

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、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、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。

IE-78327-R

インサーキット・エミュレータ

ソフトウェア編

IE-78327-R
IE-78327-R-EM

IE-78327-R

インサーキット・エミュレータ

ソフトウェア編

IE-78327-R
IE-78327-R-EM

MS-DOS™は、米国マイクロソフト社の商標です。
PC DOS™は、米国IBM社の商標です。
PC/AT™、PC/XT™は、米国IBM社の商標です。

この装置は、第一種情報装置（商工業地域において使用されるべき情報装置）で商工業地域での電波妨害禁止を目的とした情報処理装置等電波障害自主規制協議会（VCCI）基準に適合しております。

したがって、住宅地域、またはその隣接した地域で使用すると、ラジオ、テレビジョン受信機などに受信障害を与えることがあります。

ユーザーズ・マニュアルにしたがって正しく取り扱いをしてください。

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

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

【本版で改訂された主な箇所】

箇所	内容
全般	対象デバイスを追加 (μ PD78323, 78324, 78P324 : 開発中)
p. 441-448	第12章 使用上の注意事項を追加

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

はじめに

対象者 このマニュアルは16／8ビット・シングルチップ・マイクロコンピュータ78K／Ⅲシリーズを採用し、IE-78327-Rによりシステム・ディバッグを行うエンジニアを対象としています。
したがって、このマニュアルを読むエンジニアは、μPD78320、78322、78P322、78327、78328、78P328の機能、使用方法に熟知し、ディバッグの知識があることを前提とします。

構成 IE-78327-Rのマニュアルはソフトウェア編（このマニュアル）とハードウェア編の2冊に分かれています。

ソフトウェア編

機能概要
コマンド説明

ハードウェア編

基本仕様
システム構成
外部インタフェース機能

IE-78327-Rには使用時の制限事項があります。
詳細については「第12章 使用上の制限事項」にまとめてありますので、ご使用になる前に必ずお読みください。

目的 ソフトウェア編では、ディバッグなどの対象デバイスの開発作業に伴うIE-78327-Rの起動方法から、コマンドの実行方法までを理解していただくことを目的としています。

ハードウェア編では、IE-78327-Rの基本仕様と正しい外部装置の接続方法を理解していただくことを目的としています。

読み方

ソフトウェア編

- 基本的な操作手順、機能を理解しようとするとき
→ 「第2章 IE-78327-Rの起動方法」と「第3章 機能概要」を参照してください。

- 具体的なコマンドの種類、機能、入力フォーマットを理解しようとするとき
→ 「第8章 コマンドの詳細」を参照してください。

- 一通りIE-78327-Rのコントロール・ソフトウェアの機能を知りたいとき
→ 目次に従って読んでください。

ハードウェア編

- 基本仕様を理解しようとするとき
→ 「第1章 概要」と「第2章 セット・アップ」を参照してください。

- IE-78327-Rと外部機器を接続するとき
→ 「1.3 システム構成」、「第5章 ターゲット・プローブの接続」、「第6章 周辺装置の接続」、「第7章 ターゲット・システムの接続」を参照してください。

- 対象デバイスとIE-78327-Rのターゲット・インタフェース回路の相違を知りたいとき
→ 「第8章 ターゲット・インタフェース回路」を参照してください。

- IE-78327-Rのシリアルまたはパラレルのインタフェースの機能を詳しく知りたいとき
→ 「第9章 チャネル1とチャネル2の機能」、または「第10章 チャネル3とチャネル4の機能」を参照してください。

用語について

このマニュアルの中で使用する用語について、その意味を下表に示します。

用語	意味
エミュレーション・デバイス	エミュレータ内で対象デバイスのエミュレーションを行っているデバイスの総称です。 エミュレーションCPUを含みます。
エミュレーションCPU	エミュレータ内で、ユーザが作成したプログラムを実行しているCPU部分です。
対象デバイス	エミュレーションの対象となっているデバイスです（本チップ：μPD78328など）。
ターゲット・プログラム	ディバグの対象となるプログラムです（ユーザが作ったプログラム）。
ターゲット・システム	ディバグの対象となるシステムです（ユーザの作ったシステム）。 ターゲット・プログラムおよびユーザの作成したハードウェアを含みます。 狭義にはハードウェアのみを指します。

凡例

- 注 :本文中に付けた注の箇所を説明します。
- 注意 :特に気を付けて読んでいただきたい内容を表します。
- 備考 :本文の補足説明を表します。
- このマニュアルで用いている画面表示例、入力例はPC-9800シリーズをホスト・マシンとして使用した場合のものです。

その他、各章でよく使われる凡例については、その章の冒頭で説明をしています。

備考 IBM PCシリーズをホスト・マシンとした場合では、起動メッセージの一部とモジュール名の終わりを示す記号（PC-9800シリーズでは'¥'、IBM PCシリーズでは'\`'）が異なります。

目 次 要 約

- 第1章 概 説 … 1
- 第2章 IE-78327-Rの起動方法 … 9
- 第3章 機能概要 … 19
- 第4章 ディバグ手順と画面表示例 … 97
- 第5章 コマンド入力機能 … 157
- 第6章 数値、シンボル、式の記述仕様 … 167
- 第7章 コマンドの概要 … 175
- 第8章 コマンドの詳細 … 205
- 第9章 エラー・メッセージ一覧 … 369
- 第10章 オンライン・アセンブラ、逆アセンブラ仕様 … 389
- 第11章 PROMプログラマ (PG-1500、2000) の操作方法 … 433
- 第12章 使用上の注意事項 … 441

目 次

第1章 概 説 … 1

- 1. 1 特徴 … 2
- 1. 2 対象デバイス … 3
- 1. 3 IE-78327-Rの構成 … 4
- 1. 4 他機種からIE-78327-Rへのシステム・アップ … 5
- 1. 5 ホスト・マシン対象機種と対象OS … 7
 - 1. 5. 1 PC-9800シリーズの場合 … 7
 - 1. 5. 2 IBM PCシリーズの場合 … 8

第2章 IE-78327-Rの起動方法 … 9

- 2. 1 ホスト・マシンの設定 … 10
- 2. 2 コントロール・プログラムの起動 … 12
- 2. 3 起動後の操作 … 14
- 2. 4 動作環境の設定 … 15

第3章 機能概要 … 19

- 3. 1 対象デバイスの環境設定 … 20
 - 3. 1. 1 動作環境の設定 … 21
 - 3. 1. 2 クロック選択機能 … 26
 - 3. 1. 3 マッピング機能 … 27

- 3. 2 エミュレーション … 3 2
 - 3. 2. 1 リセット機能 … 3 3
 - 3. 2. 2 メモリ操作機能 … 3 4
 - 3. 2. 3 レジスタ操作機能 … 3 5
 - 3. 2. 4 実行機能 … 3 6
 - 3. 2. 5 ブレーク機能 … 4 7
 - 3. 2. 6 動作状態表示機能 … 5 3

- 3. 3 イベント検出機能 … 5 4
 - 3. 3. 1 バス・イベント検出 … 5 6
 - 3. 3. 2 外部データ検出 … 5 8
 - 3. 3. 3 プログラム実行検出 … 5 9
 - 3. 3. 4 イベント条件割り当て … 6 0
 - 3. 3. 5 バス条件 … 6 2
 - 3. 3. 6 ディレイ条件 … 6 3

- 3. 4 アナライザ … 6 5
 - 3. 4. 1 リアルタイム・トレース … 6 6
 - 3. 4. 2 実行時間、命令数計測 … 6 9
 - 3. 4. 3 内部RAMデータ・サンプル … 7 0
 - 3. 4. 4 C0カバレッジ … 7 1

- 3. 5 アセンブリ言語デバッグ機能 … 7 2
- 3. 6 PROMプログラマ制御機能 … 7 3
- 3. 7 補助機能 … 7 4
 - 3. 7. 1 オブジェクトのアップ・ロード、ダウン・ロード機能 … 7 5
 - 3. 7. 2 トリガ信号外部出力機能 … 7 6
 - 3. 7. 3 IEシンボルの操作機能 … 7 9
 - 3. 7. 4 デバッグ環境のアップ・ロード、ダウン・ロード機能 … 8 0
 - 3. 7. 5 チャネル2通信モード設定機能 … 8 1
 - 3. 7. 6 ファイルからのコマンド入力機能 … 8 2
 - 3. 7. 7 コマンド実行結果の出力機能 … 8 3
 - 3. 7. 8 コマンド・ファイルの作成機能 … 8 4
 - 3. 7. 9 コマンド・履歴の表示機能 … 8 5
 - 3. 7. 10 コマンド・ヘルプの表示機能 … 8 6
 - 3. 7. 11 ディレクトリの表示機能 … 8 7

- 3. 7. 1 2 演算機能 … 8 8
- 3. 7. 1 3 子プロセスの実行機能 … 8 9
- 3. 7. 1 4 I E - 7 8 3 2 7 - R の終了機能 … 9 0
- 3. 7. 1 5 メモリ・ワード設定機能 … 9 1
- 3. 7. 1 6 その他の機能 … 9 2

第4章 ディバグ手順と画面表示例 … 97

- 4. 1 ディバグ手順の一例 … 9 8
- 4. 2 画面表示例 … 1 0 4
 - 4. 2. 1 マクロ・サービス、割り込み処理に関するイベント検出、
トレース・データの表示 … 1 0 4
 - 4. 2. 2 外部データを用いたイベント検出、トレース・データの
表示 … 1 1 9
 - 4. 2. 3 プロシージャ実行に関するイベント検出、トレース・データ
の表示 … 1 3 0
 - 4. 2. 4 C 0 カバレッジ測定機能を用いたカバレッジ測定、測定結果
の表示 … 1 4 4

第5章 コマンド入力機能 … 157

- 5. 1 I E - 7 8 3 2 7 - R のコマンド入力機能 … 1 5 8
 - 5. 1. 1 コマンド入力形式 … 1 5 8
 - 5. 1. 2 コマンドの入力方法 … 1 6 0
 - 5. 1. 3 制御キー … 1 6 5

第6章 数値、シンボル、式の記述仕様 … 167

- 6. 1 数値の記述仕様 … 168
- 6. 2 特殊数値の記述仕様 … 170
- 6. 3 シンボルの記述仕様 … 171
- 6. 4 式の記述仕様 … 173

第7章 コマンドの概要 … 175

- 7. 1 コマンドの構造 … 176
- 7. 2 コマンド用語の定義 … 177
- 7. 3 コマンドの一覧 … 184

第8章 コマンドの詳細 … 205

- 8. 1 ライン・アセンブル (ASM) … 206
- 8. 2 バス・イベント検出の条件設定 (BRA) … 208
- 8. 3 外部データ検出の条件設定 (BRD) … 213
- 8. 4 トリガ条件の設定 (BRM) … 214
- 8. 5 プログラム実行検出の条件設定 (BRS) … 216
- 8. 6 チェック・ポイント条件の設定 (CHK) … 218
- 8. 7 クロックの選択 (CLK) … 220
- 8. 8 実行経過時間、実行命令数の表示 (CNT) … 222
- 8. 9 コマンド・ファイルの作成 (COM) … 223
- 8. 10 C0カバレッジ測定結果の表示 (CVD) … 226
- 8. 11 C0カバレッジ測定範囲の操作 (CVM) … 230
- 8. 12 逆アセンブル (DAS) … 232
- 8. 13 ディレクトリの表示 (DIR) … 233

8. 14	トリガ・ポイント位置の設定 (DLY)	… 234
8. 15	子プロセスの実行 (DOS)	… 236
8. 16	ディスエーブル条件の設定 (DSB)	… 237
8. 17	イネーブル条件の設定 (ENB)	… 239
8. 18	イベント検出器の設定状態表示 (ENV)	… 241
8. 19	コントロール・プログラムの終了 (EXT)	… 243
8. 20	コマンド・ヒストリの表示 (HIS)	… 244
8. 21	コマンド・ヘルプの表示 (HLP)	… 245
8. 22	オブジェクト、シンボル、デバッグ環境のロード (LOD)	… 246
8. 23	ファイルへの結果出力 (LST)	… 251
8. 24	マッピングの設定 (MAP)	… 254
8. 25	演算 (MAT)	… 257
8. 26	メモリの操作 (MEM)	… 258
8. 27	チャンネル2のモード設定 (MOD)	… 272
8. 28	代替メモリとユーザ・メモリ間のデータ転送 (MOV)	… 274
8. 29	トリガ信号の外部出力指定 (OUT)	… 276
8. 30	パス条件の設定 (PAS)	… 278
8. 31	PROMプログラムの制御、制御キャラクタの変更、解除 (PGM)	… 279
8. 32	サンプリング・アドレスの設定 (PSA)	… 283
8. 33	サンプル・データの表示 (PSD)	… 284
8. 34	サンプル・タイミングの設定 (PST)	… 289
8. 35	レジスタの操作 (REG)	… 290
8. 36	IE-78327-Rとエミュレーション・デバイスのリセット (RES)	… 295
8. 37	実行 (RUN)	… 296
8. 38	オブジェクト、デバッグ環境のセーブ (SAV)	… 312
8. 39	SFRの操作 (SFR)	… 316
8. 40	実行停止 (STP)	… 320
8. 41	ファイルからのコマンド入力 (STR)	… 321
8. 42	シンボルの操作 (SYM)	… 323
8. 43	トレース・データの表示 (TRD)	… 336
8. 44	トレース・データ検索条件の設定 (TRF)	… 357
8. 45	アナライザの再起動 (TRG)	… 360
8. 46	トレース・モードの設定 (TRM)	… 361
8. 47	トレース・データの選択 (TRS)	… 363
8. 48	トレース・クオリファイの設定 (TRX)	… 365

- 8. 49 オブジェクト・ファイルとメモリ内容の比較 (VRY) ... 366
- 8. 50 メモリ・ワード長の設定 (WRD) ... 368

第9章 エラー・メッセージ一覧 ... 369

- 9. 1 システム・エラー ... 370
- 9. 2 エミュレータ共通エラー ... 371
- 9. 3 アナライザ関連エラー ... 375
- 9. 4 オンライン・アセンブラ関連エラー ... 378
- 9. 5 シンボル関連エラー ... 380
- 9. 6 ファイル関連エラー (ロード/セーブ/PGM) ... 384

第10章 オンライン・アセンブラ、逆アセンブラ仕様 ... 389

- 10. 1 対象デバイスの命令セット一覧 ... 390
- 10. 2 対象デバイスの特殊機能レジスタ (sfr) 一覧 ... 407
- 10. 3 オンライン・アセンブラ仕様 ... 416
- 10. 4 逆アセンブラ仕様 ... 427

第11章 PROMプログラマ (PG-1500、2000) の操作方法 ... 433

- 11. 1 PROMプログラマの遠隔操作の開始と終了 ... 434
- 11. 2 PROMプログラマのコマンド ... 436

第12章 使用上の注意事項 … 441

12.1 IE-78327-Rに関する注意事項 … 442

12.2 開発対象製品（ μ PD78320シリーズ、 μ PD78327シリーズ）に対する注意事項 … 446

第1章 概 説

この章では I E - 7 8 3 2 7 - R の概要について説明します。

I E - 7 8 3 2 7 - R インサーキット・エミュレータは、 μ P D 7 8 3 2 2 シリーズまたは μ P D 7 8 3 2 8 シリーズを用いた応用システムのハードウェア、ソフトウェアを効率的にデバッグするための開発ツールです。

●このマニュアルでは、I E - 7 8 3 2 7 - R でエミュレーションできる対象デバイスを、 μ P D 7 8 3 2 2 シリーズと μ P D 7 8 3 2 8 シリーズという言い方で表しています。

μ P D 7 8 3 2 2 シリーズの対象デバイス

- ・ μ P D 7 8 3 2 0
- ・ μ P D 7 8 3 2 2
- ・ μ P D 7 8 P 3 2 2
- ・ μ P D 7 8 3 2 3 (開発中)
- ・ μ P D 7 8 3 2 4 (開発中)
- ・ μ P D 7 8 P 3 2 4 (開発中)

μ P D 7 8 3 2 8 シリーズの対象デバイス

- ・ μ P D 7 8 3 2 7
- ・ μ P D 7 8 3 2 8
- ・ μ P D 7 8 P 3 2 8

1.1 特徴

IE-78327-Rの特徴を次に示します。

- μ PD78322シリーズまたは μ PD78328シリーズのエミュレーションが可能
 - ・別売のターゲット・プローブとの組み合わせにより全パッケージに対応可能
- リアルタイム表示、リアルタイム・トレースが可能
 - ・豊富なブレーク、トレース機能
 - ・対象デバイスの実行を止めないで、リアルタイム・トレースの内容を見ることが可能
 - ・リアルタイム・トレーサの内容をサーチ可能
 - ・外部センス・クリップを用いて8ビットのトレース入力が可能
- プログラムのアナライズ機能
 - ・実行したメモリ・エリアの表示が可能（カバレッジ）
 - ・内部RAMの値を一定周期でトレース可能（内部RAMサンプル）
- シンボリック・デバッグ可能
- オンライン・アセンブル、逆アセンブル可能
- エミュレーション・メモリ（64Kバイト）実装
- セントロニクス・インタフェースを用いて、オブジェクト・ファイルおよびシンボル・ファイルの高速ダウン・ロードが可能（RS-232-Cインタフェースを用いたダウン・ロードの約10倍の速さ）
- 別売のエミュレーション・ボードおよび、ブレーク・ボードとIE-78327-Rのボードを交換することにより、他の78Kシリーズのエミュレータとして使用可能

1. 2 対象デバイス

IE-78327-Rでエミュレーションできるデバイスを次に示します。

表1-1 IE-78327-Rの対象デバイス

対象デバイス	内部ROM	内部RAM
μ PD78320	ROMレス	640バイト
μ PD78322	16Kバイト(マスクROM)	
μ PD78P322	16Kバイト(PROM)	
μ PD78323 ^注	ROMレス	1024バイト
μ PD78324 ^注	32Kバイト(マスクROM)	
μ PD78P324 ^注	32Kバイト(PROM)	
μ PD78327	ROMレス	512バイト
μ PD78328	16Kバイト(マスクROM)	
μ PD78P328	16Kバイト(PROM)	

注 開発中

1.3 IE-78327-Rの構成

IE-78327-Rは、次のハードウェア（筐体、ボード）で構成しています。

- ・ 筐体（新筐体）
- ・ コントロール／トレース・ボード
- ・ ブレーク・ボード（IE-78330-R-BK）
- ・ エミュレーション・ボード（IE-78327-R-EM）

ブレーク・ボードおよびエミュレーション・ボードを別売のボードに交換することで、他の78Kシリーズのインサーキット・エミュレータにシステム・アップが可能です。

備考 システム・アップとは、インサーキット・エミュレータのエミュレーション・ボードおよびブレーク・ボードを交換して、他の対象デバイスのエミュレーションが行えるようにシステムを変更することを意味します。

1.4 他機種からIE-78327-Rへのシステム・アップ

すでに75Xシリーズ、または78Kシリーズ用の一部のインサーキット・エミュレータをお持ちの場合、エミュレーション・ボードおよびブレーク・ボードを、IE-78327-R用のボードに交換することで、IE-78327-Rと同等の機能を持つエミュレータへのシステム・アップが可能です。

表1-1に、システム・アップ可能な機種、およびシステム・アップに必要なボードを示します。

お手持ちのインサーキット・エミュレータは、製品によって新筐体を使用したものと旧筐体を使用したものの2つのタイプに分けられます。このうちシステム・アップが可能なものは新筐体を使用したタイプのインサーキット・エミュレータのみです。

ボードの交換により、他機能からIE-78327-Rへシステム・アップした場合、インサーキット・エミュレータのコントロール・プログラムは、IE-78327-R用のコントロール・プログラムが必要です。

表1-2 他機種からIE-78327-Rへのシステム・アップ

お手持ちの インサーキット・ エミュレータ	筐 体	ご購入が必要なボード	
		IE-78327-R-EM	IE-78330-R-BK
IE-78112-R	旧筐体	×	×
IE-78210-R ^{注1}			
IE-78220-R ^{注1}			
IE-78310-R ^{注1}			
IE-78310A-R			
IE-75000-R	新筐体	○	○
IE-78000-R ^{注2}			
IE-78130-R			
IE-78230-R			
IE-78240-R			-
IE-78320-R ^{注1}			
IE-78330-R			○
IE-78350-R ^{注2}			
IE-78600-R ^{注2}			

- 注1. 保守製品
2. 開発中

備考1. IE-78327-R自身は、新筐体を使用しています。

2. ○………必要なボード
-………不要
×………システム・アップ不可

本来のIE-78327-Rと、他のエミュレータでIE-78327-R-EMなどを使用してシステム・アップしたインサーキット・エミュレータとの間には、使用方法での違いはありません。

(2) PC-9800シリーズの対象OS

MS-DOS ver. 2.11 / 3.10 / 3.30 / 3.30A

(3) コントロール・プログラム供給媒体

5インチFD (2HD)

3.5インチFD (2HD)

注意 8インチFD(2D)による供給は廃止致しました。
なお、現在8インチFDをご使用のユーザへは、バージョン・アップの際に
5インチFDで供給致します。ご了承ください。

1.5.2 IBM PCシリーズの場合

(1) IBM PCシリーズの対象機種

IBM PC/AT、PC/XT

(2) IBM PCシリーズの対象OS

PC DOS ver. 3.10

(3) コントロール・プログラム供給媒体

5インチFD (2HC)

第2章 IE-78327-Rの起動方法

この章では、IE-78327-Rを起動させる一連の手順について説明します。

IE-78327-Rを起動する場合、ハードウェア編 (EEU-713) の説明にしたがって使用するすべてのハードウェアの設定を完了させておく必要があります。

なお、以降の説明の中で入力しなければならない文字は大文字でも小文字でもかまいません (説明の中では大文字で記述してあります)。

また、許されていない入力を行った場合は、再度同じメッセージを出力して、入力待ちとなります。

凡 例

- _____ : 下線部分はキーボードからの入力を表します。
- ☐ : リターン・キー (CR (ODH)) の入力を表します。
- このマニュアルで用いられている画面表示例、入力例はPC-9800シリーズをホスト・マシンとして使用した場合のものです。

2.1 ホスト・マシンの設定

(1) デバイス・ドライバの確認

MS-DOS (PC-9800シリーズを使用する場合) またはPC DOS (IBM PCシリーズを使用する場合) を起動し、RS-232-C用のデバイス・ドライバおよびプリンタ用デバイス・ドライバがインストールされていることを確認してください。

インストールされていない場合は、CONFIG.SYSに次に示す形式の一文を加え、デバイス・ドライバのインストールを行ってください。

DEVICE=パス名デバイス・ドライバ名

例 MS-DOS使用の場合は、次の二文をCONFIG.SYSに加えます(ただし、RSDRV.SYSとPRINT.SYSは、起動ドライブのルート・ディレクトリにあることを前提とします)。

DEVICE=RSDRV.SYS
DEVICE=PRINT.SYS

注意 デバイス・ドライバのインストールを行ったあとは、必ずホスト・マシンをリセットして、MS-DOSまたは、PC DOSを再起動してください。

表2-1 ホスト・マシンとデバイス・ドライバの対応

ホスト・マシン デバイス・ドライバ	PC-9800シリーズ	IBM PCシリーズ
RS-232-C用 デバイス・ドライバ	RSDRV.SYS	使用するシリアルI/Fに 対応しているドライバ
プリンタ用 デバイス・ドライバ	PRINT.SYS	PRINT.SYS

なお、高速ダウン・ロード・モードを使用しない場合はプリンタ用デバイス・ドライバは特に必要ありません。ただし、IE-78327-Rのコントロール・プログラムでは必要とされないだけで、他のプログラムで必要な場合はインストールしなければなりません。

MS-DOS、またはPC DOSのバージョンによってはデバイス・ドライバのインストールが不要なものもあります（古いバージョンのもの）。

お手持ちのMS-DOS、PC DOSのマニュアルにてご確認ください。

(2) RS-232-Cの初期設定

PC-9800シリーズをホスト・マシンとする場合、SPEEDコマンドを実行してRS-232-Cの初期設定を行う必要があります。

IBM PCシリーズをホスト・マシンとする場合、MODEコマンドでCOM1の初期設定を行う必要があります。

設定する内容は次のとおりです。詳細な設定方法は、MS-DOS、PC DOSのマニュアルでご確認ください。

表2-2 RS-232C-0の初期設定

設定項目	設定内容
ボー・レート	9600bps
キャラクタ長	8ビット
パリティ・チェック	パリティ無
ストップ・ビット数	2ビット
Xパラメータ	有/無どちらでもよい

2.2 コントロール・プログラムの起動

コントロール・プログラムは次に示すファイルで構成されています。
これらのファイルがすべてカレント・ドライブのカレント・ディレクトリに存在していなければなりません。

- ・ IE78327.EXE
- ・ IE78327.HLP

コントロール・プログラムの起動は、MS-DOSまたはPC-DOSのコマンド・ライン上で **IE78327** と入力して行います。

なお、このときにはIE-78327-Rの電源が投入されていなければなりません。

正常に起動した場合、次に示すメッセージが出力されます。

・ 動作環境設定時の画面表示例

```
A>IE78327   
IE-78327 Controller (PC-9801 SERIES) Vx.x [Dd Mmm Yy]  
Copyright (C) 1990 by NEC corporation  
Power on target system (Y/N) Y   
Start on uPD78328/327 emulation system
```

備考 メッセージの出力例は、 μ PD78327シリーズ開発モードでPC-9800シリーズをホスト・マシンとして使用した場合のものです。
IBM PCシリーズをホスト・マシンとした場合では、メッセージ中のPC-9801 SERIESの部分がIBM PC SERIESと表示されます。
また、 μ PD78320シリーズ開発モードとした場合では、メッセージ中のuPD78328/327の部分がuPD78322/320と表示されます。

起動メッセージが表示されずに No Connect と出力された場合は、IE-78327-R は正常に起動できていません。

次に Abort (Y or N) と聞いてきますので Y を入力してください。

- ・ 起動メッセージが表示されなかった場合

A>IE78327

No Connect

Abort (Y or N) : Y

IE-78327-Rを正常に起動できなかった原因として、次のことが考えられます。

- ① IE-78327-Rの電源が投入されていない
- ② IE-78327-Rとホスト・マシンの接続が正常でない
- ③ デバイス・ドライバがインストールされていない
- ④ RS-232-Cの初期設定を行っていない
(SPPEDコマンドを実行していない)

以上の点について確認し、該当する項目があった場合は修正して、再度コントロール・プログラムを起動してください。

なお、どうしてもIE-78327-Rが起動できない場合は、本製品を購入された最寄りの特約店、販売員にご相談ください。

2.3 起動後の操作

IE-78327-Rが正常に起動したあとに行う操作について説明します。

(1) ターゲット・システムの電源投入の要求と確認

正常にコントロール・プログラムが起動すると、ターゲット・システムの電源の投入要求メッセージを出力します。

ターゲット・システムの電源投入後に Y と入力してください。

また、ターゲットを接続していない場合も Y と入力してください。

- ・ 電源投入の要求メッセージ

Power on target system (Y/N) Y

注意 電源投入の要求メッセージへの入力において、N と入力すると、同じメッセージを出力し続け、次のステップに進みません。

(2) 動作環境の設定

2.4 動作環境の設定 で示す操作を1度行くと、次回からは自動的に前回起動時と同じ環境設定となります。

設定内容の変更が必要かどうかを確認してきますので、前回起動時の設定内容を変更する必要がなければ N を入力してください。

設定を変更する場合は Y と入力して、再度設定を行ってください。

- ・ 動作環境設定要求のメッセージ

Create new set up mode (Y or N) Y

2.4 動作環境の設定

(1) ROMサイズの設定

ROMサイズの設定は、ターゲット・システム上のEA端子のレベルに応じて次の二とおりに分かります。

ターゲット・システムを接続していない場合は、(a) EA端子がハイ・レベルの場合と同じになります。

(a) EA端子がハイ・レベルの場合

次のメッセージが出力されますので、対象デバイスのROMサイズに合わせて設定値を入力します。

- ・ ROMサイズの設定メッセージ

Internal ROM size (8K, 16K, 24K, 32K, 40K, 48K, 56K) =

表2-3 ROMサイズの設定値

対象デバイス	設定値	対象デバイス	設定値
μPD78320	ROMレス	μPD78P324	32K
μPD78322	16K	μPD78327	ROMレス
μPD78P322		μPD78328	16K
μPD78323	ROMレス	μPD78P328	
μPD78324	32K		

備考 μPD78323、78324、78P324は開発中です。

- ・ 入力例 (μPD78322の場合)

Internal ROM size (8K, 16K, 24K, 32K, 40K, 48K, 56K) = 16K

(b) EA端子がロウ・レベルの場合 (μPD78320、78327の場合)

次のメッセージが出力され、自動的にROMサイズは0に設定されます。

- ・ ROMサイズの設定メッセージ

Internal ROM size = 0K

(2) RAMサイズの設定

次のメッセージが出力されますので、対象デバイスのRAMサイズの設定値を入力します。

- RAMサイズを選択メッセージ

Internal RAM size (256, 384, 512, 640, 768, 896, 1024, 1152, 1280) =

表2-4 RAMサイズの設定値

対象デバイス	設定値	対象デバイス	設定値
μPD78320	640	μPD78P324	1024
μPD78322		μPD78327	512
μPD78P322		μPD78328	
μPD78323	1024	μPD78P328	
μPD78324			

備考 μPD78323、78324、78P324は開発中です。

μPD78320、78322、78P322の場合、次のように入力してください。

Internal RAM size (256, 384, 512, 640, 768, 896, 1024, 1152, 1280) = 640

(3) μPD78320、78327+TAMエミュレーションの設定

次のメッセージが出力されますので、μPD78320、78327+TAMを用いて、μPD78322、78328のエミュレーションを行う場合（ターゲット・システム上でEA端子にASTB信号が入力されている場合）は Y を入力します。

- μPD78320、78327+TAMエミュレーションの設定メッセージ

uPD78327 + TAM emulation (Y/N)

次のように入力してください（EA端子にASTB信号が入力されている場合）

uPD78327 + TAM emulation (Y/N) Y

(4) 高速ダウン・ロード・モード使用の設定

次のメッセージが出力されますので、高速ダウン・ロード・モードを使用する場合は Y を、使用しない場合は N を入力してください。





- ・ ダウン・ロード・モードの選択メッセージ

Do you use high speed down load mode? (Y/N) = N

第3章 機能概要

この章では、I E - 7 8 3 2 7 - Rの動作環境の設定、コマンドの機能の概要について説明します。

凡例

- : 下線部分はキーボードからの入力を表します。
-  : リターン・キー (C R (O D H)) の入力を表します。
- <ESC> : エスケープ・キーの入力を表します。
-  : コントロール・キーを押しながら  の右側の文字を入力することを表します。
-  : カーソルを表します。
- このマニュアルで用いられている画面表示例、入力例はP C - 9 8 0 0 シリーズをホスト・マシンとして使用した場合のものです。

3.1 対象デバイスの環境設定

IE-78327-Rとホスト・マシン（PC-9800シリーズ、または、IBM PCシリーズ）を接続したときの対象デバイスの環境設定について説明します。

- (1) 動作環境の設定
- (2) クロック選択機能
- (3) マッピング

3.1.1 動作環境の設定

ターゲット・システムの電源投入後、次のメッセージが表示されます。

- ・ **Create new set up mode (Y or N) :**

これに対して **Y** を入力して動作環境の設定を行います。

また、設定した動作環境はセットアップ・ファイル (SETUP.STR) に記録されます。同じ設定で開発を続ける場合には、次回からは上記メッセージに対して **N** と入力すると、セットアップ・ファイルから自動的に動作環境が設定されます。

環境設定には、次の4つの項目があります。

- (a) ROMサイズの設定
- (b) RAMサイズの設定
- (c) μ PD78320、78327 + TAMエミュレーションの選択
- (d) ダウン・ロード・モードの選択

動作環境設定時の画面表示例を示します。

● μ PD78327、78328、78P328の開発（画面表示例）

```

A>IE78327 
IE-78327 Controller (PC-9801 SERIES) Vx.x [Dd Mmm Yy]
Copyright (C) 1990 by NEC corporation

(1) Power on target system (Y/N) Y 
      Start on uPD78328/327 emulation system注
(2) Create new set up mode (Y or N) : Y 
(3) Internal ROM size ( 8K, 16K, 24K, 32K, 40K, 48K, 56K ) = 16K 
(4) Internal RAM size ( 256, 384, 512, 640, 768, 896, 1024, 1152,
      1280 ) = 512 
      Tracer initialize
      Breaker initialize
(5) uPD78327 + TAM emulation (Y/N) N 
(6) Do you use high speed down load mode ? (Y/N) = Y 
brk:0>■
    
```

- (1) ターゲット・システムの電源投入の要求と確認
- (2) 動作環境の設定
- (3) ROMサイズの設定
- (4) RAMサイズの設定
- (5) μ PD78327 + TAMエミュレーションの選択
- (6) ダウン・ロード・モードの選択

注 μ PD78320、78322、78P322を開発する場合は、次のように表示されます。

Start on uPD78328/327 emulation system■

なお、開発モードの変更方法はハードウェア編を参照してください。

(1) ターゲット・システムの電源投入の要求と確認

正常にコントロール・プログラムを起動すると、ターゲット・システムの電源の投入要求メッセージを出力します。

ターゲット・システムの電源投入後に Y と入力します。

また、ターゲットを接続していない場合も Y と入力してください。

- ・ 電源投入の要求メッセージ

Power on target system (Y/N) Y

(2) 動作環境の設定

システム起動時に、(3) - (6) で示す環境設定の操作を一度行くと、二度目以降で設定内容の変更が不要な場合は自動的に設定を行うことができます。

設定内容の変更が必要かどうかを確認してきますので、前回起動の設定内容を変更する必要がなければ N を入力してください。

設定を変更する場合は Y と入力してください。

- ・ 動作環境設定要求のメッセージ

Create new setup mode (Y/N) Y

(3) ROMサイズの設定

ROMサイズの設定は、ターゲット・システム上の結線によって次の二通りに分かります。

ターゲット・システムを接続していない場合は、(a) $\overline{\text{EA}}$ 端子がハイ・レベルの場合と同じになります。

(a) $\overline{\text{EA}}$ 端子がハイ・レベルの場合

次のメッセージが出力されますので、 $\mu\text{PD}78322$ 、 78328 のエミュレーションを行う場合は、16K と入力します。

- ・ ROMサイズの設定メッセージ

Internal ROM size (8K, 16K, 24K, 32K, 40K, 48K, 56K) = 16K

(b) \overline{EA} 端子がロウ・レベルの場合

次のメッセージが出力され、自動的にROMサイズは0に設定されます。

・ROMサイズの設定メッセージ

Internal ROM size = 0K

(4) RAMサイズの設定

次のメッセージが出力されますので、 μ PD78320、78322、78P322のエミュレーションを行う場合は、640 [Q] と入力します。

・RAMサイズの設定メッセージ

Internal RAM size (256, 384, 512, 640, 768, 896, 1024, 1152, 1280) = 640 [Q]

μ PD78327、78328、78P328のエミュレーションを行う場合は、512 [Q] と入力します。

・RAMサイズの設定メッセージ

Internal RAM size (256, 384, 512, 640, 768, 896, 1024, 1152, 1280) = 512 [Q]

(5) μ PD78320、78327+TAMエミュレーションの選択

μ PD78320、78327+TAMを用いて μ PD78322、78328のエミュレーションを行う場合（ターゲット・システム上で \overline{EA} 端子にASTB信号が入力されている場合）、次のメッセージでY [Q] を入力します。

・ μ PD78327 + TAM emulation (Y/N) Y [Q]

Y [Q] を入力した場合、対象デバイスのメモリ空間は内部RAM、SFR空間を除いて、すべて強制的にマッピング解除（ノンマッピング）となります。

詳細は μ PD78322、78328、およびTAM（ μ PD71P301）のユーザーズ・マニュアルを参照してください。

(6) ダウン・ロード・モードの選択

次のメッセージが出力されますので、高速ダウン・ロード・モードを使用する場合は Y を、使用しない場合は N と入力してください。

・ ダウン・ロード・モードの選択メッセージ

Do you use high speed down load mode ? (Y/N) = N

3. 1. 2 クロック選択機能 (CLK コマンド)

CLKコマンドで、エミュレーションCPUに供給する動作クロックの指定を行います。CLKコマンドを実行すると、エミュレーションCPUはリセットされます。クロック・ソースは、次のどちらかを選択します。

(1) エミュレータ内クロック (CLK I)

エミュレーションCPUに接続する水晶振動子が16MHzの場合には、エミュレータ内クロック (CLK Iコマンド) を指定します。

(2) ユーザ設定クロック (CLK U)

16MHz以外 (MAX. 16MHz) の動作クロックでデバッグ作業を行う場合には、ユーザ設定クロック (CLK U コマンド) を指定します。

この場合、IE-78327-R本体内部のエミュレーション・ボード上のクロック設置用ソケット (OPCK) に動作クロックの2倍の水晶発振器を設置してください。

IE-78327-Rは、本体外部からクロックを供給することはできません (ターゲット・システムからのクロック供給は不可)。

例 12MHzでエミュレーションCPUを動作させる場合、24MHzの水晶発振器を設置します。

クロック選択の詳細については、ハードウェア編ユーザーズ・マニュアルの 第3章 動作クロックの変更 を参照してください。

3. 1. 3 マッピング機能 (MAP コマンド)

MAP コマンドは、対象デバイスの使用するメモリを指定する機能を持ちます。
MAP コマンドを実行するとエミュレーションCPUはリセットされます。

マッピング機能は、次に示す6種類のメモリ属性を指定します。

- 内部ROM (MAP I コマンド)
- ターボ・アクセス・マネージャ代替メモリ (MAP T コマンド)
- 代替メモリ (MAP W コマンド)
- ライト・プロテクト付き代替メモリ (MAP R コマンド)
- ユーザ・メモリ (MAP U コマンド)
- マッピング解除 (ノンマッピング) (MAP K コマンド)

0H-0DFFFH の56Kバイトの範囲のメモリ領域は、8Kバイト単位で次のメモリ属性を指定します。

- 内部ROM
- ターボ・アクセス・マネージャ代替メモリ
- 代替メモリ
- ライト・プロテクト付き代替メモリ
- ユーザ・メモリ
- マッピング解除

注

0E000H-0FFFFH (内部RAM、SFR領域は除く)の8Kバイトのメモリ領域は、次のメモリ属性を指定します。

- ユーザ・メモリ
- マッピング解除

注 SFR : 特殊機能レジスタ (Special Function Register)

(1) 内部ROM (MAP I コマンド)

内部ROMに指定したメモリ領域は、対象デバイスのROMと同等のメモリ領域となります。

エミュレーションCPUはIE-78327-R内のメモリに対してアクセスします。

エミュレーションCPUが、このメモリ領域に対して書き込み動作を行った場合、**write protect break** を発生します (詳細は、3.2.5 ブレーク機能を参照してください)。

IE-78327-R起動時に設定した内部ROMサイズをMAP I コマンドで変更できます。

通常、内部ROMのマッピング容量は、IE-78327-R起動時に指定したROMサイズになります。

(2) ターボ・アクセス・マネージャ代替メモリ (MAP T コマンド)

ターボ・アクセス・マネージャ代替メモリに指定したメモリ領域は、対象デバイスにTAMを接続した場合と同等のメモリ領域となります。

エミュレーションCPUは、IE-78327-R内のメモリに対してアクセスします。

エミュレーションCPUが、このメモリ領域に対して書き込み動作を行った場合、**write protect break** を発生します (詳細は 3.2.5 ブレーク機能を参照してください)。

(3) 代替メモリ (MAP W コマンド)

代替メモリに指定したメモリ領域は、対象デバイスに通常メモリ (SRAM、DRAMなど) を接続した場合と同等のメモリ領域となります。

エミュレーションCPUは、IE-78327-R内のメモリに対してアクセスします。

エミュレーションCPUがこのメモリ領域に対してターボ・アクセス (高速バースト・プログラム・フェッチ) を行った場合、Turbo access break を発生します (詳細は、3. 2. 5 ブレーク機能 を参照してください)。

(4) ライト・プロテクト付き代替メモリ (MAP R コマンド)

ライト・プロテクト付き代替メモリに指定したメモリ領域は、対象デバイスにROMを接続した場合と同等のメモリ領域となります。

エミュレーションCPUは、IE-78327-R内のメモリに対してアクセスします。

エミュレーションCPUがこのメモリ領域に対してアクセスを行った場合、以下のブレークが発生します (詳細は、3. 2. 5 ブレーク機能 を参照してください)。

- ・ Turbo access break : ターボ・アクセス動作を行った場合
- ・ Write protect break : 書き込み動作を行った場合

(5) ユーザ・メモリ (MAP U コマンド)

ユーザ・メモリに指定したメモリ領域は、ターゲット・システム上のメモリにアクセスする領域となります。

エミュレーションCPUは、ターゲット・システム上のメモリに対してアクセスします。

エミュレーションCPUがこのメモリ領域に対してターボ・アクセス (高速バースト・プログラム・フェッチ) を行った場合、Turbo access break を発生します (詳細は、3. 2. 5 ブレーク機能 を参照してください)。

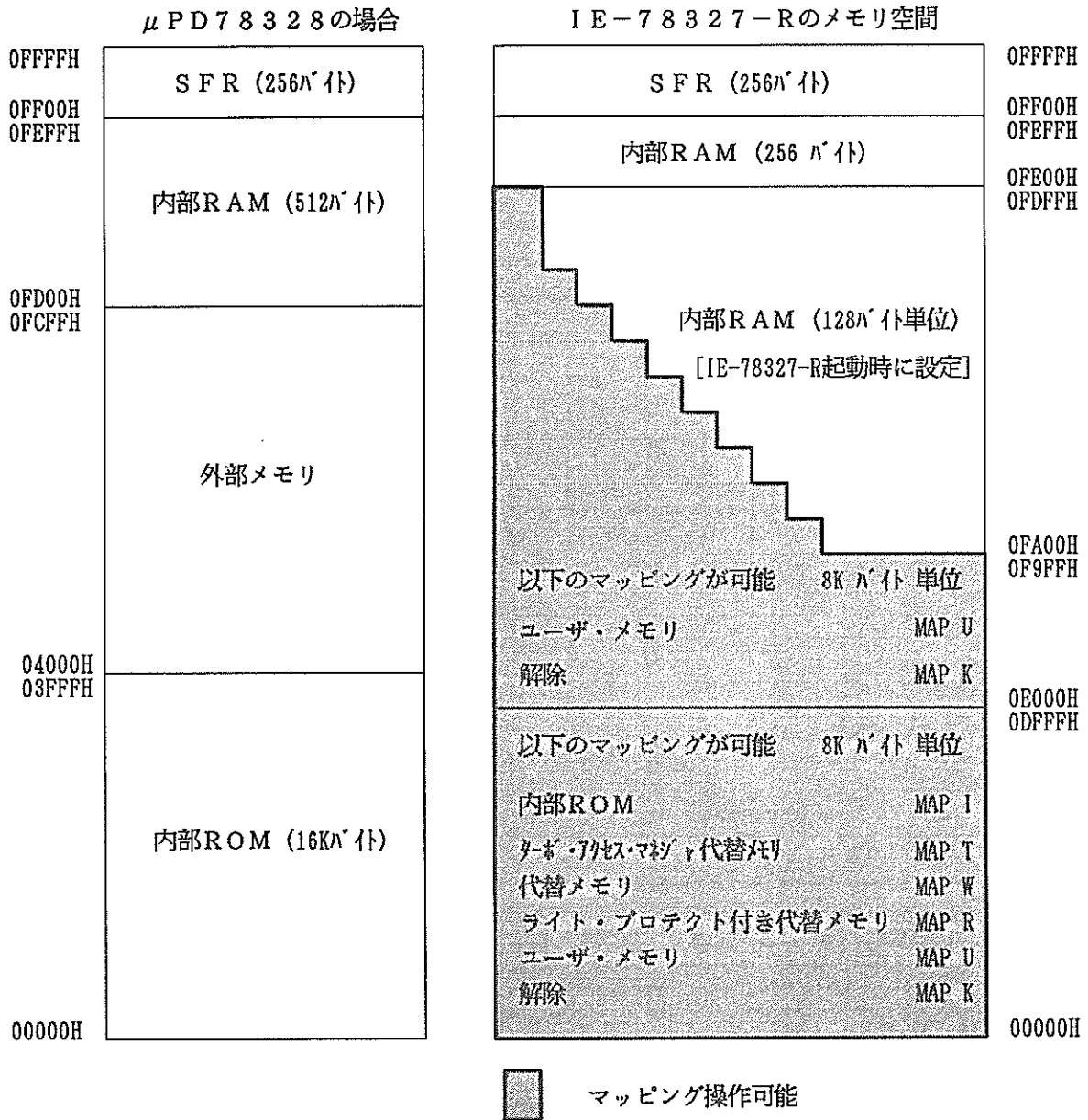
(6) マッピング解除 (MAP K コマンド)

マッピング解除に指定したメモリ領域は、メモリが割り付けられていない領域です。

エミュレーションCPUがこのメモリ領域に対してアクセスを行った場合、**Non map access break**が発生します(詳細は、3.2.5 ブレーク機能 を参照してください)。

- 対象デバイスと、IE-78327-Rのメモリ空間は次のような図で表されます。

図3-1 対象デバイスとIE-78327-Rのメモリ空間



3.2 エミュレーション

IE-78327-Rは、エミュレーションとして次の6つの機能を持っています。

- (1) リセット機能
- (2) メモリ操作機能
- (3) レジスタ操作機能
- (4) 実行機能
- (5) ブレーク機能
- (6) 動作状態表示機能

3.2.1 リセット機能 (RES コマンド)

RES コマンドが持つリセット機能には次の2つがあります。

(1) エミュレーション・デバイスのリセット (RES)

エミュレーション・デバイスをリセットします。
マッピング、メモリ、イベントやアナライザに関する情報は影響を受けません。

(2) IE-78327-R本体のリセット (RES H)

IE-78327-R 起動時の動作環境設定に戻ります。

3. 2. 2 メモリ操作機能 (ASM、DAS、MEM、MOV コマンド)

エミュレーションCPUが操作できるメモリに対して、表示、変更、転送などの操作を行う機能です。

メモリの操作は、次の方法で行うことができます。

- 16進数による変更または表示など (MEM)
- ニモニックによる変更または表示 (ASM、DAS)
- 代替メモリと
ユーザ・メモリ間のデータ転送 (MOV)

メモリの操作は、マッピング機能により指定されたメモリ領域に対して行われます。

3. 2. 3 レジスタ操作機能 (REG、SFR コマンド)

μPD78320、78322、78P322、78327、78328、
78P328の内部レジスタまたはSFRを表示または変更する機能です。

内部レジスタの操作には、REGコマンドを使用します。
特にPSWについては、フラグ単位での表示または変更ができます。

特殊機能レジスタ(SFR)の操作はSFRコマンドを使用します。

3. 2. 4 実行機能 (RUN コマンド)

エミュレーションCPUによるプログラムの実行を開始する機能です。
実行機能は、次の2つに大別されます。

- (1) リアルタイム実行
- (2) ノンリアルタイム実行

(1) リアルタイム実行機能

リアルタイム実行機能には、次の2種類があります。

- ノンブレイク・リアルタイム実行 (RUN N コマンド)
- ブレイク条件付きリアルタイム実行 (RUN B コマンド)

(a) ノンブレイク・リアルタイム実行 (RUN N コマンド)

指定されたアドレスからターゲット・プログラムの実行を開始し、ディレイ条件の成立により、リアルタイム・トレーサと内部RAMデータ・サンブラを停止します (図3-2を参照してください)。

このとき、実行中のターゲット・プログラムは強制ブレイク (フェイル・セーフ・ブレイク: STP コマンドの入力) 以外では停止できません。

図3-2 ノンブ레이크・リアルタイム実行でのエミュレーションCPU、イベント検出器、アナライザ表示プロンプト、および外部トリガ出力の動作状態

発生イベント		実行開始	インテプル条件	トリガ条件	チェックポイント条件	デバッグ条件	バス条件	トリガスタート	アナライザスタート	エミュレーション終了			
ハードウェア		RUN N ↓	ENB ↓	TRX ↓	BRM ↓	CHK ↓	DSB ↓	PAS ↓	DLY ↓	TRG ↓	STP T ↓	STP ↓	
エミュレーションCPU	動作	停止	リアルタイム実行	→	→	→	リアルタイム実行	→	→	→	→	→	停止
	割り込み	保留	受け付け	→	→	→	受け付け	→	→	→	→	→	保留
表示プロンプト		brk:0>	trc:0>	→	→	→	→	→	→	emul:0>	trc:0>	emul:0>	brk:0>
イベント検出器	リアルタイム・トレース	停止	検出	→	→	→	検出	→	停止	→	検出	停止	→
	区間トレース・モード	停止	トレース	→	→	→	トレース	→	→	停止	トレース	停止	→
	チェックポイント・モード	停止	→	→	→	→	→	→	→	→	→	→	→
	チェックポイント・トレース	停止	→	→	→	→	→	→	→	→	→	→	→
内部RAMデータ・メモリ	メモリ・メモリ	停止	リアルタイム・メモリでの書き込み	→	→	→	リアルタイム・メモリでの書き込み	→	→	停止	リアルタイム・メモリでの書き込み	停止	→
	実行時間測定	停止	→	測定	→	→	測定	停止	→	→	→	→	→
外部トリガ出力	実行命令数測定	停止	→	測定	→	→	測定	停止	→	→	→	→	→
	C0カバレッジ	停止	測定	→	→	→	測定	→	→	→	→	→	停止
外部トリガ出力		Low	→	→	→	→	→	→	High	Low	→	→	→

備考 上の図中の→は、左側の動作状態が続くことを表します。

(b) ブレーク条件付きリアルタイム実行 (RUN B コマンド)

指定されたアドレスからターゲット・プログラムの実行を開始し、ディレイ条件の成立によりエミュレーションCPU、リアルタイム・トレーサ、内部RAMデータ・サンプラを停止します(図3-3を参照してください)。

アナライザとエミュレーションCPUの実行を強制的に停止する場合、STPコマンドを入力します。

ディレイ条件の成立により、エミュレーションCPUとアナライザが停止すると自動的に1ステップ実行モードとなります。

この状態でⓐを入力すると、次の命令(1ステップ)が実行されます。

／ⓐを入力すると、プロシージャ実行を行います((b)プロシージャ実行を参照してください)。

1ステップ実行を中断する場合は、〈ESC〉を入力してください。

図3-3 ブレーク条件付きリアルタイム実行でのエミュレーションCPU、イベント検出器、アナライザ表示プロンプト、および外部トリガ出力の動作状態

発生イベント		実行開始 RUN B	ブレーク 条件 ENB	クォリアイ トレス条件 TRX	トリガ 条件 BRM	チェック ポイント 条件 CHK	デイスラップ 条件 DSB	パス 条件 PAS	ディレイ 条件 DLY	再実行 条件 <cr>or</cr>	エミュ 終了 <ESC>
ハードウェア		↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
エミュレーションCPU	動作	停止	リアルタイム実行	→	→	→	リアルタイム実行	→	→	停止	→
	割り込み	保留	受け付け	→	→	→	受け付け	→	→	保留	→
表示プロンプト		brk:0>	trc:0>	→	→	→	→	→	→	なし	→
イベント検出器		停止	検出	→	→	→	検出	→	→	→	→
	リアルタイム トレース	停止	トレース	→	→	→	トレース	→	→	停止	→
アナライザ		停止	→	→	→	→	トレース	停止	→	→	→
	リアルタイム モード	停止	→	→	→	→	トレース	停止	→	→	→
内部RAM データ サンプル		停止	→	→	→	→	→	→	→	→	→
	チェックポイント のトレース	停止	→	→	→	→	停止	→	→	→	→
パフォーマンス 測定		停止	サンプル・タイム ラグでの書き 込み	→	→	→	停止	→	→	停止	→
	実行時間測定	停止	→	測定	→	→	測定	停止	→	→	→
C0カバレージ		停止	→	測定	→	→	測定	停止	→	→	→
	実行命令数測定	停止	→	測定	→	→	測定	停止	→	→	→
外部トリガ出力		停止	測定	→	→	→	測定	→	→	停止	→
	Low	Low	→	→	→	→	→	→	High	Low	→

備考1. 上の図中の→は、左側の動作状態が続くことを表します。
 2. <cr> はリターン・キーの入力を示します。

(2) ノンリアルタイム実行機能

ノンリアルタイム実行機能には、次の2種類があります。

- ステップ実行 (RUN T コマンド)
- プロシージャ実行 (RUN T PRC付きコマンド)

(a) ステップ実行 (RUN T コマンド)

指定されたアドレスからターゲット・プログラムを1命令実行して実行トレースを行います。

ステップ実行は、指定されたレジスタ条件の一致、指定命令数の実行完了まで断続的に継続されます。

ステップ実行終了後は、1ステップ実行モードとなります。

この状態で **ⓐ** を入力すると、次の1命令(1ステップ)が実行されます。

/ **ⓐ** を入力すると、プロシージャ実行を行います((b) プロシージャ実行 を参照してください)。

実行を中断する場合は、**<ESC>** を入力してください。

備考 ステップ実行中の割り込みはすべて保留されます。

図3-4 ステップ実行状態での、エミュレーションCPU、イベント検出器、アナライザ、表示プロンプト、および外部トリガ出力の動作状態

発生イベント		コマンド入力 再実行 RUN T ⊙or/⊙			エミュレーション終了 <BSC>		
ハードウェア							
エミュレーション CPU	動作	停止	ステップ 実行	ステップ 実行 プロセッサ実行	→	停止	
	割り込み	保留	→	→	→	→	
表示プロンプト		brk:0>	なし	→	→	brk:0>	
イベント検出器		停止	→	→	→	→	
アナライザ	リアルタイム・ トレサ	全トレース・ モード	停止	トレース	全トレース メイン・ループの トレース	→	停止
		区間トレース・ モード	停止	トレース	全トレース メイン・ループの トレース	→	停止
		クオリファイ・トレース・ モード	停止	トレース	全トレース メイン・ループの トレース	→	停止
		チェック・ポイントの トレース	停止	→	→	→	→
内部RAM データ・ サンプリング	サンプリング・ メモリ	停止	→	→	→	→	
	パフォーマンス 測定						
	実行時間 測定	停止	→	→	→	→	
	実行命令 数測定	停止	→	→	→	→	
C0カバレッジ		停止	測定	→	→	停止	
外部トリガ出力		Low	→	→	→	→	

備考 上の図中の → は、左側の動作状態が続くことを表します

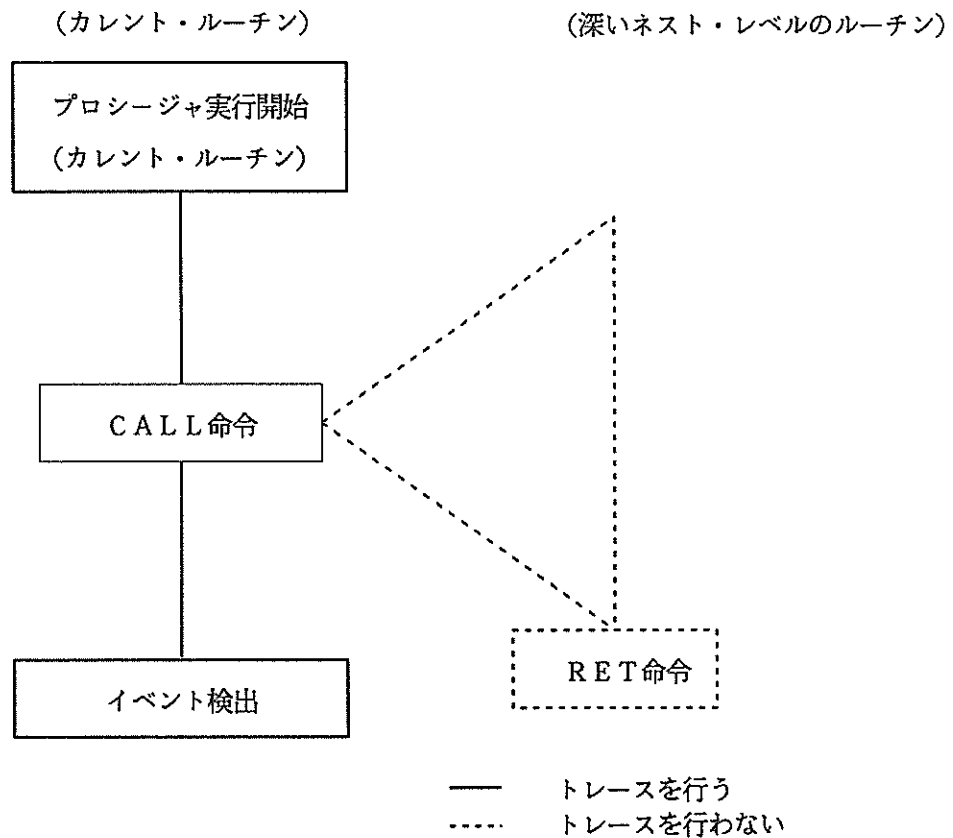
(b) プロシージャ実行 (RUN T PRC付きコマンド)

プロシージャ実行は、カレント・ルーチンとそれより浅いネスト・レベルのルーチン(メイン・ルーチンなど)のディバグを効率的に行うことができる機能です。

指定されたアドレスからターゲット・プログラムを1命令実行し、実行トレースを行います。

このとき、実行を開始したルーチン(カレント・ルーチン)よりも深いネスト・レベルのルーチン(例えば、CALL命令とRET命令で囲まれたルーチン)に対して、ステップ実行は行いますが、トレースは行いません(図3-5 プロシージャ実行とネストとの関係を参照してください)。

