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い申し上げます。

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ものですが、誤りがな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て、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が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。

ユーザーズ・マニュアル

RENESAS

保守／廃止

μPD7500シリーズ
アセンブラ

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

資料番号　EEM-647C（第4版）
発行年月　June 1991 P

ユーザーズ・マニュアル

NEC
電子デバイス

保守／廃止

μPD7500シリーズ
アセンブラ

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

保守／廃止

V30™は日本電気株式会社の商標です.
MS−DOS™は米国マイクロソフト社の商標です.
PC DOS™, PC／XT™は米国IBM社の商標です.

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。

保守／廃止

本版で改訂された主な箇所

箇　所	内　容
全　般	μPD7502A，7503A，7507B，7507H， 7507S，7508B，7508H，7519H， 7527A，7528A，7537A，7538A， 7516Hを追加

巻末にアンケート・コーナを設けております。このドキュメントに関するご意見をお気軽にお寄せください。

ま　え　が　き

1. μPD７５００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．

μPD７５００シリーズ・アセンブラーⅠ

- μPD７５００セットA
- μPD７５００セットB
- μPD７５２０
- μPD７５０１
- μPD７５０２，７５０２A
- μPD７５０３，７５０３A
- μPD７５０６
- μPD７５０７，７５０７B，７５０７H，７５０７S
- μPD７５０８，７５０８B，７５０８H

μPD７５００シリーズ・アセンブラーⅡ

- μPD７５００セットA
- μPD７５００セットB
- μPD７５１４
- μPD７５１９，７５１９H
- μPD７５２７，７５２７A
- μPD７５２８，７５２８A
- μPD７５３７，７５３７A
- μPD７５３８，７５３８A
- μPD７５１６，７５１６H
- μPD７５３３

2. 供給ファイル名

ASM75．COM
ASM75．OM0　} ……μPD7500シリーズ　アセンブラーⅠ

ASM75F．COM
ASM75F．OM0　} ……μPD7500シリーズ　アセンブラーⅡ

保守／廃止

ホスト・マシン	OS	供給媒体
PC−9800シリーズ 注1	MS−DOS　V2.11 注2　V3.10	フロッピィ・ディスク 3.5”2HD，5”2HD
IBM PC， PC／XT™	PC DOS　V3.1	フロッピィ・ディスク 5”2HC

注1. PC−9800シリーズの対象機種を次の頁に示します（１９９１年５月現在）.

2. μPD7500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（FILES）が８以上に設定されている必要があります.

CPU	8086／V30™	80286		80386	
モード	ノーマル	ノーマル	ハイレゾ	ノーマル	ハイレゾ
サポート機種	E F 1/2/3 M 2/3 VF 2 VM 0/2/4/21/11 U 2 UV 2/21/11 CV 21 UR /20 UF		XA model 1/2/3/11/21/31		
	XL model 1/2/4 VX 0/2/4/01/21/41 UX 21/41 RX 2/4/21/51 EX 2/4	XL model 1/2/4 VX 0/2/4/01/21/41 UX 21/41 RX 2/4/21/51 EX 2/4 DX 2/5/U2/U5	XL model 1/2/4		
	XL² RL 2/5/21/51 RA 2/5/21/51 ES 2/5 RS 21/51 T model W2/W5/S5/F5 LV 21/22 LX 2/4/5 LS 2/5 n nV	LX 2/4/5		XL² RL 2/5/21/51 RA 2/5/21/51 ES 2/5 RS 21/51 T model W2/W5/S5/F5 LS 2/5 nS/20	XL² RL 2/5/21/51

- 256Kバイトのメモリが必要です.
- 本プログラムは，当社提供のPC−9800シリーズ用MS−DOS上で動作します．このほかの市販ＯＳ上での動作には責任を負いかねますので，ご了承ください.
- PC−9801F1／F2／F3，VF2でご使用の際には，当社製3.5インチ，5インチ（2HD）増設ドライブをご使用ください.

保守／廃止

目　次

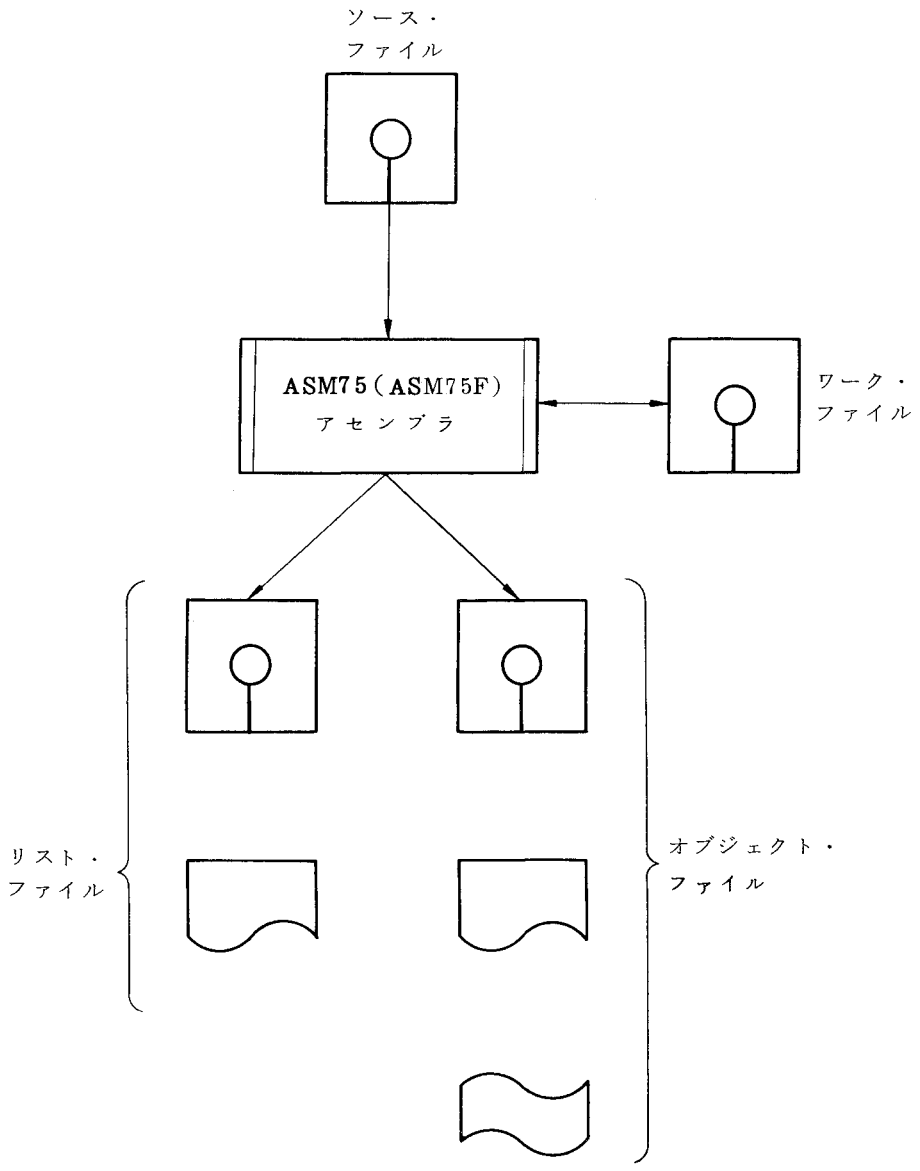
保守／廃止

第1章　ソフトウエア概要

1．1　アセンブラの機能概要

本アセンブラはμPD7500シリーズの命令とアセンブラを制御する疑似命令，さらに必要によってはマクロ名を用いて作成されたシンボリック・ソース・プログラム（ソース・コード）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・ファイルを入力し，オブジェクト・ファイルとリスト・ファイルを出力します．



保守／廃止

1.2　パス方式

本アセンブラは3PASSアセンブラです.

(1)　PASS－1

ソース・プログラムを読みシンボル・テーブルを作成します．PASS－1はモード指定（5.6参照）にかかわらず実行されます．モードSを指定するとシンボル・リスト・イメージをリスト・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（5.8参照）が行われます.

(2)　PASS－2

ソース・プログラムを読みモード指定に対応してアセンブル・リスト・イメージ，エラー・リスト・イメージ，クロス・レファレンス・リスト・イメージをリスト・ファイルに出力します.

(3)　PASS－3

モードOの指定によりソースプログラムを読みHEXオブジェクト・イメージをオブジェクト・ファイルに出力します.

1.3　ワーク・ファイル

本アセンブラはクロス・レファレンス・リストを作成する場合と，GJMP疑似命令の最適化を行う場合，ワーク・ファイルを使用して処理を行います．このため次の名称を持つ4個のファイルが作業時に指定されたユニット（5.9参照）上に作られ，処理が終了すると消去されます．したがってワーク・ファイルが作られるユニット上に同一名のファイルがないことを前もって確認してください．同一名のファイルがあると，“FILE ERROR 13”となります.

(1)　XREF75.$$$　（クロス・レファレンス・リスト作成用）
(2)　SYM75.$$$　（クロス・レファレンス・リスト作成用）
(3)　TABL75.$$$　（クロス・レファレンス・リスト作成用）
(4)　OPTI75.$$$　（GJMP疑似命令最適化用）

第2章　ソース・プログラム様式

2.1　文の構成

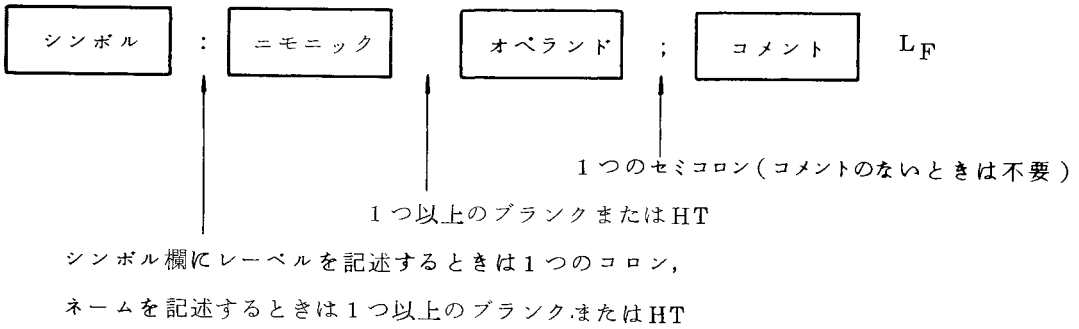
アセンブラ言語のソース・プログラムは，文（Statement）により構成されます．

1つの文は下図のように，シンボル欄，ニモニック欄，オペランド欄，コメント欄の4つのフィールドより構成され，L_F（ライン・フィード）でターミネートされます．

各欄は空白，コロン(:)またはセミコロン(;)で区切り，1行には最大80文字まで記述できます．

文の記述は自由形式で，シンボル，ニモニック，オペランド，コメントの順に記述されていれば1行の範囲で自由に書けます（各文は1行以内で記述し，次の行へ継続することはできません）．

また，コメントのみ，および空文（1行がすべてブランク）の記述も可能です．



2.2　文　　字

ソース・プログラムは次の文字を使用して記述してください．

(1)　英文字

A　B　C　D　E　F　G　H　I　J　K　L　M

N　O　P　Q　R　S　T　U　V　W　X　Y　Z

(2)　数　　字

0　1　2　3　4　5　6　7　8　9

(3)　演算子

＋　－　＊　／　（　）

(4)　特殊文字

$　'　：　；　△（ブランク）　#，@　？　／　"　%　&　．　＜　＝　＞HT

$→現在のロケーション・カウンタのアドレスを示します．

'→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.

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，それ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，MACRO命令のオペランドのデータ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，英文字で始まります.

(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，ネームはブランク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：……レーベル中にブランクがあります.
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.

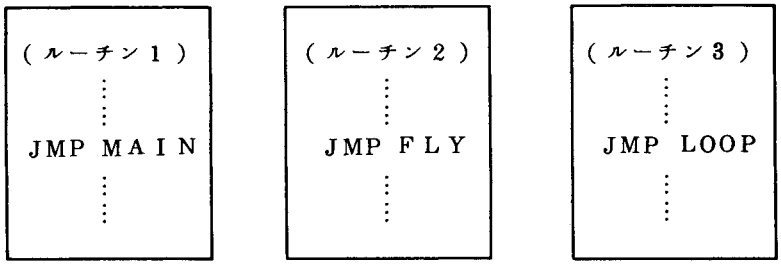
例３

```
ＡＢＣ　ＥＱＵ　３
ＸＹＺ　ＥＱＵ　ＡＢＣ
```

ＡＢＣとＸＹＺには同じデータ“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: ILS   ;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　ニモニック欄

ニモニック欄には，命令，疑似命令，マクロ名を記述します．

オペランドの必要な命令の場合，ニモニッ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 ……ニモニックとオペランドの間にブランクがありません．
	DLS	D　LS……ニモニック中にブランクが挿入されています．
	RTS	RETS……μPD７５００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もつもので，数字や文字の列で構成されます．

数字だけで構成されるものを数値定数，文字で構成され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

データの最後に文字“ B ”を付加して表現します．

例　１０１１B

(b)　８進定数

データの最後に文字“ 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

データの最後に文字“ 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

データの最後に文字“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．全てのＡＳＣＩＩ文字を使用できますので，ブランクも文字定数としてそれぞれのＡＳＣＩＩコードに変換されます．

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	レ　ジ　ス　タ　名
７５００セットＡ	ＨＬ
７５０２，７５０２Ａ	ＨＬ＋（オート・インクリメント）
７５０３，７５０３Ａ	ＨＬ－（オート・デクリメント）
７５０７，７５０７Ｂ，７５０７Ｈ，７５０７Ｓ	ＤＥ
７５０８，７５０８Ｂ，７５０８Ｈ	ＤＬ
７５１４	
７５１６，７５１６Ｈ	
７５１９，７５１９Ｈ	

製　品　群	レ　ジ　ス　タ　名
7500.セットB 7501 7506 7520	HL HL+（オート・インクリメント） HL−（オート・デクリメント）
7527，7527A 7528，7528A 7533 7537，7537A 7538，7538A	HL HL+（オート・インクリメント） HL−（オート・デクリメント） DL

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+の間にブランク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）

“$”はロケーション・カウンタの値を示します. すなわち，$をオペランドに記述した命令に割付けられたアドレス（2または3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

アドレス	シンボル	ニモニック	オペランド
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

アドレス	シンボル	ニモニック	オペランド
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

シンボル	ニモニック	オペランド
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.

例

シンボル	ニモニック	オペランド
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.
除算結果の小数点以下は切り捨てます.

(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

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

演算は次のように行なわれます.

```
      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

シンボル	ニモニック	オペランド
MASK：	LAI	6FAH AND 0FH

演算は次のように行なわれます.

```
      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

シンボル	ニモニック	オペランド
MDFY1：	LAI	0AH OR 1101B

演算は次のように行なわれます．

```
      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
MDFY2：	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ブランクが必要です．

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ブランクが必要です．

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.

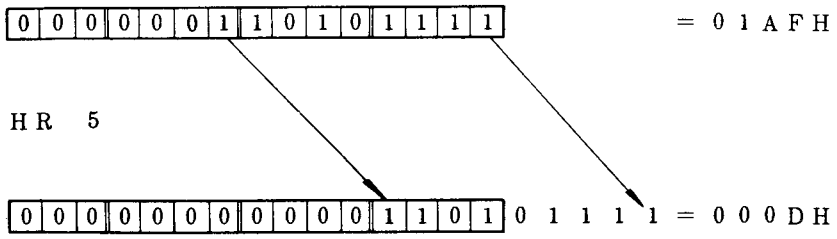
(k)　SHR　（Shift　Right）

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ブランクが必要です.

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.

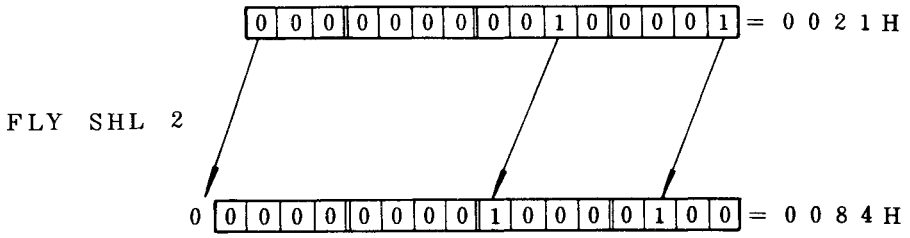
(l)　SHL　（Shift　Left）

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ブランクが必要です.

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.

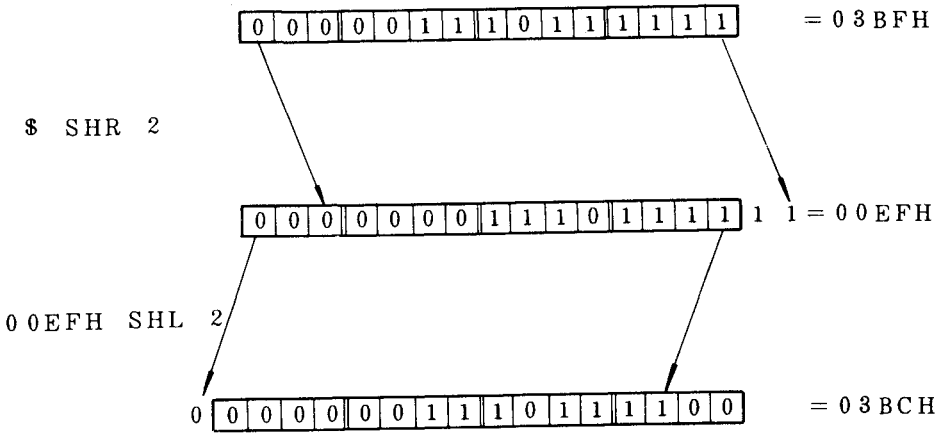


この結果，００８４Ｈ番地にジャンプします．

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の前後にはブランクが必要です．

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)　－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)　－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，その後にコメント（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すがアセンブラの処理対象にはなりません．

コメントにはすべてのASCII文字を使用できます．

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

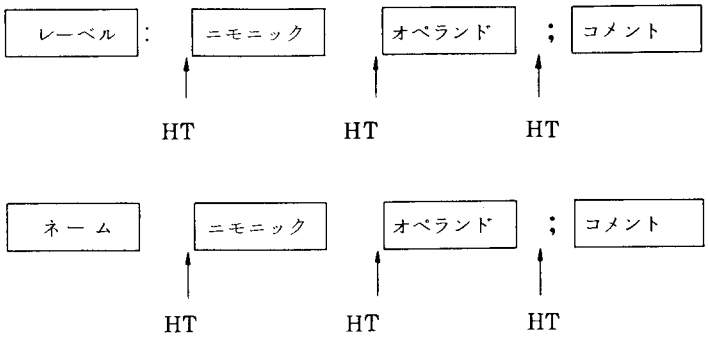
2.7　タビュレーション機能

μPD7500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，オペランド欄，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
```

←8カラム→←8カラム→←8カラム→

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，オペランド欄の先頭，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カラムから８カラムまでのブランクを発生します．

（（ＨＴのカラム位置－１）∧０ＦＦＦ８Ｈ）＋９＝次欄スタート・カラム位置

第３章　疑似命令

μPD7500 シリーズ・アセンブラのニモニック欄には，命令，疑似命令，およびマクロ名のみが記述できます．

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	ニモニック	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	ニモニック	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にも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０００００番地を割り付けます．

(ニ) ＯＲＧ直前の割り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	ニモニック	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データまたはアドレスをシンボル欄に記述したネームに割付けます．

(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	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が記述されたときは，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)　ＥＱＵで定義され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されたデータ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が記述されたときは，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されますが，二度目以降はそのネームがオペランドで指定され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
〔レーベル：〕	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)

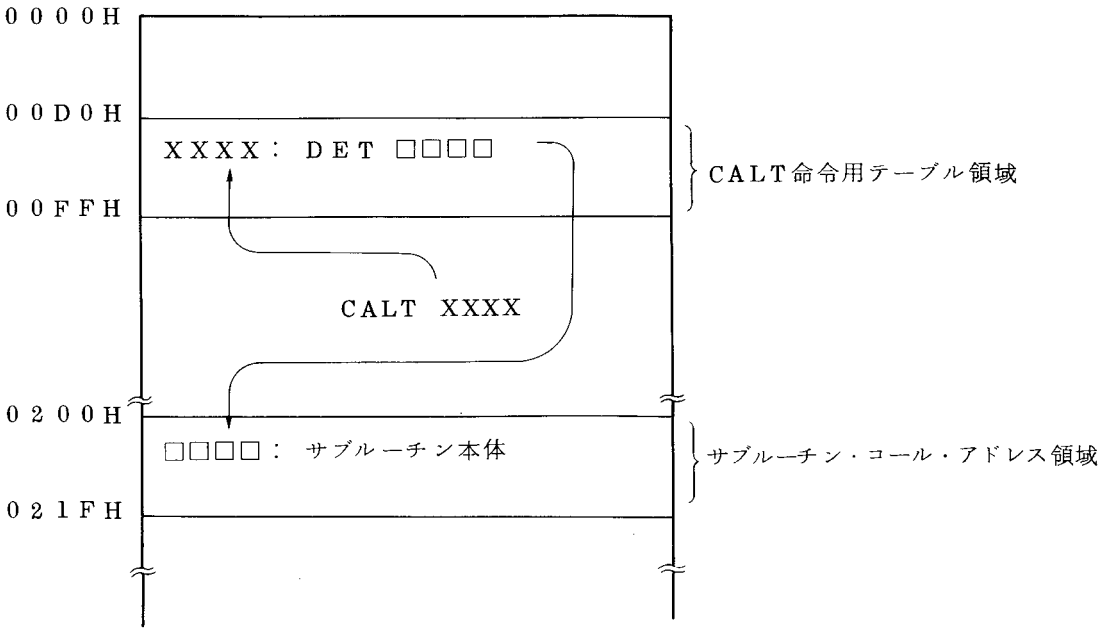
(1)　機　　能

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

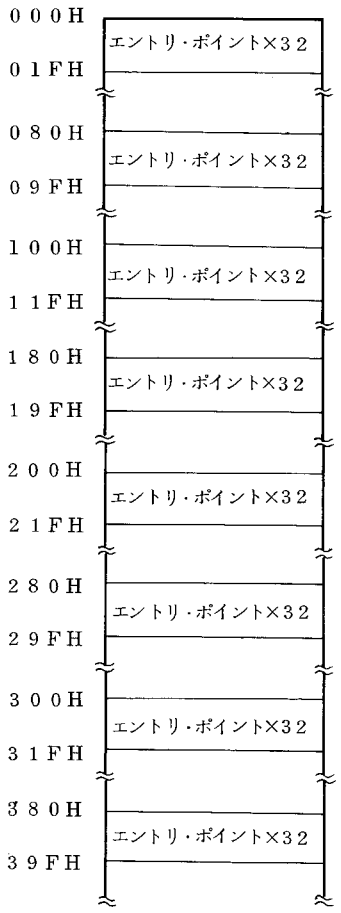
シンボル	ニモニック	オペランド	コメント
〔レーベル：〕	DET	式	〔；コメント〕

(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×ｎから１２８×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．

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：〕	IF	式	〔；コメント〕	
	⋮			
	ステートメント			
	⋮			
〔レーベル：〕	ELSE		〔；コメント〕	←ELSEを使用した場合
	⋮			
	ステートメント			
	⋮			
〔レーベル：〕	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

```
シンボル    ニモニック    オペランド
COND        SET           0FFH
            IF            COND        ┐
            LAI           5           ┘ アセンブルされる.
            ELSE                      ┐
            LAI           6           ┘ アセンブルされない.
            ENDIF
COND        SET           0
            IF            COND        ┐
            AISC          7           ┘ アセンブルされない.
            ENDIF
```

(ロ) ネスティングの例

```
                IF          TYPE        ┐ TYPE≠0 ならば
                ⋮                       ┘ アセンブルされる.
レベル2 ┌       IF          COND1       ┐ TYPE≠0 , COND1≠0
        │       ⋮                       ┘ ならばアセンブルされる.
        └       ENDIF                   ┐ TYPE≠0 ならば
                ⋮                       ┘ アセンブルされる.
レベル2 ┌       IF          COND2       ┐ TYPE≠0 , COND2≠0
        │       ⋮                       ┘ ならばアセンブルされる.
        └       ENDIF
                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

シンボル	ニモニック	オペランド	コメント
〔レーベル:〕	END		〔;コメント〕

(3) 注意事項

(イ) 1つのアセンブル単位でEND文は必ず1つだけで，かつプログラムの最後になければいけません.

3.7　マクロ定義疑似命令

3.7.1　MACROとENDM（MACRO Definition and END of Macro）

(1) 機　　能

MACRO文とENDM文の間にある一連のステートメント（マクロ・ボディ）に対しネーム（マクロ名）を割り付けます.

(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	MACRO	〔仮パラメータ・リスト〕	〔；コメント〕
	⋮		
	ENDM		

詳しくは第4章を参照してください.

3.7.2　EXITM

(1) 機　　能

マクロ展開時にEXITM疑似命令が現われると，現在のマクロ展開を終了し，ENDM文の次のステートメントからアセンブルを展開します.

(2) 記述形式

シンボル	ニモニック	オペランド	コメント
	EXITM		〔；コメント〕

詳しくは第4章（4.5）を参照してください.

3.7.3　GLOBAL

(1) 機　　能

オペランド欄で指定されたシンボルをグローバル・シンボルと宣言します.

(2) 記述形式

シンボル	ニモニック	オペランド	コメント
	GLOBAL	グローバル・シンボル・リスト	〔；コメント〕

詳しくは第4章（4.3）を参照してください.

3.8　アセンブル制御疑似命令

3.8.1　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用スペースに，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．それ以上記述され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．

3．8．2 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ないます．

(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〕

3．8．3 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されます．SPACE文自身は改行する直前に印字されます．

(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

この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，式の値は無視され改頁されます．

3．8．4 PRINT

(1) 機　能

アセンブル・リスト出力をオペランド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	{ON / OFF / SEL / NOSEL}	〔；コメント〕

(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ます．

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．

3.9　JUMP系命令の自動選択疑似命令

3.9.1　GJMP

(1) 機　能

ユーザがJUMP系命令のJCP，JMPと使い分けなくても，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（5.8参照）．

(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) 最適化オプションが指定されなかった場合

(イ) ソース・プログラムのオペランドの式にシンボルが1つも書かれていない場合

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）

(ロ) ソース・プログラムのオペランドの式のシンボルが書かれている行がすべてGJMPと書かれた行と同じか，またはそれ以前にある場合

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①，⑤参照）．

(ハ) ソース・プログラムのオペランドの式のシンボルが書かれている行が1つでもGJMPと書かれた行以降にある場合

アセンブラは，すべてJMP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JCP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)　最適化オプションが指定された場合

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

```
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
       ⋮                      ⋮                  及び '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します．

第4章　マクロ

同一プログラム中で同じ命令群を何回も使用する場合は，一般にサブルーチンにしてメモリ・サイズを節減しています．この命令群の占めるメモリ・サイズが比較的小さく，かつパラメータの使用が複雑な場合等には“マクロ機能”を使用すると便利で，プログラム作成の手間が軽減されます．

4.1　マクロの利用

μPD7500シリーズ・アセンブラにはマクロ命令として疑似命令“MACRO”と“ENDM”とがあり，2つの命令は常に“対”で使用します．

マクロは次のような手順で使用します．

1. **マクロの定義**

   まず，使用頻度の高い一連のステートメントをマクロ定義します．

   ソース・プログラムの作成段階で，ある一連のステートメントの前後に“MACRO”命令と“ENDM”命令を付加し，さらにMACRO命令にネームを付けることによりマクロが定義されます．

