

連結編集過程を考慮したソフトウェア教育用ツール (TSE)[†]

的 場 裕 司^{††} 工 藤 英 男^{†††}
松 浦 敏 雄^{†††} 吉 岡 信 夫^{††††}

アセンブリ言語によるプログラミング教育については種々の方法が報告されているが、それらには原始プログラムのアセンブラによる処理からその実行までの過程における目的モジュールやロード・モジュールと呼ばれるものの性質やそれらを処理するプログラムである連結編集プログラムや絶対ローダの機能については考慮されていないようである。

そこで、我々は原始プログラム→アセンブラ→目的モジュール→連結編集プログラム→ロード・モジュール→絶対ローダ→実行という処理過程をはっきりさせ、より良い教育効果を得るために TSE (A Tool for Software Education) と呼ぶシステムを FACOM 230-45S 上で作成したので、その機能および利用結果について報告する。

1. はじめに*

最近マイクロコンピュータにも OS が利用されるようになり、原始プログラムの翻訳からその実行までの過程が利用者にとって大変便利になってきた。

しかし、それに伴いその過程の途中に現われる目的モジュールやロード・モジュールと呼ばれるものの性質や、それらを処理するために利用される連結編集プログラムや絶対ローダと呼ばれるソフトウェアの機能^{14), 15)} について具体的に経験し理解する機会が少なくなってきたようである。我々の周囲の学生をみてもこれらの事柄に関する知識があやふやであることが分かった。

現実の計算機でこれらの事柄が具体的に経験しにくい理由として以下のことが考えられる。

(1) 目的モジュールおよびロード・モジュールに関する情報が印字出力されない場合がほとんどである。

(2) これらのモジュールの形式がマニュアル等に示されていない場合がある。

(3) 計算機内部での情報の表現が2進(8進, 16進)数表現であるため印字出力されても理解しにくい。

そこで、我々は初心者に対して相対番地や外部記号の意味やサブルーチンへの引数の授受等が表現しやす

いことを考慮して、原始プログラムの表現にはアセンブリ言語を使用し、原始プログラムの翻訳からその実行までに関する基本的な概念を把握しやすくするために内部表現が10進数である擬似計算機を想定し、それを使用して教育効果を上げるための TSE (A Tool for Software Education) と呼ぶシステムを FACOM 230-45S 上で開発し、実際のプログラミング教育において使用したので以下に報告する。

なお、アセンブリ言語の教育用システムについては種々の報告がされているが^{2)-13), 16)}、これらには上記の各モジュールに関する処理過程は考慮されていないようである。

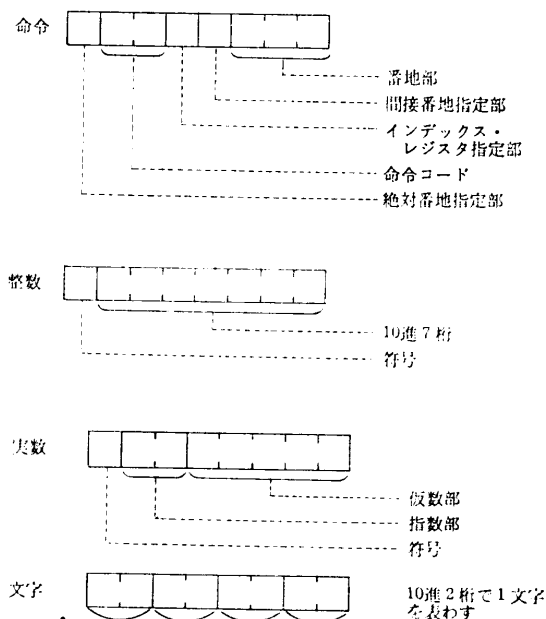


図1 語の構成

Fig. 1 Instruction and data formats.

[†] A Tool for Software Education in Consideration of the Linkage Editing Process by YUJI MATOBA (Faculty of Science, Konan University), HIDEO KUDO, TOSHIO MATSURA (Faculty of Engineering Science, Osaka University) and NOBUO YOSHIOKA (Osaka Institute of Technology).

