

解説

マイクロコンピュータのソフトウェア*

宮口 庄司**

1. はじめに

マイクロプロセッサの応用分野の拡大に伴って、そのソフトウェアは、ますます重要なものとなっている。マイクロプロセッサ応用機器の一例として、「インテリジェント・XYプロッタ（以下 I-XY を略称***）のソフトウェアを例にとり、マイクロコンピュータのソフトウェアの意義を調べてみよう（図-1 参照のこと）。

データセンタで処理された図形情報は 1~数画面分の図形コマンド例**** をまとめた形で伝送され、I-XY のバッファメモリ (RAM) に格納される。ここで回線を切断する。次に、ROM に格納されている図形コマンド解釈プログラム、直線、円弧パターン発生プログラム、および、文字パターン発生プログラムが、バッファメモリ中の図形コマンド列を解釈し、XYプロッタに図形出力を行う。I-XY は「図形コマンドという一種の言語を解釈し図形を発生する」という高度な機能を 4 種のプログラムによって実現している訳である。

マイクロプロセッサのプログラム制御によって、①通信回線使用時間が、従来方式の 1/10 から 1/1000 に短縮、②高精度、低価格サービスの実現、③データセンタの負荷の軽減、が実現している。

I-XY の制御を従来の布線論理で行おうとすると、

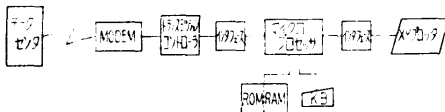


図-1 インテリジェント・XYプロッタ

* Software of Micro Computer by Shoji MIYAGUCHI (N. T. T., Musashino Electrical Communication Laboratory).

** 日本電信電話公社武蔵野電気通信研究所宅内機器研究部

*** I-XY は、マイクロプロセッサ制御の画像端末の基本的一例として、日本電信電話公社武蔵野電気通信研究所で開発されたものである。

**** 圧縮形図形情報

非常に大きな装置を作る必要がある。また、ミニコンピュータを用いるのは贅沢であり、余計な部分が多くなってしまふ。

マイクロプロセッサ（以下 μP と略称）応用機器のアプリケーションプログラム（以下 μP プログラムと略称）は、このような制御機能のほか、数値計算、データ処理などを受持つことができる。また μP 応用機器の仕様変更にも容易に対処できるので、設計に柔軟性が生れる。

μP プログラム作成のための支援ソフトウェアシステムとしては、アセンブラ、高級言語などがある。ミニコンピュータ支援ソフトウェアシステムの場合は、ミニコンピュータ OS が存在するが、 μP 支援ソフトウェアシステムとしては、OS 体系ともいふべきものが整っていない。なお、 μP プログラムの開発は、マイクロコンピュータ以外の異種コンピュータを使用するクロスソフトウェアの概念が、ミニコンピュータと比較し、より重要である。

2. μP プログラムの作成2.1 μP 応用機器の設計・製造

μP 応用機器の代表的な設計・製造手順を図-2（次頁参照）に示す。

図-2: ⑥のソフトウェア作成とは、 μP プログラムの作成を行うものであり、図-2: ③の装置固有のマイクロプログラムの作成とは、ユーザマイクロプログラミングを行うことを示す。このようにプログラムの作成を行うことが布線論理による装置設計とは大きく異なる。

2.2 μP プログラムの設計・製造

μP プログラムの設計・製造法で標準的な方法はまだ明確になっていない。これは μP 出現後日が浅いこともあるが、フローチャート作成、単体デバグ、結合デバグという従来のプログラム製造法、あるいは SP（ストラクチャード・プログラミング）手法によるプ

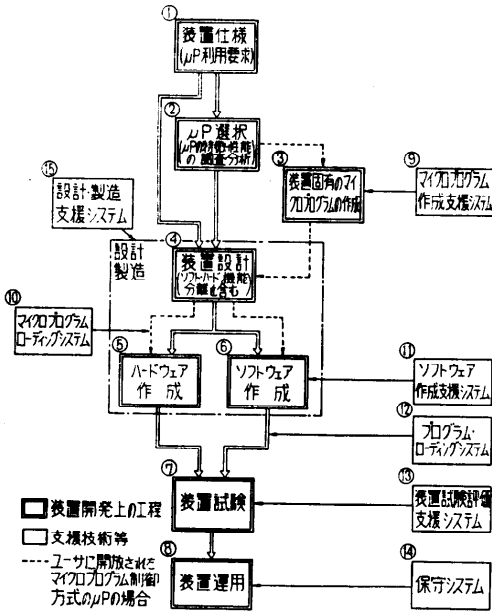


図-2 μP 応用機器の設計製造手順

プログラム製造法のいずれにせよ、μP プログラムの大きさが10 キロステップ以下のものが多く、また設計・製造法が μP 応用機器毎にかなり異なる理由によると考えられる。上手な μP プログラムの作り方として、チップメーカ、システムハウスの提供するサブルーチンを使う方法がある。四則演算、ビットシフト、高度なものでは、平方根計算、対数計算などの演算を、サブルーチンとして利用できればそれだけ μP プログラムの作成工数が少なくできる。筆者の例では、サブルーチン利用により 20% のプログラム作成が節約できた。2 回以上サブルーチンを利用する場合はメモリ節約にもかなり有効である。

