

富士通スーパーコンピュータのアーキテクチャ

伊藤 幹雄

富士通 本体事業部

富士通のスーパーコンピュータ FACOM VP シリーズ, VP-E シリーズは、その高い性能だけでなく、汎用機並みの使い勝手の良さを目標として開発された。VP アーキテクチャもこの開発思想に沿って開発されており、そのために高速性能の達成と当社の汎用機アーキテクチャとの親和性を狙いとしたアーキテクチャを採用した。

本稿では、これらの開発思想を実現する VP アーキテクチャとハードウェアについて述べる。

The architecture of Fujitsu Supercomputers.

Mikio Itoh

*Development Section 1, Mainframe Div. Fujitsu Limited
1015, Kamikadanaka, Nakahara-ku, Kawasaki, 211, Japan*

The development targets of Fujitsu Supercomputers (FACOM VP Series & VP-E Series) were its easy to use as our general purpose computers, and its high performance.

Based on this concept, we developed VP architecture which has following characteristics.

- (1) An affinity with our general purpose computer architecture (M Series architecture)*
- (2) The architecture which accomplish high performance.*

In this paper, I will describe about the characteristics of VP architecture, and also describe about VP hardware which realize VP architecture.

1. はじめに

スーパーコンピュータ FACOM VP/VP-Eシリーズはその高性能性と汎用機的使用環境を狙って開発された。これにより、VP/VP-Eシリーズは、その高い性能のみならず、消費電力、設置スペース、冷却システム等で汎用機レベルの設置性を実現し、また、汎用機レベルの高信頼性を実現した。

VPアーキテクチャも、その狙いに沿って開発されたものであり、その高性能性とMシリーズアーキテクチャとの親和性に重点においている。本稿ではFACOM VPシリーズのアーキテクチャと、それに基づいて実現されたハードウェアについて述べる。

2. FACOM VPシリーズのアーキテクチャ

スーパーコンピュータ FACOM VPシリーズのアーキテクチャの狙いとして、高性能性の実現と汎用機との親和性の2点が挙げられる。

(1) 高性能性の実現

FACOM VP/VP-Eシリーズは高い性能を実現する為に、以下のアーキテクチャを採用している。

- ベクトルレジスタの採用とベクトルレジスタを介した柔軟な命令のチェイニング機能
- スカラユニットとベクトルユニット間、およびベクトルユニット内の各パイプライン間の並列実行

(2) Mシリーズとの親和性の実現

汎用機アーキテクチャとの親和性を持つ為に、VPシリーズのアーキテクチャとして、以下を採用した。

- Mシリーズ命令にベクトル命令を追加した命令体系
 - 同一の命令カウンタによるMシリーズ命令とベクトル命令の制御
- これらがVPシリーズのアーキテクチャの大きな特徴となっているが、これにより、従来のソフトウェア資産の継承、汎用機からの移行の容易性が得られる。

3. FACOM VPシリーズのハードウェア構成

FACOM VPシリーズのアーキテクチャについて述べる前に、FACOM VP-Eシリーズのハードウェアについて説明する。

図1にFACOM VP-Eモデルのブロックダイアグラムを示す。本モデルはベクトルユニット、スカラユニット、メインメモリ(MSU)、ベクトルメモリ(VSU)およびチャンネル処理装置から成る。スカラユニットには16個の汎用レジスタ、8個の浮動小数点レジスタ等のMシリーズアーキテクチャで規定されるレジスタ類、64KBのバッファストレージおよびスカラ演算器等をもつ。

一方、ベクトルユニットはベクトルレジスタ、マスクレジスタ等のレジスタと乗算・加算/論理演算、加算/論理演算、除算の三本のベクトル演算パイプライン、マスクパイプライン、2本のロード/ストア・パイプラインを持つ。3本のベクトル演算パイプラインはその内の2本が並列動作可能である。その他の総てのパイプラインは並列動作可能である。