   マクロ定義によりMACRO命令とENDM 命令間の一連のステートメント（マクロ・ボディ）は，MACRO命令に付けられたネーム（マクロ名）と等価になります．

2. **マクロの参照**

   ソース・プログラム中で，以前に定義されたマクロを参照したい場合はそのマクロ名を通常の命令と同じように“ニモニック欄”に記述します．

   このマクロ名の記述（参照）は，マクロ・ボディ全体の参照と同じ意味になります．

3. **マクロの展開**

   マクロの定義，参照が行なわれているソース・プログラムをアセンブルすると，参照しているところにそのマクロが展開されます．

   アセンブラは，ソース・プログラム中でマクロ名の参照を見付けるとそれを対応するマクロ・ボディに展開してマクロ名の位置に挿入し，オブジェクト・プログラムに変換します．

4.1.1　マクロとサブルーチン

マクロとサブルーチンは，それぞれに特徴がありますので目的に応じて使い分けてください．

(1)　サブルーチン

サブルーチン本体は1度だけ機械語に変換され，その参照に対しては参照回数だけ1～2バイトのサブルーチン・コール命令が現われるだけです（一般には参照前後に引数設定のため命令が必要です）．したがってサブルーチンの活用によりメモリを効率良く利用できます．

また，サブルーチン化する事によりプログラムの設計が容易になり，プログラム全体の構造が理解し易くなります（プログラムの構造化）．

(2) マクロ

マクロの基本的な機能は命令群や文字列の置き換えにあります．アセンブラはマクロ参照を検出するとマクロ・ボディの仮パラメータを参照時の実パラメータに置換した命令群を展開し，それを機械語に変換します．

サブルーチン化の手法がメモリ・サイズの削減やプログラムの構造化を図るために用いられるのに対し，マクロはコーディングの能率を向上させるため，あるいはプログラムの読み易さを向上させるために用いられます．

たとえば処理手順は同じであるが使用するレジスタ名とオペランドに現われるレーベルが異なる命令群を使う場合，レジスタ名とレーベルに対して仮パラメータを割当ててマクロを定義しておけば，コーディング時にはマクロ名と実パラメータ（該当するレジスタ名とレーベル）を記述するだけで種々の命令群を記述できます．

4.2 マクロの機能

4.2.1 マクロの定義

(1) 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	MACRO	〔仮パラメータ・リスト〕	〔；コメント〕
	⋮		
	マクロ・ボディ（Macro Body）		
	⋮		
〔レーベル：〕	ENDM		〔；コメント〕

(2) 機　能

MACRO文とENDM文との間にある一連のステートメント（マクロ・ボディ）に対してネームで示されるマクロ名を割り付けます．

このネームは後でマクロ・ボディを参照する場合の代表名となります．

(3) 注意事項

(イ) マクロ・ボディ

マクロ・ボディは，“レーベルまたはネーム”，“命令”，“疑似命令”（MACROとENDMを除く），“コメント”，“マクロ名”（他のマクロの参照）で構成されます．

マクロ・ボディ内でダブル・セミコロン（；；）が記述された場合，“；；”の後は，その行の終了までマクロ定義内コメントと見なし，マクロ参照時には展開されません．

(ロ) 仮パラメータ・リスト

(i) 仮パラメータは6文字以内の英数字（シンボル記述上の規則と同じ）で構成され，一行以内であればコンマ（，）で区切っていくつでも記述できます．

(ii) 仮パラメータはマクロ・ボディ内でのみ有効です．

(iii) マクロ・ボディ内に記述された仮パラメータは，そのマクロが参照されたときに実パラメータに置き換わります．

(4) 使 用 例

AUGENとADDENの２つの数を加算して，結果をSTADR で示されるメモリにストアする簡単なマクロを考えます．

次のようにして“マクロ定義”を行います．

シンボル	ニモニック	オペランド
ADMAC	MACRO	AUGEN，ADDEN，STADR
	LAI	AUGEN
	AISC	ADDEN
	LHLI	STADR
	ST	
	ENDM	

4．2．2　マクロの参照

(1) 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：〕	ネーム	〔実パラメータ・リスト〕	〔；コメント〕

(2) 機　　能

MACRO文とENDM文で定義されたマクロ・ボディを参照します．

(3) 注意事項

(イ) ネームはMACRO 文のシンボル欄に記述された“マクロ名”で，参照する以前に定義されているものでなければいけません．

(ロ) 実パラメータとして記述できる形式は次の５種類で，マクロ展開時にマクロ内の仮パラメータに順に対応させて，そのまま置換わります．

1. 定数
2. 文字定数（引用符で囲んだASCII文字列）
3. レーベルまたはネーム
4. 式
5. スペースまたは空き（記述なし）……００が割付けられます．

(ハ) 仮パラメータから実パラメータへの置換えは，それぞれの記述順に対応させて左から順に行なわれます．

ただし，

実パラメータの数＞仮パラメータの数　の場合

余分な実パラメータは無視されます．

実パラメータの数＜仮パラメータの数　の場合

対応しない残りの仮パラメータには“NULL　STRING”（００コード）が割り付けられます．

(4) 使 用 例

以前に定義したマクロ（ADMAC）を参照します．

```
シンボル        ニモニック      オペランド
                  ⋮
                ADMAC         10H,20H,100H
                  ⋮
```

4．2．3　マクロの展開

マクロを使用したソース・プログラムをアセンブルすると，アセンブラは，まず

1. マクロ定義があると，シンボル・テーブル・エリアにマクロ・ボディをそのままのイメージを格納します．
2. 次に，マクロの参照を見付けるとそれに対応するマクロ・ボディをシンボル・テーブル・エリアより捜し出してマクロ名の位置に挿入します．（展開）．
3. 展開したプログラムを他のプログラムと同様にアセンブルします．

(1) 使 用 例

```
シンボル        ニモニック      オペランド
                  ⋮
                ADMAC         5,6,30H          ←参照
                LAI           5                ←展開
                AISC          6
                LHLI          30H
                ST
                ENDM
                  ⋮
```

4．3　マクロ内のシンボル

4．3．1　マクロ内シンボルの有効範囲

マクロ内で使用するシンボルにはグローバル・シンボル（Global Symbol）とローカル・シンボル（Local Symbol）の２種類があります．

(1) グローバル・シンボル

ソース・プログラム内のすべてのステートメントから参照できます．また，プログラム内ではＳＥＴネーム以外には，２回以上定義することはできません．

(2) ローカル・シンボル

そのマクロ内，もしくは，そのマクロを含む外側のマクロ内で定義されたシンボルは参照することができます．シンボルのサーチは現在処理中のマクロからそのマクロを含む外側のマクロに向かって行なわれ，最初に見つかったシンボルの値が有効となります．

シンボルの定義は現在処理中のマクロよりも，内側のブロック内であれば２度以上定義することができます．

また，同じシンボルでも異なったマクロであれば2度以上定義することができます．

4．3．2　グローバル宣言疑似命令

マクロ内で使用するシンボルは，特に宣言しなければローカル・シンボルとみなされます．グローバル・シンボルとするためには，そのシンボルを定義する以前にグローバル宣言疑似命令で宣言しなければなりません．

(1)　記述形式

シンボル	ニモニック	オペランド	コメント
	GLOBAL	グローバル・シンボル・リスト	〔；コメント〕

(2)　機　　能

オペランド欄で指定されたシンボルをグローバル・シンボルと宣言します．

(3)　注意事項

(イ)　グローバル・シンボルは，コンマ(,)で区切って，複数個指定することが可能です．

(ロ)　グローバル・シンボルの個数が多い場合には，“S”エラーが印字され，オーバーフロー直前までのシンボルが有効になります．

(4)　使 用 例

例1　グローバル・シンボル

```
シンボル    ニモニック    オペランド
            GLOBAL        SYM
STMAC       MACRO
SYM         SET           5          ┐
            DB            SYM        ├ ①
            ENDM                     ┘
SYM         SET           0
            DB            SYM ──→② SYM=0
              ⋮
            STMAC             ──→マクロ参照
SYM         SET           5          ┐
            DB            SYM ──→③ SYM=5 ├ マクロ展開
            ENDM                     ┘
              ⋮
            DB            SYM ──→④ SYM=5
```

①　“STMAC”というネームでマクロが定義されています．

②　SYM”には，直前のSET文で定義された値“0”が割り付けられます．

③　SYM”には，直前のSET文で定義された値“5”が割り付けられます．

④　SYM”はマクロ外でグローバル・シンボルと定義されたので，マクロ内で変更された

“SYM”の値はマクロ展開を終了してもそのまま有効であり“5”が割り付けられます.

例2　ローカル・シンボル

シンボル“SYM”は最初にグローバル宣言していません.

例1と同じマクロ“STMAC”を使用しています.

```
SYM     SET     0
        DB      SYM     ―――→ ①′SYM=0

        STMAC           ―――→ マクロ参照
SYM     SET     5
        DB      SYM     ―――→ ②′SYM=5   } マクロ展開
        ENDM
         ⋮
        DB      SYM     ―――→ ③′SYM=0
```

①′ “SYM”には，直前のSET文で定義された値“0”が割り付けられます.

②′ “SYM”はグローバル・シンボル宣言されていませんのでローカル・シンボルです.

ここでは直前のSET文で定義された値“5”が割り付けられ，このマクロ内で有効です.

③′ “SYM”はローカル・シンボルなので，マクロ内で定義された値は参照されず，再び“0”が割り付けられます.

4.4　マクロ・パラメータ

マクロ定義でオペランド欄に記述された仮パラメータは，マクロが展開されるときに実パラメータに置き換わります.

例1　パラメータを使用したマクロの定義および参照を示します.

```
シンボル        ニモニック      オペランド
PMAC            MACRO           P1,P2          ……(a)
L1              EQU             10H
                DB              P1
                DB              P2
                ENDM
                 ⋮
L1              EQU             20H
                 ⋮
```

```
                PMAC        30H,L1      ……(b)
L1              EQU         10H
                DB          30H         ……(c)
                DB          L1
                ENDM
```

(a)　マクロ定義文のＰ１，Ｐ２は仮パラメータでありマクロ内のＤＢ命令で参照しています.

(b)　マクロ参照時に，実パラメータ３０Ｈ，Ｌ１はマクロ定義(a)の仮パラメータＰ１，Ｐ２に対応しています.

(c)　マクロ展開時にＰ１には実パラメータ３０Ｈが，そのまま置換えられます．またＰ２には実パラメータＬ１が，そのまま置換えられます．Ｌ１はローカル・シンボルなのでＬ１の値は，マクロ内のＥＱＵ命令で定義された１０Ｈが与えられます.

４.５　EXITM

(1)　記述形式

シンボル	ニモニック	オペランド	コメント
	EXITM		〔；コメント〕

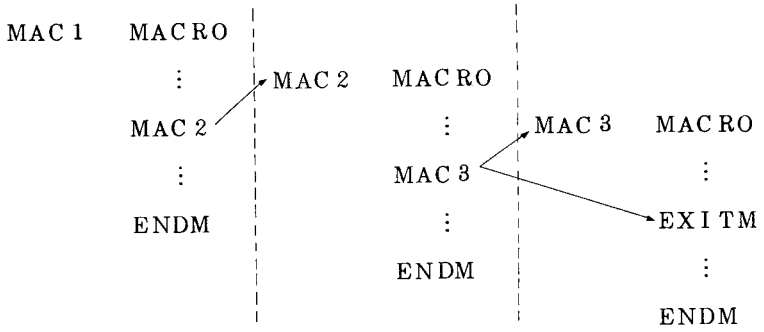
(2)　機　　能

マクロ展開時にEXITM疑似命令が現われると，現在のマクロを終了し，ENDM文の次のステートメントからアセンブルを展開します.

(3)　注意事項

(イ)　マクロ定義中では，意味をもちません.

(4)　使 用 例



上記の場合，MAC３のEXITM～ENDMはアセンブルされません.

4.6　仮パラメータと他のシンボルとの結合

仮パラメータと他のシンボルとを結合させる場合には“ & ”を使用します.

（使用例１）

```
シンボル      ニモニック      オペランド
MAC1          MACRO           P1,P2          ┐
              LDA&P1          P2             ├ マクロ定義
              ENDM                           ┘
               ⋮
              MAC1            ,XYZ           ……マクロ参照
              LDA             XYZ            ┐
              ENDM                           ├ マクロ展開
               ⋮                             ┘
              MAC1            X,B            ……マクロ参照
              LDAX            B              ┐
              ENDM                           ├ マクロ展開
                                             ┘
```

（使用例２）

```
MAC2          MACRO           P1,P2,P3       ┐
              P1&P2           P3             ├ マクロ定義
              ENDM                           ┘
               ⋮
              MAC2            LDA,,XYZ       ……マクロ参照
              LDA             XYZ            ┐
              ENDM                           ├ マクロ展開
               ⋮                             ┘
              MAC2            LDA,X,B        ……マクロ参照
              LDAX            B              ┐
              ENDM                           ├ マクロ展開
                                             ┘
```

4.7　マクロの使用例

1

```
シンボル        ニモニック      オペランド
                ORG             0H
MAC1            MACRO           P1,P2
                LLI             P1
                LAM             P2
                ENDM
                ⋮
MAC2            MACRO           P1,P2,P3
                AISC            P1
                JMP             P2
                JMP             P3
                ENDM
                ⋮
                MAC1            7,HL+      }
                MAC2            3,L1,L2    }①
                ⋮
L1:             CALL            SUB1
                JMP             L3
L2:             CALL            SUB2
L3:             NOP
                ⋮
```

上記のソース・プログラムをアセンブルにすると①は，次のように展開されます．

```
シンボル        ニモニック      オペランド
                MAC1            7,HL+
                LLI             7
                LAM             HL+
                ENDM

                MAC2            3,L1,L2
                AISC            3          …(a)
                JMP             L1
                JMP             L2
                ENDM
                ⋮
```

```
                ⋮
L1：        CALL        SUB1
            JMP         L3
L2：        CALL        SUB2
L3：        NOP
                ⋮
```

(a)　演算の結果，キャリー・フラグがリセットの場合，Ｌ１にジャンプし，SUB1 をコールします．

キャリー・フラグがセットされた場合は，スキップ条件が成立します．そして，Ｌ２にジャンプし，SUB2をコールします．この条件は，MAC１，MAC２のパラメータの値により，変更できます．

2

```
シンボル        ニモニック      オペランド
                ⋮
                GLOBAL      SUB2
MAC3            MACRO
SUB2：          LAI         1
                LHI         4
                LLI         6
                RET
                ENDM
                ⋮
                CALL        SUB2
                ⋮
                MAC3
```

上記のソース・プログラムをアセンブルすると，次のように展開されます．

```
シンボル        ニモニック      オペランド
                ⋮
                CALL        SUB2        …(a)
                ⋮
                MAC3                    …(b)
SUB2：          LAI         1
                LHI         4
                LLI         6
                RET
                ENDM
                ⋮
```

(a)　SUB2はグローバル宣言されているので，MAC3のSUB2(b)をコールします．

SUB2がGLOBAL宣言されていない場合，MAC3以外でSUB2が定義されていなければ，エラーとなります．

保守／廃止

第５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＿＿（アンダー・ライン）は，ＯＳまたは本アセンブラが出力する文字（文字列）であることを示します．また，入力文字は英小文字は使用できません．

５．１　アセンブラの起動方法

コンソールから次のように入力してください．

X＞〔d：〕ASM75 ↵　　（↵　キャリッジ・リターン）

または

X＞〔d：〕ASM75F ↵

ASM75　＝μPD７５００シリーズ　アセンブラーⅠのファイル名
ASM75F＝μPD７５００シリーズ　アセンブラーⅡのファイル名
X　　＝ログイン・ディスク番号
d　　＝ASM75・COMまたはASM75F・COMを格納しているフロッピィ・ディスクのドライブ名

また，コンソール入力は“＜”のリダイレクト機能を使用してファイルから入力することも可能です．

例　ドライブAにASM75がある場合

A＞ASM75 ↵

５．２　アセンブラ・タイトルの出力

5.1によりアセンブラが起動されると，アセンブラは次のメッセージをコンソールに出力します．

```
UPD7500 SERIES ASSEMBLER-I V×.× 〔×× ××× ××〕
  Copyright (C) 1983 NEC Corporation
```

注　μPD７５００シリーズ・アセンブラーⅡの場合

UPD7500 SERIES ASSEMBLER-Ⅱ と出力されます．

保守／廃止

5.3　入力ソース・ファイル名の指定

ソース・ファイル名（およびそれが存在するドライブ名）を指定します．

SOURCE FILE=I ）

〔I=〔d：〕プライマリ・ネーム〔．ファイル・タイプ〕〕

d：ソース・ファイルを格納しているドライブ名

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

注１　プライマリ・ネーム，ファイル・タイプは次の文字列を使用してください．
英大文字（英小文字は使用できません），数字，！＃＄％＆＂（）＋－＼＾＿＠

2　ドライブ名d：を省略した場合，ドライブAを指定したものとみなされます．

3　ファイル・タイプを省略した場合，ファイル・タイプを指定しないとみなされます．

4　ファイル名を誤って入力したり，指定なしに）のみを入力した場合，再び同じメッセージが出力されます．

5.4　日付指定

5.3に続いて日付指定のための問がコンソールに出力されます．日付以外のコメントを入力することも可能です．

DATE = d_1 ～ dn
〔d_1 ～ dn＝２０文字以内のASCII文字列．〕

入力された文字列は各種リストの日付欄に出力されます．文字列の入力を省略した場合には２０個の空白が，また文字列が２０を越えた場合には最初の２０文字が有効となります．

5.5　アセンブル対象品種の指定

アセンブル対象品種を指定します．

(1)　μPD7500シリーズーIの場合
ASSEMBLE（A／B／20／01／02／03／06／07／08）＝name ）

name＝A……７５００セットＡのソース・プログラムをアセンブルします.
name＝B……７５００セットＢの　〃
name＝20 …７５２０の　〃
name＝01 …７５０１の　〃
name＝02 …７５０２，７５０２Ａの　〃
name＝03 …７５０３，７５０３Ａの　〃
name＝06 …７５０６の　〃
name＝07 …７５０７，７５０７Ｂ，
７５０７Ｈ，７５０７Ｓの　〃
name＝08 …７５０８，７５０８Ｂ，
７５０８Ｈの　〃

注意　以上の９種類以外の指定を行なったり，指定なしに↵のみを入力した場合，再び同じメッセージが出力されます.

(2)　μＰＤ７５００シリーズ－Ⅱの場合

ＡＳＳＥＭＢＬＥ　（Ａ／Ｂ／１４／１９／２７／２８／３７／３８／１６／３３）＝ｎａｍｅ ↵

name＝A……７５００セットＡのソース・プログラムをアセンブルします.
name＝B……７５００セットＢの　〃
name＝14 …７５１４の　〃
name＝19 …７５１９，７５１９Ｈの　〃
name＝27 …７５２７，７５２７Ａの　〃
name＝28 …７５２８，７５２８Ａの　〃
name＝37 …７５３７，７５３７Ａの　〃
name＝38 …７５３８，７５３８Ａの　〃
name＝16 …７５１６，７５１６Ｈの　〃
name＝33 …７５３３の　〃

注意　以上の８種以外の指定を行なったり，指定なしに↵のみを入力した場合，再び同じメッセージが出力されます.

5.6　モード指定

5.5の対象品種の指定が終了すると次の問がコンソールに出力されますのでアセンブラの実行モードを指定します.

ＭＯＤＥ　（Ｓ／Ｌ／Ｅ／Ｘ／Ｏ）＝m_1 m_2 …mn ↵

mn ＝S…シンボル・リスト・イメージをリスト・ファイルに出力する.
　　L…アセンブル・リスト・イメージを　　〃
　　E…エラー・リストを　　〃
　　X…クロスレファレンス・リストを　　〃
　　O…ＨＥＸオブジェクト・イメージをオブジェクト・ファイルに出力する.

モードＬとＥを同時指定するとＥは無視されます.

５.７　最適化オプションの指定

ＪＵＭＰ系命令自動選択機能の最適化指定の有無を指定します.

OPTIMIZE （Y/N） ＝ r ↵

r＝Y…… 最適化パスを行ないます.
r＝N……　〃　行ないません.

注意　Ｙ，Ｎ以外の指定を行った場合は，エラーとして再び同じメッセージを出力してきます.

５.８　最適化パスの回数指定

5.7でＹを指定した場合には，次のようにコンソールに印字されますので，最適化パスの繰り返し回数を指定します.

TIMES ＝ n ↵

〔n＝1～9999までの１０進数〕

注意1　0↵または↵のみを入力した場合，最大数（9999）を入力したものとみなされます.
それ以外の場合にはエラーとして，再び同じメッセージを出力してきます.

2 最適化パスの方法

アセンブラは，ＧＪＭＰの最適化を行うために，ソース・プログラムの中間ファイルをフロッピィ・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回の読み込みで行う処理を最適化パスと呼びます.
5.7の最適化オプションが指定された場合，次の手順で処理が行われます.

(1)　PASS１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，そのとき最適化パスの状況がわかります.

TIMES = tttt, CODE = nnnn

〔 tttt＝現在の繰返し回数が１０進右づめで入る.
nnnn＝生成されたオブジェクト・コード数が１６進右づめで入る. 〕

(3) 現在のパスで，最適化パスが完了した（シンボル・テーブルの更新が行われずそのパスで生成されたコード数とその直前のパスで生成されたコード数が一致している）とき，次のメッセージがリスト上のエラー数を表示するメッセージの次に印字され，PASS－２の処理が開始されます.

OPTIMIZATION COMPLETED, TIMES = tttt

〔 tttt＝ 現在の繰返し回数が１０進右づめで入る. 〕

(4) 指定された回数内で，最適化が完了されなかった場合は，次のメッセージがリスト上のエラー数を表示するメッセージの次に印字され，PASS－２の処理が開始されます.

OPTIMIZATION TERMINATED, TIMES = tttt

〔 tttt= 最終的な繰返し回数が１０進右づめで入る. 〕

５.９　出力ファイルの装置指定

モード指定が終了するとモード指定に対応して出力ファイルの装置指定の問がコンソール上に出力されます.

```
OUTPUT FILE (CON/LST/PUN/A-P)
  LIST ..............=注1 f1 )
  OBJECT ............=注2 f2 )
  SYM-TBL(Y/N).......=注2 f3 )
TEMPORARY FILE (A-P)
  UNIT ..............=注3 f4 )
```

〔 f_1 はCON（コンソール），LST（ライン・プリンタ），およびAからPの指定が可能です.
f_2 はCON（コンソール），LST（ライン・プリンタ），PUN（紙テープ・パンチ）およびAからPの指定が可能です.
f_3 はY，Nのみ指定可能です.
f_4 はAからPのみ指定可能です. 〕

注１　モードS，L，XまたはEのうち１つ以上を指定すると出力されます.

注２　モードOを指定すると出力されます.

注３　モードXまたは最適化オプションを指定すると出力されます.

モードＬまたはＯでCONを指定した場合にCRTに表示されるリスト表示を一時停止される際に CTRL －Ｓ以外の文字を入力すると，以後一時停止ができなくなります．

５．１０　アセンブル実行開始メッセージの出力

5.9 迄の操作が終了すると，アセンブラは次のメッセージをコンソールへ出力し，アセンブラを開始します．

```
** ASSEMBLE START **
```

５．１１　アセンブル終了メッセージの出力

アセンブル実行時エラー（6.5 参照）のなかった場合，アセンブラは次の終了メッセージをコンソールへ出力します．

```
** ASSEMBLE COMPLETE,  nnnn  ERRORS **
```

〔nnnn＝１０進４桁のアセンブル・エラー数〕

上記メッセージはモードＳ，Ｌ，ＸまたはＥのうち１つ以上を指定すると出力します．

５．１２　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・プログラム・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．

```
** FILE LIST **
  SOURCE･･･= ソース・ファイル名 注1
  LIST･････= リスト・ファイル名 注2
  OBJECT･･･= オブジェクト・ファイル名 注3
  SYM-TBL  = シンボル・テーブル・ファイル名 注4
```

注１　5.3で指定したソース・ファイル名を出力します．

注２　プライマリ・ネームはソースと同一で，ファイル・タイプはＰＲＮです．

注３　モードＯを指定すると出力します．プライマリ・ネームはソースと同一で，ファイル・タイプはＨＥＸです．

注４　プライマリ・ネームはソースと同一で，ファイル・タイプはＳＹＭです．

5.13　アセンブラの操作例

(1)　ソース・プログラム例

```
        TITLE           'EXAMPLE'
;
;=========================================
;=                                       =
;=              UPD 7500 SERIES          =
;=                                       =
;=========================================
;
        ORG     10H
LABEL1  EQU     10H
LABEL2  EQU     20H
;
        LAI     5
        LHLI    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>ASM75                                                    ……アセンブラ・プログラムの起動（5.1）

UPD7500 SERIES ASSEMBLER-I VX.X  (XX XXX XX)  }
   Copyright (C) 1983 NEC Corporation         }           ……アセンブラ・タイトルの出力　（5.2）

SOURCE FILE = 7500MN.ASM                                   ……入力ソース・ファイル名の指定（5.3）

DATE = XX XXX 83                                           ……日付指定　　（5.4）

ASSEMBLE TYPE (A/B/20/01/02/03/06/07/08) = A               ……アセンブル対象品種の指定　（5.5）

MODE (S/L/E/X/O) = SLOX                                    ……モード指定　　（5.6）

OPTIMIZE(Y/N) = Y                                          ……最適化オプションの指定　（5.7）

TIMES = 2                                                  ……最適化パスの回数指定　（5.8）

OUTPUT FILE  (CON/PUN/LST/A-P)                             ……出力ファイルの装置指定　（5.9）
  LIST.......= A
  OBJECT.....= A

SYM-TBL(Y/N)= Y

TEMPORARY FILE (A-P)
  UNIT.......= A

** ASSEMBLE START **                                       ……アセンブル実行開始メッセージの出力（5.10）


OPTIMIZATION STARTS

TIMES =     1,   CODE = 0013H

OPTIMIZATION COMPLETED,   TIMES =     1

** ASSEMBLE COMPLETE,     1 ERRORS **                      ……アセンブル終了メッセージの出力（5.11）


** FILE LIST **
  SOURCE...= A:7500MN.ASM                                  }
  LIST.....= A:7500MN.PRN                                  }
  OBJECT...= A:7500MN.HEX                                  }  ……入出力ファイル・リストの出力　（5.12）
  SYM-TBL..= A:7500MN.SYM                                  }

A>
```

第６章　出力形式

６．１　オブジェクト・プログラム

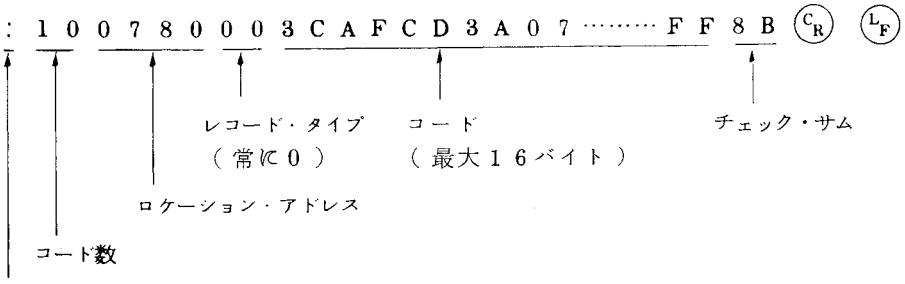
ソース・プログラムをアセンブルすると，アセンブラは絶対番地形式のオブジェクト・プログラムを操作5.9で割り当てられた装置上にＨＥＸ形式で出力します．

本アセンブラはオブジェクト出力装置ＰＵＮ(紙テープ・パンチ)が指定された場合のみ偶数パリティを付加し，他の場合にはパリティを付加しません．紙テープ・ベースでＲＯＭ発注コードを作成する場合は，なるべくＰＵＮ指定を行ってください．

いったんＦＤ上にオブジェクトを出力し，後にＰＩＰコマンドを用いて紙テープに変換するとパリティが付きませんので注意してください．

以下にＨＥＸ形式を示します．

(1)　ＨＥＸ形式



レコード・マーク …… コロン『：』

コード数 …… １６進数２桁で示されます．１レコードにおさめられるバイト数で，最大は１６バイトになります．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．現状は常に０ですが，将来の拡張のために保存してあります．

コード …… １バイトは上位４ビット，下位４ビットに分け各々１６進数で示されます．コードは最大１６バイトまでおさめることができます．

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

８ビットの２進減算器を仮定してボローを無視した減算を行ないます．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．

```
    0000  0000
  −)0001  0000
    1111  0000
```

次にロケーション・アドレスの上位８ビット０７Ｈをこの結果から減算します.

```
    1111  0000
  −)0000  0111
    1110  1001
```

更にロケーション・アドレスの下位８ビット８０Ｈを減算します.

```
    1110  1001
  −)1000  0000
    0110  1001
```

最後のデータＦＦまでこの減算方法で繰返します.

```
    0110  1001
  −)0011  1100  ……  3CH
    0010  1101
  −)1010  1111  ……  0AFH
    0111  1110
          ・
          ・
          ・
          ・
  −)1111  1111  ……  0FFH
    1000  1011
    ──────────
    チェック・サム　8BH
```

注　オブジェクト・コードの終了を示す最終レコードは次のようになります.

```
：００ ０００００ ０１ ＦＦ
  ↑     ↑      ↑   ↑
  コード数  アドレス  レコード・タイプ  チェック・サム
```

エラーはＮＯＰ（００Ｈ）を割り当てます.