μP 応用機器設計において、マイクロコンピュータを利用し、これに入出力装置などの周辺機器を付加して組立てる方法がある。この場合は応用機器特有のプログラムのみを作成し、マイクロコンピュータ・モニタと結合して μP プログラムを完成する。

モニタが利用者の要求にかなう機能、柔軟性をもっていればこの方法は有効な手法である。16 ビットマイクロコンピュータモニタのなかには、マイクロ O S⁽⁷⁾ と称してタスク管理のマクロを 20 種程度用意し、処理機能ごとにモジュール化したリロケータブルな目的プログラム (OM) を紙テープとして提供する例があるが、これら目的プログラムを利用できれば μP プログラム作成は比較的容易になろう。

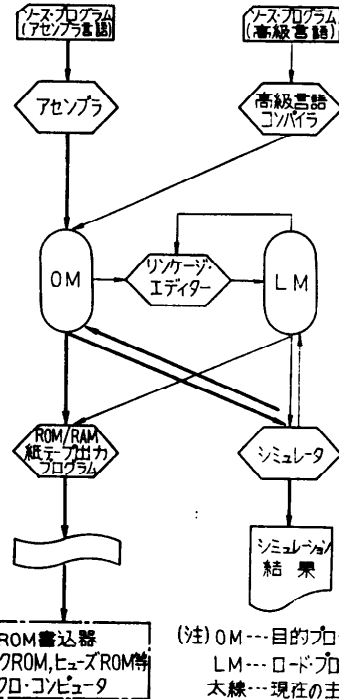


図-3 ソフトウェア作成支援システムとプログラムローディングシステム (基本部)

2.3 支援ソフトウェア

μP 応用機器の設計製造のためのソフトウェア支援システムを表-1(次頁参照)に示す。このうち基本的な支援システムは、ソフトウェア作成支援システム(図-2: ⑩参照)と、プログラム・ローディングシステム(図-2: ⑫参照)であり、この基本部分を図-3に示す⁵⁾。

アセンブラ言語で書かれたソースプログラムは、アセンブラによって目的プログラム (OM) に翻訳後、シミュレータによって目的プログラムの論理誤りを発見する (プログラム・デバグ)。

ROM/RAM 紙テープ出力プログラムは、目的プログラムを PROM 書込器などにより ROM に設定したり、マイクロコンピュータに入力するための紙テープを出力する。μP プログラムが大きくなる場合は、リンケージエディタによって目的プログラムを結合する。高級言語コンパイラとしては、PL/M がよく知られている。

ソースプログラム、目的プログラム、ロードプログラム (LM) は、紙テープあるいは MT に格納して保存、更新の対象とするが、これを受持つ支援システムについては後述する。

表-1 ソフトウェア支援システム

対応番号	名 称	内 容	備 考
⑨	マイクロプログラム作成支援システム	(1) マイクロプログラムアセンブラ (2) マイクロプログラムシミュレータ	
⑩	マイクロプログラムローディングシステム	(1) CROM 紙テープ出力プログラム (2) ロータ	CROM: 制御メモリ
⑪	ソフトウェア作成支援システム	(1) アセンブラ (2) シミュレータ (3) 高級言語 (4) ユーティリティ・プログラム (a) デバッガ (b) エディタ (c) リンケージ・エディタ (d) オブジェクト・チェッカー (5) μ P プログラム部品 (a) サブルーチン (b) システム・マクロ (c) マイクロ・コンピュータ・モニタ	1) シミュレータは、クロスソフトウェアのみ 2) マイクロ・コンピュータ・モニタとデバッガはレジデントソフトウェアのみ
⑫	プログラムローディングシステム	(1) ROM/RAM 紙テープ出力プログラム (2) ロータ	(1) P-ROM 書き器 マスク ROM, ヒューズ ROM 等のビットパターン設定用インタフェース (2) ロータ: マイクロ・コンピュータ入力プログラム・データ読み込み
⑬	装置試験・評価支援システム	(1) 検査データ作成プログラム (2) μ P プログラム・評価プログラム (3) 試験プログラム	
⑭	保守システム	(1) プログラム・データ保存・更新プログラム (2) ドキュメント作成プログラム (a) 逆アセンブラ (b) フローチャート・プログラム (3) 保守診断プログラム	
⑮	設計・製造支援システム	(1) μ P ボード設計システム CAD (2) ブロック・シミュレータ	

