

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】<http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

保守/廃止

V20HL™, V30HL™
16/8, 16ビット・マイクロプロセッサ

ソフトウェア編

μPD70108H
μPD70116H

CMOSデバイスの一般的注意事項

①静電気対策 (MOS全般)

注意 MOSデバイス取り扱いの際は静電気防止を心がけてください。

MOSデバイスは強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、NECが出荷梱包に使用している導電性のトレーやマガジン・ケース、または導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。

また、MOSデバイスを実装したボードについても同様の扱いをしてください。

②未使用入力の処理 (CMOS特有)

注意 CMOSデバイスの入力レベルは固定してください。

バイポーラやNMOSのデバイスと異なり、CMOSデバイスの入力に何も接続しない状態で動作させると、ノイズなどに起因する中間レベル入力が生じ、内部で貫通電流が流れて誤動作を引き起こす恐れがあります。プルアップかプルダウンによって入力レベルを固定してください。また、未使用端子が出力となる可能性 (タイミングは規定しません) を考慮すると、個別に抵抗を介してV_{DD}またはGNDに接続することが有効です。

資料中に「未使用端子の処理」について記載のある製品については、その内容を守ってください。

③初期化以前の状態 (MOS全般)

注意 電源投入時、MOSデバイスの初期状態は不定です。

分子レベルのイオン注入量等で特性が決定するため、初期状態は製造工程の管理外です。電源投入時の端子の出力状態や入出力設定、レジスタ内容などは保証しておりません。ただし、リセット動作やモード設定で定義している項目については、これらの動作ののちに保証の対象となります。

リセット機能を持つデバイスの電源投入後は、まずリセット動作を実行してください。

CP/M, CP/M-80, CP/M-86は米国Digital Research Inc. の商標です。

V20, V30, V20HL, V30HL, Vシリーズは日本電気株式会社の商標です。

- 本資料の内容は、後日変更する場合があります。
- 文書による当社の承諾なしに本資料の転載複製を禁じます。
- 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的所有権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
- 当社は品質、信頼性の向上に努めていますが、半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として、人身事故、火災事故、社会的な損害等を生じさせない冗長設計、延焼対策設計、誤動作防止設計等安全設計に十分ご注意願います。
- 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定して頂く「特定水準」に分類しております。また、各品質水準は以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認の上ご使用願います。
 標準水準：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
 特別水準：輸送機器 (自動車、列車、船舶等)、交通用信号機器、防災/防犯装置、各種安全装置、生命維持を直接の目的としない医療機器
 特定水準：航空機器、航空宇宙機器、海中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等
- 当社製品のデータ・シート/データ・ブック等の資料で、特に品質水準の表示がない場合は標準水準製品であることを表します。当社製品を上記の「標準水準」の用途以外でご使用をお考えのお客様は、必ず事前に当社販売窓口までご相談頂きますようお願い致します。
- この製品は耐放射線設計をしておりません。

巻末にアンケート・コーナーを設けております。このドキュメントに対するご意見をお気軽にお寄せください。

はじめに

対象者 このアプリケーション・ノートは、次の製品の機能を理解し、それらを使用した応用システムを設計するユーザを対象とします。

- μ PD70108H (別名称V20HL)
- μ PD70116H (別名称V30HL)

目的 このアプリケーション・ノートでは、上記の各製品が持つソフトウェア機能の応用方法をユーザに理解していただくことを目的としています。

構成 このアプリケーション・ノートは大きく分けて次の内容で構成しています。

- 概要
- 簡単な画像処理プログラム
- 10進フローティング・ポイント・データ演算パッケージ
- CP/M-80™エミュレーション・プログラム

読み方 このアプリケーション・ノートの読者には、マイクロコンピュータに関する一般的知識を必要とします。

本文中に出てくる各製品名は、次のように読み替えてください。

本文中の製品名	読み替える製品名
V20™	V20HL
μ PD70108	μ PD70108H
V30™	V30HL
μ PD70116	μ PD70116H

各製品の命令機能を知りたいとき

→別冊の16ビットVシリーズ™ ユーザーズ・マニュアル 命令編を参照してください。

各製品のハードウェア機能を知りたいとき

→各製品に対応した別冊のユーザーズ・マニュアル ハードウェア編を参照してください。

各製品の電気的特性を知りたいとき

→各製品に対応した別冊のデータ・シートを参照してください。

- 凡 例
- データ表記の重み : 左が上位桁、右が下位桁
 - アクティブ・ロウの表記 : $\overline{\text{XXX}}$ (端子、信号名称に上線)
 - メモリ・マップのアドレス : 上部—上位, 下部—下位
 - 注 : 本文中につけた注の説明
 - 注意 : 気をつけて読んでいただきたい内容
 - 備考 : 本文中の補足説明
 - 数の表記 : 2進数… XXXX または XXXX_B
10進数… XXXX
16進数… XXXX_H
- 2のべき数を示す接頭語 (アドレス空間、メモリ容量) :
- K (キロ) : $2^{10} = 1024$
 - M (メガ) : $2^{20} = 1024^2$

関連資料

資料名	資料番号
データ・シート	IC-8154
ユーザーズ・マニュアル ハードウェア編	IEU-761
16ビットVシリーズ ユーザーズ・マニュアル 命令編	U11301J

目次

第1章 概要 1

第2章 簡単な画像処理プログラム 2

2.1 プログラム概要 2

2.2 画像データの構成 2

2.3 プログラムの前提事項 3

2.4 画像処理のサブルーチン 3

2.5 TRANSFERサブルーチン 4

2.6 TRANSFERサブルーチン・フローチャート 7

2.7 SCALEサブルーチン 9

2.8 SCALEサブルーチン・フローチャート 11

2.9 ANDIMAGE, ORIMAGE, XORIMAGEサブルーチン 18

2.10 ANDIMAGE, ORIMAGE, XORIMAGEサブルーチン・フローチャート 18

2.11 画像処理プログラム, プログラム・リスト 19

第3章 10進フローティング・ポイント・データ演算パッケージ 41

3.1 プログラム概要 41

3.2 10進フローティング・ポイント・データのフォーマット 41

3.3 プログラムの前提事項 42

3.4 四則演算, 比較のサブルーチン 42

3.5 ワーキング・エリア 44

3.6 共通サブルーチン 44

3.7 共通サブルーチン・フローチャート 46

3.8 VFADDサブルーチン・フローチャート 68

3.9 VFSUBサブルーチン・フローチャート 73

3.10 VFMULサブルーチン・フローチャート 74

3.11 VFDIVサブルーチン・フローチャート 77

3.12 VFCMPサブルーチン・フローチャート 80

3.13 10進フローティング演算パッケージ・プログラム・リスト 81

第4章 CP/M-80TMエミュレーション プログラム 110

4.1 プログラム概要 110

4.2 メモリ・マップ 111

4.3 エミュレーション・セグメントのメモリ・マップ 112

4.4 スタートアップ・プログラムの処理 113

4.5 デフォルトFCBの作成 114

4.6 スタートアップ・プログラム・フローチャート 116

4.7 BDOSコール・エミュレーション・プログラムの処理 125

4.8 BDOSコール・エミュレーションプログラム・フローチャート 126

4.9 CP/M-80 エミュレーション・プログラム・プログラム・リスト 128

第1章 概 要

μPD70108 (別名称V20™)/70116 (別名称V30™)は、最新のCMOSプロセスを採用した16ビット・アーキテクチャのマイクロ・プロセッサです。μPD70108(V20)は、外部のデータ・バスが8ビット、μPD70116(V30)は、外部データ・バスが16ビットになっておりますが、ソフトウェア的にはコンパチブルになっております。

μPD70108(V20)/70116(V30)の命令セットは、従来の製品からの継承性を考慮しながら大幅に拡張されており、その特徴としては、次の4点が上げられます。

- 1. グラフィック処理などに適する ビット操作機能
 - ビット・フィールド操作命令..... EXT, INS
 - 1ビット操作命令..... TEST1, SET1
 - CLR1, NOT1
- 2. ビジネス用ソフトウェアで使用される BCD演算機能
 - BCDストリング加減命令..... ADD4S, SUB4S, CMP4S
 - BCDローテーション命令..... ROL4, ROR4
- 3. 高級言語サポート機能
 - スタック・フレーム操作命令..... PREPARE, DISPOSE
 - 配列インデックス・チェック命令..... CHKIND
- 4. 8ビットCPUのソフトウェアを継承するため使用するμPD8080AFエミュレーション機能
 - エミュレーション・スタート命令..... BRKEM
 - エミュレーション・モード専用命令..... CALLN, RETEM

この資料では、これらの特徴を応用した例として、

- 簡単な画像処理 (移動, 2のべき乗の拡大/縮小, 論理演算).....ビット操作機能
- 10進フローティング・ポイント・データ演算パッケージ.....BCD演算機能
- CP/M-86上でCP/Mトランジェント・プログラムを動作させるプログラム
.....μPD8080AFエミュレーション機能

の説明をします。

第2章 簡単な画像処理プログラム

2.1 プログラム概要

このプログラムは、あらかじめメモリ内にストアされている、画像データを処理し、ビデオRAMにストアするものです。

可能な処理には、平行移動、2のべき乗の拡大/縮小、画像データ間の論理積、論理和、排他的論理和があります。それぞれの処理は、サブルーチン化されており、パラメータとして基データのストアされているアドレス、処理後のデータをストアするアドレス、水平方向のパラメータ、垂直方向のパラメータを受け取り、処理を行ないます。

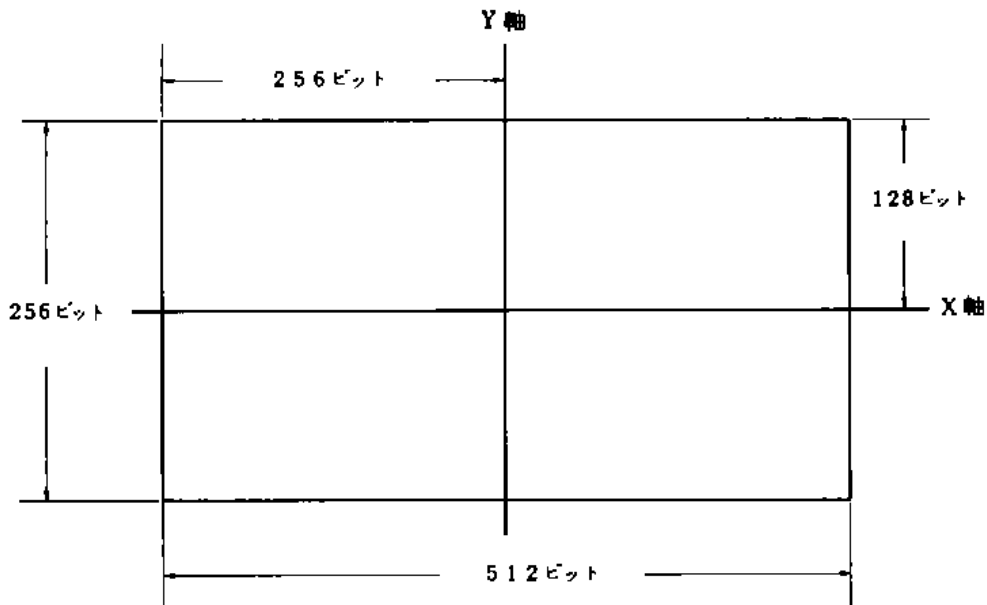
2.2 画像データの構成

このプログラムで扱う画像データは、水平512ビット、垂直256ビットの大きさを持つモノクロのデータで、1ビットで一つの画素が構成されています。

画像データの原点は、水平256ビット目、垂直128ビット目に取りられています。

第2.1図に、画像データの構成を示します。

第2.1図



2.3 プログラムの前提事項

このプログラム例は、基の画像データおよび処理後ストアされるビデオRAM領域とも、同一セグメント内に存在すること、およびサブルーチンを呼び出すメイン・プログラムと各サブルーチンが同一セグメント内に存在することを、前提に作成されています。したがってアドレス情報はすべてセグメント内のオフセットとして扱われています。

2.4 画像処理のサブルーチン

画像処理を行なうサブルーチンには、次のものがあります。

- TRANSFER** 水平方向の移動ビット数、垂直方向の移動ビット数に従い、平行移動した結果をVRAMにストアします。
- SCALE** 原点を中心に、水平方向の拡大/縮小比、垂直方向の拡大/縮小比に従い、拡大/縮小した結果をVRAMにストアします。
- ANDIMAGE** VRAMのデータと基の画像データのANDを計算して、VRAMにストアします。
- ORIMAGE** VRAMのデータと基の画像データのORを計算して、VRAMにストアします。
- XORIMAGE** VRAMのデータと基の画像データのEXORを計算して、VRAMにストアします。
- また、各サブルーチンには、下表で示すようなパラメータを与えます。

サブルーチン名	パラメータ1	パラメータ2	パラメータ3	パラメータ4
TRANSFER	基データの 先頭アドレス	VRAMの 先頭アドレス	水平の移動 ビット数 + 右方向 - 左方向	垂直の移動 ビット数 + 上方向 - 下方向
SCALE	基データの 先頭アドレス	VRAMの 先頭アドレス	水平の拡大/ 縮小比 + 拡大 - 縮小	垂直の拡大/ 縮小比 + 拡大 - 縮小
ANDIMAGE	基データの 先頭アドレス	VRAMの 先頭アドレス	なし	なし
ORIMAGE	基データの 先頭アドレス	VRAMの 先頭アドレス	なし	なし
XORIMAGE	基データの 先頭アドレス	VRAMの 先頭アドレス	なし	なし

各サブルーチンを使用する場合、まず必要なパラメータをスタックにプッシュした後、サブルーチンを呼び出します。サブルーチンから戻った時には、与えたパラメータの分、スタック・ポインタを補正する必要があります。

例

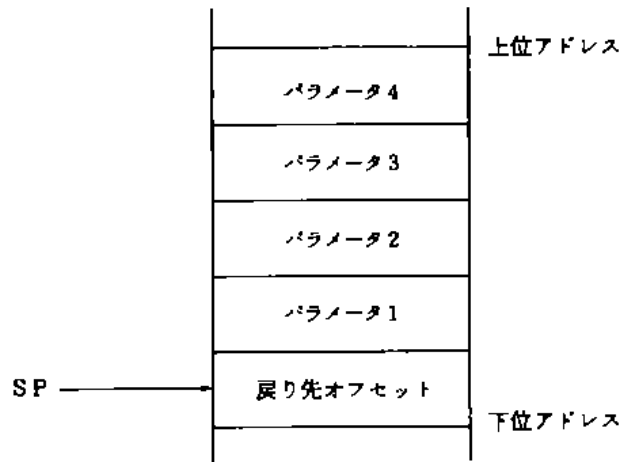
```

PUSH  YOFFSET
PUSH  XOFFSET
PUSH  OFFSET VRAM
PUSH  OFFSET SRCDATA
CALL  TRANSFER
ADD   SP, 8
⋮

```

第 2.2 図にサブルーチン・コール直後のスタックの状態を、示します。

第 2.2 図



2.5 TRANSFERサブルーチン

TRANSFER サブルーチンでは、最初に水平/垂直方向への移動ビット数をチェックし、水平が 512 ビット以上、垂直が 256 ビット以上の場合には、VRAM に転送すべきデータがないので、単に指定された VRAM の領域をクリアします。

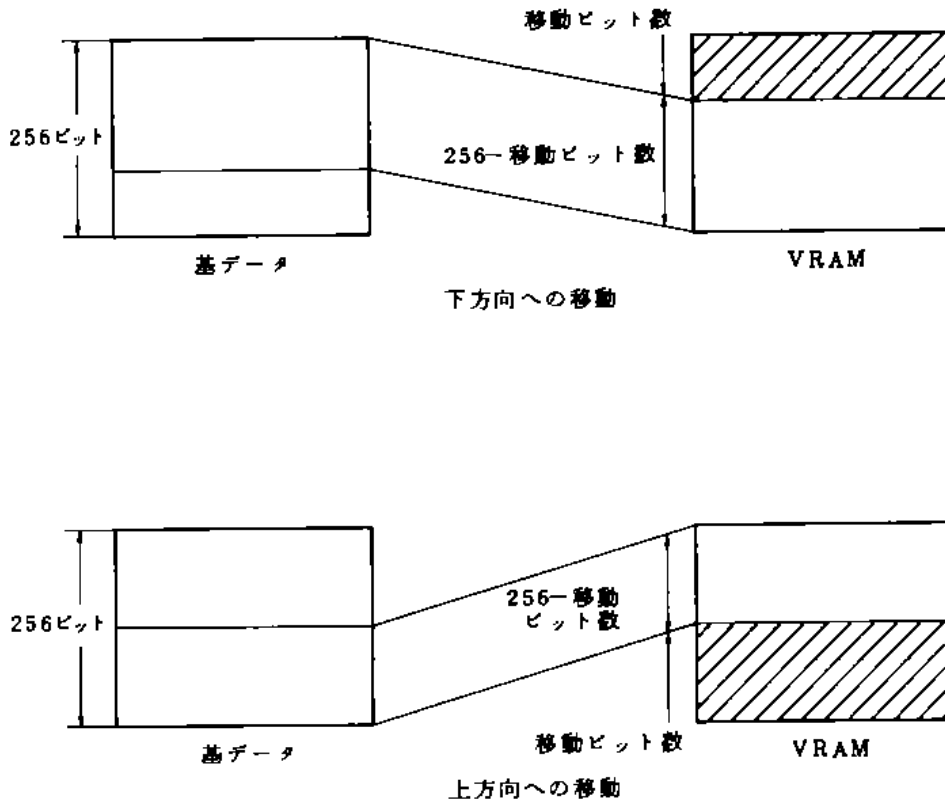
(1) 垂直方向の移動

第 2.3 図に垂直方向の移動の様子を示します。

上方向への移動の場合、VRAM の先頭行が、基データの移動ビット数目の行に相当するため、基データのアドレスを「移動ビット数 × 64」（1 行のバイト数）だけプラスします。その後、水平方向の処理を「256 - 移動ビット数」回だけ行ない、残りの行をクリアします。

下方向への移動の場合、基データの先頭行が、VRAM の移動ビット数目の行に相当するため、VRAM の最初の移動ビット数分の行（移動ビット数 × 64 バイト）をクリアし、その後、水平方向の処理を「256 - 移動ビット」回だけ行ないます。

第 2.3 図



② 水平方向の移動

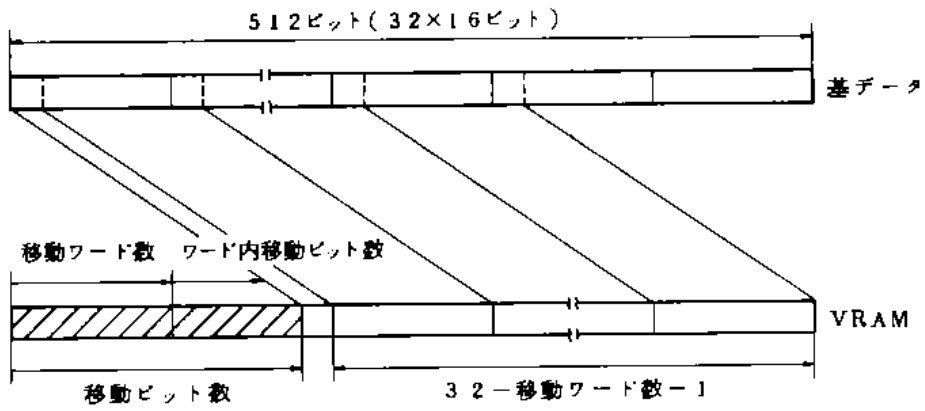
第 2.4 図に水平方向の移動の様子を示します。

水平方向の移動の場合には、移動ビット数を移動ワード数とワード内移動ビット数（16ビット以下）に分割します。

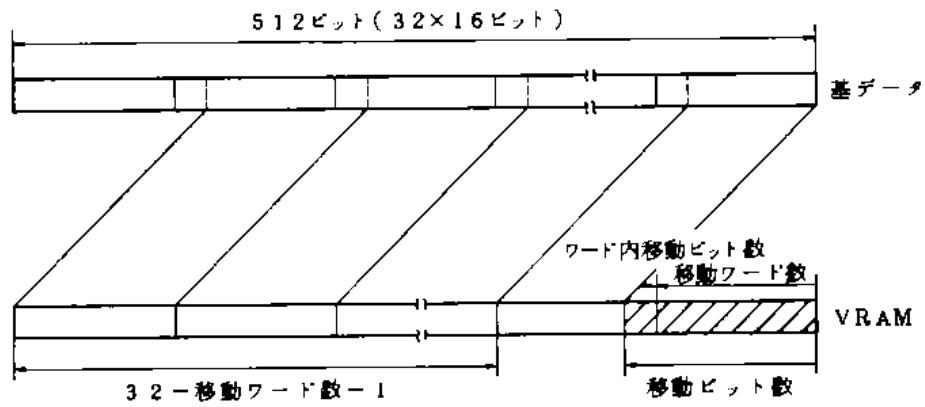
右方向への移動の場合、VRAMの処理中の行の先頭から移動ワード数だけをクリアし、ワード内移動ビット数が0でない時には、EXT命令を使って基データの先頭から“16-ワード内移動ビット数”分のデータを読み出し、ワード内移動ビット数だけ上位にシフトしSTM命令によりVRAMにストアします。その後、EXT命令とSTM命令を使い、ワード内移動ビット数が0の時は、“32-移動ワード数”回、0以外の時は、“31-移動ワード数”回、16ビット単位で転送します。

左方向への移動の場合、基データの処理中の行の先頭アドレスに“移動ワード数×2”をプラスし、ワード内移動ビット数をビット・オフセットとして、EXT命令とSTM命令を使い、“31-移動ワード数”回、16ビット単位で転送します。その後、EXT命令で、ワード内移動ビット数分のデータを読み出し（0の場合は16ビット）、STM命令でVRAMにストアし、1行の残りの部分をクリアします。

第 2.4 図

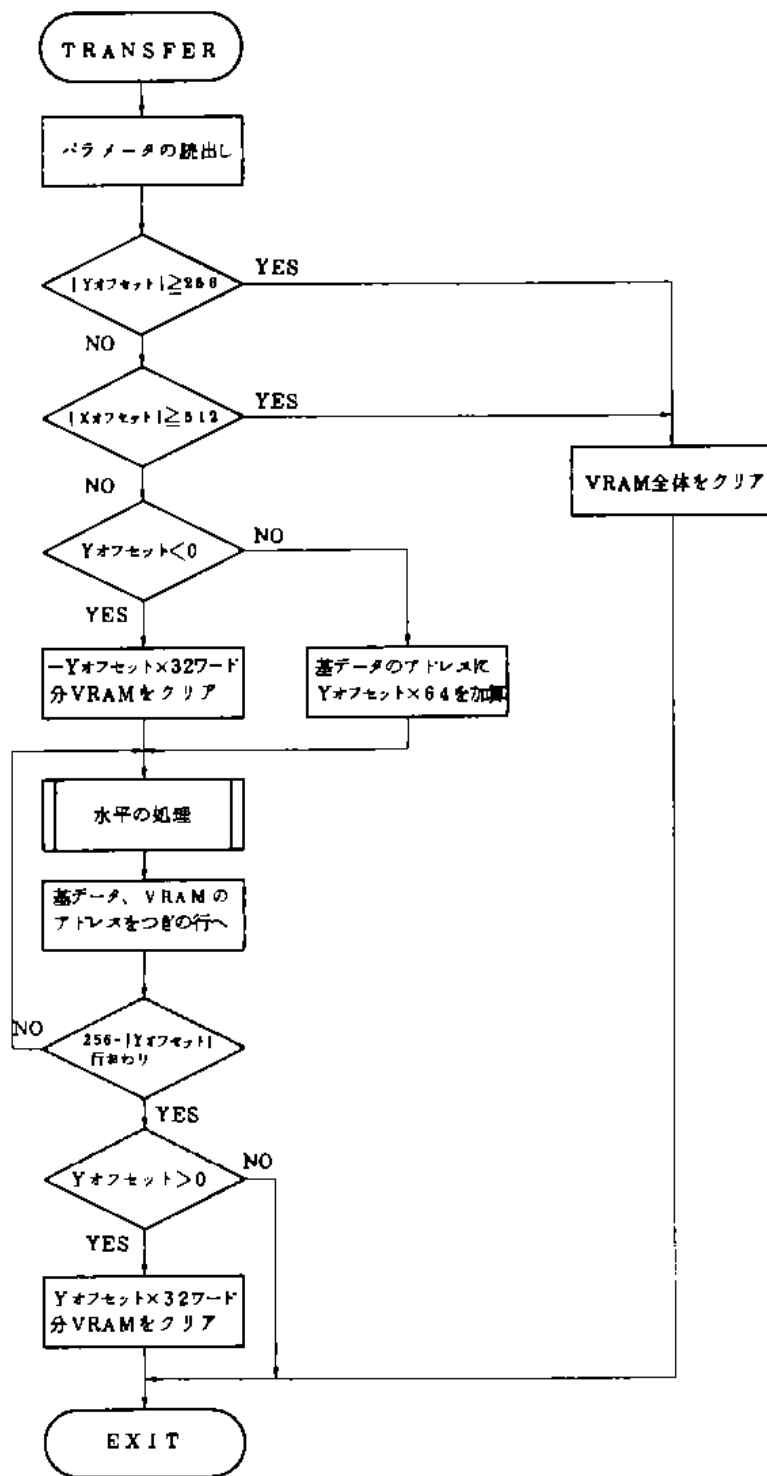


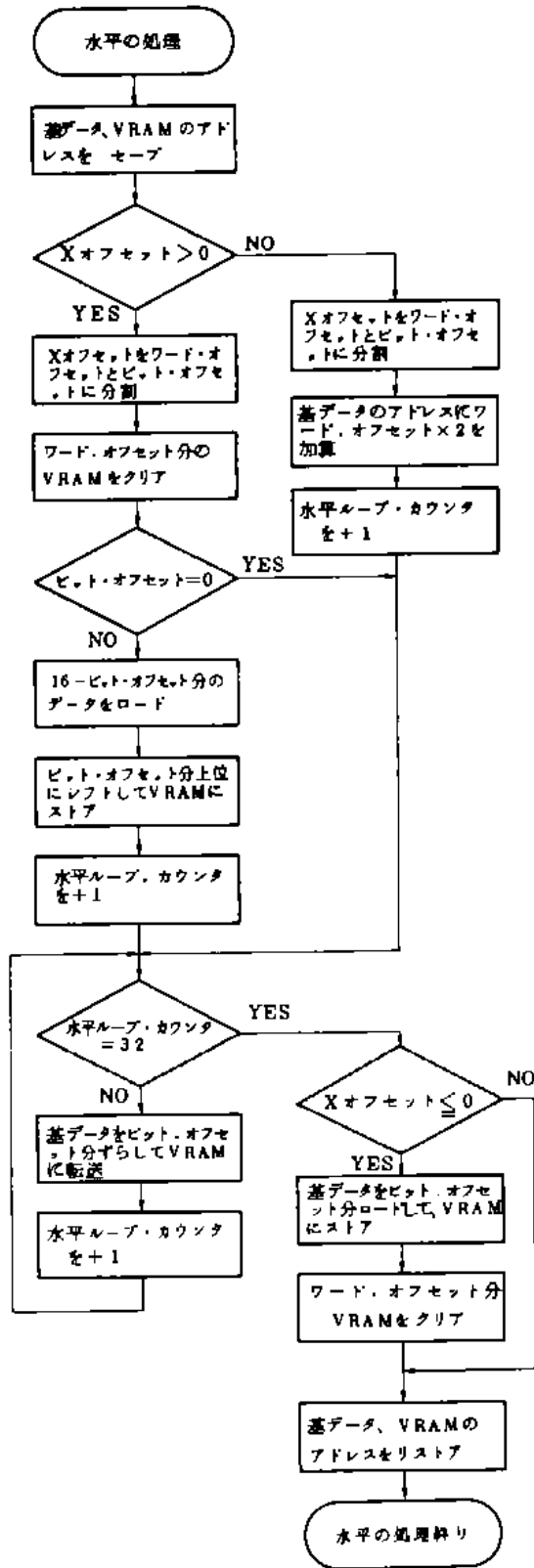
右方向への移動



左方向への移動

2.6 TRANSFERサブルーチン・フローチャート





2.7 SCALEサブルーチン

SCALEサブルーチンでは、最初に、水平方向の拡大/縮小比を最大256、垂直方向の拡大/縮小比を最大128に設定します。その後、指定された拡大/縮小比を、拡大/縮小の指定をするフラグと、2底の対数に変換し、整数に切り捨てます。

(1) 垂直方向の拡大/縮小

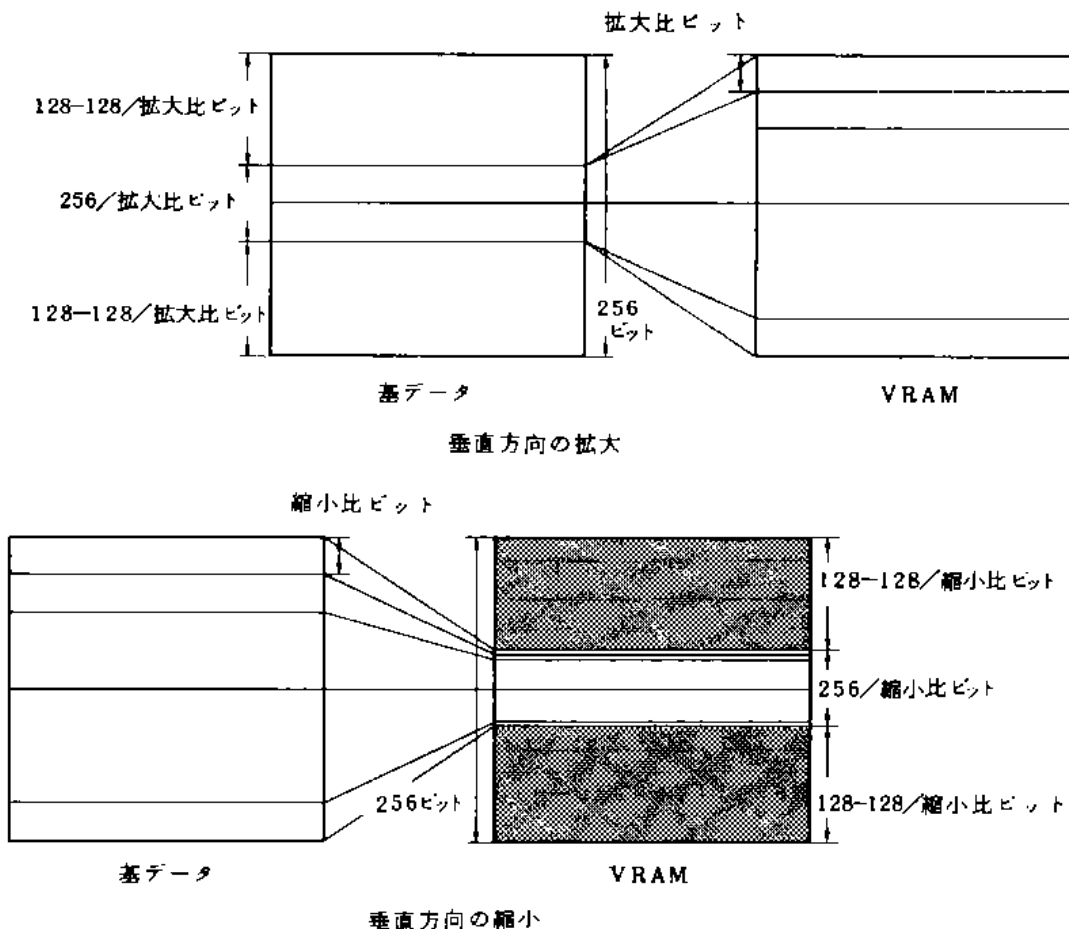
第2.5図に垂直方向の拡大/縮小の様子を示します。

垂直方向の拡大の場合、VRAMの先頭行は、基データの“128-128 / 拡大比”行目に相当するため、基データの先頭アドレスに $(128-128 / \text{拡大比}) \times 64$ (1行のバイト数) をプラスします。その後、水平の処理を行ない、処理された行を“拡大比-1”回コピーし、処理を“256/拡大比”回繰返します。

垂直方向の縮小の場合、基データの先頭行は、VRAMの“128-128 / 縮小比”行目に相当するため、VRAMの領域を“128-128 / 縮小比”行分クリアします。

その後、水平方向の処理を行なったあと、基データのアドレスに“64×縮小比”をプラスし、処理を“256 / 縮小比”回繰返します。最後にVRAM領域を“128-128 / 縮小比”行分クリアします。

第2.5図



② 水平方向の拡大/縮小

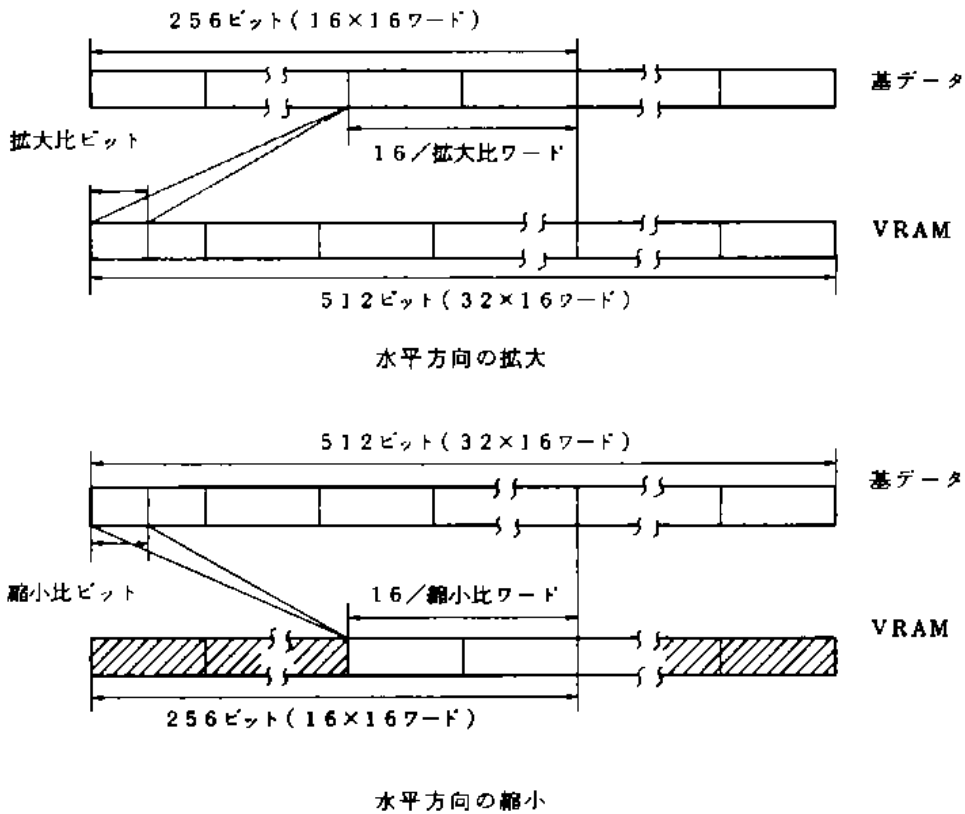
第 2.6 図に水平方向の拡大/縮小の様子を示します。

拡大比/縮小比が“1”の場合、単に基データの1行分のデータをVRAM1行に転送します。

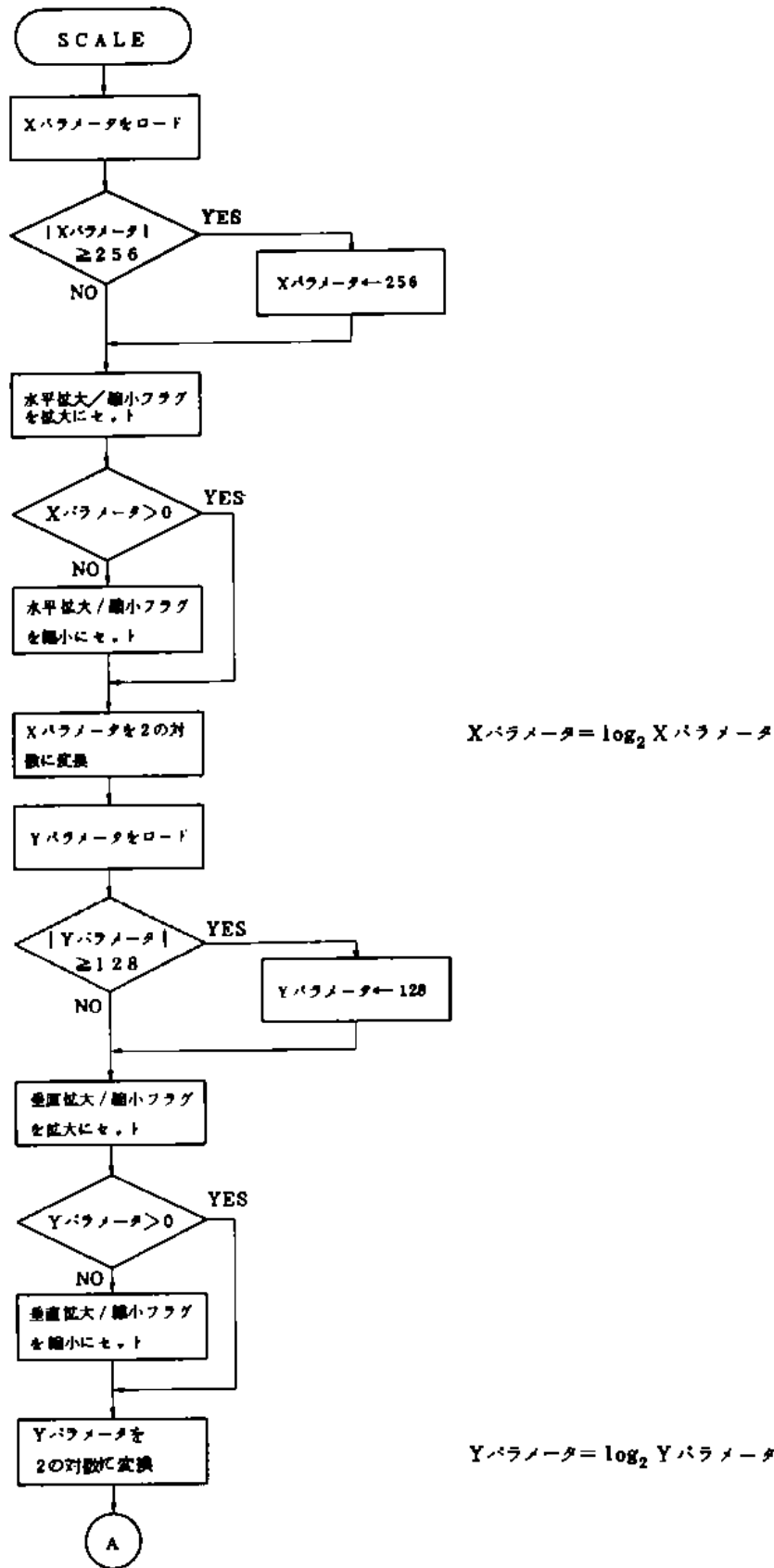
拡大の場合、VRAMの先頭ワードが基データの先頭ワードから“ $16 - 16 / \text{拡大比}$ ”目に相当するため、基データのアドレスに“ $16 - 16 / \text{拡大比}$ ”を加え、VRAMのビット・アドレスを“0”に設定します。ただし加える値が16になる場合には15を加え、VRAMのビット・アドレスを“ $256 / \text{拡大比}$ ”に設定します。その後、基データのビットの状態により拡大比が16未満の時はビット単位、16以上の時はワード単位で拡大比ビット分VRAMのデータを設定し、処理を“ $512 / \text{拡大比}$ ”回繰返します。

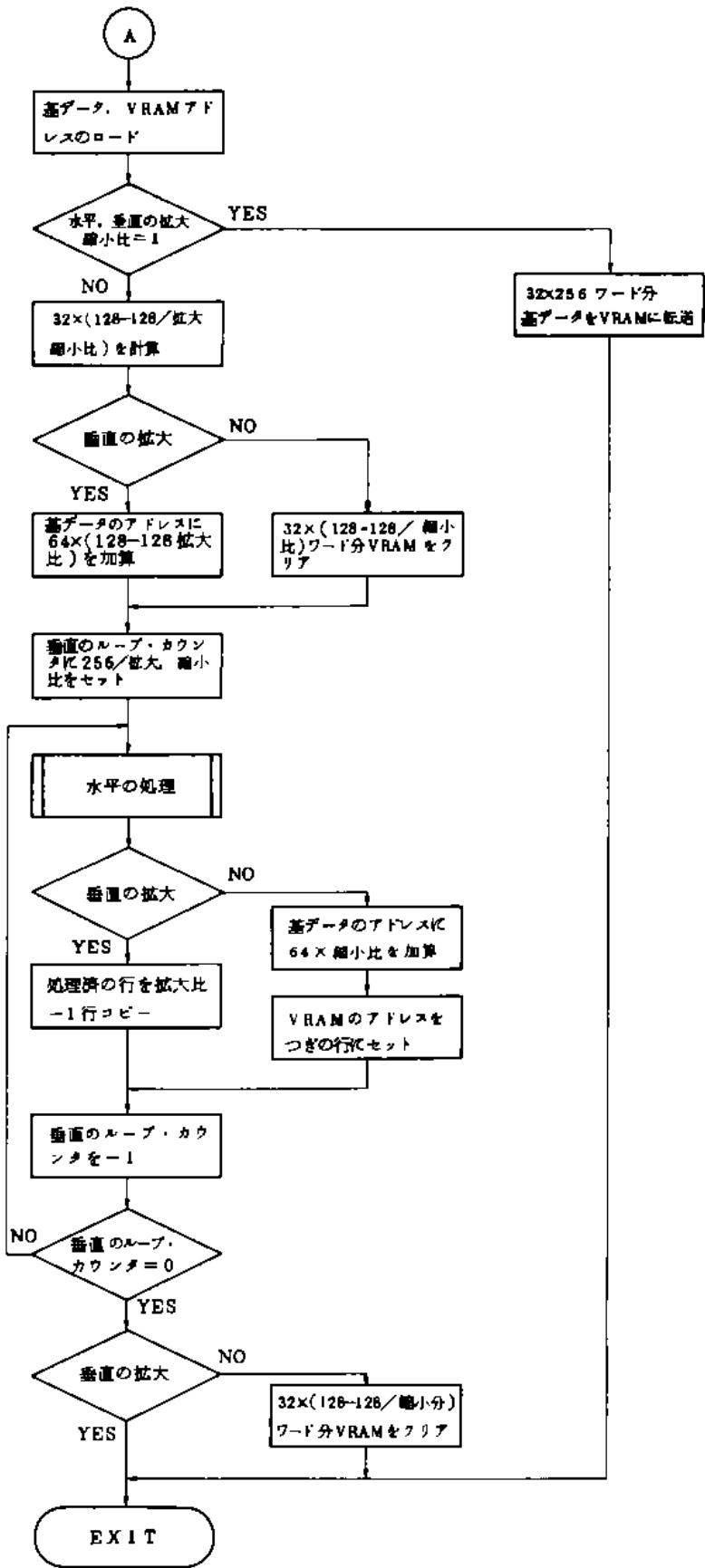
縮小の場合、基データの先頭ワードがVRAMの先頭ワードから“ $16 - 16 / \text{縮小比}$ ”目に相当するため、VRAMを“ $16 - 16 / \text{縮小比}$ ”ワード分クリアし、基データのビット・アドレスを“0”に設定します。ただしクリアするワード数が16になる場合には15ワードをクリアし基データのビット・アドレスを“ $256 / \text{縮小比}$ ”に設定します。その後、基データのビットの状態によりVRAMのデータを設定し、縮小比が16未満の場合基データのビット・アドレスを“縮小比”分、16以上の場合バイト・アドレスを“縮小比/8”分プラスし、処理を“ $512 / \text{縮小比}$ ”回繰返します。最後に、VRAMのビット・アドレスが“0”でない場合処理済みのデータをVRAMにストアした後1行の残りの部分をクリアします。