^{††} 甲南大学理学部経営理学科

^{†††} 大阪大学基礎工学部情報工学科

^{††††} 大阪工業大学工学部電子工学科

* 本報告で用いた計算機関係の用語はほとんど文献¹⁾によるものである。

2. TSE の概要

2.1 擬似計算機の機能

TSE で使用する擬似計算機 (以下 PCOM と呼ぶ) の機能を以下に列挙する。

i) 語の構成: 10進8桁 (図 1 参照). ii) 主記憶容量: 1,000 語 (0~999 番地). iii) レジスタ: 10進8桁 (演算用レジスタ, インデックス・レジスタ, スタック・ポインタ). iv) プログラム・カウンタ: 10進3桁. v) 機械語命令: 42種 (付録 1 参照). vi) 文字コード: 50種 (付録 2 参照). vii) 割込み機能: 割込み発生番地がスタック・ポインタに示される番地に格納され, 付録 3 に示す割込みの種類に対応する番地へ自動的に分岐する。

2.2 TSE の構成

TSE は PCOM に基づき FACOM 230-45 S の OS II/VS の下で動作するシステムとして 図 2 に示す 4つのルーチンから構成され, その規模は PL/I で約 3,500 行から成りオーバレイ構造にして 104 kB である。以下, 各ルーチンの機能を簡単に説明する。

(1) TSE 制御ルーチン: 図 3 に示すカード・デックのうち%記号を含むカードが TSE に対する制御文であり, これを解析し各ルーチンの起動および必要な情報の受け渡しを行う。なお, 途中のルーチンで誤りを発見した場合には次の% JOB カードから処理される。

(2) アセンブラ・ルーチン: 付録 1 の PCOM 命令と付録 4 に示す擬似命令を用いて表現された複数個のセクション* で構成される原始プログラムを翻訳して複数個の目的モジュールを出力する。

(3) 連結編集ルーチン: 格納開始番地を得て複数個セクションから成る目的モジュールを連結編集処理を行いロード・モジュールを出力する。

(4) 実行ルーチン: ロード・モジュールを PCOM の主記憶装置上に格納し, 実行開始番地を得て格納されたプログラムに制御を渡す絶対ローダの処理と, このプログラムの各命令を PCOM の命令として解釈実行を行う。

2.3 目的モジュールとロード・モジュール

TSE の主目的は, 目的モジュールおよびロード・モジュールを印字出力することにより, ソフトウェア教育をより効果的に行うことである。したがって, そ

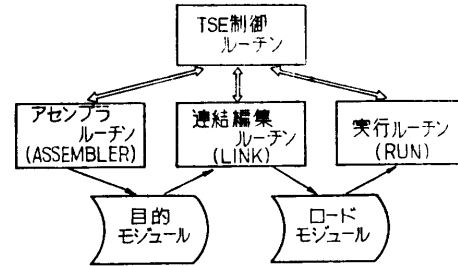


図 2 TSE の構成

Fig. 2 Configuration of TSE.

```

% JOB  使用者名,ACCT=V 会計コードV
% TSE  *
% JOB  学生番号
% ASSEMBLER  { [LIST,CROSS]
               [NOLIST,NOGROSS]
}
原始プログラム・カード
% LINK  { [LOAD=0,NOFLIST,NOMAP,NOELIST]
          [LOAD=番地,RLIST,MAP,ELIST]
}
% RUN  { [START=0]
         [START=実行開始番地,
          DUMP=ダンプ開始番地-終了番地,
          STEPDUAMP=開始番地-終了番地-間隔]
}
% DATA
データ・カード
% JEND
% JOB  学生番号
:
% JEND
:
% JEND
% JEND
  
```

(注) [] の中のパラメータは無指定のとき採用される

図 3 TSE のカード・デック構成

Fig. 3 TSE job card deck.