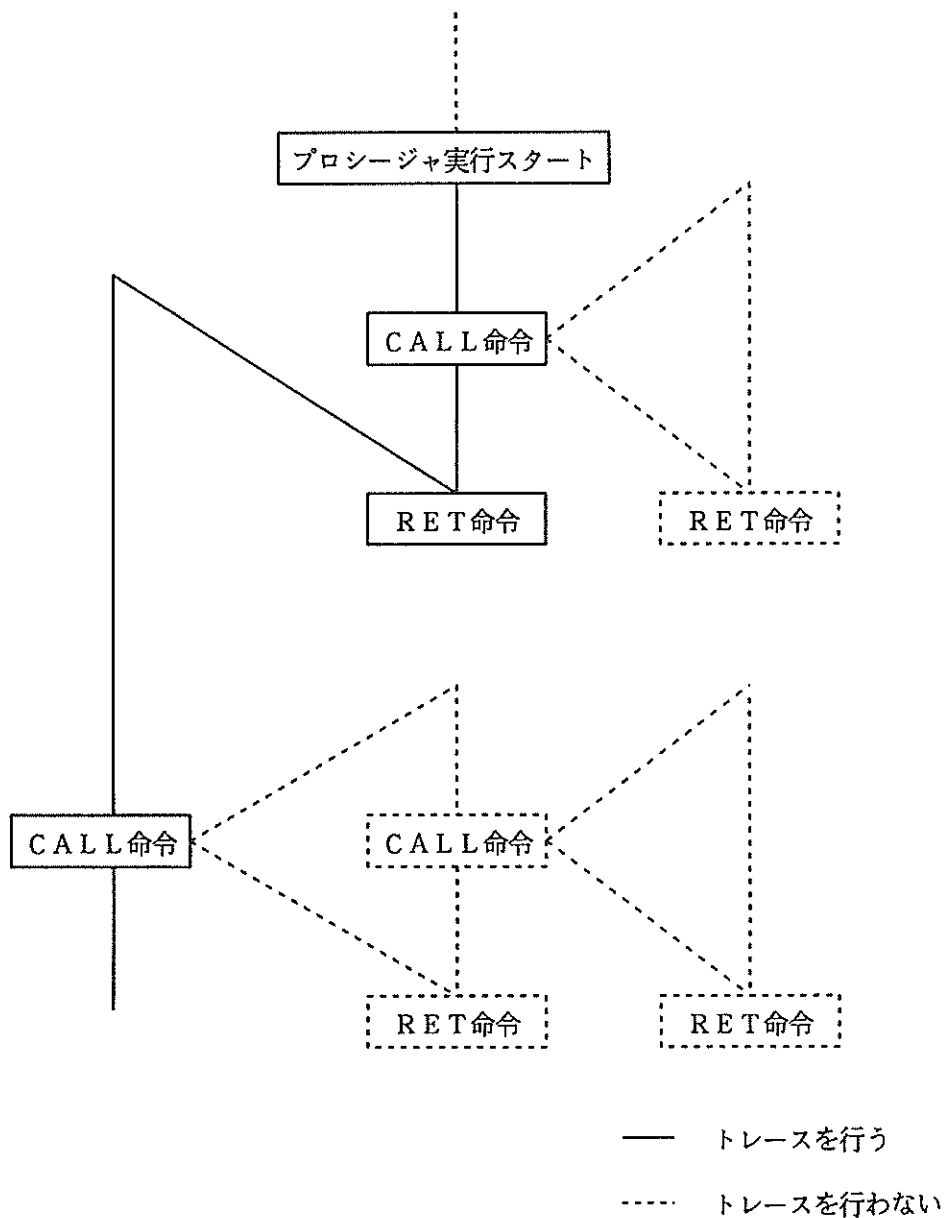
図3-5 プロシージャ実行とネストとの関係



また、実行を開始したルーチン（カレント・ルーチン）よりも浅いネスト・レベルのルーチンに戻ることができます。（図3-6 浅いネスト・レベルに戻る場合のプロシージャ実行 を参照してください）。

図3-6 浅いネスト・レベルに戻る場合のプロシージャ実行

(浅いネスト・レベルのルーチン) (カレント・ルーチン) (深いネスト・レベルのルーチン)



プロシージャ実行は、指定されたレジスタ条件の一致、指定命令数の実行完了まで断続的に継続されます。

プロシージャ実行中はイベントの検出はしません。また、割り込みはすべて保留されます。

プロシージャ実行終了後、自動的に1ステップ実行モードとなります。
この状態で **Ⓚ** を入力すると、次の1命令（1ステップ）を実行します。
/**Ⓚ** を入力すると、プロシージャ実行をします。

実行を中断する場合、**<ESC>** を入力します。

図3-7 プロシージャ実行での、エミュレーションCPU、イベント検出器、アナライザ、表示プロンプト、および外部トリガ出力の動作状態

発生イベント		コマンド入力 再実行 RUN T PRC ⊙or/⊙			エミュレーション終了 <ESC>		
ハードウェア							
エミュレーション CPU	動作	停止	プロシージャ 実行	ステップ実行 プロシージャ実行	→	停止	
	割り込み	保留	→	→	→	→	
表示プロンプト		brk:0>	なし	→	→	brk:0>	
イベント検出器		停止	→	→	→	→	
アナライザ	リアルタイム・ トレサ	全トレース・ モード	停止	メイン・ループ の全トレース	全トレース メイン・ループの トレース	→	停止
		区間トレース・ モード	停止	メイン・ループ の全トレース	全トレース メイン・ループの トレース	→	停止
		クオリファイ・トレース・ モード	停止	メイン・ループ の全トレース	全トレース メイン・ループの トレース	→	停止
		チェックポイントの トレース	停止	→	→	→	→
ザ	内部RAM データ・ サンプラ	サンプラ・ メモリ	停止	→	→	→	→
		パフォーマンス 測定	実行時間 測定	停止	→	→	→
	実行命令 数測定		停止	→	→	→	→
CO加レッジ		停止	測定	→	→	停止	
外部トリガ出力		Low	→	→	→	→	

備考 上の図中の → は、左側の動作状態が続くことを表します。

3.2.5 ブレーク機能 (STP、RES コマンド、ESCキー)

STP、RESコマンド、ESCキーは、エミュレーションCPUによるプログラムの実行を停止する機能を持ちます。

ブレーク機能には、次の4種類があります。

- ハードウェア・コンパレータによるイベント検出ブレーク
- ステップ実行時のレジスタ値によるブレーク
- フェイル・セーフ・ブレーク
- マニュアル・ブレーク

これらの各ブレーク機能と実行機能の関係を次に示します。

表3-1 ブレーク機能と実行機能の関係

実行機能 ブレーク機能	ノンブレーク・ リアルタイム実行 (RUN N)	ブレーク条件付 きリアルタイム実行 (RUN B)	ステップ実行 (RUN T)	プロシージャ 実行 (RUN T PRC付き)
イベント検出 ブレーク	×	○	×	×
レジスタ値による ブレーク	×	×	○	○
フェイル・セーフ・ ブレーク	○	○	○	○
マニュアル・ ブレーク	○	○	○	○

備考 ○ は有効、× は無効を表します。

(a) イベント検出ブレーク

イベント検出ブレークは、コマンド (RUN B) 実行中のみ有効になります。

イベント要因には次の3種類があります。

- ・ バス・イベント検出 (BRA1 - BRA4)
- ・ 外部データ検出 (BRD)
- ・ プログラム実行検出 (BRS1 - BRS4)

イベント検出によるエミュレーションCPUの実行停止時には、表3-2に示すメッセージが表示されます。

表3-2 イベント検出ブレークに対する表示メッセージ

発生要因	表示メッセージ
バス・イベント検出	Bus detection break terminated (terminatedのあとにイベント要因になった検出器を表示します)。
外部データ検出	External break terminated
プログラム実行検出	Execution break terminated (terminatedのあとにイベント要因になった検出器を表示します)。

(b) ステップ実行時のレジスタ値によるブレーク

ブレーク条件は、オペランドで次の条件を指定できます。

- 内部レジスタ条件の一致、指定命令数の実行完了 (RUN T)

コマンド (RUN T) の中にオペランドを用いて設定します。

このコマンド以外ではレジスタ値によるブレーク指定はできません。

コマンド行によるエミュレーションCPUの実行停止時には、次のメッセージが表示されます。

- **terminated**

(c) フェイル・セーフ・ブレイク

エミュレーションCPUが異常動作をした場合、強制的に発生するブレイクです。フェイル・セーフ・ブレイクは、すべての実行機能に対し有効となります。

フェイル・セーフ・ブレイクは、次のように構成されています。



注 マニュアル・ブレイクは、フェイル・セーフ・ブレイクの機能の一つですが、機械が強制的に行うのではなく、実際の操作は人間が行うことから項目を分けて項目(d)で説明します。

フェイル・セーフ・ブレイクによるエミュレーションCPUの実行停止時には、表3-3のメッセージが表示されます。

表3-3 フェイル・セーフ・ブレイクに対する表示メッセージ

発生要因	表示メッセージ
ノンマップ・エリア	Non map area break terminated
ライト・プロテクト	Write protect break terminated
ターボ・アクセス	Turbo access break terminated
マニュアル	Escape break terminated

● ノンマップ・エリア・ブレーク

マッピングされていないメモリ・エリアに対してアクセスを行った場合に発生します。

● ライト・プロテクト・ブレーク

次のメモリ領域に対し、ライト・アクセスを行った場合に発生します。

- ・ 内部ROM (MAP I)
- ・ ターボ・アクセス・マネージャ代替メモリ (MAP T)
- ・ ライト・プロテクト付き代替メモリ (MAP R)

● ターボ・アクセス・ブレーク

次のメモリ領域に対して、ターボ・アクセスを行った場合に発生します。

- ・ ライト・プロテクト付き代替メモリ (MAP R)
- ・ 代替メモリ (MAP W)
- ・ ユーザ・メモリ (MAP U)

ターボ・アクセス動作は、次の領域に対してのみ可能となります。

- ・ 内部ROM (MAP I)
- ・ ターボ・アクセス・マネージャ代替メモリ (MAP T)
- ・ 内部RAM (0FE00H-0FFFFH は除く)

(d) マニュアル・ブレーク

キー入力による強制ブレークです。すべての実行機能に対して有効です。
マニュアル・ブレークには、次の3種類があります。

● 実行停止コマンドによる強制ブレーク (STP)

実行停止コマンド (STP) を入力すると、エミュレーションCPUとアナライザの動作を停止します。

STPコマンドによる強制ブレークは、リアルタイム実行時にのみ有効となります。
内部RAM、SFRは、ブレーク直前の状態で保持されます。

● リセット・コマンドによる強制ブレーク (RES)

リセット・コマンド (RES) を入力すると、エミュレーション・デバイスとアナライザの動作が停止します。

RESコマンドによる強制ブレークは、リアルタイム実行時にのみ有効となります。
エミュレーション・デバイスは、プログラムの実行を停止すると同時にリセットされます。

● ESCキーによる強制ブレーク

<ESC> の入力で、エミュレーション・デバイスとアナライザの動作が停止します。
<ESC> 入力による強制ブレークは、ノンリアルタイム実行時にのみ有効となります。

内部RAM、SFRは、ブレーク直前の状態で保持されます。

3.2.6 動作状態表示機能

エミュレーションCPUとアナライザの動作状態に応じて、次の3種類のプロンプトを表示します。

- trc:0>
- emu:0>
- brk:0>

各プロンプトが出力されているときのエミュレーションCPUとアナライザの動作状態は、表3-4のようになります。

表3-4 エミュレーションCPU、アナライザの動作とプロンプトの関係

表示プロンプト	エミュレーションCPUの動作	アナライザの動作	備考
trc:0>	動作中	動作中	tracing
emu:0>	動作中	停止	emulating
brk:0>	停止 (ブレーク)	停止	break

これらのプロンプトが表示されているときのみ、コマンド入力が可能となります。

3.3 イベント検出機能 (BRA、BRD、BRS、DSB、ENB、CHK、TRX、DLY コマンド)

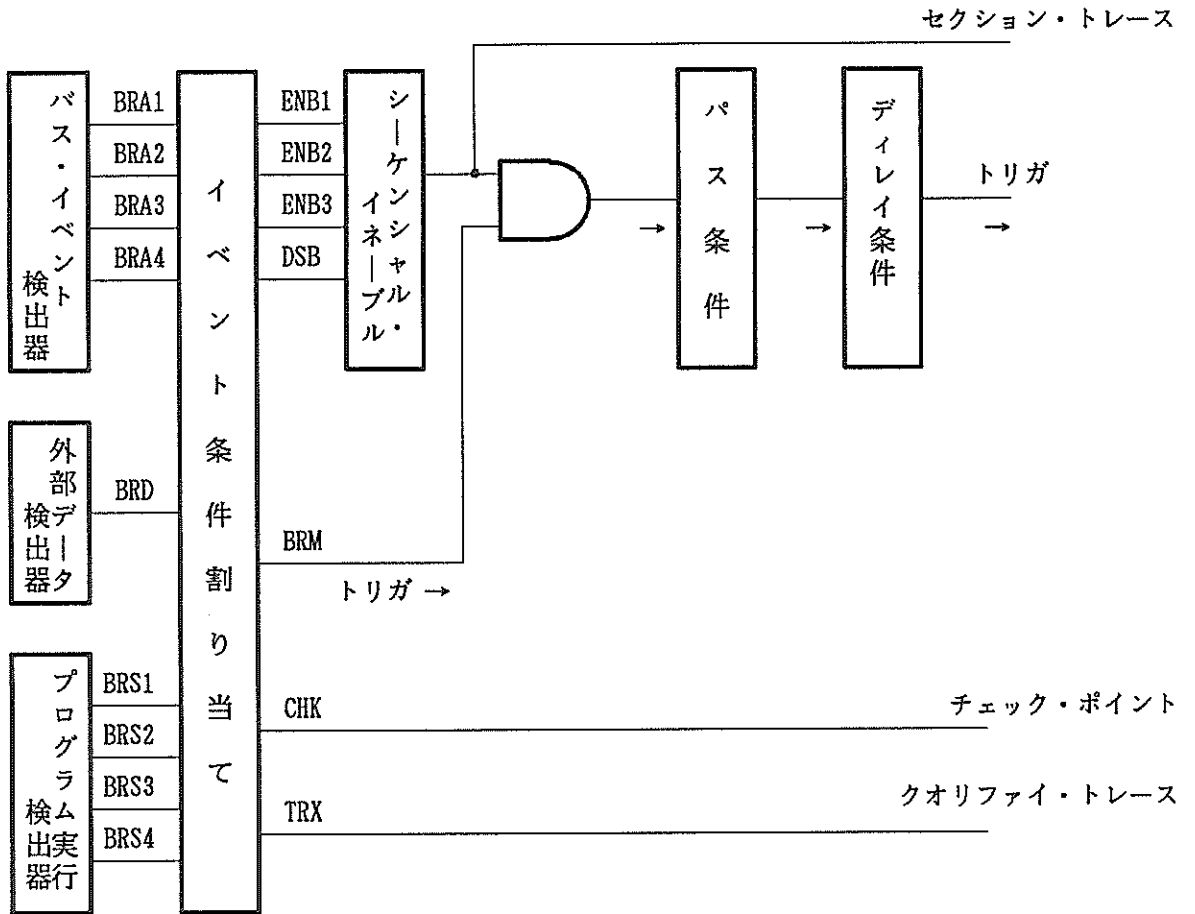
エミュレーションCPUの実行状態を常に監視し、あらかじめ設定された指定条件により、トリガ信号を出力し、エミュレーションCPUの実行、アナライザの動作を停止する機能です。

イベント検出機能は、エミュレーションCPUがリアルタイム実行時のみ有効となります。

イベント検出機能は、次のような機能を持っています。

- | | |
|----------------|---------------------------------|
| (1) バス・イベント検出 | (BRA) |
| (2) 外部データ検出 | (BRD) |
| (3) プログラム実行検出 | (BRS) |
| (4) イベント条件割り当て | (BRM、ENB1-ENB3、
DSB、CHK、TRX) |
| (5) バス条件 | (PAS) |
| (6) ディレイ条件 | (DLY) |

図3-8 イベント検出部 ブロック図



3. 3. 1 バス・イベント検出 (BRA コマンド)

エミュレーションCPUのバスに接続されたコンパレータで構成される4つのイベント検出器を持っています (BRA1 - BRA4)。

データのリード、ライトおよびプログラム (OPコード) フェッチのバス・サイクルが発生した時点で検出します。

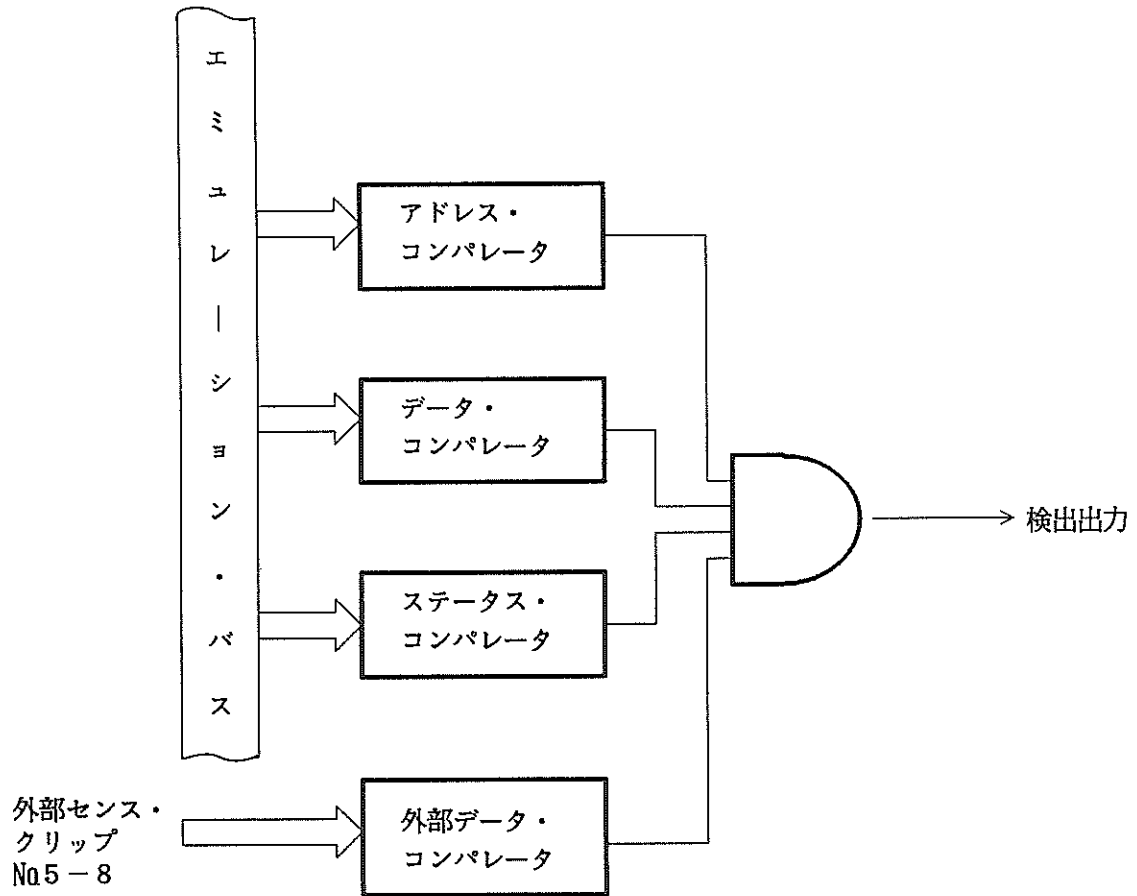
したがって、プログラム・フェッチはプリフェッチをした時点で検出します (プログラム実行時点で検出したい場合はプログラム実行検出BRSコマンドを使用してください)。

バス・イベントの検出条件は、一つのイベント検出器で次の条件を設定できます。

- ・ アドレス2ポイント
 - ・ データ (バイト)
 - ・ 検出ステータス
 - ・ 外部データ (4ビット)
- 
- AND条件となります

備考 アドレス、データ、外部データはマスク指定できます。

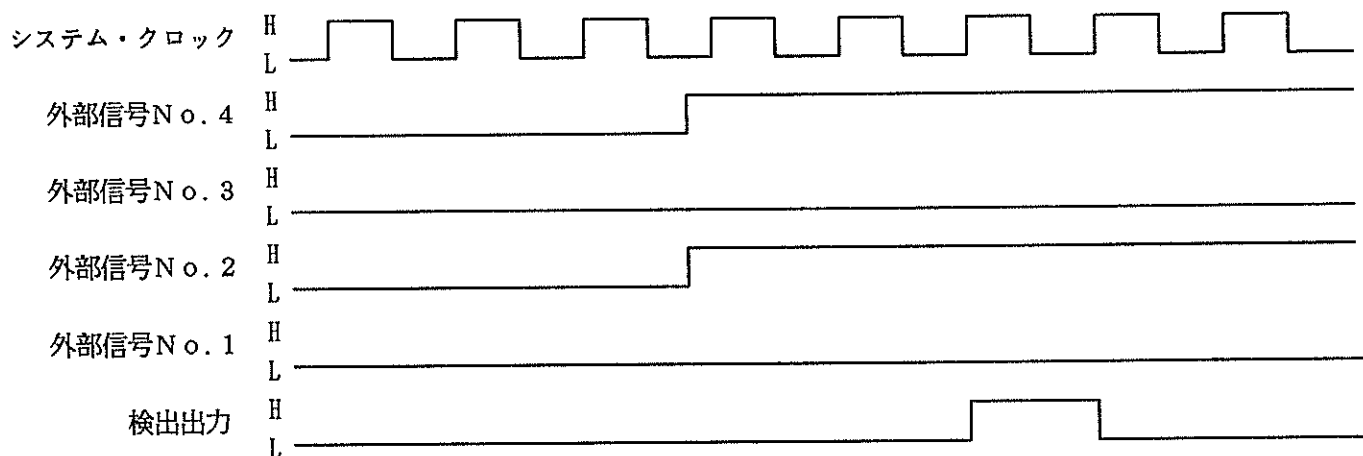
図3-9 バス・イベント検出器のブロック図



3.3.2 外部データ検出 (BRD コマンド)

外部センス・クリップNo. 1-4に入力された信号を検出します。検出のタイミングは、次のようになっています。

図3-10 外部データ1010_v (0AH)を検出する場合



検出は、外部データが目的の値に2システム・クロック以上の間保持された場合に出力されます(エッジ検出)。

検出データはマスク指定できます。

3.3.3 プログラム実行検出 (BRS コマンド)

プログラム実行を検出する4つのイベント検出器を持っています (BRS 1 - BRS 4)。プログラム実行検出は、一つのイベント検出器で次の条件を設定できます。

- ・アドレス1ポイント

● チェック・ポイント (CHK コマンド)

チェック・ポイントに割り当てられたイベントを検出すると、エミュレータはその時点で一時的にブレークします。そして、あらかじめ設定された条件にしたがって、レジスタ、メモリ、SFRの値をトレース・メモリに書き込んだのち、ふたたびターゲット・プログラムの実行を開始します。

このデータは、トレース・データ表示時に、チェック・ポイントとして表示されます（このような機能をスナップ・ショット・ダンプといいます）。

● クオリファイ・トレース (TRX コマンド)

トレース・クオリファイに割り当てられたイベントを検出すると、そのバス・サイクルのみをトレースします。このため、正常にトレース・クオリファイを行えるイベント検出器は、バス・イベント検出 (BRA1 - BRA4) のみです。

3.3.5 バス条件 (PAS コマンド)

イベントの発生回数をカウントします。あらかじめ設定された数をカウントしたら、ディレイ・カウンタをイネーブルにし、ディレイ・カウントを始めます。ディレイ条件が DLY L で設定されている場合は、アナライザはすぐに停止します (DLY L は 3.3.6 ディレイ条件 を参照してください)。

シーケンシャル・イネーブルを設定している場合は、イネーブル状態の時のイベントをカウントします。

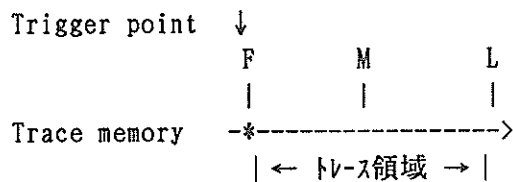
3.3.6 ディレイ条件 (DLY コマンド)

あらかじめ設定したトリガ・ポイントに対するリアルタイム・トレース、内部データ・サンプラのトレース領域を設定するコマンドです。トリガ・ポイント通過後のトレースをどこまで継続するかを選択します。

実際には、トリガ・ポイントをトレース・メモリ、およびサンプル・メモリの最初、中央、最後のいずれかに設定して、トレース領域を選択します。

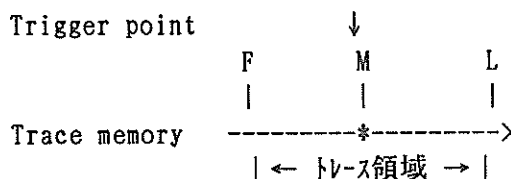
- ・ トレース・メモリおよびサンプル・メモリの最初
(トリガ・ポイント以降をトレース) DLY F
- ・ トレース・メモリおよびサンプル・メモリの中央
(トリガ・ポイント前後をトレース) DLY M
- ・ トレース・メモリおよびサンプル・メモリの最後
(トリガ・ポイント以前をトレース) DLY L

(1) トレース・メモリおよびサンプル・メモリの最初 (トリガ・ポイント以降をトレース) の場合



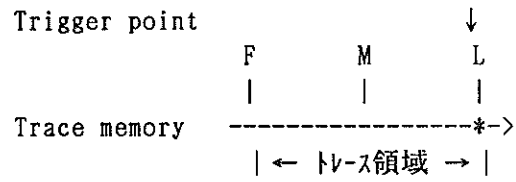
この場合、トリガ・ポイント以降のトレース・データおよびサンプル・データを参照できます。

(2) トレース・メモリおよびサンプル・メモリの中央 (トリガ・ポイント前後をトレース) の場合



この場合、トリガ・ポイントの前後のトレース・データおよびサンプル・データを参照できます。

(3) トレース・メモリおよびサンプル・メモリの最後（トリガ・ポイント以前をトレース）の場合



この場合、トリガ・ポイント以前のトレース・データおよびサンプル・データを参照できます。

3.4 アナライザ

アナライザは、ターゲット・プログラムの実行履歴などを確認できます。トリガの発生により停止します。

IE-78327-Rは、アナライザとして次の4つの機能を持っています。

- (1) リアルタイム・トレース
- (2) 実行時間、命令数計測
- (3) 内部RAMデータ・サンプル
- (4) C0カバレッジ

3.4.1 リアルタイム・トレース (RUN、TRD、TRF、TRG、TRM コマンド)

リアルタイム実行時に、アドレス・バス、データ・バス、ステータス信号などをバス・サイクルごとにトレースします。

アナライザの停止後、トレース・データを表示することによって、プログラム実行の流れがわかります。エミュレーションCPUのリアルタイム実行中も表示できます。

リアルタイム・トレースは、次の特徴を持っています。

・エミュレーションCPUの動作を考慮したトレース表示

プリフェッチしたが、実行されなかった命令の区別表示をします。また、データ・リード/ライト・サイクルと、そのサイクルを発生させた命令のペア表示をします。

・トレース・データ・サーチ機能

目的のトレース・データをサーチし、その前後約5行を表示します。また、目的の条件にあったトレース・データ、分岐命令、チェック・ポイントを選別表示します。

・クオリファイ・トレース

TRXコマンドで指定されたクオリファイ条件のバス・サイクルのみ選別トレースします。

・セクション・トレース (区間トレース)

ENBコマンドで指定されたイベント条件からDSBコマンドで指定されたイベント条件までの区間をトレースします。

・タイム・タグ

前のトレース書き込みから次のトレース書き込みまでの実行時間をシステム・クロック数で表示します。これによって、おおよその実行時間が分かります。

μ PD78320、78322、78P322、78327、78328、78P328は、アクセスするアドレス空間に応じた、次の2系統のバスを持っています。

メイン・バス：通常のメモリ・アクセスを行うバスです。

アクセス空間：0H-0FDFFH

対象メモリ：内部ROM
代替メモリ
ライト・プロテクト付き代替メモリ
ターボ・アクセス・マネージャ代替メモリ
ユーザ・メモリ
内部RAM (0FC80H-0FDFFH)
... μ PD78320, 78322, 78P322
(0FD00H-0FDFFH)
... μ PD78327, 78328, 78P328

対象デバイス内部バス：チップ内蔵のRAM、SFRに高速アクセスを行うバスです。

アクセス空間：0FE00H-0FFFFH

対象メモリ：内部RAM (0FE00H-0FEFFH)
SFR (0FF00H-0FFFFH)

これらの2系統のバスは、並行動作を行うため、同時刻にそれぞれのアクセス動作を行います。したがって、 μ PD78320、78322、78P322、78327、78328、78P328の動作を解析するには、これらの2系統のバスを同時にトレースする必要があります。

IE-78327-Rは、これらの2系統のバスを約8000フレームまで同時にトレースすることができます。

外部データのトレースは、トレース容量に限りがあるため、タイム・タグとの選択となります (TRS コマンド)。

外部データは、外部センス・クリップNo.1-8のデータをトレースします。

タイム・タグは、トレース・フレーム間のシステム・クロックをカウントします。システム・クロックの1クロックは、動作クロック16MHzの場合125ns (8MHz)です。

注意 内部RAM (0FE00H-0FEFFH) をショート・ダイレクト・アドレッシング以外のアドレッシングでリードした場合、トレース・データの値は不定となります。ただし、命令 (ターゲット・プログラム) の実行は正常に行われます。

表3-5 メイン・バス、対象デバイス内部バスのトレース・データ

トレース・データ	メイン・バス	対象デバイス内部バス
アドレス	16ビット	7ビット
データ	8ビット	16ビット
ステータス	6ビット	7ビット
外部データ	8ビット	
タイム・タグ	8ビット	
その他	7ビット	
合計	75ビット × 約8000フレーム	

3.4.2 実行時間、命令数計測

ENBコマンドで指定されたイベント条件からDSBコマンドで指定されたイベント条件までの区間の命令実行時間および実行命令数を計測します。

分解能は、 $0.2\ \mu\text{sec}$ で、 $0.4\ \mu\text{sec}$ から約14分まで計測できます。

実行命令数は、65535まで計測できます。

3.4.3 内部RAMデータ・サンプル (PSA、PSD、PST コマンド)

あらかじめ指定した内部RAMのアドレス（最大3ポイントまで）の内容をワード単位に指定したタイミング（周期）でサンプリングします。サンプリングしたデータはアナライザの停止後、表示できます。

エミュレーションCPUがリアルタイム実行中も表示できます。

サンプル・タイミングは、0.4、0.6、0.8 μsec および 1-10000 μsec (1 μsec 単位) です。

サンプリング容量は、3ワード×約2000フレームです。

3.4.4 COカバレッジ (CVD、CVM コマンド)

プログラムを実行したかどうかを各アドレスごとに記憶します。ブレイク状態のときに各アドレスのプログラムの実行状況をCVDコマンドで表示できます。

また、そのときのプログラム領域全体からみた実行部分の領域の比率をパーセンテージで表示します。

なお、プログラムの領域は、LODコマンドでプログラムをロードするときに自動的に認識されます。そのためパッチなどによりプログラムを増やした場合は、そのエリアをカバレッジ測定範囲追加コマンド(CVM Aコマンド)で測定範囲に追加します。

3.5 アセンブリ言語ディバグ機能

アセンブリ言語ディバグ機能には次の二つの機能があります。

(1) ライン・アセンブル、逆アセンブル (ASM、DAS コマンド)

ASM、DAS コマンドは、ニモニックによるメモリの内容の変更、または、表示を行うことができます。したがって、ダウン・ロードしたオブジェクトに対して、パッチや、プログラムの確認を容易に行うことができます。

通常のニモニックに加え、DB、DW、DS、ORG、END の疑似命令も使用可能です（詳細については、第10章 オンライン・アセンブラ、逆アセンブラ仕様を参照してください）。

(2) シンボリック・ディバグ機能 (LOD、SYM コマンド)

NEC製リロケータブル・アセンブラが出力するシンボル・テーブル・ファイルをロードすることで、登録したシンボルを数値の代わりに使用することができる機能です。

シンボルはLOD file S コマンドにより、モジュール単位で登録できます。

登録可能なシンボル数は最大で約2000です。登録したシンボルはSYM D コマンドにより、モジュール単位で表示できます。

シンボル・テーブル・ファイルによるシンボルの登録(LOD file S コマンド)とIEシンボルの登録とで、シンボルの二重登録となるような場合には、あとから登録するシンボルは無効となります。このような場合、シンボルの削除(SYM K コマンド)後、再度登録を行ってください。

モジュール単位でシンボルの登録を行っている場合、二重登録となるモジュールのシンボルはすべて無効となります。

3.6 PROMプログラマ制御機能 (PGM コマンド)

チャンネル2に、NEC製PROMプログラマ(PG-1500、2000)を接続すると、IE-78327-RをPROMプログラマの端末として使用できます。

このとき、IE-78327-RとPROMプログラマの間でオブジェクトのアップ・ロードまたはダウン・ロードが可能になります。

詳細は、第11章 PROMプログラマ(PG-1500、2000)の使用方法を参照してください。

チャンネル2の通信モードは、チャンネル2通信モード設定機能(MOD コマンド)で指定できます。

チャンネル2にNEC製PROMプログラマ以外の外部装置を接続してIE-78327-Rを端末として使用する場合には、使用可能な制御キャラクタが制限されます。

このような場合、制御キャラクタの変更または解除(PGM C コマンド)により、制御キャラクタの変更、使用制限解除を行ってください。

3.7 補助機能

- (1) オブジェクトのアップ・ロード、ダウン・ロード機能
- (2) トリガ信号外部出力機能
- (3) I E シンボルの操作機能
- (4) デバッグ環境のアップ・ロード、ダウン・ロード機能
- (5) チャネル2 通信モード設定機能
- (6) ファイルからのコマンド入力機能
- (7) コマンド実行結果の出力機能
- (8) コマンド・ファイルの作成機能
- (9) コマンド・ヒストリの表示機能
- (10) コマンド・ヘルプの表示機能
- (11) ディレクトリの表示機能
- (12) 演算機能
- (13) 子プロセスの実行機能
- (14) I E - 7 8 3 2 7 - R の終了機能
- (15) メモリ・ワード設定機能
- (16) その他の機能

3.7.1 オブジェクトのアップ・ロード、ダウン・ロード機能
(LOD、SAV、VRY コマンド)

マッピング状態にしたがって、オブジェクト・コードをメモリへ展開、格納する機能です。

・ オブジェクトのアップ・ロード (SAV file C コマンド)

メモリ上に展開されたオブジェクト・コードを指定したファイルにHEX形式で格納します。

・ オブジェクトのダウン・ロード (LOD file C コマンド)

NEC製リロケータブル・アセンブラ・パッケージが出力するHEX形式オブジェクト・コード、または、セーブ・コマンド (SAV file C コマンド) で作成したHEX形式オブジェクト・コードをメモリ上に展開します。

オブジェクトのアップ・ロード、ダウン・ロードは、マッピング機能により指定された領域に対して行われます。

備考 コマンド行の最後に \$V を付加することでベリファイ指定をすることができます。

3.7.2 トリガ信号外部出力機能 (OUT コマンド)

OUTコマンドは、外部測定機の同期をとるためのトリガ信号をトリガ・ポイント検出時に出力する機能です。

出力されるトリガ信号はトリガ・ポイントの条件 (BRMコマンド) で設定したイベント条件成立時に High レベルとなり、トリガ・ポイントの設定 (DLY コマンド) でのトレース・ディレイ経過後、ロウ・レベルとなります (図3-11 トリガ信号出力タイミングを参照してください)。

このトリガ信号を用いてロジック・アナライザなどにトリガをかけることができます。

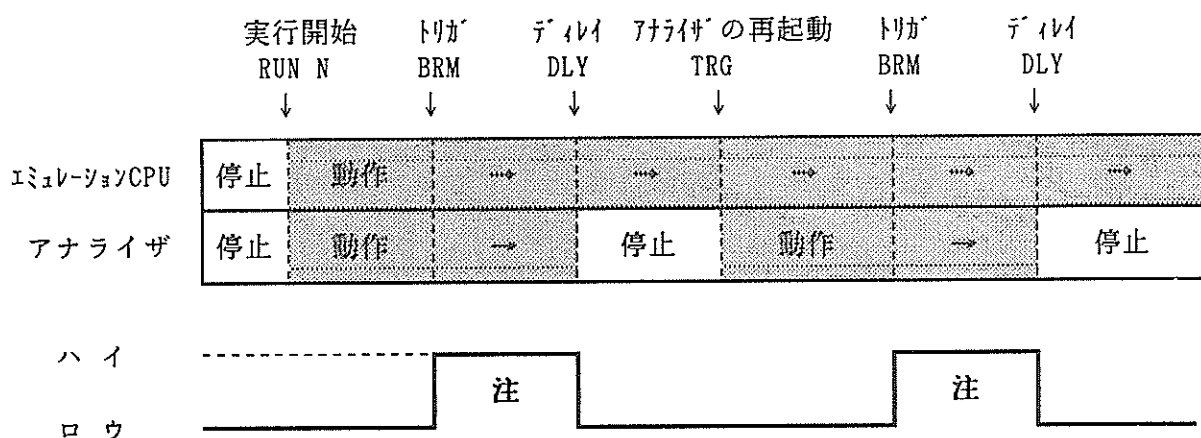
- 注意1. IE-78327-Rは、外部から入力するデータをトレース、イベント検出するため、8本の外部センス・クリップ (No. 1-8) を持っています。8本の外部センス・クリップは、通常入力ラインとなっていますが、OUT ON コマンドにより、No. 1のみをトリガ信号出力とすることができます。
2. OUTコマンドにより、外部センス・クリップのNo. 1をトリガ信号出力とする (OUT ONを実行する) 場合は、この外部センス・クリップをターゲット・システムの信号を出力するラインに接続しないでください。ターゲット・システムやIE-78327-Rを破損する原因になります。

次に、各実行機能における、トリガ信号の出力タイミングを示します。

リアルタイム実行

図3-11 トリガ信号出力タイミング (1/2)

● ノン・ブレーク・リアルタイム実行 (RUN N コマンド)

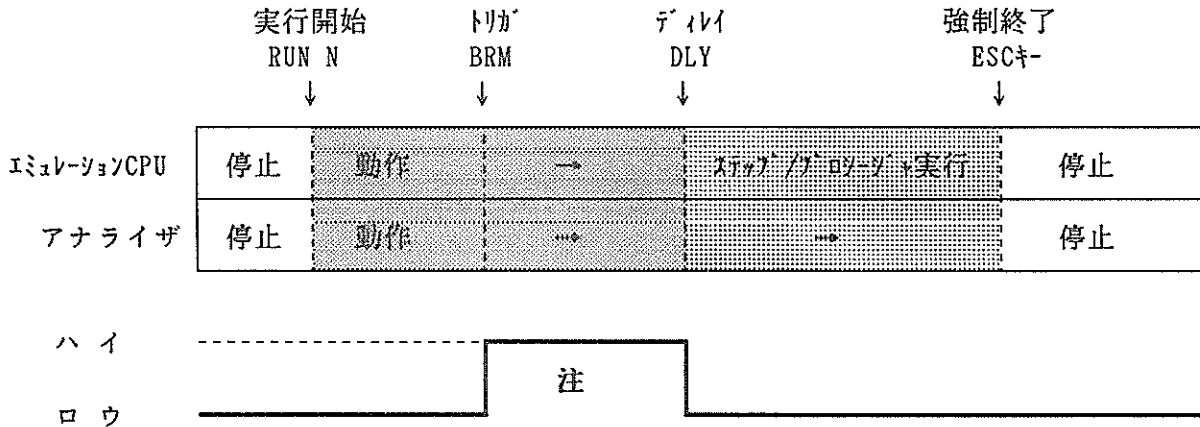


注 デレイ条件設定において DLY L を指定した場合 (BRMでのイベント検出条件成立と同時に、エミュレーションCPU、アナライザが停止する場合)でも、トリガ信号のハイ出力は1 msec以上保持されます。

備考 上の図中の → は、左側の動作状態が続くことを表します。

図3-11 トリガ信号出力タイミング (2/2)

● ブレーク条件付きリアルタイム実行 (RUN B コマンド)



注 デレイ条件設定において DLY L を指定した場合 (BRMでのイベント検出条件成立と同時に、エミュレーションCPU、アナライザが停止する場合)でも、トリガ信号のハイ出力の部分は1 msec以上保持されます。

備考 上の図中の → は、左側の動作状態が続くことを表します。

ノンリアルタイム実行

● ステップ、プロシージャ実行 (RUN T コマンド)

RUN T コマンドによるステップ動作などのノンリアルタイム実行時には、トリガ信号は出力されません。

3. 7. 3 I E シンボルの操作機能 (SYM コマンド)

キー入力により登録した I E シンボルを操作する機能です。

I E シンボルは、アセンブラのシンボル・テーブル・ファイルにより登録されたシンボルと、同等に扱われます。

登録された I E シンボルのモジュール名は、自動的に IESYMBOL となります。

I E シンボルは、コマンドによりアップ・ロード、ダウン・ロードできます。このときの I E シンボル・ファイルは、自動的に次のようになります。

- IE78327.SYM

I E シンボルの操作機能には、次の 8 種類があります。

- 登録 (SYM A)
- 変更 (SYM C)
- 表示 (SYM D)
- 削除 (SYM E)
- 全削除 (SYM K)
- ダウン・ロード (SYM L)
- アップ・ロード (SYM S)
- カレント・モジュール指定 (SYM M)

3.7.4 デバッグ環境のアップ・ロード、ダウン・ロード機能 (LOD、SAV コマンド)

頻繁に使用する特定のデバッグ環境を自動的にファイルへ格納、または、IE-78327-Rへ展開する機能です。

デバッグ環境のダウン・ロード (LOD file D コマンド) は、指定されたファイルからデバッグ環境をIE-78327-Rへ展開します。

デバッグ環境のアップ・ロード (SAV file D コマンド) は、IE-78327-R内のデバッグ環境を指定されたファイルへ格納します。

自動的にアップ・ロード、ダウン・ロードするデバッグ環境を次に示します。

- ・ 内部ROMサイズ
- ・ 内部RAMサイズ
- ・ μ PD78320、78327+TAM emulation
- ・ ダウン・ロード・モード
- ・ デバッグ環境に関する次のコマンド

BRA BRD BRS BRM CHK CLK CVM DSB
ENB DLY MAP MOD PGM PAS PSA PST
TRM TRF TRS TRX WRD

3.7.5 チャンネル2通信モード設定機能 (MOD コマンド)

チャンネル2の、次に示す通信モードを設定する機能です。

- ・ ハンドシェーク・モード
- ・ ボー・レート
- ・ キャラクタ長
- ・ パリティ・ビット
- ・ ストップ・ビット

各通信モードの詳細については、ハードウェア編 第9章 チャンネル1とチャンネル2の機能を参照してください。

3.7.6 ファイルからのコマンド入力機能 (STR コマンド)

キー入力するコマンドまたはデータをコマンド・ファイルから読み込み、自動的に実行する機能です。

コマンド・ファイルは、コマンド・ファイル作成機能 (COMコマンド) またはエディタを使用して作成してください。

エディタを使用して作成するファイル中に仮パラメータを指定できます。

仮パラメータを記述した入力ファイルを使用することで、ファイル中の仮パラメータを実パラメータ (最大4つまで指定可能) に置き換えます。

仮パラメータに対する実パラメータの設定については、第8章 8.41 ファイルからのコマンド入力を参照してください。

コマンド実行中に[^]Lキーを入力すると、コマンド実行を一時停止します。実行を再開する場合は、もう一度[^]Lキーの入力を行ってください。

[^]Kキーを入力するとSTRコマンドを終了します。

3. 7. 7 コマンド実行結果の出力機能 (L S T コマンド)

入力されたコマンドの実行結果を、指定したデバイスへ出力する機能です。

出力デバイスにファイルが指定された場合 (L S T file コマンド) は、実行結果をファイルへ格納します。

リスト装置が指定された場合 (L S T L S T コマンド) にはプリンタへ出力します。

ファイル、プリンタには、コンソールに表示された文字すべてが出力されます。
出力タイミングは、`^P`キーで制御します。

3. 7. 8 コマンド・ファイルの作成機能 (COM コマンド)

キー入力したコマンドまたはデータを、指定したデバイスへ出力する機能です。

出力デバイスにファイルが指定された場合 (COM file コマンド) にコマンド、またはデータをファイルへ格納します。

リスト装置が指定された場合 (COM LST コマンド) に、プリンタへ出力します。

ファイル、プリンタには、キー入力したコマンドまたはデータすべてが出力されます。出力タイミングは ^Oキー で制御します。

備考 COMコマンドでは、仮パラメータを指定できません。

3.7.9 コマンド・履歴の表示機能 (HIS、!n^注 コマンド)

入力された最新のコマンドを20行記憶し、コマンド番号とともに表示する機能です。

記憶されているコマンドすべての表示 (HIS コマンド) は、コマンド行の先頭にコマンド番号が付加されます。

特定のコマンドは、コマンド番号を指定することで呼び出します (!n コマンド)。
この状態で を入力すると呼び出したコマンドを実行できます。
最新のコマンド行は、!! と入力して呼び出すことができます。

注 !n : 表示したいコマンド番号をnに入力します。

3.7.10 コマンド・ヘルプの表示機能 (HLP コマンド)

コマンドの一覧、使用方法を表示する機能です。

表示されるコマンドを次に示します。

ASM	BRA	BRD	BRS	BRM
CHK	CLK	CNT	COM	CVD
CVM	DAS	DIR	DLY	DOS
DSB	ENB	EVN	EXT	HIS
HLP	LOD	LST	MAP	MAT
MEM	MOD	MOV	OUT	PAS
PGM	PSA	PSD	PST	REG
RES	RUN	SAV	SFR	STP
STR	SYM	TRD	TRF	TRG
TRM	TRS	TRX	VRY	WRD

3.7.11 ディレクトリの表示機能 (DIR コマンド)

ファイル名を表示する機能です。これは、使用しているOSのDIR/Wコマンドと同等の機能です。

特にディレクトリ名を表示する場合、前後に〈〉を付加して表示します。

3.7.12 演算機能 (MAT コマンド)

オペランドで記述された式表現を評価し、その結果を16進数、10進数、8進数、2進数で表示します。

3. 7. 1 3 子プロセスの実行機能 (DOS コマンド)

制御を一時的にOSに移す機能です。

この機能を用いることでIE-78327-Rを終了することなく、OSの内部コマンドや外部コマンドが実行できます。

再びコントロール・プログラムに制御を移す場合、EXIT ⓐ と入力してください。

3.7.14 IE-78327-Rの終了機能 (EXT コマンド)

IE-78327-Rを終了し、OSに制御を移す機能です。

注意 IE-78327-Rを終了する場合、必ずこの機能により終了してください。

3.7.15 メモリ・ワード設定機能 (WRD コマンド)

レジスタ、または、メモリ操作におけるメモリ長を設定する機能です。

メモリ長の変更単位は、バイト (WRD B コマンド)、ワード (WRD W コマンド) の二種類です。

3.7.16 その他の機能

IE-78327-Rの特殊な機能として次の三つの機能があります。

- エミュレーションCPU動作中のメモリ、または、レジスタの操作
- HALT、STOP命令実行時のIE-78327-Rの動作
- ラッチ・アップ発生時のIE-78327-Rの動作

(1) エミュレーションCPU動作中のメモリ、または、レジスタの操作

ノンブレイク・リアルタイム実行機能 (RUN N コマンド) において次に示すメモリ、または、レジスタの操作を行った場合、エミュレーションCPUの実行は一時中断されます (中断時間は、実行される各コマンドにより異なります)。

- ・ メモリ操作機能 (MEM C、または、D コマンド)
- ・ レジスタ操作機能 (REG、SFR コマンド)

注意 メモリ、または、レジスタの操作実行後、エミュレーションCPUは再び実行を開始します。しかし、上記のコマンド実行により一時プログラムの実行を中断したあとの動作となっていますので、注意してください (リアルタイム実行にはなりません)。

備考 エミュレーションCPU動作中のメモリ、または、レジスタの操作は、アナライザが停止した状態 (プロンプト emu:0>) においてのみ実行可能です。

(2) HALT、STOP命令実行時のIE-78327-Rの動作

リアルタイム実行機能 (RUN N、または、RUN B コマンド) で、HALT、または、STOP命令を実行した場合、エミュレーションCPUはスタンバイ・モードになります。このスタンバイ・モードが5秒以上継続した場合、IE-78327-Rは次の動作状態になります。

表3-6 スタンバイ・モードでのIE-78327-Rの動作状態

モード名	表示メッセージ	エミュレーションCPUの動作状態	強制的にスタンバイ・モードを解除するコマンド	
			トレーシング状態 trc:0>	エミュレーション状態 emu:0>
スタンバイ・モード	注 E-CPU STOP mode!	STOP中	RES STP	MEM REG RES SFR STP
	注 E-CPU HALT mode!	HALT中		

注 E-CPU: エミュレーションCPU

注意 ノンリアルタイム実行機能 (RUN T コマンド) で、HALT、または、STOP命令を実行してもエミュレーションCPUはスタンバイ・モードにはなりません。

(3) ラッチ・アップ発生時のIE-78327-Rの動作

IE-78327-R内のエミュレーションCPU、周辺CMOSデバイスが、ラッチ・アップが発生した場合、該当デバイスに対する電源供給を停止し、次のようなメッセージを表示します。

注

E-CPU latch up! Restart? (Y)

注 E-CPU: エミュレーションCPU

この状態で Y を入力した場合、IE-78327-Rは再起動します。

再度試みて同じ結果しか得られなかった場合、本製品を購入された最寄りの販売店に相談してください。

第4章 デイバグ手順と画面表示例

この章では、デイバグ作業の手順や、各作業で使用するコマンドについて説明しています。また、I E-78327-Rが持つ特徴的な機能をサンプル・プログラムを用いたデイバグ作業例として、画面イメージで説明します。

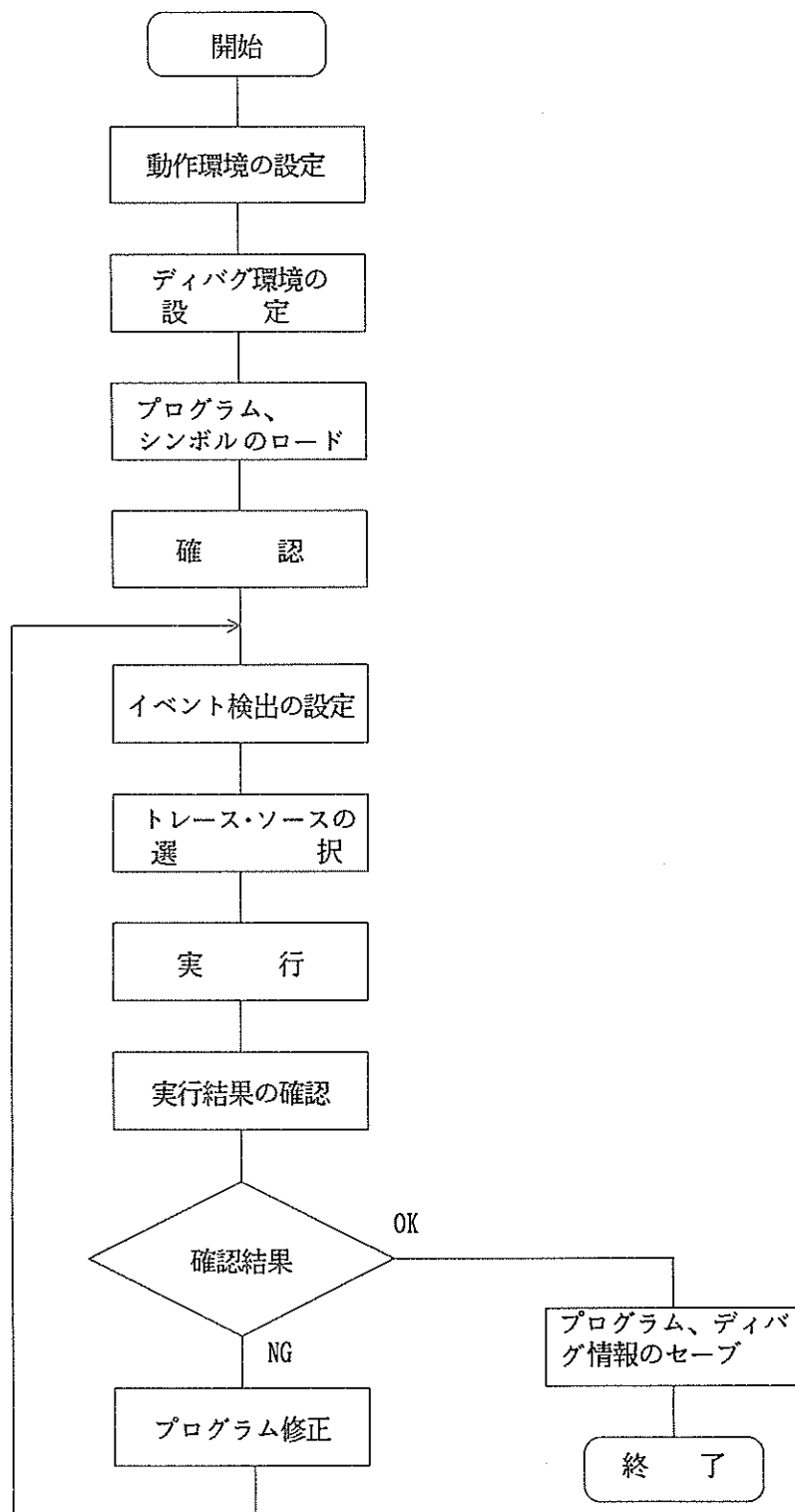
凡例

- _____ : 下線部分は、キーボードからの入力を表します。
- ☐ : リターン・キー (CR (0DH)) の入力を表します。
- R/O : Read Only (リード・オンリー) を表します。
- R/W : Read Write (リード・ライト) を表します。
- このマニュアルで用いられている画面表示例、入力例はPC-9800シリーズをホスト・マシンとして使用した場合のものです。

4.1 デバッグ手順の一例

IE-78327-Rを用いたデバッグ作業の手順の一例を次に図示します。

図4-1 デバッグ手順の一例



ディバグ手順を次の9つの項目に分けて説明します。

- ディバグ環境の設定
- プログラムまたはシンボルのロード
- 確認
- イベント検出の設定
- トレース・ソースの選択
- 実行
- 実行結果の確認
- プログラムの修正
- プログラムまたはディバグ情報のセーブ

(1) デバッグ環境の設定

動作環境の設定を終了すると、コマンド入力待ちになります。

デバッグ作業を開始する前に、通常、次に示すコマンドを用いてデバッグ環境の設定を行います。

- ・ CLK : クロックの選択
- ・ LOD file D : ファイルからのデバッグ環境ロード
- ・ MAP : マッピングの設定
- ・ OUT : トリガ信号の外部出力指定
- ・ RES : エミュレーション・デバイスのリセット
- ・ WRD : メモリ・ワード長の設定

(2) プログラムまたはシンボルのロード

デバッグ環境の設定終了後、プログラムまたはシンボルをロードします。

プログラムのロードは次のコマンドを用いて行います。

- ・ LOD file C : ファイルからのプログラム・ロード
- ・ PGM : NEC製PROMプログラマからのプログラム・ロード

シンボルのロードは、次のコマンドを用いて行います。

- ・ LOD file S : ファイルからのシンボル・ロード
- ・ SYM L : IEシンボルのロード

(3) 確認

ロードしたプログラムまたはシンボルを次に示すコマンドで確認します。

- ・ DAS : プログラムの逆アセンブル
- ・ MEM D : データの確認
- ・ SYM D : シンボルの確認

(4) イベント検出条件の設定

ディバグの初期段階ではポイントごとにイベント検出を行い、そのつどディバグ作業を行います。

イベント検出条件の設定は、次に示すコマンドを用いて行います。

- ・ BRA : バス・イベント検出の条件設定
- ・ BRD : 外部データ検出の条件設定
- ・ BRM : トリガ条件の設定
- ・ BRS : プログラム実行検出の条件設定
- ・ DLY : デイレイ条件の設定
- ・ DSB : ディスエーブル条件の設定
- ・ ENB : イネーブル条件の設定
- ・ PAS : パス条件の設定

(5) トレース・ソースの選択

ディバグ時に必要とするトレース情報を、次に示すコマンドを用いて選択します。

- ・ PSA : サンプル・アドレスの設定
- ・ PST : サンプル・タイミングの設定
- ・ TRM : トレース・モードの選択
- ・ TRS : トレース・データの選択
- ・ TRX : クオリファイ・トレース条件の設定
- ・ CHK : チェック・ポイント条件設定

(6) 実行

ロードしたプログラムを、次に示す5種類のコマンドを用いて実行します。

- ・ RUN N : ノンブレイク・リアルタイム実行
- ・ RUN B : ブレイク付きリアルタイム実行
- ・ RUN T : ステップ実行
- ・ RUN T : プロシージャ実行
(PRC付き)
- ・ TRG : アナライザの再起動

(7) 実行結果の確認

イベント検出によりエミュレーションCPUとアナライザが停止したあと、次に示すコマンドを用いて実行結果の確認を行います。

- ・ CNT : 実行経過時間、実行命令数の確認
- ・ CVD : C0カバレッジ測定結果の表示
- ・ MEM D : データの確認
- ・ PSD : サンプル・データの表示
- ・ REG D : 汎用レジスタ、フラグの確認
- ・ SFR D : SFRの確認
- ・ TRD : トレース・データの確認
- ・ TRF : トレース・データ検索条件の指定

