

# マイクロプロセッサ用標準OMフォーマット 試案 (SYSROF) について

金井 敦 杉山 和弘 神谷 芳樹  
(電電公社 横須賀通研)

## 1. はじめに

近年のマイクロプロセッサ ( $\mu P$ ) 用ソフトウェア開発環境の特徴として次のような点を指摘できる。

- ① 使用される  $\mu P$  機種が多様化している。
- ② インサーキットエミュレータ (ICE) や  $\mu P$  開発装置等の  $\mu P$  依存性が高く、多種の  $\mu P$  を使用する場合に経済的負担が大きい。
- ③ 同一  $\mu P$  に対しても多種の開発環境があり、相互の連携が円滑でないため、資源の有効利用やソフトウェアの流通が阻害されている。
- ④ 高水準言語の利用率が高まり、Adaのように処理系にデータベースを必要とする言語も用いられる等言語処理系は大型化の傾向にある。このため、中大型機によるソフト開発環境のニーズが高まっている。

⑤④の理由から言語処理を中心とした開発環境とICE等を用いる走行試験環境が分離したクロス開発形態が普及しつつある。この場合に、ソフト開発環境と走行試験環境の繋ぎとなるオブジェクトモジュール (OM) フォーマットが不統一であるため、両者の連携が円滑に進まない。

筆者らは、このようなソフトウェア開発環境に係わる問題の根源的な要因について次のように考察した。

- ① ソフトウェア開発環境を  $\mu P$  機種毎に区別してとらえる考え方は生産性向上に寄与しない。
- ② ソフトウェア開発環境には、本来、プログラム“開発”と開発したプログラムの“走行”という2つの仕事があり、言語処理を中核とした狭義の“開発環境”と“走行試験環境”に分けて考えるべきである。これら2つの環境の具備すべき条件は各々異なるので生産性向上のためには各々に対しふさわしい環境を用意することが重要である。しかし、明確に区別されていないのが現状である。具体的に、前者としてふさわしい環境は、十分な資源とソフトウェアツールの充実した中大型機を使用した環境であり、後者としては  $\mu P$  機種に密着してきめ細かい機能を装備したICEあるいは  $\mu P$  開発装置 (以後、両者を合わせて  $\mu P$  開発装置と呼ぶ) と呼ばれる装置を使用した環境である。
- ③ このような考え方でソフトウェア開発環境全体を考えたとき問題となるのが、2つの環境の結合方式である。現在2つの環境の結合を拒む大きな要因としてOMフォーマットの不統一という問題がある。

このような問題を克服するために、筆者らはシンボル情報等のデバッグ情報を含んだ標準OMフォーマット試案を提案した<sup>(1)</sup>。OMフォーマットの統一

は上記に起因する問題の解決に次のような点で貢献する。

- ① リンカの統一  
リンクをツールや言語の種類から独立させることができるので、様々なツールや言語で開発したプログラムのリンクを統一的行なう事ができる。
- ② ホストと  $\mu P$  開発装置の任意の接続  
クロス開発形態で任意のホストと任意の  $\mu P$  開発装置を接続しホストで開発したOMを  $\mu P$  開発装置へ転送してシンボリックデバッグを含む走行試験を行なえる。
- ③ 大型TSSによるクロス開発時のプログラム転送時間の短縮

大型TSSによるクロス開発形態において、従来全モジュールを結合したLMをセンタから端末へ転送していたものを、必要なモジュールのみOMの形態で転送し端末側でリンクすることが可能となり、プログラム転送時間を短縮できる。

- ④ 異なる装置で開発したプログラムの相互リンク  
中大型機、 $\mu P$  開発装置等異なった装置で開発したOMを相互にリンクできる。

## ⑤ OMレベルでのソフト流通

ソース、LMに加えてOMによってもソフトウェアを流通させることが可能となり、他のプログラムとの結合やシンボリックデバッグ等の利便性が増す。

本稿においては、2章で先に提案した標準OM (SYSROF) の基本的考え方を、3章でIEEE-P695から発表されている標準OM (MUFOM) および実際に使用されているOMとの比較を示す。4章でSYSROFの詳細を、5章で現状を示す。最後に、6章で今後の課題を示す。

