

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日
ルネサスエレクトロニクス株式会社

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

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

ご注意書き



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

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

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

保守 / 廃止

**IE-78C11 システム・ソフト
取扱説明書**



MD シリーズ用

NEC 日本電気株式会社

保守／廃止

IE-78C11 システム・ソフト 取扱説明書

MD シリーズ用

NEC 日本電気株式会社

保守／廃止

は し が き

この取扱説明書は、I E - 7 8 C 1 1 とホスト・マシンMD-080シリーズ、PDA-880もしくはMD-086シリーズを接続しホスト・マシンのコンソールよりI Eを自由に操作可能にするシステム・ソフトウェアについての操作法を述べてあります。

第1章は概説でI E - 7 8 C 1 1 の概要の説明をしてあります。

第2章はI E - 7 8 C 1 1 - Cとして(MD-080シリーズ、PDA-880で) 使用する場合の設置について説明してあります。

第3章はI E - 7 8 C 1 1 - Mとして(MD-086シリーズで) 使用する場合の設置について説明してあります。

第4章は設置後、プログラムのロード、ディバグ方法についての簡単な操作を示してあります。

第5、6、7章には起動から各コマンドの使用法の詳細について説明してあります。

保守/廃止

目 次

第1章 概 説		1-1
1.1 序 文		1-1
1.2 基本仕様		1-1
1.2.1 エミュレータ部		1-3
1.2.2 ブレーク部		1-3
1.2.3 トレース部		1-4
1.2.4 マッピング部		1-4
1.3 IE-78C11のハードウェア仕様		1-5
1.4 特 徴		1-6
1.5 開発手順		1-7
第2章 IE-78C11-Cの設置		2-1
2.1 IEコントロール・ボードの設定		2-1
2.2 IEコントロール・ボードとドライバの接続		2-5
2.3 IEコントロール・ボードの電源端子の接続		2-6
2.4 ターゲット・システムとの接続		2-7
2.4.1 ターゲット・システムとの接続		2-7
2.4.2 エミュレーション・プローブのセッティング		2-7
2.5 MD-080シリーズ, PDA-880との接続		2-9
2.5.1 MD-080FD-10と接続する場合		2-10
2.5.2 MD-080FDと接続する場合		2-16
2.5.3 PDA-880と接続する場合		2-19
2.6 PG-1000, PG-2000を接続する場合		2-22
2.7 IE-78C11システム・ソフトのバックアップ		2-26
2.8 IE-78C11システム・ソフトの起動		2-26
2.9 IE-78C11システム・ソフトの終了		2-27
2.10 2度目以降の起動		2-28
第3章 IE-78C11-Mの設置		3-1
3.1 IE番号の設定		3-1
3.1.1 IE番号とは		3-1

保守/廃止

3.1.2	IE番号の設定	3-1
3.1.3	注意事項	3-2
3.2	IEコントロール・ボードの設定	3-2
3.3	IEコントロール・ボードとドライバの接続	3-6
3.4	MD-086シリーズとの接続	3-7
3.4.1	MD-086FD-10と接続する場合	3-7
3.4.2	MD-086FDと接続する場合	3-8
3.5	ターゲット・システムとの接続	3-10
3.5.1	ターゲット・システムとの接続	3-10
3.5.2	エミュレーション・プローブのセッティング	3-10
3.6	IE-78C11システム・ソフトのバック・アップ	3-12
3.7	IE-78C11システム・ソフトの起動	3-13
3.8	IE-78C11システム・ソフトの終了	3-14
3.9	2度目以降の起動	3-16
 第4章 操作例		 4-1
 第5章 IE-78C11起動時の初期設定		 5-1
5.1	MODE0, MODE1端子の設定	5-1
5.2	μPD78C11およびμPD78C14のディバグ	5-3
5.2.1	内蔵RAMアクセス可否の設定	5-4
5.2.2	ポートDのモード設定	5-4
5.2.3	ポートFのモード設定(表5-1, 表5-2参照)	5-5
5.3	μPD78C/0のディバグ	5-15
 第6章 コマンド形態		 6-1
6.1	コマンドの基本入力形式と表現法	6-1
6.1.1	コマンド入力形式	6-1
6.1.2	マニュアルにおける表現形式	6-2
6.1.3	メイン・コマンド一覧	6-2
6.1.4	サブ・コマンド一覧	6-3
6.1.5	オペランド部一覧	6-3
6.2	オペランドの概要	6-4
6.2.1	式	6-4
6.2.2	数値群	6-4
6.2.3	予約記号	6-4
6.2.4	その他	6-5

保守/廃止

6.3	式の入力	6-5
6.3.1	数値の入力	6-5
6.3.2	シンボルの入力	6-6
6.3.3	演算子付の式の入力	6-8
6.3.4	式入力のオペランド一覧	6-8
6.4	数値群の入力	6-8
6.4.1	partition	6-9
6.4.2	partitions	6-9
6.4.3	addr	6-10
6.4.4	data-string	6-10
6.4.5	Values	6-10
6.5	予約記号	6-10
6.5.1	register-name	6-11
6.5.2	mode-register-name1	6-11
6.5.3	mode-register-name2	6-11
6.5.4	special-register-name1	6-11
6.5.5	special-register-name2	6-11
6.6	その他	6-11
6.6.1	symbol	6-11
6.6.2	command	6-11
6.6.3	filename	6-11
6.6.4	module-name	6-12
6.6.5	filematch	6-12
 第7章 コマンド詳細		 7-1
7.1	ヘルプ・コマンド	7-2
7.2	演算コマンド	7-4
7.3	数値基数指定コマンド	7-6
7.3.1	サフィックス指定	7-7
7.3.2	サフィックス表示	7-8
7.4	マッピング・コマンド	7-9
7.4.1	マッピング指定	7-10
7.4.2	マッピング表示	7-12
7.4.3	マッピング解除	7-14
7.5	シンボル・コマンド	7-16
7.5.1	シンボル表示	7-17
7.5.2	シンボル変更	7-19

保守/廃止

7.5.3	シンボル追加	7-20
7.5.4	シンボル削除	7-22
7.6	モジュール・コマンド	7-23
7.7	リセット・コマンド	7-25
7.8	クロック・コマンド	7-26
7.8.1	クロック指定	7-27
7.8.2	クロック表示	7-28
7.9	メモリ・コマンド	7-29
7.9.1	変 更	7-30
7.9.2	表 示	7-31
7.9.3	比 較	7-32
7.9.4	検 索	7-34
7.9.5	転 送	7-35
7.9.6	イニシャライズ	7-36
7.9.7	交 換	7-37
7.9.8	テ ス ト	7-39
7.10	アセンブル・コマンド	7-40
7.11	逆アセンブル・コマンド	7-43
7.12	メモリ転送コマンド	7-44
7.12.1	ユーザ・システムからエミュレーション・メモリへのメモリ転送	7-45
7.12.2	エミュレーション・メモリからユーザ・システムへのメモリ転送	7-47
7.13	ロード・コマンド	7-48
7.14	セーブ・コマンド	7-50
7.15	ベリファイ・コマンド	7-52
7.16	レジスタ・コマンド	7-53
7.16.1	変 更	7-54
7.16.2	表 示	7-56
7.17	スペシャル・レジスタ・コマンド	7-57
7.17.1	変 更	7-58
7.17.2	表 示	7-59
7.18	モード・レジスタ・コマンド	7-60
7.18.1	変 更	7-61
7.18.2	表 示	7-62
7.19	ブレイク・コントロール・コマンド群	7-63
7.19.1	BRA コマンド	7-64
7.19.2	BRD コマンド	7-66
7.19.3	BRE コマンド	7-67
7.19.4	BRT コマンド	7-68

保守/廃止

	7.19.5 BR0～BR3 コマンド	7-69
	7.19.6 BRM コマンド	7-70
7.20	トレース・コマンド群	7-71
	7.20.1 TRX コマンド	7-72
	7.20.2 TRM コマンド	7-73
	7.20.3 TRD コマンド	7-74
	7.20.4 TRP コマンド	7-78
	7.20.5 TRC コマンド	7-80
	7.20.6 TRS コマンド	7-82
7.21	エミュレーション・コマンド	7-84
	7.21.1 通常リアルタイム・エミュレーション	7-85
	7.21.2 ブレーク条件付きエミュレーション	7-87
	7.21.3 指定ステップ数実行	7-89
	7.21.4 1ステップ・エミュレーション	7-90
7.22	ストリング・コマンド	7-93
7.23	自己診断コマンド	7-94
7.24	ディレクトリ・コマンド	7-96
7.25	リスト・コマンド	7-97
7.26	終了コマンド	7-98
7.27	ページ・コマンド	7-99
7.28	PGM コマンド	7-100
付 録		付-1
付.1	IE-78C11-Cコントロール・キー一覧	付-1
付.2	IE-78C11-Cメッセージ一覧	付-2
付.3	IE-78C11-Mコントロール・キー一覧	付-5
付.4	IE-78C11-Mメッセージ一覧	付-6
付.5	予約語一覧	付-9
付.6	コマンド一覧	付-12

保守/廃止

第1章 概 説

1.1 序 文

IE-78C11システム・ソフトはMD-080シリーズ，PDA-880またはMD-086シリーズ上でIE-78C11を直接操作できるようにするためのサポート・ソフトウェアです。

IE-78C11-Cは、MD-080シリーズ^{注1}あるいはPDA-880のCP/MTM上で動作し、IE-78C11とMD-080シリーズ，PDA-880をシリアル回線で結びMD-080シリーズ，PDA-880のコンソールより制御できます。

また、IE-78C11-Mは、MD-086シリーズとIE-78C11をIEEE 796バスを介して接続し、MD-086(MP/M-86TM)のコンソールからIE-78C11を制御することができます。

これによりμPD78C10，78C11，78C14のソース・プログラムの作成からアSEMBル，デバッグまでのすべての操作がMD-080シリーズ，PDA-880またはMD-086シリーズのコンソール上で可能となり設計の効率を大幅に改善することができます。

注1 MD-086シリーズのCP/M上でも動作します。

2 CP/MTMは、米国デジタル・リサーチ社の商標です。

3 MP/M-86TMは、米国デジタル・リサーチ社の商標です。

1.2 基本仕様

IE-78C11仕様

対象CPU	μPD78C10，μPD78C11，μPD78C14	
動作CPUクロック	最大15MHz	
メモリ マップ	ユーザ・メモリ	0～FFFFH(64Kバイト)のアドレス空間を256バイト単位で、マッピング可能(リアルタイム実行)
	エミュレーション・メモリ (標準装備)	物理的アドレス空間(64Kバイト)を256バイト単位でマッピング可能(リアルタイム実行)

保守/廃止

ブ レ ー ク	ブレーク・レジスタ	<ol style="list-style-type: none"> 1. ロジカル・ブレーク・レジスタ BR0～BR3 (4種類) BR0～BR3の中でさらにフィジカル・ブレーク・レジスタの設定が可能 2. フィジカル・ブレーク・レジスタ <ol style="list-style-type: none"> 1) BRA (アドレス, コンディション, データの設定が可能) 2) BRD (外部センス・クリップのデータでブレーク) 3) BRE (オペコード・フェッチの回数) 4) BRT (ブレーク・タイマ (ms)) 3. ブレーク・モード・レジスタ BRM (各ブレーク・レジスタをハードウェアにセット)
	設定条件	<ol style="list-style-type: none"> 1. アドレス ポイント数 (複数箇所), マスク可, 範囲指定可 (64Kバイト内) 2. コンディション オペコード・フェッチ (M1リード), メモリ・リード, メモリ・ライト, メモリ・リクエスト, 条件なし 3. データ, 外部センス・データ ポイント数 (複数箇所), マスク可 4. 命令回数 (3H～FFFFH) 5. ループ回数 (1H～FFH) 6. タイマ (1H～FFFFHms)
	ブレーク要因	<ol style="list-style-type: none"> 1. アドレス 2. データ 3. コンディション (オペコード・フェッチ (M1リード), メモリ・リード, メモリ・ライト, メモリ・リクエスト) 4. 外部センス・データ 5. インストラクション・カウンタ 6. タイマ 7. MAP違反 (マッピング領域以外をアクセスしてブレーク) 8. 強制ブレーク (キー入力によるブレーク)
	トレース容量	1023フレーム
	トレース表示	インストラクション/マシン・サイクルの2種類選択可能

保守/廃止

ト レ ー ス	トレース条件	下記のいずれかを選択 1) NON (トレースしない) 2) ALL (エミュレーションCPUの実行をすべてトレース) 3) TRX (TRXで指定された条件によりトレース)
	トレース内容	アドレス (16本) / データ (8本) / $\overline{M1}$ / \overline{RD} / \overline{WR} / PA / PB または 外部センスデータ (8本)
外部トレース・ブローブ		8本
シリアル・チャンネル		2チャンネル (RS-232-C) 110~9600ボー 1. TTY1 (コンソール用) 2. TTY2 (PG用)

1.2.1 エミュレータ部

- | | |
|--|---|
| (1) エミュレーション・ブローブ | 64ピン・ICソケット挿入形式を使用 |
| (2) CPU | μ PD78C10 |
| (3) 信号ピン・インタフェース | HCMOSのパッファ付 (PD, PF, \overline{RD} , \overline{WR} , ALE) |
| (4) コントロール出力信号 | エミュレーションCPU実行中はリアル・タイムで
コントロール |
| (5) \overline{RESET} , \overline{NMI} 信号入力 | 完全エミュレート |
| (6) CPU動作クロック | 4MHz ~ 12MHz (内部クロック指定時は, 12
MHz 固定) |
| (7) MODE0, MODE1 | 完全エミュレート |

1.2.2 ブレーク部

ブレーク機能は、すべてハードウェアにより実現されています。

- | | |
|------------|---------------|
| (1) ブレーク状態 | 内部モニタ・プログラム実行 |
| (2) ブレーク機能 | 6種類 |
- 1) 3組の独立したアドレス、データ、コントロール信号の^注組み合わせによるブレーク (ループ回数設定可能)
 - 2) 指定メモリ領域以外のアクセスによるブレーク
 - 3) 8本の外部センス信号の条件設定によるブレーク
 - 4) 内部タイマによるブレーク
 - 5) コンソールからの強制ブレーク
 - 6) マルチ・ディバッガ時の他のディバッガからのブレーク・トリガ信号によるブレーク

注 オペコード・フェッチ ($\overline{M1}$ リード), メモリ・リード, メモリ・ライト

保守/廃止**1.2.3 トレース部**

(1) 三つのトレース・モード

1) ノン・トレース

2) インストラクション毎のトレース

3) マシン・サイクル毎のトレース

(2) 二つのトレース機能

1) 常時トレース

2) 指定されたアドレス範囲のトレース

(3) トレース項目

1) アドレス・バス, データ・バス

2) $\overline{M\bar{I}}$, $\overline{R\bar{D}}$, $\overline{W\bar{R}}$

3) P A, P B または 8 本の外部センス信号

1.2.4 マッピング部

(1) マッピング状態

1) ノン・マッピング (初期状態ではプログラム・メモリはノン・マッピング)

2) 外部マッピング (ユーザ)

3) 内部マッピング (エミュレーション・メモリ)

(2) マッピング対象

6 4 K バイト全域

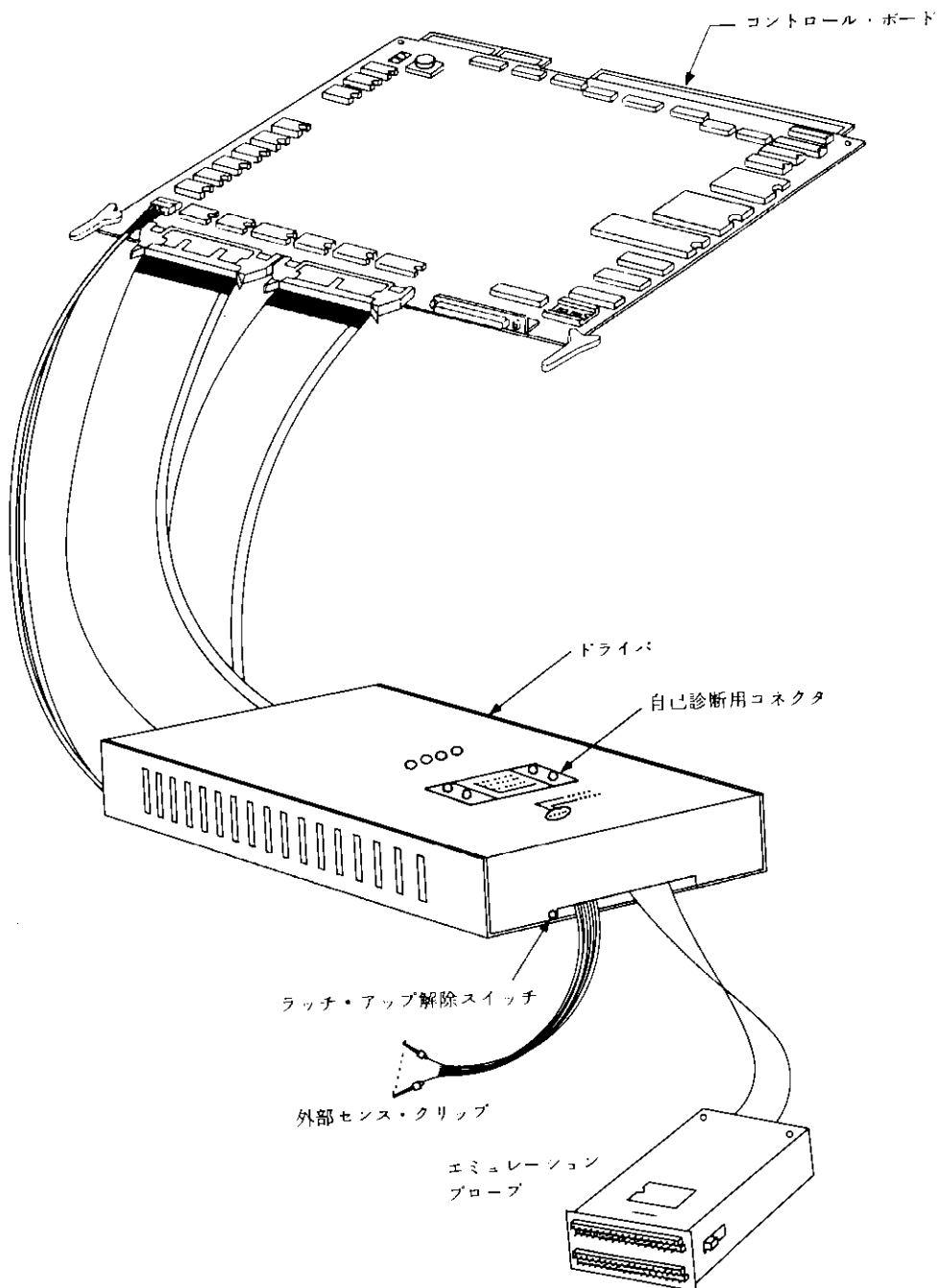
(3) マッピング単位

2 5 6 バイト単位

保守/廃止

1.3 IE-78C11のハードウェア仕様

図1-1 IE-78C11 外観図



保守/廃止

(1) 外形寸法

IEコントロール・ボード	縦	250 mm
	横	305 mm
	高	約20 mm
IE-78C11ドライブ	縦	400 mm
	横	230 mm
	高	48 mm

(2) 重量

IEコントロール・ボード	550 g
IE-78C11ドライブ	2.6 kg

(3) 所要電源

6.5 A以上 (+5 V)
500 mA以上 (± 12 V)

(4) 使用温度範囲

10 ~ 40 °C

(5) 保存温度範囲

-40 ~ +55 °C

(6) 周囲湿度範囲

10 ~ 90% (ただし結露しないこと)

1.4 特 徴

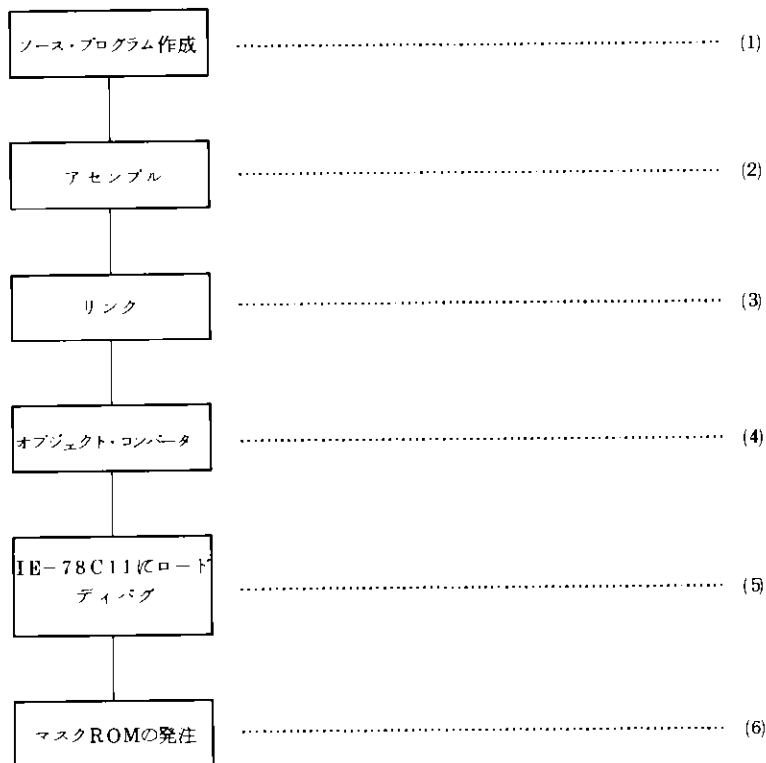
IE-78C11-C, IE-78C11-Mの機能として以下のものがあります。

- シンボリック・デバッグ機能
- シンボル・ファイルのロード, セーブ機能
- シンボルの追加, 変更, 削除, 表示機能
- オブジェクト・ファイルのロード, セーブ機能
- オンライン・アセンブル機能
- 逆アセンブル機能
- 実行結果のファイル・セーブ機能
- コマンドの自動実行機能

保守/廃止

1.5 開発手順

プログラムの開発手順として下図に示します。



- (1) エディタ等によりソース・プログラムを作成します。……………注
- (2) RA87 (アセンブラ) を使用し、(1)で作成したソース・プログラムをアセンブルします。
- (3) LK87 (リンカ) を使用し、(2)で作成されたRELファイルをリンクします。
- (4) OC87 (オブジェクト・コンバータ) を使用しシンボル・ファイルおよびヘキサ・オブジェクト・ファイルを作成します。
- OC87を使用する際に“NEW”オプションを指定してください。
- (5) (1)~(4)の作業により作成されたシンボル・ファイル、およびヘキサ・オブジェクト・ファイルをIE-78C11によりロード (LODコマンド使用)、デバッグを行います。
- (6) マスクROMを発注します。

注 IE-78C11で、SYMコマンドにより、シンボルを登録した場合APPENDというモジュール名で登録されるため、APPENDというモジュール名は使用できません。

保守/廃止

第2章 IE-78C11-Cの設置

この章では、IE-78C11-Cを起動させ簡単なオペレーションを行なうまでの手順を説明します。

2.1 IEコントロール・ボードの設定

IE-78C11コントロール・ボード上のJP1～JP7およびDP1は、MD-080シリーズ^注、PDA-880の使用に応じて図2-1～10および表2-1に従って設定してください。

注 MD-086シリーズのCP/M上での使用も可能です。

(その場合のIEコントロール・ボードの設定はMD-080シリーズと同様にしてください)

保守/廃止

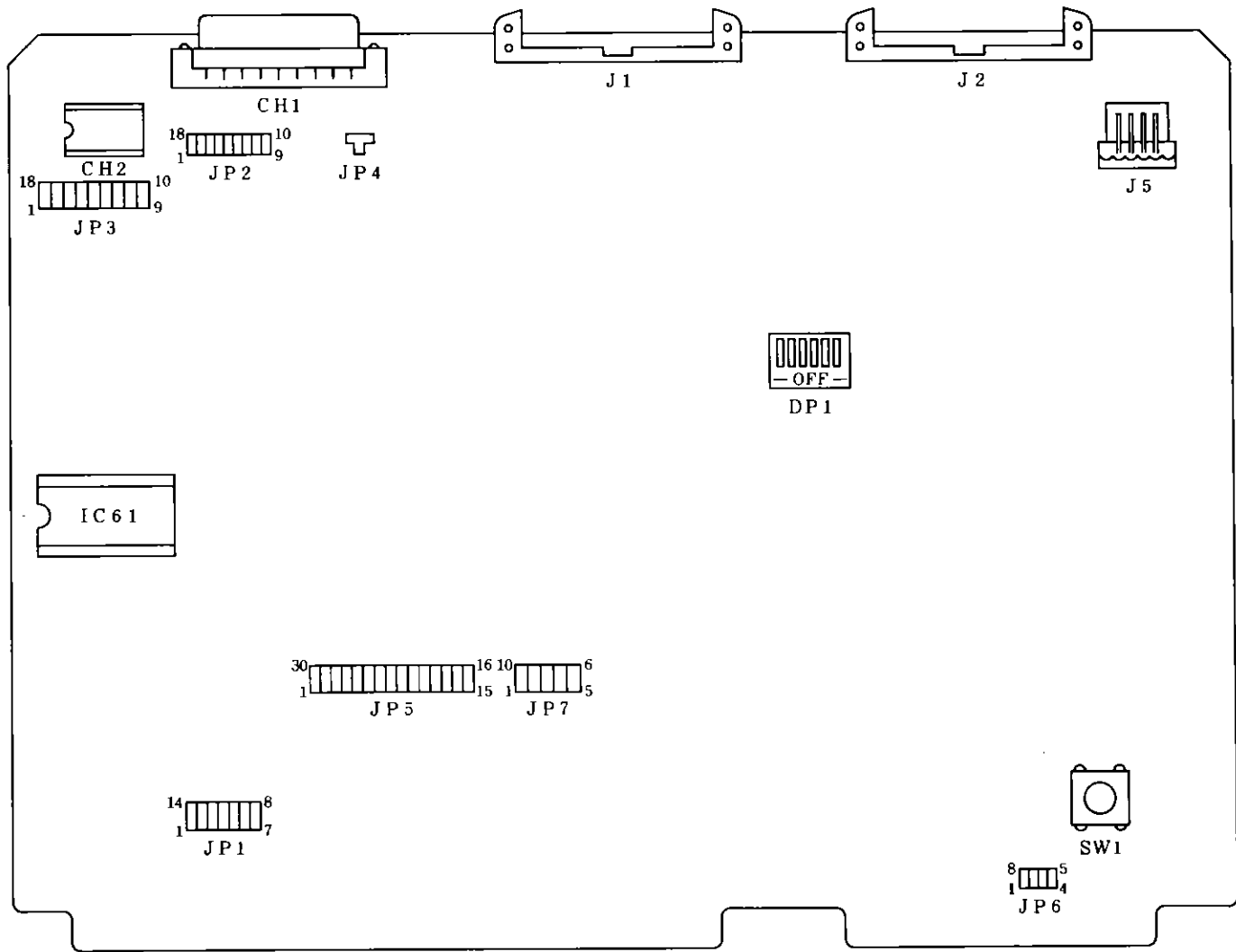


図 2-1 IE-78C11コンピュータボード

保守/廃止

図 2-2 JP1

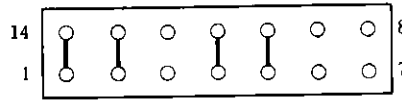


図 2-3 JP2 MD-080 シリーズと接続する場合

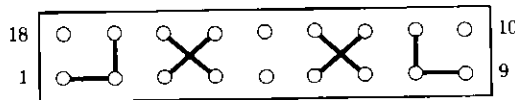


図 2-4 PDA-880 と接続する場合

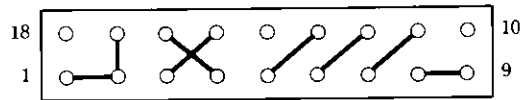


図 2-5 JP3

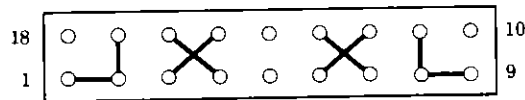


図 2-6 JP4

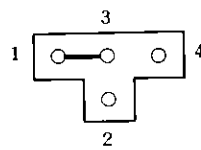
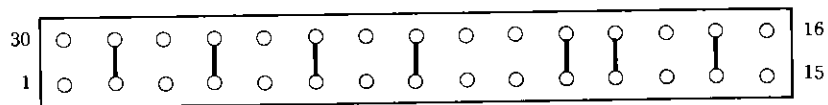


図 2-7 JP5



保守/廃止

図 2-8 JP6

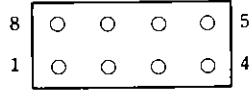


図 2-9 JP7

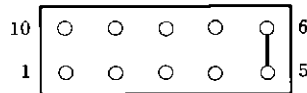
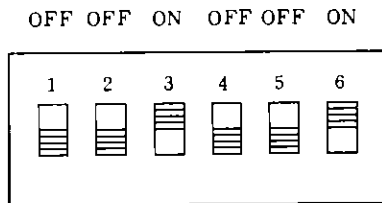


表 2-1 DP1 (ディップ・スイッチ) とボーレートの関係

ボーレート	1	2	3	4	5	6
9600	OFF	OFF	OFF	OFF	OFF	ON
4800	OFF	OFF	ON			
2400	OFF	ON	OFF			
1200	OFF	ON	ON			
600	ON	OFF	OFF			
300	ON	OFF	ON			
110	ON	ON	OFF			

MD-080シリーズのボーレートは可変ですが、PDA-880の場合は4800ボーに固定して
ください(図2-10参照)。

図 2-10

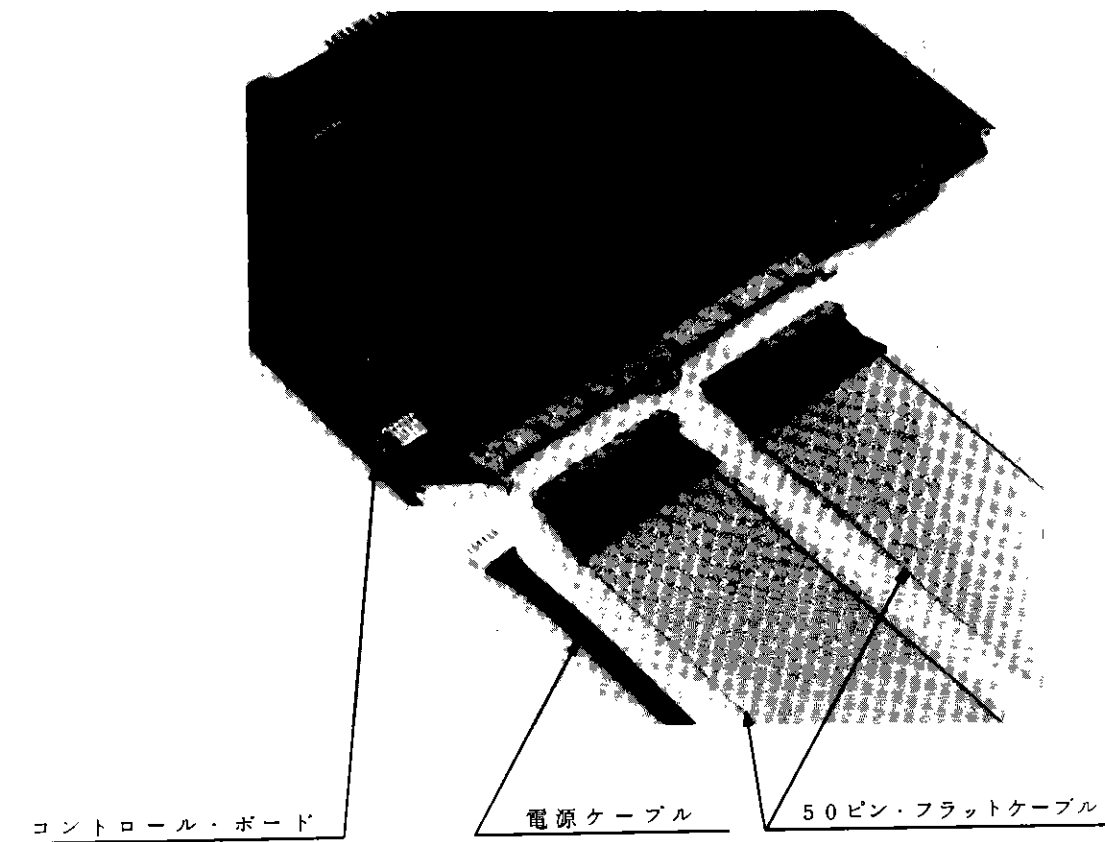


4800ボー選択例

保守/廃止**2.2 IEコントロール・ボードとドライバの接続**

- (1) IE-78C11ドライバからでている二つJ1, J2の50ピン・フラット・ケーブルの先のコネクタを、コントロール・ボード側の対応する“J1”, “J2”と名前のついたソケットに装着します。
- (2) 同様にIE-78C11ドライバよりでている電源ケーブルをコントロール・ボードのコネクタ(J5)に接続します。

図2-11



保守/廃止

2.3 IEコントロール・ボードの電源端子の接続

IEEE 796バスのP1ソケットより+5V, +12V, -12Vを供給してください。+5Vは電流容量6.5A以上, ±12Vは電流容量500mA以上の電源を利用してください。

図2-12

表2-2 電源の色別表

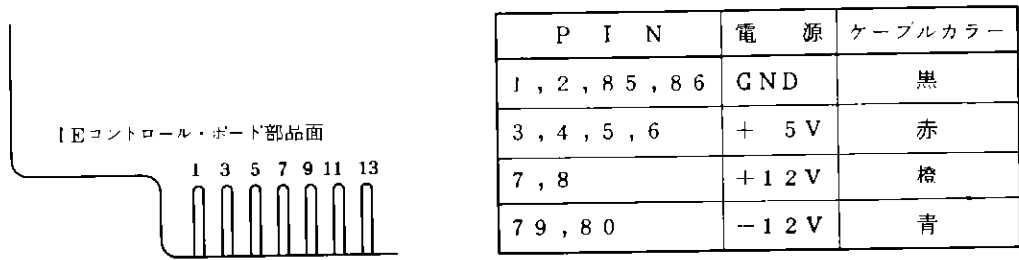
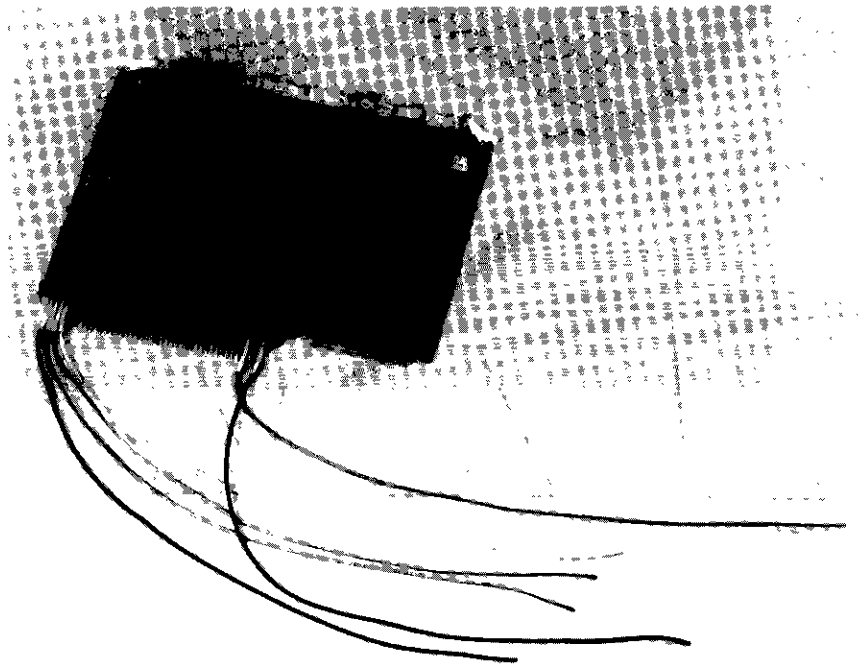


図2-13



保守/廃止

2.4 ターゲット・システムとの接続

2.4.1 ターゲット・システムとの接続

ターゲット・システムと IE-78C11 との接続は以下に述べる順序で行なってください。

- (1) IE-78C11 とターゲット・システムの電源を OFF にします。
- (2) エミュレーション・プローブをターゲット・システムの 64 ピン・ソケットに挿入してください。

ターゲット・システムがない場合でソフトウェアの開発、デバッグを行なう場合にはエミュレーション・プローブを IE-78C11 ドライバの自己診断用コネクタに挿入してください。

(CPU クロックは IE-78C11 内部のクロックを使用します)。

IE-78C11 取扱い上の注意点

- エミュレーション・プローブの MODE0, MODE1 端子は IE-78C11 内部でオープン・コレクタの出力と接続されていますから直接 Vcc につなぐと過電流が流れ誤動作したり、破壊したりする恐れがありますので、必ず抵抗 (4.7 kΩ 程度) でプルアップしてください。

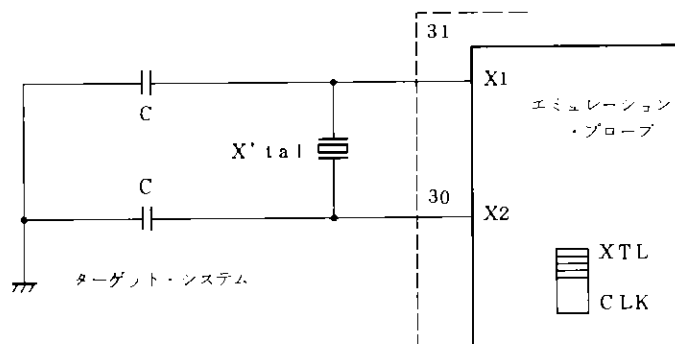
2.4.2 エミュレーション・プローブのセッティング

IE-78C11 のエミュレーション・プローブには水晶発振用の回路が内蔵されており、ターゲット・システムの発振回路を CPU クロックとして使用するには発振回路に合わせてエミュレーション・プローブの切換えスイッチを設定しなければなりません。