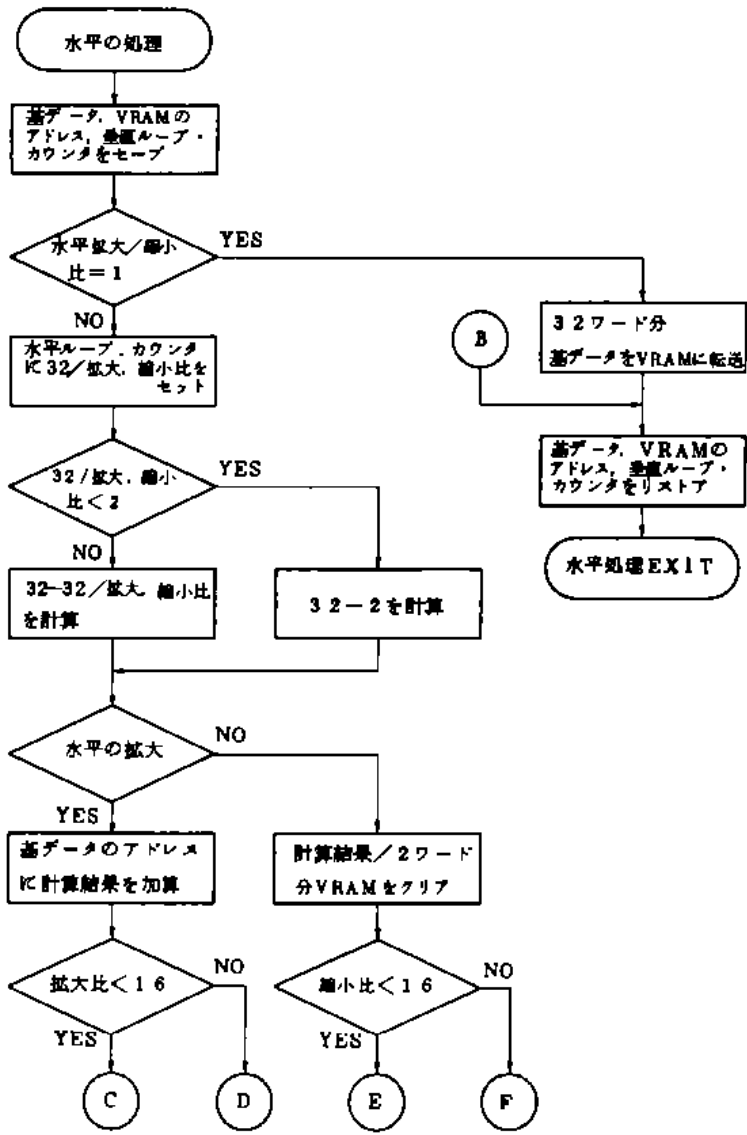
第 2.6 図

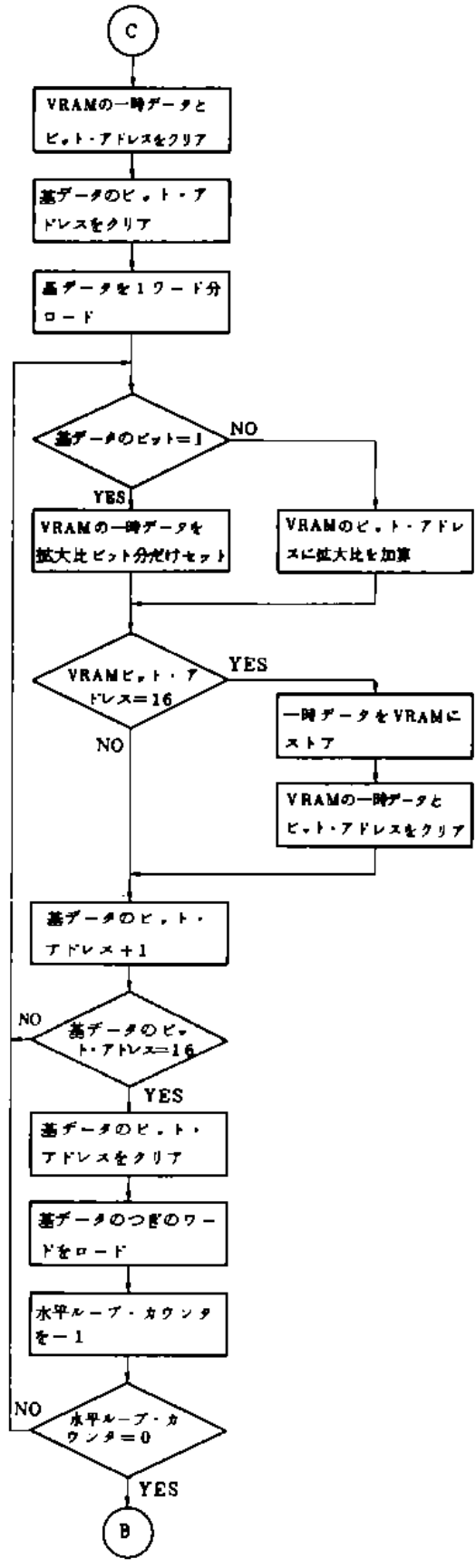


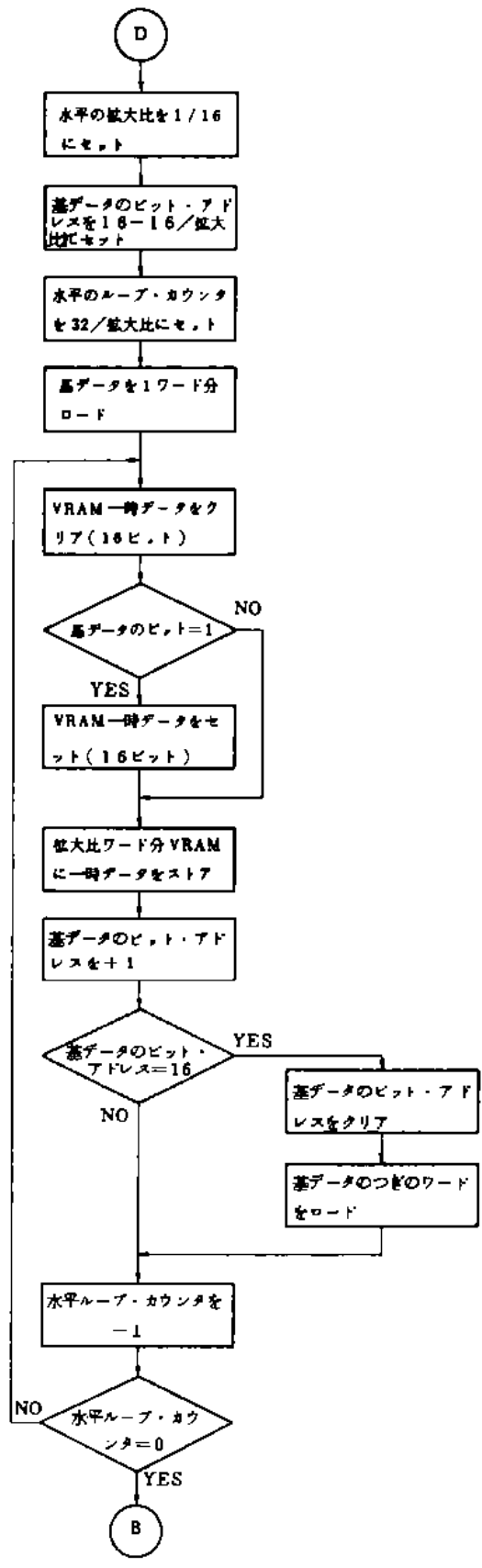
2.8 SCALEサブルーチン・フローチャート

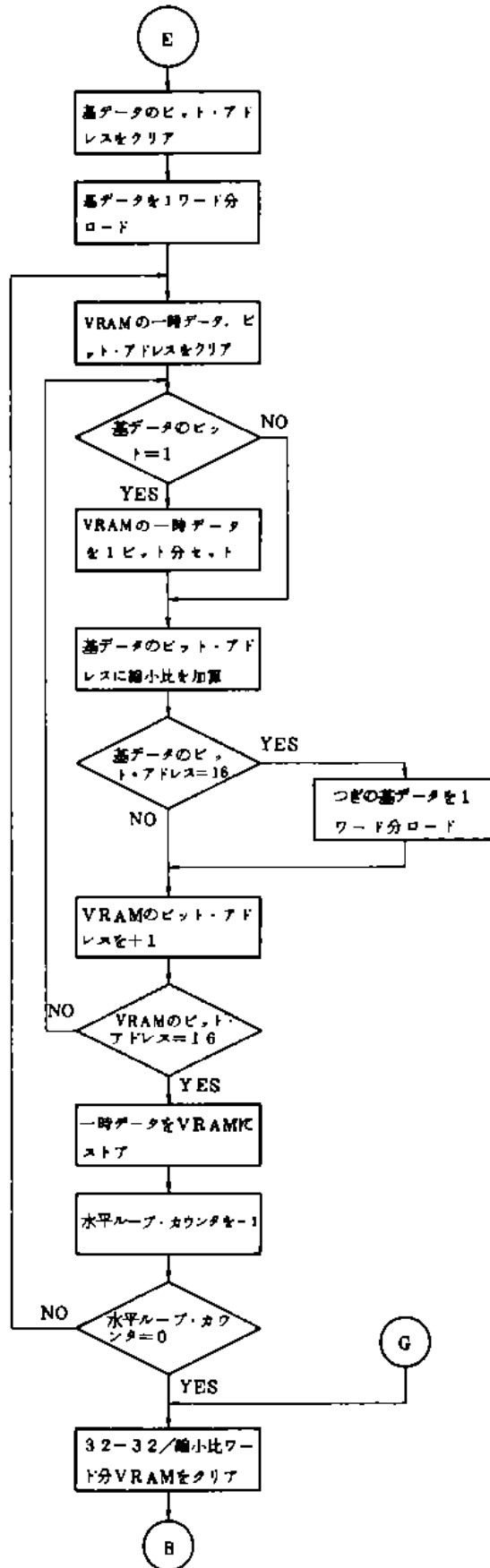


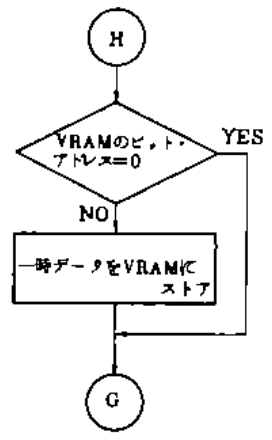
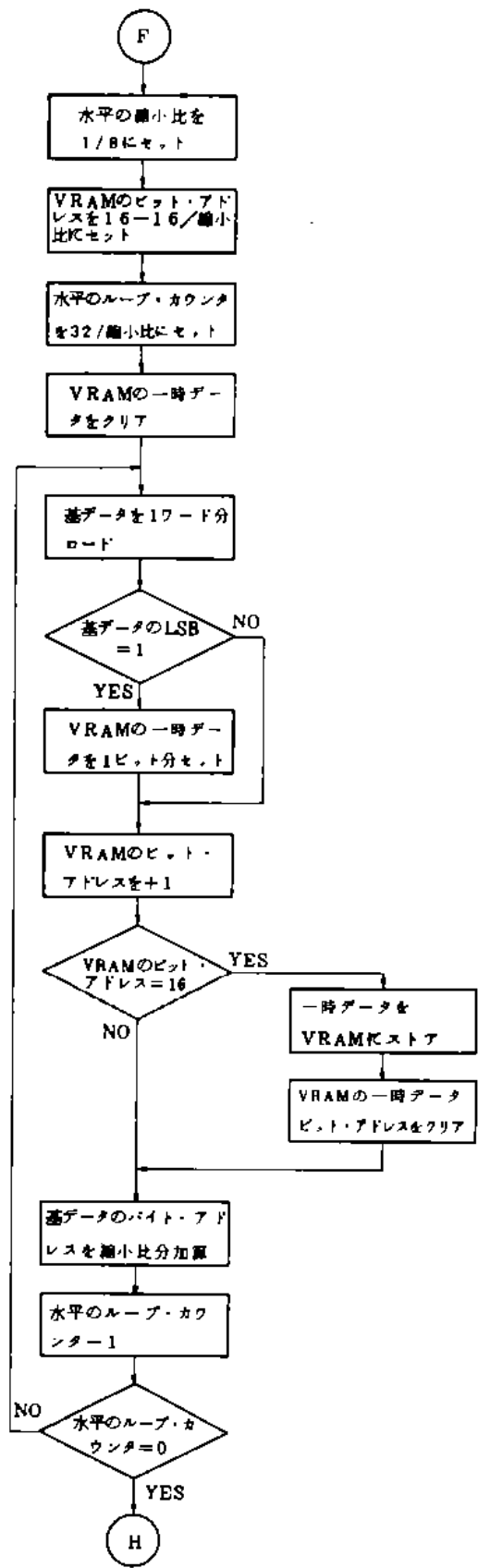








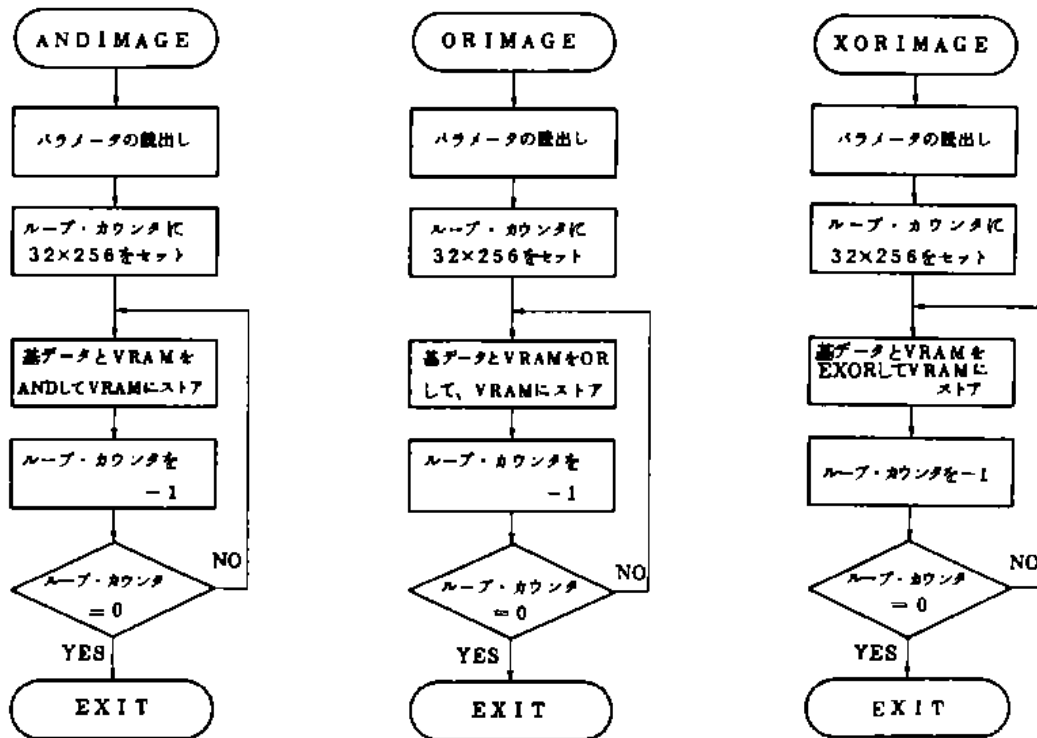




2.9 ANDIMAGE, ORIMAGE, XORIMAGE サブルーチン

ANDIMAGE, ORIMAGE, XORIMAGE サブルーチンでは、基データとVRAMの間で、ワード単位でそれぞれの演算を 32×256 回行ないます。

2.10 ANDIMAGE, ORIMAGE, XORIMAGE サブルーチン・フローチャート



UPD701C5770116 ASSEMBLER V1.0 IMAGE

DATE:25 Apr 85 11:28:39 PAGE: 1

SOURCE FILE : 1:IMAGE.SRC
 OBJECT FILE : 0:IMAGE.OBJ (IMAGE)
 COMMAND : 0:IMAGE.SRC FREE

```

LOC  OBJ  LINE  SOURCE
      1  ;*****
      2  ;=
      3  ;=  IMAGE PROCESSING EXAMPLE  ;
      4  ;=
      5  ;=          COPYRIGHT (C) 1985 NEC CORPORATION  ;
      6  ;*****
      7
      8          NAME  IMAGE
      9
     10  ;*****
     11  ;= IMAGE DATA SIZE DEFINITION  ;
     12  ;*****
     13
     14  HBIT  EQU  512          ;HORIZONTAL BIT COUNT
     15  HWORD  EQU  32         ;HORIZONTAL WORD COUNT
     16  VBIT  EQU  256         ;VERTICAL BIT COUNT
     17
     18  ;*****
     19  ;= PARAMETER ACCESS STRUCTURE DEFINITION  ;
     20  ;*****
     21
     22  PSTR  STRUC
     23
     24          DW  ?          ;OLD BP SAVE AREA
     25          DW  ?          ;RETURN ADDRESS SAVE AREA
     26  SRCAD  DW  ?          ;SOURCE DATA ADDRESS
     27  DSTAD  DW  ?          ;DESTINATION DATA ADDRESS
     28  XPARA  DW  ?          ;HORIZONTAL PARAMETER
     29  YPARA  DW  ?          ;VERTICAL PARAMETER
     30
     31  PSTR  ENDS
     32
     33  PSTR2  STRUC
     34
     35          DW  ?          ;OLD BP SAVE AREA
     36          DW  ?          ;RETURN ADDRESS SAVE AREA
     37          DW  ?          ;SOURCE DATA ADDRESS
     38          DW  ?          ;DESTINATION DATA ADDRESS
     39  XPARAM  DB  ?          ;HORIZONTAL PARAMETER LOW
     40  XPARAM  DB  ?          ;HORIZONTAL PARAMETER HIGH
     41  YPARAM  DB  ?          ;VERTICAL PARAMETER LOW
     42  YPARAM  DB  ?          ;VERTICAL PARAMETER HIGH
     43
     44  PSTR2  ENDS
     45
     46  P      EQU  (30)          ;PARAMETER BASE DEFINI
     47
     48  SEJECT
    
```

2. 11 画像処理プログラム プログラム・リスト

```

LOC 031
      LINE SOURCE
      49 ;.....
      50 ;* SOURCE IMAGE DATA AREA *
      51 ;.....
      52
----
      53 SRCSEG SEGMENT
      54
      55          SRC      DW      HWORD*4BIT DUP(?)
      56
----
      57 SRCSEG ENDS
      58
      59 ;.....
      60 ;* DESTINATION IMAGE DATA AREA *
      61 ;.....
      62
----
      63 DSTSEG SEGMENT
      64
      65          DST      DW      HWORD*4BIT DUP(?)
      66
----
      67 DSTSEG ENDS
      68
      69 ;.....
      70 ;*
      71 ;* IMAGE PROCESSING SUBROUTINES *
      72 ;*
      73 ;*
      74 ;*
      75          SP+8 ----> |-----|
      76          | Y PARAMETER |
      77          SP+6 ----> |-----|
      78          | X PARAMETER |
      79          SP+4 ----> |-----|
      80          | DST ADDR  |
      81          SP+2 ----> |-----|
      82          | SRC ADDR  |
      83          SP ---->  |-----|
      84          | RET ADDR  |
      85          |-----|
      86 ;*
      87 ;*
----
      88 IMAGE SEGMENT PUBLIC 'CODE'
      89
      90          ASSUME PS:IMAGE,DS:SRCSEG,DST:DSTSEG
      91          ;
      92          PUBLIC TRANSFER,SCALE,ANDIMAGE,ORIMAGE,XORIMAGE
      93
      94          $EJECT

```

```

LOC OBJ          LINE      SOURCE
                95          ;*****
                96          ;* TRANSFER
                97          ;*
                98          ;*
                99          ;*          [SP+8] Y OFFSET
                99          ;*          [SP+8] X OFFSET
                100         ;*
                101         ;*****
                102
0000             103 TRANSFER      PROC      NEAR
                104 ;
0000 55          105          PUSH     SP          ;SAVE OLD BASE POINTER
0001 80E0       106          MOV     PP,SP      ;GET NEW BASE POINTER
0003 60          107          PUSH     R          ;SAVE ALL REGISTER
                108 ;
0004 807E26     109          MOV     IX,P.DSTAD ;GET DESTINATION DATA ADDRESS (IX)
0007 8054C8     110          MOV     DX,P.XPARA ;GET X OFFSET (DX)
000A 81F4C0D2   111          CMP     DX,HBIT    ;CHECK !X OFFSET! > X BIT COUNT
000E 7D1E      112          BGE     TOVER
                113 ;
0010 81F4D0FE   114          CMP     DX,-HBIT
0014 7E15      115          BLE     TOVER
                116 ;
0016 80520A     117          MOV     BX,P.YPARA ;GET Y OFFSET (BX)
0019 81FB0D01   118          CMP     BX,YBIT    ;CHECK !Y OFFSET! > Y BIT COUNT
001D 7B0C      119          BGE     TOVER
                120 ;
001F 81FB0DFF   121          CMP     BX,-YBIT
0023 7E06      122          BLE     TOVER
                123 ;
0025 807E04     124          MOV     IX,P.SRCAD ;GET SOURCE DATA ADDRESS (IX)
0026 6BCB90     125          BR      TPROC
                126 ;
0028 89002D     127 TOVER:  MOV     CX,WORD-VBIT ;CLEAR DESTINATION DATA
002E 53CC      128          XOR     AX,AX
                129 ;
0030 F34E      130          REP     STP     DST
0032 694ED3     131          BR      YEXIT    ;EXIT TO MAIN PROGRAM
                132 ;
                133 ;*****
                134 ;* PAGE PROCEDURE
                135 ;*****
                136
0035 80C5      137 TPROC:  MOV     CX,BX          ;GET WORD OFFSET FROM Y OFFSET
0037 C1E105     138          SHL     CX,5
003A 7D0B      139          BGT     TPROC1
                140 ;
                141 ;*****
                142 ;* IF Y OFFSET < 0
                143 ;*****
                144
003C F7D9      145          NEG     CX          ;SET Y OFFSET TO POSITIVE
003E 53CC      146          XOR     AX,AX      ;CLEAR NO SOURCE AREA
                147 ;
0040 F34B      148          REP     STP     DST
0042 F7DB      149          NEG     BX          ;SET Y LOOP COUNTER(BX)

```

```

LOC DEJ          LINE      SOURCE
0040 E0590       150          BP      TPROC2
                  151          ;
                  152          ;.....
                  153          ;= IF Y OFFSET >= 0
                  154          ;.....
                  155          ;
0047 01E1       156  TPROC1: SHL      CW,1          ;GET BYTE OFFSET FROM Y OFFSET
0049 03F1       157          ADD      IX,CW          ;ADD BYTE OFFSET TO SOURCE ADDRESS
                  158          ;
                  159          ;.....
                  160          ;= ONE LINE PROCEDURE
                  161          ;.....
                  162          ;
004B 56         163  TPROC2: PUSH     IX          ;SAVE SOURCE & DESTINATION ADDRESS
004C 57         164          PUSH     IX
004D 2302       165          AND      DM,0W
004F 7E2C       166          ORL     TPROC3
                  167          ;
                  168          ;.....
                  169          ;= IF X OFFSET > 0
                  170          ;.....
                  171          ;
0051 38CA       172          MOV      CW,0W          ;GET BYTE OFFSET
0053 51E10FF    173          AND      CW,0FFF0H
0057 01F04      174          SHRA   CW,4
005A 5AF9       175          MOV      BM,CL          ;SET VERTICAL LOOP COUNTER (BM)
005C 33CC       176          XOR     AM,AM
                  177          ;
005E F3AF       176  REP     STP      DST          ;CLEAR NO SOURCE AREA
0060 8ACA       177          MOV      CL,DL
0062 F659       180          NEG     CL          ;GET BIT LENGTH FOR FIRST DATA
0064 7431       181          BZ      TPROC4
                  182          ;
                  183          ;.....
                  184          ;= IF BIT OFFSET NOT 0
                  185          ;.....
                  186          ;
0066 FEC9       187          DEC     CL          ;TRANSFER FIRST DATA
0068 50E10F     188          AND      CL,0FH
0069 32ED       189          XOR     CH,CW
006D 0F33CD     190          EXT     CH,CL          ;LOAD SOURCE 1-15 BIT
0070 8ACA       191          MOV      CL,DL
0072 50E10F     192          AND      CL,0FH          ;SHIFT BIT OFFSET
0075 03L0       193          SHL     AW,CL
0077 45         194          STP     DST          ;STORE DESTINATION DATA (16 BIT)
0078 FEC7       195          INC     BH
007A EB1990     196          BP      TPROC4
                  197          ;
                  198          ;.....
                  199          ;= IF X OFFSET <= 0
                  200          ;.....
                  201          ;
007B 49CA       202  TPROC3: MOV      CW,0W          ;GET BYTE OFFSET
007F F769       203          NEG     CW
0041 51E10FF    204          AND      CW,0FFF0H          ;ADD BYTE OFFSET TO SOURCE ADDRESS

```

LOC	OBJ	LINE	SOURCE
0085	C1F5D3	205	SHRA CW,3
0086	03F1	206	ADD IX,CX
008A	01F9	207	SHRA CW,1
008C	3AF9	208	MOV BX,CL
008E	FEC7	209	INC BX
0090	9AEA	210	MOV CH,DL
0092	F6DD	211	NEG CH
0094	40E50F	212	AND CH,0FH
		213	;
		214	;
		215	;* TRANSFER LOOP *
		216	;
		217	;
0097	80FF2G	218	TPROC4: CMP BH,HWORD
009A	7409	219	BZ TPROC5
		220	;
009C	0F33C50F	221	EXT CH,15
00A0	AB	222	STM DST
00A1	FEC7	223	INC BX
00A3	E8F2	224	BR TPROC4
		225	;
00A5	2302	226	TPROC5: AND DW,0W
00A7	7F1A	227	SGT TPROC6
		228	;
		229	;
		230	;* IF X CFFSET <= 0 *
		231	;
		232	;
00A9	3ACA	233	MOV CL,DL
00AB	FEC9	234	DEC CL
00AD	40E10F	235	AND CL,0FH
00B0	0F33CD	236	EXT CH,CL
00B3	AB	237	STM DST
		238	;
00B4	3ACA	239	MOV CW,0W
00B6	77D9	240	NEG CW
00B8	31E1F0FF	241	AND CW,0FFF0H
00BC	C1F9C4	242	SHRA CW,4
00BF	33C0	243	XOR AX,AX
		244	;
00C1	F3AB	245	REP STM DST
		246	;
		247	;
		248	;* END OF ONE LINE PROCEDURE *
		249	;
		250	;
00C3		251	TPROC6:
00C5	5F	252	POP IX
00C4	5E	253	POP IX
00C5	33C640	254	ADD IX,HWORD*2
00C6	83C740	255	ADD IX,HWORD*2
00C9	FEC1	256	INC BX
00CD	7403	257	BZ TPROC7
		258	;
00CF	2079FF	259	BR TPROC2

```

LOC OBJ          LINE      SOURCE
                260      ;
OUP7 804E0A      261      TPROC7: MOV     CW,P,YPARA
CQ95 23C9       262      AND     CW,CM
OJ97 7EC7       263      BLE     TERIT
                264      ;
                265      ;=====
                266      ;= IF Y OFFSET > 0
                267      ;=====
                268      ;
OJ07 C7E1D5     269      SHL     CW,3           ;CLEAR NO SOURCE AREA
OJ0C 33C0       270      ROR     AW,AW
                271      ;
OJ0E F340       272      REP     STP     DST
                273      ;
                274      ;=====
                275      ;= EXIT TO MAIN PROGRAM
                276      ;=====
                277      ;
OJ0E 01         275      TERIT: POP     0           ;RESTORE REGISTER
OJ01 38E5       279      MOV     SP,SP
OJ03 50         280      POP     EP
OJ04 C3         281      RET
                282      ;
                283      ;
                284      TRANSFER     ENDP
                285      ;
                286      REJECT

```



```

LOC  OBJ          LINE  SOURCE
                                ;*****
                                ;* ENLARGEMENT/SHRINKING
                                ;*
                                ;*
                                ;*      [SP+8] Y PARAMETER
                                ;*      [SP+6] X PARAMETER
                                ;*
                                ;*
                                ;*      POSITIVE ENLARGEMENT
                                ;*      NEGATIVE SHRINKING
                                ;*
                                ;*****
00E5          296  SCALE  PROC   NEAR
                                297
00E5 55        300          PUSH  BP           ;SAVE OLD BASE POINTER
00E6 88EC     301          MOV   BP,SP       ;GET NEW BASE POINTER
00E8 60       302          PUSH  R           ;SAVE ALL REGISTER
                                ;
00E9 3E44G8   304          MOV   AX,P.XPARA  ;GET X PARAMETER
00EC 3A000F   305          CMP   AX,MBIT/2
00EF 7ED3     306          BLE  SCALE0
                                307
                                ;
00F1 8B000F   306          MOV   AX,MBIT/2   ;SET MAX. VALUE TO MBIT/2
                                309
                                ;
00F4          310  SCALE0:
00F4 3D00FA   311          CMP   AX,-MBIT/2
00F7 7D03     312          BGE  SCALE1
                                313
                                ;
00F9 3B00FF   314          MOV   AX,-MBIT/2
                                315
                                ;
00FC          316  SCALE1:
00FC 23CC     317          AND  AX,AX       ;SET HCRIZGNTAL ENLARG/SHRINK FLAG
00FE C64609D0 318          MOV  P.XPARAH,0
0102 7D06     319          BGE  SCALE2
                                320
                                ;
0104 C64609FF 321          MOV  P.XPARAH,0FFH
0105 F7D8     322          NEG  AX
                                323
                                ;
010A 8902D0   324          SCALE2: MOV  CX,3       ;CONVERT XPARA
                                325
                                ;
                                326
                                ;
010D          327  SCALE3:
010D 0F11C0   328          TEST  AX,CL
0110 7502     329          BNZ  SCALE4
                                330
                                ;
0112 E2F6     331          BNZ  SCALE3
                                332
                                ;
0114          333  SCALE4:
0114 4846C9   334          MOV  P.XPARAL,CL  ;STORE CONVERTED XPARA TO XPARA L
0117 8B46DA   335          MOV  AX,P.YPARA  ;GET Y PARAMETER
011A 3A80D0   336          CMP  AX,VBIT/2
011B 7ED3     337          BLE  SCALE5
                                338
                                ;
011F 8B80D0   339          MOV  AX,VBIT/2
                                340
                                ;
0122          341  SCALE5:

```

```

LOC  OBJ          LINE  SOURCE
0122  3080FF      342          CRP      AW,-VBIT/2
0123  7003        343          9GE      SCALF6
          ;
          ;          344
0127  0880FF      345          MOV      AW,-VBIT/2
          ;
          ;          346
012A          347  SCALE6:
012A  23C0        348          AND      AW,AW
012C  C646C3JD    349          MOV      P.YPARAM,0          ;SET VERTICAL ENLARGE/SHRINK FLAG
0130  7D06        350          BGE      SCALE7
          ;
          ;          351
0132  C64608FF    352          MOV      P.YPARAM,0FFH
0136  1708        353          NEG      AW
          ;
          ;          354
0138  0907DD      355  SCALE7: MOV      CW,?          ;CONVERT YPARA
          ;
          ;          356
          ;          357
013A          358  SCALE8:
013A  0F11CD      359          TEST1   AW,CL
013C  75D2        360          BNZ     SCALE9
          ;
          ;          361
0140  E7F9        362          DBNZ   SCALE8
          ;
          ;          363
0142          364  SCALE9:
0142  384E0A      365          MOV      P.YPARAL,CL          ;STORE CONVERTED YPARAMETER TO YPARA L
0145  5B7604      366          MOV      IX,P.SRCAD          ;GET SOURCE & DESTINATION ADDRESS
0148  BR7E06      367          MOV      IX,P.DSTAD
014B  0A4608      368          MOV      AL,P.XPARAL
014E  0A460A      369          DR      AL,P.YPARAL
0151  75DA        370          BNZ     SCLPRC
          ;
          ;          371
          ; *****
          ; * TRANSFER SOURCE TO DESTINATION *
          ; * IF VERTICAL & HORIZONTAL FACTOR = 0 *
          ; *****
          ;          372
          ;          373
          ;          374
          ;          375
          ;          376
0153  090C2D      377          MOV      CW,MWORD*VBIT          ;COPY SOURCE TO DESTINATION
          ;
          ;          378
0156  63A5        379  REP     MOVAK  DST,SPC
          ;
          ;          380
          ; *****
          ; * EXIT TO MAIN PROGRAM *
          ; *****
          ;          381
          ;          382
          ;          383
          ;          384
0158          385  SCLERIT:
0158  01        386          POP     R
0159  8BE5        387          MOV     SP,SP
015B  5D        388          POP     BP
015C  C3        389          RET
          ;
          ;          390
          ; *****
          ; * PAGE ENLARGEMENT SHRINKING PROCEDURE *
          ; *****
          ;          391
          ;          392
          ;          393
          ;          394
015D  0B5C00      395  SCLPRC: MOV     AW,VBIT/2          ;VBIT/2 = VBIT/(2*YPARA)
0160  9A4ECA      396          MOV     CL,P.YPARAL

```

```

LOC  OBJ          LINE  SOURCE
0153 03E9          397          SHL    AW,CL
0165 898000        398          MOV    CW,VBIT/2
0168 29C8          399          SUB    CW,AW
016A C1E105        400          SHL    CW,5                ;R 32
016B 51            401          PUSH   CW
016E 3A7E0B        402          MOV    BH,P.YPARAM        ;LOAD ENLARGE/SHRINK FLAG (=H)
0171 22FF          403          AND    BH,BH
0173 7507          404          BMZ   SCLPRC1
          405          ;
          406          ;*****
          407          ;* IF VERTICAL ENLARGEMENT *
          408          ;*****
          409          ;
0175 01E1          410          SHL    CW,1
0177 35F1          411          ADD    IX,CW                ;ADD 32*(VBIT/2-VBIT/(2*YPARA)) TO SRC ADDR
0179 E00590        412          BR     SCLPRC2
          413          ;
          414          ;*****
          415          ;* IF VERTICAL SHRINKING *
          416          ;*****
          417          ;
017C          418          SCLPRC1:
017C 33C0          419          XOR    AW,AW                ;CLEAR NO SOURCE AREA
          420          ;
017E 13AB          421          REP    STM    DST
          422          ;
0180          423          SCLPRC2:
0180 32D5          424          XOR    BL,BL                ;VERTICAL LOOP COUNTER (BL) = 256/YPARA
0182 19            425          SETI  CY
0183 8A4E0A        426          MOV    CL,P.YPARAL
0186 02D9          427          RORC  BL,CL
          428          ;
0188          429          SCLPRC3:
0188 E8420C        430          CALL  SCLLIN                ;CALL 1 LINE PROCEDURE
0189 22FF          431          AND    BH,BH
018B 7520          432          BMZ   SCLPRC4
          433          ;
          434          ;*****
          435          ;* IF VERTICAL ENLARGEMENT *
          436          ;*****
          437          ;
018F 56            438          PUSH  IX                ;COPY N-1 LINES FOR ENLARGEMENT
0190 32F7          439          MOV    IX,IX
0192 83C740        440          ADC    IX,HWORD*2
0195 3001          441          MOV    AL,1
0197 8A4E0A        442          MOV    CL,P.YPARAL
019A 02EC          443          SHL    AL,CL
019C FE08          444          DEC    AL
019E 8AC3          445          MOV    CL,AL
01A0 32E9          446          MOV    CX,CH
01A2 C1E105        447          SHL    CW,5
          448          ;
01A3 F32645        449          REF  MOV3K  DST,DST
01A3 5E            450          POP   IX
01A9 35C440        451          ADD    IX,HWORD*2        ;GET NEXT SOURCE LINE ADDRESS

```

```

LOC OBJ          LINE      SOURCE
01A0 E80E90      452          BR      SCLPRC5
                453          ;
                454          ;
                455          ; * IF VERTICAL SHRINKING *
                456          ;
                457          ;
01A1              458          SCLPRC4:
01A1 3A4E6A      459          MOV     CL,P,YPARAL      ;GET NEXT SOURCE LINE ADDRESS
01B2 4B4GLO      460          MOV     AM,HWORD*2      ;ADD HWORD*2=M TO SOURCE ADDRESS
01B5 03EC        461          SHL     AM,CL
01B7 03FC        462          ADD     IX,AM
01B9 33C740      463          ADD     IX,HWORD*2
                464          ;
01B0              465          SCLPRC5:
01B0 FECD        466          DEC     BL              ;DONE ALL LINES ?
01B6 75C9        467          BNZ    SCLPRC3
                468          ;
01C0 59          469          POP     CM
01C1 22FF        470          AND     BM,BH
01C3 75D2        471          BNZ    SCLPRC6
                472          ;
01C5 EB91        473          BR      SCLPRC1
                474          ;
                475          ;
                476          ; * IF VERTICAL SHRINKING *
                477          ;
                478          ;
01C7              479          SCLPRC6:
01C7 33CG        480          XOR     AM,AM          ;CLEAR NO SOURCE AREA
                481          ;
01C9 F34B        482          REP     STP     DST
01CB EB34        483          BR      SCLPRC1
                484          ;
                485          $EJECT

```

```

LOC OBJ          LINE      SOURCE
                                     ;*****
                                     ;+ ONE LINE PROCEDURE
                                     ;*****
C1C0 56          480
C1C2 57          481
C1CF 53          482
0700 dA7E0D     490      SCLLIN: PUSH  IX          ;SAVE SRC/DST ADDR & VERTICAL LOOP COUNTER
C1D3 22FF       491          PUSH  IX
C1D5 7509       492          PUSH  BX
                                     ;LOAD X PARAMETER (BX)
                                     AND   BX,BX
                                     BNE   SLINPRC
493
494
495
496
497
498
499      ;*****
500      ;+ TRANSFER SOURCE LINE TO DESTINATION LINE IF X = 0
501      ;*****
0107 d72G0D     501          MOV   CX,WORD
502
503
504
505
506      ;+ EXIT TO PAGE PROCEDURE
507      ;*****
508
509      SCLLINEIT:
010C           510          POP   BX
010E 5F         511          POP   IX
010E 5E         512          POP   IX
010F C3         513          RET
514
515
516      ;*****
517      ;+ HORIZONTAL ENLARGEMENT/SHRINKING PROCEDURE
518      ;*****
519
520      SLINPRC:
01E0           520          MOV   CL,BX          ;GET ENLARGE/SHRINK FACTOR
01E2 33D1       521          MOV   AL,1
01E4 02E3       522          SHL  BL,CL          ;CM = 32 - 32/(2**X) OR 2
01E6 302C       523          MOV   AL,32
01E8 02E8       524          SHR  AL,CL
01EA 30D2       525          CMP  AL,2
01EC 7302       526          BML  SLINPRC1
527
528
529
530
531      SLINPRC1:
01F0           530          MOV   BX,AL          ;SET VERTICAL LOOP COUNTER (BX) &
01F2 412C       531          MOV   CL,32          ;X PARAMETER (AL)
01F4 2ACF       532          SUB  CL,BX
01F6 32E0       533          XOR  CM,CX
01F8 32E0       534          XOR  CM,CX
01F8 32E0       535          TEST P,XPARAM,OFFH
01F8 32E0       536          BZ   LLARGE
537
538
539
540      SEJECT

```

```

LOC  OBJ          LINE      SOURCE
                                     ;*****
                                     ;* HORIZONTAL ENLARGEMENT PROCEDURE
                                     ;*****
                                     ;
0201 03F1          345      LLARGE: ADD    IX,CX          ;GET SOURCE ADDRESS
0203 40F810        346      CAP    BX,16
0205 7942          347      BP    LLARGE1
                                     ;
0206 22D9          348      AND    BX,BX
020A 743E          349      BZ    LLARGE1
                                     ;
                                     ;*****
020C 33CC          352      ;* IF HORIZONTAL ENLARGEMENT FACTOR < 16
020E 3AC2          353      ;*****
0210 32ED          354      ;
0212 3B14          355      ;
                                     ;
0214              356      XOR    AX,AX          ;CLEAR DST WORK (AX) & BIT ADDRESS (CX)
0216 86E9          357      MOV    CL,AL
0218 3F11C2        358      XOR    CH,CH          ;LOAD SRC DATA TO SRC WORK (DX) &
021A 46E9          359      MOV    DW,[IX]       ;CLEAR SRC BIT ADDRESS (CX)
021B 743E          360      ;
021C              361      LLARGE0:
021E 86E9          362      XCH    CH,CL
0220 3F11C2        363      TEST   DW,CL          ;TEST SOURCE BIT
0222 46E9          364      XCH    CH,CL
0224 743E          365      BZ    LLARGE2
0225              366      ;
0226 53            367      PUSH  BX
0227              368      ;
0228              369      LLARGE3:
022A 3F15C0        370      SETI  AX,CL          ;IF SOURCE BIT = 1 THEN
022C FEC1          371      INC   CL             ; SET DESTINATION N BITS
022E FEC2          372      DEC   BL
0230 75F7          373      BNZ  LLARGE3
0231              374      ;
0232 5B            375      POP   BX
0234 8B0390        376      BR   LLARGE4
0235              377      ;
0236              378      LLARGE2:
0238 02C9          379      ADD   CL,BX          ;IF SOURCE BIT = 0 THEN
0239              380      ;
0240              381      LLARGE4:
0242 40F910        382      CMP   CL,16          ;ADD N TO DESTINATION BIT ADDRESS
0244 7505          383      BNZ  LLARGE5
0245              384      ;
0246 4B            385      STB  DST             ;IF DST BIT ADDRESS = 16 THEN
0248 33CC          386      XOR  AX,AX          ;STORE DST WORK TO DESTINATION & CLEAR DST WORK
024A 3AC2          387      MOV  CL,AL
024B              388      ;
024C              389      LLARGE5:
024E F8C5          390      INC  CX             ;ADD 1 TO SRC BIT ADDRESS
0250 30F010        391      CMP  CX,16          ;IF SRC ADDRESS = 16 THEN
0252 7506          392      BNZ  LLARGE6       ;LOAD NEXT SOURCE DATA TO SRC WORK
0253              393      ;
0254 32ED          394      XOR  CH,CH
0256 4B            395      INC  IX

```

```

LOC OBJ          LINE      SOURCE
0241 46          596          INC      IX
0242 3814        597          MOV      DW,[IX]
0244 FECE       598          DEC      BH
0246 75CC       599          BNZ     LLARGE6          ;DONE ONE LINE ?
0243 2992        600          ;
0243 2992        601          BR      SCCLINEXT
0243 2992        602          ;
0243 2992        603          ;
0243 2992        604          ;+ IF HORIZONTAL ENLARGEMENT FACTOR >= 16
0243 2992        605          ;+
0243 2992        606          ;+
0244          607          LLARGE1:
0244 COE804     608          SHR     BL,4          ;ENLARGEMENT FACTOR/16
0246 7502       609          BNZ     LLARGE7
0246 8310       610          ;
0246 8310       611          MOV     BL,16
0246 8310       612          ;
0251          613          LLARGE7:
0251 2E04       614          SUB     AL,4
0253 54C8       615          MOV     CL,AL
0255 30FC       616          MOV     AL,DFDH          ;GET SOURCE BIT ADDRESS (CL)
0257 02F8       617          SHRA   AL,CL
0259 372C       618          MOV     BH,32          ;GET HORIZONTAL LOOP COUNTER (BH)
025B 02EF       619          SHR     BH,CL
025D 34CE       620          MOV     CL,AL
025F 90E10F     621          AND    CL,DFH
0262 8B14       622          MOV     DW,[IX]          ;LOAD SOURCE DATA TO SRC WORK
0262 8B14       623          ;
0264          624          LLARGE10:
0264 33CD       625          XOR     AW,AW          ;CLEAR DST WORK
0266 DF11C2     626          TEST   DW,CL          ;TEST SOURCE BIT
0269 7401       627          BZ     LLARGE8
0269 7401       628          ;
026B 48         629          DEC     AW          ;IF SOURCE BIT = 1 THEN SET ALL DST WORK BITS
026B 48         630          ;
02AC          631          LLARGE8:
02AC 51         632          PUSH   CW
02AD 3220       633          XOR     CH,CH
02AF 3ACB       634          MOV     CL,9L
02AF 3ACB       635          ;
0271 F3AB       636          REP    STM    DST          ;STORE ENLARGEMENT FACTOR WORDS TO DESTINATION
0273 59         637          POP    CW
0274 FECE       638          INC     CL          ;SOURCE BIT ADDRESS INCREMENT
0276 30F910     639          CMP    CL,16
0279 7506       640          BNZ     LLARGE9
0279 7506       641          ;
027B 46         642          INC     IX          ;IF SRC BIT ADDRESS = 16 THEN
027C 46         643          INC     IX          ;LOAD NEXT DATA TO SRC WORK
027D 3B14       644          MOV     DW,[IX]
027F 32C9       645          XDR    CL,CL
027F 32C9       646          ;
0281          647          LLARGE9:
0281 FECE       648          DEC     BH
0283 750F       649          BNZ     LLARGE10
0283 750F       650          ;

