

第5章 ユーザーコマンドと他社ボードについて

5.1 ユーザーコマンド

ADVFSには10個のユーザ関数が用意されており、マシン語で作成されたモジュールをRAM上に展開しインタプリタから直接実行することができます。これに関するコマンドはUSERCOM0～9とSETCOMです。ユーザーコマンドはMPCINIT後、USERCOM0～9というコマンド名で設定されています。これをCOMSETでコマンド名、実行アドレスを与えることによりインタプリタよりユーザで決定したコマンド名で実行できるようになります。ユーザコマンドに対する制限は使用できるレジスタがD0～D3及びA0～A3に限られていることです。これはユーザコマンド実行中にもマルチタスクモニタがタスク制御を実施しており、モニタが保証するレジスタがこの8本に限られているためです。また、マシン語モジュールに引き渡されるデータは4つに限られています。そして、引き数は最初から順にD0～D3にセットされ引き数の数はA0にセットされています。次に操作手順に基づいてコマンドの仕様を概説します。まず、COMSETですが次のように引き数無しで実行すると現在の一覧が表示されます。初期状態ではこのようにエントリが0となっています。このままでUSERCOMを実行するとコマンド番号と引き数の数を表示します。

```
#comset
    USERCOM0 --> USERCOM0   Entry Address->&H00000000
    USERCOM1 --> USERCOM1   Entry Address->&H00000000
    USERCOM2 --> USERCOM2   Entry Address->&H00000000
    USERCOM3 --> AFFIN      Entry Address->&H0002BCF2
    USERCOM4 --> GETDG      Entry Address->&H0002BCF8
    USERCOM5 --> ATAN2      Entry Address->&H0002BCE6
    USERCOM6 --> TAN        Entry Address->&H0002BCCE
    USERCOM7 --> SIN        Entry Address->&H0002BCD4
    USERCOM8 --> COS        Entry Address->&H0002BCDA
    USERCOM9 --> ATAN      Entry Address->&H0002BCE0

#USERCOM0
0 0
#USERCOM0 1 2
0 2
#SIN 300000 1000 ADR(A)
#PR A
500
#COS 300000 1000 ADR(A)
#PR A
866
#
```

次に実際にエントリアドレスを設定してユーザコマンドの働きを確かめてみます。ADVFSには実験用モジュールが組み込まれておりそのアドレスはコマンドTESTで知ることができます。実験用モジュールは各レジスタの内容をヘキサ表示します。テストコマンドを実行することによって得られたアドレスとコマンド名(ダブルクォートで囲む文字列であること)、ユーザコマンド番号を次の例のように引き数として与えます。結果はCOMSETで確かめることができます。そして実際実行してみると与えた引き数がヘキサで表示されます。

```

#TEST
0002A776 0003158C 0008FE7E 000FCBFE 000FFA9A 000FCBFA
#COMSET 0 "AHO" &H2A776
#COMSET
    USERCOM0 --> AHO      Entry Address->&H0002A776
    USERCOM1 --> USERCOM1 Entry Address->&H00000000
    USERCOM2 --> USERCOM2 Entry Address->&H00000000
    USERCOM3 --> AFFIN    Entry Address->&H0002BCF2
    USERCOM4 --> GETDG    Entry Address->&H0002BCF8
    USERCOM5 --> ATAN2    Entry Address->&H0002BCE6
    USERCOM6 --> TAN      Entry Address->&H0002BCCE
    USERCOM7 --> SIN      Entry Address->&H0002BCD4
    USERCOM8 --> COS      Entry Address->&H0002BCDA
    USERCOM9 --> ATAN     Entry Address->&H0002BCE0
#AHO 100 200 300 400
00000064 000000C8 0000012C 00000190
#