3. 支援ソフトウェア各論

3.1 アセンブラ

アセンブラはチップメーカーやシステムハウスなどによって様々な形態で提供され流通している他、チップユーザによっても数多く自作されている。それぞれ μ P 機種 of 多様さやアセンブラプログラムの走行形態などによっていくつかの特徴をもっている。

4ビット μ P の場合は、プログラム手続部分のアドレス表現(8ビット)と、データエリアのアドレス表現(4ビット)が異なるものがあり、アセンブル対象はプログラム手続部分のみに限定しているものが多い³⁾。アセンブラ言語仕様の中にはアセンブラの負担を軽くするため、オペランドの記述を無くしたものや、記述制限を大きくしたものがある。また逆にソースプログラムの大きさの節減をね

らってアセンブラにコンパイラ的な機能を持たせたり、アドレスや定数の翻訳を工夫し、命令機能の低さをアセンブラ言語で吸収し、プログラムの負担を軽減している例もある。

```

***** 7* 07* 56 リスト *****
L.Nº F ADDR. N OBJ-CODE M
00001          ; *** N1 BIT LEFT SHIFT OF N2 BYTE DATA *****
00002          ; *** DATA SET ADDRESS 4,7=N2- ADDRESS 256=N1 *****
00003      00 00          CPG      0
00004      00 00      01 00 01      LKI      S,SHIFT
00005      00 03      11 04 01      L00P:   LKI      U,DATA+3
00006      00 06      26 04          MVI      H,4
00007      00 08      AF          XRA      A
00008      00 09      37          STC
00009      00 0A      3F          CMC
00010      00 0B      1A          L00P1:  LDAX   D
00011      00 0C      8F          ADC   A
00012      00 0D      12          STAX  D
00013      00 0E      18          UCX   D
00014      00 0F      25          DCR   H
00015      00 10      C2 08 00      JNZ   L00P1
00016      00 13      0A          LDAX  B
00017      00 14      3D          LCR   A
00018      00 15      02          STAX  B
00019      00 16      C2 03 00      JNZ   L00P
00020      00 19      76          HLT
00021      01 00          O+G    256
00022      01 00          SHIFT:  US    1
00023      01 01          DATA:  US    99H
00024          END

```

*** 1-7* 17- 27 0

図-4 アセンブルリスト例

PL/M LIST PMP-8S

```

*** PMP-8S カイロ
PMP-8S (REV.1)
000000 /CPI 1
000000 /ROM 0:255
000000 /RAM 256:511
000000 /LBAU 0:FILE
}シミュレーション・パタンの指定
}ROM値、RAM値の指定
}オブジェクト・プログラムのローディング

*** 0-ト スクリ (カイロ アドレス=0000:シヨクワウ アドレス=0019)
000000 /SET 256=1d
000000 /SET 257=1A283C4UH }指定アドレスへのデータセット
000000 /TMA A:3H }トレース範囲の指定
000000 /BRE 3H:5 }ブレイクポイントの設定
000000 /STA 0 }シミュレーション実行開始

*** シ・ツコウ カイロ
**** オール・トレース・リスト ****
アドレス ソース・イテージン ア B C D E H L 799* IE SP
00 03 LXI D 11 04 01 00 01 00 01 04 00 00 ***** 0 00 00
00 03 LXI D 11 04 01 11 01 00 01 04 00 00 ***** 0 00 00
00 03 LXI D 11 04 01 10 01 00 01 04 00 00 ***** 0 00 00
00 03 LXI D 11 04 01 0F 01 00 01 04 00 00 *****A 0 00 00

*** BP カ ノペイシタ (BP アドレス=0003)
000000 /REM A:3H }トレース指定の解除
000000 /CPI: }割込み・トレース・リストの出力継続指定

**** オール・トレース・リスト ****
アドレス ソース・イテージン ア B C D E H L 799* IE SP
00 03 LXI D 11 04 01 0E 01 00 01 04 00 00 C***** 0 00 00

*** シ・ツコウ スクリ (クワウツク= 3797)
*** (スツコウ・スワ= 594)
*** (シヨクワウ アドレス=0019)
000000 /LBAU 0:260 }ダンプ範囲の指定

**** グラフ・リスト ( 01 00 - 01 04 ) ****
アドレス ソース・イテージン
01 00 RAM 00F13400 00
A 00 B 01 C 00 U 01 E 00 H 00 L 00 SP 00 00 IE 0 799* **ZPA
000000 /ENDU }シミュレーションの終了表示
EOF

```

図-5 シミュレータ走行例

マクロ機能、条件付アセンブル機能などの付与も最近の傾向である。

アセンブルリスト例を図-4 (前頁参照) に示す。

3.2 シミュレータ

シミュレータはμPの動作を疑似するもので、チップメーカーなどからクロスセンブラと共に提供されている。各種類のシミュレータとも実行制御、命令の疑似のみならずメモリ領域の疑似機能、入出力および割込制御の機能、トレース・ダンプ等のデバッグ情報収集機能、ブレーク・ポイント設定機能等を取り入れている。処理方式としては、動作の解析と論理の修正を直に行える機能が要求されるため、インタプリタ方式、会話型が望ましいといえよう。