(8) プログラムの修正

大規模なプログラム修正の場合、ソース・レベルからプログラムを修正し、IE-78327-Rに再ロードします。

小規模なプログラム修正の場合、次に示すコマンドを用いて、プログラムまたはデータ修正を行います。

- ・ ASM : ニモニックでのプログラム修正
- ・ MEM C : メモリ内容の変更

(9) プログラムまたはディバグ情報のセーブ

ディバグ作業終了後、プログラムまたはディバグ情報をセーブします。プログラムのセーブは、次のコマンドを用いて行います。

- ・ SAV file C : ファイルへのプログラム・セーブ
- ・ PGM : NEC製PROMプログラマへのプログラム・セーブ

シンボル情報のセーブは、次のコマンドを用いて行います。LODコマンドでロードしたシンボルのセーブはできません。

- ・ SYM S : I Eシンボルのセーブ

デバッグ環境のセーブは、次のコマンドを用いて行います。

- ・ SAV file D : ファイルへのデバッグ環境セーブ

4. 2 画面表示例

IE-78327-Rが持つ特徴的な機能を、サンプル・プログラムを用いたデバッグ作業例として、画面イメージで説明します。

サンプル・プログラムは、特徴的な機能を説明することを目的に作られています。したがって、プログラム自体は特別な意味を持ちません。

4. 2. 1 マクロ・サービス、割り込み処理に関するイベント検出、 トレース・データの表示

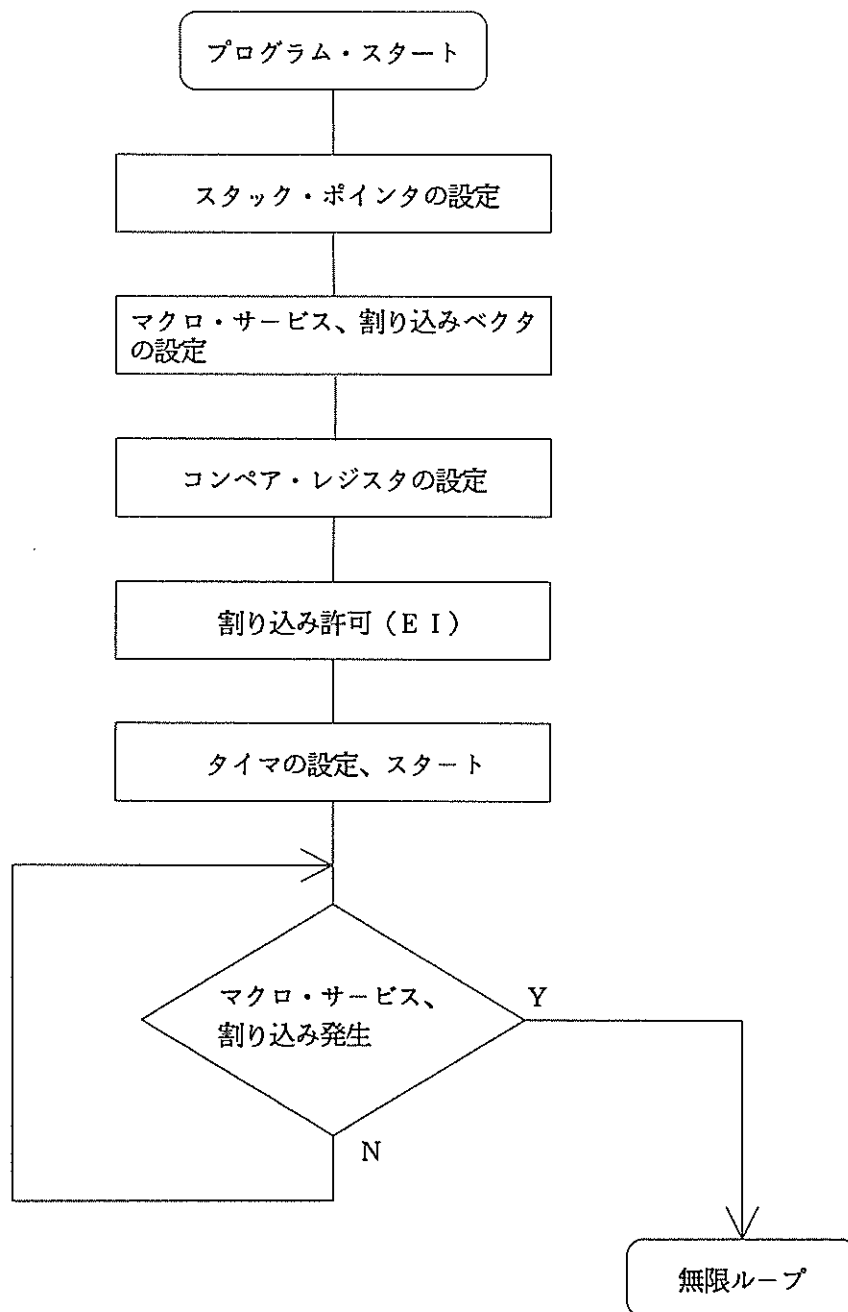
マクロ・サービスを使用したサンプル・プログラムのデバッグ作業を例に説明します。

(1) サンプル・プログラムの概略、フロー・チャート、プログラム本体

サンプル・プログラムの概略

- ・ タイマ0をフリー・ランニングさせ、あらかじめ設定したコンペア・レジスタ値とタイマ0の内容との一致でブロック転送マクロ・サービスを起動させます（ポート2のリード・データを0FD00H番地にセーブ）。
- ・ マクロ・サービスを1回起動後に割り込みを発生します。
- ・ 割り込みルーチンで無限ループを実行します。

サンプル・プログラムのフロー・チャート



サンプル・プログラム本体

```

    ORG    100H
MAIN:
    MOVW   SP, #0FE00H      ←スタック・ポインタの設定
    MOV    R1, #14H
    MOV    !0FE1EH, A
    MOV    R1, #41H
    MOV    !0FE1F, A
    MOV    0FE40H, #1H
    MOV    0FE41H, #2H
    MOVW   0FE3E, #0FD00H
    MOV    MK0H, #0EFH      ←ブロック転送マクロ・サービス、
    MOV    ISMOH, #10H      割り込みベクタの設定
    MOVW   RP7, #1000H
    MOVW   !1EH, RP7
    MOVW   CC10, #0H
    EI
    MOV    TMC0, #80H
    BR     $$
;
    END

    ORG    1000H
INT:
    BR     $INT             ←無限ループ
;
    END

```

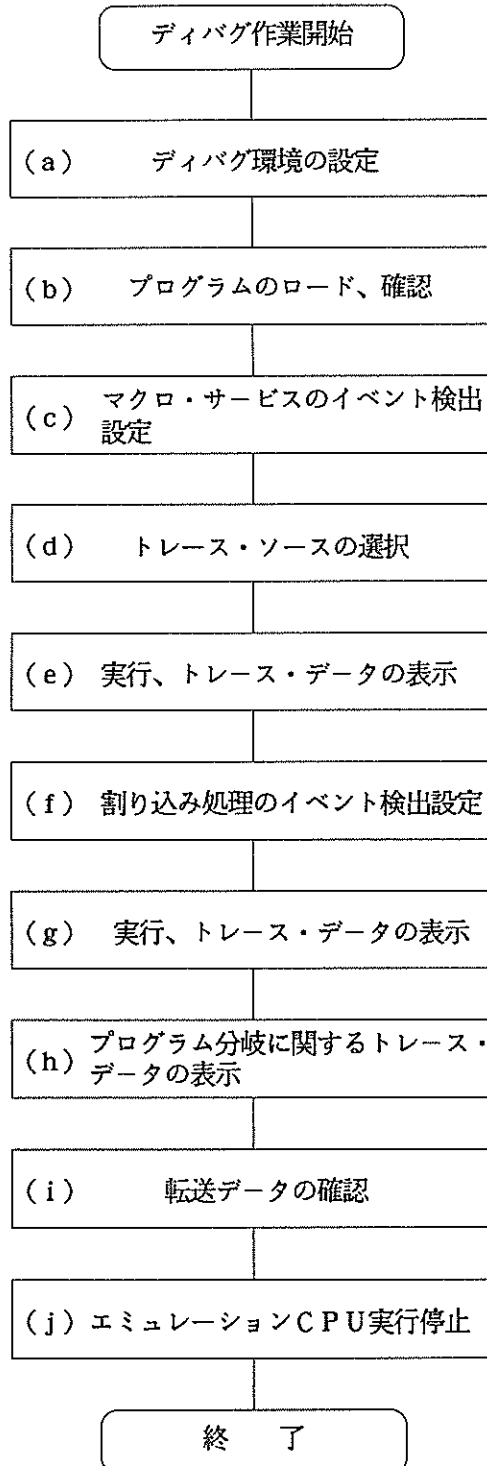
(2) サンプル・プログラムに対するデバッグ作業の概略、
フロー・チャート、画面表示例

デバッグ作業の概略

- ・ マクロ・サービスによるブロック転送をイベント検出（ポート2からのデータ・リードをイベント検出）
- ・ 割り込み処理に伴うベクタ参照をイベント検出
- ・ イベント検出後トレース・データを表示
- ・ プログラム分岐処理に関するトレース・データを検索表示
- ・ マクロ・サービスによる転送データを確認（エミュレーションCPU動作中のメモリ内容確認）
- ・ エミュレーションCPUの実行停止

デバッグ作業のフロー・チャート

- ・ (a) - (j) は次頁以降の画面表示例に対応しています。



(3) 画面表示例

(a) デバッグ環境の設定

```

brk:0>CLK I [Enter]
brk:0>RES [Enter]
brk:0>OUT OFF [Enter]
brk:0>MAP I 0K [Enter]
brk:0>MAP W 0,3FFF [Enter]
brk:0>MAP [Enter]
    0000-3FFF      R/W emulation
    4000-FC7F      Non map
    FC80-FEFF      <Internal RAM>
brk:0>WRD W [Enter]
brk:0>
    
```

・ (a) で使用されているコマンドの説明

- ・ CLK I : エミュレータ内のクロックを選択
- ・ RES : エミュレーション・デバイスをリセット
- ・ OUT OFF : トリガ信号の外部出力を禁止
- ・ MAP I 0K : 内部ROMを0Kバイトに設定
- ・ MAP W 0,3FFF : 0H-03FFFH 番地を代替メモリ・マッピングに指定
- ・ MAP : マッピング状態の確認
- ・ WRD W : メモリ長をワードに設定

(b) プログラムのロード、確認

```

brk:0>LOD MACRO C S ⓐ
object load complete
symbol table loading
PUBLIC      load complete
brk:0>DAS MAIN ⓐ
Addr Object          Mnemonic
                    PUBLICYMAIN::
0100 0B FC 00 FE      MOVW    SP,#0FE00H
                    .
                    .
011A 2B E5 EF        MOV     MKOH,#0EFH
brk:0>DAS INT ⓐ
Addr Object          Mnemonic
                    PUBLICYINT::
1000 14 FE           BR     $INT
1002 00              NOP
                    .
                    .

brk:0>

```

• (b) で使用されているコマンドの説明

- LOD MACRO C S : オブジェクトまたはシンボルをロード
- DAS MAIN : プログラムの確認
- DAS INT : プログラムの確認

(c) マクロ・サービスのイベント検出設定

```

brk:0>BRA 1 [Enter]
A 0XXXXH
V 0XXH
C NC
E 0XXXXY

A = P2 [Enter]
V = [Enter]
OP (OPecode fetch)          RP (Read by program)
RW (Read Write)            WP (Write by program)
R (Read)                   RWM (Read Write by Macro service)
W (Write)                  RM (Read by Macro service)
RWP (Read Write by program) WM (Write by Macro service)
                             NC (No condition)

C = RM [Enter]
E = 0FH [Enter]
brk:0>BRM BRA1 [Enter]
brk:0>DLY M [Enter]
    
```

・ (c) で使用されているコマンドの説明

- ・ BRA 1 : 対話形式でバス・イベント検出条件を設定 (BRA 1)
 - ・ 現在の設定状態を表示
 - ・ P 2 に対するマクロ・サービス・リード・アクセスを設定
- ・ BRM BRA1 : バス・イベント検出の 1 をトリガ条件に設定
- ・ DLY M : トリガ・ポイントをトレース・メモリの中央に設定

(d) トレース・ソースの選択

```
brk:0>TRM ALL␣  
brk:0>TRS E␣
```

・ (d) で使用されているコマンドの説明

- ・ TRM ALL : 実行開始からブレークまでの実行のトレースを指定
- ・ TRS E : 外部データのトレースを指定

(e) 実行、トレース・データの表示

```

brk:0>RUN B MAIN [C]
User-system Vcc-OFF Emulation start at 100
<External data trace mode>
trc:0>
Bus detection break terminated BRA1
brk:0>TRD I [C]
  Frame Status Address Data Label Mnemonic EX
  0079 MSRDR FF02 XX 00
T0079 MSRDR FF02 XX 00
  0081 MSWR F000 00 00
<INTCC10>
  0082 INTRD 001E 00 00
  0083 INTRD 001F 10 00
  0085 INTWR FDFE 08 00
  0086 INTWR FDFE 00 00
  0088 INTWR FDFC 2F 00
  0089 INTWR FDFD 01 00
                                PUBLICYINT::
  0090 1000 14FE BR $INT
Total frame = 4174T (L/F/T+/cr-/Frame No./.)?. <ESC>
brk:0>
    
```

・ (e) で使用されているコマンドの説明

- RUN B MAIN : ブレーク付きリアルタイム実行の開始
 - ・ トレース・モードの設定状態を表示
- TRD I : トレース・データをインストラクション表示
 - ・ トリガ・フレームをフレーム・モードで表示
 - ・ <>内に割り込み要因を示すシンボルを表示

(f) 割り込み処理のイベント検出設定

```

brk:0>BRA 2 (Q)
A OXXXXH
V OXXH
C NC
E OXXXXY

A = 1EH (Q)
V = 0 (Q)
OP (OPecode fetch)          RP (Read by program)
RW (Read Write)            WP (Write by program)
R (Read)                   RWM (Read Write by Macro service)
W (Write)                  RM (Read by Macro service)
RWP (Read Write by program) WM (Write by Macro service)
NC (No condition)

C = R (Q)
E = <ESC>
brk:0>BRM BRA2 (Q)
brk:0>DLY (Q)
TRIGGER POINT      F      M      L
                   |      |      |
TRACE MEMORY ----- * ----- >
brk:0>
brk:0>TRM (Q)
ALL
brk:0>
    
```

・ (f) で使用されているコマンドの説明

- BRA 2 : 対話形式で、バス・イベント検出条件を設定 (BRA2)
 - ・ 現在の設定状態を表示
 - ・ アドレス: 01EH 番地からデータ: 00H のリードを設定
- BRM BRA2 : バス・イベント検出の 2 をトリガ条件に設定
- DLY : トリガ・ポイントのトレース・メモリに対する設定位置を確認
 - ・ * の位置で中央に設定されている
- TRM : トレース・モードを確認
 - ・ 実行開始からブレークまでの実行トレースが設定されている

(g) 実行、トレース・データの表示

```

brk:0>RES ⓐ
brk:0>RUN N MAIN ⓐ
  User-system Vcc-ON      Emulation start at 0100
  (External data trace mode)
trc:0>
  Bus detection break    terminated BRA2
emu:0>TRD I ⓐ
  Frame Status Address Data Label Mnemonic EX
  0079  MSRDR  FF02  XX  00
  0081  MSWR  FD00  00  00
  (INTCC10)
  0082  INTRD  001E  00  00
T0082  INTRD  001E  00  00
  0083  INTRD  001F  00  00
  0085  INTWR  FDFE  08  00
  0086  INTWR  FDFE  00  00
  0088  INTWR  FDFC  2F  00
  0089  INTWR  FDFD  01  00
                                     PUBLICYINT::
  0090          1000  14FE  BR  $INT
  Total frame = 4177T (L/F/T+/cr-/Frame No./.)? <ESC>
emu:0>

```

・ (g) で使用されているコマンドの説明

- RUN N MAIN : ノンブレイク・リアルタイム実行の開始
 - ・ トレース・モードの設定状態を表示
- TRD I : トレース・データをインストラクション表示
 - ・ トリガ・フレームをフレーム・モードで表示
 - ・ () 内に割り込み要因を示すシンボルを表示

(h) プログラム分岐に関するトレース・データの表示

```

emu:0>TRD I $J Ⓜ
  Frame Status Address Data          Label Mnemonic      EX
<INTCC10>
  0082  INTRD  001E  00                                00
T0082  INTRD  001E  00                                00
  0083  INTRD  001F  10                                00
  0085  INTWR  FDFE  08                                00
  0086  INTWR  FDFE  00                                00
  0088  INTWR  FDFC  2F                                00
  0089  INTWR  FDFD  01                                00

                                PUBLICYINT::
  0090                1000  14FE                                BR    $INT
                                PUBLICYINT::
  0094                1000  14FE                                BR    $INT
                                PUBLICYINT::
  0098                1000  14FE                                BR    $INT
  Total frame = 4177T (L/F/T+/cr-/Frame No./.) ? <ESC>
emu:0>
    
```

・ (h) で使用されているコマンドの説明

TRD I \$J : プログラム分岐に関するトレース・データのみ表示

(i) 転送データの確認

```
emu:0>SFR D P2 (Q)
P2      00
emu:0>
```

・ (i) で使用されているコマンドの説明

SFR D P2 : SFRの内容表示

- ・ エミュレーションCPUの実行を中断して、SFR内容表示後、自動的に実行を再開します。

(j) エミュレーションCPUの実行停止

```

emu:0>STP (j)
Escape break      terminated
PC   SP  PSW: UF  RBS2 RBS1 RBS0 S   Z   RSS AC  IE  P/V LT  CY
1007 FDFE          0   0   0   0   0   0   0   0   0   0   0   1   0
R0   R1  R2   R3   R4   R5   R6   R7           RP4   RP5   RP6   RP7
X   A   C   B                               VP   UP   DE   HL
00  00  00  00  00  00  00  00           0000  0000  0000  0000
brk:0>
    
```

・ (j) で使用されているコマンドの説明

STP : エミュレーションCPUの実行停止

- ・ エミュレーションCPUの実行停止時、カレント・バンクのレジスタを表示

4. 2. 2 外部データを用いたイベント検出、トレース・データの表示

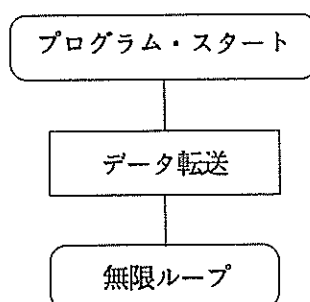
バンク切り替えを使用したサンプル・プログラムのディバグ作業を例に説明します。

(1) サンプル・プログラムの概略、フロー・チャート、プログラム本体

サンプル・プログラムの概略

- ・ 対象デバイスのアクセスできる空間64Kバイトをバンク切り替えを使用し、1Mバイトに拡張しているシステムで、アドレス84000H番地のメモリにデータ・ライトします（IEの設定としては外部センス・クリップのNo.5-8をターゲット・システムのアドレス16-19に接続しているものとします）。
- ・ 無限ループの実行をします。

サンプル・プログラムのフロー・チャート



サンプル・プログラム本体

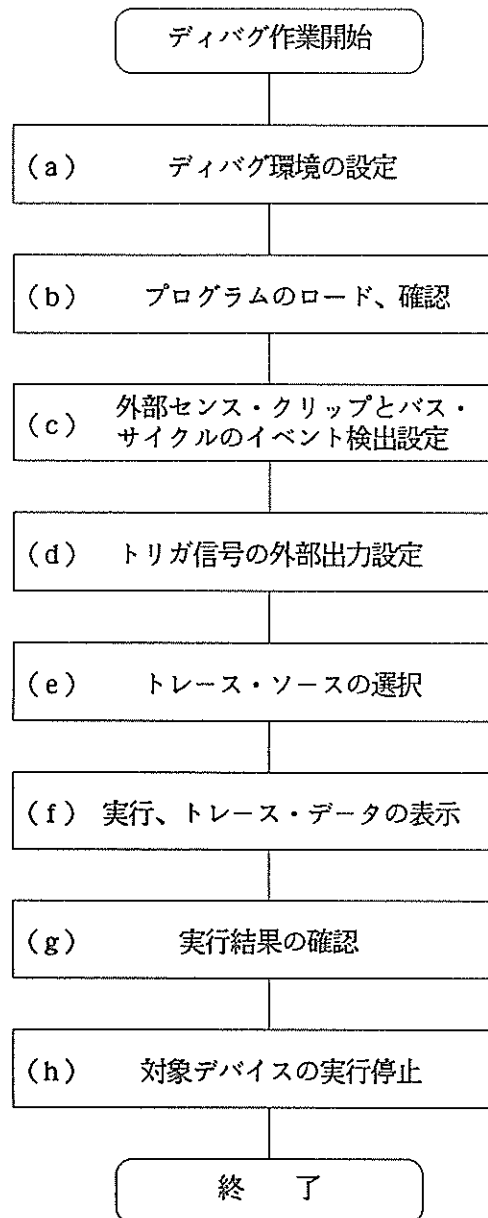
```
ORG    100H
EXTACSS:
MOV    R1, #12           ←転送データ設定
MOV    !4000H, A        ←データ転送
NOP
NOP
BR     $$               ←無限ループ
END
```

(2) サンプル・プログラムに対するデバッグ作業の概略、
フロー・チャート、画面表示例

デバッグ作業の概略

- ・ 外部センス・クリップ (No. 5 - 8) から入力するデータとバス・サイクルをAND条件でイベント検出 (84000番地に対するアクセスをイベント検出)。
- ・ イベント検出時、外部センス・クリップ (No. 1) からトリガ信号を出力。
- ・ イベント検出後トレース・データを表示。
- ・ ターゲット・システムのメモリに転送されたデータを確認 (エミュレーションCPU動作中のメモリ内容確認)。
- ・ エミュレーションCPUの実行停止。

デバッグ作業のフロー・チャート



(3) 画面表示例

(a) デバッグ環境の設定

```

brk:0>CLK I [E]
brk:0>RES [E]
brk:0>OUT OFF [E]
brk:0>MAP I 16K [E]
brk:0>MAP U 4000,7FFF [E]
brk:0>MAP [E]
  0000 - 3FFF   Internal ROM
  4000 - 7FFF   User
  8000 - FC7F   Non map
  FC80 - FEFF  <Internal RAM>
brk:0>WRD B [E]
brk:0>
    
```

・ (a) で使用されているコマンドの説明

- ・ CLK I : エミュレータ内クロックを選択
- ・ RES : エミュレーション・デバイスをリセット
- ・ OUT OFF : トリガ信号の外部出力を禁止
- ・ MAP I 16K : 内部ROMを16Kバイトに設定
- ・ MAP U 4000,7FFF : 04000H 番地-07FFFH 番地にユーザ・メモリを指定
- ・ MAP : マッピング状態の確認
- ・ WRD B : メモリ長をバイトに設定

(b) プログラムのロード、確認

```

brk:0>LOD EXSIGNAL C S ⓐ
object load complete
symbol table loading
PUBLIC      load complete
brk:0>DAS EXTACSS ⓐ
Addr  Object                Mnemonic
                                PUBLICYEXTACSS::
0100  B9 12                   MOV      R1,#12H
0102  09 F1 00 40            MOV      !4000H,A
0106  00                      NOP
                                .
                                .
brk:0>
    
```

・ (b) で使用されているコマンドの説明

- ・ LOD EXSIGNAL C S : オブジェクト、シンボルをロード
- ・ DAS EXTACSS : プログラムの確認

(c) 外部センス・クリップとバス・サイクルのイベント検出設定

```

brk:0>BRA 1 [Enter]
A 0XXXXH
V 0XXH
C NC
E 0XXXXY

A = 4000 [Enter]
V = 12H [Enter]
OP (OPecode fetch)          RP (Read by program)
RW (Read Write)            WP (Write by program)
R (Read)                   RWM (Read Write by Macro service)
W (Write)                  RM (Read by Macro service)
RWP (Read Write by program) WM (Write by Macro service)
                             NC (No condition)

C = WP [Enter]
E = 8H [Enter]
brk:0>BRM BRA1 [Enter]
brk:0>DLY M [Enter]
brk:0>
    
```

・ (c) で使用されているコマンドの説明

- ・ BRA 1 : 対話形式でバス・イベント検出条件を設定 (BRA1)
 - ・ 現在の設定状態を表示
 - ・ アドレス : 04000H 番地、
 - 外部データ : 8H 、
 - 転送データ : 12H 、
 - バス・ステータス : WP の AND 条件を設定
- ・ BRM BRA1 : バス・イベント検出の 1 をトリガ条件に設定
- ・ DLY M : トリガ・ポイントをトレース・メモリの中央に設定

(d) トリガ信号の外部出力設定

```
brk:0>OUT ON Ⓢ
```

- ・ (d) で使用されているコマンドの説明

OUT ON : トリガ信号の外部出力を許可

(e) トレース・ソースの選択

```
brk:0>TRM ALL [↵]
```

```
brk:0>TRS E [↵]
```

・ (e) で使用されているコマンドの説明

- ・ TRM ALL : 実行開始からブレークまでの実行のトレースを設定
- ・ TRS E : 外部データのトレースを設定

(f) 実行、トレース・データの表示

```
brk:0>RUN N EXTACSS [E]
User-system Vcc-OFF      Emulation start at 0100
<External data trace mode>
<Trigger output mode>
trc:0>
  Bus cycle event      terminated
emu:0>TRD I [E]
  Frame Status Address Data          Label Mnemonic          EX
  0017   WR    4000    12
  T0017   WR    4000    12
  0010           0107    00                NOP                81
  0012           0108   14FE                BR                $108H            81
  0018           0108   14FE                BR                $108H            81
  0022           0108   14FE                BR                $108H            81
  0026           0108   14FE                BR                $108H            81
  Total frame = 4112T  (L/F/T/+/cr/-/Frame No./.) ? <ESC>
```

・ (f) で使用されているコマンドの説明

- ・ RUN N EXTACSS : ノンブレイク・リアルタイム実行の開始
 - ・ トレース・モードの設定状態を表示
 - ・ トリガ信号の外部出力指定を表示
- ・ TRD I : トレース・データをインストラクションで表示
 - ・ トリガ・フレームをフレーム・モードで表示
 - ・ イベント検出と同時に、外部センス・クリップ (No.1) からトリガ信号としてハイ・レベルが出力されたことをトレース表示

(g) 実行結果の確認

```
emu:0>MEM D 4000,4000 (D)
4000 00
emu:0>
```

・ (g) で使用されているコマンドの説明

MEM D 4000,4000 : メモリ内容の表示

- ・ エミュレーションCPUの実行を中断してメモリ内容表示後、自動的に実行を再開

(h) エミュレーションCPUの実行停止

```

emu:0>STP Ⓢ
Escape break      terminated
PC   SP  PSW: UF   RBS2 RBS1 RBS0 S   Z   RSS AC  IE  P/V LT  CY
0108 0000      0   0   0   0   0   0   0   0   0   0   0   0   0
  R0  R1  R2  R3  R4  R5  R6  R7      RP4  RP5  RP6  RP7
   X  A  C  B                        VP  UP  DE  HL
   00 12 00 00 00 00 00 00      0000 0000 0000 0000
brk:0>
    
```

・ (h) で使用されているコマンドの説明

STP : エミュレーションCPUの実行停止

- ・ エミュレーションCPUの実行停止時、カレント・バンクのレジスタを表示

4. 2. 3 プロシージャ実行に関するイベント検出、トレース・データの表示

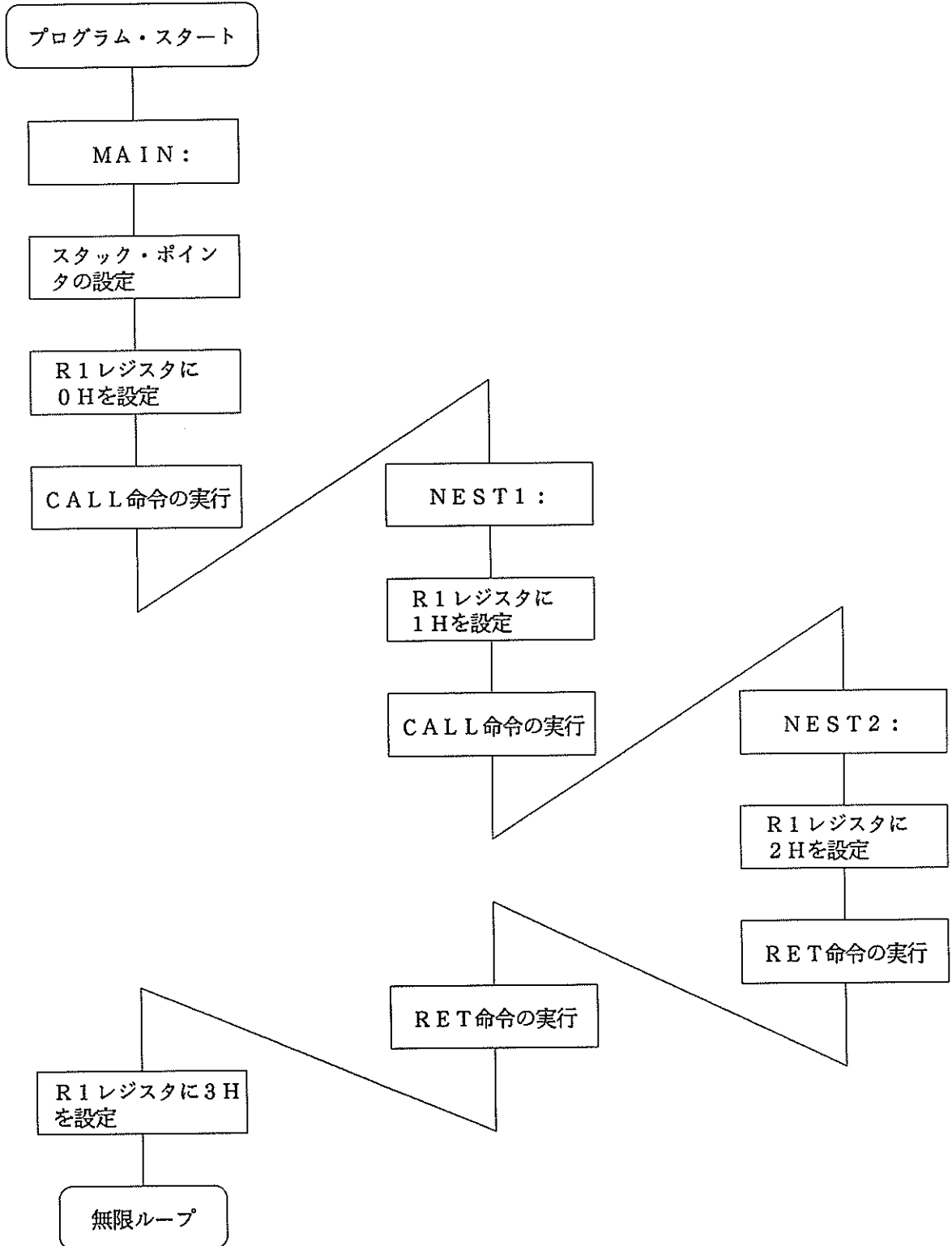
プロシージャ実行機能について、サンプル・プログラムのデバッグ作業を例に説明します。

(1) サンプル・プログラムの概略、フロー・チャート、プログラム本体

サンプル・プログラムの概略

- ・ MAIN: でR1レジスタに 0H を設定、CALL 命令を実行して深いネスト・レベル (NEST1:) へ移行します。
- ・ NEST1: でR1レジスタに 1H を設定、CALL 命令を実行して深いネスト・レベル (NEST2:) へ移行します。
- ・ NEST2: でR1レジスタに 2H を設定、RET 命令を実行してNEST1: へ復帰します。
- ・ NEST1: (復帰先) でRET 命令を実行してMAIN: へ復帰します。
- ・ MAIN: (復帰先) でR1レジスタに 3H を設定、無限ループの実行をします。

サンプル・プログラムのフロー・チャート



サンプル・プログラム本体

```

MAIN:      ORG    100H
           MOVW  SP, #0FE00H      ←スタック・ポインタの設定
           MOV   R1, #0           ←R 1 レジスタに 0H を設定
           CALL !NEST1           ←深いネスト・レベルへ移行
           MOV   R1, #3H          ←R 1 レジスタに 3H を設定
           BR    $$              ←無限ループ
           END

NEST1:    ORG    1000H
           MOV   R1, #1           ←R 1 レジスタに 1H を設定
           CALL !NEST2           ←深いネスト・レベルへ移行
           NOP
           RET                    ←元のネスト・レベルへ復帰

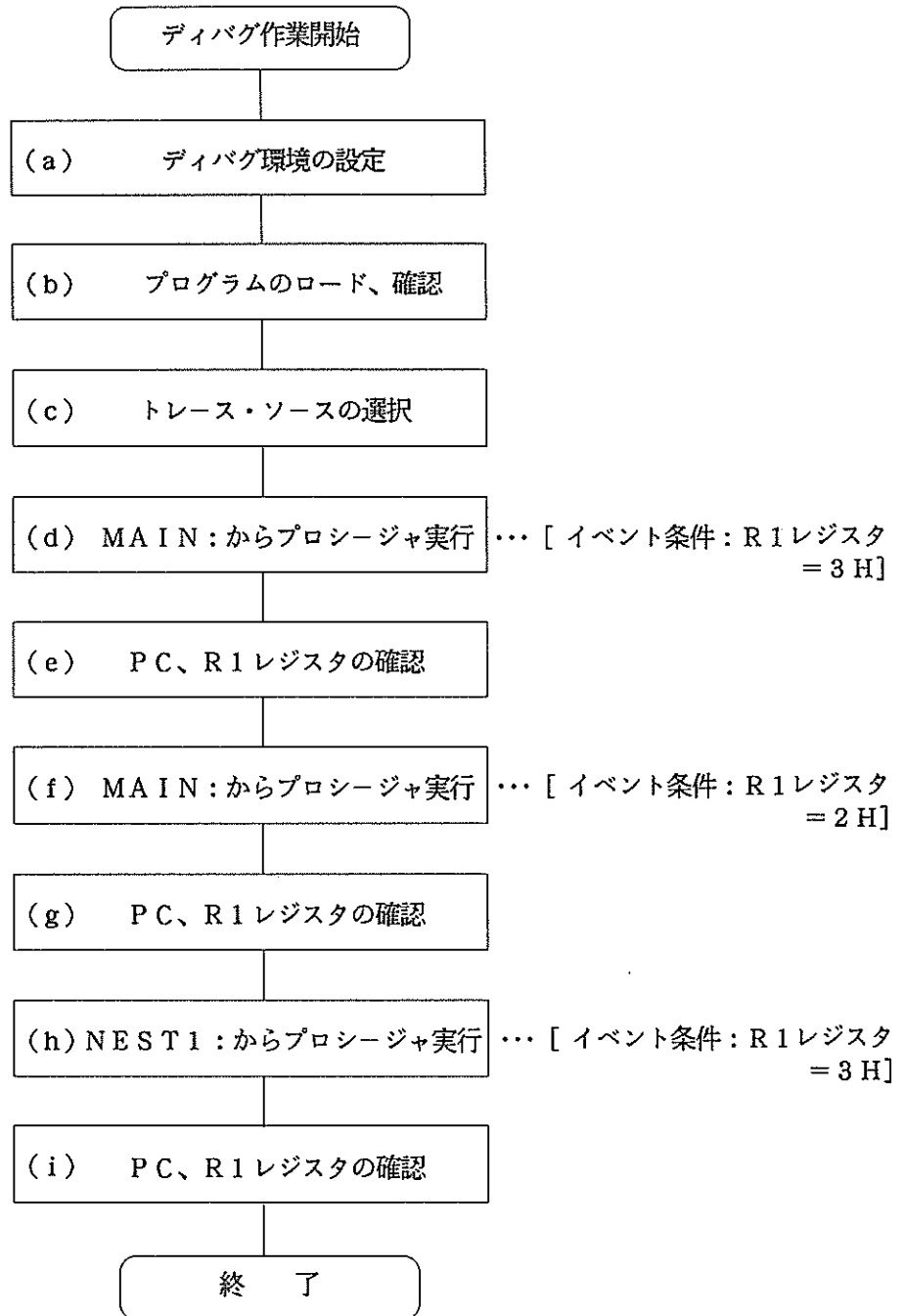
NEST2:    ORG    2000H
           MOV   R1, #2           ←R 1 レジスタに 2H を設定
           NOP
           RET                    ←元のネスト・レベルへ復帰
           END
    
```

(2) サンプル・プログラムに対するデバッグ作業の概略、
フロー・チャート、画面表示例

デバッグ作業の概略

- ・ イベント条件でR1レジスタ = 3H を指定、MAIN : からプロシージャ実行。
- ・ PC、R1レジスタ値を確認。
- ・ イベント条件でR1レジスタ = 2H を指定、MAIN : からプロシージャ実行。
- ・ PC、R1レジスタ値を確認。
- ・ イベント条件でR1レジスタ = 3H を指定、NEST1 : からプロシージャ実行。
- ・ PC、R1レジスタ値を確認。

デバッグ作業のフロー・チャート



(3) 画面表示例

(a) デバッグ環境の設定

```
brk:0>CLK I [Enter]
brk:0>RES [Enter]
brk:0>OUT OFF [Enter]
brk:0>MAP I 16K [Enter]
brk:0>MAP [Enter]
0000 - 3FFF Internal ROM
4000 - FC7F Non map
FC80 - FEFF <Internal RAM>
brk:0>WRD B [Enter]
brk:0>
```

・ (a) で使用されているコマンドの説明

- ・ CLK I : エミュレータ内クロックを選択
- ・ RES : エミュレーション・デバイスをリセット
- ・ OUT OFF : トリガ信号の外部出力を禁止
- ・ MAP I 16K : 内部ROMを16Kバイトに設定
- ・ MAP : マッピング状態の確認
- ・ WRD B : メモリ長をバイトに設定

(b) プログラムのロード、確認

```

brk:0>LOD PROCED C S Ⓜ
object load complete
symbol table loading
PUBLIC load complete
brk:0>DAS MAIN Ⓜ
Addr Object Mnemonic
PUBLICYMAIN::
0100 0B FC 00 FE MOVW SP,#0FE00H
0104 B9 00 MOV R1,#0H
.
.

brk:0>DAS NEST1 Ⓜ
Addr Object Mnemonic
PUBLICYNEST1::
1000 B9 01 MOV R1,#1H
1002 28 00 20 CALL !NEST2
.
.

brk:0>DAS NEST2 Ⓜ
Addr Object Mnemonic
PUBLICYNEST2::
2000 B9 02 MOV R1,#2H
2002 00 NOP
.
.

brk:0>

```

・ (b) で使用されているコマンドの説明

- ・ LOD PROCED C S : オブジェクト、シンボルをロード
- ・ DAS MAIN : プログラムの確認
- ・ DAS NEST 1 : プログラムの確認
- ・ DAS NEST 2 : プログラムの確認

(c) トレース・ソースの選択

```
brk:0>TRS E
```

・ (c) で使用されているコマンドの説明

TRS E : 外部データのトレースを設定

(d) MAIN:からのプロセス実行

```

brk:0>RUN T MAIN,R1=03H TRD PRC ㊟
User-system Vcc-OFF   Emulation start at 0100
< External-Sence-Cable trace mode >
Frame Status Address Data      Label Mnemonic      EX
PUBLICYMAIN::
0000      0100  0BFC00FE      MOVW   SP,#0FE00H
0004  WR   FDFE  0104
0006      0104  B900          MOV    R1,#0H
0010      0106  280010       CALL  !NEST1
0015  WR   FDFE  09           00
0016  WR   FDFF  01           00
0019      0109          MOV    R1,#3H
terminated
    
```

・ (d) で使用されているコマンドの説明

RUN T MAIN,R1=3H TRD PRC : プロセス実行の開始

- ・ イベント条件としてR1レジスタ=3Hを設定
- ・ カレント・ルーチンだけのイベント条件検出およびトレースとなる
- ・ イベント条件検出で実行を終了し、カレント・バンクのレジスタを表示

(e) PC、R1レジスタの確認

PC	SP	PSW: UF	RBS2	RBS1	RBS0	S	Z	RSS	AC	IE	P/V	LT	CY
010B	FE00	0	0	0	0	0	0	0	0	0	0	0	0
R0	R1	R2	R3	R4	R5	R6	R7	RP4	RP5	RP6	RP7		
X	A	C	B					VP	UP	DE	HL		
00	03	00	00	00	00	00	00	0000	0000	0000	0000		
One step emulation standby													
brk:0>													

・ (e) で使用されているコマンドの説明

- ・ PC = 010BH、R1レジスタ = 3H

(f) MAIN: からプロシージャ実行

```

brk:0>RUN T MAIN,R1=02H TRD PRC [D]
User-system Vcc-OFF Emulation start at 0100
< External data trace mode >
Frame Status Address Data Label Mnemonic EX
PUBLICYMAIN::
0000 0100 0BFC00FE MOVW SP,#0FE00H
0004 WR FFFC 00FE 00
0006 0104 B900 MOV R1,#0H
0010 0106 280010 CALL !NEST1
0015 WR FDFF 09 00
0016 WR FDFF 01 00
terminated
    
```

・ (f) で使用されているコマンドの説明

RUN T MAIN,R1=2H TRD PRC : プロシージャ実行の開始

- ・ イベント条件としてR1レジスタ= 2H を設定
- ・ カレント・ルーチンのみのイベント条件検出および、トレースとなる
- ・ CALL命令からの復帰時、R1レジスタの戻り値でイベント条件検出
- ・ イベント条件検出で実行を終了し、カレント・バンクのレジスタを表示

(g) PC、R1レジスタの確認

PC	SP	PSW: UF	RBS2	RBS1	RBS0	S	Z	RSS	AC	IE	P/V	LT	CY
0109	FE00	0	0	0	0	0	0	0	0	0	0	0	0
R0	R1	R2	R3	R4	R5	R6	R7	RP4	RP5	RP6	RP7		
X	A	C	B					VP	UP	DE	HL		
00	02	00	00	00	00	00	00	0000	0000	0000	0000		

One step emulation standby
brk:0>

・ (g) で使用されているコマンドの説明

- ・ PC = 0109H、R1レジスタ = 2H

(h) NEST1 : からプロシージャ実行

```

brk:0>RUN T NEST1,R1=03H TRD PRC [Q]
User-system Vcc-OFF   Emulation start at 1000
< External data trace mode >
  Frame Status Address Data      Label Mnemonic      EX
                                PUBLICYNEST1::
0000          1000  B901          MOV     R1,#1H
0004          1002  2800200      CALL   !NEST2H
0009  WR      FDFF  05
0010  WR      FDFF  10          00
0013          1005  00          NOP
0016          1006  56          RET
0018  RD      FE00  0030          00
0022          3000
terminated
    
```

・ (h) で使用されているコマンドの説明

RUN T NEST1,R1=3H TRD PRC : プロシージャ実行の開始

- ・ イベント条件としてR1レジスタ= 3H を設定
- ・ カレント・ルーチンだけのイベント条件検出および、トレースとなる
- ・ カレント・ルーチンからの浅いネスト・レベルのルーチンへの移行は可能。引き続きイベント条件検出および、トレースを行う
- ・ カレント・バンクのレジスタを表示

(i) PC、R1レジスタの確認

PC	SP	PSW:	UF	RBS2	RBS1	RBS0	S	Z	RSS	AC	IE	P/V	LT	CY
1006	FE00		0	0	0	0	0	0	0	0	0	0	0	0
R0	R1	R2	R3	R4	R5	R6	R7		RP4		RP5		RP6	RP7
X	A	C	B						VP		UP		DE	HL
00	03	00	00	00	00	00	00		0000		0000		0000	0000
One step emulation standby														
brk:0>														

・ (i) で使用されているコマンドの説明

- ・ PC = 1006H、R1レジスタ = 03H

4. 2. 4 C0カバレッジ測定機能を用いたカバレッジ測定、測定結果の表示

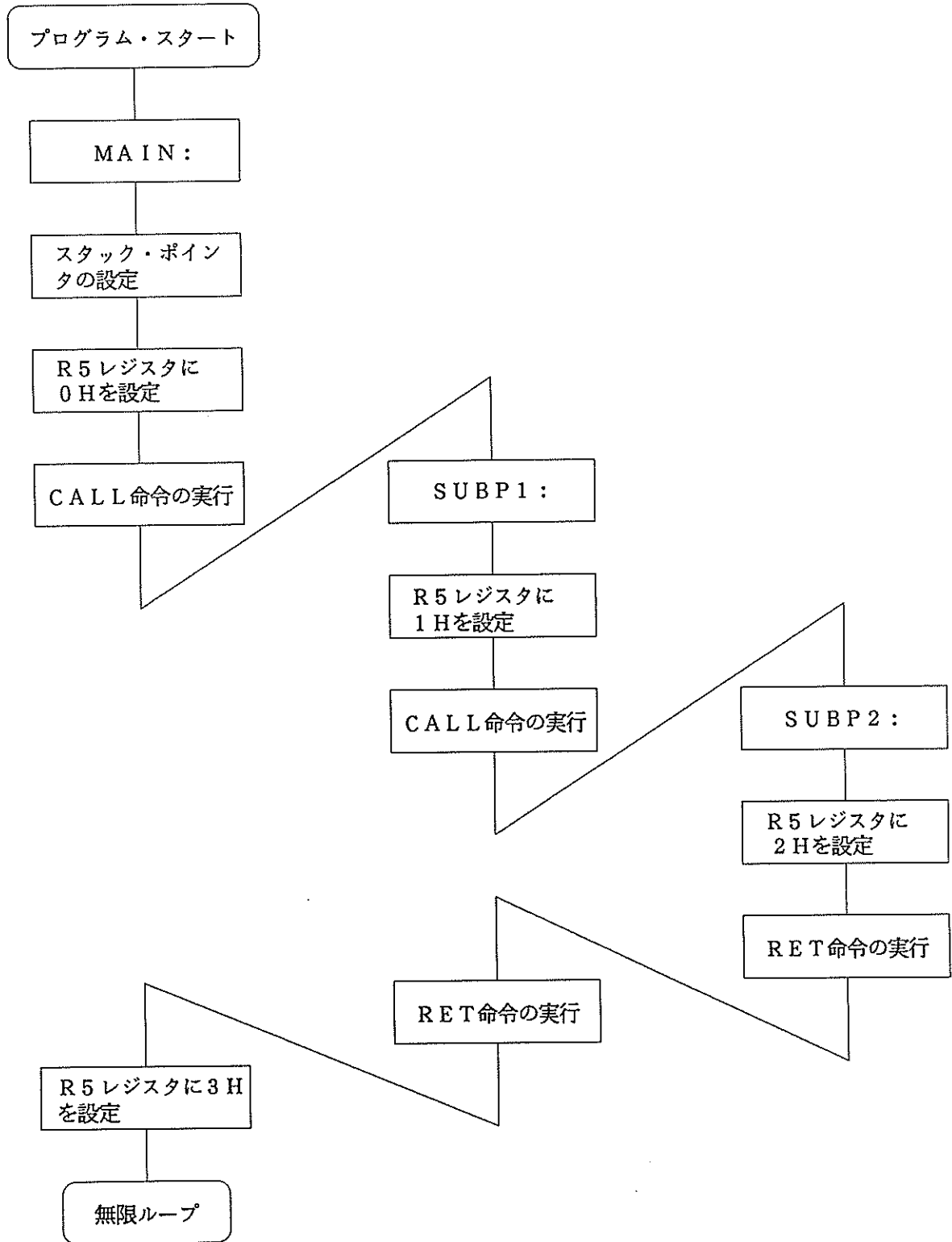
C0カバレッジ測定機能について、サンプル・プログラムのディバグ作業を例に説明します。

(1) サンプル・プログラムの概略、フロー・チャート、プログラム本体

サンプル・プログラムの概略

- ・ 00300H 番地よりのMAIN:でR5レジスタに0Hを設定、CALL命令を実行してSUBP1:へ移行。
- ・ 00380H 番地よりのSUBP1:でR5レジスタに1Hを設定、CALL命令を実行してSUBP2:へ移行。
- ・ 00480H 番地よりのSUBP2:でR5レジスタに2Hを設定、RET命令を実行してSUBP1:へ復帰。
- ・ SUBP1: (復帰先)でRET命令を実行してMAIN:へ復帰。
- ・ MAIN: (復帰先)でR5レジスタに3Hを設定、無限ループの実行。

サンプル・プログラムのフロー・チャート



サンプル・プログラム本体

```

MAIN:      ORG      300H
           MOVW    SP, #0FE00H      ←スタック・ポインタの設定
           MOV     R5, #0           ←R 5 レジスタに 0H を設定
           CALL   !SUBP1           ←深いネスト・レベルへ移行
           NOP
           MOV     R5, #3H         ←R 5 レジスタに 3H を設定
           BR     $$              ←無限ループ
           END

SUBP1:     ORG      380H
           MOV     R5, #1         ←R 5 レジスタに 1H を設定
           CALL   !SUBP2         ←SUBP2 : へ移行
           NOP
           RET                  ←MAIN : へ復帰

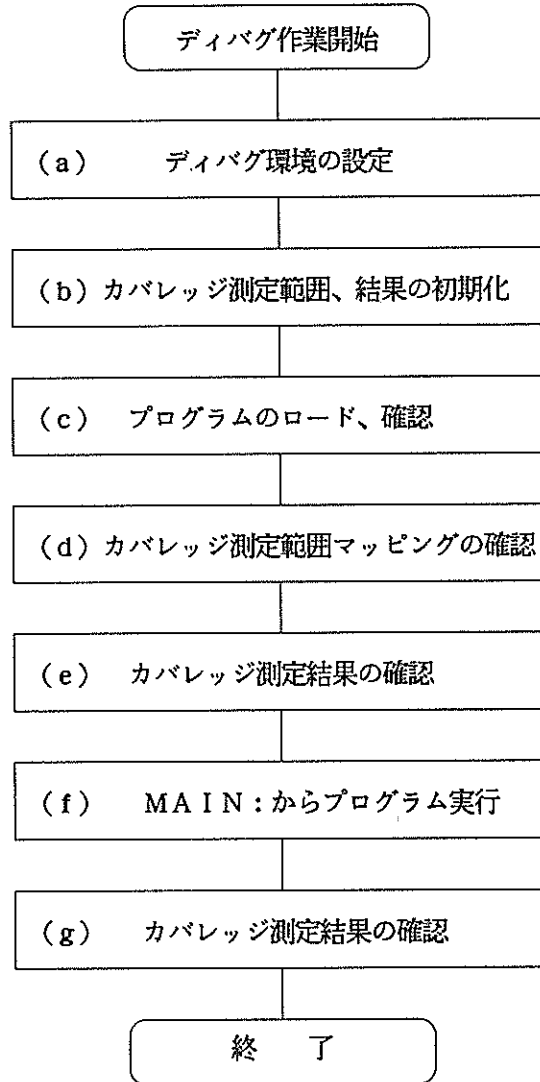
SUBP2:     ORG      400H
           MOV     R5, #2         ←R 5 レジスタに 2H を設定
           NOP
           RET                  ←SUBP1 : へ復帰
           END
    
```

(2) サンプル・プログラムに対するディバグ作業の概略、
フロー・チャート、画面表示例

ディバグ作業の概略

- ・ カバレッジ測定範囲と測定結果の初期設定
- ・ プログラム本体のロードによる、カバレッジ測定範囲のマッピング
- ・ マッピングされたカバレッジ測定範囲の確認
- ・ プログラム実行前の、カバレッジ測定結果の確認
- ・ プログラム実行後の、カバレッジ測定結果の確認

ディバグ作業のフロー・チャート



(3) 画面表示例

(a) デバッグ環境の設定

```
brk:0>CLK I [E]
brk:0>RES [E]
brk:0>OUT OFF [E]
brk:0>MAP I 16K [E]
brk:0>MAP [E]
0000 - 3FFF Internal ROM
4000 - 7FFF User
8000 - FC7F Non map
FC80 - FEFF <Internal RAM>
brk:0>WRD W [E]
brk:0>BRS 1 A=10C [E]
brk:0>BRM BRS1 [E]
brk:0>
```

・ (a) で使用されているコマンドの説明

- ・ CLK I : エミュレータ内クロックを選択
- ・ RES : エミュレーション・デバイスをリセット
- ・ OUT OFF : トリガ信号の外部出力を禁止
- ・ MAP I 16K : 内部ROMを16Kバイトに設定
- ・ MAP : マッピング状態の確認
- ・ WRD W : メモリ長をワードに設定
- ・ BRS 1 A=10C : プログラム実行検出の1を設定 (イベント・ポイント = 0010CH 番地)
- ・ BRM BRS1 : プログラム実行検出の1をトリガ条件に設定

(b) カバレッジ測定範囲、結果の初期化

```
brk:0>CVM K Ⓚ  
brk:0>
```

・ (b) で使用されているコマンドの説明

CVM K : カバレッジ測定範囲の指定解除

(c) プログラムのロード、確認

```

brk:0>LOD CVDPROG C S ⓐ
object load complete
symbol table loading
PUBLIC    load complete
brk:0>DAS MAIN ⓐ
Addr  object                Mnemonic
                                PUBLICYMAIN::
0300  0B FC 00 FE          MOVW    SP,#0FE00H
0304  BD 00                MOV     R5,#0H
                                .
                                .

brk:0>DAS SUBP1 ⓐ
Addr  Object                Mnemonic
                                PUBLICYSUBP1::
0380  BD 01                MOV     R5,#1H
0382  28 00 04            CALL   !SUBP2
                                .
                                .

brk:0>DAS SUBP2 ⓐ
Addr  Object                Mnemonic
                                PUBLICYSUBP2:
0400  BD 02                MOV     R5,#2H
0402  00                NOP
                                .
                                .

brk:0>

```

・ (c) で使用されているコマンドの説明

- ・ LOD CVDPROG C S : オブジェクト、シンボルをロード
- ・ DAS MAIN : プログラムの確認
- ・ DAS SUBP1 : プログラムの確認
- ・ DAS SUBP2 : プログラムの確認

(d) カバレッジ測定範囲マッピングの確認

```
brk:0>CVM D [C]
addr  000 100 200 300  400 500 600 700  800 900 A00 B00  C00 D00 E00 F00
0000
      . . .
1000
2000
3000
4000
5000
6000
7000
8000
9000
A000
B000
C000
D000
E000
F000
brk:0>
```

・ (d) で使用されているコマンドの説明

CVM D : カバレッジ測定範囲の表示

- ・ MAIN : が 00300H 番地から 0033FH 番地に、
SUBP1 : が 00380H 番地から 003BFH 番地
に、そしてSUBP2 : が 00400H 番地から
0043FH 番地に、それぞれマッピングされてい
る

(e) カバレッジ測定結果の確認

```
brk:0>CVD D 300,4FF 
  addr 00          1F 20          3F 40          5F 60          7F
0300 .....
0380 ....
0400 ..
0480
coverage 00.0%
brk:0>
```

・ (e) で使用されているコマンドの説明

CVD D 300,4FF : カバレッジ測定結果の表示

- ・ プログラム実行前なのでプログラム実行率は、00.0%

(f) MAIN : からプログラム実行

```

brk:0>RUN N MAIN [Q]
User-system Vcc-OFF   Emulation start at 0100
  (External data trace mode)
trc:0>
Execution break      terminated BRS1
emu:0>STP [Q]
Escape break        terminated
PC   SP  PSW: UF   RBS2 RBS1 RBS0 S   Z   RSS AC  IE  P/V LT  CY
010C FE00      0   0   0   0   0   0   0   0   0   0   0   0   0
R0   R1  R2  R3   R4   R5   R6   R7       RP4   RP5   RP6   RP7
X   A   C   B                VP   UP   DE   HL
55  02  FF  FD  4C  03  30  02       08FD  A600  29DB  FEFC
brk:0>
    
```

・ (f) で使用されているコマンドの説明

- ・ RUN N MAIN : ノンブレイク・リアルタイム実行の開始
- ・ STP : エミュレーションCPUの実行停止
- ・ エミュレーションCPUの実行停止時、カレント・バンクのレジスタを表示

(g) カバレッジ測定結果の確認

```
brk:0>CVD D 300,4FF (C)
addr 00          1F 20          3F 40          5F 60          7F
0300 *****
0380 ****
0400 **
0480
coverage 100.0%
brk:0>
```

・ (g) で使用されているコマンドの説明

CVD D 300,4FF : カバレッジ測定結果の表示

- ・ MAIN :、SUBP1 :、SUBP2 : の実行によりプログラム実行率が、100.0%と表示される

第5章 コマンド入力機能

この章では、I E - 7 8 3 2 7 - Rでのコマンドの入力方法、入力形式、制御キーについて説明します。

凡例

- _____ : 下線部分は、キーボードからの入力を表します。
- ☐ : リターン・キー (CR (ODH)) の入力を表します。
- <ESC> : エスケープ・キーの入力を表します。
- ^ : コントロール・キーを押しながら ^ の右側の文字を入力することを表します。
- ■ : カーソルを表します。
- このマニュアルで用いられている画面表示例、入力例はPC-9800シリーズをホスト・マシンとして使用した場合のものです。

5.1 IE-78327-Rのコマンド入力機能

IE-78327-Rを、PC-9800シリーズまたはIBM PCシリーズと接続して使用した場合のコマンド入力機能について説明します。

5.1.1 コマンド入力形式

コマンド入力形式には、次の2種類があります。

(1) 1ラインのコマンド入力

1ラインのコマンド入力形式は、コマンド本体、サブコマンド、オペランドをスペースで区切って最大128文字まで入力できます。

コマンド入力は、**␣** を入力して終了します。

␣ は、コマンド行のどこでも入力できます。

コマンド行の編集終了後、カーソルをコマンド行の最後まで移動する必要はありません。

<ESC> が入力された場合、入力中のコマンドを取り消します。

(2) 対話形式のコマンド入力

IE-78327-Rは、基本的に1ラインのコマンド入力形式となっています。しかし、BRA、MOD、TRFコマンドなど設定が複雑なコマンド、MEM C、PGM C、REG Cコマンドなどデータを逐次変更するコマンドについては対話形式でのコマンド入力も可能です。

␣ を入力して、データを逐次入力します。

<ESC> が入力された場合、それまで入力したデータを有効としてコマンド入力を終了します。

例1. 対話形式でのコマンド入力

```
brk:0>MOD [Enter]          ←コマンド本体のみ入力します
Mode CHAR = FLOW [Enter]
Boud 9600 = 4800 [Enter]
Long 8 = 7 [Enter]         ←対話形式での入力です
Par NON = EVEN [Enter]
Stop 2 = 1 [Enter]
brk:0>
```

2. 対話形式でのコマンド入力の終了

```
brk:0>MEM C 500 [Enter]
0500 32 = 45 [Enter]
0501 33 = <ESC>          ←コマンド入力を終了します
brk:0>
```

5. 1. 2 コマンドの入力方法

IE-78327-Rのコマンド入力方法には、次の4種類があります。

- ・キー入力によるコマンド入力
- ・ファイルからのコマンド入力
- ・コマンドの省略形入力
- ・コマンド行の再呼び出し

(1) キー入力によるコマンド入力

ⓐ を入力して、コマンド入力を終了します。

<ESC> を入力して、コマンド入力の取り消しまたは終了をします。

- ・ コマンド入力の取り消し : 1 ラインのコマンド入力で、<ESC> が入力された場合
- ・ コマンド入力の終了 : 対話形式のコマンド入力で、<ESC> が入力された場合

キー入力でコマンドを入力する場合、コマンド行の範囲で、次の編集機能を持ちます。

(a) 一文字削除 (DELキー、BSキー)

カーソル位置の直前の一文字を削除します。

(b) 一行削除 (^Xキー)

入力中のコマンド行すべてを削除します。

(c) 置き換え

カーソル位置の文字を、キー入力した文字と置き換えます。

(d) 挿入 (^Aキー)

カーソル位置へ、キー入力する文字列を挿入します。

文字列の挿入は、カーソル位置に < が表示されている挿入モードの場合に有効となります。挿入モードは、^Aキー入力によるトグル制御となっています。

(2) ファイルからのコマンド入力

あらかじめコマンドが入力されているファイル（COMコマンドやエディタなどで入力したもの）からコマンドを入力できます。

ファイルからコマンドを入力する場合、STRコマンドを使用します。
ファイルからの入力が終了すると、以降の入力はキーボードからになります。

例 指定ファイルからのコマンド入力

- ・ ファイル名 TEST.STR
- ・ ファイル内容 MAP R 2000,3FFF
MAP W 4000,5FFF
.
.
.
MEM D 300X

```
brk:0>STR TEST.STR  
brk:0>MAP R 2000,3FFF  
brk:0>MAP W 4000,5FFF  
.  
.  
.  
brk:0>MEM D 300X  
brk:0>
```

← TEST.STR ファイルからのコマンド入力を指定します

← TEST.STR ファイルより入力されたコマンド

(3) コマンドの省略形入力

表5-1に示すコマンドは、先頭の一文字だけの省略形入力が可能です。
 コマンド本体のみ入力の場合、先頭の一文字入力後 Ⓔ を入力します。
 オペランドを指定する場合、先頭の一文字入力後にスペース・キーを入力するとコマンド本体表示後オペランドの入力待ちとなります。

表5-1 省略形一覧

コマンド	省略形	コマンド	省略形	コマンド	省略形
ASM	A	HLP	H	SAV	S
CLK	C	LOD	L	TRD	T
DAS	D	MEM	M	VRV	V
EXT	E	RUN	R		

例1. CLK を省略形で入力する場合

```
brk:0>C  $\text{Ⓔ}$       ←先頭の一文字  $\text{Ⓔ}$ を入力すると、コマンド本体表示後に
                  実行されます
      ↓
brk:0>CLK  $\text{Ⓔ}$ 
  Internal
brk:0>
```