```

このように、ADVFS Cのマシン語インターフェースは4つの引き数をレジスタD 0 ~ D 3 に与えて、COMSETによって定義されたマシン語モジュールのエントリに引き渡します。これによりインタプリタ上の変数、配列変数あるいは関数によって与えられる数値を直接マシン語に引き渡すことができます。もちろんマシン語で追加したい機能の中には数値をインタプリタに戻すものもあります。この場合は引き数中に関数ADR()を使用します。関数ADR()は変数・配列のアドレスを得る関数です。たとえばA / Dコンバータをサポートするマシン語モジュールを組み込み結果を特定の変数に返すものとします。この場合は次のようになります。

```

COMSET 0 "ADIN" &HC0000
10 ADIN ADR(data)
20 PRINT data

```

ここでは変数dataに結果を返して表示します。ADINの内部処理ではD 0 に与えられた値をアドレスレジスタにコピーしておき、得られた結果をこのアドレスレジスタの指定するメモリに書き込みます。ADVFS Cの変数はすべてロング整数ですから、書き込みはmove.1 d0, (a0)の形式となります。三角関数SIN / COS / TAN / ATANをユーザコマンドとして使用することができます。この演算は68000用コンパイラのライブラリコールとなっているため、マルチタスクでは使用できません。各演算にはマルチタスクの停止が含まれていますので、注意して使用してください。引き数は整数しか扱うことができませんので、入力で10000倍の度、第二引き数を出力する数のスケールファクターとしています。

| | | |
|---------------------|----------------------|---------|
| sin A1 A2 adr(A3) | A3=sin(A1/10000)*A2 | 引き数は度です |
| cos A1 A2 adr(A3) | A3=cos(A1/10000)*A2 | 引き数は度です |
| tan A1 A2 adr(A3) | A3=tan(A1/10000)*A2 | 引き数は度です |
| atan A1 A2 adr(A3) | A3=atan(A1/1000)*A2 | 結果は度です。 |
| atan2 A1 A2 adr(A3) | A3=atan(A1/A2)*10000 | 結果は度です。 |

```

SIN 300000 1000 ADR(A)
PR A
500
ATAN 10000 1000 ADR(A)
PR A
45000

