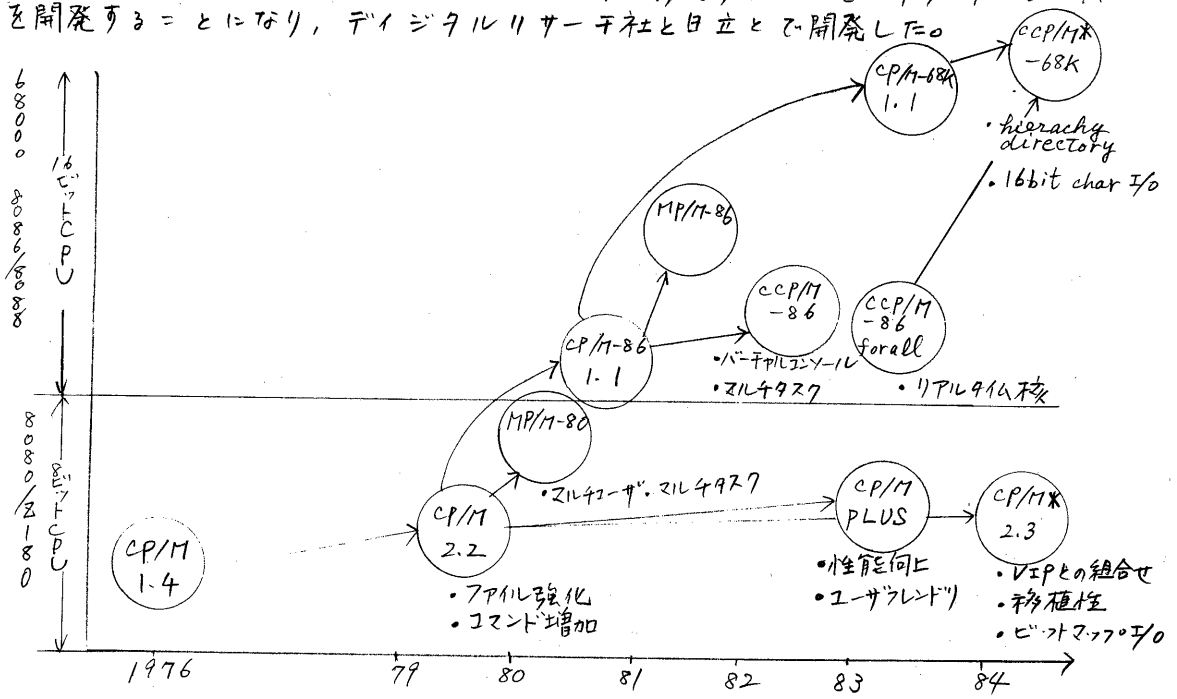
題目　　CP/M－68Kの開発

著者名　日立製作所　武蔵工場　マイコンソフト設計部　尾石辰郎

1. CP/M－68K　開発の経過

1976年米国デイジタル・リサーチ社で開発したシングルタスク・シングルユーザ用CP/M　1・4は，コンパクトにまとまり且つ移植性に優れているため，パーソナルコンピュータの標準OSとして捨てない地位を築いている。当初の8080及びZ80内蔵機種用に続いて，1980年には8086及び8088内蔵機種用CP/M－86，今回68000内蔵機種用CP/M－68Kを開発した。CP/M－86の記述言語PL/Mと日立が開発したS－PL/Hとが互換性があり，PL/MからS－PL/Hへ移行することでCP/M－68Kを開発することになり，デイジタルリサーチ社と日立とで開発した。



図－1　CP/M－68K開発の経過

2. CP/M－68Kの特長

CP/Mの販売部数は既に100万部を突破，ユーザ数は現在も増え続けCP/M上で動作する言語プロセッサ，ユーティリティ，応用ソフトウェアは2千数百種にのぼる。これらのソフトウェア資産を68000を利用したマイコン装置で利用できるようにすることがCP/M－68K開発の狙いである。