2. CLK U を省略形で入力する場合

```
brk:0>C <ス^ -ス・キー> ←先頭の一文字入力後、スペース・キーを入力すると、
                        コマンド本体表示後にオペランドの入力待ちとなります
      ↓
brk:0>CLK ■注          ←この状態でオペランドを入力します
brk:0>CLK U  $\text{Ⓔ}$         ←オペランド入力後に実行します
brk:0>
```

注 カーソル位置を示します

(4) コマンド行の再呼び出し

IE-78327-Rは、最新のコマンド行を20行記憶します。
 コマンド行の再呼び出しを行う場合、!コマンド番号 @ と入力します。

直前に入力したコマンドを呼び出す場合、!!@ と入力します。

記憶されているすべてのコマンド行とコマンド番号を表示する場合、HIS @ を入力します。

例1. 記憶されているすべてのコマンドとコマンド番号を表示

```
brk:0>HIS @ ←
  1 MEM F OXXX 0
  2 MEM D 0,0FFH
  .
  .
  .
 20 HIS ← ←直前に入力したコマンド
brk:0>
```

2. 2番のコマンド行の呼び出し

```
brk:0>!2 @
MEM D 0,0FFH
```

コマンドを実行する場合、この後に @ を入力します。
 コマンド行を修正する場合、制御キーを用いて編集します。

3. 直前に入力したコマンド行の呼び出し

```
brk:0>!! @
HIS
```

5. 1. 3 制御キー

IE-78327-Rで、使用可能な制御キーを次に示します。

表5-2 制御キー

制御キー	機 能	制御キー	機 能
^ A	挿入モード・トグル・スイッチ	^ X	一行削除
^ C	コントロール・プログラム強制終了	B S	一文字削除
^ H	一文字削除	C R	行入力の終了
^ I	スペースと同じ	D E L	一文字削除
^ J	行入力の終了	E S C	コマンド実行の終了
^ K	S T Rコマンドの強制終了	L F	行入力の終了
^ L	S T Rコマンドの一時停止と再開	T A B	スペースと同じ
^ M	行入力の終了	;	コメントの開始
^ O	C O Mコマンドの出力スイッチ	!	ヒストリの呼び出し
^ P	L S Tコマンドの出力スイッチ	←	カーソル左移動
^ Q	^ S による一時停止の解除	→	カーソル右移動
^ S	一時停止		

第6章 数値、シンボル、式の記述仕様

この章では、コマンド入力の際に使用する数値、シンボル、式の記述仕様について説明します。

6.1 数値の記述仕様

数値の記述仕様には、次の2種類があります。

(1) ビット長

次の4つの中から、使用するコマンドの中で規定されているビット長の数値を選択してください。

- ・ 16ビット
- ・ 8ビット
- ・ 4ビット
- ・ 1ビット

数値の最上位桁は、数字(0-9)を記述します。

数値の前に符号(+、-)を付加する場合、符号の次は数字(0-9)を記述します。

(2) 基数

数値の最後に次に示す4種類の符号を付加して、基数を指定できます。

- ・ H : 16進数
- ・ T : 10進数
- ・ Q : 8進数
- ・ Y : 2進数

通常、基数は各コマンドごとにその機能に応じて規定されています。

基数を示す符号が省略された場合、規定された基数で数値は処理されます。

規定外の基数を用いて数値を入力する場合、基数を示す符号を付加します。

各コマンドに応じた基数の設定は **7.3 コマンドの一覧** を参照してください。

数値の大きさと基数の対応を次に示します。

表6-1 数値の大きさと基数の対応

		基 数	
		16 進 数	10 進 数
数 値 の 大 き さ	最 小	0	0
	最 大	32ビット 0FFFFFFFH 16ビット 0FFFFH 8ビット 0FFH 4ビット 0FH 1ビット 1H	4294967295T 65535T 255T 15T 1T

		基 数	
		8 進 数	2 進 数
数 値 の 大 き さ	最 小	0	0
	最 大	32ビット 3777777777Q 16ビット 177777Q 8ビット 377Q 4ビット 17Q 1ビット 1Q	11111111111111111111111111111111Y 1111111111111111Y 1111111Y 1111Y 1Y

6.2 特殊数値の記述仕様

特殊数値とは、複数の数値の集まりを表します。

特殊数値を記述する場合、任意の数値を表現する X を使用します。

X は、16進数では 0H-FH に、8進数では 0Q-7Q に、2進数では 0Y と 1Y にそれぞれ対応します。

注意 10進数の X 表現はできません。

特殊数値の記述仕様には、次の2種類があります。

(1) 数値範囲の記述

X を使用して数値範囲を記述する場合、最下位から連続して上位桁へ X を記述します。

最上位桁まで連続して X を記述する場合、X の前に数値であることを示すために 0 を付加します。

例 数値範囲の記述

```

0XXXXH    → 0H-0FFFFH
0XXXXXXQ  → 0Q-177777Q
0XXXXXXXXY → 0Y-1111111Y
    
```

(2) マスク・データの記述

X を使用して、マスク・データを記述します。

最上位桁に X を記述する場合、X の前に数値であることを示す 0 を付加します。

例 マスク・データの記述

```

0X00H    → 0000H, 0100H, 0200H, 0300H, 0400H, 0500H, 0600H, 0700H,
           0800H, 0900H, 0A00H, 0B00H, 0C00H, 0D00H, 0E00H, 0F00H
01X1Q    → 0101Q, 0111Q, 0121Q, 0131Q, 0141Q, 0151Q, 0161Q, 0171Q
1X1010X1Y → 10101001Y, 11101001Y, 10101011Y, 11101011Y
    
```

6.3 シンボルの記述仕様

シンボルは数値の代わりに使用します。

(1) シンボル名 (構成文字、有効文字数)

シンボル名を構成する文字は、次に示す文字のみとなります。

A-Z a-z @ ? _ 0-9

先頭の文字は、数字0-9以外でなければなりません。
英小文字(a-z)は、英大文字(A-Z)として扱います。

シンボル名の有効文字数は、最大8文字です。
8文字以上の文字が記述された場合、先頭から8文字が有効となります。

(2) シンボル値 (有効桁数)

シンボル値は、16ビット・データとして扱われます。

(3) モジュール名

次の3種類のシンボルに付記するモジュール名は、それぞれ記述方法が異なります。

(a) パブリック・シンボル

パブリック・シンボルを記述する場合、シンボル名のみを記述します。

注

モジュール名(PUBLICY)を記述する必要はありません。

注 卒は、IBM PCシリーズをホスト・マシンとして使用した場合は \ と
なります。

(b) ローカル・シンボル

ローカル・シンボルを記述する場合、シンボル名の前にモジュール名を付加します。カレント・モジュールが指定 (SYM M コマンド) されている場合、モジュール名を省略してシンボル名のみ記述します。

パブリック・モジュールに同一シンボル名が存在する場合、パブリック・シンボルが選択されます。

詳細は、8.42 シンボルの操作 の項を参照してください。

(c) I E シンボル

I E シンボルを記述する場合、シンボル名のみを記述します。

注

モジュール名 (IESYMBOLY) を記述する必要はありません。

注 卒は、I B M P C シリーズをホスト・マシンとして使用した場合は \ となります。

(4) 最大シンボル数 (登録可能なシンボル数)

I E - 7 8 3 2 7 - R で記述 (登録) 可能な最大シンボル数は、次の2種類のシンボルの合計で約2000です。

(a) シンボル・テーブル・ファイル

(b) I E シンボル

たとえば、シンボル・テーブル・ファイルに1500シンボルを登録すると、I E シンボルには500シンボルまでが登録可能です。

6.4 式の記述仕様

数値と数値、シンボルとシンボル、数値とシンボルを演算子で結合した式の記述仕様について説明します。

式の記述仕様には、次の3種類の項目があります。

(1) 演算子

式には、次の演算子を使用できます。

カッコの使用は、最大32レベルまでとなります。

- | | |
|----------|------|
| ・ () | ↑ 高 |
| ・ *、/ | 優先順位 |
| ・ +、- | |
| ・ AND | ↓ 低 |
| ・ OR、XOR | |

(2) 表現方法

数値、シンボルとAND、OR、XORの各演算子を記述する場合、1つ以上のスペースが前後に必要です。

例 SYM XOR 10101100B

数値、シンボルと()、*、/、+、-の各演算子を記述する場合、スペースを挿入する必要はありません。

例 (SYM2+36H)

(3) 演算処理

演算は、すべて16ビットの整数で行われます。

演算の中間結果、最終結果が16ビット以上になった場合、17ビット目より上位は切り捨てます。

演算は、数値またはシンボルに対してだけ有効であり、予約語あるいは特殊数値に対しての演算はエラーとなります。

第7章 コマンドの概要

この章では、I E - 7 8 3 2 7 - Rで使用するコマンドの構造、用語の定義について説明し、コマンドの一覧と初期値の表を示します。

凡例

- □ : リターン・キー (CR (0DH)) の入力を表します。
- このマニュアルで用いられている画面表示例、入力例はPC-9800シリーズをホスト・マシンとして使用した場合のものです。

7.1 コマンドの構造

IE-78327-Rのコマンドは、次の3種類の項目をスペースで区切って構成します。

(1) コマンド本体

3文字で構成されます。

(2) サブコマンド

コマンド本体の処理の内容を指定します。
コマンドによっては、サブコマンドがない場合もあります。

(3) オペランド

各コマンドによって異なります。サブコマンドがあるコマンドは、サブコマンドを省略してオペランドを入力することはできません。

- コマンドの構造を次に示します。


注

b r k : 0 >	コマンド本体	サブコマンド	オペランド
↑	↑	↑	↑
プロンプト	スペース	スペース	スペース

注 プロンプトはコマンドには含まれません。

7.2 コマンド用語の定義


説明に使用する用語を次に示します。

(1) 

かっこの中に記述されている文字列の選択を表します。

(2) []

かっこの中に記述されている文字列が省略できることを表します。

(3) 

点線のかっこの中は、サブコマンドまたはオペランドの説明です。

(4) **w o r d**

数値表現の16ビット数値を表します。 基数：16進数 [H]

(5) **p a s s 8**

8ビットのパス・カウント数を表します。 基数：10進数 [T]

(6) **p a r t i t i o n**

特殊数値を表します。 基数：16進数 [H]

(7) **m a s k 4**

4ビットの数値、マスク・データを表します。 基数：2進数 [Y]

(8) **m a s k 8**

8ビットの数値、マスク・データを表します。 基数：16進数 [H]

(9) **mask16**

16ビットの数値、マスク・データを表します。 基数：16進数 [H]

(10) **data-string**

複数のバイト・データの集まりを表し、最大10個までの集まりを表現できます。

基数：16進数 [H]

(11) **bit**

数値表現の1ビット数値を表します。

基数：2進数 [Y]

(12) **step16**

16ビットの実行ステップ数を表します。

基数：10進数 [T]

(13) **point**

13ビットのサンプル・フレーム・ポインタ値を表します。

基数：10進数 [T]

(14) **number**

1から10000のサンプル・タイミングを表します。

基数：10進数 [T]

(15) **expression**

式を表します。

基数：16進数 [H]

(16) **parameter**

パラメータを表わします。

(17) register name

レジスタ名を表します。なお、特殊機能レジスタ (SFR) は (19) に示します。

レジスタ名には次のものがあります。

レジスタの種類	レジスタ名
プログラム・カウンタ	PC
スタック・ポインタ	SP
プログラム・ステータス・ワード	PSW
8ビット・ジェネラル・レジスタ	R0、R1、R2、R3、R4、 R5、R6、R7、R8、R9、 R10、R11、R12、R13、 R14、R15
16ビット・ジェネラル・レジスタ	RP0、RP1、RP2、RP3、 RP4、RP5、RP6、RP7
8ビット・インプライド・レジスタ	X、A、B、C、D、E、H、L
16ビット・インプライド・レジスタ	AX、BC、VP、UP、DE、 HL

(18) PSW flag name

PSWのフラグ名には次のものがあります。

フラグの種類	フラグ名
ユーザ・フラグ	U F
レジスタ・バンク選択フラグ2	R B S 2
レジスタ・バンク選択フラグ1	R B S 1
レジスタ・バンク選択フラグ0	R B S 0
サイン・フラグ	S
ゼロ・フラグ	Z
レジスタ・セット選択フラグ	R S S
補助キャリー・フラグ	A C
割り込み許可フラグ	I E
パリティ/オーバフロー・フラグ	P / V
割り込みレベル・トランジェント・フラグ	L T
キャリー・フラグ	C Y

(19) s f r name

S F R名を表します。

(a) μ PD78320シリーズのs f r

- 読み書き可能なS F R名を次に示します。

P0	P3	P4	P5	P8	P9	CCX0LW	CC01LW	CCX0UW
CC01UW	RTPS	BRG	RTP	RTPR	PRDC	ADM	CM00	CM01
CM02	CM03	CM10	CM11	CSIM	SBIC	SIO	ASIM	TMC
BRGM	PRM	TOC0	TOC1	RPUM	STBC	CCW	WDM	MM
PWC	FCC	IF0	IF1	MK0	MK1	PB0	PB1	ISMO
ISM1	CSE0	CSE1	INTM0	INTM1	PRSL			

EXTSFR0	EXTSFR1	EXTSFR2	EXTSFR3	EXTSFR4	EXTSFR5	EXTSFR6	EXTSFR7
EXTSFR8	EXTSFR9	EXTSFR10	EXTSFR11	EXTSFR12	EXTSFR13	EXTSFR14	EXTSFR15

- 読み出し専用のS F R名を次に示します。

P2	P7	TM0LW	CTX0LW	CT01LW	CT02LW	CT03LW	TMOUW	TM1
CTX0UW	CT01UW	CT02UW	CT03UW	ADCR	ADCRH	ASIS	RXB	ISPR

- 書き込み専用のS F R名を次に示します。

PM0	PM3	PM5	PM8	PM9	PMC0	PMC3	PMC8	TXS
-----	-----	-----	-----	-----	------	------	------	-----

(b) μ PD78327シリーズのsfr

- 読み書き可能なSFR名を次に示します。

P0	P3	P4	P5	P8	P9	CC10	BRG	P0L
POH	PRDC	RTPC	PWMC	PWMB	ADM	CM00S	CM01S	CM02S
CM03S	CM04S	CM05S	CM06	CM20	CSIM	SBIC	SIO	ASIM
CM00R	CM01R	CM02R	CM03R	CM04R	CM05R	TMC0	BRGM	TMC1
TUM	TOUT	STBC	CCW	WDM	MM	PWC	FCC	IF0
IF1	MK0	MK1	PB0	PB1	ISM0	ISM1	CSE0	CSE1
INTM0	PRSL							

EXTSFR0	EXTSFR1	EXTSFR2	EXTSFR3	EXTSFR4	EXTSFR5	EXTSFR6	EXTSFR7
EXTSFR8	EXTSFR9	EXTSFR10	EXTSFR11	EXTSFR12	EXTSFR13	EXTSFR14	EXTSFR15

- 読み出し専用のSFR名を次に示します。

P2	P7	TM2	TM0	TM1	ADCR	ADCRH	ASIS	RXB
ISPR								

- 書き込み専用のSFR名を次に示します。

PM0	PM3	PM5	PM8	PM9	PMC3	PMC8	TXS
-----	-----	-----	-----	-----	------	------	-----

(20) `symbol`

シンボル名を表します。

(21) `file`

ファイル名を表します。

(22) `module name` ¥ **注**

シンボル・テーブルのモジュール名を表します。

注 ¥は、IBM PCシリーズをホスト・マシンとして使用した場合 `\` となります。

(23) `command`

コマンド本体を表します。

7.3 コマンドの一覧

(1) コマンドの一覧表の見方

IE-78327-Rで使用する全コマンドの一覧表を次頁以降に示します。
各コマンドにおいて、次に示すシステム構成または動作状態の違いにより、
入力できるコマンド、入力できないコマンドをそれぞれ○と×で表しています。

IE-78327-Rのコマンドは基本的に1ラインのコマンド入力形式となっています。

しかし、次に示すコマンドは対話形式でのコマンド入力も可能です。

- 設定が複雑なコマンド
 - ・ BRA、MOD、TRF コマンド
- データを逐次変更するコマンド（対話形式での入力のみ）
 - ・ ASM、MEM C、PGM C、REG C、SFR C コマンド

注意 コマンド一覧表は1ラインのコマンド入力形式で示しています。

IE-78327-Rのコマンドに設定されている初期値の一覧表を、コマンドの一覧表のあとに示しています。

コマンドの種類	コマンド本体	サブコマンド	オペレータ	コンド	デフォルト	動作状態		
						trc:	emu:	brk:
ライン・アセンブル	ASM	なし	[word]	word アセンブルの開始アドレス	0H	×	×	○
各イベント検出の条件設定	バス・イベント検出の条件設定	$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$	[A=mask16][Y=mask8][C=status][B=mask4]	<div style="border: 1px dashed black; padding: 5px;"> mask16 16ビット検出アドレス範囲 (最大2箇所) mask8 8ビット検出データ mask4 4ビット外部データ status 検出ステータス </div>	A=0XXXXH Y=0XXH C=NC B=0XXXXY		×	○
	外部データ検出の条件設定	なし	[mask4]	<div style="border: 1px dashed black; padding: 5px;"> mask4 外部センス・クリップNo.1-4の信号レベル </div>	0XXXXY	×	○	○
	プログラム実行条件設定	BRS	$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$	[A=word]	<div style="border: 1px dashed black; padding: 5px;"> word 検出アドレス </div>	A=0H	×	○

注 status は、下表の中から選んでください。

OP	RW	R	W	RWP	RP	WP	RWN	RM	WN	NC
----	----	---	---	-----	----	----	-----	----	----	----

コマンドの種類	コマンド 本体	サブ コマンド	オプション コマンド	デフォルト	動作状態		
					trc:	emu:	brk:
トリガ条件の設定	B R M	なし	[BRA1][BRA2][BRA3][BRA4][BRD][BRS1][BRS2][BRS3][BRS4] [-OFF]	BRA1	×	○	○
チェック・ポイント 条件の設定	C H K	なし	[BRA1][BRA2][BRA3][BRA4][BRD][BRS1][BRS2][BRS3][BRS4] [-OFF]		×	○	○
クロックの選択	C L K	なし	[I] [U]	I	×	×	○
実行経過時間、実行 命令数の表示	C N T	なし	なし	なし	×	○	○
コマンド・ファイル の作成	C O M	なし	[file] [LST: CON:]	CON:	○	○	○

BR? 各トリガ条件
OFF 各トリガ条件の解除

REG
sfr
partition

各チェック条件
設定解除
レジスタ指定
sfr名指定
(最大5つまで)
内部RAM範囲指定

I エミュレータ内クロック
U ユーザ設定クロック

file 7桁#名
LST: 7桁#
CON: 3桁#

コマンドの種類	コマンド 本体	サブ コマンド	オ ペ ラ シ ョ ン ド	デ ィ フ ォ ー ルト	動 作 状 態		
					trc:	emu:	brk:
C O カ バ レ ッ ジ 測 定 結 果 の 表 示	測定結果の 表示	注 D	[partition]	なし	×	×	○
	測定結果の 初期化	K	[partition]	なし	×	×	○
C O カ バ レ ッ ジ 測 定 範 囲 の 操 作	測定範囲の 追加	A	[partition]	なし	×	×	○
	測定範囲の 表示	注 D	[partition]	なし	×	×	○
	測定範囲の 指定解除	K	[partition]	なし	×	×	○
逆 ア セ ン ブ ル	DAS	なし	[^{word} [-partition]]	0H	×	×	○
ディ レ ク ト リ の 表 示	DIR	なし	[file]	カレント・ドラ イブのディレ クトリ名	○	○	○
トリ ガ 、 ポ イ ン ト 位 置 の 設 定	DLY	なし	[^F [-M] L]	F トリガ・ポイント位置を M トリガ・ポイント位置を L トリガ・ポイント位置を	×	○	○

注 コマンド本体のみを入力した場合には、このサブコマンドが指定されたものとして実行します。

コマンドの種類	コマンド本体	サブコマンド	オペランド	モード	ディフォルト	動作状態		
						trc:	emu:	brk:
子プロセスの実行	DOS	なし	なし	なし	なし	○	○	○
ディスプレイエーブル条件の設定	DSB	なし	$\left[\begin{array}{l} \text{[BRA1][BRA2][BRA3][BRA4][BRD][BRS1][BRS2][BRS3][BRS4]} \\ \text{[OFF]} \end{array} \right]$	$\left[\begin{array}{l} \text{BR? 各ディスプレイエーブル条件} \\ \text{OFF 各ディスプレイエーブル条件} \\ \text{の解除} \end{array} \right]$	OFF	×	○	○
イネーブル条件の設定	ENB	$\left[\begin{array}{l} 1 \\ 2 \\ 3 \end{array} \right]$	$\left[\begin{array}{l} \text{[BRA1][BRA2][BRA3][BRA4][BRD][BRS1][BRS2][BRS3][BRS4]} \\ \text{[ON]} \end{array} \right]$	$\left[\begin{array}{l} \text{BR? 各イネーブル条件} \\ \text{ON 各イネーブル条件の解除} \end{array} \right]$	ON	×	○	○
イベント検出器の設定状態表示	EVN	なし	なし	なし	なし	○	○	○
コントロール・プログラムの終了	EXT	なし	なし	なし	なし	×	×	○
コマンド・ヒストリの表示	HIS	なし	なし	なし	なし	○	○	○
コマンド・ヘルプの表示	HLP	なし	[command]	$\left[\begin{array}{l} \text{command} \\ \text{command 本体} \end{array} \right]$	なし	○	○	○

コマンドの種類	コマンド 本体	サブ コマンド	オペランド	デフォルト	動作状態		
					trc:	eau:	brk:
オブジェクト、シンボル、ディバグ環境のロード	LOD	なし	注1 file[module nameY][D][C][S][SV]	なし	×	×	○
ファイルへの結果出力	LST	なし	[file ┌ LST: ─┐ └ CON: ─┘]	CON:	○	○	○
マッピングの設定	MAP	I	注2 partition ┌──┐ └──┘ 0K 8K 16K 24K 32K 40K 48K 56K	なし	×	×	○
		[T W R U K]	注2 [partition]	I Eミュレーション内部ROM partition マッピング範囲	なし	×	×

注1. IBM PCシリーズをホスト・マシンとした場合は、\ととなります。

2. 8K単位で設定できます。

コマンドの種類	コマンド 本体	サブ コマンド	オペランド	デフォルト	動作状態		
					trc:	emu:	brk:
演算	MAT	なし	expression	なし	○	○	○
メモリ内容の変更	MEM	C	[word]	0H	×	○	○
メモリ内容の表示	MEM	注 D	[word partition →]	0H	×	○	○
メモリの検査	MEM	E	[partition]	なし	×	×	○
メモリ内容の初期化	MEM	F	partition data-string	なし	×	×	○
メモリ内容の検索	MEM	G	partition data-string	なし	×	×	○
メモリ内容の複写	MEM	M	partition word	なし	×	×	○
メモリ内容の比較	MEM	V	partition word	なし	×	×	○
メモリ内容の交換	MEM	X	partition word	なし	×	×	○

注 コマンド本体のみを入力した場合には、このサブコマンドが指定されたものとして実行します。

コマンドの種類	コマンド本体	サブコマンド	オペラント	デフォルト	動作状態			
					trc:	emu:	brk:	
チャンネル2のモード設定	MOD	なし	<p>[MODE= [CHAR -FLOW-]] [BAUD= 19200 9600 4800 2400 1200 600 300]] [LONG= [7 8]] [PAR= [NON EVEN- ODD-]] [STOP= [1 2]]</p> <p>ハンドシェイク モード</p> <p>ポート</p> <p>キャラクタ長</p> <p>パリティ・ビット</p> <p>ストップ・ビット</p>	MODE= CHAR BAUD= 9600 LONG=8 PAR=NON STOP=2	×	○	○	
代替メモリ←→ユーザ・メモリ間のデータ転送	MOV	[U I]	partition word [SV]	<p>代替メモリ←ユーザ・メモリ ユーザ・メモリ←代替メモリ 転送先開始アドレス 転送元アドレス範囲 ペリフェイ指定</p>	なし	×	×	○
トリガ信号の外部出力指定	OUT	なし	[OFF ON]	OFF 外部センス・クリップのN0.1をトレース信号入力とします ON 外部センス・クリップのN0.1をトリガ信号出力とします	OFF	×	○	○
バス条件の設定	PAS	なし	pass8	pass8 バスカウント数	1T	×	○	○
PROMプログラムの制御、キャラクタの変更、解除	PGM	C	C	C 制御キャラクタの変更、解除の指定	なし	×	×	○
サンプリング・アドレスの設定	PSA	なし	[word][word][word]	word サンプル・アドレス	なし	×	○	○

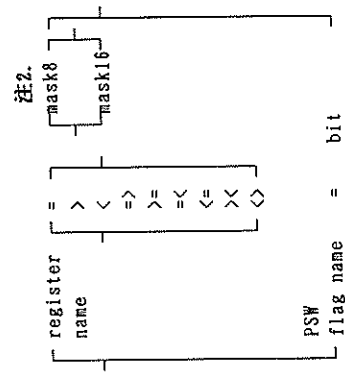
コマンドの種類	コマンド 本体	サブ コマンド	オ ペ ラ シ ョ ン ド	デ ィ ア フ ォ ー ム	動 作 状 態		
					trc:	enu:	brk:
サンプル・データの表示	PSD	なし	<div style="text-align: center;"> $\left[\begin{array}{c} \text{ALL} \\ \text{P} \\ \text{L} \\ \text{T} \end{array} \right] \left[\text{point} \right]$ $\left[\begin{array}{c} \text{SY} \\ \text{SB} \\ \text{SW} \end{array} \right]$ </div> <p>ALL point サンプル・データすべての表示 サンプル・ポイント移動数 サンプル・ポイントをサンプル・メモリの最初に設定 サンプル・ポイントをサンプル・メモリの最後に設定 サンプル・ポイントをイベント検出点に設定 SY ヒット表示指定 SB バイト表示指定 SW ワード表示指定</p>	なし	×	○	○
サンプル・タイミン グの設定	PST	なし	<div style="text-align: center;"> $\left[\begin{array}{c} \text{number} \\ .4 \\ .6 \\ .8 \end{array} \right] \left[\text{サンプル・タイミング} \right]$ </div> <p>number サンプル・タイミング .4 0.4 μsec .6 0.6 μsec .8 0.8 μsec</p>	.4	×	○	○
レジスタの レ ジ ス タ の 操 作	REG	C	<div style="text-align: center;"> $\left[\text{PSW flag name} \right] \left[\text{register name} \right]$ </div> <p>PSW flag name PSWフラグ名 register name レジスタ名</p>	R0	×	○	○
レジスタの表示	REG	注 D	<div style="text-align: center;"> $\left[\begin{array}{c} \text{ALL} \\ \text{register name} \\ \text{PSW flag name} \end{array} \right] \left[\text{レジスタの全レジスタ指定} \right]$ </div> <p>ALL 全レジスタ・バンクの全レジスタ指定 register name レジスタ名 PSW flag name PSWフラグ名</p>	カレント・バンクの全レジスタ	×	○	○

注 コマンド本体のみを入力した場合には、このサブコマンドが指定されたものとして実行します。

コマンドの種類	コマンド 本体	サブ コマンド	オ ペ ラ シ ョ ン ド	ダイフォ ールト	動 作 状 態		
					trc:	emu:	brk:
IE-78327-R、エミュレーションCPUのリセット	RES	なし	[H]	なし	○	○	○
ノンブレイク・リアルタイム実行	RUN	N	[word]	0H	×	×	○
ブレイク条件付きリアルタイム実行	RUN	B	[word]	0H	×	×	○
ステップ実行	RUN	T	[word][, -][注1] [step16]] [REG] [TRD]	なし	○	○	○
プロシージャ実行	RUN	T	[word][, -][注1] [step16]] [REG] [TRD] PRC				

mask8 8ビット・マスク・データ
mask16 16ビット・マスク・データ
bit 1ビットの数値

注1. 下記の表に置き換えてください。



2. マスク表現の場合は、=, >, <, <> のみ有効

コマンドの種類	コマンド 本体	サブ コマンド	オ ペ ラ ト ラ ン ド	デ ィ フ ォ ー ルト	動 作 状 態		
					trc:	emu:	brk:
オブジェクト、 デバッグ環境の セーブ	S A V	なし	注1 file [partition] [C] [D] [SV] file オブジェクト指定 デバッグ環境指定 セーブ・アドレス範囲 ベリファイ指定	なし	×	×	○
					×	○	○
S F R の 操 作	S F R	C	[sfr name] sfr name sfr名	P 0	×	○	○
	S F R	注2 D	[sfr name] sfr name sfr名	なし	×	○	○
実 行 停 止	S T P	なし	[T] T アナライザのみ停止	なし	○	注3 ○	×
ファイルからの コマンド入力	S T R	なし	file [parameter] file ファイル名 parameter 実パラメータ	なし	○	○	○

- 注1. partition は、5 つまで指定可能です。
 2. コマンド本体のみを入力した場合には、このサブコマンドが指定されたものとして実行します。
 3. T 指定は不可能です。

コマンドの種類	コマンド 本体	サブ コマンド	オ ペ ラ ト ラ ン ド	デ ィ ア フ ォ ー ム	動 作 状 態		
					trc:	emu:	brk:
I E シ ン ボ ル の 操 作	S Y M	A	symbol word	なし	×	○	○
			symbol word シンボル名 シンボル値				
	S Y M	C	symbol word	なし	×	○	○
			symbol word シンボル名 シンボル値				
	S Y M	E	symbol	登録され ている全 IESYMBOL	×	○	○
			symbol シンボル名				
	S Y M	L	なし	なし	×	○	○
	S Y M	S	なし	なし	×	○	○
	S Y M	K	なし	登録され ている全 シンボル	×	○	○
	S Y M	注1 D	注2 [module name]	登録され ている全 シンボル	×	○	○
			module name モジュール名				
	S Y M	M	なし	なし	×	○	○

注1. コマンド本体のみを入力した場合には、このコマンドが指定されたものとして実行します。
 2. IBM P Cシリーズをホスト・マシンとした場合 \ となります。

コマンドの種類	コマンド 本体	サブ コマンド	オ ペ ラ シ ョ ン ド	デ ィ ア フ ォ ー ム ト	動 作 状 態		
					trc:	emu:	brk:
トレース・データの表示	TRD	[F] [I]	フレーム表示 インストラクション表示 トレース・データすべて、検索条件に一致するデータすべての表示を指定 SF 検索条件に一致するフレームの前後5行の表示を指定 SJ プログラムの分岐処理に関するフレームの表示を指定 SQ 検索条件に一致するフレームの表示を指定 SC チェック・ポイントのフレームの表示を指定	なし	×	○	○
トレース・データ 検索条件の設定	TRF	なし	注 [A= [word] [V=mask8] [C=status] [E=mask8]] partition]	A=0XXXXH V=0XXH C=NC E=0XXH	×	○	○
アナライザの再起動	TRG	なし	なし	なし	×	○	×
トレース・モードの 設定	TRM	なし	[ALL] [TRX] [SEC]]	ALL 全トレース TRX クオリファアイ・トレース SEC 区間トレース	×	○	○

注 status は、次の中から選んでください。

[BROP
OP
RWI
RI
WI
RW
R
W
RWP
RP
WP
RWM
RM
WM
NC]

コマンドの種類	コマンド 本体	サブ コマンド	オ ペ ラ ン ド	デ ィ フ ォ ー ルト	動 作 状 態		
					trc:	emu:	brk:
トレース・データの 選択	TRS	なし	<div style="border: 1px dashed black; padding: 5px; display: inline-block;"> E→EXIT 外部データ T→TIME タイムタグ </div>	EXT	×	○	○
クオリファイ・トレ ース条件の設定	TRX	なし	<div style="border: 1px dashed black; padding: 5px; display: inline-block;"> BR? クオリファイ条件 OFF クオリファイの解除 </div>	OFF	×	○	○
オブジェクト・フア イルとメモリ内容の 比較	VRV	なし	file	なし	×	×	○
メモリ・ワード長の 設定	WRD	なし	<div style="border: 1px dashed black; padding: 5px; display: inline-block;"> B バイト指定 W ワード指定 </div>	B	×	○	○

第7章 コマンドの概要

(2) コマンドの初期値一覧表

IE-78327-Rのコマンドに設定されている初期値の一覧表を示します。

コマンド名	サブコマンド	初 期 値
A S M		0H
B R A	1	A=0XXXXH V=0XXH C=NC E=0XXXXY
	2	A=0XXXXH V=0XXH C=NC E=0XXXXY
	3	A=0XXXXH V=0XXH C=NC E=0XXXXY
	4	A=0XXXXH V=0XXH C=NC E=0XXXXY
B R D		0XXXXY
B R S	1	A=0H
	2	A=0H
	3	A=0H
	4	A=0H
B R M		BRA1
C H K		OFF
C L K		I
C N T		なし
C O M		CON:
C V D	D注	なし
	K	なし

注 コマンド本体のみを入力した場合には、このサブコマンドが指定されたものとして実行します。

第7章 コマンドの概要

コマンド名	サブコマンド	初 期 値
CVM	A	なし
	D注	なし
	K	なし
DAS		0H
DIR		カレント・ドライブのカレント・ディレクトリ
DLY		L
DOS		なし
DSB		OFF
ENB	1	ON
	2	ON
	3	ON
EVN		なし
EXT		なし
HIS		なし
HLP		なし
LOD		なし
LST		CON:

注 コマンド本体のみを入力した場合には、このサブコマンドが指定されたものとして実行します。

第7章 コマンドの概要

コマンド名	サブコマンド	初 期 値
MAP	I	なし
	T	なし
	W	なし
	R	なし
	U	なし
	K	なし
MAT		なし
MEM	C	0H
	D注	0H
	E	なし
	F	なし
	G	なし
	M	なし
	V	なし
	X	なし
MOD		MODE=CHAR BAUD=9600 LONG=8 PAR=NON STOP=2
MOV	U	なし
	I	なし
OUT		OFF
PAS		1T

注 コマンド本体のみを入力した場合には、このサブコマンドが指定されたものとして実行します。

第7章 コマンドの概要

コマンド名	サブコマンド	初 期 値
P G M	C	なし
		Termination of pgm/LOAD/SAVE/PGM ^Z/^B/^F/^Z Beginning of HEX-LOAD/HEX-SAVE /SYM-LOAD ^A/^E/^N Break of LOAD/SAVE ^W
P S A		なし
P S D		なし
P S T		4
R E G	C	R0
	D注	カレント・バンクの全レジスタ
R E S		なし
R U N	N	0H
	B	0H
	T	0H, 1T
S A V		なし
S F R	C	P0
	D注	読み出し可能な SFR すべて
S T P		なし
S T R		なし

注 コマンド本体のみを入力した場合には、このサブコマンドが指定されたものとして実行します。

第7章 コマンドの概要

コマンド名	サブコマンド	初 期 値
SYM	A	なし
	C	なし
	D注	登録されている全シンボル
	E	登録されている全IESYMBOL
	K	登録されている全シンボル
	L	なし
	S	なし
	M	なし
TRD	F	なし
	I	なし
TRF		A=0XXXXH V=0XXH C=NC E=0XXH
TRG		なし
TRM		ALL
TRS		EXT
TRX		OFF
VR Y		なし
WRD		B

注 コマンド本体のみを入力した場合には、このサブコマンドが指定されたものとして実行します。

(3) 数値と基数の対応

IE-78327-Rのコマンド入力時に使用する、数値と基数の対応について説明します。使用する数値は各コマンドの機能に応じて、以下のように固有の基数が規定されています。

基数を示す符号が省略された場合、規定された基数で数値は処理されます。規定外の基数を用いて数値を入力する場合、基数を示す符号を付加します。

表7-1 数値と基数の対応一覧

表 現	意 味	基 数
w o r d	16ビットの数値	16進数： H
m a s k 1 6	16ビットのマスク・データ	16進数： H
m a s k 8	8ビットのマスク・データ	16進数： H
m a s k 4	4ビットのマスク・データ	2進数： Y
p a s s 8	8ビットのパス・カウント	10進数： T
b i t	1ビットの数値	2進数： Y
s t e p 1 6	16ビットの実行ステップ数	10進数： T
p o i n t	11ビットのサブ・フレーム・ポイント数	10進数： T
n u m b e r	サンプル・タイミング	10進数： T
p a r t i t i o n	アドレス範囲	16進数： H
e x p r e s s i o n	式	16進数： H
d a t a - s t r i n g	バイト・データの集まり	16進数： H

第8章 コマンドの詳細

この章では、I E - 7 8 3 2 7 - Rで使用するすべてのコマンドについて、アルファベット順で詳細に説明します。

凡例

- _____ : 下線部分はキーボードからの入力を表します。
- ☐ : リターン・キー (CR (0DH)) の入力を表します。
- <ESC> : エスケープ・キーの入力を表します。
- ^ : コントロール・キーを押しながら ^ の右側の文字を入力することを表します。
- ■ : カーソルを表します。
- R/O : Read Only (リード・オンリー) を表します。
- R/W : Read Write (リード・ライト) を表します。
- W/O : Write Only (ライト・オンリー) を表します。
- このマニュアルで用いられている画面表示例、入力例はPC-9800シリーズをホスト・マシンとして使用した場合のものです。
- この章では、次に示す表を用いて、各コマンドがどのプロンプトのときに実行できるかを示しています。

例

A S M [word]	
基数	word:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

注意 A S Mコマンドは、プロンプトが brk:0 の場合にのみ使用できます。

8.1 ライン・アセンブル (ASM)

ASM [word]	
基数	word:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

word : アセンブルの開始アドレス

ASMコマンドは、word で指定されたアドレスからメモリの内容をニモニックで変更します。

ASM ☒ と入力した場合、前回アセンブルした次のアドレスが自動的に指定されます。

アセンブラ仕様の詳細については、第10章 オンライン・アセンブラ、逆アセンブラ仕様を参照してください。

例 200H 番地からメモリ内容を変更

```

brk:0>ASM 200H [E]
0200          NOP          ←変更前のニモニックの表示
      = MOV [HL], A [E]

      55
0201          NOP
      = MOVW AX, OFF23 [E]
Warning!(304)注 Generate code? (Y/N) N [E] ←奇数番地に対する Saddrp
      = MOVW AX, OFF23 [E]          操作のため Warning となる
Warning!(304)注 Generate code? (Y/N) Y [E]
1C 23          ←コードは生成されるが正しい
0203          NOP          動作は望めない
      = ORG 100H [E]          ←現在のロケーション・カウンタより前に
Caution!(303)注                                     さかのぼるため、Caution を表示
0100          NOP
      = MOV R1, FFH [E]          ←ニモニックが間違っているため、Error
Error!(302)注                                       を表示
      = MOV R1, #OFFH [E]
      B9 FF
0102          NOP
      = BR 500H [E]          ←ジェネリックなオブジェクト・コードが
Caution!(303)注                                       生成されたため、Caution を表示
      2C 00 05
0105          NOP
      = END [E]
brk:0>■

```

注 ()内の数字は、エラー・メッセージの番号を表します。

8.2 バス・イベント検出の条件設定 (BRA)

BRA $\left[\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \right] [A=mask16][V=mask8][C=status][E=mask4]$		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; padding: 2px;">基数</td> <td style="padding: 2px;">mask16:H mask4:Y mask8:H</td> </tr> </table>	基数	mask16:H mask4:Y mask8:H
基数	mask16:H mask4:Y mask8:H	

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

- BRA 1 : イベント検出器 1 指定
- BRA 2 : イベント検出器 2 指定
- BRA 3 : イベント検出器 3 指定
- BRA 4 : イベント検出器 4 指定
- mask16 : 16ビット検出アドレス (最大2箇所)
- mask8 : 8ビット検出データ
- status : 検出ステータス
- mask4 : 4ビット外部データ

BRAコマンドはバス・イベント検出条件を設定します。
 イベント検出条件として、アドレス、データ、ステータス、外部データの4つがあり、これらはAND条件として検出されます。
 バス・イベント検出用に4つのイベント検出器を持っているので、サブコマンドの番号を変えることで4種類の検出条件を設定することができます。

BRAn注②と入力した場合、指定したバス・イベント検出器に設定されている検出条件を表示し、対話形式の設定モードとなります。

アドレス条件は、2つまでスペースで区切って設定することができます。この場合、これらのアドレスはOR条件となります。省略した場合、アドレスはドント・ケアとなります。

データ条件は、8ビット・マスク・データで設定します。バス構成の都合上、0H-FDFHまでのアドレスに対しては下位8ビットのデータが有効になります。省略した場合、データはドント・ケアとなります。

ステータス条件は、以下のバス・サイクルの属性から選択します。省略した場合は、ステータスはドント・ケアとなります。

- ・OP : オペコード・フェッチ
- ・RW : リード・ライト
- ・R : データ・リード
- ・W : データ・ライト
- ・RWP : プログラムによるデータ・リードまたはライト
- ・RP : プログラムによるデータ・リード
- ・WP : プログラムによるデータ・ライト
- ・RWM : マクロ・サービスによるデータ・リードまたはライト
- ・RM : マクロ・サービスによるデータ・リード
- ・WM : マクロ・サービスによるデータ・ライト
- ・NC : すべてのフェッチ、リードまたはライト

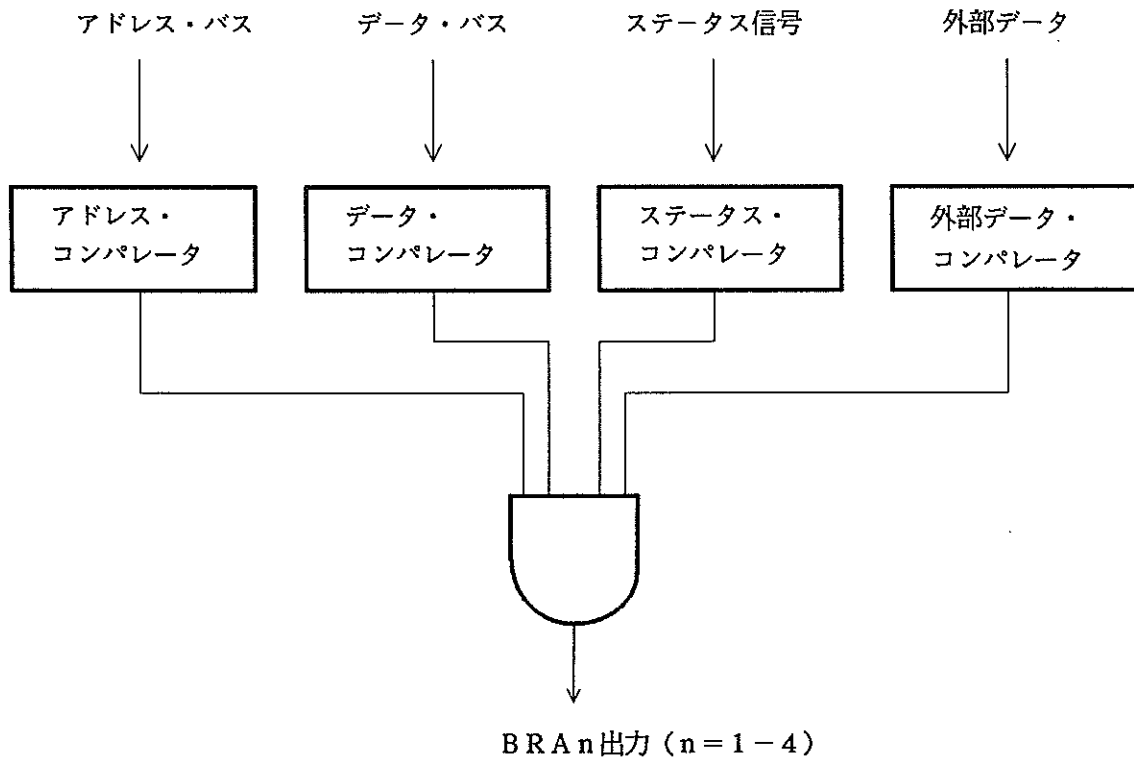
外部データ条件は、外部センス・クリップ（No. 5-8）から入力する外部データに対して、4ビット・マスク・データで設定します。省略した場合、外部データはドント・ケアとなります。

バス・イベント検出器の構成は、図8-1 バス・イベント検出器の構成図 のようになります。

注 n は1-4に対応します。

図8-1 バス・イベント検出器の構成図

次のコンパレータがBRA1-BRA4の4組あります。サブコマンドの番号を変えることで4種類の検出条件を設定することができます。



例1. 現在の設定条件を表示

```
brk:0>BRA [Enter]

      (BRA1)          (BRA2)          (BRA3)          (BRA4)
A 0XXXH             0FE40H,          0XXXXH          78XXH
   ---              1F00H              ---              ---

V 0XXH              0XXH              0XXH              0XXH

C RW                 RWP                 NC                 RW

E 0101Y             0XXXXY             0XXXXY             1000Y
brk:0>■
```

2. 対話形式でブレーク条件を設定

```
brk:0>BRA 1 [Enter]
A 0XXXXH
V 0XXH
C NC
E 0XXXXY

A = 0XXXH [Enter]
V = [Enter]
OP (OPeocode fetch)          RP (Read by program)
RW (Read Write)              WP (Write by program)
R (Read)                      RWM (Read Write by Macro service)
W (Write)                     RM (Read by Macro service)
RWP (Read Write by program)  WM (Write by Macro service)
                               NC (No condition)
                               ←新しく'RW'に設定

C = RW [Enter]
E = 100H [Enter]
入力データ・エラーです。(104)注
E = 100 [Enter] ←4ビット・マスク・データを4Hに設定
brk:0>■
```

注 () 内の数字は、エラー・メッセージの番号を表します。

例3. 1ラインでの設定

brk:0>BRA 2 A=0FE40 1F00 C=RWP (C) ←0FE40H 番地、1F00H 番地のメモリに対し、
brk:0>■ ターゲット・プログラムによりリードまたは
ライトした場合にイベント発生

8.3 外部データ検出の条件設定 (BRD)

BRD [MASK4]	
基数	MASK4:Y

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

mask4 : 外部センス・クリップNo. 1-4の信号レベル

BRDコマンドは外部入力信号レベルの検出条件を設定します。

BRD と入力した場合、現在設定されている検出レベルを表示します。

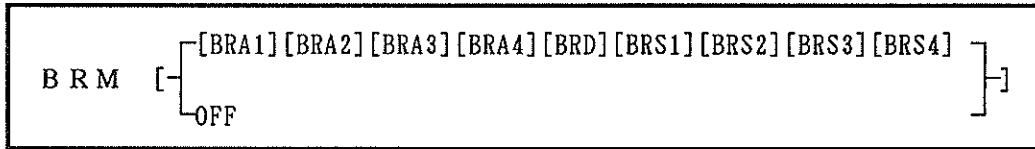
例1. 外部センス・クリップNo. 1-4の検出レベルを 3H に設定

```
brk:0>BRD 3H 
brk:0>■
```

2. 設定されている検出レベルを表示

```
brk:0>BRD 
0011Y
brk:0>■
```


8.4 トリガ条件の設定 (BRM)



trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

BR?注: 各トリガ条件
 OFF : 各トリガ条件の解除

注 BR?は、BRA1、BRA2、BRA3、BRA4、BRD、BRS1、BRS2、BRS3、BRS4のトリガ条件を表します。

BRMコマンドは、バス・イベント検出器BRA1 - BRA4、プログラム実行検出器、外部データ検出器のどの出力をトリガ信号とするかを設定します。トリガ条件は、スペースで区切ってOR条件で複数設定することができます。

BRM OFF ☒ と入力した場合、トリガ信号は発生しません。

BRM ☒ と入力した場合、現在設定されているトリガ条件を表示します。

例1. トリガ条件を設定

```
brk:0>BRM BRD BRS2 (Q) ←BRD、BR2 を設定  
brk:0>■
```

2. BRMのトリガ条件設定状態を表示

```
brk:0>BRM (Q) ←現在設定されている条件を表示  
BRD BRS2  
brk:0>■
```

8.5 プログラム実行検出の条件設定 (BRS)

BRS [$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$] [A=word]	
基数	word:H

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

- BRS 1 : イベント検出器1指定
- BRS 2 : イベント検出器2指定
- BRS 3 : イベント検出器3指定
- BRS 4 : イベント検出器4指定
- word : 検出アドレス

BRSコマンドは、プログラム実行検出器に対して、アドレスを設定します。

検出アドレスは、各イベント検出器に1ポイントのアドレスを設定します。各イベント検出器にアドレスを設定する場合、他のイベント検出器に設定されているアドレスとの間で5バイト以上の間隔を設ける必要があります。

5バイト以上の間隔がない場合、あとのポイントを検出しないことがあります。

例1. 現在の設定条件を表示

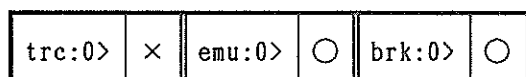
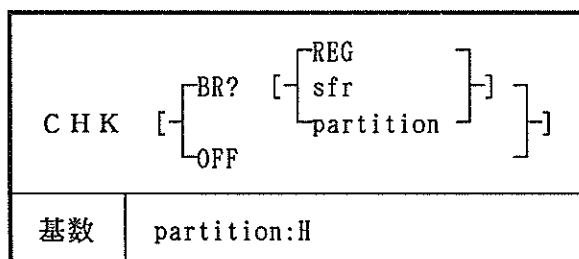
```
brk:0>BRS (␣)
  (BRS 1)  (BRS 2)  (BRS 3)  (BRS 4)
  A 0FD00H 0A000H  8000H  100H

brk:0>■
```

2. BRS 1 に設定

```
brk:0>BRS 1 A=7800 (␣)      ←7800H 番地を実行した場合に
brk:0>■                    イベント発生
```

8.6 チェック・ポイント条件の設定 (CHK)



- REG : レジスタ指定
- sfr : SFR名指定 (最大5つまで)
- partition : 内部RAM範囲指定
- BR?注 : 各チェック条件
- OFF : 設定解除

注 各チェック条件は、BRA1、BRA2、BRA3、BRA4、BRD、BRS1、BRS2、BRS3、BRS4です。
 またチェック条件は、スペースで区切って複数設定することができます。

CHKコマンドは、指定されたイベント条件が検出されたときにエミュレーションCPUを一時的に停止し、指定されたレジスタ、SFR、内部RAMの内容をトレースし、再度エミュレーションCPUを動作させます。

チェック・ポイントのトレース内容は、チェック・ポイントの次のフレームで表示されます。ただし、イベントの条件がBRAの場合、スリップすることがあります。

CHK OFF ☒ と入力した場合、チェック・ポイントの設定を解除します。

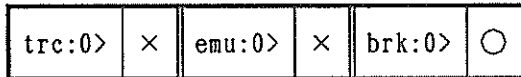
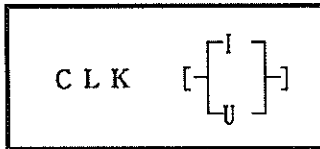
例1. チェック・ポイントにレジスタを指定

```
brk:0>CHK BRA1 BRS2 REG [Q] ←BRA1または、BRS2のどちらかの  
brk:0>■ イベント発生時にレジスタ内容をトレース
```

2. チェック・ポイントの設定を無効

```
brk:0>CHK OFF [Q]  
brk:0>■
```

8.7 クロックの選択 (CLK)



- I : エミュレータ内クロック
- U : ユーザ設定クロック

CLKコマンドは、エミュレーションCPUに供給する動作クロックの指定を行います。CLKコマンドを実行すると、エミュレーションCPUをリセットします。クロック・ソースは、次の(1)か(2)のどちらかを選択します。

(1) エミュレータ内クロック (CLK I)


エミュレーションCPUに接続する水晶振動子が16MHzの場合には、エミュレータ内クロック (CLK I コマンド) を指定します。

(2) ユーザ設定クロック (CLK U)

16MHz以外 (MAX. 16MHz) の動作クロックでデバッグ作業を行う場合には、ユーザ設定クロック (CLK U コマンド) を指定します。

この場合、IE-78327-R本体内部のエミュレーション・ボード上のクロック設置用ソケット (OPCK) に動作クロックの2倍の周波数の水晶発振器を設置してください。

例 12MHzでエミュレーションCPUを動作させる場合、24MHzの発振器を設置します。

- 備考1. IE-78327-R起動時には、エミュレータ内クロックが設定されていません。
- 2. CLK  と入力した場合、現在選択されているクロック・ソース名を表示します。
- 3. IE-78327-Rは、本体外部からクロックを供給することはできません (ターゲット・システムからのクロック供給は不可)。

例1. クロック・ソースを指定

```
brk:0>CLK U [Q]  
brk:0>■
```

←ユーザ設定クロックを指定

```
brk:0>CLK I [Q]  
brk:0>■
```

←エミュレータ内クロック (16 MHz)
を指定

2. クロックの設定状態を表示

```
brk:0>CLK [Q]  
Internal  
brk:0>■
```

←指定されているクロック・ソースを表示
←エミュレータ内クロックが指定されている
場合の表示

```
brk:0>CLK [Q]  
User  
brk:0>■
```

←ユーザ設定クロックが指定されている場合
の表示

8. 8 実行経過時間、実行命令数の表示 (CNT)

CNT

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

CNTコマンドは、ENBポイント通過後DSBポイント通過直前まで、および、ENBポイント通過後トリガ・ポイント通過直前までの実行経過時間または実行命令数の測定結果を表示します。

この測定は、DLYコマンドでのディレイ分は含みません。

実行経過時間は、最小400nsから最大858sec(約14分)まで、実行命令数は1-65535命令まで測定できます。

例1. 実行経過時間、実行命令数の表示

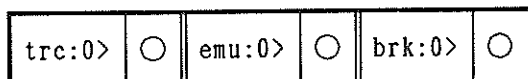
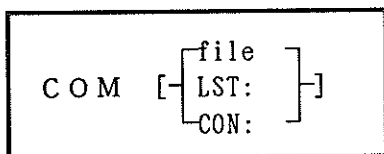
```
brk:0>CNT [C]
Emulation Time = 3min 12sec 500msec 000.0 usec ←実行経過時間の表示
Instruction Stop = 28931T ←実行命令数の表示
brk:0>■
```

2. 容量をオーバーした場合、オーバーフローのメッセージを表示

```
brk:0>CNT [C]
実行時間計測タイマーがオーバーフローしました。(200)注 ←タイマーがオーバーフローした場合
実行命令数計測カウンタがオーバーフローしました。(201)注 ←カウンタがオーバーフローした場合
brk:0>■
```

注 ()内の数字は、エラー・メッセージの番号を表します。

8.9 コマンド・ファイルの作成 (COM)



file : ファイル名
 LST: : プリンタ
 CON: : コンソール

COMコマンドは、入力されたコマンドを指定された装置に出力するため装置をオープンします。

COM file ☒ と入力した場合、入力されたコマンドは指定されたファイルへ格納されます。

ファイル名の拡張子を省略すると .STR が自動的に付加されます。

COM LST: ☒ と入力した場合、入力されたコマンドはプリンタへ出力されます。

コマンドの出力タイミングは、`^O` で制御します。

コマンドの出力停止は、次の `^O` が入力された場合となります。

COMコマンドによりファイル、プリンタをオープンした場合、必要データ出力後、必ず COM CON: ☒ と入力して装置をクローズしてください。

例1. ドライブ B に、SAMPLE.STR というファイルをオープン

```
brk:0>COM B:SAMPLE.STR (Q)
brk:0>■
```

2. ドライブ B に、オープンしようとするファイルと同一名称のファイルが存在する場合

```
brk:0>COM B:SAMPLE.STR (Q)
既に存在するファイルです。(513)注 B:SAMPLE.STR Delete ? (Y or N): Y (Q)
brk:0>■
```

ドライブ B にリード、ライト属性の SAMPLE.STR というファイルが存在している場合、上記のメッセージを表示します。

Y (Q) と入力した場合、既存のファイルは消去され、新しいファイルを作成します。なお、これ以外の入力ではファイルはオープンされません。

3. ファイルがオープンできない場合

```
brk:0>COM B:SAMPLE.STR (Q) ←R/O属性のファイル名を指定
読み取り専用のファイルです。(507)注 B:SAMPLE.STR
brk:0>■
```

ドライブ B に R/O 属性の SAMPLE.STR というファイルが存在している場合には上記のメッセージが表示され、コマンドは無視されます。

4. プリンタを正常にオープンした場合

```
brk:0>COM LST: (Q)
brk:0>■
```

注 () 内の数字はエラー・メッセージの番号を表します。

例5. プリンタが正常にオープンできない場合

brk:0>COM LST: (2)

他のコマンドでプリンタを使用しています。(109)^注

brk:0>

プリンタが他のコマンド（処理）によって使用できない場合には、上記のメッセージが表示されコマンドは無視されます。

6. ^O 入力によるコマンドの出力タイミング例

brk:0> <u>COM B:SAMPLE.STR</u> (2)	←ドライブ B に SAMPLE.STR という
brk:0> <u>RES</u> (2)	ファイルをオープン
brk:0> <u>^O CLK U</u> (2)	←コマンド行の最初で ^O を入力
brk:0> <u>OUT ON</u> (2)	
brk:0>	
brk:0> <u>^O</u>	←コマンド入力待ち状態で ^O を入力
→	SAMPLE.STR に出力されます

この場合、ファイルのオープン後に ^O が入力された直後のコマンド（CLK U）から、次の ^O が入力される直前のコマンド（OUT ON）までがファイルに出力されます。

注 () 内の数字はエラー・メッセージの番号を表します。

8.10 C0カバレッジ測定結果の表示 (CVD)

CVD [$\left[\begin{array}{c} D \\ K \end{array} \right] $ [partition]]	
基数	partition : H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

CVD D : 測定結果の表示
 CVD K : 測定結果の初期化
 partition : 表示/初期化範囲

CVDコマンドは、C0カバレッジ測定結果を表示、初期化します。

CVD $\text{\textcircled{D}}$ と入力した場合、C0カバレッジ測定結果の表示をします。

(1) CVD D コマンド

CVD [D[partition]]	
基数	partition : H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

CVD D : 測定結果の表示
 partition : 表示範囲

表示範囲を指定した場合、その範囲内で実行されたプログラム・アドレスの量を%表示し、その範囲について2バイト単位で表示を行います。

CVD D ☒ と入力した場合、マッピングされた範囲内で実行されたプログラム・アドレスの量を%表示し、64Kバイトのメモリ空間のプログラム実行マップを64バイト単位で表示します。

C0カバレッジの測定範囲の表示で、*はプログラムが実行されたことを表し、.はマッピングされていることを表します。また、空白の部分はマッピングされていないことを表し、?はマッピングされていない部分をプログラムが実行したことを表します。

例1. 測定結果の表示

```
brk:0>CVD D [Enter]
addr  000 100 200 300  400 500 600 700  800 900 A00 B00  C00 D00 E00 F00
0000  *****,** *****,.....***** *****
1000  *****,***** *****.....
2000  ???????
3000
4000
5000  .....
6000  ***** ???
7000
8000  .....
9000  .....
A000  .....
B000  .....
C000  .....
D000  .....
E000  .....
F000  .....
Coverage 19.3%
brk:0>[Enter]
```