MSUは、最大256MB、256WAY構成であり、VU、SU、CHPからアクセスされる。VSUは、VUからのみアクセス可能なメモリであり、最大768MBの容量を持つ。チャンネル処理装置は最大32チャンネルを持つ。

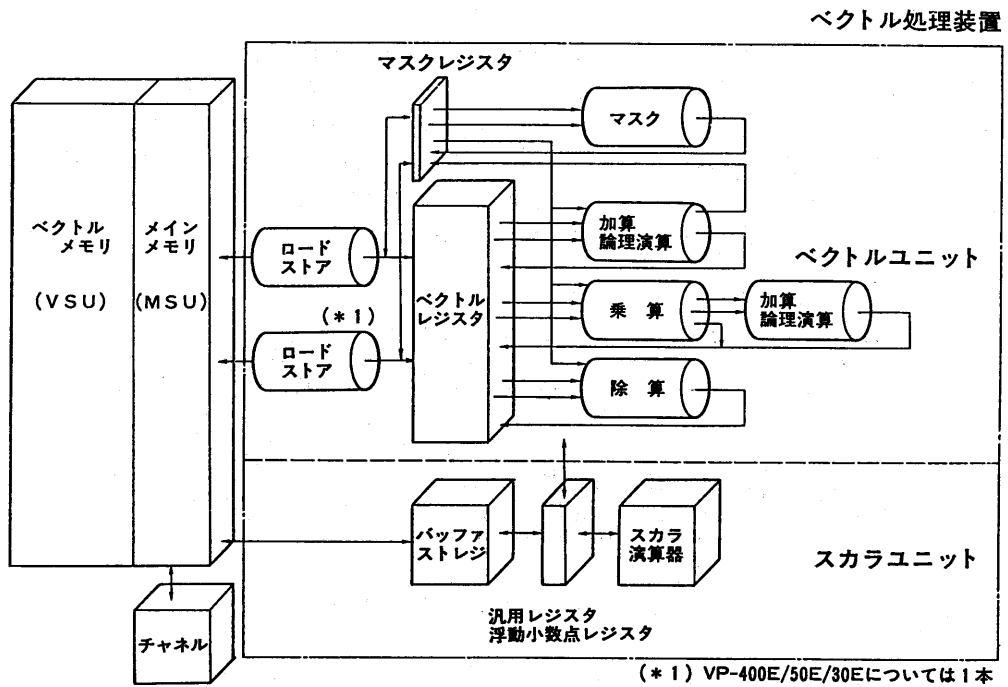


図 1. FACOM VP-Eシリーズのブロックダイアグラム

4. 汎用機との親和性

VPシリーズの命令体系は、スカラー命令として195命令、ベクトル命令として98命令から成る。表1にベクトル命令の種類を示す。

ベクトル命令	98種
Load/Store	14
四則演算 (M&A, M&S, 非正規Addなど)	25
一次リカレンス	3
比較	5
論理演算	13
マクロ (Σ , FIND MAX, FIND INDEXなど)	17
編集演算 (COMPRESS, EXPANDなど)	7
データ変換	5
その他 (制御命令)	9

表1. ベクトル命令

FACOM VPシリーズでは、すべてのMシリーズ命令をサポートしており、

Mシリーズのアーキテクチャを包含している。VPシリーズの命令体系の大きな特徴として、スカラ命令とベクトル命令が同一の命令体系内に存在する事、即ち、スカラ命令とベクトル命令が同一のコード表で表わせる事である。これにより、ベクトル命令はMシリーズの命令セットに追加された単なる1つの命令として扱える。そして、ベクトル命令もスカラ命令も同一の命令カウンタ（IC）で制御可能となり、これによって、ベクトル命令とスカラ命令は同一命令列の中での混在が許される。図2にベクトル命令のフォーマットを示す。