＜CP/M－68Kの特長＞

(1) 既存CP/Mとの互換性

CP/M2.2やCP/M－86　1.1に対して上位互換性あり。

・ディスクファイル形式

・コマンドの指定方法
・OSの持つシステム機能(マイクロプロセッサ依存項目を除く)

(2)既存応用ソフトウェアの活用が可能

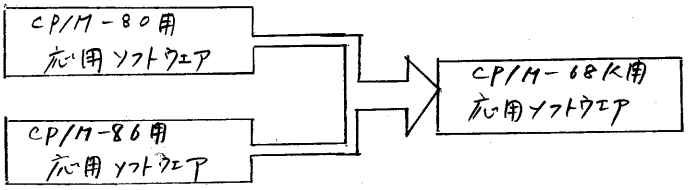


図-2 応用ソフトウェアの活用

(3)68000 アーキテクチャを活用
・任意のアドレスにプログラムをロード可能(16MBのアドレスへ直接アクセス可)
・例外処理機能の強化
・ユーザプログラムをスーパバイザモードで動作可能

(4)各種ハードへの搭載が容易

CP/M-68Kはハードウェアに依存する部分(BIOS)と依存しない部分とに分けることができる。BIOSにはシステム固有のキーボード,CRT,ディスク・ドライブ,プリンタや他のI/Oボードをコントロールするルーチンを含み,各々のハードウェアによりコントロールする方法が異なります。したがってこのBIOSさえ目的のハードに合わせれば,どのようなハードウェアでもCP/M-68Kを利用できる。

(5)C言語コンパイラをサポート

C言語コンパイラをトランジェント・コマンドの一つとしてサポート。C言語が標準仕様として持っているシステム入出力機能を実行時ライブラリとして用意している。またUNIX系OSの標準言語CがCP/M-68Kで利用できるのでUNIX応用ソフトウェアの橋渡しができる。

3、CP/M-68Kの構成

CP/M-68Kはディケットに格納されている一連のプログラムBIOS(Basic Input Output System),BDOS(Basic Disk Operating System)ccp(Command Control processor)で構成されている。OS本体内には7種類の組み込みコマンドと16種類のトランジェント・コマンドを含んでいる。

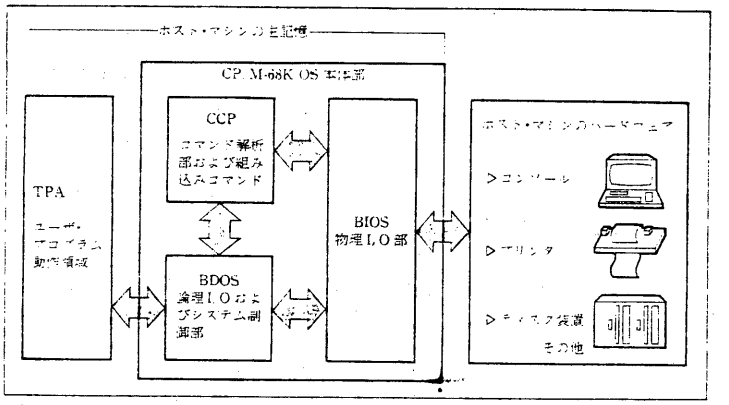


図3 cp/M-68Kの構成

4. cp/M-68Kの機能

4.1 cp/M-68K仕様

cp/M-68Kの仕様を表-1に示す。

表-1 cp/M-68Kの仕様

項目	仕様	
ハード構成	最小	(1) 68000 CPU (2) 128KB RAM (3) フロピィーディスク装置2台 (4) CRTディスプレイ1台
	拡張	(1) 16MBまで主記憶拡張 (2) プリンタ装置1台 (3) 補助入力装置1台 (4) 補助出力装置1台
ディスク装置(最大)		(1) 16ドライブ(A～P)まで (2) 1ファイルの大きさ 32MB まで (3) 1ドライブの大きさ 512MB まで
OSの機能		(1) シングルユーザ・シングルタスク (2) BDOS機能 46種類 (3) BIOS機能 22種類
コマンド		(1) 組み込みコマンド7種類 (2) トランジェント・コマンド 16種類
OSの大きさ		(1) 24KB (BDOS+CCP)+BIOSの大きさ
互換性		CP/M 2.2及び CP/M-86 1.1 に対して互換性有り (1) 8インチ片面単密度フロピィーディスケット形式 (2) BDOS, BIOS機能(但しハード依存項目を除く) (3) コマンドの指定方法