μP用シミュレータは一般に次のような問題がある。

- ① 実時間シミュレーションが難しい。
- ② 外部の入出力動作に対するシミュレーションが充分に行えない。
- ③ 汎用性に乏しい。例えば、新しいアーキテクチャにはもちろん、セカンド・ソースの機能追加に対応することも一般には難しい。
- ④ シミュレーションに長時間を要する。

256:

```

DECLARE DCL LITERALLY 'DECLARE';
DCL (A,B,C) BYTE;
DCL PTR ADDRESS;
DCL (PCOM BASED PTR) (3) BYTE;
PTR=0;
A=1; B=2;
PCOM(0)=A; PCOM(1)=B;
SUB: PROCEDURE;
DCL (U,V,W) BYTE, P ADDRESS;
DCL (COM BASED P) (3) BYTE;
P=0;
U=COM(0); V=COM(1);
W=(U+V)/2;
COM(2)=W;
RETURN;
END SUB;
CALL SUB;
C=PCOM(2);
HALT;

```

図-6 PL/M リスト例

一般的にいう μP プログラムの比較的独立した部分のデバッグはシミュレータにより行い、周辺機器を接続した実時間のプログラム・デバッグは試作品あるいはハードウェア支援システムにより行う。なおハードウェア支援システムについて後述する。

図-4 のプログラムをシミュレーションした例を、図-5 に示す。

3.3 高級言語

μP 用高級言語コンパイラは、INTEL 社が PL/M として最初に発表した (1974)。高級言語によるプログラム・コーディングはアセンブラの場合と比較すると、プログラムステップ数がかなり少なくて済むといわれているが、逆にメモリ占有量は大きくなる。マニュアルによれば、PL/M はアセンブラコーディングに対しプログラムステップ数が 1/10、メモリ増加率が 1.1 程度で済むと報告されているが¹¹⁾筆者の作成例では、それぞれ 1/5、2.5 程度であった。

PL/M は PL/1 のサブセットに μP 向きの機能を追加した言語仕様を採用している。INTEL 社の支援ソフトウェアはシンケージ・エディタがなく、ラベルと GO TO 文で絶対アドレス指定を許しているのが特徴であるが、高級言語レベルでプログラマが絶対番地を意識するのはやや抵抗がある。

PL/M と類似したクロスコンパイラは国内でもいくつか開発されつつある。レジデント・コンパイラとしては、NSC 社、INTEL 社が開発中と伝えられる。

一方、問題向き言語の開発も行われつつあり、SMS 社からコントローラおよび計測用のマクロ・アセンブラが、COMSTER 社からシーケンス制御用コンパイラ PCL が発表されている³⁾。

PL/M ソースリスト例を図-6 (前頁参照)に示す。

3.4 ユーティリティプログラム

言語処理プログラムとシミュレータ以外のソフトウェア作成支援システムを、ユーティリティプログラムとして分類する。なお、 μP 支援ソフトウェアシステムの体系は、これといった決定的な分類がないので、代表的なものを示す。

(1) デバッガ: マイクロコンピュータ上のプログラムの走行状態を確認するのに用いるプログラムである。トレーサと命名した支援システムもある。

(2) エディタ: ソースプログラムの変更・追加・削除などの編集に用いる。

(3) リンカージェディタ: 相対形式で作られた複数の目的プログラムを結合するときに用いる。開発する μP プログラムの規模の増大に伴ない、リーカージェディタが導入される傾向にある。

(4) オブジェクトチェッカ: オブジェクトテープに出力された内容と、メモリまたはファイル上のオブジェクトプログラムとを照合するプログラム。

3.5 μP プログラム部品

μP プログラム作成のためのプログラム部品として、サブルーチン、システムマクロ、マイクロコンピュータ・モニタがある。

(1) サブルーチン: μP プログラム設計にしばしば使用される機能をサブルーチン形態で結合して使われる。サブルーチンは、種別 (機能) とサブルーチンの占有メモリ量が大切である。

サブルーチンの設計には、処理速度を重視した構成と、速度よりもメモリ占有量の節約を狙った構成とする2つの方向がある。表-2 に代表的なサブルーチンを示す。

(2) システムマクロ: サブルーチンと同様な機能であるが、マクロ形態で提供するものである。

(3) マイクロコンピュータ・モニタ: マイクロコンピュータに各種の周辺機器を付加して μP 応用機器を組立てる場合、マイクロ・コンピュータモニタを、 μP プログラムの部品 (プログラム構成要素) として

