

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日  
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

$\mu$ PD7550,  $\mu$ PD7560シリーズ  
アセンブラ

PC-9800シリーズ(MS-DOS™)ベース  
IBM PC™(PC-DOS™)ベース

**$\mu$ PD7550,  $\mu$ PD7560シリーズ**  
**アセンブラ**

**PC-9800シリーズ(MS-DOS™)ベース**  
**IBM PC™(PC-DOS™)ベース**



- 文書による当社の承諾なしに本資料の転載複製を禁じます。
  - 本資料に記載された製品の使用もしくは本資料に記載の情報の使用に際して、当社は当社もしくは第三者の知的所有権その他の権利に対する保証または実施権の許諾を行うものではありません。上記使用に起因する第三者所有の権利にかかわる問題が発生した場合、当社はその責を負うものではありませんのでご了承ください。
  - 当社は品質、信頼性の向上に努めていますが、半導体製品はある確率で故障が発生します。当社半導体製品の故障により結果として、人身事故、火災事故、社会的な損害等を生じさせない冗長設計、延焼対策設計、誤動作防止設計等安全設計に十分ご注意ください。
  - 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定して頂く「特定水準」に分類しております。また、各品質水準は以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認の上ご使用願います。
    - 標準水準：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
    - 特別水準：輸送機器（自動車、列車、船舶等）、交通用信号機器、防災／防犯装置、各種安全装置、生命維持を直接の目的としない医療機器
    - 特定水準：航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等
- 当社製品のデータ・シート／データ・ブック等の資料で、特に品質水準の表示がない場合は標準水準製品であることを表します。当社製品を上記の「標準水準」の用途以外でご使用をお考えのお客様は、必ず事前に当社販売窓口までご相談頂きますようお願い致します。
- この製品は耐放射線設計をしておりません。

M4 94.11

本製品は外国為替および外国貿易管理法の規定により戦略物資等（または役務）に該当しますので、日本国外に輸出する場合には、同法に基づき日本国政府の輸出許可が必要です。

- 本資料の内容は、後日変更する場合があります。
- 文書による当社の承諾なしに本資料の転載複製を禁じます。
- この製品を使用したことにより、第三者の工業所有権等にかかわる問題が発生した場合、当社製品の構造製法に直接かかわるもの以外につきましては、当社はその責を負いませんのでご了承ください。



ま え が き

1. μPD7550,7560シリーズ・アセンブラ取扱説明書は、次の1チップ・4ビット・マイクロコンピュータに対応しています。

μPD7550,7560シリーズ・アセンブラ

- μPD7500セットA
- μPD7500セットB
- μPD7554
- μPD7556
- μPD7564
- μPD7566

2. システム構成

(1987年4月現在)

ホスト・コンピュータ		OS	メモリ・サイズ
シリーズ名	対象機種		
PC-9800	PC-9801 E	(注3) MS-DOS バージョン 2.11 3.1	256Kバイト 以上
	PC-9801 F1/F2/F3 (注1)		
	PC-9801 M2/M3		
	PC-9801 VF2 (注1)		
	PC-9801 VM0/VM2/ VM4/VM21		
	PC-9801 VX0/VX2/ VX4 (注2)		
	PC-98 XL (注2)		
IBM PC	IBM PC	PC-DOS	
	IBM PC/XT™	バージョン 3.10	

注1 8'2Dまたは5'2HDの増設ドライブをご用意ください。

注2 V30™モードに設定の上、ご使用ください。

注3 μPD7550, μPD7560シリーズアセンブラのプログラムは、弊社提供のPC-9800シリーズ用MS-DOS上にて動作いたします。

このほかの市販MS-DOS上における本プログラムの動作については、その責を負いかねますので、ご了承ください。

なお、動作環境(ホスト・コンピュータの対象機種、OSのバージョンなど)の最新情報につきましては「NEC マイクロコンピュータ・ユーザーズ・ガイド」をご参照ください。



注意

μPD7550, μPD7560シリーズアセンブラのプログラムを実行する場合、MS-DOSのシステムにCONFIG.SYSファイルが存在し、CONFIG.SYSファイル内で、オープン可能なファイル数(FILE)が8以上に設定されている必要があります。

3. アセンブラの供給形態

(1) ファイル名

AS75M.COM

AS75M.OM0

(2) フロッピー・ディスクの形式

両面倍密度8インチ・フロッピー・ディスク(8<sup>1</sup>/<sub>2</sub>D)または  
高密度5インチ・フロッピー・ディスク(5<sup>1</sup>/<sub>2</sub>HD)

IBM PC<sup>TM</sup>, PC-DOS<sup>TM</sup>, IBM PC/XT<sup>TM</sup> は、米国IBM社の商標です。

MS-DOS<sup>TM</sup>は、米国マイクロソフト社の商標です。

V30<sup>TM</sup>は、日本電気株式会社の商標です。

## 目 次

<b>第1章 ソフトウェア概要</b> .....	1-1
1.1 アセンブラの機能概要 .....	1-1
1.2 パス方式 .....	1-2
1.3 クロスレファレンス・リストの処理 .....	1-2
<b>第2章 ソース・プログラム様式</b> .....	2-1
2.1 文の構成 .....	2-1
2.2 文 字 .....	2-1
2.3 シンボル欄 .....	2-2
2.4 ニモニック欄 .....	2-4
2.5 オペランド欄 .....	2-4
2.5.1 オペランド欄の記述形式 .....	2-4
2.5.2 オペランド欄記述上の注意点 .....	2-12
2.6 コメント欄 .....	2-13
2.7 タビュレーション機能 .....	2-13
<b>第3章 疑似命令</b> .....	3-1
3.1 ロケーション・カウンタ制御疑似命令 .....	3-1
3.1.1 ORG(ORIGIN) .....	3-1
3.2 シンボル定義疑似命令 .....	3-2
3.2.1 EQU(EQUATE) .....	3-2
3.2.2 SET .....	3-3
3.3 データ定義疑似命令 .....	3-4
3.3.1 DB(Define Byte of data) .....	3-4
3.4 サブルーチン・エントリ・テーブル定義疑似命令 .....	3-4
3.4.1 DET(Define Subroutine Entry Table) .....	3-4
3.5 条件付アセンブル疑似命令 .....	3-7
3.5.1 IF と ELSE と ENDIF .....	3-7
3.6 アセンブル終了疑似命令 .....	3-8
3.6.1 END .....	3-8
3.7 アセンブル制御疑似命令 .....	3-9
3.7.1 TITLE .....	3-9
3.7.2 EJECT .....	3-9
3.7.3 SPACE .....	3-9
3.7.4 PRINT .....	3-10
3.8 JUMP系命令の自動選択疑似命令 .....	3-10
3.8.1 GJMP .....	3-10

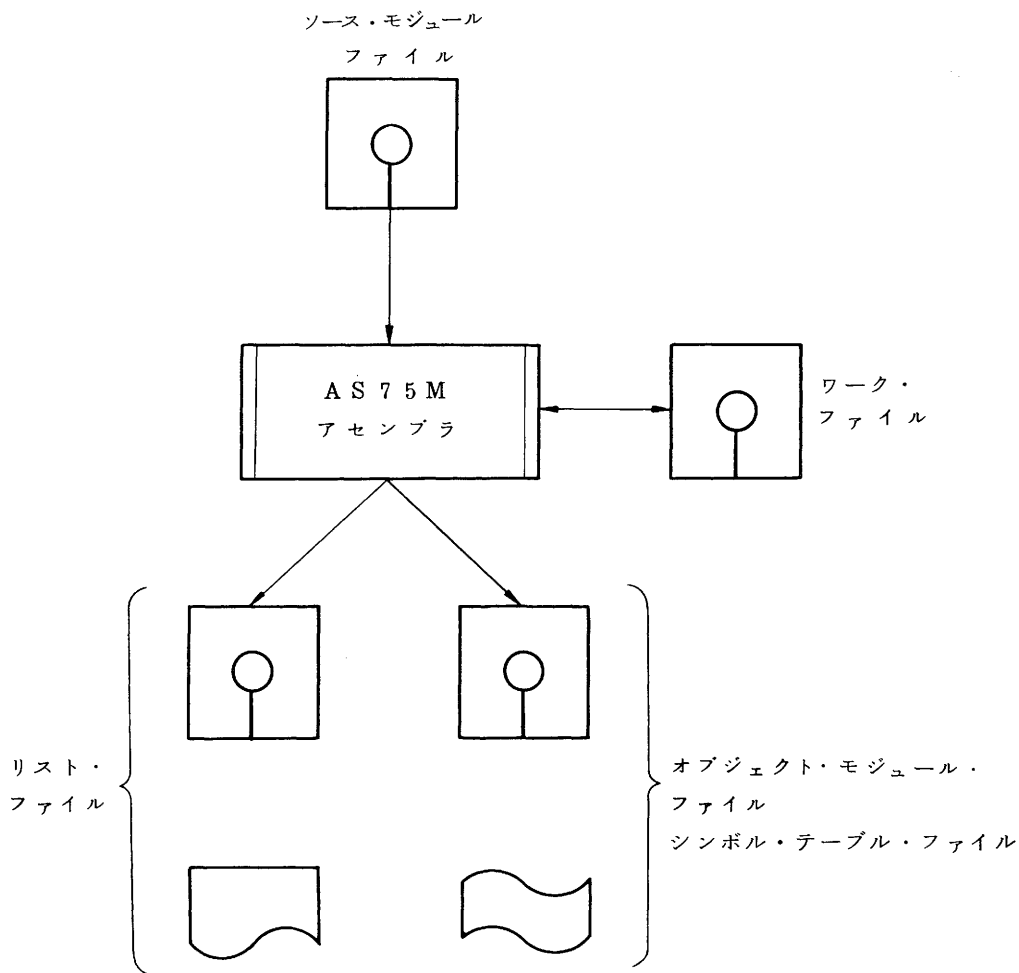
<b>第4章</b>	<b>アセンブラの操作方法</b> .....	4-1
4.1	アセンブラの起動方法 .....	4-1
4.1.1	ソース・モジュール・ファイル名とコントロールを指定する方法 .....	4-1
4.1.2	コンソールに表示されるメッセージをファイルに出力させる方法 .....	4-3
4.2	アセンブラ・タイトルの出力 .....	4-4
4.3	コントロール .....	4-4
4.3.1	入力形式 .....	4-4
4.3.2	ファイル名規則 .....	4-4
4.3.3	コントロールの種類 .....	4-5
4.4	アセンブル実行開始メッセージの出力 .....	4-1 4
4.5	アセンブル終了メッセージの出力 .....	4-1 4
4.6	入出力ファイル・リストの出力 .....	4-1 4
4.7	アセンブラの操作例 .....	4-1 5
<b>第5章</b>	<b>出力形式</b> .....	5-1
5.1	オブジェクト・モジュール・ファイル .....	5-1
5.2	出力リスト .....	5-3
5.2.1	アセンブル・リスト .....	5-3
5.2.2	エラー・リスト .....	5-6
5.2.3	シンボル・リスト .....	5-6
5.2.4	クロスレファレンス・リスト .....	5-7
5.3	アセンブル・リストに印字されるエラー・メッセージ .....	5-8
5.4	アセンブル・リストに印字されるエラー・メッセージ一覧 .....	5-1 0
5.5	コンソールに表示されるエラー・メッセージ .....	5-1 1
付録 I	予 約 語 .....	付-1
付録 II	シンボル数 .....	付-6
付録 III	インストラクション .....	付-7

# 第1章 ソフトウェア概要

## 1.1 アセンブラの機能概要

本アセンブラはμPD7550, 7560シリーズの命令とアセンブラを制御する疑似命令を, 絶対番地形式のオブジェクト・プログラム(マシン・コード)に変換します。

アセンブラはユーザの作成したソース・モジュール・ファイルを入力し, オブジェクト・モジュール・ファイルとリスト・ファイルを出力します。



## 1.2 パス方式

本アセンブラは3 PASSアセンブラです。

### (1) PASS-1

ソース・モジュールを読みシンボル・テーブルを作成します。PASS-1はコントロール指定(4.3参照)にかかわらず実行されます。コントロール指定でSYMBOLSを指定するとシンボル・リストをリスト・ファイルに出力します。JUMP系命令の最適化オプションが指定された場合は、最適化パス(4.3.3の(5)参照)が行われます。

### (2) PASS-2

ソース・モジュールを読みコントロール指定に対応してアセンブル・リスト、エラー・リスト、クロスレファレンス・リストをリスト・ファイルに出力します。

### (3) PASS-3

OBJECTの指定によりソース・モジュールを読みHEXオブジェクトをオブジェクト・モジュール・ファイルに出力します。

## 1.3 クロスレファレンス・リストの処理

本アセンブラはクロスレファレンス情報をフロッピー・ディスク(FD)上に出力して処理します。このため次の名称を持つ3個のファイルが作業用に指定されたユニット上に作られ、処理が終了すると消去されます。

- (1) ソース・ファイルのプライマリ・ネーム . \$ \$ 1
- (2) ソース・ファイルのプライマリ・ネーム . \$ \$ 2
- (3) ソース・ファイルのプライマリ・ネーム . \$ \$ 3



- ' → 文字定数を定義します。
- : → レーベルをターミネートします。
- ; → コメント欄の最初に必要です。
- △およびHT → ニモニック欄とオペランド欄およびシンボル欄（ネームのとき）とニモニック欄を分離します。

上記以外の文字は使用できません。

英小文字を使用する場合、コメント欄、文字定数を除き、英小文字を使用しても英大文字と解釈されます。

## 2.3 シンボル欄

シンボル欄にはレーベルまたはネームを記述します。

レーベルには、その付けられた命令および疑似命令に割り付けられたアドレス（ロケーション・カウンタの値）がそのまま割り付けられます。また、ネームには、EQU命令、SET命令のオペランドに記述した数値データやアドレスが割り付けられます。

シンボル記述上の規則は次のとおりです。

- (1) シンボルは英数字により構成します。ただし、先頭文字は、英文字でなければなりません。
- (2) シンボルの長さは1～6文字です。シンボル欄には6文字以上記述できますが、先頭の6文字のみが有効となります。
- (3) レーベルはコロン(:)で、ネームは空白でターミネートします。
- (4) 同一シンボルを2度以上定義することはできません（ただし、SET命令のネームを除く）。
- (5) 予約語をシンボルに使用することはできません（予約語については付録I参照）。

例1 正しい例	誤った例
F1F4:	1F4F: ……数字で始まっています。
LABEL:	LABEL ……コロンが付いていません。(レーベルとして使用する場合)
HERE:	HE RE: ……レーベル中に空白があります。
ADDA1:	ASC: ……命令(予約語)は使用できません。
ENDX:	END: ……疑似命令(予約語)は使用できません。

例2 INSTRUCTION: → INSTRU: と解釈されます。

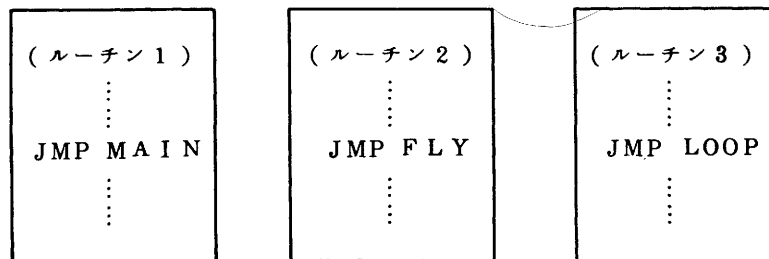
例3 ABC EQU 3                      ABCとXYZには同じデータ  
 XYZ EQU ABC                      “3”が割り付けられます。

```

例4 LOOP: LHI 5
      :
      :
      LOOP: DLS
      :
      :
      JMP LOOP
    
```

レーベル“LOOP”が2重定義されていますのでエラーとなります。

例5 レーベルの特殊な記述法として、同一番地に2個以上のレーベルを指定することができます。



上記のように個別に作成されたいくつかのルーチンがあり、しかもそれぞれの飛び先はプログラムの論理上同じ番地であると仮定します。

そのような場合、次のように記述しますと、それぞれのレーベルを統一することなく同一番地とすることができます。

```

(1) MAIN:      ;FROM ROUTINE1
      FLY:      ;FROM ROUTINE2
      LOOP: LIS ;FROM ROUTINE3
      :
      :
    
```

