドレス方式であるにもかかわらずその影響を考慮していない。

(1)はAPに有利な材料で256個のデータレジスタはオブジェクトプログラムのロード．ストア命令を少なくする。その減少の程度はプログラムの性質により可成り異なるが，航技研が現計算機を採用する際に用いたベンチマークテストの例をしらべるとあるプログラムではレジスタの数が1，4，6と増大する場合の実行命令数の変化は16，15，15であるのに対し，別のプログラムではレジスタ数が1，4，11と増大するに従い実行命令数は31，26，20と変化する。

(3)は(1)と逆の方向に働く，即ち$a=b+c$を実行する場合，汎用機ではLoad b，Floating Add c，Store aの3命令であるが，APではLoad b，Load c，Floating Add a，b，c，Store a（メモリにaをSTORE する必要があれば）となり，いずれの場合でもLoad命令が1つふえることになる。常にこの状況が実現したとするとG_0の値は単精度23.22 ns，倍精度で34.83 ns，4倍精度で58.05 ns増す必要がでてくる。

(3)と(1)の寄与の大小に関する統計データは存在しない。また3−アドレスと2−アドレス方式の科学技術計算に関する有利，不利の問題も未だ手のついていない問題である。一般的にいえることはプログラムを局所的にみて中間変数を含めた変数の数が多いプログラムではレジスターの数の多さは威力を発揮するということである。ここでは(3)と(1)の効果は相殺と考える。

(2)はAPに有利な材料であってFloating演算の50％70％，90％がV型演算に廻ったときのG_0の変化を下表に示す。

表5.2.1

精度＼条件	(a)	(b)	(c)
単精度	257 ns	240 ns	222 ns
倍精度	268 ns	246 ns	224 ns
4倍精度	334 ns	285 ns	237 ns

(a)は浮動小数点演算50％減
(b) 〃 70％減
(c) 〃 90％減

この結果T_eの値も変化するが，75 CPUを含めてこれ迄の結果を表にまとめると下表のとおりである。

表5.2.2

精度	75−CPU	AP(M&R)	AP(M&R) (a)	AP(M&R) (b)	AP(M&R) (c)
単精度	272 ns	400 ns	354 ns	337 ns	319 ns
	3.02 T_μ	4.44 T_μ	3.93 T_μ	3.74 T_μ	3.54 T_μ
倍精度	294 ns	421 ns	365 ns	343 ns	321 ns
	3.27 T_μ	4.68 T_μ	4.06 T_μ	3.81 T_μ	3.57 T_μ
4倍精度	566 ns	553 ns	431 ns	382 ns	334 ns
	6.29 T_μ	6.14 T_μ	4.79 T_μ	4.24 T_μ	3.71 T_μ

図5.1.1および表5.2.2によりFACOM−APのV型命令およびM＆R型命令に関するハードウェアの総合的実行速度の算出ができた。

6. 結　　言

本資料により明かにされたのは以下の点である。

(1) AP出現の必然性とAPの高速性の理由。

(2) FACOM−APのハードウェア構成の概略。

(3) FACOM−APの主要なV型命令の種々の実行条件のもとでの処理速度とFACOM−230−75との処理速度の比較。

(4) (3)に基ずいて，FACOM−APを使用する場合，使用者がプログラム作成上，注意すべきこと。

(5) 汎用機における，G.Mに対応するものを語長，インターリーブ数，V型命令のオペランド長の関数として与えた。

(6) M型およびR型命令に対する，G.Mをいくつかの場合に対して与えた。

(7) (5)，(6)によりFACOM−APハードウェアの総合的処理速度を明かにした。

(8) 緒言の末尾に述べた(2)，(3)の作業の基礎を与えた。

本資料を終えるにあたりFACOM−APのハードウェアの性能に関する資料を提供して下さり，また種々の議論に応じて下さった富士通(株)電算機第1技術部第1設課内田啓一郎氏，ソフトウェア事業部LP部長代理鈴木氏，FA課棚倉由行氏に謝意を表す次第である。

文　　献

1) D.J.Kuck ; ILLIAC−Ⅳ Software and Application Programming IEEE Transaction on computor 1968年 Vol. c−17, No.8, pp. 758～770

2) G.H.BARNES, R.M. Brown, M.KATO, D.J.Kuck, D.L.Slotnick, R.A.Stokes The ILLIAC−Ⅳ−computor ; IEEE Transaction on Computer, 1968年, Vol.c−17, No.8 pp. 746～757

3) 加藤満左夫, 苗村憲司 ; 並列処理計算機 オーム社 1976年

4) 元岡達編 ; 計算機システム技術 オーム社 1973年

5) W.C.HOHN & P.D.Jooes ; The control Data STAR−100 paguing station control Data corporation

6) Control Data corporation ; The Control Data STAR−computing system General Refference manual

7) Control Data corporation ; CDC Cyber 76 Handbook

8) IBM ; IBM System 360 and System 370 Model 195 Functional characteristics

9) 富士通(株) ; アレイプロセッサ・ハードウェア解説書

10) H. Lorin ; ハードウェアとソフトウェアにおける並列処理 ; 産業図書

11) D.R. Chapmann, H.Mark & H.W. Pirtle ; Computers VS. Wind Tunnels AIAA. Journal 1975

12) 富士通(株) ; FACOM−230−75 ハードウェア解説編Ⅱ, Ⅲ

航空宇宙技術研究所資料325号

昭和52年3月発行

発行所　航空宇宙技術研究所
東京都調布市深大寺町1880
電話武蔵野三鷹(0422)47-5911(大代表)〒182

印刷所　株式会社　共進
東京都杉並区久我山4－1－7(羽田ビル)

Printed in Japan