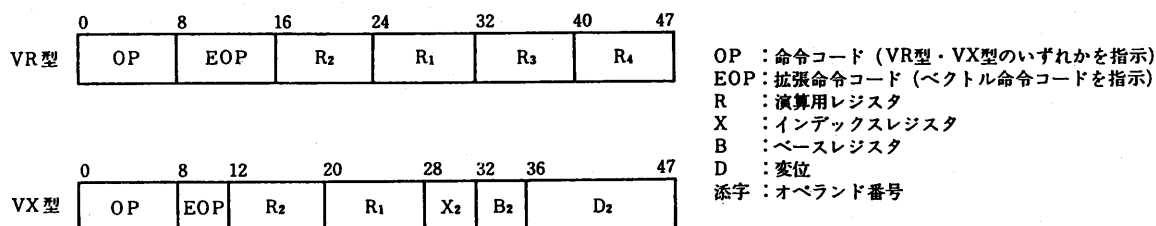


図2. 命令フォーマット

ベクトル命令は6バイト長であり、ベクトル命令形式としては、VR型とVX型がある。VR型命令は、ベクトルロード/ストア系の命令であり、VX型命令はそれ以外の主として演算命令、制御命令等である。ベクトル命令である認識とVR型かVX型かの区別はOP部でなされ、それぞれどんなベクトル命令であるかはEOP部によって示される。VR型命令の、R1、R2フィールドはベクトルレジスタを示し、R1はオペランドレジスタとして、R2はIndirect Load/Store命令の場合の各要素のアドレスとして用いる。B2、X2フィールドはベースレジスタ、インデックスレジスタとして、スカラユニットの汎用レジスタを指定する。一方、RX型命令では、R2、R3フィールドは演算パイプラインへの入力オペランドレジスタを、R1フィールドは演算結果を格納するオペランドレジスタを、R4フィールドはマスクレジスタを指定する。R1およびR2フィールドは、命令によってベクトルレジスタを示す場合とスカラユニット内の汎用レジスタや浮動小数点レジスタを指定する場合があります、R3フィールドはベクトルレジスタを指定する。この様に、本システムはベクトル命令でスカラレジスタが直接にアクセス可能、即ち、ベクトル命令によってベクトルユニットからスカラユニット内の汎用レジスタや浮動少数点レジスタを直接に参照・格納出来るアーキテクチャとなっている。

この様にスカラユニットとベクトルユニット間で非常に密接な関係を取れる配慮がアーキテクチャとして採用されている。即ち、FACOM VPシリーズのアーキテクチャは、Mシリーズ・アーキテクチャとの親和性を強く意識して作られており、これがVPアーキテクチャの大きな特徴となっている。

この様なアーキテクチャの採用は以下の効果がある。

- ハードウェア的にはベクトル命令の制御の一部をスカラユニットで行うために物量の削減が可能
- 制御プログラム(OS)的には、汎用機に近い機能を持ったOSの実

現が比較的容易に開発可能である。

- コンパイラ的には、制約条件が少なく、コード生成が容易である。
- この結果、汎用機に近い操作性と使い勝手を提供出来、また、汎用機からの移行性が容易となっている。

5. 高性能性の実現

汎用機との親和性と並んで、FACOM VPシリーズのアーキテクチャのもう一つの狙いは高性能性の発揮である。そのために幾つかの工夫がアーキテクチャに採り入れられている。それらを説明する前に、FACOM VP-Eモデルのハードウェアの命令制御方式について述べる。図3にVP-Eモデルの命令管理パイプラインを示す。

スカラユニットは、Mシリーズ論理仕様で規定されるPSW(プログラム・ステータス・ワード)内のIC部(Instruction Counter部)で示されるアドレスに基づいてMSUから命令を読み出すが、これはスカラユニットの命令管理パイプラインの命令フェッチ・ステージで行われる。フェッチされた命令はデコード・ステージで解釈され、スカラ命令であるかベクトル命令であるか判定される。スカラ命令であれば次のデータフェッチ・ステージに移り、必要なオペラ