または

```

MAIN: FLY: LOOP: LIS
    
```

```

(2) MAIN:      ;FROM ROUTINE1
      FLY:      ;FROM ROUTINE2
      LOOP:     ;FROM ROUTINE3
      LIS
      :
    
```

または

```

MAIN: FLY: LOOP:
    
```

```

LIS
    
```

以上(1), (2)の例のように、それぞれのレーベルは参照時に同じ値として評価され、ソース・プログラムの編集上大変有効な機能となります。



## 2.4 ニモニク欄

ニモニク欄には、機械語命令、疑似命令を記述します。

オペランドの必要な命令の場合、ニモニク欄とオペランド欄を区別するために、1つ以上のブランクが必要です。

例 正しい例	誤った例
JMP 24H	JMP24H .....ニモニクとオペランドの間にブランクがありません。
LDS	L DS .....ニモニク中にブランクが挿入されています。
RTS	RETS .....μPD7550, 7560シリーズの命令にRETSはありません。

## 2.5 オペランド欄

オペランド欄には命令で実行されるのに付随したコードを書きます。命令によりオペランド欄の必要ないものもあり、シングル・オペランド、ダブル・オペランドを必要とするものもあります。

ダブル・オペランドの場合、各オペランドは、“,”で区切ります。

### 2.5.1 オペランド欄の記述形式

オペランド欄の記述形式には、次のように6種類の形式があります。

#### (1) 定数 (Constant)

定数はそれ自身で定まる値を持つもので、数字や文字の列で構成します。

数字だけで構成するものを数値定数、文字で構成するものを文字定数といいます。

数値定数には、2進定数、8進定数、10進定数、16進定数があります。

##### (a) 2進定数

2進数字の最後に文字“B”を付加して表現します。

例 1011B

##### (b) 8進定数

8進数字の最後に文字“O”または文字“Q”を付加して表現します。

例 73O  
73Q

##### (c) 10進定数

10進数字の最後に文字“D”を付加するか、あるいは何も付加せずに表現します。

例 927  
927D

(d) 16進定数

16進数字の最後に文字“H”を付加して表現します。先頭の文字が0～9以外の文字で始まる場合には先頭に“0”を付加します。

例 9CH  
0ABH

(2) 文字定数

文字定数は、ASCII文字を引用符(‘)で囲んだものです。

引用符で囲まれたASCII文字はアセンブルの結果、パリティ・ビットを0とした7ビットASCIIコードに変換されます。2.2節で規定された文字を使用できますので、空白も文字定数としてそれぞれのASCIIコードに変換されます。

(ただし、HTコード(09H)は空白のASCIIコード1個に変換され、リスト上には相当数の空白が出力されます。)

引用符を文字定数として確保する場合は、引用符を2個続けます。

例 ‘A’-----41H  
 ‘ ‘-----27H  
 |  
 |-----1個の引用符が定数として確保されます。  
 ‘A’‘ ‘-----4127H  
 ‘ ‘-----20H  
 ‘<’-----203CH

(3) レジスタ名

各製品に対して、次のようなペア・レジスタがあります。

製品群	ペア・レジスタ名
7500セットA	HL
	HL+ (オート・インクリメント)
	HL- (オート・デクリメント)
	DE
	DL
7500セットB	HL
	HL+ (オート・インクリメント)
	HL- (オート・デクリメント)
	7564
	7566

注意1 これらのレジスタ名を式(6参照)の中に用いることはできません。

例 LAM HL+1 …… HL+1 と記述できません。

注意2 HL+をHL +のようにHLと+の間に空白を入れてはいけません。HL- についても同様です。

(4) \$ (ロケーション・カウンタ)

“\$”はロケーション・カウンタの値を示します。即ち、\$をオペランドに記述した命令に割付けられたアドレス(複数バイト命令の場合は1バイト目)を示します。

例 アドレス

```

100          XAL
101  LOOP:  DDRS  10H
103          JMP   $-2
105          JMP   $+100H
    
```

JMP \$-2の場合の\$は103H番地を、またJMP \$+100Hの場合の\$は105H番地を示します。JMP \$-2はJMP 101HあるいはJMP LOOPとしても同じ動作をします。

(5) シンボル

シンボルをオペランド欄に記述した場合は、そのシンボルに割付けられたアドレス(またはデータ)をオペランドの値として確保します。

```

例  HERE:    JMP  THERE
      ⋮
      THERE:  IDRS 10H
    
```

```

例  VALUE EQU  8
      LAI  VALUE
    
```

LAI VALUEはLAI 8と同じ意味を持ちます。

(6) 式(Expression)

前記の定数、\$, シンボルを演算子により結合したオペランドを“式”と呼びます。演算子は14種類(+,-,\*,/,NOT,AND,OR,XOR,EQ,NE,SHR,SHL,MOD,一符号)あり、演算実行上の優先順位が定められています。演算順位の変更には、カッコ“(”,“)”を使用します。

“式”に記述された演算は15ビットのデータを扱い、演算結果も15ビットで示されます。ただし、演算結果が命令の要求する範囲を越えている場合はエラーとなります。

(a) + (Addition)

オペランド間の加算を表します。

例	<u>アドレス</u>	<u>シンボル</u>	<u>ニモニック</u>	<u>オペランド</u>
	0010	START:	JMP	\$+6

JMP命令により16H番地へジャンプします。

(b) - (Subtraction)

オペランド間の減算を表します。

例	<u>アドレス</u>	<u>シンボル</u>	<u>ニモニック</u>	<u>オペランド</u>
	0020	BACK:	JMP	BACK-6

JMP命令実行により1AH番地へジャンプします。BACK-6は\$-6と同じ意味を持ちます。

(c) \* (Multiplication)

オペランド間の乗算を表します。

例	<u>シンボル</u>	<u>ニモニック</u>	<u>オペランド</u>
	X6:	LAI	2*3

LAI命令実行により6(2×3=6)がAレジスタにロードされます。

(d) / (Division)

オペランド間の除算を表します。

除算結果の小数点以下は切り捨てます。

例	<u>シンボル</u>	<u>ニモニック</u>	<u>オペランド</u>
	Y5:	LAI	256/50

LAI命令実行により5(256÷50=5.12)がAレジスタにロードされます。

(e) NOT

NOTに続くオペランドの各ビットを反転します。

NOTとオペランドの間にはブランクが必要です。

例	<u>シンボル</u>	<u>ニモニック</u>	<u>オペランド</u>
	COMPL:	LAI	NOT 3H AND 0FH

演算は次のように行われます。

```

NOT 3H=111 1111 1111 1100
AND ) 0FH=000 0000 0000 1111
-----
      0CH=000 0000 0000 1100
    
```

この結果、0CHがAレジスタにロードされます。

(f) AND

オペランド間で各ビットの論理積を求めます。

ANDの前後にはブランクが必要です。

例	<u>シンボル</u>	<u>ニモニック</u>	<u>オペランド</u>
	MASK:	LAI	6FAH AND 0FH

演算は次のように行われます。

```

6FAH=000 0110 1111 1010
AND ) 0FH=000 0000 0000 1111
-----
      0AH=000 0000 0000 1010
    
```

この結果、0AHがAレジスタにロードされます。

(g) OR

オペランド間で各ビットの論理和を求めます。

ORの前後にはブランクが必要です。

例	<u>シンボル</u>	<u>ニモニック</u>	<u>オペランド</u>
	MDFY1:	LAI	0AH OR 1101B

演算は次のように行われます。

```

0AH=000 0000 0000 1010
OR ) 000 0000 0000 1101
-----
      0FH=000 0000 0000 1111
    
```

この結果、0FHがAレジスタにロードされます。

(h) XOR ( Exclusive OR )

オペランド間で各ビットの排他的論理和を求めます。

XORの前後にはブランクが必要です。

例    シンボル            ニモニック            オペランド  
MDFY 2:            LAI                    9AH XOR 9DH

演算は次のように行なわれます。

9AH=	000	0000	1001	1010
XOR	) 9DH=	000	0000	1001
		1101		
		7H=	000	0000
			0000	0111

この結果, 7HがAレジスタにロードされます。

(i) EQ (Equal)

第1オペランドと第2オペランドの論理比較を行い, 等しければ0001H, 等しくなければ0000Hとします。EQと前後のオペランドとの間には空白が必要です。

例    A1 EQ A2

これは

A1=001 0010 1100 0100  
A2=001 0010 1100 0100

ならば

A1 EQ A2=000 0000 0000 0001

となります。

もしA1の値は変らずに

A2=100 1101 1100 0100

ならば

A1 EQ A2=000 0000 0000 0000

となります。

(j) NE (Not Equal)

第1オペランドと第2オペランドとの論理比較を行い, 等しくなければ0001H, 等しければ0000Hとします。NEと前後のオペランドとの間には空白が必要です。

例    A1 NE A2

ここで,

A1=001 0010 1100 0100  
A2=100 1101 1100 0100

ならば,

A1 NE A2=000 0000 0000 0001

となります。

もしA1の値が変らずに,

A2=001 0010 1100 0100

ならば,

A1 NE A2=000 0000 0000 0000

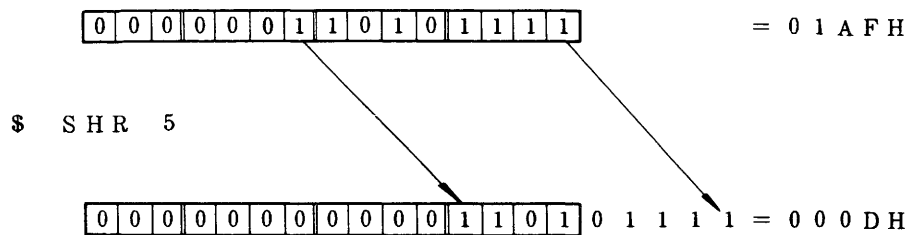
となります。

(k) SHR (Shift Right)

第1オペランドを第2オペランドで示された値だけ右シフトします。この場合、上位ビットにはシフトされたビット数だけ0が挿入されます。SHRと前後のオペランドとの間には空白が必要です。

例	アドレス	シンボル	ニモニック	オペランド
	01AF	FIELD:	LAI	\$ SHR 5

オペランドの意味は——\$で示されるアドレス(01AFH)を右に5ビット・シフト——です。演算は次のように行われます。



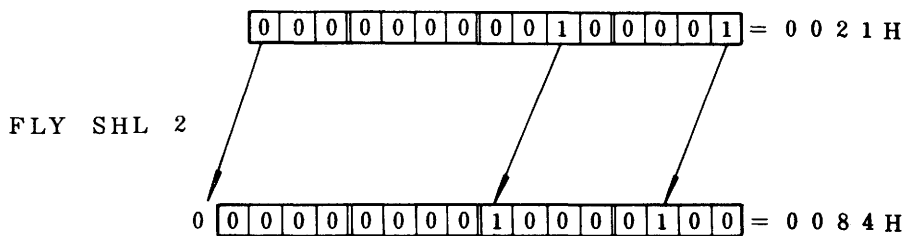
この結果、0DHがAレジスタにロードされます。

(l) SHL (Shift Left)

第1オペランドを第2オペランドで示された値だけ左シフトします。この場合、下位ビットにはシフトされたビット数だけ0が挿入されます。SHLと前後のオペランドとの間には空白が必要です。

例1	アドレス	シンボル	ニモニック	オペランド
	0021	FLY:	JMP	FLY SHL 2

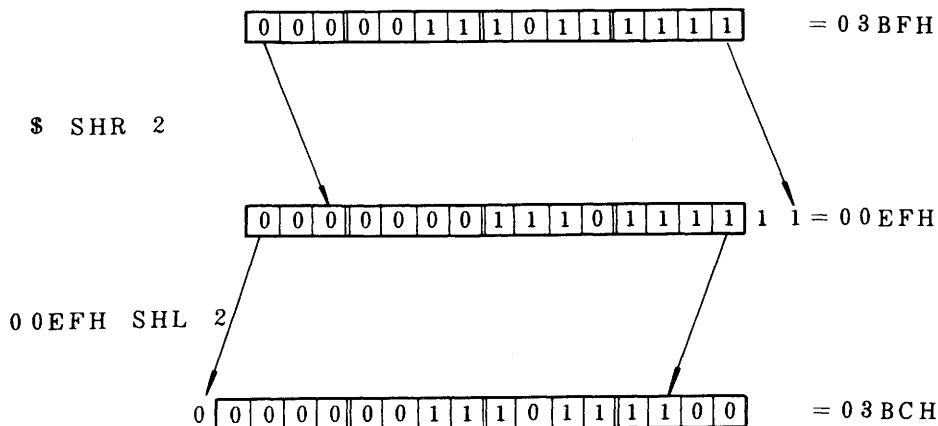
オペランドの意味は——FLYで示されるアドレス(0021H)を左に2ビット・シフト——です。演算は次のように行われます。



この結果、0084H番地にジャンプします。

例2 アドレス      シンボル                      ニモニック                      オペランド  
 03BF    LSB2:                      JMP        \$ SHR 2 SHL 2

オペランドの意味は——\$で示されるアドレス(03BFH)を右に2ビット・シフトし、その結果を左に2ビット・シフトする——です。演算は次のように行われます。



この演算は、アドレスを右に2ビット・シフトし、その後左に2ビット・シフトして元に戻したものですが、右2ビット・シフトで下位2ビットが消滅しますので次のように下位2ビットのマスクと同じ動作となります。

アドレス              シンボル                      ニモニック                      オペランド  
 03BF              LSB2:                      JMP              \$ AND 07FFCH

(m) MOD (Modulo)

オペランド間の除算を行い、その余りを求めます。

MODの前後には空白が必要です。

例 シンボル              ニモニック                      オペランド  
 REM6:              LAI                      256 MOD 50

LAI命令実行により、256を50で割った結果の余り6がAレジスタにロードされます。

(n) 一符号

オペランドの値の2の補数をとります。

例 シンボル                      ニモニック                      オペランド  
 MINUS:                      LAI                      -5 AND 0FH

“5”の2の補数がとられ、Aレジスタに0BHがロードされます。



## 2.5.2 オペランド欄記述上の注意点

### (1) 演算子の優先順位

演算子には次の優先順位があります。

- (a) ー符号, NOT
- (b) \*, /, MOD, SHR, SHL
- (c) +, -
- (d) AND
- (e) OR, XOR
- (f) EQ, NE

演算順位変更のために右カッコ“)”, 左カッコ“(”を使用することができます。

例1  $5 + 8 - 6 * 2 / 4 = 10$   
 $5 + (8 - 6) * 2 / 4 = 6$   
 $(5 + 8 - 6) * 2 / 4 = 3$   
 $2 * (0FH - (0BH AND (0AH OR 0FH))) = 8$   
 $2 * 0FH - 0BH AND 0AH OR 0FH = 0FH$

例2 式に記述された演算は左から順次行われますが、カッコおよび優先順位の異なる演算子が現われると途中結果が一時スタックに退避されます。このため、式の記述が複雑になるとスタックがオーバーフローしてエラーとなる場合があります。

### (2) 演算子に伴うブランク

演算子 NOT, MOD, SHR, SHL, AND, OR, XOR, EQ, NE とオペランドの間には少なくとも1個のブランクが必要です。

正しい例	誤った例
LAI '*' AND 0FH	LAI '*' AND0FH

### (3) オペランドの評価値

オペランド欄に記述された式の演算結果は、命令の要求を満たす値でなければいけません。

例 たとえばLAI命令の場合、オペランドは4ビットですので0H~0FHの範囲に入っていないなければいけません。

正しい例	誤った例
LAI '2*' AND 0FH	LAI '2*'
LAI 4*4-1	LAI 4*4*4

## 2.6 コメント欄

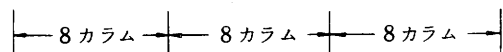
コメント欄はセミコロン“;”で始まり、その後コメント(Comment)を記述します。コメントは、プログラム・リストを参照する場合にプログラムの内容理解を助けるための注釈で、アセンブル・リスト上には出力されませんがアセンブラの処理対象にはなりません。

```
例  HERE:  LAI 0FH ; THIS IS A COMMENT
      ;
      ; BEGIN LOOP HERE
      ;
```