ここで、ターゲット・システムの発振回路を使用せずに IE-78C11 の内部クロック (12 MHz) を使用する場合は切換えスイッチの設定はどちらでもかまいません。

- 1) ターゲット・システム上の発振回路が水晶発振子 (X'tal) だけで構成されている場合はスイッチを 'XTL' 側にしてください。

図 2-14

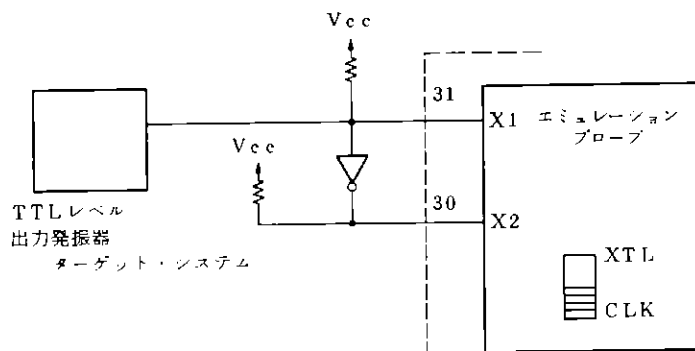


エミュレーション・プローブの切換えスイッチを 'XTL' 側にします。

保守/廃止

- 2) ターゲット・システム上の発振回路がTTLレベル出力の発振器で構成されている場合はスイッチを'CLK'側にしてください。

図 2 - 15



エミュレーション・プローブの切換えスイッチを'CLK'側にします。

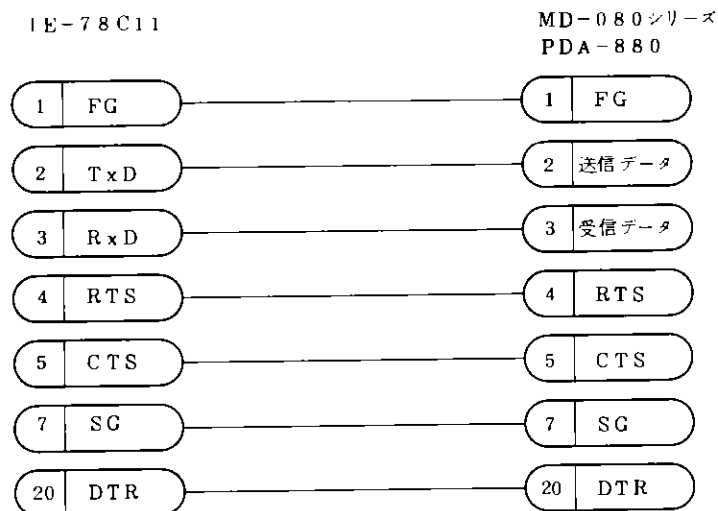
保守/廃止

2.5 MD-080 シリーズ, PDA-880 との接続

MD-080 シリーズまたは PDA-880 と接続するケーブルは添付のシリアル・ケーブル I を用いてください。

ケーブルの結線情報は下図のとおりとなっています。

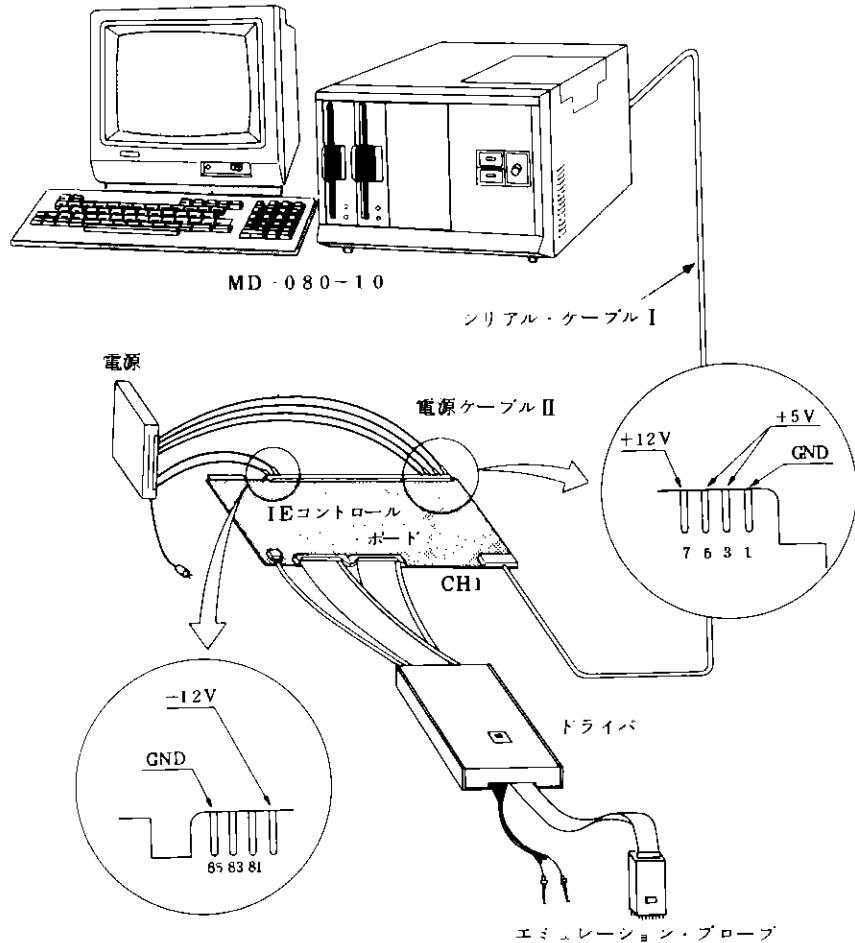
図 2-16 ケーブル結線情報



保守/廃止

2.5.1 MD-080FD-10と接続する場合

図2-17 MD-080FD-10とIE-78C11の接続図



MD-080FD-10で使用するシリアル・チャンネル(1~4)を選択します。
 次に選択したシリアル・チャンネルに沿ってMD-080FD-10のI/O インタフェース・ボードの設定をしてください。

(1) ジャンパ

ボーレートの選択, RS-232-C/TTLの選択, モデム/端末モード等の選択を行なうためのもので標準シリアル・インタフェース用として, J1~4, J9~24があります。
 各インタフェースの各機能毎のジャンパは表2-4のようになります。
 また, 図2-18にそれぞれのジャンパの基板上的位置を示します。

保守/廃止

表 2 - 4

シリアル・チャンネルNo.	CH 1	CH 2	CH 3	CH 4
RS-232-C/TTL	J 1 0	J 1 5	J 1 3	J 9
ボーレート・セレクト	0	1	2	3
モデム/端末セレクト	J 1 6 , J 2 2	J 1 4 , J 2 4	J 1 2 , J 2 3	J 1 1 , J 1 7
RTSセレクト	J 4	J 3	J 2	J 1
フレーム・グラント	J 2 1	J 2 0	J 1 9	J 1 8

保守/廃止

(2) RS-232-C/TTL選択ジャンパ

シリアル・インタフェースは、RS-232-Cを選択します。
 使用チャンネルに対応するジャンパをオープンにしてください。

表 2 - 5

シリアル・チャンネルNo.	ジャンパ
CH 1	J 1 0
CH 2	J 1 5
CH 3	J 1 3
CH 4	J 9

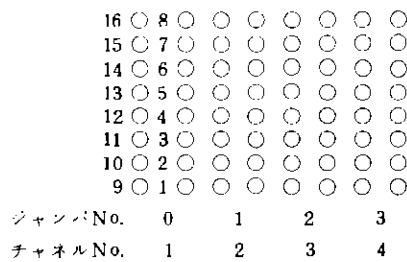
(3) ボーレート・セレクト・ジャンパ

下図のようなピン配置となっており9～16は受け側、ピン1～8がクロック入力側です。
 ここでは必要とするボーレートに応じて1組を接続してください。

2組以上接続させると誤動作、およびゲート破壊の原因となりますので必ず現在セットされているジャンパをはずし再度設定してください。

注 ボーレートはIE-78C11と同一のボーレートに設定してください。

図 2 - 1 9



各ボーレートとジャンパの関係は以下のようになります。

表 2 - 6

ジャンパ	1-9	2-10	3-11	4-12	5-13	6-14	7-15
ボーレート	9600	4800	2400	1200	600	300	150

保守/廃止

(4) モデム/端末セレクト・ジャンパ

使用チャンネルを図2-20のようにモデム・モードに設定してください。

図2-20

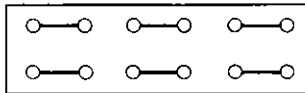


表2-7

シリアル・チャンネル No.	モード・セレクト・ジャンパ
CH1	J16, 22
CH2	J14, 24
CH3	J12, 23
CH4	J11, 17

(5) RTSセレクト・ジャンパ

モデム・モードにおけるCTS出力に、USART(: μPD8251AF)のRTSにRxRDYの条件を入れて、接続します。

ジャンパの1ピンと2ピンをショートしてください(表2-8参照)。

表2-8

シリアル・チャンネル No.	ジャンパ
CH1	J4
CH2	J3
CH3	J2
CH4	J1

保守/廃止

(6) フレーム・グラウンド・ジャンパ

フレーム・グラウンド信号をグラウンドに接続します(表2-9参照)。

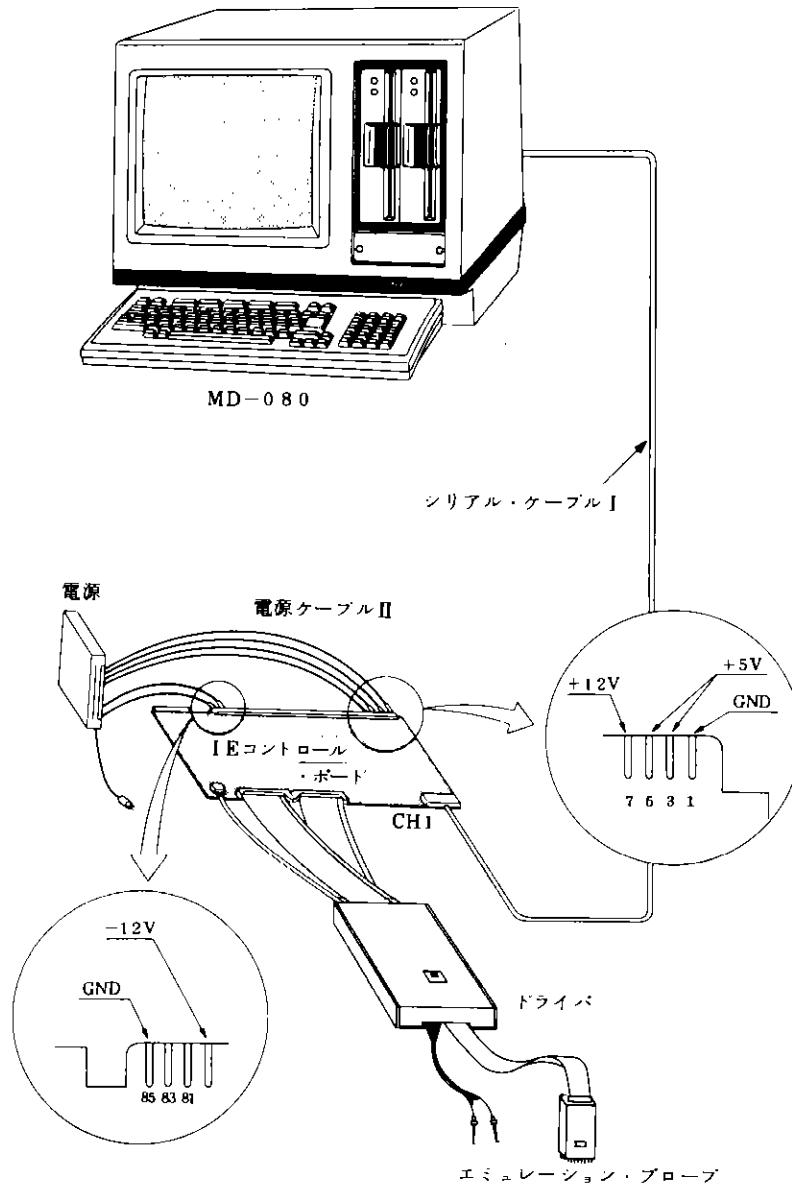
表2-9

シリアル・チャンネル No.	ジャンパ
CH 1	J 2 1
CH 2	J 2 0
CH 3	J 1 9
CH 4	J 1 8

保守/廃止

2.5.2 MD-080FDと接続する場合

図 2-21 MD-080FDとIE-78C11の接続図



保守/廃止

MD-080FDで使用するシリアル・チャンネル(1~4)を選択します。次に選択したシリアル・チャンネルに沿ってMD-080FD-10のI/Oインタフェース・ボードの設定をしてください。

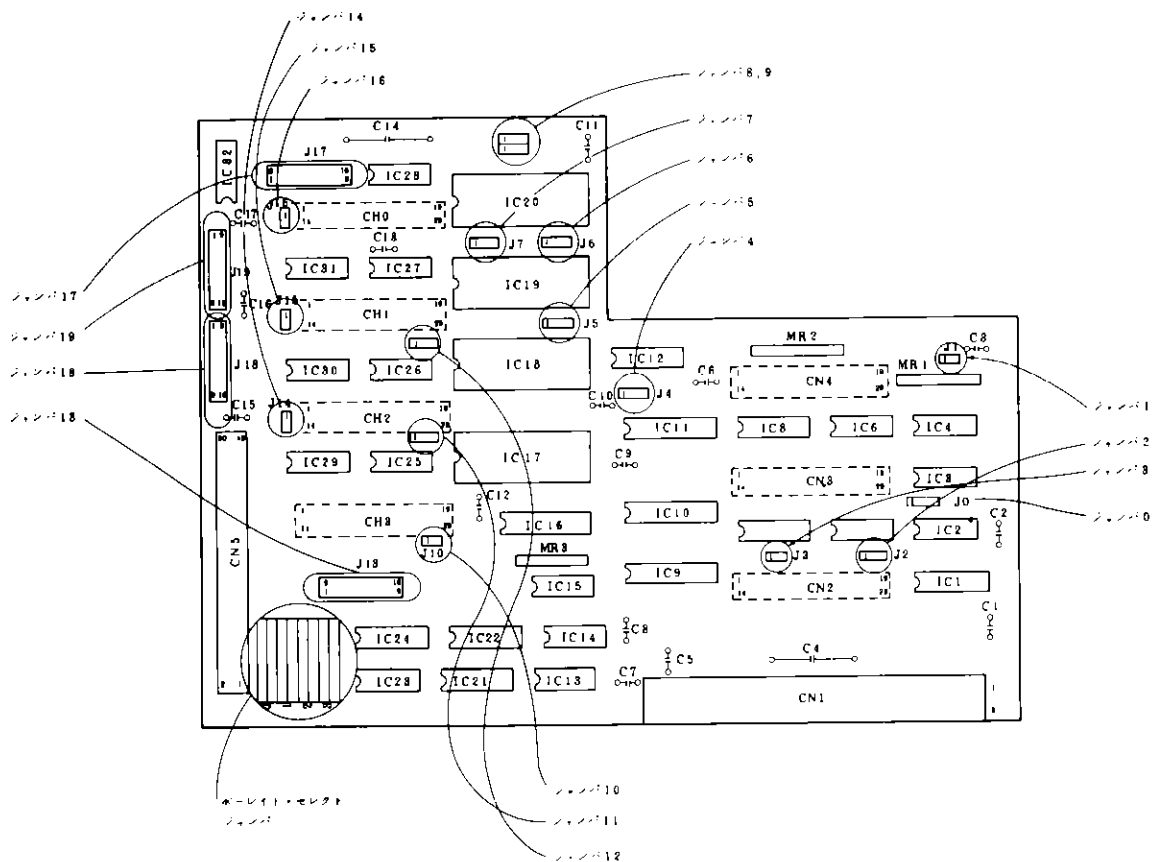
(1) チャンネル・ジャンパの設定

各チャンネルの各機能毎のジャンパは以下のようになります。また、図2-22にはジャンパ配線図を示します。

図2-10 ジャンパの構成

シリアル・チャンネルNo.	CH 1	CH 2	CH 3	CH 4
ボーレート・セレクト	0	1	2	3
モデム・モード・セレクト	J 17	J 19	J 18	J 13
RTSセレクト	J 9	J 7	J 5	J 4
フレーム・グラウンド	J 16	J 15	J 14	J 10
外部クロック	J 8	J 6	J 12	J 11

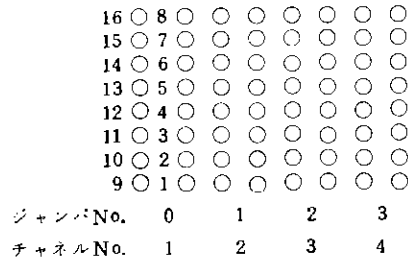
図2-22 ジャンパ配置図



保守/廃止

(2) ボーレート・セレクト・ジャンパ

図 2-23



上図（図 2-23）のような配置およびピン配置となっており、ピン 9～16 は受け側、ピン 1～8 がクロック入力側です。

ここでは必要とするボーレートに応じて 1 組を接続してください。

2 組以上接続させると誤動作およびゲート破壊の原因となりますので、必ず現在セットされているジャンパをはずし、再度設定してください。

注 ボーレートは IE-78C11 と同一のボーレートに設定してください。

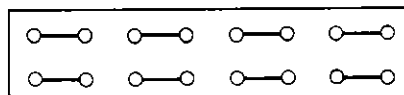
表 2-11

ジャンパ	1-9	2-10	3-11	4-12	5-13	6-14	7-15
ボーレート	9600	4800	2400	1200	600	300	150

(3) モデム・モード・ジャンパ

チャンネルはモデム・モードとして使用します。

図 2-24



(4) RTSセレクト・ジャンパ

モデム・モードにおける RTS 出力に USART の RxRDY を接続します。

2 ピンと 1 ピンをショートすると RxRDY が選択されます。

保守/廃止

- (5) フレーム・グラウンド・ジャンパ
フレーム・グラウンド信号をグラウンドに接続するために2ピンと1ピンをショートします。
- (6) 外部クロック・ジャンパ
USARTの受信ロックを内部クロックから供給するために2ピンと3ピンをショートします。

注 意

初期のMD-080FDにはシリアル・チャンネル・ナンバーが0～3までのものがありますが、チャンネル・ナンバーを1～4に置き換えて使用してください。

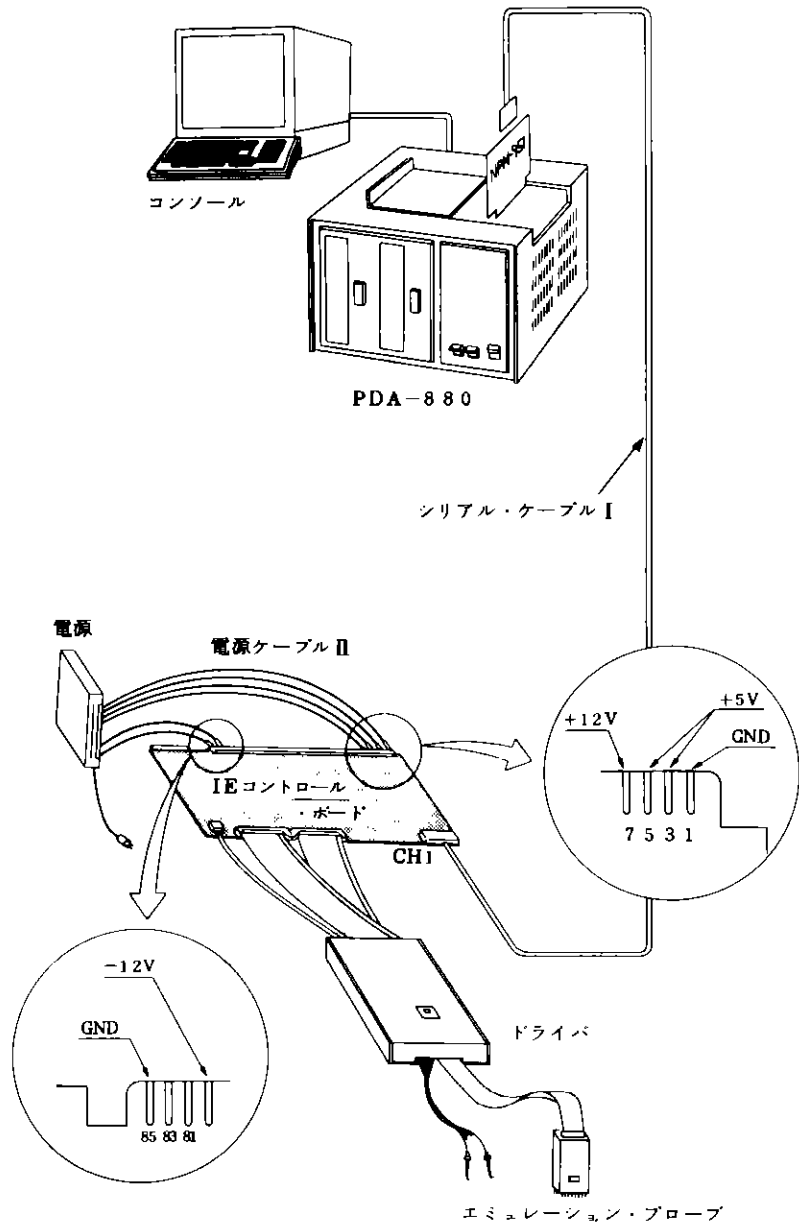
2.5.3 PDA-880と接続する場合

4800ボーを使用するためIE-78C11のJ5を4800ボーに設定してください。

IE-78C11のCH1からのケーブルを汎用シリアル・ポート・モジュール(NPM-951)のチャンネルに接続してください。

保守/廃止

図2-26 PDA-880とIE-78C11の接続図



保守/廃止

2.6 PG-1000, PG-2000 を接続する場合

図2-27 PG-1000, PG-2000接続図

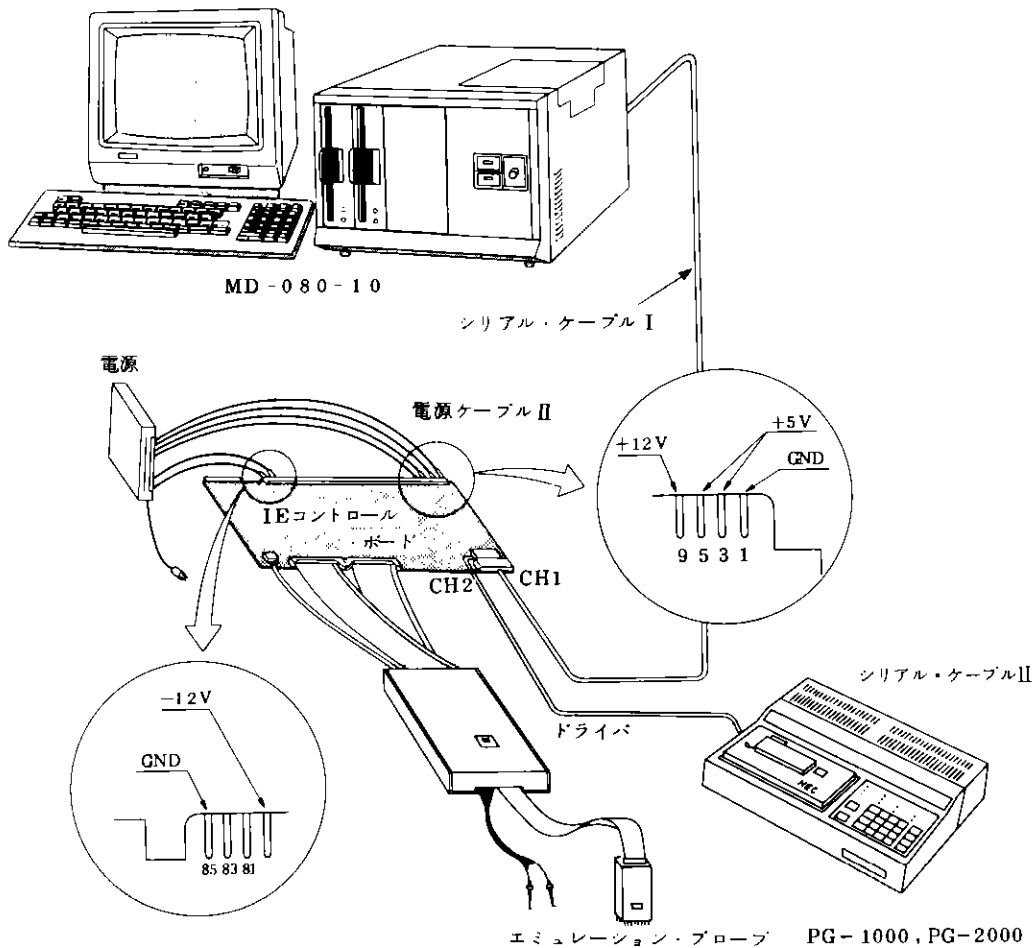


図2-27に示すようにIE-78C11コントロール・ボードのCH2にPG-1000, PG-2000を接続します。

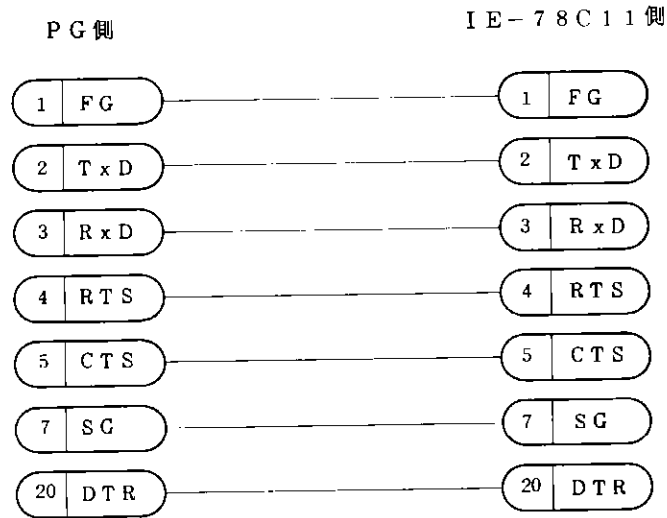
このときPG-1000, PG-2000側のボーレートはIE-78C11コントロール・ボードのボーレートと同一でなければなりません。

PG-1000, PG-2000とIE-78C11を接続するケーブルは、添付のシリアルケーブルⅡ(CH2用16ピン・ソケット)を用いてください。

結線は次図のとおりです。

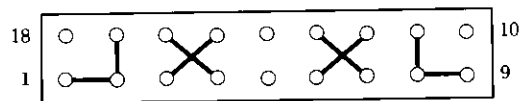
保守/廃止

図 2-28 ケーブル結線情報



コントロール・ボードの JP3 は次のように設定してください。

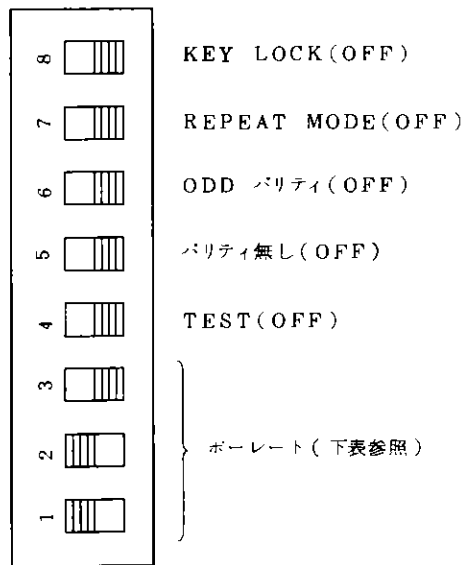
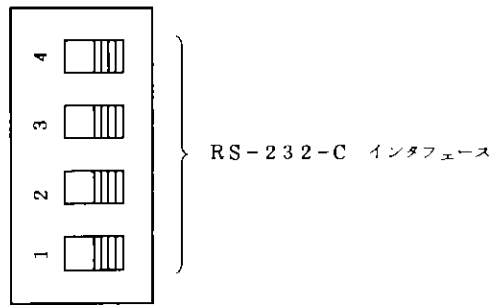
図 2-29 JP3 のジャンパ設定



保守/廃止

以下にPG-1000, PG-2000の設定を図示します。

(1) PG-1000の場合



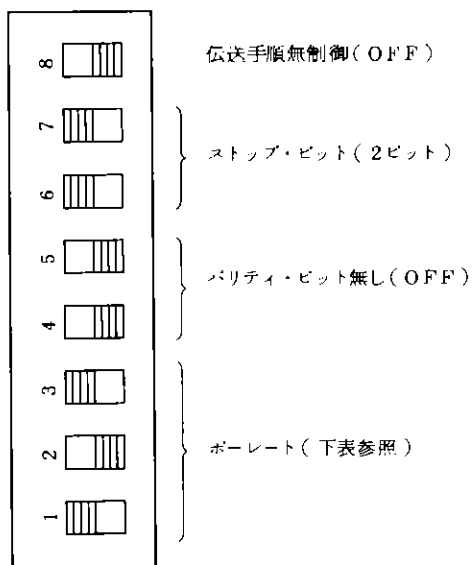
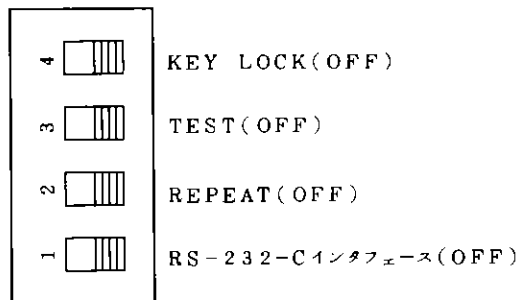
ボーレート スイッチ 番号	110	300	600	1200	2400	4800	9600	-
3	OFF	ON	OFF	ON	OFF	ON	OFF	ON
2	OFF	OFF	ON	ON	OFF	OFF	ON	ON
1	OFF	OFF	OFF	OFF	ON	ON	ON	ON

注

	ON
	OFF

保守/廃止

(2) PG-2000の場合



ボーレート スイッチ 番号	110	300	600	1200	2400	4800	9600	-
3	OFF	OFF	OFF	OFF	ON	ON	ON	ON
2	OFF	OFF	ON	ON	OFF	OFF	ON	ON
1	OFF	ON	OFF	ON	OFF	ON	OFF	ON

注 ON
 OFF

保守/廃止

2.7 IE-78C11システム・ソフトのバックアップ

IE-78C11システム・ソフト(プログラム名: IE78C11.COM)がはいっているフロッピー・ディスクには、CP/Mシステムおよびユーティリティははいっておりません。そのため、IE-78C11システム・ソフトをそのまま使用すると、CP/MシステムおよびIE-78C11システムにそれぞれ1台のドライブが必要となります。

以上のことや、システムの暴走や不慮の事故によるプログラムの破壊に対応するため、CP/Mシステム・フロッピー・ディスクを新規に作成し、IE-78C11システム・ソフト一式をコピーして使用されることをお勧めいたします。

以下に、IE-78C11システムのマスタ・フロッピー・ディスクをBドライブにセットし、新規に作成したシステム・フロッピー・ディスクをAドライブにセットした場合のコピー方法と、IE-78C11に必要なファイルの一覧を示します。

```
A>PIP A:=B:*,* [R] )
```

```
COPYING
IE78C11.COM
IE87.HLP
EX.STR
EXO.STM
EXO.HEX
```

```
A>
```

ファイル名一覧

IE78C11.COM	システム・ソフトです。
IE87.HLP	ヘルプ・ファイルです。
EX.STR	4章で使用する自動実行用のファイルです。
EXO.SYM	4章で使用するシンボル・ファイルです。
EXO.HEX	4章で使用するヘキサ・ファイルです。

注 ファイルの属性はSYS, ROとなっています。

2.8 IE-78C11システム・ソフトの起動

ターゲット・システム、IE-78C11の順に電源を投入し、以下のようにして起動させてください(アンダーライン部は、キー入力する部分です)。

```
A>IE78C11 )
```

```
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright(c) xxxx NEC Corporation
```

```
Select Channel Number ? (1/2/3/4) : 1 (1)
Initialize ? (Y/N) : Y (2)
```

```
I>
```

保守/廃止

ハードウェアが正しく接続されていない場合や、電源が入っていない場合は、下のよう出力されます。

A>IE78C11)

```
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright(c) xxxx NEC Corporation
```

Select Channel Number ? (1/2/3/4) : 1) (1)

No Connect _____ (3)

Abort ? (Y/N) : N) (4)

Select Channel Number ? (1/2/3/4) :

(1) PDA-880, MD-080シリーズの接続チャンネル

(2) 5.1, 5.1.1 参照

起動(POWER ON)時には、必ず'Y'と入力し、イニシャライズを行なってください。

イニシャライズを行わない場合、IEのメモリのマッピングが不定になります。

(3) ケーブルの接続チャンネルNo. が間違っている場合やIE-78C11の電源がOFFになっている場合に出力されます。

(4) 'Y'を入力した場合はCP/Mに戻ります。

2.9 IE-78C11システム・ソフトの終了

IE-78C11システム・ソフトを終了する場合、コマンド入力状態において'EXT)'と入力します。

I> RUN B)

```
USER-SYSTEM VDD-ON,VCC-ON
```

```
JUST MOMENT
EMULATION START AT 0000
```

```
NORMAL BREAK
TERMINATED
```

```
PSW V A B C D E H L EA SP PC
00 00 00 00 00 00 00 00 0000 014C 05D0
V' A' B' C' D' E' H' L' EA'
00 00 00 00 00 00 00 00 0000
```

```
ONE STEP EMULATION STANDBY )
```

I> EXT)

A>

二つ以上のCPUをくみあわせたマルチCPUシステムのデバッグ等の有効な方法としてエミュレーションCPUを実行させたまま終了させることもできます。

保守/廃止

```

I> RUN N )
USER-SYSTEM VDD-ON,VCC-ON
JUST MOMENT
EMULATION START AT 0000
          _____ CTRL C入力
E-CPU RUN
I> EXT )
A>

```

注 電源はターゲット・システム，IE-78C11の順で遮断してください。

2.10 2度目以降の起動

IE-78C11システム・ソフトを一度終了させた後に起動させるのには三つの場合があります。

- (1) イニシャライズ（初期設定）を行ない，再設定する場合

```

A>IE78C11 )
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright(c) xxxx NEC Corporation
Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y )

```

- (2) イニシャライズ（初期設定）を行わず，設定を前のままで使用する場合

（IE-78C11の動作を一時中断し再スタートする場合にIE-78C11の初期設定を省くことができます）。

```

A>IE78C11 )
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright(c) xxxx NEC Corporation
Select Channel Number ? (1/2/3/4) : 1 )
Initialize ? (Y/N) : N )

```

- (3) エミュレーションCPUを実行させたまま終了し再起動する場合

```

A>IE78C11 )
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright(c) xxxx NEC Corporation
Select Channel Number ? (1/2/3/4) : 1 )
E-CPU RUN
I>

```

（この状態は，エミュレーションCPUが実行状態で，CTRL C入力後コマンド入力待ちと同一の状態になります）。

保守/廃止**第3章 IE-78C11-M の設置**

この章では、IE-78C11-Mを起動させ簡単なオペレーションを行なうまでの手順を説明します。

3.1 IE 番号の設定**3.1.1 IE 番号とは**

MD-086シリーズは、マルチ・プロセッシング機能をもっています。IE-78C11は、この機能を十分活用できるように設計されています。このマルチ・プロセッシング機能により、同じIE-78C11を最大2台までならば、個々に同時に使用できます（ほかのIEとの併用も可能です）。

IE-78C11システム・ソフトでは、一つのプロセス^注が1台のIE-78C11をコントロールする方法をとっています。2台以上のIE-78C11を同時にコントロールするためには、個々に対してプロセスを起動します。