4.2 組み込みコマンド

ユーザが常時使用するコマンドを，ＯＳ本体内の一部として主記憶上に常駐しているコマンドをビルトイン・コマンドとして７種類用意している（表－２）。

表 －２ 組み込みコマンド

種類	機　能
DIR	ディスクのディレクトリからDIR属性を持つファイル名のリストを表示する
DIRS	ディスクのディレクトリからSYS属性を持つファイル名のリストを表示する
ERA	ファイルをディレクトリから消去する
REN	ファイル名を付け替える
TYPE	ファイルの内容をCRT画面上に表示する
USER	システムに登録するユーザ番号を指定する
SUBMIT	サブミット・ファイル中のコマンド・ラインを順次取り出して実行する

4.3 トランジェント・コマンド

ディスクから主記憶上へロードして実行するコマンド・プログラムをトランジェント・コマンドとして１６種類用意している（表－３）。ＰＩＰ，ＥＤ，ＳＴＡＴ，ＤＤＴなどＣＰ／Ｍユーザによく知られているコマンドの他に，６８０００アセンブラ（ＡＳ６８），リンケージ・エディタ（ＬＯ６８）等がある。

表３ トランジェント・コマンド

種類	機　能
STAT	ファイルの使用状況表示とI/Oステータスの管理を行う
ED	ソース・プログラムの作成，修正，編集を行うテキスト・エディタ
PIP	周辺装置間でファイルの転送を行う
DUMP	ファイル内容を16進およびASCII文字を用いて表示する
DDT	コマンド・ファイルのデバグを行う
AS68*	アセンブラ・ソース・プログラムをアセンブルしてオブジェクト・プログラムを作成する
LO68*	オブジェクト・プログラムを結合してコマンド・ファイルを作成する
NM68*	オブジェクト・ファイル，コマンド・ファイル中のシンボル・テーブルの内容を表示する
SENDC68*	絶対番地形式のコマンド・ファイルをSタイプ・レコードに変換する
SIZE68*	コマンド・ファイルの大きさを表示する
RELOC*	リロケーション情報を含むコマンド・ファイルを絶対番地形式のコマンド・ファイルとする
AR68*	オブジェクト・プログラムを結合してライブラリ・ファイルを作成する
FORMAT	ディスクのイニシャライズを行う
INIT*	ディスク中にあるファイルをすべて消去する
COPY	ディスクの内容を別のディスクへバックアップする
C(CP68*/C068/C168)	C言語コンパイラ

注）*の付いたコマンドはCP/M-86の標準コマンドに対して追加・変更されたコマンド

4.4 ＢＤＯＳ機能

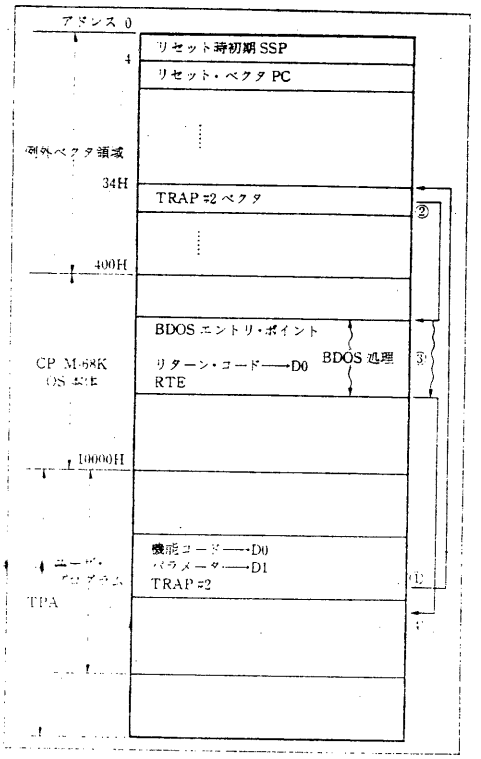
ＯＳ本体部にＴＰＡ（Transient program Area）内で実行されるユーザプログラムに対して４６種類のシステム・コール（BDOS）機能を用意している。システム・コールは①ディスクファイルの入出力機能②ディスク・ドライブ操作機能③非ディスク装置の入出力機能④実行制御機能に大別できる。ユーザプログラムからこのＢＤＯＳ機能を呼び出すためには次の手順でプログラムしなければならない。