```

UPD7C106/7J116 ASSEMBLER W.LD IMAGE

DATE:25 Apr 85 17-23-39 PAGE: 14

LOC	OBJ	LINE	SOURCE		
0285	E954FF	051		9P	SCLLIMRTT
		052	:		
		053	REJECT		


```

LOC  OBJ      LINE      SOURCE
                                054      ;*****
                                055      ; HORIZONTAL SHRINKING PRECEDURE
                                056      ;*****
0276  01E9      056      LSHRNK: SHR      CW,1          ;CLEAR NO SOURCE AREA
028A  51        059      PUSH     CW
0293  5900      060      MOV      DW,AW
0280  33CC      061      XOR      AW,AW
                                062      ;
028F  F3AB      063      REP      STY      DST
0291  38C2      064      MOV      AW,DW
0293  30F910    065      CMP      BL,16
0290  7939      066      BP      LSHRNK1
                                067      ;
0298  2208      068      AND     BL,BL
029A  7435      069      BZ      LSHRNK1
                                070      ;
                                071      ;*****
                                072      ; * IF HORIZONTAL SHRINKING FACTOR < 16
                                073      ;*****
                                074
029C  32ED      075      XOR      CH,CH          ;CLEAR SRC BIT ADDRESS (CH) 8
029E  3214      076      MOV      DW,[CX]       ;LOAD SOURCE DATA TO SRC WORK (DW)
                                077      ;
02A0          078      LSHRNK5:
02A0  32E9      079      XOR      CL,CL          ;CLEAR DST BIT ADDRESS (CL) 8
02A2  33CC      080      XOR      AW,AW         ;DST WORK (AL)
                                081      ;
02A4          082      LSHRNK4:
02A4  30E9      083      XCH     CH,CL
02A6  3F11C2     084      TEST    DW,CL          ;TEST SOURCE BIT
02A9  36E9      085      XCH     CH,CL
02A9  7403      086      RZ      LSHRNK2
                                087      ;
02A0  3F15C0     088      SETI   AW,CL          ;IF SOURCE BIT = 1 THEN SET DESTINATION BIT
                                089      ;
02B0          090      LSHRNK2:
02B0  02E5      091      ADD     CH,BL          ;ADD SHRINK FACTOR TO SRC BIT ADDRESS
02B2  30F910    092      CMP     CH,16
02B3  7506      093      RAZ    LSHRNK3
                                094      ;
02B7  46        095      INC     IX              ;IF SOURCE ADDRESS = 16 THEN
02B8  46        096      INC     IX              ;LOAD NEXT SOURCE DATA TO SRC WORK
02B9  8914      097      MOV     DW,[IX]
02B8  32ED      098      XOR     CH,CH
                                099      ;
0250          700      LSHRNK3:
02B0  FE01      701      INC     CL              ;INCREMENT DESTINATION BIT ADDRESS
02B1  80F910    702      CMP     CL,16
02C2  75E0      703      RAZ    LSHRNK4
                                704      ;
02C4  AB      705      STP     DST            ;CLEAR DST WORK
02C5  FE0F      706      DEC     BH              ;DONE ONE LINE ?
02C7  7507      707      BNZ    LSHRNK5
                                708      ;

```

```

LOC  OBJ          LINE      SOURCE
02C9                709      LSHRNK11:
02C9 59            710          POP     CW
02CA 33CC          711          KOR     AW,AW          ;CLEAR NO SOURCE AREA
                                712          ;
02CC F342          713          REP     STP     DST
02CE E9GEFF        714          BR      SCLLINEKIT
                                715          ;
                                716          ;*****
                                717          ;* IF HCRIZONTAL SHRINK FACTOR >= 10
                                718          ;*****
                                719          ;
02D1                720      LSHRNK1:
02D1 CGEEC3        721          SHR     BL,3          ;SHRINK FACTOR /8
02D4 7502          722          BNZ     LSHRNK6
                                723          ;
02D6 332D          724          MOV     BL,32
                                725          ;
02D8                726      LSHRNK2:
02D8 2C04          727          SUB     AL,4
02DA 34CE          728          MOV     CL,AL
02DC 30FC          729          MOV     AL,DF0H      ;GET DESTINATION BIT ADDRESS (CL)
02DE 02FE          730          SHRA   AL,CL
02E0 372D          731          MOV     BH,32
02E2 02EF          732          SHR     BH,CL        ;GET HCRIZONTAL LOOP COUNTER (BH)
02E4 34CE          733          MOV     CL,AL
02E6 30E1DF        734          AND    CL,DF0H
02E9 33CC          735          KOR     AW,AW        ;CLEAR DST WORK & LOAD SOURCE TO SRC WORK
                                736          ;
02EA                737      LSHRNK3:
02EA 5814          738          MOV     DW,(IX)
02EB DF19C2D1      739          TESTI  DW,1          ;TEST SOURCE BIT
02ED 7405          740          BZ     LSHRNK7
                                741          ;
02F3 DF15CD        742          SETI  AW,CL          ;IF SOURCE BIT = 1 THEN SET DESTINATION BIT
                                743          ;
02F6                744      LSHRNK4:
02F6 FECE          745          INC     CL          ;INCREPENT DESTINATION ADDRESS
02F8 30191D        746          CMP     CL,10
02FA 7505          747          BNZ     LSHRNK6
                                748          ;
                                749          ;
02FD AB            750          STP     DST          ;IF DESTINATION BIT ADDRESS = 10 THEN
02FE 32C9          751          XOR     CL,CL        ;STORE DESTINATION & CLEAR DST WORK
0300 33CC          752          KOR     AW,AW
                                753          ;
0302                754      LSHRNK5:
0302 53            755          PUSH   BH           ;GET NEXT SOURCE BYTE ADDRESS
0303 32FF          756          XOR     BH,BH
0305 03F3          757          ADD     IX,BH
0307 50            758          POP     BH
0308 FECE          759          DEC     BH          ;DONE ONE LINE ?
030A 75DF          760          BNZ     LSHRNK9
                                761          ;
030C 22E9          762          AND    CL,CL        ;IF DESTINATION ADDRESS NOT 0 THEN
030E F401          763          BZ     LSHRNK10

```

LOC	OBJ	LINE	SOURCE
		766	;
0510	AB	767	STP DST ;STORE DESTINATION DATA
		768	;
0311		769	LSHRNK10:
0311	C020	769	BA LSHRNK11
		770	;
		771	SCALE EMDP
		772	;
		773	SEJECT

```

LOC  OBJ          LINE  SOURCE
                                774  ;*****
                                775  ;* AND OPERATION BETWEEN SOURCE AND DESTINATION *
                                776  ;*****
                                777
0313          778  ANDIMAGE      PROC      NEAR
                                779
0313 55          780          PUSH     BP          ;SAVE OLD BASE POINTER
0314 98EC        781          MOV      BP,SP      ;GET NEW BASE POINTER
0316 56          782          PUSH     IX          ;SAVE REGISTERS
0317 57          783          PUSH     IY
0318 51          784          PUSH     CW
                                785  ;
0314 390020      786          MOV      CW,MWORD+VBIT ;SET LOOP COUNTER
                                787  ;
031C AD          788  [AND:  LDW      SRC      ;LOAD SOURCE DATA
031D 262305      789          AND      AW,DST[LY]   ;AND OPERATION
0320 AD          790          STM      DST      ;STORE DESTINATION
0321 22F9        791          DENZ   IAND
                                792  ;
0323 59          793          POP      CW          ;RESTORE REGISTERS
0324 5F          794          POP     IY
0325 5E          795          POP     IX
0326 8Bb3        796          MCV     SP,AP
0328 5B          797          POP     AP
0329 E3          798          RET
                                799
0300          800  ANDIMAGE      ENDP
0301          801
0302          802  SELECT

```

```

LOC  JOB          LINE  SOURCE
                                803 ;*****
                                804 ; OR OPERATION BETWEEN SOURCE AND DESTINATION *
                                805 ;*****
                                806
032A          807 ORIMAGE PROC  NBAR
                                808
032A 55          809          PUSH  SP          ;SAVE OLD BASE POINTER
032B 56EC        810          MOV   BP,SP      ;GET NEW BASE POINTER
032C 56          811          PUSH  IX          ;SAVE REGISTERS
032E 57          812          PUSH  IY
032F 51          813          PUSH  CX
                                814 ;
                                815          MOV   CX,WORD+VBIT ;SET LOOP COUNTER
                                816 ;
0333 4D          817 [OR:  LDM   SRC          ;LOAD SOURCE DATA
0334 260B05      818          OR   AM,DST[IX]    ;OR OPERATION
0337 48          819          STP   DST          ;STORE DESTINATION
0338 22F9        820          DENZ  IOR
                                821 ;
033A 59          822          POP   CX          ;RESTORE REGISTERS
033B 5F          823          POP   IY
033C 5E          824          POP   IX
033D 38E5        825          MOV   SP,BP
033F 50          826          POP   BP
0340 C3          827          RET
                                828
                                829 ORIMAGE ENDP
                                830
                                831 SEJECT

```

```

LOC OBJ          LINE      SOURCE
                532      ;*****
                533      ;*  ERROR OPERATION BETWEEN SOURCE AND DESTINATION  *
                534      ;*****
                535
0341             536      XORIMAGE      PROC      NEAR
                537
0341 55          538          PUSH      BP          ;SAVE OLD BASE POINTER
0342 30E0        539          MOV      BP,SP      ;GET NEW BASE POINTER
0344 56          540          PUSH      IX          ;SAVE REGISTERS
0345 57          541          PUSH      IY
0346 51          542          PUSH      CX
                543      ;
0347 490C20      544          MOV      CX,WORD*VRIT      ;SET LOOP COUNTER
                545      ;
034A AD          546      ICR:  LDM      SRC          ;LOAD SOURCE DATA
0349 263305      547          XOR      AX,DST[IX]      ;XOR OPERATION
034E AD          548          STB      DST          ;STORE DESTINATION
034F 82F9        549          DBNZ     ICR,H      ;
                550      ;
0351 59          551          POP      CX          ;RESTORE REGISTERS
0352 5F          552          POP      IY
0353 3E          553          POP      IX
0354 8BE5        554          MOV      SP,BP
0356 5D          555          POP      BP
0357 C3          556          RET
                557
                558      XORIMAGE      ENDP
                559
----           560      IMAGE      ENDS
                561
                562          END

```

*** XREF SYMBOL LISTING ***

NAME	TYPE	VALUE	ATTRIBUTES	XREF
??SEG	SEGMENT		SIZE=000CH PARA PUBLIC	
ANDIMAGE	NEAR	0313	IMAGE PUBLIC 92# 77# 80#	
DST	WORD	0000	DSTSEG 65# 110 148 179 194 222 237 245 272 379 421 440 449 492 503 585 636 663 705 713 750 763 78#	
			290 818 819 847 848	
DST10	WORD	0006	27# 10# 35#	
DSTSEG	SEGMENT		SIZE=400CH PARA 63# 67# 90	
NOIT	NUMBER	0200	14# 111 114 305 309 311 314	
HWOPD	NUMBER	0020	15# 53 65 127 218 254 255 377 440 451 460 463 501 786 815 844	
LAND	NEAR	031C	IMAGE 78# 79#	
IMAGE	SEGMENT		SIZE=0358H PARA PUBLIC 'CODE' 8 8# 90 86#	
IOR	NEAR	0333	IMAGE 817# 820	
IROP	NEAR	034A	IMAGE 846# 849	
LLARGE	NEAR	0201	IMAGE 536 545#	
LLARGE1	NEAR	024A	IMAGE 547 550 607#	
LLARGE1C	NEAR	0264	IMAGE 624# 649	
LLARGE2	NEAR	0229	IMAGE 565 578#	
LLARGE3	NEAR	021E	IMAGE 569# 573	
LLARGE4	NEAR	022D	IMAGE 576 581#	
LLARGE5	NEAR	0237	IMAGE 583 599#	
LLARGE6	NEAR	0214	IMAGE 561# 592 599	
LLARGE7	NEAR	0231	IMAGE 609 613#	
LLARGE8	NEAR	026C	IMAGE 627 631#	
LLARGE9	NEAR	0261	IMAGE 640 647#	
LSHRNK	NEAR	0289	IMAGE 658 658#	
LSHRNK1	NEAR	0231	IMAGE 566 669 720#	
LSHRNK1C	NEAR	0311	IMAGE 763 767#	
LSHRNK1#	NEAR	02C7	IMAGE 709# 769	
LSHRNK2	NEAR	023D	IMAGE 676 690#	
LSHRNK3	NEAR	023D	IMAGE 693 700#	
LSHRNK4	NEAR	02A4	IMAGE 682# 703	
LSHRNK5	NEAR	0240	IMAGE 673# 707	
LSHRNK6	NEAR	02D8	IMAGE 722 726#	
LSHRNK7	NEAR	02F6	IMAGE 740 744#	
LSHRNK8	NEAR	0302	IMAGE 747 754#	
LSHRNK9	NEAR	02E8	IMAGE 737# 760	
ORIMAGE	NEAR	032A	IMAGE PUBLIC 92# 907# 929#	
P	-----	000C	[] 4# 10# 110 117 124 261 304 318 321 334 335 349 352 365 366 367 368 369 396 402 426 442 459 472	
			535	
PSTR	STRUC		SIZE=000CH 22# 31#	
PSTR2	STRUC		SIZE=000CH 33# 44#	
SCALE	NEAR	00E5	IMAGE PUBLIC 92# 292# 77#	
SCALE0	NEAR	00F4	IMAGE 306 310#	
SCALE1	NEAR	00FC	IMAGE 312 316#	
SCALE2	NEAR	0104	IMAGE 319 324#	
SCALE3	NEAR	010D	IMAGE 327# 331	
SCALE4	NEAR	0114	IMAGE 329 333#	
SCALE5	NEAR	0122	IMAGE 337 341#	
SCALE6	NEAR	012A	IMAGE 343 347#	
SCALE7	NEAR	0132	IMAGE 350 355#	
SCALE8	NEAR	013B	IMAGE 358# 362	
SCALE9	NEAR	0142	IMAGE 360 364#	
SCALE11	NEAR	0158	IMAGE 385# 473 483	

NAME	TYPE	VALUE	ATTRIBUTES	XREF
SCLLIN	NEAR	01C0	IMAGE	430 450#
SCLLINEXIT	NEAR	01B0	IMAGE	509# 601 631 714
SCLPRC	NEAR	0150	IMAGE	570 393#
SCLPRC1	NEAR	0170	IMAGE	404 418#
SCLPRC2	NEAR	0130	IMAGE	412 423#
SCLPRC3	NEAR	0103	IMAGE	429# 467
SCLPRC4	NEAR	01AF	IMAGE	432 458#
SCLPRC5	NEAR	0160	IMAGE	452 465#
SCLPRC6	NEAR	01C7	IMAGE	471 479#
SLINPRC	NEAR	01E0	IMAGE	455 519#
SLINPRC1	NEAR	01F0	IMAGE	526 530#
SPC	WORD	0000	SRCSEG	35# 319 503 788 817 846
SRC40	WORD	0004		26# 124 366
SRCSEG	SEGMENT		SIZE=4000H	PARA 53# 57# 90
TEXT	NEAR	00E0	IMAGE	131 263 279#
TCVER	NEAR	002F	IMAGE	112 115 119 122 127#
TPROC	NEAR	0035	IMAGE	125 137#
TPRCC1	NEAR	0047	IMAGE	139 156#
TPRCC2	NEAR	004B	IMAGE	150 163# 250
TPRCC3	NEAR	0070	IMAGE	166 202#
TPROC4	NEAR	0077	IMAGE	181 196 218# 224
TPROC5	NEAR	00A5	IMAGE	219 226#
TPROC6	NEAR	00E3	IMAGE	227 251#
TPROC7	NEAR	0002	IMAGE	257 261#
TRANSFER	NEAR	0000	IMAGE	PUBLIC 92# 103# 284#
VSIT	NUMBER	0100		16# 55 63 118 121 127 336 339 342 345 377 595 398 786 815 844
XDRIMAGE	NEAR	0341	IMAGE	PUBLIC 92# 856# 858#
XPARA	WORD	00C8		28# 110 304
XPARAH	BYTE	0009		40# 318 321 535
XPARAL	BYTE	0008		39# 334 368 493
YPARA	WORD	00CA		29# 117 261 335
YPARAH	BYTE	000B		42# 349 352 402
YPARAL	BYTE	000A		41# 365 369 398 426 442 459

ASSEMBLY COMPLETE, 0 ERROR(S) FOUND

第3章 10進フローティング・ポイント・データ演算パッケージ

3.1 プログラム概要

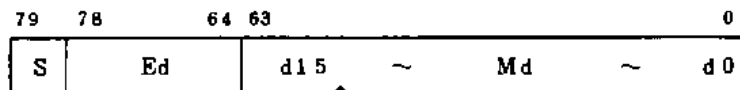
このプログラムは、あらかじめメモリ内にストアされている二つの10進フローティング・ポイント・データについて四則演算と比較をします。

それぞれの演算および比較の処理は、サブルーチン化されており、ソース・データのストアされているアドレスおよびデスティネーション・データのストアされているアドレスをパラメータとして受け取り、演算結果をデスティネーション・データにストアします。また比較結果または演算中に発生した例外事項は、ステータスとしてAWレジスタにセットされます。

3.2 10進フローティング・ポイント・データのフォーマット

このプログラムで扱う10進フローティング・ポイント・データのフォーマットを第3.1図に示します。

第3.1図



- S : 符号ビット (0 = 正, 1 = 負), ▲ 小数点
- Ed : 2進表現された指数部 (バイアス方式), バイアス = 3FFFH
- Md : BCD表現された仮数部 (16桁)

* このフォーマットでの数の体系

- ノーマル数 仮数部のすべての桁が10未満 (BCD) で最上位桁が"0"でなく, 指数部が"0"でも最大値でもない数
- アンノーマル数 仮数部のすべての桁が10未満 (BCD) で最上位桁が"0"であり, 指数部が"0"でも最大値でもない数
- デノーマル数 仮数部のすべての桁が10未満 (BCD) で少なくとも1桁以上が"0"でなく, 指数部が"0"である数
- 無限大 仮数部の最上位桁が"8" (MSB=1) で, その他の桁が"0"であり, 指数部が最大値である数

ノット・ア・デシマル (NAD)	仮数部の少なくとも1桁以上が10以上(BCDでない)である数
不 定	仮数部の最上位桁が"12"(COH)で、その他の桁が"0"であり、指数部が最大値である数
ゼ ロ	仮定数部のすべての桁が"0"であり、指数部が"0"である数

3.3 プログラムの前提事項

このプログラム例は、ソース・データおよびデスティネーション・データとも、同一セグメント内に存在することと、サブルーチンを呼び出すメイン・プログラムと各サブルーチンが同一セグメント内に存在することを、前提に作成されています。したがってアドレス情報はすべてセグメント内のオフセット内のオフセットとして扱われています。

3.4 四則演算と比較のサブルーチン

四則演算と比較を行なうサブルーチンには、次のものがあります。

VFADD	10進フローティング・ポイント・データの加算
VFSUB	10進フローティング・ポイント・データの減算
VF MUL	10進フローティング・ポイント・データの乗算
VFDIV	10進フローティング・ポイント・データの除算
VFCMP	10進フローティング・ポイント・データの比較

すべてのサブルーチンに対して、ソース・データのストアされているアドレスと、デスティネーション・データのストアされているアドレスを、パラメータとして与えます。

各サブルーチンを使用する場合、まず必要なパラメータをスタックにプッシュした後、サブルーチンを呼び出します。なお、スタック・ポイントの値は、サブルーチン内で、与えたパラメータの分を補正してから戻りますので、呼び出したプログラムでのスタック・ポイントの補正は不用です。

例

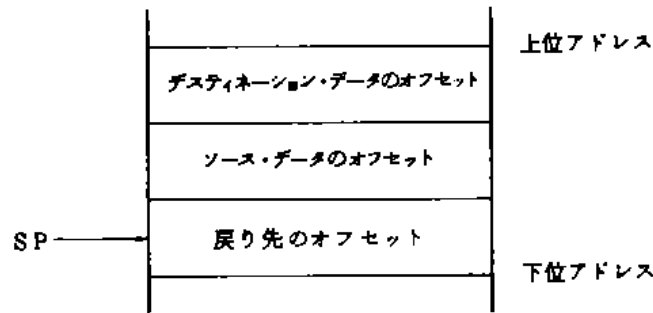
```

PUSH   OFFSET DSTDATA
PUSH   OFFSET SRCDATA
CALL   VFADD
      ⋮

```

第 3.2 図にサブルーチン・コール直後のスタックの状態を、示します。

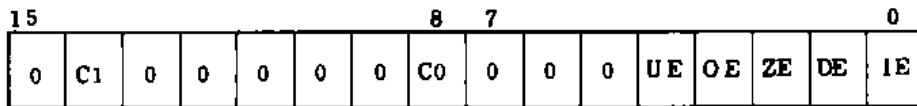
第 3.2 図



また、サブルーチンから戻る時には、比較および演算実行中に発生した例外事項の内容を示すため、AWレジスタにステータスをセットしてから戻ります。

第 3.3 図にステータス・ワード (AWレジスタ) のフォーマットを示します。

第 3.3 図



- | | |
|-----------------|--|
| オーバーフロー
(OE) | 演算の結果、指数部が最大値より大きくなった時、セットされます。
演算に適した符号の無限大を演算結果としてセットします。 |
|-----------------|--|
- | | |
|----------------|---|
| アンダフロー
(UE) | 演算の結果、指数部が“0”より小さくなった時、セットされます。アンノーマル数に変換後、演算結果としてセットします。変換後仮数部がすべて“0”になった場合には、ゼロをセットします。 |
|----------------|---|
- | | |
|------------------|---|
| ゼロ・デバインド
(ZE) | 比除数がゼロ、無限大以外で除数がゼロの除算の時、セットされます。符号をソース・データとデスティネーション・データの符号の排他的論理和にした無限大を演算結果としてセットします。 |
|------------------|---|
- | | |
|---------------------|---|
| デノーマル・オペランド
(DE) | 四則演算の時、少なくとも一方のデータがデノーマル数の時、セットされます。アンノーマル数に変換後、演算を実行します。比較の時、少なくとも一方のデータがデノーマル数かアンノーマル数の時、セットされます。デノーマル数は、アンノーマル数に変換して、比較を実行します。 |
|---------------------|---|

ノット・ア・デシマル (NAD) 少なくとも一方のデータがノット・ア・デシマルの時、セットされます。
 不定を演算結果としてセットします。

C1, C0 比較実行後の結果を示します。

C1 C0	比 較 結 果
0 0	ソース・データ > デスティネーション・データ
0 1	ソース・データ < デスティネーション・データ
1 0	ソース・データ = デスティネーション・データ

3.5 ワーキング・エリア

このプログラムで共通に使用するワーキング・エリアには、以下のものがあります。

- MNTSソース・データの仮数部をストアする領域
- EXPSソース・データの指数部をストアする領域
- SGNSソース・データの符号をストアする領域
 正....."0" 負....."FF"
- STASソース・データのステータスをストアする領域
 ノーマル数, アンノーマル数....."0"
 ゼロ....."1"
 無限大....."2"
- MNTDデスティネーション・データの仮数部をストアする領域
- EXPDデスティネーション・データの指数部をストアする領域
- SGNDデスティネーション・データの符号をストアする領域
 正....."0" 負....."FF"
- STADデスティネーション・データのステータスをストアする領域
 ノーマル数, アンノーマル数....."0"
 ゼロ "1"
 無限大 "2"
 除算時の仮数部の一致 "3"
- WORK1.....演算用ワーク・エリア
- WORK2.....演算用ワーク・エリア
- COUNT1.....カウンタ
- COUNT2.....カウンタ

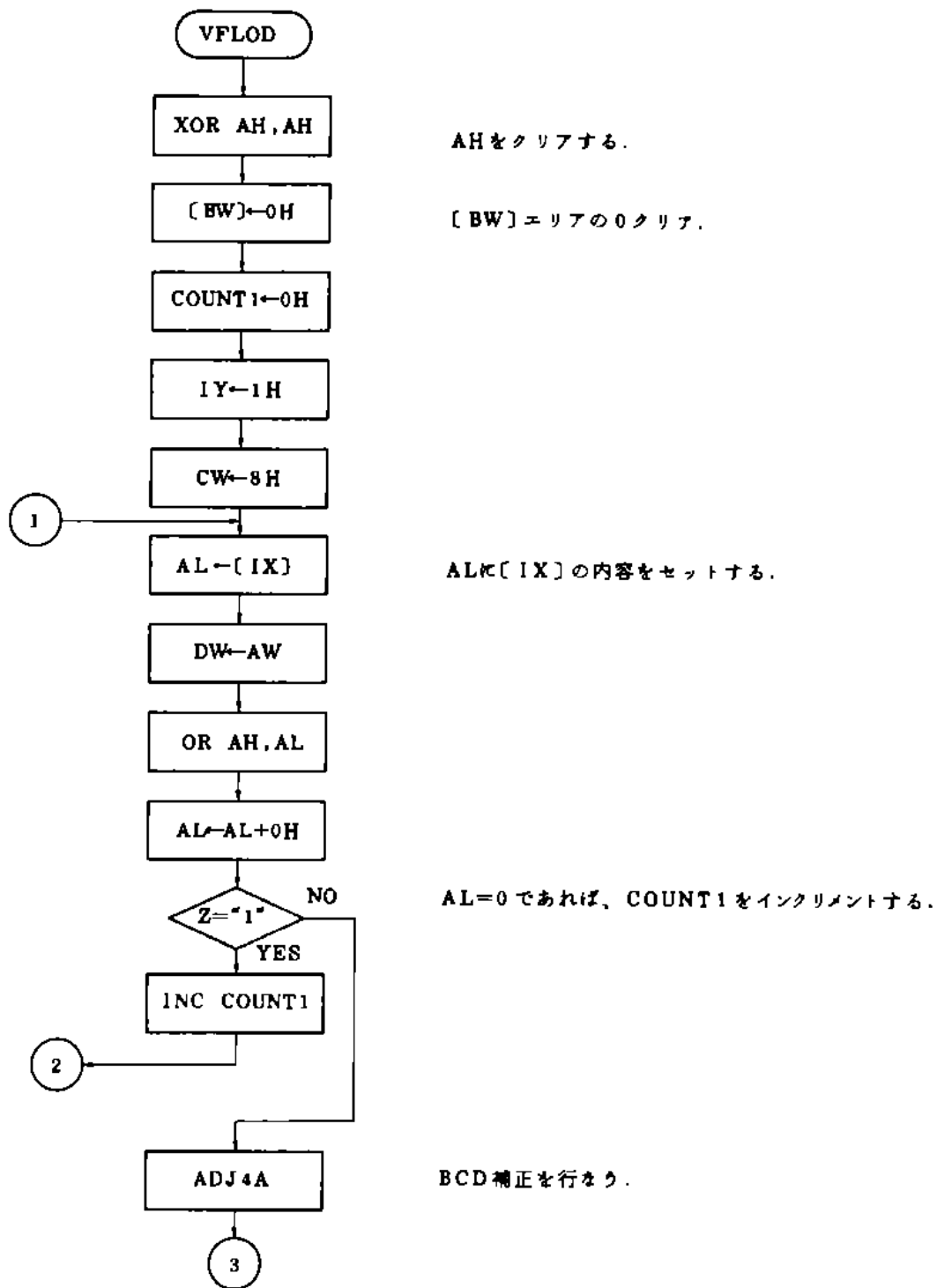
3.6 共通サブルーチン

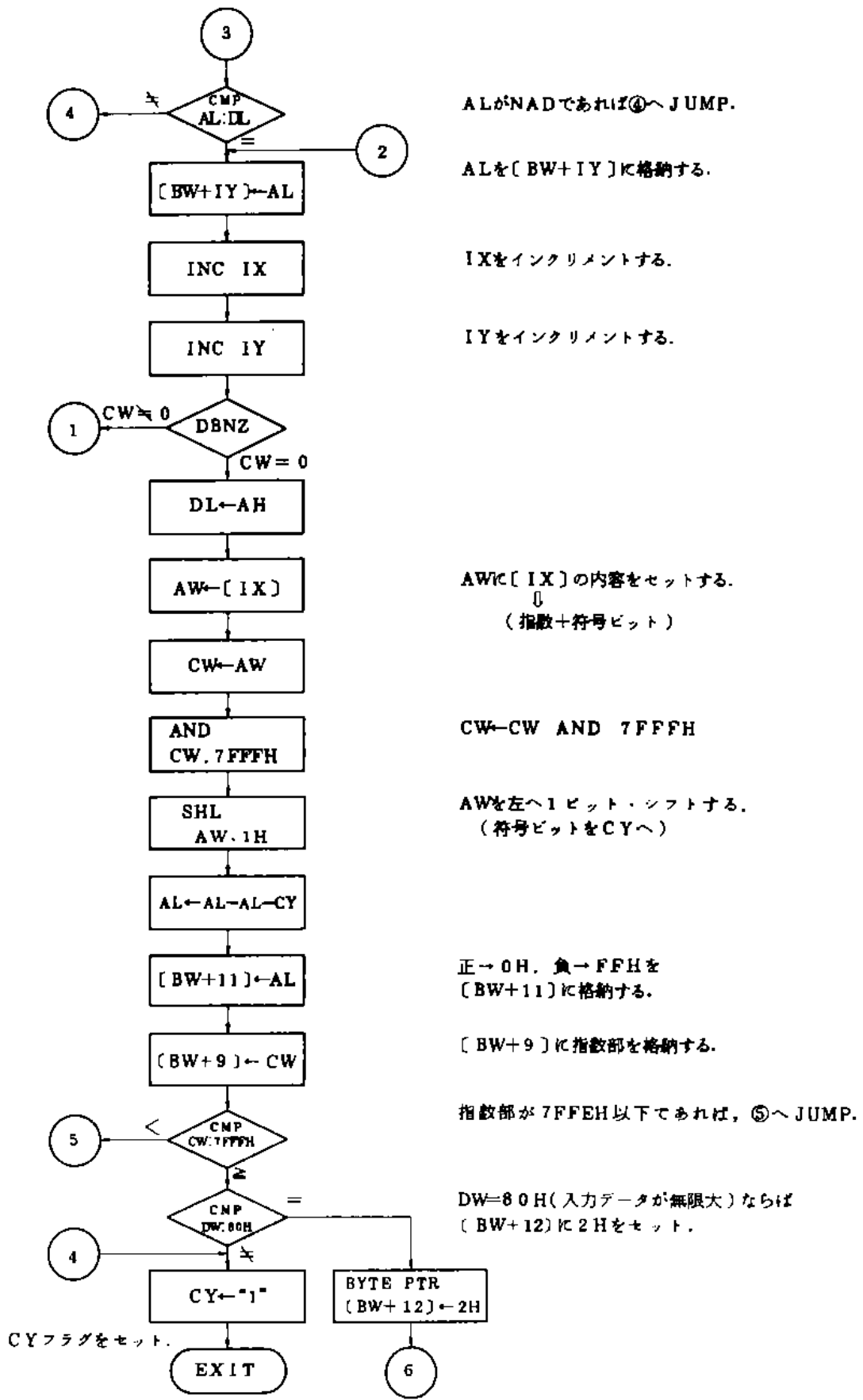
10進フローティング・ポイント・データ演算パッケージで共通に使用するサブルーチンとして以下のサブルーチンがあります。

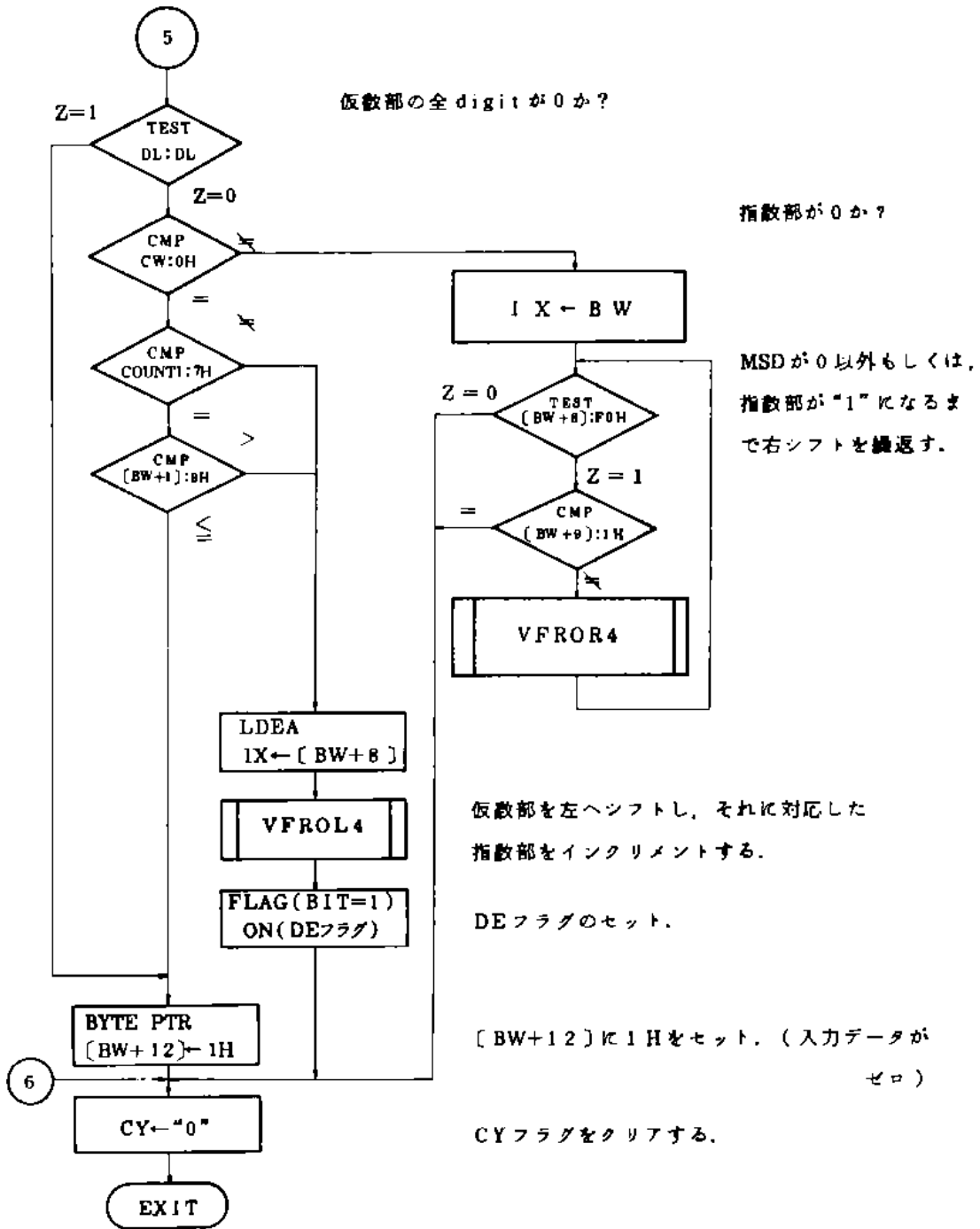
- VFL0D.....ソース・データあるいはデスティネーション・データを展開して、それぞれのワーキング・エリアにストアし数値のチェックを行なってステータスにセットします。さらにデータがNAD、不定の場合は、CYフラグをセットします。
- VFLDR.....デスティネーション・データおよびソース・データを展開して、それぞれのワーキング・エリアにVLF0Dを使ってストアします。その後ステータス、符号をレジスタにセットします。
 - AL.....ソース・データのステータス
 - AH.....デスティネーション・データのステータス
 - DL.....ソース・データの符号
 - DH.....デスティネーション・データの符号
- VFADSD.....ワーキング・エリアの仮数部を加算して、結果をワーキング・エリアにストアします。
- VFSBSD.....ワーキング・エリアの仮数部を減算して、結果をワーキング・エリアにストアします。
- VFCPSD.....ワーキング・エリアの仮数部を減算して、結果をストアしないでCPUのフラグのみをセットします。
- VFMLSD.....ワーキング・エリアの仮数部を乗算して、結果をワーキング・エリアにストアします。
- VFDVSD.....ワーキング・エリアの仮数部を除算して、結果をワーキング・エリアにストアします。
- VFNOMA.....ワーキング・エリアに展開されたソース・データおよびデスティネーション・データの仮数部の最上位ディジットが、“0”以外になるまで仮数部を上位にシフトし、指数部をデクリメントします。
(ノーマル数に変換します。)
- VFXC0.....ワーキング・エリアのデータを指定されたバイト数分、交換します。
- VFMVSD.....IX, IY, CWで指定されたアドレス間でブロック転送を行ないます。
- VFCLR.....WORK2エリアを“0”クリアします。
- VFROL4.....ソース・データあるいはデスティネーション・データを展開したワーキング・エリアの仮数部を下位にシフトし、指数部をインクリメントします。
- VFROR4.....ソース・データあるいはデスティネーション・データを展開したワーキング・エリアの仮数部を上位にシフトし、指数部をデクリメントします。
- VFROLW.....WORK2エリアを下位に1桁シフトします。
- VFZE.....デスティネーション・データにゼロをセットします。
- VFOE.....デスティネーション・データに無限大をセットします。
- VFIE.....デスティネーション・データに不定をセットします。
- VFSTR.....演算後、ワーキング・エリアの仮数部の丸めを行ない、デスティネーション・データにストアします。

3.7 共通サブルーチン・フローチャート

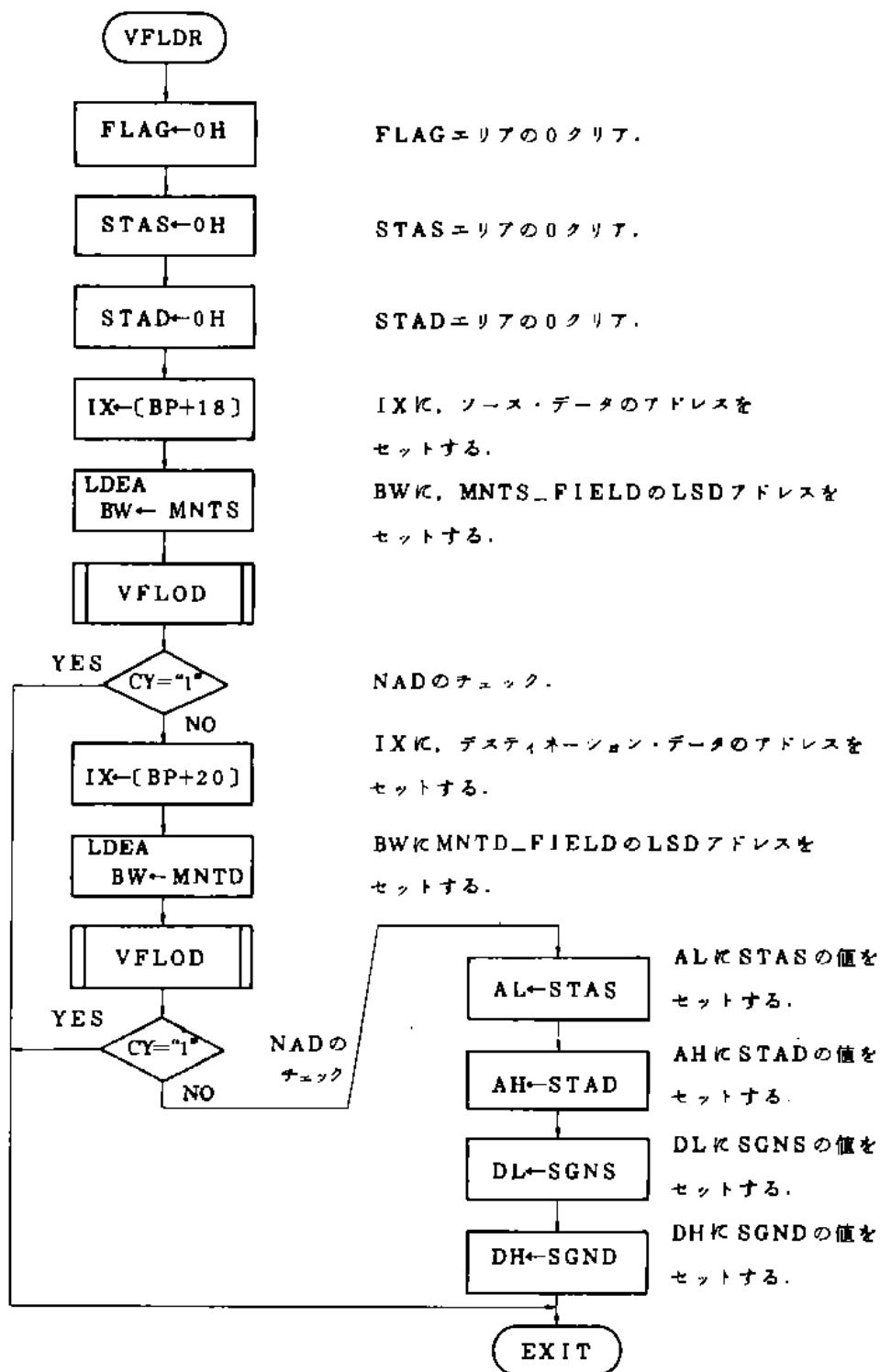
VFLOD (データの入力&展開)



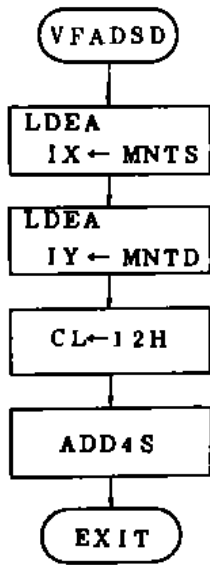




VFLDR (データの入力とステータスのセット)



VFADSD (仮数部の加算)



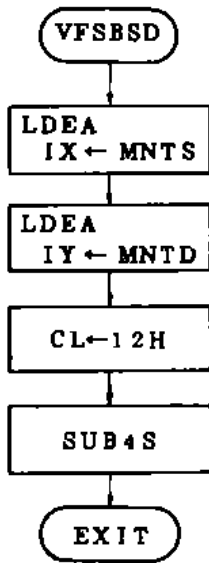
IXにMNTS_FIELDのLSDアドレスを
セットする。

IYにMNTD_FIELDのLSDアドレスを
セットする。

CLに12H (BCD桁数)をセットする。

IX、IYでアドレスされるバケットBCDス
tringをCLレジスタの値だけ加算し、結
果をIYでアドレスされるstringにストア
する。

VFSBSD (仮数部の減算)



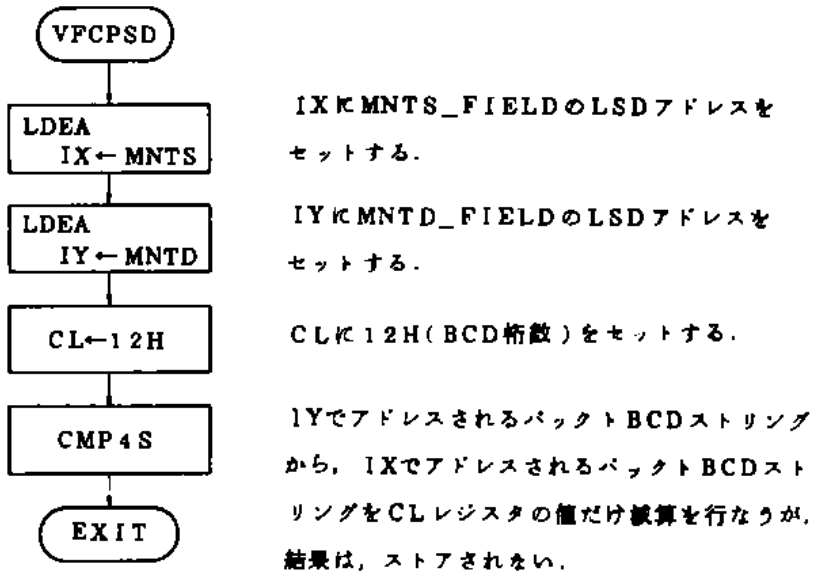
IXにMNTS_FIELDのLSDアドレスを
セットする。

IYにMNTD_FIELDのLSDアドレスを
セットする。

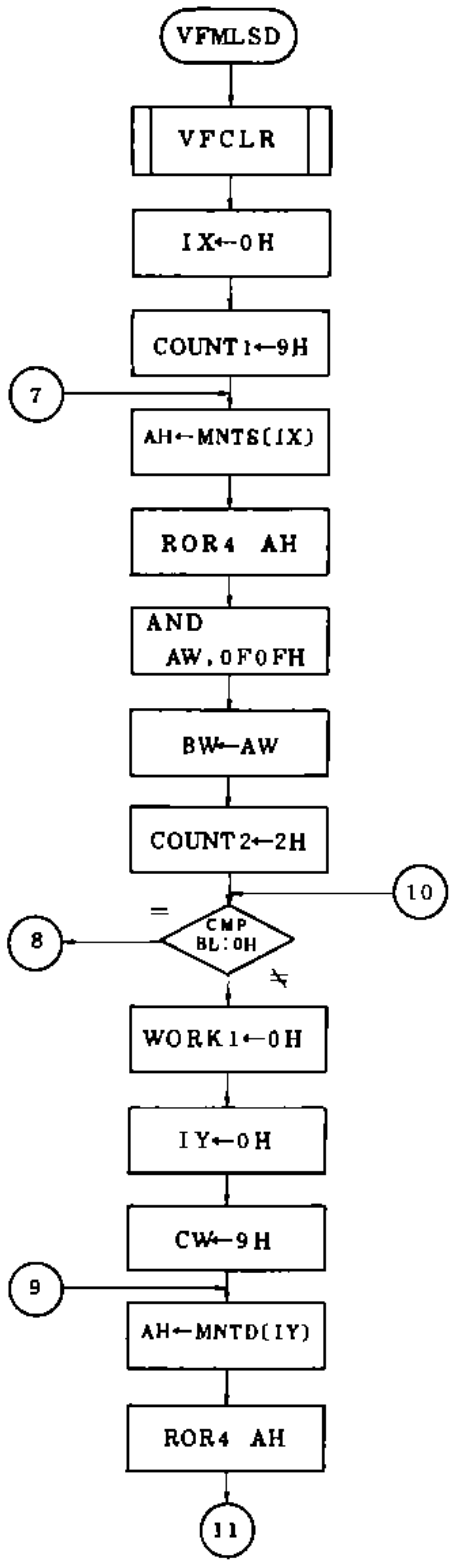
CLに12H (BCD桁数)をセットする。

IYでアドレスされるバケットBCDstring
から、IXでアドレスされるバケットBCDス
tringを、CLレジスタの値だけ減算し、結果を
IYでアドレスされるstringにストアする。

VFCPSD(仮数部の仮減算)



VFMLSD (仮数部の乗算)



WORK2_FIELDの0クリア.