IE番号は、そのプロセスが、どのIE-78C11を操作するかという指定情報になります。

注 MP/M-86における一つのプログラム実行単位

3.1.2 IE 番号の設定

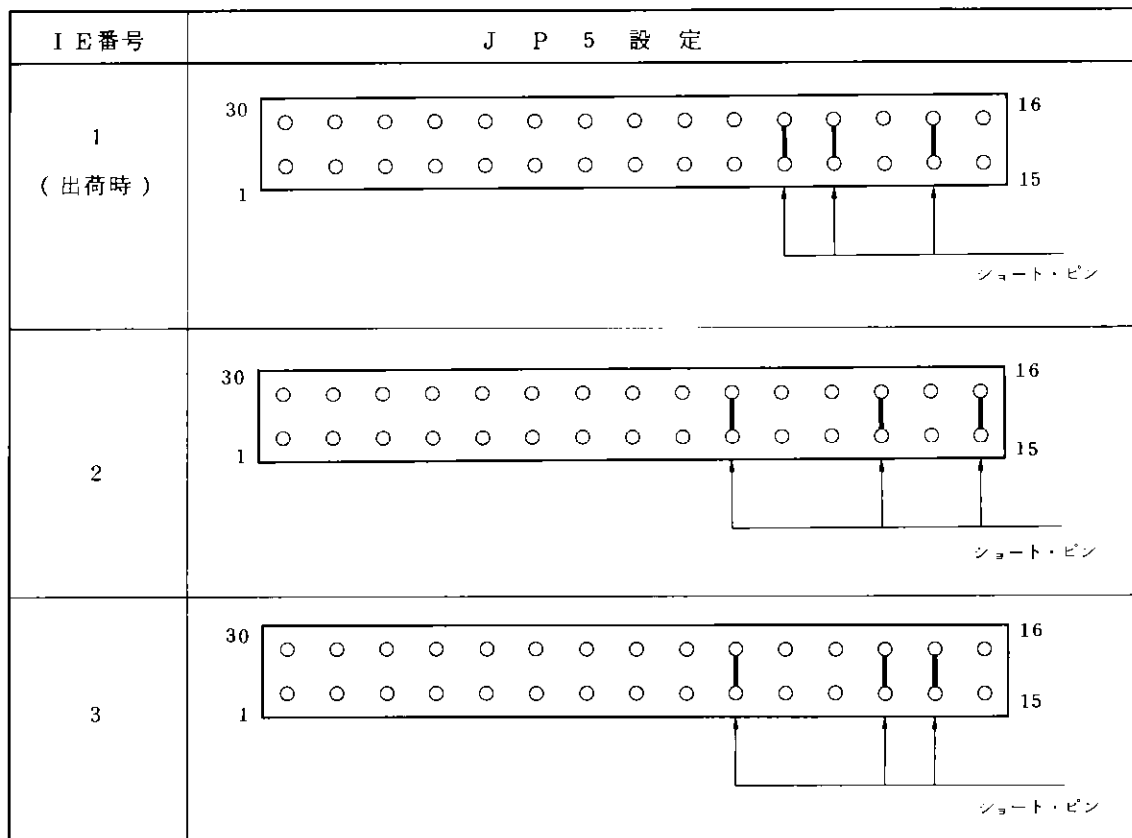
J P 5で“IE番号”を設定します。

図3-1に、その設定パターンを示します。

（図で示した部分のショート・ピンを変更することによって、IE番号が変わります）

保守/廃止

図 3-1



3.1.3 注意事項

複数のIE-78C11および他のIEを同一のIE番号に設定し、スロットに挿入しないでください。故障の原因となります。

システム・ソフト起動時、他で使用中の“IE番号”を指定すると、そこでのディバグ作業を混乱させてしまいますので、“IE番号”の指定には各自注意してください。

3.2 IEコントロール・ボードの設定

この節では、IEコントロール・ボードのジャンパおよび、スイッチの設定方法についての説明をします。

IE-78C11コントロール・ボードのジャンパおよび、スイッチの位置を図3-2に示します。

保守/廃止

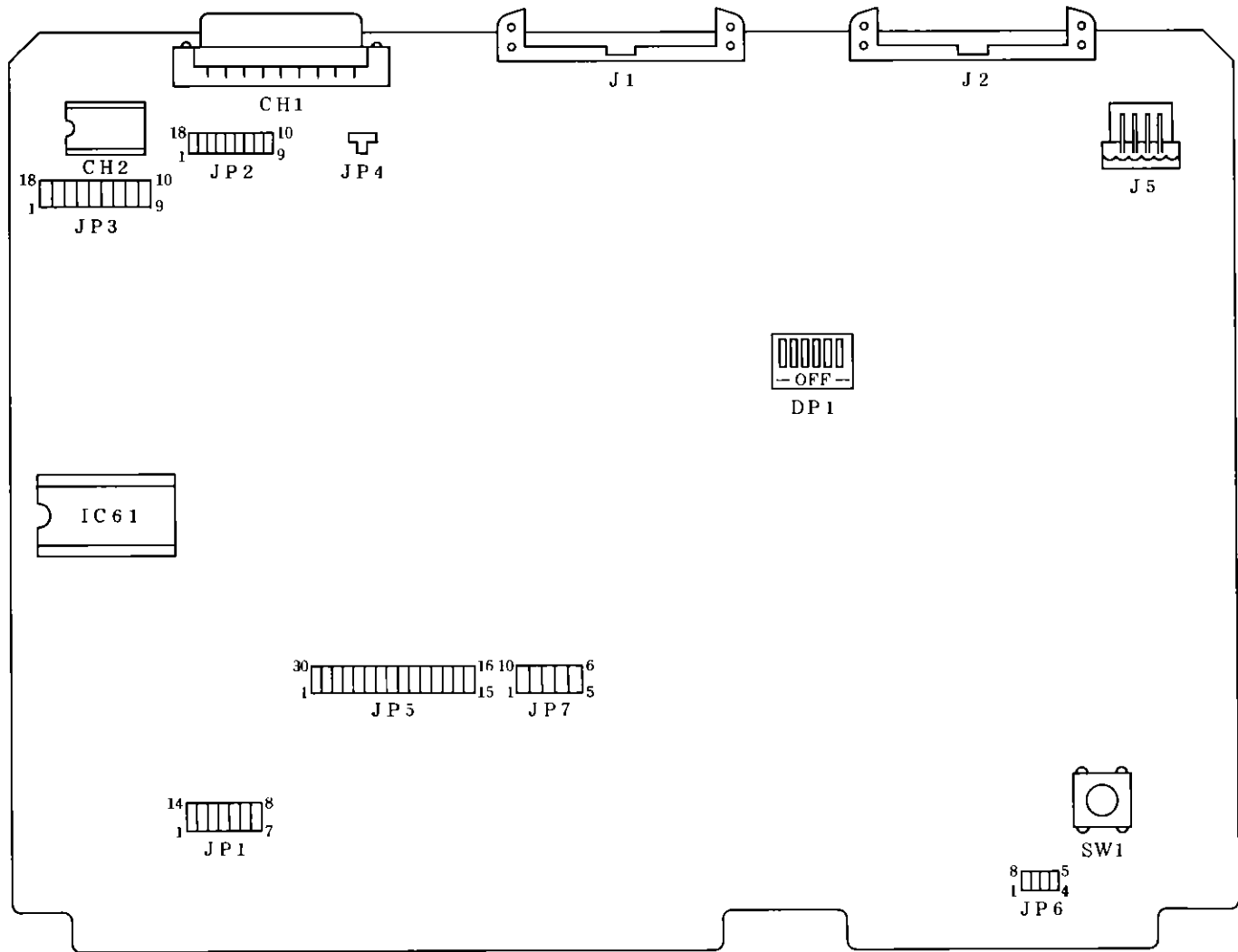
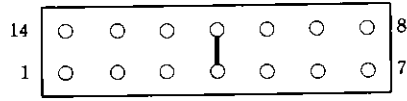


図 3-2 IE-78C11コントロールボード

保守/廃止

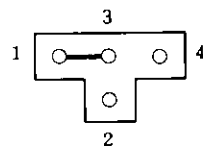
図 3 - 3 JP1



JP2・JP3

JP2・JP3の設定は不用です。

図 3 - 4 JP4



JP5

IE番号を指定する場合は、3.1.2 IE番号の設定を参照してください。

図 3 - 5 に、IE番号が '1' の場合を示します。

図 3 - 5

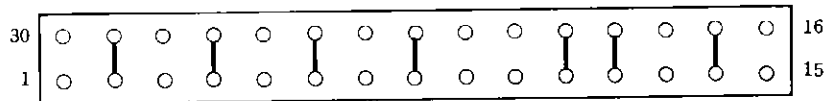


図 3 - 6 JP6

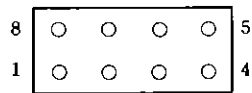
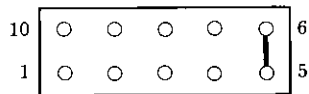


図 3 - 7 JP7

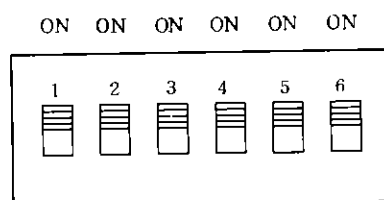


保守/廃止

D P 1

D P 1 (D I P S W) は図 3 - 8 のように設定してください。

図 3 - 8 D P 1

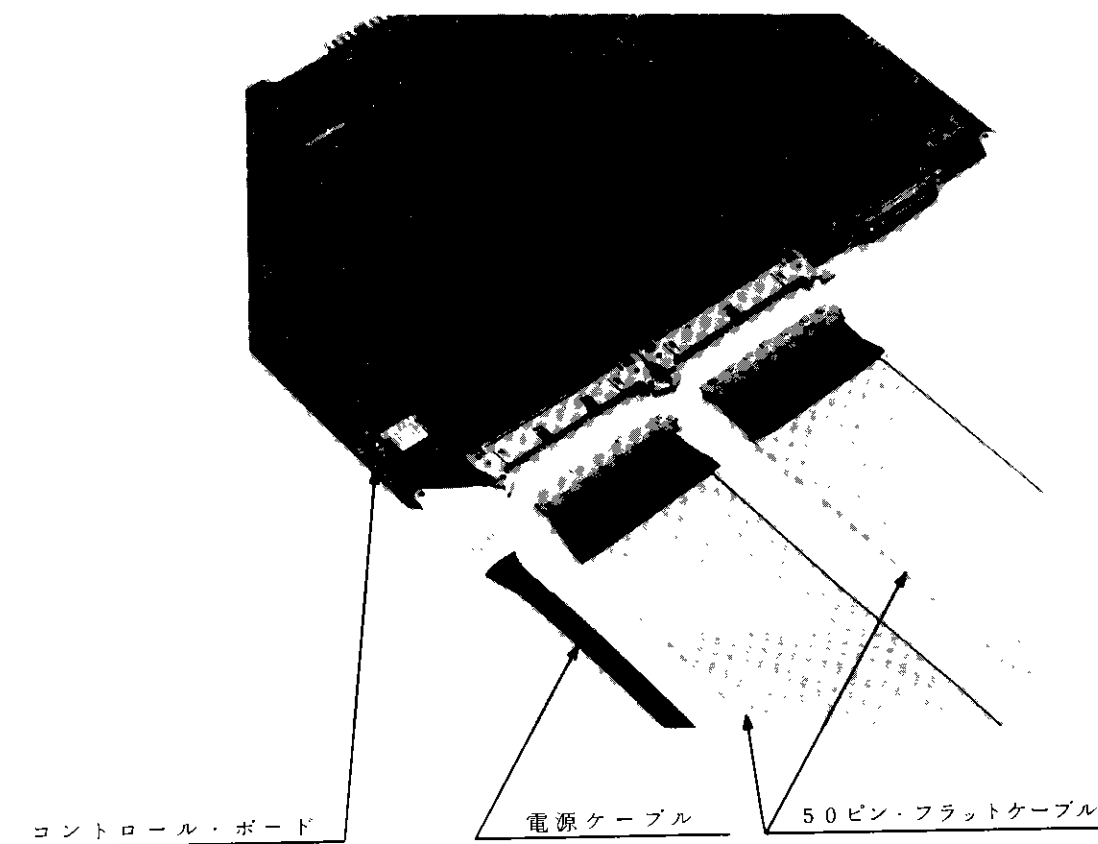


保守/廃止

3.3 IEコントロール・ボードとドライバの接続

- (1) IE-78C11ドライバからでてくる二つ(J1, J2)の50ピン・フラット
 - ・ケーブルの先のコネクタを、コントロール・ボード側の対応する“J1”, “J2”と名前のついたソケットに装着します。
- (2) 同様にIE-78C11ドライバよりでてくる電源ケーブルのコネクタ(J5)に電源ソケットを装着します。

図3-9



保守/廃止

3.4 MD-086 シリーズとの接続

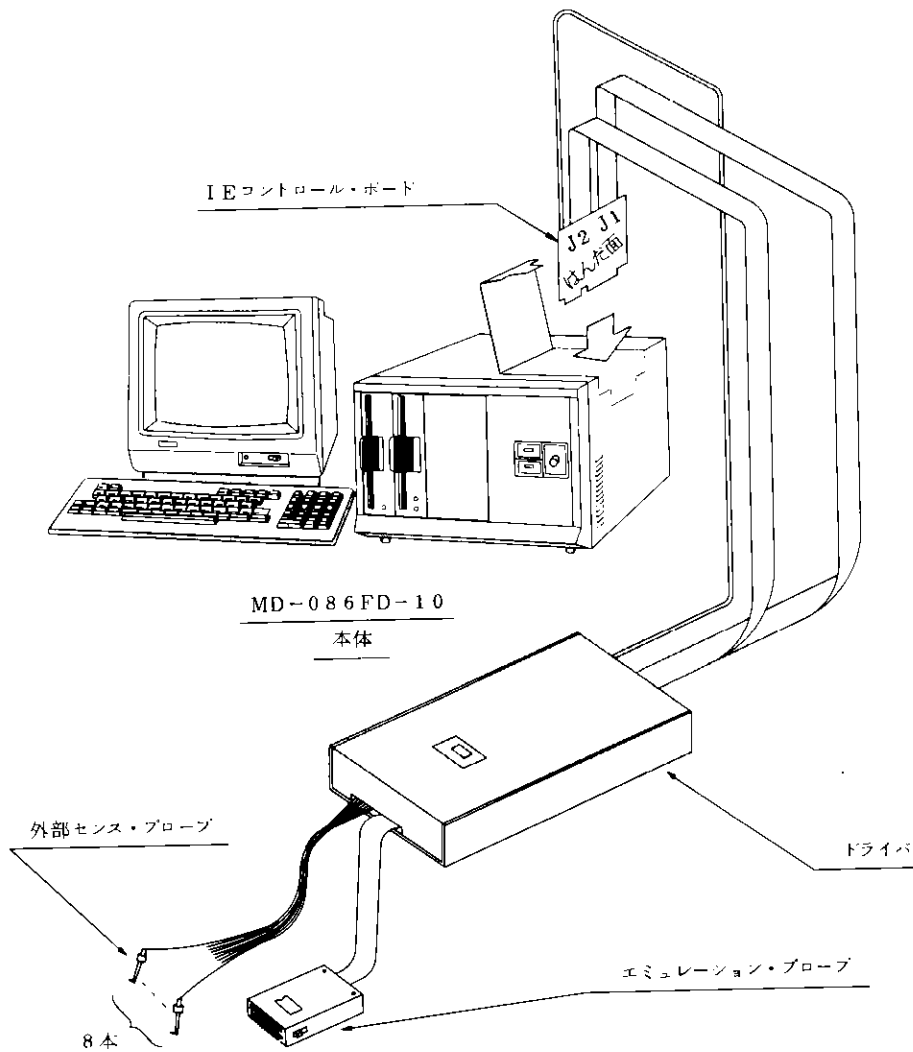
3.4.1 MD-086FD-10 と接続する場合

MD-086FD-10 本体のカバーを開けて IE コントロール・ボードを挿入してください。

挿入時は、MD-086FD-10 の電源を必ず切って行なってください。

図 3-10 に接続図を示します。

図 3-10



保守/廃止

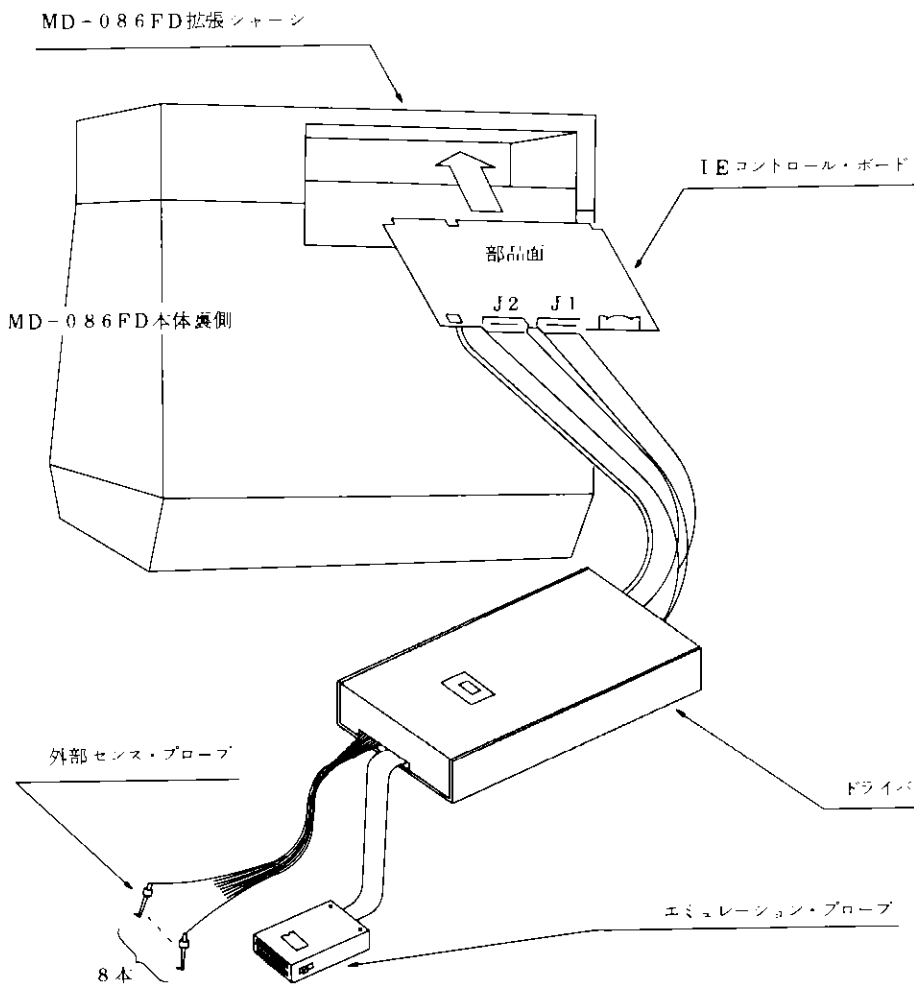
3.4.2 MD-086FDと接続する場合

MD-086FDには、IEコントロール・ボードの部品面を上にして挿入するものと、はんだ面を上にして挿入するものの2種類がありますので注意してください。

IEコントロール・ボードを挿入するときは、MD-086 および拡張シャーシの電源を遮断した上で行ってください。

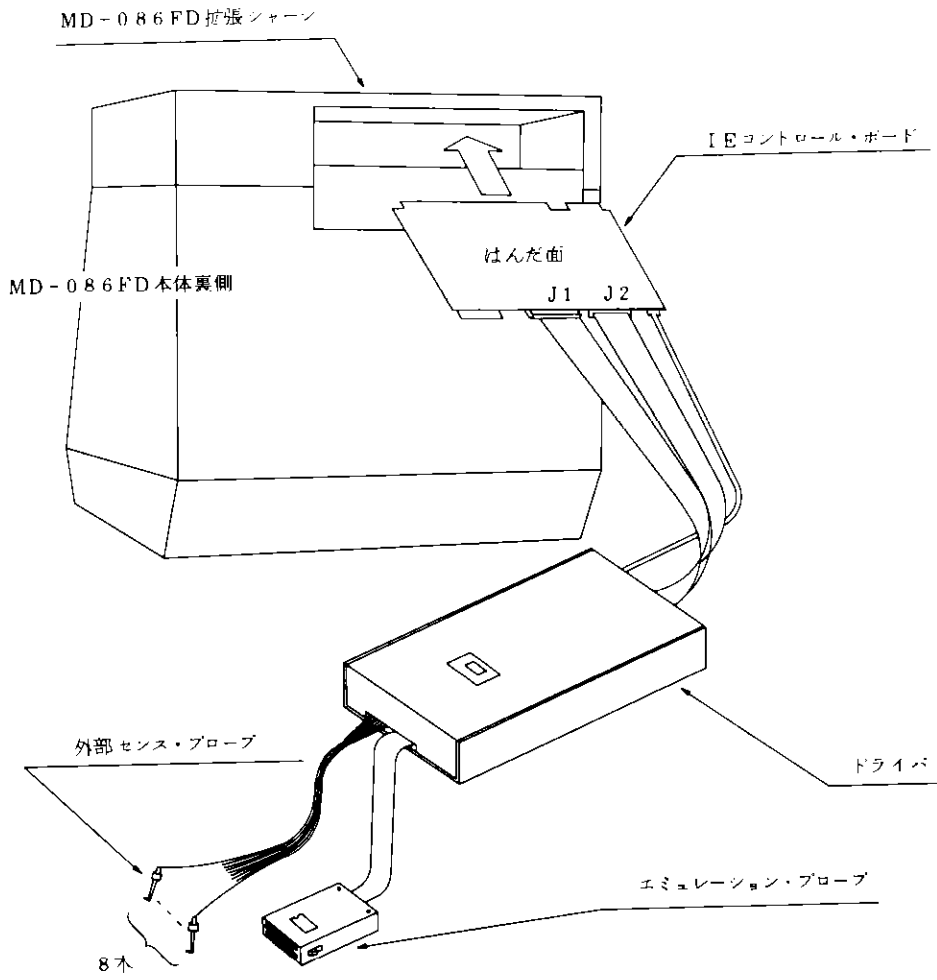
図3-11、図3-12に接続図を示します。

図3-11



保守/廃止

図3-12



保守/廃止

3.5 ターゲット・システムとの接続

3.5.1 ターゲット・システムとの接続

ターゲット・システムと IE-78C11 との接続は以下に述べる順序で行なってください。

- (1) IE-78C11 とターゲット・システムの電源を OFF にします。
- (2) エミュレーション・プローブをターゲット・システムの 64 ピン・ソケットに挿入します。
エミュレーション・プローブをターゲット・システムに接続せず IE-78C11 によるソフトウェアの開発、デバッグを行なう場合にはエミュレーション・プローブを IE-78C11 ドライバの 64 ピン・コネクタに挿入してください (CPU クロックは IE-78C11 内部のクロックを使用します)。

IE-78C11 取扱以上の注意点

- エミュレーション・プローブの MODE0, MODE1 端子は IE-78C11 内部でオープン・コレクタの出力と接続されていますから、直接 Vcc につなぐと過電流が流れ誤動作する恐れがありますので、必ず抵抗 (4.7 kΩ 程度) でプルアップしてください。

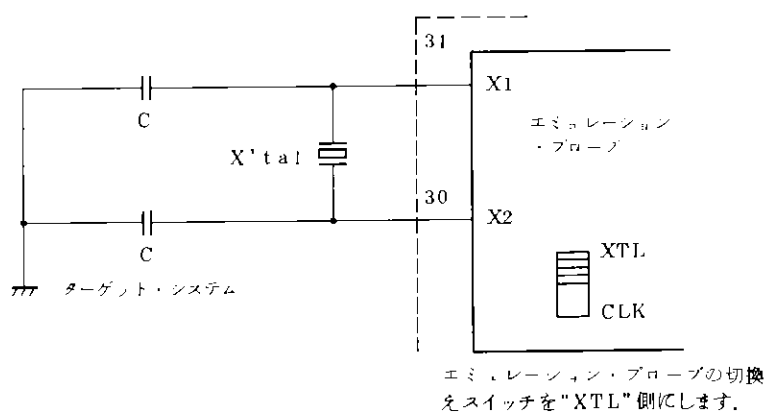
3.5.2 エミュレーション・プローブのセッティング

IE-78C11 のエミュレーション・プローブには水晶発振用の回路が内蔵されていますので、ターゲット・システムの発振回路を CPU クロックとして使用する場合には発振回路に合わせてエミュレーション・プローブの切換えスイッチを設定しなければなりません。

ここで、ターゲット・システムの発振回路を使用せずに IE-78C11 の内部クロック (12 MHz) を使用する場合には切換えスイッチはどちらでもかまいません。

- 1) ターゲット・システム上の発振回路が水晶発振子 (X'tal) だけで構成されている場合はスイッチを "XTL" 側にしてください。

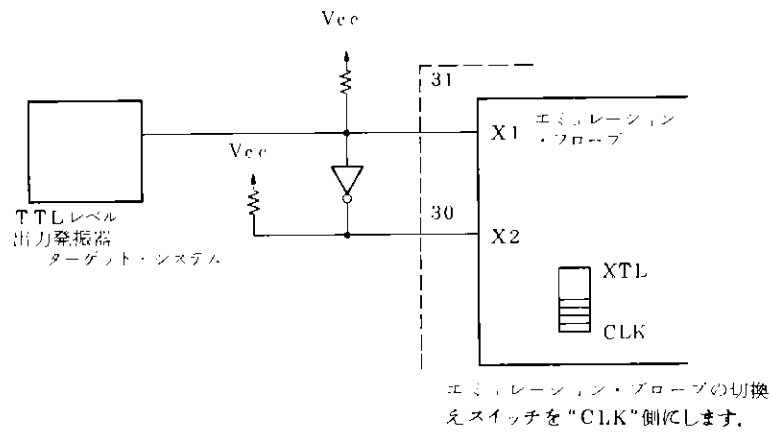
図 3-13



保守/廃止

- 2) ターゲット・システム上の発振回路がTTLレベル出力の発振器で構成されている場合は、スイッチを“CLK”側にしてください。

図 3 - 1 4



保守/廃止

3.6 IE-78C11システム・ソフトのバック・アップ

IE-78C11システム・ソフト(プログラム名: IE78C11.COM)がはいっているフロッピー・ディスクには、MP/M-86システムおよびユーティリティははいっていません。そのため、IE-78C11システム・ソフトをそのまま使用すると、MP/M-86システムおよびIE-78C11システムにそれぞれ1台のドライブが必要となります。

以上のことや、システムの暴走や不慮の事故によるプログラムの破壊に対応するため、MP/M-86システム・フロッピー・ディスクを新規に作成し、IE-78C11システム・ソフト一式をコピーして使用されることをお勧めします。

以下にIE-78C11システムのマスタ・フロッピーをBドライブにセットし新規に作成したシステム・フロッピー・ディスクをAドライブにセットした場合のコピー方法と、IE-78C11に必要なファイルの一覧を示します。

```

0A>PIP A:=B:*.*[R]      (AドライブにPIP.COMが必要です。)
XX:XX:XX A:PIP.COM

COPYING
IE78C11.COM
IE87.HLP
EX.STR
EXO.SYM
EXO.HEX

0A>

```

ファイル名一覧

IE78C11.COM	システム・ソフトです。
IE87.HLP	ヘルプ・ファイルです。
EX.STR	4章で使用するシンボル・ファイルです。
EXO.SYM	4章で使用するシンボル・ファイルです。
EXO.HEX	4章で使用するヘキサ・ファイルです。

注 ファイルの属性はSYS,ROとなっています。

保守/廃止

3.7 IE-78C11システム・ソフトの起動

ターゲット・システム，IE-78C11の順に電源を投入し，以下のようにして起動させていただきます。

```

OA>IE78C11
XX:XX:XX A:IE78C11.CMD

IE-78C11 System Software Vx.x [dd mmm yy]
Copyright(c) xxxx NEC Corporation

Select IE-Number ? (1/2/3) : 1 (1)
Initialize ? (Y/N) : Y (2)

I>

```

ハードウェアが正しく接続されていない場合は下のように出力されます。

```

OA>IE78C11
XX:XX:XX A:IE78C11.CMD

IE-78C11 System Software vx.x [dd mmm yy]
Copyright(c) xxxx NEC Corporation

Select IE-Number ? (1/2/3) : 2 (1)

No Connect (3)
Abort ? (Y/N) : N (4)

Select IE-Number ? (1/2/3) :

```

- (1) 接続したIE番号を入力します。
- (2) 5.1, 5.1.1 参照
起動(POWER ON)時には，必ず“Y”と入力しイニシャライズを行なってください。
イニシャライズを行わない場合，IEのメモリのマッピングは不定となります。
- (3) 接続されていないIE番号を入力した場合に出力されます。
- (4) “N”を入力した場合はIE番号の設定入力待ちの状態に戻ります。

保守/廃止

3.8 IE-78C11 システム・ソフトの終了

IE-78C11 システム・ソフトを終了する場合コマンド入力状態において“EXT)”と入力します。

```

I>RUN N )
USER-SYSTEM YDD-ON

JUST MOMENT
EMULATION START

NORMAL BREAK
TERMINATED

PSW Y A B C D E H L EA SP PC
00 00 00 00 00 00 00 00 0000 014C 05D0
   Y' A' B' C' D' E' H' L' EA'
   00 00 00 00 00 00 00 00 0000

ONE STEP EMULATION STANDBY

I>EXT )
0A>

```

CTRL Cを入力することによっても、システムを終了することができます。

```

I>MAP )
$ IE-INTRM MAPPING $
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
0000-FE00

I> ----- CTRL/C入力
Abort ? (Y/N) : Y )
(5)

0A>

```

(1) “Y”を入力した場合、システム・ソフトは終了しMP/M-86に戻ります。

CTRL C入力は、コマンド実行中でもシステム・ソフトを終了することができます。
エミュレーション実行中の **CTRL** C入力は、エミュレーションCPUを実行させたままシステム・ソフトを終了します。

```

I>RUN N )
USER-SYSTEM YDD-ON

JUST MOMENT
EMULATION START

I> ----- CTRL/C入力
Abort ? (Y/N) : Y )
(6)

0A>

```

保守/廃止

(2) “Y”を入力した場合、システム・ソフトは終了しMP/M-86に戻ります。この時も、エミュレーションCPUは実行したままになっています。

“Abort?(Y/N):”の状態で、“N”を入力した場合、システム・ソフトの終了は行なわれません。

注 電源はターゲット・システム、IE-78C11の順で解除してください。

保守/廃止

3.9 2度目以降の起動

IE-78C11システム・ソフトを一度終了させた後に起動させるのには三つの場合があります。

- (1) イニシャライズ(初期設定)を行ない、再設定する場合

```
OA>IE78C11 )
XX:XX:XX A:IE78C11.CMD

IE-78C11 System Software Vx.x [dd mmm yy]
Copyright(c) xxxx NEC Corporation

Select IE-Number ? (1/2/3) : 1 )
Initialize ? (Y/N) : Y )
```

- (2) イニシャライズ(初期設定)を行わず、設定を前のままで使用する場合

(IE-78C11の動作を一時中断し再スタートする場合に IE-78C11の初期設定を省くことができます)

```
OA>IE78C11 )
XX:XX:XX A:IE78C11.CMD

IE-78C11 System Software Vx.x [dd mmm yy]
Copyright(c) xxxx NEC Corporation

Select IE-Number ? (1/2/3) : 1 )
Initialize ? (Y/N) : N )
```

- (3) エミュレーションCPUを実行させたまま終了し再起動する場合

(この場合、エミュレーションCPUが実行状態であるためRESコマンドとESCキー以外は受けつけません)

```
OA>IE78C11 )
XX:XX:XX A:IE78C11.CMD

IE-78C11 System Software Vx.x [dd mmm yy]
Copyright(c) xxxx NEC Corporation

Select IE-Number ? (1/2/3) : 1 )
E-CPU RUN
```

I> -----ESCキー入力

```
ESC BREAK
TERMINATED
```

```
PSW V A B C D E H L EA SP PC
04 08 01 02 03 04 05 06 07 0000 0000 000E
```

```
V' A' B' C' D' E' H' L' EA'
00 00 00 00 00 00 00 00 0000
```

I>

保守/廃止

CTRL D入力により、コンソールの切離し(デタッチ)を行なっていたときは、再度
CTRL D入力を行なうかまたは“ATTACH IE78C11”を入力することにより、再びシステム・ソフトにもどります。

```
I>MAP _ )
```

```
$ IE-INTROM MAPPING $
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
0000-FE00
```

```
I>MAP W 0.FF _ )
```

```
I> _____ CTRL/D入力
Attach: Tmp0
```

```
0A> ATTACH IE78C11 _ )
XX:XX:XX A:IE78C11.CMD
```

```
Attach: IE78C11
```

```
I> MAP _ )
```

```
$ IE-INTROM MAPPING $
$ IE-INTRAM MAPPING $
0000-00FF FF00-FFFF
$ USER $
$ NON MAPPING $
0100-FE00
```

```
I>
```

保守/廃止

第4章 操作例

以下は、IE-78C11-Cの起動よりプログラムのディバグまでの操作例です（IE-78C11-Mは、起動時のみ操作方法が異なります）。

```

A>IE78C11 )
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright (C) xxxx NEC Corporation

Select Channel Number ? (1/2/3/4) : 1 ) (1)
Initialize ? (Y/N) : Y ) (2)

USER-SYSTEM VDD-ON

CPU 78C10 ) (3)
EXTERNAL MEMORY SIZE 64K BYTE
RAE (E:ENABLE,D:DISABLE)=E )

I>MAP ) (4)

$ IE-INTROM MAPPING $
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
0000-FEFF
I> MAP W 0,FFFF ) (5)

I> LOD B:EXO ) (6)

Object
Load Complete

Symbol Table
Load Complete

I> DAS 1000,101B ) (7)

ADRS OBJECT LABEL MNEMONIC
1000 3400FF ADDATA: LXI H,BUFF0
1003 2410FF LXI D,BUFF1
1006 400B10 CALL BFADD
1009 00 NOP
100A FF FINAL: JR FINAL
100B 482A BFADD: CLC
100D 6B04 MY! C,00004H
100F 2B BFAD1: LDAX H
1010 70D4 ADCX D+
1012 3D STAX H+
1013 53 DCR C
1014 FA JR BFAD1
1015 481A SKN CY
1017 B8 RET
1018 B9 RETS
1019 00 NOP
101A 00 NOP
101B 00 NOP
I> SYM D ) (8)

[EXAMPLE0] ) (9)
PUBLIC SYMBOL

LOCAL SYMBOL

ADDATA 1000 C BFAD1 100F C BFADD 100B C
BUFF0 FF00 D BUFF1 FF10 D FINAL 100A C
I> BRM BRA ) (10)

I> BRA A=[EXAMPLE0]FINAL ) (11)

```

保守 / 廃止

I>IRM ALL) _____ (12)

I>IRC 1) _____ (13)

I>MEM C [EXAMPLE0]BUFF0) _____ (14)

FF00 00-21 00-43 00-85 00-87 00-Q1 00-)
I> MEM C [EXAMPLE0]BUFF1 _____ (15)

FF10 00-10 00-32 00-54 00-76 00-Q1 00-)
I>RUN B 1000) _____ (16)

USER-SYSTEM VDD-ON

JUST MOMENT
EMULATION START AT 1000

NORMAL BREAK _____ (17)
TERMINATED

```
PSW V A B C D E H L EA SP PC
10 00 02 00 FF FF 15 FF 05 0000 0700 100A
   V' A' B' C' D' E' H' L' EA'
   00 FF FF FF 01 4C 00 00 014C
```

ONE STEP EMULATION STANDBY)

I>MEM D FFOX) _____ (18)

FF00 31 75 B9 FD 02 00 00 00 00 00 00 00 00 00 00 _____ (19)

I> TRD ALL) _____ (20)

ADRS	OBJECT	LABEL	MNEMONIC	PA	PB
1000	3400FF	ADDATA:	LXI H,BUFF0	00	00
1003	2410FF		LXI D,BUFF1	00	00
1006	400B10		CALL BFADD	00	00
100B	482A	BFADD:	CLC	00	00
100D	6B04		MVI C,00004H	00	00
100F	2B	BFADI:	LDAX H	00	00
1010	70D4		ADCX D+	00	00
1012	3D		STAX H+	00	00
1013	53		DCR C	00	00
1014	FA		JR BFADI	00	00
100F	2B	BFADI:	LDAX H	00	00
1010	70D4		ADCX D+	00	00
1012	3D		STAX H+	00	00
1013	53		DCR C	00	00
1014	FA		JR BFADI	00	00
100F	2B	BFADI:	LDAX H	00	00
1010	70D4		ADCX D+	00	00
1012	3D		STAX H+	00	00
1013	53		DCR C	00	00
1014	FA		JR BFADI	00	00
100F	2B	BFADI:	LDAX H	00	00
1010	70D4		ADCX D+	00	00
1012	3D		STAX H+	00	00
1013	53		DCR C	00	00
1014	FA		JR BFADI	00	00
1015	481A		SKN CY	00	00
1017	B8		RET	00	00
1018	B9		RETS	00	00
1009	00		NOP	00	00
100A	FF	FINAL:	JR FINAL	00	00

I>ASM [EXAMPLE0]BFADD) _____ (22)

```
ADRS OBJECT LABEL MNEMONIC
100B 482A BFADD: CLC
      = )
100D 6B04 MVI C,00004H
      =MVI C,3 )
100D 6B03
100F 2B BFADI: LDAX H
      =END )
```


保守/廃止

I>MOD EXAMPLE0) _____ (23)

I>MEM C BUFF0) _____ (24)

FF00 31-21 75-43 B9-65 FD-87 02-01 00-)
I> MEM C BUFF1 _____ (25)

FF10 10-10 32-32 54-54 76-76 01-)
I> RUN B 1000) _____ (26)

USER-SYSTEM VDD-ON

JUST MOMENT
EMULATION START AT 1000

NORMAL BREAK
TERMINATED

```
PSW Y A B C D E H L EA SP PC
10 00 FD 00 FF FF 14 FF 04 0000 0700 100A
   Y' A' B' C' D' E' H' L' EA'
   00 FF FF FF 01 4C 00 00 014C
```

ONE STEP EMULATION STANDBY)

I>MEM D FFOX) _____

FF00 31 75 B9 FD 01 00 00 00 00 00 00 00 00 00 00 00 _____ (27)

I> DAS 1000,1020) _____ (28)

ADRS	OBJECT	LABEL	MNEMONIC
1000	3400FF	ADDATA:	LXI H,BUFF0
1003	2410FF		LXI D,BUFF1
1006	400B10		CALL BFADD
1009	00		NOP
100A	FF	FINAL:	JR FINAL
100B	482A	BFADD:	CLC
100D	6B03		MVI C,00003H
100F	2B	BFAD1:	LDAX H
1010	70D4		ADCX D+
1012	3D		STAX H+
1013	53		DCR C
1014	FA		JR BFAD1
1015	481A		SKN CY
1017	B8		RET
1018	B9		RETS
1019	00		NOP
101A	00		NOP
101B	00		NOP
101C	00		NOP
101D	00		NOP
101E	00		NOP
101F	00		NOP
1020	00		NOP