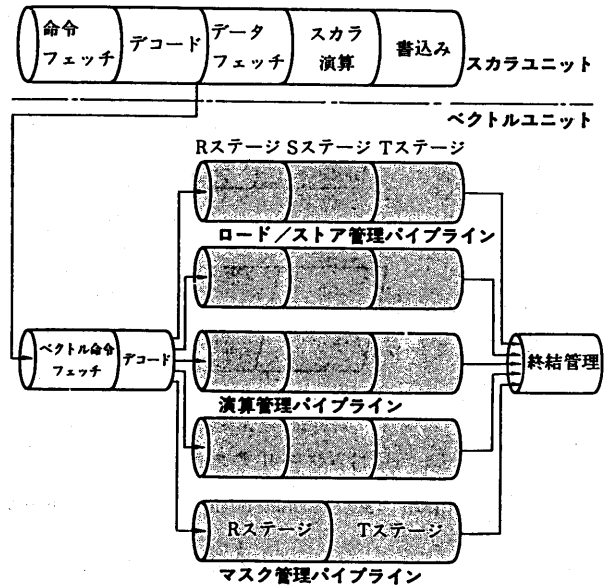


図3. 命令管理パイプライン

ンドがバッファストレージより取り出される。次のスカラ演算ステージでは処理が行われ、書き込みステージで汎用レジスタ(GPR)、浮動小数点レジスタ(FPR)やMSUに結果が格納される。命令デコードの結果、ベクトル命令であれば、命令をベクトルユニットに転送する。この場合にGPR、FPR等のスカラレジスタの内容が必要であれば、それらを読み出してベクトル命令と共にベクトルユニットに転送する。命令がベクトルロード/ストアの場合は、命令の X_2 、 B_2 、 D_2 フィールドを用いてアドレス計算を行い、オペランド・アドレスがベクトル命令と共にベクトルユニットに転送される。このアドレス計算はスカラパイプラインのデータフェッチ・ステージにあるスカラ命令のオペランドアドレス計算用のアドレス加算器を使用している。ベクトルユニットに転送されて来たベクトル命令は、ベクトルユニットのデコードステージで、EOPフィールドが解釈され、どのようなベクトル命令であるかが認識される。命令の種類によって、2本のロード/ストア管理パイプライン、2本の演算管理パイプライン、1本のマスク管理パイプラインの内のどれに投入されるかが判断され、そのパイプ

ラインが空いていれば投入され、処理が実行される。処理が完了すると終結ステージに入るが、ここでスカラユニット内のGPR, FPRへの処理結果の格納が必要ならばデータがスカラユニットに送られ、レジスタへの書き込みが行われる。

この様なハードウェア制御の下で、高性能性を実現するために、以下のアーキテクチャ的工夫が成されている。

(1) スカラ命令とベクトル命令の並列動作とその順序性の保障

VPの高性能性の実現のために、スカラ命令とベクトル命令の並列動作を可能にしている。図4にスカラ命令とベクトル命令の並列動作を図示する。この図で解る通り、先行するベクトル命令の完了を待たずに後続するスカラ命令が完了している。これは、先行するベクトル命令がベクトルユニットに送られた後は、後続するスカラ命令はスカラパイプラインを走ることが出来る。