本論にはいる前に、本稿で使用用語の定義を示す。

- ① セグメント：18086等の  $\mu P$  で用いられているアドレス空間の区切り
- ② モジュール：OMの一塊である。SYSROFでは、ヘッダレコードとトレーラレコードで囲まれた部分をいう (ヘッダレコードとトレーラレコードも含む)。
- ③ セクション：再配置をするときの最小単位である。モジュールは複数のセクションから構成される。

## 2. SYSROFの基本的考え方

標準OMフォーマットとして具備しなければならない条件を要求の強い順に以下に示す。

- ①各種のμP用のオブジェクトを同一フォーマットで表現できなければならない(機種独立性)。
- ②高級言語デバグを容易にするため、シンボル情報等のデバグ情報を持たなければならない(デバグ情報充足性)。
- ③μP開発装置上でリンクが行なわれることが多い事から、リンクの処理が軽いことが必要である(処理容易性)。
- ④今後のμP開発装置の発展や言語の発達に伴い仕様上の情報追加が容易でなければならない(情報追加容易性)。
- ⑤μP開発装置のデバグ能力は様々な相違があるので、OM内に含まれる情報の取捨選択が容易なフォーマット構成である必要がある(情報選択性)。
- ⑥比較的資源の小さいμP開発装置にもOMを格納しなければならないこと、また通信路を伝送させる場合の時間短縮のため、なるべく少ないバイト数で表現される必要がある(コンパクト性)。

SYSROFにおいては、上の要求条件を次のように解決した。

- ①については、μP機種をOM内の情報に含める方も考えられるが、リンクがμP機種を意識

しないでOM内の情報だけを参照することにより、処理が可能なように、アドレス更新単位、セグメント方式フラグあるいはアドレスの分割等の情報を設定した。

- ②については、高級言語イメージのデバグに必要な情報として、シンボル名およびアドレス、構造体、プロシージャの包含関係、名標の型情報および行番号情報をOM内に設定した。

- ③については、MUFOMで採用されているコマンド方式もあるが、多くのμP開発装置で使用され各レコード内に設定される各種情報の位置や表現方法等がすでに定まっています処理のしやすいブロック化ベタ詰め方式とした。

- ④については、レコードタイプを増設することにより容易に情報を追加できるようにしている。

- ⑤については、OM情報の種類により、オブジェクト情報、シンボル情報等10種類のレコードに分類し、このレコード単位に情報を取捨選択できるようにした。

- ⑥については、大型計算機上での処理のしやすさから必要以上に領域を取っている情報がある。このため、コンパクト性は十分とは言えず今後の課題である。

### 3. SYSROFとその他OMの比較

本章では、実際に使用されている4種類のOMフォーマットおよびIEEE-P695から提案されている標準OMフォーマットであるMUFOMの特徴を比較する。主な項目について比較したものを表1に示す。各OMフォーマットにおいて特徴的なこ

とを以下に示す。

- ①MUFOMを除いて、多少の相違はあるものの、情報がレコード単位にタイプ分け(例えば、ヘッダレコード、オブジェクトレコード等)されているのに対し、MUFOMは、コマンド形式(プログラム言語的表現)を取っている。

- ②標準OMに対する要求の中で最も必要と思われる機種独立性を持っているOMは、MAPS(DIPS上のコンパイラ等で使用されているOM)、SYSROFおよびMUFOMである。

- ③上記3つのOMの中では、MAPSは処理容易性、コンパクト性に欠け、MUFOMはデバグ情報に欠けている。

### 4. SYSROFの仕様

#### 4.1 概要

SYSROFは、情報の単位ごとに以下に示す11のレコードタイプを持つ。

- ①ヘッダ情報レコード (レコードタイプ0)
- ②セクション情報レコード (レコードタイプ1)
- ③外部参照シンボル情報レコード (レコードタイプ2)
- ④外部定義シンボル情報レコード (レコードタイプ3)
- ⑤外部入り口シンボル情報レコード (レコードタイプ4)
- ⑥リロケーション情報レコード (レコードタイプ5)
- ⑦オブジェクト情報レコード (レコードタイプ6)
- ⑧シンボル情報第1レコード (レコードタイプ7)