I> EXT)

A>

保守/廃止

- (1) IE-78C11と接続されているチャンネル・ナンバを入力します。
- (2) イニシャライズを行ないます。
- (3) CPU: μ PD78C10, RAEビットオンの場合の初期設定を行ないます。
- (4) マッピングを確認します。
- (5) 0~FEFFH番地を内部メモリにマッピングします。
- (6) 今回ディバグを行なうプログラムをロードします。
 HEXオブジェクト=B:EXO. HEX
 シンボル・ファイル=B:EXO. SYM
- (7) ディバグを行なう1000H~101BH番地を逆アセンブルします。
 1000H~101BH番地のプログラムは、ローカル・シンボル'BUFF0'(FF00DH番地)からの4バイトのデータとローカル・シンボル'BUFF1'(FF10DH番地)からの4バイトのデータを加え、その結果をBUFF0から4バイト書込むプログラムです。
- (8) シンボルを、すべて表示します。
- (9) モジュール名 EXAMPLE0
- (10) ブレーク・レジスタ(BRA)のブレーク・アドレスをモジュール名'EXAMPLE0'のローカル・シンボル'FINAL'(100AH)番地に設定します。
- (11) ブレーク・モード・レジスタ(BRM)にBRAを登録します。
- (12) トレース・モード(TRM)をすべてトレースする条件(ALL)に設定します。
- (13) コマンドにより、トレースの表示形式をインストラクション表示(I)に指定します。
- (14) BUFF0を87654321に設定します。
- (15) BUFF1を76543210に設定します。
- (16) プログラムを実行します。
- (17) ブレーク条件によってブレークしました。
- (18) 実行結果を確認します。
- (19) FF04H番地の値が実行前01から、実行後02に変わっていることがわかります。
 FF04H番地は変わってはならないプログラムですが実際には変わっています。
 そこで、プログラム修正を行ないます。
- (20) 実行結果をトレースします。
- (21) サブルーチンです。
 BFAD1(100FH番地)を5回通過していることがわかります(1回多く通過していることがわかりました)。
- (22) BFAD1を通過する回数を1回減らします。
 アセンブル・コマンドによってカウンタ数を4から3にします。
- (23) モジュール・コマンドによってカレント・モジュールをEXAMPLE0に設定します。
- (24) モジュール・コマンドの指定によりモジュール名を省略できます。
 再度BUFF0番地以後のデータを設定します。
- (25) BUFF1番地以後のデータを設定します。

保守/廃止

- ㉔ プログラムを実行します（ブレイク条件は以前のままです）。
- ㉕ 実行結果を確認します（正しい結果が得られました）。
- ㉖ 再度、プログラムを逆アセンブルしてみます。

※ 第4章の操作例は、添付のフロッピー・ディスクの中に EX.STR というファイル名で入っています。
このファイルをストリング・コマンド（STR）で次のように入力することにより再現することができます。

1 > STR EX 2

保守/廃止

第5章 IE-78C11起動時の初期設定

5.1 MODE0, MODE1 端子の設定

IE-78C11は電源投入時にユーザ・システムのMODE0, MODE1端子の状態を読み込み、その値によりCPUのモードを設定します。

したがって、ユーザ・システム上でMODE0, 1端子を希望の設定にプルアップ、ダウンする必要があります。

表 5-1

MODE1	MODE0	モ ー ド
0	0	μ PD78C10 外部4Kバイト
0	1	μ PD78C10 外部16Kバイト
1	0	μ PD78C11, μ PD78C14
1	1	μ PD78C10 外部64Kバイト

ユーザ・システムが無い場合、ドライバ上の自己診断用コネクタに、エミュレーション・プローブを挿入することにより、ソフトウェア・デバッグが行なえます。しかし、この場合MODE1, MODE0端子は、共に'1'となり μ PD78C10で外部64Kバイト・モードとなります。また、ポートA, B, Cが内部で接続されているため、ポートA, B, Cを共に出力モードとして使用することはできません。

このような制限を受けたくない場合、エミュレーション・プローブをどこにも接続しない状態で使用します。この状態で電源投入を行なうと、タイトル・メッセージを出力した後'USER - SYSTEM VDD-OFF'のメッセージを出力し停止します。このとき、ESCキー入力を行なうと'MODE0='と出力しますので'0'か'1'を入力します。次に'MODE1='と出力しますので同様に'0'か'1'を入力してください。

これで任意のモードに設定ができます。

保守/廃止

例

```

IE-78C11 MONITOR V2.0 [17 DEC 85]
COPYRIGHT (C) 1985 NEC CORPORATION

USER-SYSTEM VDD-OFF ←← E S Cキー入力

SELECT MODE
MODE0= Q
MODE1= Q

CPU 78C10
EXTERNAL MEMORY SIZE 4K BYTE
RAE (E:ENABLE, D:DISABLE)=

```

注 ハードウェア・バージョンがAのものは、使用できません。
 ハードウェア・バージョンは、コントロール/トレース・ボードのCH1コネクタの横に
 捺印された10桁のシリアル・ナンバーの左から2番目のアルファベットで表されます。

XXXXXXXXX4

└─ ハードウェア・バージョン

ハードウェア・バージョンAについて

ハードウェア・バージョンがAのものはリセット回路が図5-1のようになっており、プローブに何も接続しない状態ではリセット入力オープンのため、HC14は'0'とみなし μ PD78C10はリセット状態となり、ユーザ・プログラムの実行ができません。

図5-2あるいは図5-3のいずれかの処置を行なうことによりユーザ・プログラムの実行が可能になります。

図5-1

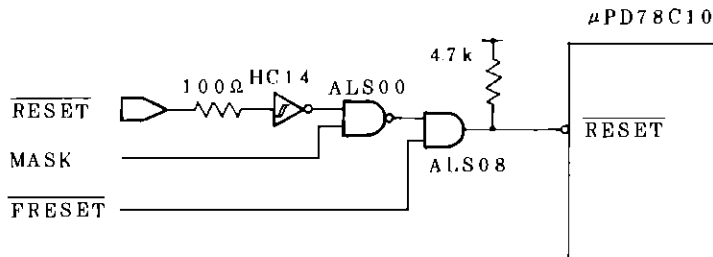
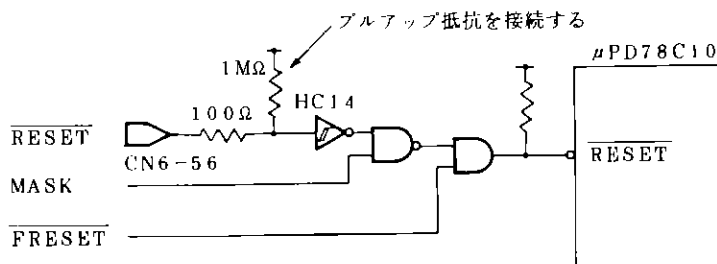
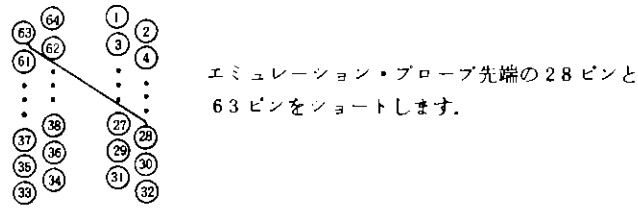


図5-2



保守/廃止

図5-3



5.2 μPD78C11およびμPD78C14のデバッグ

μPD78C11はROM4Kバイト、RAM256バイト、μPD78C14はROM16Kバイト、RAM256バイト内蔵のCPUです。ユーザ・システムのMODE0、MODE1端子をそれぞれMODE0=0、MODE1=1に設定すると初期設定時にモニタが'SELECT CPU(78C11:1/78C14:4)= 'とメッセージを出力しますので'1'か'4'を入力してCPUの選択を行なってください。

図5-4 μPD78C11メモリ・マップ

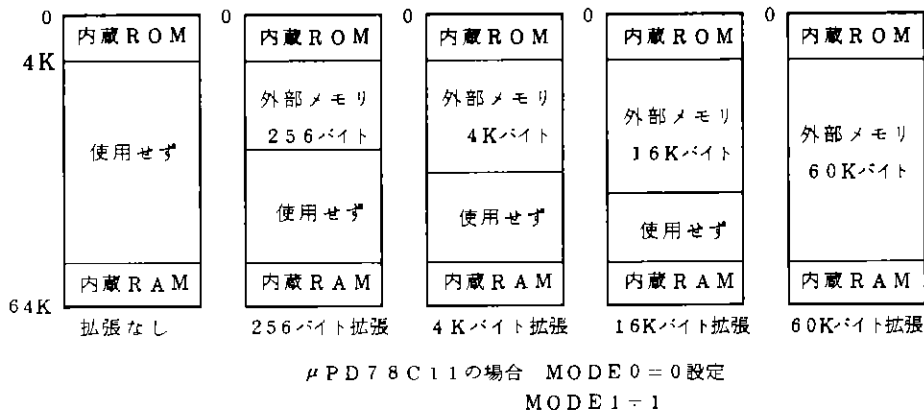
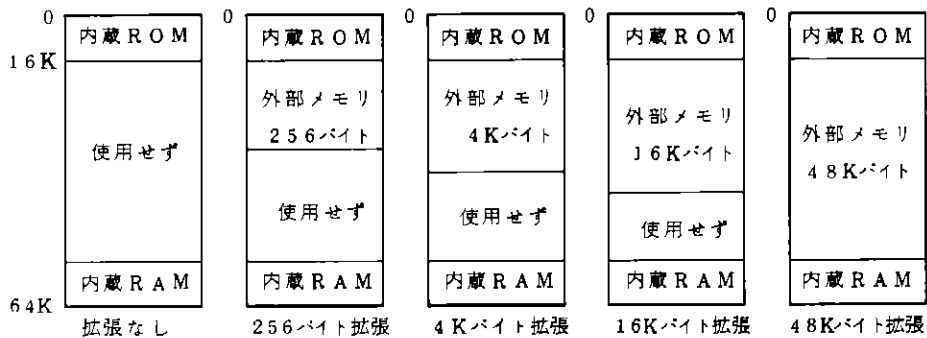


表5-2

拡張メモリ	PF0	PF1	PF2	PF3	PF4	PF5	PF6	PF7
256バイト	ポート	ポート	ポート	ポート	ポート	ポート	ポート	ポート
4Kバイト	AB8	AB9	AB10	AB11	ポート	ポート	ポート	ポート
16Kバイト	AB8	AB9	AB10	AB11	AB12	AB13	ポート	ポート
60Kバイト	AB8	AB9	AB10	AB11	AB12	AB13	AB14	AB15

保守/廃止

図 5 - 5 μPD78C14メモリ・マップ



μPD78C14の場合 MODE0=0設定
MODE1=1

表 5 - 3

拡張メモリ	PF0	PF1	PF2	PF3	PF4	PF5	PF6	PF7
256バイト	ポート	ポート	ポート	ポート	ポート	ポート	ポート	ポート
4Kバイト	AB8	AB9	AB10	AB11	ポート	ポート	ポート	ポート
16Kバイト	AB8	AB9	AB10	AB11	AB12	AB13	ポート	ポート
48Kバイト	AB8	AB9	AB10	AB11	AB12	AB13	AB14	AB15

IE-78C11 起動時、以下の設定を行なってください。

- (1) 内蔵RAMアクセス可否の設定 (RAEビット)
- (2) ポートDのモード設定 (入出力モード)
- (3) ポートFのモード設定 (入出力モード)

5.2.1 内蔵RAMアクセス可否の設定

ユーザ・プログラム中で設定するMMレジスタのRAEビットにあわせてイネーブル (Eを入力) またはディスエーブル (Dを入力) にしてください。

5.2.2 ポートDのモード設定

- (1) ポートDをポートとして使用する場合、そのポートを入力として使用するか、出力として使用するかを設定します。
- (2) 外部にメモリを拡張した場合、ポートDの指定はありません。

保守/廃止

5.2.3 ポートFのモード設定(表5-1, 表5-2参照)

- (1) ポートFをポートとして使用する場合、入力に使用するか出力に使用するかを1ビットずつ指定します。
- (2) 外部に60Kバイトまたは48Kバイトを拡張した場合、ポートの指定はありません。

注 RAE, PD, PFの指定は初期化プログラムでのMMレジスタ, MFレジスタの設定と一致しなければなりません。

一致していない場合はIE-78C11への初期設定入力がIE-78C11のハードウェアに設定され、エミュレーションCPUと異なるため正しく動作しません。

以下に初期設定時の例を示します(一部はキー入力する部分です)。

例1~16はIE-78C11-Cの場合について書かれておりますのでIE-78C11-Mについては第3章を参照してください。

例1 外部メモリなし RAEビットオン

```
IE-78C11 System Software Vx.x [dd mmm yy]
  Copyright (C) xxxx NEC Corporation

Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y

USER-SYSTEM VDD-ON

SELECT CPU (78C11:1/78C14:4)=1
EXTERNAL MEMORY SIZE (0:NON,1:256,2:4K,3:16K,4:60K)=0
RAE (E:ENABLE,D:DISABLE) =E
PORT-D MODE (1:INPUT,0:OUTPUT)=1
PORT-F MODE (1:INPUT,0:OUTPUT)
PF-01234567
  11110000

I> MAP
$ IE-INTROM MAPPING $
0000-0FFF
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
1000-FE0F
I>
```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
      ORG      0
      MVI     A,00001000B      ; RAE:E PD:I
      MOV     MM,A            ; SETMM-REG
      MVI     A,00001111B      ; 00001111 MODE
      MOV     MF,A            ; SET MF-REG

```


保守/廃止

例 2 外部メモリなし RAEビットオフ

```

IE-78C11 System Software Vx.x [dd mmm yy]
  Copyright (C) xxxx NEC Corporation

Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y

USER-SYSTEM VDD-ON

SELECT CPU (78C11:1/78C14:4)=1
EXTERNAL MEMORY SIZE (0:NON,1:256,2:4K,3:16K,4:60K)=0
RAE (E=ENABLE,D:DISABLE) = 0
PORT-D MODE (1:INPUT,0:OUTPUT)= 0
PORT-F MODE (1:INPUT,0:OUTPUT)
PF-01234567
  00001111

I> MAP

$ IE-INTRM MAPPING $
0000-0FFF
$ IE-INTRAM MAPPING $
$ USER $
$ NON MAPPING $
1000-FFFF
I>

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
ORG      0
MVI     A,00000001B    ; RAE:D PD:0
MOV     MM,A          ; SET MM-REG
MVI     A,11110000B   ; 11110000 MODE
MOV     MF,A          ; SET MF-REG

```

保守/廃止

例 3 外部メモリ 256 バイト RAE ビットオン

```

IE-78C11 System Software Vx.x [dd mmm yy]
  Copyright (C) xxxx NEC Corporation

Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y

USER-SYSTEM VDD-ON

SELECT CPU (78C11:1/78C14:4)=1
EXTERNAL MEMORY SIZE (0:NON,1:256,2:4K,3:16K,4:60K)=1
RAE (E:ENABLE,D:DISABLE) = E
PORT-F MODE (1:INPUT,0:OUTPUT)
PF-01234567
  LLLLQOO

```

```

1> MAP 2

$ IE-INTRM MAPPING $
0000-0FFF
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
1000-FE0F
1>

```

```

;
; SAMPLE OF MODE INITIALIZE PROGRAM
;
      ORG      0
      MVI     A,00001010B      ; RAE:E 256BYTE MODE
      MOV     MM,A            ; SET MM-REG
      MVI     A,00011111B      ; 00011111 MODE
      MOV     MF,A            ; SET MF-REG

```

保守/廃止

例 4 外部メモリ 256 バイト RAE ビットオフ

```
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright (C) xxxx NEC Corporation

Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y

USER-SYSTEM VDD-ON

SELECT CPU (78C11:1/78C14:4)=1
EXTERNAL MEMORY SIZE (0:NON,1:256,2:4K,3:16K,4:60K)=1
RAE (E:ENABLE,D:DISABLE) = D
PORT-F MODE (I:INPUT,O:OUTPUT)
PF-01234567
  00011111

I> MAP
$IE-INTRAM MAPPING $
0000-0FFF
$ IE-INTRAM MAPPING $
$ USER $
$ NON MAPPING $
1000-FFFF
I>
```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
ORG      0
MVI     A,00000010B      ; RAE:D 256BYTE MODE
MOV     MM,A            ; SET MM-REG
MVI     A,11111000B     ; 1111000 MODE
MOV     MF,A           ; SET MF-REG

```

保守/廃止

例 5 外部メモリ 4 K バイト RAE ビットオン

```

IE-78C11 System Software Vx.x [dd mmm yy]
  Copyright (C) xxxx NEC Corporation

Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : y

USER-SYSTEM VDD-ON

SELECT CPU (78C11:1/78C14:4)=1
EXTERNAL MEMORY SIZE (0:NON,1:256,2:4K,3:16K,4:60K)=2
RAE (E:ENABLE,D:DISABLE) = E
PORT-F MODE (I:INPUT,O:OUTPUT)
PF-01234567
  XXXX1100

I> MAP
$ IE-INTROM MAPPING $
0000-0FFF
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
1000-FEFF
I>

```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

;
; SAMPLE OF MODE INITIALIZE PROGRAM
;
      ORG      0
      MVI     A,00001100B    ; RAE:E 4KBYTE MODE
      MOV     MM,A          ; SET MM-REG
      MVI     A,00110000B   ; 00110000 MODE
      MOV     MF,A         ; SET MF-REG

```

保守/廃止

例 6 外部メモリ 4 Kバイト RAE ビットオフ

```

IE-78C11 System Software Vx.x [dd mm yy]
  Copyright (C) xxxx NEC Corporation

Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y

USER-SYSTEM YDD-ON

SELECT CPU (78C11:1/78C14:4)=1
EXTERNAL MEMORY SIZE (0:NON,1:256,2:4K,3:16K,4:60K)=2
RAE (E:ENABLE,D:DISABLE) = D
PORT-FMODE (1:INPUT,0:OUTPUT)
PF-01234567
  XXXX0011

I> MAP
$ IE-INTROM MAPPING $
0000-0FFF
$ IE-INTRAM MAPPING $
$ USER $
$ NON MAPPING $
1000-FFFF
I>

```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
:
: ORG 0
: MVI A,00000100B ; RAE:D 4KBYTE MODE
: MOV MM,A ; SET MM-REG
: MVI A,11000000B ; 11000000 MODE
: MOV MF,A ; SET MF-REG

```

保守/廃止

例7 外部メモリ16Kバイト RAEビットオン

```

IE-78C11 System Software Vx.x [dd mmm yy]
  Copyright (C) xxxx NEC Corporation

Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y

USER-SYSTEM VDD-ON

SELECT CPU (78C11:1/78C14:4)=1
EXTERNAL MEMORY SIZE (0:NON,1:256,2:4K,3:16K,4:60K)= 3
RAE (E:ENABLE,D:DISABLE) = E
PORT-F MODE (1:INPUT,0:OUTPUT)
PF-01234567
  XXXXXXIQ

I> MAP
$ IE-INTRM MAPPING $
0000-0FFF
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
1000-FE00
I>

```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
  ORG      0
  MVI     A,00001110B      ; RAE:E 16KBYTE MODE
  MOV     MM,A             ; SET MM-REG
  MVI     A,01000000B      ; 01000000 MODE
  MOV     MF,A             ; SET MF-REG

```

保守/廃止

例 8 外部メモリ 16 K バイト RAE ビットオフ

```
IE-78C11 System Software Vx.x [dd mmm yy]
  Copyright (C) xxxx NEC Corporation

Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y

USER-SYSTEM VDD-ON

SELECT CPU (78C11:1/78C14:4)=1
EXTERNAL MEMORY SIZE (0:NON,1:256,2:4K,3:16K,4:60K)= 3
RAE (E:ENABLE,D:DISABLE) = D
PORT-F MODE (I:INPUT,O:OUTPUT)
PF-01234567
  XXXXXX01

I>MAP
$ IE-INTRM MAPPING $
0000-0FFF
$ IE-INTRAM MAPPING $
$ USER $
$ NON MAPPING $
1000-FE0F
I>
```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
  ORG      0
  MVI     A,00000110B      ; RAE:D 16KBYTE MODE
  MOV     MM,A            ; SET MM-REG
  MVI     A,10000000B     ; 10000000 MODE
  MOV     MF,A            ; SET MF-REG

```

保守/廃止

例 9 外部メモリ 60 K バイト R A E ビットオン

```

IE-78C11 System Software Vx.x [dd mmm yy]
  Copyright (C) xxxx NEC Corporation

Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y

USER-SYSTEM VDD-ON

SELECT CPU (78C11:1/78C14:4)=1
EXTERNAL MEMORY SIZE (0:NON,1:256,2:4K,3:16K,4:60K)=4
RAE (E:ENABLE,D:DISABLE) = E

1>MAP

$ IE-INTROM MAPPING $
0000-OFFF
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
1000-FE00
1>

```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
:
:   ORG      0
:   MVI     A,00001111B      ; RAE:E 64KBYTE MODE
:   MOV     MM,A            ; SET MM-REG

```


保守/廃止

例 10 外部メモリ 60K バイト RAE ビットオフ

```
IE-78C11 System Software Vx.x [dd mmm yy]
  Copyright (C) xxxx NEC Corporation

Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y

USER-SYSTEM VDD-ON

SELECT CPU (78C11:1/78C14:4)=1
EXTERNAL MEMORY SIZE (0:NON,1:256,2:4K,3:16K,4:60K)= 4
RAE (E:ENABLE,D:DISABLE) = D

I> MAP 1

$ IE-INTROM MAPPING $
0000-0FFF
$ IE-INTRAM MAPPING $
$ USER $
$ NON MAPPING $
1000-FFFF
I>
```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
:   ORG      0
:   MVI     A,00000111B      ; RAE:D 60KBYTE MODE
:   MOV     MM,A            ; SET MM-REG

```

保守/廃止

5.3 μ PD78C10のディバグ

μ PD78C10は μ PD78C11および μ PD78C14のROMなしバージョンです。そしてこの μ PD78C10は前に説明したMODE0, MODE1端子の状態により以下に示すような3種類のモードになります(モード・レジスタMMはRAEビット以外は使用せず0にしなければなりません。図5-2参照)。

図5-3 メモリ・マップ

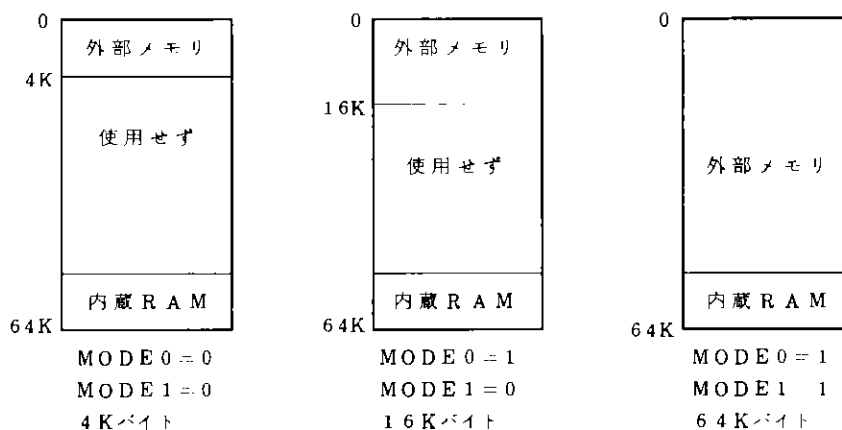


表5-4 PF0~7の動作

外部メモリ	PF0	PF1	PF2	PF3	PF4	PF5	PF6	PF7
4Kバイト	AB8	AB9	AB10	AB11	ポート	ポート	ポート	ポート
16Kバイト	AB8	AB9	AB10	AB11	AB12	AB13	ポート	ポート
64Kバイト	AB8	AB9	AB10	AB11	AB12	AB13	AB14	AB15

μ PD78C10のMMレジスタはRAEビットのみに使用します。

その他のビットは、すべて0にしなければなりません。

内蔵RAM (FF00H~FFFFH番地)については、このRAEビットで μ PD78C10と同様に制御ができます。

μ PD78C10のポートDは、マルチ・プレックスド・アドレス/データ・バスとして固定され、ポートFの下位4ビットは常にアドレス・バスとして使用し上位4ビットは三つのモード(4Kモードの場合すべてポート、16Kモードの場合上位2ビットのみポート、64Kモードの場合すべてアドレス)になります。

以下に μ PD78C10の場合と同様に初期化の入力例、出力例およびそれに対応する μ PD78C10の初期化プログラムの例を記しますので参考にしてください。

保守/廃止

例 11 外部メモリ 4 K バイト R A E ビットオン

```
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright (C) xxxx NEC Corporation
```

```
Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y
```

```
USER-SYSTEM VDD-ON
```

```
CPU 78C10
EXTERNAL MEMORY SIZE 4K BYTE
RAE (E:ENABLE,D:DISABLE) = E
PORT-F MODE (I:INPUT,O:OUTPUT)
PF-01234567
XXXX1010
```

```
1>MAP
```

```
$ IE-INTRAM MAPPING $
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
0000-FE00
1>
```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
:
:   ORG      0
:   MVI     A,00001000B ; RAE:E
:   MOV     MM,A        ; SET MM-REG
:   MVI     A,01010000B ; 01010000 MODE
:   MOV     MF,A        ; SET MF-REG

```

保守/廃止

例 12 外部メモリ 4 K バイト R A E ビットオフ

```
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright (C) xxxx NEC Corporation
```

```
Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y
```

```
USER-SYSTEM VDD-ON
```

```
CPU 78C10
EXTERNAL MEMORY SIZE 4K BYTE
RAE (E:ENABLE,D:DISABLE) = D
PORT-F MODE (1:INPUT,0:OUTPUT)
PF-01234567
XXXXQIQI
```

```
1> MAP
```

```
$ IE-INTRAM MAPPING $
$ IE-INTRAM MAPPING $
$ USER $
$ NON MAPPING $
0000-FFFF
```

```
1>
```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
ORG      0
MVI     A,00000000B ; RAE:D
MOY     MM,A        ; SET MM-REG
MVI     A,10100000B ; 10100000 MODE
MOY     MF,A        ; SET MF-REG

```

保守/廃止

例 13 外部メモリ 16 K バイト R A E ビットオン

```
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright (C) xxxx NEC Corporation
```

```
Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y
```

```
USER-SYSTEM VDD-ON
```

```
CPU 78C10
EXTERNAL MEMORY SIZE 16K BYTE
RAE (E:ENABLE,D:DISABLE) = E
PORT-F MODE (I:INPUT,O:OUTPUT)
PF-01234567
XXXXXX11
```

```
1> MAP
```

```
$ IE-INTRM MAPPING $
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
0000-FE00
```

```
1>
```

以下は上記のセッティングに対応した初期化プログラムです。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
:
:   ORG      0
:   MVI     A,00001000B      ; RAE:E
:   MOV     MM,A            ; SET MM-REG
:   MVI     A,11000000B     ; 11000000 MODE
:   MOV     MF,A            ; SET MF-REG

```

保守/廃止

例 14 外部メモリ 16 K バイト R A E ビットオフ

```
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright (C) xxxx NEC Corporation
```

```
Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y
```

```
USER-SYSTEM VDD-ON
```

```
CPU 78C10
EXTERNAL MEMORY SIZE 16K BYTE
RAE (E:ENABLE,D:DISABLE) = D
PORT-F MODE (I:INPUT,O:OUTPUT)
PF-01234567
XXXXXX00
```

```
1> MAP 2
```

```
$ IE-INTROM MAPPING $
$ IE-INTROM MAPPING $
$ USER $
$ NON MAPPING $
0000-FFFF
```

```
1>
```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
ORG      0
MVI     A,00000000B    ; RAE:D
MOV     MM,A          ; SET MM-REG
MVI     A,00000000B    ; 00000000 MODE
MOV     MF,A          ; SET MF-REG

```

保守/廃止

例 15 外部メモリ 64 Kバイト RAEビットオン

```
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright (C) xxxx NEC Corporation
```

```
Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y
```

```
USER-SYSTEM VDD-ON
```

```
CPU 78C10
EXTERNAL MEMORY SIZE 64K BYTE
RAE (E:ENABLE,D:DISABLE) = E
```

```
1> MAP
```

```
$ IE-INTROM MAPPING $
$ IE-INTROM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
0000-FE00
```

```
1>
```

以下は上記のセッティングに対応した初期化プログラムの例です。

```

:
: SAMPLE OF MODE INITIALIZE PROGRAM
:
:
:   ORG      0
:   MVI     A,00001000B      ; RAE:E
:   MOV     MM,A            ; SET MM-REG
:
:
:

```

保守/廃止

例 16 外部メモリ 64 K バイト RAE ビットオフ

```
IE-78C11 System Software Vx.x [dd mmm yy]
Copyright (C) xxxx NEC Corporation
```

```
Select Channel Number ? (1/2/3/4) : 1
Initialize ? (Y/N) : Y
```

```
USER-SYSTEM VDD-ON
```

```
CPU 78C10
EXTERNAL MEMORY SIZE 64K BYTE
RAE (E:ENABLE,D:DISABLE) = D
```

```
1> MAP
```

```
$ IE-INTRAM MAPPING $
$ IE-INTRAM MAPPING $
$ USER $
$ NON MAPPING $
0000-FFFF
```

```
1>
```

以下は上記のセッティングに対応した初期化プログラムの例です。

```
...
SAMPLE OF MODE INITIALIZE PROGRAM
...
ORG      0
MYI     A,00000000B      ; RAE:D
MOV     MM,A             ; SET MM-REG
```


保守/廃止

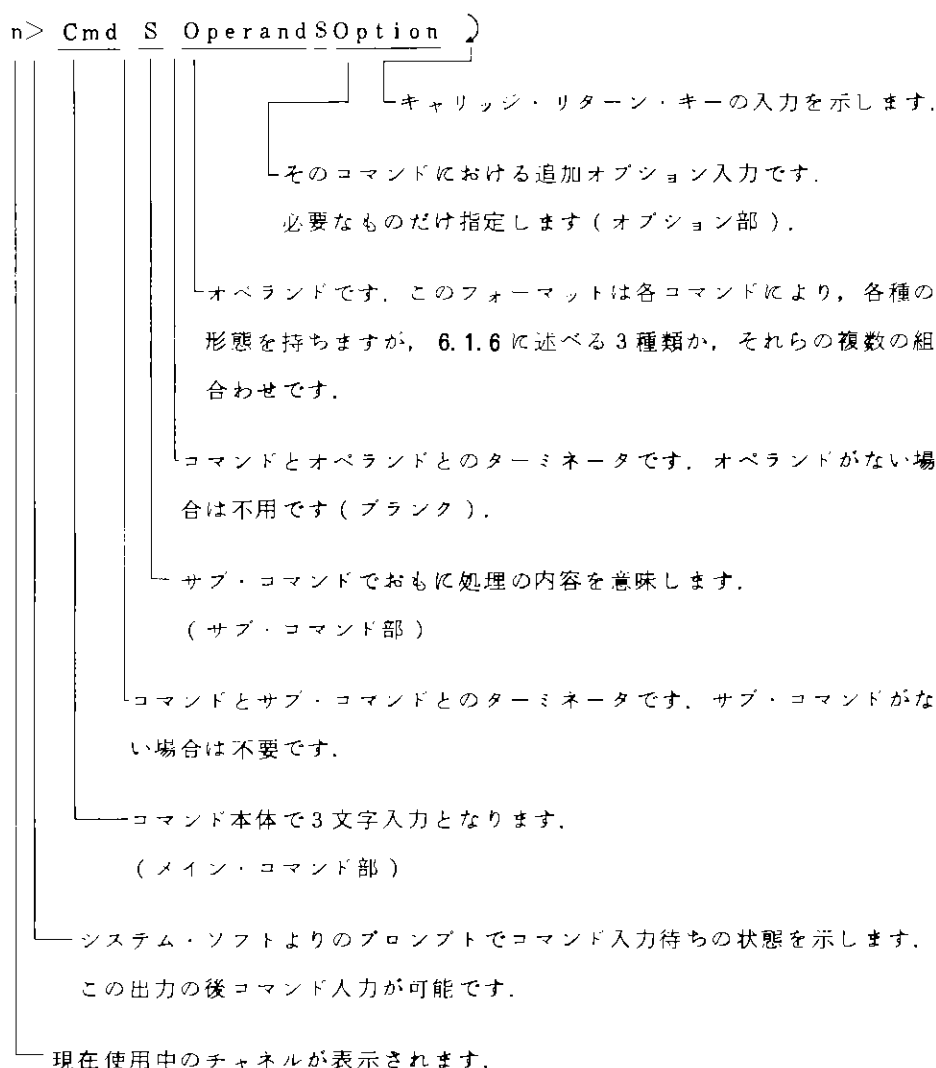
第6章 コマンド形態

6.1 コマンドの基本入力形式と表現法

ここでは、各コマンドについての入力形式や本マニュアルでの表現法についてまとめてあります。

6.1.1 コマンド入力形式

以下に入力コマンドの基本形を示します。



注 英字については、大文字のみ受け付けます。

保守/廃止

6.1.2 マニュアルにおける表現形式

大文字	実際にその文字が出力されるか入力することを意味します(6.1.3, 6.1.4 参照).
小文字	オペランド部の指定
)	キャリッジ・リターン入力です.
{ 文字列 :	{ } の中に記述されている文字列のどれかを選ぶことを意味します.
[文字列 : : :	[] の中に記述されている文字列のどれかを選ぶか、またはまったく入力をしていないでもよいことを意味します. 入力しなかった場合の処理は各コマンドで説明します.
文字列.....	その文字列に相当する入力を連続して入力することが可能であることを示しています.