```

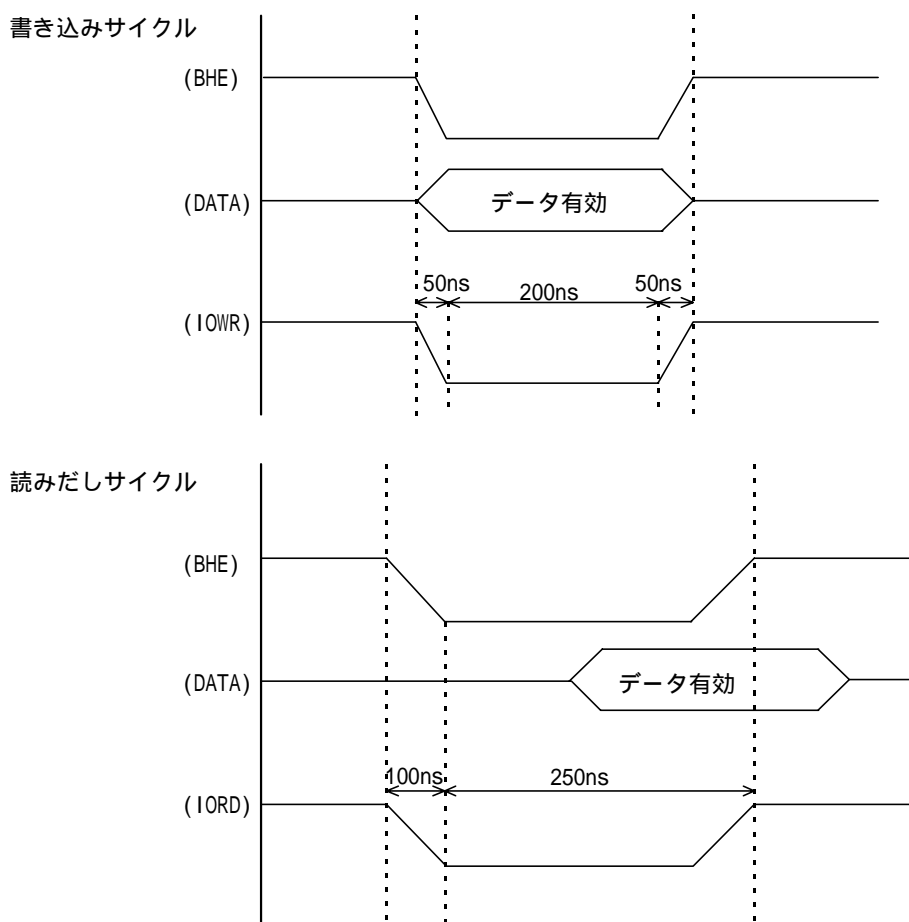
ユーザコマンドは必要に応じて追加されます。

5.2 他社ボードとの組み合わせ使用について

1) ボードの使用に当たって注意すること

- I/Oアクセスに限ること!
- アクセスはワードに限られていること!
- C-MOSバスであること!

MPC-684はバスを98I/Oバスに準拠しており、市販の98用I/Oボードをいくつか使用することができますが、次のような制限を持っています。第一にMPC-684はハード的にワードアクセス、ワード書き込みしかできません。従って同一番地の偶数・奇数番地に配置された異なる8ビットデバイスをアクセスする場合に注意が必要です。マシン語上ではバイトアクセスをしているにも拘らずハード的にはワードアクセスされ、指定バイト以外はダミーアクセスされてしまいます。二番目の注意事項としてはMPC-684のバスバッファはC-MOSを対象としており、TTLもしくはLS-TTL等で構成されたI/Oボードの使用には注意が必要です。TTLでは、C-MOSはただ一つのドライブ能力しかありませんのでほかには何も接続することはできません。MPC-684に使用されているバスバッファのLS-TTLに対するファンアウトは15となっていますが実際の使用環境では5程度と考えた方が無難です。市販の98ボードはLS-TTLで構成されていることが多いのですが、こうしたI/Oインターフェースでは数枚併用しても差し支えない範囲と考えられます。弊社純正のMOP-048、MIP-048ではC-MOS構成のバスとなっていますのでこうした考慮は無用です。



() 付信号は“L”アクティブを示す

2) マシン語にすべきか

ADVFSCではバスに対するコマンドとしてIOR(), WIR(), IOW, WOWを備えています。あまり高速でない使用方法であればマシン語を使用しなくても十分にできるものと思います。次にコンテック製のA/Dコンバータを使用したサンプルプログラムを例示します。A/DコンバータのアドレスはD0に設定されています。このときMPC-684のボードアドレスはE0やC0に変更しておきます。

```

*MAIN_CPU
DO
  IOW &H17 &HD0
  WAIT IOR(&HD1)&&H40==&H40
  data=WIR(&HD0)&&HFFF
  IF IOR(&HD1)&&H10==16 THEN
    data=0-data
  END IF
  TIME 1000
LOOP

```

'A / Dスタート
'A / D終了街ち
'データ取得16ビット
'符号判断

A D V F S C サポートでの応答スピードはインタプリタの実行スピードの制限を受けます。その見積はシングルタスクで一行 250 μ sec として後は通過する行数を乗じたものとなります。この例では A / D スタート後変換終了まで 5 行となりますので約 1.25 msec となります。従って実際の制御では 10 msec 程度応答を期待するものに用途は限られます。もちろんマルチタスクを必要とするシステムでは 10 msec 程度の時間的余裕は不可欠です。マルチタスク有効下では高速サポートを目的としたマシン語化は得策ではありません。マシン語化するのが必要なのは次のような場合と考えられます。第一は用途をシングルタスクに絞り高速サポートする場合、もう一つはデータの取得にインタプリタの手に余るような複雑な演算を必要とするような場合です。前記の例の A / D コンバータでは 16 ビットデータのため演算は複雑ではありませんのでマシン語化はあまり意味はありません。

3) 動作確認ボード

| | |
|-----------|----------------------------------|
| メーカー | アドテック |
| D / A ボード | AB98 - 06BV / 1、 AB98 - 56BV / 1 |
| A / D ボード | AB98 - 30B |