- 機能コード番号をデータレジスタ０（D0）に格納
- 各機能コードに対するパラメータをデータレジスタ１（D1）に格納
- Trap#2命令を実行

このようにＢＤＯＳ機能の呼び出しには，６８０００の例外処理命令Trapを使用する。６８０００は０番地から（３ＦＦＦ）Ｈまでの各々４バイトには２５５個の例外ベクタを持つ。各ベクタの値は，例外処理手続きプログラムの先頭アドレスを指す。例外事象が発生するとプロセッサがベクタ値をプログラム・カウンタに設定し，プロセッサのモードをスーパバイザ・モードへ切り替えて実行を開始する。
図－４に示すように，ＢＤＯＳ機能の呼び出しもTrap#2に対応する例外ベクタがＢＤＯＳプログラムのエントリ・ポイントを指し，Trap命令を実行すると，スーパバイザ・モードに移行した後ただちにＢＤＯＳ処理プログラム

へ制御が移る。BDOSの処理が終了すると，その処理結果をデータレジスタ0(D0)に設定，ユーザ・モードに移行後，Trap#2命令の次の命令からユーザプログラムの実行を再開する。68000のアーキテクチャ用に新設したシステム・コール，例外ベクタ宣言(機能コード61)，スーパバイザ・モードへの移行(62番)，TPA範囲の参照／設定(63番)について述べる。

図4 BDOSの呼び出し手順



(1) 例外ベクタの設定機能

68000例外ベクタ値の設定をBDOSを経由して行う機能で，本機能で設定できる例外ベクタを表-4に示す。

本機能呼び出し時には，D0に機能コード61，D1に例外パラメータブロック(図5)のアドレスを格納し，Trap#2を実行する。

本機能が使用されるとユーザが指定したベクタ値をBDOS内の管理テーブルに記録するのみで，主記憶装置上のベクタ領域に直接ベクタ値を書き込まず，実際のベクタ値はBDOSの例外処理手続きの先頭を指している。したがって例外ベクタを設定したベクタ番号に対応する例外事象が発生するとBDOSの例外処理手続きの実行を開始する。

表-4 例外ベクタの設定できるベクター一覧

ベクタ番号	例外事象
2	Bus Error
3	Address Error
4	Illegal Instruction
5	Zero Divide
6	CHK Instruction
7	TRAPV Instruction
8	Privilege Violation
9	Trace
10	Line 1010 Emulator
11	Line 1111 Emulator
32	Trap 0
33	Trap 1
36	Trap 4
37	Trap 5
38	Trap 6
39	Trap 7

ベクタ番号
新たに指定するベクタ値
BDOSによって返される旧ベクタ値

図-5 例外パラメータブロック(EPB)

この処理手続きでは，実行モードをスーパバイザ・モードからユーザプログラムのモードへ切り換え，ユーザが宣言した例外処理手続きへJMPする。このためユーザの例外処理手続きの最後で，例外事象が発生した次の命令に復帰するにはリターン&リストア命令（RTR）を使用するのが好ましい。

(2) TPA範囲の参照／設定機能

ユーザプログラムでTPA範囲の参照または，設定を行うことができる。本機能呼び出し時には，DCに機能コード63をトランジェント・プログラム・エリア・ブロック(TPAB)（図-7）のアドレスをDEに設定し，Trap#2を実行する。この機能は，次に説明するRSX機能とともに使用するのが代表的例である。