6.1.3 メイン・コマンド一覧

MAT	:	演算コマンド
SUF	:	数値基数指定コマンド
MAP	:	マッピング・コマンド
SYM	:	シンボル・コマンド
RES	:	リセット・コマンド
CLK	:	クロック・コマンド
MEM	:	メモリ・コマンド
ASM	:	アセンブル・コマンド
DAS	:	逆アセンブル・コマンド
MOV	:	メモリ転送コマンド
LOD	:	ロード・コマンド
SAV	:	セーブ・コマンド
VRV	:	ベリファイ・コマンド
REG	:	レジスタ・コマンド
MDR	:	モード・レジスタ・コマンド
SPR	:	スペシャル・レジスタ・コマンド
BR?	:	ブレーク・コントロール・コマンド群
TR?	:	トレース・コントロール・コマンド群
RUN	:	エミュレーション・コマンド
STR	:	ストリング・コマンド
DIG	:	自己診断コマンド
DIR	:	ディレクトリ・コマンド
LST	:	リスト・コマンド
HLP	:	ヘルプ・コマンド
MOD	:	モジュール・コマンド

保守/廃止

PAG : ページ・コマンド
 EXT : 終了コマンド
 PGM : PGMコマンド

注1 ' ? ' は、BR? に対してはA, D, E, T, 0~3, Mのうちいずれかが対応し、
 TR? に対してはM, S, X, C, D, Pが対応します。

2 PGMコマンドはIE-7811H-Cが保有するコマンドでIE-7811H-Mでの使用はできません。

6.1.4 サブ・コマンド一覧

A : Address or Append
 B : Break
 C : Change, Code or Condition
 D : Display
 E : Examination
 F : Fill
 G : Get
 H : Hard
 I : IE or Instruction
 K : Kill
 L : Loop
 M : Move or Mode
 N : Normal
 PB : Port B
 R : internal Rom
 S : Step, Select or Symbol
 T : Trace
 U : User
 V : Verify
 W : internal r/W memory
 X : eXchange

6.1.5 オペランド部一覧

addr	アドレス値 (16ビット)
addrx	アドレス値による数値範囲
partition	addr, addrおよびaddrxによる数値範囲
addrs	addr, partitionおよびpartitionsによる アドレス群

保守/廃止

<code>partitions</code>	<code>partition</code> をスペースでくぎり、複数個の入力形式
<code>data-string</code>	‘,’ で区切られたデータ列 (8ビット, 最大10個)
<code>register-name</code>	レジスタ名
<code>mode-register-name1</code>	モード・レジスタ名 (セット可能)
<code>mode-register-name2</code>	モード・レジスタ名 (リード可能)
<code>special-register-name1</code>	特殊レジスタ名 (セット可能)
<code>special-register-name2</code>	特殊レジスタ名 (リード可能)
<code>filename</code>	人出力ファイルに関する指定
<code>step-No.</code>	ステップ数 (16ビット)
<code>value</code>	数値データ (8ビット, 16ビット)
<code>expression</code>	演算式
<code>line-No.</code>	トレース・ライン数 (10ビット)
<code>values</code>	マスク表現可能な数値データ
<code>command</code>	コマンド名
<code>module-name</code>	モジュール名
<code>filematch</code>	ファイル名の省略
<code>symbol</code>	シンボル
<code>vreg</code>	Vレジスタ値

6.2 オペランドの概要

6.1.5 でオペランドの一覧を示しましたが、この22種類の内容は基本的には次の4種類に大別されます。

6.2.1 式

ある具体的数値を表現します。

`addr`, `step-No.`, `value`, `expression`, `line-No.`, `vreg` がこれに相当します。

6.2.2 数値群

ある具体的数値の一群を表現するものです。

一般に対応するコマンドではその一群すべてに対して処理を行なうこととなります。

`partition`, `addrs`, `partitions`, `data-string`, `values` がこれに相当します。

6.2.3 予約記号

予約された文字列のうち一つを、そこに使用します。そのときの文字列が指す対象に対して処理を行ないます。

`register-name`, `mode-register-name1`, `mode-register-name2`, `special-register-name1`, `special-register-name2` がこれにあたります。

保守/廃止

6.2.4 その他

6.2.1～6.2.3に含まれないものです。

symbol, command, filename, module-name, filematchがあります。

6.3 式の入力

オペランドが式となっている部分に対しては、直接数値を入力すること、あらかじめ登録されているシンボルを入力してそれが示す数値のかわりとする、またそれらを演算子によって組合わせた形で一つの数値を代表させることができます。

たとえば“7”という入力も“4+3”という入力も“NANA”(NANAが7という数値を代表している)という入力もすべて式です。そしてそのどの入力も7という数値を入力したことになります。

6.3.1 数値の入力

数値は0～9, A～Fの文字列に最後に基数文字を付けた形が基本形です。

ここで基数とは、その文字列が16進, 10進, 8進, 2進のいずれかを指示するものです。以下にその一覧を示します。

基数文字	進 数
H	16進
T	10進
Q	8進
Y	2進

もし、基数文字を省略した場合は現在設定されている内部基数が採用されます(この内部基数は起動時は16進に設定され、変更や表示はSUFコマンドで行なえます)。

基数が16進の場合、頭の1文字はかならず0～9でなければいけません。もしA～Fで始めるとシンボルと解釈されるからです。ただし、その文字列のシンボルが登録されていない場合にかぎって16進数として解釈されます。

また、有効桁数以上の入力があった場合は、ビット・イメージに変換されたのち有効部が採用されます(ただし、無視される上位桁の中に0～9, A～F以外の文字があるとエラーとなります)。有効桁数は各オペランドの種類により異なりますので各コマンドの説明を参照してください。

以下に記述例を示しますので参考にしてください。

保守/廃止

内部基数	入 力	値 (10進)
16進の場合	1 0	1 6
	1 0 1	2 5 7
10進の場合	1 0	1 0
	1 0 1	1 0 1
8進の場合	1 0	8
	1 0 1	8 5
2進の場合	1 0	2
	1 0 1	5

例

	入 力	16進	10進
	1 2 A 4 F 6 H (16ビット入力の場合)	A 4 F 6	4 5 0 4 6
	1 F 5 9 B C H (8ビット入力の場合)	B C	1 8 8
	1 3 2 5 2 T (16ビット入力の場合)	3 3 C 4	1 3 2 5 2
	1 3 2 5 2 T (8ビット入力の場合)	C 4	1 9 6
	1 1 1 0 1 1 0 1 0 Y (16ビット入力の場合)	1 D 2	4 6 6
	1 1 1 0 1 1 0 1 0 Y (8ビット入力の場合)	D 2	2 1 0
	1 B Y	エラー	
	1 G 6 6 Q	エラー	
	0 X 0 0 1 3 6 H	エラー	

6.3.2 シンボルの入力

シンボルはアセンブラにより作成されたシンボルおよびSYMコマンドにより作成したシンボルが使用できます。シンボルとして登録された文字列は、その文字列をオペランド部に入力すると、そのシンボルがもつ値が入力されたとみなして処理されます。またシンボルは後述するASMコマンドおよびDASコマンドも使用されます。

アセンブラより作成されたシンボルは、ローカルおよびパブリックの2種類があります。さらに各シンボルは三つのタイプのうちの一つをもちます。このうちタイプはコマンドのオペランド部では区別されません(各種の逆アセンブル出力時に区別されます)。ローカルおよびパブリックの入力は、同一シンボル名が各モジュール内に存在する可能性があるため、以下の方法で選択できるようになっています。

(a) パブリック・シンボルの入力

パブリック・シンボルはそのままそのシンボル名を入力することで参照できます。

例 MEM C START)

保守/廃止

(b) ローカル・シンボルの入力

そのローカル・シンボルが属しているモジュール名をシンボル名の前に「 [] 」でかこんで入力することで指定します。

例 ASM [SUB] IN0)

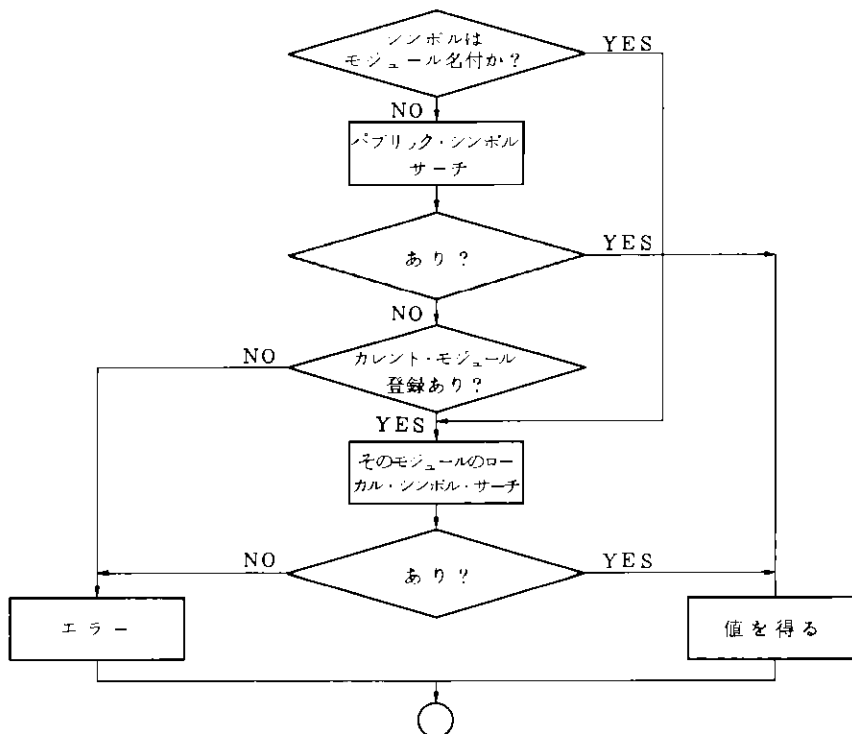
(c) カレント・モジュールによる入力

ローカル・シンボルを入力するときに毎回モジュール名を入力するのはめんどりです。そのためカレント・モジュールという概念をIEはもっています。カレント・モジュールはMODコマンドにより設定され、一度設定されると次に再度MODコマンドによって新しいモジュール名が設定されるまでそのモジュール名が有効になります。この状態でシンボル名を入力するとそのモジュール内のシンボルが指定されます。

例 MOD SUB)
ASM IN0)

(d) シンボルの選択フロー

以上の内容の処理の流れを以下に示します。



保守/廃止

6.3.3 演算子付の式の入力

オペランド入力において、数値およびシンボルの演算（＋，－，＊，／，（，），AND，OR，XOR）が可能です。

実行例を以下に示します。

```

I> MEM D 2+5, 29*2 )
0007 00 00 00 00 00 00 00 00 00
0010 00 00 00 00 00 00 00 00 00
0020 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00
0050 00 00 00
I> MEM D 2E/2, 20 )
0017 00 00 00 00 00 00 00 00 00
0020 00
I> MEM D 3 AND 7, F )
0013 00 00 00 00 00 00 00 00 00
I>

```

6.3.4 式入力のオペランド一覧

- (a) `addr`
メモリのアドレスを意味しており、16ビット長です。
- (b) `step-No.`
ステップ実行のステップ数で16ビット長です（しかし、0、1、2という値は使用できません）。
- (c) `value`
8ビットまたは16ビット長です。
- (d) `expression`
MATコマンドでのオペランドです。
16ビット長です。
- (e) `line-No.`
トレース関係コマンドでの1トレース単位を意味しています。
10ビット長です。
- (f) `vreg`
PAGコマンドでVレジスタに与える値です。
8ビット長です。

6.4 数値群の入力

数値群とはある値からある値までのすべての数値、n個の単独の数値のあつまり、ある値に対してのマスク処理、の三つの種類の総称です。各オペランドにより多少入力形式が異なりますので個々に説明を行いません。

保守/廃止

6.4.1 partition

ある二つのアドレス(16ビット長)の範囲を示すものです。入力法としては二つの方法が選べます。一つは式の項で述べた addr を二つ', ' で区切って入力する方法で、もう一つは、数値入力形式の値の最下位桁から任意桁までを 'X' 文字で連続にうめることにより 'X' でうめた桁が 0 から現在指定されている基数の最大までを範囲とみなす方法です(ただし、基数が10進の場合はこの方法は利用できません)。

以下に二つの入力法の例を示しますので参考にしてください。

(a) addr, addr形式

0, FF	0H~0FFHまで(FFがシンボルにないとき) 0~FF(FFというシンボル)のアドレスまで (FFがシンボルにあるとき)
START, END	STARTが示すアドレスからENDが示すアドレスまで

(b) X文字入力形式

0XX(内部基数16進)	0000~00FF(16進)
0XX(" 2進)	0000~0003(16進)
1XX(" 16進)	0100~01FF(16進)
XXX(" 16進)	XXXというシンボルがある場合は、そのシンボル値となります。もしシンボルにXXXがない場合は、0000~0FFF(16進)という意味になりますが、できるだけ次の行のように左桁に0をつけた入力をおすすめします。
0XXX(内部基数16進)	0000~0FFF(16進)
0XXY(" 2進)	0000~0003(16進) (Yの基数が2進を意味します)
X1X	エラー(Xが連続でない)
XX1234H	エラー(Xが上位桁にある)

6.4.2 partitions

partitionが1組のアドレスを示しているのに対して、partitionsは複数のアドレス範囲を指定できます(1組でも許されます)。記述方法はpartitionの入力形式をスペースで区切りただ列記することで実現します。これはSAVコマンドでのオペランドの入力で使用します。

保守/廃止

例

```

0, FF□200, 23F□500, 5FF□700, 7FF (内部基数16進)
    0000~00FF (16進)
    0200~023F (16進)
    0500~05FF (16進)
    0700~07FF (16進)

START, START+0FFH□DATA, DATA□OR□100H□1000H, 10FFH
    START~START+0FFH
    DATA~DATA□OR□100H
    1000H~10FFH

START, END
    START~END (1組も可能)

```

6.4.3 addr

partitions が複数のアドレス範囲を示しているのに対し、addr はそれに加えて単なるアドレス値を addr 形式で入力し、それをスペースで区切って入力することもできるものです。この形式は主にブレイク・アドレスの入力で複数のアドレスの組合せをブレイク・ポイントに設定するときに利用します。

形式は次のとおりです。

$$\left\{ \begin{array}{l} \text{addr} \\ \text{partition} \end{array} \right\} \square \left\{ \begin{array}{l} \text{addr} \\ \text{partition} \end{array} \right\} \square \dots$$

6.4.4 data-string

MEM G および MEM F 等を対象とする1バイト・データの一連のつらなりを表すものです。入力形式は

value(8), value(8)

となります。

6.4.5 values

X 表現が可能な8ビット・データです。

6.5 予約記号

予約記号はほとんどが μ PD78C11, 78C10 の持つレジスタの名前といえます。5種類はそれぞれが次に示すレジスタ群を代表しています。

保守/廃止

6.5.1 register-name

汎用レジスタ群です。以下に示すものがあります。

PSW, V, A, B, C, D, E, H, L, EA, SP, PC, V', A', B', C', D', E',
H', L', EA

6.5.2 mode-register-name1

モード・レジスタ・ネームです。以下に示すものがあります。

MA, MB, MCC, MC, MM, MF, TMM, ETMM, EOM, SML, SMH, ANM, ZCM

6.5.3 mode-register-name2

mode-register-name1のうち、 μ PD78C11, 78C10, 78C14がREAD可能なものです。

以下に示すものがあります。

TMM, EOM, SMH, ANM

6.5.4 special-register-name1

μ PD78C11, 78C10, 78C14の有する特殊レジスタのなかでWRITE可能なレジスタで以下に示すものがあります。

PA, PB, PC, PD, PF, MKH, MKL, TXB, TM0, TM1, ETM0, ETM1

6.5.5 special-register-name2

μ PD78C11, 78C10, 78C14の有する特殊レジスタのREAD可能なレジスタで以下に示すものがあります。

PA, PB, PC, PD, PF, MKH, MKL, PXB, CR0, CR1, CR2, CR3, ECNT,
ECPT

6.6 その他

6.6.1 symbol

プログラムで使用されているシンボルを入力します。

6.6.2 command

メイン・コマンド名です。コマンド名を大文字3文字で入力します。

6.6.3 filename

このオペランドはファイルの入出力の場合に用います。

基本形式は次のようになります。

[drive:] primary-name[.extention-name]

このfilenameの表現は、通常のCP/M, MP/M-86で用いられるものと同じです。

保守/廃止

この filename のうち drive : が省略された場合は、カレント・ディスクとなります。

また、extention-name が省略された場合は、オブジェクト・コードの入出力に対しては 'HEX'、シンボル・ファイルに対しては 'SYM'、また LST コマンドでは 'TXT'、STR コマンドでは 'STR' として処理されます。

注 ファイル・ネームに使用できる文字としては、英字の大文字、数字に加えて @, ", #, %, +, -, ' があります。

6.6.4 module-name

プログラム中に作成されているモジュール名を入力します。

6.6.5 filematch

ファイル名を代行できる特殊キーで *, ? があります。

保守/廃止**第7章 コマンド詳細**

この章では、各コマンドについて説明します。

例に用いるシンボル、シンボル値およびシンボル・タイプは以下のとおりです。

モジュール名	パブリック・シンボル			ローカル・シンボル		
	シンボル名	シンボル値	タイプ	シンボル名	シンボル値	タイプ
MMAIN	SS	0006	C	START	0000	C
				OWARI	001F	C
MSO	CLRFF	0022	C	CLRFE	0026	C

保守/廃止**7.1 ヘルプ・コマンド**

```
HLP[ command ] )
```

command で指定したコマンドの入力形式を表示します。

command の指定を省略した場合、コマンドの説明を順次表示します。

このとき 1 画面毎に Next (Y/N)? と出力されますので続行の場合 'Y' を入力してください。

例

```
1 > HLP ) _____ (1)
```

```
1 > HLP MEM ) _____ (2)
```

(1) コマンド入力形式を順次表示します。

(2) MEM コマンドの入力形式を表示します。

保守/廃止

実行例

```

1> HLP )
HLP COMMAND
  1> HLP [command]
MAT COMMAND
  1> MAT expression
SUF COMMAND
  1> SUF
  1> SUF H
  1> SUF T
  1> SUF Q
  1> SUF Y
Next (Y/N)? Y ----- Yと入力すると次のコマンドを表示します。
MAP COMMAND
  1> MAP
  1> MAP W
  1> MAP R
  1> MAP U
  1> MAP K
  1> MAP W partition
  1> MAP R partition
  1> MAP U partition
  1> MAP K partition
SYM COMMAND
  1> SYM D [[module-name] symbol]
  1> SYM C [[module-name] symbol]
  1> SYM A symbol
  1> SYM K
Next (Y/N)? N ----- Nと入力するとプロンプトにもどります。
1> HLP RES )
RES COMMAND
  1> RES
  1> RES H
1>

```

保守/廃止

7.2 演算コマンド

MAT expression)

expression (式) の演算結果を 16 進, 10 進, 8 進, 2 進で表示します。

数値の後に基数 (H, T, Q, Y) が省略された場合は S U F コマンド (7.3 参照) の指定に従います。

演算子は以下のとおりです。

四則演算	+	:	加算
	-	:	減算
	*	:	乗算
	/	:	除算
論理演算	A N D	:	論理積
	O R	:	論理和
	X O R	:	排他論理和
	N O T	:	否定

カッコ処理 () :

カッコ処理の使用回数 (ネスティング) は 32 組までです。

ただし, 入力数値, 演算結果が 2 バイトを越える時は下位 2 バイトのデータが有効となります。

演算優先順位

- 1 ()
- 2 *, /
- 3 +, -
- 4 N O T
- 5 A N D
- 6 O R, X O R

例

- 1 > MAT 5H+7H) _____ (1)
- 1 > MAT 5 AND 7) _____ (2)
- 1 > MAT 3*(2+5)) _____ (3)
- 1 > MAT SS+[MMAIN]OWARI) _____ (4)

(1) 5H+7H の演算を行ない, 結果を 16 進, 10 進, 8 進, 2 進で表示します。

(2) 5 AND 7 の演算を行ない, 結果を 16 進, 10 進, 8 進, 2 進で表示します。論理演算を行なう場合は, 演算子の前後にスペースを入力してください。

(3) () 表現が可能です。

(4) シンボルが使用できます。

保守/廃止

実行例

I> MAT 5H+7H)CH,12T,14Q,1100Y
I> MAT 5 AND 7)5H,5T,5Q,101Y
I> MAT 3*(2+5))15H,21T,25Q,10101Y
I> MAT SS+(MMAIN)OWARI)24H,36T,44Q,100100Y
I> MAT 101011Y/13Q)3H,3T,3Q,11Y
I> MAT 5 AND 1)

1H,1T,1Q,1Y

保守/廃止

7.3 数値基数指定コマンド

$$\text{SUF} \left[\left\{ \begin{array}{c} \text{H} \\ \text{T} \\ \text{Q} \\ \text{Y} \end{array} \right\} \right])$$

H : 16進

T : 10進

Q : 8進

Y : 2進

入力する数の基数指定および、現在指定されている基数を表示します。

保守/廃止

7.3.1 サフィックス指定

SUF { H
T
Q
Y })

指 定

入力数値の表現を指定します。

初期値はH(16進)が指定されています。

例

1> SUF T)

入力する数値表現を10進数に指定します。

実行例

1> SUF H)

1> MAT 10)

10H,16T,20Q,10000Y

1> SUF T)

1> MAT 10)

AH,10T,12Q,1010Y

1>



7.3.2 サフィックス表示

SUF)

現在指定されている基数を表示します。

例

1> SUF)

実行例

1> SUF T)

1> SUF)

T

1>

保守/廃止

7.4 マッピング・コマンド

$$\text{MAP} \left[\begin{array}{l} \text{W} \\ \text{R} \\ \text{U} \\ \text{K} \end{array} \right] \left[\text{partition} \right] \left. \vphantom{\begin{array}{l} \text{W} \\ \text{R} \\ \text{U} \\ \text{K} \end{array}} \right\}$$

64 Kバイトのメモリ空間において、使用する領域の配置を設定または表示します。

W : エミュレーションRAMエリア
 R : エミュレーションROMエリア
 U : ユーザ・システム
 K : 解除

IE-78C11は、最大64 Kバイトのメモリをアクセスできます。

このメモリは256バイトごとにユーザのターゲット・システムのメモリやIE-78C11内部のエミュレーション・メモリにマッピングすることができます。

マッピングの種類には、4種類^注ありMAPコマンドにより設定します。

ただし、電源投入時、システム・リセット時のモード設定により自動的にROMマッピング、およびRAMマッピングされる部分もあります(5章参照)。

ユーザ・システムが未完成でもエミュレーション・メモリをROM、RAMとしてマッピングすることによりソフトウェア・デバッグが可能となります。

注 エミュレーション・メモリをRAMとしてマッピング
 エミュレーション・メモリをROMとしてマッピング
 ユーザ・システムにマッピング
 マッピングしない

エミュレーション・メモリをROMとしてマッピングすると、エミュレーション中に、その領域をライト・アクセスした場合プロテクトされます。

保守/廃止

7.4.1 マッピング指定

$$\text{MAP} \left\{ \begin{array}{l} \text{W} \\ \text{R} \\ \text{U} \\ \text{K} \end{array} \right\} \text{partition)}$$

マッピング指定の可能な範囲は次の(1)~(4)の条件により異なります。

範囲を越えて指定された場合は“MAPPING ERROR”となり、入力されたコマンドは無効となります。

(1) ターゲットCPUがμPD78C10でRAEがENABLEの時

エミュレーションRAMエリア	(W)	0 0 0 0 - F F F F
エミュレーションROMエリア	(R)	0 0 0 0 - F E F F
ユーザ・システム・エリア	(U)	0 0 0 0 - F E F F
解 除	(K)	0 0 0 0 - F E F F

(2) ターゲットCPUがμPD78C10でRAEがDISABLEの時

エミュレーションRAMエリア	(W)	0 0 0 0 - F F F F
エミュレーションROMエリア	(R)	0 0 0 0 - F F F F
ユーザ・システム・エリア	(U)	0 0 0 0 - F F F F
解 除	(K)	0 0 0 0 - F F F F

(3) ターゲットCPUがμPD78C11でRAEがENABLEの時

エミュレーションRAMエリア	(W)	1 0 0 0 - F F F F
エミュレーションROMエリア	(R)	0 0 0 0 - F E F F
ユーザ・システム・エリア	(U)	1 0 0 0 - F E F F
解 除	(K)	1 0 0 0 - F E F F

(4) ターゲットCPUがμPD78C11でRAEがDISABLEの時

エミュレーションRAMエリア	(W)	1 0 0 0 - F F F F
エミュレーションROMエリア	(R)	0 0 0 0 - F F F F
ユーザ・システム・エリア	(U)	1 0 0 0 - F F F F
解 除	(K)	1 0 0 0 - F F F F

(5) ターゲットCPUがμPD78C14でRAEがENABLEの時

エミュレーションRAMエリア	(W)	4 0 0 0 - F F F F
エミュレーションROMエリア	(R)	0 0 0 0 - F E F F
ユーザ・システム・エリア	(U)	4 0 0 0 - F E F F
解 除	(K)	4 0 0 0 - F E F F

(6) ターゲットCPUがμPD78C14でRAEがDISABLEの時

エミュレーションRAMエリア	(W)	4 0 0 0 - F F F F
エミュレーションROMエリア	(R)	0 0 0 0 - F F F F
ユーザ・システム・エリア	(U)	4 0 0 0 - F F F F
解 除	(K)	4 0 0 0 - F F F F

保守/廃止

例

```

1> MAP W 0, FF ) _____ (1)
1> MAP R 1000, 1100 ) _____ (2)
1> MAP U 2000, 2F01 ) _____ (3)
1> MAP K 0, FF ) _____ (4)

```

(1) プログラム・メモリの 0H～FFH 番地のアドレス空間をエミュレーション RAM に指定します。

(2) プログラム・メモリの 1000H～11FFH 番地のアドレス空間をエミュレーション ROM に指定します。

エミュレーション ROM に指定されたエリアは実行中の書込みが禁止されます。

(3) プログラム・メモリの 2000H～2FFFH 番地のアドレス空間をユーザ・システム側に指定します。

(4) プログラム・メモリの 0H～FFH 番地のアドレス空間のマッピング指定を解除します。

注 マッピングは、256 バイトのバウンダリで行なわれるために、スタート・アドレスの下位 8 ビットは 00H にエンド・アドレスの下位 8 ビットは FFH にそれぞれ設定されます。

実行例

```

1> MAP W 0, FF ) _____ 0-FFまでを256バイト単位で内部RAM
                               にマッピングします。
1> MAP W )
$ IE-INTRAM MAPPING $
0000-00FF
1> MAP R 1000H, 1100H )
1> MAP R )
$ IE-INTRAM MAPPING $
1000-11FF _____ マッピング指定は1000-1100ですが256
                               バイト単位ですから1000-11FFとなります。
1> MAP U 2000H, 2F01H )
1> MAP U )
$ USER $
2000-2FFF
1> MAP )
$ IE-INTRAM MAPPING $
1000-11FF
$ IE-INTRAM MAPPING $
0000-00FF
$ USER $
2000-2FFF
$ NON MAPPING $
0100-0FFF 1200-1FFF 3000-FFFF
1> MAP K 0, FF )
$ IE-INTRAM MAPPING $
1000-11FF
$ IE-INTRAM MAPPING $
$ USER $
2000-2FFF
$ NON MAPPING $
0000-0FFF 1200-1FFF 3000-FFFF
1>

```

保守/廃止

7.4.2 マッピング表示

$$\text{MAP} \left[\begin{array}{c} \text{W} \\ \text{R} \\ \text{U} \end{array} \right] \text{)}$$

例

```

1> MAP W ) _____ (1)
1> MAP R ) _____ (2)
1> MAP U ) _____ (3)
1> MAP ) _____ (4)

```

- (1) エミュレーションRAMにマッピングされたエリアを表示します。
- (2) エミュレーションROMにマッピングされたエリアを表示します。
- (3) ユーザ・システム側にマッピングされたエリアを表示します。
- (4) マッピングされたエリアをすべて表示します。

実行例

```

1> MAP )
$ IE-INTRAM MAPPING $
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
0000-FE00

1> MAP W 0,FF )
1> MAP )
$ IE-INTRAM MAPPING $
$ IE-INTRAM MAPPING $
0000-00FF FF00-FFFF
$ USER $
$ NON MAPPING $
0100-FE00

1> MAP K 0,FF )
1> MAP )
$ IE-INTRAM MAPPING $
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
0000-FE00
1> MAP W [MMAIN]START,[MMAIN]OWARI )
1> MAP )
$ IE-INTRAM MAPPING $
$ IE-INTRAM MAPPING $
0000-00FF FF00-FFFF
$ USER $
$ NON MAPPING $
0100-FE00

```


保守/廃止

1> MAP K [MMAIN]START,[MMAIN]OWARI)

1> MAP)

```
$ IE-INTRAM MAPPING $  
$ IE-INTRAM MAPPING $  
FF00-FFFF  
$ USER $  
$ NON MAPPING $  
0000-FE00
```

1> MAP W)

```
$ IE-INTRAM MAPPING $  
FF00-FFFF
```

1> MAP R)

```
$ IE-INTRAM MAPPING $
```

1> MAP U)

```
$ USER $
```

1>

保守/廃止

7.4.3 マッピング解除

MAP K)

I E - 7 8 C 1 1 システムを起動時に設定されたマッピング状態にします。

- CPUが μ PD78C11でRAEがENABLEのとき

実行例

1> MAP K)

1> MAP)

```
$ IE-INTROM MAPPING $
0000-0FFF
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
1000-FE00
```

1>

- CPUが μ PD78C11でRAEがDISABLEのとき

実行例

1> MAP K)

1> MAP)

```
$ IE-INTROM MAPPING $
0000-0FFF
$ IE-INTRAM MAPPING $
$ USER $
$ NON MAPPING $
1000-FFFF
```

1>

- CPUが μ PD78C10でRAEがENABLEのとき

実行例

1> MAP K)

1> MAP)

```
$ IE-INTROM MAPPING $
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
0000-FE00
```

1>

保守 / 廃止

- CPUが μ PD78C10でRAEがDISABLEのとき

実行例

```

1> MAP K )
1> MAP )
$ IE-INTROM MAPPING $
$ IE-INTRAM MAPPING $
$ USER $
$ NON MAPPING $
0000-FFFF
1>

```

- CPUが μ PD78C14でRAEがENABLEのとき

```

1> MAP K )
1> MAP )
$ IE-INTROM MAPPING $
0000-3FFF
$ IE-INTRAM MAPPING $
FF00-FFFF
$ USER $
$ NON MAPPING $
4000-FE0F
1>

```

- CPUが μ PD78C14でRAEがDISABLEのとき

```

1> MAP K )
1> MAP )
$ IE-INTROM MAPPING $
$ IE-INTRAM MAPPING $
$ USER $
$ NON MAPPING $
0000-FFFF
1>

```

保守/廃止

7.5 シンボル・コマンド

$$\text{SYM} \left\{ \begin{array}{l} \text{D [[module-name] symbol]} \\ \text{C [[module-name] symbol} \\ \text{A symbol} \\ \text{K} \end{array} \right\})$$

D : 表示

C : 変更

A : 追加

K : 削除

シンボル・テーブルに登録されているシンボルの表示、変更、削除または追加を行なうコマンドです。

表 示

登録されているシンボル名、シンボル値、シンボル・タイプを表示します。

変 更

登録されているシンボルのシンボル値、シンボル・タイプの変更またはシンボルの削除を行ないます。

追 加

新しくシンボルを追加します。

削 除

登録されているシンボルをすべて削除します。

シンボル・タイプを以下に説明します。

シンボル・タイプ・コード	シンボル・タイプ名
N (0 0)	Number (定数)
C (0 1)	Code (ROM レーベル)
D (0 2)	Data (ROM レーベル)

保守/廃止

7.5.1 シンボル表示

```
SYM D [[module-name]symbol] )
```

シンボル値、シンボル・タイプを表示します。

例

```
1> SYM D ) _____ (1)
1> SYM D SS ) _____ (2)
1> SYM D [MS0]CLRFE ) _____ (3)
1> SYM D [MMAIN ] _____ (4)
```

- (1) 登録されているシンボルのシンボル値、シンボル・タイプを表示します。
- (2) シンボル“SS”のシンボル値、シンボル・タイプを表示します。
- (3) モジュール“MS0”のローカル・シンボル“CLRFE”のシンボル値、シンボル・タイプを表示します。
- (4) モジュール“MMAIN”に登録されているシンボルのすべてのシンボル値、シンボル・タイプを表示します。

保守/廃止

実行例

```

I> SYM D )
[MMAIN]
PUBLIC SYMBOL
SS      0006 C _____ シンボル・タイプ
LOCAL SYMBOL
OWARI   001F C   START   0000 C _____ シンボル値
[MS0]
PUBLIC SYMBOL
CLRFF   0022 C
LOCAL SYMBOL
CLRFE   0026 C
[APPEND] _____ 追加されたシンボルのモジュール
                        です。
ABC     0014 C
I> SYM D SS )
SS 0006 C
I> SYM D [MS0] CLRFE )
[MS0]CLRFE 0026 C
I> SYM D [MMAIN] )
[MMAIN]
PUBLIC SYMBOL
SS      0006 C
LOCAL SYMBOL
OWARI   001F C   START   0000 C
I>

```

保守/廃止

7.5.2 シンボル変更

```
SYM C [module-name]symbol )
```

登録されているシンボルのシンボル値、シンボル・タイプの変更およびシンボルの削除を行いません。

例

```
1> SYM C [MS0]CLRFE )----- (1)
```

```
1> SYM C SS )----- (2)
```

(1) モジュール“MS0”のローカル・シンボル“CLRFE”のシンボル値、シンボル・タイプを変更します。

(2) パブリック・シンボル“SS”のシンボル値、シンボル・タイプを変更します。

実行例

```
1> SYM C SS )
SS 0006 C - 0 N )
```

```
1> SYM D SS )
SS 0000 N
```

```
1> SYM C [MS0]CLRFE )
[MS0]CLRFE 0026 C - 30 C )
```

```
1> SYM D [MS0]CLRFE )
[MS0]CLRFE 0030 C
```

```
1> SYM C SS )
SS 0000 N - - )
```

削除する場合は‘-’（ハイフン）を入力します。

```
1> SYM D SS )
Symbol Not Found
```

```
1>
```

保守/廃止

7.5.3 シンボル追加

SYM A symbol)

シンボルを追加登録します。

例

I> SYM A XYZ)

シンボル“XYZ”を追加登録します。

ここで、追加されたシンボルはパブリック・シンボルであり、モジュール名は“APPEND”となります。

実行例

I> SYM D)

[MMAIN]

PUBLIC SYMBOL

SS 0006 C

LOCAL SYMBOL

OWARI 001F C START 0000 C

[MS0]

PUBLIC SYMBOL

CLRFF 0022 C

LOCAL SYMBOL

CLRFE 0030 C

I> SYM A XYZ)

XYZ - 100 C)

I> SYM A XYZ)

XYZ 0100 C - 120 C)

シンボル'XYZ'が存在する場合はSYM Cと同じ動作となります。

I> SYM D XYZ)

XYZ 0120 C

保守/廃止

I> SYM D)

[MMAIN]

PUBLIC SYMBOL

SS 0006 C

LOCAL SYMBOL

OWARI 001F C START 0000 C

[MSO]

PUBLIC SYMBOL

CLRFF 0022 C

LOCAL SYMBOL

CLRFE 0030 C

[APPEND]

XYZ 0120 C

I>

追加されたシンボルはモジュール名
'APPEND' のパブリック・シ
ンボルとなります。

保守/廃止

7.5.4 シンボル削除

SYM K)

例

I> SYM K)

登録されているシンボルをすべて削除します。

実行例

I> SYM D)

[MMAIN]

PUBLIC SYMBOL

SS 0006 C

LOCAL SYMBOL

OWARI 001F C START 0000 C

[MS0]

PUBLIC SYMBOL

CLRFF 0022 C

LOCAL SYMBOL

CLRFE 0030 C

[APPEND]

XYZ 0120 C

I> SYM K)I> SYM D) ————— シンボルがない場合は何も表示しません。

I>

保守/廃止

7.6 モジュール・コマンド

MOD [{ module }])

カレント・モジュールの指定，解除，および現在指定されているモジュール名を表示します。

これによって指定されたモジュールのローカル・シンボルは，モジュール名を入力せずにローカル・シンボルを入力することができます。

例

```

1> MOD MMA I N ) _____ (1)
1> MOD - ) _____ (2)
1> MOD ) _____ (3)

```

- (1) カレント・モジュールを MMA I N に指定します。
- (2) カレント・モジュールを解除します。
- (3) 現在指定されているカレント・モジュールを表示します。

保守 / 廃止

実行例

```

I> SYM D )
[MMAIN]
PUBLIC SYMBOL
SS      0006 C
LOCAL SYMBOL
OWARI   001F C      START   0000 C
[MSO]
PUBLIC SYMBOL
CLRFF   0022 C
LOCAL SYMBOL
CLRFE   0026 C
[APPEND]
ABC     0014 C
I> MEM D [MMAIN]START,[MMAIN]OWARI )
0000 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
0010 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
I> MOD MMAIN ) _____ モジュール名を指定します。

I> MEM D START,OWARI ) _____ モジュール名を省略できます。

0000 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
0010 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32
I> MOD ) _____ モジュール名の確認です。

MMAIN
I> MOD - ) _____ ' - ' (ハイフン) の入力で現在指定さ
                     れているカレント・モジュールを解除し
I> MOD ) _____ ます。
-
I>

```

保守/廃止

7.7 リセット・コマンド

RES [H])

ディバッガのハードウェアおよびエミュレーションCPUをリセットします。

Hが省略された場合は、エミュレーションCPUをリセットします。

例

1> RES) _____ (1)

1> RES H) _____ (2)

(1) エミュレーションCPUをリセットします。

(2) ディバッガのハードウェアおよびエミュレーションCPUをリセットします。

実行例

1> REG

```
PSW V A B C D E H L EA SP PC
00 00 00 FF 00 BB 41 00 00 0000 0700 276B
V' A' B' C' D' E' H' L' EA'
00 FF FF FF 01 40 00 00 014C
```

1> RES

システム・リセット後のレジスタの値は
保証できません。

1> REG

```
PSW V A B C D E H L EA SP PC
00 00 00 FF 00 BB 41 00 00 0000 0700 0000
V' A' B' C' D' E' H' L' EA'
00 FF FF FF 00 04 08 FD 014C
```

1> RES H

USER-SYSTEM VDD-ON

CPU 78C10
EXTERNAL MEMORY SIZE 64K BYTE
RAE (E:ENABLE,D:DISABLE)= E

1>

保守/廃止

7.8 クロック・コマンド

$$\text{CLK} [\left\{ \begin{array}{c} \text{U} \\ \text{I} \end{array} \right\}] \text{)}$$

U：ユーザ・システム側

I：内部クロック

クロック・ソースを指定，表示します。

指 定

Uが指定された場合はクロック・ソースとしてユーザ・システム側を使用します。

Iが指定された場合はクロック・ソースとして内部クロックを使用します。

表 示

現在指定されているクロック・ソース名を表示します。

注 クロックの変更はエミュレーションCPUをリセットした状態から行なっていますのでクロック変更後はエミュレーションCPUはリセット状態にあります。

保守/廃止

7.8.1 クロック指定

$$\text{CLK} \left\{ \begin{array}{l} \text{U} \\ \text{I} \end{array} \right\})$$

例

1 > CLK U) _____ (1)

1 > CLK I) _____ (2)

(1) クロック・ソースとしてユーザ・システム側のクロックを使用します。

(2) クロック・ソースとして内部クロックを使用します。

保守/廃止

7.8.2 クロック表示

CLK)

現在指定されているクロック・ソース名を表示します。

実行例

I> CLK 1)

I> CLK 2)

1
I>

保守/廃止

7.9 メモリ・コマンド

```
MEM C[ addr ] )
MEM D[ { addr
        } partition ] )
MEM E[ partition ][ $A ] )
MEM { F
      } partition data-string )
MEM { M
      } partition addr )
MEM { V
      }
MEM { X
      }
```

マッピングされているメモリの内容を変更、表示、検索、転送、イニシャライズ、交換、テストするコマンドです。

- 変更 (C) : プログラム・メモリの内容を変更します。
- 変更 (D) : プログラム・メモリの内容を表示します。
- 比較 (V) : partition で指定されたプログラム・メモリの内容と addr を先頭アドレスとする領域の内容を比較します。
比較したプログラム・メモリの内容が異なった場合、アドレスおよび内容の表示を行いません。
- 検索 (G) : data-string で指定したデータ列を partition で指定したアドレス内のデータと比較し、同一のデータ列が存在した場合は、アドレスの先頭番地を表示します (データ列は最大 10 個までです)。
- 転送 (M) : partition で指定した範囲の内容を addr を先頭アドレスとする領域に転送します。
- 初期化 (F) : partition で指定したアドレス範囲に data-string で指定したデータ列を書込みます (データ列は最大 10 個までです)。
- 交換 (X) : partition で指定したアドレス範囲の内容と addr を先頭アドレスとする領域の内容を交換します。ただし、交換するエリアは重複してはなりません。
- テスト (E) : partition で指定した範囲のプログラム・メモリのテストを行いません。
partition が省略された場合は、マッピングされているすべての範囲で実行し、実行後にメモリの内容を破壊してしまいます。
\$A が指定された場合はメモリの内容は破壊されません。

保守/廃止

7.9.1 変更

MEM C [addr])