表1. 各種OMの比較

項番	標準OMの具備条件	A	B	C	MAPS	SYSROF	MUFOM	
1	機種独立性	×	×	×	○	○	○	
2	デバグ情報充足性	シンボル名	○	○	○	○	○	
		構造体、手続きの入れ子関係	×	×	○	○	○	規定
		型情報	×	×	○	○	○	なし
		行番号	×	×	○	○	○	
3	処理容易性	○	○	○	△	○	△	
4	情報追加容易性	○	○	○	○	○	○	
5	情報選択性	○	○	○	△	○	○	
6	コンパクト性	○	○	○	×	△	×	

A, B, C: 代表的なICEで使用されているOM  
 MAPS: DIPS上のマルチターゲットクロスコンパイラ・アセンブラで使用されているOM

⑨シンボル情報第2レコード (LJ-トタイプ8)

⑩ソースラインナンバー情報レコード(LJ-トタイプ9)

⑪トレーラ情報レコード (LJ-トタイプ10)

SYSROFは、セクション単位にオブジェクトを一塊として扱う。各レコードはセクション単位に図1に示すように繰り返えられる。表現は、16進のアスキーコード表現である。

各レコードは、付録1のように、先頭にレコードタイプ(2byte)とレコード長(2byte)が、最後にチェックサム(1byte)が設定される。ここで、レコードタイプとはレコード種別を示すレコード番号である。レコード長とは、1レコードの全バイト数の2分の1の値(最大値は60である)である。チェックサムとは、レコードの先頭からASCIIコードの上位4ビットを0として0から減算した結果である。

#### 4.2 各レコードの説明

##### 4.2.1 ヘッダ情報レコード

レコードタイプは0である。ヘッダ情報レコードは以下に示す情報から構成される。

(1) ファイル名(8文字)

ファイル名が、左詰めではいる。最大8文字である。8文字に満たない場合は、残りの部分にスペース(20H)が詰められる。

(2) スタートアドレス(4byte)

プログラムモジュールのスタート番地である。絶対形式の時のみ有効である。

(3) プログラム長(4バイト)

(モジュール終了アドレス-開始アドレス+1)

の値が入る。モジュールが1セクションから構成されているときのみ有効である。

(4) プログラムタイプ(4byte)

プログラムのタイプにより、表2に示す値が設定される。

表2. プログラムタイプ

値	プログラムタイプ
0	すべてのセクションが絶対形式
1	すべてのセクションが相対形式
2	絶対形式と相対形式のセクションが混在

(5) ツール識別(4byte)

0Mを作成したツールを示す値である。現在、0から14をNTTが使用している。その他は未定義である。

(6) アドレス更新単位(4byte)

1アドレスが示すビット数である。

(7) セクション数(4byte)

本モジュールに含まれるセクション数である。

(8) 作成日付け(8文字)

本モジュールが作成された日付けである。YY-MM-DD(YY:年、MM:月、DD:日)で表わす。

(9) CPUタイプ(4byte)

セグメントタイプ(例えば、18086)の場合

<ヘッダ情報>

<セクション情報>

<外部参照シンボル情報><外部参照シンボル情報>.....

<外部定義シンボル情報><外部定義シンボル情報>.....

<外部入り口シンボル情報><外部入り口シンボル情報>.....

<リロケーション情報><リロケーション情報>.....

<オブジェクト情報><オブジェクト情報>.....

<シンボル情報1><シンボル情報2><シンボル情報1><シンボル情報2>.....

<行番号情報><行番号情報>.....

<セクション情報>

<外部参照シンボル情報><外部参照シンボル情報>.....

<外部定義シンボル情報><外部定義シンボル情報>.....

<外部入り口シンボル情報><外部入り口シンボル情報>.....

<リロケーション情報><リロケーション情報>.....

<オブジェクト情報><オブジェクト情報>.....

<シンボル情報1><シンボル情報2><シンボル情報1><シンボル情報2>.....

<行番号情報><行番号情報>.....

.....