これらの表現形式をどのようにするかが重要なポイントとなり, 我々が採用した形式を 図 4 および 図 5 に示す。

目的モジュールは 図 4 のように 4つの部分で構成され, その内容は次のようになっている。

(1) セクション表示部: このモジュールが原始プログラムの何番目のセクションであるかを表わす。

(2) ENDT (Entry Directory Table) 部: 擬似命令 ENTRY で定義されているシンボルがどの番地に対応するかを表わし, シンボルごとに 2語を用いて表現され上位 1語にシンボル (4文字以内) の PCOM の文字コードが, 下位 1語にそのシンボルに対応する番地が入る。

(3) TEXT 部: PCOM の機械語表現のプログラムが格納される部分である。

(4) EXDT (External Directory Table) 部: 擬似命令 EXTRN で定義されているシンボルがどの位置で使用されているかを表わし, その形式は ENDT

* セクションとは原始プログラムを構成する単位で, 擬似命令 END で終わる命令の集合である。

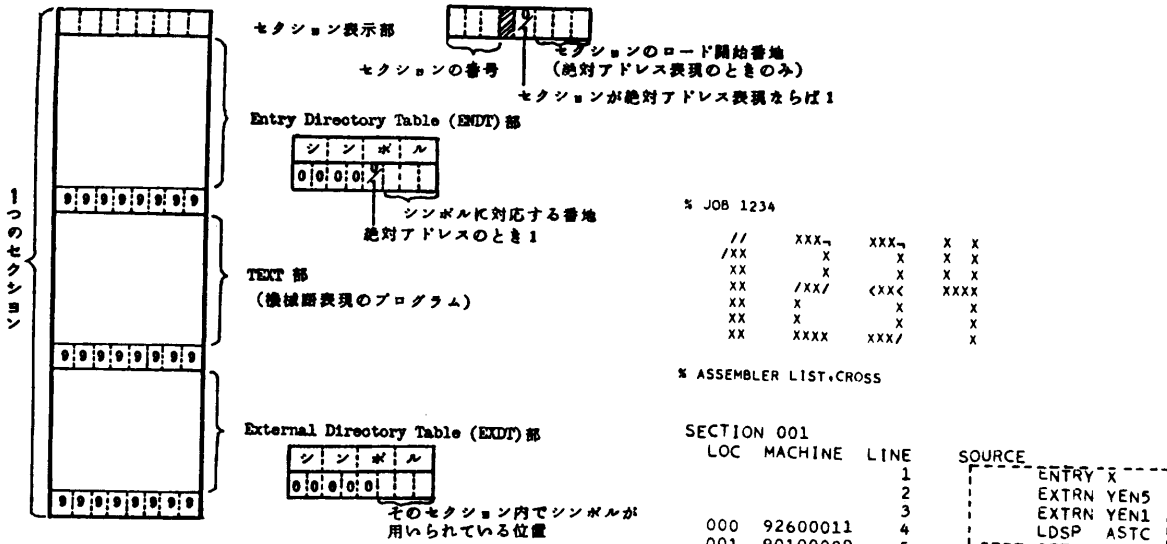


図 4 目的モジュールの形式
Fig. 4 Object module format.

部と似ているが、下位1語にはそのシンボルが使用されている TEXT 部の位置が入る。なお、99999999は上記の各部分の区切りを示すものである。これに対して、ロード・モジュール 図 5 に示すように単純な形であり、セクションの格納開始番地、格納すべき語数および内容という構成になっている。

3. TSE の使用例

3.1 例題とその説明

ここでは、あまり大きくない金額をデータとして読み、それが5円および1円硬貨のどのような組合せで表現されるかを計算するプログラムを示す。ただし、TSE の特徴を示すため外部記号を用い原始プログラムを2つのセクションで構成した。