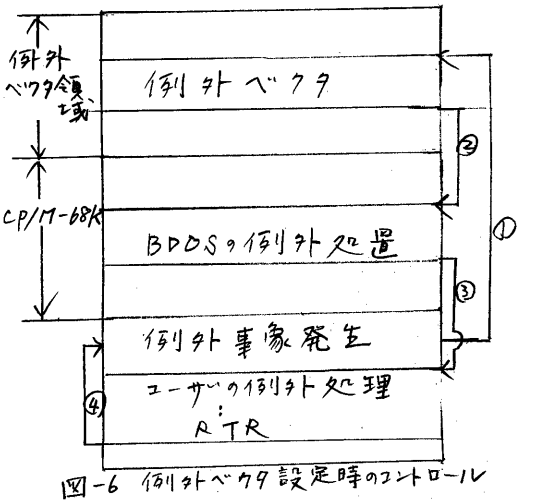


図-6 例外ベクタ設定時のコントロール

(3) RSX (Resident System extension) 機能

OS本体部とは別に，メモリ上に常駐するコマンド・プログラムを作成できる。このコマンド・プログラムをRSXと呼ぶ。RSXはTPA内にロードされるがユーザプログラムからはOS本体部の一部のように扱われる。既存CP/Mでは，他のI/Oを接続する場合やOSの機能拡張を行う場合，付加機能をBIOSに入れるかまたは新しいcp/mを生成しユーザに提供していた。その結果OS本体部が肥大化したり，付加機能を追加するたびにOS本体部を再生成しなければならないといった煩雑さがあった。RSXを用いればOS本体部に手を加えることなく，ユーザの要求に合わせたシステム機能の拡張を行うことができる。RSXとなるプログラムはトランジェント・プログラムと同様，ディスクからTPAへロードされ実行されるが，トランジェントプログラムとは異なりプログラム実行終了後も主記憶装置上に常駐する。

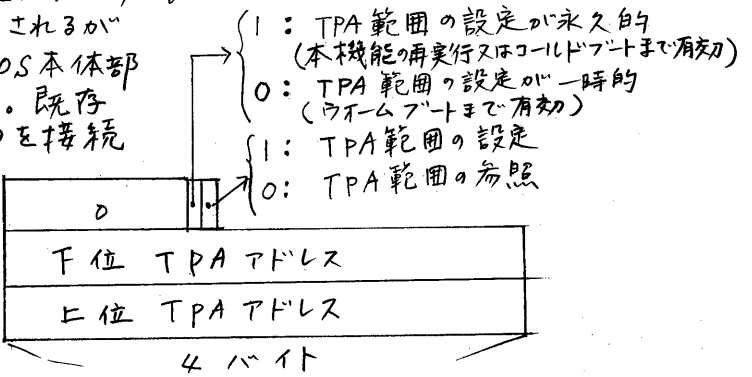


図-7 TPAB の形式

＜RSXの利用手順＞

(i) RSXプログラムの作成

通常のコマンド・プログラムを作成するのと同様，要求される機能を

実行するプログラムをRSXプログラムとして作成。このプログラムの先頭には，RSXプログラムがTrap #0 又は 1で呼び出された時に行う処理を付け加える。

(ii) RSX設定プログラムの作成

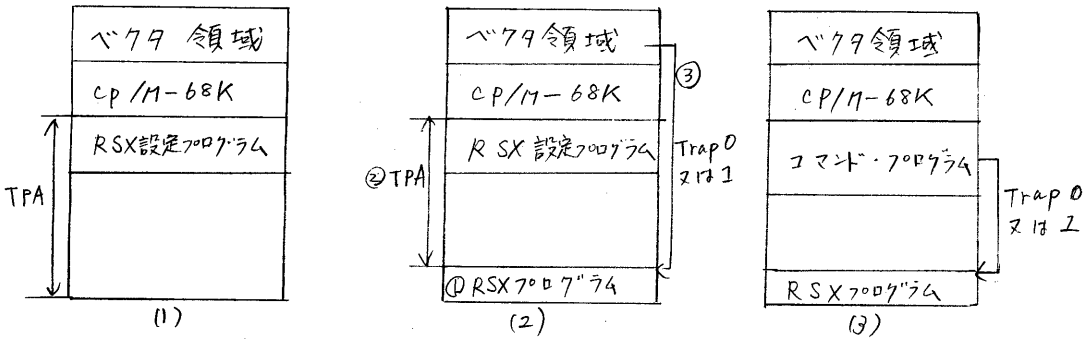
このプログラムはまずRSXプログラムをメモリ上にロード（BDOS機能コード59を使用）して，現在のTPAからRSXプログラム領域を除いてTPA領域として設定（BDOS機能コード63を使用）する。その後例外ベクタのTrap #0 又はTrap#1にRSXプログラムの先頭アドレスを設定（BDOS機能コード61を使用）する。