<トレーラ情報>

図1. 各レコードの出現順番

は0が、ノンセグメントタイプ（例えば、M68000）の場合に1が設定される。  
 (10) セグメントサイズ (4 byte)  
 1セグメントが占めるアドレスサイズである。  
 (11) セグメントシフト (1 byte)  
 実アドレスを計算するときのセグメントアドレスのビットシフト数である。

#### 4.2.2 セクション情報レコード

レコードタイプは1である。ヘッダ情報レコードは以下に示す情報から構成される。

- (1) セクション名 (8文字)  
 セクション名が、左詰めである。最大8文字である。8文字に満たない場合は、残りの部分にスペース (20H) が詰められる。
- (2) セグメントアドレス (4 byte)  
 セグメントアドレスが入る。絶対形式の場合のみ有効である。他の場合は、0が詰められる。
- (3) セクションアドレス (4 byte)  
 セクションの開始アドレスが入る。絶対形式の場合のみ有効である。他の場合は、0が詰められる。
- (4) セクションタイプ (4 byte)  
 セクションのタイプにより、表3に示す値が設定される。

表3. セクションタイプ

セクションタイプ	値	セクションの意味
ABS	0	命令部、定数データ、変数データにかかわりなく、配置アドレスがすでに決定しているオブジェクトが設定されるセクションである。
INS	1	命令部のオブジェクトが設定されるセクションである。
COS	1	定数データのオブジェクトが設定されるセクションである。
VRS	2	変数のデータが設定されるセクションである。
SPS	3	様々な用途に使用されるセクションである。

- (5) 境界調整数 (4 byte)  
 セクション割り付け時のアドレス境界条件（例えば、2バイト境界のとき2）が設定される。
- (6) 外部定義シンボル個数 (4 byte)

外部定義シンボル情報レコードで定義された外部定義シンボルの個数である。

- (7) 外部参照シンボル個数 (4 byte)  
 外部参照シンボル情報レコードで定義された外部参照シンボルの個数である。
- (8) リロケーション情報個数 (4 byte)  
 リロケーション情報が差し示しているリロケーション対象の個数である。リロケーション情報が複数に分離している場合はまとめて一個と数える。
- (9) 外部入り口シンボル個数 (4 byte)  
 外部入り口シンボル情報レコードで指定されたシンボルの個数。
- (10) オリジン情報個数 (4 byte)  
 オブジェクト情報レコード内に出現したオリジン情報の個数。
- (11) シンボル個数 (4 byte)  
 シンボル情報レコードで定義されたシンボル個数。
- (12) セクション長 (4 byte)  
 そのセクション内のオブジェクトのバイト数。

#### 4.2.3 外部参照シンボル情報

レコードタイプは2である。ヘッダ情報レコードは以下に示す情報から構成される。

- (1) 外部参照シンボル名 (8文字)  
 他のセクションで定義されかつ本セクションから参照されるシンボル名である。最大8文字である。8文字に満たない場合は、残りの部分にスペース (20H) が詰められる。

#### 4.2.4 外部定義シンボル情報

レコードタイプは3である。ヘッダ情報レコードは以下に示す情報から構成される。

- (1) 外部定義シンボル名 (8文字)  
 本セクション内で定義され、他のセクションで定義されるシンボル名である。最大8文字である。8文字に満たない場合は、残りの部分にスペース (20H) が詰められる。
- (2) 外部定義シンボルアドレス (4 byte)  
 直前の外部定義シンボルのアドレスである。
- (3) シンボルタイプ (4 byte)  
 外部定義変数の時0が、外部定義入り口名のとき1が入る。

#### 4.2.5 外部入り口シンボル情報レコード

レコードタイプは4である。ヘッダ情報レコードは以下に示す情報から構成される。

- (1) 外部入り口シンボル番号 (4 byte)  
 外部参照シンボル情報レコードで定義されたシンボルの出現順番で外部入り口シンボルを差し示す。

#### 4.2.6 リロケーション情報レコード

レコードタイプは5である。ヘッダ情報レコード

は以下に示す情報から構成される。

(1) リロケーションタイプ (4 byte)

リロケーション対象が分割されているかどうか、およびセグメントアドレスか通常のアドレスかにより、表4に示す値が入る。

表4. リロケーションタイプ

値	リロケーションタイプ
0	アドレスは1フィールドからなる。
1	アドレスは2フィールドに分割されている。
2	アドレスは3フィールドに分割されている。
3	セグメントアドレスは1フィールドからなる。
4	セグメントアドレスは2フィールドに分割されている。
5	セグメントアドレスは3フィールドに分割されている。

(2) アドレスデータのサイン (4 byte)

リロケーション情報が差し示すオブジェクト内の相対アドレスの値が正の場合は0が、負の場合は1が設定される。ただし、オブジェクト内の相対アドレスが負の場合は、そのアドレスは2の補数表現されている。