AHに、MNTS[IX]をセットする.

AHをパックトBCDデータとして扱い、ALレジスタの下位4ビットを介して、1桁分右へ回転する.

AWの正規化.

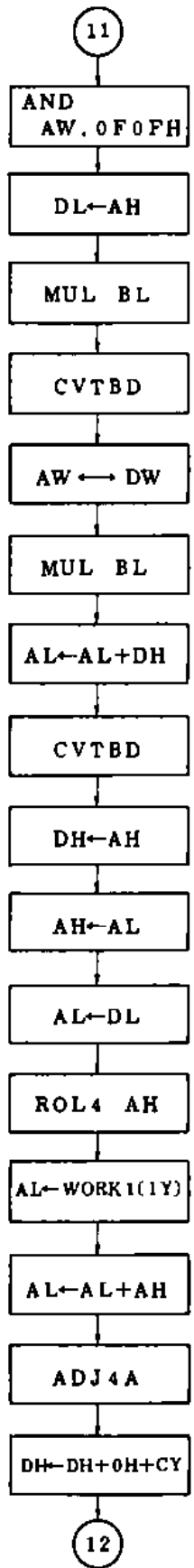
BL = 0か?

WORK1の0クリア.

WORK1の0クリア.

AHに、MNTD[IY]をセットする.

AHをパックトBCDデータとして扱い、ALレジスタの下位4ビットを介して、1桁分右へ回転する.



AWの正規化.

$AW \leftarrow AL * BL$

レジスタALの2進数8ビットを、2桁のアンパクトBCDに変換.

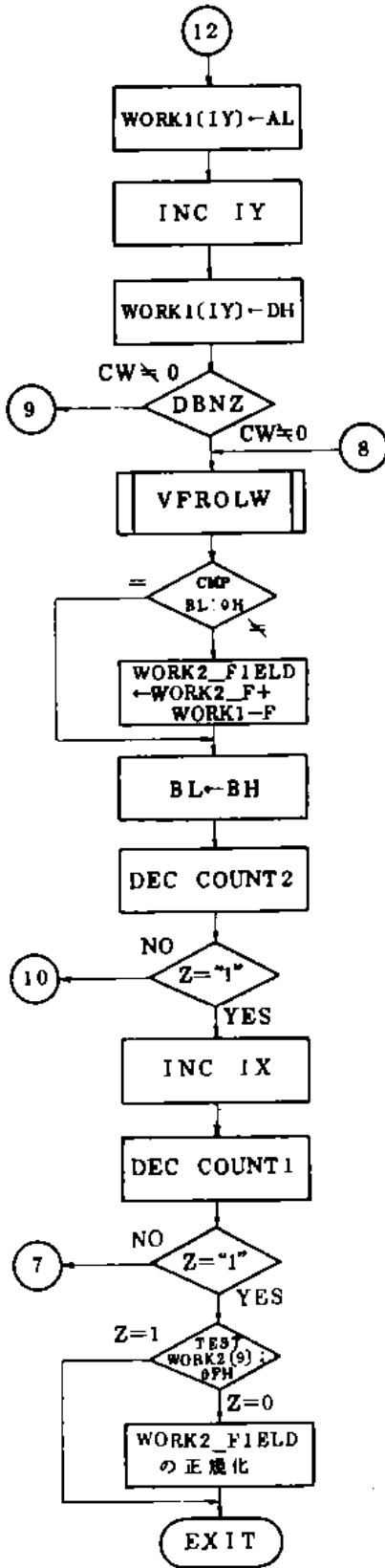
AWレジスタとDWレジスタの交換.

$AW \leftarrow AL * BL$

レジスタALの2進数8ビットを、2桁のアンパクトBCDに変換する.

AHをパケットBCDデータとして扱い、ALレジスタの下位4ビットを介して、1桁分左へ回転する.

BCD補正を行なう.



IYをインクリメントする。

WORK2_FIELDを左へシフトする。

BL=0か？

WORK2_FIELD ← WORK2_FIELD
+ WORK1_FIELD

COUNT2をデクリメントする。

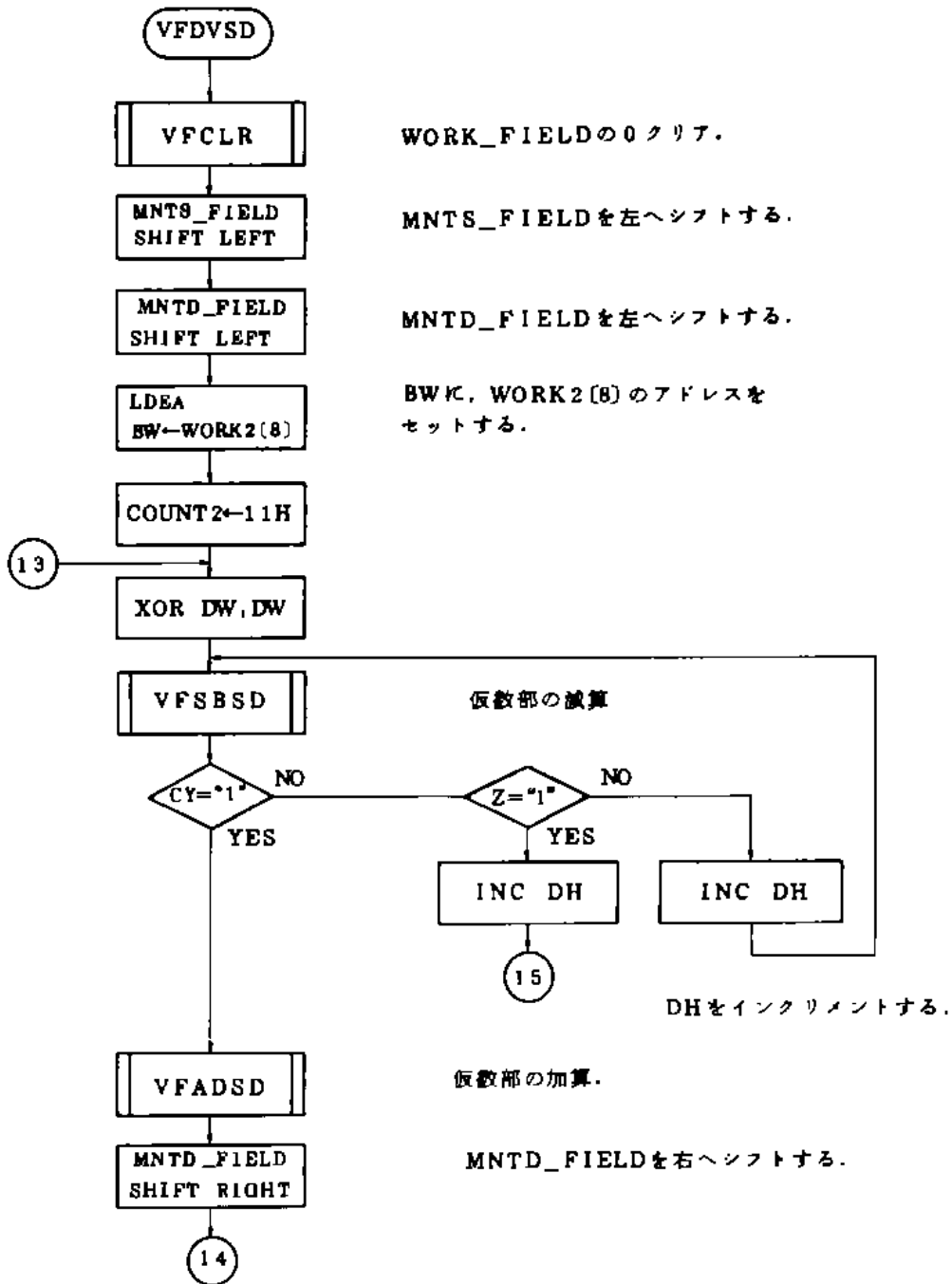
IXをインクリメントする。

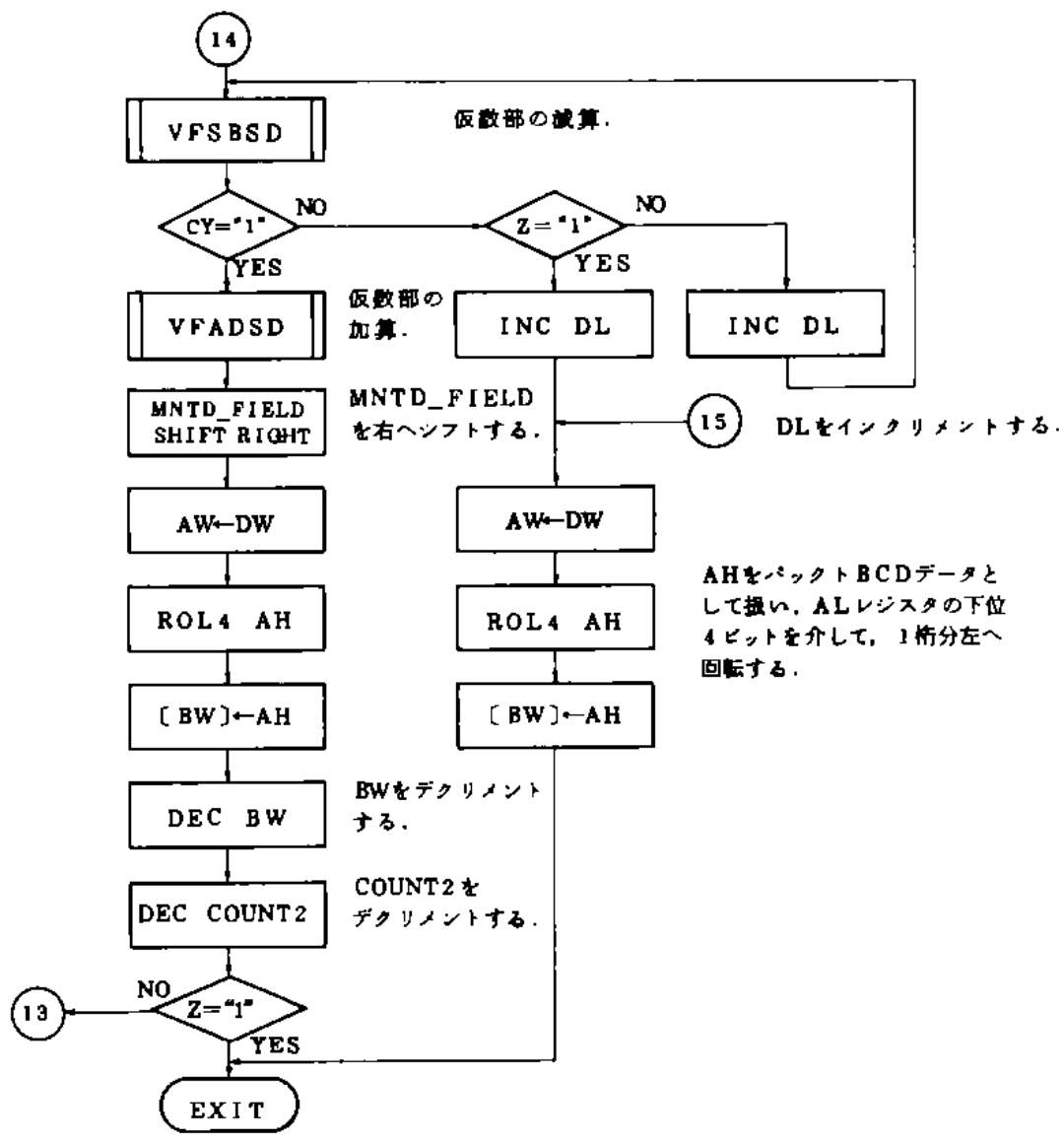
COUNT1をデクリメントする。

WORK2(9)の上位digitが0か？

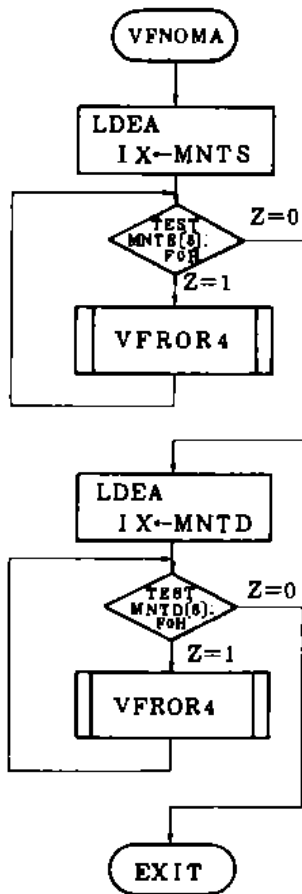
WORK2_FIELDを左へシフトし、
EXPD(指数部)をインクリメントする。

VFDVSD (仮数部の除算)





VFNOMA (乗除算における仮数部の正規化)



IXにMNTS-FIELDのLSD

アドレスをセットする。

MSDが0以外になるまでMNTS-FIELD

を右へシフトする。

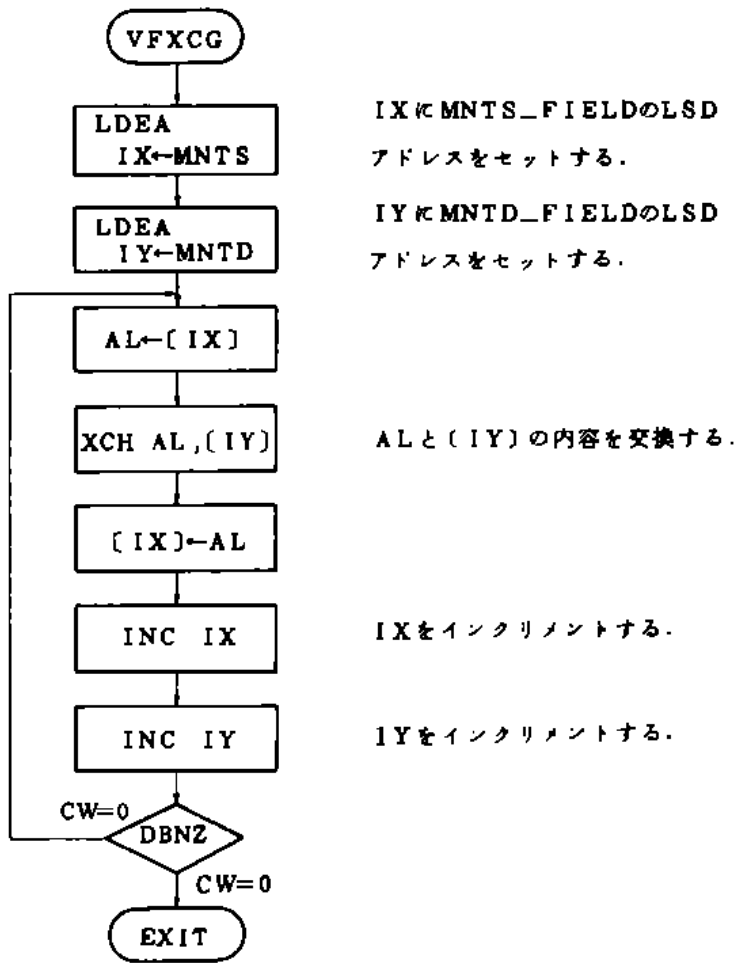
IXにMNTD-FIELDのLSDアドレスを

セットする。

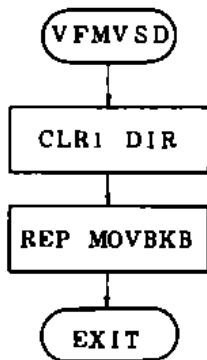
MSDが0以外になるまでMNTD-FIELD

を右へシフトする。

VFXCG (データの交換)



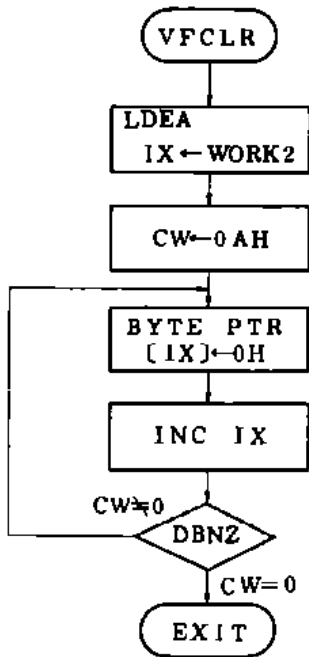
VFMVSD (ブロック転送)



DIRフラグをクリアする。

IXでアドレスされるブロックを、IYでアドレスされるブロックへバイト単位で、転送する。

VFCLR (WORK2エリアの0クリア)

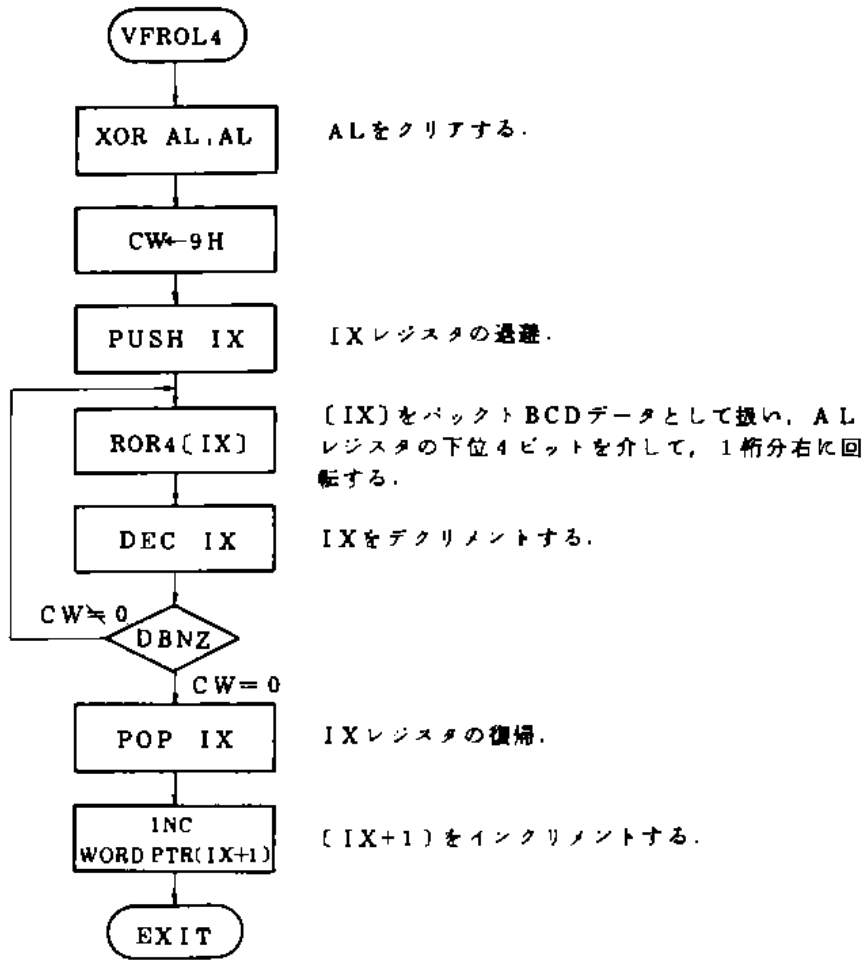


IXにWORK2のLSDアドレスをセットする。

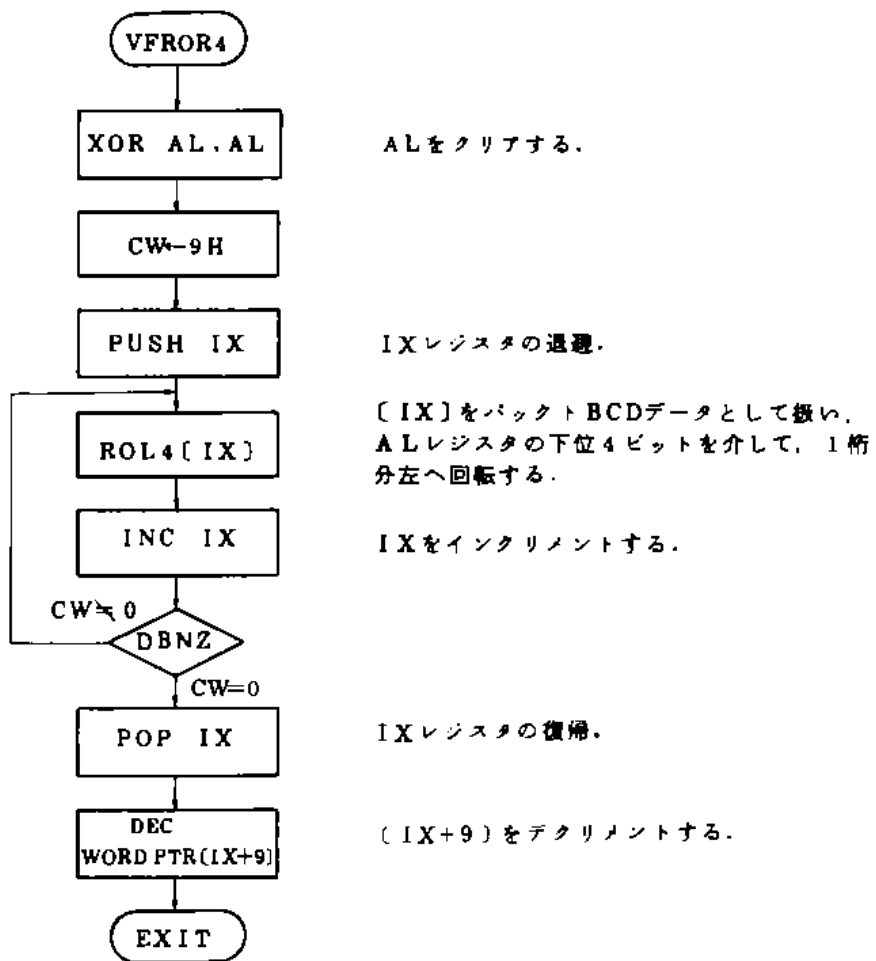
[IX]を0クリアする。

IXをインクリメントする。

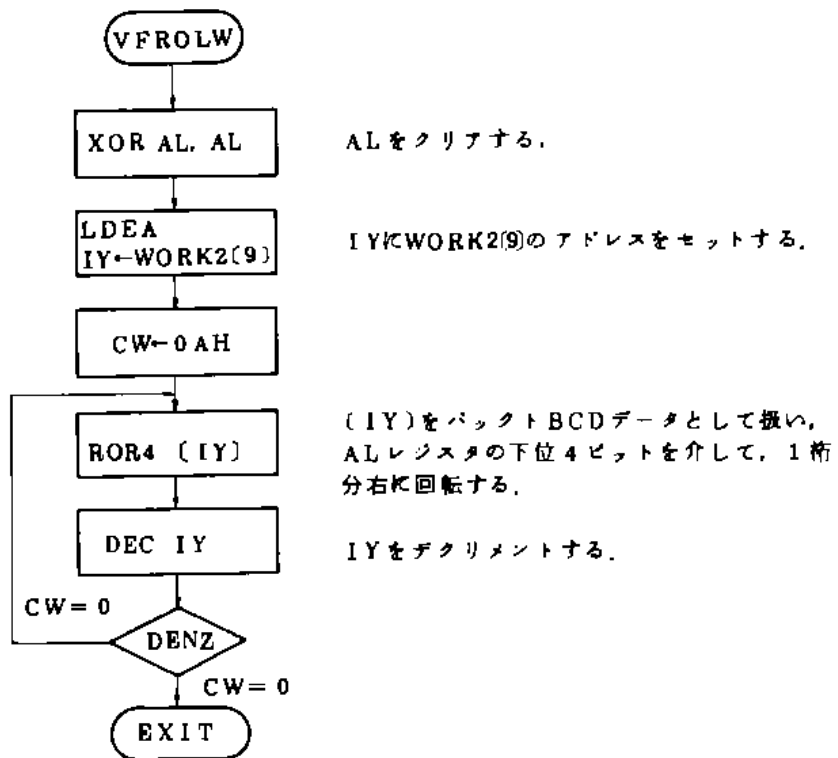
VFROL4 (仮数部を1/10倍して、指数部をインクリメント)



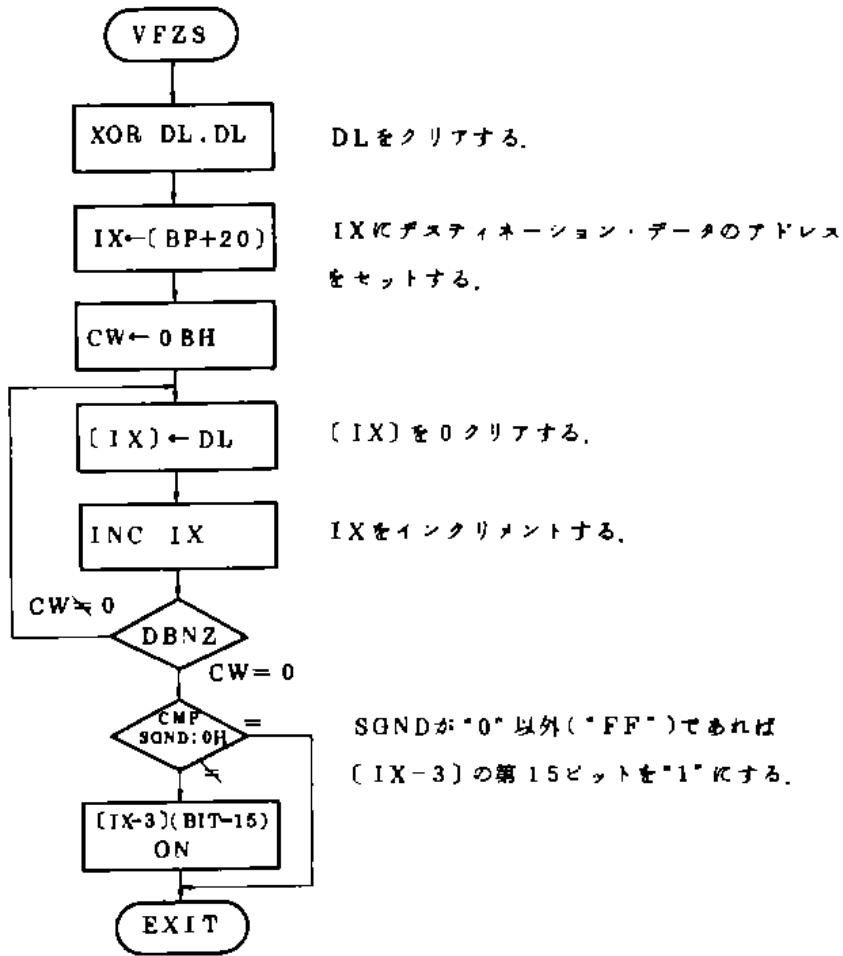
VFROR4 (仮数部を10倍して, 指数部をデクリメント)



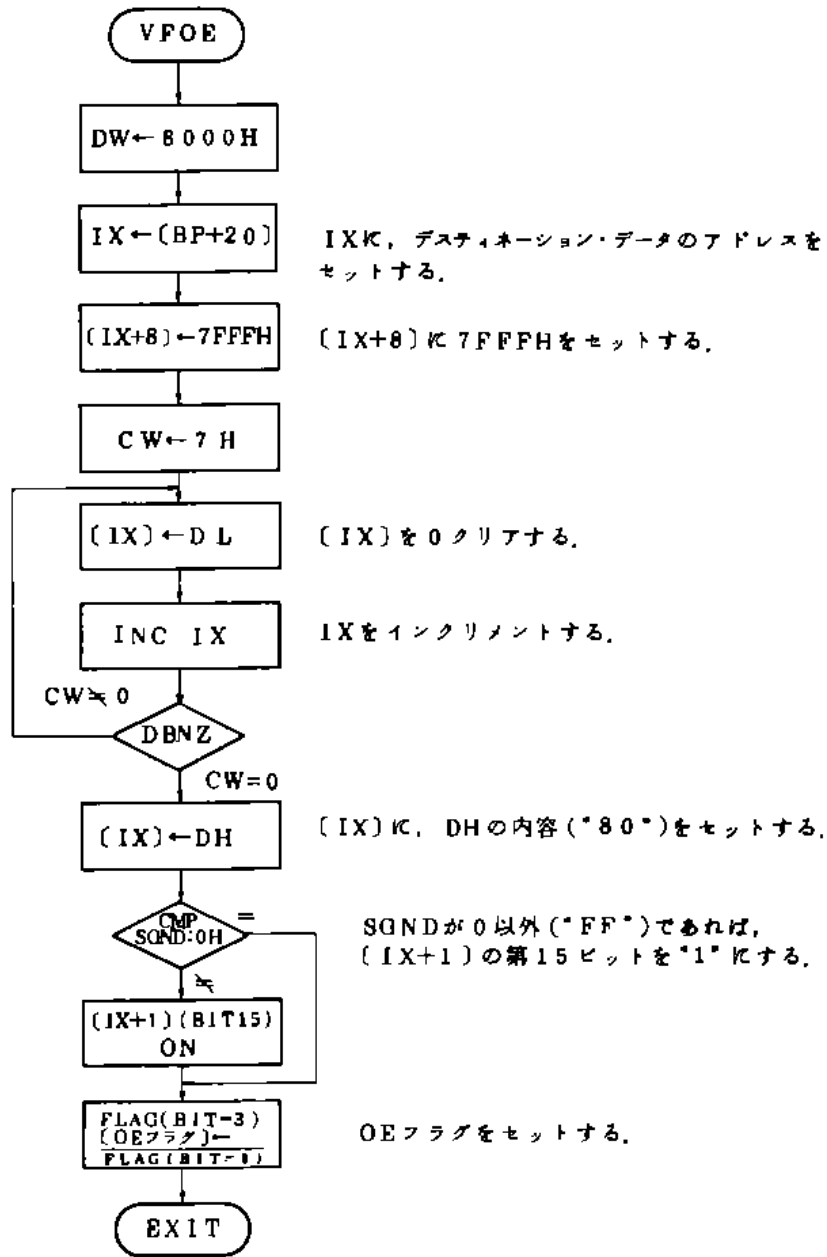
VFROLW(WORK2エリアを1/10倍)



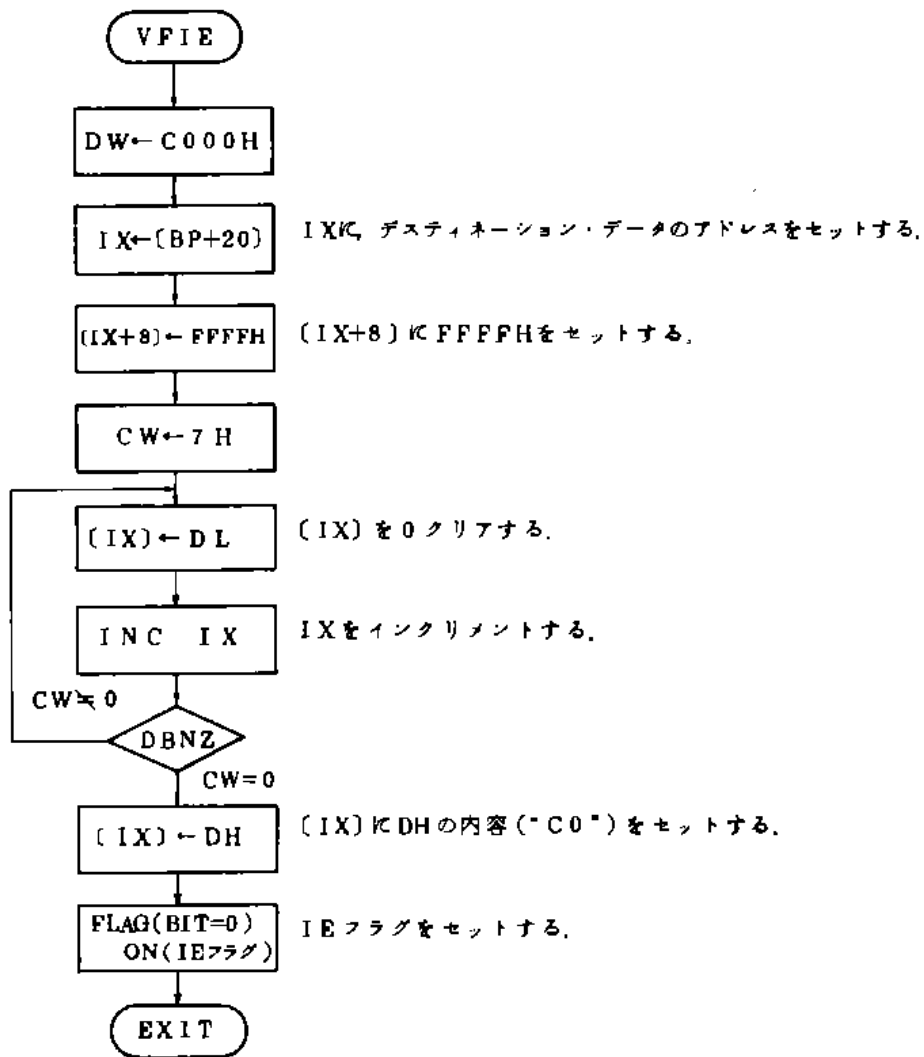
VFZS (デスティネーション・データにゼロをセット)



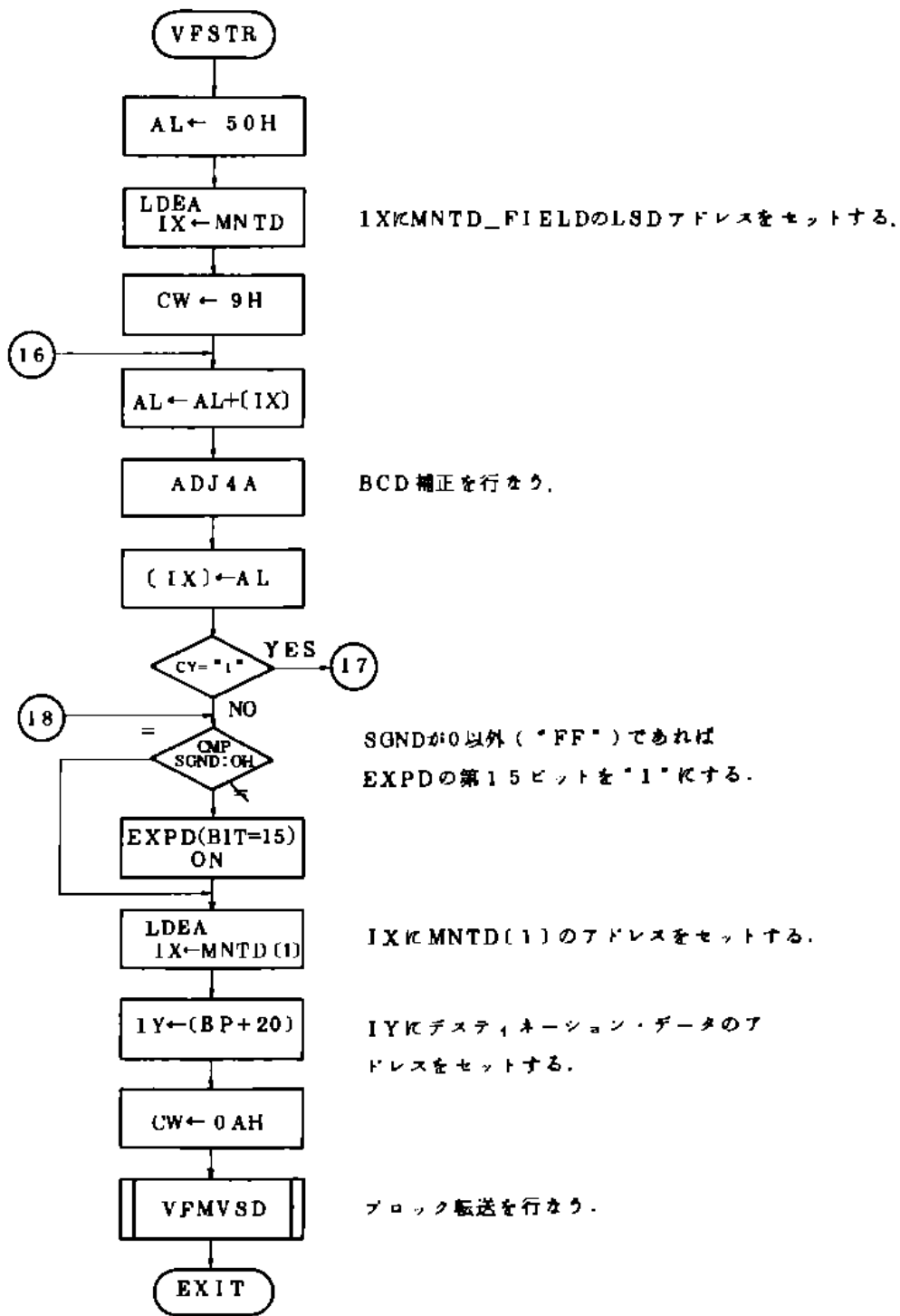
VFOE (デスティネーション・データに無限大をセット)

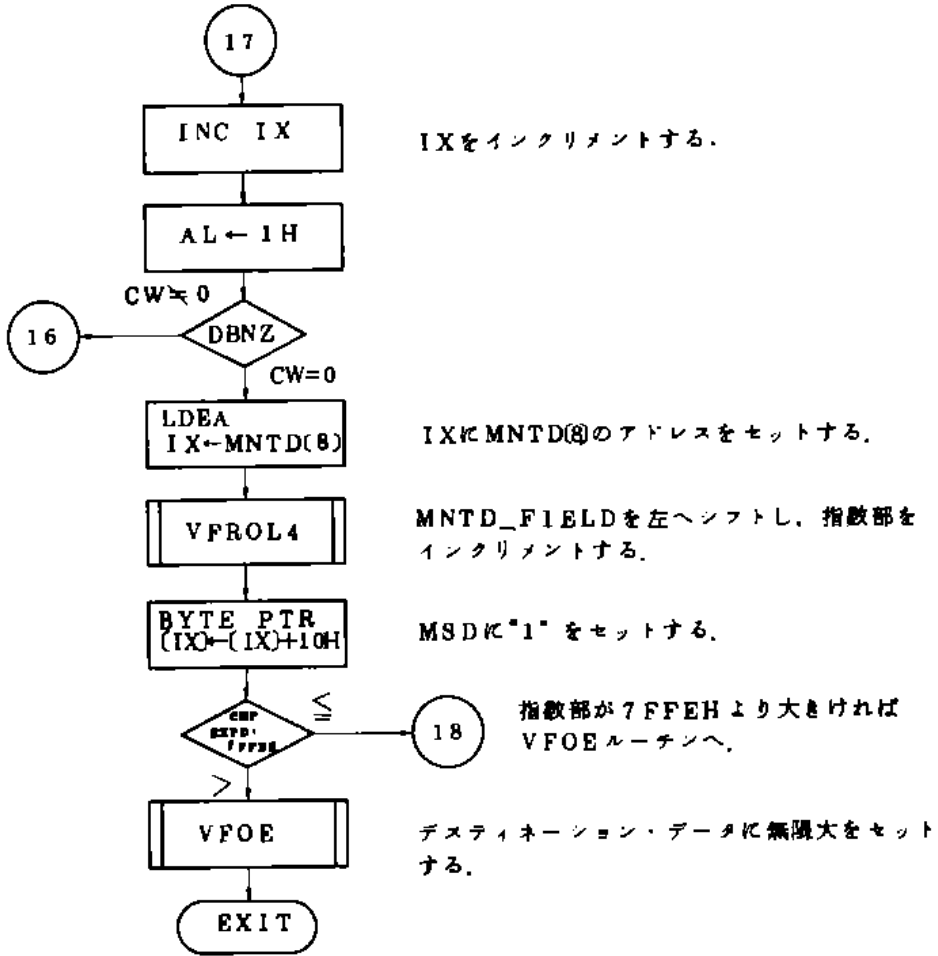


VFIE (デスティネーション・データに不定をセット)