(iii) RSX設定プログラムの実行

CCPによりRSX設定プログラムをメモリへロードして実行を開始する。

(iv) RSX機能の利用

BDOS機能の呼び出しと同様，RSXで指定された方法でパラメータを設定し，Trap#0又はTrap#1 命令を実行する。



(1) RSXプログラムとRSX設定プログラムをコマンド・ファイルに作成後，CCPにより RSX 設定プログラムをメモリへロードして実行開始する

(2) RSX設定プログラムは①RSXプログラムのメモリへのロード②TPA範囲の設定 ③Trapベクタの設定を行う

(3) RSX設定プログラム完了後，コマンドプログラムはTrap（0又は1）命令により RSX 機能を利用できる。

図-8 RSX 機能

(4) ユーザプログラムをスーパーバイザモードで実行する方法

68000ユイクロ プロセッサは，実行モードとしてスーパバイザモードとユーザモードの2種類のモードをサポートしている。スーパバイザモードはOSなどの管理プログラム実行時に割り当てられ，その下で走るユーザプログラムはユーザモードで動作するように設定する場合が多い。スーパバイザモードではユーザモードを下位概念に置くため，いくつかの特権命令の実行が許されている。またスタックポインタも各々の実行モードに対応してスーパバイザス

スタックポインタとユーザスタックポインタの2つが用意されている。ユーザモードではユーザスタックポインタのみを，スーパバイザモードでは両方のスタックポインタを操作することができる。cp/M-68Kでも，OSはスーパバイザモード，ユーザプログラムはユーザモードで動作する。ユーザプログラムからBDOSやBIOS機能をTrap命令で呼び出すと，スーパバイザモードで入出力動作が行われる。

cp/M-68Kでは，BDOSの機能（機能コード62番）としてスーパバイザモードへの移行機能をもうけて，ユーザプログラムがスーパバイザモードで動作することを可能にしている。これはスーパバイザモードを必要とする特殊な応用プログラムや新しいOSを開発できるようにするものである。この時ユーザプログラム側で注意しなければならないのは，スタックポインタの値とスタックエリアの内容である。スーパバイザモードへ移行するとスタックはスーパバイザスタックに切り換りOSの使用するスタックエリアと共有することになる。このためスタックエリアの内容破壊やスタックポインタの変更を行うとシステムが暴走する場合がある。そこでユーザプログラムは，自分のプログラムエリア内にスーパバイザスタック用のエリアを確保しておき，スーパバイザモードへ移行後ただちにスタックポインタを切り換えることが望ましい。ユーザプログラムがOSの機能を使用する時には，OSの管理するスーパバイザスタックへ再び切り換える。スーパバイザモードからユーザモードへ戻るには特権命令"MOVE SR"を用いる。

```
MOVE.W   #63, D0
TRAP     #2          ] スーパバイザモードへの移行

MOVE.L   A7, OSSTACK
LEA      BOTTOM, A7  ] OSのスーパバイザスタックを退避，
                       スーパバイザスタックを新しく設定。

┌──────────┐
│ 処理プログラム │
└──────────┘

MOVE.L   A7, MYSTACK
MOVEA.L  OSSTACK, A7 ] BDOS機能を使用するため，OSの
                       スーパバイザスタックへ切り換える。

┌──────────┐
│ BDOS機能呼び出し │
└──────────┘

MOVE.L   A7, OSSTACK
MOVEA.L  MYSTACK, A7 ] 再びユーザのスーパバイザスタックへ
                       戻す。

┌──────────┐
│ 処理プログラム │
└──────────┘

MOVEA.L  OSSTACK, A7
MOVE     SR, D0
AND.W    #DFFF, D0   ] OSのスーパバイザスタックへ切り換え，
MOVE     D0, SR      ] ユーザモードへ戻す。
```