(3) アドレス内のビット位置 (4 byte)

リロケーション対象アドレス内の当該リロケーション情報が対象とするビット位置である。

(4) フィールド長 (4 byte)

当該リロケーション情報が対象とするフィールドのビット長である。

(5) 外部参照シンボル番号 (4 byte)

外部参照シンボルレコードに出現したシンボルの出現順番が入る。外部シンボルを参照しない場合は0が入る。

(6) リロケーション対象アドレス (4 byte)

オブジェクトプログラム内のリロケーション対象を差し示すセクション内相対アドレスである。

#### 4. 2. 7 オブジェクト情報レコード

レコードタイプは6である。ヘッダ情報レコードは以下に示す情報から構成される。

(1) 開始表示フラグ (4 byte)

次のエリアに開始/終了アドレス情報がある場合

は1が、ない場合は0が入る。

(2) 開始アドレス (4 byte)

オブジェクトプログラムの開始アドレスである。相対形式の場合はセクション内相対アドレスが、絶対形式の場合は絶対アドレスが入る。

(3) 終了アドレス (4 byte)

オブジェクトプログラムの終了アドレスである。相対形式の場合はセクション内相対アドレスが、絶対形式の場合は絶対アドレスが入る。

(4) オブジェクトプログラム

オブジェクトコードである。

#### 4. 2. 8 シンボル情報第1レコード

レコードタイプは7である。ヘッダ情報レコードは以下に示す情報から構成される。

(1) シンボル名 (10文字)

ソースプログラム内で定義されたシンボル名が左詰めで入る。最大10文字である。10文字に満たない場合は残りの部分にスペース(20H)が詰められる。

(2) アドレス (4 byte)

シンボルのアドレスが入る。相対形式の場合はセクション内相対アドレスが、絶対形式の場合は絶対アドレスが入る。

(3) データ長 (1 byte)

そのシンボルが占める領域の大きさがアドレス更新単位で入る。シンボルの種類により、表5に示す値が入る。

表5. データ長

名標の種類	値の意味
主構造体	構造体の大きさ
副構造体	副構造体の大きさ
配列	配列の要素の大きさ
ラベル (アセンブラ)	ラベルの付いた命令あるいはデータの大きさ
ラベル (コンパイラ)	0
その他	データの占める領域の大きさ

(4) 属性 (1 byte)

表6のように、シンボルの属性により値が入る。

表6. 属性

シンボルのタイプ	値
スカラー変数 ／主構造体名	1
副構造体名／構造体要素名	2
配列名／主構造体の配列名	3
副構造体配列名 ／構造体要素配列名	4
プログラムモジュール名 ／セクション名／手続き名	5
ラベル名	6
その他	未定義

(5) 配列の次元 (1 byte)

配列でない場合は0が、配列の場合はその次元数が入る。

(6) 配列長 (2 byte)

配列の場合、その配列が占める領域の大きさが、アドレス更新単位で入る。

#### 4.2.9 シンボル情報第2レコード

レコードタイプは8である。ヘッダ情報レコードは以下に示す情報から構成される。

(1) 第一／二／三添字の最大値 (2 byte x 3)

シンボルが配列名の場合、配列宣言時の値が入る。

(2) 主構造体名 (10文字)

シンボルが構造体の場合、そのシンボルが属する主構造体の名前が入る。

(3) 副構造体名 (10文字)

シンボルが構造体の場合、そのシンボルが属する直接外側の構造体名が入る。

(4) モジュール名 (8文字)

シンボルが属するモジュールの名前 (最も外側の手続き名) がはいる。

(5) 内部手続き名 (8文字)

シンボルが属する直接外側の手続き名が入る。

#### 4.2.10 行番号情報

レコードタイプは9である。ヘッダ情報レコードは以下に示す情報から構成される。

(1) 行番号 (4 byte)

ソースプログラム中の行番号が16進で入る。

(2) アドレス (4 byte)

行番号に対応するアドレスが入る。相対形式の場

合はセクション内相対アドレスが、絶対形式の場合は絶対アドレスが入る。

#### 4.2.11 トレーラ情報

レコードタイプは10である。ヘッダ情報レコードは以下に示す情報から構成される。

(1) 終了情報 (10文字)

固定文字、” / END OF \*\*\*\* ” が入る。  
\*\*\*\* は自由に設定可能 (4文字) である。

#### 4.3 通信規約

本節では、SYSROFを転送する場合の通信規約をしめす。

伝送コードは、アスキーコードによるヘキサ形式とする。ひとつのオブジェクトモジュールを転送する場合のフォーマットを以下に示す。

```
[CR][LF][SP].....[SP]
```

一回以上繰り返し

.