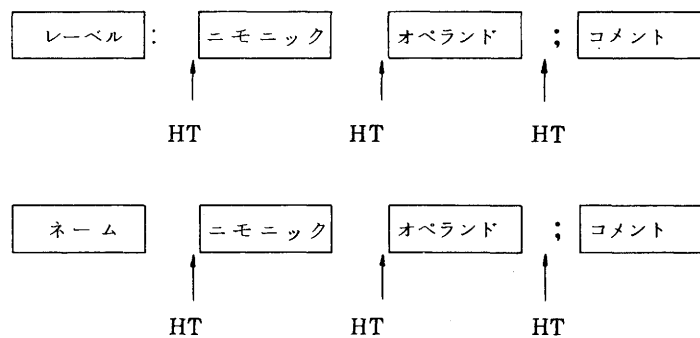
## 2.7 タビュレーション機能

μPD7550,7560シリーズ・アセンブラには、アセンブル処理で出力されるアセンブル・リストを見やすい形式にするために、タビュレーション機能が用意されています。タビュレーション機能を使用するとソース・プログラムのシンボル欄、ニモニック欄、オペラント欄、コメント欄がそれぞれ次に示すように整理できます。

```
TRNS1:  LHLI    07H    ;SRC1
TRNS2:  LHLI    0BH    ;SRC2
        LDS          ;A←SRC,L←L-1
        XADR    13H    ;STORE A TO (13H)
        LDS
        XADR    12H    ;STORE A TO (12H)
        LDS
        XADR    11H    ;STORE A TO (11H)
        LDS
        XADR    10H    ;STORE A TO (10H)
        END
```



タビュレーション動作を行なう場合は、ソース・プログラムのニモニック欄、オペラント欄の先頭、およびコメント欄の始まりを示すセミコロン(;)の前に“HT”(Horizontal Tabulation 09H)を挿入します。このようにしますと、アセンブル・リスト上の各欄が見やすく整理されます。



HTは論理的にはソース文中のブランクの挿入可能な個所であればどこにでも挿入でき、挿入される  
カラム位置によって1から8個までのブランクを発生します。

$$((HT \text{ のカラム位置} - 1) \wedge 0FFF8H) + 9 = \text{次欄スタート・カラム位置}$$

## 第3章 疑似命令

μPD7550,7560シリーズ・アセンブラのニモニク欄には、機械語命令、疑似命令が記述できます。

以下の説明で、[ ]で囲んだ項目は記述しなくてもよいことを示します。また、シンボルの内コロン(:)でターミネートされるものをレーベル、ブランクでターミネートされるものをネームと呼びます。

### 3.1 ロケーション・カウンタ制御疑似命令

#### 3.1.1 ORG (ORIGIN)

(1) 機能

アセンブラのロケーション・カウンタにオペランドで指定された値をセットします。

(2) 記述形式

シンボル	ニモニク	オペランド	コメント
[レーベル:]	ORG	式	[ ; コメント ]

(3) 注意事項

- (イ) オペランドの式としてレーベルやネームも使用できますが、そのレーベルまたはネームはそれ以前に定義されているものでなければなりません。
- (ロ) ORG文のシンボル欄に記述されたレーベルに割り付けられる値はオペランドで指定された値ではなく、直前のロケーション・カウンタの値です。

例	アドレス	シンボル	ニモニク	オペランド
	015D		ST	
	015E		ILS	
	015F		JMP	NEXT
		START:	ORG	200H
	0200	NEXT:	ASC	

この例ではJMPは2バイト命令ですのでレーベルSTARTに161Hが割り付けられます。また、ORG命令のオペランドは200Hを示し、ASC命令に200Hが割り付けられ、レーベルNEXTには200Hが割り付けられます。

- (ハ) プログラムの先頭にORG文によるアドレス指定がない場合は、アセンブラはロケーション・カウンタに0000番地を割り付けます。
- (ニ) ORG直前の割付けアドレスよりも低いアドレスをORGで指定した場合はOエラーとなります。

この場合ORG命令機能は有効ですが、正しいオブジェクトを得るためにはOエラーがなくなるようにしてください。

次の例では10H番地と11H番地のオブジェクトが重複しているため、これらを重複しないように訂正しなければなりません。

エラー	アドレス	オブジェクト	ニモニク	オペランド
	000E	20	DB	20H
	000F	20	DB	20H
	0010	20	DB	20H
	0011	20	DB	20H
O		0010	ORG	10H
	0010	20	DB	20H
	0011	20	DB	20H
	0012	20	DB	20H

### 3.2 シンボル定義疑似命令

#### 3.2.1 EQU ( EQUATE )

(1) 機能

オペランドで指定された式の値またはアドレスをシンボル欄に記述したネームに割り付けます。

(2) 記述形式

シンボル	ニモニク	オペランド	コメント
ネーム	EQU	式	[ ;コメント ]

(3) 注意事項

- (イ) オペランドにレーベルまたはネームを記述するときは、それはすでに定義されているものでなければいけません。
- (ロ) ネームは必ずblankでターミネートします。
- (ハ) シンボル欄, ニモニク欄, オペランド欄の記述にエラーがあるとそのネームは登録されませんので、そのネームを参照した文もエラーになります。
- (ニ) EQUで定義したネームは同一プログラム内では他の値に再定義できません。

(4) 使用例

シンボル	ニモニック	オペランド
SUBR	EQU	7
SEG1	EQU	0B3H
	⋮	
	CALL	SUBR
	⋮	
	JMP	SEG1
	⋮	
	ORG	100H
	⋮	

### 3.2.2 SET

(1) 機能

オペランド欄で指定した式の値またはアドレスをシンボル欄に記述したネームに割り付けます。

(2) 記述形式

シンボル	ニモニック	オペランド	コメント
ネーム	SET	式	[ ; コメント ]

(3) 注意事項

- (イ) オペランド欄にレーベルまたはネームを記述するときは、それはすでに定義されているものでなければいけません。
- (ロ) ネームは必ずブランクでターミネートします。
- (ハ) シンボル欄，ニモニック欄，オペランド欄の記述にエラーがあるとそのネームは登録されませんので，そのネームを参照した文もエラーになります。
- (ニ) ネームとその値は一番最初にSETステートメントで定義しますが，二度目以降はそのネームがオペランドで指定した値に変更されます。変更された値は次のSETステートメントがくるまでその値が保持されます。

(4) 使用例

シンボル	ニモニック	オペランド	コメント
IMMED	SET	5	
	LAI	IMMED	; IMMED=5
IMMED	SET	10H-6	
	LAI	IMMED	; IMMED=0AH

### 3.3 データ定義疑似命令

#### 3.3.1 DB(Define Byte of data)

(1) 機能

8ビットのデータを確保します。

(2) 記述形式

シンボル	ニモニック	オペランド	コメント
[ レベル: ]	DB	式または文字定数	[ ; コメント ]

(3) 注意事項

(イ) オペランドとしては次の2種類があります。

(i) 式

式の値を8ビットのデータとして確保します。式の値が8ビットを越えた場合は下位8ビットがデータとして確保され、エラー・コード(V)が印字されます。

(ii) 文字定数

1文字に対してそれぞれ7ビットASCIIコードが確保され、文字数分のASCIIコードが順にメモリに割り当てられます。

(ロ) オペランドはコンマ(,)で区切っていくつでも指定することが可能です。

(4) 使用例

シンボル	ニモニック	オペランド	マシン・コード
DATA1:	DB	0A3H	A3
DATA2:	DB	2FH-0AH	25
WORD1:	DB	'WOR'	574F52
WORD2:	DB	5*2, 'U', 'M'	0A554D

### 3.4 サブルーチン・エントリ・テーブル定義疑似命令

#### 3.4.1 DET(Define Subroutine Entry Table)

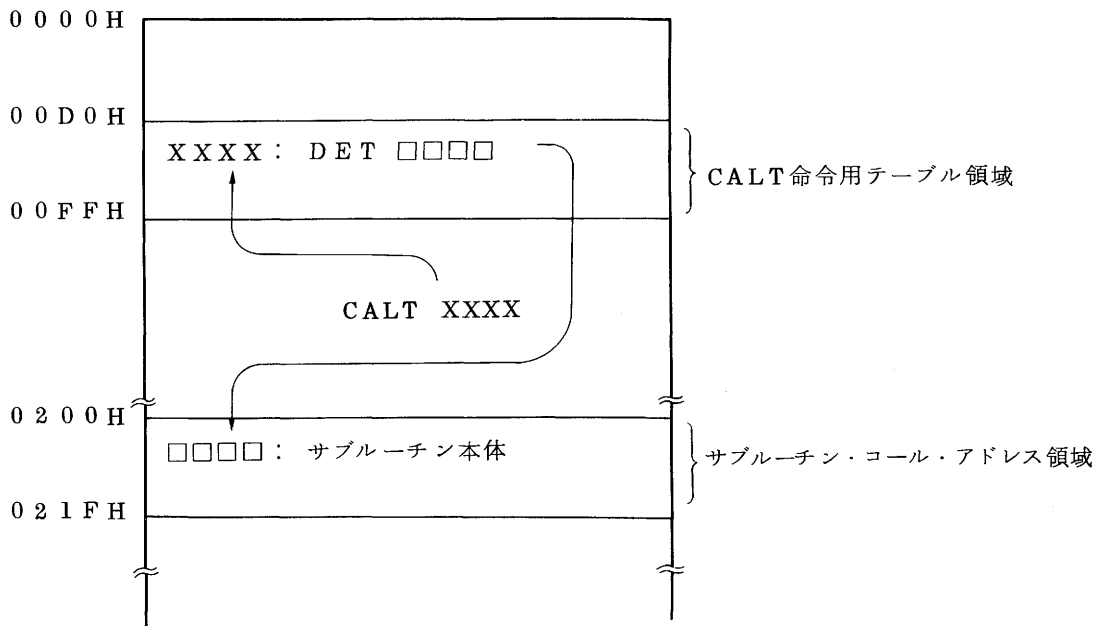
(1) 機能

CALT命令が参照するサブルーチン・エントリ・テーブルを作成します。

サブルーチン・コール命令としてCALL, CALTの二種あり, CALL命令は2バイト命令で, CALT命令は1バイト命令です。したがって使用頻度の高いサブルーチンのコールには, CALT命令を使用するとプログラム・サイズが大きくなるのを防ぐことができます。

ただし, CALT命令のオペランドには, サブルーチン・エントリ・テーブル・アドレスを記述するため, CALT命令用に定められたテーブル領域(0D0H~0FFH)に疑似命令DETにより, あらかじめサブルーチン・コール・アドレス(CALT命令が参照するサブルーチンの先頭番地)を記述します。

図 3-1 プログラム・メモリ・マップ



(2) 記述形式

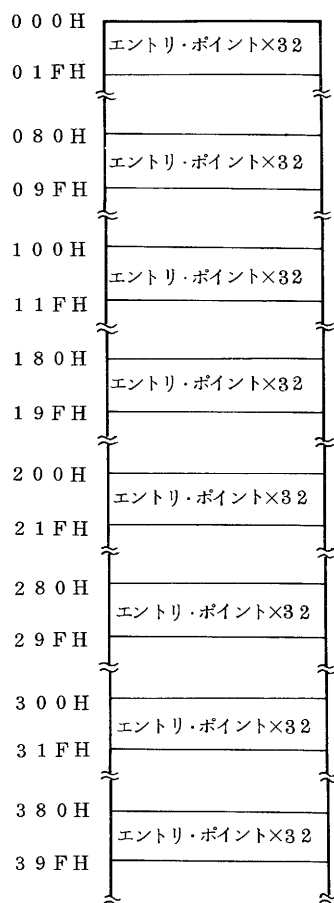
シンボル	ニモニック	オペランド	コメント
[ レーベル: ]	DET	式	[ ; コメント ]

(3) 注意事項

- (イ) DET命令は208～255(00D0H～00FFH)番地の範囲, すなわちCALT命令のエントリ・テーブル内に記述しなければなりません. この範囲外に記述するとAエラーとなります.
- (ロ) 式の値は $128 \times n$ から $128 \times n + 31$  ( $n$ は0～7の整数)の範囲, すなわちCALT命令が参照するサブルーチンの先頭番地を記述しなければなりません. この範囲外であるとVエラーとなります.



図3-2 CALT命令が参照するサブルーチンのエントリ・ポイント



(4) 使用例

CALT命令の参照するテーブルが“TABLE”，参照するサブルーチンが“SUBR”の場合を示します。

アドレス	シンボル	ニモニック	オペランド	コメント
0H		ORG	0D0H	
D0H	TABLE:	DET	SUBR	;ENTRY TABLE FOR 'CALT'
	:	:		
F0H		CALT	TABLE	;CALL SUBROUTINE 'SUBR'
	:	:		
	:	ORG	100H	
100H	SUBR:	LAI	5	;SUBROUTINE CALLED
	:	:		;BY 'CALT' TABLE
		RT		
		:		

### 3.5 条件付アセンブル疑似命令

#### 3.5.1 IFとELSEとENDIF

(1) 機能

(イ) IFとENDIFの場合

オペランドが“0”以外の値の場合は、IFとENDIFで囲まれたステートメントがアセンブルの対象となります。

オペランドが“0”の場合は、IFとENDIFで囲まれたステートメントはアセンブルの対象となりません。

(ロ) IFとELSEとENDIFの場合

オペランドが“0”以外の値の場合は、IFとELSEで囲まれたステートメントがアセンブルの対象となり、ELSEとENDIFで囲まれたステートメントはアセンブルされません。

オペランドが“0”の場合は、IFとELSEで囲まれたステートメントはアセンブルされず、ELSEとENDIFで囲まれたステートメントがアセンブルの対象となります。

(2) 記述形式

<u>シンボル</u>	<u>ニモニック</u>	<u>オペランド</u>	<u>コメント</u>
[レベル:]	IF	式	[ ;コメント]
	⋮		
	ステートメント		
	⋮		
[レベル:]	ELSE		[ ;コメント] ← ELSEを使用した 場合
	⋮		
	ステートメント		
	⋮		
[レベル:]	ENDIF		[ ;コメント]

(3) 注意事項

(イ) IFとそれに対応するENDIF文の間に含まれるすべてのステートメントは、IF-ENDIFブロックとして定義されます。

このブロックは最大8レベルまでネスティングが可能です。

(ロ) ELSEはオプションですので、必ずしも指定する必要はありません。ただし、指定する場合には、IF-ENDIFブロックに対して1つだけしか使用できません。

(4) 使用例

(イ) ELSEを使用した例

<u>シンボル</u>	<u>ニモニック</u>	<u>オペランド</u>	
COND	SET	0FFH	
	IF	COND	↑ アセンブルされる。
	LAI	5	
	ELSE		
	LAI	6	↑ アセンブルされない。
	ENDIF		
COND	SET	0	
	IF	COND	↑ アセンブルされない。
	AISC	7	
	ENDIF		

(ロ) ネスティングの例

レベル 1 {	レベル 2 {	IF	TYPE	↑ TYPE≠0 ならば アセンブルされる。	
		⋮			
		IF	COND1		↑ TYPE≠0, COND1≠0 ならばアセンブルされる。
		⋮			
		ENDIF			
		⋮			
		IF	COND2		↑ TYPE≠0 ならば アセンブルされる。
		⋮			
		ENDIF			↑ TYPE≠0, COND2≠0 ならばアセンブルされる。
		ENDIF			

### 3.6 アセンブル終了疑似命令

#### 3.6.1 END

(1) 機能

アセンブラにプログラムの終了を指示します。

(2) 記述形式

<u>シンボル</u>	<u>ニモニック</u>	<u>オペランド</u>	<u>コメント</u>
[ レベル: ]	END		[ ;コメント ]