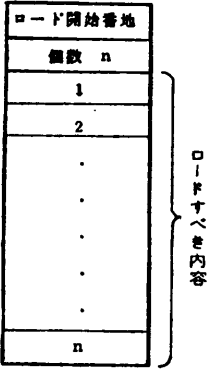


図 5 ロード・モジュールの形式
Fig. 5 Load module format.

```

% JOB 1234
// XXX~ XXX~ X X
/XX X X X X
XX X X X X
XX /XX/ <XX< XXXX
XX X X X
XX X X X
XX XXXX XXX/ X

% ASSEMBLER LIST,CROSS

SECTION 001
LOC MACHINE LINE SOURCE
1 ENTRY X
2 EXTRN YEN5
3 EXTRN YEN1
4 LDSP ASTC
5 STRT:GET X
6 LDJ C5
7 JSR YEN5
8 LDJ C1
9 JSR YEN1
10 BRN STRT
11 C5 :DEC 5
12 C1 :DEC 1
13 X :WRKS 1
14 STCK:WRKS 1
15 ASTC:ADR STCK
16 END

SYMBOL TABLE (CROSS)
X 009RG YEN5 *** F YEN1 *** E
STRT 001R C5 007R C1 008R
STCK 010R ASTC 011R
    
```

```

SECTION 002
LOC MACHINE LINE SOURCE
1 ENTRY YEN5
2 ENTRY YEN1
3 EXTRN X
4 YEN5:DA X
5 YEN1:STX N
6 DIV N
7 STA ANS
8 PUT ANS
9 MLT N
10 SUB X
11 MLT MONE
12 STA X
13 RTS
14 MONE:DEC -1
15 N :WRKS 1
16 ANS :WRKS 1
17 END

SYMBOL TABLE (CROSS)
YEN5 009RG YEN1 001RG X *** E
MONE 010R N 011R ANS 012R
    
```

図 6 アセンブル時の出力リスト
Fig. 6 Assembly list of an example program.

具体的な原始プログラムは 図 6 の点線で囲んだ部分である。セクション1において、シンボルXはこのセクション内で使用されるのみならず擬似命令 ENTRY によりセクション2においても共通領域として使用可能である。

% LINK LOAD=100,RLIST,MAP,ELIST

OUTPUT LIST FROM ASSEMBLER (RLIST)	目的モジュール									
SEQ.	0	1	2	3	4	5	6	7	8	9
0	00100000	24000000	00000009	99999999	92600011	90100009	92100007	04900000	92100008	04900000
10	93000001	00000005	00000001	00000000	00000000	90000010	99999999	25051465	00000004	25051461
20	00000006	99999999	00200000	25051465	00000000	25051461	00000001	99999999	00500000	92000011
30	91400011	90800012	90200012	91300011	01200000	91300010	00800000	05000000	10000001	00000000
40	00000000	99999999	24000000	00000001	24000000	00000007	24000000	00000009	99999999	

ENTRY SYMBOL TABLE (MAP) ENTRY 擬似命令で用いられたシンボルと
それに対応する番地の一覧表

SYMBOL	ADDRESS
X	109
YEN5	112
YEN1	113

EXTERNAL SYMBOL TABLE (MAP) EXTRN 擬似命令で用いられたシンボルが
実際に使用されている番地を示す一覧表

SECTION NO.	SYMBOL	ADDRESS
1	YEN5	103
1	YEN1	105
2	X	112
2	X	118
2	X	120

SECTION TABLE (MAP) 各セクションがロードされるべき番地を
示す一覧表

SECTION NO.	START	END
1	100	111
2	112	124

OUTPUT LIST FROM LINKER (ELIST) ロード・モジュール

SEQ.	0	1	2	3	4	5	6	7	8	9
0	00000100	00000012	02600111	00100109	02100107	04900112	02100108	04900113	03000101	00000005
10	00000001	00000000	00000000	00000110	00000112	00000013	00500109	02000123	01400123	00800124
20	00200124	01300123	01200109	01300122	00800109	05000000	10000001	00000000	00000000	

PROGRAM LIST ON MEMORY (ELIST) ロード・モジュールを
主記憶装置上にロードした状態

ADDRESS	0	1	2	3	4	5	6	7	8	9
100	02600111	00100109	02100107	04900112	02100108	04900113	03000101	00000005	00000001	00000000
110	00000000	00000110	00500109	02000123	01400123	00800124	00200124	01300123	01200109	01300122
120	00800109	05000000	10000001	00000000	00000000					

図 7 連結編集時の出力リスト

Fig. 7 Object and load modules of an example program.