.

スペースの総数が100以上であること

```
[CR][LF][SP].....[SP]
```

```
[CR][LF]<   ロート   >
```

.

.

.

```
[CR][LF]<   ロート   >
```

```
[CR][LF][SP].....[SP]
```

一回以上繰り返し

.

.

スペースの総数が100以上であること

```
[CR][LF][SP].....[SP]
```

ここで、

[CR]=0DH

[LF]=0AH

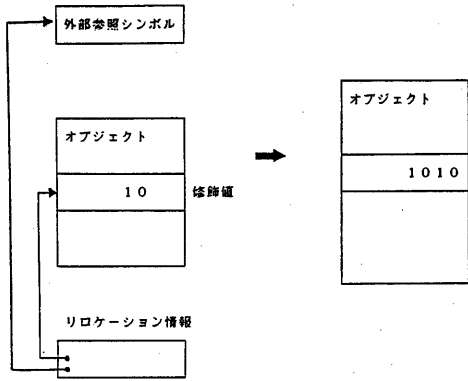
[SP]=20H

#### 4.4 リンク方式

本節では、リロケーション情報から絶対アドレスを計算する方法について述べる。

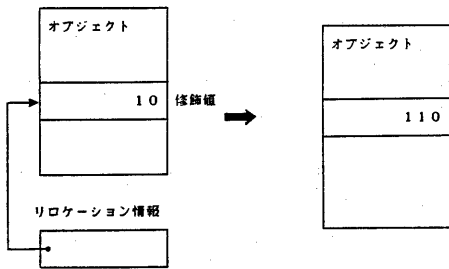
#### 4.4.1 リロケーション情報の表現方式

基本的な表現方法は、図2に示すようにセクション内部でのリロケーションと外部参照シンボルによ



外部参照シボルのアドレス：1000

(a) 外部参照シボルを参照する場合



セクション開始アドレス：100

(b) セクション内の情報だけによるリロケーション

図2. リンク方式

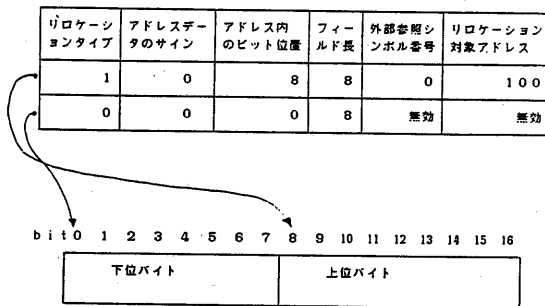


図3. 分割されている場合の表現方式

るリロケーションがある。前者は、リロケーション情報レコード内のリロケーション対象アドレス情報からオブジェクト内のリロケーション対象を見つけ出し、リロケーション対象アドレスに入っている修飾値(オフセット)とセクション開始アドレスを加えて絶対アドレスとする。後者は、リロケーション対象アドレスからオブジェクト内のリロケーション対象を見つけ出し、その修飾値と、外部参照シボル(外部参照シボル番号)のアドレスを知りそのアドレスに修飾値を加えた値が絶対アドレスとなる。

#### 4.4.2 リロケーション対象データが分割されている場合の表現方式と処理

リロケーション対象が複数に分割されている場合(例えば、バイト反転機種I8086等)の表現を示す。

リロケーション情報内のリロケーションタイプが、リロケーション対象アドレスがいくつかのフィールドに分割されているかを示している。例として、Z-80のようにアドレスが2つのフィールドに別れている場合の表現方法について示す。図3に示すように、最初のリロケーション情報のリロケーションタイプを見ることによりそのフィールドがいくつに分割されているか知ることができる。その分割数と同数のリロケーション情報が一塊となって分割のされかたを示す。

#### 5. 現状

図4に示すように、現在、電電公社の大型計算機DIPS上のマルチターゲットクロスコンパイラMAPS-C・μP用AdaとマルチターゲットクロスアセンブラPMP-AEが出力するOM(出力するOMフォーマットはすべて同一である)とSYSROFとの間での双方向のコンバータ(DLOAD、ULOAD)を試作した。また、