6．2　出力リスト

出力リストには，アセンブル・リスト，シンボル・リストおよびクロスレファレンス・リストがあります．

(1)　アセンブル・リスト

アセンブル・リストは，以下の情報とシンボル・リストを含んでいます．

(a)　E　　1カラム　エラー・コード

(b)　STNO　4カラム　ステートメント・ナンバー

(c)　ADRS　4カラム　ロケーション・アドレス

(d)　OBJ　8カラム　オブジェクト・コード

(e)　M　　1カラム　マクロ・ネスティング・レベル

(f)　SOURCE STATEMENT　ソース・ステートメント

アセンブル・リストの例

```
** UPD7500-A  ASSEMBLE LIST **        (    XX XXX 83          )  PAGE     1

                ( EXAMPLE                                  )

E STNO ADRS OBJECT    M    SOURCE STATEMENT

     1                              TITLE           EXAMPLE'
     2                    ;
     3                    ;=============================================
     4                    ;=                                           =
     5                    ;=              UPD 7500 SERIES              =
     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,  TIMES =     1
```

(2) 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．

```
** UPD7500-A  SYMBOL LIST **            (    XX XXX 83          )  PAGE     1
SYMBOL ADRS  SYMBOL ADRS  SYMBOL ADRS  SYMBOL ADRS  SYMBOL ADRS
LABEL1 0010  LABEL2 0020  SUBR   0200  TABLE  00D0
```

(3) 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．

```
** UPD7500-A  CROSS REFERENCE LIST **  (    XX XXX 83          )  PAGE     1
SYMBOL DEF. REF.  ( STATEMENT NUMBER )

LABEL1 0010
LABEL2 0011
SUBR   0027 0021
TABLE  0021 0024  0031
```

6.3 ソース・ファイル・エラー・メッセージ

ソース・プログラム中にエラーがある場合は，アセンブル・リスト上の対応する文の左側（E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が印字されます．

(1) A（ADDRESS ERROR）

ロケーション・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）

左かっこと右かっこの整合がとれていないか，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 JMP 102B …… 2進定数に“2”は使えない．
　 JMP 9Q …… 8進定数に“9”は使えない．

(6) 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ことを示します（予約語については付録を参照）．

例 ILS : ASC ……予約語（命令）をレーベルに使用している．
　 END : ST ……予約語（疑似命令）　〃
　 SHR : ASC ……予約語（演算子）　〃
　 HL : LAI ……予約語（レジスタ名）　〃

Lエラーのあったソース・ステートメントに対しては，アセンブラはマシン・コードを生成しません．

(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．

SYMBOL1 EQU 1---┐ 有効文字数が6文字のため，同じネーム
SYMBOL2 EQU 2---┘ （SYMBOL）となる．

(8) N （NESTING ERROR）

MACRO文，IF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）

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付けられた値と，PASS2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）．

保守／廃止

(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）

式の記述が複雑な場合に途中結果をロードするスタックがオーバフローしたことを示します.

(13) T

ＩＦ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　3.9**参照）.

(17) X

マクロ参照のオペランド欄に記述されたパラメータに誤りがあることを示します.

6.4　ソース・ファイル・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	MACRO文,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.
X		マクロ参照時のパラメータの記述が間違っている.

6.5　実行時のエラー・メッセージ

アセンブラはアセンブル続行が不可能なエラーが生ずると次に示すように原因を表すメッセージとファイル・リスト（5.9参照）をコンソールに出力して処理を中断し，制御をOSへ戻します.

(1) EOF ERROR

ソース・プログラム中にEND文がなかったことを示します.

(2) TABLE ERROR

シンボル・テーブルまたはマクロ・ボディを登録するためのエリアがないことを示します.

(3) STACK AREA OVERFLOWD

スタック・エリアがオーバフローしたことを示します.

(4) MACRO REFERENCE OVERFLOWED

マクロの展開が32767回以上になったことを示します.

(5) CAN NOT FILE OPEN

ファイルがオープンされなかったことを示します.

(6) FILE NOT FOUND

入力ファイル名が，ディスクに登録されていないことを示します.

(7)　FILE ERROR 13

ファイルがすでに存在していることを示します.

(8)　FILE ERROR 15

ソース・プログラム中に126文字を越える行があることを示します.

付録Ⅰ　予約語

予約語には，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・プログラム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	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．

保守／廃止

予約語一覧（７５００セットＡ）

命令	ACSC	ADSC		AESC	AISC	AHSC	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		
	ENDM	EQU	EXITM	GJMP	GLOBAL	IF		
	MACRO	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（７５００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	XAH	XAL	
	XAM	XDS	XHDR	XIS	XLDR		
疑似命令	DB	EJECT	ELSE	END	ENDIF	ENDM	
	EQU	EXITM	GJMP	GLOBAL	IF	MACRO	
	ORG	PRINT	SET	SPACE	TITLE		
演算子	AND EQ MOD NE NOT OR SHL SHR						
	XOR						
レジスタ	HL HL+ HL−						

保守／廃止

予約語一覧（７５０１）

命令	ACSC	AISC	ANL	ASC	CAL	CALL	
	CMA	DDRS	DLS	EXL	HALT	IDRS	
	ILS	IP	IPL	IP1	IP54	JAM	JCP
	JMP	L	LADR	LAI	LAM	LAMT	
	LDS	LHI	LHLI	LIS	NOP		
	OP	OPL	OP3	OP54	ORL	RAR	
	RC	RMB	RT	RTS	SC	SIO	
	SKABT	SKAEI	SKAEM	SKC	SKI		
	SKLEI	SKMBF	SKMBT	SKMBT	SMB		
	ST	STII	STOP	TAMMOD			
	TAMSIO	TAMSP	TCNTAM	TIMER			
	TSIOAM	X	XADR	XAH	XAL		
	XAM	XDS	XHDR	XIS	XLDR		
疑似命令	DB	EJECT	ELSE	END	ENDIF	ENDM	
	EQU	EXITM	GJMP	GLOBAL	IF	MACRO	
	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（７５０２，７５０２Ａ）

命令	ACSC	AISC	ANL	ANP	ASC	CALL		
	CALT	CMA	DDRS	DES	DI	DLS	EI	
	EXL	HALT	IDRS	IES	ILS	IP	IPL	IP1
	IP54	JAM	JCP	JMP	L	LADR	LAI	
	LAM	LAMT	LDEI	LDI	LDS	LEI	LHI	
	LHLI	LHLT	LIS	LLI	NOP	OP	OPL	OP3
	OP54	ORL	ORP	POPDE	POPHL			
	PSHDE	PSHHL	RAR	RC	RMB	RT		
	RTPSW	RTS	SC	SIO	SKABT	SKAEI		
	SKAEM	SKC	SKDEI	SKEEI	SKHEI			
	SKI	SKLEI	SKMBF	SKMBT	SMB	ST		
	STOP	TAD	TAE	TAH	TAL	TAMMOD		
	TAMSIO	TAMSP	TCNTAM	TDA	TEA			
	THA	TIMER	TLA	TSIOAM	TSPAM	X		
	XAD	XADR	XAE	XAH	XAL	XAM		
	XHDR	XIS	XLDR					
疑似命令	DB	DET	EJECT	ELSE	END	ENDIF	ENDM	
	EQU	EXITM	GJMP	GLOBAL	IF	MACRO		
	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			

保守／廃止

予約語一覧（７５０３，７５０３Ａ，７５０７，７５０７Ｂ，７５０７Ｈ，
７５０７Ｓ，７５０８，７５０８Ｂ，７５０８Ｈ，７５１４）

命令	ACSC	AISC	ANL	ANP	ASC	CALL		
	CALT	CMA	DDRS	DES	DI	DLS	EI	
	EXL	HALT	IDRS	IES	ILS	IP	IPL	IP1
	IP54	JAM	JCP	JMP	L	LADR	LAI	
	LAM	LAMTL	LDEI	LDI	LDS	LEI	LHI	
	LHLI	LHLT	LIS	LLI	NOP	OP	OPL	OP3
	OP54	ORL	ORP	POPDE	POPHL			
	PSHDE	PSHHL	RAR	RC	RMB	RT		
	RTPSW	RTS	SC	SIO	SKABT	SKAEI		
	SKAEM	SKC	SKDEI	SKEEI	SKHEI			
	SKI	SKLEI	SKMBF	SKMBT	SMB	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		
	ENDM	EQU	EXITM	GJMP	GLOBAL	IF		
	MACRO	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（７５２０）

命令	ACSC	AISC	ASC	CAL	CALL	CMA	
	DDRS	DLS	EXL	IDRS	ILS	IPL	IP1
	JAM	JCP	JMP	L	LADR	LAI	LAM
	LAMT	LDS	LHI	LHLI	LIS	NOP	OPL
	OP3	RC	RMB	RT	RTS	SC	SKABT
	SKAEI	SKAEM	SKC	SKMBF	SKMBT		
	SMB	ST	STII	X	XADR	XAH	XAL
	XAM	XDS	XIS				
疑似命令	DB	EJECT	ELSE		END	ENDIF	ENDM
	EQU	EXITM	GJMP		GLOBAL	IF	MACRO
	ORG	PRINT	SET		SPACE	TITLE	
演算子	AND EQ MOD NE NOT OR SHL SHR XOR						
レジスタ	HL HL+ HL−						

保守／廃止

予約語一覧（７５０６）

命令	ACSC	AISC	ANL	ASC	CAL	CALL	
	CMA	DDRS	DLS	EXL	HALT	IDRS	
	ILS	IP	IPL	IP1	IP54	JAM	JCP
	JMP	L	LADR	LAI	LAM	LAMT	
	LDS	LHI	LHLI	LIS	NOP		
	OP	OPL	OP54	ORL	RAR		
	RC	RMB	RT	RTS	SC		
	SKABT	SKAEI	SKAEM	SKC	SKI		
	SKLEI	SKMBF	SKMBT	SMB			
	ST	STII	STOP	TAMMOD			
	TAMSP	TIMER					
	X	XAH	XAL				
	XAM	XDS	XHDR	XIS	XLDR		
疑似命令	DB	EJECT	ELSE	END	ENDIF	ENDM	
	EQU	EXITM	GJMP	GLOBAL	IF	MACRO	
	ORG	PRINT	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（７５１６，７５１６Ｈ，７５１９，７５１９Ｈ）

命令	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 注	JMPL	L	LADR	
	LAI	LAM	LDEI	LDI	LDS	LEI	LHI	
	LHLI	LHLT	LIS	LLI	NOP	OP	OPL	OP3
	OP54	ORL	ORP	POPDE	POPHL	PSHDE		
	PSHHL	RAL	RAR	RC	RMB	RT	RTPSW	
	RTS	SC	SDSB	SESB	SHSB	SIO	SKABT	
	SKAEI	SKAEM	SKC	SKDEI	SKEEI			
	SKHEI	SKI	SKLEI	SKMBF	SKMBT	SKMEI		
	SLSB	SMB	ST	STOP	TAD	TAE		
	TAH	TAL	TAMMOD	TAMSIO	TAMSP			
	TCNTAN	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		
	ENDM	EQU	EXITM	GJMP	GLOBAL	IF		
	MACRO	ORG	PRINT	SET	SPACE			
	TITLE							
演算子	ADD EQ MOD NE NOT OR SHL SHR							
	XOR							
レジスタ	HL HL+ HL− DE DL							

注　７５１６，７５１６Ｈのみ使用できます．

保守／廃止

予約語一覧（７５２７，７５２７Ａ，７５２８，７５２８Ａ，７５３３，
７５３７，７５３７Ａ，７５３８，７５３８Ａ）

命令	ACSC	AISC	ANL	注 ANP	ASC	CALL	CALT
	CMA	DDRS	DI	DLS	EI	EXL	HALT
	IDRS	ILS	IP	IPL	IP54	JAM	JCP
	JMP	L	LADR	LAI	LAM	LAMTL	LDS
	LHLI	LHLT	LIS	NOP	OP	OPL	OP3
	OP54	ORL	注 ORP	RAR	RC	RMB	RPBL
	RT	RTPSW	RTS	SC	SIO	SKABT	SKAEI
	SKAEM	SKC	SKI	SKLEI	SKMBF	SKMBT	SMB
	SPBL	ST	STOP	TAMMOD	TAMSIO		
	TAMSP	TCNTAM	TIMER	TSIOAM	TSPAM		
	X XADR	XAH	XAL	XAM	XDS	XHDR	XIS
	XLDR						
疑似命令	DB	DET	EJECT	ELSE	END	ENDIF	
	ENDM	EQU	EXITM	GJMP	GLOBAL	IF	
	MACRO	ORG	PRINT	SET	SPACE		
	TITLE						
演算子	AND EQ MOD NE NOT OR SHL SHR						
	XOR						
レジスタ	HL HL+ HL− DL						

注　７５３３のみ使用できます．

付録II　シンボル数

本アセンブラで使用できる最大シンボル数は，４８００個です．ただし，マクロを使用した場合，およそ次の式で計算されるシンボル数だけ使用可能なシンボル数が減少します．

$$\text{減少するシンボル数} = \frac{\text{マクロ・ボディを構成する文字数の合計}}{12}$$

ただし，マクロ・ボディを構成する文字数を計算する場合，タブ文字やＣＲ，ＬＦは，それぞれ１文字として数えます．

保守／廃止

付録III　インストラクション

μPD7500H SET A インストラクション活用表

命令群	ニーモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	マシンサイクル	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	$I_3I_2I_1I_0$	1	A←n4	n4=0-FH	縦積みLAI
	LDI n4	0 0 1 1 0 0 1 0	1 1 1 0 $I_3I_2I_1I_0$	2	D←n4	n4=0-FH	
	LEI n4	0 0 1 1 0 0 0 0	1 1 1 0 $I_3I_2I_1I_0$	2	E←n4	n4=0-FH	
	LHI n4	0 0 1 1 0 0 1 1	1 1 1 0 $I_3I_2I_1I_0$	2	H←n4	n4=0-FH	
	LLI n4	0 0 1 1 0 0 0 1	1 1 1 0 $I_3I_2I_1I_0$	2	L←n4	n4=0-FH	
	LAM pr	0 1 0 R_4	0 0 R_1R_0	½	A←(pr), pr=DL, DE, HL−, HL+, HL		L=FH(HL−) L=0 (HL+)
	LADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 0 $D_3D_2D_1D_0$	2	A←(mem)	mem=0-FFH	
	LDEI byte	0 1 0 0 $I_7I_6I_5I_4$	1 1 1 1 $I_3I_2I_1I_0$	2	DE←byte	byte=0-FFH	
	LHLI byte	0 1 0 0 $I_7I_5I_6I_4$	1 1 1 0 $I_3I_2I_1I_0$	2	HL←byte	byte=0-FFH	縦積み LHLI LHLT
	LHLT taddr	1 1 0 0	$P_3P_2P_1P_0$	2	HL←T(000001100 P_3 P_2 P_1 P_0) taddr=00C0H-00CFH		縦積み LHLI LHLT
	LAMT	0 1 0 1	1 1 1 0	2	A←TABLE(BNK, PC_{11-6},0, C, A)$_H$ (HL)←TABLE(BNK, PC_{11-6},0, C, A)$_L$		
	LAMTL	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 0	3	A←T(BNK, PC_{11-8},A, (HL))$_H$ (HL)←T(BNK, PC_{11-8},A, (HL))$_L$		
	ST	0 1 0 1	0 1 1 1	1	(HL)←A		
	TAD	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 0	2	D↔A		
	TAE	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 0	2	E↔A		
	TAH	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 0	2	H↔A		
	TAL	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 0	2	L↔A		
	TDA	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 1	2	A↔D		
	TEA	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 1	2	A↔E		
	THA	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 1	2	A↔H		
	TLA	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 1	2	A↔L		
	XAD	0 1 0 0	1 0 1 0	1	A↔D		
	XAE	0 1 0 0	1 0 1 1	1	A↔E		
	XAH	0 1 1 1	1 0 1 0	1	A↔H		
	XAL	0 1 1 1	1 0 1 1	1	A↔L		
	XAM pr	0 1 0 R_4	0 1 R_1R_0	½	A↔(pr) pr=DL, DE, HL−, HL+, HL		L=FH(HL−) L=0 (HL+)
	XADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 1 $D_3D_2D_1D_0$	2	A↔(mem)	mem=0-FFH	
	XHDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 0 $D_3D_2D_1D_0$	2	H↔(mem)	mem=0-FFH	
	XLDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 1 $D_3D_2D_1D_0$	2	L↔(mem)	mem=0-FFH	
演算命令	AISC n4	0 0 0 0	$I_3I_2I_1I_0$	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
	ADSC	0 0 1 1 1 0 1 0	1 1 1 0 1 0 0 1	⅔	A←A+D		carry
	AESC	0 0 1 1 1 0 0 0	1 1 1 0 1 0 0 1	⅔	A←A+E		carry
	AHSC	0 0 1 1 1 0 1 1	1 1 1 0 1 0 0 1	⅔	A←A+H		carry
	ALSC	0 0 1 1 1 0 0 1	1 1 1 0 1 0 0 1	⅔	A←A+L		carry
	SDSB	0 0 1 1 1 0 1 0	1 1 1 0 1 0 0 0	⅔	A←A−D		borrow
	SESB	0 0 1 1 1 0 0 0	1 1 1 0 1 0 0 0	⅔	A←A−E		borrow
	SHSB	0 0 1 1 1 0 1 1	1 1 1 0 1 0 0 0	⅔	A←A−H		borrow
	SLSB	0 0 1 1 1 0 0 1	1 1 1 0 1 0 0 0	⅔	A←A−L		borrow
	EXL	0 1 1 1	1 1 1 0	1	A←A∀(HL)		
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	A←A∧(HL)		
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	A←A∨(HL)		
レジスタアキュームレータ操作命令	CMA	0 1 1 1	1 1 1 1	1	A←$\overline{A}$		
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	C←A_0, A_3←C A_{n-1}←A_n (n=1-3)		
	RAL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 1	2	C←A_3, A_n←A_{n-1} A_0←C (n=1-3)		
キャリーフラグ操作命令	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	IES	0 1 0 0	1 0 0 1	½	E←E+1		E=0
	ILS	0 1 0 1	1 0 0 1	½	L←L+1		L=0
	IDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 0 1	2	DE←DE+1		
	IHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 0 1	2	HL←HL+1		
	IDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 1 $D_3D_2D_1D_0$	⅔	(mem)←(mem)+1 mem=0-FFH		(mem)=0
	DES	0 1 0 0	1 0 0 0	½	E←E−1		E=FH
	DLS	0 1 0 1	1 0 0 0	½	L←L−1		L=FH
	DDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 0 0	2	DE←DE−1		
	DHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 0 0	2	HL←HL−1		
	DDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 0 $D_3D_2D_1D_0$	⅔	(mem)←(mem)−1 mem=0-FFH		(mem)=FH

命令群	ニーモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	マシンサイクル	オペレーション		スキップ条件
メモリビットリ操作命令	RMB bit	0 1 1 0	1 0 B_1B_0	1	(HL) bit←0	bit=0-3	
	SMB bit	0 1 1 0	1 1 B_1B_0	1	(HL) bit←1	bit=0-3	
ジャンプ命令	JMP addr1	0 0 1 0 $P_7P_6P_5P_4$	$P_{11}P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	PC_{11-0}←P_{11-0}	addr1=0-1FFFH	
	JMPL addr	0 0 1 1 0 0 P_{13} 0 $P_7P_6P_5P_4$	1 1 1 1 $P_{11}P_{10}P_9P_8$ $P_3P_2P_1P_0$	3	BNK←P_{13}, PC_{11-0}←P_{11-0} addr=0000H-1FFFH		
	JCP addr	1 0 P_5P_4	$P_3P_2P_1P_0$	1	PC_{5-0}←P_{5-0}	addr=0-1FFFH	
	JAM addr3	0 0 1 1 0 0 0 1	1 1 1 1 $P_3P_2P_1P_0$	2	PC_{11-8}←P_{3-0}, PC_{7-4}←A_{3-0} PC_{3-0}←(HL)	addr3=0-FH	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 $P_7P_6P_5P_4$	0 $P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	(SP−1) (SP−2) (SP−4)←PC_{11-0} (SP−3)←PSW, SP←SP−4 BNK←0, PC_{11-0}←0, P_{10-0} caddr=0000H-07FFH		
	CALT taddr1	1 1 P_5P_4	$P_3P_2P_1P_0$	2	(SP−1) (SP−2) (SP−4)←PC_{11-0} (SP−3)←PSW, SP←SP−4 BNK←0, $PC_{11,10,6,5}$←0 $PC_{9-7,4-0}$←T(0000011$P_5P_4P_3P_2P_1P_0$) taddr1=00D0H-00FFH		
	RT	0 1 0 1	0 0 1 1	2	PC_{11-0}←(SP) (SP+2) (SP+3) BNK←(SP+1)$_L$, SP←SP+4		
	RTS	0 1 0 1	1 0 1 1	3	PC_{11-0}←(SP) (SP+2) (SP+3) BNK←(SP+1)$_L$, SP←SP+4 then skip unconditionally		無条件
	RTPSW	0 1 0 0	0 0 1 1	2	PC_{11-0}←(SP) (SP+2) (SP+3) PSW←(SP+1), SP←SP+4		
	PSHDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 0	2	(SP−1)←D, (SP−2)←E SP←SP−2		
	PSHHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 0	2	(SP−1)←H, (SP−2)←L SP←SP−2		
	POPDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 1	2	E←(SP), D←(SP+1) SP←SP+2		
	POPHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 1	2	L←(SP), H←(SP+1) SP←SP+2		
	TAMSP	0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 1	2	SP_{7-4}←A SP_{3-1}←$(HL)_{3-1}$, SP_0←0		
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	A←SP_{7-4} $(HL)_{3-1}$←SP_{3-1}, $(HL)_0$←0		
スキップ命令	SKC	0 1 0 1	1 0 1 0	½	skip if C=1		C=1
	SKMBT bit	0 1 1 0	0 1 B_1B_0	½	skip if (HL) bit=1	bit=0-3	(HL) bit=1
	SKABT bit	0 1 1 1	0 1 B_1B_0	½	skip if A bit=1	bit=0-3	A bit=1
	SKMBF bit	0 1 1 0	0 0 B_1B_0	½	skip if (HL) bit=0	bit=0-3	(HL) bit=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	1 1 1 1 $I_3I_2I_1I_0$	⅔	skip if A=n4	n4=0-FH	A=n4
	SKDEI n4	0 0 1 1 0 1 1 0	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if D=n4	n4=0-FH	D=n4
	SKEEI n4	0 0 1 1 0 1 0 0	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if E=n4	n4=0-FH	E=n4
	SKHEI n4	0 0 1 1 0 1 1 1	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if H=n4	n4=0-FH	H=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if L=n4	n4=0-FH	L=n4
	SKMEI n4	0 0 1 1 0 1 1 1	1 1 1 1 $I_3I_2I_1I_0$	⅔	skip if (HL)=n4	n4=0-FH	(HL)=n4
SIO制御命令	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	SIO_{7-4}←A SIO_{3-0}←(HL)		
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	A←SIO_{7-4} (HL)←SIO_{3-0}		
	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_{7-4}←A MOD_{3-0}←(HL)		
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0_0 1 0	2	Start Timer		
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	A←CT_{7-4} (HL)←CT_{3-0}		
割込制御命令	EI n4	0 0 1 1 1 0 0 1	1 1 1 1 $I_3I_2I_1I_0$	2	IME←1 IE3-0←IE3-0∨n4	(n4=0) (n4=1-FH)	
	DI n4	0 0 1 1 1 0 0 0	1 1 1 1 $I_3I_2I_1I_0$	2	IME←0 IE3-0←IE3-0∧$\overline{n4}$	(n4=0) (n4=1-FH)	
	SKI n4	0 0 1 1 0 1 0 0	1 1 1 1 $I_3I_2I_1I_0$	⅔	skip if INT RQF=1 then reset INT RQF		INT RQF=1
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	1 1 1 1 $D_7D_6D_5D_4$	2	A←Port (addr5)	addr5= 0, 1, 4-BH	
	IP1	0 1 1 1	0 0 0 1	1	A←Port 1		
	IP54	0 0 1 1 0 0 1 1	1 1 1 1 1 0 0 0	2	A←Port 5 (HL)←Port 4		
	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	1 1 1 1 $D_3D_2D_1D_0$	2	Port/Mode reg. (addr4)←A addr4=1-FH		
	OP3	0 1 1 1	0 0 1 1	1	Port 3←A		
	OP54	0 0 1 1 0 0 1 1	1 1 1 1 1 1 0 0	2	Port 5←A Port 4←(HL)		
	ANP addr6, n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 0 $D_3D_2D_1D_0$	2	Port (addr6)←Port (addr6)∧n4 addr6.=2-BH, n4=0-FH		
	ORP addr6, n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 1 $D_3D_2D_1D_0$	2	Port (addr6)←Port (addr6)∨n4 addr6.=2-BH, n4=0-FH		
	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 Operation		