また、シンボル YEN5 および YEN1 はこのセクション内で定義されていないが、それぞれ 5 円および 1 円硬貨の数を求めるサブルーチンの入口名として使用することを擬似命令 EXTRN により示している。

一方セクション 2 において、シンボル X はセクション 1 のものを使用することを擬似命令 EXTRN により示し、シンボル YEN5 および YEN1 はセクション 1 から使用可能とすることを擬似命令 ENTRY により示している。

以下では、この原始プログラムを 図 2 にそって実行可能なプログラム (ロード・モジュール) にしていく過程について説明する。

3.2 TSE による処理

前記の 2 つのセクションから成る原始プログラムが TSE のアセンブラにより処理されたときの出力リストを 図 6 に示す。

この例では、原始プログラムの先頭番地が指定されていないので 0 番地からの相対番地として処理されている。したがって、シンボルを用いている命令に対応する機械語表現では、最上位桁を 9 で表わすことによ

りその番地の再配置が必要であることを示している。

また、シンボル・テーブルにおいて、記号 ***E で表わすシンボルは擬似命令 EXTRN で定義される外部記号であり、これらが用いられている命令に対応する番地部は 000 となっている。記号 G が付けられているシンボルは擬似命令 ENTRY で定義される外部記号であり、ほかのセクションでも利用可能であることを示している。

次に、例題について TSE の連結編集プログラムにより処理された出力リストを 図 7 に示す。なお、図 6 と 図 7 を簡略化したものを 図 8 に示し、以下の説明ではこれを用いる。

(a) 欄は目的モジュールであり、その左側には対応する原始プログラムを付記してある。セクション 1 の ENDT 部には擬似命令 ENTRY で定義されたシンボル X に対応する PCOM の文字コード (24) と相対番地 (009) が入っている。TEXT 部には図 6 の機械語表現がそのまま入っている。EXDT 部には擬似命令 EXTRN により定義されたシンボル YEN5 および YEN1 に対応する PCOM の文字コード