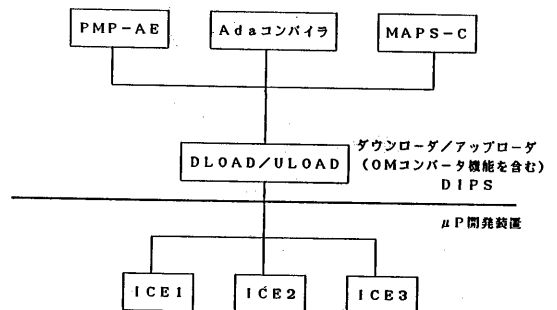


図4. 現在の接続状況

μP開発装置上のツールとしては、SYSROFを各種のフォーマットに変換するOMコンバータを2機種に、また、SYSROFリンカを1機種に試作中である。現在2400/4800bpsの回線を用いてダウンラインロードを行なっている。

### 6. 検討および今後の課題

(1) SYSROFを標準仕様とするためには、以下の点が問題となるので今後改善するように検討して行きたい。

①配列が3次元までしか考慮されていない。各次元の最大値はほとんど意味がないので削除することにより次元数に関係ないフォーマットとすることができる。

②属性の表現が、構造体、配列等の情報が分離していない、浮動小数点、ストリング等の詳細な属性表現がないという問題がある。

③ツール名が汎用的な意味を持よように定義する必要がある。

(2) シンボル情報第二レコードには、親の構造体名、手続き名があるが、それらが同一名の場合一意に決定できない場合がある。このため、名前だけでなく、アドレス情報も追加することにより、同一名でも一意に認識出来るようにする。

(3) 現在のSYSROFは、大型計算機においても処理がしやすいようにレコード内の各情報はワード境界で配置されるようにしてある。このため、必要以上に大きなエリアが取られている情報もある。大型計算機は処理能力が高くむしろダウンロードがボトルネックとなるので、転送量が小さくなるように、なるべく無駄な情報がない仕様とすべきであろう。

また、図5に示すように、リロケーション情報だけでOMの72%を占める。リロケーション情報のビットパターンは非常に似通っているため、より冗長性の少ない表現方式とすることができると思われる。

(4) 現在、当研究所では、マクロコンピュータ用マルチターゲットAdaを開発中であるが、動的に割り付けアドレスの変化する変数のデバッグ時の読み出し、あるいは、アセンブライメージではなくソースイメージでデバッグを行なうための情報追加が今後必要になってこよう。しかし、当面は、現状のμP開発装置のデバッグの機能がそれほど高くないことから、現在の仕様で十分と考えている。

### 7. おわりに

SYSROFとMUFOMは共に、機種独立性が高いOMフォーマットであるが、それぞれ特徴があり、そのフォーマットに適した使用環境があるように思われる。しかし、異なるμP開発装置上で同一OMによるデバッグという観点からは、MUFOMにおけるデバッグ情報の未定義部分は大きな障害となる。

前章に示したようにSYSROFにはまだ問題点が若干ある。しかし、OMフォーマットの標準化は、ソフトウェアに関する標準化の中で実現可能でかつメリットの高い残された数少ない分野の1つであり、上記の問題点を解決しつつ、関係方面の幅広い意見を聞きながら各種の標準化委員会等の場を通して標準化を働きかけて行く考えである。

### 謝辞

MUFOMについて貴重な御意見を頂いた、小原啓義早大教授ならびに山田剛助手に感謝いたします。

### <参考文献>

(1) 杉山、神谷、"ソフト流通をねらうμP用オブジェクトフォーマット標準化の試み、"情報処理全国大会大28回(昭和59年前期)。

(2) IEEE P695 Working Group, "The Micro-processor Universal Format for Object Modules-Proposed Standard," IEEE Micro vol.3, No.4, pp.48-66(1983,8)。

(3) 山田、内田、小原、"IEEE P695に準拠した汎用機械語開発ソフトウェア"、情報処理第32回マイクロコンピュータ研究会(昭和59年8月31日)。

(4) 山田、内田、小原、"オブジェクトモジュール標準形式IEEE-P695の応用と処理系"、情報処理学会第34回マイクロコンピュータ研究会(昭和59年12月18日)。