表-2 サブルーチン一覧

分類	サブルーチン品目例
型変換	1バイト2進→外部16進 内部10進→1バイト2進 グレイコード→1バイト2進 など
算術演算	2バイト2進加算/減算 2バイト2の補数 2バイト算術右シフト 1バイト絶対値 など
論理演算	2バイト排他的論理和 4バイト論理積 など
関数	正弦 余弦 対数 指数 平方根 など
編集	文字編集 ゼロサプレス ビット反転 など
その他	CRC (Cyclic Redundancy Check) 乱数 等差数列 など

使用する。なお、3.1~3.5 は、図-2 のソフトウェア作成支援システムと保守システム (エディタ) に該当する。

3.6 プログラムローディングシステム

μP プログラムの目的プログラムは一般に紙テープを介して、マスク ROM あるいはヒューズ ROM に設定されたり、マイクロコンピュータの入力プログラムまたはデータとなる。この紙テープは通常 ROM 紙テープまたは RAM 紙テープと呼ばれ、プログラムローディングシステムによって作成する。

紙テープフォーマットとして、BNPF コードがよく知られているが、その他、HEXA フォーマット (16進数フォーマット)、バイナリ・フォーマットがある (図-7 参照)。なお紙テープの代りに MT が媒体として使われることも多い。

3.7 マイクロプログラム作成支援システム

マイクロプログラム制御方式 μP のプログラム開発を行う場合は、マイクロプログラムアセンブラが必要となる⁶⁾。

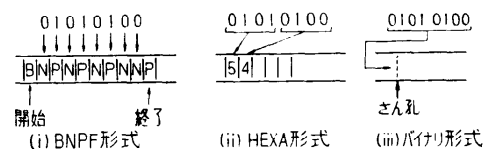
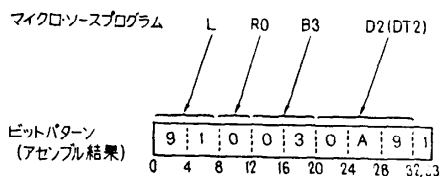


図-7 BNPF, HEXA バイナリー各形式の概念



補足説明: 本例では、一語が0ビットから33ビットまでの34ビットで、0~7, 8~11, 12~19, 20~31の4つのフィールドと、データパリティ(32ビット目)、アドレスパリティ(33ビット目)から構成されている。

L, R0, B3はマイクロワードと呼ばれ、フィールドビットパターンとの対応が既に定義されている。D2は20~31のフィールドにつけられたフィールド名で、ここに、リテラル値DT2(レーベル値、値:0A9)が入ることを示す(MAPASSの例⁹⁾)。

図-8 マイクロプログラムアセンブラの原理

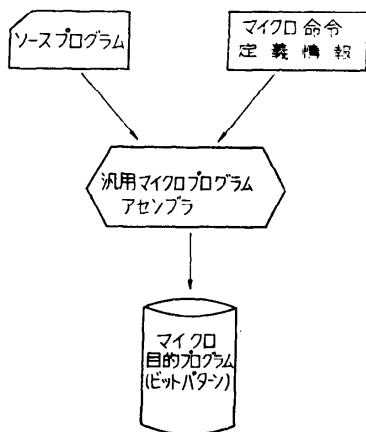


図-9 汎用マイクロプログラムアセンブラ

マイクロプログラムアセンブラは、マイクロ命令を構成するフィールドにマイクロワードとしての情報を記号または数字により指定するもので、普通のアセンブラと類似している(図-8参照)。

マイクロプログラム支援システムは、従来からマイクロプログラム制御の汎用計算機的设计・製造に使われてきている。一般には、マイクロ命令形式を定義する情報を、マイクロプログラムアセンブラに入力する方式として、多機種向けの汎用マイクロプログラムアセンブラが望ましい(図-9参照)。

INTEL社は、3000シリーズのマイクロプログラムアセンブラとしてCROMIS⁹⁾を開発しており、またSMS社は、マイクロプログラム制御方式のマイクロプログラムアセンブラとしてRAPID¹⁰⁾を開発して

いるが、普通のアセンブラ程には普及していない。

マイクロプログラムアセンブラの1つの例として、MPASS²⁾*について述べる。

MPASSは、マイクロ命令などのアーキテクチャに関するものをマイクロプログラムの定義部として記述し、以降に続くマイクロプログラムアセンブラのための変換用索引テーブルを作る。そしてマイクロプログラムアセンブラ部では、この定義で処理された変換用テーブルに従ってマイクロプログラム・ソースステートメントを変換し、マイクロ命令ビットパターンを出力する。MPASSのアセンブルリストを図-10(次頁参照)に示す。

マイクロプログラム制御方式システム開発のためにはマイクロプログラム・シミュレータも道具**として使われている。これはマイクロ命令の動作機能を擬似するもので、プログラム・ロード機能や、メモリ内容のディスプレイ、変更といった機能を有している。