(3) 注意事項

- (イ) アセンブラは、ENDが現われるまでアセンブルを行います。END以降にソース文が存在する場合はコメントとして処理します。

## 3.7 アセンブル制御疑似命令

### 3.7.1 TITLE

#### (1) 機能

アセンブル・リスト上で改ページを行い、次ページから各リストのタイトル欄に、オペランドで確保された文字列を印字します。

#### (2) 記述形式

シンボル	ニモニック	オペランド	コメント
[レーベル:]	TITLE	'文字列'	[ ;コメント]

#### (3) 注意事項

- (イ) 文字列は最大32文字まで記述することができます。それ以上記述した文字は無視されます。
- (ロ) アセンブラはTITLE文が現われると改ページをして新ページの最初の行にTITLE文自身を印字します。

### 3.7.2 EJECT

#### (1) 機能

EJECT文自身を印字した後、アセンブル・リスト上で改ページを行います。

#### (2) 記述形式

シンボル	ニモニック	オペランド	コメント
[レーベル:]	EJECT		[ ;コメント]

### 3.7.3 SPACE

#### (1) 機能

アセンブル・リスト上で、オペランドの式の値だけ改行します。SPACE文自身は改行する直前に印字します。

#### (2) 記述方式

シンボル	ニモニック	オペランド	コメント
[レーベル:]	SPACE	式	[ ;コメント]

#### (3) 注意事項

SPACE命令は同一頁内で有効です。SPACE文を記述した頁内で、SPACE文以降の空行数を式の値が越えた場合は、式の値は無視して改頁します。

### 3.7.4 PRINT

(1) 機能

アセンブル・リスト出力をオペランドの指定にしたがって制御します。

(2) 記述形式

シンボル	ニモニック	オペランド	コメント				
[レーベル:]	PRINT	<table border="1"> <tr><td>ON</td></tr> <tr><td>OFF</td></tr> <tr><td>SEL</td></tr> <tr><td>NOSEL</td></tr> </table>	ON	OFF	SEL	NOSEL	[ ;コメント]
ON							
OFF							
SEL							
NOSEL							

(3) オペランドの意味

ON …… PRINT OFF状態を解除します。PRINT ON文自身の行から印字を始めて、

OFF …… PRINT OFF文の次の行からの印字を省略します。

SEL …… PRINT NOSEL状態を解除します。PRINT OFF状態では PRINT SEL文自身は印字されません。

NOSEL ……条件付アセンブル疑似命令(IF, ELSE, ENDIF)および条件が不成立でアセンブルされなかったソース・プログラムの印字を省略します。PRINT OFF状態では PRINT NOSEL文自身は印字されません。

(4) 注意事項

アセンブラのスタート時では、PRINT ONかつPRINT SELの状態になっています。

## 3.8 JUMP系命令の自動選択疑似命令

### 3.8.1 GJMP

(1) 機能

ユーザがGJMPと書けばアセンブラが自動的にJCP, JMPを選択し、そのオブジェクト・コードを生成します(4.3.3の(5)参照)。

(2) 記述方式

シンボル	ニモニック	オペランド	コメント
[レーベル:]	GJMP	式	[ ;コメント]

(3) 注意事項

(i) 最適化オプションを指定しない場合

(1) ソース文のオペランドの式にシンボルを1つも書かない場合

アセンブラはJCP命令あるいはJMP命令を選択し、JCP命令が使える場合は、JCP命令のオブジェクト・コードを生成し、その他はJMP命令のオブジェクト・コードを生成します。

(4)使用例②参照)

(a) オペランドの式のシンボルを定義した行がG J M Pを書いた行と同じか、またはその行より先頭方向にある場合

アセンブラはJ C P命令あるいはJ M P命令を選択し、J C P命令が使える場合は、J C P命令のオブジェクト・コードを生成し、その他はJ M P命令のオブジェクト・コードを生成します(4)使用例①, ⑤参照)。

(b) オペランドの式のシンボルを定義した行がG J M Pを書いた行よりE N D方向にある場合

アセンブラは、すべてJ M P命令のオブジェクト・コードを生成します。ただし、その中でJ C P命令が使える可能性のある場合は、リスト上のエラー・メッセージ欄に' W 'を印字します(4)使用例③, ④参照)。

(ii) 最適化オプションを指定した場合

アセンブラは、J C P命令あるいはJ M P命令を選択し、J C P命令が使える場合は、J C P命令のオブジェクト・コードを生成し、その他はJ M P命令のオブジェクト・コードを生成します。

(4) 使用例

E	ADDR	OBJECT	SOURCE	
	0103		L0:	
	0110	83	GJMP L0	← JCP命令生成①
	0120	A3	GJMP 123H	← JCP命令生成②
	0123		L1:	
	0200	22F3	GJMP L2	← JMP命令生成③
W	02F0	22F3	GJMP L3	← JMP命令生成 および'W'表示④
	02F3		L2:	
	0300		ORG \$-0DH	
	02F3		L3:	
	0300	2103	GJMP L0	← JMP命令生成⑤

注意

GJMP疑似命令のオペランドに前方参照となるシンボルを記述し、その分岐範囲が1バイト命令で分岐可能な最大値であった場合、アセンブラは1バイトではなく、2バイト分岐命令のコードを生成する。



## 第4章 アセンブラの操作方法

以下の説明において“ [ ”, “ ] ”で囲まれた文字列は省略可能であることを示します。また入力文字は英小文字を使用することができますが、英大文字と解釈されます。

### 4.1 アセンブラの起動方法

プログラムの起動は、コマンド・ライン上に、プログラム名およびパラメータを指定することにより行ないます。

コンソールから次のように入力してください。

```
X>[d:]AS75M ソース・モジュール・ファイル名 [コントロール][コンソールに出力
されるメッセージをファイルに出力させる指定] )
          ( ) キャリッジ・リターン)
```

一行で指定可能な文字数は127文字(“ ”を含む)までです。それ以降は無効となります。

X : ログイン・ディスク名

ソース・モジュール・ : アセンブルするソース・モジュール・ファイルのファイル名  
ファイル名 : 媒体はFDに限ります。

: ファイル名の形式は 4.3.2 ファイル名規則を参照してください。

コントロール : 4.3 コントロールを参照してください。

コンソールに出力され

るメッセージをファイ

ルに出力させる指定 : 4.1.2 を参照してください。

#### 4.1.1 ソース・モジュール・ファイル名とコントロールを指定する方法

コマンド・ライン上でソース・モジュール・ファイル名と、コントロールを指定する方法は次の三通りあります。

##### (1) コマンド・ライン上に直接記述する方法

OSの出力するプロンプタに続けてプログラム名、ソース・モジュール・ファイル名を指定し、次にコントロールを入力します。

```
AS75M ソース・モジュール・ファイル名 [コントロール][ . . . ]
```



例 A>AS75M TEST.ASM DATE=59/3/24 )

(2) パラメータ・ファイルによる方法

コントロールやソース・モジュール・ファイルの指定をあらかじめディスク内に用意したパラメータ・ファイルより入力します。

パラメータ・ファイルを示す記号として「!」を用います。これは、同じ指定を何度も行う場合、指定するパラメータが多い時に使用します。

AS75M !パラメータ・ファイル名

(ただしパラメータ・ファイルの一番初めはソース・モジュール・ファイルを指定してください。)

パラメータファイル規約

[ソース・モジュール・ファイル名] コントロール [・・・] )

- ・パラメータ・ファイルが複数行から成る場合、2行目以降が1行目の継続行となります。
- ・パラメータ・ファイル内にある空行は解釈されません。
- ・パラメータ・ファイル内にパラメータ・ファイル指定(!ファイル名)、コンソールに出力されるメッセージをファイルに出力する指定(>ファイル名, >>ファイル名)を指定することもできます。
- ・パラメータ・ファイル内で同種のコントロールを複数回指定した場合は、後に指定したコントロールが優先されます。

例 A>AS75M !TEST.JOB )

TEST.JOBの内容

TEST.ASM DATE=59/3/24 )

この場合、(1)の例と同じ結果になります。

(3) (1)と(2)を併用する方法

パラメータ・ファイルの内容に対し、コマンド・ラインから追加あるいは変更を可能にする方法です。

ソース・モジュール・ファイル名はパラメータ・ファイル、コマンド・ラインのいずれからでも入力できます。

(ただし、ソース・モジュール・ファイルの指定が一番初めになければいけません。したがってコマンド・ラインかパラメータ・ファイルかソース・モジュール・ファイルの指定のある方を先に記述してください。)

AS75M ソース・モジュール・ファイル [コントロール...] !パラメータ・ファイル [コントロール...]  
 または  
 AS75M !パラメータ・ファイル [コントロール...]  
 (パラメータ・ファイルの先頭はソース・モジュール・ファイルの指定)

例 A>AS75M !EX1.JOB NOPR )  
 EX1.JOBの内容 TEST.ASM DATE=59/3/24  
 この場合 TEST.ASM DATE=59/3/24 NOPR )  
 を指定したことになります。

- 注1. コントロールはコマンド・ラインおよびパラメータ・ファイルのいずれからの入力も有効となります。また順序に制限はありません。(ただし、注2,3の場合は除きます。)
2. 同種のコントロールをコマンド・ラインおよびパラメータ・ファイルから指定した場合、コマンド・ラインからの入力が優先され、パラメータ・ファイルから入力したパラメータは無視されます。

例 A>AS75M !EX.JOB NOPR )  
 EX.JOBの内容  
 TEST.ASM PRINT=TEST.PRN DATE=59/3/24  
 この場合 TEST.ASM NOPR DATE=59/3/24 )を  
 指定したことになります。

3. 同種のコントロールをコマンド・ライン(またはパラメータ・ファイル)から指定した場合、あとから指定したコントロールが優先されます。

例 A>AS75M TEST.ASM PR NOPR )  
 この場合 TEST.ASM NOPRを指定したことになります。  
 (この場合コンソールにメッセージが出力されます。5.5参照)

#### 4.1.2 コンソールに表示されるメッセージをファイルに出力させる方法

実行時、コンソールに表示されるメッセージをファイルに出力させる方法です。記号として'>', '>>'を用います。

- (1) >ファイル名 プログラム実行時、コンソールに表示されるメッセージを代りにファイル名で示したファイルに出力します。すでに同名のファイルが存在している場合は、上書きします。

例 A>AS75M TEST.ASM >MASSAG.JOB

- (2) >>ファイル名 プログラム実行時，コンソールに表示されるメッセージを代りにファイル名で示したファイルに出力します。すでに同名のファイルが存在している場合は，そのファイルに追加します。

例 A>AS75M TEST.ASM >>MASSAG.JOB

## 4.2 アセンブラ・タイトルの出力

4.1によりアセンブラが起動されるとコンソールに起動開始メッセージが表示されます。

```
UPD7550/60 SERIES ASSEMBLER VX.X [XX XXX XX]
Copyright (C) 1984 NEC Corporation
```

## 4.3 コントロール

コントロールとは，各プログラムの実行に先立って必要な情報を指示するためのものです。

### 4.3.1 入力形式

- ・コントロールの入力形式は次のとおりです。

```
コントロール名 [= (パラメータ)]
```

- ・各コントロールの指定順序は自由です。

### 4.3.2 ファイル名規則

- (1) フロッピー・ディスク上のファイルの場合

```
[ドライブ名:]プライマリ・ネーム[.[ファイル・タイプ]]
```

ドライブ名：ファイルを格納しているFDがセットされているドライブ名

プライマリ・ネーム：8文字以内の文字列

ファイル・タイプ：3文字以内の文字列

<省略時解釈>

省略時解釈は，ファイルの種類毎に異なり，表4-1のとおりとなります。

表4-1 ドライブ名およびファイル・タイプの省略時解釈

項番	ファイルの種類	ドライブ名	ファイル・タイプ
1	ソース・モジュール・ファイル	ログイン・ ディスク	ASM
2	オブジェクト・モジュール・ファイル		HEX
3	リスト・ファイル		PRN
4	シンボル・テーブル・ファイル		SYM

注1. プライマリ・ネーム, ファイル・タイプは, 次の文字を使用してください.

英大文字, 英小文字 (英大文字と解釈されます), 数字, ! # \$ % & " ( ) + - ^ \_ @

2. ドライブ名を省略した場合, ログイン・ディスクを指定したものとみなされます.
3. ファイル名を ' プライマリ・ネーム. ' と入力した場合, ファイル・タイプなしのファイル名を指定したものとみなされます.

(2) フロッピー・ディスク以外の場合

装置名:

装置名には次のものが指定できます.

- { CON.....コンソール
- { LST.....シリアル・プリンタ
- { PUN.....紙テープ

4.3.3 コントロールの種類

アセンブラでは次の5種類のコントロールを提供しています.

- 出力ファイル指定
- 出力リスト項目指定
- 処理日付指定
- シリーズ指定
- G J M P 最適化指定

表 4-2 基本コントロール一覧

項番	分類	コントロール形式	短縮形	省略時の解釈	
a	出力ファイル指定用	オブジェクト モジュール・ファイル	OBJECT [=ファイル名] NOOBJECT	OJ NOOJ	OJ=ソース・モジュール・ファイル名 ただし、ファイル・タイプは'HEX'
		リスト・ファイル	PRINT [=ファイル名] NOPRINT	PR NOPR	PR=ソース・モジュール・ファイル名 ただし、ファイル・タイプ'PRN'
	ワーク・ファイル	WORKFILES [=ドライブ名]	WF	WF=ソース・モジュール・ファイル 格納ドライブ	
	シンボル・テーブル・ ファイル	DEBUG [=ファイル名] NODEBUG	DB NODB	DB=ソース・モジュール・ファイル名 ただし、ファイル・タイプは'SYM'	
b	出力項目 指定用	シンボル・リスト	SYMBOLS NOSYMBOLS	SB NOSB	NO SB
		エラー・リスト	ERRORPRINT NOERRORPRINT	EP NOEP	NO EP
	クロス・レファレンス ・リスト	XREF NOXREF	XR NOXR	NO XR	
c	処理日付 指定用	DATE [=最大12文字]	DA	日付は出力されません	
d	シリーズ 指定用	MOD=チップの数字部分 または A か B	MD	MD=B	
e	GJMP最適化 指定用	OPTIMIZE [=回数] NOOPTIMIZE	OP NOOP	NO OP	

(1) 出力ファイル指定用コントロール

出力ファイル指定のためのコントロールには次のものがあります。

- ・オブジェクト・モジュール・ファイル出力指定
- ・リスト・ファイル出力指定
- ・ワーク・ファイル作成ドライブ指定
- ・シンボル・テーブル・ファイル出力指定

(a) オブジェクト・モジュール・ファイル出力指定 (省略可)

<形式>

- ・オブジェクト・モジュール・ファイルを作成する場合

OBJECT [=ファイル名]      または      OJ [=ファイル名]

- ・オブジェクト・モジュール・ファイルを作成しない場合

NOOBJECT      または      NOOJ

<補足>

- ・出力媒体は、FD, 紙テープが使用可能です。

各出力媒体を指定する場合は

FD ——— OBJECT = ファイル名

紙テープ ——— OBJECT = PUN : と指定します。

- 本コントロールまたは本コントロールのファイル名指定を省略することができます。

省略時解釈： OBJECT=ソース・モジュール・ファイル名のプライマリ・ネーム・HEX

この場合のドライブ名は、ソース・モジュール・ファイルと同一のドライブ名になります。

例 Bのドライブ上のFDにOMFというファイル名でオブジェクト・モジュール・ファイルを作成

…OBJECT = B : OMF …

Bのドライブ上のFDにOMF.HEXというオブジェクト・モジュール・ファイルが作成されます。

(b) リスト・ファイル出力指定（省略可）

<形式>

- リスト・ファイルを出力する場合

PRINT [=ファイル名] または PR [=ファイル名]

- リスト・ファイルを出力しない場合

NOPRINT または NOPR

<補足>

- リスト・ファイルの出力媒体はFD, SP, コンソールが使用可能です。

各出力媒体を指定する場合は

FD ——— PRINT = ファイル名

SP ——— PRINT = LST :

コンソール ——— PRINT = CON :

と指定します。

- 本コントロールまたは本コントロールのファイル名指定を省略することができます。

省略時解釈： PRINT=ソース・モジュール・ファイル名のプライマリネーム・PRN

この場合のドライブ名は、ソース・モジュール・ファイルと同一のドライブ名になります。

例 SPへのリスト出力

…PRINT = LST : …

リスト・ファイルが、SPに出力されます。

(c) ワーク・ファイル作成ドライブ指定 (省略可)

<形式>

WORKFILES = ドライブ名      または      WF = ドライブ名

<補足>

- ・本コントロールを省略することができます。

省略時解釈: ソース・モジュール・ファイルと同一のドライブ名

例 B のドライブ上の FD にワーク・ファイルを割り当てる。

…WORKFILES=B:

1.3 に記述されている作業用ファイルが B のドライブ上の FD に作成され、処理が終了すると消去されます。

(d) シンボル・テーブル・ファイル出力指定 (省略可)

<形式>

- ・シンボル・テーブル・ファイルを作成する場合

DEBUG [=ファイル名]      または      DB [=ファイル名]

- ・シンボル・テーブル・ファイルを作成しない場合

NODEBUG      または      NO DB

<補足>

- ・出力媒体は FD, 紙テープに出力できます。

各出力媒体を指定する場合は

FD ——— DEBUG = ファイル名

紙テープ — DEBUG = PUN : と指定します。

- ・本コントロールまたは本コントロールのファイル名の指定を省略することができます。

省略時の解釈: DB: ソース・モジュール・ファイルのプライマリ・ネーム .SYM

この場合のドライブ名は, ソース・モジュール・ファイルと同一のドライブ名になります。

例 B のドライブ上の FD にシンボル・テーブル・ファイルを作成

…DEBUG=B: STF.SYM . . .