(5) 伊東、佐藤、長野、神谷、"総合ソフトウェア生産システムの実用化"、研究実用化報告、VOL.33, No.12, 2879-2894(昭和59年12月)。

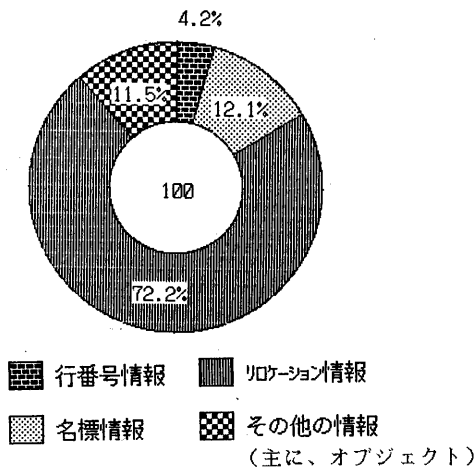


図5. SYSROFにおける各情報の占める割合



付録1.SYSRPROFにおける各レコードのフォーマット

\*:レコードタイプ  
#:レコード長  
&:チェックサム

0.ヘッダ-情報レコード

2	2	8	4	4	4	4	4	4	4	4	8
*	#	ファイ ル名	開始ア ドレス	プログ ラム長	プログラ ムタイプ	ツール 種別	アドレ ス単位	セクシ ョン数	作成 日付		

4 4 1 1バイト

CPU タイプ	セグメン トサイズ	セグメン トシフト	&
------------	--------------	--------------	---

1.セクション情報レコード

2	2	8	4	4	4	4	4	4	4	1	バイト	
*	#	セクシ ョン名	セグメン トアドレス	セクション アドレス	セクシヨ ンタイプ	境界 調整数	外部定義シ ンボル個数					

外部参照シ ンボル個数	リクエシヨ ン情報個数	外部入りロシ ンボル個数	オリジ ン情報個数	シンボ ル個数	セクシ ョン長	&
----------------	----------------	-----------------	--------------	------------	------------	---

2.外部参照シンボル情報レコード

2	2	8	1	バイト
*	#	外部参照 シンボル名	.....最大6回繰り返し	&

3.外部定義シンボル情報レコード

2	2	8	4	4	1	バイト
*	#	外部定義 シンボル名	外部定義シ ンボル タイプ	シンボル タイプ	.....最大2回繰り返し	&

4.外部入りロシンボル情報レコード

2	2	4	1	バイト
*	#	外部入りロ シンボル番号	.....最大13回繰り返し	&

5.リクエシヨン情報レコード

2 2 4 4 4 4 4 4

*	#	リクエシ ョンタイプ	アドレスデ ータのサイ ン	アドレ スのビット 位置	フィー ルド長	外部参照 シンボル番 号
---	---	---------------	---------------------	--------------------	------------	--------------------

4 1バイト

リクエシヨ ン対象アド レス	.....最大2回繰り返し	&
----------------------	---------------	---

6.オブジェクト情報レコード

2	2	4	(4)	(4)	最大51(43)	1	バイト
---	---	---	-----	-----	----------	---	-----

*	#	開始表示 フラグ	開始ア ドレス	終了ア ドレス	オブジェク トプログラ ム	&
---	---	-------------	------------	------------	---------------------	---

7.シンボル情報第一レコード

2	2	10	4	2	1	1	2	1	バイト
---	---	----	---	---	---	---	---	---	-----

*	#	シンボ ル名	アド レス	デー タ長	属性	配列の 次元	配列 長	&
---	---	-----------	----------	----------	----	-----------	---------	---

8.シンボル情報第二レコード

2	2	2	2	2	10	10	10	8	8	1	バイト
---	---	---	---	---	----	----	----	---	---	---	-----

*	#	第一送字 の最大値	第二送字 の最大値	第三送字 の最大値	主構造 体名	副構造 体名	モジ ュール名	内部手続 名	&
---	---	--------------	--------------	--------------	-----------	-----------	------------	-----------	---

9.行番号情報レコード

2 2 4 4 1バイト

*	#	行 番号	アド レス	.....最大6回繰り返し	&
---	---	---------	----------	---------------	---

10.トレーラ情報レコード

2 2 10 10 1バイト

*	#	終了情報 "/ENDOF****"	&
---	---	----------------------	---