2. 範囲を指定して測定結果を表示

```
brk:0>CVD D 300,4FF [Enter] ← 300H-4FFH 番地の測定結果を表示
addr 00          1F 20          3F 40          5F 60          7F
0300 *****
0380 ******,..... *****
0400 *****
0480 *****
coverage 89.8%
brk:0>[Enter]
```

(2) CVD K コマンド

CVD K [partition]

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

CVD K : 測定結果の初期化
 partition : 初期化範囲

CVD K コマンドは、C0カバレッジ測定結果の初期化を行います。
 CVD K ☐ と入力した場合、全空間の初期化を行います。

例 C0カバレッジ測定結果の初期化

```
brk:0>CVD K 300,4FF ☐
brk:0>■
```


8. 1 1 C 0カバレッジ測定範囲の操作 (CVM)

CVM [-[A [partition] [D][partition] K [partition]]]	
基数	partition:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

- CVM A : 測定範囲の追加
- CVM D : 測定範囲の表示
- CVM K : 測定範囲の指定解除
- Partition : C 0カバレッジ範囲

CVMコマンドは、C 0カバレッジ測定範囲の追加、表示、解除をします。

CVM A partition ☒ と入力した場合、指定範囲をC 0カバレッジ測定の範囲に追加します。

CVM D ☒ と入力した場合、すべてのC 0カバレッジ測定範囲を表示します。

CVM D partition ☒ と入力した場合、指定されたC 0カバレッジ範囲を表示します。

CVM K ☒ と入力した場合、すべてのC 0カバレッジ測定範囲の解除と測定結果の初期化を行います。

CVM K partition ☒ と入力した場合、指定されたC 0カバレッジ範囲の解除と測定結果の初期化を行います。

カバレッジの測定範囲の表示を行うとき、. はC 0カバレッジの測定範囲であることを表し、空白の部分はC 0カバレッジの測定範囲でないことを表します。

LODコマンドにおいて、オブジェクト・データをロードすると、ロード・アドレスがカバレッジ測定範囲に自動的に追加されます。

例1. C0カバレッジ測定範囲の追加

```
brk:0>CVM A 8000,0EFFF ☐ ← 8000H-0EFFFH 番地の測定範囲を追加
brk:0>■
```

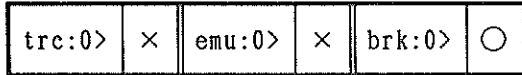
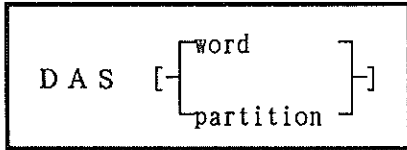
2. C0カバレッジ測定範囲の表示

```
brk:0>CVM D ☐
addr  000 100 200 300  400 500 600 700  800 900 A00 B00  C00 D00 E00 F00
0000 .....
1000 .....
2000 .....
3000 } C0カバレッジの測定範囲ではありません。
4000 }
5000 ....
6000 .....
7000 .....
8000 .....
9000 .....
A000 .....
B000 .....
C000 .....
D000 .....
E000 .....
F000 .....
brk:0>■
```

3. C0カバレッジ測定範囲の指定解除

```
brk:0>CVM K ☐
brk:0>■
```

8.12 逆アセンブル (DAS)



word : 逆アセンブルの開始アドレス
 partition : 逆アセンブルのアドレス範囲

DAS コマンドは、指定されたアドレスからメモリ内容をニモニックで表示します。

開始アドレスだけが指定された場合、指定されたアドレスから11行分を表示します。
 開始または終了アドレスが指定された場合、指定された範囲を表示します。

DAS と入力した場合、前回の逆アセンブルで表示された次のアドレスから11行分を表示します。

逆アセンブラの仕様については、第10章 オンライン・アセンブラ、逆アセンブラ仕様を参照してください。

例 逆アセンブル表示

```

brk:0>DAS 100                 ←100H 番地から11行分を表示
  Addr Object          Mnemonic
  0100 B8 00           MOV     R0,#0H
  0102 B9 01           MOV     R1,#1H
  .
  .
  0111 00              NOP
brk:0>■
  
```

8. 13 ディレクトリの表示 (DIR)

DIR [file]

trc:0> ○ emu:0> ○ brk:0> ○

file : ファイル名

DIRコマンドは、ディレクトリ名、ファイル名を表示します。
 ファイル名を省略すると、カレント・ドライブのカレント・ディレクトリにあるファイル名を表示します。
 ディレクトリ名を表示する場合、前後に < > を付加して表示します。
 ドライブ名を指定すると、そのドライブのファイル名を表示します。同様にパス名（ディレクトリ）や、ファイル名を指定したファイル名の表示も可能です。

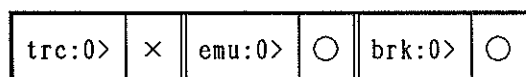
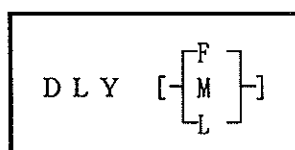
例1. カレント・ドライブのすべてのディレクトリ名、ファイル名の表示

```
brk:0>DIR Ⓢ
Directory = A:Y
COMMAND COM PRINT SYS RSDRV SYS CONFIG SYS FORMAT EXE
SPEED COM SETUP STR IE78330 COM SYMDEB EXE <TEST >
brk:0>■
```

2. パス名を指定してディレクトリ名、ファイル名の表示

```
brk:0>DIR YTEST Ⓢ
Directory = A:YTEST
< . > <.. > TEST1 HEX TEST1 DBG
brk:0>■
```

8.14 トリガ・ポイント位置の設定 (DLY)



- F : トリガ・ポイント位置をトレース・メモリおよび、サンプル・メモリの最初に設定
- M : トリガ・ポイント位置をトレース・メモリおよび、サンプル・メモリの中央に設定
- L : トリガ・ポイント位置をトレース・メモリおよび、サンプル・メモリの最後に設定

DLYコマンドは、BRMコマンドで設定したトリガ・ポイントを、トレース・メモリおよび、サンプル・メモリの最初、中央、最後に設定します。このコマンドにより、トリガ・ポイントに対するリアルタイム・トレーサおよび内部RAMデータ・サンプラのトレース領域を選択します。

設定するトリガ・ポイント位置は、以下の3ポイントから選択します。

- ・ トレース・メモリおよび、サンプル・メモリの最初 F
 (リアルタイム・トレーサおよび内部RAMデータ・サンプラは、トリガ・ポイント以降をトレースする)
- ・ トレース・メモリおよび、サンプル・メモリの中央 M
 (リアルタイム・トレーサおよび内部RAMデータ・サンプラは、トリガ・ポイントを中心としてその前後をトレースする)
- ・ トレース・メモリおよび、サンプル・メモリの最後 L
 (リアルタイム・データ・サンプラはトリガ・ポイントをトレースの最後として、それ以前をトレースする)

DLY @ と入力した場合、現在の設定状態を表示します。

例1. トリガ・ポイント位置の設定

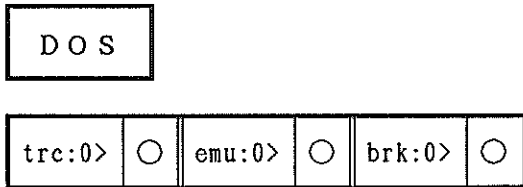
brk:0>DLY F [Q] ←トリガ・ポイント位置をトレース・メモリおよび、サンプル・メモリの最初に設定
brk:0>■

brk:0>DLY L [Q] ←トリガ・ポイント位置をトレース・メモリおよび、サンプル・メモリの最後に設定
brk:0>■

2. トリガ・ポイント位置の設定状態を表示

```
brk:0>DLY [Q]
Trigger point   F      M      L
                |      |      |
Trace memory   -----*--> ←現在のトリガ・ポイント位置を * で表示
brk:0>■
```

8.15 子プロセスの実行 (DOS)



DOS コマンドは、MS-DOS、PC DOS の環境設定時で "CONFIG.SYS" 内の shell コマンドで指定した、環境文字列テーブル中の "COMSPEC" に設定されているファイル名のコマンド・プロセッサを子プロセスとして実行し、コントロール・プログラムの制御を一時中断します。

"COMSPEC" に設定されているファイルがない場合は、環境設定でパス指定されているパスで MS-DOS、PC DOS の標準コマンド・プロセッサである "COMMAND.COM" をサーチして実行します。

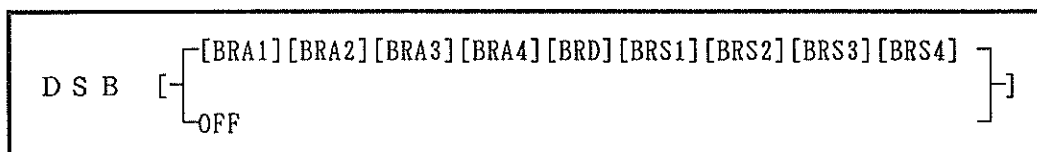
コマンド・プロセッサがサーチできない場合は、子プロセスの実行は行いません。

子プロセスとして MS-DOS、PC DOS の標準コマンド・プロセッサである "COMMAND.COM" が実行された場合、コントロール・プログラムに制御を戻すときには、文字列 EXIT ⓐ を入力します。

例 子プロセスの実行

brk:0> <u>DOS</u> ⓐ	←制御を OS に移動
A>	←OS のプロンプトの表示
A> <u>EXIT</u> ⓐ	←コントロール・プログラムに制御を移動
return from child	
brk:0>■	←コントロール・プログラムのプロンプトの表示

8.16 ディスエーブル条件の設定 (DSB)



trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

BR? 注 : 各ディスエーブル条件
 OFF : ディスエーブル条件の解除

注 BR?は、BRA1、BRA2、BRA3、BRA4、BRD、BRS1、BRS2、BRS3、BRS4のディスエーブル条件を表します。

DSBコマンドは、ディスエーブル条件の設定を行いません。

ディスエーブル条件が満たされると次のイネーブル条件が満たされるまでは、トリガ条件が満たされてもトリガ信号は出力されません。

イネーブル条件がシーケンシャル・イネーブルとなっていた場合は、すべてクリアされ、ENB1より再度検出を行います。

ディスエーブル条件は、スペースで区切って複数設定することができます。

DSB ☒ と入力した場合、現在設定されているディスエーブル条件を表示します。

DSB OFF ☒ と入力した場合、現在設定されているディスエーブル条件を解除します。

例1. ディスエーブルの条件を設定

```
brk:0>DSB BRA1 BRA2 (C)
brk:0>■
```

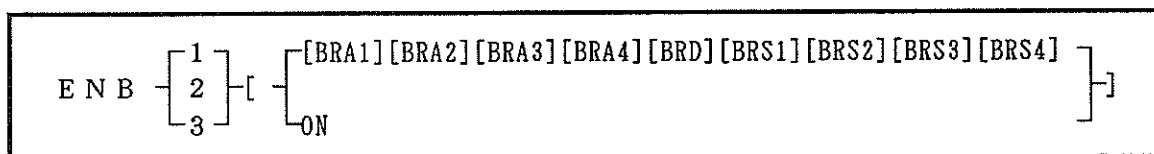
←ディスエーブル条件にBRA1,
BRA2を設定

2. 設定状態を表示

```
brk:0>DSB (C)
BRA1 BRA2
brk:0>■
```

←BRA1、BRA2が設定されている

8.17 イネーブル条件の設定 (ENB)



trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

- ENB 1 : イネーブル1 指定
- ENB 2 : イネーブル2 指定
- ENB 3 : イネーブル3 指定
- BR? ^{注1} : 各イネーブル条件
- ON : 各イネーブル条件のパス指定

ENB コマンドは、イネーブル条件を設定します。
イネーブル条件が満たされ、イネーブル状態になるとトリガ条件が満たされることにより、トリガ信号が出力されます。

イネーブル条件は、スペースで区切って複数設定することができます。

イネーブル・ポイントを複数指定した場合は、シーケンシャル・イネーブルとなり、ENB 1, ENB 2, ENB 3の順にイネーブル・ポイントを通過したときのみ、トリガが有効になります。

ENB 1, ENB 2, ENB 3の順にシーケンシャル・イベント検出を行っている途中、DSBコマンドで指定されたイベントが発生すると、それまでのイベント検出を無効にし、再度シーケンシャル・イベント検出を行います。

ENB n ON ^{注2}と入力した場合、本イネーブル条件はパス（イネーブル・ポイントの通過の有無にかかわらず、イネーブル条件が満たされているようにすること）され、次のイネーブル条件が有効となります。たとえば、ENB 1, ENB 2をONにした場合、ENB 3の条件を満たすだけでトリガが発生します。

ENB n ^{注2}と入力した場合、指定したイネーブル条件を表示します。

ENB ^{注2}と入力した場合、設定状態を表示します。

注1. BR? は、BRA 1、BRA 2、BRA 3、BRA 4、BRD、BRS 1、BRS 2、BRS 3、BRS 4のイネーブル条件を表します。

2. n は 1 - 3 を表します。

例1. イネーブル条件を設定

```
brk:0>ENB 1 BRA1 BRS1 (Q) ←イネーブル条件1をBRA1, BRS1に設定  
brk:0>■
```

```
brk:0>ENB 3 BRA4 (Q) ←イネーブル条件3をBRA4に設定  
brk:0>■
```

2. ENB 1 の設定状態を表示

```
brk:0>ENB 1 (Q)  
1 BRA1 BRS1  
brk:0>■
```

3. イネーブルの条件をすべて表示

```
brk:0>ENB (Q)  
1 BRA1 BRS1  
2 ON  
3 BRA4  
brk:0>■
```

8.18 イベント検出器の設定状態表示 (ENV)

ENV					
trc:0>	<input type="radio"/>	emu:0>	<input type="radio"/>	brk:0>	<input type="radio"/>

ENVコマンドは、イベントの設定状態を表示します。

イベントの設定状態の表示は、二画面にまたがって表示するので、Next ? (Y/N) と表示し、次画面への移行を確認します。

Y と入力した場合、次画面を表示します。

N と入力した場合、コマンドを終了します。

例 イベント状態を表示

brk:0>EVN Q

```

Enable1      ENB1 ON
Enable2      ENB2 ON
Enable3      ENB3 ON
Disable      DSB OFF
Trigger      BRM  BRA1 BRS1
Check point  CHK  OFF
Qualify trace TRX  OFF
Pass count   PAS  1T
Delay count  DLY  M
Trigger out  OUT  ON
    
```

```

External data BRD  OXXXXY
Program execute BRS1 A=0XXXXH
                BRS2 A=0XXXXH
                BRS3 A=0XXXXH
                BRS4 A=0XXXXH
    
```

←ここまで表示して一時停止

Next ? (Y/N) Y Q

←次画面を表示

```

Bus detect    BRA1 A=0XXXXH
              V=0XXH
              C=NC
              E=0XXXXY
              BRA2 A=0XXXXH
              V=0XXH
              C=NC
              E=0XXXXY
              BRA3 A=0XXXXH
              V=0XXH
              C=NC
              E=0XXXXY
              BRA4 A=0XXXXH
              V=0XXH
              C=NC
              E=0XXXXY
    
```

8. 19 コントロール・プログラムの終了 (EXT)

EXT

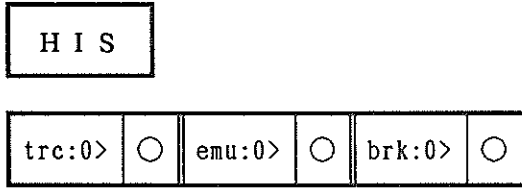
trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

EXTコマンドは、IE-78327-Rコントロール・プログラムを終了しOSに制御を戻します。このとき、オープンされているファイルはすべてクローズされます。

例 コントロール・プログラムの終了

```
brk:0>EXT Ⓢ
A>■ ←OSのプロンプト
```

8. 20 コマンド・ヒストリの表示 (HIS)



HIS コマンドは、ヒストリ・メモリに記憶されている最新のコマンド行（最大20行分）をコマンド番号とともに古いものから順番に表示します。

ヒストリ・メモリへの登録は、コマンド実行ごとに自動的行われます。

特定のコマンド行を読み出す場合、!n と入力します（nはコマンド番号）。

最新のコマンド行を読み出す場合、!! と入力します。

読み出したコマンドを実行する場合、そのコマンド行で を入力します。

例1. ヒストリ・メモリに登録されているコマンドを表示

```
brk:0>HIS 
  1 LOD TEST
  2 CLK
  .
  .
 20 HIS
brk:0>■
```

2. コマンド番号を指定して特定のコマンドを読み出す

```
brk:0>!2            ←ヒストリ・メモリの2番目のコマンドを読み出す
  CLK            ←このコマンドを実行する場合  を入力
  Internal
brk:0>■
```


8. 2 1 コマンド・ヘルプの表示 (HLP)


HLP [command]

trc:0> ○ emu:0> ○ brk:0> ○


command : コマンド本体

HLPコマンドは、コマンドの一覧、コマンドの使用方法を表示します。

コマンド本体を指定した場合、指定したコマンドの使用方法を表示します。HLP  と入力した場合、コマンドの一覧表示後にコマンド本体の入力待ちとなります。

HLPコマンドを終了する場合、 を入力します。

例 コマンドの一覧、コマンドの使用方法の表示

brk:0>HLP 


Command Table

ASM	BRA	BRD	BRS	BRM
CHK	CVD	CVM	CLK	CNT
COM	DAS	DLY	DSB	DIR
DOS	ENB	EVN	EXT	HIS
HLP	LOD	LST	MAP	MAT
MEM	MOD	MOV	OUT	PAS
PGM	PSA	PSD	PST	REG
RES	RUN	SAV	SFR	STP
STR	SYM	TRD	TRG	TRF
TRM	TRS	TRX	VRY	WRD

←コマンドの
一覧を表示

HLP>DIR  ←DIRコマンドを指定

DIRコマンドの説明

HLP>  ←ヘルプ・コマンドを終了

brk:0>■

8. 2 2 オブジェクト、シンボル、デバッグ環境のロード (LOD)

```
LOD file[module name¥][[D][C][S]][$V]
```

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

- file : ファイル名
- module name¥^注 : モジュール名
- D : デバッグ環境指定
- C : オブジェクト指定
- S : シンボル指定
- \$V : ベリファイ指定

注 ¥ は IBM PC シリーズをホスト・マシンとして使用した場合 \ となります。

LOD コマンドは、デバッグ環境、オブジェクト・コード、シンボル・テーブルを指定されたファイルから連続または個別にロードします。

シンボル・テーブルをロードする際は、必要なモジュールを指定してロードすることができます。

\$V を指定するとベリファイ付きでロードを行います。また、ロード中にエラーが検出されると LOD コマンドを終了します。

IE-78327-R 起動時に高速ダウン・ロード・モードを指定した場合、チャンネル 4 を使用してロードを行います。しかし他のプロセスでプリンタを使用しているときは次のメッセージを表示し、チャンネル 1 を使用してロードを行います。

Select serial interface

例1. デバッグ環境、オブジェクト、シンボルの連続ロード

```
brk:0>LOD SAMPLE Q注
Debug condition load (Y/N)? Y      ←デバッグ環境をロードするか聞いてきます
debug condition load complete      ←のでYを入力
object load complete
symbol table loading              ←正常終了した場合のメッセージ
PUBLIC      load complete
MOD01      load complete
MOD02      load complete
brk:0>■
```

2. デバッグ環境のロード

```
brk:0>LOD SAMPLE.DBG D Q      ←コマンドの最後にスイッチDをつけます
debug condition load complete    ←正常終了した場合のメッセージ
brk:0>■
```

```
brk:0>LOD SAMPLE D Q          ←ファイル名の拡張子を省略した場合、
debug condition load complete    .DBG が設定されます。
brk:0>■
```

注 拡張子、スイッチを省略した場合、連続ロードとなります。

例3. オブジェクト・コードのロード

```
brk:0>LOD SAMPLE.HEX C [C] ←コマンドの最後にスイッチCをつけます
object load complete ←正常終了した場合のメッセージ
brk:0>■
```

```
brk:0>LOD SAMPLE C [C] ←ファイル名の拡張子を省略した場合、HEX
object load complete が設定されます
brk:0>■
```

オブジェクト・コードをロード中にエラーを検出した場合、次のメッセージを表示します。

チェックサムエラーです。(516) 注 ←チェック・サム・エラーを検出
不正文字がありました。(515) 注 ←レコード中に許されない文字を検出
マッピングされていません。(107) 注 ←マッピングされていないメモリにロードしようとした

注 () 内の数字は、エラー・メッセージの番号を表します。

例4. シンボル・テーブルのロード

```
brk:0>LOD SAMPLE.SYM S Ⓢ
symbol table loading
PUBLIC      load complete
MOD01       load complete
MOD02       load complete
brk:0>■
```

←コマンド行の最後にスイッチSを付加
 ←シンボル・ロードの開始メッセージ
] ←モジュール・ブロックのロード終了
 メッセージ

```
brk:0>LOD SAMPLE S Ⓢ
symbol table loading
PUBLIC      load complete
MOD01       load complete
MOD02       load complete
brk:0>■
```

←ファイル名の拡張子を省略した場合
 S Y Mが設定されます

```
brk:0>LOD SAMPLE PUBLICY MOD02Y S Ⓢ
symbol table loading
PUBLIC      load complete
MOD01       pass
MOD02       load complete
brk:0>■
```

←モジュールを指定した場合指定したモジュールだけをロードします

5. すでにロード済みのシンボル・モジュールを再ロードした場合

```
brk:0>LOD SAMPLE S Ⓢ
symbol table loading
PUBLIC      Pass
MOD01       load complete
MOD02       Pass
brk:0>■
```

←ロードされません
 ←ロードされません

例6. ロード中にエラーを検出した場合

```
brk:0>LOD SAMPLE PUBLICY MOD02Y S (2)
symbol table loading
PUBLIC    load complete
MOD01     pass
MOD02     失敗しました。(518)注 ←モジュール・ブロックをロード中に
brk:0>■                               エラーを検出した場合のメッセージ
```

注 ()内の数字はエラー・メッセージの番号を表します。

8.23 ファイルへの結果出力 (LST)

```
L S T [-[file
      LST:
      CON: ]]
```

```
trc:0> ○ emu:0> ○ brk:0> ○
```

file : ファイル名
 LST: : プリンタ
 CON: : コンソール

L S T コマンドは、コンソールに表示されたデータを指定された装置に出力するため、装置をオープンします。

LST file ㊦ と入力した場合、表示データは指定されたファイルへ格納されます。ファイル名の拡張子を省略すると .TXT が設定されます。

LST LST: ㊦ と入力した場合、表示データはプリンタへ出力されます。

データの出力タイミングは、[^]P で制御します。データの出力停止は、次の [^]P が入力された場合となります。

L S T コマンドによりファイル、プリンタをオープンした場合、必要データ出力後に必ず LST CON: ㊦ と入力して装置をクローズしてください。

例1. ドライブBに SAMPLE.TXT というファイルをオープンする場合

```
brk:0>LST B:SAMPLE.TXT 
brk:0>■
```

2. ドライブBに、オープンしようとするファイルと同一名称のファイルが存在する場合

```
brk:0>LST B:SAMPLE.TXT  注
既に存在するファイルです。(513)B:SAMPLE.TXT Delete ? (Y or N): Y 
brk:0>■
```

ドライブBにR/W属性の SAMPLE.TXT というファイルが存在する場合、上記のメッセージを表示します。

Y と入力した場合、既存のファイルは消去され、新しくオープンします。

Y 以外を入力した場合、ファイルはオープンされません。

3. ファイルがオープンできない場合

```
brk:0>LST B:SAMPLE.TXT  ←R/O属性のファイル名を指定
読取り専用のファイルです。(507) 注 B:SAMPLE.
brk:0>■
```

ドライブBにR/O属性の SAMPLE.TXT というファイルが存在している場合、上記のメッセージを表示してコマンドは無視されます。

4. プリンタを正常にオープンした場合

```
brk:0>LST LST: 
brk:0>■
```

注 () 内の数字はエラー・メッセージの番号を表します。

例5. プリンタが正常にオープンできない場合

```
brk:1>LST LST: (2)
```

他のコマンドでプリンタを使用しています。(109) 注

```
brk:1>
```

プリンタが他のコマンド（処理）によって使用できない場合、上記のメッセージが表示され、コマンドは無視されます。

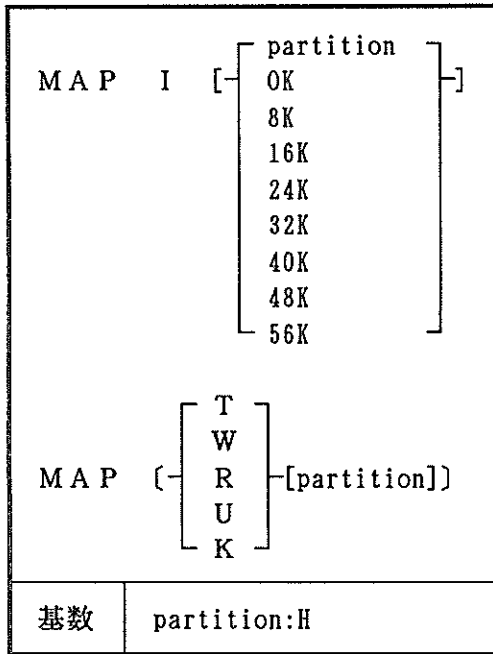
6. ^P 入力によるコマンドの出力タイミング例

brk:0>LST B:SAMPLE.TXT (2)	←ドライブ B に、SAMPLE.TXT という
brk:0>RES (2)	ファイルをオープン
brk:0>^P CLK U (2)	←コマンド行の最初で ^P を入力
brk:0>OUT ON (2)	
brk:0>■	
brk:0>^P	←コマンド入力待ち状態で ^P を入力
→ SAMPLE.TXT へ出力	

この場合、ファイルのオープン後 ^P が入力された直後のコマンド（CLK U）から次の ^P が入力される直前のコマンド（OUT ON）までがファイルに出力されます。

注 () 内の数字はエラー・メッセージの番号を表します。

8.24 マッピングの設定 (MAP)



trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

- MAP I : 内部ROM
- MAP T : ターボ・アクセス・マネージャ代替メモリ
- MAP W : 代替メモリ
- MAP R : ライト・プロテクト付き代替メモリ
- MAP U : ユーザ・メモリ
- MAP K : マッピングの解除 (ノンマッピング)
- partition : マッピング範囲

MAP コマンドは、対象デバイスの使用するメモリ属性を指定します。

0H-0DFFFH のメモリ領域は、8 Kバイト単位で次のメモリ属性を指定します。

- 内部ROM
- ターボ・アクセス・マネージャ代替メモリ
- 代替メモリ
- ライト・プロテクト付き代替メモリ
- ユーザ・メモリ
- マッピング解除


0E000H-0FFFFH (内部RAM、SFR領域は除く) の8 Kバイトのメモリ領域は、以下のメモリ属性を指定します。


- ユーザ・メモリ
- マッピング解除

MAP コマンドを実行すると、エミュレーションCPUはリセットされます。

内部ROMマッピングに設定されている範囲は、MAP I コマンド以外で変更することはできません

コマンド入力時にサブコマンドのみを指定すると、サブコマンドで指定したマッピングの範囲が表示されます。

MAP K  と入力した場合、内部ROM以外のすべてのマッピングが解除されます。

MAP  と入力した場合、現在のマッピング状態を表示します。

例1. マッピングの設定

```
brk:0>MAP I 8K [C]      ← 0H 番地から 1FFFH 番地までを内部ROMに
brk:0>■                設定

brk:0>MAP R 2000,3FFF [C] ← 2000H 番地から 3FFFH 番地までをライト・
brk:0>■                プロテクト付き代替メモリに設定
```

2. マッピング状態の表示

```
brk:0>MAP [C]
0000-1FFF Internal ROM
2000-3FFF R/O emulation
4000-5FFF R/W emulation
6000-7FFF User
8000-DFFF Turbo emulation
E000-FC7F Non map
FD00-FE7F (Internal RAM) ←内部RAM空間を示しています
brk:0>■                この領域に対しては、MAPコマンド
                        で操作できません

brk:0>MAP R [C]        ←ライト・プロテクト付き代替メモリの
2000-3FFF              範囲を表示
brk:0>■
```

3. マッピングの解除

```
brk:0>MAP K 2000,5FFF [C] ← 2000H 番地から 5FFFH 番地までのマッピン
brk:0>MAP K 0,1FF [C]     グを解除
マッピング・エラーです。(106) 注 ←内部ROMの範囲を指定したためエラー
brk:0>■
```

注 () 内の数字はエラー・メッセージの番号を表します。

8.25 演算 (MAT)

MAT expression	
基数	expression:H

trc:0>	○	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

expression : 式

MATコマンドは、オペランドに記述された式表現を評価し、その結果を16進数、10進数、8進数、2進数の順に表示します。

式には次に示す演算子が記述できます。カッコの使用は、最大32レベルまでです。

()	高
* /	↑
+ -	優先順位
AND	↓
OR XOR	低

演算は、すべて16ビットの整数で行われます。演算の中間結果、最終結果が16ビット以上になった場合、17ビット目より上位は切り捨てます。

例 演算

```
brk:0>MAT 5H+7H AND 17Q  ←式の入力
0CH,12T,14Q,1100Y ←演算結果の表示
brk:0>■
```


8. 26 メモリの操作 (MEM)

MEM C [word]
MEM D [[word] partition]
MEM E [partition]
MEM [F] partition data-string [G]
MEM [M] Partition word [V] [X]
基数 word:H partition:H data-string:H

trc:0>	×	emu:0>	注	brk:0>	○
--------	---	--------	---	--------	---

- | | | | |
|-------------|-------------|-----------|------------|
| MEM C | : メモリ内容の変更 | MEM G | : メモリ内容の検索 |
| MEM D | : メモリ内容の表示 | MEM M | : メモリ内容の複写 |
| MEM E | : メモリの検査 | MEM V | : メモリ内容の比較 |
| MEM F | : メモリ内容の初期化 | MEM X | : メモリ内容の交換 |
| word | : 開始アドレス | partition | : アドレス範囲 |
| data-string | : データ列 | | |

MEM コマンドは、WRD コマンドで指定したメモリ長で、メモリ内容の変更、表示、初期化、検索などを行います。

MEM  と入力した場合、メモリ内容の表示を行います。

注 emu:0> では、MEM C、D のみ実行可能です。

(1) MEM C コマンド

MEM C [word]	
基数	word:H

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

word : 変更開始アドレス

MEM C コマンドは、マッピングされている 0H-0FEFFH 番地のメモリ内容を変更します。

オペランドでメモリの変更開始アドレスを指定した場合、指定したアドレスから順番にメモリ内容を変更します。

メモリ内容を変更しない場合は 、変更を終了する場合には または、 または、 を入力します。

MEM C と入力した場合、前回メモリ内容を変更した次のアドレスを開始アドレスとして、メモリ内容を変更します。

注意 トレース(trc:>)およびエミュレーション(emu:>)時にメモリの内容を操作するコマンドを入力すると、エミュレーションCPUの実行は一時中断されます。

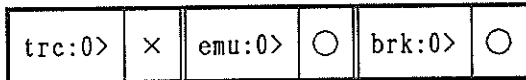
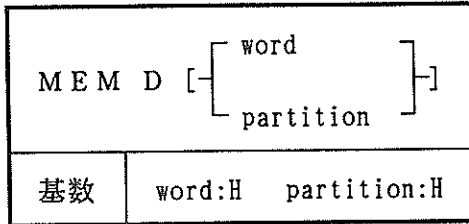
例1. バイト単位でのメモリ内容を変更 (WRDコマンドでB指定の場合)

```
brk:0>MEM C 100 [Q]      ← 100H 番地からメモリ内容を変更
0100 00 11 [Q]
0101 11 22 [Q]
0102 22 . [Q]           ←メモリ内容の変更終了
brk:0>■
```

2. ワード単位でのメモリ内容を変更 (WRDコマンドでW指定の場合)

```
brk:0>MEM C 100 [Q]      ←100H 番地からメモリ内容を変更
0100 1100 2211 [Q]
0102 3322 4433 [Q]
.
.
.
.
0108 9988 . [Q]         ←メモリ内容の変更終了
brk:0>■
```


(2) MEM D コマンド



word : 表示開始アドレス
partition : 表示アドレス範囲

MEM D コマンドは、マッピングされている 0H-0FEFFH 番地のメモリ内容を表示します。

オペランドでメモリの表示開始アドレスを指定した場合、指定したアドレスから 11 行分のメモリ内容を表示します。メモリの表示範囲を指定した場合、指定した範囲のメモリ内容を表示します。

MEM D  と入力した場合、前回メモリ内容を表示した次のアドレスを開始アドレスとして 11 行分のメモリ内容を表示します。

バイト単位でのメモリ内容の表示の場合は、16 進数と、次ページに示す ASCII 文字で表示されます。

ワード単位でのメモリ内容の表示は、16 進数だけで表示します。

表8-1 バイト単位でのASCII文字データ

		下 位 4 ビ ッ ト															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
上 位 4 ビ ッ ト	2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
	6	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	7	p	q	r	s	t	u	v	w	x	y	z	{		}	.	.

例1. バイト単位でのメモリ内容の表示 (WRDコマンドでB指定の場合)

```
brk:0>MEM D 100,17F ☐ ← 100H-17FH までのメモリ内容を表示
0100      00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F .....
0110      30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 0123456789:;<=>?
0120      40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F @ABCDEFGHIJKLMNO
0130      50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F PQRSTUVWXYZabcde
      .
      .
      .
0170      20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F !"#$%&'()*+,-./
```

└─ アドレス ─┬──────────┬── データ (16 バイト) ─────────┬── ASCII文字 ─┘
brk:0>■

2. ワード単位での 100H-12FH までのメモリ内容の表示
(WRD コマンドでW指定の場合)

```
brk:0>MEM D 100,12F ☐ ← 100H-12FH までのメモリ内容を表示
0100      0100 0302 0504 0706 0908 0B0A 0D0C 0F0E
0110      3130 3332 3534 3736 3938 3B3A 3D3C 3F3E
0120      4140 4342 4544 4746 4948 4B4A 4C4D 4F4E
brk:0>■
```

(3) MEM E コマンド

MEM E [partition]	
基数	partition:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

partition : 検査アドレス範囲

MEM E コマンドは、マッピングされている 0H-0FE7FH 番地のメモリを検査します。

オペランドで検査の範囲を指定する方法と、MEM E ☐ と入力してマッピングされているすべてのメモリを検査する方法があります。

例 メモリの検査

```
brk:0>MEM E 0XXX ☐      ← 0000H-0FFFH までのメモリを検査
complete                ← 正常終了した場合のメッセージ
brk:0>■
```

```
brk:0>MEM E 1000,1FFF ☐ ← 1000H-1FFFH までのメモリを検査
1021                    ← メモリ検査で異常を検出したアドレスを
brk:0>■                表示します
```

メモリ検査で異常を検出した場合、そのアドレス以後の検査は行いません。

(4) MEM F コマンド

MEM F partition data-string	
基数	partition:H data-string:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

partition : 初期化アドレス範囲
 data-string : 初期化データ列

MEM F コマンドは、マッピングされている 0H-0FE7FH 番地のメモリ内容を任意のデータで初期化します。

初期化データに指定できるデータ列は、最大10個までです。初期化データには、マスク・データを記述できます。この場合、マスクされたビットは変更されません。

例1. バイト単位でメモリ内容をデータ列で初期化する場合
(WRDコマンドでB指定の場合)

```
brk:0>MEM F 100,1FF 1,2,3,4,5,6,7,8,9,0 [E] ← 100H-1FFH までを1,2,3,4,
brk:0>■                                     5,6,7,8,9,0 のバイト・デー
                                                タ列で初期化

brk:0>MEM D 100,1FF [E]
0100  01 02 03 04 05 06 07 08 09 00 01 02 03 04 05 06  .....
0110  07 08 09 00 01 02 03 04 05 06 07 08 09 00 01 02  .....
0120  03 04 05 06 07 08 09 00 01 02 03 04 05 06 07 08  .....
      .
      .
      .
01E0  05 06 07 08 09 00 01 02 03 04 05 06 07 08 09 00  .....
01F0  01 02 03 04 05 06 07 08 09 00 01 02 03 04 05 06  .....
brk:0>■
```

2. メモリ内容をマスク・データで初期化する場合

```
brk:0>MEM F 100,11F 3X [E]
brk:0>■
```

```
brk:0>MEM D 100,13F [E]
0100  31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36  1234567890123456
0110  37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32  7890123456789012
0120  03 04 05 06 07 08 09 00 01 02 03 04 05 06 07 08  .....
0130  09 00 01 02 03 04 05 06 07 08 09 00 01 02 03 04  .....
brk:0>■
```

備考 マスク・データ 3 X で初期化した場合、上記のように下位 4 ビットを初期化しません。

例4. ワード単位でメモリ内容をデータ列で初期化する場合
(WRDコマンドでW指定の場合)

```
brk:0>MEM F 100,17F 1,2,3,4,5,6,7,8,9,0 [C] ← 100H-17FH までを1,2,3,4,
brk:0>█                                         5,6,7,8,9,0 のワード・デー
                                                タ列で初期化

brk:0>MEM D 100,17F [C]
0100  0001 0002 0003 0004 0005 0006 0007 0008  .....
0110  0009 0000 0001 0002 0003 0004 0005 0006  .....
0120  0007 0008 0009 0000 0001 0002 0003 0004  .....
0130  0005 0006 0007 0008 0009 0000 0001 0002  .....
      .
      .
      .
      .
0170  0007 0008 0009 0000 0001 0002 0003 0004  .....
brk:0>█
```

5. マスク・データによる初期化

```
brk:0>MEM F 100,11F 808X [C]

brk:0>MEM D 100,13F [C]
0100  8081 8082 8083 8084 8085 8086 8087 8088  .....
0110  8089 8080 8081 8082 8083 8084 8085 8086  .....
0120  0007 0008 0009 0000 0001 0002 0003 0004  .....
0130  0005 0006 0007 0008 0009 0000 0001 0002  .....
brk:0>█
```

備考 マスク・データ'808X'で初期化した場合、上記のように下位4ビットを初期化しません。

(5) MEM G コマンド

MEM G partition data-string	
基数	partition:H data-string:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

partition : 検索アドレス範囲
 data string : 検索データ列

MEM G コマンドは、マッピングされている 0H-0FE7FH 番地の指定したデータ列を検索します。

検索データに指定できるデータ列は、最大10個までです。検索データには、マスク・データを記述できます。

例1. バイト単位でメモリ内容を検索 (WRDコマンドでB指定の場合)

```
brk:0>MEM G 100,11F 30,31,32 [C] ← 100H-11FH の範囲から30,31,32
0109                               の連続したデータ列をサーチ
0113                               ] データ列を検出したアドレス
011D                               ]
brk:0>■
```

```
brk:0>MEM D 100,11F [C]
0100  31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36  1234567890123456
0110  37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32  7890123456789012
brk:0>■
```

2. ワード単位でメモリ内容を検索 (WRDコマンドでW指定の場合)

```
brk:0>MEM G 100,12F 12,13,14 [C] ← 100H-12FH の範囲から12,13,14
0124                               の連続したデータ列をサーチ
brk:0>■
```

```
brk:0>MEM D 100,12F [C]
0100  0000 0001 0002 0003 0004 0005 0006 0007  .....
0110  0008 0009 000A 000B 000C 000D 000E 000F  .....
0120  0010 0011 0012 0013 0014 0015 0016 0017  .....
brk:0>■
```

(6) MEM M コマンド

MEM M partition word	
基数	partition:H word:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

word : 複写先開始アドレス
 partition : 複写元アドレス範囲

MEM M コマンドは、マッピングされている 0H-0FE7FH 番地のメモリ内容の複写をします。

メモリ内容の複写は、複写範囲で指定されたメモリの内容を複写先アドレスで指定されたアドレス以降に複写します。

例 メモリ内容を複写

brk:0>MEM M 100,10F 120 ☐ ← 100H-10FH までを 120H 以降にコピー
 brk:0>■

brk:0>MEM D 100,13F ☐ ←コピー後のメモリ内容を表示

0100	31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36	1234567890123456
0110	37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32	7890123456789012
0120	31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36	1234567890123456
0130	09 00 01 02 03 04 05 06 07 08 09 00 01 02 03 04

brk:0>■

(7) MEM V コマンド

MEM V partition word	
基数	partition:H word:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

partition : 比較元アドレス範囲
 word : 比較先開始アドレス

MEM V コマンドは、マッピングされている 0H-0FE7FH 番地のメモリ内容の比較をします。

メモリ内容の比較は、比較範囲で指定されたメモリの内容を比較先アドレスで指定されたアドレス以降のメモリの内容と比較します。

比較の結果、メモリ内容が一致しない場合はそのアドレスとデータが表示されます。

例 メモリ内容を比較

```
brk:0>MEM V 100,10F 110 [C] ← 100H-10FH までを 110H 以降と比較
Source  distination
0100 01-0110 30
0101 02-0111 41
0105 06-0115 00
010D 04-011D 01
brk:0>■

brk:0>MEM D 100,11F [C]
0100  01 02 03 04 05 06 07 08 09 00 01 02 03 04 05 06 .....
0110  00 01 02 03 04 05 06 07 08 09 00 01 02 03 04 05 06 .....
brk:0>■
```

← 一致しなかった比較元と比較先のアドレスとデータを対で表示

(8) MEM X コマンド

MEM X partition word	
基数	partition:H word:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

partition : 交換元アドレス範囲
 word : 交換先開始アドレス

MEM X コマンドは、マッピングされている 0H-0FE7FH 番地のメモリ内容の交換をします。

メモリ内容の交換は、交換範囲で指定されたメモリの内容を交換先アドレスで指定されたアドレス以降のメモリの内容と交換します。

例 メモリ内容を交換

```
brk:0>MEM D 100,12F ☐
0100  31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36      1234567890123456
0110  37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32      7890123456789012
0120  03 04 05 06 07 08 09 00 01 02 03 04 05 06 07 08      .....
```

```
brk:0>MEM X 100,10F 120 ☐ ← 100H-10FH までを 120H 以降と交換
brk:0>■
```

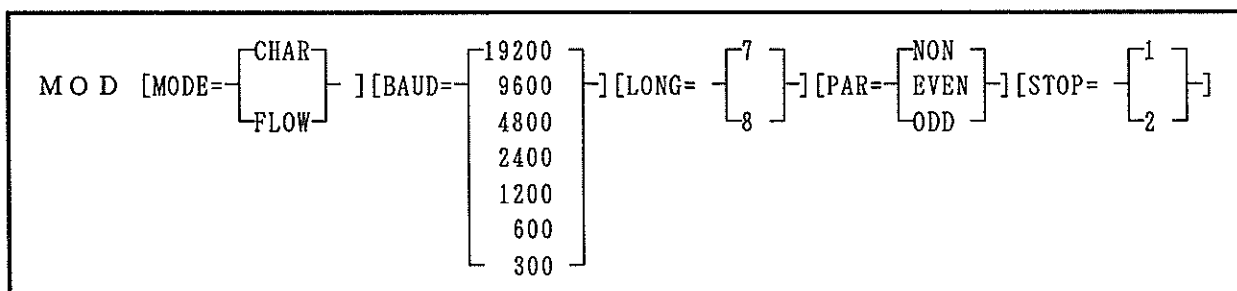
```
brk:0>MEM D 100,12F ☐ ←交換後のメモリ内容の表示
0100  03 04 05 06 07 08 09 00 01 02 03 04 05 06 07 08      .....
```

```
0110  37 38 39 30 31 32 33 34 35 36 37 38 39 30 31 32      7890123456789012
```

```
0120  31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36      1234567890123456
```

```
brk:0>■
```

8. 27 チャンネル2のモードの設定 (MOD)



trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

- MODE : ハンドシェイク・モード
- BAUD : ボー・レート
- LONG : キャラクタ長
- PAR : パリティ・ビット
- STOP : ストップ・ビット

MODコマンドはチャンネル2の通信モードの設定をします。通信モードの設定はコマンド入力時に1ラインで設定する方法と、対話形式で設定する方法があります。

MOD と入力した場合、通信モードの設定を対話形式で行います。

例1. チャネル2の通信モードを1ライン設定

```
brk:0>MOD MODE=CHAR BAUD=4800 LONG=8 PAR=NON STOP=2 [Enter] ← 1ラインで設定
brk:0>■
```

2. チャネル2の通信モードを対話形式で設定

```
brk:0>MOD [Enter] ←対話形式で設定
Mode CHAR = FLOW [Enter] ←フロー制御に変更
Baud 4800 = 9600 [Enter] ←ボー・レートを9600ボーに変更
Long 8 = [Enter] ←キャラクタ長は変更しない
Par NON = EVEN [Enter] ←偶数パリティに変更
Stop 2 = 1 [Enter] ←ストップ・ビットを1に変更
brk:0>■
```

8. 28 代替メモリとユーザ・メモリ間のデータ転送 (MOV)

MOV	$\left[\begin{array}{c} U \\ I \end{array} \right]$	partition word [\$V]
基数	word:H	partition:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

- MOV U : 代替メモリ→ユーザ・メモリ
- MOV I : ユーザ・メモリ→代替メモリ
- partition : 転送元アドレス範囲
- word : 転送先開始アドレス
- \$V : ベリファイ指定

MOV コマンドは、IE-78327-R 代替のメモリ、ターボ・アクセス・マネージャ代替メモリ、ライト・プロテクト付き代替メモリとターゲット・システムとのメモリ間で、partition で指定されたアドレス範囲のメモリ内容を word を先頭とするメモリ・エリアへ転送します。

MOV U コマンドは、IE-78327-R の代替メモリ、ターボ・アクセス・マネージャ代替メモリ、ライト・プロテクト付き代替メモリからターゲット・システムのメモリへデータ転送をします。

MOV I コマンドは、ターゲット・システムのメモリから IE-78327-R の代替メモリ、ターボ・アクセス・マネージャ代替メモリ、ライト・プロテクト付き代替メモリへデータ転送をします。

転送先に指定されるメモリのアドレス範囲は、マッピングされている必要がありますが、転送元はマッピングされている必要はありません。

コマンド行の最後に \$V を指定することで、ベリファイ機能が付加されます。

例1. 代替メモリからユーザ・メモリへ、ベリファイ付きでのメモリ内容の転送

```
brk:0>MOV U 4000,7FFF 4000 $V [Q] ←4000H-7FFFH の代替メモリのデータをターゲット・システム内の 4000H 番地以降の RAM へ、ベリファイ付きで転送します
brk:0>■
```

2. ユーザ・メモリから代替メモリへ、メモリ内容の転送

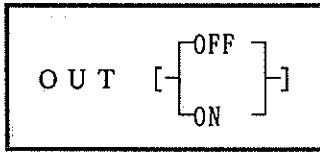
```
brk:0>MAP [Q]
0000 - 3FFF Internal ROM
4000 - 7FFF R/W emulation ←転送先となるメモリは、代替メモリ・マッピングとなっています
8000 - FC7F Non map
FC80 - FFFF <Internal RAM>
brk:0>
brk:0>MOV I 4000,7FFF 4000 [Q] ← 4000H-7FFFH のターゲット・システムのメモリのデータを代替メモリの 4000H 以降のメモリへ転送します
brk:0>■
```

3. 正しくデータ転送ができなかった場合

```
brk:0>MOV U 4000,7FFF 4000 $V [Q]
6200H
6300H
7000H
brk:0>■
```

] ←正常にデータ転送ができなかった番地を表示

8. 29 トリガ信号の外部出力指定 (OUT)



trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

- OFF : 外部センス・クリップのNo. 1をトレース信号入力とします。
- ON : 外部センス・クリップのNo. 1をトリガ信号出力とします。

OUT コマンドは、外部センス・クリップのNo. 1をトレース信号入力として扱うか、トリガ信号出力として扱うかを指定します。

OUT ☒ と入力した場合は、現在の状態を表示します。

OUT コマンドにより、外部センス・クリップのNo. 1をトリガ信号出力とする (OUT ON を実行する) 場合は、この外部センス・クリップをターゲット・システムの信号を出力するラインに接続しないでください。ターゲット・システムや IE-78327-R を破損する原因になります。

例1. 出力指定

```
brk:0>OUT ON (2)      ←トリガ信号出力に設定
brk:0>■
```

2. 出力ラインに接続されている場合

```
brk:0>OUT ON (2)
外部トリガ出力がショートしています。(204)注 ←信号の衝突がおきるため
brk:0>■                                       コマンドは受け付けない
```

3. 現在の状態を表示

```
brk:0>OUT (2)          ←現在の状態を表示
OFF                    ←トレース信号入力指定になっています
brk:0>■
```

注 () 内の数字はエラー・メッセージの番号を表しています。

8.30 パス条件の設定 (PAS)

PAS [pass8]	
基数	pass8 : T

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

pass8 : パス条件 (パス・カウント数)

PAS コマンドは、イベント発生回数を設定します。

PAS と入力した場合、設定されているパス条件を表示します。

例1. パス条件を設定する場合

```
brk:0>PAS 7       ←パス・カウント数を7回に設定します
brk:0>■
```

2. 設定されているパス条件を表示する場合

```
brk:0>PAS 
7T      ←7回が設定されています
brk:0>■
```

8.31 PROMプログラムの制御、制御キャラクタの変更、解除 (PGM)

PGM [C]

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

C : 制御キャラクタの変更、または解除の指定

PGMコマンドは、チャンネル2に接続したNEC製PROMプログラマ(PG-1500、2000)を遠隔操作するため、IE-78327-Rを端末モードにします。

端末モードへ移行する場合は PGM ☒ を入力します。

端末モードを終了する場合、^Zを入力します。

- 注意1. チャンネル2にNEC製PROMプログラマ(PG-1500、2000以外の外部装置)を接続して、IE-78327-Rを端末として使用する場合は、次の制御キャラクタが使用できません。

^A ^B ^C ^D ^E ^F ^H ^I ^J
^K ^L ^M ^N ^Q ^S ^W ^Z

上記の制御キャラクタを使用する場合、PGM C ⊞ と入力して、制御キャラクタの使用制限解除や、一部の制御キャラクタの変更を行ってください。

2. 制御キャラクタの変更は、対話形式で行います。
特定のキャラクタとして、次の16種類のキャラクタが使用できます。

^A ^B ^E ^F ^G ^N ^O ^P
^R ^T ^U ^V ^W ^X ^Y ^Z

同一のキャラクタは重複して設定できません。
DELキーまたは^Hを入力すると、初期値に戻ります。

<ESC> 入力による変更中断の場合、それまでに変更したキャラクタを無効にします。

3. 制御キャラクタの使用制限解除の変更を行うと、自動的に端末モードとなります。

例1. チャンネル2にPG-1500を接続してIE-78327-Rを端末として使用

```
brk:0>PGM C
Beginning of PGM mode      ← PGMコマンド開始メッセージ
PG>                          ← PGから出力されたプロンプト

PG>^Z C                    ← 端末モードを終了します
Exit PGM mode (Y/N) Y C ← 端末モード終了確認メッセージに対しYを入力
Termination of PGM mode    ← 端末モード終了メッセージ
brk:0>■
```

2. すべての制御キャラクタを送受信可能とする

```
brk:0>PGM C C
Put through all control character (Y/N) Y C ← この場合、すべての制御
Beginning of PGM mode                               キャラクタを送受信可能

^Z                                                    ← ^Zが入力された場合、以下のメッセージを出力
EXIT PGM mode (Y/N) Y C ← この場合CH2に^Zを送信し、端末モードを
                                                    継続します

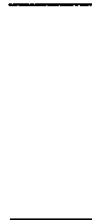
^Z
EXIT PGM mode (Y/N) Y C ← 端末モードを終了する場合、Yを入力します
Termination of PG mode ← 端末モード終了メッセージ
brk:0>■
```

例3. 一部の制御キャラクタを変更

brk:0>PGM C

Put through all control character (Y/N) N

```
Termination of "PGM"      ... ^Z
Beginning   of "HEX LOAD" ... ^A
Beginning   of "HEX SAVE" ... ^E
Beginning   of "SYM LOAD" ... ^N
Termination of "LOAD"     ... ^B
Termination of "SAVE"     ... ^F
Break       of "LOAD/SAVE"... ^W
```



←初期値を表示

Termination of "PGM" ... ^Z ■ ←現在の値を表示し、入力待ちになります

- この状態で任意のキャラクタを入力します。

```
Termination of "PGM"      ... ^A  ← ^Z を ^A に変更
Beginning   of "HEX LOAD" ... ^Z  ← ^A を ^Z に変更
Beginning   of "HEX SAVE" ... ^N  ← ^E を ^N に変更
Beginning   of "SYM LOAD" ... ^E  ← ^N を ^E に変更
Termination of "LOAD"     ... ^B  ←  入力に変更しない
Termination of "SAVE"     ... ^X  ← ^F を ^X に変更
Break       of "LOAD/SAVE"... ^G  ← ^W を ^G に変更
```

- ひとつおりの変更が終わると次のメッセージを表示してPGMモードになります。

Beginning of PGM mode

8. 3 2 サンプル・アドレスの設定 (PSA)

PSA [word][word][word]

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

word : サンプル・アドレス

PSA コマンドは、サンプリングしたい内部RAMのアドレスを設定します。

サンプル・アドレスは、0FE00H-0FEFFH 番地の範囲で3ポイントまで設定することができます。データは、ワードでサンプリングしているため奇数アドレスを設定した場合、そのアドレスと、一番地前のアドレスのペアとなります。

PSA と入力した場合、設定されているアドレスを表示します。

サンプル・アドレスが設定されていない場合、--と表示します。

例1. 3ポイントのパラメータ・アドレスを設定する場合

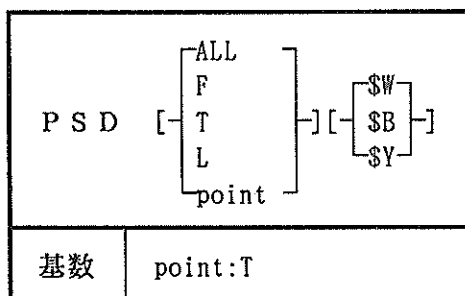
```
brk:0>PSA 0FE40 0FE42 0FE44     ←サンプル・アドレス3個を設定
brk:0>■
```

```
brk:0>PSA 
0FE40 0FE42 0FE44    ←偶数アドレスに続く奇数アドレスも設定
brk:0>■              されています
```

2. 現在の設定状態を表示

```
brk:0>PSA     ←現在の設定を表示
--
brk:0>■
```

8.33 サンプル・データの表示 (PSD)



trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

- ALL : 最初のサンプル・データからサンプル・データをすべて表示
- F : サンプル・データの最初から表示
- T : サンプル・データのイベント検出点から最後5行表示
- L : サンプル・データの最後から前に表示
- point : 指定したサンプル・フレームNoより表示
- \$W : ワード表示指定
- \$B : バイト表示指定
- \$Y : ビット表示指定

PSDコマンドは、サンプラ・メモリに書き込まれているサンプル・データを表示します。このサンプル・データは、SPAコマンドであらかじめ設定した3ポイントの内部RAMデータをPSTコマンドで設定したインターバルで、リアルタイム・エミュレーション中にサンプルしたデータです。

サンプル・データの表示は、現在のサンプル・ポイントまたは指定するサンプル・ポイントを基準として表示を始めます。

エミュレーション終了時には、サンプル・ポイントは、最終のフレームに位置しています。

サンプル表示エリアの変更のためのサンプル・ポイント指定には、次の2種類があります。


- ・ サンプル・データ表示後の促進入力時にポイントを指定する
- ・ SPDコマンド入力時にオペランドで指定する



以下に、促進入力時のサンプル・ポイント指定機能一覧を示します。

	促進入力 コマンド	機 能
絶対アドレス	frame No	指定したフレームより11行表示
+ / cr / - の移動	+ / c r	直前に表示した後の11行表示
	-	直前に表示した最初のフレームから前の11行表示
サンプル・ポイント の操作	F	最初の11行を表示
	L	最後の11行を表示
トリガ検索	T	トリガ・フレームの前後11行を表示

外部データのデータ表示は、表示指定に関係なくビット表示となります。

- 注意1. PSAコマンドでサンプル・アドレス設定終了後、一度ライト動作が行われるまでは、サンプル・データは不定値となります。
2. サンプル・データは再リアルタイム実行によりクリアされます。
 3. シングル・ステップ実行中、または、プロシージャ実行中は内部RAMデータ・サンブラは停止します。

PSD  と入力した場合、カレント・ポイントから11行表示します。

データ・サンプリングの終了時の PSD  と入力した場合、PSD T  と同じ内容となります。

表示指定 [\$W/\$B/\$Y] を省略した場合は、WRDコマンドで設定されたメモリ・ワード長で表示します。

例1. サンプル終了直後のサンプル・データをバイト単位で指定表示
(WRD コマンドでB指定の場合)

```
brk:0>PSD [Q]
frame  FE40  FE41  FE42  FE43  FE44  FE45
0051   00   23   CF   23   A0   CF
0052   00   23   CA   27   A0   CF
0053   00   23   C5   2C   A0   CF
0054   00   23   C0   33   A0   CF
0055   00   23   B8   30   A0   CF
T0056  00   23   B4   2F   A0   CF
0057   00   23   A6   35   A0   CF
Total frame = 0057 (F/L/T+/cr/-/Frame No./.) ? F [Q]
0000   45   23   AA   23   A0   CF
0001   45   23   AA   27   A0   CF
0002   45   23   AA   2C   A0   CF
0003   45   23   AA   33   A0   CF
0004   45   23   AA   30   A0   CF
0005   00   23   AA   23   A0   CF
0006   00   23   CA   27   A0   CF
0007   00   23   C5   2C   A0   CF
0008   00   23   C0   33   A0   CF
0009   00   23   B8   30   A0   CF
0010   00   23   B8   30   A0   CF
Total frame = 0057 (F/L/T+/cr/-/Frame No./.) ? <ESC>
brk:0>■
```

2. サンプル終了直後のサンプル・データをワード単位で指定表示
(WRD コマンドでW指定の場合)

```
brk:0>PSD [Q]
frame  FE40  FE42  FE44
0051  2300  23CF  CFA0
0052  2300  27CA  CFA0
0053  2300  2CC5  CFA0
0054  2300  33C0  CFA0
0055  2300  30B8  CFA0
T0056 2300  2FB4  CFA0
0057  2300  35A6  CFA0
Total frame = 0057 (F/L/T+/cr/-/Frame No./.) ? <ESC>
brk:0>■
```

例3. \$W 指定表示

```
brk:0>PSD 5 $W ␣
frame  FE42      FE44
0005   23CF      CFA0
0006   2567      CFA0
0007   2568      CFA0
0008   2569      CFA0
Total frame = 0008 (F/L/T+/cr-/Frame No./.) ? <ESC>
brk:0>■
```

4. \$B 指定表示

```
brk:0>PSD 5 $B ␣
frame  FE42      FE43      FE44      FE45
0005   CF        23        A0        CF
0006   67        25        A0        CF
0007   68        25        A0        CF
0008   69        25        A0        CF
Total frame = 0008 (F/L/T+/cr-/Frame No./.) ? <ESC>
brk:0>■
```

