マイクロプログラムアセンブラの一構成法

星　元雄　小島猛志　中林　撰　　林　英也　太田明博
（武蔵野通研）　　（沖電気工業株式会社）

1　まえがき

マイクロプログラム制御は計算機の中央処理装置，周辺制御装置等で広く使用されるようになり，電子交換機の中央処理装置，データチャネル装置への適用も検討されている。そのためマイクロプログラムを効率良く設計作成するためのサポートシステムが必要となった。

こゝで報告するマイクロプログラムアセンブラ（以下ＭＰＡＳＳと称す）は電子交換機のマイクロプログラムへの適用を直接目的として作成した。しかし汎用化をはかったため他の分野にも適用可能である。

2　マイクロプログラムアセンブラの特徴

マイクロプログラムアセンブラはプログラマの書いたニーモニックコードをオブジェクトのビットパターンへ変換するという点では一般のアセンブラと変わりない。マイクロプログラムは一般のアセンブラ言語で書かれたプログラムと比較して以下の点が異なる。

(1)そのマシンコードで作られるプログラム数が少ない。
一般のプログラムでは一つのマシンに対して多くのプログラムが作られるが、マイクロプログラムは一つのマシンに対して通常ハードウェア設計時に作られるだけである。

(2)プログラム規模が小さい。
制御メモリのコスト，実行時間等の制約から、プログラム規模は高々１～２Ｋ語程度と考えられる。

(3)プログラム設計者はハードウェアに熟知している。
一般に装置設計者自身がプログラミングを行なう。

(4)デバッグ手段として論理シミュレータがある。
最近、論理装置の設計には論理シミュレータが使われることが多い。その場合、マイクロプログラムを論理装置の一部とみて、そのデバッグを論理シミュレータで行なうという考え方もある。

(5)パリティ付与，特殊な番地割付方法が考えられる。
マイクロプログラムはＲＯＭへロードする場合が多く、ロードするためのハードウェアにパリティ付加機能が無い場合，アセンブラがパリティを付加する必要がある。マイクロプログラムは特に実行時間短縮の要求が大きいため，ダイナミックステップを減らすために多重分岐が採用されることがあり、番地割付方法が複雑な場合がある。

(6)マイクロプログラムの評価は装置設計の大切な評価の一部となる。
装置設計の評価要因の一つに制御メモリを有効に用いたかどうかがあげられる。

これらの相違点はあくまで相対的なものであり、今後マイクロプログラム技術がエミュレータの分野等更に一般化した場合傾向が異なってくることもあろう。

3　MPASSの設計条件

前項の特徴(1)～(6)を考慮して以下の項目を設計条件とした。

(i)マイクロ命令形式に関する情報を入力情報の一部として汎用化をはかる。

前項(1)，(2)の条件から，マイクロプログラムを適用するマシンごとにそのためのアセンブラを作成するのは得策でない。従ってマイクロ命令語長，フィールド構成，パリティ付加方法等を異にする各種のマシンに共通して利用できる汎用性のあるアセンブラが要求される。そのためには，マイクロ命令形式を定義する情報を入力情報の一部として加える。

(ii)アセンブラの出力媒体はMTとし、制御メモリロードのためのインタフェースは別途用意することを想定する。

これはマシンごとに制御メモリへロードする方法を異にするためである。

(iii)高級なプログラム編集機能を必要としない。

プログラム規模が小さい事から、リンケージエディット，マクロ命令は不要と考えた。但し、ソースモジュールのアップデート機能は必要であるが、これにはOSのユーティリティを利用でき、アセンブラの機能としては必要ない。

(iv)チェック項目はシンタクスチェック，メモリ領域チェックのみとする。

ハードウェアに密着したチェック項目は論理シミュレータの範囲と考える。

(v)パリティの付加

(vi)評価データの収集

制御メモリの利用度合を評価するデータを収集する。

以上をまとめて表3.1に示す。

4　システム構成

システム構成を図4.1に示す。処理の流れを図4.2に示す。

表3.1　一般のアセンブラとMPASSの機能の比較

項目		一般のアセンブラ	MPASS
1	基本機能	ニーモニックコードで記述した命令群をマシンコードへ変換	同左
2	入力情報	プログラム	マイクロ命令形式定義*，マイクロプログラム
3	出力情報	リスト，リロケータブルモジュール	リスト，ロードモジュール*，評価データ*
4	マクロ命令機能	あり	なし*
5	パリティ付加機能	なし	あり*