図-9 ユーザプログラムをスーパバイザモードで動かす例

5．ＢＩＯＳとその作成方法

ＣＰ／Ｍ－６８Ｋを各種のハードウェアへ容易に移植できるのは、その構成がハードウェア環境に依存する部分（ＢＩＯＳ）と依存しない部分とに分離され、その間を単純なインタフェースで結合しているからである。ＣＰ／Ｍ－６８Ｋをユーザシステムに搭載するには、ＢＩＯＳ部分をそのハードに適用するよう変更すればよい。

5．1　ＢＩＯＳの機能

ＢＩＯＳの機能は２２種類のシステムコールからなり、①システムコントロール機能②非ディスク入出力機能（１文字入出力機能）③ディスクに対する入出力機能に大別できる。ＢＩＯＳの機能呼び出しは次の順で行う。

(i)　機能コード番号をデータレジスタ０（Ｄ０）に設定。

(ii)　各機能コードに対応するパラメータをデータレジスタ１（Ｄ１）とデータレジスタ（Ｄ２）に設定。

(iii)　Ｔrap＃３を実行

機能コード０～２０までのＢＩＯＳ仕様は既存ＣＰ／Ｍと同一であり、機能コード２１（ｆlush buffers）と機能コード２２（set Exception Vector）の２つの機能が付加された。

flush buffer …ＢＩＯＳの持つディスクの読み書きバッファの内容を実際のディスクに書き込み、バッファの内容と実ディスクの内容とを一致させる。

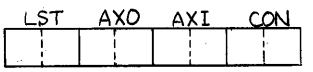
Set Exception Vector…例外ベクタ領域に例外ベクタ値を設定する。

２２種類のＢＩＯＳ機能仕様と実現方式を決定後、各ＢＩＯＳ機能を１モジュールとした手続きを作成。ＣＰ／Ｍ－６８Ｋでは既存のＣＰ／Ｍのように、各機能コードごとのジャンプテーブルを用意する必要はない。ＢＩＯＳのソースプログラムをアセンブル又はコンパイルして作成できたオブジェクトプログラムは、リンケージエディタ（Ｌ０６８）を使用して、ＢＤＯＳとＣＣＰからなるライブラリファイル（ＣＰＭＬＩＢ）と結合する。その結果新しいＣＰ／Ｍ－６８Ｋシステム（ＣＰＭ・ＳＹＳ）を作成することができる。

5．2　非ディスク入出力機能

ディスクに関する入出力機能の他に、４個の論理入出力装置に対して１文字単位でのアクセス機能を持つ。この４個の論理入出力装置は「ＣＯＮ」，「ＬＳＴ」，「ＡＸ０」，「ＡＸＩ」と名付けられ、物理入出力装置との結び付けはＢＩＯＳが行ない、各論理装置１個について、最大４種類の物理装置を割り当て使用できる。論理装置と物理装置との結合状態を示す変数をＩＯＢＹＴＥ

と呼び、ＢＩＯＳ内の１バイト変数で、各論理装置２ビットずつの４つのフィールドに分けて使用する。この２ビットのフィールドの持つ値（０～３）によって物理装置との対応付けを示す。ＩＯＢＹＴＥの値はＢＤＯＳ機能（機能コード７，８）またはＢＩＯＳ機能（機能コード19，20）を使用して、プログラム実行時に動的に参照／設定することができる。