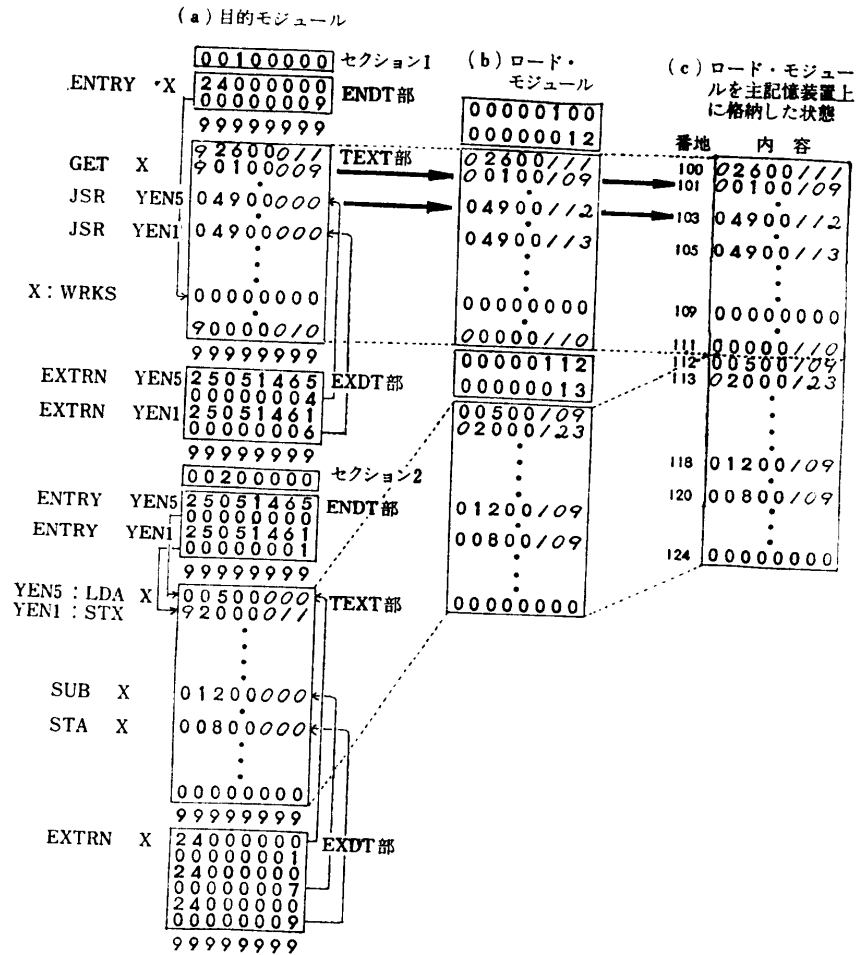


図 8 各モジュール間の関連
Fig. 8 Relation between modules.

(25051465 および 25051461) とそれらのシンボルが使用されている TEXT 部の位置 (4 番目と 6 番目) が入っている。

(b) 欄のロード・モジュールは TSE の連結編集プログラムにより、格納開始番地 (制御文で 100 番地が指定されている) と目的モジュールから生成されたものである。「ロードすべき内容」は目的モジュールの TEXT 部の各命令に対応する最上位桁や番地部が修正され完全な機械語として表現されている。なお、修正を受ける部分とそれに対応するところはイタリックで示してある。

たとえば, GET X の目的モジュールは 90100009 であるが, このセクションの格納開始番地が 100 番地であるので, ロード・モジュールでは 00100109 という形に再配置される。

また, JSR YEN5 の目的モジュールは 04900000

であるが, 格納開始番地 (100 番地) とセクション 1 の大きさ (12 語) より YEN5 の番地が 112 番地となり, ロード・モジュールでは 04900112 となる。

最後の (c) 欄は, ロード・モジュールが TSE の絶対ロードにより処理され PCOM の主記憶装置内に格納されたときの状態を示している。

なお, 図 7 に示されている各種のテーブル類は, 原始プログラム, 目的モジュールおよびロード・モジュールの関係を分かりやすくするために出力されている。

3.3 実施結果

情報工学科 2 年次を対象として, 昭和 54 年度後期のプログラミングの授業において TSE を用いた結果について述べる。

授業では TSE の説明を施しながら逐次具体的な課題について演習を行ったときの TSE の使用状況を表 1 に示す。なお, 課題の概要は, はじめ簡単なア

表 1 TSE の使用状況
Table 1 A TSE's actual records.

課 題 の 内 容	1 ジョブの 当カード枚数	1 当 り人 の ジョブ件数
a, b, x, y, のデータを読みある計算 をする	61	3.3
つるかめ算の処理	64	3.4
N個のデータを読みそれらの和を求める	49	3.6
数式の逆ポーリッシュ表現	160	12.0
初期値, 誤差を与えて平方根の近似計算	110	5.4
n! の計算でnがいくらのときオーバ ・フローするか (そのときメッセージ出力)	64	4.6
与えられた金額N円を現行の貨幣を用いて 表わせ	120	8.2
人数=44人 合 計	628	40.5

センブリ言語による演習を, 次に浮動小数点演算と割込み処理を, さらにマクロやサブルーチンを用いてプログラムを表現するものであった。

特に目的モジュールとロード・モジュールについては, 原始プログラムにおけるセクションの順序を変更したり, 連結編集プログラムにおいて格納開始番地を変更したりした場合の影響等を具体的に示すことができ教育上大変便利であるのみならず, 学生も興味を持って体験的に理解しているようである。

さらに, TSE を用いることにより3年次における実験の予備知識が修得でき, 計算機に関する教育・実験がより効果的に行うことが可能になった。

4. む す び

ここでは TSE と呼ぶソフトウェア教育用ツールを作成した。それはアセンブラ, 連結編集プログラム, 絶対ロードと呼ばれる各種のソフトウェア間での情報がどのような形式で授受されているかを具体的に示すことにより, 外部記号や再配置という概念を理解させることを目的としている。