μPD7500H SET B インストラクション活用表

命令群	ニーモニック	命令コード		マシンサイクル	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	$I_3 I_2 I_1 I_0$	1	A←n4　n4=0-FH	縦積LAI
	LHI n3	0 0 1 0	$1\ I_2 I_1 I_0$	1	H←0$I_2 I_1 I_0$　n3=0-7	
	LAM pr1	0 1 0 1	$0\ 0 R_1 R_0$	½	A←(pr1)　pr1=HL−,HL+,HL	L=FH(HL−) L=0(HL+)
	LADR mem	0 0 1 1 $D_7 D_6 D_5 D_4$	1 0 0 0 $D_3 D_2 D_1 D_0$	2	A←(mem)　mem=0-FFH	
	LHLI n5	1 1 0 I_4	$I_3 I_2 I_1 I_0$	1	HL←000$I_4 I_3 I_2 I_1 I_0$　n5=0-1FH	縦積LHLI
	LAMT	0 1 0 1	1 1 1 0	2	A←TABLE(BNK,PC_{11-6},0,C,A_{5-0})H (HL)←TABLE(BNK,PC_{11-6},0,C,A_{5-0})L	
	LAMTL	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 0	3	A←TABLE(BNK,PC_{11-8},A_{7-4},(HL)$_{7-4}$)H (HL)←TABLE(BNK,PC_{11-8},A_{3-0},(HL)$_{3-0}$)L	
	ST	0 1 0 1	0 1 1 1	1	(HL)←A	
	STII n4	0 1 0 0	$I_3 I_2 I_1 I_0$	1	(HL)←n4,L←L+1　n4=0-FH	
	XAH	0 1 1 1	1 0 1 0	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	A↔L	
	XAM pr1	0 1 0 1	$0\ 1 R_1 R_0$	½	A↔(pr1)　pr1=HL−,HL+,HL	L=FH(HL−) L=0(HL+)
	XADR mem	0 0 1 1 $D_7 D_6 D_5 D_4$	1 0 0 1 $D_3 D_2 D_1 D_0$	2	A↔(mem)　mem=0-FFH	
	XHDR mem	0 0 1 1 $D_7 D_6 D_5 D_4$	1 0 1 0 $D_3 D_2 D_1 D_0$	2	H↔(mem)　mem=0-FFH	
	XLDR mem	0 0 1 1 $D_7 D_6 D_5 D_4$	1 0 1 1 $D_3 D_2 D_1 D_0$	2	L↔(mem)　mem=0-FFH	
演算命令	AISC n4	0 0 0 0	$I_3 I_2 I_1 I_0$	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)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	A←A∨(HL)	
アキュームレーター・キャリー・フラグ操作命令	CMA	0 1 1 1	1 1 1 1	1	A←$\bar{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	C←A_0,A_3←C An←A_{n+1}　(n=1-3)	
	RAL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 1	2	C←A_3,A_0←C, An←A_{n-1}　(n=1-3)	
	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 $D_7 D_6 D_5 D_4$	1 1 0 1 $D_3 D_2 D_1 D_0$	⅔	(mem)←(mem)+1	(mem)=0
	DLS	0 1 0 1	1 0 0 0	½	L←L−1	L=FH
	DDRS mem	0 0 1 1 $D_7 D_6 D_5 D_4$	1 1 0 0 $D_3 D_2 D_1 D_0$	⅔	(mem)←(mem)−1	(mem)=FH
ビットメモリ操作命令	RMB bit	0 1 1 0	$1\ 0 B_1 B_0$	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	$1\ 1 B_1 B_0$	1	(HL)bit←1　bit=0-3	
ジャンプ命令	JMP addr2	0 0 1 0 $P_7 P_6 P_5 P_4$	$0 P_{10} P_9 P_8$ $P_3 P_2 P_1 P_0$	2	PC_{11-0}←0,P_{10-0}	
	JMPL addr	0 0 1 1 0 0 P_{13} 0 $P_7 P_6 P_5 P_4$	1 1 1 1 $P_{11} P_{10} P_9 P_8$ $P_3 P_2 P_1 P_0$	3	BNK←P_{13} PC_{11-0}←$P_{11\ 0}$	
	JCP addr	1 0 $P_5 P_4$	$P_3 P_2 P_1 P_0$	1	PC_{5-0}←P_{5-0}	
	JAM addr3	0 0 1 1 0 0 0 1	1 1 1 1 $P_3 P_2 P_1 P_0$	2	PC_{11-8}←P_{3-0},PC_{7-4}←$A_{3\ 0}$ PC_{3-0}←(HL)$_{3-0}$	

命令群	ニーモニック	命令コード		マシンサイクル	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 $P_7 P_6 P_5 P_4$	$0\ P_{10} P_9 P_8$ $P_3 P_2 P_1 P_0$	2	(SP-1)(SP-2)(SP-4)←PC_{11-0} (SP-3)←PSW,SP←SP-4 BNK←0,PC_{11-0}←0,P_{10-0}	
	CAL caddr1	1 1 1 P_4	$P_3 P_2 P_1 P_0$	2	(SP-1)(SP-2)(SP-4)←PC_{11-0} (SP-3)←PSW,SP←SP-4 BNK←0 PC_{11-0}←0001$P_4 P_3$000$P_2 P_1 P_0$	
	RT	0 1 0 1	0 0 1 1	2	PC_{11-0}←(SP)(SP+2)(SP+3) BNK←(SP+1),SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	3	PC_{11-0}←(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_{7-4}←A SP_{3-1}←(HL)$_{3-1}$,SP_0←0	
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	A←SP_{7-4} (HL)$_{3-1}$←SP_{3-1},(HL)$_0$←0	
スキップ命令	SKC	0 1 0 1	1 0 1 0	½	Skip if C=1	C=1
	SKABT bit	0 1 1 1	$0\ 1 B_1 B_0$	½	Skifp if A_{bit}=1　bit=0-3	A_{bit}=1
	SKMBT bit	0 1 1 0	$0\ 1 B_1 B_0$	½	Skip if $(HL)_{bit}$=1　bit=0-3	$(HL)_{bit}$=1
	SKMBF bit	0 1 1 0	$0\ 0 B_1 B_0$	½	Skip if $(HL)_{bit}$=0　bit=0-3	$(HL)_{bit}$=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	1 1 1 1 $I_3 I_2 I_1 I_0$	⅔	Skip if A=n4　n4=0-FH	A=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 1 $I_3 I_2 I_1 I_0$	⅔	Skip if L=n4　n4=0-FH	L=n4
	SKMEI n4	0 0 1 1 0 1 1 1	1 1 1 1 $I_3 I_2 I_1 I_0$	⅔	Skip if (HL)=n4　n4=0-FH	(HL)=n4
	SKI n4	0 0 1 1 0 1 0 0	1 1 1 1 $I_3 I_2 I_1 I_0$	⅔	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_H←A,SIO_L←(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	A←SIO_H,(HL)←SIO_L	
	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_H←A,MOD_L←(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	CT_{7-0}←0,INTT RQF←0	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	A←CT_{7-4},(HL)←CT_{3-0}	
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	1 1 1 1 $D_3 D_2 D_1 D_0$	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	1 1 1 1 $D_3 D_2 D_1 D_0$	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	

保守／廃止

μPD7520　インストラクション活用表

命令群	ニモニック	オペランド	命令コード	マシンサイクル	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_3 I_2 I_1 I_0	1	A←n4	たてつみ LAI
	LHI	n2	0 0 1 0 1 0 I_1 I_0	1	H←n2	
	LHLI	n5	1 1 0 I_4 I_3 I_2 I_1 I_0	1	HL←n5 (H←0I_4, L←$I_{3\ 0}$)	たてつみ LHLI
	LAMT		0 1 0 1 1 1 1 0	2	A←ROM($PC_{9\ 6}$,0,C,A)$_H$ (HL)←ROM($PC_{9\ 6}$,0,C,A)$_L$	
	L		0 1 0 1 0 0 1 0	1	A←(HL)	
	LIS		0 1 0 1 0 0 0 1	½~3	A←(HL), L←L+1 Skip if L=0	L=0
	LDS		0 1 0 1 0 0 0 0	½~3	A←(HL), L←L−1 Skip if L=15	L=15
	LADR	addr1	0 0 1 1 1 0 0 0 / 0 0 D_5 D_4 D_3 D_2 D_1 D_0	2	A←(addr1)	
	ST		0 1 0 1 0 1 1 1	1	(HL)←A	
	STII	n4	0 1 0 0 I_3 I_2 I_1 I_0	1	(HL)←n4 L←L+1	
	XAH		0 1 1 1 1 0 1 0	1	A↔H	
	XAL		0 1 1 1 1 0 1 1	1	A↔L	
	X		0 1 0 1 0 1 1 0	1	A↔(HL)	
	XIS		0 1 0 1 0 1 0 1	½~3	A↔(HL), L←L+1 Skip if L=0	L=0
	XDS		0 1 0 1 0 1 0 0	½~3	A↔(HL), L←L−1 Skip if L=15	L=15
	XADR	addr1	0 0 1 1 1 0 0 1 / 0 0 D_5 D_4 D_3 D_2 D_1 D_0	2	A↔(addr1)	
演算命令	AISC	n4	0 0 0 0 I_3 I_2 I_1 I_0	½~3	A←A+n4 Skip on carry	Carry
	ASC		0 1 1 1 1 1 0 1	½~3	A←A+(HL) Skip on carry	Carry
	ACSC		0 1 1 1 1 1 0 0	½~3	A, C←A+(HL)+C Skip on carry	Carry
	EXL		0 1 1 1 1 1 1 0	1	A←A ∀ (HL)	
	CMA		0 1 1 1 1 1 1 1	1	A←$\overline{A}$	
	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	½~3	L←L+1 Skip if L=0	L=0
	IDRS	addr1	0 0 1 1 1 1 0 1 / 0 0 D_5 D_4 D_3 D_2 D_1 D_0	⅔~4	(addr1)←(addr1)+1 Skip if (addr1)=0	(addr1) =0

命令群	ニモニック	オペランド	命令コード	マシンサイクル	オペレーション	スキップ条件
増減命令	DLS		0 1 0 1 1 0 0 0	½~3	L←L−1 Skip if L=15	L=15
	DDRS	addr1	0 0 1 1 1 1 0 0 / 0 0 D_5 D_4 D_3 D_2 D_1 D_0	⅔~4	(addr1)←(addr1)−1 Skip if (addr1)=15	(addr1) =15
操作メモリビット命令	RMB	bit	0 1 1 0 1 0 B_1 B_0	1	(HL)bit←0	
	SMB	bit	0 1 1 0 1 1 B_1 B_0	1	(HL)bit←1	
ジャンプ命令	JMP	addr	0 0 1 0 0 0 P_9 P_8 / P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0	2	$PC_{9\ 0}$←P_{9-0}	
	JAM	addr2	0 0 1 1 1 1 1 1 / 0 0 0 1 0 0 P_1 P_0	2	PC_{9-8}←P_{1-0} PC_{7-4}←A PC_{3-0}←(HL)	
	JCP	addr	1 0 P_5 P_4 P_3 P_2 P_1 P_0	1	PC_{5-0}←P_{5-0}	
サブルーチン制御命令	CALL	addr	0 0 1 1 0 0 P_9 P_8 / P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0	2	STACK←PC+2 PC_{9-0}←P_{9-0}	
	CAL	addr	1 1 1 P_4 P_3 P_2 P_1 P_0	1	STACK←PC+1 PC_{9-0}←01$P_4P_3$000$P_2P_1P_0$	
	RT		0 1 0 1 0 0 1 1	1	PC←STACK	
	RTS		0 1 0 1 1 0 1 1	2~3	PC←STACK, then skip.	無条件
スキップ命令	SKC		0 1 0 1 1 0 1 0	½~3	Skip if C=1	C=1
	SKMBT	bit	0 1 1 0 0 1 B_1 B_0	½~3	Skip if (HL)bit=1	(HL)bit =1
	SKMBF	bit	0 1 1 0 0 0 B_1 B_0	½~3	Skip if (HL)bit=0	(HL)bit =0
	SKABT	bit	0 1 1 1 0 1 B_1 B_0	½~3	Skip if Abit=1	Abit =1
	SKAEI	n4	0 0 1 1 1 1 1 1 / 0 1 1 0 I_3 I_2 I_1 I_0	⅔~4	Skip if A=n4	A=n4
	SKAEM		0 1 0 1 1 1 1 1	½~3	Skip if A=(HL)	A =(HL)
入出力命令	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	
	OPL		0 1 1 1 0 0 1 0	1	PORT(L)←A	
	OP3		0 1 1 1 0 0 1 1	1	PORT3←A	
その他	NOP		0 0 0 0 0 0 0 0	1	No operation	

μPD7501 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		マシン・サイクル	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	$I_3 I_2 I_1 I_0$	1	A←n4　n4=0-FH	縦積LAI
	LHI n3	0 0 1 0	1 $I_2 I_1 I_0$	1	H←0$I_2I_1I_0$　n3=0-7	
	LAM pr	0 1 0 1	0 0 R_1R_0	½	A←(pr)　pr=HL−,HL+,HL	L=FH(HL−) L=0(HL+)
	LADR mem	0 0 1 1 0 $D_6D_5D_4$	1 0 0 0 $D_3D_2D_1D_0$	2	A←(mem)　mem=0-5FH	
	LHLI n5	1 1 0 I_4	$I_3 I_2 I_1 I_0$	1	HL←000$I_4I_3I_2I_1I_0$　n5=0-1FH	縦積LHLI
	LAMT	0 1 0 1	1 1 1 0	2	A←ROM(PC_{9-6}, 0, C, A_{3-0})$_H$ (HL)←ROM(PC_{9-6}, 0, C, A_{3-0})$_L$	
	ST	0 1 0 1	0 1 1 1	1	(HL)←A	
	STII n4	0 1 0 0	$I_3 I_2 I_1 I_0$	1	(HL)←n4, L←L+1　n4=0-FH	
	XAH	0 1 1 1	1 0 1 0	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	A↔L	
	XAM pr	0 1 0 1	0 1 R_1R_0	½	A↔(pr)　pr=HL−,HL+,HL	L=FH(HL−) L=0(HL+)
	XADR mem	0 0 1 1 0 $D_6D_5D_4$	1 0 0 1 $D_3D_2D_1D_0$	2	A↔(mem)　mem=0-5FH	
	XHDR mem	0 0 1 1 0 $D_6D_5D_4$	1 0 1 0 $D_3D_2D_1D_0$	2	H↔(mem)　mem=0-5FH	
	XLDR mem	0 0 1 1 0 $D_6D_5D_4$	1 0 1 1 $D_3D_2D_1D_0$	2	L↔(mem)　mem=0-5FH	
演算命令	AISC n4	0 0 0 0	$I_3 I_2 I_1 I_0$	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)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	A←A∨(HL)	
アキュームレータ・キャリー・フラグ操作命令	CMA	0 1 1 1	1 1 1 1	1	A←$\bar{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	C←A_0, A_3←C A_n←A_{n+1}　(n=1-3)	
	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 $D_6D_5D_4$	1 1 0 1 $D_3D_2D_1D_0$	⅔	(mem)←(mem)+1 mem=0-5FH	(mem)=0
	DLS	0 1 0 1	1 0 0 0	½	L←L−1	L=FH
	DDRS mem	0 0 1 1 0 $D_6D_5D_4$	1 1 0 0 $D_3D_2D_1D_0$	⅔	(mem)←(mem)−1 mem=0-5FH	(mem)=FH
メモリビット操作命令	RMB bit	0 1 1 0	1 0 B_1B_0	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	1 1 B_1B_0	1	(HL)bit←1　bit=0-3	
ジャンプ命令	JMP addr	0 0 1 0 $P_7P_6P_5P_4$	0 0 P_9P_8 $P_3P_2P_1P_0$	2	PC_{9-0}←P_{9-0}	
	JCP addr	1 0 P_5P_4	$P_3P_2P_1P_0$	1	PC_{5-0}←P_{5-0}	
	JAM addr3	0 0 1 1 0 0 0 1	1 1 1 1 0 0 P_1P_0	2	PC_{9-8}←P_{1-0}, PC_{7-4}←A_{3-0} PC_{3-0}←(HL)$_{3-0}$	

命令群	ニモニック	命令コード		マシン・サイクル	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 $P_7P_6P_5P_4$	0 0 P_5P_8 $P_3P_2P_1P_0$	2	(SP-1)(SP-2)(SP-4)←PC_{9-0} (SP-3)←PSW, SP←SP-4 PC_{9-0}←P_{9-0}	
	CAL caddr1	1 1 1 P_4	$P_3P_2P_1P_0$	2	(SP-1)(SP-2)(SP-4)←PC_{9-0} (SP-3)←PSW, SP←SP-4 PC_{9-0}←01$P_4P_3$000$P_2P_1P_0$	
	RT	0 1 0 1	0 0 1 1	2	PC_{9-0}←(SP)(SP+2)(SP+3) SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	3	PC_{9-0}←(SP)(SP+2)(SP+3) 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_{6-4}←A_{2-0} SP_{3-1}←(HL)$_{3-1}$, SP_0←0	
スキップ命令	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_1B_0	½	Skifp if A_{bit}=1　bit=0-3	A_{bit}=1
	SKMBT bit	0 1 1 0	0 1 B_1B_0	½	Skip if (HL)$_{bit}$=1　bit=0-3	(HL)$_{bit}$=1
	SKMBF bit	0 1 1 0	0 0 B_1B_0	½	Skip if (HL)$_{bit}$=0　bit=0-3	(HL)$_{bit}$=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	1 1 1 1 $I_3 I_2 I_1 I_0$	⅔	Skip if A=n4　n4=0-FH	A=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 1 $I_3 I_2 I_1 I_0$	⅔	Skip if L=n4　n4=0-FH	L=n4
	SKI n3	0 0 1 1 0 1 0 0	1 1 1 1 0 $I_2 I_1 I_0$	⅔	Skip if INT RQF=1 then reset INT RQF	INT RQF=1
制御命令SIO	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	SIO_{7-4}←A SIO_{3-0}←(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	A←SIO_{7-4} (HL)←SIO_{3-0}	
	SIO	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 1	2	Start SIO 3-BIT CNT←0, INT0/S RQF←0	
制御命令タイマ	TAMMOD	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 1	2	MOD_{7-4}←A MOD_{3-0}←(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	Start Timer CT_{7-0}←0, INTT RQF←0	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	A←CT_{7-4} (HL)←CT_{3-0}	
入出力命令	IPL	0 1 1 1	0 0 0 0	1	A←Port(L)	
	IP addr2	0 0 1 1 1 1 0 0	1 1 1 1 0 $D_2D_1D_0$	2	A←Port(addr2) addr2=0,1,4-6	
	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 addr1	0 0 1 1 1 1 1 0	1 1 1 1 $D_3D_2D_1D_0$	2	Port/Mode reg.(addr1)←A addr1=3-6, $\overline{BH}$, $\overline{CH}$, $\overline{EH}$, 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)	
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 Operation	

保守／廃止

μPD7502　インストラクション活用表

（適用品種：μPD7502, 7502A）

命令群	ニモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	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	$I_3 I_2 I_1 I_0$	1	1	A←n4　n4=0−F	縦積みLAI
	LDI n4	0 0 1 1 0 0 1 0	1 1 1 0 $I_3 I_2 I_1 I_0$	2	2	D←n4　n4=0−F	
	LEI n4	0 0 1 1 0 0 0 0	1 1 1 0 $I_3 I_2 I_1 I_0$	2	2	E←n4　n4=0−F	
	LHI n4	0 0 1 1 0 0 1 1	1 1 1 0 $I_3 I_2 I_1 I_0$	2	2	H←n4　n4=0−F	
	LLI n4	0 0 1 1 0 0 0 1	1 1 1 0 $I_3 I_2 I_1 I_0$	2	2	L←n4　n4=0−F	
	LAM pr	0 1 0 R_4	0 0 $R_1 R_0$	1	½	A←(pr), pr=DE,DL,HL,HL+,HL−	
	LADR mem	0 0 1 1 0 $D_6 D_5 D_4$	1 0 0 0 $D_3 D_2 D_1 D_0$	2	2	A←(mem)　mem=0−7F	
	LDEI byte	0 1 0 0 $I_7 I_6 I_5 I_4$	1 1 1 1 $I_3 I_2 I_1 I_0$	2	2	DE←byte　byte=0−FF	
	LHLI byte	0 1 0 0 $I_7 I_6 I_5 I_4$	1 1 1 0 $I_3 I_2 I_1 I_0$	2	2	HL←byte　byte=0−FF	縦積み LHLL LHLT
	LHLT taddr1	1 1 0 0	$P_3 P_2 P_1 P_0$	1	2	$H \leftarrow ROM(0001100P_3P_2P_1P_0)_H$ $L \leftarrow ROM(0001100P_3P_2P_1P_0)_L$ taddr1=0C0−0CF	縦積み LHLL LHLT
	LAMT	0 1 0 1	1 1 1 0	1	2	$A \leftarrow ROM(PC_{10-6},0,C,A)_H$ $(HL) \leftarrow ROM(PC_{10-6},0,C,A)_L$	
	ST	0 1 0 1	0 1 1 1	1	1	(HL)←A	
	TAD	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 0	2	2	D←A	
	TAE	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 0	2	2	E←A	
	TAH	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 0	2	2	H←A	
	TAL	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 0	2	2	L←A	
	TDA	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 1	2	2	A←D	
	TEA	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 1	2	2	A←E	
	THA	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 1	2	2	A←H	
	TLA	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 1	2	2	A←L	
	XAD	0 1 0 0	1 0 1 0	1	1	A↔D	
	XAE	0 1 0 0	1 0 1 1	1	1	A↔E	
	XAH	0 1 1 1	1 0 1 0	1	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	1	A↔L	
	XAM pr	0 1 0 R_4	0 1 $R_1 R_0$	1	½	A↔(pr) pr=DE,DL,HL,HL+,HL−	
	XADR mem	0 0 1 1 0 $D_6 D_5 D_4$	1 0 0 1 $D_3 D_2 D_1 D_0$	2	2	A↔(mem)　mem=0−7F	
	XHDR mem	0 0 1 1 0 $D_6 D_5 D_4$	1 0 1 0 $D_3 D_2 D_1 D_0$	2	2	H↔(mem)　mem=0−7F	
	XLDR mem	0 0 1 1 0 $D_6 D_5 D_4$	1 0 1 1 $D_3 D_2 D_1 D_0$	2	2	L↔(mem)　mem=0−7F	
演算命令	AISC n4	0 0 0 0	$I_3 I_2 I_1 I_0$	1	½	A←A+n4　n4=0−F skip if carry	carry
	ASC	0 1 1 1	1 1 0 1	1	½	A←A+(HL) skip if carry	carry
	ACSC	0 1 1 1	1 1 0 0	1	½	A,C←A+(HL)+C skip if carry	carry
	EXL	0 1 1 1	1 1 1 0	1	1	A←A⊻(HL)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	2	A←A∨(HL)	
アキュームレジスタ操作命令	CMA	0 1 1 1	1 1 1 1	1	1	$A \leftarrow \overline{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	2	C←A_0, A_3←C $A_{n-1} \leftarrow A_n$　(n=1−3)	
キャリーフラグ操作命令	RC	0 1 1 1	1 0 0 0	1	1	C←0	
	SC	0 1 1 1	1 0 0 1	1	1	C←1	
増減命令	IES	0 1 0 0	1 0 0 1	1	½	E←E+1 skip if E=0	E=0
	ILS	0 1 0 1	1 0 0 1	1	½	L←L+1 skip if L=0	L=0
	IDRS mem	0 0 1 1 0 $D_6 D_5 D_4$	1 1 0 1 $D_3 D_2 D_1 D_0$	2	⅔	(mem)←(mem)+1 skip if(mem)=0　mem=0−7F	mem=0
	DES	0 1 0 0	1 0 0 0	1	½	E←E−1 skip if E=F	E=F
	DLS	0 1 0 1	1 0 0 0	1	½	L←L−1 skip if L=F	L=F
	DDRS mem	0 0 1 1 0 $D_6 D_5 D_4$	1 1 0 0 $D_3 D_2 D_1 D_0$	2	⅔	(mem)←(mem)−1 skip if(mem)=F　mem=0−7F	mem=F
メモリビット操作命令	RMB bit	0 1 1 0	1 0 $B_1 B_0$	1	1	(HL)bit←0　bit=0−3	
	SMB bit	0 1 1 0	1 1 $B_1 B_0$	1	1	(HL)bit←1　bit=0−3	
ジャンプ命令	JMP addr	0 0 1 0 $P_7 P_6 P_5 P_4$	$0P_{10} P_9 P_8$ $P_3 P_2 P_1 P_0$	2	2	$PC_{10-0} \leftarrow P_{10-0}$ addr=0−7FF	
	JCP addr	1 0 $P_5 P_4$	$P_3 P_2 P_1 P_0$	1	1	PC[illegible] addr 0−7FF	