(注)　＊印は主な相違点

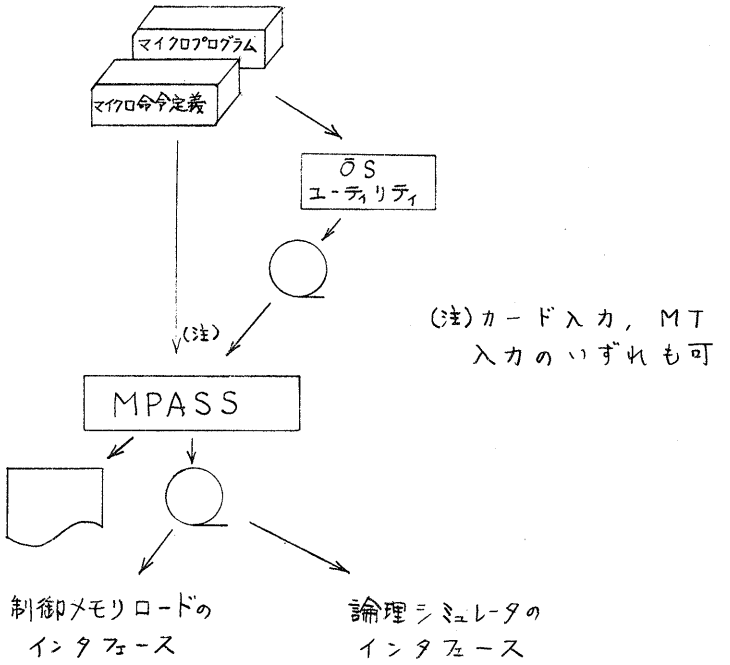


図4.1 システム構成

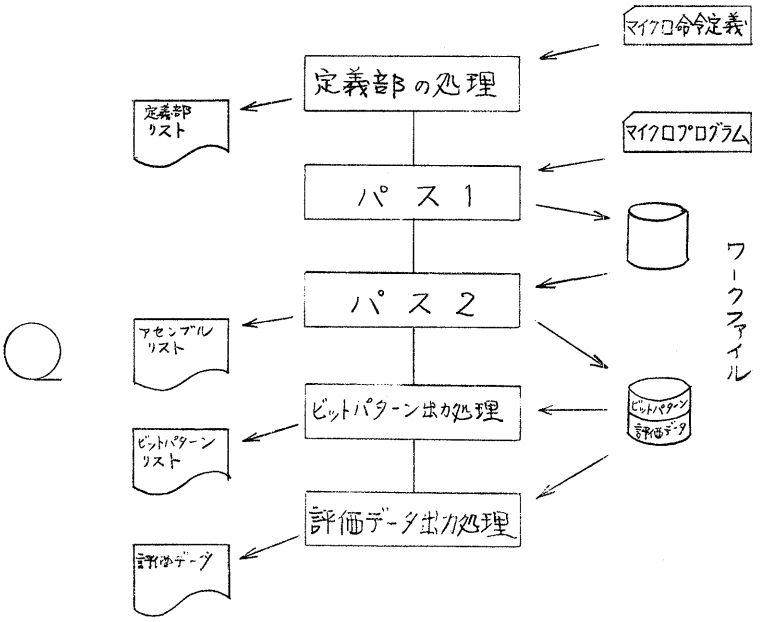


図4.2 処理の流れ

4.1　入力形態

アセンブラへの入力媒体としてカード入力，ＭＴ入力のいずれも可能である。ＭＴファイルの作成およびその更新にはＯＳのユーティリティを用いる。

入力情報はマイクロ命令形式定義部とマイクロプログラム部より成るため入力形態として以下の３通りのバリエーションが可能である。

(a)定義部，プログラム部ともにカード入力
(b)定義部ＭＴ入力，プログラム部カード入力
(c)定義部，プログラム部ともにＭＴ入力

5　マイクロプログラムの記述

5.1　一般則

マイクロ命令仕様の異なる各種の装置に適用できるよう汎用化をはかったため記述形式もフリーフォーマットとした。

(1)フリーフォーマット。
文の終りは；（セミコロン）で示す。１つの文が複数行にまたがってもよい。１つの行に複数の文を書いてはならない（アセンブルリスト出力時、ソースイメージとマシンコードを同一行に印字することから設定した条件）。
レーベルは文の始めに書き，：（コロン）で区切る。
コメントは／＊と＊／でかこむ。