本文ではその構成と使用例さらにプログラミングの授業において使用した結果について述べた。

TSE を授業に取り入れた経験では, 通常の計算機環境において利用者には原始プログラムしか見えないが, このような道具を用いることにより目的モジュールやロード・モジュールおよびそれらを処理するソフトウェアの存在や役割を意識するようになってきたようである。

なお, 現在のように計算機に関するハードウェア・ソフトウェア両面の進歩が激しい時代において, ここて述べた教育目的および方法が適当なものであるかどうかについては種々の意見があるものと思われるが, 情報工学科のように計算機を専門とする学生の教育には, TSE のような道具を用いることも適当な試みであると考えている。

また, TSE のアセンブラにはマクロ定義機能, 実行時には1ステップごとの実行やメモリ・ダンプ等の機能が含まれており, さらにエラー・メッセージがカナ文字で出力される等教育に適した多くの機能が組み込まれていることを追記しておく。

最後に, TSE の作成に際し御助力を頂いたコンピューターサービス(株)の高田司郎氏に感謝致します。

参 考 文 献

- 1) 日本ユニバック編: コンピュータ英和・和英辞典, 共立出版。
- 2) 島崎他: 情報工学科におけるプログラミング実習の一例, 情報処理学会論文誌, Vol. 21, No. 2, pp. 83-90 (1980).
- 3) McCharen, J.D.: A Description of a First Course in Assembler Language, SIGCSE, Vol. 6, No. 3, pp. 36-43 (1974).
- 4) Ballaben, G. and Ercoli, P.: Computer Aided Teaching of Assembler Programming, Computers in Education, North-Holland, pp. 217-221 (1975).
- 5) Donovan, J.J.: Tools and Philosophy for Software Education, CACM, Vol. 19, No. 8, pp. 430-436 (1976).
- 6) Haghese, C.E. and Pfeeger, C.P.: ASSIST: An Environment Simulator for IBM 360 Systems Software Development, IEEE Trans. on Software Engineering, Vol. SE-4, No. 6, pp. 526-530 (1978).
- 7) 樽美和幸, 宇津宮孝一, 荒牧重登: 計算機理解援助用シミュレータ SIM 960, 信学技報, Vol. 79, No. 78, ET 79-4, pp. 27-32 (1979).
- 8) 佐藤他: TSS によるアセンブリ言語教育用システム, 信学技報, Vol. 78, No. 112, ET 78-7, pp. 13-16 (1978).
- 9) 的場裕司, 吉岡信夫, 佐藤武久: プログラミング教育用擬似計算機システムについて, 情報処理, Vol. 17, No. 2, pp. 118-123 (1976).
- 10) Lawes, C.P.: Description of an Introductory Course in Programming Language, SIGCSE, Vol. 6, No. 3, pp. 69-78 (1974).
- 11) Sjoerdsma, T.: An Interactive Pseudo-Assembler for Introductory Computer Science, SIGCSE, Vol. 8, No. 1, pp. 342-349 (1976).

- 12) Albright, R. A. and Smith, B. D. : BRAINO: A Machine Language Simulator, SIGCSE, Vol. 10, No. 4, pp. 59-65 (1978).
- 13) Mavaddat, F. : An Experiment in Teaching Programming Languages, SIGCSE, Vol. 8, No. 3, pp. 45-59 (1976).
- 14) Donovan, J. J. : Systems Programming,

- McGraw-Hill (1972).
- 15) Barron, D. W. : Assemblers and Loaders, Macdonald (1972).
- 16) 阿草他: TSS の下でのアセンブリ言語および計算機構造の教育援助システム, 情報処理学会論文誌, Vol. 21, No. 6, pp. 482-490 (1980).
(昭和55年8月13日受付)
(昭和56年4月27日採録)