なお、マイクロプログラムローディングシステムについては省略したが、プログラム・ローディングシステムと同様な形態が必要となる。

3.8 装置試験・評価支援システム

装置試験システムは、装置がほぼ完成する段階で試験データを大量に必要とする場合、あるいは人手では作りにくい場合、試験データをプログラムによって作成する。

評価プログラムは、例えば μ Pプログラムの命令使用頻度などを測定する。

3.9 保守システム

μ PプログラムをMT、紙テープなどの媒体に保存し、プログラム変更などのアフタサービスなどに対処するためのプログラム群である。ソフトウェア作成支援システム、およびマイクロプログラム作成支援システムのプログラムと同一のものがある。保守システムとしては他に大型計算機で利用されているフローチャートプログラムなどのドキュメント作成プログラムも将来作られるであろう。これが完備されていればプログラムをよく理解できるし、変更する際も大いに役立つ。

3.10 設計製造支援システム

μ P応用機器設計製造のためのDAシステムともいうべきもので、 μ Pを組込んだ回路のシミュレーション、I/Oインタフェース回路のシミュレーション、プリント板の設計などを行うものである。

μ Pアーキテクチャが進歩し、 μ P組込回路やプリ

* 日本電信電話公社武蔵野電気通信研究所で開発

** ファームウェア形態

4.2 クロスソフトウェア

クロスソフトウェアは、新しいコンピュータやミニコンピュータ（ターゲットマシン）のプログラム開発を、強力な OS の機能、各種の周辺装置の恩恵が得られる大型コンピュータ（ホストマシン）で行い、ソフトウェア開発の能率向上を図ろうとするものである。

クロスソフトウェアには FORTRAN などで作成したポータビリティを持ったものと、主としてアセンブラ記述により走行マシンを限定した専用のものがある。チップメーカーは前者を MT に格納して販売しているが、後者については主にミニコンピュータを用い、クロスソフトウェアとミニコンピュータを一組にして販売する例が多いようである。

12, 16 ビット系のチップの中には、LSI-11, IMP 6100, TMS 9900 などのように既存のミニコンファミリ-の1つとしてソフトウェア互換性を強調したものがみられる。この場合クロスソフトウェアとレジダントソフトウェアの区別がなく、ソフトウェア資産の活用といった面では効率が良い。

次に TSS 用クロスソフトウェアについて述べる。

TSS 用クロスソフトウェアは、市販の FORTRAN 記述クロスソフトウェアを、TSS 用に流用しているものが多い。市販用クロスソフトウェアは、ポータビリティを考慮した設計（即ち、汎用化設計）によって処理速度が犠牲になり易いので、TSS に流用するよりは TSS 専用のクロスソフトウェアを新規に作った方がよい。

5. ハードウェア支援システム

ハードウェア支援システムの開発状況、機能、動向に関し以下に述べる。

5.1 開発状況

ハードウェア支援システム用マイクロコンピュータが各種開発されているが、これは μP プログラム開発向きというよりは、ホストマシンで開発したプログラムをマイクロコンピュータ上で走行確認する、いわゆるハードウェアシミュレータとしての機能が強い。最近は二次記憶装置としてフロッピー・ディスクを備え、プログラムの流れを追跡できるよう CRT ディスプレイ装置を付加したものが現われている。

5.2 機能

ハードウェア支援システムの特徴を中心に述べる。

(1) オンラインデバグ

インタフェース回路、周辺装置を組入れて μP プロ

グラムのオンラインデバグ（入出力装置の動作、外部割込動作など）の試験が行える。

μP プログラムデバグ機能としては、1ステップ実行機能、アドレスストップ機能（ブレーク・ポイント）がついているものが多く、ROM またはカセット MT にデバグ用プログラムを内蔵している。

(2) PROM へのオンライン書込み

PROM 書込装置を付加することにより、デバグが完了した μP プログラムを書込むことができる。

(3) 生産用に直結

モジュール構成（CPU モジュール、I/O モジュールなど）を採用しているものがあり、各モジュール・ボードをそのまま実システムへ適用できる。

5.3 今後の動向

各種 μP のエミュレータ方式のハードウェア支援システム、あるいは周辺機器との自由度の大きいインタフェースを持つハードウェア支援システムができれば利用者にとって好ましいといえよう。

6. 今後の課題

6.1 アセンブラの動向

支援ソフトウェアとして基本的なアセンブラは、チップメーカーおよびユーザによって多数重複して作成されているのが現状であり、この点かなりの無駄がある。アセンブラの開発を容易に行う手段を開発することは緊急の課題といえよう。

アセンブラーアセンブラ技術により目的とするアセンブラを能率よく作成する方式（方式-I）、あるいは汎用アセンブラ方式（方式-II）は重要である（図-11参照）。方式-Iは目的とする特定のアセンブラを生成する方法で、ミニコン・ベースのものが最近発表されている。電総研の汎用アセンブラ方式は、マイクロ