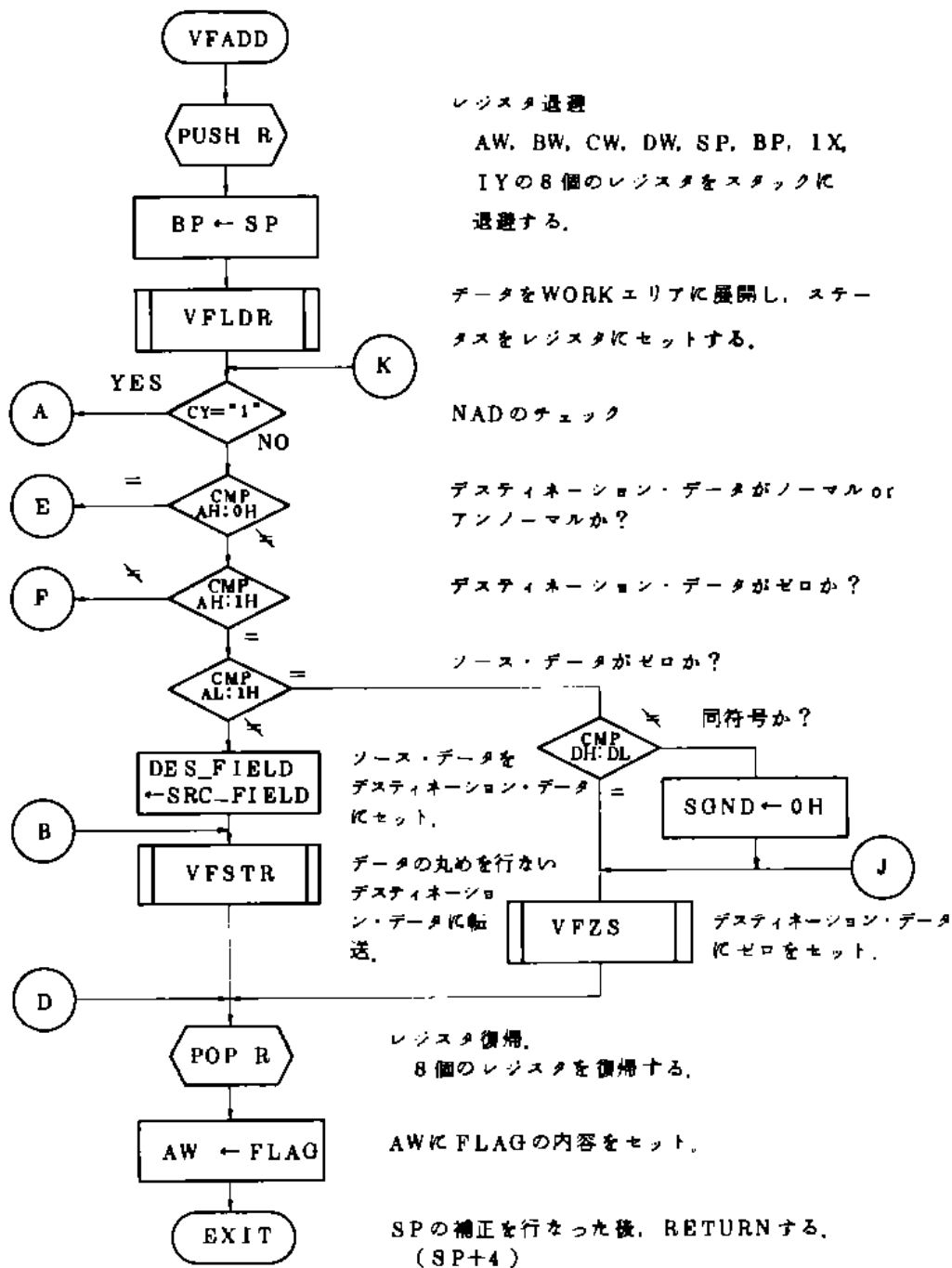


VFSTR (演算後の丸めと結果のストア)

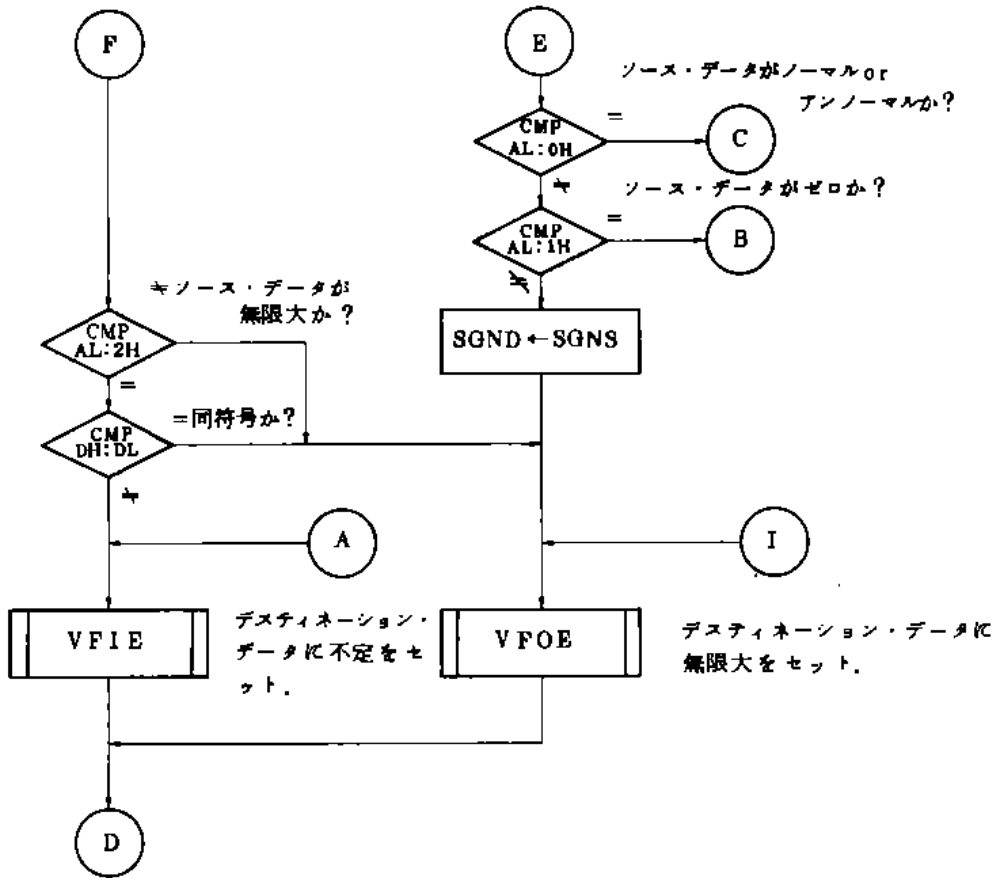




3.8 VFADD サブルーチン・フローチャート

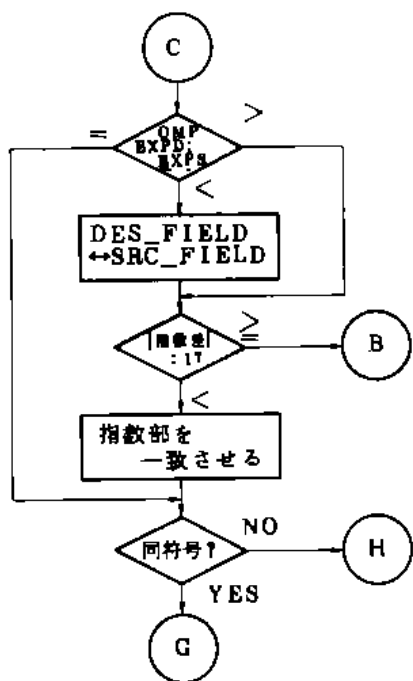


(加減算の例外処理)



AH	AL	PROCESS
0	0	正規の演算処理ルーチンへ
0	1	演算を行なわないでVFSTRルーチンへ
0	2	VFOEルーチンへ
1	0	SRC_FIELDをDES_FIELDへ転送後 VFSTRルーチンへ
1	1	VFZSルーチンへ
1	2	VFOEルーチンへ
2	0	VFOEルーチンへ
2	1	VFOEルーチンへ
2	2	同符号であればVFOE, 異符号ではVFIEへ
0 → ノーマル or アンノーマル 1 → ゼロ 2 → 無限大		

(加減算時の前処理)



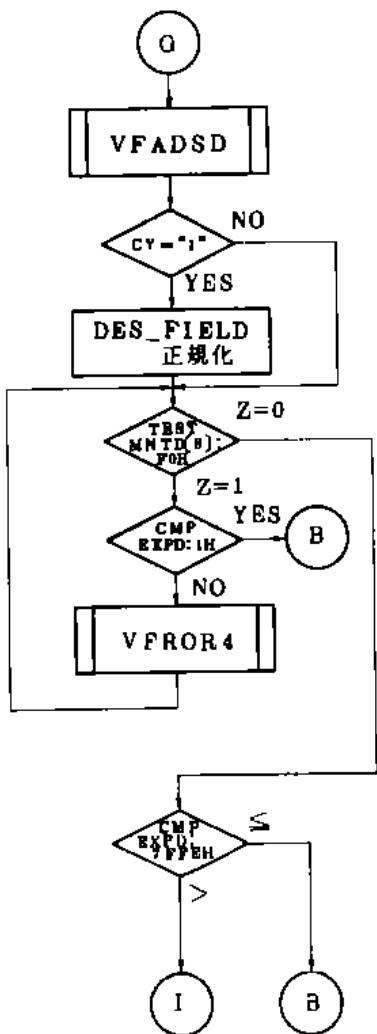
指数部の比較を行ない、大きい方を DES_FIELD(MNTD, EXPD, SGND)に格納する。

指数差が17以上であれば、演算を行なわないで戻る。

SRC_FIELDを左へシフトして指数部を一致させる。

符号の比較(SGND, SGNS)を行ない同符号であれば加算を、異符号であれば減算を行なり。

(加算処理)

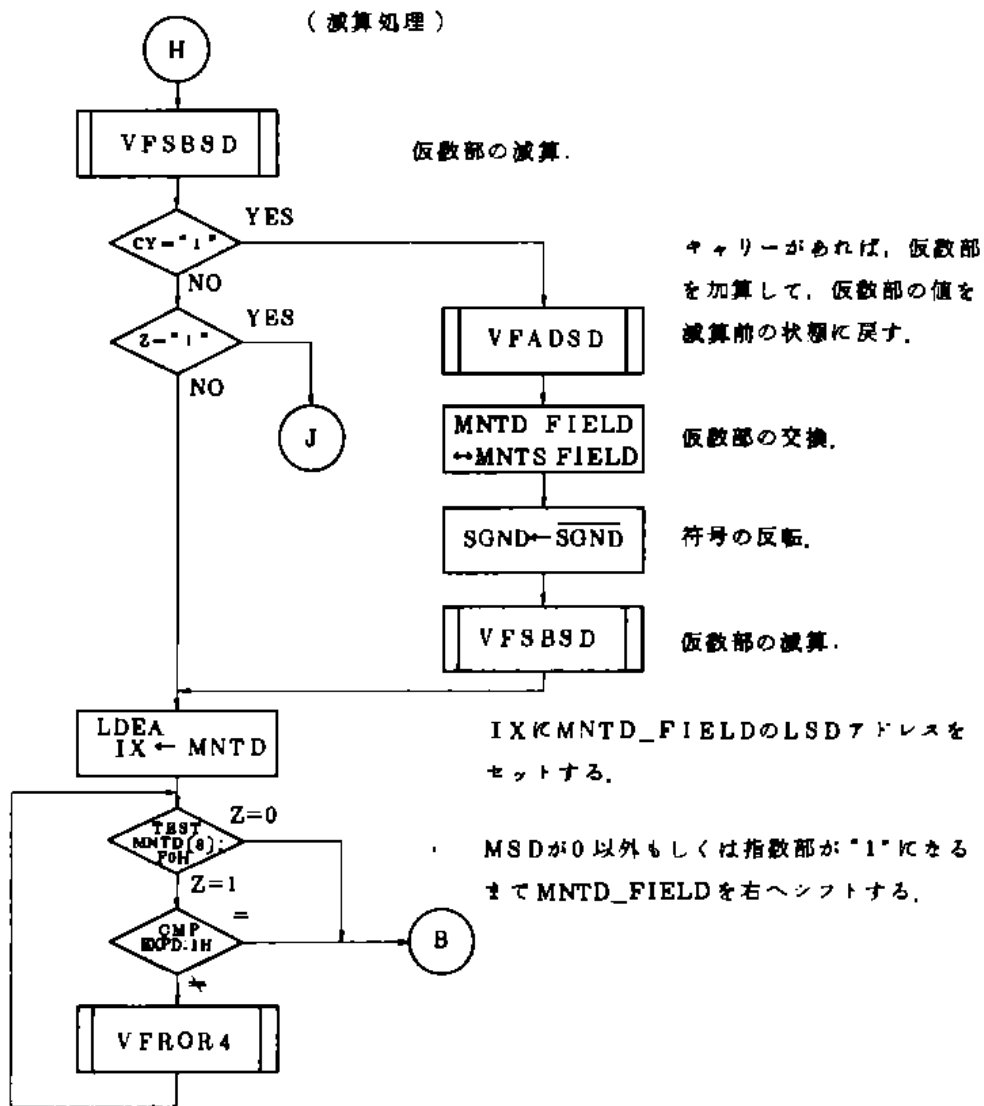


仮数部の加算.

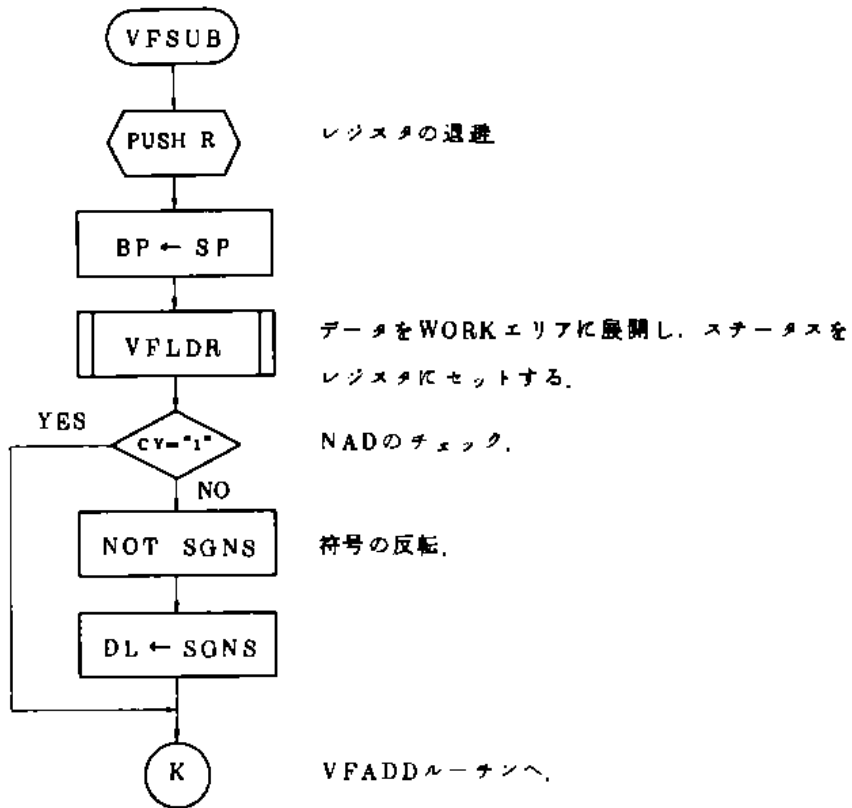
最上位ディジットからの桁上りがあれば、MNTD_FIELDを左へシフトし、指数部をインクリメントする。そして、MSDに"1"をセットする。

MSDが0以外もしくは指数部が"1"になるまでMNTD_FIELDを右へシフトする。

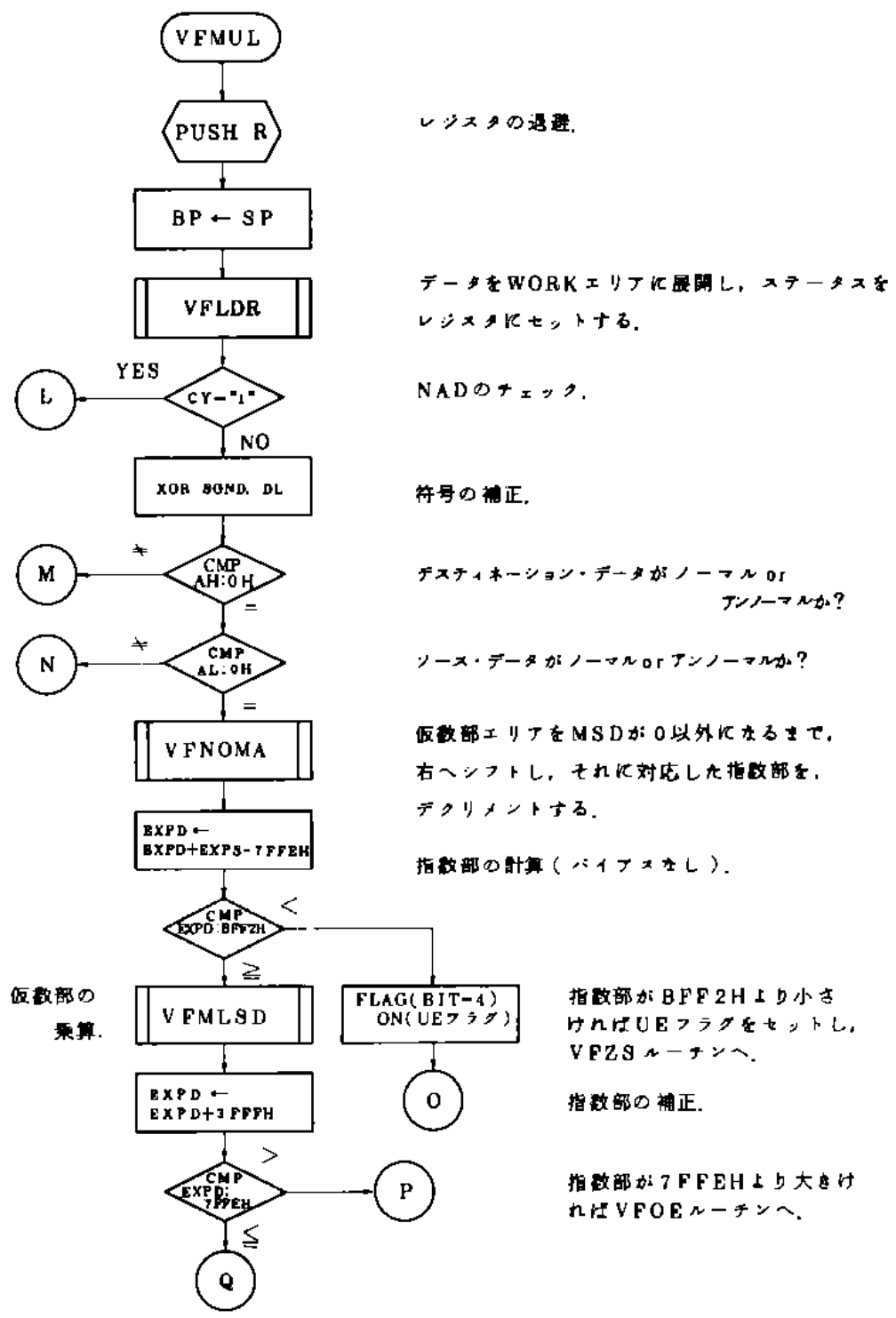
指数部が7FPEHより大きければVFOEルーチンへ。



3.9 VFSUBサブルーチン・フローチャート



3.10 VFMULサブルーチン・フローチャート



レジスタの退避.

データをWORKエリアに展開し、ステータスをレジスタにセットする.

NADのチェック.

符号の補正.

デスティネーション・データがノーマル or アンノーマルか?

ソース・データがノーマル or アンノーマルか?

仮数部エリアをMSDが0以外になるまで、右へシフトし、それに対応した指数部を、デクリメントする.

指数部の計算 (バイアスなし).

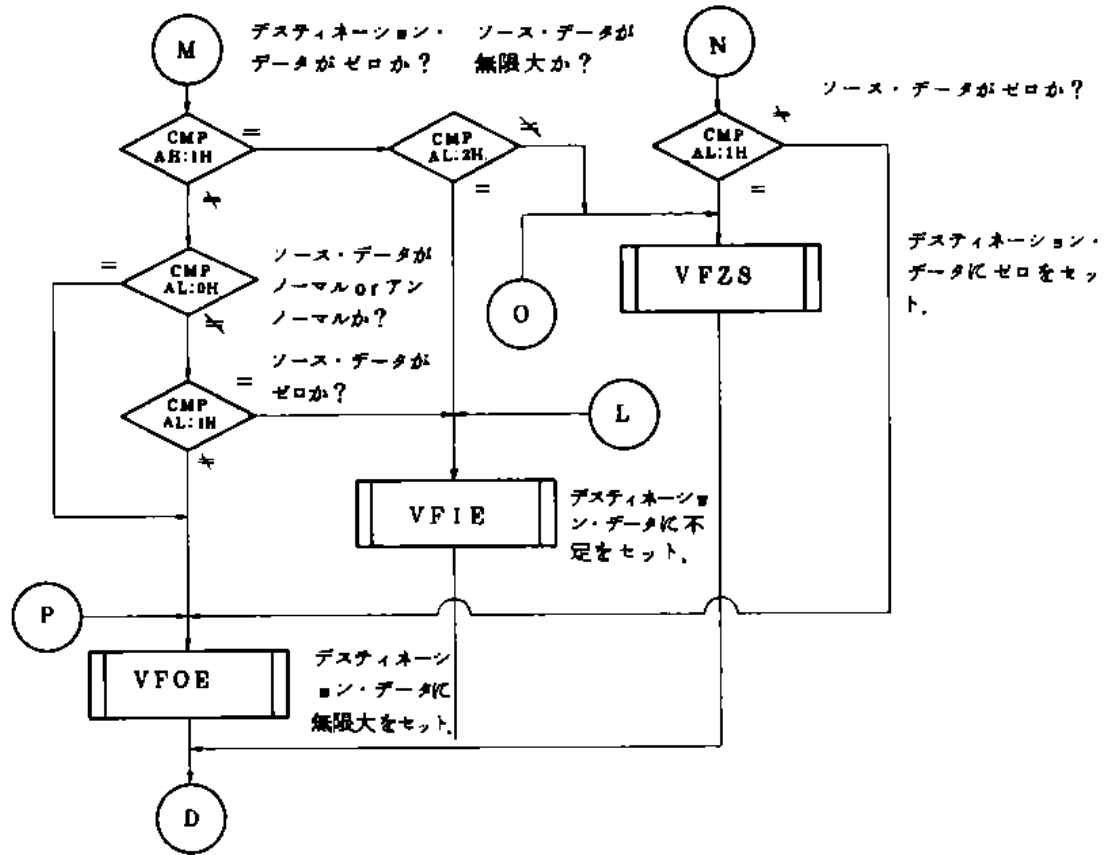
仮数部の乗算.

指数部がBFF2Hより小さければUEフラグをセットし、VFZSルーチンへ.

指数部の補正.

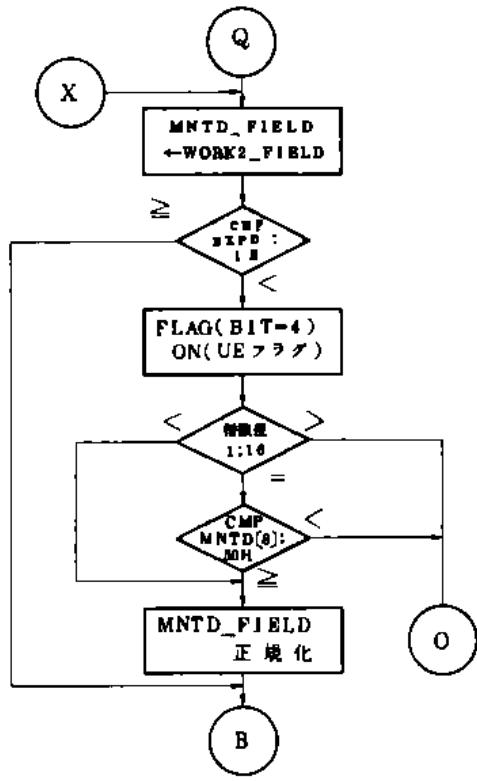
指数部が7FFEHより大きければVFOEルーチンへ.

(乗算の例外処理)



AH	AL	PROCESS
0	0	正規の演算処理ルーチンへ
0	1	VFZSルーチンへ
0	2	VFOEルーチンへ
1	0	VFZSルーチンへ
1	1	VFZSルーチンへ
1	2	VFIEルーチンへ
2	0	VFOEルーチンへ
2	1	VFIEルーチンへ
2	2	VFOEルーチンへ
0 → ノーマル or アンノーマル 1 → ゼロ 2 → 無限大		

(乗除算時の結果の丸め)



演算結果を、MNTD_FIELDへ転送する。

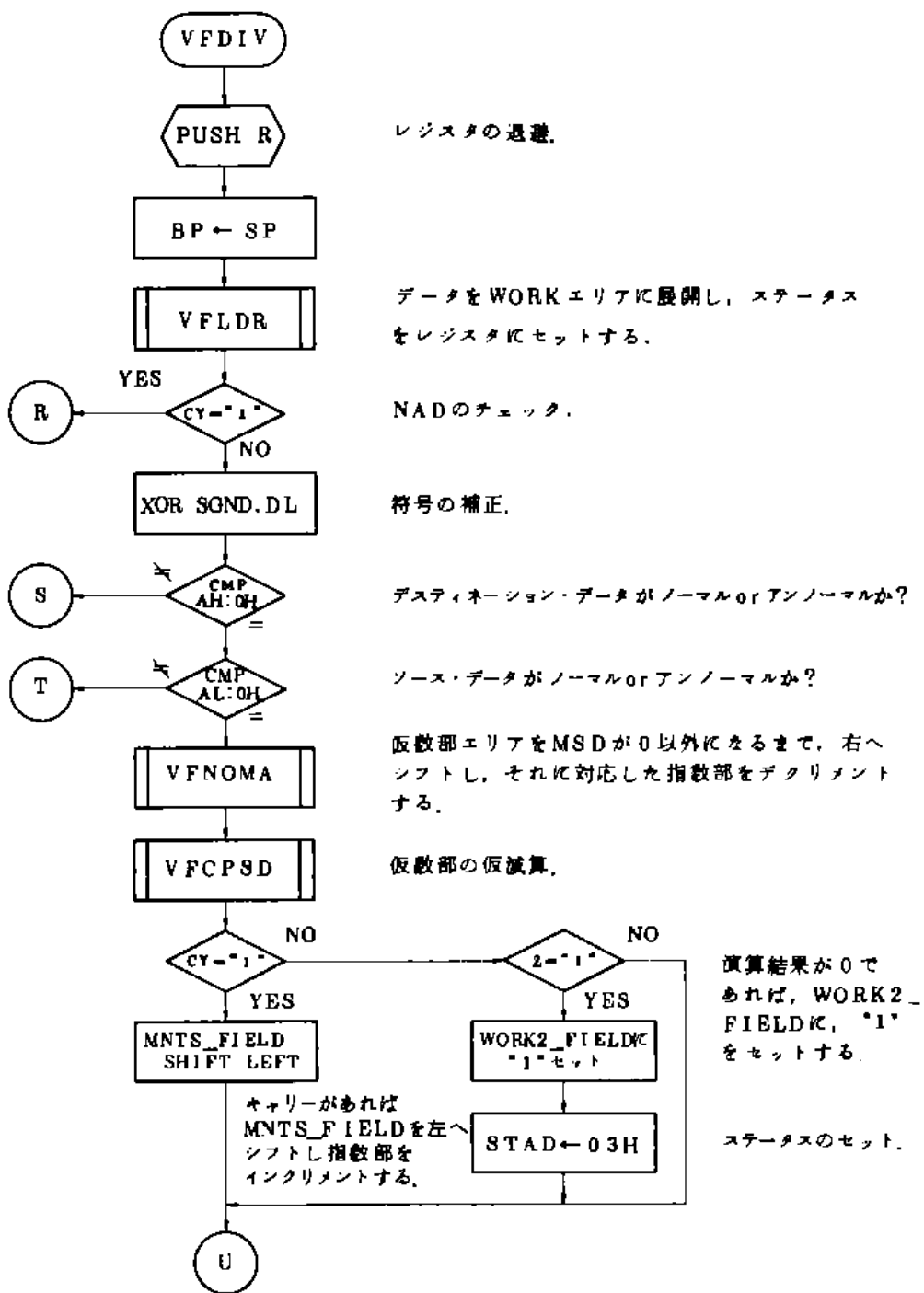
指数部が“1”以上であればVFSTRルーチンへ

UEフラグのセット。

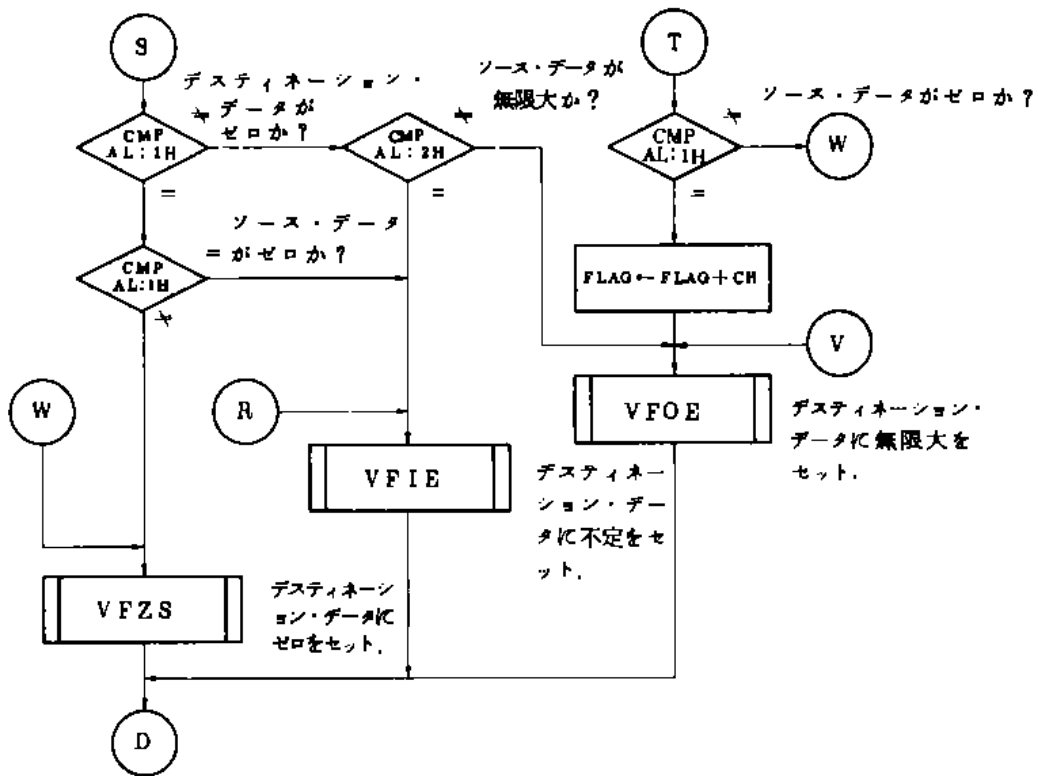
値“1”から指数部を減算し、指数差が16より大きければVFZSルーチンへ。

指数差が16より小さい場合、もしくは、指数差が16で、MSDが“5”以上であれば、MNTD_FIELDを左へシフトし、指数部をインクリメントする。

3.11 VFDIVサブルーチン・フローチャート



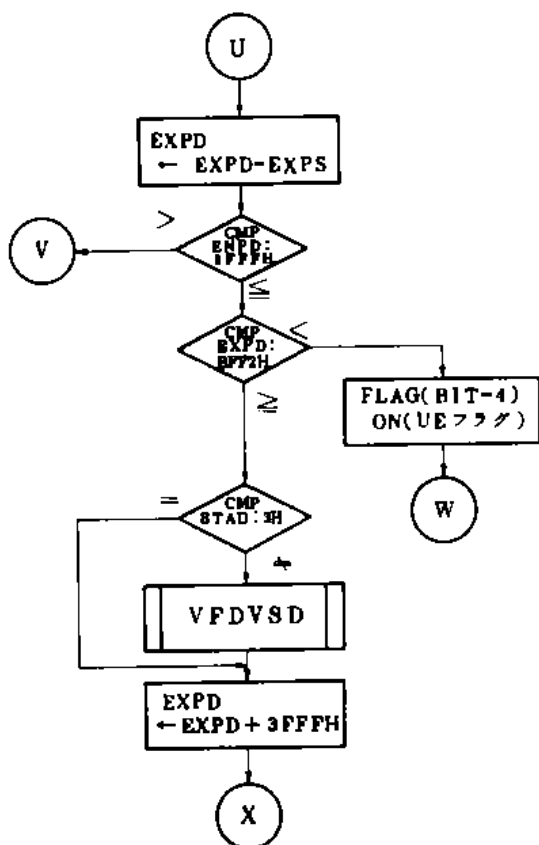
(除算の例外処理)



AH	AL	PROCESS
0	0	正規の演算処理ルーチンへ
0	1	DEフラグをセットし、VFOEルーチンへ
0	2	VFZSルーチンへ
1	0	VFZSルーチンへ
1	1	VFIEルーチンへ
1	2	VFZSルーチンへ
2	0	VFOEルーチンへ
2	1	VFOEルーチンへ
2	2	VFIEルーチンへ
3	×	演算を行なわないで、乗算正規化ルーチン

0 → ノーマル or アンノーマル 1 → ゼロ 2 → 無限大 3 → 仮数部が一致

(除算処理)



指数部の計算。

指数部が3FFFHより大きければ
VFOEルーチンへ。

指数部がBFF2Hより小さければ
UEフラグをセットし、VFZS
ルーチンへ。

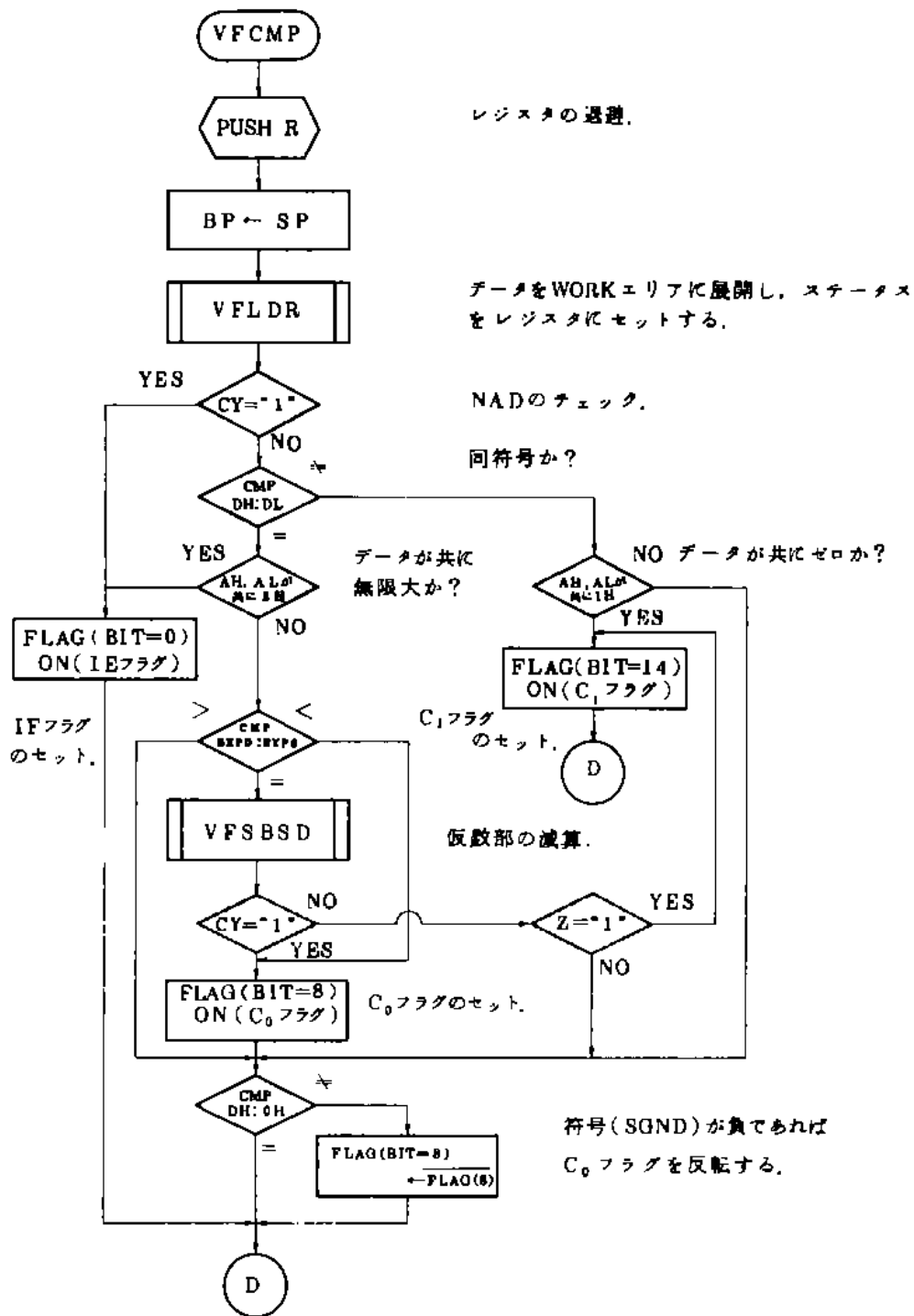
STAD≠3であればVFDVSD
ルーチンへ。

仮数部の除算。

指数部の補正。

VFMUL正規化ルーチンへ。

3.12 VFCMP サブルーチン・フローチャート



3. 13 10進フローティング演算パッケージ・プログラムのリスト

UPD709/70116 ASSEMBLER V1.0 VFPAK

DATE:24 Apr 83 17-22- 1 PAGE: 1

SOURCE FILE : A:VFPAK.SRC
OBJECT FILE : A:VFPAK.REL (VFPAK)
COMMAND : A:VFPAK.SRC AREF

```
LOC OBJ LINE SOURCE
1 ;*****
2 ;
3 ; FLOAING POINT ARITHMETIC PACKAGE FOR DECIMAL
4 ;
5 ; FLOAING POINT DATA FORMAT
6 ;
7 ; SIGN: 1 BIT
8 ; EXPONENTIAL: 15 BIT (BIASED BINARY,BIAS = 03FFFF)
9 ; MANTISSA: 64 BIT (16 DIGIT BCD)
10 ;
11 ; OFFSET +9 !S! EXP(HIGH) !
12 ; +8 ! EXP(LOW) !
13 ; +7 ! D15 ! D14 !
14 ; +6 ! D13 ! D12 !
15 ; : : :
16 ; : : :
17 ; 0 ! D1 ! D0 !
18 ;
19 ;
20 ; (-1) X SIGN X MANTISSA X (10 ** (EXPONENTIAL-03FFFF))
21 ;
22 ; COPYRIGHT (C) 1985 NEC CORPORATION
23 ;*****
24 ;
25 ; NAME VFPAK
26 ;
27 ; PUBLIC VFADD,VFSUB,VFMUL
28 ; PUBLIC VFDIV,VFCMP
29 ;
30 ; ASSURE PS:CODE,DSO:DATA
31 ;
32 ;*****
33 ; FLOAING POINT PACKAGE 'DATA'
34 ;*****
35 ;
36 ; DATA SEGMENT PUBLIC 'DATA'
37 ;
38 ;*****
39 ; TEMPORARY FIELD
40 ;*****
41 ;
42 ; MNTS DB 0 DUP(?) ;SOURCE MANTISSA FIELD
43 ;
44 ; EXP5 DB ? ;SOURCE EXPONENTIAL FIELD
45 ; SGN5 DB ? ;SOURCE SIGN FIELD
46 ; STAS DB ? ;SOURCE STATUS FIELD
47 ;
48 ; MNTD DB 0 DUP(?) ;DESTINATION MANTISSA FIELD
```

LOC	OBJ	L/ML	SOURCE				
)						
0016	????	43	EXP0	0V	?		;DESTINATION EXPONENTIAL FIELD
0018	??	49	SGND	0B	?		;DESTINATION SIGN FIELD
0019	??	50	STAD	0B	?		;DESTINATION STATUS FIELD
		51					
		52					
001A	????	52	FLAG	0V	?		;FLAG STATUS FIELD
		53					
001C	(10	54	WORK1	0B	10	DUP(?)	;WORK1 WORK FIELD
	??						
)						
0026	(10	55	WORK2	0B	10	DUP(?)	;WORK2 WORK FIELD
	??						
)						
0030	??	56	COUNT1	0B	?		;COUNT1 WORK FIELD
0031	??	57	COUNT2	0B	?		;COUNT2 WORK FIELD
		58					
----		59	DATA	ENDS			
		60					
		61	REJECT				

```

LOC  OBJ          LINE  SOURCE
                                62      ;
                                63      ;
                                64      ;
                                65      ;
                                66      ;
-----          67      CODE   SEGMENT PUBLIC 'CODE'
                                68      ;
                                69      ;
                                70      ;
                                71      ;
                                72      ;
                                73      ;
                                74      ;
                                75      ;
                                76      ;
0000          77      VFADD  PROC   NEAR
                                78      ;
0000 60          79      PUSH   R           ;SAVE REGISTERS
C001 40EC        80      MOV    BP,SP
C003 E27303      81      CALL  VFLDR           ;DATA LOAD & STATUS SET
                                82      ;
0006 7212        83      VFADD1: BC   VFADD3           ;IF CY = 1 THEN VFADD3
                                84      ;
000B 80FC00      85      VFADD2: CMP   AH,0H           ;IF AH = 0(H)
C008 7426        86      BZ    VFADD6           ; THEN VFADD6
                                87      ;
000B 50FC01      88      CMP   AH,01H          ;IF AH = 1(H)
C010 740E        89      BZ    VFADD4           ; THEN VFADD4
                                90      ;
0012 3C02        91      CMP   AL,02H          ;IF AL NOT = 2(H)
C014 7529        92      BNJ  VFADD7           ; THEN VFADD7
                                93      ;
0016 3AF2        94      CMP   DH,0L           ;IF DH = 0L
0018 7423        95      BZ    VFADD7           ; THEN VFADD7
                                96      ;
001A E83705      97      VFADD3: CALL  VFI   ;INDEFINITE SET
C018 E83790      98      BR   VFADD10          ;JUMP VFADD10
                                99      ;
0020 3C01        100     VFADD4: CMP   AL,01H          ;IF AL NOT = 1(H)
0022 7521        101     BZ    VFADD8           ; THEN VFADD8
                                102     ;
0024 3AF2        103     CMP   DH,0L           ;IF DH = 0L
C026 7403        104     BE   VFADD5           ; THEN VFADD5
                                105     ;
C028 C00E180000  * 106     MOV   SGR0,DH
                                107     ;
0028 28E204      108     VFADD5: CALL  VFI   ;ZERO SET
C030 E22490      109     BR   VFADD10          ;JUMP VFADD10
                                110     ;
C033 3C0C        111     VFADD6: CMP   AL,0H           ;IF AL = 0(H)
C035 7426        112     BZ    VFADD11          ; THEN VFADD11
                                113     ;
0037 3C01        114     CMP   AL,01H          ;IF AL = 1(H)
0039 741E        115     BZ    VFADD9           ; THEN VFADD9
                                116     ;