B のドライブ上の FD に STF.SYM というファイル名でシンボル・テーブル・ファイルを作成します。

(2) 出力リスト項目指定用コントロール

リスト・ファイルに出力するリストの種類を指定するためのコントロールです。

出力リストには、次のものがあります。

- ・アセンブル・リスト
- ・シンボル・リスト
- ・クロス・レファレンス・リスト
- ・エラー・リスト

なお、アセンブル・リストは、PRINTコントロールを指定した場合、無条件に出力されます。

(a) シンボル・リスト出力指定（省略可）

<形式>

- ・シンボル・リストを出力する場合

**SYMBOLS**      または      **SB**

- ・シンボル・リストを出力しない場合

**NOSYMBOLS**      または      **NOSB**

<補足>

- ・NOPRINTコントロールを指定した場合はSYMBOLSの指定が無視されます。
- ・本コントロールを省略することができます。

省略時の解釈： **NOSYMBOLS**

例① シンボル・リストをSPに出力

…PRINT=LST: SYMBOLS…

② シンボル・リストを出力しない

…NOSYMBOLS…

または

コントロールを指定しない。

(b) クロス・レファレンス・リスト出力指定

<形式>

- ・クロス・レファレンス・リストを出力する場合

**XREF**      または      **XR**

- ・クロス・レファレンス・リストを出力しない場合

**NOXREF**      または      **NOXR**



<補足>

- NOPRINT コントロールを指定した場合は、XREF の指定が無視されます。
- 本コントロールを省略することができます。

省略時解釈: NOXREF

例① クロス・レファレンス・リストをSPに出力

…PRINT=LST: XREF…

② クロス・レファレンス・リストを出力しない

…NOXREF…

または

コントロールを指定しない

(c) エラー・リスト出力指定 (省略可)

<形式>

- エラー・リストを出力する場合

ERRORPRINT      または      EP

- エラー・リストを出力しない場合

NOERRORPRINT      または      NOEP

<補足>

- NOPRINT コントロールを指定した場合、ERRORPRINT の指定が無視されます。
- エラー・リストを出力する場合、アセンブル・リストも出力します。
- 本コントロールを省略することができます。

省略時の解釈: NOERRORPRINT

例① エラー・リストをSPに出力

…PRINT=LST: ERRORPRINT…

② エラー・リストを出力しない

…NOERRORPRINT…

または

コントロールを指定しない

(3) 処理日付指定用コントロール (省略可)

アセンブルを行なった日の日付を通知するためのコントロールです。出力リストのヘッダ (DATE欄) に指定した日付が印字されます。

<形式>

DATE [=日付]      または      DA [=日付]

日付：12文字以内のASCII文字列

<補足>

- 本コントロールまたは本コントロールの日付の指定を省略した場合は、日付は出力されません。
- 12文字に満たない場合は、ブランクが補われます。
- 13文字以上指定した場合は最初の12文字が有効となります。
- DATEコントロールで使用できる文字は2.2節で規定された文字と同様ですが日付の文字に△(ブランク)あるいは=を使用する場合は、日付の指定を引用符' , 'で囲んで使用します。また、引用符を文字として使用したい場合は、引用符を2個続けます。

例① DATE欄に84/5/5と印字する。

...DATE = 84/5/5...

例② DATE欄に84 5 5と印字する。

...DATE = '84 5 5'...

例③ DATE欄に84'5 9と印字する。

...DATE = '84' '5 9'...

(4) シリーズ指定用コントロール(省略可)

アセンブル対象品種を指定するためのコントロールです。

<形式>

MOD =シリーズ番号      または      MD =シリーズ番号

シリーズ番号

A ..... μPD7500セットA

B ..... μPD7500セットB

54 ..... μPD7554

56 ..... μPD7556

64 ..... μPD7564

66 ..... μPD7566

<補足>

- 本コントロールは省略することができます。

省略時の解釈： MOD = B

例 指定チップを7554とします。

...MD = 54...

(5) GJMP 最適化指定用コントロール (省略可)

JUMP 系命令自動選択機能の最適化の有無を指定するコントロールです。

<形式>

- GJMP 疑似命令の最適化を行う場合

OPTIMIZE [=最大回数]      または      OP [=最大回数]

- GJMP 疑似命令の最適化を行わない場合

NOOPTIMIZE      または      NOOP

<補足>

- 最適化繰返し最大回数に記述できるのは 1 以上 9999 (10 進数で表した場合で) 以下の数字です。
- 最適化繰返し最大回数は省略することができます。  
このとき最適化繰返し最大回数は 9999 回となります。  
本コントロールは省略することができます。

省略時の解釈:      NOOPTIMIZE

最適化パスの方法

アセンブラは、GJMP の最適化を行なうために、ソース・モジュール・ファイルの中間ファイルをフロッピー・ディスクに作成し、これを指定された回数 (または、最適化が完了するまで) 繰り返し読みます。

この 1 回の読み込みで行なう処理を最適化パスと呼びます。

最適化コントロールが指定された場合、次の手順で処理が行われます。

- ① PASS 1 終了時に、次のメッセージがコンソール上に表示され、最適化パスが開始されます。

OPTIMIZATION STARTS

- ② 指定された回数だけ最適化パスを繰返します。

各最適化パス終了時に、次のメッセージがコンソールに表示され、そのときの最適化パスの状況がわかります。

TIMES = t t t t , CODE = n n n n

t t t t = 現在の繰返し回数が 10 進右づめで入ります。 n n n n = 生成されたオブジェクト・コード数が 16 進右づめで入ります。
--

- ③ 現在のパスで、最適化パスが完了したとき、リスト上のエラー数を示すメッセージを出してから、下記のメッセージが出力され、PASS-2 の処理が開始されます。

OPTIMIZATION COMPLETED, TIMES = t t t t

[ t t t t = 現在の繰返し回数が 10 進右づめで入ります。 ]

- ④ 指定された回数内で、最適化が完了しなかった場合は、リスト上のエラー数を示すメッセージを出力してから下記のメッセージが出力され、PASS-2 の処理が開始されます。

OPTIMIZATION TERMINATED, TIMES = t t t t

[ t t t t = 最終的な繰返し回数が 10 進右づめで入ります。 ]

この場合、生成されたオブジェクト・コードは、最小コード数ではなく、また正常なオブジェクトではないので、最適化を完了するだけの回数を指定して再度アセンブルしてください。

## 4.4 アセンブル実行開始メッセージの出力

アセンブラが起動し、入力したコントロールの解析が終了すると、アセンブラは次のメッセージをコンソールへ出力し、アセンブルを開始します。

ASSEMBLE START

## 4.5 アセンブル終了メッセージの出力

アセンブラは、アセンブル処理が終了すると次のメッセージをコンソールに表示し制御をOSに戻します。

- ・正常終了の場合

ASSEMBLE COMPLETE, nnnn ERRORS

[ nnnn = 10進4桁のアセンブル・エラー数 ]

- ・異常終了の場合

PROGRAM ABORTED

## 4.6 入出力ファイル・リストの出力

最後にアセンブラはソース・モジュール・ファイル名と完成した出力ファイル名をコンソールへ出力します。ただし 'PROGRAM ABORTED' とコンソールに出力された場合（アセンブルを中止した場合）は入出力ファイル・リストは出力されません。

### FILE LIST

SOURCE . . . = ソース・モジュール・ファイル名 <sup>注1</sup>

LIST . . . . . = リスト・ファイル名 <sup>注2</sup>

OBJECT . . . = オブジェクト・ファイル名 <sup>注2</sup>

SYM-TBL . . = シンボル・テーブル・ファイル名 <sup>注2</sup>

注1 4.1 で指定したソース・モジュール・ファイル名を出力します。

2 各ファイルの出力の指定をした時に出力されます。

## 4.7 アセンブラの操作例

(1) ソース・プログラム例

```

                TITLE      '      EXAMPLE'
;
;=====
;=
;=                UPD 7550/60 SERIES      =
;=
;=====
;
                ORG        10H
LABEL1 EQU        10H
LABEL2 EQU        20H
;
                LAI        5
                LHLD      10H
                XAM        HL
;
                DB        0FH
                DB        10H
;
                ORG        0D0H
TABLE: DET        SUBR
;
                ORG        100H
                CALT      TABLE
;
                ORG        200H
SUBR:  ASC
                XAM        HL
                RT
;
                CALT      TABLE
;
                ORG        100H
                DB        10H
                DB        0FFH
;
                END
    
```

(2) 入力例

```
A>AS75M 7500MN.ASM PR=A: SB XR OP=2 MD=A DATE=XX/XXX/XX      ..... アセンブラ・プログラムの起動 (4.3)
UPD7550/60 SERIES ASSEMBLER VX.X (XX XXX XX)                }..... アセンブラ・タイトルの出力 (4.2)
  Copyright (C) 1984 NEC Corporation
ASSEMBLE START                                               ..... アセンブル実行開始メッセージの出力
                                                                (4.4)
OPTIMIZATION STARTS
TIMES=      1, CODE = 0013H
OPTIMIZATION COMPLETE, TIMES =      1                        ..... アセンブル終了メッセージの出力
                                                                (4.5)
ASSEMBLER COMPLETE,      1 ERRORS

FILE LIST
SOURCE... = A:7500MN.ASM
LIST.... = A:7500MN.PRN
OBJECT... = A:7500MN.HEX
SYM-TBL.. = A:7500MN.SYM
}..... 入出力ファイル・リストの出力 (4.6)

A>
```

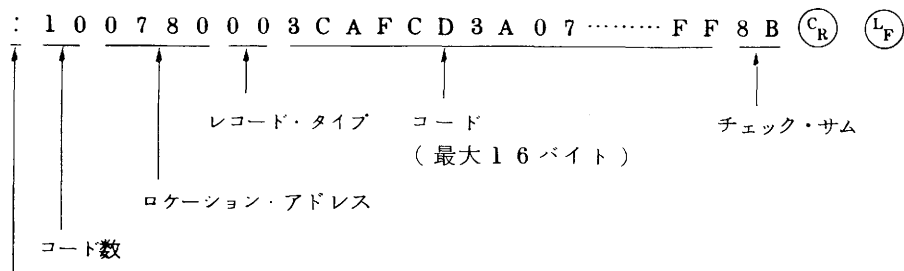
## 第5章 出力形式

### 5.1 オブジェクト・モジュール・ファイル

ソース・モジュール・ファイルをアセンブルすると、アセンブラは絶対番地形式のオブジェクト・モジュール・ファイルを4.3コントロールで割り当てられた装置上にHEX形式で出力します。

以下にHEX形式を示します。

(1) HEX形式



レコードのはじまりを示すコード

レコード・マーク ..... コロン『:』

コード数 ..... 1レコードにおさめられるバイト数が(最大は16バイト) 16進数2桁で示されます。コード数が0のとき紙テープは終りになります。

ロケーション・アドレス ..... 16進数4桁で示されます。

レコード・タイプ ..... 16進数2桁で示されます。通常は00でオブジェクト・コード終了を示す終了レコードのときのみ01になります。

コード ..... 1バイトのコードは16進数2桁で示されます。コードは最大16バイトまで出力することができます。

チェック・サム ..... コロンを除くすべてのデータを初期値0から順に減算した値が入ります。この値は16進数2桁で示されます。

#### チェック・サムの算出方法

ボローを無視した8ビット減算を行います。上記のデータ例ではまず初期値00Hからコード数の10Hを減算します。





## 5.2 出力リスト

本アセンブラが出力するリストは次の4種類です。

- (i) シンボル・リスト
- (ii) アセンブル・リスト
- (iii) エラー・リスト
- (iv) クロスレファレンス・リスト

上記の(i)~(iv)は同一のファイル(ただしPRINTコントロールによる)に上記の順番で出力されます。

### 5.2.1 アセンブル・リスト

アセンブル結果を出力します。

- (1) 出力形式

UPD7500-A ASSEMBLER VX.X				( XX/XXX/XX ) PAGE : 1	
** EXAMPLE **					
E	STNO	ADRS	OBJECT	SOURCE STATEMENT	
	1			;=====	
	2			;=	
	3			;=	
	4			;=	
	5			;=	
	6			;=	
	7			;=====	
	8			;	
	9		0010	ORG	10H
	10		0010	LABEL1 EQU	10H
	11		0020	LABEL2 EQU	20H
	12			;	
	13	0010	15	LAI	5
	14	0011	4E10	LHLI	10H
	15	0013	56	XAM	HL
	16			;	
	17	0014	0F	DB	0FH
	18	0015	10	DB	10H
	19			;	
	20		00D0	ORG	0D0H
	21	00D0	80	TABLE: DET	SUBR
	22			;	
	23		0100	ORG	100H
	24	0100	D0	CALT	TABLE
	25			;	
	26		0200	ORG	200H
	27	0200	7D	SUBR: ASC	
	28	0201	56	XAM	HL
	29	0202	53	RT	
	30			;	
	31	0203	D0	CALT	TABLE
	32			;	
0	33		0100	ORG	100H
	34	0100	10	DB	10H
	35	0101	FF	DB	0FFH
	36			;	
	37			END	

ERROR = 1  
 OPTIMIZATION COMPLETED, TIME = 1

(2) 出力項目の説明

表 5-1

項番(注)	内 容
①	アセンブル対象チップ名 (MODコントロールでの指定値)
②	日付 (DATEコントロールでの指定値) DATEコントロールを指定しないときは起動時の日付
③	出力リストのページ番号を10進表示
④	タイトル (TITLEコントロールでの指定値) TITLEコントロールを指定しないときは空白
⑤	アセンブル・エラー・フラグ エラー・フラグの種類と意味については5.3参照してください。
⑥	ソース・ステートメント番号を10進4桁表示
⑦	ロケーション・アドレスを16進4桁表示
⑧	オブジェクト・コードを16進表示
⑨	ソース・ステートメントのイメージ
⑩	アセンブル・エラー個数 (エラー・フラグ数) を10進表示
⑪	最適化を実行した回数を10進表示 (OPTIMIZEコントロールの指定した場合のみ)