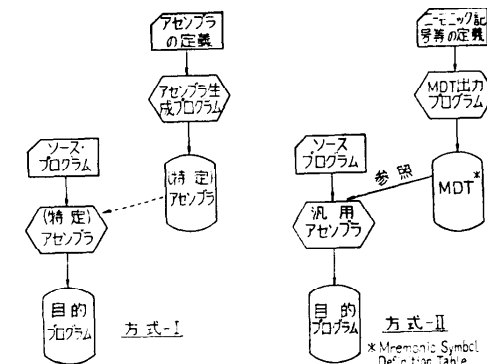


図-11 アセンブラの作り方

* Mnemonic Symbol Definition Table

プログラム制御方式計算機（モジュール型複合計算機 ACE 等）を対象としたものであり⁴⁾、命令の構成が整然とした形式（フィールド・センシティブ）の μP には適用が可能である。ASR 社は一定の様式にかなう μP について方式-II の MDT 部分を人手によって作成する方法であり、交換可能なチップ数は、数種である。方式-I と方式-II のいずれにせよ、今後の μP 用アセンブラの進む方向を示唆しているといえよう。

マイクロプログラムアセンブラは同様な理由により汎用化指向が必要である。しかしマイクロプログラミング機能をユーザに解放することは、マイクロプログラム・アセンブラ提供者のアフタサービスに解決を要する問題が多い。

6.2 高級言語の動向

高級言語で作成したプログラムは、メモリ占有量が比較的に大きくなるため普及のネックになっていると考えられる。高級言語コンパイラがアセンブラ程度に普及するためには、次の動きが必要であろう。

- ① 多機種・少量生産 μP 応用機器の開発
- ② メモリ価格の低下
- ③ メモリ拡大率の小さい高級言語コンパイラの出現
- ④ μP プログラムの流通
- ⑤ ハードに対するソフト生産性の相対的低下、あるいは μP プログラム大規模化などによる μP プログラム開発費の増加

プログラム教育とプログラム流通の観点から、高級言語の仕様は、 μP とは独立することが望ましいが、 μP 向きにどの程度の言語仕様にするかという点では解決を要する問題が多い。なお XY プロッタ、CRT ディスプレイ向きの図形処理言語やその他の問題向言語も望まれよう。

PL/M 言語は PL/I 系の言語であるが、最近注目をあびている APL 系の μP 高級言語が出現する可能性も充分考えられよう。

μP 対応の目的プログラムを複数個出力する汎用高級言語コンパイラの生成についても、汎用アセンブラと同様なソフトウェア汎用化技術が要請される。

6.3 紙テープフォーマット

ROM および RAM 紙テープフォーマットとして、BNPF コードはかなり標準化されているが、紙テープ量が増加する欠点がある。HEXA フォーマット、バイナリ・フォーマットはそれぞれ各社独自の形式が使われており、特に標準化されていない。このため

表-3 紙テープフォーマットを区分する要因

項目	内容
形式	バイナリ形式 BNPFF 形式) ... HEXA 形式) { 4ビット方式 (片側オール0 片側制御情報) { 7単位方式 (8ビット目0 8ビット目1) { 8単位方式 (偶数パリティ 奇数パリティ)
レコード構成	印字する場合の一行の文字数 区切り記号 CR, CR・LF, LF(NL)
挿入記号	スペース, ; ; /%*, SE その他の記号
エラーチェック	なし あり...チェックサム (加法, 減法), CDC (多項式), 垂直パリティ (偶, 奇), 水平パリティ (偶, 奇)
アドレス付与方式	アドレスなし アドレスあり...チップ単位式, 連続方式 付与方法...8進, 10進, 16進
チップ終了記号	なし あり...*, その他の記号
テープヘッダ情報	なし あり...テープナンバ, キーワード, チップ烙印記号その他
その他	相対形式か否か, 相対形式ではシンボルテーブルの形式, チップ単位式の場合は1チップ内の部分出力の可能性

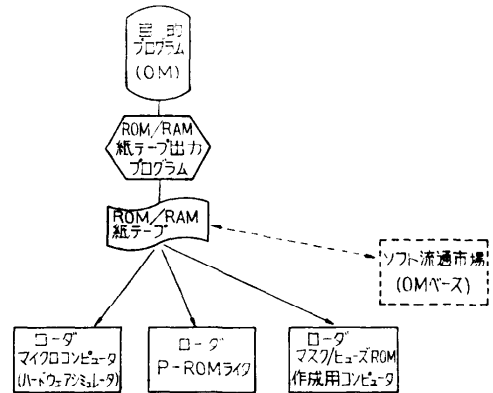


図-12 ROM/RAM 紙テープを介したプログラムの流れ

P-ROM 書込器とのインターフェースがバラバラである他、 μP プログラムの流通などに支障が生じている。

表-3 に紙テープ・フォーマットを区分する主な要因を示す。たとえば同じ HEXA フォーマットでもパリティビットの取り扱いや、レコード区切り記号、エラーチェックの方法などでそれぞれ互換性のない紙テープとなる。