プログラム・メモリの内容を変更します。

変更後は ') ' , E S C キー入力により復帰できます。

スペース・キーを入力すると, そのアドレスのメモリの内容を変更せず, 次のアドレスへ移行します。

例

1> MEM C addr) _____ (1)

1> MEM C) _____ (2)

(1) プログラム・メモリの内容を addr 番地から 8 ビット単位で変更します。

(2) プログラム・メモリの 0 H 番地の内容を 8 ビット単位で変更します。

実行例

1> MEM C 10T)

000A 00-11 00-22 00-33 00-44 00-) — 8ビット入力により次のアドレスに
1> MEM C 12Q) 移行します。
変更後は ') ' 入力により復帰できます。

000A 11-22 22-33 33-44 44-55 00-)
1> MEM C A)

000A 33-I
INPUT DATA ERROR — 8ビット以外の入力はエラーとなります。
0A番地の内容は破壊されません。

1> MEM C A)

000A 33-Y
INPUT DATA ERROR

1> MEM C B)

000B 44-44 55-55 66-66 00-00 00-00
0010 00-00 00-00 00-) — スペースを入力するとメモリ内容の
変更をせず次のアドレスへ移行します。

1> MEM C 123456789) — アドレスは下4桁を有効とします。
6789 00-

1> MEM C [MMAIN]START) — MMMAINフェーズのローカリゼーションSTARTのアドレス
0000 00- から変更を開始します。

1>

保守/廃止

7.9.2 表示

```
MEM D [ { addr
          } ] )
```

プログラム・メモリの内容を表示します。

例

```
1> MEM D 7F ) _____ (1)
1> MEM D 0, 1F ) _____ (2)
1> MEM D ) _____ (3)
```

- (1) プログラム・メモリの 7FH 番地の内容を表示します。
 (2) プログラム・メモリの 0～1FH の番地の内容を表示します。
 (3) プログラム・メモリの 0 番地の内容を表示します。

実行例

```
1> MEM D 7F ) _____ 7FH番地の内容を表示します。
007F 7F
1> MEM D 0,1F ) _____ 0～1FH番地の内容を表示します。
0000 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0010 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
1> MEM D X ) _____ X = 0～F
0000 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
1> MEM D 3F0,410 )
03F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0400 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0410 00
1> MEM D 3X ) _____ 3X = 30～3F
0030 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
1> MEM D [MMAIN]START,SS ) _____ MMINEジュールのローカル・シブSTARTのアドレス
からバリアク・シブSSまでを表示します。
0000 00 01 02 03 04 05 06
1>
```

保守/廃止

7.9.3 比較

```
MEM V partition addr )
```

partition で指定されたプログラム・メモリの内容と addr を先頭アドレスとする領域内容を比較して異なった内容があれば、そのアドレスおよび内容の表示をします。

例

```
1> MEM V 0, F 20 )
```

プログラム・メモリの 0H~FH 番地の内容と 20H 番地以降の内容を比較します。

実行例

```
1> MEM F XXXX 1,2,3,4,5,6,7,8,9,0 )
```

```
1> MEM V 0,F 20 )
```

```
0000 01 0020 03
0001 02 0021 04
0002 03 0022 05
0003 04 0023 06
0004 05 0024 07
0005 06 0025 08
0006 07 0026 09
0007 08 0027 00
0008 09 0028 01
0009 00 0029 02
000A 01 002A 03
000B 02 002B 04
000C 03 002C 05
000D 04 002D 06
000E 05 002E 07
000F 06 002F 08
```

```
1> MEM V 0,1F 50 )
```

比較した内容が同一であった場合は出力がありません。

```
1>
```

保守/廃止

1> MEM Y (MMAIN)START,(MMAIN)OWARI SS)

```
0000 01 0006 06
0001 02 0007 07
0002 03 0008 08
0003 04 0009 09
0004 05 000A 00
0005 06 000B 01
0006 07 000C 02
0007 08 000D 03
0008 09 000E 04
0009 00 000F 05
000A 01 0010 06
000B 02 0011 07
000C 03 0012 08
000D 04 0013 09
000E 05 0014 00
000F 06 0015 01
0010 07 0016 02
0011 08 0017 03
0012 09 0018 04
0013 00 0019 05
0014 01 001A 06
0015 02 001B 07
0016 03 001C 08
0017 04 001D 09
0018 05 001E 00
0019 06 001F 01
001A 07 0020 02
001B 08 0021 03
001C 09 0022 04
001D 00 0023 05
001E 01 0024 06
001F 02 0025 07
```

1>

保守/廃止

7.9.4 検 索

MEM G partition data-string)

data-string で指定したデータ列を partition で指定したアドレス内のデータと比較し、同一のデータが存在した場合はアドレスの先頭番地を表示します（データ列は最大10個までです）。

例

1> MEM G 0,1F 55,AA,33)

プログラム・メモリの 0H～1FH 番地内で連続したデータ列 50,AA,33 を検索します。

実行例

1> MEM F 0,100 55,AA,33,AA) 0H～100H 番地までのプログラム・メモリに
55 AA 33 AA のデータ列を書込みます。

1> MEM D 0,1F)

```
0000 55 AA 33 AA 55 AA 33 AA 55 AA 33 AA 55 AA 33 AA
0010 55 AA 33 AA 55 AA 33 AA 55 AA 33 AA 55 AA 33 AA
```

1> MEM G 0,1F 55,AA,33)

```
0000 ----- メモリの内容が 55 AA 33 の順になっている
0004                アドレスの先頭番地を 0H～1FH の範囲で出力
0008                します。
000C
0010
0014
0018
001C
```

1> MEM G [MMAIN]START,[MMAIN]OWARI 55,AA,33)

```
0000
0004
0008
000C
0010
0014
0018
001C
```

1> MEM F 0,100 66,22,3F,77,55,22,6F,77)

1> MEM G 0,1F XX,22,XF,77)

```
0000
0004
0008
000C
0010
0014
0018
001C
```

1> MEM D 0,1F)

```
0000 66 22 3F 77 55 22 6F 77 66 22 3F 77 55 22 6F 77
0010 66 22 3F 77 55 22 6F 77 66 22 3F 77 55 22 6F 77
```

1>

保守/廃止

7.9.5 転送

MEM M partition addr)

partition で指定された範囲の内容を addr を先頭アドレスとする領域に転送します。

例

1> MEM M 0, 1F 100)

プログラム・メモリの 0H~1FH 番地の内容を 100H 番地から 11FH 番地に転送します。

実行例

1> MEM D 0,1F)

```
0000 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0010 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
```

1> MEM D 100,11F)

```
0100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

1> MEM M 0,1F 100)

0H~1FH 番地の内容を 100H 番地から 11FH 番地に転送します。

1> MEM D 0,1F)

転送後の 0H~1FH 番地の内容は変化しません。

```
0000 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0010 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
```

1> MEM D 100,11F)

```
0100 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0110 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
```

1> MEM F 20,FF 0)

1> MEM D 20,50)

```
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00
```

1> MEM M [MMAIN]START,SS [MS0]CLRFE)

1> MEM D 20,30)

```
0020 00 00 00 00 00 00 00 01 02 03 04 05 00 00 00 00
0030 00
```

1>

保守/廃止

7.9.6 イニシャライズ

MEM F partition data-string)

partition で指定したアドレス範囲 data-string で指定したデータ列を書込みます (データ列は最大 10 個までです)。

例

1> MEM F 0, FF 0)

プログラム・メモリの 0H~FFH 番地に 0 を書込みます。

実行例

1> MEM F 0,FF 0)

1> MEM F [MMAIN]START,[MMAIN]OWARI 55)

1> MEM D XX)

```
0000 55 55 55 55 55 55 55 55 55 55 55 55 55 55
0010 55 55 55 55 55 55 55 55 55 55 55 55 55 55
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

1> MEM F XX 1,2,3,4,5,6,7,8,9,0)

1> MEM D XX)

```
0000 01 02 03 04 05 06 07 08 09 00 01 02 03 04 05 06
0010 07 08 09 00 01 02 03 04 05 06 07 08 09 00 01 02
0020 03 04 05 06 07 08 09 00 01 02 03 04 05 06 07 08
0030 09 00 01 02 03 04 05 06 07 08 09 00 01 02 03 04
0040 05 06 07 08 09 00 01 02 03 04 05 06 07 08 09 00
0050 01 02 03 04 05 06 07 08 09 00 01 02 03 04 05 06
0060 07 08 09 00 01 02 03 04 05 06 07 08 09 00 01 02
0070 03 04 05 06 07 08 09 00 01 02 03 04 05 06 07 08
0080 09 00 01 02 03 04 05 06 07 08 09 00 01 02 03 04
0090 05 06 07 08 09 00 01 02 03 04 05 06 07 08 09 00
00A0 01 02 03 04 05 06 07 08 09 00 01 02 03 04 05 06
00B0 07 08 09 00 01 02 03 04 05 06 07 08 09 00 01 02
00C0 03 04 05 06 07 08 09 00 01 02 03 04 05 06 07 08
00D0 09 00 01 02 03 04 05 06 07 08 09 00 01 02 03 04
00E0 05 06 07 08 09 00 01 02 03 04 05 06 07 08 09 00
00F0 01 02 03 04 05 06 07 08 09 00 01 02 03 04 05 06
```

1>

保守/廃止

7.9.7 交換

MEM X partition addr)

partition で指定したアドレス範囲の内容と addr を先頭アドレスとする領域の内容を交換します。

ただし、交換するエリアは重複してはなりません。

例

1> MEM X 0, 5F 200)

プログラム・メモリの 0H~5FH 番地の内容と 200H 番地から 25FH 番地の内容を交換します。

実行例

1> MEM D 0,FF)

```

0000 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0010 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
0020 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
0030 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
0040 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
0050 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
0060 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
0070 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
0080 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
0090 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
00A0 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
00B0 B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
00C0 C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
00D0 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
00E0 E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
00F0 F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
> MEM D 200,2FF )

```

```

0200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```



1> MEM X 0,5F 200)

1> MEM D XX)

```

0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
0070 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
0080 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
0090 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
00A0 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
00B0 B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
00C0 C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
00D0 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
00E0 E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
00F0 F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
    
```

1> MEM D 2XX)

```

0200 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0210 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
0220 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
0230 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
0240 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
0250 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
0260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    
```

0 H ~ 5 F H 番地までのメモリの内容が 2 0 0 H から 2 5 F H 番地に転送されています。

1> MEM X [(MMAIN)START, (MMAIN)OWARI 200)

1> MEM D XX)

```

0000 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0010 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
0070 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
0080 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
0090 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
00A0 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
00B0 B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
00C0 C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
00D0 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
00E0 E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
00F0 F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
    
```

1>

保守/廃止

7.9.8 テスト

```
MEM E [ partition ] [ $A ] )
```

partition で指定された範囲のプログラム・メモリのテストを行ないます。

省略された場合はすべてのプログラム・メモリのテストを行ないます。

テスト後のメモリの内容は \$A を指定した場合には破壊されません (指定のない場合には破壊されてしまいます)。

例

```
1 > MEM E 0, FFF ) _____ (1)
```

```
1 > MEM E $A ) _____ (2)
```

```
1 > MEM E 0, FFF $A ) _____ (3)
```

```
1 > MEM E ) _____ (4)
```

(1) プログラム・メモリの 0H ~ FFFH 番地のテストを行ないます。

(2) すべてのプログラム・メモリの簡易テストを行ないます。

(3) プログラム・メモリの 0H ~ FFFH 番地の簡易テストを行ないます。

(4) すべてのプログラム・メモリのテストを行ないます。

保守/廃止

7.10 アセンブル・コマンド

ASM[addr])

addr で指定されたアドレスのメモリの内容をニモニックとオペランドで変更します。

addr が省略された場合は 0 H 番地以降が指定されます。

終了は END 疑似命令で行ないます。

例

```
1> ASM 1000 )
```

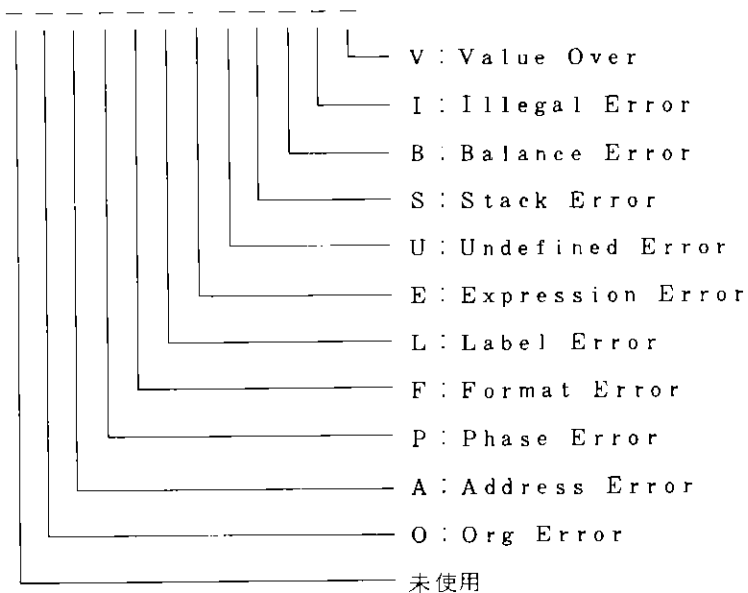
1000 H 番地のメモリの内容をニモニックとオペランドで変更します。

登録方法は以後の手順で行ないます。

- (a) addr で指定したアドレスのメモリ内容を逆アセンブルし、表示します。
- (b) '=' の後にラベル、ニモニックおよびオペランドを入力してください。
- (c) 入力した命令がアセンブルされ、オブジェクト・コードを表示します。
- (d) 表示されたオブジェクト・コードは ')' キーを入力すると、メモリに書込まれます。
ただし、スペース・キーを入力すると書込みを行ないません。

エラー・フラグの説明

入力した内容がエラーの時は '-' またはエラー・コードが出力されます。



Format Error の時は下のようになります。

```
-----F-----
```

保守/廃止

なお、アセンブル時の数値入力は、数値入力の後に基数（H, T, Q, Y）がない限り S U F コマンドで指定された基数に従います。

数値入力は式処理機能があり、M A T コマンドと同じ演算子を使用できます。

以下にエラー・コードを説明します。

Value Over	オペランド部の数値入力オーバーフローです。
Illegal Error	使用できない文字が記述されています。
Balance Error	括弧または引用符の記述がまちがっています。
Stack Error	式の記述が複雑すぎます。
Undefined Error	アセンブル未定義入力があります。
Expression Error	演算子の記述がまちがっています。
Label Error	解析不可能なアセンブル入力になっています。
Format Error	入力形式がまちがっています。
Phase Error	ラベルを二重定義しているか、またはラベルが予約語です。
Address Error	アドレス入力が F F F F H 番地を越えています。
Org Error	オルグで指定されたアドレス値が現在のアドレス値より低い場合に 表示されます。

使用可能な疑似命令は、次の五つです。

ORG

DB

DW

DS

END

保守/廃止

実行例

1> LOD B:MMAIN)

Object
Load Complete

Symbol Table
Load Complete

1> ASM)

ADRS	OBJECT	LABEL	MNEMONIC	
0000	040080	START:	LXI	SP,08000H
		=	<u>LXI SP,7000</u>)	
0000	040070)			
0003	240001	=	LXI	D,00100H
		=		
0006	402A00	=	CALL	0002AH
		=	<u>JMP OWARI</u>)	
0006	00000000	----		
0006	402A00	F--U----	CALL	0002AH
		=	<u>JMP [MMAIN]OWARI</u>)	
0006	541F00)			
0009	240002	=	LXI	D,00200H
		=	<u>END</u>)	

スペースを入力すると、アドレスの移動がなく、再度同じアドレスのアセンブルが可能です。

1> ASM 1000)

ADRS	OBJECT	LABEL	MNEMONIC	
1000	00		NOP	
		=	<u>LSTOUT: CALL 10</u>)	
1000	401000)			
1003	00		NOP	
		=	<u>END</u>)	

アセンブル・コマンドの終了時に入力してください。

1>

保守/廃止

7.11 逆アセンブル・コマンド

```
DAS [ { addr  
      { partition } ] )
```

指定アドレスまたは指定したアドレス範囲のメモリの内容をニモニックで表示します。

アドレスまたはアドレス範囲が省略された場合は 0 H 番地を表示します。

例

```
1> DAS 0, 1F ) _____ (1)
```

```
1> DAS ) _____ (2)
```

(1) 0 H ~ 1 F H 番地のメモリの内容をニモニックで表示します。

(2) 0 H 番地のメモリの内容をニモニックで表示します。

実行例

```
1> LOD B:MMAIN S ) _____ シンボル・ファイルをロードします。
```

```
1> DAS [MMAIN]START,[MMAIN]OWARI )
```

ADRS	OBJECT	LABEL	MNEMONIC
0000	040080	START:	LXI SP,08000H
0003	240001		LXI D,00100H
0006	402A00		CALL 0002AH
0009	240002		LXI D,00100H
000C	402200		CALL CLRFF
000F	6BFF		MVI C,000FFH
0011	340001		LXI H,00100H
0014	240003		LXI D,00300H
0017	10	BLOCK	D+
0018	340002		LXI H,00200H
001B	240004		LXI D,00400H
001E	10	BLOCK	D+
001F	541F00	OWARI:	JMP OWARI

```
1>
```

保守/廃止

7.12 メモリ転送コマンド

```
MOV  { I }
      { U } partition addr )
```

I : ユーザ・システムからエミュレーション・メモリへの転送
 U : エミュレーション・メモリからユーザ・システムへの転送
 システム間のメモリ転送を行ないます。

例

```
1> MOV I partition addr ) _____ (1)
1> MOV U partition addr ) _____ (2)
```

- (1) ユーザ・システムからエミュレーション・メモリへのプログラム転送です。
 partition で指定したユーザ・システム側のメモリの内容を addr を先頭アドレスとするエミュレーション・メモリへ転送します。
 マッピングは被転送領域であるエミュレーション・メモリに指定してください。
- (2) エミュレーション・メモリ・システムへのプログラム転送
 partition で指定したエミュレーション・メモリの内容を addr を先頭アドレスとするユーザ・システム側のメモリへ転送します。
 マッピングは被転送領域であるユーザ・システムに指定してください。

保守/廃止

7.12.1 ユーザ・システムからエミュレーション・メモリへのメモリ転送

```
MOV I partition addr )
```

partition で指定したユーザ・システム側のメモリ内容をエミュレーション・メモリの addr 番地以降に転送します。

例

```
I> MOV I 0, FF 0 )
```

ユーザ・システム側の 0H~FFH 番地の内容をエミュレーション・メモリの 0H 番地から FFH 番地へ転送します。

実行例

```
I> MAP W XXXX )
```

```
I> MAP U XX )
```

————— ユーザ領域をマッピングします。

```
I> MAP )
```

```
$ IE-INTROM MAPPING $
$ IE-INTRAM MAPPING $
0100-FFFF
$ USER $
0000-00FF
$ NON MAPPING $
```

```
I> MEM D XX )
```

```
0000 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0010 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
0020 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
0030 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
0040 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
0050 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
0060 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
0070 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
0080 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
0090 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
00A0 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
00B0 B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
00C0 C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
00D0 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
00E0 E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
00F0 F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
```

保守/廃止

1> MAP W XX)

1> MEM D XX)

```

0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
    
```

1> MOV I 0,FF 0)

1> MEM D XX)

```

0000 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0010 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
0020 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
0030 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
0040 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
0050 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
0060 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
0070 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
0080 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
0090 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
00A0 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
00B0 B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
00C0 C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
00D0 D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
00E0 E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
00F0 F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
    
```

1>

保守/廃止

7.12.2 エミュレーション・メモリからユーザ・システムへのメモリ転送

```
MOV U partition addr )
```

partition で示すエミュレーション・メモリの内容をユーザ・システム側の addr を先頭アドレスとするメモリに転送します。

例

```
1> MOV U XX 0 )
```

エミュレーション・メモリの 0H ~ FFH 番地のメモリの内容をユーザ側の 0H 番地から FFH 番地へ転送します。

実行例

```
1> MAP W XXXX )
1> MEM F XXX 0,11,22,33,44,55 )
1> MAP U XXX )
1> MEM F XX AA,BB,CC,DD,EE,FF )
1> MEM D X )
0000 AA BB CC DD EE FF AA BB CC DD EE FF AA BB CC DD
1> MOV U XX 0 )
1> MEM D XX )
0000 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33
0010 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11
0020 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
0030 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33
0040 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11
0050 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
0060 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33
0070 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11
0080 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
0090 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33
00A0 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11
00B0 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00C0 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33
00D0 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11
00E0 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00F0 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33
1> MAP )
$ IE-INTR0M MAPPING $
$ IE-INTRAM MAPPING $
1000-FFFF
$ USER $
0000-0FFF
$ NON MAPPING $
1>
```

保守/廃止

7.13 ロード・コマンド

```
LOD filename [ { C } ] [ { S } ] )
```

filename で指定した HEX 形式のオブジェクト・ファイル、シンボル・ファイルの内容をロードします。

drive を省略した場合カレント・ディスクが指定されます。サブ・コマンドを省略した場合 HEX オブジェクト・ファイルおよびシンボル・ファイルの両方がロードされます。

filename ロードすべきファイル

サブ・コマンド

C HEX オブジェクト・ファイルのみのロード

S シンボル・ファイルのみのロード

filename について

filename はファイル・タイプ (extension-name) の省略が可能です。

ファイル・タイプを省略した場合、HEX オブジェクト・ファイルの場合は 'HEX'、シンボル・ファイルの場合は 'SYM' としてロードします。

例

```
1> LOD B:TEST ) ----- (1)
1> LOD AAA. HEX ) ----- (2)
1> LOD B:TEST S ) ----- (3)
1> LOD AAA. SYM S ) ----- (4)
1> LOD BBB C ) ----- (5)
```

- (1) BドライブのHEXオブジェクト・ファイル、'TEST. HEX'、シンボル・ファイル 'TEST. SYM' の順でロードします。
- (2) カレント・ディスクのHEXオブジェクト・ファイル 'AAA. HEX' とシンボル・ファイル 'AAA. SYM' をロードします。
- (3) Bドライブのシンボル・ファイル 'TEST. SYM' をロードします。
- (4) カレント・ディスクのシンボル・ファイル 'AAA. SYM' をロードします。
- (5) カレント・ディスクのHEXオブジェクト・ファイル 'BBB. HEX' をロードします。

注 HEXオブジェクト・ファイルのロード中にエラーが生じた場合は、その時点でエラー・メッセージを出力し、終了します。

保守/廃止

実行例

```

I> LOD B:TEST )
Object -----
Load Complete      HEXオブジェクト・ファイルとしてB:TEST. HEX
                   をロードします。
Symbol Table -----
Load Complete      シンボル・ファイルとしてB:TEST. SYMをロードし
I> LOD AAA.HEX )
Object -----
Load Complete      HEXオブジェクト・ファイルとしてカレント・ディスクの
                   AAA. HEXをロードします。
Symbol Table -----
Load Complete      シンボル・ファイルとしてのカレント・ディスクのAAA.
I> LOD BBB.C )
Object -----
Load Complete      HEXオブジェクト・ファイルとしてのカレント・ディスク
                   のBBB. HEXをロードします。
I> LOD B:TEST.S )
Symbol Table -----
Load Complete      シンボル・ファイルとしてB:TEST. SYMをロードし
I> LOD B:TEST.HEX.C )
Object -----
Load Complete      HEXオブジェクト・ファイルとしてB:TEST. HEX
                   をロードします。
I> LOD AAA.SYM.S )
Symbol Table -----
Load Complete      シンボル・ファイルとしてカレント・ディスクのAAA.
I> LOD B:TEST1 )
Object -----
Load Complete      HEXオブジェクト・ファイルとしてB:TEST1. HEX
                   をロードとします。
Symbol Table -----
File Not Found     B:TEST1. SYMが存在しなかったことを示します。
I> LOD B:TEST2.HEX )
Object -----
File Not Found     B:TEST2. HEXが存在しなかったことを示します。
I> LOD B:TEST3 )
Object -----
File Not Found     B:TEST3. HEXが存在しなかったことを示します。
I>

```

保守/廃止

7.14 セーブ・コマンド

```
SAV filename [ C ] partition )
```

```
SAV filename S )
```

指定したファイルに `partition` で指定した範囲のオブジェクト・コードをHEX形式でセーブします。

またシンボルも指定ファイルへセーブすることができます。

`filename` セーブすべきファイル名

`partition` セーブすべきプログラム・メモリの範囲
(IE-78C11-Mの場合、X表現は使用できません。)

サブ・コマンド

C プログラム・メモリのみセーブ

S シンボルのみセーブ

`filename`

`filename`はファイル・タイプ(`extension-name`)の省略が可能です。

ファイル・タイプを省略した場合は`filename`のファイル・タイプはプログラム・メモリの場合‘HEX’となりシンボル・ファイルの場合‘SYM’となります。

注 シンボル・ファイルまたはヘキサ・ファイルがすでに存在する場合には現在存在するファイルを消去するかどうか聞いてきます。消去し、セーブを行なう場合、‘Y’を入力してください。セーブ中にディスク容量がいっぱいになった場合、セーブできませんのでディスクの交換を行ない、再度セーブを行なってください。

例

```
1> SAV B:TEST 0, FF ) _____ (1)
```

```
1> SAV TEST.HEX C 0, FF ) _____ (2)
```

```
1> SAV B:TEST S ) _____ (3)
```

(1) 0H~FFH番地までの範囲にあるプログラム・メモリの内容をBドライブのTEST.HEXに、またシンボルをBドライブのTEST.SYMにセーブします。

(2) 0H~FFH番地までの範囲にあるプログラム・メモリの内容をカレント・ディスクのTEST.HEXにセーブします。

(3) シンボルをドライブBのTEST.SYMにセーブします。

保守/廃止

実行例

I> SAV B:TEST O,FF)

Object _____ メモリの内容をB:TEST. HEXにセーブします。

Save Complete

Symbol Table _____ シンボルをB:TEST. SYMにセーブします。

Save Complete

I> SAV B:SSS C O,IF)

Object

File Already Exists. Delete ? (Y/N) : Y _____ 指定したファイルがすでに存在する場合にそれを消去し、セーブを行なうときは 'Y' を入力します。

Save Complete

I> SAV B:TEST.HEX C O,FF)

Object _____ メモリの内容をB:TEST. HEXにセーブします。

Save Complete

I> SAV B:TEST S)

Symbol Table

File Already Exists. Delete ? (Y/N) : Y

Save Complete

I> SAV B:TTT.SYM S)

Symbol Table

Save Complete

I> SAV B:SSS C O,F)

Object

File Already Exists. Delete ? (Y/N) : N _____ 消去しない場合、Nを入力します。セーブは行なわれません。

I>

I> SAV B:SSO C O,FFF)

Object

Disk Full

Disk Change ? (Y/N) : N _____ N入力で次のコマンド入力状態になります。

I> SAV B:SS1 C O,FFF)

Object

Disk Full

Disk Change ? (Y/N) : Y

Disk Change OK ? (Y/N) : N _____ Y入力されるまでメッセージは出力されません。

Disk Change OK ? (Y/N) : Y

Disk Change OK ? (Y/N) : Y

I> SAV B:SS1 C O,FFF) _____ ディスクを交換した後、再度セーブを行ないます。

Object

Save Complete

I>



7.15 ベリファイ・コマンド

```
VRY filename )
```

filename で指定したHEXファイルの内容とIEのメモリの内容とを比較します。

比較した内容で異なったメモリがあった場合は、そのアドレス、メモリの内容、ファイルの内容を表示します。

ファイル・タイプを省略した場合、'HEX'となります。

例

```
I> VRY TEST ) _____ (1)
```

```
I> VRY B:TEST1.HEX ) _____ (2)
```

(1) カレント・ディスクのTEST.HEXの内容とIEのメモリの内容と比較します。

(2) BドライブのTEST1.HEXの内容とIEのメモリの内容と比較します。

実行例

```
I> VRY TEST )
```

ADDR=0000,	DATA(RAM)=01,	DATA(FILE)=05
ADDR=0003,	DATA(RAM)=00,	DATA(FILE)=07
ADDR=001A,	DATA(RAM)=1F,	DATA(FILE)=FA

			TEST.HEXの内容
			IEのメモリの内容
			アドレス

```
I> VRY B:TEST1.HEX ) _____ 同一の場合は出力しません。
```

```
I>
```


保守/廃止

7.16 レジスタ・コマンド

$$\text{REG} \left[\left\{ \begin{array}{c} \text{C} \\ \text{D} \end{array} \right\} [\text{register-name}] \right]$$

C：変更

D：表示

μPD78C11, 78C10, 78C14 の有するレジスタの値を表示したり, 変更したりするコマンドです。
レジスタには次に示すものがあります。

PSW	フラグ・レジスタ	PC	プログラム・カウンタ
V	レジスタ V	V'	レジスタ V' (裏)
A	アキュムレータ	A'	アキュムレータ (裏)
B	レジスタ B	B'	レジスタ B' (裏)
C	レジスタ C	C'	レジスタ C' (裏)
D	レジスタ D	D'	レジスタ D' (裏)
E	レジスタ E	E'	レジスタ E' (裏)
H	レジスタ H	H'	レジスタ H' (裏)
L	レジスタ L	L'	レジスタ L' (裏)
EA	拡張アキュムレータ	EA'	拡張アキュムレータ (裏)
SP	スタック・ポインタ		

保守/廃止

7.16.1 変 更

 REG C [register-name])

指定したレジスタの内容を変更します。

レジスタ名を省略した場合、レジスタの内容を順に変更します。

例

1 > REG C EA') _____ (1)

1 > REG C) _____ (2)

(1) レジスタ EA' の内容を変更します。

この時変更データを入力せずに ') ' を入力した場合は内容変更をせずに終了します。

(2) すべてのレジスタの内容を順次変更します。

' スペース ' の入力で内容を変更せずに次のレジスタ変更に移れます。

') ' を入力した場合は現在のレジスタの変更はせずに終了します。

レジスタの変更順序は下のとおりです。

PSW → V → A → B → C → D → E → H → L → EA → SP → PC → V' → A' → B' → C' → D' → E'
 → H' → L' → EA'

EA' を終了した時点で変更を終了します。

保守 / 廃止

実行例

1> RUN N.O)

USER-SYSTEM VDD-ON

JUST MOMENT

EMULATION START AT 0000 - - - - - E S C 入力

ESC BREAK

TERMINATED

```
PSW V A B C D E H L EA SP PC
00 00 00 18 18 26 25 00 00 0000 0700 C84C
   V' A' B' C' D' E' H' L' EA'
   00 FF FF FF 01 4C 00 00 014C
```

1> REG C A)

A 00=11)

1> REG C)

PSW 00=11

V 00=22

A 11=33

B 18=44

C 18=55

D 26=66

E 25=77

H 00=00

L 00=88

EA 0000=99

SP 0700=1000

PC C84C=50

V' 00=AA

A' FF=BB

B' FF=CC

C' FF=DD

D' 01=EE

E' 4C=EE

H' 00=12

L' 00=34

EA' 014C=56

1> REG D)

```
PSW V A B C D E H L EA SP PC
11 22 33 44 55 66 77 00 88 0099 1000 0050
```

V' A' B' C' D' E' H' L' EA'

AA BB CC DD EE FF 12 34 0056

1>

----- 変更しない場合、スペースを入力します。

保守/廃止

7.16.2 表示

 REG[D[register-name]])

レジスタの内容を表示します。

レジスタを省略した場合、すべてのレジスタの内容を表示します。

例

```

1> REG D A ) _____ (1)
1> REG D ) _____ (2)
1> REG ) _____ (3)
  
```

- (1) レジスタ A の内容を表示します。
- (2) すべてのレジスタの内容を表示します。
- (3) すべてのレジスタの内容を表示します。

実行例

```

1> REG )
PSW V A B C D E H L EA SP PC
11 22 33 44 55 66 77 00 88 0099 1000 0050
V' A' B' C' D' E' H' L' EA'
AA BB CC DD EE FF 12 34 0056
1> REG D A )
A 33
1>
  
```

保守/廃止**7.17 スペシャル・レジスタ・コマンド**

```
SPR [ { C[ special-register-name 1 ]
        D[ special-register-name 2 ] } ] )
```

C : 変更

D : 表示

μPD78C11, 78C10, 78C14 の有する特殊レジスタの表示および変更を行います。
特殊レジスタには以下のものがあります。

special-register-name 1

PA	PORT	Aレジスタ
PB	PORT	Bレジスタ
PC	PORT	Cレジスタ
PD	PORT	Dレジスタ
PF	PORT	Fレジスタ
MKH	MASK	Highレジスタ
MKL	MASK	Lowレジスタ
TXB	Tx	BUFFERレジスタ
TM0	TIMER	REG0
TM1	TIMER	REG1
ETM0	TIMER/EVENT	COUNTER REG0
ETM1	TIMER/EVENT	COUNTER REG1

special-register-name 2

PA	PORT	Aレジスタ
PB	PORT	Bレジスタ
PC	PORT	Cレジスタ
PD	PORT	Dレジスタ
PF	PORT	Fレジスタ
MKH	MASK	Highレジスタ
MKL	MASK	Lowレジスタ
RXB	Rx	BUFFERレジスタ
CR0	A/D変換	リザルト0
CR1	A/D変換	リザルト1
CR2	A/D変換	リザルト2
CR3	A/D変換	リザルト3
ECNT	TIMER/EVENT	COUNTER UPCOUNTER
ECPT	TIMER/EVENT	COUNTER CAPTUREレジスタ

保守/廃止

7.17.1 変 更

SPR C [special-register-name1])

指定した特殊レジスタの内容を変更します。

特殊レジスタ名を省略した場合、特殊レジスタの内容を順に変更します。

例

1> SPR C PF) _____ (1)

1> SPR C) _____ (2)

(1) PFレジスタの内容を変更します。

(2) すべての特殊レジスタの内容を順に変更します。

‘)’を入力した場合、特殊レジスタの内容を変更せずに終了します。

特殊レジスタの変更順序は次のとおりです。

PA→PB→PC→PD→PF→MKH→MKL→TXB→TMO→TM1→ETMO→ETM1

ETM1を変更した時点で終了します。

実行例

1> SPR C PF)

PF A0=05)

1> SPR C)

```

PA 56=00
PB 2E=11
PC CC=22
PD C3=33
PF 00=44
MKH 07=55
MKL FE=66
TXB --=77
TMO ---=88
TM1 ---=90
ETMO --=AA
ETM1 ---=BB
1> SPR )
    
```

```

PA PB PC PD PF MKH MKL RXB CRO CR1 CR2 CR3 ECNT ECPT
00 00 00 C3 00 55 66 FF 00 00 00 00 0000 FFFF
1> SPR C )
    
```

```

PA 00=00 _____
PB 00=00 _____
PC 00=00 _____
PD C3=C3 _____
PF 00=00 _____
MKH 55=55 _____
MKL 66=66 _____
TXB ---- _____
TMO ---- _____
TM1 ---- _____
ETMO ---- _____
ETM1 ---- _____
1>
    
```

レジスタの内容を変更しない場合、スペースを入力します。

保守/廃止

7.17.2 表 示

```
SPR [ D [ special-register-name 2 ] ] )
```

特殊レジスタの内容を表示します。

特殊レジスタ名を省略した場合、すべての特殊レジスタの内容を表示します。

例

```
1> SPR D ECPT ) _____ (1)
```

```
1> SPR D ) _____ (2)
```

```
1> SPR ) _____ (3)
```

(1) ECPTレジスタの内容を表示します。

(2) すべての special-register-name 2 の内容を表示します。

(3) 動作は(2)と同様です。

実行例

```
1> SPR )
```

```
PA PB PC PD PF MKH MKL RXB CRO CRI CR2 CR3 ECNT ECPT
00 00 00 C3 00 55 66 FF 00 00 00 00 0000 FFFF
```

```
1> SPR D ECPT )
```

```
ECPT FFFF
```

```
1> SPR D TMO )
```

```
WRITE ONLY REGISTER _____ special-register-name1を指定すると
1> エラーになります。
```

保守/廃止

7.18 モード・レジスタ・コマンド

$$\text{MDR} \left[\left\{ \begin{array}{l} \text{C}[\text{mode-reg-name 1}] \\ \text{D}[\text{mode-reg-name 2}] \end{array} \right\} \right] \left. \right\} \left. \right\}$$

C：変更

D：表示

μPD78C11, 78C10, 78C14 の有するモード・レジスタの表示および変更を行ないます。
モード・レジスタには以下のものがあります。

mode-register-name 1

MA	MODE Aレジスタ
MB	MODE Bレジスタ
MC	MODE Cレジスタ
MCC	MODE CONTROL Cレジスタ
MM	MEMORY MAPPINGレジスタ
MF	MODE Fレジスタ
TMM	タイマ・モード・レジスタ
ETMM	タイマ/イベント・カウンタ・モード・レジスタ
EOM	タイマ/イベント・カウンタ・アウトプット・モード・レジスタ
SML	シリアル・モード・レジスタ (LOW)
SMH	シリアル・モード・レジスタ (HIGH)
ANM	A/Dチャンネル・モード・レジスタ
ZCM	ゼロクロス・モード・レジスタ

mode-register-name 2

TMM	タイマ・モード・レジスタ
EOM	タイマ/イベント・カウンタ・アウトプット・モード・レジスタ
SMH	シリアル・モード・レジスタ
ANM	A/Dチャンネル・モード・レジスタ

保守/廃止

7.18.1 変 更

```
MDR C [ mode-register-name ] )
```

指定したモード・レジスタの内容を変更します。

モード・レジスタ名を省略した場合、モード・レジスタの内容を順に変更します。

例