注 出力形式の中の番号です。

(3) アセンブル・リストの例

UPD7500-A ASSEMBLER VX.X

( XX/XXX/XX ) PAGE : 1

\*\* EXAMPLE

\*\*

E	STNO	ADRS	OBJECT	SOURCE	STATEMENT
	1				
	2				
	3				
	4				=====
	5				;=
	6				UPD 7550/60 SERIES
	7				=====
	8				;
	9		0010	ORG	10H
	10		0010	LABEL1 EQU	10H
	11		0020	LABEL2 EQU	20H
	12				;
	13	0010	15	LAI	5
	14	0011	4E10	LHLI	10H
	15	0013	56	XAM	HL
	16				;
	17	0014	0F	DB	0FH
	18	0015	10	DB	10H
	19				;
	20		00D0	ORG	0D0H
	21	00D0	80	TABLE: DET	SUBR
	22				;
	23		0100	ORG	100H
	24	0100	D0	CALT	TABLE
	25				;
	26		0200	ORG	200H
	27	0200	7D	SUBR: ASC	
	28	0201	56	XAM	HL
	29	0202	53	RT	
	30				;
	31	0203	D0	CALT	TABLE
	32				;
0	33		0100	ORG	100H
	34	0100	10	DB	10H
	35	0101	FF	DB	0FFH
	36				;
	37			END	

ERROR = 1  
OPTIMIZATION COMPLETED, TIME = 1

### 5.2.2 エラー・リスト

エラー・リストはアセンブルの結果、エラーの検出されたステートメント行のみを印字するものです。形式としては、アセンブル・リストと同様です。

(1) エラー・リストの例

```

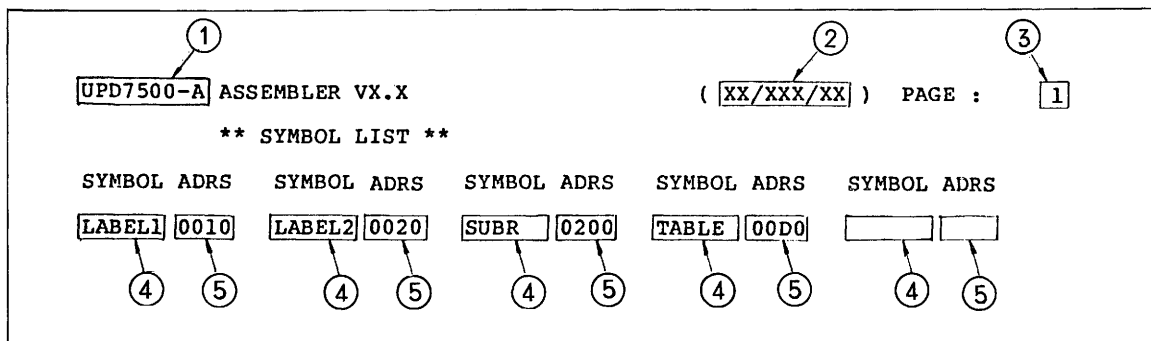
UPD7500-A ASSEMBLER VX.X                ( XX/XXX/XX ) PAGE
      ** EXAMPLE                        **
E STNO ADRS OBJECT   SOURCE STATEMENT
O  33          0100   ORG    100H

ERROR = 1
    
```

### 5.2.3 シンボル・リスト

シンボル・リストは、シンボルとそのシンボルに割り付けられた値の対照表です。シンボルは、アルファベット順に出力されます。

(1) 出力形式



(2) 出力項目の説明

表 5-2

項番注	内 容
①	アセンブル対象デバイス名 (MODコントロールでの指定値)
②	日付 (DATEコントロールでの指定値) DATEコントロールを指定しないときは空白
③	出力リストのページ番号を10進表示
④	シンボル名
⑤	シンボル値を16進4桁表示

注 出力形式の中の番号です。

(3) シンボル・リスト例

```

UPD7500-A ASSEMBLER VX.X                ( XX/XXX/XX ) PAGE : 1
      ** SYMBOL LIST **
SYMBOL ADRS  SYMBOL ADRS  SYMBOL ADRS  SYMBOL ADRS  SYMBOL ADRS
LABEL1 0010  LABEL2 0020  SUBR  0200  TABLE 00D0
    
```

5.2.4 クロスレファレンス・リスト

クロスレファレンス・リストは、シンボルとその定義された文番号と、参照している文番号の対応表です。シンボルはアルファベット順に出力されます。

(1) 出力形式

```

① UPD7500-A ASSEMBLER VX.X                ( ② XX/XXX/XX ) PAGE : ③ 1
      ** CROSS REFERENCE LIST **
SYMBOL DEF. REF. ( STATEMENT NO. )
LABEL1 0010
LABEL2 0011
SUBR   0027 0021
TABLE 0021 0024 0031
④ ⑤ ⑥
    
```

(2) 出力項目の説明

表 5-3

項 番注	内 容
①	アセンブル対象チップ名 (MOD コントロールでの指定値)
②	日付 (DATE コントロールでの指定値) DATE コントロールを指定しないときは空白
③	出力リストのページ番号を10進表示
④	シンボル名
⑤	定義ステートメント番号を10進表示
⑥	参照ステートメント番号を10進表示

注 出力形式の中の番号です。

(3) クロスリファレンス・リスト例

```

UPD7500-A ASSEMBLER VX.X                ( XX/XXX/XX ) PAGE : 1
      ** CROSS REFERENCE LIST **
SYMBOL DEF. REF. ( STATEMENT NO. )
LABEL1 0010
LABEL2 0011
SUBR    0027 0021
TABLE  0021 0024 0031
    
```

### 5.3 アセンブル・リストに印字されるエラー・メッセージ

ソース・モジュール中にエラーがある場合は、アセンブル・リスト上に対応する左側（アセンブル・エラー・フラグ欄）に次に示すエラー・コードが印字されます。

1ステートメント中に複数個のエラーがある場合には最初のエラー・フラグが印字されます。

(1) A (ADDRESS ERROR)

ローケーション・カウンタの値がアセンブル対象品の最大値を越えていることを示します。

(2) B (BALANCE ERROR)

左括弧と右括弧の対応がとれていないか、または引用符の使用が間違っていることを示します。

例 LAI ((DECM-3) MOD 3 …左右の括弧の数が違う。  
 LAI 'A'' AND 0FH ………左右の引用符の数が違う。

(3) E (EXPRESSION ERROR)

演算子の記述が間違っていることを示します。演算子がない場合、ニモニックの綴りを間違っ  
 て記述した場合などにこのエラーとなります。

例 INCR ……… INCRという命令はない。  
 LAM H ………オペランドの記述が違う。

(4) F (FORMAT ERROR)

オペランドがないことを示します。

例 CALL  
 JMP

(5) I (ILLEGAL CHARACTER)

使用できない文字が記述されていることを示します。数値定数が間違っている場合もこのエラーとなります。

例 J M P 1 0 2 B ..... 2進定数に“2”は使えない。  
 J M P 9 Q ..... 8進定数に“9”は使えない。

(6) L (LABEL OR NEME ERROR)

レーベルまたはネームに予約語を使用していることを示します(予約語については付録を参照)。

例 I L S : A S C ..... 予約語(命令)をレーベルに使用している。  
 E N D : S T ..... 予約語(疑似命令) ”  
 S H R : A S C ..... 予約語(演算子) ”  
 H L : L A I ..... 予約語(レジスタ名) ”

(7) M (MULTIPLE DEFINITION ERROR)

同一シンボルが2度以上定義されている場合にMエラーとなります。

S Y M B O L 1 E Q U 1 ---- } 有効文字数が6文字のため、同じネーム  
 S Y M B O L 2 E Q U 2 ---- } (SYMBOL)となる。

(8) N (NESTING ERROR)

I F文のネスティングが8レベルを越えていることを示します。

(9) O (ORG ERROR)

O R G直前の割り付けアドレスよりも低いアドレスをO R Gで指定していることを示します。

(10) P (PHASE ERROR)

シンボルに対して最後の最適化パスで割り付けられた値と、P A S S 2以降で割り付けられた値が異なることを示します(最適化パスが完了しなかったことを示します)。

(11) R (REFERENCE ERROR)

分岐命令で参照するアドレスが間違っていることを示します。

(12) S (STACK OVERFLOW)

式の記述が複雑で演算の途中結果をセーブするスタックがオーバーフローしたことを示します。

(13) T

I F文のオペランドが間違っていることを示します。



オペランドに '\$', レーベル, レーベルを使用して定義されたネーム, 前方のネームのいずれかが記述されている場合に T エラーとなります。

(14) U (UNDEFINED SYMBOL)

未定義のシンボルを参照していることを示します。

(15) V (VALUE OVERFLOW)

オペランドに記述された値が命令の要求する範囲を越えていることを示します。

例 LAI 1FFH ……LAI のオペランドは 0~0FH の範囲

(16) W (WARNING)

疑似命令 GJMP および機械語命令 JMP において, JCP 命令が使える可能性のあることを示します。これはエラーではありませんので, エラー数には含まれません (第 3 章 3.8 参照)

## 5.4 アセンブル・リストに印字される エラー・メッセージ一覧

エラー・コード		エラー・メッセージ
A	Address error	ロケーション・カウンタの値が最大値を越えている。
B	Balance error	括弧または引用符の使用が間違っている。
E	Expression error	演算子の記述が間違っている。
F	Format error	オペランドが不足している。
I	Illegal Character	使用できない文字が記述されている。
L	Label or Name error	レーベルまたはネームに予約語を使用している。
M	Multiple definition	同じシンボルが 2 度以上定義されている。
N	Nesting error	IF 文のネスティングが 8 レベルを越えている。
O	ORG error	ORG で指定したアドレスが直前のアドレスよりも低い。
P	Phase error	最適化パス時と PASS2 で異なる値がシンボルに割り付けられた。
R	Reference error	分岐命令で参照するアドレスが間違っている。
S	Stack overflow	式の記述が複雑すぎる
T		IF 文のオペランドの記述が間違っている。
U	Undefined Symbol	未定義のシンボルを参照している。
V	Value overflow	オペランド値が命令の要求範囲を越えている。
W	Warning	JCP 命令が使える可能性がある。

## 5.5 コンソールに表示されるエラー・メッセージ

コンソールに表示されるエラー・メッセージには次の3種類があります。

- (a) 起動コマンドに対するエラー・メッセージ
- (b) アセンブラの内部処理上のエラー・メッセージ
- (c) 致命的なファイル I/O エラーに対するエラー・メッセージ

(1) 起動コマンドに対するエラー・メッセージ

形式： \*\*\* ERROR エラー番号 エラー・メッセージ

PROGRAM ABORTED

エラー番号 F001	メッセージ	MISSING FILE SPECIFICATION
	原因	ソース・モジュール・ファイル名の指定がありません。
	プログラムの処理	OSにコントロールを戻す。
	ユーザの処置	ソース・モジュール・ファイル名を指定して再実行してください。
エラー番号 F002	メッセージ	ILLEGAL FILE SPECIFICATION-(ファイル名)注
	原因	ファイル名が正しくありません。
	プログラムの処理	OSにコントロールを戻す。
	ユーザの処置	正しいファイル名を指定して再実行してください。
エラー番号 F003	メッセージ	FILE SPECIFICATION CONFLICTED-(ファイル名)注
	原因	互いに異なるファイル名を指定しなければならないコントロールに、同じ名前前のファイル名を指定しています。
	プログラムの処理	OSにコントロールを戻す。
	ユーザの処置	正しいファイル名を指定して再実行してください。
エラー番号 F004	メッセージ	FILE ALREADY EXISTS-(ファイル名)注
	原因	指定したファイルが既に存在して、書き込み保護されています。
	プログラムの処理	OSにコントロールを戻す。
	ユーザの処置	書き込みの可能または存在しないファイルを指定して再実行してください。
エラー番号 F005	メッセージ	FILE NOT FOUND-(ファイル名)注
	原因	ソース・モジュール・ファイルが存在しません
	プログラムの処理	OSにコントロールを戻す。
	ユーザの処置	正しいソース・モジュール・ファイル名を指定して再実行してください。
エラー番号 F006	メッセージ	ILLEGAL FILE-(ファイル名)注
	原因	扱えないファイルが指定されています。
	プログラムの処理	OSにコントロールを戻す。
	ユーザの処置	正しいソース・モジュール・ファイル名を指定して再実行してください。

エラー番号 F 0 0 7	メッセージ	ILLEGAL OR MISSING PARAMETER :(パラメータ) 注
	原因	パラメータが必要なコントロールにパラメータが指定されていないか不当なパラメータが指定されています。
	プログラムの処理	OSにコントロールを戻す。
	ユーザの処置	正しいパラメータを指定して再実行してください。
エラー番号 F 0 0 8	メッセージ	CONTROL IS NOT RECOGNIZED :(文字列) 注
	原因	コントロールとして不当な文字列が指定されています。
	プログラムの処理	OSにコントロールを戻す。
	ユーザの処置	正しいコントロールを指定して再実行してください。
エラー番号 F 0 0 9	メッセージ	INVALID SYNTAX
	原因	コントロールの区切りが誤っています。
	プログラムの処理	OSにコントロールを戻す。
	ユーザの処置	正しいコントロールを指定して再実行してください。

注( )でかこまれた部分は、エラーとなった入力イメージが出力されます。

(2) アセンブラの内部処理上のエラー・メッセージ

形式: \*\*\* ERROR エラー番号 メッセージ

エラー番号 A 0 0 1	メッセージ	SYMBOL TABLE OVERFLOW
	原因	シンボルの数が多いためワーク・エリアがなくなりました。
	プログラムの処理	OSにコントロールを戻す。注1
	ユーザの処置	シンボル数を削減してください。
エラー番号 A 0 0 2	メッセージ	SOURCE LINE IS TOO LONG—ALLOWED ONLY UP TO 128
	原因	ソース・ステートメント長がオーバーしています。
	プログラムの処理	OSにコントロールを戻す。注1
	ユーザの処置	ソース・ステートメント文を128文字までに修正してください。
エラー番号 A 0 0 3	メッセージ	MISSING 'END' IN SOURCE MODULE
	原因	ソース・ファイルにEND文がありません。
	プログラムの処理	END文があるものとして処理を続ける。
	ユーザの処置	END文を追加してください。
エラー番号 A 0 0 4	メッセージ	TEXT EXISTS BEHIND END STATEMENT—IGNORE
	原因	ENDのうしろにソース文がかかれています。
	プログラムの処理	END以降を無効行としてコメント扱いにする。
	ユーザの処置	END以降の文を無効行として扱ってよいか確認してください。
エラー番号 A 0 0 5	メッセージ	DUPLICATE CONTROL—(コントロール名) 注2
	原因	コントロールを2重に指定しています。
	プログラムの処理	コマンド・ラインからの入力を優先させ、パラメータ・ファイルからの入力を無視する。
	ユーザの処置	なし

注1 'PROGRAM ABORTED' とメッセージを出力してから OS にコントロールを戻します。

注2 ( ) でかこまれた部分はエラーとなった入力イメージが出力されます。

(3) 致命的なファイル I/O エラーに対するエラー・メッセージ

形式: FATAL I/O ERROR -

DEVICE: 装置名

FILE NAME: ファイル名

ERROR: エラー・メッセージ

PROGRAM ABORTED

メッセージ	READ ERROR
原因	フロッピー・ディスクの読み込みエラーが起りました。
プログラムの処理	OS にコントロールを戻す。
ユーザの処置	なし
メッセージ	WRITE ERROR
原因	フロッピー・ディスクに書き込みエラーが起りました。
プログラムの処理	OS にコントロールを戻す。
ユーザの処置	新しいフロッピー・ディスクを使用してください。
メッセージ	NO DIRECTORY EXIST
原因	フロッピー・ディスクにファイルを登録するディレクト・エリアが足りません。
プログラムの処理	OS にコントロールを戻す。
ユーザの処置	新しいフロッピー・ディスクを使用してください。
メッセージ	END OF VOLUME
原因	フロッピー・ディスク上にファイルを書き込むエリアが足りません。
プログラムの処理	OS にコントロールを戻す。
ユーザの処置	新しいフロッピー・ディスクを使用してください。
メッセージ	FILE CLOSE FAILED
原因	ファイルがクローズできません。
プログラムの処理	OS にコントロールを戻す。
ユーザの処置	なし
メッセージ	SELECT ERROR
原因	存在しないデバイス名を指定しています。
プログラムの処理	OS にコントロールを戻す。
ユーザの処置	デバイス名を直してください。