```

LOC	OBJ	LINE	SOURCE	
0039	88161800	R 117	MOV	SGMB,DL
		118		
003F	E3EAD4	119	VFADD7: CALL	VFDE ;INFINITY SET
0042	EB1290	120	BH	VFADD10 ;JUMP VFADD10
		121		
0045	803AC000	R 122	VFADD8: LDEA	IX,MNTS ;IX ← MNTS ADDR
0049	803E0000	R 123	LDEA	IY,MNTD ;IY ← MNTD ADDR
0040	890800	124	MOV	CW,0DH
0050	E87804	125	CALL	VFRVSD ;IY STRING ← IX STRING
		126		
0053	EB1A05	127	VFADD9: CALL	VFSTR ;ROUND & STORE
		128		
0056	81	129	VFADD1C: POP	R ;RESTORE REGISTERS
0057	A11AC0	R 130	MOV	AW,FLAG
005A	C20400	131	RET	A ;EXIT & SP ADJUST
		132		
005D	8B0E1600	R 133	VFADD11: MOV	CW,EXPD
0061	8B0EC900	R 134	CMF	CW,EXPS ;IF EXPD > EXPS
0063	770C	135	BH	VFADD12 ; THEN VFADD12
		136		
0067	741E	137	BZ	VFADD14 ;IF EXPD = EXPS THEN VFADD14
		138		
0069	890C00	139	MOV	CW,0CH
006C	E84CC4	140	CALL	VFXCG ;DES_FIELD ↔ SRC_FIELD
006F	8B0E1600	R 141	MOV	CW,EXPD
		142		
0073	7B0E0900	R 143	VFADD12: SUB	CW,EXPS
0077	85F911	144	CMF	CW,011H ;IF CW = > 1(CH)
007A	78D7	145	BGE	VFADD9 ; THEN VFADD9
		146		
007C	80360800	R 147	LDEA	IX,MNTS(B) ;IX ← MNTS(B) ADDR
		148		
0080	51	149	VFADD13: PUSH	CW ;SAVE CW
0081	E35CC4	150	CALL	VFR0L4 ;ROTATE LEFT & INC EXP.
0084	59	151	POP	CW ;RESTORE CW
0085	62F9	152	DBNZ	VFADD13 ;JUMP VFADD13 UNTIL CW = 0(CH)
		153		
0087	3AF2	154	VFADD14: CMF	DH,DL ;IF DH NOT = DL
0089	733C	155	BWZ	VFADD18 ; THEN VFADD18
		156		
0038	E805C2	157	CALL	VFADD8 ;ADDITION MNT.
008E	73D4	158	BMC	VFADD15 ;IF CY = 0 THEN VFADD15
		159		
0090	80361300	R 160	LDEA	IX,MNTD(B) ;IX ← MNTD(B) ADDR
0094	E849C4	161	CALL	VFR0L4 ;ROTATE LEFT & INC EXP.
0097	800410	162	ADD	BYTE PTR [IX],C10H
		163		
009A	8036C000	R 164	VFADD15: LDEA	IX,MNTD ;IX ← MNTD ADDR
		165		
009E	F6081500FD	R 166	VFADD16: TEST	MNTD(B),0FDH ;MNTD(B) AND 0(CH)
00A3	750C	167	BWZ	VFADD17 ;IF Z = 0 THEN VFADD17
		168		
00A5	533E160001	R 169	CMF	EXPD,01H ;IF EXPD = 1(CH)
00A4	7447	170	BZ	VFADD9 ; THEN VFADD9
		171		

LOC	OBJ	LINE	SOURCE		
004C	E24204	172	CALL	VFR0R4	; ROTATE RIGHT & DEC EXP.
004F	EBE9	173	BR	VFADD16	; JUMP VFADD16
		174	;		
00B1	313E1600FE7F	175	VFADD17: CMP	EXP0, 07FFEH	; IF EXP0 = < 7FFEH
00B7	769A	176	BNM	VFADD9	; THEN VFADD9
		177	;		
00B9	E884	178	BR	VFADD7	; JUMP VFADD7
		179	;		
00B8	E83202	180	VFADD18: CALL	VFS0SD	; SUBTRACTION MNT.
00B2	7205	181	BC	VFADD19	; IF CY = 1 THEN VFADD19
		182	;		
00C0	7713	183	BH	VFADD20	; IF Z = 0 THEN VFADD20
		184	;		
00C2	E968FF	185	BR	VFADD5	; JUMP VFADD5
		186	;		
00C5	E89802	187	VFADD19: CALL	VFADSD	; ADDITION MNT.
00C8	890900	188	MOV	CM, 09H	
00C9	E8E903	189	CALL	VFXCG	; DES_FIELD <-> SRC_FIELD
00CE	F6101800	190	NOT	SGND	; SGND <- NCT SGND
00D2	E89802	191	CALL	VFS0SD	; SUBTRACTION MNT.
		192	;		
00D3	3D360000	193	VFADD20: LDA	IX, MNT0	; IX <- MNT0 ADDR
		194	;		
00D9	F6061500F0	195	VFADD21: TEST	MNT0[B], 01CH	; MNT0[B] AND 01CH
00DE	750C	196	BNZ	VFADD22	; IF Z = 0 THEN VFADD22
		197	;		
00E0	835E1600J1	198	CMP	EXP0, 01H	; IF EXP0 = 1(H)
00E5	7405	199	BZ	VFADD22	; THEN VFADD22
		200	;		
00E7	E807C4	201	CALL	VFR0R4	; ROTATE RIGHT & DEC EXP.
00EA	EBE9	202	BR	VFADD21	; JUMP VFADD21
		203	;		
00EC	E964FF	204	VFADD22: BR	VFADD9	; JUMP VFADD9
		205	;		
		206	VFADD	ENDP	
		207	;		
		208	SEJECT		

LOC	OBJ	LINE	SOURCE
		209	;
		210	;
		211	;.....
		212	; FLDATING POINT SUBTRACTION
		213	;.....
		214	VFSUB PROC NEAR
		215	;
		216	PUSH R ;SAVE REGISTERS
		217	MOV BP,SP
		218	CALL VFLDR ;DATA LOAD & STATUS SET
		219	BC VFSUB1 ;IF CY = 1 THEN VFSUB1
		220	;
		221	NOT SGNS ;SGNS <- NOT SGNS
		222	MOV DL,SGNS
		223	;
		224	VFSUB1: BR VFA001 ;JUMP VFA001
		225	;
		226	VFSUB ENDP
		227	;
		228	SEJECT

LOC	OBJ	LINE	SOURCE
		229	;
		230	;
		231	;.....
		232	; FLOATING POINT MULTIPLICATION
		233	;.....
		234	;
C102		234	VFMUL PRCC NEAR
		235	;
O102 60		236	PUSH R ;SAVE REGISTERS
O103 5BEC		237	MOV BP,SP
O103 682302		238	CALL VFLDR ;DATA LOAD & STATUS SET
O103 7212		239	BC VFMUL1 ;IF CY = 1 THEN VFMUL1
		240	;
O10A 50161800	R	241	IOR SGND,0L
O10E 30FC0D		242	CMP AH,0H ;IF AH = 0(H)
O111 742C		243	JS VFMUL4 ; THEN VFMUL4
		244	;
O113 30FC01		245	CMP AH,01H ;IF AH NOT = 1(H)
O116 750A		246	BWZ VFMUL2 ; THEN VFMUL2
		247	;
O118 3C02		248	CMP AL,02H ;IF AL NOT = 2(H)
O11A 7511		249	BWZ VFMUL3 ; THEN VFMUL3
		250	;
O11C 6B3504		251	VFMUL1: CALL VJIE ;INDEFINITE SET
O11F 6936FF		252	BB VFADD10 ;JUMP VFADD10
		253	;
O122 3C0D		254	VFMUL2: CMP AL,0H ;IF AL = 0(H)
O124 7413		255	BZ VFMUL5 ; THEN VFMUL5
		256	;
O126 3C01		257	CMP AL,01H ;IF AL = 1(H)
O128 74F2		258	BZ VFMUL1 ; THEN VFMUL1
		259	;
O12A 6B0F9D		260	BR VFMUL5 ;JUMP VFMUL5
		261	;
O12D 68E203		262	VFMUL3: CALL VJIS ;ZERO SET
O130 6923FF		263	BR VFADD10 ;JUMP VFADD10
		264	;
O133 3C0D		265	VFMUL4: CMP AL,0H ;IF AL = 0(H)
O135 740A		266	JS VFMUL6 ; THEN VFMUL6
		267	;
O137 3C01		268	CMP AL,01H ;IF AL = 1(H)
O139 74F2		269	BZ VFMUL3 ; THEN VFMUL3
		270	;
O13B 68EE03		271	VFMUL5: CALL VJIE ;INFINITY SET
O13E 6913FF		272	BB VFADD10 ;JUMP VFADD10
		273	;
O141 6856C3		274	VFMUL6: CALL VJNOMN ;NORMALIZATION MNT.
O144 3816C90D	R	275	MOV DM,EXPS
O148 0116160D	R	276	ADD EXPD,DM
O14C 312E1600FE7F	R	277	SUB EXPD,07FFEH
O152 313E1600F20F	R	278	CMP EXPD,0BFF2H ;IF EXPD > 96F2(H)
O156 700E		279	SGE VFMUL7 ; THEN VFMUL7
		280	;
O15A 0F1D061A0024	R	281	SET1 FLAG,04H ;FLAG(BIT = 4) ← "1"
O16D 68C9		282	BR VFMUL3 ;JUMP VFMUL3
		283	;

LOC	OBJ		LINE	SOURCE	
0162	E82502		284	VFMUL7: CALL	VFMUL5 ;MULTIPLICATION MNT.
0165	8106160CF43F	R	285	ADD	EXP0,03FFFH
0168	813E160CF47F	R	286	CMP	EXP0,07FFEH
0171	7FC8		287	BGT	VFMUL5 ; THEN VFMUL5
			288	;	
0173	8D362600	R	289	VFMUL8: LDEA	IX,WORK2 ;IX <- WORK2 ADDR
0177	803E0D0C	R	290	LDEA	IT,MNT0 ;IT <- MNT0 ADDR
0178	890900		291	MOV	CW,09H
017E	E84005		292	CALL	VFMUL9
0181	833E160001	R	293	CMP	EXP0,01H
0186	7D28		294	BGE	VFMUL12 ; THEN VFMUL12
			295	;	
0188	0F10061A0004	R	296	SETT	FLAG,04H ;FLAG(BIT * 4) <- "1"
018E	990100		297	MOV	CW,01H
0191	200E1600	R	298	SUB	CW,EXP0
0195	83F910		299	CMP	CW,010H
0198	720E		300	BC	VFMUL10 ; THEN VFMUL10
			301	;	
019A	7402		302	9Z	VFMUL9 ;IF CW = 10(H) THEN VFMUL9
			303	;	
019C	E98F		304	BR	VFMUL3 ;JUMP VFMUL3
			305	;	
019E	803E150050	R	306	VFMUL9: CMP	MNT0[8],050H ;IF MNT0[8] < 50(H)
01A3	7288		307	BC	VFMUL3 ; THEN VFMUL3
			308	;	
01A5	90361500	R	309	VFMUL10: LDEA	IX,MNT0[2] ;IX <- MNT0[2] ADDR
			310	;	
01A9	51		311	VFMUL11: PUSH	CW ;SAVE CW
01AA	E83303		312	CALL	VFMUL4 ;ROTATE LEFT & INC EXP.
01AB	59		313	POP	CW ;RESTORE CW
01AE	E2F9		314	DBNZ	VFMUL11 ;JUMP VFMUL11 UNTIL CW = 0(H)
			315	;	
01B0	E9ACFE		316	VFMUL12: BR	VFMUL9 ;JUMP VFMUL9
			317	;	
			318	VFMUL ENDP	
			319	;	
			320	SEJECT	

LOC	OBJ	LINE	SOURCE
		321	;
		322	;
		323	;
		324	; FLOATING POINT DIVISION
		325	;
		326	VF DIV PROC NEAR
		327	;
		328	PUSH R ;SAVE REGISTERS
		329	MOV BP,SP
		330	CALL VFLDP ;DATA LOAD & STATUS SET
		331	BC VF DIV3 ;IF CY = 1 THEN VF DIV3
		332	;
		333	XOR SGND,DL
		334	CMP AH,0H ;IF AH = 0(N)
		335	BZ VF DIV4 ; THEN VF DIV4
		336	;
		337	CMP AH,01H ;IF AH = 1(N)
		338	BZ VF DIV1 ; THEN VF DIV1
		339	;
		340	CMP AL,02H ;IF AL = 2(N)
		341	BZ VF DIV3 ; THEN VF DIV3
		342	;
		343	BR VF DIV5 ;JUMP VF DIV5
		344	;
		345	VF DIV1: CMP AL,01H ;IF AL = 1(N)
		346	BZ VF DIV3 ; THEN VF DIV3
		347	;
		348	VF DIV2: CALL VFIS ;ZERO SET
		349	BR VF ADD10 ;JUMP VF ADD10
		350	;
		351	VF DIV3: CALL VFIE ;INDEFINITE SET
		352	BR VF ADD10 ;JUMP VF ADD10
		353	;
		354	VF DIV4: CMP AL,0H ;IF AL = 0(N)
		355	BZ VF DIV6 ; THEN VF DIV6
		356	;
		357	CMP AL,01H ;IF AL NOT = 1(N)
		358	BNZ VF DIV2 ; THEN VF DIV2
		359	;
		360	ADD FLAG,0CH
		361	;
		362	VF DIV5: CALL VFDE ;INFINITY SET
		363	BR VF ADD10 ;JUMP VF ADD10
		364	;
		365	VF DIV6: CALL VFNOMA ;NORMALIZATION MNT.
		366	CALL VFCPSD ;COMPARISON MNT.
		367	BC VF DIV7 ;IF CY = 1 THEN VF DIV7
		368	;
		369	BNZ VF DIV8 ;IF Z = 0 THEN VF DIV8
		370	;
		371	CALL VFLR ;CLEAR WORK2_FIELD
		372	MOV WORK2[8],010H
		373	MOV STAT,03H
		374	BR VF DIV8 ;JUMP VF DIV8
		375	;

```

LOC  OBJ          LINE      SOURCE
C200  8D36C800      R        376  VFDIV7: LDEA    IX,MNTS[B]      ;IX ← MNTS[B] ADDR
D211  EBCC02                377          CALL    VFDIV4      ;ROTATE LEFT & INC EXP.
                378          ;
Q214  80160900      R        379  VFDIV8: MOY    DM,EXPS
Q218  29161600      R        380          SUB    EXP0,DM
Q21C  813E16JGFFJF  R        381          CMP    EXP0,03FFFFH   ;IF EXP0 > 3FFFFH
Q222  7FC9                382          BGT    VFDIV5      ; THEN VFDIV5
                383          ;
C224  813E1600+2BF  R        384          CMP    EXP0,0BFF2H   ;IF EXP0 = > 0BFF2H
C22A  7008                385          BGE    VFDIV9      ; THEN VFDIV9
                386          ;
C22C  2F10C61A0004  R        387          SETI1  FLAG,04H      ;FLAG(BIT = 4) ← "1"
Q232  EBA0                388          BR     VFDIV2      ;JUMP VFDIV2
                389          ;
Q234  503E190003    R        390  VFDIV9: CMP    STAB,03H   ;IF STAB = 3(H)
Q239  7403                391          NZ    VFDIV10    ; THEN VFDIV10
                392          ;
D23B  EBE3D1                393          CALL   VFDIV8      ;DIVISION MNT.
                394          ;
D23E  81061600+7JF  R        395  VFDIV1C:ADD   EXP0,03FFFFH
D244  E92CFF                396          BR     VFMUL8      ;JUMP VFMUL8
                397          ;
                398  VFDIV  ENDP
                399          ;
                400  SEJECT

```

```

LOC 09J          LINE      SOURCE
                401      ;
                402      ;
                403      ;          FLOATING POINT COMPARISON
                404      ;
                405      ;
O247            406      VFCMP  PROC  NEAR
                407      ;
O247 60         408      MOV  R          ;SAVE REGISTERS
O248 8BEE       409      MOV  SP,SP
O24A E8BE00     410      CALL VFLDR  ;DATA LOAD & STATUS SET
O24B 720C       411      BC   VFCMP1  ;IF CY = 1 THEN VFCMP1
                412      ;
O24F 3AF2       413      CMP  DH,DL   ;IF DH = DL
O251 7411       414      BZ   VFCMP2  ; THEN VFCMP2
                415      ;
O253 300101     416      CMP  AH,0101H  ;IF AH = 1C1(H)
O256 7423       417      BZ   VFCMP4  ; THEN VFCMP4
                418      ;
O25B EB3C90     419      BR   VFCMP6  ;JUMP VFCMP6
                420      ;
O25B DF1D061A00D0  R  421      VFCMP1: SET1  FLAG,0H  ;FLAG(BIT = 0) <- "1"
O261 E9F2FD     422      BR   VFADD10 ;JUMP VFADD10
                423      ;
O264 300202     424      VFCMP2: CMP  AH,0202H  ;IF AH = 202(H)
O267 74F2       425      BZ   VFCMP1  ; THEN VFCMP1
                426      ;
O269 411600     427      MOV  AH,EXPD  ;IF EXPD < EXPS
O26C 3B040900   R  428      CMP  AH,EXPS  ; THEN VFCMP5
O270 7212       429      BC   VFCMP5
                430      ;
O272 7516       431      BNZ  VFCMP6  ;IF EXPD NOT = EXPS
                432      ;
O274 E9F900     433      VFCMP3: CALL  VF599D  ;SUBTRACTION RNT.
O277 720B       434      BC   VFCMP5  ;IF CY = 1 THEN VFCMP5
                435      ;
O279 750F       436      BNZ  VFCMP6  ;IF Z = 0 THEN VFCMP6
                437      ;
O27B DF1D061A00DE  R  438      VFCMP4: SET1  FLAG,0EH  ;FLAG(BIT = 14) <- "1"
O281 E902FD     439      BR   VFADD10 ;JUMP VFADD10
                440      ;
O284 DF1D061A00DE  R  441      VFCMP5: SET1  FLAG,0BH  ;FLAG(BIT = 8) <- "1"
                442      ;
O28A 30FE00     443      VFCMP6: CMP  DH,0H   ;IF DH = 0(H)
O290 7406       444      BZ   VFCMP7  ; THEN VFCMP7
                445      ;
O29F DF1F061A00DE  R  446      NOT1  FLAG,0BH  ;FLAG(BIT = 8) <- NOT (BIT = 2)
                447      ;
O29S E9BEFD     448      VFCMP7: BR   VFADD10 ;JUMP VFADD10
                449      ;
                450      VFCMP  ENDP
                451      ;
                452      SEJECI

```

LOC	OBJ	LINE	SOURCE
		453	;
		454	;
		455	; VFL0D (PUBLIC)
		456	;
		457	; DATA LOAD & SEPARATION
		458	;
		459	;
0293		460	VFL0D PROC NEAR
		461	;
0293 32E4		462	XOR AH, AH
029A C607C0		463	MOV BYTE PTR [BX], CH
0296 C6063D0000	R	464	MOV COUNT1, 0H
02A2 8F0100		465	MOV IX, 01H
02A5 8908C0		466	MOV CX, 08H
		467	;
02A6 3A04		468	VFL0D1: MOV AL, [IX]
02AA 38D0		469	MOV DX, AH
02AC 0AEC		470	OR AH, AL
02AE 0400		471	ADD AL, 0H
02B0 7507		472	BNZ VFL0D2 ; IF Z = 0 THEN VFL0D2
		473	;
02B2 FE063000	R	474	INC COUNT1
02B6 180690		475	BB VFL0D3 ; JUMP VFL0D3
		476	;
02B9 27		477	VFL0D2: ADJ4A ; ADJUST NIBBLE ADD
02BA 3AC2		478	CMF AL, DL ; IF AL NOT = DL
02BC 7520		479	BNZ VFL0D4 ; THEN VFL0D4
		480	;
02BE 8801		481	VFL0D5: MOV [BX+IX], AL
02C0 46		482	INC IX
02C1 47		483	INC IX
02C2 E2E4		484	DBNZ VFL0D1 ; JUMP VFL0D1 UNTIL CX = 0(N)
02C4 8A04		485	MOV DL, AH
02C6 6804		486	MOV AH, [IX]
02C8 98C8		487	MOV CX, AH
02CA B9E1FF7F		488	AND CX, 07FFFFH
02CE 07E0		489	SHL AH, 01H ; SHIFT LEFT AH
02D0 1AC0		490	SUBC AL, AL ; AL ← AL - AL - CY
02D2 884709		491	MOV [BX+1], AL
02D4 994F09		492	MOV [BX+9], CX
02D6 81F91F7F		493	CMF CX, 07FFFFH ; IF CX < 7FFF(H)
02D8 720F		494	BC VFL0D5 ; THEN VFL0D5
		495	;
02DE 81FA8000		496	CMF DX, 080H ; IF DX NOT = 80(H)
02E2 7507		497	BNZ VFL0D4 ; THEN VFL0D4
		498	;
02E4 C647CC02		499	MOV BYTE PTR [BX+12], 02H
02E8 EB3F90		500	BB VFL0D1C ; JUMP VFL0D1C
		501	;
02EB F9		502	VFL0D4: SETI CY ; CY ← "1"
02EC C3		503	RET
		504	;
02E6 8402		505	VFL0D5: TEST DL, DL ; DL AND DL
02EF 7434		506	BZ VFL0D9 ; IF Z = 1 THEN VFL0D9
		507	;

```

LDC OBJ          LINE      SOURCE
C2F1 33F900      506          CMP     CW,0H          ;IF CW NOT = 0(H)
C2F4 751C        507          SHZ     VFLOD7        ; THEN VFLOD7
                    510          ;
C2F6 803E3000D7  R      511          CMP     COUNT1,07H    ;IF COUNT1 NOT = 7(H)
C2FB 7506        512          BAZ     VFLOD6        ; THEN VFLOD6
                    513          ;
C2FD 407FC109    514          CRP     BYTE PTR [9W+1],09H ;IF [8W+1] = < 9(H)
C301 7622        515          BNH     VFLOD9        ; THEN VFLOD9
                    516          ;
C303 9D7708      517          VFLOD6: LDEA    IX,[9W+8] ;IX <- [8W+8] ADDR
C306 EB5701      518          CALL   VFR0L4        ;ROTATE LEFT & INC EXP.
C309 0F10C6140D01  #      519          SET1   FLAG,01H     ;FLAG(BIT = 1) <- "1"
C30F EB1890      520          BR     VFLOD10       ;JUMP VFLOD10
                    521          ;
C312 8813        522          VFLOD7: MOV     IX,BW
                    523          ;
C314 764708F0    524          VFLOD8: TEST   BYTE PTR [8W+8],0F0H ;[8W+8] AND F0(H)
C31B 750F        525          PRZ     VFLOD10     ;IF Z = 0 THEN VFLOD10
                    526          ;
C31A 837F0901    527          CMP     WORD PTR [8W+9],01H ;IF [8W+9] = 1(H)
C31E 7409        528          BE     VFLOD10     ; THEN VFLOD10
                    529          ;
C320 E8CE01      530          CALL   VFR0R4        ;ROTATE RIGHT & DEC EXP.
C323 EBEF        531          BR     VFLOD8        ;JUMP VFLOD8
                    532          ;
C325 C6470C01    533          VFLOD9: MOV     BYTE PIR [8W+12],01H
                    534          ;
C329 FB          535          VFLOD10: CLR1   CY
C32A C3          536          RET
                    537          ;
                    538          VFLOD  ENDP
                    539          ;
                    540          SEJECT

```

LOC	OBJ	LINE	SOURCE
		541	;
		542	;
		543	; VFLDR (PUBLIC)
		544	;
		545	; DATA LOAD & STATUS SET
		546	;
		547	;
0328		548	VFLDR PROC NEAR
		549	;
0329	C7061A000000	R 550	MOV FLAG,0H
0331	C606CC0000	R 551	MOV STAS,0H
0336	C608190000	R 552	MOV STAD,0H
0338	8E761?	R 553	MOV IX,[BP+18]
033E	8D1EG000	R 554	LDEA 9M,MNTS ;BM <- MNTS ADDR
0342	EB53FF	R 555	CALL VFLOD ;DATA LOAD & SEPARATION
0345	7219	R 556	BC VFLOD1 ;IF CY = 1 THEN VFLOD1
		557	;
0347	8B7614	R 558	MOV IX,[BP+20]
034A	8D1EG000	R 559	LDEA 9M,MNTD ;BM <- MNTD ADDR
034E	EB47FF	R 560	CALL VFLOD ;DATA LOAD & SEPARATION
0351	720F	R 561	BC VFLOD1 ;IF CY = 1 THEN VFLOD1
		562	;
0353	A00C00	R 563	MOV AL,STAS
0356	8A261900	R 564	MOV AH,STAD
035A	8A16C000	R 565	MOV DL,SGNS
035E	8A361500	R 566	MOV DH,SGND
		567	;
0362	C3	R 568	VFLDR1: RET
		569	;
		570	VFLDR ENDP
		571	;
		572	\$EJECT

```

LOC  OBJ          LINE  SOURCE
;
;
;
573      ;
574      ;*****
575      ;      VFASD (PUBLIC)
576      ;*****
577      ;      MNTD_FIELD <- MNTD_FIELD + MNTS_FIELD
578      ;*****
579      ;
C363      580      VFASD  PROC   NEAR
;
C363  8036C000      R  581      ;
C367  803EC000      R  582      LDEA  IX,MNTS      ;IX <- MNTS ADDR
C369  8112          583      LDEA  IY,MNTD      ;IY <- MNTD ADDR
C36A  0F2C          584      MOV   CL,012H
C36B  C3          585      ADD4S      ;ADD NIBBLE STRING
C36C  C3          586      RET
;
;
587      ;
588      VFASD  ENDP
;
;
589      ;
590      ;*****
591      ;      VFSBSD (PUBLIC)
592      ;*****
593      ;      MNTD_FIELD <- MNTD_FIELD - MNTS_FIELD
594      ;*****
595      ;
C370      596      VFSBSD  PROC   NEAR
;
C370  8036C000      R  597      ;
C374  803EC000      R  598      LDEA  IX,MNTS      ;IX <- MNTS ADDR
C378  8112          599      LDEA  IY,MNTD      ;IY <- MNTD ADDR
C37A  0F22          600      MOV   CL,012H
C37C  C3          601      SUB4S      ;SUBTRACT NIBBLE STRING
C37D  C3          602      RET
;
;
603      ;
604      VFSBSD  ENDP
;
;
605      ;
606      ;*****
607      ;      VFCPSD (PUBLIC)
608      ;*****
609      ;      COMPARE MNTD_FIELD <-> MNTS_FIELD
610      ;*****
611      ;
C37b      612      VFCPSD  PROC   NEAR
;
C37b  8036C000      R  613      ;
C381  803EC000      R  614      LDEA  IX,MNTS      ;IX <- MNTS ADDR
C383  8112          615      LDEA  IY,MNTD      ;IY <- MNTD ADDR
C387  0F26          616      MOV   CL,012H
C389  C3          617      CMP4S      ;COMPARE LEFT NIBBLE STRING
C38A  C3          618      RET
;
;
619      ;
620      VFCPSD  ENDP
;
;
621      ;
622      REJECT

```

LOC	OBJ	LINE	SOURCE
		623	;
		624	;
		625	VFFLSD (PUBLIC)
		626	;
		627	MNTD_FIELD ← MNTD_FIELD + MNTS_FIELD
		628	;
		629	;
038A		630	VFFLSD PROC NEAR
		631	;
038A	EB43D1	632	CALL VFCLR ;CLEAR WORK2_FIELD
0390	9E00C0	633	MOV IX,0H
0390	C606300009	634	MOV COUNT1,09H
		635	;
0395	8A44C000	636	VFFLSD1:MOV AH,MNTS(IX)
0399	2F2AC4	637	ROR4 AH ;ROTATE RIGHT NIBBLE AH
039C	250FDF	638	AND AH,0F0FH
039F	89DE	639	MOV BX,AH
05A1	C606310002	640	MOV COUNT2,02H
		641	;
05A6	80F900	642	VFFLSD2:CMP BL,0H ;IF BL = 0(H)
05A9	7440	643	BE VFFLSD4 ; THEN VFFLSD4
		644	;
03AB	C6061C0000	645	MOV WORK1,0H
0390	8F00C0	646	MOV IX,0H
0393	890900	647	MOV CX,09H
		648	;
0396	8A450000	649	VFFLSD5:MOV AH,MNTB(IX)
039A	0F2AC4	650	ROR4 AH ;ROTATE RIGHT NIBBLE AH
039D	250FDF	651	AND AH,0F0FH
03C0	8A04	652	MOV BL,AH
03C2	F6EB	653	MUL BL
03C4	D40A	654	CVTBD ;CONVERT BINARY TO DECIMAL
03C6	92	655	XCH AH,BH ;AH ↔ BH
03C7	F6EB	656	MUL BL
03C9	02C6	657	ADD AL,BH
03C2	D40A	658	CVTBD ;CONVERT BINARY TO DECIMAL
03C0	8AF4	659	MOV BH,AH
03CF	8AEC	660	MOV AH,AL
03D1	8AC2	661	MOV AL,DL
03D3	2F28C4	662	ROR4 AH ;ROTATE LEFT NIBBLE AH
03D6	8A851C00	663	MOV AL,WORK1(IX)
03BA	32C4	664	ADD AL,AH
03DC	27	665	ADJ4A ;ADJUST NIBBLE ADD
03D0	9006C0	666	ADD BH,0H ;BH ← BH + 0(H) + CY
03E0	8A851C00	667	MOV WORK1(IX),AL
03E4	47	668	INC IX
03E5	8A851C00	669	MOV WORK1(IX),BH
03E9	E2CB	670	DBNZ VFFLSD3 ;JUMP VFFLSD3 UNTIL CX = 0(H)
		671	;
03E3	E814C1	672	VFFLSD4:CALL VFRQLW ;ROTATE LEFT WORK2_FIELD
03EE	90F800	673	CMP BL,0H ;IF BL = 0(H)
03F1	740E	674	BE VFFLSD5 ; THEN VFFLSD5
		675	;
03F3	54	676	PUSH IX ;SAVE IX
03F4	30361C00	677	LDIA IX,WORK1 ;IX ← WORK1 ADDR

LOC	OBJ		LINE	SOURCE	
03F8	3D3E7600	R	678	LDEA	IX,WORK2
03FC	3114		679	MOV	CL,D14H
03FE	0F2C		680	ADDAS	
0400	5E		681	POP	IX
			682		;RESTORE IX
			683		;
0401	BAD1		683	VFPLSD5:MOV	BL,BH
0403	FE0E3100	R	684	DEC	COUNT2
0407	7598		685	BNZ	VFPLSD2
			686		;IF Z = 0 THEN VFPLSD2
			687		;
C409	46		687	INC	IX
040A	FE0E3000	R	688	DEC	COUNT1
040E	74D2		689	BZ	VFPLSD6
			690		;IF Z = 1 THEN VFPLSD6
			691		;
0410	EBB3		691	BR	VFPLSD1
			692		;JUMP VFPLSD1
			693		;
0412	F6D62F000F	R	693	VFPLSD6:TEST	BYTE PTR WORK2[9],DFH
0417	74D7		694	BZ	VFPLSD7
			695		;WORK2[9] AND F(H)
			696		;IF Z = 1 THEN VFPLSD7
			697		;
0419	E8E600		696	CALL	VFROLW
041C	FFD61600	R	697	INC	EXPD
			698		;ROTATE LEFT WORK2_FIELD
			699		;
0420	C3		699	VFPLSD7:RET	
			700		;
			701	VFPLSD ENDP	
			702		;
			703	SEJECT	

```

LOC  OBJ          LINE  SOURCE
                                704      ;
                                705      ;*****
                                706      ;      VFDVSD (PUBLIC)
                                707      ;*****
                                708      ;      MNTD_FIELD <- MNTD_FIELD / MNTS_FIELD
                                709      ;*****
                                710      ;
0421          711      VFDVSD PROC    NEAR
                                712      ;
0421  EBAE00          713      CALL    VFCLR                ;CLEAR WORK2_FIELD
0424  B036C800        R          714      LDEA    IX,MNTS[B]          ;IX <- MNTS[B] ADDR
0428  E88500          715      CALL    VFR0L4            ;ROTATE LEFT & INC EXP.
0429  FFOEC900        R          716      DEC     EXP5
042F  D0361500        R          717      LDEA    IX,MNTD[B]          ;IX <- MNTD[B] ADDR
0433  EBAAC0          718      CALL    VFR0L4            ;ROTATE LEFT & INC EXP.
0436  FFOE1600        R          719      DEC     EXPB
043A  D01E2E00        R          720      LDEA    BM,WORK2[B]        ;BM <- WORK2[B] ADDR
043E  C806310011      R          721      MOV     COUNT2,011H
                                722      ;
0443  3302           723      VFDVSD1: XOR     DM,DM
                                724      ;
0445  E82E00          725      VFDVSD2: CALL    VFSB5D          ;SUBTRACTION MNT.
0448  720B           726      BC      VFDVSD4            ;IF CY = 1 THEN VFDVSD4
                                727      ;
044A  7404           728      BZ      VFDVSD3            ;IF Z = 1 THEN VFDVSD3
                                729      ;
044C  FE06           730      INC     DM
044E  EBF3           731      BR      VFDVSD2            ;JUMP VFDVSD2
                                732      ;
0450  FE06           733      VFDVSD3: INC     DM
0452  EB3E00          734      BR      VFDVSD8            ;JUMP VFDVSD8
                                735      ;
0455  EB0BFF          736      VFDVSD4: CALL    VFAD5D          ;ADDITION MNT.
0458  D036C000        R          737      LDEA    IX,MNTD            ;IX <- MNTD ADDR
045C  EB9200          738      CALL    VFR0R4            ;ROTATE RIGHT & DEC EXP.
045F  FFD61600        R          739      INC     EXPB
                                740      ;
0463  E90AFF          741      VFDVSD5: CALL    VFSB5D          ;SUBTRACTION MNT.
0466  720B           742      BC      VFDVSD7            ;IF CY = 1 THEN VFDVSD7
                                743      ;
0468  7404           744      BZ      VFDVSD6            ;IF Z = 1 THEN VFDVSD6
                                745      ;
046A  FE02           746      INC     DL
046C  EBF5           747      BR      VFDVSD5            ;JUMP VFDVSD5
                                748      ;
046E  FE02           749      VFDVSD6: INC     DL
0470  EB2090          750      BR      VFDVSD8            ;JUMP VFDVSD8
                                751      ;
0473  EB0BFE          752      VFDVSD7: CALL    VFAD5D          ;ADDITION MNT.
0476  D036C000        R          753      LDEA    IX,MNTD            ;IX <- MNTD ADDR
047A  E87400          754      CALL    VFR0R4            ;ROTATE RIGHT & DEC EXP.
047D  FFD61600        R          755      INC     EXPB
0481  3BC2           756      MOV     AM,DM
0483  OF28C4          757      ROL4    AM                ;ROTATE LEFT MIDDLE AM
0486  3827           758      MOV     [BM],AM

```

LOC	OBJ	LINE	SOURCE
C498	4B	759	DEC Bw
C499	FEDE3100	760	DEC COUNT2
C4A0	75B4	761	BWZ VFBVSD1
		762	;
		763	BR VFBVSD9
		764	;
		765	VFBVSDP:MOV AM,BW
C492	9BC2	766	ROL4 AM
C494	DF28C4	767	MOV [BW],AM
C497	9B27	768	;
		769	VFBVSD0:RET
		770	;
		771	VFBVSD [WDP
		772	;
		773	SEJECT

;IF Z = 0 THEN VFBVSD1

;JUMP VFBVSD9

;ROTATE LEFT NIBBLE AM

001

```

LOC  OBJ                LINE  SOURCE
                                774      ;
                                775      ;.....
                                776      ;      VFNOMA (PUBLIC)
                                777      ;.....
                                778      ;      NORMALIZATION MNTD_FIELD & MNTS_FIELD
                                779      ;.....
                                780      ;
049A                781      VFNOMA PROC      NEAR
                                782      ;
049A 80360000      R      783      LDEA      IX,MNTS      ;IX ← MNTS ADDR
                                784      ;
049E F6060500F0    R      785      VFNOMA1:TEST  MNTS(8),OFOW      ;MNTS(8) AND FO(4)
04A3 7505                786      SNZ      VFNOMA2      ;IF Z = 0 THEN VFNOMA2
                                787      ;
04A5 E84900                788      CALL      VFROR4      ;ROTATE RIGHT & DEC EXP.
04A8 EBF4                789      BR      VFNOMA1      ;JUMP VFNOMA1
                                790      ;
04AA 3D5E0000      R      791      VFNOMA2:LDEA  IX,MNTD      ;IX ← MNTD ADDR
                                792      ;
04AE F6061500F0    R      793      VFNOMA3:TEST  MNTD(8),OFOW      ;MNTD(8) AND FO(4)
04F3 7505                794      BNZ      VFNOMA4      ;IF Z = 0 THEN VFNOMA4
                                795      ;
04B5 E839C0                796      CALL      VFROR4      ;ROTATE RIGHT & DEC EXP.
04B6 EBF4                797      BR      VFNOMA3      ;JUMP VFNOMA3
                                798      ;
049A C3                799      VFNOMA4:RET
                                800      ;
                                801      VFNOMA ENDP
                                802      ;
                                803      SEJECT

```

```

LOC  OBJ                LINE  SOURCE
                                304      ;
                                305      ;*****
                                306      ;      VFXCG (PUBLIC)
                                307      ;*****
                                308      ;      DESTINATION_FIELD <-> SOURCE_FIELD
                                309      ;*****
                                310      ;
C439                311      VFXCG  PROC    NEAR
                                312      ;
0488 8036C000        R      313          LDEA  IX,MNTS      ;IX <- MNTS ADDR
048F 8D3EC000        R      314          LDEA  IY,MNTD      ;IY <- MNTD ADDR
                                315      ;
C4C3 8404                316      VFXCG1: MOV   AL,[IX]
04C5 8603                317          RCM   AL,[IY]      ;AL <-> [IY]
04C7 8804                318          MOV   [IX],AL
04C9 46                319          INC   IX
04CA 47                320          INC   IY
04CB E2F6                321          DBNZ  VFXCG1      ;JUMP VFXCG1 UNTIL CM = 0CM)
04CD C3                322          RET
                                323      ;
                                324      VFXCG  ENDP
                                325      ;
                                326      ;*****
                                327      ;      VFRVSD (PUBLIC)
                                328      ;*****
                                329      ;      IY BLOCK <- IX BLOCK (BYTE)
                                330      ;*****
04CE                331      ;
                                332      VFRVSD PROC    NEAR
                                333      ;
04CE FE                334          CLR1  DIR      ;DIR <- "0"
04CF F3A4                335          REP   MOVKB      ;IY BLOCK <- IX BLOCK (BYTE)
04D1 C3                336          RET
                                337      ;
                                338      VFRVSD ENDP
                                339      ;
                                340      $EJECT

```

```

LOC OBJ                LINE    SOURCE
                        841      ;
                        842      ;.....
                        843      ;     VFCLR (PUBLIC)
                        844      ;.....
                        845      ;     CLEAR WORK2_FIELD
                        846      ;.....
                        847      ;
04B2                   848     VFCLR PROC    NEAR
                        849      ;
04D2 3D362600          850      LDEA    IX,WORK2           ;IX ← WORK2 ADDR
04D6 39D400            851      MOV     CW,0AH
                        852      ;
04D9 06D400           853     VFCLR1: MOV   BYTE PTR [IX],CX
04DC 46                854      INC     IX
04DD E2FA             855      DBNZ   VFCLR1           ;JUMP VFCLR1 UNTIL CW = 0(H)
04DF C3                856      RET
                        857      ;
                        858     VFCLR  ENDP
                        859      ;
                        860      ;.....
                        861      ;     VFRQL4 (PUBLIC)
                        862      ;.....
                        863      ;     ROTATE LEFT & INC EXPONENTIAL
                        864      ;.....
                        865      ;
04E0                   866     VFRQL4 PROC  NEAR
                        867      ;
04E3 32CC             868      XOR     AL,AL
04E2 99D9DD          869      MOV     CW,D9H
04E5 56              870      PUSH   IX           ;SAVE IX
                        871      ;
04E6 DF2404          872     VFRQL1: RORL  BYTE PTR [IX]           ;ROTATE RIGHT NIBBLE [IX]
04E9 4E              873      DEC     IX
04EA E2FA           874      DBNZ   VFRQL1           ;JUMP VFRQL1 UNTIL CW = 0(H)
04EC 5E              875      POP     IX           ;RESTORE IX
04ED FF4401          876      INC     WORD PTR [IX+1]
04F0 C3              877      RET
                        878      ;
                        879     VFRQL4 ENDP
                        880      ;
                        881     SEJECT

```

```

LOC  OBJ          LINE      SOURCE
                                882      ;
                                883      ;.....
                                884      ;      VFPOR4 (PUBLIC)
                                885      ;.....
                                886      ;      ROTATE RIGHT & DEC EXPONENTIAL
                                887      ;.....
                                888      ;
                                889      VFPOR4 PROC      NEAR
                                890      ;
                                891      XOP      AL,AL
                                892      MOV      CM,D9H
                                893      PUSH     IX          ;SAVE IX
                                894      ;
                                895      VFPOR4: ROL4   BYTE PTR [IX]      ;ROTATE LEFT NIBBLE [IX]
                                896      INC      IX
                                897      DBNZ    VFPOR1      ;JUMP VFPOR1 UNTIL CM = C(4)
                                898      POP      IX          ;RESTORE IX
                                899      DEC     WORD PTR [IX+9]
                                900      RET
                                901      ;
                                902      VFPOR4 ENDP
                                903      ;
                                904      ;.....
                                905      ;      VFPOLW (PUBLIC)
                                906      ;.....
                                907      ;      ROTATE LEFT WORK2_FIELD
                                908      ;.....
                                909      ;
                                910      VFPOLW PROC      NEAR
                                911      ;
                                912      XOP      AL,AL
                                913      LDBA   IT,WORK2[9]      ;IT ← WORK2[9] ADDR
                                914      MOV      CM,DAH
                                915      ;
                                916      VFPOLW: ROR4   BYTE PTR [IT]      ;ROTATE RIGHT NIBBLE [IT]
                                917      DEC     IT
                                918      DBNZ    VFPOLW1      ;JUMP VFPOLW1 UNTIL CM = D(4)
                                919      RET
                                920      ;
                                921      VFPOLW ENDP
                                922      ;
                                923      BEJECT

```

```

LOC  OBJ          LINE      SOURCE
                                924      ;
                                925      ;
                                926      ;
                                927      ;
                                928      ;
                                929      ;
                                930      ;
0512          931      VFZ5   PROC   NEAR
                                932      ;
0512 3202          933      LOP    DL,DL
0514 8B7614        934      MOV    IX,CBP+20J
0517 3908C0        935      MOV    CW,08H
                                936      ;
051A 8B14          937      VFZ51: MOV   [IX],DL
051C 46            938      INC    IX
051D E2FB          939      DBNZ  VFZ51          ;JUMP VFZ51 UNTIL CW = 0(H)
051F 803E180000    940      CMP    SGND,0H      ;IF SGND = 0(H)
052A 7405          941      BZ    VFZ52          ; THEN VFZ52
                                942      ;
0526 0F1D44F60F    943      SETI  WORD PTR [IX-3],0FH ;[IX-3](BIT = 15) <- "1"
                                944      ;
052B C3            945      VFZ52: RET
                                946      ;
                                947      VFZ5   ENDP
                                948      ;
                                949      ;
                                950      ;
                                951      ;
                                952      ;
                                953      ;
                                954      ;
052C          955      VFOE   PROC   NEAR
                                956      ;
052C 8A0080        957      MOV    DW,08000H
052F 3B7614        958      MOV    IX,CBP+20J
0532 C74408F17F    959      MOV    [IX+5],07FFFH
0537 B90700        960      MOV    CW,07H
                                961      ;
053A 8514          962      VFOE1: MOV   [IX],DL
053C 46            963      INC    IX
053D E2FB          964      DBNZ  VFOE1          ;JUMP VFOE1 UNTIL CW = 0(H)
053F 8B54          965      MOV    [IX],0H
0541 803E180000    966      CMP    SGND,0H      ;IF SGND = 0(H)
0544 7405          967      BZ    VFOE2          ; THEN VFOE2
                                968      ;
0548 3F1D44010F    969      SETI  WORD PTR [IX+1],0FH ;[IX+1](BIT = 15) <- "1"
                                970      ;
054D 0F1F061A0003  971      VFOE2: NOTI  FLAG,03H   ;FLAG(BIT = 3) <- NOT (BIT = 3)
0553 C3            972      RET
                                973      ;
                                974      VFOE   ENDP
                                975      ;
                                976      ;
                                976      BEJEC1

```



```

LOC  OBJ                LINE  SOURCE
                                977  ;
                                978  ;*****
                                979  ;   VFIE  (PUBLIC)
                                980  ;*****
                                981  ;   DESTINATION_DATA <- INDEFINITE
                                982  ;*****
                                983  ;
0554                984  VFIE  PROC  NEAR
                                985  ;
C334  3A00C0          986                MOV  DW,0C000H
0557  5B7614          987                MOV  IX,[BP+20]
055A  C744C8FFFF     988                MOV  [[X+8],0FFFFH
055F  890700          989                MOV  CW,07H
                                990  ;
C562  8814           991  VFIE1:  MOV  [IX],DL
C564  46             992                INC  IX
C565  62F9           993                DBNZ VFIE1 ;JUMP VFIE1 UNTIL CW = 0(M)
C567  8834           994                MOV  [IX],DH
C569  0F19061A000C   R  995                SETI FLAG,DH ;FLAG(BIT = 0) <- "1"
C56F  C3             996                RET
                                997  ;
                                998  VFIE  ENDP
                                999  ;
1200                $EJECT