```
1> MDR C MA ) _____ (1)
```

```
1> MDR C ) _____ (2)
```

(1) MAレジスタの内容を変更します。

‘)’を入力した場合は内容を変更せずに終了します。

(2) すべてのモード・レジスタの内容を順次変更します。

スペース・キーを入力した場合、内容を変更せずに次のレジスタ変更に移ります。

‘)’を入力した場合、現在のモード・レジスタの内容を変更せずに終了します。

モード・レジスタの変更順序は次のとおりです。

MA→MB→MCC→MC→MM→MF→TMM→ETMM→EOM→SML→SMH→ANM→ZCM

ZCMを変更した時点で終了します。

実行例

```
1> MDR C MA
MA --=11
1> MDR C
MA --=00
MB --=11
MCC --=22
MC --=33
MM --=44
MF --=55
TMM FF=66
ETMM --=77
EOM 00=88
SML --=99
SMH 00=AA
ANM --=BB
ZCM --=CC
1>
```

保守/廃止

7.18.2 表 示

```
MDR[ D[ mode-register-name 2 ] ] )
```

モード・レジスタの内容を表示します。

モード・レジスタ名を省略した場合、すべてのモード・レジスタの内容を表示します。

例

```
1> MDR D TMM ) _____ (1)
```

```
1> MDR D ) _____ (2)
```

```
1> MDR ) _____ (3)
```

(1) TMMレジスタの内容を表示します。

(2) すべてのモード・レジスタ(リード可)の内容を表示します。

(3) 動作は(2)と同様です。

実行例

```
1> MDR )
```

```
TMM EOM SMH ANM  
66 88 AA BB
```

```
1> MDR D )
```

```
TMM EOM SMH ANM  
66 88 AA BB
```

```
1> MDR D TMM )
```

```
TMM 66
```

```
1> MDR D SML ) _____ mode-register-name 2 以外は表示できません。
```

```
WRITE ONLY REGISTER
```

```
1>
```

保守/廃止

7.19 ブレーク・コントロール・コマンド群

ブレーク・コマンドは以下に説明する3種類のブレーク・レジスタによって構成されています。

- フィジカル・ブレーク・レジスタ (Physical break register) ————— (1)
- ロジカル・ブレーク・レジスタ (Logical break register) ————— (2)
- ブレーク・モード・レジスタ (Break mode register) ————— (3)

それぞれのレジスタについて概要を説明します。

- (1) フィジカル・ブレーク・レジスタはブレーク条件を物理的にセット保存できるレジスタで以下の4種類があります。

BRA, BRD, BRE, BRT

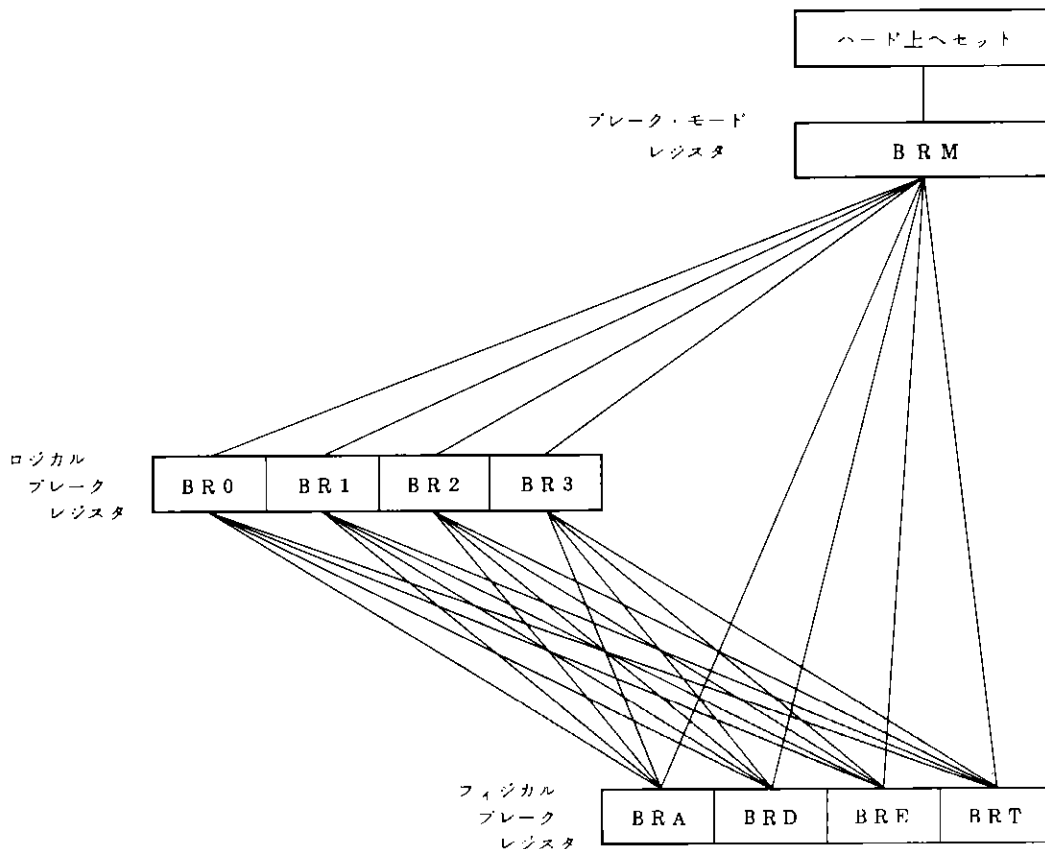
- (2) ロジカル・ブレーク・レジスタはフィジカル・レジスタの組合わせをセット保存できるレジスタで以下の4種類があります。

BR0, BR1, BR2, BR3

- (3) ブレーク・モード・レジスタは(1), (2)で示したフィジカル・ブレーク・レジスタおよびロジカル・ブレーク・レジスタをハード上にセットするレジスタで、ブレーク条件のハード上にセットした後ブレーク条件のデータ変更を行っても変更データはハード上にセットされません。再度ブレーク・モード・レジスタにセットしなおしてください。

注 初期状態は、未設定になっています。

図7-19 ブレーク・レジスタ・コマンドの関係図



保守/廃止

7.19.1 BRA コマンド

BRA [A=addr] [V=values] [C=cond] [L=loop])

addr	アドレス部	addr または partition をスペースで区切り最大 5 箇所まで指定できます。
values	データ部	8 ビットまでのデータで X 表現が可能です。 ただし、10 進の X 表現は禁止です。
cond	コンディション部	次のコンディション指定が行なえます。 省略された場合は条件なしとなります。 OP : オペコード・フェッチ (M1 リード) RD : メモリ・リード WR : メモリ・ライト MR : メモリ・リクエスト NC : 条件なし
loop	ループ指定部	アドレス部、データ部、コンディション部によって指定した条件に対してのループ (通過回数) の指定が行なえます。 ループの許容回数は 1 ~ 255 回です。

例

```

1> BRA A=10 ) _____ (1)
1> BRA ) _____ (2)
1> BRA A=0, FH 30H, 3FH 50H, 5FH ) _____ (3)
1> BRA V=55T ) _____ (4)
1> BRA C=WR L=3 ) _____ (5)
1> BRA A=20H V=0 L=5T ) _____ (6)
1> BRA A=1XXH V=1X0X1Y C=RD L=2Q ) _____ (7)

```

- (1) ブレーク・アドレスを 10 (サフィックスが 16 進なら 10H, 10 進なら AH, 8 進なら 8H, 2 進なら 2H) 番地に指定します。
- (2) BRA に指定されているブレーク条件を表示します。
- (3) アドレスの 0H~FH, 30H~3FH, 50H~5FH のいずれかをアクセスしたらブレークするように指定します。
- (4) データ 55T をアクセスしたらブレークするように指定します。
- (5) データを 3 回書込んだらブレークするように指定します。
- (6) アドレス 20H で 0H を 5 回アクセスしたらブレークするように指定します。
- (7) アドレス 100H~1FFH の範囲でデータ 11H 13H 19H 1BH のいずれかを 2 回

保守/廃止

リードしたらブレークするように指定します。

実行例

```
I> BRA _
--
I> BRA A=10
I> BRA _
A=10H
I> BRA A=1XXH V=1XOX1Y C=RD L=2Q
I> BRA _
A=1XXH V=1XOX1Y C=RD L=2Q
I> BRA A=SS
I> BRA _
A=0006H
I>
```

保守/廃止

7.19.2 BRD コマンド

BRD [values])

BRD コマンドは外部センス・クリップで8本のデータでブレイクさせるためのオプション・ブレイク・コマンドです。

例

1> BRD 5AH)————— (1)

1> BRD)————— (2)

(1) ブレイク・データを5AHに指定します。

(2) 現在指定されているデータを表示します。

指定されていない場合は‘—’を表示します。

またvaluesはマスク表現可能な数値データで1バイトです。

実行例

1> BRD)

--

1> BRD 5AH)

1>

保守/廃止

7.19.3 BRE コマンド

```
BRE [count] )
```

BRE コマンドはオペコード・フェッチの回数 (count) を指定します。

例

```
1> BRE 1000H ) _____ (1)
```

```
1> BRE ) _____ (2)
```

- (1) オペコード・フェッチの回数を1000Hに指定します。
なお、count 数は2バイトで3H~FFFFHの範囲です。
- (2) 現在指定されているオペコード・フェッチの回数を表示します。
指定されていない場合、' -- ' を表示します。

実行例

```
1> BRE )  
--  
1> BRE 1000 )  
1> BRE )  
1000H  
1>
```

保守/廃止

7.19.4 BRT コマンド

```
BRT [time] )
```

BRT コマンドはブレーク・タイマ・コマンドです。

例

```
1> BRT 1300H ) _____ (1)
```

```
1> BRT ) _____ (2)
```

(1) ブレーク・タイマを、1300H(ms)に指定します。

なお、timeは2バイト・データで(1H~FFFFHmsの範囲)を入力します。

(2) 現在指定されているブレーク・タイムを表示します。

指定されていない場合、'---'を表示します。

実行例

```
1> BRT )
```

```
--
```

```
1> BRT 1300 )
```

```
1> BRT )
```

```
1300H
```

```
1>
```


保守/廃止

7.19.5 BR0～BR3 コマンド

BR0 [[BRA], [BRD], [BRE], [BRT]])

ロジカル・ブレイク・コマンドにはBR0～BR3までありそれぞれ同等の機能をもっています。機能は、あらかじめブレイク時に登録したいフィジカル・レジスタをロジカル・ブレイク・レジスタに登録しておけばBRMコマンドでブレイク条件を登録する時にロジカル・ブレイク・レジスタを一つ登録すればよいことになります。

例

1> BR0 BRA, BRD, BRE, BRT) _____ (1)

1> BR0) _____ (2)

(1) ロジカル・ブレイク・コマンド (BR0) にフィジカル・ブレイク・コマンド (BRA, BRD, BRE, BRT) を登録します。

フィジカル・ブレイク・コマンドを複数入力するときは、その間を ' , ' で区切ります。

(BR1～BR3も同様)

ブレイクはいずれかの条件が満足したらブレイクします。

(2) 現在登録されているフィジカル・ブレイク・コマンドを表示します。

登録されていない場合、' -- ' を表示します。

実行例

1> BR0)

--

1> BR0 BRA, BRD BRE, BRT)

1> BR0)

BRA, BRD, BRE, BRT

1>

保守/廃止

7.19.6 BRM コマンド

```
BRM [[BRA], [BRD], [BRE], [BRT], [BR0], [BR1],
                                           [BR2], [BR3]] )
```

BRMはブレーク付エミュレーション(RUN B)を行なうときのブレーク条件を登録するコマンドです。

フィジカル・ブレーク・コマンドとロジカル・ブレーク・コマンドをセットしてもコマンドによって登録しなければ、'RUN B'のブレーク条件とは、なりません。

例

```
1> BRM BRA, BR0, BR1 ) _____ (1)
1> BRM BR0, BR1, BR2, BR3 ) _____ (2)
1> BRM ) _____ (3)
```

(1) ブレーク条件にBRA, BR0, BR1を登録します。

フィジカル・ブレーク・コマンド, ロジカル・ブレーク・コマンドの両方を任意の個数だけ登録できます。

(2) フィジカル・ブレーク・コマンド, ロジカル・ブレーク・コマンド, ブレーク・モード・コマンドはどちらを先に登録してもかまいません。

ブレークはどれか一つBRMに登録されたものを満たせばブレークします。

(3) 現在登録されているロジカル・ブレーク・コマンド, フィジカル・ブレーク・コマンドを表示します。

実行例

```
1> BRM )
--
1> BRM BRA,BRO BRI )
1> BRM )
BRA,BRO,BRI
1>
```

保守/廃止

7.20 トレース・コマンド群

トレース・コマンドには TRX, TRD, TRM, TRP, TRC, TRS の六つのコマンドがあります。

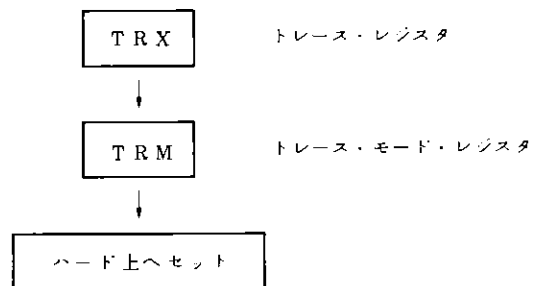
- TRX : トレース条件の設定
- TRD : トレース・データの表示
- TRM : トレース・モードの変更
- TRP : トレース・ポインタの移動
- TRC : 表示形態の変更
- TRS : ソース選択

トレース容量 1023 フレームです。

POWER ON 時には、トレース・サイクル (TRC) はマシン・サイクル (M)、トレース・モード (TRM) はノン・トレース (NON)、ソース選択 (TRS) はポート B (PB) に設定されます。

トレース・データがないときには TRD は何も表示しません。

図 7-20 トレース・コマンドの関係図



保守/廃止

7.20.1 TRX コマンド

TRX[A=addr] [C=cond] [V=values] [PA=values] [PB=values])

addr : アドレス addr または partition をスペースで区切り最大5箇所まで指定できます。

values : データ X表現が可能です。ただし10進のX表現は禁止です。

cond : 条件 下記の条件を選択します。省略の場合はNCとなります。

OP : オペコード・フェッチ (M1リードの場合)

RD : メモリ・リード

WR : メモリ・ライト

NC : 条件なし

条件の指定はアドレスとデータの組合わせに対する条件を指定します。

トレース条件は指定された条件 (AND条件) をすべて満たしたとき初めてトレースされます。

例

1> TRX A=0,1FFF 3000 V=00 PA=0) _____ (1)

1> TRX) _____ (2)

(1) トレース・アドレスは 'A=' に続いてスペースをターミネイトとして最大5個まで指定可能です。

トレース条件のデータは 'V=' に続いて指定できX表現が可能です。

ポート A, B の条件指定によるトレースも行なうことができます。

(2) TRX に指定されているトレース条件を表示します。

指定がない場合は 'ー' を表示します。

実行例

1> TRX)

--

1> TRX A=0H,1FFFH 3000H V=0H C=RD PA=0H)

1> TRX)

A=0H,1FFFH 3000H V=0H C=RD PA=0H

1> TRX A=SS)

1> TRX)

A=0006H

1>

保守/廃止

7.20.2 TRMコマンド

```
TRM [ { NON
      { ALL } ] )
      { TRX }
```

TRMコマンドはトレースする条件を設定するコマンドです。

トレース条件には以下のような種類があります。

NON : トレースしない

ALL : すべてトレースする

TRX : トレース条件を設定する

例

```
1> TRM NON ) _____ (1)
```

```
1> TRM ALL ) _____ (2)
```

```
1> TRM TRX ) _____ (3)
```

```
1> TRM ) _____ (4)
```

- (1) トレース条件をNON(トレースしない)に設定します。
- (2) トレース条件をALL(すべてトレースする)に設定します。
- (3) トレース条件をTRXに設定します。
- (4) 現在設定されているトレース条件を表示します。

実行例

```
1> TRM )
NON
1> TRM ALL )
1> TRM )
ALL
1> TRM TRX )
1> TRM )
TRX
1>
```

保守/廃止

7.20.3 TRDコマンド

```
TRD [ { [ + ] line-No.
        -line-No.
        ALL
      } ] )
```

TRDコマンドはリアルタイム実行後にその実行結果を表示するコマンドです。
トレースするときの条件はTRMコマンドにより指定します。

例

```
1> TRD 10 ) _____ (1)
1> TRD ALL ) _____ (2)
1> TRD ) _____ (3)
```

- (1) 現在のポインタから10ラインだけトレースを実行し結果を表示します。
ポインタは10移動します(+は省略可能です)。
- (2) トレース・データをすべて表示します。
- (3) 現在のポインタから全トレース・データを表示します。

TRMがNONに指定されている場合、データはトレースされず、またそれ以前のトレース・データも破壊されます。

TRPコマンドでポインタの移動をしない限り次にTRDコマンドを実行するときは、そのポインタの位置からトレース・ダンプを行ないます。

TRDコマンドは、TRPコマンドと密接に関係していますのでTRPコマンドも参照してください。

エミュレーション実行後のトレース・ポインタは、最も新しいところになっています。

保守/廃止

実行例

1> TRM ALL

1> RUN S 0.10) _____ 0H番地から10Hステップ実行させます。

USER-SYSTEM VDD-ON

JUST MOMENT
EMULATION START AT 0000

STEP BREAK
TERMINATED

```
PSW V A B C D E H L FA SP PC
04 08 01 02 03 04 05 06 07 0000 0000 000E
   V' A' B' C' D' E' H' L' EA'
   00 00 00 00 00 00 00 00 00 0000
```

ONE STEP EMULATION STANDBY)

1> TRP)

0010

1> TRD ALL) _____ トレース結果を表示します。
トレース後のポインタはCH番地です。

OLDEST	OBJECT	LABEL	MNEMONIC	PA	PB
ADRS	6901	START:	MVI	A,00001H	00 00
0000	6A02		MVI	B,00002H	00 00
0002	6B03		MVI	C,00003H	00 00
0004	6C04		MVI	D,00004H	00 00
0006	6D05		MVI	E,00005H	00 00
0008	6E06		MVI	H,00006H	00 00
000A	6F07		MVI	L,00007H	00 00
000C	6808		MVI	V,00008H	00 00
0010	EF	EXCHG1:	JR	START	00 00
0000	6901	START:	MVI	A,00001H	00 00
0002	6A02		MVI	B,00002H	00 00
0004	6B03		MVI	C,00003H	00 00
0006	6C04		MVI	D,00004H	00 00
0008	6D05		MVI	E,00005H	00 00
000A	6E06		MVI	H,00006H	00 00
000C	6F07		MVI	L,00007H	00 00

NEWEST
1> TRC M) _____

表示形態をマシン・サイクルにします。

保守/廃止

1> IRD-100

ADRS	DATA	CYC.	PA	PB
OLDEST				
0000	69	OP	00	00
0001	01	RD	00	00
0002	6A	OP	00	00
0003	02	RD	00	00
0004	6B	OP	00	00
0005	03	RD	00	00
0006	6C	OP	00	00
0007	04	RD	00	00
0008	6D	OP	00	00
0009	05	RD	00	00
000A	6E	OP	00	00
000B	06	RD	00	00
000C	6F	OP	00	00
000D	07	RD	00	00
000E	68	OP	00	00
000F	08	RD	00	00
0010	EF	OP	00	00
0000	69	OP	00	00
0001	01	RD	00	00
0002	6A	OP	00	00
0003	02	RD	00	00
0004	6B	OP	00	00
0005	03	RD	00	00
0006	6C	OP	00	00
0007	04	RD	00	00
0008	6D	OP	00	00
0009	05	RD	00	00
000A	6E	OP	00	00
000B	06	RD	00	00
000C	6F	OP	00	00
000D	07	RD	00	00

NEWEST

1> TRP_Q

ポインタを一番古い所へ移動します。

1> TRD_B

ADRS	DATA	CYC.	PA	PB
OLDEST				
0000	69	OP	00	00
0001	01	RD	00	00
0002	6A	OP	00	00
0003	02	RD	00	00
0004	6B	OP	00	00
0005	03	RD	00	00
0006	6C	OP	00	00
0007	04	RD	00	00

保守 / 廃止

1> TRD +100 2

ADRS	DATA	CYC.	PA	PB
OLDEST				
0008	6D	OP	00	00
0009	05	RD	00	00
000A	6E	OP	00	00
000B	06	RD	00	00
000C	6F	OP	00	00
000D	07	RD	00	00
000E	68	OP	00	00
000F	08	RD	00	00
0010	EF	OP	00	00
0000	69	OP	00	00
0001	01	RD	00	00
0002	6A	OP	00	00
0003	02	RD	00	00
0004	6B	OP	00	00
0005	03	RD	00	00
0006	6C	OP	00	00
0007	04	RD	00	00
0008	6D	OP	00	00
0009	05	RD	00	00
000A	6E	OP	00	00
000B	06	RD	00	00
000C	6F	OP	00	00
000D	07	RD	00	00

NEWEST

1> TRD -5

ADRS	DATA	CYC.	PA	PB
0009	05	RD	00	00
000A	6E	OP	00	00
000B	06	RD	00	00
000C	6F	OP	00	00
000D	07	RD	00	00

NEWEST

1> TRA -6

1> TRD

ADRS	DATA	CYC.	PA	PB
0008	6D	OP	00	00
0009	05	RD	00	00
000A	6E	OP	00	00
000B	06	RD	00	00
000C	6F	OP	00	00
000D	07	RD	00	00

NEWEST

1>

保守/廃止

7.20.4 TRP コマンド

$$\text{TRP} \left[\left\{ \begin{array}{l} 0 \\ N \\ [+]\text{line-No.} \\ -\text{line-No.} \end{array} \right\} \right])$$

TRP コマンドは、現在のトレース・ポインタを移動させるコマンドです。エミュレーション実行直後のトレース・ポインタは最も新しいところを示しています。

例

```
1> TRP 0 ) _____ (1)
1> TRP N ) _____ (2)
1> TRP ±line-No. ) _____ (3)
1> TRP ) _____ (4)
```

- (1) トレース・データの最も古いところへポインタを移動します。
- (2) トレース・データの最後にトレースしたところにポインタを移動します。
- (3) 現在のポインタから数値で指定した数だけ移動します。
(+ : N (NEWEST) 方向で省略可能, - : O (OLDEST) 方向)
- (4) エミュレーション後のトレース実行の総数を表示します。

保守/廃止

実行例

1> TRP 0) _____ トレース・ポインタを最も古いところに移動します。
 1> TRP N) _____ トレース・ポインタを最も新しいところに移動します。
 1> TRP 0) _____
 1> TRD +100) _____ 現在のポインタより100ライントレース・データを表示します。

ADRS	DATA	CYC.	PA	PB
0000	69	OP	00	00
0001	01	RD	00	00
0002	6A	OP	00	00
0003	02	RD	00	00
0004	6B	OP	00	00
0005	03	RD	00	00
0006	6C	OP	00	00
0007	04	RD	00	00
0008	6D	OP	00	00
0009	05	RD	00	00
000A	6E	OP	00	00
000B	06	RD	00	00
000C	6F	OP	00	00
000D	07	RD	00	00
000E	68	OP	00	00
000F	08	RD	00	00
0010	EF	OP	00	00
0000	69	OP	00	00
0001	01	RD	00	00
0002	6A	OP	00	00
0003	02	RD	00	00
0004	6B	OP	00	00
0005	03	RD	00	00
0006	6C	OP	00	00
0007	04	RD	00	00
0008	6D	OP	00	00
0009	05	RD	00	00
000A	6E	OP	00	00
000B	06	RD	00	00
000C	6F	OP	00	00
000D	07	RD	00	00

NEWEST _____

最も新しいトレース・データを表示し、それ以降データがないことを示します。

1> TRD -5) _____ 現ポインタの五つ前から、トレース・データを現ポインタまで表示します。

ADRS	DATA	CYC.	PA	PB
0009	05	RD	00	00
000A	6E	OP	00	00
000B	06	RD	00	00
000C	6F	OP	00	00
000D	07	RD	00	00

1> TRP -6) _____

1> TRD) _____ 現在のポインタより最終のトレース・データまでを表示します。

ADRS	DATA	CYC.	PA	PB
0008	6D	OP	00	00
0009	05	RD	00	00
000A	6E	OP	00	00
000B	06	RD	00	00
000C	6F	OP	00	00
000D	07	RD	00	00

1>

保守/廃止

7.20.5 TRCコマンド

$$\text{TRC} \left[\left\{ \begin{array}{c} \text{I} \\ \text{M} \end{array} \right\} \right]$$

トレース・ダンプ時の表示形態（マシン・サイクルまたはインストラクション）の選択または表示を行ないます。

I：インストラクション表示

M：マシン・サイクル表示

起動時には、マシン・サイクルに指定されます。

例

```
1> TRC I ) _____ (1)
1> TRC M ) _____ (2)
1> TRC ) _____ (3)
```

- (1) インストラクション表示に指定します。
- (2) マシン・サイクル表示に指定します。
- (3) 現在の状態（MまたはI）を表示します。

実行例

```
1> TRC I )
1> RUN S 0.5 )
USER-SYSTEM VDD-ON
JUST MOMENT
EMULATION START AT 0000
STEP BREAK
TERMINATED
PSW V A B C D E H L EA SP PC
00 08 01 02 03 04 05 06 07 0000 0000 000A
   V' A' B' C' D' E' H' L' EA'
   00 00 00 00 00 00 00 00 0000
```

保守/廃止

I> TRD ALL _____ インストラクション表示

ADRS	OBJECT	LABEL	MNEMONIC		PA	PB
0000	6901	START:	MVI	A,00001H	00	00
0002	6A02		MVI	B,00002H	00	00
0004	6B03		MVI	C,00003H	00	00
0006	6C04		MVI	D,00004H	00	00
0008	6D05		MVI	E,00005H	00	00

I> TRC M

I> RUN S 0.4

USER-SYSTEM VDD-ON
JUST MOMENT
EMULATION START AT 0000

STEP BREAK
TERMINATED

PSW	V	A	B	C	D	E	H	L	EA	SP	PC
00	08	01	02	03	04	05	06	07	0000	0000	0008
	V'	A'	B'	C'	D'	E'	H'	L'	EA'		
	00	00	00	00	00	00	00	00	0000		

ONE STEP EMULATION STANDBY

I> TRD ALL _____ マシン・サイクル表示

ADRS	DATA	CYC.	PA	PB
0000	69	OP	00	00
0001	01	RD	00	00
0002	6A	OP	00	00
0003	02	RD	00	00
0004	6B	OP	00	00
0005	03	RD	00	00
0006	6C	OP	00	00
0007	04	RD	00	00

I>

保守/廃止

7.20.6 TRSコマンド

```
TRS [ { PB } ] )
      { EX }
```

トレース (PB と EX) の選択または表示をします。

PB : ポート B

EX : 外部センス・データ

起動時には PB に設定されます。

例

```
1 > TRS EX ) _____ (1)
```

```
1 > TRS ) _____ (2)
```

- (1) 外部ポートをトレースします。
- (2) 現在選択されているソース名を表示します。

実行例

```
1 > TRS )
```

```
PB
1 > RUN S.0.5 )
```

```
USER-SYSTEM VDD-ON
JUST MOMENT
EMULATION START AT 0000
```

```
STEP BREAK
TERMINATED
```

```
PSW V A B C D E H L EA SP PC
00 08 01 02 03 04 05 06 07 0000 0000 000A
   V' A' B' C' D' E' H' L' EA'
   00 00 00 00 00 00 00 00 00 0000
```

```
ONE STEP EMULATION STANDBY )
```

```
1 > TRC 1 )
```

```
1 > TRD ALL )
```

ADRS	OBJECT	LABEL	MNEMONIC	PA PB	—	ポート B の
0000	6901	START:	MYI	A,00001H	00 00	トレース・
0002	6A02		MYI	B,00002H	00 00	データを表
0004	6B03		MYI	C,00003H	00 00	示します。
0006	6C04		MYI	D,00004H	00 00	
0008	6D05		MYI	E,00005H	00 00	

保守 / 廃止

I> TRS_EX _

I> RUN S 0.5 _

USER-SYSTEM VDD-ON

JUST MOMENT
EMULATION START AT 0000

STEP BREAK
TERMINATED

```
PSW Y A B C D E H L EA SP PC
00 08 01 02 03 04 05 06 07 0000 0000 000A
   Y' A' B' C' D' E' H' L' EA'
   00 00 00 00 00 00 00 00 0000
```

ONE STEP EMULATION STANDBY _

I> TRD_ALL _

ADRS	OBJECT	LABEL	MNEMONIC	PA EX	外部センス・クリッ プのトレース・デー タを表示します。
0000	6901	START:	MVI	A,00001H	00 FF
0002	6A02		MVI	B,00002H	00 FF
0004	6B03		MVI	C,00003H	00 FF
0006	6C04		MVI	D,00004H	00 FF
0008	6D05		MVI	E,00005H	00 FF

I>

保守/廃止

7.21 エミュレーション・コマンド

RUN $\left\{ \begin{matrix} N \\ B \end{matrix} \right\}$ [addr])

RUN S [addr][, step-No.])

RUN T [addr][, {

step-No.	$\left\{ \begin{matrix} = \\ < \\ > \\ <> \\ >< \\ <= \\ =< \\ >= \\ => \end{matrix} \right\}$	value
register-name		

)] [\$ $\left\{ \begin{matrix} E \\ D \end{matrix} \right\}$] [\$R])

N : 通常リアルタイム・エミュレーション

B : ブレーク条件付きエミュレーション

S : 指定ステップ数実行

T : 1ステップ・エミュレーション

D : 表示ディスエーブル

E : 表示イネーブル

R : レジスタ表示指定

保守/廃止

7.21.1 通常リアルタイム・エミュレーション

RUN N[addr])

指定したアドレスからエミュレーションを開始します。

addrを省略した場合のプログラム・カウンタがスタート・アドレスに指定されます。

ESC により実行を中止し、そのときのレジスタの内容を表示します。

IE-78C11-Cの場合、**CTRL** Cによりブレイクせずに“E-CPU RUN”を出力しプロンプトに戻り、RESコマンドと**ESC**キーのみ受けつけます。

例

1> RUN N) _____ (1)

1> RUN N 0) _____ (2)

(1) 現在のプログラム・カウンタで示されるアドレスからリアルタイムでエミュレーションを開始します。

(2) 0H番地からリアルタイムでエミュレーションを開始します。

注 **ESC** を入力して強制ブレイクしようとするとき、**ESC** を入力する前に何らかのキーを入力していた場合、エミュレーションCPUはブレイクしますが、レジスタを表示しない場合があります。

保守/廃止

実行例

1> RUN N Q 2

USER-SYSTEM VDD-ON

JUST MOMENT
EMULATION START AT 0000 ← [ESC入力]

ESC BREAK
TERMINATED

```
PSW V A B C D E H L EA SP PC
04 08 01 02 03 04 05 06 07 0000 0000 000A
   V' A' B' C' D' E' H' L' EA'
   00 00 00 00 00 00 00 00 0000
```

1> RUN N Q 2

USER-SYSTEM VDD-ON

JUST MOMENT
EMULATION START AT 0000

E-CPU RUN ← [CTRL]/C入力 (注)
1> ← [ESC入力]

ESC BREAK
TERMINATED

```
PSW V A B C D E H L EA SP PC
04 08 01 02 03 04 05 06 07 0000 0000 000E
   V' A' B' C' D' E' H' L' EA'
   00 00 00 00 00 00 00 00 0000
```

1>

注 [CTRL] C入力によりIE-78C11-Cではプロンプトに戻り、これにより以後に
力できるのは、RESコマンドまたは [ESC] キーのみとなります。
IE-78C11-Mの場合 'Abort ? (Y/N)' と出力し、'Y' と入力があった場
合はMP/M-86に戻ります。

保守/廃止

7.21.2 ブレーク条件付きエミュレーション

```
RUN B[ addr ] )
```

指定したアドレスからブレーク・モードの条件が満たされるまでリアルタイムでエミュレーションを行ないブレーク以後は1ステップ動作となります。

addrを省略した場合は現在のプログラム・カウンタがスタート・アドレスに指定されます。

ESC , **CTRL C** のシーケンスは通常リアルタイム・エミュレーションと同様です。

例

```
1> RUN B ) _____ (1)
```

```
1> RUN B 0 ) _____ (2)
```

(1) 現在のプログラム・カウンタで示されるアドレスからブレーク条件付きエミュレーションを開始します。

(2) 0H番地からブレーク条件付きエミュレーションを開始します。

1ステップ動作

スペース・キーを入力することにより、1ステップ実行します。

また、') ' を入力することにより、終了します。

保守/廃止

実行例

```

1> BRM BRA )
1> BRA A=10 )
1> RUN B 0 )

USER-SYSTEM VDD-ON

JUST MOMENT
EMULATION START AT 0000

NORMAL BREAK
TERMINATED

PSW Y A B C D E H L EA SP PC
00 00 18 18 18 18 19 00 11 0000 0700 0011
   Y' A' B' C' D' E' H' L' EA'
   00 FF FF FF 01 4C 00 00 014C

ONE STEP EMULATION STANDBY _____ スペース入力

ADRS OBJECT          LABEL          MNEMONIC
0011 32              INX              H

PSW Y A B C D E H L EA SP PC
00 00 18 18 18 18 19 00 12 0000 0700 0012
   Y' A' B' C' D' E' H' L' EA'
   00 FF FF FF 01 4C 00 00 014C _____ スペース入力

ADRS OBJECT          LABEL          MNEMONIC
0012 32              INX              H

PSW Y A B C D E H L EA SP PC
00 00 18 18 18 18 19 00 13 0000 0700 0013
   Y' A' B' C' D' E' H' L' EA'
   00 FF FF FF 01 4C 00 00 014C )
1>
    
```

保守/廃止

7.21.3 指定ステップ数実行

```
RUN S[ addr ][ , step-No. ] )
```

指定したアドレスから指定したステップ数をリアルタイムで実行します。

addrを省略した場合は、現在のプログラム・カウンタがスタート・アドレスに指定されます。

ステップ数を省略した場合は、1ステップ動作となります。

指定ステップ数実行後は1ステップ動作となります。

ESCのシーケンスは通常リアルタイム・エミュレーションと同様とします。

例

```
1> RUN S 0, 5 ) _____ (1)
1> RUN S , 50 ) _____ (2)
1> RUN S 1000 ) _____ (3)
1> RUN S ) _____ (4)
```

(1) 0番地から5ステップをリアルタイムで実行し以後1ステップ動作となります。

(2) 現在のプログラム・カウンタの値から50Hステップをリアルタイムで実行し以後1ステップ動作となります。

(3) 1000H番地より1ステップ動作となります。

(4) 現在のプログラム・カウンタの値から1ステップ動作となります。

実行例

```
I> RUN S 0,5 )
USER-SYSTEM VDD-ON
JUST MOMENT

STEP BREAK
TERMINATED

PSW V A B C D E H L EA SP PC
00 02 00 80 80 80 80 00 00 0000 0700 0005
   V' A' B' C' D' E' H' L' EA'
   00 FF FF FF 01 4C 00 00 014C

ONE STEP EMULATION STANDBY )
I>
```

保守/廃止

7.21.4 1ステップ・エミュレーション

$$\text{RUN T[addr][, } \left\{ \begin{array}{l} \text{step-No.} \\ \left. \begin{array}{l} (=) \\ (>) \\ (<) \\ (>=) \\ (<=) \\ (= >) \\ (= <) \\ (> <) \\ (< >) \end{array} \right\} \text{value} \right\}] [\$ \left\{ \begin{array}{l} D \\ E \end{array} \right\}] [\$ R])$$

指定したアドレスから指定した条件に従って1命令ずつアドレス、命令コード、ニモニック、レジスタの内容を表示しながら実行します。

これ以後は、1ステップ動作となります。

アドレスを省略した場合は現在のプログラム・カウンタがスタート・アドレスに指定されます。

指定条件を省略した場合は1ステップ実行をします。

\$D、\$Eの指定は、ニモニックを\$Dでは表示を行わず、\$Eでは表示を行いません。

省略した場合は前回の1ステップ・エミュレーション時の指定によります。

初期値は\$Eです。\$Rの指定はレジスタの内容の表示を行いません。

指定のない場合は表示を行いません。

レジスタ条件のvalueは指定したレジスタのビット数と同じビット数としてあつかわれます。

ESC のシーケンスは通常リアルタイム・エミュレーションと同様とします。

例

1> RUN T 10, 5 \$R) _____ (1)

1> RUN T , PC=19 \$R) _____ (2)

1> RUN T) _____ (3)

(1) 10番地から1ステップ・エミュレーションをレジスタの内容を表示しながら5ステップだけ実行します。

以後は、1ステップ動作となります。

(2) 現在のプログラム・カウンタの値からPCが19になるまでレジスタの内容を表示しながら実行し、以後1ステップ動作となります。

(3) 現在のプログラム・カウンタの内容がスタート・アドレスに指定されて、1ステップ実行し

保守/廃止

ます。

実行例

1> RUN T 10,5\$R)

```
ADRS   OBJECT   LABEL           MNEMONIC
0010   00
```

```
PSW V A B C D E H L EA SP PC
00 02 00 80 80 80 80 00 00 0000 0700 0011
V' A' B' C' D' E' H' L' EA'
00 FF FF FF 01 4C 00 00 014C
```

```
ADRS   OBJECT   LABEL           MNEMONIC
0011   00
```

```
PSW V A B C D E H L EA SP PC
00 02 00 80 80 80 80 00 13 0000 0700 0012
V' A' B' C' D' E' H' L' EA'
00 FF FF FF 01 4C 00 00 014C
```