(2)数値はすべて２進数（'で示す），１０進数（無表示），１６進数（''で示す）のいずれでも記述できる。

(3)シンボル名は８文字以内の英数字（＄も可とした）。

本言語はマイクロ命令形式定義部とマイクロプログラム部とからなる。

<プログラム> ::= <マイクロ命令形式定義部> <マイクロプログラム部>

5.2　マイクロ命令形式定義部

マイクロ命令形式定義部では(1)マイクロ命令語長，(2)メモリ領域，(3)パリティ形式，(4)フィールド位置　(5)フィールド内のビットパターンに対するニーモニックコードの定義，(6)マイクロ命令構成の定義を行なう。

<マイクロ命令形式定義部> ::= <ＣＭＤＥＦ文>
[<ＦＩＥＬＤ文>]$_1^n$ (注)
[<ＩＮＳＴＲＵＣＴ文>]$_1^n$
ＤＥＦＥＮＤ；

(注)　記号 [ ]$_i^n$ は [ ] の中の内容を連続的に i 個から n 個まで並べられることを意味する。

〈CMDEF文〉::= CMDEF 〈語長属性〉〔〈メモリ領域属性〉〕
〔〈パリティ属性〉〕$_0^{16}$;

語長属性は〈MSDビット位置番号〉-〈LSDビット位置番号〉で示し、ビット位置呼称が右昇順か左昇順かも定義できる。
メモリ領域属性は〈スタート番地〉-〈エンド番地〉で示し、プログラムが入るべき領域を定義する。
パリティ属性にはデータパリティとアドレスパリティの2種類がある。

〈データパリティ属性〉::= PD(〈パリティ挿入ビット〉)={ODD|
EVEN}(〈フィールド開始ビット〉,
〈フィールド長〉〔+〈フィールド開始ビット〉
,〈フィールド長〉〕$_0^{15}$);

〈アドレスパリティ属性〉::= PA(〈パリティ挿入ビット〉)=
{ODD|EVEN};

データパリティは通常のパリティである。パリティの対象となるフィールドは16個までコンカティネーションできる。アドレスパリティはその語の番地の偶奇によりパリティを決める方式である。

マイクロ命令をフィールドの集まりと見做し、FIELD文により，各フィールドについてそのフィールド名，ビット位置，マイクロオーダ名を定義する。

〈FIELD文〉::= FIELD〈フィールド名〉〈フィールド属性〉
〈マイクロオーダ属性〉
〔DEFAULT(〈数〉)〕〔BRANCH〕;

〈フィールド属性〉::= (〈フィールド開始ビット〉,〈フィールド長〉〔+
〈フィールド開始ビット〉,〈フィールド長〉〕$_0^3$)

〈マイクロオーダ属性〉::= 〔〈マイクロオーダ名〉(〈ビットパターン〉)〕$_0^{4096}$

フィールドに入るべきビットパターンに対応するニーモニックコードをマイクロオーダと呼ぶ。マイクロオーダは必らずしもすべてのビットパターンに対して定義する必要はなく（例えば8ビットのフィールドでは256通りのビットパターンが存在する）、頻繁に現われるビットパターンについてのみ定義し，残りについてはフィールド名と組合せてリテラル表現ができる。デフォルト値の指定（指定がなければall '0'）も可能である。BRANCH指定があると，番地を示すフィールドと見做し，そこに入る値がメモリ領域内にあるかどうかチェックする。

INSTRUCT文はマイクロプログラムのシンタクスチェックのための情報を提供する。

〈INSTRUCT文〉::= INSTRUCT〈タイプ名〉:
〔〈フィールド名〉〕$_0^{256}$,
〔〈マイクロオーダ名〉〕$_0^{4096}$;

タイプ名はその命令形式の呼び名である。2番目の属性はそのマイクロ命令を構成するフィールド群を定義する。フィールド名の1つを( )でくくることができる。これは命令の形式を識別するキーフィールドとなる。3番目の属性は( )でかこまれたフィールド名に属するマイクロオーダの部分集合である。第3属性が省略されると( )でかこまれたフィールド名に属するマイクロオーダのうち、それ以前のINSTRUCT文で使用されなかったものすべてが第3属性にあると見做す。第2属性に( )でくくったものがないとき第3属性は,とともに省略しなければならない。そのような文は1文しか許さない。この命令形式に決まるのは他のいずれの命令形式にも決まらなかった場合である。