命令群	ニモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	オペレーション	スキップ条件
ジャンプ命令	JAM addr2	0 0 1 1 0 0 0 1	1 1 1 1 0 $P_2 P_1 P_0$	2	2	$PC_{10-8} \leftarrow P_{2-0}$, $PC_{7-4} \leftarrow A_{3-0}$ $PC_{3-0} \leftarrow$(HL) addr2=0−7	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 $P_7 P_6 P_5 P_4$	$0P_{10} P_9 P_8$ $P_3 P_2 P_1 P_0$	2	2	(SP−1)(SP−2)(SP−4)←PC (SP−3)←PSW, PC←P_{10-0} SP←SP−4　addr=0−7FF	
	CALT taddr2	1 1 $P_5 P_4$	$P_3 P_2 P_1 P_0$	1	2	SP−1・SP−2・SP−4←PC SP−3←PSW PC←ROM(0001100$P_5P_4P_3P_2P_1P_0$) SP←SP−4　taddr2=0D0−0FF	
	RT	0 1 0 1	0 0 1 1	1	2	PC←(SP)(SP+2)(PS+3) SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	1	3	PC←(SP)(SP+2)(SP+3) SP←SP+4 then skip unconditionally	無条件
	RTPSW	0 1 0 0	0 0 1 1	1	2	PC←(SP)(SP+2)(SP+3) PSW←(SP+1) SP←SP+4	
	PSHDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 0	2	2	(SP−1)←D (SP−2)←E SP←SP−2	
	PSHHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 0	2	2	(SP−1)←H (SP−2)←L SP←SP−2	
	POPDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 1	2	2	E←(SP) D←(SP+1) SP←SP+2	
	POPHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 1	2	2	L←(SP) H←(SP+1) SP←SP+2	
	TAMSP	0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 1	2	2	$SP_{7-4} \leftarrow A$ $SP_{3-1} \leftarrow (HL)_{3-1}$, $SP_0 \leftarrow 0$	
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	2	$A \leftarrow SP_{7-4}$ $(HL)_{3-1} \leftarrow SP_{3-1}$, $(HL)_0 \leftarrow 0$	
スキップ命令	SKC	0 1 0 1	1 0 1 0	1	½	skip if C=1	C=1
	SKMBT bit	0 1 1 0	0 1 $B_1 B_0$	1	½	skip if(HL)bit=1 bit=0−3	(HL)bit=1
	SKABT bit	0 1 1 1	0 1 $B_1 B_0$	1	½	skip if Abit=1 bit=0−3	Abit=1
	SKMBF bit	0 1 1 0	0 0 $B_1 B_0$	1	½	skip if(HL)bit=0 bit=0−3	(HL)bit=0
	SKAEM	0 1 0 1	1 1 1 1	1	½	skip if A=(HL)	A=(HL)
	SKAEI n4	0 0 1 1 0 1 1 0	1 1 1 1 $I_3 I_2 I_1 I_0$	2	⅔	skip if A=n4　n4=0−F	A=n4
	SKDEI n4	0 0 1 1 0 1 1 0	1 1 1 0 $I_3 I_2 I_1 I_0$	2	⅔	skip if D=n4　n4=0−F	D=n4
	SKEEI n4	0 0 1 1 0 1 0 0	1 1 1 0 $I_3 I_2 I_1 I_0$	2	⅔	skip if E=n4　n4=0−F	E=n4
	SKHEI n4	0 0 1 1 0 1 1 1	1 1 1 0 $I_3 I_2 I_1 I_0$	2	⅔	skip if H=n4　n4=0−F	H=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 0 $I_3 I_2 I_1 I_0$	2	⅔	skip if L=n4　n4=0−F	L=n4
御SIO命令制	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	2	$SIO_{7-4} \leftarrow A$ $SIO_{3-0} \leftarrow$(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	2	$A \leftarrow SIO_{7-4}$ $(HL) \leftarrow SIO_{3-0}$	
	SIO	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 1	2	2	3Bit CNT←0 INT 0/S RQF←0	
御タイマ命令制	TAMMOD	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 1	2	2	$MOD_{7-4} \leftarrow A$ $MOD_{3-0} \leftarrow$(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	2	Start Timer	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	2	$A \leftarrow CT_{7-4}$ $(HL) \leftarrow CT_{3-0}$	
割込制御命令	EI n3	0 0 1 1 1 0 0 1	1 1 1 1 0 $I_2 I_1 I_0$	2	2	Enable Interrupt n3=0−7	
	DI n3	0 0 1 1 1 0 0 0	1 1 1 1 0 $I_2 I_1 I_0$	2	2	Disable Interrupt n3=0−7	
	SKI n3	0 0 1 1 0 1 0 0	1 1 1 1 0 $I_2 I_1 I_0$	2	⅔	Skip if Interrupt n3=0−7	INT RQF=1
入出力命令	IPL	0 1 1 1	0 0 0 0	1	1	A←PORT(L)	
	IP addr3	0 0 1 1 1 1 0 0	1 1 1 1 0 $D_2 D_1 D_0$	2	2	A←PORT(addr3)　addr3=0, 1, 4−6	
	IP1	0 1 1 1	0 0 0 1	1	1	A←PORT1	
	IP54	0 0 1 1 0 0 1 1	1 1 1 1 1 0 0 0	2	2	A←RORT5 (HL)←PORT4	
	OPL	0 1 1 1	0 0 1 0	1	1	PORT/MODE REG(L)←A	
	OP addr1	0 0 1 1 1 1 1 0	1 1 1 1 $D_2 D_1 D_0$	2	2	PORT/MODE REG(addr1)←A addr1=3−6, B, C, E, F	
	OP3	0 1 1 1	0 0 1 1	1	1	PORT3←A	
	OP54	0 0 1 1 0 0 1 1	1 1 1 1 1 1 0 0	2	2	PORT5←A PORT4←(HL)	
	ANP addr4, n4	0 1 0 0 $I_3 I_2 I_1 I_0$	1 1 0 0 0 $D_2 D_1 D_0$	2	2	PORT(addr4)←PORT(addr4)∧n4 addr4=3.4,5,6　n4=0−F	
	ORP addr4, n4	0 1 0 0 $I_3 I_2 I_1 I_0$	1 1 0 1 0 $D_2 D_1 D_0$	2	2	PORT(addr4)←PORT(addr4)∨n4 addr4=3,4,5,6　n4=0−F	
御CPU命令制	HALT	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 0	2	2	set HALT mode	
	STOP	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 1	2	2	set STOP mode	
	NOP	0 0 0 0	0 0 0 0	1	1	No Operation	

μPD7503　インストラクション活用表

（適用品種：μPD7503, 7503A）

命令群	ニーモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	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	$I_3I_2I_1I_0$	1	1	A←n4　n4=0-F	縦積みLAI
	LDI n4	0 0 1 1 0 0 1 0	1 1 1 0 $I_3I_2I_1I_0$	2	2	D←n4　n4=0-F	
	LEI n4	0 0 1 1 0 0 0 0	1 1 1 0 $I_3I_2I_1I_0$	2	2	E←n4　n4=0-F	
	LHI n4	0 0 1 1 0 0 1 1	1 1 1 0 $I_3I_2I_1I_0$	2	2	H←n4　n4=0-F	
	LLI n4	0 0 1 1 0 0 0 1	1 1 1 0 $I_3I_2I_1I_0$	2	2	L←n4　n4=0-F	
	LAM pr	0 1 0 R_4	0 0 R_1R_0	1	½	A←(pr), pr=DL,DE,HL−,HL+,HL	L=F(HL−) L=0(HL+)
	LADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 0 $D_3D_2D_1D_0$	2	2	A←(mem)　mem=0-DF	
	LDEI byte	0 1 0 0 $I_7I_6I_5I_4$	1 1 1 1 $I_3I_2I_1I_0$	2	2	DE←byte　byte=0-FF	
	LHLI byte	0 1 0 0 $I_7I_6I_5I_4$	1 1 1 0 $I_3I_2I_1I_0$	2	2	HL←byte　byte=0-FF	縦積み LHLI LHLT
	LHLT taddr1	1 1 0 0	$P_3P_2P_1P_0$	1	2	H←ROM(00001100$P_3P_2P_1P_0$)H L←ROM(00001100$P_3P_2P_1P_0$)L taddr1=0C0−0CF	縦積み LHLI LHLT
	LAMTL	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 0	2	3	A←ROM(PC_{11-4},A,$(HL)_{3-0}$)H (HL)←ROM(PC_{11-4},A,$(HL)_{3-0}$)L	
	ST	0 1 0 1	0 1 1 1	1	1	(HL)←A	
	TAD	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 0	2	2	D←A	
	TAE	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 0	2	2	E←A	
	TAH	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 0	2	2	H←A	
	TAL	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 0	2	2	L←A	
	TDA	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 1	2	2	A←D	
	TEA	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 1	2	2	A←E	
	THA	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 1	2	2	A←H	
	TLA	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 1	2	2	A←L	
	XAD	0 1 0 0	1 0 1 0	1	1	A↔D	
	XAE	0 1 0 0	1 0 1 1	1	1	A↔E	
	XAH	0 1 1 1	1 0 1 0	1	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	1	A↔L	
	XAM pr	0 1 0 R_4	0 1 R_1R_0	1	½	A↔(pr)　pr=DL,DE,HL−,HL+,HL	L=F(HL−) L=0(HL+)
	XADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 1 $D_3D_2D_1D_0$	2	2	A↔(mem)　mem=0-DF	
	XHDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 0 $D_3D_2D_1D_0$	2	2	H↔(mem)　mem=0-DF	
	XLDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 1 $D_3D_2D_1D_0$	2	2	L↔(mem)　mem=0-DF	
演算命令	AISC n4	0 0 0 0	$I_3I_2I_1I_0$	1	½	A←A+n4　n4≠0-F	carry
	ASC	0 1 1 1	1 1 0 1	1	½	A←A+(HL)	carry
	ACSC	0 1 1 1	1 1 0 0	1	½	A,C←A+(HL)+C	carry
	EXL	0 1 1 1	1 1 1 0	1	1	A←A⊻(HL)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	2	A←A∨(HL)	
アキュームレータ操作命令	CMA	0 1 1 1	1 1 1 1	1	1	A←$\overline{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	2	C←A_0, A_3←C A_{n-1}←An　(n=1-3)	
キャリーフラグ操作命令	RC	0 1 1 1	1 0 0 0	1	1	C←0	
	SC	0 1 1 1	1 0 0 1	1	1	C←1	
増減命令	IES	0 1 0 0	1 0 0 1	1	½	E←E+1	E=0
	ILS	0 1 0 1	1 0 0 1	1	½	L←L+1	L=0
	IDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 1 $D_3D_2D_1D_0$	2	⅔	(mem)←(mem)+1 mem=0-DF	(mem)=0
	DES	0 1 0 0	1 0 0 0	1	½	E←E−1	E=F
	DLS	0 1 0 1	1 0 0 0	1	½	L←L−1	L=F
	DDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 0 $D_3D_2D_1D_0$	2	⅔	(mem)←(mem)−1 mem=0-DF	(mem)=F
メモリ操作命令	RMB bit	0 1 1 0	1 0 B_1B_0	1	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	1 1 B_1B_0	1	1	(HL)bit←1　bit=0-3	
ジャンプ命令	JMP addr	0 0 1 0 $P_7P_6P_5P_4$	$P_{11}P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	2	PC11-0←P11-0 addr=0-FFF	
	JCP addr	1 0 P_5P_4	$P_3P_2P_1P_0$	1	1	PC_{5-0}←P_{5-0}　addr=0-FFF	

命令群	ニーモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	オペレーション	スキップ条件
ジャンプ命令	JAM addr2	0 0 1 1 0 0 0 1	1 1 1 1 $P_3P_2P_1P_0$	2	2	PC_{11-8}←P_{11-8}, PC_{7-4}←A_{3-0} PC_{3-0}←(HL) addr2=0-FFH	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 $P_7P_6P_5P_4$	$0P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	2	(SP−1)(SP−2)(SP−4)←PC (SP−3)←PSW, PC←P_{10-0} SP←SP−4　caddr=0-7FF	
	CALT taddr2	1 1 P_5P_4	$P_3P_2P_1P_0$	1	2	(SP−1)(SP−2)(SP−4)←PC (SP−3)←PSW PC←ROM(0000111$P_5P_4$$P_3P_2P_1P_0$) SP←SP−4, taddr2=080-0FF	
	RT	0 1 0 1	0 0 1 1	1	2	PC←(SP)(SP+2)(SP+3) SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	1	3	PC←(SP)(SP+2)(SP+3) SP←SP+4 then skip unconditionally	無条件
	RTPSW	0 1 0 0	0 0 1 1	1	2	PC←(SP)(SP+2)(SP+3) PSW←(SP+1) SP←SP+4	
	PSHDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 0	2	2	(SP−1)←D (SP−2)←E SP←SP−2	
	PSHHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 0	2	2	(SP−1)←H (SP−2)←L SP←SP−2	
	POPDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 1	2	2	E←(SP) D←(SP+1) SP←SP+2	
	POPHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 1	2	2	L←(SP) H←(SP+1) SP←SP+2	
	TAMSP	0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 1	2	2	SP_{7-4}←A SP_{3-1}←$(HL)_{3-1}$, SP_0←0	
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	2	A←SP_{7-4} $(HL)_{3-1}$←SP_{3-1}, $(HL)_0$←0	
スキップ命令	SKC	0 1 0 1	1 0 1 0	1	½	skip if C=1	C=1
	SKMBT bit	0 1 1 0	0 1 B_1B_0	1	½	skip if(HL)bit=1　bit=0-3	(HL)bit=1
	SKABT bit	0 1 1 1	0 1 B_1B_0	1	½	skip if Abit=1　bit=0-3	Abit=1
	SKMBF bit	0 1 1 0	0 0 B_1B_0	1	½	skip if(HL)bit=0　bit=0-3	(HL)bit=0
	SKAEM	0 1 0 1	1 1 1 1	1	½	skip if A=(HL)	A=(HL)
	SKAEI n4	0 0 1 1 0 1 1 0	1 1 1 1 $I_3I_2I_1I_0$	2	⅔	skip if A=n4　n4=0-F	A=n4
	SKDEI n4	0 0 1 1 0 1 1 0	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if D=n4　n4=0-F	D=n4
	SKEEI n4	0 0 1 1 0 1 0 0	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if E=n4　n4=0-F	E=n4
	SKHEI n4	0 0 1 1 0 1 1 1	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if H=n4　n4=0-F	H=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if L=n4　n4=0-F	L=n4
制御SIO命令	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	2	SIO_{7-4}←A SIO_{3-0}←(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	2	A←SIO_{7-4} (HL)←SIO_{3-0}	
	SIO	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 1	2	2	3-BIT CNT←0 INT0/S RQF←0	
制御タイマ命令	TAMMOD	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 1	2	2	MOD_{7-4}←A MOD_{3-0}←(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	2	Start Timer	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	2	A←CT_{7-4} (HL)←CT_{3-0}	
割込制御命令	EI n3	0 0 1 1 1 0 0 1	1 1 1 1 0 $I_2I_1I_0$	2	2	IME←1　(n3=0) IE2-0←IE2-0∨n3(n3=1-7)	
	DI n3	0 0 1 1 1 0 0 0	1 1 1 1 0 $I_2I_1I_0$	2	2	IME←0　(n3=0) IE2-0←IE2-0∧n3 (n3=1-7)	
	SKI n3	0 0 1 1 0 1 0 0	1 1 1 1 0 $I_2I_1I_0$	2	⅔	Skip if Interrupt n3=1-7	INT RQF=1
入出力命令	IPL	0 1 1 1	0 0 0 0	1	1	A←PORT(L)	
	IP addr3	0 0 1 1 1 1 0 0	1 1 1 1 0 $D_2D_1D_0$	2	2	A←PORT(addr3)　addr3=0,1,4-6	
	IP1	0 1 1 1	0 0 0 1	1	1	A←PORT1	
	IP54	0 0 1 1 0 0 1 1	1 1 1 1 1 0 0 0	2	2	A←PORT5 (HL)←PORT4	
	OPL	0 1 1 1	0 0 1 0	1	1	PORT/MODE REG(L)←A	
	OP addr1	0 0 1 1 1 1 1 0	1 1 1 1 $D_2D_1D_0$	2	2	PORT/MODE REG(addr1)←A addr1=3-6, B, C, E, F	
	OP3	0 1 1 1	0 0 1 1	1	1	PORT3←A	
	OP54	0 0 1 1 0 0 1 1	1 1 1 1 1 1 0 0	2	2	PORT5←A PORT4←(HL)	
	ANP addr4, n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 0 0 $D_2D_1D_0$	2	2	PORT(addr4)←PORT(addr4)∧n4 addr4=3-6, n4=0-F	
	ORP addr4, n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 1 0 $D_2D_1D_0$	2	2	PORT(addr4)←PORT(addr4)∨n4 addr4=3-6, n4=0-F	
制御CPU命令	HALT	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 0	2	2	Set HALT mode	
	STOP	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 1	2	2	Set STOP mode	
	NOP	0 0 0 0	0 0 0 0	1	1	No Operation	

保守／廃止

μPD7506　インストラクション活用表

命令群	ニーモニック	命令コード		サイクル・マシン	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	$I_3 I_2 I_1 I_0$	1	A←n4　n4=0-FH	縦積LAI
	LHI n3	0 0 1 0	$1 I_2 I_1 I_0$	1	H←$0I_2I_1I_0$　n3=0-7	
	LAM pr	0 1 0 1	$0 0 R_1 R_0$	½	A←(pr)　pr=HL−,HL+,HL	L=FH(HL−) L=0(HL+)
	LADR mem	0 0 1 1 $0 0 D_5 D_4$	1 0 0 0 $D_3 D_2 D_1 D_0$	2	A←(mem)　mem=0-3FH	
	LHLI n5	1 1 0 I_4	$I_3 I_2 I_1 I_0$	1	HL←$000I_4I_3I_2I_1I_0$　n5=0-1FH	縦積LHLI
	LAMT	0 1 0 1	1 1 1 0	2	A←ROM(PC_{3-6}, 0, C, A_{3-0})$_H$ (HL)←ROM(PC_{9-6}, 0, C, A_{3-0})$_L$	
	ST	0 1 0 1	0 1 1 1	1	(HL)←A	
	STII n4	0 1 0 0	$I_3 I_2 I_1 I_0$	1	(HL)←n4, L←L+1　n4=0-FH	
	XAH	0 1 1 1	1 0 1 0	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	A↔L	
	XAM pr	0 1 0 1	$0 1 R_1 R_0$	½	A↔(pr)　pr=HL−,HL+,HL	L=FH(HL−) L=0(HL+)
	XADR mem	0 0 1 1 $0 0 D_5 D_4$	1 0 0 1 $D_3 D_2 D_1 D_0$	2	A↔(mem)　mem=0-3FH	
	XHDR mem	0 0 1 1 $0 0 D_5 D_4$	1 0 1 0 $D_3 D_2 D_1 D_0$	2	H↔(mem)　mem=0-3FH	
	XLDR mem	0 0 1 1 $0 0 D_5 D_4$	1 0 1 1 $D_3 D_2 D_1 D_0$	2	L↔(mem)　mem=0-3FH	
演算命令	AISC n4	0 0 0 0	$I_3 I_2 I_1 I_0$	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)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	A←A∨(HL)	
アキュームレータ・キャリ・フラグ操作命令	CMA	0 1 1 1	1 1 1 1	1	A←$\bar{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	C←A_0, A_3←C A_n←A_{n+1}　(n=1-3)	
	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 $0 0 D_5 D_4$	1 1 0 1 $D_3 D_2 D_1 D_0$	⅔	(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 $0 0 D_5 D_4$	1 1 0 0 $D_3 D_2 D_1 D_0$	⅔	(mem)←(mem)−1 mem=0-3FH	(mem)=FH
メモリビット操作命令	RMB bit	0 1 1 0	$1 0 B_1 B_0$	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	$1 1 B_1 B_0$	1	(HL)*bit*←1　bit=0-3	
ジャンプ命令	JMP addr	0 0 1 0 $P_7 P_6 P_5 P_4$	0 0 $P_9 P_8$ $P_3 P_2 P_1 P_0$	2	PC_{9-0}←P_{9-0}	
	JCP addr	1 0 $P_5 P_4$	$P_3 P_2 P_1 P_0$	1	PC_{5-0}←P_{5-0}	
	JAM addr3	0 0 1 1 0 0 0 1	1 1 1 1 0 0 $P_1 P_0$	2	PC_{9-8}←P_{1-0}, PC_{7-4}←A_{3-0} PC_{3-0}←(HL)$_{3-0}$	

命令群	ニーモニック	命令コード		サイクル・マシン	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 $P_7 P_6 P_5 P_4$	0 0 $P_9 P_8$ $P_3 P_2 P_1 P_0$	2	(SP-1)(SP-2)(SP-4)←PC_{9-0} (SP-3)←PSW, SP←SP-4 PC_{9-0}←P_{9-0}	
	CAL caddr1	1 1 1 P_4	$P_3 P_2 P_1 P_0$	2	(SP-1)(SP-2)(SP-4)←PC_{9-0} (SP-3)←PSW, SP←SP-4 PC_{9-0}←$01P_4P_3000P_2P_1P_0$	
	RT	0 1 0 1	0 0 1 1	2	PC_{9-0}←(SP)(SP+2)(SP+3) SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	3	PC_{9-0}←(SP)(SP+2)(SP+3) 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_{5-4}←A_{1-0} SP_{3-1}←(HL)$_{3-1}$, SP_0←0	
スキップ命令	SKC	0 1 0 1	1 0 1 0	½	Skip if C=1	C=1
	SKABT bit	0 1 1 1	$0 1 B_1 B_0$	½	Skip if A_{bit}=1　bit=0-3	A_{bit}=1
	SKMBT bit	0 1 1 0	$0 1 B_1 B_0$	½	Skip if (HL)$_{bit}$=1　bit=0-3	(HL)$_{bit}$=1
	SKMBF bit	0 1 1 0	$0 0 B_1 B_0$	½	Skip if (HL)$_{bit}$=0　bit=0-3	(HL)$_{bit}$=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	1 1 1 1 $I_3 I_2 I_1 I_0$	⅔	Skip if A=n4　n4=0-FH	A=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 1 $I_3 I_2 I_1 I_0$	⅔	Skip if L=n4　n4=0-FH	L=n4
	SKI n2	0 0 1 1 0 1 0 0	1 1 1 1 $0 0 I_1 I_0$	⅔	Skip if INT RQF=1 then reset INT RQF	INT RQF =1
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_H←A, MOD_L←(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	Start Timer CT_{7-0}←0, INTT RQF←0	
入出力命令	IPL	0 1 1 1	0 0 0 0	1	A←Port(L)	
	IP addr2	0 0 1 1 1 1 0 0	1 1 1 1 $0 D_2 D_1 D_0$	2	A←Port(addr2) addr2=0,1,4-6	
	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 addr1	0 0 1 1 1 1 1 0	1 1 1 1 $D_3 D_2 D_1 D_0$	2	Port/Mode reg.(addr1)←A addr1=1,2,4-6,CH,EH,FH	
	OP54	0 0 1 1 0 0 1 1	1 1 1 1 1 1 0 0	2	Port5←A, Port4←(HL)	
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 Operation	

μPD7507　インストラクション活用表

（適用品種：μPD7507，7507B，7507H，7507S）

命令群	ニモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	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	$I_3I_2I_1I_0$	1	1	A←n4　n4=0-FH	縦積みLAI
	LDI n4	0 0 1 1 0 0 1 0	1 1 1 0 $I_3I_2I_1I_0$	2	2	D←n4　n4=0-FH	
	LEI n4	0 0 1 1 0 0 0 0	1 1 1 0 $I_3I_2I_1I_0$	2	2	E←n4　n4=0-FH	
	LHI n4	0 0 1 1 0 0 1 1	1 1 1 0 $I_3I_2I_1I_0$	2	2	H←n4　n4=0-FH	
	LLI n4	0 0 1 1 0 0 0 1	1 1 1 0 $I_3I_2I_1I_0$	2	2	L←n4　n4=0-FH	
	LAM pr	0 1 0 R_4	0 0 R_1R_0	1	½	A←(pr) pr=DL,DE,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	LADR mem	0 0 1 1 0 $D_6D_5D_4$	1 0 0 0 $D_3D_2D_1D_0$	2	2	A←(mem)　mem=0-7FH	
	LDEI byte	0 1 0 0 $I_7I_6I_5I_4$	1 1 1 1 $I_3I_2I_1I_0$	2	2	DE←byte　byte=0-FFH	
	LHLI byte	0 1 0 0 $I_7I_6I_5I_4$	1 1 1 0 $I_3I_2I_1I_0$	2	2	HL←byte　byte=0-FFH	縦積み LHLI LHLT
	LHLT taddr1	1 1 0 0	$P_3P_2P_1P_0$	1	2	H←ROM(0001100$P_3P_2P_1P_0$)H L←ROM(0001100$P_3P_2P_1P_0$)L taddr1=0C0H−0CFH	縦積み LHLI LHLT
	LAMTL	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 0	2	3	A←ROM(PC_{10-8},A,(HL))H (HL)←ROM(PC_{10-8},A,(HL))L	
	ST	0 1 0 1	0 1 1 1	1	1	(HL)←A	
	TAD	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 0	2	2	D←A	
	TAE	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 0	2	2	E←A	
	TAH	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 0	2	2	H←A	
	TAL	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 0	2	2	L←A	
	TDA	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 1	2	2	A←D	
	TEA	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 1	2	2	A←E	
	THA	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 1	2	2	A←H	
	TLA	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 1	2	2	A←L	
	XAD	0 1 0 0	1 0 1 0	1	1	A↔D	
	XAE	0 1 0 0	1 0 1 1	1	1	A↔E	
	XAH	0 1 1 1	1 0 1 0	1	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	1	A↔L	
	XAM pr	0 1 0 R_4	0 1 R_1R_0	1	½	A↔(pr) pr=DL,DE,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	XADR mem	0 0 1 1 0 $D_6D_5D_4$	1 0 0 1 $D_3D_2D_1D_0$	2	2	A↔(mem)　mem=0-7FH	
	XHDR mem	0 0 1 1 0 $D_6D_5D_4$	1 0 1 0 $D_3D_2D_1D_0$	2	2	H↔(mem)　mem=0-7FH	
	XLDR mem	0 0 1 1 0 $D_6D_5D_4$	1 0 1 1 $D_3D_2D_1D_0$	2	2	L↔(mem)　mem=0-7FH	
演算命令	AISC n4	0 0 0 0	$I_3I_2I_1I_0$	1	½	A←A+n4　n4=0-FH	carry
	ASC	0 1 1 1	1 1 0 1	1	½	A←A+(HL)	carry
	ACSC	0 1 1 1	1 1 0 0	1	½	A,C←A+(HL)+C	carry
	EXL	0 1 1 1	1 1 1 0	1	1	A←A⊻(HL)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	2	A←A∨(HL)	
アキュームレジスタ操作命令	CMA	0 1 1 1	1 1 1 1	1	1	A←$\overline{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	2	C←A_0, A_3←C A_{n-1}←An　(n=1-3)	
キャリーフラグ操作命令	RC	0 1 1 1	1 0 0 0	1	1	C←0	
	SC	0 1 1 1	1 0 0 1	1	1	C←1	
増減命令	IES	0 1 0 0	1 0 0 1	1	½	E←E+1	E=0
	ILS	0 1 0 1	1 0 0 1	1	½	L←L+1	L=0
	IDRS mem	0 0 1 1 0 $D_6D_5D_4$	1 1 0 1 $D_3D_2D_1D_0$	2	⅔	(mem)←(mem)+1 mem=0-7FH	(mem)=0
	DES	0 1 0 0	1 0 0 0	1	½	E←E−1	E=FH
	DLS	0 1 0 1	1 0 0 0	1	½	L←L−1	L=FH
	DDRS mem	0 0 1 1 0 $D_6D_5D_4$	1 1 0 0 $D_3D_2D_1D_0$	2	⅔	(mem)←(mem)−1 mem=0-7FH	(mem)=FH
メモリビット操作命令	RMB bit	0 1 1 0	1 0 B_1B_0	1	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	1 1 B_1B_0	1	1	(HL)bit←1　bit=0-3	
ジャンプ命令	JMP addr	0 0 1 0 $P_7P_6P_5P_4$	0$P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	2	PC_{10-0}←P_{10-0} addr=0-7FFH	
	JCP addr	1 0 P_5P_4	$P_3P_2P_1P_0$	1	1	PC_{5-0}←P_{5-0}　addr=0-7FFH	