```

```

LOC  OBJ          LINE      SOURCE
                                1001      ;
                                1002      ;
                                1003      ;
                                1004      ;
                                1005      ;
                                1006      ;
                                1007      ;
                                1008      ;
                                1009      ;
0570          1010      VFSTR  PROC   NEAR
                                1011      ;
0570 3050          1012      MOV    AL,050H
0572 30360000      R      1013      LDEA  IX,MNTD          ;IX ← MNTD ADDR
0576 390900          1014      MOV    CM,09H
                                1015      ;
0579 0204          1016      VFSTR1: ADD  AL,[IX]
057B 27            1017      ADJAA          ;ADJUST NIBBLE ADD
057C 8804          1018      MOV    [IX],AL
057E 731B          1019      BNC   VFSTR3          ;IF CY = 0 THEN VFSTR3
                                1020      ;
0580 46            1021      INC   IA
0581 3001          1022      MOV    AL,01H
0583 2274          1023      DBNZ  VFSTR1          ;JUMP VFSTR1 UNTIL CM = 0(H)
0585 80361500      R      1024      LDEA  IA,MNTD[B]      ;IX ← MNTD[B] ADDR
0589 E834FF          1025      CALL  VPROL4          ;ROTATE LEFT 5 INC EXP.
059C 300410          1026      ADD   BYTE PTR [IX],010H
                                1027      ;
058F 313E1600FE7F  R      1028      VFSTR2: CMP  EXPD,07FFEH          ;IF EXPD = 7FFF(H)
0595 7604          1029      BNP   VFSTR3          ; THEN VFSTR3
                                1030      ;
0597 E392FF          1031      CALL  WFOE          ;INFINITY SET
059A C3            1032      RET
                                1033      ;
059d 303E180000      R      1034      VFSTR3: CMP  SGND,0H          ;IF SGND = 0(H)
05A0 7406          1035      BZ    VFSTR4          ; THEN VFSTR4
                                1036      ;
05A2 0F1D061600CF  R      1037      SETI  EXPD,0FH          ;EXPD(BIT = 15) ← "1"
                                1038      ;
05A6 8036CF00      R      1039      VFSTR4: LDEA IX,MNTD[E1]          ;IX ← MNTD[E1] ADDR
05AC 487E14          1040      MOV    IY,[9P+20]
05AF 390A00          1041      MOV    CM,0AH
05B2 E819FF          1042      CALL  VPROVD          ;IY STRING ← IX STRING
05B5 C3            1043      RET
                                1044      ;
                                1045      VFSTR  ENDP
                                1046      ;
-----          1047      CODE  ENDS
                                1048      END

```

*** XREF SYMBOL LISTING ***

NAME	TYPE	VALUE	ATTRIBUTES	XREF
??SEG	SEGMENT		SIZE=000CH PARA PUBLIC	
CODE	SEGMENT		SIZE=0536H PARA PUBLIC 'CODE'	30 67# 1047#
COUNT1	BYTE	0030	DATA	56# 464 474 511 634 698
COUNT2	BYTE	0031	DATA	57# 64C 684 721 76C
DATA	SEGMENT		SIZE=0032H PARA PUBLIC 'DATA'	50 36# 59#
ERPD	WORD	0016	DATA	42# 133 141 16# 175 198 27c 277 27# 285 28# 293 29# 380 381 384 395 427 697 71# 73# 755 102#
				1037
EXPS	WORD	0009	DATA	43# 134 143 275 37# 42# 71#
FLAG	WORD	001A	DATA	52# 13C 281 29# 360 387 421 43# 441 44# 51# 55C 971 995
MNTD	BYTE	000#	DATA	47# 123 160 1# 164 16# 193 195 2# 30# 30# 55# 583 59# 615 64# 717 737 753 791 793 914 1013 1024
				103#
MNTS	BYTE	0000	DATA	42# 122 147 37# 554 582 59# 614 63# 714 783 785 813
SGND	BYTE	0018	DATA	49# 10# 117 19C 241 333 56# 94C 96# 1034
SGNS	BYTE	000#	DATA	44# 221 222 565
STAD	BYTE	0C19	DATA	50# 373 39C 552 564
STAS	BYTE	000C	DATA	45# 551 5#
VFADD	NEAR	0000	CODE	PUBLIC 27# 77# 206#
VFADD1	NEAR	000#	CODE	83# 224
VFADD10	NEAR	0056	CODE	9# 10# 120 12# 252 263 272 34# 352 363 422 43# 44#
VFADD11	NEAR	0050	CODE	112 133#
VFADD12	NEAR	0073	CODE	135 143#
VFADD13	NEAR	0080	CODE	149# 152
VFADD14	NEAR	0087	CODE	137 154#
VFADD15	NEAR	009A	CODE	158 164#
VFADD16	NEAR	009E	CODE	166# 173
VFADD17	NEAR	00B1	CODE	167 175#
VFADD18	NEAR	00B#	CODE	155 18C#
VFADD19	NEAR	00C5	CODE	181 187#
VFADD2	NEAR	000#	CODE	85#
VFADD20	NEAR	00#	CODE	183 193#
VFADD21	NEAR	0009	CODE	195# 202
VFADD22	NEAR	00EC	CODE	19# 19# 204#
VFADD3	NEAR	001A	CODE	83 97#
VFADD4	NEAR	0020	CODE	89 100#
VFADD5	NEAR	002#	CODE	104 10# 185
VFADD6	NEAR	0033	CODE	86 111#
VFADD7	NEAR	003F	CODE	92 95 119# 17#
VFADD8	NEAR	0045	CODE	101 122#
VFADD9	NEAR	0053	CODE	115 127# 145 170 17# 204 31#
VFADD#	NEAR	03#	CODE	157 187 58# 58# 73# 752
VFLR	NEAR	0402	CODE	371 632 113 84# 85#
VFLR1	NEAR	0409	CODE	853# 855
VFCPP	NEAR	0247	CODE	PUBLIC 28# 406# 450#
VFCPP1	NEAR	0259	CODE	411 421# 425
VFCPP2	NEAR	0264	CODE	414 424#
VFCPP3	NEAR	0274	CODE	433#
VFCPP4	NEAR	027B	CODE	417 43#
VFCPP5	NEAR	02#	CODE	42# 434 441#
VFCPP6	NEAR	028A	CODE	41# 451 45# 443#
VFCPP7	NEAR	0295	CODE	444 44#
VFCPS0	NEAR	037#	CODE	36# 612# 62#
VFR1V	NEAR	01#	CODE	PUBLIC 2# 32# 39#

NAME	TYPE	VALUE	ATTRIBUTES	XREF
VFDLV1	NEAR	0100	CODE	338 343#
VFDLV10	NEAR	023E	CODE	391 395#
VFDLV2	NEAR	0104	CODE	348# 350 385
VFDLV3	NEAR	010A	CODE	331 341 346 351#
VFDLV4	NEAR	01E0	CODE	335 354#
VFDLV5	NEAR	01E0	CODE	343 362# 382
VFDLV6	NEAR	01F3	CODE	355 365#
VFDLV7	NEAR	0200	CODE	367 376#
VFDLV8	NEAR	0214	CODE	369 374 379#
VFDLV9	NEAR	0234	CODE	385 390#
VFDVSD	NEAR	0421	CODE	393 711# 771#
VFDVSD1	NEAR	0443	CODE	723# 761
VFDVSD2	NEAR	0445	CODE	725# 731
VFDVSD3	NEAR	0450	CODE	728 735#
VFDVSD4	NEAR	0455	CODE	726 736#
VFDVSD5	NEAR	0463	CODE	741# 747
VFDVSD6	NEAR	046E	CODE	744 749#
VFDVSD7	NEAR	0473	CODE	742 752#
VFDVSD8	NEAR	0492	CODE	734 750 765#
VFDVSD9	NEAR	0499	CODE	765 769#
VFIE	NEAR	0354	CODE	97 251 351 924# 998#
VFIE1	NEAR	0562	CODE	991# 993
VFL00	NEAR	0320	CODE	81 218 238 330 410 548# 570#
VFLDR1	NEAR	0362	CODE	556 561 568#
VFL00	NEAR	0298	CODE	460# 538# 555 560
VFL001	NEAR	02AB	CODE	468# 484
VFL0010	NEAR	0329	CODE	500 520 525 528 535#
VFL002	NEAR	02W9	CODE	472 477#
VFL003	NEAR	028E	CODE	475 481#
VFL004	NEAR	02E0	CODE	479 497 502#
VFL005	NEAR	02E0	CODE	494 505#
VFLC06	NEAR	0303	CODE	312 317#
VFLC07	NEAR	0312	CODE	509 522#
VFLC08	NEAR	0314	CODE	524# 531
VFLC09	NEAR	0325	CODE	506 515 533#
VFMLS0	NEAR	036A	CODE	284 630# 701#
VFMLS01	NEAR	0393	CODE	636# 691
VFMLS02	NEAR	03A6	CODE	642# 685
VFMLS03	NEAR	03B6	CODE	649# 670
VFMLS04	NEAR	03E8	CODE	643 672#
VFMLS05	NEAR	0401	CODE	674 683#
VFMLS06	NEAR	0412	CODE	689 693#
VFMLS07	NEAR	0420	CODE	694 695#
VFMUL	NEAR	0102	CODE	PUBLIC 27# 234# 318#
VFMUL1	NEAR	011C	CODE	235 251# 258
VFMUL10	NEAR	01A5	CODE	300 309#
VFMUL11	NEAR	01A9	CODE	311# 316
VFMUL12	NEAR	01dC	CODE	294 316#
VFMUL2	NEAR	0122	CODE	246 254#
VFMUL3	NEAR	0120	CODE	249 262# 269 282 304 307
VFMUL4	NEAR	0133	CODE	243 265#
VFMUL5	NEAR	0138	CODE	255 26C 271# 287
VFMUL6	NEAR	0141	CODE	266 274#
VFMUL7	NEAR	0162	CODE	279 284#
VFMUL8	NEAR	0173	CODE	289# 306

NAME	TYPE	VALUE	ATTRIBUTES	REF
VFMUL9	NEAR	019E	CODE	302 306#
VFMVSD	NEAR	04CE	CODE	125 292 B32# B3E# 1042
VFNCR4	NEAR	0494	CODE	274 365 781# 8C1#
VFNCR41	NEAR	049E	CODE	785# 789
VFNCR42	NEAR	04AA	CODE	786 791#
VFNCR43	NEAR	04AE	CODE	791# 797
VFNCR44	NEAR	04BA	CODE	794 799#
VFOE	NEAR	052C	CODE	119 271 362 955# 974# 1031
VFOE1	NEAR	053A	CODE	962# 964
VFOE2	NEAR	054D	CODE	967 971#
VFRDL1	NEAR	04E9	CODE	872# 874
VFRDL4	NEAR	04E0	CODE	150 161 312 377 518 715 718 866# 879# 1025
VFRCLM	NEAR	0502	CODE	672 696 910# 921#
VFRDLW1	NEAR	0506	CODE	916# 91E
VFROR1	NEAR	04F7	CODE	895# 897
VFROR4	NEAR	04F1	CODE	172 201 530 738 754 788 796 889# 9C2#
VFSB5D	NEAR	037D	CODE	18C 191 433 596# 604# 725 741
VFSTR	NEAR	057D	CODE	127 101G# 1045#
VFSTR1	NEAR	0579	CODE	1016# 1C23
VFSTR2	NEAR	056F	CODE	1028#
VFSTR3	NEAR	059B	CODE	1019 1C29 1034#
VFSTR4	NEAR	05A9	CODE	1035 1C59#
VFSUB	NEAR	00E7	CODE	PUBLIC 27# 214# 226#
VFSU37	NEAR	00FF	CODE	215 226#
VFXCG	NEAR	040B	CODE	14C 189 811# 824#
VFXCG1	NEAR	04C3	CODE	816# 821
VFZ5	NEAR	0512	CODE	108 262 348 931# 947#
VFZ51	NEAR	051A	CODE	937# 93E
VFZ52	NEAR	052B	CODE	941 945#
WORK1	BYTE	0C1C	DATA	54# 645 663 667 669 677
WORK2	BYTE	0C26	DATA	55# 289 372 672 693 720 85C 913

ASSEMBLY COMPLETE, 0 ERROR(S) FOUND

第4章 CP/M-80エミュレーション・プログラム

4.1 プログラム概要

このプログラムは、V20/V30ベースのCP/M-86システムにおいて、8080エミュレーション機能により、CP/M-80用に開発された応用プログラムを動作させるものです。

このプログラムは、大きく二つの部分に分割されます。一つは、応用プログラムを動作可能にするために、エミュレーションに使用するセグメントを初期設定するスタートアップ・プログラム、もう一つは、応用プログラムが動作している時に発行するCP/M-80のBDOSコールを、CP/M-86のBDOSコールを使用してエミュレーションするBDOSコール・エミュレーション・プログラムです。

スタートアップ・プログラムでは、最初に、プログラムのスタート時にパラメータで指定されたCP/M-80の応用プログラムをディスクから読出し、エミュレーションで使用するセグメントにロードします。その後、エミュレーションで使用するセグメントを、CP/M-80で応用プログラムが動作する時と同じような状態に設定し、CP/M-80のエミュレーションに必要な8080の命令をセットします。最後にBRKEM命令を使用して応用プログラムをスタートします。

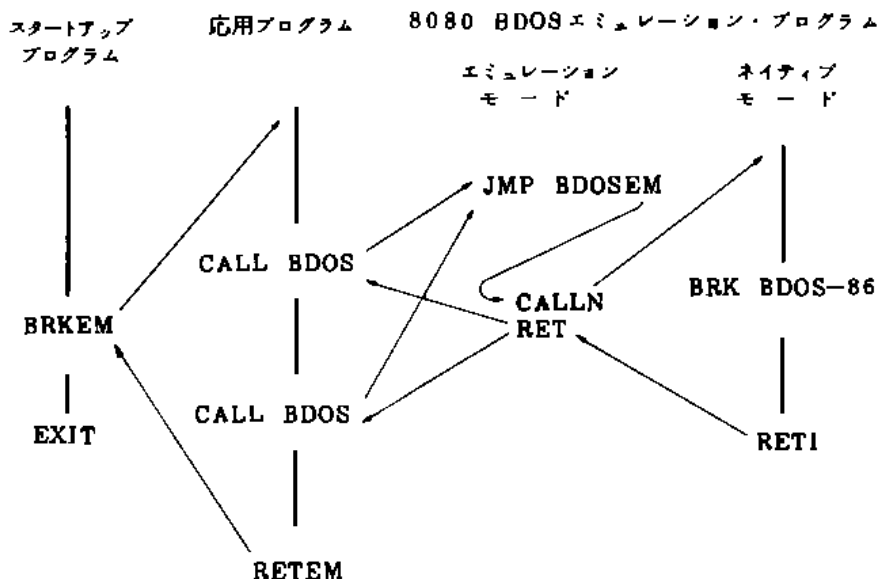
BDOSコール・エミュレーション・プログラムでは、応用プログラムがBDOSコールを発行した場合、CALLN命令により一旦ネイティブ・モードに戻り、CP/M-86のBDOSコールを発行して、CP/M-80のBDOSコールをエミュレーションします。

このプログラムは、使用する場合、コマンドに続き、実行したい応用プログラムのコマンド名および応用プログラムに対するパラメータを入力します。

例 A>RUNCPM ASM FILE

第4.1図に、このプログラムの動作時のネイティブ・モードと8080エミュレーション・モードの状態せいの様子を示します。

第 4.1 図



4.2 メモリ・マップ

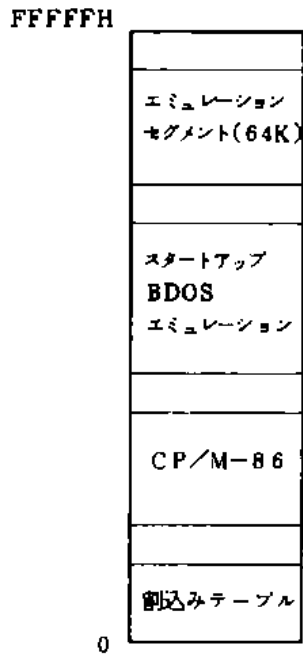
このプログラムは、CP/M-86 のサポートするメモリ・モデルのうちのスモール・モデルで動作します。スモール・モデルでは、プログラムのスタート時に、コードとデータの二つのセグメントが設定され、コードのセグメント・アドレスは、PSレジスタ、データのセグメント・アドレスはDS0、DS1レジスタにセットされます。

このプログラムでは、コードのセグメントにはネイティブ・モードで動作するスタートアップ・プログラムとBDOSコール・エミュレーション・プログラムをストアし、データのセグメントは64Kバイトの8080のアドレス空間として使用するエミュレーション・セグメントとして使用します。エミュレーション・セグメントは、64Kバイトの大きさが必要ですが、プログラムのスタート時に確保されている事が前提となっています。

なおデータのセグメントの先頭の256バイトは、ユーザ・プログラムでは使用できないベース・ページであり、デフォルトFCB、デフォルト・バッファ等の領域として使用されていますが、このプログラムでは、この領域をCP/M-80のベース・ページとしても使用しています。

第4.2図にこのプログラムがメモリにロードされた直後の状態を示します。

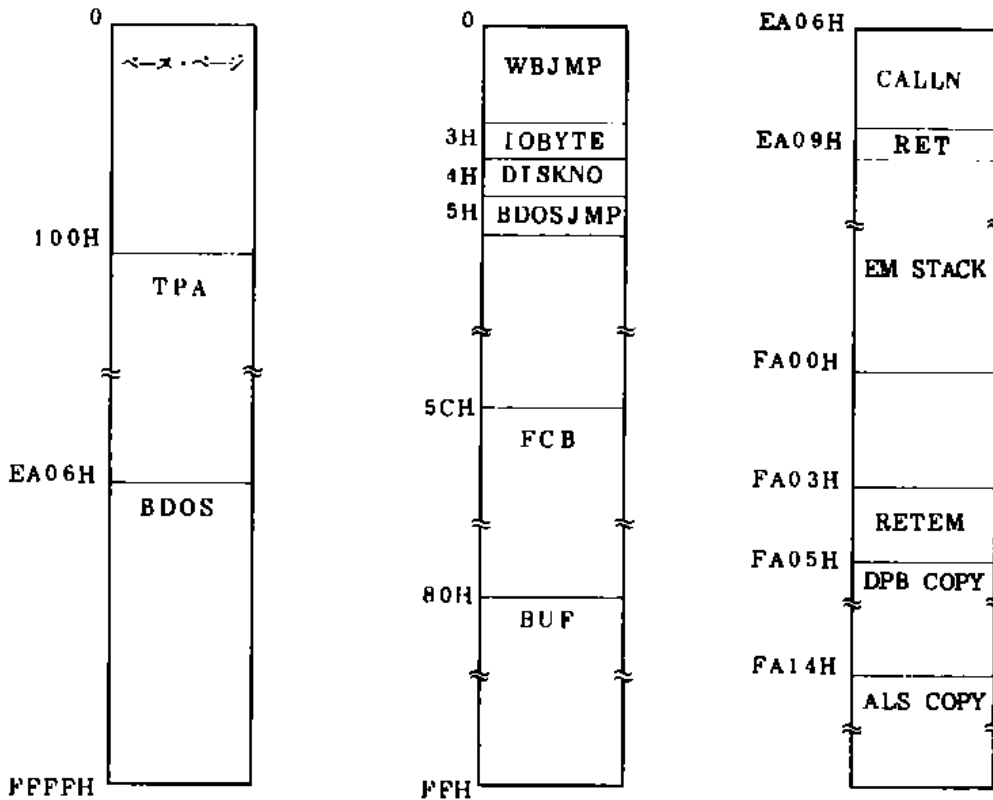
第 4.2 図



4.3 エミュレーション・セグメントのメモリ・マップ

第 4.3 図にエミュレーション・セグメントのメモリ・マップを示します。
尚、このセグメント内の命令は、すべて 8080 の命令として実行されます。

第 4.3 図



- WBJMPウォーム・ブート・エントリ 応用プログラムが、CP/M-80を再立上げするときこのアドレスをコールします。このプログラムでは、単にネイティブ・モードに戻るために、RETEM命令へのジャンプ命令がストアされます。
- IOBYTE.....I/Oバイト 応用プログラムが論理I/Oと物理I/Oの対応を調べるとき、リードします。CP/M-86のI/Oバイトをストアします。
- DISKNO.....デフォルト・ディスク N. O. 応用プログラムがデフォルト・ディスク番号を調べるとき、リードします。CP/M-86のデフォルト・ディスクN. O.をストアします。
- BDOSJMP.....応用プログラムのBDOSコール・アドレス このプログラムでは、BDOSコール・エミュレーション・プログラムをコールするため、CALLN命令へのジャンプ命令がストアされます。
- FCBデフォルトFCB (ファイル・コントロール・ブロック) 応用プログラムのスタート時、CP/M-80により設定されます。このプログラムでは、セット・アップ・ルーチンにより設定されます。
- BUFデフォルト・バッファ 応用プログラムのスタート時、CP/M-80により設定されます。このプログラムでは、セットアップ・ルーチンにより設定されます。
- TPAトランジェント・プログラム・エリア CP/M-80の応用プログラムがストアされる領域
- CALLNBDOSコール・エミュレーション・プログラムをコールするためのCALLN命令をストアする領域
- RETBDOSコール後、応用プログラムに戻るためのRET命令をストアする領域
- EM STACKエミュレーション・モード中のスタック領域
- RETEM応用プログラム実行終了後、ネイティブ・モードに戻るためのRETEM命令をストアする領域
- DPB COPYCP/M-86のディスク・パラメータ・ブロック領域のコピー・エリア
- ALS COPYCP/M-86のディスク・アロケーション・テーブル領域のコピー・エリア

4.4 スタートアップ・プログラムの処理

CP/M-86により設定された、デフォルト・バッファ・エリア、デフォルトFCBエリアをチェックし、コマンドの直後に入力された応用プログラム名が正しく入力されていない場合には、入力されたパラメータの後に"?"を付けたエラー・メッセージを出力して、プログラムを終了します。その後、デフォルトFCBエリアのファイル・タイプを"COM"に設定し、ファイルをリード・オープンします。ファイルがディスク上に存在しなかった場合、応用プログラム名の後に"?"を付

けたエラー・メッセージを出力して、プログラムを終了します。

正常にオープンできた場合には、応用プログラムのスタート・アドレスに相当するエミュレーション・セグメントのオフセット100Hからファイルをリードします。

そして、エミュレーション・セグメント内のWBJMP, BDOSJMP, CALLN, RET, RETEMのそれぞれの領域に、8080の命令や、エミュレーション・モード専用命令等をストアし、IOBYTE DISKNOの領域には、CP/M-86の値をストアします。

ディフォルト・バッファ・エリアには、CP/M-86により設定されたデータから、応用プログラム名を取り除いたデータをストアし、そのデータをもとに、ディフォルトFCBエリアを初期設定します。

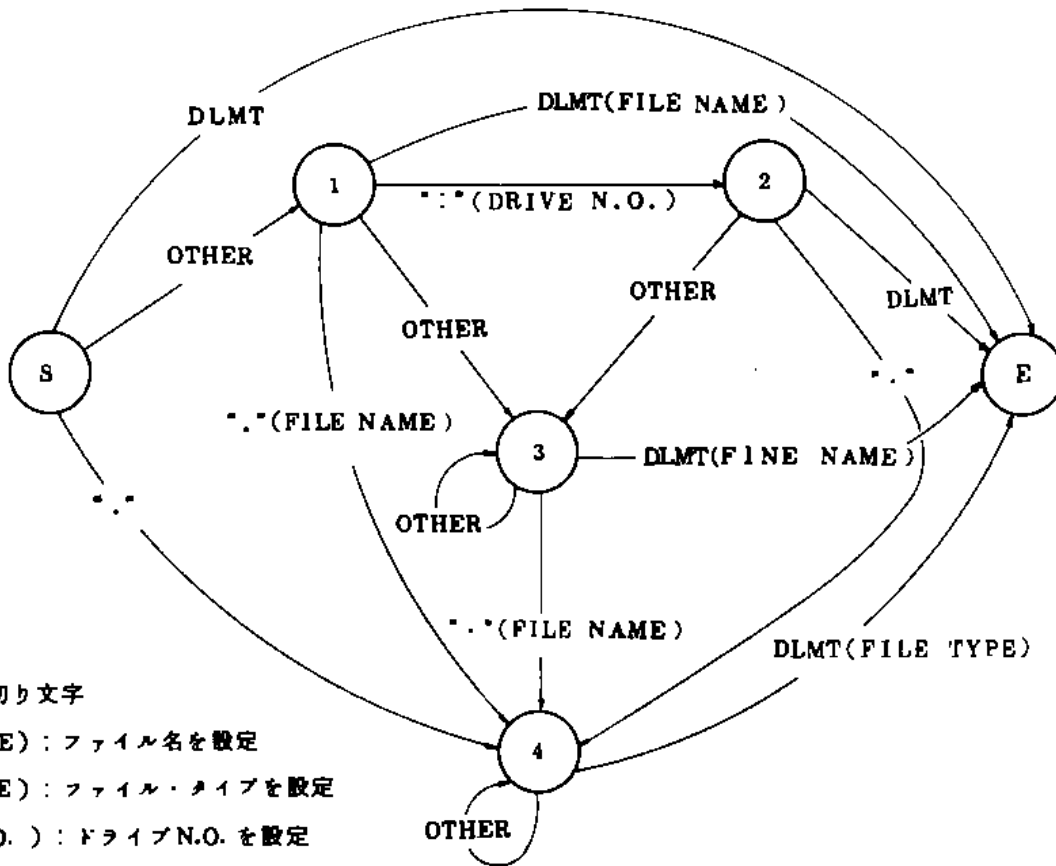
最後にBRKEM, CALLN命令のための割込みテーブルを、BRKEM命令用では、エミュレーション・セグメントの100H番地、CALLN命令用では、BDOSコール・エミュレーション・プログラムのスタート・アドレスに初期設定し、エミュレーション・モードで使用するスタック領域の先頭にRETEM命令のストアされているアドレスをストアした後、BRKEM命令により、8080のエミュレーションをスタートします。

4.5 デイフォルトFCBの作成

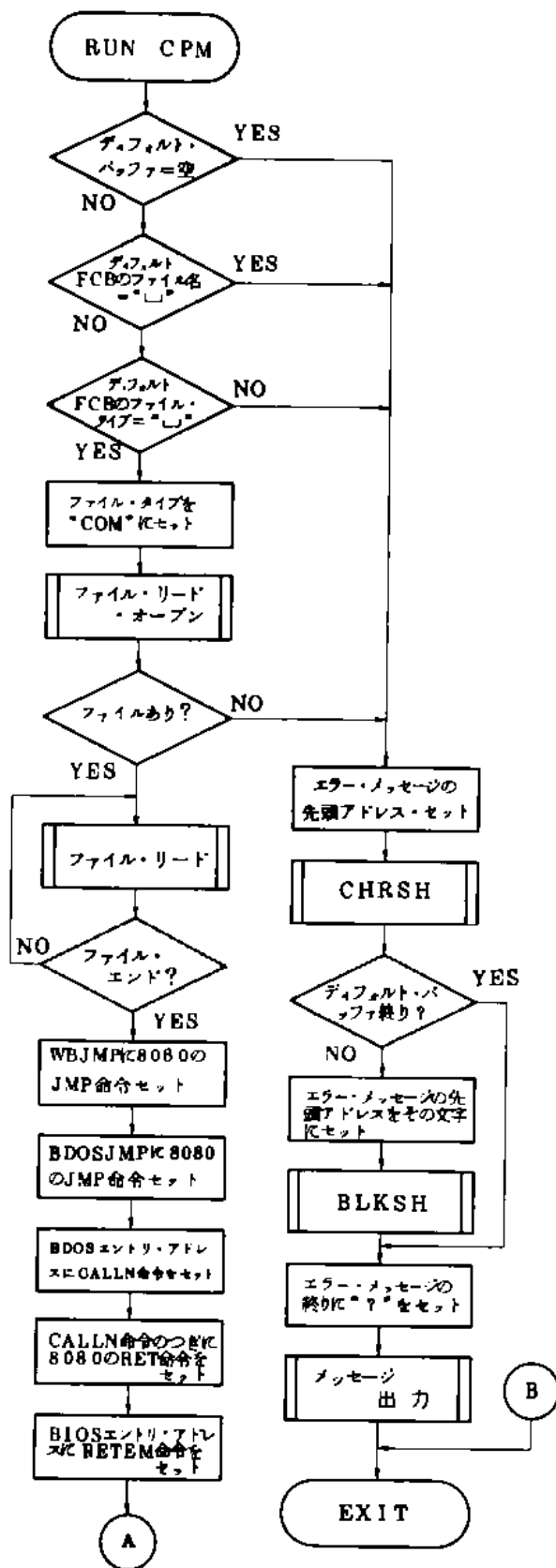
スタートアップ・プログラムの中で、応用プログラムのためのディフォルトFCBエリアを作成する時には、第4.4図で示す状態せんい図に従い、ドライブ番号、ファイル名、ファイル・タイプを設定します。この場合、区切り文字(DLMT)として"=", ":", ";", "<", ">", ".", " ", スペースを使用しています。

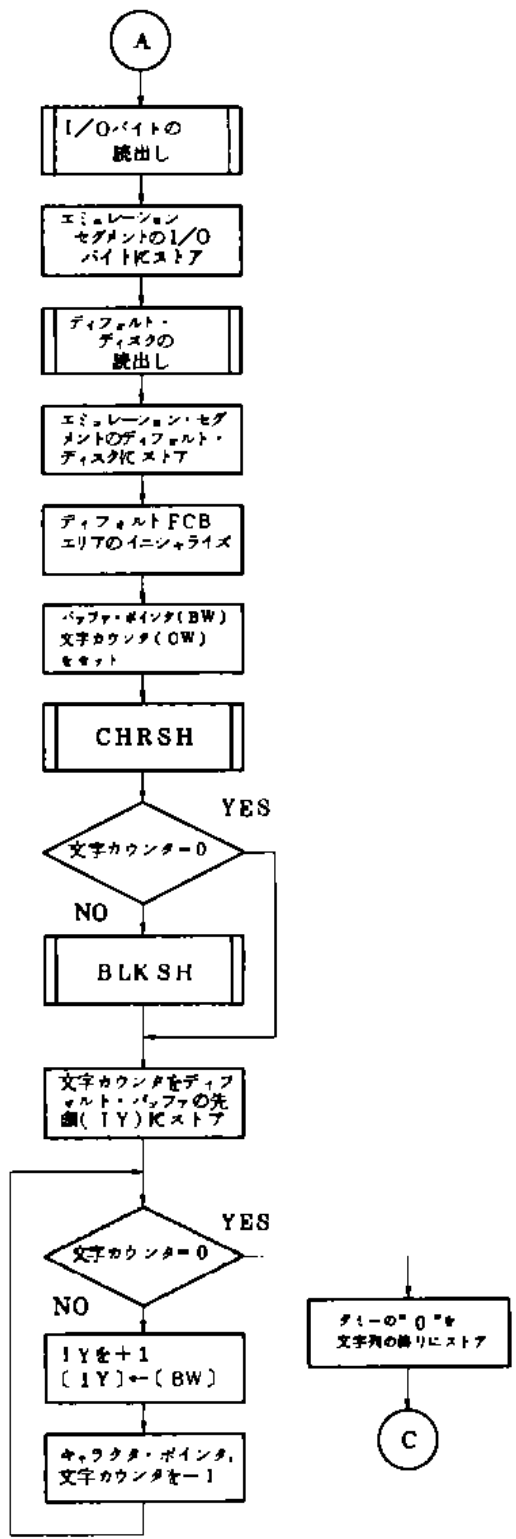
ディフォルトFCBエリアには、最初のパラメータに対するFCBを作成しますが、最初のパラメータと2番目のパラメータの区切り文字がスペースの場合、ディフォルトFCBエリアの一部に、2番目のパラメータに対するドライブ番号、ファイル名、ファイル・タイプを示すデータを設定します。

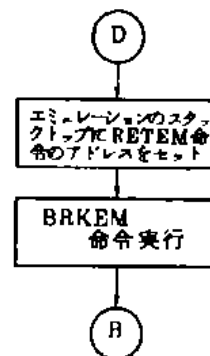
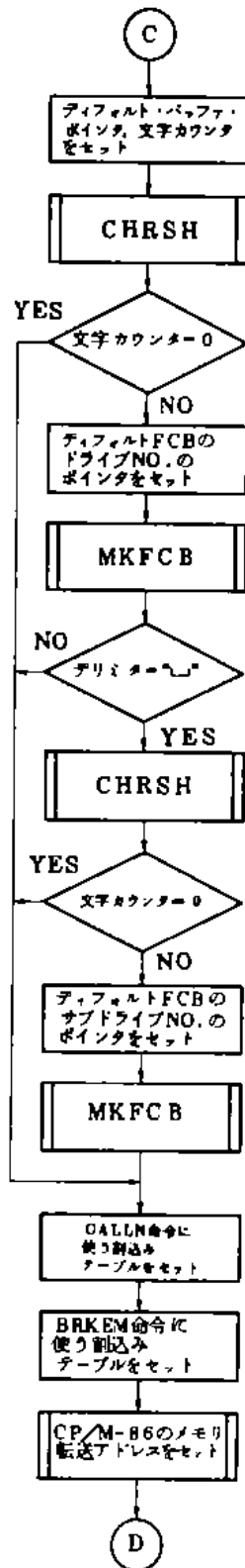
第 4.4 図

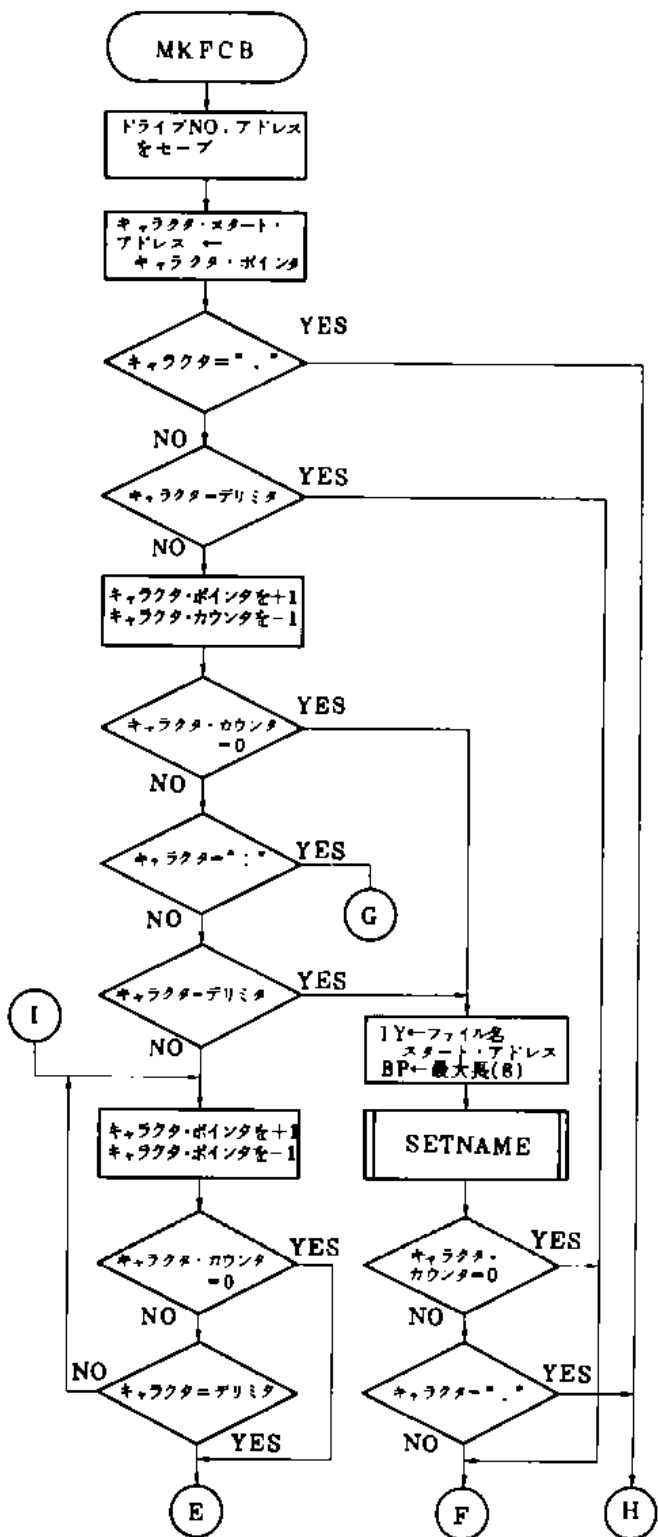


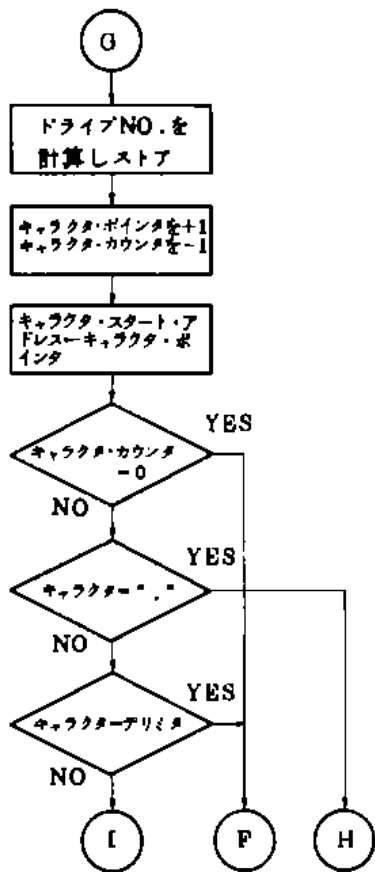
4.6 スタートアップ・プログラム・フローチャート

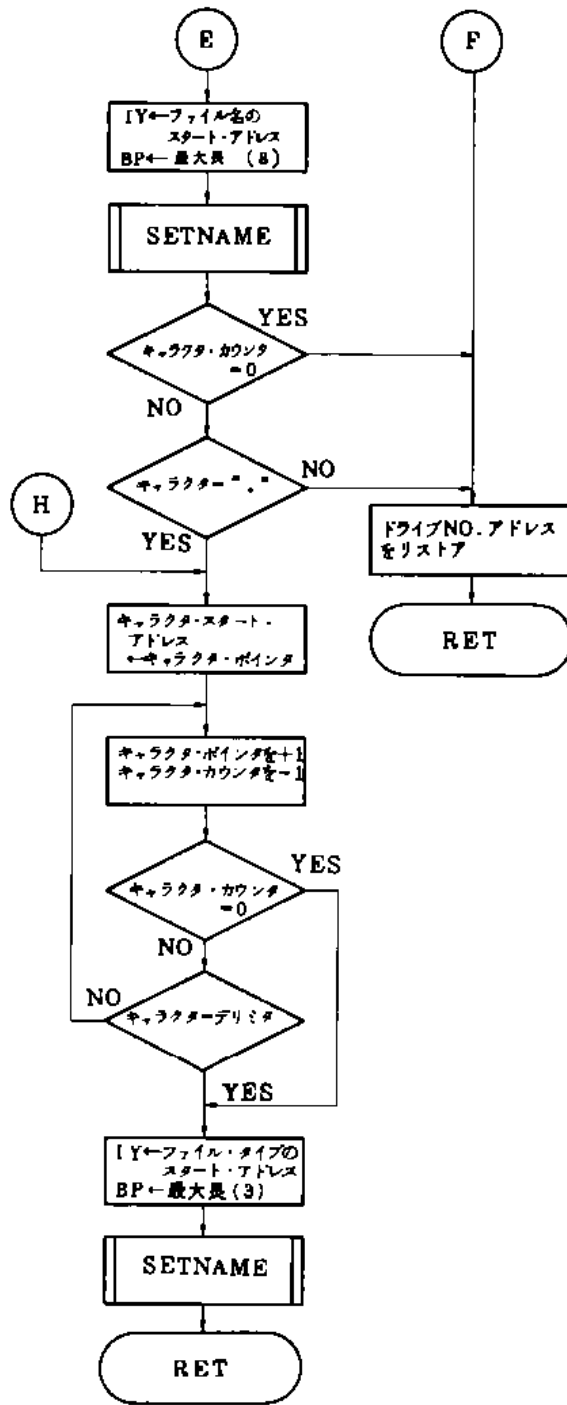


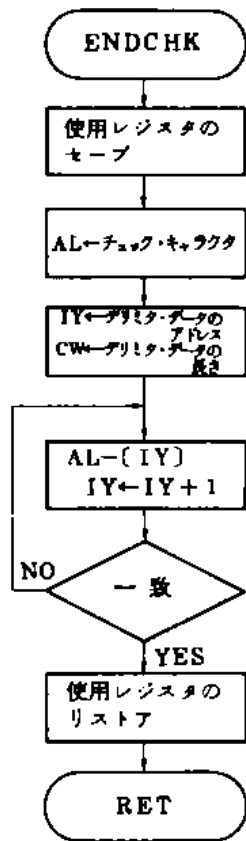


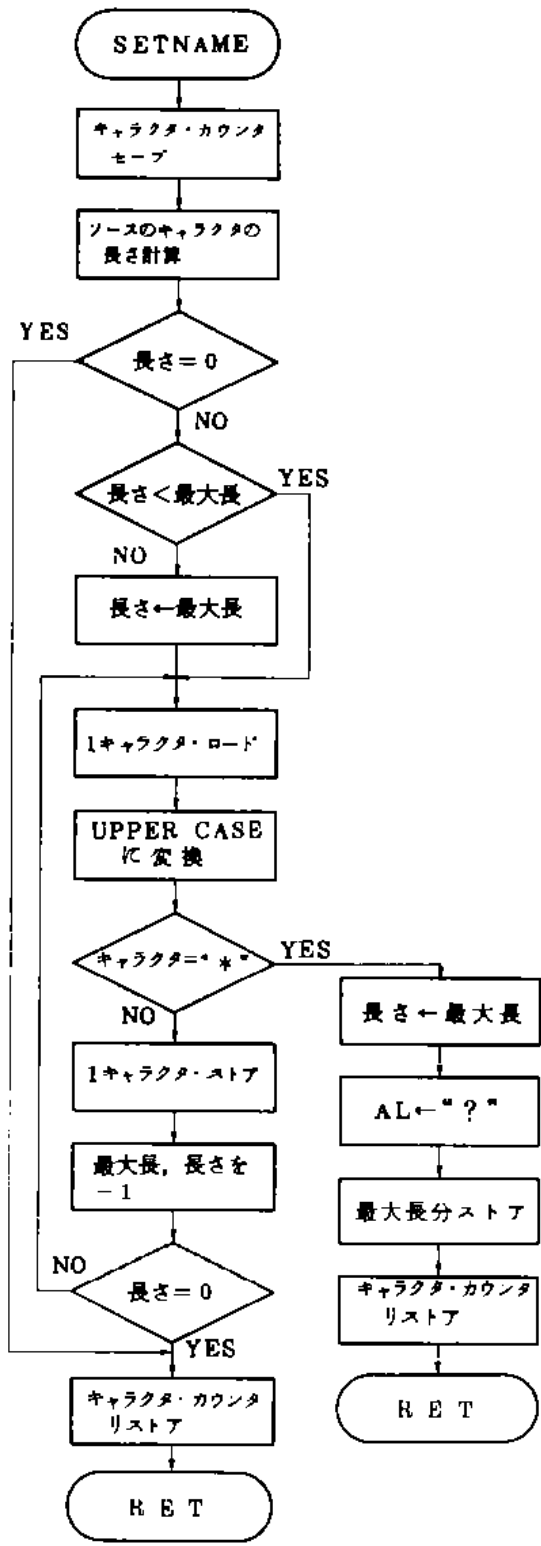


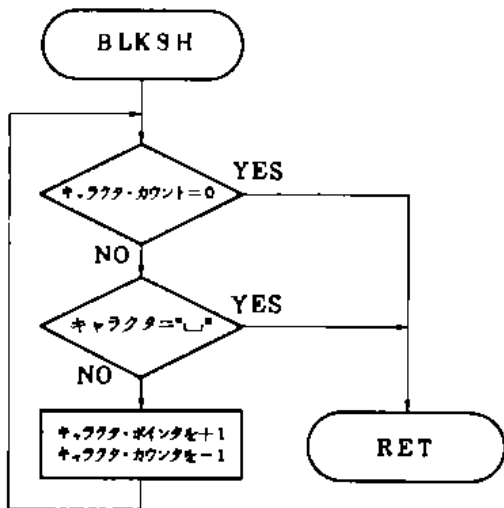
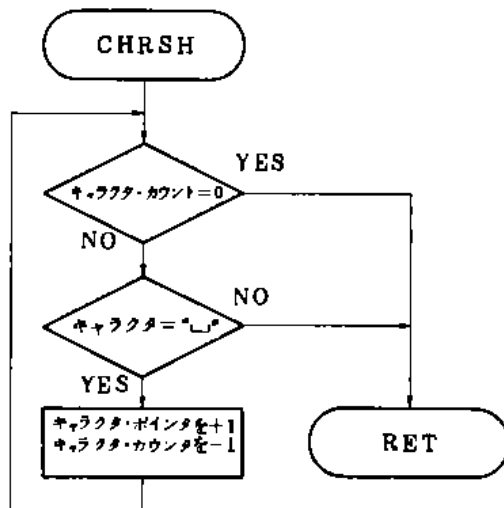