5.3 マイクロプログラム部

〈マイクロプログラム部〉::= MP ;
〔{〈LOC文〉|〈EQUAL文〉|
〈マイクロ命令文〉|〈CONST文〉}〕$_0^n$
MPEND ;

〈LOC文〉::= LOC〈レーベル名〉=〈式〉 ;

〈式〉::= 〈項〉〔〈演算子〉〈項〉〕$_0^n$

〈項〉::= 〈数〉|〈マイクロオーダ名〉|〈EQUAL定義名〉|〈レーベル名〉|@

〈演算子〉::= +|−|*|/|&|,|#

表5.1 特殊記号の意味

	意味
@	これが使われているマイクロ命令の絶対アドレスを示す
+	ロジカル加算
−	ロジカル減算
*	左ロジカルシフト
/	右ロジカルシフト
&	論理積
,	論理和
#	排他的論理和

〈ＥＱＵＡＬ文〉::＝ＥＱＵＡＬ〈ＥＱＵＡＬ定義名〉＝〈式〉 ；

〈マイクロ命令文〉::＝〔〈レーベル名〉:〕〔〈フィールド子〉〕$_{0}^{256}$ ；

〈フィールド子〉::＝〈マイクロオーダ名〉|〈リテラル〉

〈リテラル〉::＝〈フィールド名〉（〈式〉）

〈ＣＯＮＳＴ文〉::＝〔〈レーベル名〉:〕ＣＯＮＳＴ{〈2進数〉|〈16進数〉}

マイクロ命令文は定義部のＦＩＥＬＤ文で定義したマイクロオーダ名を羅列したものである。マイクロオーダ名の記述順序に任意でよい。1つのマイクロ命令文中に同じフィールドに属するマイクロオーダ名が2つ以上あってはならない。必要とするビットパターンに対応するマイクロオーダ名が定義されていない場合はリテラル表現ができる。

マイクロ命令文にはレーベルをつけられる。一般にレーベルが参照されるのはジャンプ先を示す場合であり、これはリテラル表現で示される。リテラル表現に式を導入したのはそのためである。すなわち、

(i) ジャンプ先を絶対番地で示す場合：　χ　　（χ：ジャンプ先レーベル）

(ii) ジャンプ先を相対番地で示す場合：　χ－@

(iii) ベースアドレス（セクタ）方式が採用されている場合：

χ＊n　または　χ／n　（χを左または右にnビットシフトする）

等によりジャンプ先を表現でき、マイクロプログラム制御方式において種々考えられる分岐方法に対して融通性がある。

ＥＱＵＡＬ定義文は、定数をＥＱＵＡＬ定義をにより間接的に定義したり，式を一度一つのＥＱＵＡＬ定義名に直して用いたりするのに使用する。

レーベル名を絶対番地と対応づけるにはＬＯＣ文を用いる。アセンブラの処理を簡単化するため，ＬＯＣ文，ＥＱＵＡＬ文の式の値はその文が現われた時点で確定（アセンブル処理の1パス目で確定）していなければならないという制約をもうけた。ＬＯＣ文の値はメモリ領域チェックを行なうが，ＥＱＵＡＬ文の値はチェックしない。

ＣＯＮＳＴ文はマイクロ命令の1語をすべて2進数または16進数表現するためのものである。その中にはパリティビットも含める。従って、パリティバス等正常状態とは異なるビットパターンを作り出すことが可能である。

マイクロプログラムの記述例を図5.1に示す。

5.4　評価データ

マイクロプログラム方式を評価する一基準として，マイクロ命令のフィールド構成，フィールド長が適当であったかどうか評価することとし，そのためのデータを編集出力するようにした。収集データは以下の5項目である。

(1) 使用したメモリ領域。

(2) 各マイクロ命令形式（ＩＮＳＴＲＵＣＴ文に対応）の使用頻度，ならびにその中の各フィールドにおける各マイクロオーダ名の使用頻度。

(3)各マイクロ命令形式ごとに，任意の2つのフィールド（図5.2においてX軸のフィールドとY軸のフィールド）が同時に有効に使われた（デフォルト値でなかった）回数。

(4)BRANCH属性をもつフィールドにおける値の分布。

(5)語長内の各ビット位置における'0'と'1'の出現頻度。

評価データの出力例を図5.2に示す。

(3)でもし2つのフィールドが同時に使われる回数が少なければ、図5.3のようにそのマイクロ命令形式を2つの異ったマイクロ命令形式に分けて語長を短かくできる可能性があると解釈できる。(4)ではジャンプ先を入れるフィールドのフィールド長が適当であったかどうか評価する。