## 付録 I 予約語

予約語には、機械語命令、疑似命令、演算子、レジスタ名の4種類があります。予約語は、アセンブラの識別のためにあらかじめ予約してある文字列で、指定の目的以外には転用できません。

ソース・モジュールの各欄に記述できる予約語の種類と予約語一覧を次に示します。

シンボル欄	すべての予約語が記述できない。
ニモニック欄	機械語命令および疑似命令のみ記述できる。
オペランド欄	演算子およびレジスタ名のみ記述できる。
コメント欄	すべての予約語が記述できる。

予約語一覧 (μPD7500セットA)

命	ACSC	ADSC	AESC	AHSC	AISC	ALSC		
	ANL	ANP	ASC	CALL	CALT	CMA	DDE	
	DDRS	DES	DHL	DI	DLS	EI	EXL	HALT
	IDE	IDRS	IES	IHL	ILS	IP	IPL	IP1
	IP54	JAM	JCP	JMP	JMPL	L	LADR	
	LAI	LAM	LAMT	LAMTL	LDEI	LDI		
	LDS	LEI	LHI	LHLI	LHLT	LIS	LLI	
	NOP	OP	OPL	OP3	OP54	OPL	ORP	
	POPDE	POPHL		PSHDE	PSHHL	RAL		
	RAR	RC	RMB	RPBL	RT	RTPSW	RTS	
	SC	SDSB	SESB	SHSB	SIO	SKABT		
	SKAEI	SKAEM		SKC	SKDEI	SKEEI		
	SKHEI	SKI	SKLEI	SKMBF	SKMBT			
	SKMEI	SLSB	SMB	SPBL	ST			
	STOP	TAD	TAE	TAH	TAL			
	TAMMOD	TAMSIO	TAMSP	TCNTAM				
	TDA	TEA	THA	TIMER	TLA	TSIOAM		
	TSPAM	X	XAD	XADR	XAE	XAH	XAL	
	XAM	XDS	XHDR	XIS	XLDR			
	疑似命令	DB	DET	EJECT	ELSE	END	ENDIF	
EQU		GJMP	IF					
ORG		PRINT	SET	SPACE				
TITLE								
演算子	AND	EQ	MOD	NE	NOT	OR	SHL	SHR
	XOR							
レジスタ	HL	HL+	HL-	DE	DL			

予約語一覧 ( μPD7500セットB )

命 令	ACSC	AISC	ANL	ASC	CAL	CALL	
	CMA	DDRS	DLS	EXL	HALT	IDRS	
	ILS	IP	IPL	IP1	IP54	JAM	JCP
	JMP	JMPL	L	LADR	LAI	LAM	LAMT
	LAMTL	LDS	LHI	LHLI	LIS	NOP	
	OP	OPL	OP3	OP54	ORL	RAL	RAR
	RC	RMB	RPBL	RT	RTS	SC	SIO
	SKABT	SKAEI	SKAEM	SKC	SKI		
	SKLEI	SKMBF	SKMBT	SKMEI	SMB		
	SPPL	ST	STII	STOP	TAMMOD		
	TAMSIO	TAMSP	TCNTAM	TIMER			
	TSIOAM	TSPAM	X XADR	XADR	XAH	XAL	
	XAM	XDS	XHDR	XIS	XLDR		
疑似 命令	DB	EJECT	ELSE	END	ENDIF		
	EQU	GJMP	IF				
	ORG	PRINT	SET	SPACE	TITLE		
演算 子	AND	EQ	MOD	NE	NOT	OR	SHL SHR
	XOR						
レジ スタ	HL	HL+	HL-				



予約語一覧 (μPD7554, 7564)

命 令	ACSC	AISC	ASC	CAL	CALL	
	CMA	DDRS	DLS	EXL	HALT	
	IDRS	ILS	IPL	JCP	JMP	L
	LAI	LAM	LDS	LHI	LHLI	
	LIS	NOP	OPL	RC	RMB	RPBL
	RT	RTS	SC	SIO	SKABT	
	SKAEI	SKAEM	SKC	SKI	SKMBF	
	SKMBT	SMB	SPBL	ST	STII	
	STOP	TAMSIO	TAMSP	TCNTAM	TIMER	
	TSIOAM	X	XAL	XAM	XDS	XIS
疑 似 命 令	DB	EJECT	ELSE	END	ENDIF	
	EQU	GJMP	IF	ORG		
	PRINT	SET	SPACE	TITLE		
演 算 子	AND	EQ	MOD	NE	NOT	OR
	XOR				SHL	SHR
レ ジ ス タ	HL	HL+	HL-			

予約語一覧 ( μPD7556, 7566 )

命 令	ACSC	AISC	ASC	CAL	CALL	
	CMA	DDRS	DLS	EXL	HALT	
	IDRS	ILS	IP1	IPL	JCP	
	JMP	L	LAI	LAM	LDS	
	LHI	LHLI	LIS	NOP	OPL	RC
	RMB	RPBL	RT	RTS	SC	SKABT
	SKAEI	SKAEM	SKC	SKI	SKMBF	
	SKMBT	SMB	SPBL	ST	STII	
	STOP	TAMSP	TCNTAM	TIMER	X	XAL
	XDS	XIS				
疑 似 命 令	DB	EJECT	ELSE	END	ENDIF	
	EQU	GJMP	IF	ORG		
	PRINT	SET	SPACE	TITLE		
演 算 子	AND	EQ	MOD	NE	NOT	OR SHL SHR
	XOR					
レ ジ ス タ	HL	HL+	HL-			

## 付録II シンボル数

本プログラム中で使用できるシンボルの最大数は、約4800個です。

付録III インストラクション

μPD7500 SET A インストラクション活用表

Table of instructions for μPD7500 SET A, including columns for mnemonic, opcode, address, operation, and skip conditions. Instructions include LAI, LDI, LEI, LHI, LLI, LAM, LADR, LDEI, LHLL, LHLT, LAMT, LAMTL, ST, TAD, TAE, TAH, TAL, TDA, TEA, THA, TLA, XAD, XAE, XAH, XAL, XAM, XADR, XHDR, XLDR, AISC, ASC, ACSC, ASC, AESC, AHSC, ALSC, SDSB, SESB, SHSB, SLSC, EXL, ANL, ORL, CMA, RAR, RAL, RC, SC, IES, ILS, IDE, IHL, IDRS, DES, DLS, DDE, DHL, DDRS.

Table of instructions for μPD7550, 7560, including columns for mnemonic, opcode, address, operation, and skip conditions. Instructions include RMB, SMB, JMP, JMPL, JCP, JAM, CALL, CALT, RT, RTS, RTPSW, PSHDE, PSHHL, POPDE, POPHL, TAMSP, TSPAM, SKC, SKMBT, SKABT, SKMBF, SKAEM, SKAEI, SKDEI, SKEEI, SKHEI, SKLEI, SKMEI, TAMSIO, TSIOAM, SIO, TAMMOD, TIMER, TCNTAM, EI, DI, SKI, IPL, IP, IP1, OP54, OPFL, OP, OP3, OP54, ANP, ORP, RPBL, SPBL, HALT, STOP, NOP.

## μ PD7500 SET B インストラクション活用表

命令群	ニモニック	命令コード	オペレーション	スキップ条件
ロ	LAI n4	0 0 0 1 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	A←n4 n4 0 FH	縦格LAI
	LHI n3	0 0 1 0 I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	H←0I <sub>2</sub> I <sub>1</sub> I <sub>0</sub> n3 0 7	
	LAM pr1	0 1 0 1 0 0 R <sub>1</sub> R <sub>0</sub>	A←(pr1) pr1-HL, HL+, HL	L←FH(HL-) L=0(HL+)
	LADR mem	0 0 1 1 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	A←(mem) mem 0 FFH	
	LHLI n5	1 1 0 1 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	HL←000I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub> n5 0 1FH	縦格LHLI
	LAMT	0 1 0 1 1 1 1 0	A←TABLE(BNK, PC <sub>n+4</sub> , 0, C, A <sub>3</sub> sh) (HL)←TABLE(BNK, PC <sub>n+4</sub> , 0, C, A <sub>3</sub> sh)	
	LAMTL	0 0 1 1 0 0 1 1 1 1 1 1 0 1 0 0	A←TABLE(BNK, PC <sub>n+4</sub> , A <sub>3</sub> , (HL) sh) (HL)←TABLE(BNK, PC <sub>n+4</sub> , A <sub>3</sub> , (HL) sh)	
	ST	0 1 0 1 0 1 1 1	(HL)←A	
	STII n4	0 1 0 0 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	(HL)←n4, L←L+1 n4 0-FH	
	XAH	0 1 1 1 1 0 1 0	A←H	
XAL	0 1 1 1 1 0 1 1	A←L		
XAM pr1	0 1 0 1 0 1 R <sub>1</sub> R <sub>0</sub>	A←(pr1) pr1-HL, HL+, HL	L←FH(HL-) L=0(HL+)	
XADR mem	0 0 1 1 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	A←(mem) mem 0 FFH		
XHDR mem	0 0 1 1 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	H←(mem) mem 0 FFH		
XLDR mem	0 0 1 1 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	L←(mem) mem=0 FFH		
演算命令	AISC n4	0 0 0 0 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	A←A+n4 n4 0 FH	carry
	ASC	0 1 1 1 1 1 0 1	A←A+(HL)	carry
	ACSC	0 1 1 1 1 1 0 0	A, C←A+(HL)+C	carry
	EXL	0 1 1 1 1 1 1 0	A←A∨(HL)	
	ANL	0 0 1 1 1 0 1 1 0 0 1 0	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1 0 1 1 0	A←A∨(HL)	
フラグ操作命令	CMA	0 1 1 1 1 1 1 1	A← $\bar{A}$	
	RAR	0 0 1 1 1 0 1 1 0 0 1 1	C←A <sub>0</sub> , A <sub>3</sub> ←C An←An+1 (n=1 3)	
	RAL	0 0 1 1 1 0 1 1 0 1 1 1	C←A <sub>3</sub> , A <sub>0</sub> ←C An←An-1 (n=1 3)	
	RC	0 1 1 1 1 0 0 0	C←0	
	SC	0 1 1 1 1 0 0 1	C←1	
	増減命令	ILS	0 1 0 1 1 0 0 1	L←L+1
IDRS mem		0 0 1 1 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	(mem)←(mem)+1	(mem)=0
DLS		0 1 0 1 1 0 0 0	L←L-1	L=FH
DDRS mem		0 0 1 1 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	(mem)←(mem)-1	(mem)=FH
ビット操作命令	RMB bit	0 1 1 0 1 0 B <sub>1</sub> B <sub>0</sub>	(HL)bit←0 bit 0 3	
	SMB bit	0 1 1 0 1 1 B <sub>1</sub> B <sub>0</sub>	(HL)bit←1 bit 0 3	
ジャンプ命令	JMP addr2	0 0 1 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	PC <sub>n</sub> ←P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	JMPL addr	0 0 1 1 0 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	BNK←P <sub>3</sub> PC <sub>n</sub> ←P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	JCP addr	1 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	PC <sub>5</sub> ←P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	JAM addr3	0 0 1 1 0 0 0 1 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	PC <sub>n</sub> ←P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> , PC <sub>7</sub> ←A <sub>3</sub> P <sub>0</sub> PC <sub>3</sub> ←(HL) <sub>3</sub> P <sub>0</sub>	

命令群	ニモニック	命令コード	オペレーション	スキップ条件	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	(SP 1)(SP 2)(SP 4)←PC <sub>n</sub> (SP 3)←PSW, SP←SP-4 BNK←0, PC <sub>n</sub> ←0, P <sub>10</sub> ←0	
	CAL caddr1	1 1 1 P <sub>3</sub>	P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	(SP 1)(SP 2)(SP 4)←PC <sub>n+4</sub> (SP 3)←PSW, SP←SP-4 BNK←0 PC <sub>n+4</sub> ←0001P <sub>3</sub> 000P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	RT	0 1 0 1 0 0 1 1	2	PC <sub>n</sub> ←(SP)(SP+2)(SP+3) BNK←(SP+1), SP←SP+4	
	RTS	0 1 0 1 1 0 1 1	3	PC <sub>n</sub> ←(SP)(SP+2)(SP+3) BNK←(SP+1), SP←SP+4 then skip unconditionally	無条件
	TAMSP	0 0 1 1 0 0 1 1 1 1 1 1 0 0 0 1	2	SP <sub>7</sub> ←A SP <sub>3-1</sub> ←(HL) <sub>3-1</sub> , SP <sub>0</sub> ←0	
	TSPAM	0 0 1 1 0 0 1 1 1 1 1 1 0 1 0 1	2	A←SP <sub>7</sub> (HL) <sub>3-1</sub> ←SP <sub>3-1</sub> , (HL) <sub>0</sub> ←0	
	SKC	0 1 0 1 1 0 1 0	1/2	Skip if C=1	C=1
	SKABT bit	0 1 1 1 0 1 B <sub>1</sub> B <sub>0</sub>	1/2	Skip if A <sub>bit</sub> =1 bit=0-3	A <sub>bit</sub> =1
	SKMBT bit	0 1 1 0 0 1 B <sub>1</sub> B <sub>0</sub>	1/2	Skip if (HL) <sub>bit</sub> =1 bit=0-3	(HL) <sub>bit</sub> =1
	SKMBF bit	0 1 1 0 0 0 B <sub>1</sub> B <sub>0</sub>	1/2	Skip if (HL) <sub>bit</sub> =0 bit=0-3	(HL) <sub>bit</sub> =0
スキップ命令	SKAEM	0 1 0 1 1 1 1 1	1/2	Skip if A=(HL)	A=(HL)
	SKAEI n4	0 0 1 1 0 1 1 0 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	1/2	Skip if A=n4 n4=0-FH	A=n4
	SKLEI n4	0 0 1 1 0 1 0 1 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	1/2	Skip if L=n4 n4=0-FH	L=n4
	SKMEI n4	0 0 1 1 0 1 1 1 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	1/2	Skip if (HL)=n4 n4=0-FH	(HL)=n4
	SKI n4	0 0 1 1 0 1 0 0 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	1/2	Skip if INT RQF=1 then reset INT RQF	INT RQF=1
	SIO制御命令	TAMSIO	0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 0	2	SIO <sub>n</sub> ←A, SIO <sub>t</sub> ←(HL)
		TSIOAM	0 0 1 1 0 0 1 1 1 1 1 1 1 0 1 0	2	A←SIO <sub>n</sub> , (HL)←SIO <sub>t</sub>
		SIO	0 0 1 1 0 0 1 1 1 1 1 1 0 0 1 1	2	Start SIO
	タイマ制御命令	TAMMOD	0 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1	2	MOD <sub>n</sub> ←A, MOD <sub>t</sub> ←(HL)
		TIMER	0 0 1 1 0 0 1 1 1 1 1 1 0 0 1 0	2	CT <sub>7-0</sub> ←0, INTT RQF←0
TCNTAM		0 0 1 1 0 0 1 1 1 1 1 1 1 0 1 1	2	A←CT <sub>7</sub> , (HL)←CT <sub>3-0</sub>	
入出力命令		IPL	0 1 1 1 0 0 0 0	1	A←Port(L)
	IP addr5	0 0 1 1 1 1 0 0 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	2	A←Port(addr5) addr5=0, 1, 4-BH	
	IP1	0 1 1 1 0 0 0 1	1	A←Port1	
	IP54	0 0 1 1 0 0 1 1 1 1 1 1 1 0 0 0	2	A←Port5, (HL)←Port4	
	OPL	0 1 1 1 0 0 1 0	1	Port/Mode reg.(L)←A	
	OP addr4	0 0 1 1 1 1 1 0 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	2	Port/Mode reg.(addr4)←A addr4=1 FH	
	OP3	0 1 1 1 0 0 1 1	1	Port3←A	
	OP54	0 0 1 1 0 0 1 1 1 1 1 1 1 1 0 0	2	Port5←A, Port4←(HL)	
	RPBL	0 1 0 1 1 1 0 0	1	8243 Port bit(L)←0	
	SPBL	0 1 0 1 1 1 0 1	1	8243 Port bit(L)←1	
CPU制御命令	HALT	0 0 1 1 0 0 1 1 1 1 1 1 0 1 1 0	2	Set Halt Mode	
	STOP	0 0 1 1 0 0 1 1 1 1 1 1 0 1 1 1	2	Set Stop Mode	
	NOP	0 0 0 0 0 0 0 0	1	No Operatio	