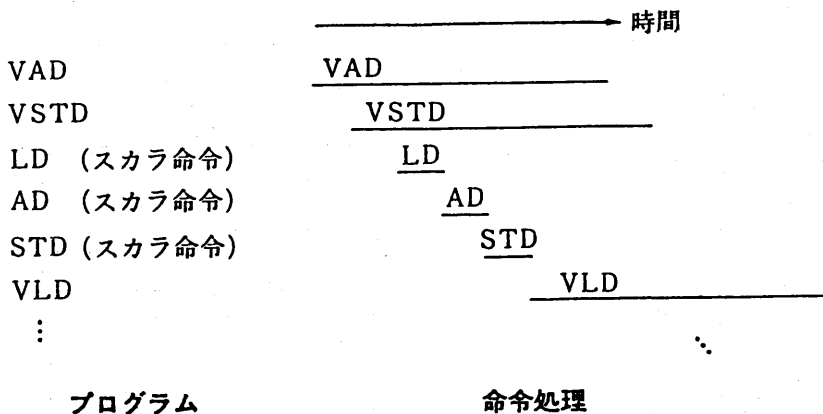


図4. スカラ命令とベクトル命令の並列実行

この場合の第一の問題としては、スカラユニットとベクトルユニットの両者によるスカラレジスタへの参照・更新が競合した場合のアクセス順序の保障である。VPシリーズのアーキテクチャでは、これをハードウェアで保障する事を定めている。即ち、スカラパイプラインのデコード・ステージでスカラレジスタの競合をチェックし、先行するスカラ命令が更新しようとしているレジスタを後続するベクトル命令が参照しようとしている場合は、後続のベクトル命令をパイプライン上でインターロックして、更新が完了するまでスカラレジスタが参照されない様にしている。また逆に先行するベクトル命令が更新しようとしているスカラレジスタを後続するスカラ命令が参照しようとしている場合は、後続するスカラ命令をパイプライン上でインターロックして更新完了までスカラレジスタの参照を行わせない。

もう一つの問題としては、ベクトルユニットとスカラユニットからのMSUの更新・参照順序の保障である。この場合、ハードウェアはPOST命令とWAIT命令という同期命令、即ち、POST命令に先行する命令によるメモリアクセスとWAIT命令より後の命令によるメモリアクセス

の間でアクセスの逐次化を行う機能を用意し、ソフトウェアで参照順序保障を行っている。以下にその方法の例を示す。

…VST 1…VPOST…VST 2…VWAIT…LOAD…

このケースではベクトルストア命令(VST 1)とスカラロード命令(LOAD)の間で参照・更新順序は逐次化されるが、VPOST命令に後続するベクトルストア命令(VST 2)とスカラロード命令(LOAD)では逐次化の保障は無い。

この様にスカラレジスタの更新・参照順序の保障をハードウェアで、そして、MSUアクセスの更新・参照順序の保障をソフトウェアで対処し、効率良いスカラ命令とベクトル命令の並列実行を実現している。

(2) ベクトル命令間の並列実行/追い越しとその順序性の保障

図3で示す様に、ベクトルユニットに送られて来た命令は、5本の管理パイプラインの内的一本に投入される。しかし、例えば、除算命令が続いた場合、管理パイプラインは空いていても、除算パイプラインは一本しか無いので、後続する除算命令を演算管理パイプラインに投入することは出来ない。しかし、例えば、その後に加算命令があった場合は、先行する除算命令を追い越して加算命令を演算管理パイプラインに投入することが出来る制御を行っている。しかし、この場合もスカラ命令とベクトル命令間のスカラレジスタの逐次性の保障と同様に、ベクトル命令間でベクトルレジスタ、マスクレジスタ、スカラレジスタの逐次性の保障が必要である。この場合も同様にハードウェアにて逐次性のチェックを行うアーキテクチャを採用している。すなわち、ベクトルユニットのデコード・ステージでベクトルレジスタ間の競合をチェックし、レジスタアクセスの順序保障を行っている。また、ベクトル命令間のスカラレジスタの競合はスカラユニットで保障している。もう一つの問題として、2本のロード/ストアパイプライン間のメモリアクセス順序を保障する必要がある。この場合もスカラユニット-ベクトルユニット間のメモリアクセスの逐次化と同様に、ソフトウェアによって保障を行っている。ハードウェアはこのために以下の3通りの逐次化機能を提供している。また、この3つの方法は、若番ほど、逐次化の時のオーバーヘッドが大きい。