5. \$Y 指定表示

```
brk:0>PSD 5 $Y ␣
frame  FE42      FE43      FE44      FE45
0005   11001111 00100011 10100000 11001111
0006   01100111 00100101 10100000 11001111
0007   01101000 00100101 10100000 11001111
0008   01101001 00100101 10100000 11001111
Total frame = 0008 (F/L/T+/cr-/Frame No./.) ? <ESC>
brk:0>■
```

例6. P S A指定アドレス1ポイントの場合

```
brk:0>PSD 51 [Enter]
frame   FE40
  0051   2300
  0052   2300
  0053   2300
  0054   2300
  0055   2300
T0056   2300
  0057   2300
Total frame = 0057 (F/L/T+/cr-/Frame No./.) ? <ESC>
brk:0>■
```

7. サンプル・データなしの場合

```
brk:0>PSD T [Enter]
サンプル・データがありません。注(206)
brk:0>■
```

8. P S A指定アドレスなしの場合

```
brk:0>PSD T [Enter]
P S Aの設定がありません。注(205)
brk:0>■
```

注 () 内の数字はエラー・メッセージの番号を表します。

8.34 サンプル・タイミングの設定 (PST)

P S T [$\left[\begin{array}{c} \text{number} \\ .4 \\ .6 \\ .8 \end{array} \right]$]	
基数	number : T

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

number : サンプル・タイミング
 .4 : 0.4 μ sec
 .6 : 0.6 μ sec
 .8 : 0.8 μ sec

PSTコマンドは、内部RAMデータのサンプル・タイミングを設定します。

サンプル・タイミングは、0.4、0.6、0.8 μ sおよび1-10000 μ sまで1 μ s単位に設定できます。

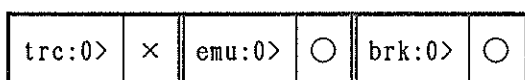
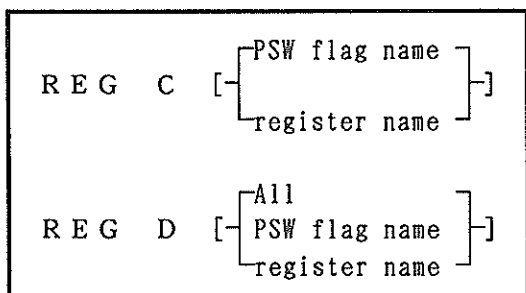
例1. サンプル・タイミングを0.8 μ Sに設定する場合

```
brk:0>PST .8 
brk:0>■
```

2. 現在の設定を表示

```
brk:0>PST 
0.8usec
brk:0>■
```


8.35 レジスタの操作 (REG)



- ALL : 全レジスタ・バンクの全レジスタ指定
- PSW flag name : UF、RBS2、RBS1、RBS0、S、Z、RSS、AC、IE、P/V、LT、CY
- register name : PC、SP、PSW、R0、R1、R2、R3、R4、R5、R6、R7、RP4、RP5、RP6、RP7、X、A、B、C、VP、UP、DE、HL

REG コマンドは、汎用レジスタおよびPC、SP、PSWの表示または変更をします。

PSW (プログラム・ステータス・ワード) については、フラグ単位での変更または表示が可能です。

REG  と入力した場合、カレント・レジスタ・バンクのレジスタとPC、SP、PSWの内容を表示します。

注意 トレース(trc:0>)またはエミュレーション(emu:0>)時にレジスタの操作(読み出し、書き込み)を行うと、エミュレーションCPUの実行は一時中断されます。

(1) REG C コマンド

REG C [<table style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 0 5px;">{</td> <td style="padding: 0 5px;">PSW flag name</td> <td style="padding: 0 5px;">}</td> </tr> <tr> <td style="padding: 0 5px;">{</td> <td style="padding: 0 5px;">register name</td> <td style="padding: 0 5px;">}</td> </tr> </table>]	{	PSW flag name	}	{	register name	}
{	PSW flag name	}				
{	register name	}				

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

PSW flag name : UF、RBS2、RBS1、RBS0、S、Z、RSS、AC、
IE、P/V、LT、CY

register name : PC、SP、PSW、R0、R1、R2、R3、R4、R5、
R6、R7、RP4、RP5、RP6、RP7、X、A、B、C、
VP、UP、DE、HL

REG C コマンドは、指定されたレジスタから順番に内容を変更します。

レジスタ名にPSWが指定された場合、UFからCYの順に1ビットごとに変更ができます。PSWのフラグ名が直接指定された場合、指定されたフラグ名から順番に変更します。

REG C と入力した場合、カレント・レジスタ・バンクの汎用レジスタ (R0-R7、RP4-RP7) とPC、SPの内容を順番に変更します。

レジスタの内容を変更しない場合は を、レジスタ内容の変更を終了する場合は <ESC> を入力します。

例1. カレント・レジスタ・バンクの汎用レジスタとPC、SPの内容を変更

```
brk:0>REG C [0]
RO(X)      00 = 10 [0]
R1(A)      10 = 20 [0]
.
.
SP          9000 = [0]      ←変更しません
brk:0>■
```

2. PSWの全フラグを順番に変更

```
brk:0>REG C PSW [0]
UF          0 = 1 [0]
RBS2       1 = 0 [0]
.
.
CY          0 = 1 [0]
brk:0>■
```

3. PSWのフラグを指定して内容を順番に変更

```
brk:0>REG C RBS2 [0]
RBS2       1 = 0 [0]
RBS1       1 = [0]
RBS0       0 = 1 [0]
S          1 = <ESC>      ←変更を終了します
brk:0>■
```

(2) REG D コマンド


REG D [<table style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 0 5px;">[</td> <td style="padding: 0 5px;">ALL</td> <td style="padding: 0 5px;">]</td> </tr> <tr> <td style="padding: 0 5px;">[</td> <td style="padding: 0 5px;">PSW flag name</td> <td style="padding: 0 5px;">]</td> </tr> <tr> <td style="padding: 0 5px;">[</td> <td style="padding: 0 5px;">register name</td> <td style="padding: 0 5px;">]</td> </tr> </table>]	[ALL]	[PSW flag name]	[register name]
[ALL]							
[PSW flag name]							
[register name]							

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

- ALL : 全レジスタ・バンクの全レジスタ指定
- PSW flag name : UF、RBS2、RBS1、RBS0、S、Z、RSS、AC、
IE、P/V、LT、CY
- register name : PC、SP、PSW、R0、R1、R2、R3、R4、R5、
R6、R7、RP4、RP5、RP6、RP7、X、A、B、C、
VP、UP、DE、HL

ALLを指定すると、全レジスタ・バンク（0-7）のすべてのレジスタ内容を表示します。

PSWを指定すると、PSWの全フラグ内容を表示します。

REG D  と入力した場合、カレント・バンクの全レジスタとPC、SP、PSWの内容を表示します。

例1. レジスタを指定して内容を表示

```
brk:0>REG D PC ☐ ← P C の内容を表示
PC      1000
brk:0>■
```

2. 全レジスタ・バンクのすべてのレジスタを表示

```
brk:0>REG D ALL ☐
PC      SP  PSW:  UF  RBS2 RBS1 RBS0  S  Z  RSS  AC  IE  P/V  LT  CY
1000    7000          0  0  1  1  0  1  0  0  0  1  0  1

      R0  R1  R2  R3  R4  R5  R6  R7      RP4  RP5  RP6  RP7
BANK0  01  02  03  04  05  06  07  08      0909  0A0A  0B0B  0C0C
BANK1  11  12  13  14  15  16  17  18      1919  1A1A  1B1B  1C1C
BANK2  21  22  23  24  25  26  27  28      2929  2A2A  2B2B  2C2C
BANK3  31  32  33  34  35  36  37  38      3939  3A3A  3B3B  3C3C
BANK4  98  76  54  32  10  AA  BB  CC      DEF0  5562  1F20  3434
BANK5  00  00  00  00  22  3F  1C  52      0006  AEFC  7000  1020
BANK6  11  22  33  44  55  66  77  88      9999  AAAA  BBBB  CCCC
BANK7  71  72  73  74  75  76  77  78      7979  7A7A  7B7B  7C7C
brk:0>■
```

3. カレント・バンクの全レジスタ内容を表示

```
brk:0>REG D ☐
PC      SP  PSW:  UF  RBS2 RBS1 RBS0  S  Z  RSS  AC  IE  P/V  LT  CY
1000    7000          0  0  1  1  0  1  0  0  0  1  0  1

      R0  R1  R2  R3  R4  R5  R6  R7      RP4  RP5  RP6  RP7
      X  A  C  B          VP  UP  DE  HL
      31  32  33  34  35  36  37  38      3939  3A3A  3B3B  3C3C
brk:0>■
```

8. 36 IE-78327-Rとエミュレーション・デバイスのリセット (RES)

RES [H]

trc:0>	<input type="radio"/>	emu:0>	<input type="radio"/>	brk:0>	<input type="radio"/>
--------	-----------------------	--------	-----------------------	--------	-----------------------

H : IE-78327-Rのリセット

RES コマンドは次の2つのリセット機能を持っています。

- ・エミュレーション・デバイスのリセット . . . RES
- ・IE-78327-Rのリセット . . . RES H

トレーシング (trc:0>) 時、またはエミュレーティング (emu:0>) 時に RES を入力すると、実行を中断してブレーク (brk:0>) します。

例1. エミュレーション・デバイスのリセット

```
brk:0>RES 
brk:0>■
```

2. IE-78327-Rのリセット

```
brk:0>RES H 
[
  IE-78327-Rの起動メッセージ
brk:0>■
```

8.37 実行 (RUN)

<pre> RUN N [word] RUN B [word] RUN T [word][, ^{注1}] [REG] [TRD] _{step16} RUN T [word][, ^{注1}] [REG] [TRD] PRC _{step16} </pre>	<p style="text-align: center;">注1</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">register name</td> <td style="padding: 2px;">=</td> <td style="border: 1px solid black; padding: 2px;">mask8</td> </tr> <tr> <td></td> <td style="padding: 2px;">></td> <td></td> </tr> <tr> <td></td> <td style="padding: 2px;"><</td> <td></td> </tr> <tr> <td></td> <td style="padding: 2px;">=></td> <td></td> </tr> <tr> <td></td> <td style="padding: 2px;">>=</td> <td></td> </tr> <tr> <td></td> <td style="padding: 2px;">=<</td> <td></td> </tr> <tr> <td></td> <td style="padding: 2px;"><=</td> <td></td> </tr> <tr> <td></td> <td style="padding: 2px;">><</td> <td></td> </tr> <tr> <td></td> <td style="padding: 2px;"><></td> <td></td> </tr> </table> <p style="text-align: center;">注2</p> <p style="text-align: center;">PSW flag name = bit</p>	register name	=	mask8		>			<			=>			>=			=<			<=			><			<>	
register name	=	mask8																										
	>																											
	<																											
	=>																											
	>=																											
	=<																											
	<=																											
	><																											
	<>																											
<p>基数</p>	<p>word:H step16:T mask8:H mask16:H bit:Y</p>																											

trc:n>	×	emu:n>	×	brk:n>	○
--------	---	--------	---	--------	---

- 注1. 上記の表に置き換えてください。
 2. マスク表現の場合は = <> >< のみ有効です。

- RUN N : ノンブレーク・リアルタイム実行
 RUN B : ブレーク条件付きリアルタイム実行
 RUN T : ステップ実行
 RUN T PRC : プロシージャ実行
 word : 実行開始アドレス
 PRC : プロシージャ実行指定
- register name : PC, SP, R0, R1, R2, R3, R4, R5, R6, R7, RP4, RP5, RP6, RP7, X, A, B, C, VP, UP, DE, HL
- PSW flag name : UF, RBS2, RBS1, RBS0, S, Z, RSS, AC, IE, P/V, LT, CY
- REG : レジスタ表示指定
 TRD : トレース表示指定
 bit : 1ビットの数値
 step16 : 16ビット・ステップ数
 mask8 : 8ビット・マスク・データ
 mask16 : 16ビット・マスク・データ

RUNコマンドは、エミュレーションCPUによるターゲット・プログラムの実行を開始します。

サブコマンド、またはオペランドを指定することで、4種類の実行機能が選択できます。

実行開始アドレスは、オペランドで指定します。

0FDFH を越えた実行開始アドレスの指定はできません。

(1) RUN N コマンド

RUN N [word]	
基数	word:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

word : 実行開始アドレス

RUN N コマンドは、指定されたアドレスからターゲット・プログラムを実行して、ディレイ条件の成立により、リアルタイム・トレーサと内部RAMデータ・サンブラを停止します。

この時、実行中のターゲット・プログラムは、STP コマンド、または、強制ブレーク（フェイル・セーフ・ブレーク）以外では停止できません。

実行開始アドレスが省略された場合、現在のPCの値が実行開始アドレスになります。

次ページにRUN N コマンドの動作状態例を示します。

注意 トレース(trc:0>)およびエミュレーション(emu:0>)時に、レジスタ、メモリ、SFRの内容の操作を行うコマンドを入力すると、エミュレーションCPUの実行は一時中断されます。

図8-2 ノンブ레이크・リアルタイム実行でのエミュレーションCPU、イベント検出器、アナライザ表示プロンプト、および外部トリガ出力の動作状態

発生イベント		実行開始	インブール 条件	ENB	クロック トリガ 条件	TRX	クロック トリガ 条件	BRM	チェック ポイント 条件	CHK	ハイスループ 条件	DSB	パルス条件 DLY	トリガ スタート TRG	トリガ ストップ STP T	エミュ 終了 STP	
ハードウェア	動作	RUN N	↓	→	→	→	→	→	→	→	→	↓	↓	↓	↓	↓	
	割り込み	停止	リアルタイム実行	→	→	→	→	→	→	→	→	→	→	→	→	→	停止
		保留	受け付け	→	→	→	→	→	→	→	→	→	→	→	→	→	保留
表示プロンプト		brk:0>	trc:0>	→	→	→	→	→	→	→	→	→	→	→	→	→	brk:0>
イベント検出器		停止	検出	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	リアルタイム・モード	停止	検出	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	区間モード	停止	検出	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	リアルタイム・モード	停止	検出	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	リアルタイム・モード	停止	検出	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	リアルタイム・モード	停止	検出	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	リアルタイム・モード	停止	検出	→	→	→	→	→	→	→	→	→	→	→	→	→	→
アナライザ		停止	検出	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	内部RAMデータ・トリガ	停止	トリガ・タイム	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	外部トリガ	停止	トリガ・タイム	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	実行時間測定	停止	測定	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	実行命令数測定	停止	測定	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	C0カバレレッジ	停止	測定	→	→	→	→	→	→	→	→	→	→	→	→	→	→
	外部トリガ出力	Low	→	→	→	→	→	→	→	→	→	→	→	→	→	→	→

備考 上の図中の→は、左側の動作状態が続くことを表します。

例1. ディレイ条件の成立によりアナライザのみ停止した場合

```
brk:0>RUN N 200 (Q)      ← 200H 番地からターゲット・プログラムを実行
User-system Vcc-ON      Emulation start at 0200
trc:0>
emu:0>■
```

2. S T P T コマンドでアナライザを停止した場合

```
brk:0>RUN N 200 (Q)
User-system Vcc-ON      Emulation start at 0200
trc:0>STP T (Q)        ←アナライザを停止
emu:0>■
```

3. S T P コマンドでエミュレーションCPUの実行
及びアナライザを停止した場合

```
brk:0>RUN N 200 (Q)
User-system Vcc-ON      Emulation start at 0200
trc:0>STP (Q)          ←エミュレーションCPUの実行を停止
Escape break terminated
PC   SP  PSW: UF  RBS2 RBS1 RBS0  S   Z   RSS  AC  IE  P/V  LT  CY
0109 7000      0   0   1   1   0   1   0   0   0   1   0   1
R0  R1  R2  R3  R4  R5  R6  R7      RP4  RP5  RP6  RP7
X   A   C   B      VP   UP   DE   HL
30  10  20  30  20  20  60  70      1000 2000 3000 4000
brk:0>■
```

S T P コマンドでアナライザとエミュレーションCPUの実行を停止すると、その時点でのカレント・レジスタ・バンクのレジスタ内容を表示します。

(2) RUN B コマンド

RUN B [word]	
基数	word:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---


word : 実行開始アドレス


RUN B コマンドは、指定されたアドレスからターゲット・プログラムの実行を開始し、ディレイ条件の成立により、エミュレーションCPUとリアル・タイム・トレーサ、内部RAMデータ・サンブラを停止します。

実行開始アドレスが省略された場合、現在のPC（カレントPC）の値が実行開始アドレスになります。

アナライザとエミュレーションCPUの実行を強制的に停止する場合、STP コマンドを入力します。

ディレイの成立により、エミュレーションCPUとアナライザが停止すると自動的に1ステップ実行、または、プロシージャ実行モードとなります。

この状態で  を入力すると、次の1命令が実行されます。

/  を入力すると、プロシージャ実行を行います。

1ステップ実行、または、プロシージャ実行を中断する場合は、<ESC> を入力してください。

次ページにRUN B コマンドの動作状態例を示します。

図8-3 ブレーク条件付きリアルタイム実行でのエミュレーションCPU、イベント検出器、アナライザ表示プロンプト、外部トリガ出力の動作状態

発生イベント		実行開始 RUN B	ブレーク 条件 ENB	クォリファイ トリス条件 TRX	トリガ 条件 BRM	チェック ポイント条件 CHK	ディスアブル 条件 DSB	パス 条件 PAS	デレイ 条件 DLY	再実行 再実行 <cr>or</cr>	エミュレーション 終了 <ESC>
ハードウェア	動作	停止	リタイム実行	→	→	→	リタイム実行	→	→	停止	→
	割り込み	保留	受け付け	→	→	→	受け付け	→	→	保留	→
表示プロンプト		brk:0>	trc:0>	→	→	→	→	→	→	なし	brk:0>
イベント検出器	リタイム・トリス	停止	検出	→	→	→	検出	→	→	→	→
	全トリス・モード	停止	トリス	→	→	→	トリス	→	→	停止	→
アナライザ	区間トリス・モード	停止	→	→	→	→	トリス	停止	→	→	→
	クォリファイ・トリス・モード	停止	→	トリス	停止	→	→	→	→	→	→
内部RAMデータ・サンプル	チェックポイントのトリス	停止	→	→	→	→	トリス	→	→	→	→
	サンプル・メモリ	停止	サンプル・タイム ゲでの書き込み	→	→	→	サンプル・タイム ゲでの書き込み	→	→	停止	→
パフォーマンス測定	実行時間測定	停止	→	測定	→	→	測定	停止	→	→	→
	実行命令数測定	停止	→	測定	→	→	測定	停止	→	→	→
外部トリガ出力	C0カバレッジ	停止	測定	→	→	→	測定	→	→	停止	→
		Low	→	→	→	→	→	High	Low	→	→

備考1. 上の図中の→は、左側の動作状態が続くことを表します。
 2. <cr> はリターン・キーの入力を示します。

例 トリガ発生によりアナライザとエミュレーションCPUの実行が停止した場合

```

brk:0>RUN B 100 [Enter]          ← 100H 番地からターゲット・プログラムを実行
User-system Vcc-ON      Emulation start at 0100
trc:0>
Execution break terminated ←トリガ発生によりエミュレーションCPUの実行を停止
PC   SP  PSW: UF  RBS2 RBS1 RBS0 S  Z  RSS AC  IE  P/V LT  CY
0109 7000      0  0  1  1  0  1  0  0  0  1  0  1
R0  R1  R2  R3  R4  R5  R6  R7      RP4  RP5  RP6  RP7
X   A   C   B                VP   UP   DE   HL
30  10  20  30  20  20  60  70      1000 2000 3000 4000
One step emulation standby [Enter] ←1ステップ実行を開始
Frame Status Address Data Label Mnemonic
0022      010A      MOV R1,#3H
PC   SP  PSW: UF  RBS2 RBS1 RBS0 S  Z  RSS AC  IE  P/V LT  CY
010A 7000      0  0  1  1  0  1  0  0  0  1  0  1
R0  R1  R2  R3  R4  R5  R6  R7      RP4  RP5  RP6  RP7
X   A   C   B                VP   UP   DE   HL
30  03  20  30  20  20  60  70      1000 2000 3000 4000
One step emulation standby <ESC> ←1ステップ実行を終了
brk:0>■
    
```

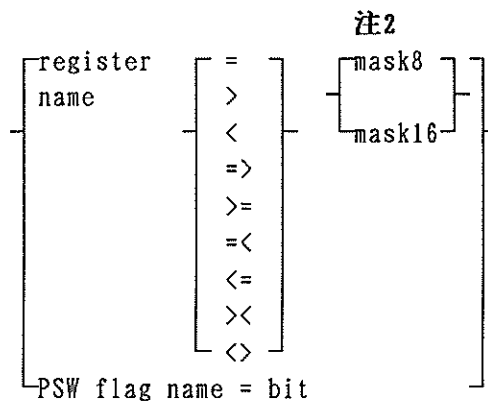
(3) RUN T コマンド

RUN T [word][, } ^{注1} } step16] [REG] [TRD]					
基数	word:H	step16:T	mask8:H	mask16:H	bit:Y

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

- word : 実行開始アドレス
- REG : レジスタ表示指定
- TRD : トレース表示指定
- step16 : 16ビット・ステップ数
- mask8 : 8ビット・マスク・データ
- mask16 : 16ビット・マスク・データ
- register name : PC, SP, R0, R1, R2, R3, R4, R5, R6, R7, RP4, RP5, RP6, RP7, X, A, B, C, VP, UP, DE, HL
- PSW flag name : UF, RBS2, RBS1, RBS0, S, Z, RSS, AC, IE, P/V, LT, CY
- bit : 1ビット数値

注1. 次に示す図に置き換えてください。



2. mask8, mask16 にマスク・データを使用する場合、 = <> >< の条件のみ指定できます。

R U N T コマンドは、指定されたアドレスからターゲット・プログラムを1命令ずつ実行して、実行トレースを行います。

実行開始アドレスが省略された場合、現在のP C（カレントP C）の値が実行開始アドレスになります。

R E Gを指定した場合は、実行命令ごとにカレント・レジスタの内容を表示します。

T R Dを指定した場合は、実行命令ごとに実行結果の逆アセンブルの内容を表示します。

ステップ実行は、指定されたレジスタ条件の成立、指定命令数の実行完了まで断続的に継続されます。

ステップ実行終了後は、1ステップ実行、または、プロシージャ実行モードとなります。

この状態で **Ⓚ** を入力すると、次の1命令が実行されます。

/ **Ⓚ** を入力すると、プロシージャ実行を行います。

1ステップ実行、または、プロシージャ実行を中断する場合は、<ESC> を入力してください。

次ページにR U N T コマンドの動作状態図を示します。

注意 ステップ実行中の割り込みはすべて保留されます。

図8-4 ステップ実行状態での、エミュレーションCPU、イベント検出器、アナライザ、表示プロンプト、外部トリガ出力の動作状態

発生イベント		コマンド入力 RUN T	再実行 ⓐor/ⓑ	エミュレーション終了 <ESC>
ハードウェア エミュレーション CPU	動作	停止	ステップ 実行	ステップ実行 プログラムの実行
	割り込み	保留	→	→
表示プロンプト		brk:0>	なし	→
イベント検出器		停止	→	→
アナライザ	全トレース・モード	停止	トレース	全トレース メインループの トレース
	区間トレース・モード	停止	トレース	全トレース メインループの トレース
	クオリファイ・トレース・モード	停止	トレース	全トレース メインループの トレース
	チェックポイントの トレース	停止	→	→
内部RAM データ・ サンプル	サンプル・ メモリ	停止	→	→
パフォーマンス 測定	実行時間 測定	停止	→	→
	実行命令 数測定	停止	→	→
C0加算レジスタ		停止	測定	→
外部トリガ出力		Low	→	→

備考 上の図中の → は、左側の動作状態が続くことを表します

例 ステップ実行

```
brk:0>RUN T 100,R1=1 TRD ⓐ          ← 100H 番地からターゲット・プログラムを実
                                         行R1=1 で実行を停止

User-system Vcc-ON      Emulation start at 0100
Frame Status Address Data          Label Mnemonic
0000          0100                MOVW   SP,#0FE00H
0006          0104                MOV    R0,#0H
0008          0106                MOV    R1,#1H
terminated
PC   SP  PSW: UF  RBS2 RBS1 RBS0  S   Z   RSS AC  IE  P/V LT  CY
0108 FE00      0   0   1   1   0   1   0   0   0   1   0   1
   R0  R1  R2  R3  R4  R5  R6  R7          RP4  RP5  RP6  RP7
   X   A   C   B                      VP   UP   DE   HL
   03  01  20  30  20  20  60  70          1000  2000  3000  4000
One step emulation standby ⓑ          ← 1 ステップ実行を開始
Frame Status Address Data          Label Mnemonic          EX
0000          0108                MOV    R2,#2H          00
PC   SP  PSW: UF  RBS2 RBS1 RBS0  S   Z   RSS AC  IE  P/V LT  CY
0108 FE00      0   0   1   1   0   1   0   0   0   1   0   1
   R0  R1  R2  R3  R4  R5  R6  R7          RP4  RP5  RP6  RP7
   X   A   C   B                      VP   UP   DE   HL
   03  01  02  30  20  20  60  70          1000  2000  3000  4000
One step emulation standby <ESC>     ← 1 ステップ実行を終了
brk:0>■
```

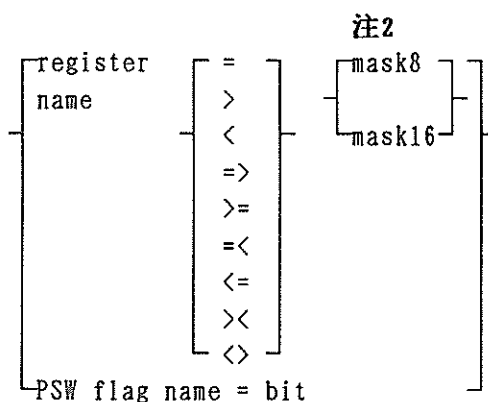
(4) RUN T PRC付きコマンド

RUN T [word][, $\left[\begin{array}{c} \text{注1} \\ \text{step16} \end{array} \right]$][REG][TRD] PRC					
基数	word:H	step16:T	mask8:H	mask16:H	bit:Y

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

- word : 実行開始アドレス
- REG : レジスタ表示指定
- TRD : トレース表示指定
- step16 : 16ビット・ステップ数
- mask8 : 8ビット・マスク・データ
- mask16 : 16ビット・マスク・データ
- PRC : プロシージャ実行指定
- register name : PC, SP, R0, R1, R2, R3, R4, R5, R6, R7, RP4, RP5, RP6, RP7, X, A, B, C, VP, UP, DE, HL
- PSW flag name : UF, RBS2, RBS1, RBS0, S, Z, RSS, AC, IE, P/V, LT, CY
- PRC : プロシージャ実行指定
- bit : 1ビット数値

注1. 次に示す図に置き換えてください。



2. mask8, 16 にマスク・データを使用する場合、 = <> >< の条件のみ指定できます。

RUN T PRC付きコマンドは、指定されたアドレスからターゲット・プログラムを1命令ずつ実行し、実行トレースを行います。このとき、実行を開始したルーチン（カレント・ルーチン）よりも深いルーチン（CALL命令とRET命令で囲まれたルーチンとコンテキスト・ルーチン）に対しては、ターゲット・プログラムはステップ実行されますが、トレースは行いません。

実行開始アドレスが省略された場合、現在のPC（カレントPC）の値が実行開始アドレスになります。

REGを指定した場合は、実行命令ごとにカレント・レジスタの内容を表示します。

TRDを指定した場合は、実行命令ごとに実行結果の逆アセンブルの内容を表示します。

ステップ実行は、指定されたレジスタ条件の成立、指定命令数の実行完了まで断続的に継続されます。

プロシージャ実行終了後は、1ステップ実行、または、プロシージャ実行モードとなります。

この状態で **Ⓚ** を入力すると、次の1命令が実行されます。

/ **Ⓚ** を入力すると、プロシージャ実行を行います。

1ステップ実行、または、プロシージャ実行を中断する場合は、**<ESC>** を入力してください。

次ページにRUN T PRC コマンドの動作状態図を示します。

注意 プロシージャ実行中の割り込みはすべて保留されます。

図8-5 プロシージャ実行での、エミュレーションCPU、イベント検出器、アナライザ、表示プロンプト、外部トリガ出力の動作状態

発生イベント		マイト入力 RUN T	再実行 ⓐor/ⓑ	エミュレーション終了 <ESC>
ハードウェア エミュレーション CPU	動作	停止	プロシージャ 実行	ステップ実行 プロシージャ実行
	割り込み	保留	→	→
表示プロンプト		brk:0>	なし	→
イベント検出器		停止	→	→
アナライザ	全トレース・モード	停止	メイン・ルチンの全トレース	全トレース メイン・ルチンのトレース
	区間トレース・モード	停止	メイン・ルチンの全トレース	全トレース メイン・ルチンのトレース
	クオリファイ・トレース・モード	停止	メイン・ルチンの全トレース	全トレース メイン・ルチンのトレース
	チェック・ポイントのトレース	停止	→	→
内部RAMデータ・サンプラ	サンプラ・メモリ	停止	→	→
パフォーマンス測定	実行時間測定	停止	→	→
	実行命令数測定	停止	→	→
COカレッジ		停止	測定	→
外部トリガ出力		Low	→	→

備考 上の図中の → は、左側の動作状態が続くことを表します。

例 プロシージャ実行

```
brk:0>RUN T 100,R0=3H TRD PRC ⓐ注
User-system Vcc-ON      Emulation start at 100
Frame Status Address Data Label Mnemonic
0000          0100                MOVW  SP,#0FE00H
0006          0104                MOV   R1,#0H
0010          0106                CALL  !NEST1
0015  WR      FDFE  01
0016  WR      FDFE  09
      terminated
PC   SP  PSW: UF  RBS2 RBS1 RBS0 S  Z  RSS AC  IE  P/V LT  CY
0109 FE00      0   0   0   0   0   0   0   0   0   0   0   0   0
      R0  R1  R2  R3  R4  R5  R6  R7          RP4  RP5  RP6  RP7
      X  A  C  B                      VP  UP  DR  HL
      03  00  00  00  00  00  00  00          0000  0000  0000  0000
One step emulation standby ⓑ ← 1 ステップ実行を開始
Frame Status Address Data Label Mnemonic
0000          0109                MOV   R3,#3H
PC   SP  PSW: UF  RBS2 RBS1 RBS0 S  Z  RSS AC  IE  P/V LT  CY
0109 FE00      0   0   0   0   0   0   0   0   0   0   0   0   0
      R0  R1  R2  R3  R4  R5  R6  R7          RP4  RP5  RP6  RP7
      X  A  C  B                      VP  UP  DE  HL
      03  00  00  00  00  00  00  00          0000  0000  0000  0000
One step emulation standby <ESC> ← 1 ステップ実行を終了
brk:0>■
```

注 100H 番地からターゲット・プログラムを実行、R0=3 で実行を停止。

8.38 オブジェクト、ディバグ環境のセーブ (SAV)

SAV	file[partition][C][D][\$V]
基数	partition:H

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

- file : ファイル名
- partition : セーブ・アドレス範囲
- C : オブジェクト指定
- D : ディバグ環境指定
- \$V : ベリファイ指定

SAVコマンドは、オブジェクト・コード、ディバグ環境をホスト・マシンの指定したファイルにセーブします。

セーブ範囲は、五つまでスペースで区切って設定することができます。

セーブ範囲が省略された場合、マッピングされているメモリ内容すべてと、内部RAMの内容(0FE00H-0FE7FH)をセーブします。

オペランドでCを指定した場合、オブジェクト・コードのセーブを行い、Dを指定した場合、ディバグ環境のセーブを行います。C、Dの指定がない場合、最初にオブジェクト・コードをセーブし、次にディバグ環境をセーブします。

オブジェクト・コード、ディバグ環境を一度にセーブすると、ファイル名で拡張子を付加することはできません。この場合、拡張子はHEX(オブジェクト・ファイル)、DBG(ディバグ環境ファイル)が自動的に付加されます。

コマンド行の最後に\$Vを指定した場合、オブジェクト・コードをセーブ後、ファイルとメモリ内容のベリファイを行います。

セーブするディバグ環境を以下に示します。

- 内部ROMサイズ
- 内部RAMサイズ
- μ PD78320、78327 + TAM emulation の設定状態
- ダウン・ロード・モード
- ディバグ環境に関する以下のコマンドの設定状態

BRA	BRD	BRS	BRM	CHK	ENB	DSB	CLK
CVM	PAS	DLY	MAP	MOD	PGM	PSA	PST
TRM	TRF	TRS	TRX	WRD			

例1. オブジェクトとディバグ環境を一度にセーブ

```
brk:0>SAV B:SAMPLE ⓐ  
object save complete  
debug condition save complete  
brk:0>■
```

□ ←正常終了した場合のメッセージ

2. セーブ範囲を指定してベリファイ付きでオブジェクト・コードをセーブ

```
brk:0>SAV B:SAMPLE.HEX 0XXX 2000,27FF C $V ⓐ注  
object save complete  
object verify complete  
brk:0>■
```

□ ←正常終了した場合のメッセージ

注 0-0FFF、2000-27FF 番地の内容をベリファイ付きでセーブ

3. ディバグ環境のセーブ

```
brk:0>SAV B:SAMPLE.DBG D ⓐ  
debug condition save complete  
brk:0>■
```

←正常終了した場合のメッセージ

例4. ドライブBに指定したファイルが、すでに存在する場合

```
brk:0>SAV B:SAMPLE.HEX C 
  既に存在するファイルです。(513)注 B:SAMPLE.HEX Delete ? (Y or N):Y 
brk:0>
```

リード、ライト属性の SAMPLE.HEX というファイルが存在している場合、上記のメッセージを表示します。

Y を入力すると SAMPLE.HEX を削除して新しく SAMPLE.HEX というファイルを作成します。

Y 以外の入力をしててもファイルは作成されません。

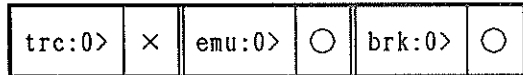
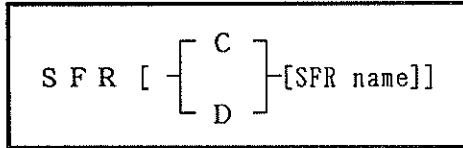
5. ドライブBに SAMPLE.HEX というファイルがオープンできない場合

```
brk:0>SAV B:SAMPLE.HEX C           ←R/O属性のファイル名を指定
  読取り専用のファイルです。(507)注 B:SAMPLE.HEX
brk:0>
```

ドライブBにR/O属性の SAMPLE.HEX というファイルが存在している場合、上記のメッセージを表示しコマンドは無視されます。

注 () 内の数字はエラー・メッセージの番号を表しています。

8. 39 SFRの操作 (SFR)



SFR C : SFRの変更
 SFR D : SFRの表示

SFR name : μ PD78320シリーズのSFR

P0, [P2], P3, P4, P5, [P7], P8, P9, [TMOLW], [CTXOLW], [CT01LW], [CT02LW], [CT03LW], CCXOLW, CC01LW, <PM0>, <PM3>, <PM5>, <PM8>, <PM9>, [TMOUW], [TM1], [CTXOUW], [CT01UW], [CT02UW], [CT03UW], CCXOUW, CC01UW, <PMC0>, RTPS, <PMC3>, <PMC8>, BRG, RTP, RTPR, PRDC, ADM, [ADCR], [ADCRH], CM00, CM01, CM02, CM03, CM10, CM11, CSIM, SBIC, SIO, ASIM, [ASIS], [RXB], <TXS>, TMC, BRGM, PRM, TOC0, TOC1, RPUM, STBC, CCW, WDM, MM, PWC, FCC, IF0, IF1, MK0, MK1, PB0, PB1, ISM0, ISM1, CSE0, CSE1, INTM0, INTM1, [ISPR], PRSL, EXTSFR0, EXTSFR1, EXTSFR2, EXTSFR3, EXTSFR4, EXTSFR5, EXTSFR6, EXTSFR7, EXTSFR8, EXTSFR9, EXTSFR10, EXTSFR11, EXTSFR12, EXTSFR13, EXTSFR14, EXTSFR15

: μ PD78327シリーズのSFR

P0, (P2), P3, P4, P5, (P7), P8, P9, (TM2), CC10, <PM0>, <PM3>, <PM5>, <PM8>, <PM9>, (TMO), (TM1), <PMC3>, <PMC8>, BRG, POL, POH, PRDC, RTPC, PWMC, PWMB, ADM, (ADCR), (ADCRH), CM00S, CM01S, CM02S, CM03S, CM04S, CM05S, CM06, CM20, CSIM, SBIC, SIO, ASIM, (ASIS), (RXB), <TXS>, CM00R, CM01R, CM02R, CM03R, CM04R, CM05R, TMC0, BRGM, TMC1, TUM, TOUT, STBC, CCW, WDM, MM, PWC, FCC, IF0, IF1, MK0, MK1, PB0, PB1, ISM0, ISM1, CSE0, CSE1, INTM0, (ISPR) PRSL, EXTSFR0, EXTSFR1, EXTSFR2, EXTSFR3, EXTSFR4, EXTSFR5, EXTSFR6, EXTSFR7, EXTSFR8, EXTSFR9, EXTSFR10, EXTSFR11, EXTSFR12, EXTSFR13, EXTSFR14, EXTSFR15

SFR コマンドは、SFRの変更、または表示を行います。
 SFR ☉ と入力した場合、SFRを表示します。

- 注意1. ()で示したSFRは、読み出し専用、< >で示したSFRは書き込み専用です。
2. emu:0> においてSFRの操作を行う場合、SFRの読み出し、書き込み時に実行が一時中断されます。

(1) SFR C コマンド

SFR C [SFR name]

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

SFR C コマンドは、SFRの変更を行います。

SFR C と入力した場合、P0-PRSL までマッピング・アドレス昇順に変更します。

オペラントでSFRを指定した場合、指定したSFRから順番に変更します。

SFRの内容を変更しない場合、 を入力します。

SFRの変更を途中でやめる場合、<ESC> を入力します。

注意 () で示したSFRは、読み出し専用、< > で示したSFRは書き込み専用です。

例1. すべてのSFRの内容を順番に変更

```
brk:0>SFR C           ←書き込み可能なSFRすべてを変更
P0          77 = 88 
P1          88 =           ←変更なし
.
.
PRSL       00 = 02 
brk:0>■
```

2. SFRを指定して内容を順番に変更

```
brk:0>SFR C ASIM         ←ASIMレジスタから変更
ASIM       03 = 04 
TXS       -- = 0AA     ←書き込み専用レジスタの場合 -- を表示
.
.
TMC0      20 = <ESC>      ←変更を終了
brk:0>■
```

3. 読み出し専用SFRを指定した場合

```
brk:0>SFR C ASIS       ←読み出し専用のSFRを指定
Read only                                     ←読み出し専用のメッセージを表示
brk:0>■
```


(2) SFR D コマンド

```
SFR D [SFR name]
```

```
trc:0> × emu:0> ○ brk:0> ○
```

SFR D コマンドは、読み出し可能なSFRを表示します。

オペランドでSFRを指定した場合、指定したSFRを表示します。
書き込み専用のSFRを指定した場合、--が表示されます。

SFR D と入力した場合、P0-PRSL までアドレス昇順に表示します。

SFRの表示は一画面に収まらないため、Next ? (Y/N) を表示し入力待ちとなります。

Y を入力すると次の画面を表示し、Y 以外を入力するとコマンドを終了します。

注意 ()で示したSFRは、読み出し専用、< >で示したSFRは書き込み専用です。

例1. すべてのSFRの内容を表示

```
brk:0>SFR D           ←読み出し可能なSFRすべてを表示
P0      P2      P3      P4      P5      P7      P8      P9      TM2     CC10    TM0
00      00      00      00      00      00      00      00      0000    0000    0000
      .
      .
      .
ISM1    CSE0    CSE1    INTM0  PRSL
2828    9876    5432    10     32

Next ? (Y/N) Y       ←次の画面を表示します

EXTSFR0  EXTSFR1  EXTSFR2  EXTSFR3  EXTSFR4  EXTSFR5  EXTSFR6  EXTSFR7
00        00        00        00        00        00        00        00

EXTSFR8  EXTSFR9  EXTSFR10 EXTSFR11 EXTSFR12 EXTSFR13 EXTSFR14 EXTSFR15
00        00        00        00        00        00        00        00
brk:0>■
```

例2. SFRを指定して表示

```
brk:0>SFR D P5 [↵] ← P 5 レジスタの内容を表示
P5      01
brk:0>■
```



```
brk:0>SFR D TxS [↵] ←書き込み専用SFRを指定
TxS     -- ←書き込み専用なので -- 表示
brk:0>■
```

8.40 実行停止 (STP)

STP

trc:0>	○	emu:0>	○	brk:0>	×
--------	---	--------	---	--------	---

STP T

trc:0>	○	emu:0>	×	brk:0>	×
--------	---	--------	---	--------	---

T : アナライザのみ停止

STP コマンドは、次の2つの機能を持っています。

- ・ エミュレーションCPUとアナライザの両方を停止 . . . STP ☒
- ・ アナライザのみの停止 . . . STP T ☒

8. 4 1 ファイルからのコマンド入力 (STR)

STR file[parameter]

trc:0>	<input type="radio"/>	emu:0>	<input type="radio"/>	brk:0>	<input type="radio"/>
--------	-----------------------	--------	-----------------------	--------	-----------------------

file : ファイル名
parameter : 実パラメータ

STRコマンドは、指定されたファイルからコマンド、データを入力します。
入力ファイルにはCOMコマンド、またはエディタなどで作成したファイルを使用してください。

ファイルからのコマンド、データの入力を一時中断する場合は、[^]Lキーを入力します。
入力を再開する場合は、もう一度[^]Lキーを入力してください。
コマンドまたはデータの入力を終了する場合には[^]Kキーを入力します。

仮パラメータを記述した入力ファイル^注を使用して、ファイル中の仮パラメータを実パラメータに置き換えることができます。実パラメータは、最大4つまで指定することができます。

入力ファイルで仮パラメータを使用する場合、\$0、\$1、\$2、\$3 と記述します。
仮パラメータの\$とアセンブラの相対アドレスを指定する\$を区別するため、相対アドレスを指定する場合は \$\$ と記述してください。

注 仮パラメータを使用するファイルは、エディタなどで作成してください。

例1. ファイルからコマンドを入力

```
brk:0>STR B:SAMPLE.STR (2) ←ドライブBの SAMPLE.STR を入力ファイルに指定
brk:0>CLK I ←このコマンドよりファイルから入力
brk:0>RES
brk:0>OUT OFF
brk:0>■
```

- ・ 例1のファイル内容を以下に示します。

```
CLK I
RES
OUT OFF
```

2. 仮パラメータを記述したファイルからコマンドを入力

```
brk:0>STR SAMPLE.STR U ON (2) ←実パラメータU ONを指定
brk:0>CLK U ←このコマンドよりファイルから入力
brk:0>RES
brk:0>OUT ON
brk:0>■
```

- ・ 例2のファイル内容を以下に示します。

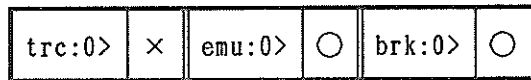
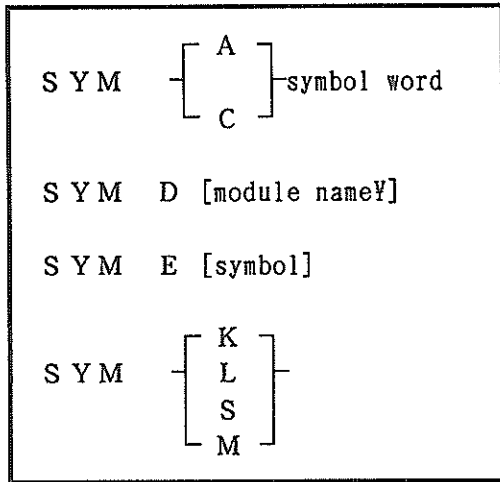
```
CLK $0
RES
OUT $1
```

3. 指定したファイルが存在しない場合

```
brk:0>STR B:SAMPLE.STR (2)
          ファイルが見つかりません。(509)注B:SAMPLE.STR ←ファイルが見つからない
brk:0>■                                     場合のメッセージ
```

注 ()内の数字は、エラー・メッセージの番号を表します。

8.42 シンボルの操作 (SYM)



- SYM A : IESYMBOL の登録
- SYM C : IESYMBOL の変更
- SYM D : IESYMBOL、プリロード・シンボルの表示
- SYM E : IESYMBOL の削除
- SYM K : IESYMBOL、プリロード・シンボルの削除
- SYM L : IESYMBOL のロード
- SYM S : IESYMBOL のセーブ
- SYM M : カレント・モジュールの指定
- symbol : シンボル名
- word : シンボル値
- module name¥ : モジュール名

(1) SYM A コマンド

SYM A symbol word	
基数	word:H

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

symbol : シンボル名
word : シンボル値

SYM A コマンドは、IEシンボルを登録します。

すでに登録されているIEシンボルと同一名称のシンボル、または予約語と同一名称のシンボルは登録できません。

SYM A コマンドで登録されたシンボルに対しては、IESYMBOL というモジュール名が付けられます。

シンボル・タイプは、コード・タイプとなります。

例 IEシンボルの登録

```
brk:0>SYM A SYMBOL01 1000 (C) ←シンボル値が 1000H
brk:0>■ の SYMBOL01 を登録
```

(2) SYM C コマンド



SYM C symbol word	
基数	word:H

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

symbol : シンボル名
word : シンボル値

SYM C コマンドは、指定された I E シンボルのシンボル値を変更します。

例 I E シンボルのシンボル値を変更

brk:0>SYM C SYMBOL01 2000  ←SYMBOL01 という I E シンボルのシンボル値
brk:0> を 2000H に変更

(3) SYM D コマンド

SYM D [module name¥]

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

module name¥^{注1}: モジュール名

SYM D コマンドは、登録されている I E シンボルまたはプリロード・シンボルを表示します。

モジュール名が指定された場合、該当するモジュールのシンボルすべてを表示します。オペランドで PUBLICY を指定した場合、I E シンボルまたは PUBRIC モジュールすべてを表示します。^{注1}

SYM D @ と入力した場合、登録されているすべての I E シンボルまたはプリロード・シンボルをモジュールごとに表示します。

登録されているシンボルがない場合、次のメッセージを表示します。

シンボルがありません。(404)^{注2}

指定したモジュール名が登録されていない場合、次のメッセージを表示します。

モジュールが見つかりません。(519)^{注2}

注1. ¥は、IBM PC シリーズをホスト・マシンとして使用した場合には、\ となります。

2. () 内の数字は、エラー・メッセージの番号を表します。

例1. 登録されている I E シンボルまたはプリロード・シンボルすべてを表示

```
brk:0>SYM D ☐
module : IESYMBOL
    1000 INT1          2000 INT2
module : PUBLIC
    0100 START        0600 DATA          01FF FINISH
module : SUBPRG1
    0250 START        0280 FINISH
brk:0>■
```

2. パブリック・シンボルのみを表示

```
brk:0>SYM D PUBLIC ☐
module : IESYMBOL
    1000 INT1          2000 INT2
module : PUBLIC
    0100 START        0600 DATA          01FF FINISH
brk:0>■
```

3. モジュール名を指定してシンボルを表示

```
brk:0>SYM D SUBPRG1 ☐
module : SUBPRG1
    0250 START        0280 FINISH
brk:0>■
```

(4) SYM E コマンド

S Y M E symbol

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

symbol : シンボル名

S Y M E コマンドは、指定した I E シンボルを削除します。

シンボルの指定がない場合、S Y M A コマンドで定義されたシンボルをすべて削除します。

例1. 指定した I E シンボルの削除

```
brk:0>SYM E INT1  ←INT1 を削除
brk:0>■
```

2. すべての I E シンボルの削除

```
brk:0>SYM E 
brk:0>■
```

(5) SYM K コマンド

S Y M K

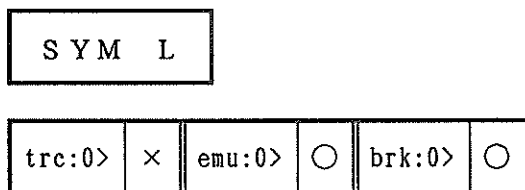
trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

SYM K コマンドは、登録されている I E シンボルまたはプリロード・シンボルすべてを削除します。

例 すべてのシンボルの削除

```
brk:0>SYM K [C]
brk:0>■
```

(6) SYM L コマンド



SYM L コマンドは、IEシンボル・ファイルから IESYMBOL をロードします。ロードされる IEシンボル・ファイルは、カレント・ドライブ上の IE78327.SYM というファイルになります。

例1. IEシンボル・ファイルのロード

```
brk:0>SYM L (4)
brk:0>■
```

2. IEシンボル・ファイルが存在しない場合

```
brk:0>SYM L (4)
IE78327.SYM が見つかりません。(400)注 ← μPD78327シリーズを開発
brk:0>■                               する場合でIEシンボルのファイル
                                         がない場合のメッセージです
```

注 () 内の数字はエラー・メッセージの番号を表します。

(7) SYM S コマンド

S Y M S

trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

S Y M S コマンドは、IESYMBOL を I E シンボル・ファイルへ格納します。

I E シンボル・ファイルのファイル名は IE78327.SYM というファイル名が自動的に設定されます。

ドライブ・ユニットは、カレント・ドライブのカレント・ディレクトリが選択されます。

例1. IESYMBOL を I E シンボル・ファイルへ格納

```
brk:0>SYM S (Y)
brk:0>■
```

2. IESYMBOL が存在しない場合

```
brk:0>SYM S (Y)
シンボルがありません。(404)注← IESYMBOL が存在しない場合のメッセージ
brk:0>■
```

3. I E シンボル・ファイルがすでに存在する場合

```
brk:0>SYM S (Y)
すでに存在するファイルです。(513)注 IE78327.SYM Delete ? (Y/N) : Y (Y)
brk:0>■
```

IE78327.SYM というリード、ライト属性のファイルがすでに存在する場合、上記のメッセージに対して Y (Y) を入力すると、すでに存在している同名のファイルを削除し、新しく IE78327.SYM ファイルを作成します。

Y (Y) 以外が入力された場合、コマンドは無視されます。

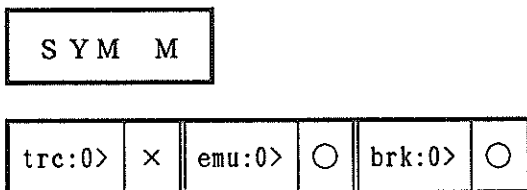
4. I E シンボルのファイルが作成できない場合

```
brk:0>SYM S (Y)
読み取り専用のファイルです。(507)注 IE78327.SYM
brk:0>■
```

ファイルの属性が R/O の場合、上記のメッセージを表示し、コマンドは無視します

注 () の数字は、エラー・メッセージの番号を表します。

(8) SYM M コマンド



SYM M コマンドは、設定されているカレント・モジュール名の表示または変更をします。

カレント・モジュールが設定されている場合、カレント・モジュール内のシンボルは、モジュール名を省略して記述できます。

同一シンボル名がパブリック・シンボル・モジュールに存在する場合、そのシンボルはパブリック・シンボルと判断されます（図8-6 カレント・モジュールが設定されている場合のシンボル処理フロー 参照）。

他のモジュールに属するローカル・シンボルを記述する場合、ローカル・シンボル名の前にモジュール名を付加します。

例1. カレント・モジュール名の表示

```
brk:0>SYM M (Q)
MOD01 Y = ■ ←現在のカレント・モジュール名を表示
```

```
brk:0>SYM M (Q)
MOD01 Y = (Q) ←カレント・モジュール名を変更しないときは
brk:0>■ (Q) を入力
```

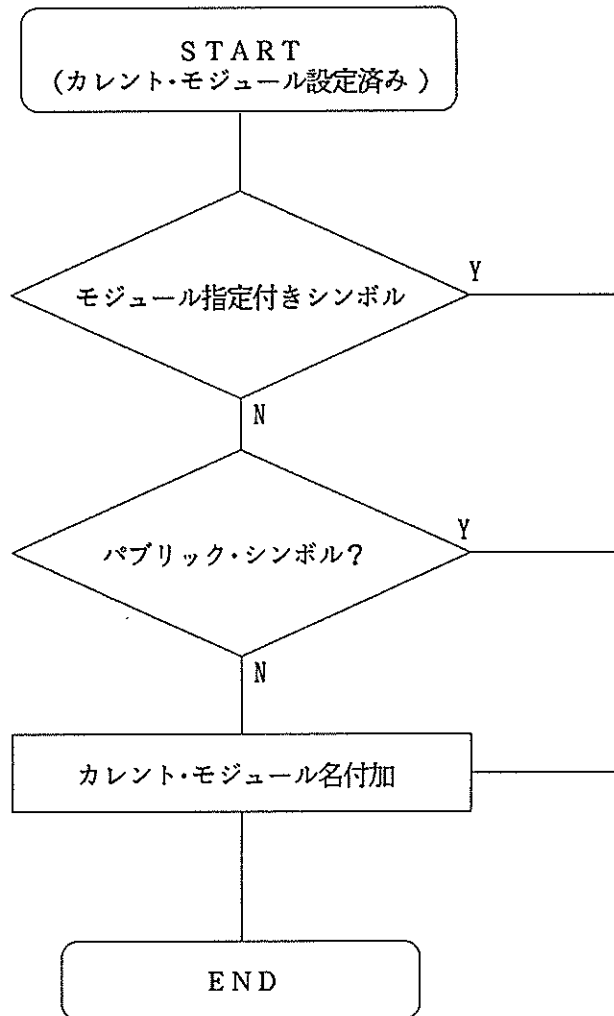
2. カレント・モジュール名を変更

```
brk:0>SYM M (Q)
MOD01 Y = MOD02Y (Q) ←カレント・モジュールを MOD02 に変更
brk:0>■
```

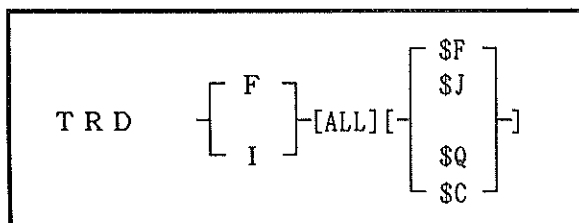
```
brk:0>SYM M (Q)
MOD02 Y = (Q) ←カレント・モジュールが MOD02 に変更
brk:0>■
```

備考 卒は I B M P C シリーズをホスト・マシンを使用した場合には、 \ と な り ま す。

図8-6 カレント・モジュールが設定されている場合のシンボル処理フロー



8.43 トレース・データの表示 (TRD)



trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

- TRD F : フレーム表示
- TRD I : インストラクション表示
- ALL : カレント・トレース・ブロック内の最初のフレームから、トレース・データすべて、または、検索条件に一致するデータすべての表示を指定
- \$F : 検索条件に一致するフレームの前後約5行の表示を指定
- \$J : プログラムの分岐処理に関するフレームの表示を指定
- \$Q : 検索条件に一致するフレームの表示を指定
- \$C : チェック・ポイントのフレームの表示を指定

TRD コマンドは、カレント・トレース・ブロックのトレース・データの表示、または、カレント・トレース・ブロックの検索表示を行います。

サブコマンド、オペランドの組み合わせにより8種類の表示形式でトレース・データを表示できます。

トレース・データの表示コマンドに関して、次の4機能を説明します。

- トレース・ブロック
- トレース・ポインタの操作
- トレース・データの検索表示
- トレース・データ表示時のマーキング情報

(a) トレース・ブロック

トレース・メモリへの書き込みは、常に前回のトレース・データに追加されます。ただし、RES H コマンド時のみトレース・メモリはクリアされます。

したがって、エミュレーション・スタート・コマンド (RUN B, RUN N, RUN T, TRG) の入力から、次のエミュレーション・スタート・コマンドの入力までに、エミュレーションCPUがターゲット・プログラムを実行して得たトレース結果をトレース・ブロックと言います。

トレース・データの表示コマンド (TRD) は、カレント・トレース・ブロックのトレース・データの表示、またはカレント・トレース・ブロックの検索表示を行います。

(b) トレース・ポイントの操作

トレース・データの表示と検索表示は、現在のトレース・ポイント、または、指定するトレース・ポイントを基準として表示を始めます。

エミュレーション終了時には、トレース・ポイントはカレント・トレース・ブロックの最後のフレームに位置しています。

トレース表示エリアの変更のためのトレース・ポイントの指定は、トレース・データ表示後の促進入力時のポイント指定、または、TRDコマンド入力時のALLオペランド入力により行います。

次に、促進入力時のトレース・ポイント指定機能一覧を示します。

表8-2 促進入力時のトレース・ポイント指定機能一覧

トレース・ポイントの指定内容	促進入力コマンド	機能	カレント・トレース・ブロック内外
絶対アドレス	frame No	指定したフレームより11行表示	内/外
カレント・トレース・ブロック変更	前	P 前のトレース・ブロックの最後の11行を表示	外
	後	N 後のトレース・ブロックの最初の11行を表示	
+/cr/-の移動	+ / c r	直前に表示した後の11行表示	内
	-	直前に表示した最初のフレームから前の11行表示	
カレント・トレース・ブロック内の操作	F	最初の11行を表示	内
	L	最後の11行を表示	
トリガ検索	T	トリガ・フレームの前後11行を表示	

表示対象のトレース・ブロックの変更はカレント・トレース・ブロック変更コマンド(P, N)または、絶対アドレスによるトレース・ポイントの指定により行います。

(c) トレース・データの検索表示

カレント・トレース・ブロックから、希望するトレース・フレームを表示する機能をトレース・データの検索表示と言います。

検索表示条件の指定は、TRDコマンド入力時の"\$"オプションの指定、または、トレース表示後の促進入力時のトリガ検索指定により行います。

次に、TRDコマンド入力時の検索表示条件一覧を示します。

表8-3 TRDコマンド入力時の検索表示条件一覧

オプション	検索表示内容
\$F	検索条件に一致するフレームの前後5行を表示します
\$J	プログラムの分岐命令に関するフレームを表示します
\$Q	検索条件に一致するフレームを表示します
\$C	チェック・ポイントのフレームを表示します

\$Fまたは、\$Qの検索条件は、トレース・データ検索条件の設定コマンド(TRF)であらかじめ設定しておきます。

(d) トレース・データ・マーキング情報

トレース・データの表示には、トレース・メモリにトレース・フレームが書き込まれたときのエミュレーションCPUの動作状態（実行モード）、トレース・モード、そして特殊フレームを示すために、各種のマーキング情報が自動的に付加されます。

次に、エミュレーションCPUの実行モードとトレース・モード用のマーキング情報一覧を示します。


表8-4 実行モード、トレース・モード用のマーキング情報一覧

実行モード	トレース・モード	マーキング情報
リアルタイム・エミュレーション	ALL	[ALL trace mode terminated.]
	SEC	[SEC trace mode terminated.]
	TRX	[TRX trace mode terminated.]
シングル・ステップ実行	(実行ステップ全トレース)	[one step emulation terminated.]
プロシージャ実行	(メイン・ルーチンのみ全トレース)	[procedure emulation terminated.]