6 今後の課題

本アセンブラはマイクロ命令形式等のハードウェア情報を入力情報の中で定義することによって汎用化を試みた。この方向を押し進めると汎用言語プロセッサ（アセンブラの自動発生）の考え方に近づく。ここでは、そこまで進まないことにより定義部の構成を簡単で分り易いものにできた。しかし、以下の点で更に融通性をもたせることが望まれる。

(1)本アセンブラではシンタクス解析を容易にするためキーパラメータ方式とした。そのためマイクロオーダ名の記述順序は任意でよいが、欠点としてマイクロオーダ名の二重定義を禁止している。その結果、例えばある同じレジスタを指定する場合でも、何個所かの異なったフィールドに出現するときは、それぞれ異なったマイクロオーダ名を定義しなければならない不便さがある。これを改善するには、ポジショナルパラメータの考え方を入れるか、マイクロオーダ名の間の関係に意味をもたせるとかして二重定義を許すことが考えられる。

(2)本アセンブラを汎用アセンブラとして見た場合、既成のアセンブラ言語で書かれた小規模プログラムをアセンブルできれば便利である。そのためには項の区切り、文の区切りを示す識別子の与え方を検討する必要がある。

その他今回割愛したがフローチャートの自動作成も保守用のドキュメントとして有効であろう。

7 あとがき

本報告ではマイクロプログラムアセンブラ（MPAS5）の言語仕様を中心に記述した。本アセンブラの特徴は定義文の導入による汎用化、式、フィールドのコンカティネーション等の導入による記述性の向上，評価データの収集である。

本アセンブラはDIPS-1の上で実行する。電子交換機処理装置の実験機のマイクロプログラム作成に使われており、機能的に要求を十分満たすことを確認した。また汎用アセンブラとして論理シミュレータへ入力する試験プログラムのアセンブルに使うことも検討している。

おわりに、日頃御指導いただく武蔵野通研鵜野室長、沖電気工業上原室長、柴田研究主任に深謝致します。

図 5.1 記述例

```
MICRO-PROGRAM ASSEMBLER  "MPASS010"   DEFINITION PART SOURCE LIST                    DATE 74/12/03 (22:00:43)

STMTNO   123456789 123456789 123456789 123456789 123456789 123456789 123456789   ID SQNO.
    1    /*                                                                  */ CHM01020
                 CMDEF  32-0                                                     CHM01030
                        PD(32)=ODD(0,31)                                        ;CHM01040
    2    /*                                                                  */ CHM01070
                 FIELD  INT(31,1)   NINT     ('0')      DEFAULT  ('0')        CHM01080
                                    YINT     ('1')                           ;CHM01090
    3            FIELD  TYP(29,2)                                             CHM01100
                                    BUS      ('00')                          CHM01110
                                    ME       ('01')                          CHM01120
                                    TJ       ('10')                          CHM01130
                                    ICT      ('11')                         ;CHM01140
    4            FIELD  RCR(28,1)   RCR      ('1')      DEFAULT  ('0')        CHM01150
                                                                            ;CHM01160
    5            FIELD  SCH(24,4)                        DEFAULT  ('0000')     CHM02010
                                    JEND     ('0001')                        CHM02020
                                    JENDH    ('0010')                        CHM02030
                                    JENDE    ('0011')                        CHM02040
                                    SSEL     ('0100')                        CHM02050
                                    RSEL     ('0101')                        CHM02060
                                                                                  定義部
   28    /*                                                                  */ CHM10010
                 INSTRUCT BUS;    INT (TYP) RCR SCH ISA MCHK ADDER DB REG PB   CHM10020
                                BR RB1 RB0 ,BUS                               ;CHM10030
   29            INSTRUCT ME ;    INT (TYP) MA KEY QDR ADDER DB REG PB BR RB1  CHM10040
                            RB0 ;ME                                           ;CHM10050
   30            INSTRUCT TJ ;    INT (TYP) TS TP DB REG EXPV JP JA ;TJ       ;CHM10060
   31            INSTRUCT ICT;    INT (TYP) RCR SCH CST CC FFC CTL LTL ;ICT   ;CHM10070
   32    /*                                                                  */ CHM10080
                 DEFEND                                                        ;CHM10090

MICRO-PROGRAM ASSEMBLER  "MPASS010"   PROGRAM PART SOURCE LIST                   DATE 74/12/03 (22:00:53)

STMTNO   123456789 123456789 123456789 123456789 123456789 123456789 123456789   ID SQNO.    ADR.  MICROCODE
    1            MP                                                           ;CHM10000
    2            LOC    CHWLR="21                                             ;CHM13000
                                                                                  プログラム部
    6            LOC    DTL01="25  ←― DTL01 を 25番地に指定する。             ;CHM13040

   21  DTL01:     ME     DA      RD   MRQ DB1 REG2 REG      ADD WR            ;CHM10720  0025  038043085
   22             TJ     AND2  TP("20)      DB4       M  NXT  JA(DTL03)       ;CHM10730  0026  054408027
   23  DTL03:     BUS    ISA MCK3                        MEM                   ;CHM10740  0027  100E00020
   24             TJ     AND1  TP("08)      DBC       M  NXT  JA(DTS02)       ;CHM10750  0028  052118032
   25  DTL02:     ME     SYS0  KEY WT       DB3            DB                 ;CHM10760  0029  029806010
   26             BUS    ISA MCK1                        MEM                   ;CHM10770  002A  000A00020
   27             ME     SYS1  KEY WT       DB6            DB                 ;CHM10780  002B  12B80C010
   28             BUS    ISA MCK1                        MEM                   ;CHM10790  002C  000A00020
   29         YINT ICT RCR          ISB  RPC1                                 ;CHM10800  002D  0F0066000
    :            :
  195            MPEND                                                        ;CHM12160
```