図-12 に示すように紙テープは μP プログラム流通の媒体として重要であり、フォーマットの標準化が強

くのぞまれる*。

マイクロプログラム・アセンブラの場合もマイクロプログラミングが一般に普及すれば紙テープ・フォーマットに関する同様の問題が生ずるであろう。

6.4 μ P プログラムの流通

μ P プログラムの開発はかなり工数がかかるので、「プログラム・バンク」などがあって μ P プログラムが流通できる組織が生まれることが望まれる。流通するプログラム単位としては独立した μ P プログラムと、サブルーチンのようなプログラム部品とがある。

プログラム流通上は、 μ P 種別が沢山あることから、例えば、 μ P-A 用のサブルーチンは μ P-B 用のサブルーチンに流用できないのが難点である。この点 μ P 機種によらない高級言語で記述したプログラムが生まれれば便利であろう。 μ P プログラムをライブラリとして整理し、ユーザに提供しようとしている団体、メーカーが国内に2~3生れている。

6.5 TSS による支援ソフトウェア

クロスソフトウェアを TSS によりサービスする場合、経済性にやや難点があるといわれていたが、最近の TSS によるクロスソフトサービスは1年前と比較し、数倍改善されてきている。

TSS によるクロスソフトサービスのキーポイントは、汎用アセンブラを構成要素とする汎用クロスソフトウェアサービスであろう。汎用アセンブラによれば、マイクロプログラム制御方式 μ P あるいはビットスライス μ P も対象とできる他、新しいチップが発表されると同時にクロスソフトサービスが開始できるという利点がある。また、多数の μ P 応用機器を長期間にわたって開発、保守する場合、汎用アセンブラを使えば、 μ P 機種が増えてもアセンブラは増えない利点がある。

汎用アセンブラ・ベースの TSS クロスソフトウェアは μ P プログラムの流通体制作りにも貢献できる。また μ P にこだわらず、ミニコンピュータ、超小型計算機あるいはインテリジェント・ターミナルなどのソフトウェア作成にも適用できる可能性も生まれる。

7. む す び

マイクロコンピュータのソフトウェアについてその現状と今後の課題について述べた。 μ P の利用が多様化、高度化するにつれ、 μ P プログラムはますます重要なものと認められているが、一方では生産性の低さ

が μ P 普及のネックとなっている。 μ P に関する支援ソフトウェア体系として決定版といえるようなものはなく、今後の動向によるが、クロスソフトウェアによる支援ソフトウェアの概念は定着しよう。 μ P プログラム開発支援システムとしては、クロスソフトウェア支援システムと、ハードウェア支援システムとが併立することとなる。

今後の課題としては、長期間にわたる μ P プログラム保守体制、汎用アセンブラ、ROM/RAM 紙テープの標準化、 μ P プログラムの流通などが考えられる。

おわりに、日頃から懇切な御指導を戴いている横須賀電気通信研究所 高島部長、戸田調査役に深甚の謝意を表します。

参 考 文 献

- 1) 末永康仁他：マイクロコンピュータ制御のインテリジェント XY プロッタ装置、画像工学研究会資料、IE 75-2 (1975-04)。
 - 2) 星 元雄他：マイクロプログラムアセンブラの一構成法、情報処理学会設計自動化研究会資料、No. 75-24 (1975-2)。
 - 3) 特集：マイクロコンピュータ、日経エレクトロニクス、No. 121, pp. 58~157 (1975.11)。
 - 4) 飯塚他：モジュール型複合計算機 (ACE)、情報処理学会アーキテクチャ研究会資料、74-4 (1974.10)。
 - 5) Howard Falk: Microcomputer software makes its debut, IEEE Spectrum, Vol. 11, No. pp. 77~84 (Oct. 1974)。
 - 6) Charles W. Flink, II: A Microprogrammed Environment for a Software Development System, COMPCON '75-Fall Committee, Session 5.
 - 7) PANAFACOM-16A セルフスタンディングシステム説明書 (暫定版)、パナファコム社。
 - 8) MRTOS (μ -Real Time Operating System) 解説書: ASR 社。
 - 9) Intel Series 3000 Cross Microprogramming System CROMIS Reference Specification, 98-113A, INTEL 社。
 - 10) RAPID a Configuration-Independent Symbolic Assembler for Read-Only Store, SMS 社。
 - 11) A guide to PL/M Programming: INTEL 社。
 - 12) MCS-X クロスアセンブラ, ASR 社。
- 〔補足〕 No. 7~13 はプログラムマニュアルを示す。
(昭和50年12月6日受付)
(昭和51年1月23日再受付)

* 媒体としては紙テープの他 MT もあるが省略した。