シングル・ステップ実行用マーキング情報と、プロシージャ実行用マーキング情報は、実行モードが変更になったときのみ表示されます。

次に、特殊フレーム用のマーキング情報一覧を示します。

表8-5 特殊フレーム用のマーキング情報一覧

エミュレーションCPU/トレース状態	マーキング情報
割り込みベクタ発生時	<INTxx>
トリガ・フレーム	T
チェック・ルーチン	<CHK> 

備考 区間トレース・モードまたは、クオリファイ・トレース・モード時のトリガ・ポイントは、トリガ表示フレームとトリガ表示フレームの1つ前のフレームの間に発生しています。

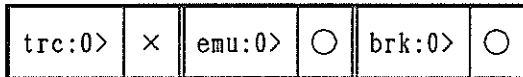
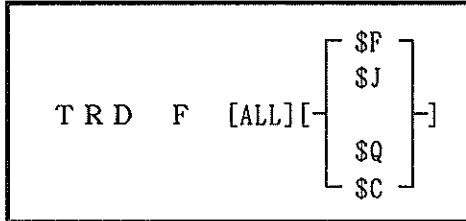
リアルタイム・トレーサのトレース・データとトレース・ブロック及びマーキング情報例を示します。

表8-6 リアルタイム・トレーサの動作例

実行コマンド	トレース・モード (TRM)	タイム・タ グ 選択	リアルタイム・トレーサ のトレース・データ	トレース・ ブロック	マーキング情報
RUN_B ----- []	オール・トレース ↑ ↑ ↑ ↑ ↑ ↑	EXT ↑ ↑ ↑ ↑ ↑ ↑	----- T ----- ----- D -----	①	- [ALL trace mode terminated.] - [one step emulation terminated.]
RUN_B ----- []	クォリファイ・トレース ↑ ↑ ↑	EXT ↑ ↑ ↑	----- T -----	②	- [TRX trace mode terminated.] - [one step emulation terminated.]
RUN_N	セクション・トレース ↑ ↑ ↑	TIME ↑ ↑ ↑	----- T ----- ----- D -----	③	- [SEC trace mode terminated.]
TRG	セクション・トレース ↑ ↑ ↑ ↑	TIME ↑ ↑ ↑ ↑	----- T ----- ----- D -----	④	- [SEC trace mode terminated.]
RUN_T_PRC ----- [] [] []	プロシージャ・トレース ↑ ↑ ↑ ↑	TIME ↑ ↑ ↑ ↑	----- -----	⑤	- [procedure emulation terminated.] - [one step emulation terminated.]

- T...パス条件成立を示します。
- D...ディレイ条件成立を示します。

(1) TRD F コマンド



- ALL : カレント・トレース・ブロック内の最初のフレームから、トレース・データすべて、または、検索条件に一致するデータすべての表示を指定
- \$F : 検索条件に一致するフレームの前後約5行の表示を指定
- \$J : プログラムの分岐処理に関するフレームの表示を指定
- \$Q : 検索条件に一致するフレームの表示を指定
- \$C : チェック・ポイントのフレームの表示を指定

TRD F コマンドは、カレント・トレース・ブロックのトレース・フレームをフレーム・モードで表示、または、カレント・トレース・ブロックのトレース・データを検索し、フレーム・モードで表示します。

トレース・フレームの表示内容を次に示します。

表8-7 トレース・フレームの表示内容 (TRDF)

ラベル名	意味
Frame	トレース・メモリのフレームNoでトレース・メモリに書き込まれた時間的順序を表しています。表示できる範囲は 0000-8191 です。
Status	<p>トレース・フレームのバス・サイクルのステータスを示します。</p> <ul style="list-style-type: none"> ・OP : オペコード・フェッチです。 ・RD : リード・アクセスです。 ・WR : ライト・アクセスです。 ・INTRD : 割り込み処理によるリード・サイクルです。 ・INTWR : 割り込み処理によるライト・サイクルです。 ・MSRD : マクロ・サービスによるリード・サイクルです。 ・MSWR : マクロ・サービスによるライト・サイクルです。 ・BROP : 分岐後の最初のフェッチです。 <p>注・WST : BCUバスのライト起動サイクル (内部ライト・サイクル)</p>
Address	フェッチまたはリード, ライト・アクセスしたアドレスです。
Data	フェッチまたはリード, ライト・アクセスしたデータです。ただし、無効フェッチの場合にはDataが () で囲まれます。
Exu	<p>エミュレーションCPUの Exu の動作情報です。</p> <ul style="list-style-type: none"> ・M1 : 命令の実行開始を示します。
8--EX--1	<p>外部センス・データを示します。</p> <p>本フレーム・データは、トレース・データの選択コマンドで TRS E が選択されたときに表示されます。</p>
Clock	<p>タイム・タグ (フレーム間のシステム・クロック数) を示します。</p> <p>本フレーム・データは、トレース・データの選択コマンドで TRS T が選択されたときに表示されます。</p>

注意 内部RAM (0FE00H-0FEFFH) をショート・ダイレクト・アドレッシング以外のアドレッシングとした場合、トレース・データの値は不定となります。ただし、命令 (ターゲット・プログラム) の実行は正常に行われます。

注 WST サイクルは、デバイス内部のライト・サイクルです。実際にメモリに書き込まれるサイクルは、それ以後のWRサイクルです。

オペランドで \$F、\$J、\$Q、\$C を指定して、トレース・データの検索表示を行います。

・ TRD F [$\left[\begin{array}{c} \$F \\ \$J \\ \$Q \\ \$C \end{array} \right]$] ⊕ と入力した場合

現在のトレース・ポイントから、カレント・トレース・ブロック内の正の方向に向かって検索条件に一致するデータを検索します。一致データがあった場合は検出データを表示し、表示促進入力待ちとなります。

・ TRD F ALL [$\left[\begin{array}{c} \$F \\ \$J \\ \$Q \\ \$C \end{array} \right]$] ⊕ と入力した場合

カレント・トレース・ブロック内の最初のフレームから、カレント・トレース・ブロック内の正の方向に向かって検索条件に一致するデータを検索します。一致データがあった場合は検出データを表示し、表示促進入力待ちとなります。

注意1. 先頭フレームのタイム・タグは、不定値となるため“xxx”と表示されます。

2. タイム・タグ値が、255システム・クロック以上の場合は、すべて“xxx”と表示されます。

例1. トレース・データすべてをフレーム表示

```

emu:0>TRD F ALL [Enter]
  Frame Status Address Data Exu 8--EX--1
  0000   OP    0100    65      00000000
  0001   OP    0101    00     M1 00000000
      .
      .
  2047   OP   010A    C8      00000000
emu:0>

```

2. 検索条件に一致するフレームの前後5行を表示

```

emu:0>TRD F $F [Enter]
  Frame Status Address Data Exu 8--EX--1
  0029   OP    0100    65      00000000
  0030   OP    0101    00     M1 00000000
      .
      .
!0033   OP    0134    C8      00000000
      .
      .
  0039   OP    0158    42      00000000
Total frame = 2048 (F/L/T/P/N/+/cr/-/Frame No./.) ?

```

3. 検索条件に一致するフレームすべてを表示

```

emu:0>TRD F $Q [Enter]
  Frame Status Address Data Exu 8--EX--1
!0012   WR    8001    32      00000000
!0065   WR    8001    32      00000000
      .
      .
!0143   WR    8001    32      00000000
emu:0>

```

例4. プログラム分岐処理に関するフレームを表示

```

emu:0>TRD F $J ␣
  Frame Status Address Data Exu 8--EX--1
  0000  BROP  0100   65      00000000
      .
      .
      .
< INTCM10 >
  0063  INTRD  001E   00      00000000
  0064  INTRD  001F   40      00000000
  0065  INTWR  FDFE   08      00000000
  0066  INTWR  FDFE   07      00000000
  Total frame = 1074 (F/L/T/P/N+/cr-/Frame No./.) ? <ESC>
emu:0>■

```

5. 前のトレース・ブロックの最後の11行を表示

```

emu:0>TRD F ␣
  Frame Status Address Data Exu 8--EX--1
  0027  OP    0100   65      00000000
  0030  OP    0101   00      M1 00000000
      .
      .
      .
  0033  OP    0134   C8      00000000
      .
      .
      .
  0037  OP    0158   42      00000000
  Total frame = 2048 (F/L/T/P/N+/cr-/Frame No./.) ? P ␣
  Frame Status Address Data Exu 8--EX--1
  0018  OP    0100   65      00000000
  0020  OP    0101   00      M1 00000000
      .
      .
      .
  0024  OP    0120   A0      00000000
      .
      .
      .
  0028  OP    0111   30      00000000
  Total frame = 2048 (F/L/T/P/N+/cr-/Frame No./.) ? <ESC>
emu:0>■

```

例6. うしろのトレース・ブロックの最初の11行を表示

```

emu:0>TRD F Q
  Frame  Status  Address  Data  Exu  8--EX--1
  0015    OP      0100    65           00000000
  0016    OP      0101    00    M1  00000000
      .
      .
  0020    OP      0120    A0           00000000
      .
      .
  0025    OP      0111    30           00000000
  Total frame = 2048 (F/L/T/P/N+/cr/-/Frame No./.) ? N Q
  Frame  Status  Address  Data  Exu  8--EX--1
  0026    OP      0100    65           00000000
  0030    OP      0101    00    M1  00000000
      .
      .
  0033    OP      0134    C8           00000000
      .
      .
  0036    OP      0158    42           00000000
  Total frame = 2048 (F/L/T/P/N+/cr/-/Frame No./.) ? <ESC>
emu:0>■

```

例7. フレームNoを指定して1024フレーム目から11行を表示

```

emu:0>TRD F Q
Frame Status Address Data Exu 8--EX--1
0029 OP 0100 65 00000000
0030 OP 0101 00 M1 00000000
.
.
0033 OP 0134 C8 00000000
.
.
0039 OP 0158 42 00000000
Total frame = 2048 (F/L/T/P/N+/cr-/Frame No./.) ? 1024Q
Frame Status Address Data Exu 8--EX--1
1024 RD FE10 65 00000000
1025 OP 0101 00 M1 00000000
.
.
1030 OP 0134 C8 00000000
.
.
1034 WR FE12 42 00000000
Total frame = 2048 (F/L/T/P/N+/cr-/Frame No./.) ? <ESC>
emu:0>■
    
```

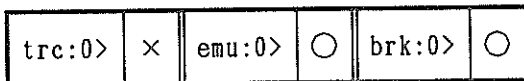
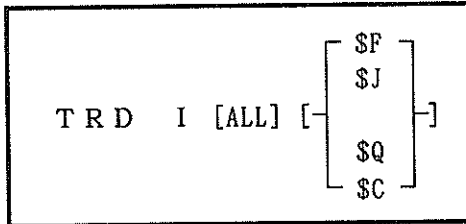

例8. トレース・データのマーキング情報表示

```

emu:0>TRD F ALL (C)
  Frame  Status  Address  Data  Exu  8--EX--1
  0000                00    M1  00000000
  0001    RD     0100     00          00000000
  0002    OP     0101     12          00000000
  0003    OP     0010     35          00000000
  0004                45    M1  00000000
  0005    OP     0124     0B          00000000
  0006    OP     0125     0C          00000000
<CHK>
FE00    01
  Frame  Status  Address  Data  Exu  8--EX--1
  0009                00    M1  00000000
  0010    OP     0125     00          00000000
  0011    WR     FF34     15          00000000
T0012  BROP     0230     45          00000000
  0013                BB    M1  00000000
  0014    OP     4508     30          00000000
  0015    OP     4411     02          00000000
[ALL trace mode terminated.]
  0016                00    M1  00000000
  0017    OP     0012     55          00000000
  0018    OP     0013     12          00000000
[one step emulation terminated.]
  0019                00    M1  00000000
  0020    OP     0012     55          00000000
  0021    OP     0013     12          00000000
[procedure emulation terminated.]
emu:0>■

```

(2) TRD I コマンド



- ALL : カレント・トレース・データすべて、または、
検索条件に一致するデータすべての表示を指定
- \$F : 検索条件に一致するフレームの前後約5行の表示を指定
- \$J : プログラムの分岐処理に関するフレームの表示を指定
- \$Q : 検索条件に一致するフレームの表示を指定
- \$C : チェック・ポイントのフレームの表示を指定

TRD I コマンドは、カレント・トレース・ブロックのトレース・フレームを
インストラクション・モードで表示します。
または、カレント・トレース・ブロックのトレース・データを検索し、インスト
ラクション・モードで表示します。

トレース・フレームの表示内容を次に示します。

表8-8 トレース・フレームの表示内容 (TRD I)

ラベル名	意 味
Frame	トレース・メモリのフレームNoでトレース・メモリに書き込まれた時間的順序を表しています。表示できる範囲は 0000-8191 です。
Status	<p>トレース・フレームのバス・サイクルのステータスを示します。</p> <ul style="list-style-type: none"> ・OP : オペコード・フェッチです。 ・RD : リード・アクセスです。 ・WR : ライト・アクセスです。 ・INTRD : 割り込み処理によるリード・サイクルです。 ・INTWR : 割り込み処理によるライト・サイクルです。 ・MSRD : マクロ・サービスによるリード・サイクルです。 ・MSWR : マクロ・サービスによるライト・サイクルです。 ・BROP : 分岐後の最初のフェッチです。
Address	フェッチまたはリード, ライト・アクセスしたアドレスです。
Data	フェッチまたはリード, ライト・アクセスしたデータです。ただし、無効フェッチの場合にはDataが () で囲まれます。
Label	レーベル・シンボルを表示します。
Mnemonic	命令のニモニックを表示します。
EX	<p>外部センス・データを示します。</p> <p>本フレーム・データは、トレース・データの選択コマンドで TRS E が選択されたときに表示されます。</p>
Clock	<p>タイム・タグ (フレーム間のシステム・クロック数) を示します。</p> <p>本フレーム・データは、トレース・データの選択コマンドで TRS T が選択されたときに表示されます。</p>

注意 内部RAM (0FE00H-0FEFFH) をショート・ダイレクト・アドレッシング以外のアドレッシングとした場合、トレース・データの値は不定となります。ただし、命令 (ターゲット・プログラム) の実行は正常に行われます。

オペランドで \$F、\$J、\$Q、\$C を指定して、トレース・データの検索表示を行います。

・ TRD I [$\left[\begin{array}{c} \$F \\ \$J \\ \\ \$Q \\ \$C \end{array} \right]] \textcircled{\text{C}}$ と入力した場合

現在のトレース・ポイントから、カレント・トレース・ブロック内の正の方向に向かって検索条件に一致するデータを検索します。一致データがあった場合は検出データを表示し、表示促進入力待ちとなります。

・ TRD I ALL [$\left[\begin{array}{c} \$F \\ \$J \\ \\ \$Q \\ \$C \end{array} \right]] \textcircled{\text{C}}$ と入力した場合

カレント・トレース・ブロック内の最初のフレームから、カレント・トレース・ブロック内の正の方向に向かって検索条件に一致するデータを検索します。一致データがあった場合は検出データを表示し、表示促進入力待ちとなります。

- 注意1. 先頭フレームのタイム・タグは、不定値となるため"xxx"と表示されます。
2. タイム・タグ値が、255システム・クロック以上の場合、すべて"xxx"と表示されます。
 3. 命令に対するリード、またはライト・アクセスを表示するので、実際のエミュレーションCPUの動作とトレース・データの表示の順番が異なる場合があります。
 4. クオリファイ・トレース・モード (TRM TRX) でトレースされたデータは、インストラクション・トレース表示 (TRD I) を行っても、フレーム表示となります。
 5. 区間トレース・モード (TRM SEC) でトレースされたデータの先頭フレームと最終フレームは、インストラクション・トレース表示 (TRD I) を行うと、不定値となる場合があります。

例1. トレース・データすべてをインストラクション表示

```

emu:0>TRD I ALL [Enter]
  Frame Status Address Data          Label Mnemonic
  PUBLICYEXTACSS::
0000          0100  80                MOVW   RPO, #1234H
0003          0103  80                MOVW   !4000H, RPO
          .
          .
1019          0109  81                BR     $109H
emu:0>
  
```

2. 検索条件に一致するフレームの前後5行をインストラクション表示

```

emu:0>TRD I $F [Enter]
  Frame Status Address Data          Label Mnemonic
  PUBLICYEXTACSS::
0000          0100  80                MOVW   RPO, #1234H
0003          0103  80                MOVW   !4000H, RPO
          .
          .
!0007注      0107  81                NOP
          .
          .
0014  WR     4001  12 81
Total frame = 1020 (F/L/T/P/N/+/cr-/Frame No./.) ?
  
```

注 !マークは、検索条件に一致したトレース・データを表しています。

例3. 検索条件に一致するフレームをインストラクション表示

```

emu:0>TRD I $Q [Q]
  Frame Status Address Data          Label Mnemonic
  PUBLICYEXTACSS::
!0007          0107  81                NOP
!0028          0107  81                NOP
      .
      .
!1003          0107  81                NOP
  Total frame = 1020 (F/L/T/P/N+/cr/-/Frame No./.) ? █
  
```

4. プログラム分岐処理に関するフレームをインストラクション表示

```

emu:0>TRD I $J [Q]
  Frame Status Address Data          Label Mnemonic
  0053          012F                BR      $12FH
  0057          012F                BR      $12FH

  0063  INTRD  001E  00
T0063  INTRD  001E  00 ← イベントを検出したフレーム
  0064  INTRD  001F  10
  0065  INTWR  FDFE  08
  0066  INTWR  FDFE  00
  0068  INTWR  FDFC  2F
  Total frame = 1074 (F/L/T/P/N+/cr/-/Frame No./.) ? <ESC>
emu:0>█
  
```

例5. 区間トレースに関するフレームをインストラクション表示

```
brk:0>TRD I ALL (Q)
Frame Status Address Data Label Mnemonic EX
8180 0180 2C2003 BR !320H
T8185 BROP 0320 14 2F
8185 0320 14FE BR $322H
8188 0322 00 NOP
8191 OP 0321 FE 3F
[SEC trace mode terminated.]
brk:0>■
```

6. クオリファイ・トレースしたフレームを表示

```
brk:0>TRD I ALL (Q)
Frame Status Address Data Exu 7--EX--0
8178 OP 0164 1B 00000011
8179 OP 0165 (00) 00000011
<INTCM10>
8180 BROP 0180 m2C 00000011
8181 20 M1 00000011
8182 OP 0181 20 00000011
8183 OP 0182 03 00000011
8184 OP 0183 (00) 00000011
<CHK>
P5 10
Frame Status Address Data Exu 7--EX--0
T8185 BROP 0320 m14 00101111
8186 FE M1 00111111
8187 OP 0321 FE 00111111
8188 OP 0322 (00) 00111111
[TRX trace mode terminated.]
brk:0>
```

8.44 トレース・データ検索条件の設定 (TRF)

TRF [A= <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 0 5px;">word</td> <td rowspan="2" style="font-size: 2em; vertical-align: middle;">]</td> </tr> <tr> <td style="padding: 0 5px;">partition</td> </tr> </table>] [V=mask8] [C=status] [E=mask8]		word]	partition
word]			
partition				
基数	word:H partition:H mask8:H			


trc:0>	×	emu:0>	○	brk:0>	○
--------	---	--------	---	--------	---

word : 検索アドレス
partition : 検索アドレス範囲
status : 検索ステータス
mask8 : 8ビット・マスク・データ

TRF コマンドはトレース・データの検索条件を設定します。

検索条件の設定には、コマンドを1ラインで入力する方法と、対話形式で入力する方法があります。

対話形式で検索条件を設定する場合、現在の検索条件の表示後、各条件ごとに順次設定します。

TRF  と入力した場合、現在設定されている検索条件を表示し、対話形式の設定モードとなります。

検索アドレスは、5つまで（スペースで区切って）設定できます。設定を省略した場合、すべてのアドレス領域が設定されます。

検索データは、8ビット・マスク・データで設定します。設定を省略した場合、すべてのデータが設定されます。

検索ステータスは、次のバス・サイクルの属性から選択します。設定を省略した場合、オペコード・フェッチを含むすべてのリードまたはライトが設定されます。

- ・BROP : 分岐後の最初のフェッチ
- ・OP : オペコード・フェッチ
- ・RWI : 割り込みによるリード・ライト
- ・RI : 割り込みによるリード
- ・WI : 割り込みによるライト
- ・RW : データ・リード・ライト
- ・R : データ・リード
- ・W : データ・ライト
- ・RWP : プログラムによるデータ・リードまたはライト
- ・RP : プログラムによるデータ・リード
- ・WP : プログラムによるデータ・ライト
- ・RWM : マクロ・サービスによるデータ・リードまたはライト
- ・RM : マクロサービスによるデータ・リード
- ・WM : マクロ・サービスによるデータ・ライト
- ・NC : オペコード・フェッチを含むすべてのリードまたはライト

検索する外部データは、外部センス・クリップ（No. 1 - 8）から入力する外部データに対して、8ビット・マスク・データで設定します。

なお、設定を省略した場合、すべての外部データが設定されます。

例1. 検索条件を1ラインで設定

```
brk:0>TRF A=300X 0FE00,0FE7F V=55H C=WP E=0XXH [注]
brk:0>
```

注 3000H-300FH、0FE00H-0FE7FH 番地に 55H をプログラム・ライトしたトレース・データを検索します。

2. 検索条件を対話形式で設定

```
brk:0>TRF [ ]
A=3000H,300FH,FE00H,FE7FH
V=55H
C=WP
E=0XXH
```

←現在設定されている検索条件の表示


```
A=4000 [ ]
V=0XXH [ ]
BROP (BRanch OPecode fetch)
OP (OPecode fetch)
RWI (Read Write by Interrupt)
RI (Read by Interrupt)
WI (Write by Interrupt)
NC (No Condition)
RW (Read Write)
R (Read)
W (Write)
RWP (Read Write by Program)
RP (Read by program)
WP (Write by Program)
RWM (Read Write by Macro service)
RM (Read by Macro service)
WM (Write by Macro sevice)
```

← 4000H 番地に変更
← 0XXH に変更


```
C=BROP [ ]
E=1H [ ]
brk:0>■
```

← BROP に変更
← 1H に変更

8.45 アナライザの再起動 (TRG)

TRG

trc:0>	×	emu:0>	○	brk:0>	×
--------	---	--------	---	--------	---

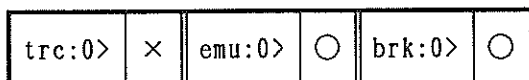
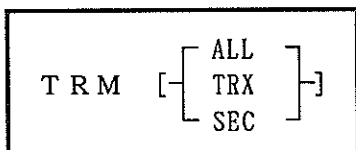
TRGコマンドは、アナライザ（リアルタイム・トレーサと内部データ・サンプラ）を再起動します。

アナライザの再起動は、エミュレーションCPU動作中で、アナライザのみ停止している状態（プロンプト：emu:0>）のときに行います。

例 アナライザの再起動

```
emu:0>TRG 
Analyzer start
trc:0>■
```

8.46 トレース・モードの設定 (TRM)



ALL : 全トレース
 TRX : クオリファイ・トレース
 SEC : 区間トレース

TRM コマンドは、トレース・モードを選択します。

TRM ALL と入力した場合、RUN B、またはNコマンド入力後からトレースします。

TRM TRX と入力した場合、TRX コマンドで指定されたイベント発生時のみトレースします。

TRM SEC と入力した場合、ENBからDSB間の動作のみトレースします。

TRM と入力した場合、現在のトレース・モードを表示します。

例1. トレース・モードをクオリファイ・トレースに設定

```
brk:0>TRM TRX [Q]  
brk:0>■
```

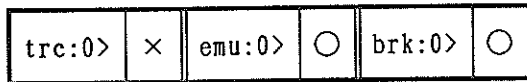
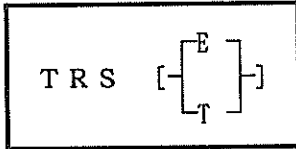
2. 現在選択されているトレース・ソースの表示

```
brk:0>TRM [Q]  
TRX ←クオリファイ・トレースが選択されています  
brk:0>■
```

3. トレース・モードを区間トレースに設定

```
brk:0>TRM SEC [Q]  
brk:0>■
```

8. 47 トレース・データの選択 (TRS)



E : 外部データ
 T : タイム・タグ

TRS コマンドは、トレース・データの外部データとタイム・タグのどちらかのトレースを選択します。

TRS E と入力した場合、外部データを選択します。

TRS T と入力した場合、タイム・タグを選択します。

TRS と入力した場合、現在の選択状態を表示します。

例1. トレース・データを外部データに選択

```
brk:0>TRS E [E]  
brk:0>
```

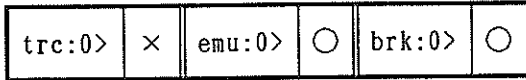
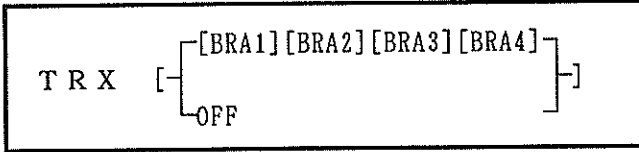
2. トレース・データをタイム・タグに設定

```
brk:0>TRS T [T]  
brk:0>■
```

3. 現在の設定を表示

```
brk:0>TRS [E] ←現在の設定を表示  
Time_tag  
brk:0>■ ←タイム・タグが設定されています
```

8.48 トレース・クオリファイの設定 (TRX)



BR? 注: クオリファイ条件
 OFF : クオリファイの解除

注 BR?は、BRA1、BRA2、BRA3、BRA4 のクオリファイ条件を表しています。

TRXコマンドは、トレース・クオリファイの条件としてイベント要因を指定します。クオリファイ条件は、4つまで指定することができます。

TRX OFF と入力した場合、トレース・クオリファイの指定を解除します。

TRX と入力した場合、クオリファイ条件を表示します。

TRXコマンドは、TRMコマンドでTRXが選択されていなければ機能しません。

例1. クオリファイ条件に BRA1、BRA2 を設定

```
brk:0>TRX BRA1 BRA2 
brk:0>■
```

2. 設定されているクオリファイ条件を表示

```
brk:0>TRX 
BRA1 BRA2
brk:0>■
```


8. 49 オブジェクト・ファイルとメモリ内容の比較 (VRY)

VRY	file
-----	------

trc:0>	×	emu:0>	×	brk:0>	○
--------	---	--------	---	--------	---

file : ファイル名

VRYコマンドは、ホスト・マシンのファイルに格納されているオブジェクト・コードとメモリ内容をベリファイします。

メモリ内容とのベリファイはマッピング状態に応じて行れます。

オブジェクト・ファイルとメモリ内容が異なる場合、アドレス、オブジェクト・データおよびメモリの内容を表示します。

例1. ファイルのデータとメモリ内容をベリファイ

```
brk:0>VRY SAMPLE.HEX (Q) ←SAMPLE.HEX というファイルとメモリ内容をベリファイ
object verify complete ←正常終了した場合のメッセージ
brk:0>■
```

2. ベリファイした結果に異常があった場合

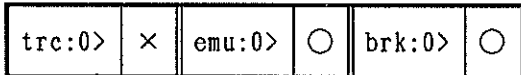
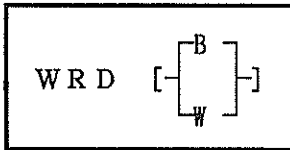
```
brk:0>VRY SAMPLE.HEX (Q)
object verify
Address File Memory ←ベリファイした結果に異常があった場合のメッセージ
0123 00 01
1234 FF FE
brk:0>■
```

3. 指定したファイルが存在しない場合

```
brk:0>VRY SAMPLE.HEX (Q)
ファイルが見つかりません。(509)注SAMPLE.HEX ← SAMPLE.HEX というファイル
名が見つからない場合のメ
ッセージ
brk:0>■
```

注 () 内の数字は、エラー・メッセージの番号を表しています。

8.50 メモリ・ワード長の設定 (WRD)



B : バイト指定
W : ワード指定

WRDコマンドは、メモリ操作における、データの表示、変更時のデータ長（メモリ・ワード長）を設定します。

データ長は、バイト、ワードのいずれかを選択します。

WRD と入力した場合、現在設定されているデータ長を表示します。

例1. メモリ・ワード長をバイトに設定

```
brk:0>WRD B 
brk:0>■
```

2. メモリ・ワード長を確認

```
brk:0>WRD 
Word ←ワードが設定されています
brk:0>■
```

第9章 エラー・メッセージ一覧

この章では、IE-78327-Rから出力されるすべてのエラー・メッセージについて説明します。エラー・メッセージの説明は検索に便利のように番号順に載せてあります。エラー・メッセージの番号とエラーの種類を以下の表に示します。

表9-1 エラー・メッセージの番号と種類

エラー・メッセージの番号	エラーの種類
000-002	システム・エラー
100-112	エミュレータ共通エラー
200-210	アナライザ関連エラー
300-304	オンライン・アセンブラ関連エラー
400-412	シンボル関連エラー
500-518	ファイル関連エラー(ロード/セブ/PGM)

出力されるエラー・メッセージは、次に示すIE-78327-Rのシステム構成の違いに応じて異なります。

ホスト・マシン	メッセージの言語
PC-9800シリーズ	日本語
IBM PC/XT、PC/AT	英語

凡例

- このマニュアルで用いられている画面表示例、入力例はPC-9800シリーズをホスト・マシンとして使用した場合のものです。

9.1 システム・エラー (000-002)

エラー番号：(000)

メッセージ (日本語)：通信異常です。(000)

(英語)：Communication error. (000)

説明：IE-78327-Rとホスト・マシンの通信が正常にできなかった場合に出力します。

エラー番号：(001)

メッセージ (日本語)：ハードウェアの異常です。RES コマンドを入力して下さい。(001)

(英語)：Illegal hardware. Enter RES command. (001)

説明：ハードウェアに異常が発生した場合に出力します。

エラー番号：(002)

メッセージ (日本語)：ハードウェアの異常です。RES H コマンドを入力して下さい。(002)

(英語)：Illegal hardware. Enter RES H command. (002)

説明：ハードウェアに異常が発生した場合に出力します。

9.2 エミュレータ共通エラー (100-112)

エラー番号：(100)

メッセージ (日本語)：コマンドまたはデータの入力文字数が多すぎます。(100)

(英語)：Command/Data too long. (100)

説明：128文字以上のコマンド/データが入力された場合に出力します。

エラー番号：(101)

メッセージ (日本語)：コマンドが違います。(101)

(英語)：Unrecognized command. (101)

説明：入力したコマンドが存在しない場合に出力します。

エラー番号：(102)

メッセージ (日本語)：コマンド形式が違います。(102)

(英語)：Command format error. (102)

説明：コマンド名は正しいが、オペランドが正しくない場合に出力します。

エラー番号：(103)

メッセージ(日本語)：実行できないコマンドです。(103)

(英語)：Unexecutable command. (103)

説明：入力したコマンドがこのモードで実行できない場合に出力します。

エラー番号：(104)

メッセージ(日本語)：入力データ・エラーです。(104)

(英語)：Input data error. (104)

説明：入力したデータが正しくない場合に出力します。

エラー番号：(105)

メッセージ(日本語)：中断しました。(105)

(英語)：aborted. (105)

説明：処理の中断。

エラー番号：(106)

メッセージ(日本語)：マッピング・エラーです。(106)

(英語)：Mapping error. (106)

説明：指定されたアドレス範囲に、マッピングされていないメモリ・エリアがある場合に出力します。

エラー番号：(107)

メッセージ(日本語)：マッピングされていません。(107)

(英語)：Non map area access. (107)

説明：コマンド実行中、マッピングされていないメモリにアクセスしようとした場合に出力します。

エラー番号：(108)

メッセージ(日本語)：テストできるメモリがありません。(108)

(英語)：Can not test. (108)

説明：テストできるメモリがない場合に出力します。

エラー番号：(109)

メッセージ(日本語)：他のコマンドでプリンタを使用しています。(109)

(英語)：List device is used by other command. (109)

説明：他のコマンド処理でプリンタを使用している場合に出力します。

エラー番号：(110)

メッセージ(日本語)：重複しています。モジュール名 (110)

(英語)：Warning double define :モジュール名 (110)

説明：LODコマンドで、同一モジュール名が複数指定された場合に出力します。

エラー番号：(111)

メッセージ（日本語）：HLP コマンドは実行できません。(111)

（英語）：Can not execute HLP command ! (111)

説明：起動時にIE78327.EXEと同一のディレクトリにヘルプ・ファイルが存在しない場合に出力します。

エラー番号：(112)

メッセージ（日本語）：キーワードが違います。(112)

（英語）：Keyword Error. (112)

説明：HLP コマンドで、コマンド名が正しくない場合に出力します。

9.3 アナライザ関連エラー (200-210)

エラー番号：(200)

メッセージ (日本語)：実行時間計測タイマーがオーバーフローしました。(200)

(英語)：Emulation Timer overflow. (200)

説明：実行時間計測タイマがオーバーフローしました。

エラー番号：(201)

メッセージ (日本語)：実行命令数計測カウンタがオーバーフローしました。(201)

(英語)：Instruction counter overflow. (201)

説明：実行命令数計測カウンタがオーバーフローしました。

エラー番号：(204)

メッセージ (日本語)：外部トリガ出力がショートしています。(204)

(英語)：External trigger line short. (204)

説明：外部センス・クリップのNo.1が出力端子に接続されています。

エラー番号：(205)

メッセージ(日本語)：PSAの設定がありません。(205)

(英語)：No PSA address. (205)

説明：PSA(サンプル・アドレス)の設定がありません。

エラー番号：(206)

メッセージ(日本語)：サンプル・データがありません。(206)

(英語)：No sampled data. (206)

説明：サンプル・データがありません。

エラー番号：(207)

メッセージ(日本語)：トレース・データがありません。(207)

(英語)：No traced data. (207)

説明：トレース・データがありません。

エラー番号：(208)

メッセージ(日本語)：見つかりません。(208)

(英語)：Not found. (208)

説明：検索データが見つかりません。

エラー番号：(209)

メッセージ(日本語)：トレース・ブロックがありません。(209)

(英語)：Trace block not found. (209)

説明：トレース・ブロックがありません。

エラー番号：(210)

メッセージ(日本語)：トリガフレームがありません。(210)

(英語)：Trigger frame not found. (210)

説明：トリガ・フレームがトレース・データにありません。

9.4 オンライン・アセンブラ関連エラー (300-304)

エラー番号：(300)

メッセージ (日本語)：アセンブル範囲を越えました。(300)

(英語)：Assemble area over! (300)

説明：ASMコマンドで、アクセスできるメモリの範囲を越えた場合に出力します。

エラー番号：(301)

メッセージ (日本語)：逆アセンブル範囲を越えました。(301)

(英語)：Disassemble area over! (301)

説明：DASコマンドで、アクセスできるメモリの範囲を越えた場合に出力します。

エラー番号：(302)

メッセージ：Error! (302)

説明：オブジェクト・コードを生成できないか、あきらかにエラーである場合に出力します。

番号：(303)

メッセージ：Caution! (303)

説明：ジェネリックなオブジェクトが生成された、あるいは注意を要する場合に出力します。

番号：(304)

メッセージ：Warning! (304)

説明：オブジェクトの生成はできるが、正しい動作は望めない場合に出力します。

9.5 シンボル関連エラー (400-409)

エラー番号 : (400)

メッセージ (日本語) : IE78327.SYM が見つかりません。(400)

(英語) : IE78327.SYM file not found. (400)

説明 : SYM L コマンドで、IE78327.SYMがカレント・ディスク上に存在しなかった場合に出力します。

エラー番号 : (401)

メッセージ (日本語) : I Eシンボル・ファイルの形式が違います。(401)

(英語) : Illegal IESYMBOL file. (401)

説明 : SYM L コマンドで、アペンド・シンボル・ファイルの形式が正しくない場合に出力します。

エラー番号 : (402)

メッセージ (日本語) : シンボルが重複しています。(402) シンボル名

(英語) : Double define symbol. (402) シンボル名

説明 : すでに登録されているシンボルを登録しようとした場合に出力します。

エラー番号：(403)

メッセージ(日本語)：モジュール名が重複しています。(403) モジュール名

(英語)：Double define module name. (403) モジュール名

説明：表示されたモジュールが、すでに登録されている場合に出力します。

エラー番号：(404)

メッセージ(日本語)：シンボルがありません。(404)

(英語)：No symbol. (404)

説明：シンボルがない場合に出力します。

エラー番号：(405)

メッセージ(日本語)：予約されたシンボルです。(405)

(英語)：Reserved symbol. (405)

説明：SYM A コマンドで、予約語がシンボルとして定義された場合に出力します。

エラー番号：(406)

メッセージ(日本語)：IEシンボル・テーブルがいっぱいです。(406)

(英語)：IESYMBOL table full. (406)

説明：SYM A/L コマンドで、IEシンボル・セーブ・エリアに空がない場合に出力します。

エラー番号：(407)

メッセージ（日本語）：シンボルが見つかりません。(407)

（英語）：Symbol not found. (407)

説明：SYM C/E コマンドで、指定されたシンボルが存在しない場合
に出力します。

エラー番号：(408)

メッセージ（日本語）：シンボル・モジュール・テーブルがいっぱいです。(408)

（英語）：Symbol module table full. (408)

説明：LODコマンドで、入力できるモジュール数をオーバーした場合に
出力します。

エラー番号：(409)

メッセージ（日本語）：シンボル・テーブルがいっぱいです。(409)

（英語）：Symbol table full. (409)

説明：LODコマンドで、シンボル・セーブ・エリアに空がない場合に出
力します。

エラー番号：(410)

メッセージ(日本語)：モジュールが見つかりません。(410)

(英語)：Module not found. (410)

説明：SYM Dコマンドで、指定されたモジュール名が存在しない場合に出力します。

エラー番号：(411)

メッセージ(日本語)：重複しています。モジュール名(411)

(英語)：Warning double define：モジュール名(411)

説明：LODコマンドで、同一モジュール名が複数指定された場合に出力します。

エラー番号：(412)

メッセージ(日本語)：モジュール名 指定されたモジュール名は存在しません。(412)

(英語)：モジュール名 Module not exist. (412)

説明：LODコマンドで、シンボル・ファイル内に指定されたモジュール名のシンボル・ブロックが存在しなかった場合に出力します。

9.6 ファイル関連エラー (ロード/セーブ/PGM) (500-520)

エラー番号: (500)

メッセージ (日本語): クローズできません。(500) ファイル名

(英語): Can not close (500) ファイル名

説明: 表示されたファイルのクローズが正常にできなかった場合に出力します。

エラー番号: (501)

メッセージ (日本語): クローズできません。×××コマンドを中止しました。(501)

(英語): Can not close ファイル名. Cancel ××× command. (501)

説明: ×××コマンド実行中、表示されたファイルのクローズが正常にできなかった場合に出力します (×××はSTR, LST, COMの各コマンド)。

エラー番号: (502)

メッセージ (日本語): オープンできません。(502) ファイル名

(英語): Can not open. (502) ファイル名

説明: 指定されたファイルがオープンできなかった場合に出力します。

エラー番号：(503)

メッセージ(日本語)：読取りができません。(503) 771#名

(英語)：Disk read error. (503) 771#名

説明：表示されたファイルの読み込みで異常を見つけた場合に出力します。

エラー番号：(504)

メッセージ(日本語)：読取りができません。771#名 STRコマンドを中止します。(504)

(英語)：Disk read error.771#名 Cancel STR command (504)

説明：STRコマンド実行中、表示されたファイルの読み込みで異常を見つけた場合に出力します。

エラー番号：(505)

メッセージ(日本語)：書き込みができません。(505) 771#名

(英語)：Disk write error. (505) 771#名

説明：表示されたファイルの書き込みで異常を見つけた場合に出力します。

エラー番号：(506)

メッセージ(日本語)：書き込みができません。771#名 XXXコマンドを中止します。(506)

(英語)：Disk write error.771#名 Cancel XXX command. (506)

説明：XXXのコマンド実行中、表示されたファイルの書き込みで異常を見つけた場合に出力します。(XXXは、LST, COMの各コマンド)

エラー番号：(507)

メッセージ（日本語）：読取り専用のファイルです。(507) ファイル名

（英語）：Read only file. (507) ファイル名

説明：ファイルの属性がR/Oのファイルに対し、同一名のファイルを新たにメイクしようとした場合に出力します。

エラー番号：(508)

メッセージ（日本語）：ファイルが作れません。(508) ファイル名

（英語）：File make error. (508) ファイル名

説明：表示されたファイルを作成できなかった場合に出力します。

エラー番号：(509)

メッセージ（日本語）：ファイルが見つかりません。(509)

（英語）：File not found. (509)

説明：指定されたファイル名が存在しない場合に出力します。

エラー番号：(510)

メッセージ（日本語）：ヘルプ・ファイルが見つかりません。(510)

（英語）：HELP file not found. (510)

説明：HELPコマンド実行時、IE78327.EXEと同一のディレクトリにヘルプ・ファイルが見つからなかった場合に出力します。

エラー番号：(511)

メッセージ（日本語）：ファイル名が不正です。(511)

（英語）：Bad file entry. (511)

説明：ファイル名の記述が正しくない場合に出力します。

エラー番号：(512)

メッセージ（日本語）：既にオープンされたファイルです。(512) ファイル名

（英語）：File already opened. (512) ファイル名

説明：すでにオープン済みのファイル名を指定した場合に出力します。

エラー番号：(513)

メッセージ（日本語）：既に存在するファイルです。(513)

（英語）：File already exists. (513)

説明：指定されたファイル名がすでに存在する場合に出力します。

エラー番号：(514)

メッセージ（日本語）：予約されたファイル名です。(514)

（英語）：Reserved file name. (514)

説明：システム・ソフトが使う、予約されたファイル名を指定した場合に出力します。

エラー番号：(515)

メッセージ(日本語)：不正文字がありました。(515)

(英語)：Bad character. (515)

説明：オブジェクトのロード/セーブ時に、正しくない文字を検出した場合に出力します。

エラー番号：(516)

メッセージ(日本語)：チェックサムエラーです。(516)

(英語)：Check sum error. (516)

説明：オブジェクトのロード/セーブ時にチェック・サム・エラーを検出した場合に出力します。

エラー番号：(517)

メッセージ(日本語)：レコード形式が違います。(517)

(英語)：Illegal record. (517)

説明：LODコマンドで、シンボル・テーブル・ファイルのレコード形式が正しくない場合に出力します。

エラー番号：(518)

メッセージ(日本語)：失敗しました。(518)


(英語)：Load failed. (518)

説明：LODコマンドで、シンボル/オブジェクトのロード中にエラーを検出した場合に出力します。

第10章 オンライン・アセンブラ、逆アセンブラ仕様

この章では、IE-78327-Rの対象デバイス（ μ PD78320、78322、78P322、78327、78328、78P328）の命令セット、特殊機能レジスタ（SFR）、オンライン・アセンブラ、逆アセンブラの仕様について説明します。

凡例

-  : リターン・キー（CR（0DH））の入力を表します。
- R/O : Read Only（リード・オンリー）を表します。
- R/W : Read Write（リード・ライト）を表します。
- W/O : Write Only（ライト・オンリー）を表します。
- このマニュアルで用いられている画面表示例、入力例はPC-9800シリーズをホスト・マシンとして使用した場合のものであります。

10.1 対象デバイスの命令セット一覧

対象デバイス（ μ PD78320、78322、78P322、78327、78328、78P328）の命令セットは、命令機能に応じて次のように分類されます。

- 8ビット・データ転送命令
- 16ビット・データ転送命令
- 8ビット演算命令
- 16ビット演算命令
- 乗除算命令
- 符号付き乗除算命令
- 増減命令
- シフト，ローテート命令
- BCD補正命令
- データ変換命令
- ビット操作命令
- コール，リターン命令
- スタック操作命令
- 特殊命令
- 無条件分岐命令
- 条件付き分岐命令
- コンテキスト・スイッチング命令
- スtring命令
- CPU制御命令

● オペランドの表現形式と記述方法

表現形式	記述方法
r r1 r2	R0, R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15 R0, R1, R2, R3, R4, R5, R6, R7 C, B
rp rp1 rp2	RP0, RP1, RP2, RP3, RP4, RP5, RP6, RP7 RP0, RP1, RP2, RP3, RP4, RP5, RP6, RP7 DE, HL, VP, UP
sfr sfrp	特殊機能レジスタ略号 注1 特殊機能レジスタ略号 (16ビット操作可能レジスタ) 注2
post	RP0, RP1, RP2, RP3, RP4, RP5/PSW, RP6, RP7 (複数記述可能。ただし、RP5はPUSH, POP命令, PSWはPUSHU, POPUに限る。)
mem	[DE], [HL], [DE+], [HL+], [DE-], [HL-], [VP], [UP]; レジスタ・インダイレクト・モード [DE+A], [HL+A], [DE+B], [HL+B], [VP+DE], [VP+HL]; ベース・インデクスト・モード [DE+byte], [HL+byte], [VP+byte], [UP+byte], [SP+byte]; ベース・モード word[A], word[B], word[DE], word[HL]; インデクスト・モード
saddr saddrp	0FE20H-0FF1FH イミューディエト・データまたはレーベル 0FE20H-0FF1EH イミューディエト・データ (ただしbit0 = 0) またはレーベル (16ビット操作時)
\$addr16 !addr16	0000H-0FDFH イミューディエト・データまたはレーベル; レラティブ・アドレッシング 0000H-0FDFH イミューディエト・データまたはレーベル; イミューディエト・アドレッシング (ただしMOV命令では0FFFFHまで記述可能)
addr11 addr5	800H-0FFFH イミューディエト・データまたはレーベル 40H-7EH イミューディエト・データ (ただしbit0 = 0) 注3 またはレーベル
word byte bit n	16ビット・イミューディエト・データまたはレーベル 8ビット・イミューディエト・データまたはレーベル 3ビット・イミューディエト・データまたはレーベル 3ビット・イミューディエト・データ (0-7)

- 注1. 表10-1、10-3 8ビット特殊機能レジスタ (SFR) 一覧 参照。
 2. 表10-2、10-4 16ビット特殊機能レジスタ (SFR) 一覧 参照。
 3. bit0=1 (奇数アドレス) へのワード・アクセスはしないでください。

- 備考1. rpとrplは記述できるレジスタ名は同じですが、発生するコードが異なります
(μ PD78322 ユーザーズ・マニュアル (IEU-619)、 μ PD78328 ユーザーズ・マニュアル (IEU-693) 参照)。
2. r, r1, rp, rplおよびpostは、絶対名称 (R0-R15, RP0-RP7) のほかに機能名称 (X, A, C, B, E, D, L, H, AX, BC, DE, HL, VP, UP) でも記述可能。
3. イミューディエト・アドレッシングは、全空間をアドレス可能。レラティブ・アドレッシングは、次に続く命令の先頭アドレスから-128~+127の範囲のみアドレス可能。
4. オペレーション説明上の凡例 (1/2)

記号	説明
A	Aレジスタ; 8ビット・アキュムレータ
X	Xレジスタ
B	Bレジスタ
C	Cレジスタ
D	Dレジスタ
E	Eレジスタ
H	Hレジスタ
L	Lレジスタ
R0-R15	レジスタ0-15 (絶対名称)
AX	レジスタ・ペア (AX); 16ビット・アキュムレータ
BC	レジスタ・ペア (BC)
DE	レジスタ・ペア (DE)
HL	レジスタ・ペア (HL)
RP0-RP7	レジスタ・ペア0-7 (絶対名称)
PC	プログラム・カウンタ
SP	スタック・ポインタ
UP	ユーザ・スタック・ポインタ
PSW	プログラム・ステータス・ワード
CY	キャリー・フラグ
AC	補助キャリー・フラグ
Z	ゼロ・フラグ
P/V	パリティ/オーバーフロー・フラグ
T	サイン・フラグ
TPF	テーブル・ポジション・フラグ
RBS	レジスタ・バンク・セレクト・フラグ
RSS	レジスタ・セット・セレクト・フラグ
IE	割り込み許可フラグ
STBC	スタンバイ・コントロール・レジスタ
WDM	ウォッチドッグ・タイマ・モード・レジスタ
()	()内のアドレスまたはレジスタの内容で示されるメモリの内容。(+), (-)の場合は、命令実行後()内の内容が+1または-1される。
(())	(())内のアドレスで示されるメモリの内容により示されるメモリの内容

備考4. オペレーション説明上の凡例（2／2）

記号	説明
××H	16進数
×H, ×L	16ビット・レジスタの上位8ビット, 下位8ビット
!××	ダイレクト・アドレッシングによるアドレス
\$××	レラティブ・アドレッシングによるアドレス

5. フラグ欄の記号

記号	説明
(ブランク)	変化なし
0	0にクリアされる
1	1にセットされる
×	結果に従ってセット、クリアされる
P	P/Vフラグが、パリティ・フラグとして動作する
V	P/Vフラグが、オーバフロー・フラグとして動作する
R	以前に退避した値がリストアされる

命令群	ニモニック	オペランド	オペレーション	フラグ							
				S	Z	AC	P/V	CY			
8ビット・データ転送命令	MOV	r1, #byte	r1 ← byte								
		saddr, #byte	(saddr) ← byte								
		sfr注, #byte	sfr ← byte								
		r, r1	r ← r1								
		A, r1	A ← r1								
		A, saddr	A ← (saddr)								
		saddr, A	(saddr) ← A								
		saddr, saddr	(saddr) ← (saddr)								
		A, sfr	A ← sfr								
		sfr, A	sfr ← A								
		A, mem	A ← (mem)								
		mem, A	(mem) ← A								
		A, [saddrp]	A ← ((saddrp))								
		[saddrp], A	((saddrp)) ← A								
		A, !addr16	A ← (!addr16)								
		!addr16, A	(!addr16) ← A								
		PSWL, #byte	PSWL ← byte				×	×	×	×	×
		PSWH, #byte	PSWH ← byte								
		PSWL, A	PSWL ← A				×	×	×	×	×
		PSWH, A	PSWH ← A								
	A, PSWL	A ← PSWL									
	A, PSWH	A ← PSWH									
	XCH	A, r1	A ↔ r1								
		r, r1	r ↔ r1								
		A, mem	A ↔ (mem)								
		A, saddr	A ↔ (saddr)								
		A, sfr	A ↔ sfr								
		A, [saddrp]	A ↔ ((saddrp))								
saddr, saddr		(saddr) ↔ (saddr)									
16ビット・データ転送命令	MOVW	rpl, #word	rpl ← word								
		saddrp, #word	(saddrp) ← word								
		sfrp, #word	sfrp ← word								
		rp, rpl	rp ← rpl								
		AX, saddrp	AX ← (saddrp)								
		saddrp, AX	(saddrp) ← AX								
		saddrp, saddrp	(saddrp) ← (saddrp)								
		AX, sfrp	AX ← sfrp								
		sfrp, AX	sfrp ← AX								
		rpl, !addr16	rpl ← (addr16)								
		!addr16, rpl	(addr16) ← rpl								
		AX, mem	AX ← (mem)								
mem, AX	(mem) ← AX										

注 sfr に STBC, WDM を記述した場合は専用命令となり、バイト数がこの命令とは異なります。

命令群	ニモニク	オペランド	オペレーション	フラグ					
				S	Z	AC	P/V	CY	
注	XCHW	AX, saddrp	AX ↔ (saddrp)						
		AX, sfrp	AX ↔ sfrp						
		saddrp, saddrp	(saddrp) ↔ (saddrp)						
		rp, rpl	rp ↔ rpl						
8 ピ ツ ト 演 算 命 令	ADD	A, #byte	A, CY ← A+byte	×	×	×	V	×	
		saddr, #byte	(saddr), CY ← (saddr)+byte	×	×	×	V	×	
		sfr, #byte	sfr, CY ← sfr+byte	×	×	×	V	×	
		r, rl	r, CY ← r+rl	×	×	×	V	×	
		A, saddr	A, CY ← A+(saddr)	×	×	×	V	×	
		A, sfr	A, CY ← A+sfr	×	×	×	V	×	
		saddr, saddr	(saddr), CY ← (saddr)+(saddr)	×	×	×	V	×	
		A, mem	A, CY ← A+(mem)	×	×	×	V	×	
	ADDC	mem, A	(mem), CY ← (mem)+A	×	×	×	V	×	
		A, #byte	A, CY ← A+byte+CY	×	×	×	V	×	
		saddr, #byte	(saddr), CY ← (saddr)+byte+CY	×	×	×	V	×	
		sfr, #byte	sfr, CY ← sfr+byte+CY	×	×	×	V	×	
		r, rl	r, CY ← r+rl+CY	×	×	×	V	×	
		A, saddr	A, CY ← A+(saddr)+CY	×	×	×	V	×	
		A, sfr	A, CY ← A+sfr+CY	×	×	×	V	×	
		saddr, saddr	(saddr), CY ← (saddr)+(saddr)+CY	×	×	×	V	×	
	SUB	A, mem	A, CY ← A+(mem)+CY	×	×	×	V	×	
		mem, A	(mem), CY ← (mem)+A+CY	×	×	×	V	×	
		A, #byte	A, CY ← A-byte		×	×	×		
		saddr, #byte	(saddr), CY ← (saddr)-byte	×	×	×	V	×	
		sfr, #byte	sfr, CY ← sfr-byte	×	×	×	V	×	
		r, rl	r, CY ← r-rl	×	×	×	V	×	
		A, saddr	A, CY ← A-(saddr)	×	×	×	V	×	
		A, sfr	A, CY ← A-sfr	×	×	×	V	×	
	SUBC	saddr, saddr	(saddr), CY ← (saddr)-(saddr)	×	×	×	V	×	
		A, mem	A, CY ← A-(mem)	×	×	×	V	×	
		mem, A	(mem), CY ← (mem)-A	×	×	×	V	×	
		A, #byte	A, CY ← A-byte-CY	×	×	×	V	×	
saddr, #byte		(saddr), CY ← (saddr)-byte-CY	×	×	×	V	×		
sfr, #byte		sfr, CY ← sfr-byte-CY	×	×	×	V	×		
r, rl		r, CY ← r-rl-CY	×	×	×	V	×		
A, saddr		A, CY ← A-(saddr)-CY	×	×	×	V	×		
	A, sfr	A, CY ← A-sfr-CY	×	×	×	V	×		
	saddr, saddr	(saddr), CY ← (saddr)-(saddr)-CY	×	×	×	V	×		
	A, mem	A, CY ← A-(mem)-CY	×	×	×	V	×		
	mem, A	(mem), CY ← (mem)-A-CY	×	×	×	V	×		

注 16ビット・データ転送命令

命令群	ニモニック	オペランド	オペレーション	フラグ				
				S	Z	AC	P/V	CY
8 ビット 演算 命令	AND	A, #byte	$A \leftarrow A \wedge \text{byte}$	×	×		P	
		saddr, #byte	$(\text{saddr}) \leftarrow (\text{saddr}) \wedge \text{byte}$	×	×		P	
		sfr, #byte	$\text{sfr} \leftarrow \text{sfr} \wedge \text{byte}$	×	×		P	
		r, rl	$r \leftarrow r \wedge \text{rl}$	×	×		P	
		A, saddr	$A \leftarrow A \wedge (\text{saddr})$	×	×		P	
		A, sfr	$A \leftarrow A \wedge \text{sfr}$	×	×		P	
		saddr, saddr	$(\text{saddr}) \leftarrow (\text{saddr}) \wedge (\text{saddr})$	×	×		P	
		A, mem	$A \leftarrow A \wedge (\text{mem})$	×	×		P	
	mem, A	$(\text{mem}) \leftarrow (\text{mem}) \wedge A$	×	×		P		
	OR	A, #byte	$A \leftarrow A \vee \text{byte}$	×	×		P	
		saddr, #byte	$(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$	×	×		P	
		sfr, #byte	$\text{sfr} \leftarrow \text{sfr} \vee \text{byte}$	×	×		P	
		r, rl	$r \leftarrow r \vee \text{rl}$	×	×		P	
		A, saddr	$A \leftarrow A \vee (\text{saddr})$	×	×		P	
		A, sfr	$A \leftarrow A \vee \text{sfr}$	×	×		P	
		saddr, saddr	$(\text{saddr}) \leftarrow (\text{saddr}) \vee (\text{saddr})$	×	×		P	
		A, mem	$A \leftarrow A \vee (\text{mem})$	×	×		P	
	mem, A	$(\text{mem}) \leftarrow (\text{mem}) \vee A$	×	×		P		
	XOR	A, #byte	$A \leftarrow A \nabla \text{byte}$	×	×		P	
		saddr, #byte	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$	×	×		P	
		sfr, #byte	$\text{sfr} \leftarrow \text{sfr} \nabla \text{byte}$	×	×		P	
		r, rl	$r \leftarrow r \nabla \text{rl}$	×	×		P	
		A, saddr	$A \leftarrow A \nabla (\text{saddr})$	×	×		P	
		A, sfr	$A \leftarrow A \nabla \text{sfr}$	×	×		P	
		saddr, saddr	$(\text{saddr}) \leftarrow (\text{saddr}) \nabla (\text{saddr})$	×	×		P	
		A, mem	$A \leftarrow A \nabla (\text{mem})$	×	×		P	
	mem, A	$(\text{mem}) \leftarrow (\text{mem}) \nabla A$	×	×		P		
	CMP	A, #byte	A-byte		×	×	×	V
saddr, #byte		(saddr)-byte		×	×	×	V	×
sfr, #byte		sfr-byte		×	×	×	V	×
r, rl		r-rl		×	×	×	V	×
A, saddr		A-(saddr)		×	×	×	V	×
A, sfr		A-sfr		×	×	×	V	×
saddr, saddr		(saddr)-(saddr)		×	×	×	V	×
A, mem		A-(mem)		×	×	×	V	×
mem, A	(mem)-A		×	×	×	V	×	