IOBYTEの構成

非ディスク入出力機能はＩＯＢＹＴＥの値によって、物理装置を決め、各々に対する手続きを用意することで実現する。ＢＩＯＳ機能コード５（ＬＳＴ装置への出力）のプログラム例を図－10に示す。

```
        case （LST in IOBYTE）    /＊ LSTのIOBYTE＊/
0：IOBYEの値0に対するリスト出力手続き；/＊シリアルプリンタ出力＊/
1：IOBYEの値1に対するリスト出力手続き；/＊パラレルプリンタ出力＊/
2：IOBYEの値2に対するリスト出力手続き；/＊タイピュータ出力＊/
3：IOBYEの値3に対するリスト出力手続き；/＊パンチャ出力＊/
    end  case;
```

図-10　非ディスク入出力プログラム例

５．３　ディスク入出力機能

ディスク入出力処理概要を図－11に示す。

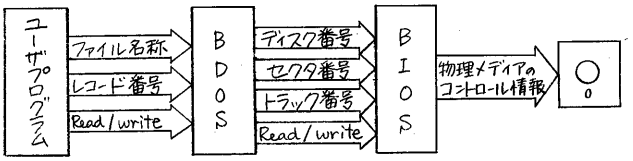


図-11 ディスク入出力処理概要

ユーザプログラムからのディスク入出力要求を①ディスク番号②セクタ番号③トラック番号に変換して、ＢＩＯＳの機能を呼び出す。

サポートするディスクの形式はＢＩＯＳプログラム内のＤＰＨ（Disk Prameter Hedar）に記述され、ディスクメディアの属性はＤＰＢ（Disk Parameter Block）に記述される。これらＤＰＨとＤＰＢのエントリを決定し、サポートするディスクの形式を決めるのはディスク容量と下記５項目である。

(i)　セクタ間スキュを設定するか否か。設定する場合にはその値をいくつにするか。

(ii)　１トラックを何セクタとするか。

(iii)　１ブロックを何セクタとするかを８，６，32，64，128の内から選ぶ。ブロックとはファイルの割り付け単位である。

(iv) ディレクトリエントリ（1エントリの大きさは32バイト）を何個設定するか。

(v) ディスクの先頭のスキップトラック数（通常はコールドローダの格納部分）を何トラックとするか。

表-5　DPHの構成

フィールド名称	フィールド位置	意味
XLT	0~3	セクタ番号の論理・物理変換テーブルのアドレス
—	4~9	BDOS内で使用
DIRBUF	10~13	ディレクトリ操作用作業アドレス
DPB	14~17	DPBのアドレス
CSV	18~21	ディスク交換時，ディレクトリをチェックするための作業領域アドレス
ALV	22~25	ディスク中のファイルのアロケーション情報を格納するための作業領域アドレス

表-6　DPBの構成

フィールド名称	フィールド位置	意味
SPT	0~1	1トラック当りのセクタ数
BSH	2~2	ブロックサイズを表わすシフト数 $2^n \times 128$（バイト）がブロックサイズ
BLM	3~3	ブロックマスク （ブロックサイズ−1）
EXM	4~4	拡張ブロックマスク （ブロックサイズ/1024−1）
DSM	5~6	ディスクの総ブロック数 （ディスク容量/ブロックサイズ−1）
DRM	7~8	ディレクトリ最大エントリ数−1
—	9~10	BIOS内で使用
CKS	11~12	ディレクトリのチェックベクトルサイズ （(DRM+1)/4）
OFF	13~14	ディレクトリの最初にあるシステム用トラック数

6. あとがき

CP/Mファミリーの1つとして68000用CP/Mを開発することにより、①68000応用装置のOSの開発が容易に行えること②既存応用ソフトウェアの活用が可能なこと③応用ソフトウェアの開発が容易に行える等の効果が期待できる。一方、今後のマイクロコンピュータの利用形態は、OA，ビジネス分野，インテリジェント端末，グラフィック処理など、より高度な機能と使い勝手の良いシステムへの要求が高まっている。また、機器組み込み型の制御装置，プロセス制御などにも利用できるリアルタイム機能を有するOSのニーズも高まっている。これらの要求に答えるべくCP/Mファミリーの思想を受け継いだ次世代OSの開発も進んでいる。

— 以上 —