命令群	ニモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	オペレーション	スキップ条件
ジャンプ命令	JAM addr2	0 0 1 1 0 0 0 1	1 1 1 1 0 $P_2P_1P_0$	2	2	PC_{10-8}←P_{2-0}, PC_{7-4}←A_{3-0} PC_{3-0}←(HL) addr2=0-7	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 $P_7P_6P_5P_4$	0$P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	2	(SP−1)(SP−2)(SP−4)←PC (SP−3)←PSW, PC←P_{10-0} SP←SP−4　caddr=0-7FFH	
	CALT taddr2	1 1 P_5P_4	$P_3P_2P_1P_0$	1	2	(SP−1)(SP−2)(SP−4)←PC (SP−3)←PSW $PC_{7,6}$←ROM(00011$P_5P_4P_3P_2P_1P_0$) $PC_{10,5,4}$←0 SP←SP−4　taddr2=0D0H-0FFH	
	RT	0 1 0 1	0 0 1 1	1	2	PC←(SP)(SP+2)(SP+3) SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	1	3	PC←(SP)(SP+2)(SP+3) SP←SP+4 then skip unconditionally	無条件
	RTPSW	0 1 0 0	0 0 1 1	1	2	PC←(SP)(SP+2)(SP+3) PSW←(SP+1) SP←SP+4	
	PSHDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 0	2	2	(SP−1)←D (SP−2)←E SP←SP−2	
	PSHHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 0	2	2	(SP−1)←H (SP−2)←L SP←SP−2	
	POPDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 1	2	2	E←(SP) D←(SP+1) SP←SP+2	
	POPHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 1	2	2	L←(SP) H←(SP+1) SP←SP+2	
	TAMSP	0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 1	2	2	SP_{7-4}←A SP_{3-1}←$(HL)_{3-1}$, SP_0←0	
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	2	A←SP_{7-4} $(HL)_{3-1}$←SP_{3-1}, $(HL)_0$←0	
スキップ命令	SKC	0 1 0 1	1 0 1 0	1	½	skip if C=1	C=1
	SKMBT bit	0 1 1 0	0 1 B_1B_0	1	½	skip if(HL)bit=1　bit=0-3	(HL)bit=1
	SKABT bit	0 1 1 1	0 1 B_1B_0	1	½	skip if Abit=1　bit=0-3	Abit=1
	SKMBF bit	0 1 1 0	0 0 B_1B_0	1	½	skip if(HL)bit=0　bit=0-3	(HL)bit=0
	SKAEM	0 1 0 1	1 1 1 1	1	½	skip if A=(HL)	A=(HL)
	SKAEI n4	0 0 1 1 0 1 1 0	1 1 1 1 $I_3I_2I_1I_0$	2	⅔	skip if A=n4　n4=0-FH	A=n4
	SKDEI n4	0 0 1 1 0 1 1 0	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if D=n4　n4=0-FH	D=n4
	SKEEI n4	0 0 1 1 0 1 0 0	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if E=n4　n4=0-FH	E=n4
	SKHEI n4	0 0 1 1 0 1 1 1	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if H=n4　n4=0-FH	H=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if L=n4　n4=0-FH	L=n4
御SIO命令制	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	2	SIO_{7-4}←A SIO_{3-0}←(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	2	A←SIO_{7-4} (HL)←SIO_{3-0}	
	SIO	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 1	2	2	3Bit CNT←0 INT0/S RQF←0	
御タイマ命令制	TAMMOD	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 1	2	2	MOD_{7-4}←A MOD_{3-0}←(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	2	Start Timer	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	2	A←CT_{7-4} (HL)←CT_{3-0}	
割込制御命令	EI n3	0 0 1 1 1 0 0 1	1 1 1 1 0 $I_2I_1I_0$	2	2	IME←1　(n3=0) IE2-0←IE2-0∨n3(n3=1-7)	
	DI n3	0 0 1 1 1 0 0 0	1 1 1 1 0 $I_2I_1I_0$	2	2	IME←0　(n3=0) IE2-0←IE2-0∧$\overline{n3}$(n3=1-7)	
	SKI n3	0 0 1 1 0 1 0 0	1 1 1 1 0 $I_2I_1I_0$	2	⅔	Skip if Interrupt n3=1-7	INT RQF=1
入出力命令	IPL	0 1 1 1	0 0 0 0	1	1	A←PORT(L)	
	IP addr3	0 0 1 1 1 1 0 0	1 1 1 1 0 $D_2D_1D_0$	2	2	A←PORT(addr3)　addr3=0,1,4-7	
	IP1	0 1 1 1	0 0 0 1	1	1	A←PORT1	
	IP54	0 0 1 1 0 0 1 1	1 1 1 1 1 0 0 0	2	2	A←RORT5 (HL)←PORT4	
	OPL	0 1 1 1	0 0 1 0	1	1	PORT/MODE REG(L)←A	
	OP addr1	0 0 1 1 1 1 1 0	1 1 1 1 $D_3D_2D_1D_0$	2	2	PORT/MODE REG(addr1)←A addr1=1-7, CH, EH, FH	
	OP3	0 1 1 1	0 0 1 1	1	1	PORT3←A	
	OP54	0 0 1 1 0 0 1 1	1 1 1 1 1 1 0 0	2	2	PORT5←A PORT4←(HL)	
	ANP addr4, n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 0 0 $D_2D_1D_0$	2	2	PORT(addr4)←PORT(addr4)∧n4 addr4=2-7, n4=0-FH	
	ORP addr4, n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 1 0 $D_2D_1D_0$	2	2	PORT(addr4)←PORT(addr4)∨n4 addr4=2-7, n4=0-FH	
御CPU命令制	HALT	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 0	2	2	Set HALT mode	
	STOP	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 1	2	2	Set STOP mode	
	NOP	0 0 0 0	0 0 0 0	1	1	No Operation	

保守／廃止

μPD7508　インストラクション活用表

（適用品種：μPD7508，7508B，7508H）

命令群	ニモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マサイクル	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	$I_3I_2I_1I_0$	1	1	A←n4　n4=0-FH	縦積みLAI
	LDI n4	0 0 1 1 0 0 1 0	1 1 1 0 $I_3I_2I_1I_0$	2	2	D←n4　n4=0-FH	
	LEI n4	0 0 1 1 0 0 0 0	1 1 1 0 $I_3I_2I_1I_0$	2	2	E←n4　n4=0-FH	
	LHI n4	0 0 1 1 0 0 1 1	1 1 1 0 $I_3I_2I_1I_0$	2	2	H←n4　n4=0-FH	
	LLI n4	0 0 1 1 0 0 0 1	1 1 1 0 $I_3I_2I_1I_0$	2	2	L←n4　n4=0-FH	
	LAM pr	0 1 0 R_4	0 0 R_1R_0	1	½	A←(pr) pr=DL,DE,HL−,HL+,HL	L=FH(HL−) L=0　(HL+)
	LADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 0 $D_3D_2D_1D_0$	2	2	A←(mem)　mem=0-DFH	
	LDEI byte	0 1 0 0 $I_7I_6I_5I_4$	1 1 1 1 $I_3I_2I_1I_0$	2	2	DE←byte　byte=0-FFH	
	LHLI byte	0 1 0 0 $I_7I_6I_5I_4$	1 1 1 0 $I_3I_2I_1I_0$	2	2	HL←byte　byte=0-FFH	縦積み LHLI LHLT
	LHLT taddr1	1 1 0 0	$P_3P_2P_1P_0$	1	2	H←ROM(00001100$P_3P_2P_1P_0$)$_H$ L←ROM(00001100$P_3P_2P_1P_0$)$_L$ taddr1=0C0H−0CFH	縦積み LHLI LHLT
	LAMTL	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 0	2	3	A←ROM(PC$_{11\text{-}8}$,A,(HL)$_H$)$_H$ (HL)←ROM(PC$_{11\text{-}8}$,A,(HL)$_L$)$_L$	
	ST	0 1 0 1	0 1 1 1	1	1	(HL)←A	
	TAD	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 0	2	2	D←A	
	TAE	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 0	2	2	E←A	
	TAH	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 0	2	2	H←A	
	TAL	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 0	2	2	L←A	
	TDA	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 1	2	2	A←D	
	TEA	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 1	2	2	A←E	
	THA	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 1	2	2	A←H	
	TLA	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 1	2	2	A←L	
	XAD	0 1 0 0	1 0 1 0	1	1	A↔D	
	XAE	0 1 0 0	1 0 1 1	1	1	A↔E	
	XAH	0 1 1 1	1 0 1 0	1	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	1	A↔L	
	XAM pr	0 1 0 R_4	0 1 R_1R_0	1	½	A↔(pr) pr=DL,DE,HL−,HL+,HL	L=FH(HL−) L=0　(HL+)
	XADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 1 $D_3D_2D_1D_0$	2	2	A↔(mem)　mem=0-DFH	
	XHDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 0 $D_3D_2D_1D_0$	2	2	H↔(mem)　mem=0-DFH	
	XLDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 1 $D_3D_2D_1D_0$	2	2	L↔(mem)　mem=0-DFH	
演算命令	AISC n4	0 0 0 0	$I_3I_2I_1I_0$	1	½	A←A+n4　n4=0-FH	carry
	ASC	0 1 1 1	1 1 0 1	1	½	A←A+(HL)	carry
	ACSC	0 1 1 1	1 1 0 0	1	½	A, C←A+(HL)+C	carry
	EXL	0 1 1 1	1 1 1 0	1	1	A←A∨(HL)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	2	A←A∨(HL)	
アキュームレータ操作命令	CMA	0 1 1 1	1 1 1 1	1	1	A←$\overline{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	2	C←A_0, A_3←C A_{n-1}←A_n　(n=1-3)	
キャリーフラグ操作命令	RC	0 1 1 1	1 0 0 0	1	1	C←0	
	SC	0 1 1 1	1 0 0 1	1	1	C←1	
増減命令	IES	0 1 0 0	1 0 0 1	1	½	E←E+1	E=0
	ILS	0 1 0 1	1 0 0 1	1	½	L←L+1	L=0
	IDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 1 $D_3D_2D_1D_0$	2	⅔	(mem)←(mem)+1 mem=0-DFH	(mem)=0
	DES	0 1 0 0	1 0 0 0	1	½	E←E−1	E=FH
	DLS	0 1 0 1	1 0 0 0	1	½	L←L−1	L=FH
	DDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 0 $D_3D_2D_1D_0$	2	⅔	(mem)←(mem)−1 mem=0-DFH	(mem)=FH
メモリビット操作命令	RMB bit	0 1 1 0	1 0 B_1B_0	1	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	1 1 B_1B_0	1	1	(HL)bit←1　bit=0-3	
ジャンプ命令	JMP addr	0 0 1 0 $P_7P_6P_5P_4$	$P_3P_2P_1P_0$ $P_3P_2P_1P_0$	2	2	PC$_{11\text{-}0}$←P$_{11\text{-}0}$　addr=0-FFFH	
	JCP addr	1 0 P_5P_4	$P_3P_2P_1P_0$	1	1	PC$_{5\text{-}0}$←P$_{5\text{-}0}$　addr=0-FFFH	

命令群	ニモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マサイクル	オペレーション	スキップ条件
命令ネジャン	JAM addr2	0 0 1 1 0 0 0 1	1 1 1 1 $P_3P_2P_1P_0$	2	2	PC$_{11\text{-}8}$←P$_{3\text{-}0}$, PC$_{7\text{-}4}$←A$_{3\text{-}0}$ PC$_{3\text{-}0}$←(HL)　addr2=0-FH	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 $P_7P_6P_5P_4$	0$P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	2	(SP-1)(SP-2)(SP-4)←PC$_{11\text{-}0}$ (SP-3)←PSW, PC$_{11\text{-}0}$←0,P$_{10\text{-}0}$ SP←SP−4　caddr=0-7FFH	
	CALT taddr2	1 1 P$_5$P$_4$	P$_3$P$_2$P$_1$P$_0$	1	2	(SP−1)(SP−2)(SP−4)←PC$_{11\text{-}0}$ (SP−3)←PSW PC$_{11\text{-}8}$,PC$_{7\text{-}4}$←ROM(000011P$_5$P$_4$P$_3$P$_2$P$_1$P$_0$) PC$_{3\text{-}0}$←0 SP←SP−4　taddr2=0D0H-0FFH	
	RT	0 1 0 1	0 0 1 1	1	2	PC$_{11\text{-}0}$←(SP)(SP+2)(SP+3) SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	1	3	PC$_{11\text{-}0}$←(SP)(SP+2)(SP+3) SP←SP+4 then skip unconditionally	無条件
	RTPSW	0 1 0 0	0 0 1 1	1	2	PC$_{11\text{-}0}$←(SP)(SP+2)(SP+3) PSW←(SP+1) SP←SP+4	
	PSHDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 0	2	2	(SP−1)←D (SP−2)←E SP←SP−2	
	PSHHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 0	2	2	(SP−1)←H (SP−2)←L SP←SP−2	
	POPDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 1	2	2	E←(SP) D←(SP+1) SP←SP+2	
	POPHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 1	2	2	L←(SP) H←(SP+1) SP←SP+2	
	TAMSP	0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 1	2	2	SP$_{7\text{-}4}$←A SP$_{3\text{-}0}$←(HL)$_{3\text{-}0}$,SP$_0$←0	
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	2	A←SP$_{7\text{-}4}$ (HL)$_{3\text{-}1}$←SP$_{3\text{-}1}$, (HL)$_0$←0	
スキップ命令	SKC	0 1 0 1	1 0 1 0	1	½	skip if C=1	C=1
	SKMBT bit	0 1 1 0	0 1 B_1B_0	1	½	skip if(HL)bit=1　bit=0-3	(HL)bit=1
	SKABT bit	0 1 1 1	0 1 B_1B_0	1	½	skip if Abit=1　bit=0-3	Abit=1
	SKMBF bit	0 1 1 0	0 0 B_1B_0	1	½	skip if(HL)bit=0　bit=0-3	(HL)bit=0
	SKAEM	0 1 0 1	1 1 1 1	1	½	skip if A=(HL)	A=(HL)
	SKAEI n4	0 0 1 1 0 1 1 0	1 1 1 1 $I_3I_2I_1I_0$	2	⅔	skip if A=n4　n4=0-FH	A=n4
	SKDEI n4	0 0 1 1 0 1 1 0	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if D=n4　n4=0-FH	D=n4
	SKEEI n4	0 0 1 1 0 1 0 0	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if E=n4　n4=0-FH	E=n4
	SKHEI n4	0 0 1 1 0 1 1 1	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if H=n4　n4=0-FH	H=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 0 $I_3I_2I_1I_0$	2	⅔	skip if L=n4　n4=0-FH	L=n4
御SIO命令制	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	2	SIO$_{7\text{-}4}$←A SIO$_{3\text{-}0}$←(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	2	A←SIO$_{7\text{-}4}$ (HL)←SIO$_{3\text{-}0}$	
	SIO	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 1	2	2	3Bit CNT←0 INT0/S RQF←0	
御タイマ命令制	TAMMOD	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 1	2	2	MOD$_{7\text{-}4}$←A MOD$_{3\text{-}0}$←(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	2	Start Timer	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	2	A←CT$_{7\text{-}4}$ (HL)←CT$_{3\text{-}0}$	
割込制御命令	EI n3	0 0 1 1 1 0 0 1	1 1 1 1 0 $I_2I_1I_0$	2	2	IME←1　(n3=0) IE2-0←IE2-0∨n3(n3=1-7)	
	DI n3	0 0 1 1 1 0 0 0	1 1 1 1 0 $I_2I_1I_0$	2	2	IME←0　(n3=0) IE2-0←IE2-0∧$\overline{n3}$(n3=1-7)	
	SKI n3	0 0 1 1 0 1 0 0	1 1 1 1 0 $I_2I_1I_0$	2	⅔	Skip if INT RQF=1 then reset INT RQF n3=1-7	INT RQF=1
入出力命令	IPL	0 1 1 1	0 0 0 0	1	1	A←PORT(L)	
	IP addr3	0 0 1 1 1 1 0 0	1 1 1 1 0 $D_2D_1D_0$	2	2	A←PORT(addr3)　addr3=0,1,4-7	
	IP1	0 1 1 1	0 0 0 1	1	1	A←PORT1	
	IP54	0 0 1 1 0 0 1 1	1 1 1 1 1 0 0 0	2	2	A←RORT5 (HL)←PORT4	
	OPL	0 1 1 1	0 0 1 0	1	1	PORT/MODE REG(L)←A	
	OP addr1	0 0 1 1 1 1 1 0	1 1 1 1 $D_3D_2D_1D_0$	2	2	PORT/MODE REG(addr1)←A addr1=1-7, CH, EH, FH	
	OP3	0 1 1 1	0 0 1 1	1	1	PORT3←A	
	OP54	0 0 1 1 0 0 1 1	1 1 1 1 1 1 0 0	2	2	PORT5←A PORT4←(HL)	
	ANP addr4, n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 0 0 $D_2D_1D_0$	2	2	PORT(addr4)←PORT(addr4)∧n4 addr4=2-7, n4=0-FH	
	ORP addr4, n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 1 0 $D_2D_1D_0$	2	2	PORT(addr4)←PORT(addr4)∨n4 addr4=2-7, n4=0-FH	
御CPU命令制	HALT	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 0	2	2	Set HALT mode	
	STOP	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 1	2	2	Set STOP mode	
	NOP	0 0 0 0	0 0 0 0	1	1	No Operation	

μPD7514　インストラクション活用表

命令群	ニーモニック	命令コード D7D6D5D4	命令コード D3D2D1D0	バイト	マシンサイクル	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	I3 I2 I1 I0	1	1	A←n4　n4=0-FH	縦積みLAI
	LDI n4	0 0 1 1 0 0 1 0	1 1 1 0 I3 I2 I1 I0	2	2	D←n4　n4=0-FH	
	LEI n4	0 0 1 1 0 0 0 0	1 1 1 0 I3 I2 I1 I0	2	2	E←n4　n4=0-FH	
	LHI n4	0 0 1 1 0 0 1 1	1 1 1 0 I3 I2 I1 I0	2	2	H←n4　n4=0-FH	
	LLI n4	0 0 1 1 0 0 0 1	1 1 1 0 I3 I2 I1 I0	2	2	L←n4　n4=0-FH	
	LAM pr	0 1 0 R4	0 0 R1 R0	1	½	A←(pr) pr=DL,DE,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	LADR mem	0 0 1 1 D7D6D5D4	1 0 0 0 D3D2D1D0	2	2	A←(mem)　mem=0-FFH	
	LDEI byte	0 1 0 0 I7 I6 I5 I4	1 1 1 1 I3 I2 I1 I0	2	2	DE←byte　byte=0-FFH	
	LHLI byte	0 1 0 0 I7 I6 I5 I4	1 1 1 0 I3 I2 I1 I0	2	2	HL←byte　byte=0-FFH	縦積み LHLI LHLT
	LHLT taddr1	1 1 0 0	P3P2P1P0	1	2	H←ROM(00001100P3P2P1P0)H L←ROM(00001100P3P2P1P0)L taddr1=0C0H−0CFH	縦積み LHLI LHLT
	LAMTL	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 0	2	3	A←ROM(PC11-8,A,(HL))H (HL)←ROM(PC11-8,A,(HL))L	
	ST	0 1 0 1	0 1 1 1	1	1	(HL)←A	
	TAD	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 0	2	2	D←A	
	TAE	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 0	2	2	E←A	
	TAH	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 0	2	2	H←A	
	TAL	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 0	2	2	L←A	
	TDA	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 1	2	2	A←D	
	TEA	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 1	2	2	A←E	
	THA	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 1	2	2	A←H	
	TLA	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 1	2	2	A←L	
	XAD	0 1 0 0	1 0 1 0	1	1	A↔D	
	XAE	0 1 0 0	1 0 1 1	1	1	A↔E	
	XAH	0 1 1 1	1 0 1 0	1	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	1	A↔L	
	XAM pr	0 1 0 R4	0 1 R1 R0	1	½	A↔(pr) pr=DL,DE,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	XADR mem	0 0 1 1 D7D6D5D4	1 0 0 1 D3D2D1D0	2	2	A↔(mem)　mem=0-FFH	
	XHDR mem	0 0 1 1 D7D6D5D4	1 0 1 0 D3D2D1D0	2	2	H↔(mem)　mem=0-FFH	
	XLDR mem	0 0 1 1 D7D6D5D4	1 0 1 1 D3D2D1D0	2	2	L↔(mem)　mem=0-FFH	
演算命令	AISC n4	0 0 0 0	I3 I2 I1 I0	1	½	A←A+n4　n4=0-FH	carry
	ASC	0 1 1 1	1 1 0 1	1	½	A←A+(HL)	carry
	ACSC	0 1 1 1	1 1 0 0	1	½	A,C←A+(HL)+C	carry
	EXL	0 1 1 1	1 1 1 0	1	1	A←A⊻(HL)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	2	A←A∨(HL)	
アキュームレータ操作命令	CMA	0 1 1 1	1 1 1 1	1	1	A←$\overline{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	2	C←A0, A3←C An-1←An　(n=1-3)	
キャリーフラグ操作命令	RC	0 1 1 1	1 0 0 0	1	1	C←0	
	SC	0 1 1 1	1 0 0 1	1	1	C←1	
増減命令	IES	0 1 0 0	1 0 0 1	1	½	E←E+1	E=0
	ILS	0 1 0 1	1 0 0 1	1	½	L←L+1	L=0
	IDRS mem	0 0 1 1 D7D6D5D4	1 1 0 1 D3D2D1D0	2	⅔	(mem)←(mem)+1 mem=0-FFH	(mem)=0
	DES	0 1 0 0	1 0 0 0	1	½	E←E−1	E=FH
	DLS	0 1 0 1	1 0 0 0	1	½	L←L−1	L=FH
	DDRS mem	0 0 1 1 D7D6D5D4	1 1 0 0 D3D2D1D0	2	⅔	(mem)←(mem)−1 mem=0-FFH	(mem)=FH
メモリビット操作命令	RMB bit	0 1 1 0	1 0 B1 B0	1	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	1 1 B1 B0	1	1	(HL)bit←1　bit=0-3	
ジャンプ命令	JMP addr	0 0 1 0 P11P10P9P8	P7P6P5P4 P3P2P1P0	2	2	PC11-0←P11-0　addr=0-FFFH	
	JCP addr	1 0 P5 P4	P3P2P1P0	1	1	PC5-0←P5-0　addr=0-FFFH	

命令群	ニーモニック	命令コード D7D6D5D4	命令コード D3D2D1D0	バイト	マシンサイクル	オペレーション	スキップ条件
ジャンプ命令	JAM addr2	0 0 1 1 0 0 0 1	1 1 1 1 P3P2P1P0	2	2	PC11-8←P3-0, PC7-4←A3-0 PC3-0←(HL)　addr2=0-FH	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 P7P6P5P4	0P11P9P8 P3P2P1P0	2	2	(SP-1)(SP-2)(SP-4)←PC11-0 (SP-3)←PSW, PC11-0←0,P10-0 SP←SP−4　caddr=0-7FFH	
	CALT taddr2	1 1 P5 P4	P3P2P1P0	1	2	(SP−1)(SP−2)(SP−4)←PC11-0 (SP−3)←PSW PC11-10,3-1←0 SP←SP−4　taddr2=0D0H-0FFH	
	RT	0 1 0 1	0 0 1 1	1	2	PC11-0←(SP)(SP+2)(SP+3) SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	1	3	PC11-0←(SP)(SP+2)(SP+3) SP←SP+4 then skip unconditionally	無条件
	RTPSW	0 1 0 0	0 0 1 1	1	2	PC11-0←(SP)(SP+2)(SP+3) PSW←(SP+1) SP←SP+4	
	PSHDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 0	2	2	(SP−1)←D (SP−2)←E SP←SP−2	
	PSHHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 0	2	2	(SP−1)←H (SP−2)←L SP←SP−2	
	POPDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 1	2	2	E←(SP) D←(SP+1) SP←SP+2	
	POPHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 1	2	2	L←(SP) H←(SP+1) SP←SP+2	
	TAMSP	0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 1	2	2	SP7-4←A SP3-1←(HL)3-1, SP0←0	
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	2	A←SP7-4 (HL)3-1←SP3-1, (HL)0←0	
スキップ命令	SKC	0 1 0 1	1 0 1 0	1	½	skip if C=1	C=1
	SKMBT bit	0 1 1 0	0 1 B1 B0	1	½	skip if(HL)bit=1　bit=0-3	(HL)bit=1
	SKABT bit	0 1 1 1	0 1 B1 B0	1	½	skip if Abit=1　bit=0-3	Abit=1
	SKMBF bit	0 1 1 0	0 0 B1 B0	1	½	skip if(HL)bit=0　bit=0-3	(HL)bit=0
	SKAEM	0 1 0 1	1 1 1 1	1	½	skip if A=(HL)	A=(HL)
	SKAEI n4	0 0 1 1 0 1 1 0	1 1 1 1 I3 I2 I1 I0	2	⅔	skip if A=n4　n4=0-FH	A=n4
	SKDEI n4	0 0 1 1 0 1 1 0	1 1 1 0 I3 I2 I1 I0	2	⅔	skip if D=n4　n4=0-FH	D=n4
	SKEEI n4	0 0 1 1 0 1 0 0	1 1 1 0 I3 I2 I1 I0	2	⅔	skip if E=n4　n4=0-FH	E=n4
	SKHEI n4	0 0 1 1 0 1 1 1	1 1 1 0 I3 I2 I1 I0	2	⅔	skip if H=n4　n4=0-FH	H=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 0 I3 I2 I1 I0	2	⅔	skip if L=n4　n4=0-FH	L=n4
御SIO命令制	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	2	SIO7-4←A SIO3-0←(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	2	A←SIO7-4 (HL)←SIO3-0	
	SIO	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 1	2	2	3Bit CNT←0 INT0/S RQF←0	
御タイマ命令制	TAMMOD	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 1	2	2	MOD7-4←A MOD3-0←(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	2	Start Timer	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	2	A←CT7-4 (HL)←CT3-0	
割込制御命令	EI n3	0 0 1 1 1 0 0 1	1 1 1 1 0 I2 I1 I0	2	2	IME←1　(n3=0) IE2-0←IE2-0∨n3(n3=1-7)	
	DI n3	0 0 1 1 1 0 0 0	1 1 1 1 0 I2 I1 I0	2	2	IME←0　(n3=0) IE2-0←IE2-0∧$\overline{n3}$(n3=1-7)	
	SKI n3	0 0 1 1 0 1 0 0	1 1 1 1 0 I2 I1 I0	2	⅔	Skip if INT RQF=1 then reset INT RQF n3=1-7	INT RQF=1
入出力命令	IPL	0 1 1 1	0 0 0 0	1	1	A←PORT(L)	
	IP addr3	0 0 1 1 1 1 0 0	1 1 1 1 0 D2D1D0	2	2	A←PORT(addr3)　addr3=0,1,4-7	
	IP1	0 1 1 1	0 0 0 1	1	1	A←PORT1	
	IP54	0 0 1 1 0 0 1 1	1 1 1 1 1 0 0 0	2	2	A←RORT5 (HL)←PORT4	
	OPL	0 1 1 1	0 0 1 0	1	1	PORT/MODE REG(L)←A	
	OP addr1	0 0 1 1 1 1 1 0	1 1 1 1 D3D2D1D0	2	2	PORT/MODE REG(addr1)←A addr1=1-7, BH, CH, EH, FH	
	OP3	0 1 1 1	0 0 1 1	1	1	PORT3←A	
	OP54	0 0 1 1 0 0 1 1	1 1 1 1 1 1 0 0	2	2	PORT5←A PORT4←(HL)	
	ANP addr4, n4	0 1 0 0 I3 I2 I1 I0	1 1 0 0 0 D2D1D0	2	2	PORT(addr4)←PORT(addr4)∧n4 addr4=2-7, n4=0-FH	
	ORP addr4, n4	0 1 0 0 I3 I2 I1 I0	1 1 0 1 0 D2D1D0	2	2	PORT(addr4)←PORT(addr4)∨n4 addr4=2-7, n4=0-FH	
御CPU命令制	HALT	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 0	2	2	Set HALT mode	
	STOP	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 1	2	2	Set STOP mode	
	NOP	0 0 0 0	0 0 0 0	1	1	No Operation	