4.7 BDOSコール・エミュレーション・プログラムの処理

CP/M-80のBDOSコールでは、BDOSコールの種類をCレジスタ、8ビットのパラメータをEレジスタ、16ビットのパラメータをDEレジスタにより指定し、処理の結果は、8ビットの場合ALレジスタ、16ビットの場合HLレジスタにセットされます。これに対し、CP/M-86のBDOSコールでは、BDOSコールの種類をCLレジスタ、8ビットのパラメータをDLレジスタ、16ビットのパラメータをDWレジスタにより指定し（アドレスをパラメータとする場合セグメントはDS0により指定）処理の結果は、8ビットの場合ALレジスタ、16ビットの場合BWレジスタにセットされます。（処理結果がアドレスの場合セグメントはDS1にセットされます。）

また、同じ処理番号のBDOSコールは、CP/M-80とCP/M-86で同じ処理を行いません。

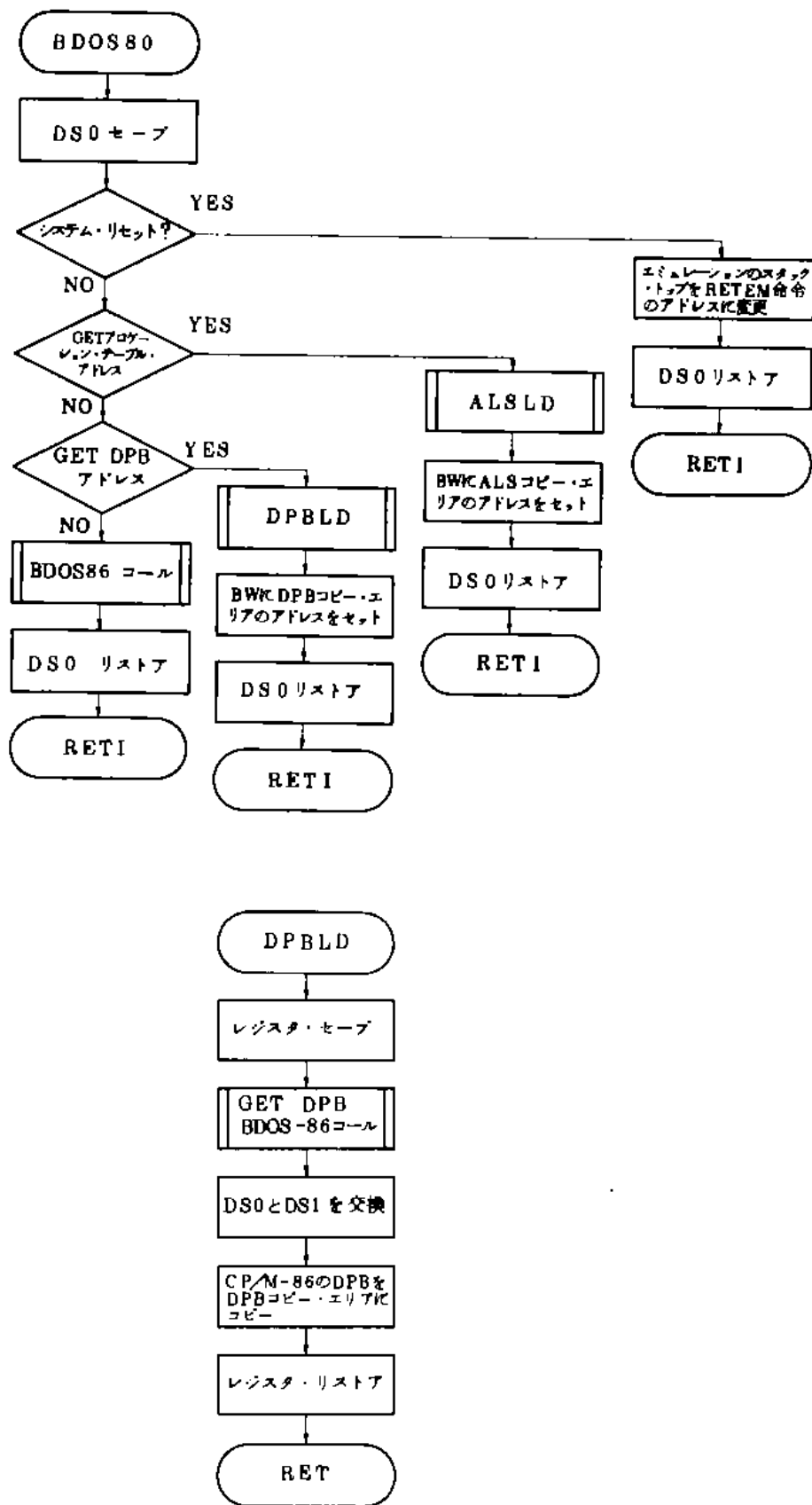
エミュレーション・モードでは、8080のA, C, DE, HLの各レジスタは、ネイティブ・モードのAL, CL, DW, BWレジスタに割当てて使用されており、CP/M-80のBDOSコールは、ほとんどの場合、そのままCP/M-86のBDOSコールでエミュレーションできます。ただし、処理結果としてアドレスが得られるようなBDOSコールでは、DS1とBWレジスタに結果がセットされますが、エミュレーション・モード中は、DS1で指定されるセグメントに対してアクセスする手段がないため、そのままではBDOSコールはエミュレーションできません。

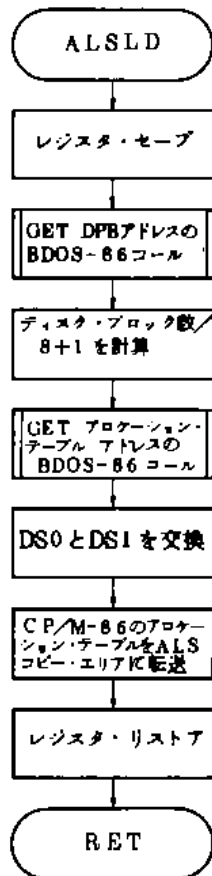
処理結果がアドレスで得られるBDOSコールとして、ディスク・パラメータ・ブロックのアドレスを求めるものと、ディスク・アロケーション・テーブルのアドレスを求めるものがあります。

これらのBDOSコールの場合、CP/M-86のBDOSコールで得られた結果によりエミュレーション・セグメント内にディスク・パラメータ・ブロックやアロケーション・テーブルをコピーし、CP/M-80の処理結果としてコピーした領域のアドレスをHLレジスタにセットして、BDOSコールからリターンします。

また、システム・リセットのBDOSコールは、このプログラムの場合、エミュレーション・モードを終了し、ネイティブ・モードに戻ったあと、CP/M-86のシステム・リセットのBDOSコールを実行する必要があるため、このプログラムでは、エミュレーション・モードで使用しているスタックを操作し、8080のRET命令の戻り先をRETEM命令のアドレスに変更した後リターンします。

4.8 BDOSコール・エミュレーション・プログラムチャート





4.9 CP/M-80E ミュレーション・プログラム・
プログラムのリスト

UPD70105/70116 ASSEMBLER V1.0 RUNCPM

DATE:25 Apr 85 11-28-19 PAGE: 1

SOURCE FILE : A:RUNCPM.SRC
OBJECT FILE : A:RUNCPM.REL (RUNCPM)
COMMAND : A:RUNCPM.SRC XREF

```

LOC 09J          LINE      SOURCE
                  1          ;.....
                  2          ;*
                  3          ;*
                  4          ;*   RUN CP/M-80 TRANSIENT PROGRAM
                  5          ;*   CN V20/V30 BASED CP/M-86
                  6          ;*
                  7          ;*   USAGE: RUNCPM  COMMAND  ARGUMENTS
                  8          ;*
                  9          ;*   COPYRIGHT (C) 1985 NEC CORPORATION
                 10          ;.....
                 11          NAME      RUNCPM
                 12
                 13          ;.....
                 14          ;* CP/M-86 BOOS CALL N.O. DEFINITION
                 15          ;.....
                 16
                 17          RETCPM      EQU      0
                 18          GET10BYTE  EQU      7
                 19          SOUT      EQU      9
                 20          OPEN      EQU      15
                 21          READ      EQU      20
                 22          GETDISK   EQU      25
                 23          SETDMA    EQU      26
                 24          SETSEG    EQU      51
                 25          ;
                 26          ;.....
                 27          ;* INTERRUPT VECTOR DEFINITION
                 28          ;.....
                 29
                 30          PROSB6    EQU      224
                 31          EDOSEMULATE EQU      255
                 32          EENTRY     EQU      254
                 33          ;
                 34          $EJECT
    
```

OC0D
OC07
OC09
J00F
UC14
UC19
OC1A
UC33

JCE0
OCFF
OCFE


```

LDC OBJ          LINE      SOURCE
                35          ;*****
                36          ;= SEGMENT FOR 8080 EMULATION CODE (64K BYTE)
                37          ;*****
                38
                39          EMSEG  SEGMENT1
                40          ;
0000            41          ORG    0
0000 ??         42          M3JMP  02    ?    ;WBOOT JMP INSTRUCTION FOR CP/M-80
0001 ?????     43          M8ADDR 04    ?
0003 ??        44          10BYTE 08    ?    ;10BYTE FOR CP/M-80
0004 ??        45          DISKNO 08    ?    ;DEFAULT DISK N.O. FOR CP/M-80
0005 ??        46          8005JMP 09    ?    ;8005 JMP INSTRUCTION FOR CP/M-80
0006 ?????     47          805ADDR 04    ?
                48          ;
0008            49          ORG    5CH
035C (16       50          FCB    03    16 DUP(?) ;DEFAULT FCB FOR CP/M-80
??
)
                51          ;
006C (20       52          SUBFCB  08    20 DUP(?)
??
)
                53          ;
0090            54          ORG    90H
0090 (128      55          BUF    08    128 DUP(?) ;DEFAULT BUFFER AREA FOR CP/M-80
??
)
                56          ;
0100            57          ORG    100H
0100            58          TPA    LABEL BYTE ;TPA AREA FOR CP/M-80
0100            59          ORG    DEC06H
E006 (3        60          CALLCODE 08    3 DUP(?) ;8005 ENTRY ADDRESS
??
)
E059 ??        61          RETCODE 08    ?    ;RET INSTRUCTION FROM 8005
                62          ;
E00A            63          ORG    0FAD0H
FADD            64          ENSTACK LABEL WORD ;EMULATION STACK AREA
                65          ;
FADD            66          ORG    07A03H
FAD3 (12       67          RETERCODE 08    2 DUP(?) ;WBOOT ENTRY ADDRESS
??
)
                68          ;
FAD5 (13       69          OP80C  08    13 DUP(?) ;DISK PARAMETER BLKCK COPY FOR CP/M-80
??
)
                70          ;
FAD6            71          ALS80  LABEL BYTE ;DISK ALLOCATION TABLE COPY FOR CP/M-80
                72          ;
                73          EMSEG  ENDS
                74          ;
                75          SEJECT

```

```

LOC  OBJ          LINE      SOURCE
                                     ;*****
                                     ;=  PROGRAM SEGMENT FOR
                                     ;=  START UP ROUTINE & BDCS-80 EMULATOR
                                     ;*****
-----
91      RUNCPM SEGMENT
92      ASSUME  PS:RUNCPM,DS:EMSEG,DS1:EMSEG
93      ;
0000  4B8000      84      MOV     BX,OFFSET BUF          ;LOAD CHARACTER POINTER &
0003  1A07        85      MOV     AL,[BX]                ;CHARACTER COUNTER
0005  1A0E        86      MOV     CL,AL
0007  350C        87      MOV     CH,0
0009  43          88      INC     BX
000A  220C        89      AND     AL,AL
000C  751E        90      OR     CX,CF
91      ;
92      ;*****
93      ;=  ERROR MESSAGE OUTPUT ROUTINE
94      ;*****
95      ;
000E  4B8100      96      ERPO01: MOV    BX,OFFSET BUF+1
0011  3B03        97      MOV    DX,BX
0013  28002       98      CALL  CHRSH                   ;SEARCH 1'ST CHARACTER OF 1'ST ARGUMENT
0016  2305        99      CMZ   ERPO02
100     ;
0018  3B03       101     MOV    DX,BX                   ;SET 8DOS PARAMETER (START ADDR OF MSG
001A  2B0502     102     CALL  BLKSH                   ;SEARCH END CHARACTER OF 1'ST ARGUMENT
001D  ;
103     ERPO02:
104     ;
0018  C6073F     105     ERPO03: MOV    BYTE PTR [BX],'?'   ;SET ERROR MSG
0020  C6470124   106     MOV    BYTE PTR [BX+1],'S'
0024  3109       107     MOV    CL,SOUT                ;OUTPUT MESSAGE
0026  C0E0       108     BRK   9D0586
109     ;
0028  3100       110     EXIT:  MOV    CL,RETCPM         ;EXIT
002A  C0EC       111     BRK   8D05E6
112     ;
002C  ;
002C  103E5B002D 113     CHKFMAL:
0031  7409       114     CMP    FC5+1,' '              ;CHECK FILE NAME
0031  7409       115     JZ    ERPO01
116     ;
0033  403E65002D 117     CMP    FC8+9,' '              ;CHECK FILE TYPE
0036  7504       118     JNZ   ERPO01
119     ;
003A  51         120     PUSH  CX                      ;SAVE CHARACTER COUNTER
003B  C606650043 121     MOV    FC8+9,'C'              ;SET FILE TYPE TO 'CCM'
0040  C60665004F 122     MOV    FC8+10,'D'
0045  C60667004D 123     MOV    FC8+11,'M'
124     ;
004A  310F       125     MOV    CL,OPEN                ;READ OPEN
126     ;
004C  8A5C00     127     MOV    DX,OFFSET FC8
004F  C0EC       128     BRK   9D0586
129     ;
0051  59         130     POP  CX

```

LOC	OBJ	LINE	SOURCE
C352	FECC	131	INC AL ;CHECK XXX.COM FILE ON DISK ?
0054	7482	132	BZ ERROR1
		133	;
		134	;
		135	*****
		136	;* LOAD CP/M-80 TRANSIENT PROGRAM *
		137	*****
0056	3A8C00	138	MOV DW,OFFSET IPA-20H
0057	31C28000	139	RDCOM: ADD DW,80H
0058	52	140	PUSH DW
005E	811A	141	MOV CL,SETDMA ;SET LOAD ADDRESS
0060	C8FC	142	BRK 800586
		143	;
0062	8114	144	POV CL,READ ;READ 1 SECTOR
0064	DA3C30	145	MOV DW,OFFSET FCB
0067	C8E0	146	BRK 800526
		147	;
0069	5A	148	PDP DW
006A	22CC	149	RND AL,AL ;CHECK END OF FILE ?
006C	74E8	150	BZ RDCOM
		151	;
		152	*****
		153	;* SET UP FOR CP/M-80 BASE PAGE & BDOS/BIOS AREA *
		154	*****
		155	*****
006E	C6063030C3	156	MOV WBJMP,0C3H ;SETUP FOR WBOOT JMP CODE FOR 8030
0073	C706010C03FA	157	POV WBAADR,OFFSET PETERCODE
		158	;
0079	C6060530C3	159	MOV 9005JMP,0C3H ;SETUP FOR BDOS JMP CODE FOR 806C
007E	C706063036EC	160	MOV 9005ADDR,OFFSET CALLNCODE
		161	;
0084	C60606ECCD	162	MOV CALLNCODE,CE3H ;SETUP FOR CALLN CODE FOR 2080
0089	C60607ECCD	163	MOV CALLNCODE+1,DEDH
008E	C60608ECCF	164	MOV CALLNCODE+2,BD05EMULATE
		165	;
0093	C60609ECC9	166	MOV RETCODE,0C9H ;SETUP FOR RET CODE FOR 808U
		167	;
0093	C60603FAED	168	MOV PETERCODE,0EDH ;SETUP FOR RETEM CODE FOR 2020
0098	C60604FAFD	169	MOV PETERCODE+1,DFBH
		170	;
0042	3107	171	MOV CL,GETIOBYTE ;SETUP FOR CP/M-80 IOBYTE
0044	C8E0	172	BRK 800526
		173	;
00A6	42E300	174	MOV IOBYTE,AL
		175	;
00A9	3119	176	MOV CL,GETDISK ;SETUP FOR CP/M-80 DEFAULT DISK N.O.
00AB	C8E0	177	BRK 800526
		178	;
00AB	A20400	179	MOV DISKNO,AL
		180	;
0090	8C03	181	MOV AD,DS0 ;INITIALIZE DEFAULT FCB AREA
0092	3ECC	182	MOV DS1,4H
0094	8F3C00	183	MOV IT,OFFSET FCB
0097	300C	184	MOV AL,D
0099	AA	185	STP FCB

LOC	OBJ	LINE	SOURCE
		186	;
008A	8020	187	MOV AL,' '
008C	8908C0	188	MOV CW,8*3
008F	F3AA	189	REP ST# FCB
		190	;
00C1	800C	191	MOV AL,0
00C3	890400	192	MOV CW,LENGTH FCB-1-8-3
00C6	F3AA	193	REP ST# FCB
		194	;
00C3	800C	195	MOV AL,0
00CA	AA	196	ST# SUBFCB
		197	;
00CB	802C	198	MOV AL,' '
00CD	8908C0	199	MOV CW,8*3
00D0	F3AA	200	REP ST# SUBFCB
		201	;
00D2	800C	202	MOV AL,0
00D4	890800	203	MOV CW,LENGTH SUBFCB-1-8-3
00D7	F3AA	204	REP ST# SUBFCB
		205	;
0089	8880D0	206	MOV BW,OFFSET BUF ;SETUP FOR DEFAULT BUFFER AREA
008C	8A0F	207	MOV CL,[BW] ;LOAD CHARACTER POINTER B
008E	850C	208	MOV CH,0 ;CHARACTER COUNTER
0090	43	209	INC BW
00E1	883201	210	CALL CHRSH ;SEARCH 1'ST CHARACTER OF 1'ST ARG
00E4	E303	211	BCWZ SETLEN
		212	;
00E6	E83901	213	CALL BLKSH ;SEARCH END CHARACTER OF 1'ST ARG
00E9	8F3C00	214	SETLEN: MOV IY,OFFSET BUF ;STORE CHARACTER LENGTH
00EC	8800	215	MOV [IY],CL
00EE	E30A	216	SETCHR: BCWZ SETBMY
		217	;
00FD	47	218	INC IY
00F1	8A0F	219	MOV AL,[BW] ;REMOVE COMMAND NAME
00F3	8805	220	MOV [IY],AL
00F5	43	221	INC BW
00F6	FEC9	222	DEC CL
00F8	E8F4	223	OR SETCHR
		224	;
00FA	47	225	SETBMY: INC IY ;STORE DUMMY
00FB	C80500	226	MOV BYTE PTR [IY],0
		227	;
		228	*****
		229	; MAKE DEFAULT FCB
		230	*****
		231	;
00FE	8880D0	231	MOV BW,OFFSET BUF ;MAKE FCB
0101	8A0F	232	MOV CL,[BW]
0103	850C	233	MOV CH,0
0105	43	234	INC BW
0106	880001	235	CALL CHRSH ;SEARCH 1'ST CHARACTER 1'ST ARG
0107	E31A	236	BCWZ SETITBL
		237	;
0108	8F5C00	238	MOV IY,OFFSET FCB
010E	E84700	239	CALL #KFCB
0111	803F20	240	CMP BYTE PTR [BW],' ' ;IF DELIMITER = ' ' THEN MAKE SUB FCB

LOC	OBJ	LINE	SOURCE
0114	7502	241	BNZ SETITBL
		242	;
0116	E8FD00	243	CALL CMPSH ;SEARCH 1'ST CHARACTER OF 2'ND A4G
0119	E306	244	BCWZ SETITBL
		245	;
0116	BFACD0	246	MOV IX,OFFSET SUBFCB
0116	E83700	247	CALL PKFCB
0121		248	SETITBL:
0121	BC0A	249	MOV BM,B50 ;SET INTERRUPT TABLE
0123	380000	250	MOV AW,0
0126	BE01	251	MOV D50,AW
		252	;
0128	H8FC03	253	MOV PW,BDOSEMULATE+4 ;CALLN VECTER TABLE
0129	C7072E02	254	MOV WORD PTR [BW],OFFSET BDOSSDE
012F	83C302	255	ADD BM,2
0132	HCCF	256	MOV [BW],PS
		257	;
0134	39F2G3	258	MOV BM,EMENTRY+4 ;BRKEN VECTER TABLE
0137	C707C0U1	259	MOV WORD PTR [BW],OFFSET TPA
0139	53C302	260	ADD BM,2
013E	3917	261	MOV [BW],BW
		262	;
0140	BEBA	263	MOV D50,BW
		264	;
0142	3AB0J0	265	MOV BW,OFFSET EUF ;SET DEFAULT DMA ADDRESS FOR CP/M-80
0145	911A	266	MOV [L,SEYDMA
0147	CD0D	267	BRK ED0586
		268	;
0149	30FF09	269	MOV BP,OFFSET ENSTACK-2 ;SETUP SP FOR CP/M-80 E
		270	;RETURN ADDRESS FOR TRANSIENT PRO.
014C	5EC7443C03FA	271	MOV D50:[BP],OFFSET RETEMCODE
0152	JFFFFE	272	BRKEN EMENTRY ;START EMULATION MCODE
		273	;
0155	E9D0FE	274	SR EXIT
		275	;
		276	SEJECT

LOC	OBJ	LINE	SOURCE
		277	;
		278	;* MAKE DRIVE N.O FILE NAME & FILE TYPE *
		279	;
		280	;* PARAMETER IY:DRIVE N.O. ADDRESS *
		281	;* BW: 1*ST CHARACTER ADDRESS *
		282	;* CW: CHARACTER COUNT *
		283	;
		284	;
0158	57	285	PKFCB: PUSH IY ;SAVE FILE NAME START ADDRESS
0159	4363	286	MOV IX,BW
0159	403F2E	287	CMP BYTE PTR [BX],'.'
015E	7457	288	JZ PKFCA
		289	;
0160	EB69D0	290	CALL ENDCMK
0163	741F	291	BZ PKFCBEX
		292	;
0165	43	293	PKFC1: INC BX
0166	49	294	DEC CW
0167	74DA	295	JZ PKFC11
		296	;
0169	403F3A	297	CMP BYTE PTR [BX],'.'
016C	7417	298	JZ PKFC2
		299	;
016E	EB5D00	300	CALL ENDCMK
0171	752B	301	BWZ PKFC3
		302	;
0173		303	PKFC11:
0173	5F	304	POP IY ;SET FILE NAME
0174	57	305	PUSH IY
0175	67	306	INC IY
0176	400E00	307	MOV SP,B
0179	EB6F00	308	CALL SETNAME
017C	E305	309	BCWZ PKFCBEX
		310	;
017E	403F2E	311	CMP BYTE PTR [BX],'.'
0181	7434	312	BZ PKFC4
		313	;
0183		314	PKFCBEX:
0185	5F	315	POP IY ;EXIT
0184	C5	316	RET
		317	;
0185	2A47FF	318	PKFC2: MOV AL,[BX-1] ;SET DRIVE N.O.
0186	245F	319	AND AL,5FH
018A	2C40	320	SUB AL,'A'-1
018C	8803	321	MOV [IY],AL
018E	43	322	INC BX
018F	49	323	DEC CW
0190	69F3	324	MOV IX,BW
0192	74EF	325	BZ PKFCBEX
		326	;
0194	403F2E	327	CMP BYTE PTR [BX],'.'
0197	741E	328	BZ PKFCA
		329	;
0199	EB32D0	330	CALL ENDCMK
019C	74E5	331	BZ PKFCBEX

LOC	OBJ	LINE	SOURCE
		332	;
019E	43	333	MKFC3: INC SW
019F	49	334	DEC CW
01A0	7405	335	BZ MKFC31
		336	;
01A2	582900	337	CALL ENDCMK
01A5	75F7	338	BNZ MKFC3
		339	;
01A7	5F	340	MKFC31: POP IY ;SET FILE NAME
01A8	57	341	PUSH IY
01A9	47	342	INC IY
01AA	5D0200	343	MOV BP,B
01AB	583900	344	CALL SETNAME
01AD	5301	345	BCWZ MKFCBEX
		346	;
01B2	503F2E	347	CMP BYTE PTR [BX],'
01B5	75CC	348	BNZ MKFCBEX
		349	;
01B7	8BF3	350	MKFC4: MOV IX,BX
01B9	46	351	INC IX
01BA	43	352	MKFC41: INC SW
01BB	49	353	DEC CW
01BC	7405	354	BZ MKFC42
		355	;
01BE	580900	356	CALL ENDCMK
01C1	75F7	357	BNZ MKFC41
		358	;
01C3	5F	359	MKFC42: POP IY ;SET FILE TYPE
01C4	53C709	360	ADD IY,9
01C7	5D0300	361	MOV BP,3
01CA	581E00	362	CALL SETNAME
01CD	53	363	RET ;EXIT
		364	;
		365	SEJECT

```

LOC  OBJ          LINE      SOURCE
                                     ;*****
                                     ;* MKFCB DELIMITER CHECK SUB ROUTINE *
366                                     ;*
367                                     ;*
368                                     ;*
369                                     ;* PARAMETER 96: CURRENT CHARACTER ADDRESS *
370                                     ;*****
371
01CE  51          372      ENDCHK: PUSH   CW
01CF  57          373              PUSH   IX
01D0  06          374              PUSH   BX1
01D1  0E          375              PUSH   PS
01D2  07          376              POP    DS1
377      ;
378      ;      ASSUME DS1:RUNCPM
379      ;
01D3  3A07        380              MOV    AL,[BX]
01D5  3F230190    381              MOV    IX,OFFSET ENDDATA
01D7  09C80040    382              MOV    CX,LENGTH ENDDATA
01D8  F28E        383      REPZ   CMPS  ENDDATA
384      ;
01D9  07          385              POP    BX1
386      ;
387      ;      ASSUME DS1:EMSEG
388      ;
01E0  5F          389              POP    IX
01E1  59          390              POP    CX
01E2  03          391              RET
392      ;
01E3  303A593C5E3F2E2C  393      ENDDATA DB  '=><_ ' ;DELIMITER CHARACTERS
394      ;
395      ;*****
396      ;* MKFCB FILE NAME & FILE TYPE SET SUB ROUTINE *
397      ;*
398      ;* PARAMETER 06: END CHARACTER ADDRESS *
399      ;*          IX: START CHARACTER ADDRESS *
400      ;*          BP: MAX LENGTH *
401      ;*          IX: DESTINATION START ADDRESS *
402      ;*****
403
01E8          404      SETNAME:
01E9  51          405              PUSH   CW
01EA  8BCB        406              MOV    CW,BX ;GET CHARACTER LENGTH
01EB  2BCE        407              SUB    CW,IX
01EC  E322        408              SCMB  SETNA1
409      ;
01ED  3BCB        410              CMP    CW,BP
01EE  7202        411              BL   SETNA2
412      ;
01EF  68CD        413              MOV    CW,BP ;IF LENGTH > MAX LENGTH
01F0          414      SETNA2:
01F1  FC          415              CLD
01F2          416      SETNA3:
01F3  AC          417              LDB  RUF ;LOAD CHARACTER
01F4  3C61        418              CMP  AL,'a' ;CONVERT LOWER CASE TO UPPER CASE
01F5  7206        419              BL   SETNA3
420      ;

```


LOC	OBJ	LINE	SOURCE
01F5	3C7A	421	EMP AL,'?'
0200	7302	422	9ML SETNA3
		423	;
0202	2C2C	424	SU? AL,'B'-'A'
0204		425	SETNA3:
0204	3C2A	426	EMP AL,'@'
0206	7406	427	BZ SETNA4
		428	;
0208	AA	429	STM FCB ;SET CHARACTER TO FCB AREA
0209	4D	430	DEC BP
020A	E2ED	431	DBNZ SETNA5
		432	;
020C	59	433	POP CM
020D	C3	434	RET
		435	;
020E		436	SETNA4:
020E	3BCD	437	MOV CM,BP
0210	303F	438	MOV AL,'?'
0212	F3AA	439	REP STP FCB ;IF WILD CARD THEN SET CHARACTER TO '?'
0214		440	SETNA1:
0214	59	441	POP CM
0215	C3	442	RET
		443	;
		444	REJECT

LOC	OBJ	LINE	SOURCE
		445	;
		446	;A SEARCH 1'ST CHARACTER (NOT ' ')
		447	;
		448	;A PARAMETER BW: CHARACTER ADDRESS
		449	;A CW: CHARACTER LENGTH
		450	;
		451	;
0216	E309	452	CHPSH: BCWZ CHPSH1
		453	;
0219	303F20	454	EMP BYTE PTR [BW],'
0218	7504	455	BNZ CHPSH1
		456	;
0216	43	457	INC BW
0216	49	458	DEC CW
0217	E8F5	459	OR CHPSH
		460	;
0221	C3	461	CHPSH1: RET
		462	;
		463	;
		464	;A SEARCH 1'ST ' '
		465	;
		466	;A PARAMETER BW: CHARACTER ADDRESS
		467	;A CW: CHARACTER LENGTH
		468	;
		469	;
0222	E309	470	BLKSH: BCWZ BLKSH1
0224	303F20	471	EMP BYTE PTR [BW],'
0227	7404	472	BZ BLKSH1
		473	;
0229	43	474	INC BW
022A	49	475	DEC CW
022B	E8F5	476	SR BLKSH
		477	;
022D	C3	478	BLKSH1: RET
		479	;
		480	REJECT

```

LOC OBJ          LINE      SOURCE
                                481      ;*****
                                482      ;*  BDOS-80 EMULATOR          *
                                483      ;*                               *
                                484      ;*  PARAMETER  CL:  BDOS CALL N.O.  *
                                485      ;*                               *
                                486      ;*  CW:  BDOS CALL ADDRESS PARAMETER *
                                487      ;*****
022E             488      BDOS90E:
022E 1E          489      PUSH    DSI
022F 50F9D0      490      CMP     CL,D          ;CHECK EXIT CALL ?
0232 741E          491      BZ     BDOSEXIT
                                492      ;
                                493      CMP     CL,1BH      ;GET ALLOCATION TABLE ADDRESS ?
0234 50F71B      494      BZ     GETALS
                                495      ;
                                496      CMP     CL,1FH      ;GETDISK PARAMETER BLOCK ADDRESS ?
0239 80F91F      497      BZ     GETDPB
023C 7404          498      ;
023E C0E0          499      BRK     BDOSR0      ;CALL BDOS-80
                                500      ;
0240 1F          501      POP     DSI
0241 CF          502      RETI
                                503      ;
0242 E815D0      504      GETDPB: CALL   DPBLD      ;COPY DISK PARAMETER BLOCK
0243 50U5FA      505      MOV     DI,OFFSET DPPBC ;RETURN DPB COPY AREA ADDRESS
0248 1F          506      POP     DSI
0249 CF          507      RETI
                                508      ;
024A 5523D0      509      GETALS: CALL   ALSLD      ;COPY DISK ALLOCATION TABLE
024B 5014FA      510      MOV     DI,OFFSET ALSBC ;RETURN ALLOCATION TABLE COPY AREA ADDRESS
0250 1F          511      POP     DSI
0251 CF          512      RETI
                                513      ;
0252             514      BDOSEXIT:
0252 3EC746D003FA 515      MOV     DI:WORD PTR [BP],OFFSET RETENCODE ;CHANGE BDOS RETURN ADDRESS TO WBOOT ADDRESS
0258 1F          516      POP     DSI
0259 CF          517      RETI
                                518      REJECT

```

```

LOC  OBJ          LINE      SOURCE
                                     ;*****
                                     ;= COPY DISK PARAMETER BLOCK TO CP/M-80 READABLE AREA =
                                     ;*****
025A  6C          523      DPBLD:  PUSH   R
025B  B11F        524      MOV    CL,1FH                ;GET DPB ADDRESS OF CP/M-80
025D  C0EC        525      BRK   B00586
                                     ;
025F  1E          527      PUSH  D50                    ;EXCHANGE D50 & D51
0260  U6          528      PUSH  D51
0261  1F          529      POP   D50
0262  07          530      POP   D51
0263  8BF3        531      MOV   IX,BW
0265  3F05FA      532      MOV   IX,OFFSET DPBSC       ;SET DPB ADDRESS OF CP/M-80
0269  B90FJ0      533      MOV   CX,LENGTH DPBSC      ;SET DPB LENGTH
026A  FC          534      CLRI  D1P
026C  F3A4        535      REP  MOVWB DPBBD,[IX]
                                     ;
026E  07          537      POP   R
026F  C3          538      RET
                                     ;
540      ;*****
541      ;= COPY DISK ALLOCATION TABLE TO CP/M-80 READABLE AREA =
542      ;*****
543
0270  60          544      ALSLD: PUSH  R
0271  B11F        545      MOV   CL,1FH                ;GET DISK PARAMETER BLOCK
0273  C0EC        546      BRK   B00586
                                     ;
0275  26624F05    548      MOV   CX,D51:[BX*5]         ;GET BLOCK SIZE OF DISK
0279  C1E903      549      SHR   CX,5                  ;SIZE/2+1
027C  41          550      INC   CX
027D  51          551      PUSH  CX
027E  3119        552      MOV   CL,1BH                ;GET ALLOCATION TABLE ADDRESS
0280  C0E0        553      BRK   B00586
                                     ;
0242  39          555      POP   CX
0283  1E          556      PUSH  D50                    ;EXCHANGE D50 & D51
0284  06          557      PUSH  D51
0285  1F          558      POP   D50
0286  07          559      POP   D51
0287  8BF3        560      MOV   IX,BW
0289  3F14FA      561      MOV   IX,OFFSET ALS8C      ;SET ALLOCATION TABLE ADDRESS OF CP/M-80
028C  FC          562      CLRI  D1P
028D  F3A4        563      REP  MOVWB ALS8D,[IX]
028F  61          564      POP   R
0290  C3          565      RET
                                     ;
-----          567      ;
568      RUACPM  ENDS
569      ;
570      ENC

```

*** TABLE SYMBOL LISTING ***

NAME	TYPE	VALUE	ATTRIBUTES	XREF
??SEG	SEGMENT		SIZE=000CH PARA PUBLIC	
AL590	BYTE	FA14	EMSEG	71# 510 561 563
AL5LD	NEAR	D27C	RUNCPM	50# 344#
B00580E	NEAR	J22E	RUNCPM	254 488#
B00536	NUMBER	0CE0		30# 108 111 120 142 146 172 177 267 499 525 546 553
B005ADDR	WORD	0006	EMSEG	47# 160
B005EMULATE	NUMBER	00FF		51# 164 253
B005EXIT	NEAR	J232	RUNCPM	491 514#
B005JMP	BYTE	0005	EMSEG	46# 159
HLK5H	NEAR	Q222	RUNCPM	102 213 470# 476
HLK5H1	NEAR	J220	RUNCPM	470 472 478#
JUF	BYTE	0080	EMSEG	55# 84 96 206 214 231 265 417
CALLNCODE	BYTE	ECG6	EMSEG	60# 160 162 163 164
CHKPNAME	NEAR	J02C	RUNCPM	90 113#
CHR5H	NEAR	J216	RUNCPM	98 210 235 243 452# 459
CHR5H1	NEAR	Q221	RUNCPM	452 455 461#
DISKNO	BYTE	J004	EMSEG	45# 179
DP88J	BYTE	FA05	EMSEG	69# 505 532 533 535
DP8LO	NEAR	J25A	RUNCPM	504 523#
EMENTRY	NUMBER	00FE		32# 252 272
EMSEG	SEGMENT		SIZE=FA14H PARA	39# 73# 82 82 387
EMSTACK	WORD	FA00	EMSEG	64# 269
EMDCHK	NEAR	Q10E	RUNCPM	290 300 330 337 356 372#
EMDDATA	BYTE	Q1E3	RUNCPM	381 382 383 393#
ERR0R1	NEAR	000E	RUNCPM	96# 115 119 132
ERR0R2	NEAR	U010	RUNCPM	99 103#
ERR0R3	NEAR	Q010	RUNCPM	105#
EXIT	NEAR	J228	RUNCPM	110# 274
FCB	BYTE	J05C	EMSEG	50# 116 117 121 122 123 127 145 183 185 189 192 193 238 42# 43#
GETALS	NEAR	Q24A	RUNCPM	494 509#
GETDISK	NUMBER	0019		22# 176
GETDPB	NEAR	J242	RUNCPM	497 504#
GETIOBYTE	NUMBER	0007		19# 171
IQ0YTE	BYTE	J003	EMSEG	44# 174
NKFC1	NEAR	Q165	RUNCPM	293#
NKFC11	NEAR	J173	RUNCPM	295 303#
NKFC2	NEAR	Q185	RUNCPM	298 318#
NKFC3	NEAR	J19E	RUNCPM	301 313# 338
NKFC31	NEAR	Q1A7	RUNCPM	315 340#
NKFC4	NEAR	Q187	RUNCPM	288 312 322 350#
NKFC41	NEAR	Q18A	RUNCPM	352# 357
NKFC42	NEAR	Q1C3	RUNCPM	354 359#
NKFC3	NEAR	Q158	RUNCPM	239 247 285#
NKFC0EX	NEAR	Q163	RUNCPM	291 309 314# 525 331 345 348
OPEN	NUMBER	000F		20# 125
HDCON	NEAR	U059	RUNCPM	139# 150
READ	NUMBER	J014		21# 164
RETCODE	BYTE	EC09	EMSEG	61# 166
RETCPM	NUMBER	U000		17# 110
RETCODE	BYTE	FA03	EMSEG	67# 157 169 169 271 515
RUNCPM	SEGMENT		SIZE=C291H PARA	11 61# 82 378 563#
SETCHR	NEAR	J0EE	RUNCPM	216# 223

NAME	TYPE	VALUE	ATTRIBUTES	XREF
SETDMA	NUMBER	001A	23# 141 26C	
SETDMY	NEAR	00FA	RUNCPM 216 225#	
SETITBL	NEAR	0121	RUNCPM 236 247 244 242#	
SETLEM	NEAR	0CE9	RUNCPM 211 214#	
SETMA1	NEAR	021A	RUNCPM 40E 440#	
SETMA2	NEAR	01F3	RUNCPM 411 414#	
SETMA3	NEAR	0204	RUNCPM 419 422 425#	
SETMA4	NEAR	020E	RUNCPM 427 436#	
SETMA5	NEAR	01F9	RUNCPM 416# 431	
SETNAME	NEAR	01E8	RUNCPM 30E 344 362 404#	
SETSEG	NUMBER	0033	24#	
SOUT	NUMBER	0009	19# 107	
SUBFC0	BYTE	006C	EMSEG 32# 19E 200 203 204 246	
TPA	BYTE	0100	EMSEG 5E# 128 259	
#BADD#	WORD	0001	EMSEG 43# 157	
#SJPP	BYTE	006D	EMSEG 47# 156	

ASSEMBLY COMPLETE, 0 ERROR(S) FOUND

アンケート記入のお願い

お手数ですが、このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

[ドキュメント名] V20HL, V30HL アプリケーション・ノート ソフトウェア編
(U12231JJ1V0AN00 (第1版))

[お名前など] (さしつかえのない範囲で)
御社名 (学校名, その他) ()
ご住所 ()
お電話番号 ()
お仕事の内容 ()
お名前 ()

1. ご評価 (各欄に○をご記入ください)

項 目	大変良い	良 い	普 通	悪 い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン, 字の大きさなど					
その他 ()					
()					

2. わかりやすい所 (第 章, 第 章, 第 章, 第 章, その他)
理由 []

3. わかりにくい所 (第 章, 第 章, 第 章, 第 章, その他)
理由 []

4. ご意見, ご要望

5. このドキュメントをお届けしたのは
NEC販売員, 特約店販売員, NEC半導体ソリューション技術本部員,
その他 ()

ご協力ありがとうございました。

下記あてにFAXで送信いただくか、最寄りの販売員にコピーをお渡しください。

キ
リ
ト
リ

— お問い合わせは、最寄りのNECへ —

【営業関係お問い合わせ先】

半導体第一販売事業部 半導体第二販売事業部 半導体第三販売事業部	〒108-01 東京都港区芝五丁目7番1号 (NEC本社ビル)	東京 (03)3454-1111 (大代表)	
中部支社 半導体第一販売部 半導体第二販売部	〒460 名古屋市中区錦一丁目17番1号 (NEC中部ビル)	名古屋 (052)222-2170 名古屋 (052)222-2190	
関西支社 半導体第一販売部 半導体第二販売部 半導体第三販売部	〒540 大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪 (06) 945-3178 大阪 (06) 945-3200 大阪 (06) 945-3208	
北海道支社 東北支社 岩手支店 山形支店 郡山支店 いわき支店 新潟支店 土浦支店 水戸支店 神奈川支社 群馬支店	札幌 (011)231-0101 仙台 (022)267-8740 盛岡 (0198)51-4344 山形 (0236)23-5511 郡山 (0249)23-5511 いわき (0246)21-5511 新潟 (0250)30-2155 土浦 (0298)23-0101 水戸 (029)226-1717 相模原 (045)324-5524 高崎 (0273)26-1265	太田支店 宇都宮支店 小山支店 長野支店 甲府支店 埼玉支店 千葉支店 神岡支店 北井支店 太田 (0276)46-4011 宇都宮 (028)621-2281 小山 (0285)24-5011 長野 (0283)35-1662 甲府 (0552)24-4141 大宮 (048)641-1411 立川 (0425)26-5091 千葉 (043)238-8116 神岡 (054)255-2211 金沢 (0762)23-1621 福井 (0776)22-1666	富山支店 三番支店 京都支店 神戸支店 中国支店 鳥取支店 岡山支店 四国支店 新居浜支店 松山支店 九州支店 富山 (0764)31-8461 津 (0592)25-7341 京都 (075)344-7824 神戸 (078)333-3854 広島 (082)242-5504 鳥取 (0857)27-5311 岡山 (086)225-4455 高松 (0878)36-1200 新居浜 (0897)32-5001 松山 (089)945-4148 福岡 (092)271-7700

【本資料に関する技術お問い合わせ先】

半導体ソリューション技術本部 マイクロコンピュータ技術部	〒210 川崎市幸区塚越三丁目484番地	川崎 (044)548-8890	半導体 インフォメーションセンター FAX(044)548-7900 (FAXにてお問い合わせします)
半導体販売技術本部 日本販売技術部	〒108-01 東京都港区芝五丁目7番1号 (NEC本社ビル)	東京 (03)3798-9619	
半導体販売技術本部 中部販売技術部	〒460 名古屋市中区錦一丁目17番1号 (NEC中部ビル)	名古屋 (052)222-2125	
半導体販売技術本部 西日本販売技術部	〒540 大阪市中央区城見一丁目4番24号 (NEC関西ビル)	大阪 (06) 945-3383	