① POST/WAIT命令

② Indirect Access命令の挿入：Indirect Access命令にベクトルユニット内アクセスの逐次化機能を持たせている。

③ IDによる方法：ベクトルロード/ストア命令にID部を持ち、IDが同じ命令間では実行順序保障を行う。ハードウェア制御的にはIDによりロード/ストア管理パイプラインを固定する。但し、Indirect Access命令にはIDの機能は無い。

これらにより、ベクトルパイプライン間の並列実行/追い越し実行が可能となり、ベクトルパイプラインを有効に使用する事が出来る。

(3) ベクトル・パイプラインのリンケージ

FACOM VPシリーズの持つベクトル命令数は98種と少ない。そして、その殆どがシンプルな命令である。しかし、FACOM VPシリーズのハードウェアは、これらの命令セットを組み合わせた命令列が、複合命令を用意するのと同程度の性能を発揮出来る様に、柔軟なリンケージ

機能を装備している。図5にリンケージの概念図を示す。

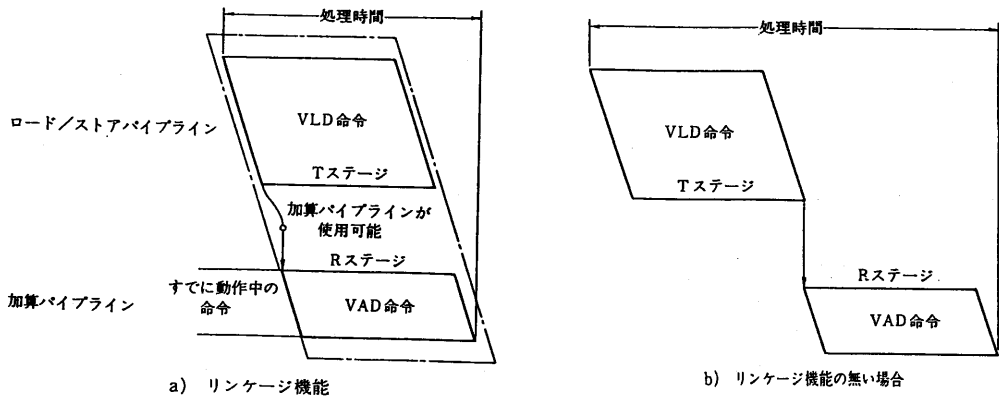


図5. リンケージ機能

図5にて、a)はリンケージ機能が有効なケースである。この場合は、先行するベクトル・ロード命令のデータ格納レジスタが次のベクトル加算命令で使用されている。しかし、VLD命令の各要素はすべての要素の処理が完了しない内に加算パイプラインに送り込まれている。この機能をリンケージという。この場合は仮にLOAD&ADDと言う複合命令を新設した場合と同程度の性能を発揮している。また、リンケージ機能を持たないb)のケースとの実行時間の差に注目願いたい。また、この機能は、ほぼどのタイミングからでもリンケージを開始できる柔軟なハードウェア制御が採用されている。

6. まとめ

以上にFACOM VP/VP-Eシリーズのアーキテクチャについて述べた。このアーキテクチャの開発思想は高性能性と汎用期との親和性の両立を目指している点に、スーパーコンピュータのアーキテクチャとしての特徴がある。今後もこのアーキテクチャを発展させて、これらの長所を伸ばして行きたいと思う。

7. 参考文献

- 1) FACOM VPシリーズ ハードウェア機能解説書
- 2) FACOM MPシリーズ ハードウェア機能解説書
- 3) "スーパーコンピュータ FACOM VPのハードウェア"
雑誌FUJITSU VOL. 35 NO4
- 4) 内田啓一郎 & 伊藤幹雄 "HIGH SPEED VECTOR PROCESSORS IN JAPAN"
computer physics communications 37(1985) 7-13