保守／廃止

μPD7519/7519H　インストラクション活用表

命令群	ニーモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	マシンサイクル	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	$I_3I_2I_1I_0$	1	A←n4　n4=0-FH	縦積みLAI
	LDI n4	0 0 1 1 0 0 1 0	1 1 1 0 $I_3I_2I_1I_0$	2	D←n4　n4=0-FH	
	LEI n4	0 0 1 1 0 0 0 0	1 1 1 0 $I_3I_2I_1I_0$	2	E←n4　n4=0-FH	
	LHI n4	0 0 1 1 0 0 1 1	1 1 1 0 $I_3I_2I_1I_0$	2	H←n4　n4=0-FH	
	LLI n4	0 0 1 1 0 0 0 1	1 1 1 0 $I_3I_2I_1I_0$	2	L←n4　n4=0-FH	
	LAM pr	0 1 0 R_4	0 0 R_1R_0	½	A←(pr),pr=DL,DE,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	LADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 0 $D_3D_2D_1D_0$	2	A←(mem)　mem=0-FFH	
	LDEI byte	0 1 0 0 $I_7I_6I_5I_4$	1 1 1 1 $I_3I_2I_1I_0$	2	DE←byte　byte=0-FFH	
	LHLI byte	0 1 0 0 $I_7I_5I_6I_4$	1 1 1 0 $I_3I_2I_1I_0$	2	HL←byte　byte=0-FFH	縦積み LHLI LHLT
	LHLT taddr	1 1 0 0	$P_3P_2P_1P_0$	2	HL←ROM (00001100 $P_3P_2P_1P_0$) taddr=0C0H-0CFH	縦積み LHLI LHLT
	LAMTL	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 0	3	A←ROM $(PC_{11-8},A,(HL))_H$ $(HL)\leftarrow ROM\ (PC_{11-8},A,(HL))_L$	
	ST	0 1 0 1	0 1 1 1	1	(HL)←A	
	TAD	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 0	2	D←A	
	TAE	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 0	2	E←A	
	TAH	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 0	2	H←A	
	TAL	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 0	2	L←A	
	TDA	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 1	2	A←D	
	TEA	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 1	2	A←E	
	THA	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 1	2	A←H	
	TLA	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 1	2	A←L	
	XAD	0 1 0 0	1 0 1 0	1	A↔D	
	XAE	0 1 0 0	1 0 1 1	1	A↔E	
	XAH	0 1 1 1	1 0 1 0	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	A↔L	
	XAM pr	0 1 0 R_4	0 1 R_1R_0	½	A↔(pr)pr=DL,DE,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	XADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 1 $D_3D_2D_1D_0$	2	A↔(mem)　mem=0-FFH	
	XHDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 0 $D_3D_2D_1D_0$	2	H↔(mem)　mem=0-FFH	
	XLDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 1 $D_3D_2D_1D_0$	2	L↔(mem)　mem=0-FFH	
演算命令	AISC n4	0 0 0 0	$I_3I_2I_1I_0$	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
	ADSC	0 0 1 1 1 0 1 0	1 1 1 0 1 0 0 1	⅔	A←A+D	carry
	AESC	0 0 1 1 1 0 0 0	1 1 1 0 1 0 0 1	⅔	A←A+E	carry
	AHSC	0 0 1 1 1 0 1 1	1 1 1 0 1 0 0 1	⅔	A←A+H	carry
	ALSC	0 0 1 1 1 0 0 1	1 1 1 0 1 0 0 1	⅔	A←A+L	carry
	SDSB	0 0 1 1 1 0 1 0	1 1 1 0 1 0 0 0	⅔	A←A−D	borrow
	SESB	0 0 1 1 1 0 0 0	1 1 1 0 1 0 0 0	⅔	A←A−E	borrow
	SHSB	0 0 1 1 1 0 1 1	1 1 1 0 1 0 0 0	⅔	A←A−H	borrow
	SLSB	0 0 1 1 1 0 0 1	1 1 1 0 1 0 0 0	⅔	A←A−L	borrow
	EXL	0 1 1 1	1 1 1 0	1	A←A∨(HL)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	A←A∨(HL)	
アキュームレータ操作命令	CMA	0 1 1 1	1 1 1 1	1	$A\leftarrow\overline{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	C←A_0, A_3←C A_{n-1}←An　(n=1-3)	
	RAL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 1	2	C←A_3, An←A_{n-1} A_0←C　(n=1-3)	
キャリーフラグ操作命令	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	IES	0 1 0 0	1 0 0 1	½	E←E+1	E=0
	ILS	0 1 0 1	1 0 0 1	½	L←L+1	L=0
	IDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 0 1	2	DE←DE+1	
	IHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 0 1	2	HL←HL+1	
	IDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 1 $D_3D_2D_1D_0$	⅔	(mem)←(mem)+1　mem=0-FFH	(mem)=0
	DES	0 1 0 0	1 0 0 0	½	E←E−1	E=FH
	DLS	0 1 0 1	1 0 0 0	½	L←L−1	L=FH
	DDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 0 0	2	DE←DE−1	
	DHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 0 0	2	HL←HL−1	
	DDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 0 $D_3D_2D_1D_0$	⅔	(mem)←(mem)−1　mem=0-FFH	(mem)=FH

命令群	ニーモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	マシンサイクル	オペレーション	スキップ条件
ビット操作命令	RMB bit	0 1 1 0	1 0 B_1B_0	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	1 1 B_1B_0	1	(HL)bit←1　bit=0-3	
ジャンプ命令	JMP addr	0 0 1 0 $P_{11}P_{10}P_9P_8$	$P_7P_6P_5P_4$ $P_3P_2P_1P_0$	2	$PC_{11-0}\leftarrow P_{11-0}$　addr=0-FFFH	
	JCP addr	1 0 P_5P_4	$P_3P_2P_1P_0$	1	$PC_{5-0}\leftarrow P_{5-0}$　addr=0-FFFH	
	JAM addr1	0 0 1 1 0 0 0 1	1 1 1 1 $P_3P_2P_1P_0$	2	$PC_{11-8}\leftarrow P_{3-0}$, $PC_{7-4}\leftarrow A_{3-0}$ $PC_{3-0}\leftarrow$(HL)　addr1=0-FH	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 $P_7P_6P_5P_4$	0 $P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	(SP−1)(SP−2)(SP−4)←PC_{11-0} (SP−3)←PSW, SP←SP−4 $PC_{11-0}\leftarrow 0$, P_{10-0} caddr=000H-7FFH	
	CALT taddr1	1 1 P_5P_4	$P_3P_2P_1P_0$	2	(SP−1)(SP−2)(SP−4)←PC_{11-0} (SP−3)←PSW, SP←SP−4 $PC_{11,10,6,5}\leftarrow 0$ $PC_{9-7,4-0}\leftarrow$ROM(000011$P_5P_4P_3P_2P_1P_0$) taddr1=0D0H-0FFH	
	RT	0 1 0 1	0 0 1 1	2	$PC_{11-0}\leftarrow$(SP)(SP+2)(SP+3) SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	3	$PC_{11-0}\leftarrow$(SP)(SP+2)(SP+3) SP←SP+4 then skip unconditionally	無条件
	RTPSW	0 1 0 0	0 0 1 1	2	$PC_{11-0}\leftarrow$(SP)(SP+2)(SP+3) PSW←(SP+1), SP←SP+4	
	PSHDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 0	2	(SP−1)←D, (SP−2)←E SP←SP−2	
	PSHHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 0	2	(SP−1)←H, (SP−2)←L SP←SP−2	
	POPDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 1	2	E←(SP), D←(SP+1) SP←SP+2	
	POPHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 1	2	L←(SP), H←(SP+1) SP←SP+2	
	TAMSP	0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 1	2	$SP_{7-4}\leftarrow$A $SP_{3-1}\leftarrow$(HL)$_{3-1}$, $SP_0\leftarrow 0$	
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	A←SP_{7-4} $(HL)_{3-1}\leftarrow SP_{3-1}$, $(HL)_0\leftarrow 0$	
スキップ命令	SKC	0 1 0 1	1 0 1 0	½	skip if C=1	C=1
	SKMBT bit	0 1 1 0	0 1 B_1B_0	½	skip if (HL)bit=1　bit=0-3	(HL)bit=1
	SKABT bit	0 1 1 1	0 1 B_1B_0	½	skip if Abit=1　bit=0-3	Abit=1
	SKMBF bit	0 1 1 0	0 0 B_1B_0	½	skip if (HL)bit=0　bit=0-3	(HL)bit=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	1 1 1 1 $I_3I_2I_1I_0$	⅔	skip if A=n4　n4=0-FH	A=n4
	SKDEI n4	0 0 1 1 0 1 1 C	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if D=n4　n4=0-FH	D=n4
	SKEEI n4	0 0 1 1 0 1 0 0	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if E=n4　n4=0-FH	E=n4
	SKHEI n4	0 0 1 1 0 1 1 1	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if H=n4　n4=0-FH	H=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if L=n4　n4=0-FH	L=n4
	SKMEI n4	0 0 1 1 0 1 1 1	1 1 1 1 $I_3I_2I_1I_0$	⅔	skip if (HL)=n4　n4=0-FH	(HL)=n4
SIO制御命令	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	$SIO_{7-4}\leftarrow$A $SIO_{3-0}\leftarrow$(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	A←SIO_{7-4} (HL)←SIO_{3-0}	
	SIO	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 1	2	Start SIO 3-BIT CNT←0, INT0/S RQF←0	
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	$MOD_{7-4}\leftarrow$A $MOD_{3-0}\leftarrow$(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	Start Timer $CT_{7-0}\leftarrow 0$, INTT RQF←0	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	A←CT_{7-4} (HL)←CT_{3-0}	
割込制御命令	EI n3	0 0 1 1 1 0 0 1	1 1 1 1 0 $I_2I_1I_0$	2	IME←1　(n3=0) IE2-0←IE2-0∨n3　(n3=1-7)	
	DI n3	0 0 1 1 1 0 0 0	1 1 1 1 0 $I_2I_1I_0$	2	IME←0　(n3=0) IE2-0←IE2-0∧$\overline{n3}$　(n3=1-7)	
	SKI n3	0 0 1 1 0 1 0 0	1 1 1 1 0 $I_2I_1I_0$	⅔	skip if INT RQF=1 then reset INT RQF	INT RQF=1
入出力命令	IPL	0 1 1 1	0 0 0 0	1	A←Port(L)	
	IP addr3	0 0 1 1 1 1 0 0	1 1 1 1 $D_3D_2D_1D_0$	2	A←Port(addr3)　addr3= 0,1,4-6,8-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 addr2	0 0 1 1 1 1 1 0	1 1 1 1 $D_3D_2D_1D_0$	2	Port/Mode reg.(addr2)←A addr2=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)	
	ANP addr4,n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 0 $D_3D_2D_1D_0$	2	Port(addr4)←Port(addr4)∧n4 addr4.=2-6,8-BH,n4=0-FH	
	ORP addr4, n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 1 $D_3D_2D_1D_0$	2	Port(addr4)←Port(addr4)∨n4 addr4.=2-6,8-BH, n4=0-FH	
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 Operation	

μPD7527A, 7537A　インストラクション活用表

（適用品種：μPD7527, 7527A, 7537, 7537A）

命令群	ニモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	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	$I_3\ I_2\ I_1\ I_0$	1	1	A←n4　n4=0-FH	縦積みLAI
	LAM pr	0 1 0 R_4	0 0 R_1R_0	1	½	A←(pr)　pr=DL,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	LADR mem	0 0 1 1 0 $D_6D_5D_4$	1 0 0 0 $D_3D_2D_1D_0$	2	2	A←(mem)　mem=0-7FH	
	LHLI byte	0 1 0 0 $I_7\ I_6\ I_5\ I_4$	1 1 1 0 $I_3\ I_2\ I_1\ I_0$	2	2	HL←byte　byte=0-FFH	縦積み LHLI LHLT
	LHLT taddr1	1 1 0 0	$P_3P_2P_1P_0$	1	2	H←ROM(0001100$P_3P_2P_1P_0$)$_H$ L←ROM(0001100$P_3P_2P_1P_0$)$_L$ taddr1=0C0H−0CFH	縦積み LHLI LHLT
	LAMTL	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 0	2	3	A←ROM(PC_{10-8}A,(HL))$_H$ (HL)←ROM(PC_{10-8}A,(HL))$_L$	
	ST	0 1 0 1	0 1 1 1	1	1	(HL)←A	
	XAH	0 1 1 1	1 0 1 0	1	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	1	A↔L	
	XAM pr	0 1 0 R_4	0 1 R_1R_0	1	½	A↔(pr)　pr=DL,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	XADR mem	0 0 1 1 0 D_6 D_5D_4	1 0 0 1 $D_3D_2D_1D_0$	2	2	A↔(mem)　mem=0-7FH	
	XHDR mem	0 0 1 1 0 $D_6D_5D_4$	1 0 1 0 $D_3D_2D_1D_0$	2	2	H↔(mem)　mem=0-7FH	
	XLDR mem	0 0 1 1 0 $D_6D_5D_4$	1 0 1 1 $D_3D_2D_1D_0$	2	2	L↔(mem)　mem=0-7FH	
演算命令	AISC n4	0 0 0 0	$I_3\ I_2\ I_1\ I_0$	1	½	A←A+n4　n4=0-FH	carry
	ASC	0 1 1 1	1 1 0 1	1	½	A←A+(HL)	carry
	ACSC	0 1 1 1	1 1 0 0	1	½	A,C←A+(HL)+C	carry
	EXL	0 1 1 1	1 1 1 0	1	1	A←A∀(HL)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	2	A←A∨(HL)	
レジスタ操作命令	CMA	0 1 1 1	1 1 1 1	1	1	**A←$\overline{A}$**	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	2	C←A_0, A_3←C A_{n-1}←An　(n=1-3)	
フラグ操作命令	RC	0 1 1 1	1 0 0 0	1	1	C←0	
	SC	0 1 1 1	1 0 0 1	1	1	C←1	
増減命令	ILS	0 1 0 1	1 0 0 1	1	½	L←L+1	L=0
	IDRS mem	0 0 1 1 0 $D_6D_5D_4$	1 1 0 1 $D_3D_2D_1D_0$	2	¾	(mem)←(mem)+1 mem=0-7FH	(mem)=0
	DLS	0 1 0 1	1 0 0 0	1	½	L←L−1	L=FH
	DDRS mem	0 0 1 1 0 $D_6D_5D_4$	1 1 0 0 $D_3D_2D_1D_0$	2	¾	(mem)←(mem)−1 mem=0-7FH	(mem)=FH
ビットメモリ操作命令	RMB bit	0 1 1 0	1 0 B_1B_0	1	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	1 1 B_1B_0	1	1	(HL)bit←1　bit=0-3	
ジャンプ命令	JMP addr	0 0 1 0 $P_7P_6P_5P_4$	0 $P_{10}P_9P_8$ $P_3\ P_2\ P_1P_0$	2	2	PC_{10-0}←P_{10-0}　addr=0-7FFH	
	JCP addr	1 0 P_5P_4	$P_3P_2P_1P_0$	1	1	PC_{5-0}←P_{5-0}　addr=0-7FFH	
	JAM addr2	0 0 1 1 0 0 0 1	1 1 1 1 0 $P_2P_1P_0$	2	2	PC_{10-6}←P_{2-0}, PC_{7-4}←A_{3-0} PC_{3-0}←(HL)　addr2=0-7	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 $P_7P_6P_5P_4$	0$P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	2	(SP-1)(SP-2)(SP-4)←PC_{10-0} (SP-3)←PSW, PC_{10-0}←P_{10-0} SP←SP−4　caddr=0-7FFH	
	CALT taddr2	1 1 P_5P_4	$P_3P_2P_1P_0$	1	2	(SP-1)(SP-2)(SP-4)←$PC_{0\ 3}$,SP-3)←PSW $PC_{7-4,0}$←ROM(0001P$_5$P$_4$P$_3$P$_2$P$_1$P$_0$) $PC_{10,9,8}$←0 SP←SP−4　taddr2=0D0H-0FFH	
	RT	0 1 0 1	0 0 1 1	1	2	PC_{10-0}←(SP)(SP+2)(SP+3) SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	1	3	PC_{10-0}←(SP)(SP+2)(SP+3) SP←SP+4 **then skip unconditionally**	無条件
	RTPSW	0 1 0 0	0 0 1 1	1	2	PC_{10-0}←(SP)(SP+2)(SP+3) PSW←(SP+1) SP←SP+4	
	TAMSP	0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 1	2	2	SP_{7-4}←A SP_{3-1}←(HL)$_{3-1}$, SP_0←0	
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	2	A←SP_{7-4} (HL)$_{3-1}$←SP_{3-1}, (HL)$_0$←0	

命令群	ニモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	オペレーション	スキップ条件
スキップ命令	SKC	0 1 0 1	1 0 1 0	1	½	skip if C=1	C=1
	SKMBT bit	0 1 1 0	0 1 B_1B_0	1	½	skip if(HL)bit=1　bit=0-3	(HL)bit=1
	SKABT bit	0 1 1 1	0 1 B_1B_0	1	½	skip if Abit=1　bit=0-3	Abit=1
	SKMBF bit	0 1 1 0	0 0 B_1B_0	1	½	skip if(HL)bit=0　bit=0-3	(HL)bit=0
	SKAEM	0 1 0 1	1 1 1 1	1	½	skip if A=(HL)	A=(HL)
	SKAEI n4	0 0 1 1 0 1 1 0	1 1 1 1 $I_3\ I_2\ I_1\ I_0$	2	¾	skip if A=n4　n4=0-FH	A=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 0 $I_3\ I_2\ I_1\ I_0$	2	¾	skip if L=n4　n4=0-FH	L=n4
御命令制SIO	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	2	SIO_{7-4}←A SIO_{3-0}←(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	2	A←SIO_{7-4} (HL)←SIO_{3-0}	
	SIO	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 1	2	2	3Bit CNT←0 INT0/S RQF←0	
御命令制タイマ	TAMMOD	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 1	2	2	MOD_{7-4}←A MOD_{3-0}←(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	2	Start Timer	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	2	A←CT_{7-4} (HL)←CT_{3-0}	
割込制御命令	EI n2	0 0 1 1 1 0 0 1	1 1 1 1 0 0 $I_1\ I_0$	2	2	IME←1　(n2=0) IE1-0←IE1-0∨n2(n2=1−3)	
	DI n2	0 0 1 1 1 0 0 0	1 1 1 1 0 0 $I_1\ I_0$	2	2	IME←0　(n2=0) IE1-0←IE1-0∧$\overline{n2}$(n2=1-3)	
	SKI n2	0 0 1 1 0 1 0 0	1 1 1 1 0 0 $I_1\ I_0$	2	¾	Skip if INT RQF=1 **then reset INT RQF n2=1-3**	INT RQF=1
入出力命令	IPL	0 1 1 1	0 0 0 0	1	1	A←PORT(L)	
	IP addr3	0 0 1 1 1 1 0 0	1 1 1 1 $D_3D_2D_1D_0$	2	2	A←PORT(addr3) addr3=0,4,5,AH,BH	
	IP54	0 0 1 1 0 0 1 1	1 1 1 1 1 0 0 0	2	2	A←RORT5 (HL)←PORT4	
	OPL	0 1 1 1	0 0 1 0	1	1	PORT/MODE REG(L)←A	
	OP addr1	0 0 1 1 1 1 1 0	1 1 1 1 $D_3D_2D_1D_0$	2	2	PORT/MODE REG(addr1)←A addr1=2-5,8-BH,CH,FH	
	OP3	0 1 1 1	0 0 1 1	1	1	PORT3←A	
	OP54	0 0 1 1 0 0 1 1	1 1 1 1 1 1 0 0	2	2	PORT5←A PORT4←(HL)	
	RPBL	0 1 0 1	1 1 0 0	1	1	Port bit(L)←0	
	SPBL	0 1 0 1	1 1 0 1	1	1	Port bit(L)←1	
御命令制CPU	HALT	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 0	2	2	Set HALT mode	
	STOP	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 1	2	2	Set STOP mode	
	NOP	0 0 0 0	0 0 0 0	1	1	No Operation	

保守／廃止

μPD7528A, 7538A インストラクション活用表

（適用品種：μPD7528, 7528A, 7538, 7538A）

命令群	ニーモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	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	$I_3\ I_2\ I_1\ I_0$	1	1	A←n4　n4=0-FH	縦積みLAI
	LAM pr	0 1 0 R_4	0 0 R_1R_0	1	½	A←(pr)　pr=DL,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	LADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 0 $D_3D_2D_1D_0$	2	2	A←(mem)　mem=0-9FH	
	LHLI byte	0 1 0 0 $I_7\ I_6\ I_5\ I_4$	1 1 1 0 $I_3\ I_2\ I_1\ I_0$	2	2	HL←byte　byte=0-FFH	縦積み LHLI LHLT
	LHLT taddr1	1 1 0 0	$P_3P_2P_1P_0$	1	2	H←ROM(00001100$P_3P_2P_1P_0$)$_H$ L←ROM(00001100$P_3P_2P_1P_0$)$_L$ taddr1=0C0H−0CFH	縦積み LHLI LHLT
	LAMTL	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 0	2	3	A←ROM(PC_{11-8},A,(HL))$_H$ (HL)←ROM(PC_{11-8},A,(HL))$_L$	
	ST	0 1 0 1	0 1 1 1	1	1	(HL)←A	
	XAH	0 1 1 1	1 0 1 0	1	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	1	A↔L	
	XAM pr	0 1 0 R_4	0 1 R_1R_0	1	½	A↔(pr)　pr=DL,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	XADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 1 $D_3D_2D_1D_0$	2	2	A↔(mem)　mem=0-9FH	
	XHDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 0 $D_3D_2D_1D_0$	2	2	H↔(mem)　mem=0-9FH	
	XLDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 1 $D_3D_2D_1D_0$	2	2	L↔(mem)　mem=0-9FH	
演算命令	AISC n4	0 0 0 0	$I_3\ I_2\ I_1\ I_0$	1	½	A←A+n4　n4=0-FH	carry
	ASC	0 1 1 1	1 1 0 1	1	½	A←A+(HL)	carry
	ACSC	0 1 1 1	1 1 0 0	1	½	A,C←A+(HL)+C	carry
	EXL	0 1 1 1	1 1 1 0	1	1	A←A∀(HL)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	2	A←A∨(HL)	
レジスタ操作命令 アキュームレータ操作命令	CMA	0 1 1 1	1 1 1 1	1	1	A←$\overline{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	2	C←A_0, A_3←C A_{n-1}←An　(n=1-3)	
キャリフラグ操作命令	RC	0 1 1 1	1 0 0 0	1	1	C←0	
	SC	0 1 1 1	1 0 0 1	1	1	C←1	
増減命令	ILS	0 1 0 1	1 0 0 1	1	½	L←L+1	L=0
	IDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 1 $D_3D_2D_1D_0$	2	¾	(mem)←(mem)+1 mem=0-9FH	(mem)=0
	DLS	0 1 0 1	1 0 0 0	1	½	L←L−1	L=FH
	DDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 0 $D_3D_2D_1D_0$	2	¾	(mem)←(mem)−1 mem=0-9FH	(mem)=FH
ビットメモリ操作命令	RMB bit	0 1 1 0	1 0 B_1B_0	1	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	1 1 B_1B_0	1	1	(HL)bit←1　bit=0-3	
ジャンプ命令	JMP addr	0 0 1 0 $P_7P_6P_5P_4$	$P_{11}P_{10}P_9P_8$ $P_3\ P_2\ P_1P_0$	2	2	PC_{11-0}←P_{11-0}　addr=0-FFFH	
	JCP addr	1 0 P_5P_4	$P_3P_2P_1P_0$	1	1	PC_{5-0}←P_{5-0}　addr=0-FFFH	
	JAM addr2	0 0 1 1 0 0 0 1	1 1 1 1 $P_3P_2P_1P_0$	2	2	PC_{11-8}←P_{3-0}, PC_{7-4}←A_{3-0} PC_{3-0}←(HL)　addr2=0-FH	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 $P_7P_6P_5P_4$	0$P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	2	(SP−1)(SP−2)(SP−4)←PC_{11-0} (SP−3)←PSW,PC_{11-0}←0,P_{10-0} SP←SP−4　caddr=0-7FFH	
	CALT taddr2	1 1 P_5P_4	$P_3P_2P_1P_0$	1	2	(SP−1)(SP−2)(SP−4)←PC_{11-0}(SP−3)←PSW $PC_{3-1,0,0,0}$←ROM(0000110$P_5P_4P_3P_2P_1P_0$) $PC_{11,10,9,8}$←0 SP←SP−4　taddr2=0D0H-0FFH	
	RT	0 1 0 1	0 0 1 1	1	2	PC_{11-0}←(SP)(SP+2)(SP+3) SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	1	3	PC_{11-0}←(SP)(SP+2)(SP+3) SP←SP+4 then skip unconditionally	無条件
	RTPSW	0 1 0 0	0 0 1 1	1	2	PC_{11-0}←(SP)(SP+2)(SP+3) PSW←(SP+1) SP←SP+4	
	TAMSP	0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 1	2	2	SP_{7-4}←A $SP_{3\ 1}$←$(HL)_{3-1}$,SP_0←0	
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	2	A←SP_{7-4} $(HL)_{3-1}$←SP_{3-1}, $(HL)_0$←0	

命令群	ニーモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	オペレーション	スキップ条件
スキップ命令	SKC	0 1 0 1	1 0 1 0	1	½	skip if C=1	C=1
	SKMBT bit	0 1 1 0	0 1 B_1B_0	1	½	skip if(HL)bit=1　bit=0-3	(HL)bit=1
	SKABT bit	0 1 1 1	0 1 B_1B_0	1	½	skip if Abit=1　bit=0-3	Abit=1
	SKMBF bit	0 1 1 0	0 0 B_1B_0	1	½	skip if(HL)bit=0　bit=0-3	(HL)bit=0
	SKAEM	0 1 0 1	1 1 1 1	1	½	skip if A=(HL)	A=(HL)
	SKAEI n4	0 0 1 1 0 1 1 0	1 1 1 1 $I_3\ I_2\ I_1\ I_0$	2	¾	skip if A=n4　n4=0-FH	A=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 0 $I_3\ I_2\ I_1\ I_0$	2	¾	skip if L=n4　n4=0-FH	L=n4
御SIO命令制	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	2	$SIO_{7\ 4}$←A SIO_{3-0}←(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	2	A←SIO_{7-4} (HL)←SIO_{3-0}	
	SIO	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 1	2	2	3Bit CNT←0 INT0/S RQF←0	
御タイマ命令制	TAMMOD	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 1	2	2	MOD_{7-4}←A MOD_{3-0}←(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	2	Start Timer	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	2	A←$CT_{7\ 4}$ (HL)←CT_{3-0}	
割込制御命令	EI n2	0 0 1 1 1 0 0 1	1 1 1 1 0 0 $I_1\ I_0$	2	2	IME←I　(n2=0) IE1-0←IE1-0∨n2(n2=1−3)	
	DI n2	0 0 1 1 1 0 0 0	1 1 1 1 0 0 $I_1\ I_0$	2	2	IME←0　(n2=0) IE 1-0←IE1-0∧$\overline{n2}$(n2=1-3)	
	SKI n2	0 0 1 1 0 1 0 0	1 1 1 1 0 0 $I_1\ I_0$	2	¾	Skip if INT RQF=1 **then reset INT RQF n2=1-3**	INT RQF=1
入出力命令	IPL	0 1 1 1	0 0 0 0	1	1	A←PORT(L)	
	IP addr3	0 0 1 1 1 1 0 0	1 1 1 1 $D_3\ D_2\ D_1\ D_0$	2	2	A←PORT(addr3) addr3=0,4,5,AH,BH	
	IP54	0 0 1 1 0 0 1 1	1 1 1 1 1 0 0 0	2	2	A←RORT5 (HL)←PORT4	
	OPL	0 1 1 1	0 0 1 0	1	1	PORT/MODE REG(L)←A	
	OP addr1	0 0 1 1 1 1 1 0	1 1 1 1 $D_3\ D_2\ D_1\ D_0$	2	2	PORT/MODE REG(addr1)←A addr1=2-5,8-BH,CH,FH	
	OP3	0 1 1 1	0 0 1 1	1	1	PORT3←A	
	OP54	0 0 1 1 0 0 1 1	1 1 1 1 1 1 0 0	2	2	PORT5←A PORT4←(HL)	
	RPBL	0 1 0 1	1 1 0 0	1	1	Port bit(L)←0	
	SPBL	0 1 0 1	1 1 0 1	1	1	Port bit(L)←1	
御CPU命令制	HALT	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 0	2	2	Set HALT mode	
	STOP	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 1	2	2	Set STOP mode	
	NOP	0 0 0 0	0 0 0 0	1	1	No Operation	