付 録

付録1 機械語命令一覧表(42種類)

命令 コード記号	機 能
00 HLT	ストップ
01 GET	データをmにセット
02 PUT	(m) を出力
03 FPUT	(m) を浮動小数点型で出力
04 CPUT	Accの値をnとすると、mからm+n/4番地の内容を文字型で出力
05 LDA	(m) → Acc
06 LDLA	(m [L4]) → Acc [L4]
07 LDRA	(m [R4]) → Acc [R4]
08 STA	(Acc) → m
09 STLA	(Acc [L4]) → m [L4]
10 STRA	(Acc [R4]) → m [R4]
11 ADD	(Acc) + (m) → Acc
12 SUB	(Acc) - (m) → Acc
13 MLT	(Acc) × (m) → Acc
14 DIV	(Acc) ÷ (m) → Acc
15 FADD	(Acc) + (m) → Acc
16 FSUB	(Acc) - (m) → Acc
17 FMLT	(Acc) × (m) → Acc
18 FDIV	(Acc) ÷ (m) → Acc
19 CMP	(Acc) = (m) なら 0 → Acc (Acc) ≠ (m) なら 1 → Acc
20 STX	(XR) → m
21 LDx	(m) → XR
22 INCX	(XR) + 1 → XR
23 ADDX	(XR) + (m) → XR
24 SUBX	(XR) - (m) → XR
26 LDSP	(m) → SP
27 STSP	(SP) → m
28 PUSH	(Acc) → (SP), (SP) - 1 → SP
29 POP	(SP) + 1 → SP, (SP) → Acc
30 BRN	m ヘブランチ (m → PC)
31 BMT	(Acc) < 0 なら m ヘブランチ
32 BPL	(Acc) ≥ 0 なら m ヘブランチ
33 BEQ	(Acc) = 0 なら m ヘブランチ
34 BNE	(Acc) ≠ 0 なら m ヘブランチ
37 BZX	(XR) = 0 なら m ヘブランチ
38 BNZ	(XR) ≠ 0 なら m ヘブランチ
40 SHRL	(Acc) を右シフト (含符号)
41 SHLL	(Acc) を左シフト (含符号)
42 SHR	(Acc) を右シフト (除符号)
43 SHL	(Acc) を左シフト (除符号)
49 JSR	(PC) → (SP), (SP) - 1 → SP, m ヘブランチ
50 RTS	(SP) + 1 → SP, (SP) ヘブランチ

(X) : X の内容を意味する
 m : 実際の番地を意味する
 Acc : 演算レジスタ
 XR : インデックス・レジスタ
 SP : スタック・ポインタ
 PC : プログラム・カウンタ
 [L4] : 左側4けたの部分
 [R4] : 右側4けたの部分

付録2 類似計算機用文字コード一覧表(50種類)

	上位けた				
	0	1	2	3	6
0	J	T	(0
1	A	K	U)	1
2	B	L	V	,	2
3	C	M	W	.	3
4	D	N	X	^	4
5	E	O	Y	;	5
6	F	P	Z	:	6
7	G	Q	+	&	7
8	H	R	-	@	8
9	I	S	*	=	9

付録3 内部割込み一覧表

割込みの種類	番地
データ・エラー	0
Accのオーバー・フロー	1
Accのアンダー・フロー	2
0による除算	3
不正命令	4
PCのオーバー・フロー	5

付録4 類似命令一覧表(14種類)

記号	機 能
ORG	以下の命令をセットする番地を指定する
LIST	マクロ展開したプログラムのリストを出す
NLIST	マクロ展開しない
ENTRY	ENTRYシンボルの定義
EXTERN	EXTERNALシンボルの定義
DEC	/0進整数をセットする
CHAR	文字定数をセットする
WRKS	作業用番地を指定した個数だけ確保する
ADR	アドレス定数を定義する
MACRO	マクロ定義の開始
IFT	条件付きマクロ展開(真)
IFF	条件付きマクロ展開(偽)
MEND	マクロ定義の終了
END	プログラムの終了を示す