```
ADRS   OBJECT   LABEL           MNEMONIC
0012   00
```

```
PSW V A B C D E H L EA SP PC
00 02 00 80 80 80 80 00 00 0000 0700 0013
V' A' B' C' D' E' H' L' EA'
00 FF FF FF 01 4C 00 00 014C
```

```
ADRS   OBJECT   LABEL           MNEMONIC
0013   00
```

```
PSW V A B C D E H L EA SP PC
00 02 00 80 80 80 80 00 00 0000 0700 0014
V' A' B' C' D' E' H' L' EA'
00 FF FF FF 01 4C 00 00 014C
```

```
ADRS   OBJECT   LABEL           MNEMONIC
0014   00
```

```
PSW V A B C D E H L EA SP PC
00 02 00 80 80 80 80 00 00 0000 0700 0015
V' A' B' C' D' E' H' L' EA'
00 FF FF FF 01 4C 00 00 014C
```

ONE STEP EMULATION STANDBY)

1>

保守/廃止

I> RUN T PC=19\$R)

ADRS	OBJECT	LABEL	MNEMONIC
0015	00		NOP

PSW	V	A	B	C	D	E	H	L	EA	SP	PC
00	02	00	80	80	80	80	00	00	0000	0700	0016
	V'	A'	B'	C'	D'	E'	H'	L'	EA'		
	00	FF	FF	FF	01	4C	00	00	014C		

ADRS	OBJECT	LABEL	MNEMONIC
0016	00		NOP

PSW	V	A	B	C	D	E	H	L	EA	SP	PC
00	02	00	80	80	80	80	00	13	0000	0700	0017
	V'	A'	B'	C'	D'	E'	H'	L'	EA'		
	00	FF	FF	FF	01	4C	00	00	014C		

ADRS	OBJECT	LABEL	MNEMONIC
0017	00		NOP

PSW	V	A	B	C	D	E	H	L	EA	SP	PC
00	02	00	80	80	80	80	00	00	0000	0700	0018
	V'	A'	B'	C'	D'	E'	H'	L'	EA'		
	00	FF	FF	FF	01	4C	00	00	014C		

ADRS	OBJECT	LABEL	MNEMONIC
0018	00		NOP

PSW	V	A	B	C	D	E	H	L	EA	SP	PC
00	02	00	80	80	80	80	00	13	0000	0700	0019
	V'	A'	B'	C'	D'	E'	H'	L'	EA'		
	00	FF	FF	FF	01	4C	00	00	014C		

ONE STEP EMULATION STANDBY)

I>

保守/廃止

7.22 スtring・コマンド

```
STR filename )
```

ファイルに登録したIE-78C11システム・ソフトの一連のコマンドを、そのfilenameを指定することによって連続自動実行します。

filename : コマンドに登録したファイルの名前

ファイル・タイプを省略した場合、ファイル・タイプは‘STR’となります。

例

```
1> STR B:ABC ) ----- (1)
```

```
1> STR XYZ.SRC ) ----- (2)
```

(1) ドライブBのABC. STRに登録してある一連のコマンドを実行します。

(2) カレント・ディスクのXYZ. SRCに登録してある一連のコマンドを実行します。

CTRL Y入力によりString実行は中止できます。

実行例

```
A> TYPE B:MANU12.STR )
```

```
MAP W XXXX
MEM F XX 0
MEM D XX
MEM D X
```

上記のファイルをSTRコマンドにより実行すると以下ようになります。

```
1> STR B:MANU12 )
```

```
1> MAP W XXXX )
```

```
1> MEM F XX 0 )
```

```
1> MEM D XX )
```

```
0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
1> MEM D X )
```

```
0000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1>
```

7.23 自己診断コマンド

DIG)

IE-78C11ハードウェアの検査(自己診断)を行ないます。

このコマンドによりIE-78C11の動作が不良と思われるとき、その場で診断することができます。

自己診断機能は、以下の項目をチェックします。

- (1) 外部発振回路
- (2) 内部メモリ
- (3) MODE0, MODE1
- (4) ポートD, ポートF
- (5) ポートA, ポートB, ポートC
- (6) アナログ入力
- (7) シリアル入出力

DIGコマンドを使用する場合は以下の手順で行なってください。

- (1) IE-78C11システム・ソフトを終了します。
- (2) IE-78C11ターゲット・システムの電源を切りエミュレーション・プローブをドライバの自己診断用コネクタに挿入しIE-78C11の電源を投入します。このときプローブのスイッチは'XTL'側にします。
- (3) IE-78C11 Cを起動させます。
(IE-78C11-Mの場合は3章を参照してください。)
- (4) イニシャライズを行ないます。
- (5) 自己診断コマンド(DIG)を実行します。

保守/廃止

実行例

```

I> EXT ) _____ (1)
A> IE78C11 ) _____ (2)
IE-78C11 System Software Vx.x[dd mmm yy]
Copyright(c)xxxx NEC Corporation
Select Channel Number ? (1/2/3/4) : 1 _____ (3)
Initialize ? (Y/N) : Y ) _____ (4)
USER-SYSTEM VDD-ON
CPU 78C10
EXTERNAL MEMORY SIZE 64K BYTE
RAE (E:ENABLE,D:DISABLE)=D
I> DIG ) _____ (5)
EXTERNAL CLOCK TEST.OK
ALTERNATE RAM TEST...OK
USER MEMORY TEST....OK
HARDWART TEST
MO,M1.....OK
PORT D,F....OK
PORT A,B,C...OK
ANALOG IN...OK
SERIAL I/O...OK
DIAGNOSTIC COMPLETE
IE-78C11 MONITOR Vx.x[dd mmm yy]
COPYRIGHT (c)xxxx NEC CORPORATION
USER-SYSTEM VDD-ON
CPU 78C10
EXTERNAL MEMORY SIZE 64K BYTE
RAE(E:ENABLE,D:DISABLE)=E
I>

```

注 自己診断の結果NGの場合はIE-78C11の電源を切った上で購入先にご連絡ください。

保守/廃止

7.24 ディレクトリ・コマンド

```
DIR [ { filename } ] )
```

ファイル名を表示します。

filename : ファイル名

filematch : ファイル名の省略

複数のファイルを一度に指定する場合に用いるキャラクタは文字、数字、ピリオドおよび*、?という2個の特殊な記号を使います。

例

```
1> DIR B: ) _____ (1)
1> DIR TEST. HEX ) _____ (2)
1> DIR *. HEX ) _____ (3)
```

- (1) ドライブB上のすべてのファイル名をリストアウトします。
- (2) カレント・ディスク上のTEST. HEXファイルの有無を確認します。
- (3) カレント・ディスク上のファイル名のエクステンションがHEXのファイル名すべてをリストアウトします。

実行例

```
1> DIR )
A: IE78C11      COM : IE87      HLP
1> DIR B: )
B: TEST1      HEX : TEST      SYM : PIP      COM : TEST     HEX
B: TEST2      SYM : MANU11    TXT : SS       SYM : TTT      HEX
B: SS         HEX : TTT      SYM : SSS      HEX : SSS      SYM
B: MANU2      TXT : MANU3    TXT : MANU0    TXT : MANU4    TXT
B: MANU5      TXT : MANU8    TXT : MANU7    TXT : MANU6    TXT
B: MANU9      TXT : CHDAS    SYM : CHDAS    HEX : MANU10   TXT
B: SSSS      TXT : MMAIN    SYM : MANU1    TXT
1> DIR B:*.TXT )
B: MANU11    TXT : MANU2    TXT : MANU3    TXT : MANU0    TXT
B: MANU4     TXT : MANU5    TXT : MANU8    TXT : MANU7    TXT
B: MANU6     TXT : MANU9    TXT : MANU10   TXT : SSSS     TXT
B: MANU1     TXT
```

保守/廃止

7.25 リスト・コマンド

LST[filename])

コンソールに出力されるすべての出力を、指定したファイルへ書込みます。

filename : ファイル名

ファイル・タイプを省略した場合、ファイル・タイプは 'TXT' となります。

指定したファイルが R/O または SYS 属性のときは 'File Already Exists' と出力し書込みは行ないません。

例

1> LST B:TEST) _____ (1)

1> LST) _____ (2)

(1)から(2) (または 'EXT)') の入力までにコンソール上に出力されたすべての出力を、指定した B ドライブの TEST. TXT に書込みます。

実行例

1> LST B:TEST)

1> MEM D 0,7)

0000 00 00 00 00 00 00 00 00

1> DAS 200,202)

ADRS	OBJECT	LABEL	MNEMONIC
200	3400FF	ADDATA:	LXI H,BUFF0
201	2410FF		LXI D,BUFF1
202	400B10		CALL BFADD

1> LST)

1>

この部分が B ドライブの
TEST ファイルに書込
まれます。

保守/廃止

7.26 終了コマンド

EXT)

このコマンドを実行すると、CP/MまたはMP/M-86に戻ります。

このコマンドはシステム・ソフトの使用を終了するときに使用します。

LSTコマンド実行後にこのコマンドを実行すると、ファイルの書込みを終了しCP/M, またはMP/M-86へ戻ります。

例

1> EXT)

CP/Mへ戻ります。

実行例

1> EXT)

A>

保守/廃止

7.27 ページ・コマンド

$$\text{PAG } \left\{ \begin{array}{l} \text{D} \\ \text{C } \text{vreg} \end{array} \right\} \text{)}$$

Vレジスタの値を表示したり変更するコマンドです。

vregの範囲は0H~FFHの値です。

このコマンドによりセットするVレジスタの値ではありません。

これは逆アセンブル表示に使用するためのコマンドです。

例

```
1> PAG D ) _____ (1)
```

```
1> PAG C FF ) _____ (2)
```

(1) Vレジスタの値を表示します。

(2) Vレジスタの値をFFHに指定します。

実行例

```
1> PAG D )
```

```
FF
1> SYM_D )
```

[APPEND]

```
SD002    FF02 D    SDSTA    5502 D
```

```
1> DAS C )
```

ADRS	OBJFCT	LABEL	MNEMONIC	
000C	74D002		ADCW SD002	← シンボル値がFF02のシンボルを表示します。

```
1> DAS C )
```

ADRS	OBJECT	LABEL	MNEMONIC	
000C	74D002		ADCW SDSTA	← シンボル値が5502のシンボルを表示します。

```
1> PAG C 33 )
```

ADRS	OBJECT	LABEL	MNEMONIC	
000C	74D002		ADCW 00002H	← シンボル値が3302のシンボルが存在しない場合は、シンボルを表示しません。

保守/廃止

7.28 PGM コマンド

PGM)

IE-78C11のCH2とPG-1000, PG-2000をRS-232-Cにより接続した場合に、PG-1000, PG-2000をIE-78C11側のコンソールにより制御可能とするコマンドです。

CTRL Zによりコマンドは終了します。

1> PGM)

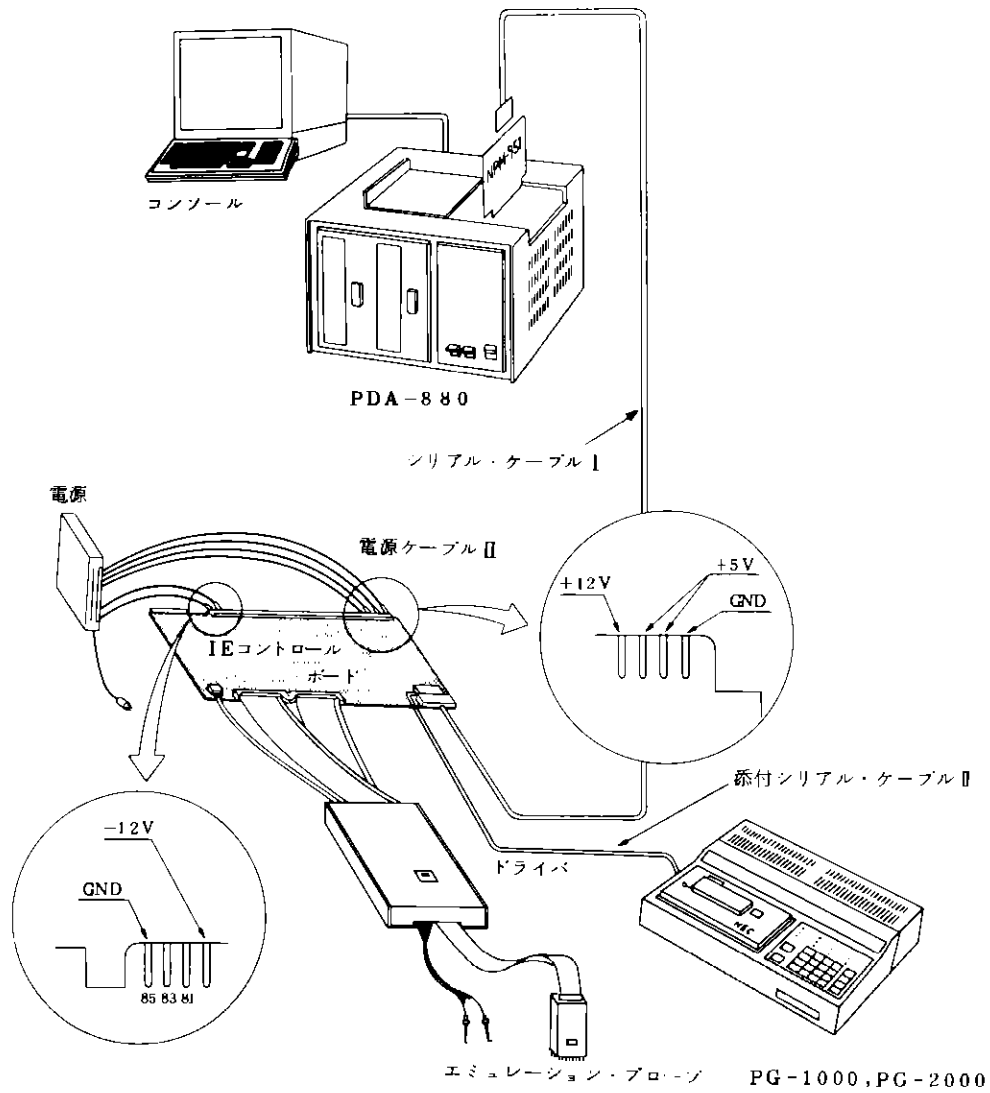
PGM MODE …… これよりIE-78C11のCH2に接続されているPG-1000, PG-2000の操作が、IE-78C11側のコンソールにより可能となります。ここで、PG-1000の場合RES-SERI, PG-2000の場合RES-REM-STARTと入力してください。

注 CH1, CH2のボーレートはDP1により同じ値に設定されます。

このコマンドはIE-78C11-Cでのみ使用できます。

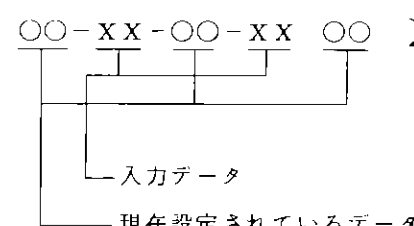
保守/廃止

図7-28 IE-78C11とPG-1000, PG-2000の接続図



保守/廃止

PG-1000, PG-2000 のコマンド一覧

A	*As, e, r)	パラメータの設定を行ないます。
E	*Er)	PGバッファのデータの変更です。 形式は以下のとおりです。 *Er) r ○○-XX-○○-XX ○○) 
F	*Fr, re, d)	PGバッファの初期設定を行ないます。
I	*I)	インテリジェント・モード (コンソールへエコー・バックするモード) へ移行します。
J	*J)	PTRより入力します。(PG-1000のみ) パリティ・エラー時は '?' を表示します。
O	*Or, re)	PGバッファの内容表示です。 表示形式は以下のとおりです。 r 00 00 00 00 00 00 00
R	*Rr, re)	PROMの0番地よりの内容をPGバッファへ読み出します。
S	*S)	PROMセレクトです。
T	*T)	トランジェント・モード (コンソールへエコー・バックしないモード) へ移行します。
V	*Vs, e, r)	PROMの内容とPGバッファの内容との比較し、データが異なるときは '?' を表示します。
W	*Ws, e, r)	PGバッファの内容をPROMに書込みます。ライト・エラー時は '?' を表示します。
Y	*Y)	現在のパラメータを表示します。
Z	*Z)	PROMブランク・チェックです。
L	*Lbias) PARTITION=i, ie)	IE-78C11のマッピングされているi番地からie番地までのデータをPGのバッファのi+bias番地からie+bias番地まで転送します。
P	*Pr, re) BIAS=i	PGのバッファのr番地からre番地までのデータをIE-78C11のマッピングされているr+i~re+i番地まで転送します。

保守/廃止

s : PROM開始番地
 e : PROM終了番地
 r : PGバッファ開始番地
 re : PGバッファ終了番地
 d : バイト・データ
 i : IE-78C11のスタート・アドレス
 ie : IE-78C11のエンド・アドレス

エラー条件

S > e

r > re

16進以外の入力

PGMコマンドを実行しPG-1000, PG-2000を操作する例を示します。

実行例

```

I> MAP W XXXX ) _____ (1)
I> MEM F XXXX 0 ) _____ (2)
I> PGM ) _____ (3)

PGM MODE

*I ) _____ (4)
*FO,3FFF,0 ) _____ (5)
*R ) _____ (6)
*00,3F ) _____ (7)
0000 69 00 6A 01 6B 02 6C 03 6D 04 6E 05 6F 06 68 07
0010 48 AF 48 AC 69 10 6A 11 6B 12 6C 13 6D 14 6E 15
0020 6F 16 68 17 54 24 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
*PG,27 ) _____ (8)
BIAS = 0 ) _____ (9)
*
PGM END
I> MEM D 0,3F ) _____ (10)

0000 69 00 6A 01 6B 02 6C 03 6D 04 6E 05 6F 06 68 07
0010 48 AF 48 AC 69 10 6A 11 6B 12 6C 13 6D 14 6E 15
0020 6F 16 68 17 54 24 00 00 00 00 00 00 00 00 00
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I>
  
```

- (1) エミュレーション・メモリをRAMとしてマッピングします。
- (2) エミュレーション・メモリをクリアします。
- (3) これ以降はPGの入力となります。
- (4) 'I)'と入力します。
- (5) PGのメモリをクリアします。
- (6) ROMの内容をPGのメモリへ読み込みます。
- (7) リードした内容(PG)を表示します。
- (8) PG-1000, PG-2000のメモリの内容をエミュレーション・メモリへ転送します。
- (9) **CTRL** Zを入力しPGMコマンドを終了します。
- (10) これ以降はIEの入力状態となります。

保守/廃止

付 録

付. 1 IE-78C11-C コントロール・キー一覧

コントロール・キー	説 明
[DEL] , [RUB]	最後に入力した文字を1文字ずつ削除します。 (削除された文字はスクリーンから消えます。)
[CTRL] H	同 上
[CTRL] U	入力した行の全部をキャンセルします。
[CTRL] X	入力した行の全部をキャンセルしスクリーン上からも消去します。
[CTRL] R	入力したラインを次のラインに再表示します。
[CTRL] E	表示上だけの復帰, 改行を行ないます。
[CTRL] M	キャリッジ・リターンと同様です。
[CTRL] J	同 上
[CTRL] C	コマンドの実行を中止します。 (ただしE-CPUが動作中は強制ブレイクせずにプロンプトに戻ります。)
[CTRL] S	出力の一時停止です。
[CTRL] Q	出力の再開です。
[CTRL] P	コンソール出力を同時にLSTデバイスへも出力し, 再度の[CTRL] P入力により解除されます。
[CTRL] D	画面表示の停止です。 (CPU実行を終了しコンソール待ちになるとプロンプトを出力します。)
[ESC]	[CTRL] Cと同じ動作をします。 (ただしE-CPUが動作中はブレイクレジスタ等を表示します。)
[CTRL] Y	ストリング・コマンドの実行を中止します。

保守/廃止

付.2 IE-78C11-C メッセージ一覧

メッセージ	内 容
UNRECOGNIZED COMMAND	<p>解説不能なコマンドです。</p> <p>(該当するコマンドが見つからなかったことを示します。)</p> <p>コマンドを確認の上、再入力してください。</p>
COMMAND FORMAT ERROR	<p>文法誤りです。</p> <p>(コマンド入力形式が違っていることを示します。)</p> <p>コマンドを確認の上、入力し直してください。</p>
MAPPING ERROR	<p>アドレスの指定誤りです。(マッピングされていないアドレスが指定されていることを示します。)</p> <p>マッピング・エリアを入力し直すかアドレスの指定を変更してください。</p>
NON-MAP AREA ACCESS	<p>マッピング・エリア・オーバー(コマンド処理中にマッピングされていないエリアをアクセスしたことを)を示します。</p> <p>マッピング・エリアを確認してください。</p>
MEMORY FAILURE	<p>メモリの異常です。</p> <p>(ユーザ・メモリ(IE-78C11の内部メモリ)に、エラーが発生したことを示します。)</p> <p>自己診断(DIGコマンド)を行ない内部メモリを確認してください。</p>
SYSTEM DOWN	<p>システムの異常です。(IE-78C11のファームウェアが正しく動作していないことを示します。)</p> <p>リセットして再スタートしてください。</p>
NO CONNECT	<p>IEとMD 080シリーズ/PDA-880が正しく接続されていない場合や、ユーザ・システムが正しく設定されていない場合を示します。接続、および設定を確認してください。</p>
LOAD FAILED	<p>ロード中に異常があったことを示します。</p> <p>ロード・コマンドを確認の上、コマンドを再実行してください。</p>
SAVE FAILED	<p>セーブ中に異常があったことを示します。</p> <p>セーブ・コマンドを確認の上、コマンドを再実行してください。</p>
FILE NOT FOUND	<p>指定したファイルが見つからないことを示します。</p> <p>DIRコマンドによりファイル名を確認してください。</p>
DIRECTORY FULL	<p>ディレクトリがいっぱいであることを示します。</p> <p>ディスクを交換し、最初から起動してください。</p>

保守/廃止

メ ッ セ ー ジ	内 容
DISK FULL	ディスクがいっぱいになり書込みができないことを示します。 ディスクを交換してください。
SYMBOL NOT FOUND	指定したシンボルが見つからないことを示します。 シンボル名の確認をしてください。
RESERVED WORD	指定したシンボルが予約語であるためにシンボルとして使用できないことを示します。 予約語でないシンボルを指定してください。
COMMAND NOT FOUND	指定したコマンドが見つからなかったことを示します。 コマンド名を確認の上、入力し直してください。
IE87. HLP NOT FOUND ON DEFAULT OR A:	ヘルプ・ファイルがデフォルトまたはAドライブに存在しないことを示します。
MODULE NOT FOUND	指定したモジュールが見つからなかったことを示します。 モジュール名を確認の上入力し直してください。
COMMAND ERROR	コマンドの使用方法が誤っていることを示します。 コマンドの取扱説明を確認してください。
NO FILE	ディレクトリが見つからなかったことを示します。 ディレクトリを確認の上入力し直してください。
TYPE ERROR	シンボル・タイプが誤っていることを示します。 シンボル・タイプを確認してください。
OVER FLOW	シンボル領域がオーバ・フローになったことを示します。
INPUT DATA ERROR	入力されたデータは使用できません。 データを入力し直してください。
E-CPU RUN	ユーザ・プログラム実行中に CTRL C を入力したときで、 CPU が動作していることを示します。 ブレイクさせるときは ESC またはリセット・コマンドを入力 してください。
POINTER OVER SET NEWEST POINTER OVER SET OLDEST	トレース・ポインタの移動が実行結果のフレーム数を越えて設定さ れたことを示します。
ADDRESS OVER	PGM コマンド実行時に PG でロード・コマンドが動作していると きアドレスをオーバしたことを示します。
CHECK SUM ERROR	ファイル中の 16 進形式オブジェクト・コードのチェック・サム値 が一致しません。16 進形式オブジェクト・コードをつくりなおし てください。
ILLEGAL FILE	入力されたファイルが異常です。

保守/廃止

メ ッ セ ー ジ	内 容
SERIAL I/O ERROR	シリアル回線の異常です。 シリアル回線を確認してください。
FILE ALREADY EXISTS	指定したファイルがSYS属性の場合、またはR/Oの場合です。 ファイル・ネームの指定を変更してください。
FILE ALREADY EXISTS, DELETE (Y/N):	指定したファイルが指定されたドライブに存在する場合は、 Yと入力すると書き込みを開始します。以前のファイルを保護する場 合はNを入力しファイル・ネームの指定を変更してください。
COMMAND TOO LONG	コマンドの入力行で字数が多すぎます。
NON BREAK	強制ブレークを行なってもブレークしない時出力されます。 RESコマンドでブレークしてください。
FILE CLOSE ERROR	クローズしようとしたファイルがディスク上にみつきりません。
NON TRACE DATA	トレース・データがありません。

保守/廃止

付.3 IE-78C11-M コントロール・キー一覧

コントロール・キー	説明
DEL , RUB	最後に入力した文字を1文字ずつ削除します。 (削除された文字はスクリーンから消えます。)
CTRL H	同上
CTRL U	入力した行の全部をキャンセルします。
CTRL X	入力した行の全部をキャンセルしスクリーン上からも消去します。
CTRL R	入力したラインを次のラインに再表示します。
CTRL E	表示上だけの復帰, 改行を行いません。
CTRL M	CRLFと同様です。
CTRL J	同上
CTRL C	コマンドの実行を中断して‘ABORT (Y/N)?’というメッセージが表示されます。ここで‘Y’を入力するとコマンドの実行を中止して, MP/Mへ戻ります。 ‘N’を入力すると元のコマンドの実行を再開します。
CTRL S	出力の一時停止です。
CTRL Q	出力の再開です。
CTRL P	コンソール出力を同時にLSTデバイスへも出力し再度の CTRL P入力により解除されます。
CTRL D	システム・ソフトをコンソールから分離します。 (再入力によりコンソール接続待ちとなっているシステム・ソフトをコンソールに接続します。)
ESC	コマンドの実行を中止します。 (ただしE-CPUが動作中はブレークレジスタ等を表示します。)
CTRL Y	STRコマンドを中止します。

保守/廃止

付. 4 IE-78C11-M メッセージ一覧

メッセージ	内 容
UNRECOGNIZED COMMAND	<p>解説不能なコマンドです。 (該当するコマンドが見つからなかったことを示します。)</p> <p>コマンドを確認の上、再入力してください。</p>
COMMAND FORMAT ERROR	<p>文法誤りです。 (コマンド入力形式が違っていることを示します。)</p> <p>コマンドを確認の上、入力し直してください。</p>
MAPPING ERROR	<p>アドレスの指定誤りです。(マッピングされていないアドレスが指定されていることを示します。)</p> <p>マッピング・エリアを入力し直すかアドレスの指定を変更してください。</p>
NON-MAP AREA ACCESS	<p>マッピング・エリア・オーバ(コマンド処理中にマッピングされていないエリアをアクセスしたことを)を示します。</p> <p>マッピング・エリアを確認してください。</p>
MEMORY FAILURE	<p>メモリの異常です。 (ユーザ・メモリ(IE-78C11の内部メモリ)に、エラーが発生したことを示します。)</p> <p>自己診断(DIGコマンド)を行ない内部メモリを確認してください。</p>
SYSTEM DOWN	<p>システムの異常です。(IE-78C11のファームウェアが正しく動作していないことを示します。)</p> <p>リセットして再スタートしてください。</p>
NO CONNECT	<p>接続したIE番号と設定したIE番号がまちがっている場合やユーザ・システムが正しく設定されていない場合を示します。接続したIE番号またはユーザ・システムの確認してください。</p>
LOAD FAILED	<p>ロード中に異常があったことを示します。</p> <p>ロード・コマンドを確認の上、コマンドを再実行してください。</p>
SAVE FAILED	<p>セーブ中に異常があったことを示します。</p> <p>セーブ・コマンドを確認の上、コマンドを再実行してください。</p>
FILE NOT FOUND	<p>指定したファイルが見つからないことを示します。</p> <p>DIRコマンドによりファイル名を確認してください。</p>
DIRECTORY FULL	<p>ディレクトリがいっぱいであることを示します。</p> <p>ディスクを交換し、最初から起動してください。</p>

保守/廃止

メッセージ	内 容
DISK FULL	ディスクがいっぱいになり書込みができないことを示します。 ディスクを交換してください。
SYMBOL NOT FOUND	指定したシンボルが見つからないことを示します。 シンボル名の確認をしてください。
RESERVED WORD	指定したシンボルが予約語であるためにシンボルとして使用できないことを示します。 予約語でないシンボルを指定してください。
COMMAND NOT FOUND	指定したコマンドが見つからなかったことを示します。 コマンド名を確認の上、入力し直してください。
IE87. HLP NOT FOUND ON DEFAULT OR A:	ヘルプ・ファイルがデフォルトまたはAドライブに存在しないことを示します。
MODULE NOT FOUND	指定したモジュールが見つからなかったことを示します。 モジュール名を確認の上入力し直してください。
COMMAND ERROR	コマンドの使用法が誤っていることを示します。 コマンドの取扱説明を確認してください。
NO FILE	ディレクトリが見つからなかったことを示します。 ディレクトリを確認の上入力し直してください。
TYPE ERROR	シンボル・タイプが誤っていることを示します。 シンボル・タイプを確認してください。
OVER FLOW	シンボル領域がオーバーフローになったことを示します。
INPUT DATA ERROR	入力されたデータは使用できません。 データを入力し直してください。
POINTER OVER SET NEWEST POINTER OVER SET OLDEST	トレース・ポインタの移動が実行結果のフレーム数を越えて設定されたことを示します。
E-CPU RUN	CPUが動作していることを示します。 ブレイクさせる時はESCまたはリセット・コマンドを入力してください。
NON BREAK	強制ブレイクを行ってもブレイクしないとき出力されます。ファームウェアが正しく動作していないことを示します。 RESコマンドでブレイクしてください。
CHECK SUM ERROR	ファイル中の16進形式オブジェクト・コードのチェック・サム値が一致しません。16進形式オブジェクト・コードをつくりなおしてください。
ILLEGAL FILE	入力されたファイルが異常です。

保守/廃止

メッセージ	内 容
VERIFY ERROR	データの読みができません。 マッピング領域、およびデータを確認してください。
FILE ALREADY EXISTS	指定したファイルがSYS属性の場合、またはR/Oの場合です。 ファイル・ネームの指定を変更してください。
FILE ALREADY EXISTS DELETE (Y/N)	指定したファイルが指定されたドライブに存在する場合は、 Yと入力すると書き込みを開始します。以前のファイルを保護する場 合はNを入力しファイル・ネームの指定を変更してください。
COMMAND TOO LONG	コマンドの入力行で字数が多すぎます。
FILE CLOSE ERROR	クローズしようとしたファイルがディスク上にみつかりません。
NON TRACE DATA	トレース・データがありません。

保守/廃止

付.5 予約語一覧

A	ACI	ADC	ADCW
ADCX	ADD	ADDNC	ADDNCW
ADDNCX	ADDW	ADDX	ADI
ADINC	AN4	AN5	AN6
AN7	ANA	ANAW	ANAX
AND	ANI	ANIW	AT
ANW			
B	BIT	BITADDRE	BLOCK
BYTE			
C	CALB	CALF	CALLTABL
CALL	CALT	CLC	CODE
COMMON	COMPLETE	CR0	CR1
CR2	CR3	CSEG	CY
D	DAA	DADC	DADD
DADDNC	DAN	DATA	DB
DCR	DCRW	DCX	DEQ
DGT	DI	DIV	DLT
DMOV	DNE	DOFF	DON
DOR	DRLI	DRLR	DS
DSBB	DSEG	DSLL	DSLRL
DSUB	DSUBNB	DW	DXR
E	EA	EADD	EAH
EAL	ECNT	ECPT	EI
END	EOF	EOM	EQ
EQA	EQAW	EQAX	EQI
EQIW	EQU	ER	ESUB
ETM0	ETM1	ETMM	EXA
EXH	EXTRN	EXX	
F1	F2	FAD	FE0
FE1	FEIN	FIXEDARE	FNMI
FSR	FST	FT0	FT1
GE	GJMP	GT	GTA
GTAW	GTAX	GTI	GTIW
H	HC	HIGH	HLT

保守/廃止

INPAGE	INR	INRW	INX
JB	JEA	IMP	JR
JRE			
L	LBCD	LDAW	LDAX
LDEAX	LDED	LE	LHLD
LOW	LSPD	LT	LTA
LTAW	LTAX	LTI	LIW
LXI			
MA	MB	MC	MCC
MF	MKH	MKL	MM
MOD	MOV	MUL	MVI
MVIW	MVIX		
NAME	NE	NEA	NEAW
NEAX	NEGA	NEI	NEIW
NOP	NOT	NOTHING	NUMBER
OFFA	OFFAW	OFFAX	OFFI
OFFIW	ONA	ONAW	ONAX
ONI	ONIW	OR	ORA
ORAW	ORAX	ORG	ORI
ORIW	OV		
PA	PAGE	PB	PC
PD	PF	POP	PUBLIC
PUSH			
RET	RETI	RETS	RLD
RLL	RLR	RRD	RXB
SB	SBB	SBBW	SBBX
SBCD	SBI	SDED	SET
SHL	SHLD	SHR	SK
SKIT	SKN	SKNIT	SLL
SLLC	SLR	SLRC	SMH
SML	SOFTI	SP	SSPD
STACK	STAW	STAX	STC
STEAX	STKLN	STOP	SUB
SUBNB	SUBNBW	SUBNBX	SUBW
SUBX	SUI	SUINB	
TABLE	TM0	TM1	TMM
TXB			

保守/廃止

UNIT			
V	VREG		
XOR	XRA	XRAW	XRAX
XRI			
Z	ZCM		

保守/廃止

付.6 コマンド一覧

コマンド名		形式
ヘルプ・コマンド		HLP [command]
演算コマンド		MAT expression 演算子 +, -, *, /, AND, OR, NOT, XOR
数値基数指定 コマンド	指定	SUF $\left\{ \begin{array}{c} H \\ T \\ Q \\ Y \end{array} \right\}$
	表示	SUF
マッピング・ コマンド	指定	MAP $\left\{ \begin{array}{c} W \\ R \\ U \end{array} \right\}$ partition
	表示	MAP $\left[\begin{array}{c} W \\ R \\ U \end{array} \right]$
	解除	MAP K [partition]
シンボル・コマンド	表示	SYM D [[module-name]symbol]
	変更	SYM C [module-name]symbol
	追加	SYM A symbol
	削除	SYM K
モジュール・ コマンド	指定	MOD module-name
	表示	MOD
	削除	MOD -
リセット・コマンド	IE	RES H
	チップ	RES
クロック・コマンド	指定	CLK $\left\{ \begin{array}{c} U \\ I \end{array} \right\}$
	表示	CLK
メモリ・コマンド	変更	MEM C [addr]
	表示	MEM D $\left\{ \begin{array}{c} [addr] \\ [partition] \end{array} \right\}$
	比較	MEM V partition addr
	検索	MEM G partition data-string

保守/廃止

コマンド名		形 式
メモリ・コマンド	転送	MEM M partition addr
	イニシャ ライズ	MEM F partition data-string
	交換	MEM X partition addr
	テスト	MEM E [partition][\$A]
アセンブル・コマンド		ASM [addr]
逆アセンブル・コマンド		DAS [{ partition } [addr]]
メモリ転送コマンド		MOV { I U } partition addr
ロード・コマンド		LOD filename [{ C S }]
セーブ・コマンド		SAV { filename [C] partition } [filename S]
ベリファイ・コマンド		VRY filename
レジスタ・コマンド	変更	REG C [register-name]
	表示	REG D [register-name]
スペシャル・ レジスタ・コマンド	変更	SPR C [special-register-name1]
	表示	SPR D [special-register-name2]
モード・レジスタ・ コマンド	変更	MDR C [mode-register-name1]
	表示	MDR D [mode-register-name2]
ブレイク・ コントロール・ コマンド群	BRA	BRA [A=addr] [V=values] [C=code] [L=loop]
	BRD	BRD [data]
	BRE	BRE [count]
	BRT	BRT [time]
	BR 0~3	BR { 0 1 2 3 } [[BRA], [BRD], [BRE], [BRT]]
BRM	BRM [[BRA], [BRD], [BRE], [BRT], [BR0], [BR1], [BR2], [BR3]]	
トレース・ コマンド群	TRX	TRX [A=addr] [V=values] [C=cond] [PA=values] [PB=values]
	TRD	TRD [{ ALL [上] time-No. }]

保守/廃止

コマンド名		形 式
トレース・ コマンド群	TRP	TRP $\left[\begin{array}{c} [+] \text{ line-No.} \\ 0 \\ N \end{array} \right]$
	TRC	TRC $\left[\begin{array}{c} M \\ I \end{array} \right]$
	TRS	TRS $\left[\begin{array}{c} PB \\ EX \end{array} \right]$
エミュレー ション・ コマンド	通常 リアルタイム	RUN N [addr]
	ブレーク 条件つき	RUN B [addr]
	指定ステップ数 実行	RUN S [addr][, step-No.]
	1ステップ・ エミュレー ション	RUN T [addr] $\left[\begin{array}{c} , \text{ step-No.} \\ \left. \begin{array}{c} = \\ > \\ < \\ => \\ >= \\ = < \\ <= \\ > < \\ < > \end{array} \right\} \text{ value} \end{array} \right]$ [\$E] [\$R]
string コマンド		STR filename
自己診断コマンド		DIG
ディレクトリ・コマンド		DIR $\left[\begin{array}{c} \text{filename} \\ \text{filematch} \end{array} \right]$
		LST [filename]
終了コマンド		EXT
ページ・コマンド	表示	PAG D
	変更	PAG C vreg
PGMコマンド		PGM