μPD7516H インストラクション活用表

(適用品種：μPD7516, 7516H)

命令群	ニモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	マサイクル	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	$I_3I_2I_1I_0$	1	A←n4　n4=0-FH	縦積みLAI
	LDI n4	0 0 1 1 0 0 1 0	1 1 1 0 $I_3I_2I_1I_0$	2	D←n4　n4=0-FH	
	LEI n4	0 0 1 1 0 0 0 0	1 1 1 0 $I_3I_2I_1I_0$	2	E←n4　n4=0-FH	
	LHI n4	0 0 1 1 0 0 1 1	1 1 1 0 $I_3I_2I_1I_0$	2	H←n4　n4=0-FH	
	LLI n4	0 0 1 1 0 0 0 1	1 1 1 0 $I_3I_2I_1I_0$	2	L←n4　n4=0-FH	
	LAM pr	0 1 0 R_4	0 0 R_1R_0	½	A←(pr),pr=DL,DE,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	LADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 0 $D_3D_2D_1D_0$	2	A←(mem)　mem=0-FFH	
	LDEI byte	0 1 0 0 $I_7I_6I_5I_4$	1 1 1 1 $I_3I_2I_1I_0$	2	DE←byte　byte=0-FFH	
	LHLI byte	0 1 0 0 $I_7I_5I_6I_4$	1 1 1 0 $I_3I_2I_1I_0$	2	HL←byte　byte=0-FFH	縦積み LHLI LHLT
	LHLT taddr	1 1 0 0	$P_3P_2P_1P_0$	2	HL←TABLE(0000001100$P_3P_2P_1P_0$) taddr=00C0H−00CFH	縦積み LHLI LHLT
	LAMTL	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 0	3	A←TABLE(BNK,$PC_{11-8}A_{3-0}$,$(HL)_{3-0}$)H (HL)←TABLE(BNK,$PC_{11-8}A_{3-0}$,$(HL)_{3-0}$)L	
	ST	0 1 0 1	0 1 1 1	1	(HL)←A	
	TAD	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 0	2	D←A	
	TAE	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 0	2	E←A	
	TAH	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 0	2	H←A	
	TAL	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 0	2	L←A	
	TDA	0 0 1 1 1 0 1 0	1 1 1 0 1 0 1 1	2	A←D	
	TEA	0 0 1 1 1 0 0 0	1 1 1 0 1 0 1 1	2	A←E	
	THA	0 0 1 1 1 0 1 1	1 1 1 0 1 0 1 1	2	A←H	
	TLA	0 0 1 1 1 0 0 1	1 1 1 0 1 0 1 1	2	A←L	
	XAD	0 1 0 0	1 0 1 0	1	A↔D	
	XAE	0 1 0 0	1 0 1 1	1	A↔E	
	XAH	0 1 1 1	1 0 1 0	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	A↔L	
	XAM pr	0 1 0 R_4	0 1 R_1R_0	½	A↔(pr),pr=DL,DE,HL−,HL+,HL	L=FH(HL−) L=0 (HL+)
	XADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 1 $D_3D_2D_1D_0$	2	A↔(mem)　mem=0-FFH	
	XHDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 0 $D_3D_2D_1D_0$	2	H↔(mem)　mem=0-FFH	
	XLDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 1 $D_3D_2D_1D_0$	2	L↔(mem)　mem=0-FFH	
演算命令	AISC n4	0 0 0 0	$I_3I_2I_1I_0$	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
	ADSC	0 0 1 1 1 0 1 0	1 1 1 0 1 0 0 1	⅔	A←A+D	carry
	AESC	0 0 1 1 1 0 0 0	1 1 1 0 1 0 0 1	⅔	A←A+E	carry
	AHSC	0 0 1 1 1 0 1 1	1 1 1 0 1 0 0 1	⅔	A←A+H	carry
	ALSC	0 0 1 1 1 0 0 1	1 1 1 0 1 0 0 1	⅔	A←A+L	carry
	SDSB	0 0 1 1 1 0 1 0	1 1 1 0 1 0 0 0	⅔	A←A−D	borrow
	SESB	0 0 1 1 1 0 0 0	1 1 1 0 1 0 0 0	⅔	A←A−E	borrow
	SHSB	0 0 1 1 1 0 1 1	1 1 1 0 1 0 0 0	⅔	A←A−H	borrow
	SLSB	0 0 1 1 1 0 0 1	1 1 1 0 1 0 0 0	⅔	A←A−L	borrow
	EXL	0 1 1 1	1 1 1 0	1	A←A∀(HL)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	A←A∨(HL)	
アキュームレータ操作命令	CMA	0 1 1 1	1 1 1 1	1	A←$\overline{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	C←A_0, A_3←C A_{n-1}←A_n　(n=1-3)	
	RAL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 1	2	C←A_3, A_n←A_{n-1} A_0←C　(n=1-3)	
キャリフラグ操作命令	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	IES	0 1 0 0	1 0 0 1	½	E←E+1	E=0
	ILS	0 1 0 1	1 0 0 1	½	L←L+1	L=0
	IDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 0 1	2	DE←DE+1	
	IHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 0 1	2	HL←HL+1	
	IDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 1 $D_3D_2D_1D_0$	⅔	(mem)←(mem)+1　mem=0-FFH	(mem)=0
	DES	0 1 0 0	1 0 0 0	½	E←E−1	E=FH
	DLS	0 1 0 1	1 0 0 0	½	L←L−1	L=FH
	DDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 0 0	2	DE←DE−1	
	DHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 0 0	2	HL←HL−1	
	DDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 0 $D_3D_2D_1D_0$	⅔	(mem)←(mem)−1　mem=0-FFH	(mem)=FH

命令群	ニモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	マサイクル	オペレーション	スキップ条件
メモリビット操作命令	RMB bit	0 1 1 0	1 0 B_1B_0	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	1 1 B_1B_0	1	(HL)bit←1　bit=0-3	
ジャンプ命令	JMP addr 1	0 0 1 0 $P_7P_6P_5P_4$	$P_{11}P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	PC_{11-0}←P_{11-0}　addr1=0-1FFFH	
	JMPL addr	0 0 1 1 1 1 1 1	00P_{13}0$P_{11}P_{10}P_9P_8$ B_3 $P_7P_6P_5P_4P_3P_2P_1P_0$		BNK←P_{13} PC_{11-0}←P_{11-0}	
	JCP addr 1	1 0 P_5P_4	$P_3P_2P_1P_0$	1	PC_{5-0}←P_{5-0}　addr1=0-1FFFH	
	JAM addr 2	0 0 1 1 0 0 0 1	1 1 1 1 $P_3P_2P_1P_0$	2	PC_{11-8}←P_{3-0}, PC_{7-4}←A_{3-0} PC_{3-0}←(HL)　addr2=0-FH	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 $P_7P_6P_5P_4$	0 $P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	(SP−1)(SP−2)(SP−4)←PC_{11-0} (SP−3)←PSW, SP←SP−4 BNK←0, PC_{11-0}←0, P_{10-0} caddr=0000H−07FFH	
	CALT taddr1	1 1 P_5P_4	$P_3P_2P_1P_0$	2	(SP−1)(SP−2)(SP−4)←PC_{11-0} (SP−3)←PSW, SP←SP−4 BNK←0, $PC_{11-10,6,5}$←0 $PC_{9-7,4-0}$←TABLE(0000011 $P_5P_4P_3P_2P_1P_0$) taddr1=00D0H−00FFH	
	RT	0 1 0 1	0 0 1 1	2	PC_{11-0}←(SP)(SP+2)(SP+3) BNK←$(SP+1)_{1,}$SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	3	PC_{11-0}←(SP)(SP+2)(SP+3) BNK←$(SP+1)_{1,}$SP←SP+4 then skip unconditionally	無条件
	RTPSW	0 1 0 0	0 0 1 1	2	PC_{11-0}←(SP)(SP+2)(SP+3) PSW←(SP+1), SP←SP+4	
	PSHDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 0	2	(SP−1)←D, (SP−2)←E SP←SP−2	
	PSHHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 0	2	(SP−1)←H, (SP−2)←L SP←SP−2	
	POPDE	0 0 1 1 1 0 0 0	1 1 1 0 1 1 1 1	2	E←(SP), D←(SP+1) SP←SP+2	
	POPHL	0 0 1 1 1 0 0 1	1 1 1 0 1 1 1 1	2	L←(SP), H←(SP+1) SP←SP+2	
	TAMSP	0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 1	2	SP_{7-4}←A SP_{3-1}←$(HL)_{3-1}$, SP_0←0	
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	A←SP_{7-4} $(HL)_{3-1}$←SP_{3-1}, $(HL)_0$←0	
スキップ命令	SKC	0 1 0 1	1 0 1 0	½	skip if C=1	C=1
	SKMBT bit	0 1 1 0	0 1 B_1B_0	½	skip if (HL)bit=1　bit=0-3	(HL)bit=1
	SKABT bit	0 1 1 1	0 1 B_1B_0	½	skip if Abit=1　bit=0-3	Abit=1
	SKMBF bit	0 1 1 0	0 0 B_1B_0	½	skip if (HL)bit=0　bit=0-3	(HL)bit=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	1 1 1 1 $I_3I_2I_1I_0$	⅔	skip if A=n4　n4=0-FH	A=n4
	SKDEI n4	0 0 1 1 0 1 1 0	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if D=n4　n4=0-FH	D=n4
	SKEEI n4	0 0 1 1 0 1 0 0	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if E=n4　n4=0-FH	E=n4
	SKHEI n4	0 0 1 1 0 1 1 1	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if H=n4　n4=0-FH	H=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 0 $I_3I_2I_1I_0$	⅔	skip if L=n4　n4=0-FH	L=n4
	SKMEI n4	0 0 1 1 0 1 1 1	1 1 1 1 $I_3I_2I_1I_0$	⅔	skip if (HL)=n4　n4=0-FH	(HL)=n4
SIO制御命令	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	SIO_{7-4}←A SIO_{3-0}←(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	A←SIO_{7-4} (HL)←SIO_{3-0}	
	SIO	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 1	2	Start SIO 3-BIT CNT←0, INT0/S RQF←0	
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_{7-4}←A MOD_{3-0}←(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	Start Timer CT_{7-0}←0, INTT RQF←0	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	A←CT_{7-4} (HL)←CT_{3-0}	
割込制御命令	EI n3	0 0 1 1 1 0 0 1	1 1 1 1 0 $I_2I_1I_0$	2	IME←1　(n3=0) IE2-0←IE2-0∨n3　(n3=1-7)	
	DI n3	0 0 1 1 1 0 0 0	1 1 1 1 0 $I_2I_1I_0$	2	IME←0　(n3=0) IE2-0←IE2-0∧$\overline{n3}$　(n3=1-7)	
	SKI n3	0 0 1 1 0 1 0 0	1 1 1 1 0 $I_2I_1I_0$	⅔	skip if INT RQF=1 then reset INT RQF	INT RQF=1
入出力命令	IPL	0 1 1 1	0 0 0 0	1	A←Port(L)	
	IP addr4	0 0 1 1 1 1 0 0	1 1 1 1 $D_3D_2D_1D_0$	2	A←Port(addr4)　addr4= 0,1,4-6,8-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 addr3	0 0 1 1 1 1 1 0	1 1 1 1 $D_3D_2D_1D_0$	2	Port/Mode reg.(addr3)←A addr3=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)	
	ANP addr5,n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 0 $D_3D_2D_1D_0$	2	Port(addr5)←Port(addr5)∧n4 addr5.=2-6,8-BH,n4=0-FH	
	ORP addr5,n4	0 1 0 0 $I_3I_2I_1I_0$	1 1 0 1 $D_3D_2D_1D_0$	2	Port(addr5)←Port(addr5)∨n4 addr5.=2-6,8-BH,n4=0-FH	
制CPU命令制	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 Operation	

保守／廃止

μPD7533　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 $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	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	$I_3\ I_2\ I_1\ I_0$	1	1	A←n4　n4=0-FH	縦積みLAI
	LAM pr	0 1 0 R_1	0 0 R_1R_0	1	½	A←(pr)　pr=DL,HL−,HL+,HL	L=FH(HL−) L=0　(HL+)
	LADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 0 $D_3D_2D_1D_0$	2	2	A←(mem)　mem=0-9FH	
	LHLI byte	0 1 0 0 $I_7\ I_6\ I_5\ I_4$	1 1 1 0 $I_3\ I_2\ I_1\ I_0$	2	2	HL←byte　byte=0-FFH	縦積み LHLI LHLT
	LHLT taddr1	1 1 0 0	$P_3P_2P_1P_0$	1	2	H←ROM(00001100$P_3P_2P_1P_0$)H L←ROM(00001100$P_3P_2P_1P_0$)L taddr1=0C0H−0CFH	縦積み LHLI LHLT
	LAMTL	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 0	2	3	A←ROM(PC_{11-8},A,(HL))H (HL)←ROM(PC_{11-8},A,(HL))L	
	ST	0 1 0 1	0 1 1 1	1	1	(HL)←A	
	XAH	0 1 1 1	1 0 1 0	1	1	A↔H	
	XAL	0 1 1 1	1 0 1 1	1	1	A↔L	
	XAM pr	0 1 0 R_1	0 1 R_1R_0	1	½	A↔(pr)　pr=DL,HL−,HL+,HL	L=FH(HL−) L=0　(HL+)
	XADR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 0 1 $D_3D_2D_1D_0$	2	2	A↔(mem)　mem=0-9FH	
	XHDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 0 $D_3D_2D_1D_0$	2	2	H↔(mem)　mem=0-9FH	
	XLDR mem	0 0 1 1 $D_7D_6D_5D_4$	1 0 1 1 $D_3D_2D_1D_0$	2	2	L↔(mem)　mem=0-9FH	
演算命令	AISC n4	0 0 0 0	$I_3\ I_2\ I_1\ I_0$	1	½	A←A+n4　n4=0-FH	carry
	ASC	0 1 1 1	1 1 0 1	1	½	A←A+(HL)	carry
	ACSC	0 1 1 1	1 1 0 0	1	½	A,C←A+(HL)+C	carry
	EXL	0 1 1 1	1 1 1 0	1	1	A←A∀(HL)	
	ANL	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 0	2	2	A←A∧(HL)	
	ORL	0 0 1 1 1 0 1 1	1 1 1 1 0 1 1 0	2	2	A←A∨(HL)	
アキュームレータ操作命令	CMA	0 1 1 1	1 1 1 1	1	1	$A\leftarrow\overline{A}$	
	RAR	0 0 1 1 1 0 1 1	1 1 1 1 0 0 1 1	2	2	C←A_0, A_3←C A_{n-1}←An　(n=1-3)	
フィアクリー操作命令	RC	0 1 1 1	1 0 0 0	1	1	C←0	
	SC	0 1 1 1	1 0 0 1	1	1	C←1	
増減命令	ILS	0 1 0 1	1 0 0 1	1	½	L←L+1	L=0
	IDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 1 $D_3D_2D_1D_0$	2	⅔	(mem)←(mem)+1 mem=0-9FH	(mem)=0
	DLS	0 1 0 1	1 0 0 0	1	½	L←L−1	L=FH
	DDRS mem	0 0 1 1 $D_7D_6D_5D_4$	1 1 0 0 $D_3D_2D_1D_0$	2	⅔	(mem)←(mem)−1 mem=0-9FH	(mem)=FH
メモリビット操作命令	RMB bit	0 1 1 0	1 0 B_1B_0	1	1	(HL)bit←0　bit=0-3	
	SMB bit	0 1 1 0	1 1 B_1B_0	1	1	(HL)bit←1　bit=0-3	
ジャンプ命令	JMP addr	0 0 1 0 $P_7P_6P_5P_4$	$P_{11}P_{10}P_9P_8$ $P_3\ P_2\ P_1P_0$	2	2	PC_{11-0}←P_{11-0}　addr=0-FFFH	
	JCP addr	1 0 P_5P_4	$P_3P_2P_1P_0$	1	1	PC_{5-0}←P_{5-0}　addr=0-FFFH	
	JAM addr2	0 0 1 1 0 0 0 1	1 1 1 1 $P_3P_2P_1P_0$	2	2	PC_{11-8}←P_{3-0}, PC_{7-4}←$A_{3\ 0}$ PC_{3-0}←(HL)　addr2=0-FH	
サブルーチン・スタック制御命令	CALL caddr	0 0 1 1 $P_7P_6P_5P_4$	0 $P_{10}P_9P_8$ $P_3P_2P_1P_0$	2	2	(SP−1)(SP−2)(SP−4)←PC_{11-0} (SP−3)←PSW,PC_{13-0}←0,P_{10-0} SP←SP−4　caddr=0-7FFH	
	CALT taddr2	1 1 P_5P_4	$P_3P_2P_1P_0$	1	2	(SP−1)(SP−2)(SP−4)←PC_{11-0}(SP−3)←PSW $PC_{13-8,7-6}$←ROM(00001$P_5P_4P_3P_2P_1P_0$) $PC_{11,10,5,4}$←0 SP←SP−4　taddr2=0D0H-0FFH	
	RT	0 1 0 1	0 0 1 1	1	2	PC_{11-0}←(SP)(SP+2)(SP+3) SP←SP+4	
	RTS	0 1 0 1	1 0 1 1	1	3	PC_{11-0}←(SP)(SP+2)(SP+3) SP←SP+4 then skip unconditionally	無条件
	RTPSW	0 1 0 0	0 0 1 1	1	2	PC_{11-0}←(SP)(SP+2)(SP+3) PSW←(SP+1) SP←SP+4	
	TAMSP	0 0 1 1 0 0 1 1	1 1 1 1 0 0 0 1	2	2	SP_{7-4}←A SP_{3-1}←$(HL)_{3-1}$,SP_0←0	
	TSPAM	0 0 1 1 0 0 1 1	1 1 1 1 0 1 0 1	2	2	A←SP_{7-4} $(HL)_{3-1}$←$SP_{3\ 1}$,$(HL)_0$←0	

命令群	ニモニック	命令コード $D_7D_6D_5D_4$	命令コード $D_3D_2D_1D_0$	バイト	マシンサイクル	オペレーション	スキップ条件
スキップ命令	SKC	0 1 0 1	1 0 1 0	1	½	skip if C=1	C=1
	SKMBT bit	0 1 1 0	0 1 B_1B_0	1	½	skip if(HL)bit=1　bit=0-3	(HL)bit=1
	SKABT bit	0 1 1 1	0 1 B_1B_0	1	½	skip if Abit=1　bit=0-3	Abit=1
	SKMBF bit	0 1 1 0	0 0 B_1B_0	1	½	skip if(HL)bit=0　bit=0-3	(HL)bit=0
	SKAEM	0 1 0 1	1 1 1 1	1	½	skip if A=(HL)	A=(HL)
	SKAEI n4	0 0 1 1 0 1 1 0	1 1 1 1 $I_3\ I_2\ I_1\ I_0$	2	⅔	skip if A=n4　n4=0-FH	A=n4
	SKLEI n4	0 0 1 1 0 1 0 1	1 1 1 0 $I_3\ I_2\ I_1\ I_0$	2	⅔	skip if L=n4　n4=0-FH	L=n4
御SIO命令制	TAMSIO	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 0	2	2	SIO_{7-4}←A SIO_{3-0}←(HL)	
	TSIOAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 0	2	2	A←SIO_{7-4} (HL)←SIO_{3-0}	
	SIO	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 1	2	2	3Bit CNT←0 INT0/S RQF←0	
御タイマ命令制	TAMMOD	0 0 1 1 0 0 1 1	1 1 1 1 1 1 1 1	2	2	MOD_{7-4}←A MOD_{3-0}←(HL)	
	TIMER	0 0 1 1 0 0 1 1	1 1 1 1 0 0 1 0	2	2	Start Timer	
	TCNTAM	0 0 1 1 0 0 1 1	1 1 1 1 1 0 1 1	2	2	A←CT_{7-4} (HL)←CT_{3-0}	
割り込み制御命令	EI n2	0 0 1 1 1 0 0 1	1 1 1 1 0 0 I_1I_0	2	2	IME←1　(n2=0) IE1-0←IE1-0∨n2(n2=1−3)	
	DI n2	0 0 1 1 1 0 0 0	1 1 1 1 0 0 I_1I_0	2	2	IME←0　(n2=0) IE1-0←IE1-0∧$\overline{n2}$(n2=1-3)	
	SKI n2	0 0 1 1 0 1 0 0	1 1 1 1 0 0 I_1I_0	2	⅔	Skip if INT RQF=1 **then reset INT RQF n2=1-3**	INT RQF=1
入出力命令	IPL	0 1 1 1	0 0 0 0	1	1	A←PORT/RESULT REG(L)	
	IP addr3	0 0 1 1 1 1 0 0	1 1 1 1 $D_3D_2D_1D_0$	2	2	A←PORT/RESULT REG(addr3) addr3=0,1,4-AH	
	IP54	0 0 1 1 0 0 1 1	1 1 1 1 1 0 0 0	2	2	A←PORT5 (HL)←PORT4	
	OPL	0 1 1 1	0 0 1 0	1	1	PORT/MODE REG(L)←A	
	OP addr1	0 0 1 1 1 1 1 0	1 1 1 1 $D_3D_2D_1D_0$	2	2	PORT/MODE REG(addr1)←A addr1=2-7,AH,CH,EH,FH	
	OP3	0 1 1 1	0 0 1 1	1	1	PORT3←A	
	OP54	0 0 1 1 0 0 1 1	1 1 1 1 1 1 0 0	2	2	PORT5←A PORT4←(HL)	
	ANP addr4,n4	0 1 0 0 $I_3\ I_2\ I_1\ I_0$	1 1 0 0 0 $D_2D_1D_0$	2	2	PORT(addr4)←PORT(addr4)∧n4 addr4=2-7,n4=0-FH	
	ORP addr4,n4	0 1 0 0 $I_3\ I_2\ I_1\ I_0$	1 1 0 1 0 $D_2D_1D_0$	2	2	PORT(addr4)←PORT(addr4)∨n4 addr4=2-7,n4=0-FH	
御CPU命令制	HALT	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 0	2	2	Set HALT mode	
	STOP	0 0 1 1 0 0 1 1	1 1 1 1 0 1 1 1	2	2	Set STOP mode	
	NOP	0 0 0 0	0 0 0 0	1	1	No Operation	

保守／廃止

アンケート記入のお願い

お手数ですが，このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

［ドキュメント名］ μPD7500シリーズ　アセンブラ　ユーザーズ・マニュアル
（EEM−647C（第4版），June 1991P）

［お名前など］（さしつかえのない範囲で）

御社名（学校名，その他）　（　　）
ご住所　（　　）
お電話番号　（　　）
お仕事の内容　（　　）
お名前　（　　）

1．ご評価（各欄に○をご記入ください）

項　　目	大変良い	良　い	普　通	悪　い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン，字の大きさなど					
そ　の　他（　　）					
（　　）					

2．わかりやすい所（第　章，第　章，第　章，第　章，その他　　）
理由［　　］

3．わかりにくい所（第　章，第　章，第　章，第　章，その他　　）
理由［　　］

4．ご意見，ご要望

5．このドキュメントをお届けしたのは
NEC 販売員，特約店販売員，NEC 半応技本部員，その他（　　）

ご協力ありがとうございました。
下記あてに FAX で送信いただくか，最寄りの販売員にコピーをお渡しください。

NEC 半導体応用技術本部インフォメーションセンター
FAX：(044)548-7900（直通FAXでの24時間受付）

キリトリ

保守／廃止

保守／廃止

アンケート記入のお願い

お手数ですが，このドキュメントに対するご意見をお寄せください。今後のドキュメント作成の参考にさせていただきます。

［ドキュメント名］ μPD7500シリーズ　アセンブラ　ユーザーズ・マニュアル
（EEM−647C（第4版），June 1991P）

［お名前など］（さしつかえのない範囲で）

御社名（学校名，その他）　（　　）
ご住所　（　　）
お電話番号　（　　）
お仕事の内容　（　　）
お名前　（　　）

1．ご評価（各欄に○をご記入ください）

項　　目	大変良い	良　い	普　通	悪　い	大変悪い
全体の構成					
説明内容					
用語解説					
調べやすさ					
デザイン，字の大きさなど					
そ　の　他（　　）					
（　　）					

2．わかりやすい所（第　章，第　章，第　章，第　章，その他　　）
理由［　　］

3．わかりにくい所（第　章，第　章，第　章，第　章，その他　　）
理由［　　］

4．ご意見，ご要望

5．このドキュメントをお届けしたのは
NEC 販売員，特約店販売員，NEC 半応技本部員，その他（　　）

ご協力ありがとうございました。
下記あてに FAX で送信いただくか，最寄りの販売員にコピーをお渡しください。

NEC 半導体応用技術本部インフォメーションセンター
FAX：(044)548-7900（直通FAXでの24時間受付）

キリトリ

保守／廃止

保守／廃止

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)	
半導体第一、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(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	〒460 名古屋市中区栄四丁目14番5号(松下中日ビル)	名古屋(052)242－2755

北海道支社	札幌	(011)231－0161
東北支社	仙台	(022)261－5511
岩手支店	盛岡	(0196)51－4344
山形支店	山形	(0236)23－5511
福島支店	郡山	(0249)23－5511
茨城支店	いわき	(0246)21－5511
栃木支店	宇都宮	(0286)21－2281
群馬支店	高崎	(0273)26－1255
埼玉支店	大宮	(048)641－1411
千葉支店	千葉	(0472)27－5441
神奈川支店	横浜	(045)324－5511
山梨支店	甲府	(0552)24－4141
長野支店	長野	(0262)35－1444
松本支店	松本	(0263)35－1666
諏訪支店	諏訪	(0266)53－5350
静岡支店	静岡	(054)255－2211
沼津支店	沼津	(0559)63－4455
浜松支店	浜松	(0534)52－2711
金沢支店	金沢	(0762)23－1621
福井支店	福井	(0776)22－1866
富山支店	富山	(0764)31－8461
京都支店	京都	(075)221－8511
神戸支店	神戸	(078)332－3311
広島支店	広島	(082)242－5504
鳥取支店	鳥取	(0857)27－5311
岡山支店	岡山	(0862)25－4455
高松支店	高松	(0878)36－1200
新居浜支店	新居浜	(0897)32－5001
松山支店	松山	(0899)45－4111
九州支社	福岡	(092)271－7700
北九州支店	北九州	(093)541－2887

（技術お問い合せ先）

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)	東京(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 専用マイクロコンピュータ技術部	〒210 川崎市川崎区駅前本町15番5号(十五番館)	川崎(044)246－3923

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