(1) MICRO-PROGRAM ASSEMBLER "MPASS010" MEMORY MAP LIST

REGION	DON'T CARE.		
USED	0021 - 0032	0012 (18)	WORDS
USED	0037 - 00CA	0094 (148)	WORDS
TOTAL		00A6 (166)	WORDS

(2) MICRO-PROGRAM ASSEMBLER "MPASS010" FIELD USAGE LIST

INSTRUCTION BUS		36 (21.7 %)	
FIELD INT			
	NINT	0 (.0 %)	
	YINT	3 (8.3 %)	
	LITERAL	0 (.0 %)	
	DEFAULT	33 (91.7 %)	
FIELD TYP			
	BUS	36 (100.0 %)	

(2)、(3)は命令フォーマット毎に出力し、(4)は全体について出力する。

(3) SIMULTANEOUS USAGE OF FIELDS

	1	2	3	4	5	6	7	8	9	10	11	12	13
1 INT	3	3	3	5	0	0	1	5	1	1	2	1	1
2 TYP		36	3	4	21	28	5	6	15	5	31	16	14
11 BR											31	11	9
12 RBI												16	14
13 RBO													14

(4)

NOCHK	0 (.0 %)	
CONST	0 (.0 %)	
ERROR	1 (.6 %)	
TOTAL	166 (100.0 %)	

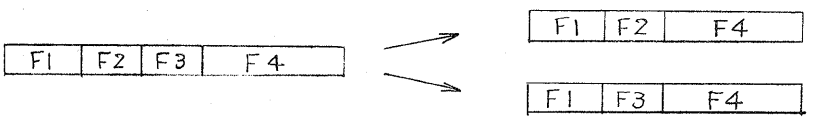
(5) MICRO-PROGRAM ASSEMBLER "MPASS010" BRANCH FIELD BIT USAGE LIST

FIELD TP			
	00000000	2 (3.8 %)	
	00000001	3 (5.8 %)	
	0000001X	1 (1.9 %)	
	000001XX	8 (15.4 %)	
	00001XXX	14 (26.9 %)	
	0001XXXX	10 (19.2 %)	
	001XXXXX	6 (11.5 %)	
	01XXXXXX	5 (9.6 %)	
	1XXXXXXX	3 (5.8 %)	

(6) MICRO-PROGRAM ASSEMBLER "MPASS010" 0/1 USAGE LIST

BIT	0		1	
32	89	(53.6 %)	77	(46.4 %)
31	153	(92.2 %)	13	(7.8 %)
1	132	(79.5 %)	34	(20.5 %)
0	110	(66.3 %)	56	(33.7 %)
TOTAL	4201	(76.7 %)	1277	(23.3 %)

図5.2 評価データ



F2とF3が同時に使われない場合

図5.3 マイクロ命令語長の短縮