μPD7554/64 インストラクション活用表

命令群	ニモニック	命令コード	アドレス	オペレーション	スキップ条件
ロード・ストア命令	LAI	n4	0 0 0 1 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	1 A←n4 n4=0-FH	縦横LAI
	LHI	n2	0 0 1 0 1 0 I <sub>1</sub> I <sub>0</sub>	1 H←I <sub>1</sub> I <sub>0</sub> n2=0-3	
	LAM	pr	0 1 0 1 0 0 R <sub>3</sub> R <sub>2</sub>	½ A←(pr) pr=HL-, HL+, HL	L=FH(HL-) L=0(HL+)
	LHLI	n5	1 1 0 I <sub>4</sub> I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	1 HL←000I <sub>4</sub> I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub> n5=0-1FH	縦横LHLI
	ST		0 1 0 1 0 1 1 1	1 (HL)←A	
	STH	n4	0 1 0 0 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	1 (HL)←n4, L←L+1 n4=0-FH	
	XAL		0 1 1 1 1 0 1 1	1 A←L	
	XAM	pr	0 1 0 1 0 1 R <sub>3</sub> R <sub>2</sub>	½ A←(pr) pr=HL-, HL+, HL	L=FH(HL-) L=0(HL+)
演算命令	AISC	n4	0 0 0 0 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	½ A←A+n4 n4=0-FH	carry
	ASC		0 1 1 1 1 1 0 1	½ A←A+(HL)	carry
	ACSC		0 1 1 1 1 1 0 0	½ A, C←A+(HL)+C	carry
	EXL		0 1 1 1 1 1 1 0	1 A←A∨(HL)	
アキームレジスタ操作命令	CMA		0 1 1 1 1 1 1 1	1 A← $\bar{A}$	
	RC		0 1 1 1 1 0 0 0	1 C←0	
	SC		0 1 1 1 1 0 0 1	1 C←1	
増減命令	ILS		0 1 0 1 1 0 0 1	½ L←L+1	L=0
	IDRS	mem	0 0 1 1 0 0 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	½ (mem)←(mem)+1 mem=0-3FH	(mem)=0
	DLS		0 1 0 1 1 0 0 0	½ L←L-1	L=FH
命令	DDRS	mem	0 0 1 1 0 0 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	½ (mem)←(mem)-1 mem=0-3FH	(mem)=FH
	メモリビット操作命令	RMB	bit	0 1 1 0 1 0 B <sub>3</sub> B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	1 (HL)bit←0 bit=0-3
SMB		bit	0 1 1 0 1 1 B <sub>3</sub> B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	1 (HL)bit←1 bit=0-3	
ジャンプ命令	JMP	addr	0 0 1 0 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2 PC <sub>3-0</sub> ←P <sub>3-0</sub>	
	JCP	addr	1 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	1 PC <sub>3-0</sub> ←P <sub>3-0</sub>	

命令群	ニモニック	命令コード	アドレス	オペレーション	スキップ条件	
サブルーチン・スタック制御命令	CALL	caddr	0 0 1 1 P <sub>7</sub> P <sub>6</sub> P <sub>5</sub> P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2 (SP-1)/(SP-2)/(SP-4)←PC <sub>3-0</sub> (SP-3)←PSW, SP←SP-4 PC <sub>3-0</sub> ←P <sub>3-0</sub>		
	CAL	caddr1	1 1 1 P <sub>4</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2 (SP-1)(SP-2)(SP-4)←PC <sub>3-0</sub> (SP-3)←PSW, SP←SP-4 PC <sub>3-0</sub> ←01P <sub>4</sub> P <sub>3</sub> 000P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>		
	RT		0 1 0 1 0 0 1 1	2 PC <sub>3-0</sub> ←(SP)/(SP+2)/(SP+3) SP←SP+4		
	RTS		0 1 0 1 1 0 1 1	3 PC <sub>3-0</sub> ←(SP)/(SP+2)/(SP+3) SP←SP+4 then skip unconditionally	無条件	
	TAMSP		0 0 1 1 0 0 1 1	2 SP <sub>3-0</sub> ←A <sub>1-0</sub> SP <sub>3-0</sub> ←(HL) <sub>3-1</sub> , SP <sub>0</sub> ←0		
	SKC		0 1 0 1 1 0 1 0	½ Skip if C=1	C=1	
	SKABT	bit	0 1 1 1 0 1 B <sub>3</sub> B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	½ Skip if A <sub>bit</sub> =1 bit=0-3 A <sub>bit</sub> =1		
	SKMBT	bit	0 1 1 0 0 1 B <sub>3</sub> B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	½ Skip if (HL) <sub>bit</sub> =1 bit=0-3 (HL) <sub>bit</sub> =1		
スキップ命令	SKMBF	bit	0 1 1 0 0 0 B <sub>3</sub> B <sub>2</sub> B <sub>1</sub> B <sub>0</sub>	½ Skip if (HL) <sub>bit</sub> =0 bit=0-3 (HL) <sub>bit</sub> =0		
	SKAEM		0 1 0 1 1 1 1 1	½ Skip if A=(HL)	A=(HL)	
	SKAEI	n4	0 0 1 1 0 1 1 1 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	½ Skip if A=n4 n4=0-FH	A=n4	
	SKI	n2	0 0 1 1 0 1 0 0 I <sub>1</sub> I <sub>0</sub>	½ Skip if INT RQF=1 then reset INT RQF	INT RQF=1	
	TAMSIO		0 0 1 1 0 0 1 1	2 SIO <sub>H</sub> ←A, SIO <sub>L</sub> ←(HL)		
	TSIOAM		0 0 1 1 0 0 1 0	2 A←SIO <sub>H</sub> , (HL)←SIO <sub>L</sub>		
	SIO		0 0 1 1 0 0 1 1	2 Start SIO		
	TIMER		0 0 1 1 0 0 1 0	2 Start Timer CT <sub>7-0</sub> ←0, INTT RQF←0		
タイマ制御命令	TCNTAM		0 0 1 1 0 0 1 1	2 A←CT <sub>7-4</sub> , (HL)←CT <sub>3-0</sub>		
	IPL		0 1 1 1 0 0 0 0	1 A←Port(L)		
	OPL		0 1 1 1 0 0 1 0	1 Port/Mode reg.(L)←A		
	RPBL		0 1 0 1 1 1 0 0	1 Port Bit(L)←0		
入出力命令	SPBL		0 1 0 1 1 1 0 1	1 Port Bit(L)←1		
	CPU制御命令	HALT		0 0 1 1 0 0 1 1	2 Set Halt Mode	
		STOP		0 0 1 1 0 0 1 1	2 Set Stop Mode	
NOP			0 0 0 0 0 0 0 0	1 No Operation		

## μPD7556/66 インストラクション活用表

命令群	ニモニック	命令コード	アドレス	オペレーション	スキップ条件
ロード・ストア命令	LAI	n4	0 0 0 1 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	1 A←n4	n4=0-FH 縦横LAI
	LHI	n2	0 0 1 0 1 0 I <sub>1</sub> I <sub>0</sub>	1 H←I <sub>1</sub> I <sub>0</sub>	n2=0-3
	LAM	pr	0 1 0 1 0 0 R <sub>1</sub> R <sub>0</sub>	½ A←(pr) pr=HL-, HL+, HL	L= FH(HL-) L=0(HL+)
	LHLI	n5	1 1 0 1 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	1 HL←000I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	n5=0-1FH 縦横LHLI
	ST		0 1 0 1 0 1 1 1	1 (HL)←A	
	STH	n4	0 1 0 0 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	1 (HL)←n4, L←L+1	n4=0-FH
	XAL		0 1 1 1 1 0 1 1	1 A↔L	
	XAM	pr	0 1 0 1 0 1 R <sub>1</sub> R <sub>0</sub>	½ A←(pr) pr=HL-, HL+, HL	L= FH(HL-) L=0(HL+)
演算命令	AISC	n4	0 0 0 0 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	½ A←A+n4	n4=0-FH carry
	ASC		0 1 1 1 1 1 0 1	½ A←A+(HL)	carry
	ACSC		0 1 1 1 1 1 0 0	½ A, C←A+(HL)+C	carry
	EXL		0 1 1 1 1 1 1 0	1 A←A∨(HL)	
アキユームレタキヤ リーフタグ操作命令	CMA		0 1 1 1 1 1 1 1	1 A←Ā	
	RC		0 1 1 1 1 0 0 0	1 C←0	
	SC		0 1 1 1 1 0 0 1	1 C←1	
増減命令	ILS		0 1 0 1 1 0 0 1	½ L←L+1	L=0
	IDRS	mem	0 0 1 1 0 0 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	½ (mem)←(mem)+1 mem=0-3FH	(mem)=0
	DLS		0 1 0 1 1 0 0 0	½ L←L-1	L= FH
	DDRS	mem	0 0 1 1 0 0 D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	½ (mem)←(mem)-1 mem=0-3FH	(mem)=FH
メモリビット 操作命令	RMB	bit	0 1 1 0 1 0 B <sub>1</sub> B <sub>0</sub>	1 (HL)bit←0	bit=0-3
	SMB	bit	0 1 1 0 1 1 B <sub>1</sub> B <sub>0</sub>	1 (HL)bit←1	bit=0-3
ジャンプ命令	JMP	addr	0 0 1 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2 PC <sub>3-0</sub> ←P <sub>3-0</sub>	
	JCP	addr	1 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	1 PC <sub>5-0</sub> ←P <sub>3-0</sub>	

命令群	ニモニック	命令コード	アドレス	オペレーション	スキップ条件	
サブルーチン・スタック制御命令	CALL	caddr	0 0 1 1 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2 0 0 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	(SP-1)←(SP-2)←(SP-4)←PC <sub>3-0</sub> (SP-3)←PSW, SP←SP-4 PC <sub>3-0</sub> ←P <sub>3-0</sub>	
	CAL	caddr1	1 1 1 1 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	2 P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	(SP-1)←(SP-2)←(SP-4)←PC <sub>3-0</sub> (SP-3)←PSW, SP←SP-4 PC <sub>3-0</sub> ←01P <sub>3</sub> 000P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>	
	RT		0 1 0 1 0 0 1 1	2	PC <sub>3-0</sub> ←(SP)←(SP+2)←(SP+3) SP←SP+4	
	RTS		0 1 0 1 1 0 1 1	3	PC <sub>3-0</sub> ←(SP)←(SP+2)←(SP+3) SP←SP+4 then skip unconditionally	無条件
	TAMSP		0 0 1 1 0 0 1 1 0 0 1 1 0 0 0 1	2	SP <sub>3-0</sub> ←A <sub>3-0</sub> SP <sub>3-1</sub> ←(HL) <sub>3-1</sub> , SP <sub>0</sub> ←0	
	SKC		0 1 0 1 1 0 1 0	½	Skip if C=1	C=1
	SKABT	bit	0 1 1 1 0 1 B <sub>1</sub> B <sub>0</sub>	½	Skip if A <sub>bit</sub> =1	bit=0-3 A <sub>bit</sub> =1
	SKMBT	bit	0 1 1 0 0 1 B <sub>1</sub> B <sub>0</sub>	½	Skip if (HL) <sub>bit</sub> =1	bit=0-3 (HL) <sub>bit</sub> =1
スケキ スケキ スケキ スケキ スケキ スケキ スケキ スケキ	SKMBF	bit	0 1 1 0 0 0 B <sub>1</sub> B <sub>0</sub>	½	Skip if (HL) <sub>bit</sub> =0	bit=0-3 (HL) <sub>bit</sub> =0
	SKAEM		0 1 0 1 1 1 1 1	½	Skip if A=(HL)	A=(HL)
	SKAEI	n4	0 0 1 1 0 1 1 0 I <sub>3</sub> I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>	½	Skip if A=n4	n4=0-FH A=n4
	SKI	n2	0 0 1 1 0 1 0 0 0 0 1 0 I <sub>1</sub> I <sub>0</sub>	½	Skip if INT RQF=1 then reset INT RQF	INT RQF=1
	TIMER		0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 0	2	Start Timer CT <sub>7-0</sub> ←0, INTT RQF←0	
	TCNTAM		0 0 1 1 0 0 1 1 0 0 1 1 1 0 1 1	2	A←CT <sub>7-4</sub> , (HL)←CT <sub>3-0</sub>	
	IPL		0 1 1 1 0 0 0 0	1	A←Port(L)	
	IP1		0 1 1 1 0 0 0 1	1	A←Port1	
入出力命令	OPL		0 1 1 1 0 0 1 0	1	Port Mode reg.(L)←A	
	RPBL		0 1 0 1 1 1 0 0	1	Port Bit(L)←0	
	SPBL		0 1 0 1 1 1 0 1	1	Port Bit(L)←1	
	CPU制御命令	HALT		0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 0	2	Set Halt Mode
STOP			0 0 1 1 0 1 1 1 0 0 1 1 0 1 1 1	2	Set Stop Mode	
NOP			0 0 0 0 0 0 0 0	1	No Operation	

# — NEC 日本電気株式会社 —

本社	〒108-01 東京都港区芝五丁目7番1号(日本電気本社ビル)	大 阪	06-945-3178
半導体		大 阪	06-945-3200
第一、第二 販売事業部	〒108-01 東京都港区芝五丁目7番1号(日本電気本社ビル) 東京	大 阪	03-3454-1111
関西支社 半導体販売部	〒540 大阪市中央区城見一丁目4番24号(日本電気関西ビル)	大 阪	06-945-3178
		大 阪	06-945-3200
中部支社 半導体販売部	〒480 名古屋市中区栄四丁目14番5号(松下ビル) 名古屋	大 阪	052-242-2755

北海道支社	札幌	011-231-0161	立川支社	立川	0425-26-0911
東北支社	仙台	022-261-5511	千葉支社	千葉	0472-27-5441
関東支社	東京	0196-51-4344	横浜支社	横浜	054-255-2211
北関東支社	宇都宮	0236-23-5511	川崎支社	川崎	0559-63-4455
中部支社	名古屋	0249-23-5511	相模原支社	相模原	0534-52-2711
近畿支社	大阪	0246-21-5511	松戸支社	松戸	0762-23-1621
中国支社	岡山	0258-36-2155	武蔵野支社	武蔵野	0776-22-1866
四国支社	高松	0292-26-1717	府中支社	府中	0764-31-8461
九州支社	福岡	045-324-5511	所沢支社	所沢	078-332-3311
		0273-26-1255	狭山市支社	狭山市	075-221-8511
		0276-46-4011	東武野田支社	東武野田	078-332-3311
		0286-21-2281	東武東上支社	東武東上	082-242-5504
		0285-24-5011	東武東横支社	東武東横	0857-27-5311
		0262-35-1444	東武東横支社	東武東横	0862-25-4455
		0263-35-1666	東武東横支社	東武東横	0878-36-1200
		0266-53-5350	東武東横支社	東武東横	0897-32-5000
		0552-24-4141	東武東横支社	東武東横	0899-45-4111
		048-641-1411	東武東横支社	東武東横	092-271-7700
			東武東横支社	東武東横	093-541-2887

(技術お問い合わせ先)

半導体応用技術本部 第一応用システム技術部	〒108-01 東京都港区芝五丁目7番1号(日本電気本社ビル)	電 話	03-3798-6105
半導体応用技術本部 第二応用システム技術部	〒540 大阪市中央区城見一丁目4番24号(日本電気関西ビル)	大 阪	06-945-3383
半導体応用技術本部	〒210 川崎市幸区幸町三丁目484番地(川崎技術センター)	電 話	044-533-1111

インフォメーションセンター  
FAX(044) 548-7900  
(24時間受付)