命令群	ニモニック	オペランド	オペレーション	フラグ					
				S	Z	AC	P/V	CY	
16ビット演算命令	ADDW	AX, #word	AX, CY ← AX+word	×	×	×	V	×	
		saddrp, #word	(saddrp), CY ← (saddrp)+word	×	×	×	V	×	
		sfrp, #word	sfrp, CY ← sfrp+word	×	×	×	V	×	
		rp, rpl	rp, CY ← rp+rpl	×	×	×	V	×	
		AX, saddrp	AX, CY ← AX+(saddrp)	×	×	×	V	×	
		AX, sfrp	AX, CY ← AX+sfrp	×	×	×	V	×	
		saddrp, saddrp	(saddrp), CY ← (saddrp)+(saddrp)	×	×	×	V	×	
	SUBW	AX, #word	AX, CY ← AX-word	×	×	×	V	×	
		saddrp, #word	(saddrp), CY ← (saddrp)-word	×	×	×	V	×	
		sfrp, #word	sfrp, CY ← sfrp-word	×	×	×	V	×	
		rp, rpl	rp, CY ← rp-rpl	×	×	×	V	×	
		AX, saddrp	AX, CY ← AX-(saddrp)	×	×	×	V	×	
		AX, sfrp	AX, CY ← AX-sfrp	×	×	×	V	×	
		saddrp, saddrp	(saddrp), CY ← (saddrp)-(saddrp)	×	×	×	V	×	
	CMPW	AX, #word	AX-word	×	×	×	V	×	
		saddrp, #word	(saddrp)-word	×	×	×	V	×	
		sfrp, #word	sfrp-word	×	×	×	V	×	
		rp, rpl	rp-rpl	×	×	×	V	×	
		AX, saddrp	AX-(saddrp)	×	×	×	V	×	
		AX, sfrp	AX-sfrp	×	×	×	V	×	
		saddrp, saddrp	(saddrp)-(saddrp)	×	×	×	V	×	
乗除算命令	MULU	r1	AX ← AX×r1						
	DIVUW	r1	AX(商), r1(余り) ← AX÷r1						
	MULUW	rpl	AX(上位16ビット), rpl(下位16ビット) ← AX×rp						
	DIVUX	rpl	AXDE(商), rpl(余り) ← AXDE÷rpl						
注	MULW	rpl	AX(上位16ビット), rpl(下位16ビット) ← AX×rpl						
増減命令	INC	r1	r1 ← r1+1	×	×	×	V		
		saddr	(saddr) ← (saddr)+1	×	×	×	V		
	DEC	r1	r1 ← r1-1	×	×	×	V		
		saddr	(saddr) ← (saddr)-1	×	×	×	V		
	INCW	rp2	rp2 ← rp2+1						
		saddrp	(saddrp) ← (saddrp)+1						
DECW	rp2	rp2 ← rp2-1							
	saddrp	(saddrp) ← (saddrp)-1							

注 符号付き乗算命令

命令群	ニモニック	オペランド	オペレーション	フラグ
				S Z AC P/V CY
シフト・ローテート命令	ROR	r1, n	$(CY, r17 \leftarrow r10, r1_{n-1} \leftarrow r1_n) \times n$	P ×
	ROL	r1, n	$(CY, r10 \leftarrow r17, r1_{n+1} \leftarrow r1_n) \times n$	P ×
	RORC	r1, n	$(CY \leftarrow r10, r17 \leftarrow CY, r1_{n-1} \leftarrow r1_n) \times n$	P ×
	ROLC	r1, n	$(CY \leftarrow r17, r10 \leftarrow CY, r1_{n+1} \leftarrow r1_n) \times n$	P ×
	SHR	r1, n	$(CY \leftarrow r10, r17 \leftarrow 0, r1_{n-1} \leftarrow r1_n) \times n$	× × 0 P ×
	SHL	r1, n	$(CY \leftarrow r17, r10 \leftarrow 0, r1_{n+1} \leftarrow r1_n) \times n$	× × 0 P ×
	SHRW	rpl, n	$(CY \leftarrow rpl0, rpl_{15} \leftarrow 0, rpl_{n-1} \leftarrow rpl_n) \times n$	× × 0 P ×
	SHLW	rpl, n	$(CY \leftarrow rpl_{15}, rpl0 \leftarrow 0, rpl_{n+1} \leftarrow rpl_n) \times n$	× × 0 P ×
	ROR4	[rpl]	$A_{3-0} \leftarrow (rpl)_{3-0},$ $(rpl)_{7-4} \leftarrow A_{3-0},$ $(rpl)_{3-0} \leftarrow (rpl)_{7-4}$	
ROL4	[rpl]	$A_{3-0} \leftarrow (rpl)_{7-4},$ $(rpl)_{3-0} \leftarrow A_{3-0},$ $(rpl)_{7-4} \leftarrow (rpl)_{3-0}$		

命令群	ニモニク	オペランド	オペレーション	フラグ					
				S	Z	AC	P/V	CY	
注1	ADJBA		Decimal Adjust Accumulator	×	×	×	P	×	
	ADJBS								
注2	CVTBW		A7=0のとき X ← A, A ← 00H A7=1のとき X ← A, A ← FFH						
ビット 操作 命令	MOVI	CY, saddr. bit	CY ← (saddr. bit)					×	
		CY, sfr. bit	CY ← sfr. bit					×	
		CY, A. bit	CY ← A. bit					×	
		CY, X. bit	CY ← X. bit					×	
		CY, PSWH. bit	CY ← PSWH. bit					×	
		CY, PSWL. bit	CY ← PSWL. bit					×	
		saddr. bit, CY	(saddr. bit) ← CY						
		sfr. bit, CY	sfr. bit ← CY						
		A. bit, CY	A. bit ← CY						
		X. bit, CY	X. bit ← CY						
		PSWH. bit, CY	PSWH. bit ← CY						
		PSWL. bit, CY	PSWL. bit ← CY						
	ANDI	CY, saddr. bit	CY ← CY ∧ (saddr. bit)						×
		CY, /saddr. bit	CY ← CY ∧ (saddr. bit)						×
		CY, sfr. bit	CY ← CY ∧ sfr. bit						×
		CY, /sfr. bit	CY ← CY ∧ sfr. bit						×
		CY, A. bit	CY ← CY ∧ A. bit						×
		CY, /A. bit	CY ← CY ∧ A. bit						×
		CY, X. bit	CY ← CY ∧ X. bit						×
		CY, /X. bit	CY ← CY ∧ X. bit						×
		CY, PSWH. bit	CY ← CY ∧ PSWH. bit						×
		CY, /PSWH. bit	CY ← CY ∧ PSWH. bit						×
		CY, PSWL. bit	CY ← CY ∧ PSWL. bit						×
		CY, /PSWL. bit	CY ← CY ∧ PSWL. bit						×
	OR I	CY, saddr. bit	CY ← CY ∨ (saddr. bit)						×
		CY, /saddr. bit	CY ← CY ∨ (saddr. bit)						×
		CY, sfr. bit	CY ← CY ∨ sfr. bit						×
		CY, /sfr. bit	CY ← CY ∨ sfr. bit						×
		CY, A. bit	CY ← CY ∨ A. bit						×
		CY, /A. bit	CY ← CY ∨ A. bit						×
		CY, X. bit	CY ← CY ∨ X. bit						×
		CY, /X. bit	CY ← CY ∨ X. bit						×
		CY, PSWH. bit	CY ← CY ∨ PSWH. bit						×
		CY, /PSWH. bit	CY ← CY ∨ PSWH. bit						×
		CY, PSWL. bit	CY ← CY ∨ PSWL. bit						×
		CY, /PSWL. bit	CY ← CY ∨ PSWL. bit						×

- 注1. BCD補正命令
2. データ変換命令

命令群	ニモニク	オペランド	オペレーション	フラグ
				S Z AC P/V CY
ビット 操作 命令	XOR1	CY, saddr. bit	$CY \leftarrow CY \nabla (saddr. bit)$	×
		CY, sfr. bit	$CY \leftarrow CY \nabla sfr. bit$	×
		CY, A. bit	$CY \leftarrow CY \nabla A. bit$	×
		CY, X. bit	$CY \leftarrow CY \nabla X. bit$	×
		CY, PSWH. bit	$CY \leftarrow CY \nabla PSWH. bit$	×
		CY, PSWL. bit	$CY \leftarrow CY \nabla PSWL. bit$	×
	SET1	saddr. bit	$(saddr. bit) \leftarrow 1$	
		sfr. bit	$sfr. bit \leftarrow 1$	
		A. bit	$A. bit \leftarrow 1$	
		X. bit	$X. bit \leftarrow 1$	
		PSWH. bit	$PSWH. bit \leftarrow 1$	
		PSWL. bit	$PSWL. bit \leftarrow 1$	× × × × ×
	CLR1	saddr. bit	$(saddr. bit) \leftarrow 0$	
		sfr. bit	$sfr. bit \leftarrow 0$	
		A. bit	$A. bit \leftarrow 0$	
		X. bit	$X. bit \leftarrow 0$	
		PSWH. bit	$PSWH. bit \leftarrow 0$	× × × × ×
		PSWL. bit	$PSWL. bit \leftarrow 0$	
	NOT1	saddr. bit	$(saddr. bit) \leftarrow \overline{(saddr. bit)}$	
		sfr. bit	$sfr. bit \leftarrow \overline{sfr. bit}$	
		A. bit	$A. bit \leftarrow \overline{A. bit}$	
		X. bit	$X. bit \leftarrow \overline{X. bit}$	
		PSWH. bit	$PSWH. bit \leftarrow \overline{PSWH. bit}$	
		PSWL. bit	$PSWL. bit \leftarrow \overline{PSWL. bit}$	× × × × ×
SET1	CY	$CY \leftarrow 1$	1	
CLR1	CY	$CY \leftarrow 0$	0	
NOT1	CY	$CY \leftarrow \overline{CY}$	×	

命令群	ニモニック	オペランド	オペレーション	フラグ
				S Z AC P/V CY
コ ー ル ・ リ タ ー ン 命 令	CALL	!addr16	(SP-1) ← (PC+3) _H , (SP-2) ← (PC+3) _L , PC ← !addr16, SP ← SP-2	
		rp1	(SP-1) ← (PC+2) _H , (SP-2) ← (PC+2) _L , PC _H ← rp1 _H , PC _L ← rp1 _L , SP ← SP-2	
		[rp1]	(SP-1) ← (PC+2) _H , (SP-2) ← (PC+2) _L , PC _H ← (rp1+1), PC _L ← (rp1), SP ← SP-2	
	CALLF	!addr11	(SP-1) ← (PC+2) _H , (SP-2) ← (PC+2) _L , PC ₁₅₋₁₁ ← 00001, PC ₁₀₋₀ ← !addr11, SP ← SP-2	
	CALLT	[addr5]	(SP-1) ← (PC+1) _H , (SP-2) ← (PC+1) _L , PC _H ← (TPF, 00000000, addr5+1), PC _L ← (TPF, 00000000, addr5), SP ← SP-2	
	BRK		(SP-1) ← PSW _H , (SP-2) ← PSW _L , (SP-3) ← (PC+1) _H , (SP-4) ← (PC+1) _L , PC _L ← (003EH), PC _H ← (003FH), SP ← SP-4, IE ← 0	
	RET		PC _L ← (SP), PC _H ← (SP+1), SP ← SP+2	
	RETB		PC _L ← (SP), PC _H ← (SP+1), PSW _L ← (SP+2), PSW _H ← (SP+3) SP ← SP+4	R R R R R
	RETI		PC _L ← (SP), PC _H ← (SP+1), PSW _L ← (SP+2), PSW _H ← (SP+3) SP ← SP+4	R R R R R

命令群	ニモニク	オペランド	オペレーション	フラグ
				S Z AC P/V CY
スタック操作命令	PUSH	sfrp	$(SP-1) \leftarrow sfr_H, (SP-2) \leftarrow sfr_L$ $SP \leftarrow SP-2$	
		post	$\{(SP-1) \leftarrow post_H, (SP-2) \leftarrow post_L, SP \leftarrow SP-2\} \times n回$ 注1	
		PSW	$(SP-1) \leftarrow PSW_H, (SP-2) \leftarrow PSW_L$ $SP \leftarrow SP-2$	
	PUSHU	post	$\{(UP-1) \leftarrow post_H, (UP-2) \leftarrow post_L, UP \leftarrow UP-2\} \times n回$ 注1	
	POP	sfrp	$sfr_L \leftarrow (SP), sfr_H \leftarrow (SP+1)$ $SP \leftarrow SP+2$	
		post	$\{post_L \leftarrow (SP), post_H \leftarrow (SP+1), SP \leftarrow SP+2\} \times n回$ 注1	
		PSW	$PSW_L \leftarrow (SP), PSW_H \leftarrow (SP+1), SP \leftarrow SP+2$	R R R R R
	POPU	post	$\{post_L \leftarrow (UP), post_H \leftarrow (UP+1), UP \leftarrow UP+2\} \times n回$ 注1	
	MOVW	SP, #word	$SP \leftarrow word$	
		SP, AX	$SP \leftarrow AX$	
		AX, SP	$AX \leftarrow SP$	
	INCW	SP	$SP \leftarrow SP+1$	
	DECW	SP	$SP \leftarrow SP-1$	
特殊命令	CHKL	sfr	(端子レベル)∨(出力バッファ前の信号レベル)	× × P
	CHKLA	sfr	$A \leftarrow \{(端子レベル) \vee (出力バッファ前の信号レベル)\}$	× × P
注2	BR	!addr16	$PC \leftarrow !addr16$	
		rpl	$PC_H \leftarrow rpl_H, PC_L \leftarrow rpl_L$	
		[rpl]	$PC_H \leftarrow (rpl+1), PC_L \leftarrow (rpl)$	
		\$addr16	$PC \leftarrow \$addr16$	

- 注1. nはpostとして記述したレジスタの数です。
 2. 無条件分岐命令

命令群	ニモニク	オペランド	オペレーション	フラグ
				S Z AC P/V CY
条件付き命令	BC	\$addr16	PC ← \$addr16 if CY=1	
	BL			
	BNC	\$addr16	PC ← \$addr16 if CY=0	
	BNL			
	BZ	\$addr16	PC ← \$addr16 if Z=1	
	BE			
	BNZ	\$addr16	PC ← \$addr16 if Z=0	
	BNE			
	BV	\$addr16	PC ← \$addr16 if P/V=1	
	BPE			
	BNV	\$addr16	PC ← \$addr16 if P/V=0	
	BPO			
	BN	\$addr16	PC ← \$addr16 if S=1	
	BP	\$addr16	PC ← \$addr16 if S=0	
	BGT	\$addr16	PC ← \$addr16 if (P/V ≠ S) ∨ Z=0	
	BGE	\$addr16	PC ← \$addr16 if P/V ≠ S=0	
	BLT	\$addr16	PC ← \$addr16 if P/V ≠ S=1	
	BLE	\$addr16	PC ← \$addr16 if (P/V ≠ S) ∨ Z=1	
	BH	\$addr16	PC ← \$addr16 if Z ∨ CY=0	
	BNH	\$addr16	PC ← \$addr16 if Z ∨ CY=1	
	BT	saddr.bit, \$addr16	PC ← \$addr16 if (saddr.bit)=1	
		sfr.bit, \$addr16	PC ← \$addr16 if sfr.bit=1	
		A.bit, \$addr16	PC ← \$addr16 if A.bit=1	
		X.bit, \$addr16	PC ← \$addr16 if X.bit=1	
		PSWH.bit, \$addr16	PC ← \$addr16 if PSWH.bit=1	
		PSWL.bit, \$addr16	PC ← \$addr16 if PSWL.bit=1	
	BF	saddr.bit, \$addr16	PC ← \$addr16 if (saddr.bit)=0	
		sfr.bit, \$addr16	PC ← \$addr16 if sfr.bit=0	
A.bit, \$addr16		PC ← \$addr16 if A.bit=0		
X.bit, \$addr16		PC ← \$addr16 if X.bit=0		
PSWH.bit, \$addr16		PC ← \$addr16 if PSWH.bit=0		
PSWL.bit, \$addr16		PC ← \$addr16 if PSWL.bit=0		
BTCLR	saddr.bit, \$addr16	PC ← \$addr16 if (saddr.bit)=1 then reset (saddr.bit)		
	sfr.bit, \$addr16	PC ← \$addr16 if sfr.bit=1 then reset sfr.bit		
	A.bit, \$addr16	PC ← \$addr16 if A.bit=1 then reset A.bit		
	X.bit, \$addr16	PC ← \$addr16 if X.bit=1 then reset X.bit		
	PSWH.bit, \$addr16	PC ← \$addr16 if PSWH.bit=1 then reset PSWH.bit		
	PSWL.bit, \$addr16	PC ← \$addr16 if PSWL.bit=1 then reset PSWL.bit	× × × × ×	

命令群	ニモニク	オペランド	オペレーション	フラグ
				S Z AC P/V CY
条件付き分岐命令	BFSET	saddr.bit, \$addr16	PC ← \$addr16 if (saddr.bit)=0 then set (saddr.bit)	
		sfr.bit, \$addr16	PC ← \$addr16 if sfr.bit=0 then set sfr.bit	
		A.bit, \$addr16	PC ← \$addr16 if A.bit=0 then set A.bit	
		X.bit, \$addr16	PC ← \$addr16 if X.bit=0 then set X.bit	
		PSWH.bit, \$addr16	PC ← \$addr16 if PSWH.bit=0 then set PSWH.bit	
		PSWL.bit, \$addr16	PC ← \$addr16 if PSQL.bit=0 then set PSQL.bit	× × × × ×
注	DBNZ	r2, \$addr16	r2 ← r2-1, then P0 ← \$addr16 if r2 ≠ 0	
		saddr, \$addr16	(saddr) ← (saddr)-1, then PC ← \$addr16 if (saddr) ≠ 0	
注	BRKCS	RBn	PC _H ↔ R5, PC _L ↔ R4, R7 ← PSWH, R6 ← PSQL, RBS2-0 ← n, RSS ← 0, IE ← 0	
	RETCS	!addr16	PC _H ← R5, PC _L ← R4, R5, R4 ← !addr16, PSWH ← R7, PSWL ← R6	R R R R R
	RETCSB	!addr16	PC _H ← R5, PC _L ← R4, R5, R4 ← !addr16, PSWH ← R7, PSWL ← R6	R R R R R

注 コンテキスト・スイッチング命令

命令群	ニモニック	オペランド	オペレーション	フラグ					
				S	Z	AC	P/V	CY	
ストリ命令	MOV M	[DE+], A	(DE+) ← A, C ← C-1 End if C=0						
		[DE-], A	(DE-) ← A, C ← C-1 End if C=0						
	MOV BK	[DE+], [HL+]	(DE+) ← (HL+), C ← C-1 End if C=0						
		[DE-], [HL-]	(DE-) ← (HL-), C ← C-1 End if C=0						
	XCHM	[DE+], A	(DE+) ↔ A, C ← C-1 End if C=0						
		[DE-], A	(DE-) ↔ A, C ← C-1 End if C=0						
	XCHBK	[DE+], [HL+]	(DE+) ↔ (HL+), C ← C-1 End if C=0						
		[DE-], [HL-]	(DE-) ↔ (HL-), C ← C-1 End if C=0						
	CMPME	[DE+], A	(DE+)-A, C ← C-1 End if C=0 or Z=0		×	×	×	V	×
		[DE-], A	(DE-)-A, C ← C-1 End if C=0 or Z=0		×	×	×	V	×
	CMPBKE	[DE+], [HL+]	(DE+)-(HL+), C ← C-1 End if C=0 or Z=0		×	×	×	V	×
		[DE-], [HL-]	(DE-)-(HL-), C ← C-1 End if C=0 or Z=0		×	×	×	V	×
	CMPMNE	[DE+], A	(DE+)-A, C ← C-1 End if C=0 or Z=1		×	×	×	V	×
		[DE-], A	(DE-)-A, C ← C-1 End if C=0 or Z=1		×	×	×	V	×
	CMPBKNE	[DE+], [HL+]	(DE+)-(HL+), C ← C-1 End if C=0 or Z=1		×	×	×	V	×
		[DE-], [HL-]	(DE-)-(HL-), C ← C-1 End if C=0 or Z=1		×	×	×	V	×
	CMPMC	[DE+], A	(DE+)-A, C ← C-1 End if C=0 or CY=0		×	×	×	V	×
		[DE-], A	(DE-)-A, C ← C-1 End if C=0 or CY=0		×	×	×	V	×
CMPBK C	[DE+], [HL+]	(DE+)-(HL+), C ← C-1 End if C=0 or CY=0		×	×	×	V	×	
	[DE-], [HL-]	(DE-)-(HL-), C ← C-1 End if C=0 or CY=0		×	×	×	V	×	

命令群	ニモニック	オペランド	オペレーション	フラグ				
				S	Z	AC	P/V	CY
ストリング命令	CMPMNC	[DE+], A	(DE+)-A, C ← C-1 End if C=0 or CY=1	×	×	×	V	×
		[DE-], A	(DE-)-A, C ← C-1 End if C=0 or CY=1	×	×	×	V	×
	CMPBKNC	[DE+], [HL+]	(DE+)-(HL+), C ← C-1 End if C=0 or CY=1	×	×	×	V	×
		[DE-], [HL-]	(DE-)-(HL-), C ← C-1 End if C=0 or CY=1	×	×	×	V	×
CPU制御命令	MOV	STBC, #byte	STBC ← byte ^注					
		WDM, #byte	WDM ← byte ^注					
	SWRS		RSS ← $\overline{\text{RSS}}$					
	SEL	RBn	RSS ← 0, RBS2-0 ← n					
		RBn, ALT	RSS ← 1, RBS2-0 ← n					
	NOP		No Operation					
EI		IE ← 1(Enable Interrup)						
DI		IE ← 0(Disable Interru)						

注 STBCレジスタ、WDMレジスタ操作命令のオペコードが異常のとき、オペコード・トラップ割り込みを発生します。

トラップ時のオペレーション

(SP-1) ← PSW_H, (SP-2) ← PSW_L
 (SP-3) ← (PC-4)_H, (SP-4) ← (PC-4)_L
 PC_L ← (003CH), PC_H ← (003DH),
 SP ← SP-4, IE ← 0

10.2 対象デバイスの特殊機能レジスタ (s f r) 一覧

s f r は対象デバイスに μ PD78320シリーズを用いた場合と μ PD78327シリーズを用いた場合とでは、その名称とアドレスが異なります。

また、対象デバイスのs f rはレジスタのビット幅に応じて次のように分類されます。

- 8ビット特殊機能レジスタ
- 16ビット特殊機能レジスタ

次ページより μ PD78320シリーズと μ PD78327シリーズのs f rを示します。

● μ PD78320シリーズのsfr

表10-1 8ビット特殊機能レジスタ一覧 (1/2)

レジスタ	SFR	レジスタ	SFR	レジスタ	SFR	レジスタ	SFR
FF00	P0	FF20	PM0	FF40	PMC0	FF60	RTP
FF01		FF21		FF41	RTPS	FF61	RTPR
FF02	P2	FF22		FF42		FF62	PRDC
FF03	P3	FF23	PM3	FF43	PMC3	FF63	
FF04	P4	FF24		FF44		FF64	
FF05	P5	FF25	PM5	FF45		FF65	
FF06		FF26		FF46		FF66	
FF07	P7	FF27		FF47		FF67	
FF08	P8	FF28	PM8	FF48	PMC8	FF68	ADM
FF09	P9	FF29	PM9	FF49		FF69	
FF0A		FF2A		FF4A		FF6A	
FF0B		FF2B		FF4B		FF6B	ADCRH
FF0C		FF2C		FF4C		FF6C	
FF0D		FF2D		FF4D		FF6D	
FF0E		FF2E		FF4E		FF6E	
FF0F		FF2F		FF4F		FF6F	
FF10		FF30		FF50		FF70	
FF11		FF31		FF51		FF71	
FF12		FF32		FF52		FF72	
FF13		FF33		FF53		FF73	
FF14		FF34		FF54		FF74	
FF15		FF35		FF55		FF75	
FF16		FF36		FF56		FF76	
FF17		FF37		FF57		FF77	
FF18		FF38		FF58		FF78	
FF19		FF39		FF59		FF79	
FF1A		FF3A		FF5A		FF7A	
FF1B		FF3B		FF5B		FF7B	
FF1C		FF3C		FF5C		FF7C	
FF1D		FF3D		FF5D		FF7D	
FF1E		FF3E		FF5E		FF7E	
FF1F		FF3F		FF5F		FF7F	

● μ PD78320シリーズのsfr

表10-1 8ビット特殊機能レジスタ一覧 (2/2)

レジスタ	SFR	レジスタ	SFR	レジスタ	SFR	レジスタ	SFR
FF80	CSIM	FFA0		FFC0	STBC	FFE0	IF0L
FF81		FFA1		FFC1	CCW	FFE1	IF0H
FF82	SBIC	FFA2		FFC2	WDM	FFE2	IF1L
FF83		FFA3		FFC3		FFE3	
FF84		FFA4		FFC4	MM	FFE4	MK0L
FF85		FFA5		FFC5		FFE5	MK0H
FF86	SIO	FFA6		FFC6	PWC	FFE6	MK1L
FF87		FFA7		FFC7		FFE7	
FF88	ASIM	FFA8		FFC8		FFE8	PB0L
FF89		FFA9		FFC9	FCC	FFE9	PB0H
FF8A	ASIS	FFAA		FFCA		FFEA	PB1L
FF8B		FFAB		FFCB		FFEB	
FF8C	RXB	FFAC		FFCC		FFEC	ISM0L
FF8D		FFAD		FFCD		FFED	ISM0H
FF8E	TXS	FFAE		FFCE		FFEE	ISM1L
FF8F		FFAF		FFCF		FFEF	
FF90		FFB0	TMC	FFD0	EXTSFR0	FFF0	CSE0L
FF91		FFB1	BRGM	FFD1	EXTSFR1	FFF1	CSE0H
FF92		FFB2	PRM	FFD2	EXTSFR2	FFF2	CSE1L
FF93		FFB3		FFD3	EXTSFR3	FFF3	
FF94		FFB4		FFD4	EXTSFR4	FFF4	INTM0
FF95		FFB5		FFD5	EXTSFR5	FFF5	INTM1
FF96		FFB6		FFD6	EXTAFR6	FFF6	
FF97		FFB7		FFD7	EXTSFR7	FFF7	
FF98		FFB8	TOC0	FFD8	EXTSFR8	FFF8	ISPR
FF99		FFB9	TOC1	FFD9	EXTSFR9	FFF9	PRSL
FF9A		FFBA		FFDA	EXTSFR10	FFFA	
FF9B		FFBB		FFDB	EXTSFR11	FFFB	
FF9C		FFBC		FFDC	EXTSFR12	FFFC	
FF9D		FFBD		FFDD	EXTAFR13	FFFD	
FF9E		FFBE		FFDE	EXTSFR14	FFFE	
FF9F		FFBF	RPUM	FFDF	EXTAFR15	FFFF	

● μ PD78320シリーズのsfr

表10-2 16ビット特殊機能レジスタ一覧 (1/2)

レジスタ	SFR	レジスタ	SFR	レジスタ	SFR	レジスタ	SFR
FF00		FF20		FF40		FF60	
FF01		FF21		FF41		FF61	
FF02		FF22		FF42		FF62	
FF03		FF23		FF43		FF63	
FF04		FF24		FF44		FF64	
FF05		FF25		FF45		FF65	
FF06		FF26		FF46		FF66	
FF07		FF27		FF47		FF67	
FF08		FF28		FF48		FF68	
FF09		FF29		FF49		FF69	
FF0A	TM0LW	FF2A	TM0UW	FF4A		FF6A	ADCR
FF0B		FF2B		FF4B		FF6B	
FF0C		FF2C	TM1	FF4C	BRG	FF6C	
FF0D		FF2D		FF4D		FF6D	
FF0E		FF2E		FF4E		FF6E	
FF0F		FF2F		FF4F		FF6F	
FF10	CTX0LW	FF30	CTX0UW	FF50		FF70	CM00
FF11		FF31		FF51		FF71	
FF12	CT01LW	FF32	CT01UW	FF52		FF72	CM01
FF13		FF33		FF53		FF73	
FF14	CT02LW	FF34	CT02UW	FF54		FF74	CM02
FF15		FF35		FF55		FF75	
FF16	CT03LW	FF36	CT03UW	FF56		FF76	CM03
FF17		FF37		FF57		FF77	
FF18	CCX0LW	FF38	CCX0UW	FF58		FF78	
FF19		FF39		FF59		FF79	
FF1A	CC01LW	FF3A	CC01UW	FF5A		FF7A	
FF1B		FF3B		FF5B		FF7B	
FF1C	CC10	FF3C		FF5C		FF7C	CM10
FF1D		FF3D		FF5D		FF7D	
FF1E		FF3E		FF5E		FF7E	CM11
FF1F		FF3F		FF5F		FF7F	

● μ PD78320シリーズのsfr

表10-2 16ビット特殊機能レジスタ一覧 (2/2)

レジスタ	SFR	レジスタ	SFR	レジスタ	SFR	レジスタ	SFR
FF80		FFA0		FFC0		FFE0	IF0
FF81		FFA1		FFC1		FFE1	
FF82		FFA2		FFC2		FFE2	IF1
FF83		FFA3		FFC3		FFE3	
FF84		FFA4		FFC4		FFE4	MK0
FF85		FFA5		FFC5		FFE5	
FF86		FFA6		FFC6		FFE6	MK1
FF87		FFA7		FFC7		FFE7	
FF88		FFA8		FFC8		FFE8	PB0
FF89		FFA9		FFC9		FFE9	
FF8A		FFAA		FFCA		FFEA	PB1
FF8B		FFAB		FFCB		FFEB	
FF8C		FFAC		FFCC		FFEC	ISM0
FF8D		FFAD		FFCD		FFED	
FF8E		FFAE		FFCE		FFEE	ISM1
FF8F		FFAF		FFCF		FFEF	
FF90		FFB0		FFD0		FFF0	CSE0
FF91		FFB1		FFD1		FFF1	
FF92		FFB2		FFD2		FFF2	CSE1
FF93		FFB3		FFD3		FFF3	
FF94		FFB4		FFD4		FFF4	
FF95		FFB5		FFD5		FFF5	
FF96		FFB6		FFD6		FFF6	
FF97		FFB7		FFD7		FFF7	
FF98		FFB8		FFD8		FFF8	
FF99		FFB9		FFD9		FFF9	
FF9A		FFBA		FFDA		FFFA	
FF9B		FFBB		FFDB		FFFB	
FF9C		FFBC		FFDC		FFFC	
FF9D		FFBD		FFDD		FFFD	
FF9E		FFBE		FFDE		FFFE	
FF9F		FFBF		FFDF		FFFF	

● μ PD78327シリーズのsfr

表10-3 8ビット特殊機能レジスタ一覧 (1/2)

レジスタ	SFR	レジスタ	SFR	レジスタ	SFR	レジスタ	SFR
FF00	P0	FF20	PM0	FF40		FF60	POL
FF01		FF21		FF41		FF61	POH
FF02	P2	FF22		FF42		FF62	PRDC
FF03	P3	FF23	PM3	FF43	PMC3	FF63	RTPC
FF04	P4	FF24		FF44		FF64	PWMC
FF05	P5	FF25	PM5	FF45		FF65	
FF06		FF26		FF46		FF66	PWMB
FF07	P7	FF27		FF47		FF67	
FF08	P8	FF28	PM8	FF48	PMC8	FF68	ADM
FF09	P9	FF29	PM9	FF49		FF69	
FF0A		FF2A		FF4A		FF6A	
FF0B		FF2B		FF4B		FF6B	ADCRH
FF0C		FF2C		FF4C		FF6C	
FF0D		FF2D		FF4D		FF6D	
FF0E		FF2E		FF4E		FF6E	
FF0F		FF2F		FF4F		FF6F	
FF10		FF30		FF50		FF70	
FF11		FF31		FF51		FF71	
FF12		FF32		FF52		FF72	
FF13		FF33		FF53		FF73	
FF14		FF34		FF54		FF74	
FF15		FF35		FF55		FF75	
FF16		FF36		FF56		FF76	
FF17		FF37		FF57		FF77	
FF18		FF38		FF58		FF78	
FF19		FF39		FF59		FF79	
FF1A		FF3A		FF5A		FF7A	
FF1B		FF3B		FF5B		FF7B	
FF1C		FF3C		FF5C		FF7C	
FF1D		FF3D		FF5D		FF7D	
FF1E		FF3E		FF5E		FF7E	
FF1F		FF3F		FF5F		FF7F	

● μ PD78327シリーズのsfr

表10-3 8ビット特殊機能レジスタ一覧 (2/2)

レジスタ	SFR	レジスタ	SFR	レジスタ	SFR	レジスタ	SFR
FF80	CSIM	FFA0		FFC0	STBC	FFE0	IF0L
FF81		FFA1		FFC1	CCW	FFE1	IF0H
FF82	SBIC	FFA2		FFC2	WDM	FFE2	IF1L
FF83		FFA3		FFC3		FFE3	
FF84		FFA4		FFC4	MM	FFE4	MK0L
FF85		FFA5		FFC5		FFE5	MK0H
FF86	SIO	FFA6		FFC6	PWC	FFE6	MK1L
FF87		FFA7		FFC7		FFE7	
FF88	ASIM	FFA8		FFC8		FFE8	PB0L
FF89		FFA9		FFC9	FCC	FFE9	PB0H
FF8A	ASIS	FFAA		FFCA		FFEA	PB1L
FF8B		FFAB		FFCB		FFEB	
FF8C	RxB	FFAC		FFCC		FFEC	ISM0L
FF8D		FFAD		FFCD		FFED	ISM0H
FF8E	TxS	FFAE		FFCE		FFEE	ISM1L
FF8F		FFAF		FFCF		FFEF	
FF90		FFB0	TMC0	FFD0	EXTSFR0	FFF0	CSE0L
FF91		FFB1	BRGM	FFD1	EXTSFR1	FFF1	CSE0H
FF92		FFB2	TMC1	FFD2	EXTSFR2	FFF2	CSE1L
FF93		FFB3		FFD3	EXTSFR3	FFF3	
FF94		FFB4	TUM	FFD4	EXTSFR4	FFF4	INTMO
FF95		FFB5		FFD5	EXTSFR5	FFF5	
FF96		FFB6		FFD6	EXTAFR6	FFF6	
FF97		FFB7		FFD7	EXTSFR7	FFF7	
FF98		FFB8		FFD8	EXTSFR8	FFF8	ISPR
FF99		FFB9		FFD9	EXTSFR9	FFF9	PRSL
FF9A		FFBA	TOUT	FFDA	EXTSFR10	FFFA	
FF9B		FFBB		FFDB	EXTSFR11	FFFB	
FF9C		FFBC		FFDC	EXTSFR12	FFFC	
FF9D		FFBD		FFDD	EXTAFR13	FFFD	
FF9E		FFBE		FFDE	EXTSFR14	FFFE	
FF9F		FFBF		FFDF	EXTAFR15	FFFF	

● μ PD78327シリーズのsfr

表10-4 16ビット特殊機能レジスタ一覧 (1/2)

レジスタ	SFR	レジスタ	SFR	レジスタ	SFR	レジスタ	SFR
FF00		FF20		FF40		FF60	
FF01		FF21		FF41		FF61	
FF02		FF22		FF42		FF62	
FF03		FF23		FF43		FF63	
FF04		FF24		FF44		FF64	
FF05		FF25		FF45		FF65	
FF06		FF26		FF46		FF66	
FF07		FF27		FF47		FF67	
FF08		FF28		FF48		FF68	
FF09		FF29		FF49		FF69	
FF0A		FF2A	TM0	FF4A		FF6A	ADCR
FF0B		FF2B		FF4B		FF6B	
FF0C	TM2	FF2C	TM1	FF4C	BRG	FF6C	
FF0D		FF2D		FF4D		FF6D	
FF0E		FF2E		FF4E		FF6E	
FF0F		FF2F		FF4F		FF6F	
FF10		FF30		FF50		FF70	CM00S
FF11		FF31		FF51		FF71	
FF12		FF32		FF52		FF72	CM01S
FF13		FF33		FF53		FF73	
FF14		FF34		FF54		FF74	CM02S
FF15		FF35		FF55		FF75	
FF16		FF36		FF56		FF76	CM03S
FF17		FF37		FF57		FF77	
FF18		FF38		FF58		FF78	CM04S
FF19		FF39		FF59		FF79	
FF1A		FF3A		FF5A		FF7A	CM05S
FF1B		FF3B		FF5B		FF7B	
FF1C	CC10	FF3C		FF5C		FF7C	CM06
FF1D		FF3D		FF5D		FF7D	
FF1E		FF3E		FF5E		FF7E	CM20
FF1F		FF3F		FF5F		FF7F	

● μ PD78327シリーズのsfr

表10-4 16ビット特殊機能レジスタ一覧 (2/2)

レジスタ	SFR	レジスタ	SFR	レジスタ	SFR	レジスタ	SFR
FF80		FFA0		FFC0		FFE0	IF0
FF81		FFA1		FFC1		FFE1	
FF82		FFA2		FFC2		FFE2	IF1
FF83		FFA3		FFC3		FFE3	
FF84		FFA4		FFC4		FFE4	MK0
FF85		FFA5		FFC5		FFE5	
FF86		FFA6		FFC6		FFE6	MK1
FF87		FFA7		FFC7		FFE7	
FF88		FFA8		FFC8		FFE8	PB0
FF89		FFA9		FFC9		FFE9	
FF8A		FFAA		FFCA		FFEA	PB1
FF8B		FFAB		FFCB		FFEB	
FF8C		FFAC		FFCC		FFEC	ISM0
FF8D		FFAD		FFCD		FFED	
FF8E		FFAE		FFCE		FFEE	ISM1
FF8F		FFAF		FFCF		FFEF	
FF90	CM00R	FFB0		FFD0		FFF0	CSE0
FF91		FFB1		FFD1		FFF1	
FF92	CM01R	FFB2		FFD2		FFF2	CSE1
FF93		FFB3		FFD3		FFF3	
FF94	CM02R	FFB4		FFD4		FFF4	
FF95		FFB5		FFD5		FFF5	
FF96	CM03R	FFB6		FFD6		FFF6	
FF97		FFB7		FFD7		FFF7	
FF98	CM04R	FFB8		FFD8		FFF8	
FF99		FFB9		FFD9		FFF9	
FF9A	CM05R	FFBA		FFDA		FFFA	
FF9B		FFBB		FFDB		FFFB	
FF9C		FFBC		FFDC		FFFC	
FF9D		FFBD		FFDD		FFFD	
FF9E		FFBE		FFDE		FFFE	
FF9F		FFBF		FFDF		FFFF	

10.3 オンライン・アセンブラ仕様

オンライン・アセンブラ仕様は、ASMコマンドにおいて適用されます。
アセンブラ仕様は、次に示す14項目で規定されています。

- ① 文字セット
- ② シンボル定義
- ③ コメント行
- ④ オペランドに用いる数値表現
- ⑤ オペランドに用いるシンボル表現
- ⑥ オペランドに用いる式表現
- ⑦ 疑似命令
- ⑧ 命令コードの生成規則
- ⑨ `sfr` 操作命令に対するエラー・チェック
- ⑩ `saddr` 空間操作命令に対するエラー・チェック
- ⑪ アドレッシング・モードの省略
- ⑫ `PUSHR`、`POPR`、`PUSHU`、`POPU`のオペランド記述
- ⑬ エラー表示
- ⑭ 予約語一覧

(1) 文字セット

本アセンブラで使用できる文字を次に示します。

A-Z a-z @ ? _ 0-9 + - * /
\$! [] # () ; . , ¥ \

¥注1、\注2は、シンボル表現としてのみ使用できます。

小文字は大文字相当として扱われます。

注1. PC-9800シリーズと接続して使用する場合。

2. IBM PCシリーズと接続して使用する場合。

(2) シンボル定義

このアセンブラでは、ラベルなどのシンボル定義はできません。
しかし、数値の代わりに定義済みのシンボルを用いることは可能です。

(3) コメント行

このアセンブラでは、; から ⊞ までをコメントとして扱います。

(4) オペランドに用いる数値表現

コマンド入力時に使用する数値の記述仕様にしがいます。

詳細は、第6章 数値、シンボル、式の記述仕様 を参照してください。

(5) オペランドに用いるシンボル表現

コマンド入力時に使用するシンボルの記述仕様にしがいます。

詳細は、第6章 数値、シンボル、式の記述仕様 を参照してください。

(6) オペランドに用いる式表現

コマンド入力時に使用する、式の記述仕様にしたがいます。

詳細は、第6章 数値、シンボル、式の記述仕様 を参照してください。

(7) 疑似命令

このアセンブラでは次の疑似命令をサポートしています。

(a) ORG addr16

ORG命令は、ORG命令の次の命令を addr16 に置きます。

addr16 が、カレント・ロケーションよりも小さい値だった場合、Caution が表示されます。

addr16 > 0FE7FH の場合、Error が表示されます。

(b) DB byte, …byte

DB命令は、カレント・ロケーションにバイト・データを置きます。

複数のバイト・データが、で区切られている場合、カレント・ロケーション以降にバイト・データを順番に置きます。

データを置くロケーションが 0FE7FH を越えた場合、オペランドに word データがあった場合、Error が表示されます。

エラーがあった場合、データはすべて無効となります。

(c) DW word, …word

DW命令は、カレント・ロケーションに word データの Low byte、カレント・ロケーション+1 に High byte を置きます。

複数の word データが、で区切られている場合、カレント・ロケーション以降に上記の規則にしたがって順番に置きます。

データを置くロケーションが 0FE7FH を越えた場合、Error が表示されます。

エラーがあった場合、データはすべて無効となります。

(d) DS word

DS命令は、DS命令の次の命令を、カレント・ロケーション + word に置きます。カレント・ロケーション + word が 0FE7FH を越えた場合、Error が表示されます。カレント・ロケーション + word が 0FFFFH を越えた場合、オーバフローした上位桁は切り捨てられ、Caution が表示されます。

(e) END

END命令によって、ASMコマンドを終了します。

(8) 命令コードの生成規則

μPD78320、78322、78P322、78327、78328、78P328は、0FE20H-0FFFFHのメモリ空間に対して次の2種類のアドレッシングができます。

(a) ショート・ダイレクト・アドレッシング

IE-78327-Rのオンライン・アセンブラは、0FE20H-0FF1FHのメモリ空間に対してショート・ダイレクト・アドレッシングを適用します。

(b) 特殊機能レジスタ(SFR)アドレッシング

IE-78327-Rのオンライン・アセンブラは、0FF20H-0FFFFHのメモリ空間に対して特殊機能レジスタ(SFR)アドレッシングを適用します。

(9) sfr操作命令に対するエラー・チェック

sfr空間(特殊機能レジスタ空間)に対する操作命令に対して、次ページに示すアドレス表現方法に応じてエラー・チェックを行います。

(a) アドレスが `s f r` 予約語で表現された場合

`s f r`、`s f r p`のチェック

`s f r` に対する16ビット操作命令または `s f r p` に対する8ビット操作命令の場合、Warning が表示されます。

読み出し、書き込み専用の属性のチェック

R/Oの `s f r` に対する書き込み命令、W/Oの `s f r` に対する読み出し命令の場合、Warning が表示されます。

(b) アドレスが数値、シンボル、式で表現された場合

`s f r` 存在のチェック

`s f r` が存在しない場合、Warning が表示されます。

`s f r`、`s f r p`のチェック

`s f r` に対する16ビット操作命令および、`s f r p` に対する8ビット操作命令の場合、Warning が表示されます。

読み出し、書き込み専用の属性のチェック

R/Oの `s f r` に対する書き込み命令、W/Oの `s f r` に対する読み出し命令の場合、Warning が表示されます。

(10) `saddr` 空間操作命令に対するエラー・チェック

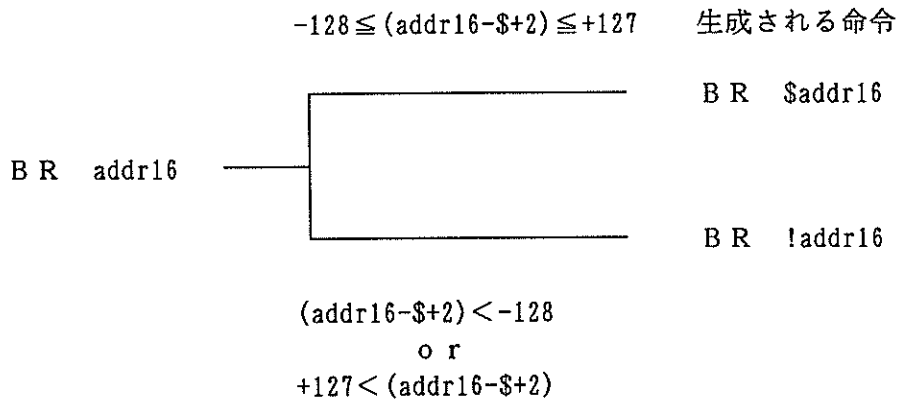
`saddr` 空間に対する操作命令に対して、16ビット操作時はアドレスの偶数または奇数のチェックをします。

奇数アドレスに対する16ビット操作命令の場合、Warning を表示します。

(11) アドレッシング・モードの省略

ブランチ系の命令で適用するアドレッシング・モード (Absolute or Relative) が命令によってははっきりと決まっている場合、アドレッシング・モードは省略できます。

ブランチ命令 (BR) でアドレッシング・モードを省略した場合、`addr16` の値により Caution を表示して最短コードを生成します。
最短コードの生成規則を次に示します。



ロケーション・カウンタの \$ を用いる場合、アドレッシング・モードは省略できません。

例1. アドレッシング・モードの省略

(通常のコマンド)		(省略形)
C A L L !addr16	→	C A L L addr16
C A L L \$addr16	→	C A L L addr16

2. ロケーション・カウンタを用いたアドレッシング

BC \$ \$ ± n ← \$ ± n 番地へのジャンプを表します。

		アドレッシング・モードの \$
		ロケーション・カウンタの \$

(12) **PUSHR、POPR、PUSHU、POPU**のオペランド記述

オペランドの記述の順番は規定されません。

同じオペランドが二つ以上記述された場合、Error を表示します。

(13) エラー表示

本アセンブラのエラー・メッセージには、次の3種類があります。

(a) Error

オブジェクト・コードを生成できない場合、または明らかにエラーである場合に表示されます。

例 Error となる命令記述

```

ORG 0FF00H ← 0FF00H 番地にプログラムは置けません
DB 0FF00H ← ワード・データは記述できません
MOV J, #byte ← シンボルが登録されていません
BR $ ← アドレスの指定がありません
    
```

(b) Warning

オブジェクト・コードは生成できますが、正しい動作が望めない場合に表示されます。

例 Warning となる命令記述

```

MOV sfr, #byte ← sfr に対して8ビット操作をしています
MOVW sfr, #word ← sfr に対して16ビット操作をしています
MOVW odd saddr, #word ← 奇数の saddr に対して16ビット操作をしています
BR !0FF00H ← sfr 空間へブランチしています
    
```

(c) Caution

アドレスが自動生成された場合に表示します。

例 アドレスが自動生成される場合

```

ORG $-100H
BR 100H
    
```

(14) 予約語一覧表

IE-78327-Rのオンライン・アセンブラの予約語を示します。

A	BTCLR	CY
AC	BV	D
ADCR	BZ	DB
ADCRH	C	DBNZ
ADD	CALL	DE
ADDC	CALLF	DEC
ADDW	CALLT	DECW
ADJ4A	CC10	DI
ADJ4S	CCW	DIVUW
ADM	CHK	DIVUX
ALT	CHKR	DS
AND	CLR1	DW
AND1	CM00R	E
ASIM	CM01R	EI
ASIS	CM02R	END
AX	CM03R	EXTSFR1
B	CM04R	EXTSFR2
BC	CM05R	EXTSFR3
BE	CM00S	EXTSFR4
BF	CM01S	EXTSFR5
BFSET	CM02S	EXTSFR6
BGE	CM03S	EXTSFR7
BGT	CM04S	EXTSFR8
BH	CM05S	EXTSFR9
BL	CM06	EXTSFR10
BLE	CM20	EXTSFR11
BLT	CMP	EXTSFR12
BN	CMPBKC	EXTSFR13
BNC	CMPBKE	EXTSFR14
BNE	CMPBKNC	EXTSFR15
BNH	CMPBKNE	FCC
BNL	CMPMC	H
BNV	CMPME	HL
BNZ	CMPMNC	IE
BP	CMPMNE	IFO
BPE	CMPW	IFOH
BPO	CSE0	IFOL
BR	CSE0H	IF1
BRG	CSE0L	IF1L
BRGM	CSE1	INC
BRK	CSE1L	INCW
BRKCS	CSIM	INTMO
BT	CVTBW	ISMO

ISM0H	POP	RORC
ISM0L	POPU	RP0
ISM1	PRDC	RP1
ISM1L	PRSL	RP2
ISPR	PSW	RP3
L	PSWH	RP4
LT	PSWL	RP5
MK1	PUSH	RP6
MK0H	PUSHU	RP7
MK0L	PWC	RSS
MK1	PWMB	RTPC
MK1L	PWMC	RXB
MM	R0	S
MOV	R1	SBIC
MOV1	R2	SEL
MOVBK	R3	SET1
MOVM	R4	SHL
MOVW	R5	SHLW
MULU	R6	SHR
MULUW	R7	SHRW
MULW	R8	SIO
NOP	R9	SP
NOT1	R10	STBC
OR	R11	SUB
OR1	R12	SUBC
ORG	R13	SUBW
P0	R14	SWRS
P2	R15	TMO
P3	RB0	TM2
P4	RB1	TMC0
P5	RB2	TMC1
P7	RB3	TUM
P8	RB4	TOUT
P9	RB5	TXS
PB0	RB6	UF
PB0H	RB7	UP
PB0L	RBS0	VP
PB1	RBS1	WDM
PB1L	RBS2	X
PC	RET	XCH
PM0	RETB	XCHBK
PM3	RETCS	XCHM
PM5	RETCS	XCHW
PM8	RETI	XOR
PM9	ROL	XOR1
PMC3	ROL4	Z
PMC8	ROLC	
P0L	ROR	
P0H	ROR4	

10.4 逆アセンブラ仕様

逆アセンブラ仕様は、DAS、TRDI コマンドなど、逆アSEMBル表示を行うコマンドにおいて適用されます。

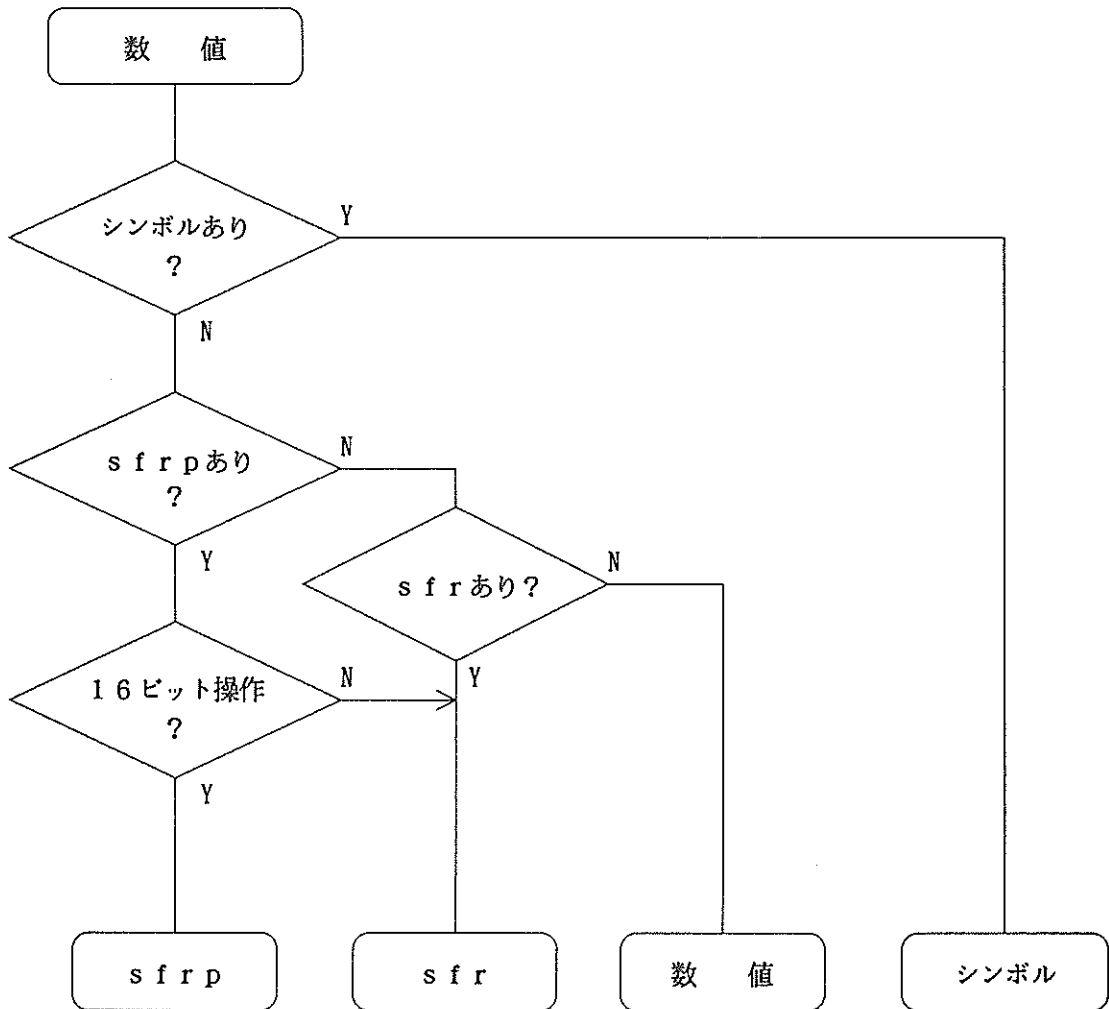
逆アセンブラ仕様は、次の7種類の項目で規定されています。

- ① オペランドの数値表示
- ② オペランドのシンボル表示
- ③ レーベル行の表示
- ④ ブランチ命令の表示
- ⑤ sfr、saddr 操作命令チェック
- ⑥ MOV STBC, #byte / MOV WDM, #byte 命令のチェック
- ⑦ エラー表示

(1) オペランドの数値表示

数値に対して登録済みシンボル、s f r、s f r pが対応する場合、シンボル名で表示します。

次に数値のシンボルへの変換フローを示します。



数値に対応する登録済みシンボル、s f r、s f r pが存在しない場合、16進数で数値を表示して基数（H）を付加します。

数値の先頭文字は必ず数字（0 - 9）で始まります。

(2) オペランドのシンボル表示

シンボルはモジュール名を省略して表示します。

(3) レーベル行の表示

カレント・ロケーションに対応するシンボルがある場合、レーベル行としてシンボルを表示します。

例 レーベル行の表示

注

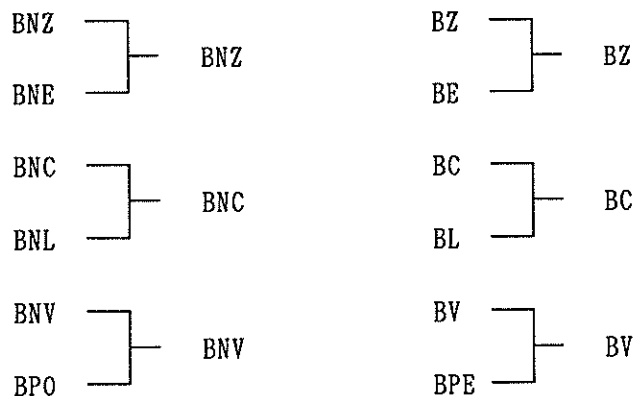
module¥local symbol : ←ローカル・シンボルの場合、: を付加します。
 PUBLIC¥public symbol :: ←パブリック・シンボルの場合、:: を付加します。

注 ¥は、IBM PCシリーズをホスト・マシンとして使用した場合には \ となります。

(4) ブランチ命令の表示

ブランチ系の命令では、同一のオブジェクト・コードに対して2種類のニモニックが割り当てられている場合があります。

このような場合、次のような一般的によく使われるニモニックを表示します。



(5) `sfr`、`saddr` 操作命令のチェック

`sfr`、`saddr` 空間に対する操作命令をチェックし、不正な動作となる場合にはエラーを表示します。

次に示す動作を行った場合、エラーが表示されます。

(a) `sfr` の場合

- 存在しない `sfr`、`sfrp` に対するアクセス
- R/O の `sfr` に対する書き込み動作
- W/O の `sfr` に対する読み出し動作
- `sfrp` 以外の空間に対する 16 ビット操作

(b) `saddr` の場合

- 奇数アドレスに対する 16 ビット操作

(6) `MOV STBC`、`#byte` または、`MOV WDM`、`#byte` 命令のチェック

`MOV STBC`、`#byte` または、`MOV WDM`、`#byte` 命令において、オペランドのバイトのコンプリメントをとったものと、`byte` が一致しなかった場合、エラーを表示します。
このような場合、ニモニックのオペランドにはバイトが表示されます。

`STBC`、`WDM` レジスタに対する `MOV sfr`、`#byte` 命令となる場合、エラーを表示します。

`STBC`、`WDM` レジスタに対しては、対象デバイス制御命令以外ではアクセスできません。

(7) エラー表示

エラー・メッセージには、次の2種類があります。

(a) ???

逆アセンブルできない場合、ニモニックの所に???が表示されます。

2バイト以上の長さを持つ命令コードで、2バイト目以降をデコードしてエラーであった場合、1バイト目をエラーとして表示します。

2バイト目は、次の命令の1バイト目として逆アセンブルします。

例 逆アセンブルできない場合

ADDR	OBJECT	MNEMONIC	
		注	
		PUBLICYSTART::	
0100	0B FC 00 FE	MOVW SP, #STACK	
0104	01	???	←逆アセンブルできません
0105	64 00 10	MOVW RP2, #WORK1	
0108	25	???	←逆アセンブルできません
0109	59	MOV [HL+], A	

注 ¥ は IBM PC シリーズをホストとして使用した場合には、 \ となります。

(b) ?

逆アセンブルはできますが正しい動作を望めない場合、逆アセンブルしたニモニックに?を付加します。

例 正しい動作が望めない場合

ADDR	OBJECT	MNEMONIC	
0105	0C 21 34 12	? MOVW 0FE21H, #1234H	←奇数の saddr に16ビット操作をしています
0113	3A 02 20	? MOV P2, #20H	←R/O の sfr に書き込み動作をしています
0120	10 43	? MOV A, PMCS	←W/O の sfr より読み込み動作をしています
0127	2C 00 FF	? BR !OFF00H	←sfr 空間へブランチしていません

第11章 PROMプログラマ (PG-1500、2000) の操作方法

本章では、IE-78327-RとPROMプログラマ (PG-1500、2000) を接続した場合のPROMプログラマ遠隔操作方法について説明しています。

IE-78327-RとPROMプログラマの接続方法については、ハードウェア編 第6章 周辺装置の接続 を参照してください。

PG-1500、2000の詳しい操作方法については、それぞれのユーザーズ・マニュアルを参照してください。

凡例

- _____ : 下線部分は、キーボードからの入力を表します。
- ☐ : リターン・キー (CR (ODH)) からの入力を表します。
- <BSC> : エスケープ・キーの入力を表します。
- ^ : コントロール・キーを押しながら、^ の右側の文字を入力することを表します。
- このマニュアルで用いられている画面表示例、入力例はPC-9800シリーズをホスト・マシンとして使用した場合のものです。

11.1 PROMプログラマの遠隔操作の開始と終了

(1) 遠隔操作の開始方法

PROMプログラマ制御コマンド (PGMコマンド) を入力すると、PROMプログラマのプロンプトが表示されます。

プロンプトが出力されない場合は、接続方法などの確認を行ってください。

例1. PG-1500の遠隔操作の開始

```
brk:0>PGM [Q]
Beginning of PGM mode ← PROMプログラマの遠隔操作開始メッセージ
PG> ← PG-1500から出力されたプロンプト
```

2. PG-2000の遠隔操作の開始

```
brk:0>PGM [Q]
Beginning of PGM mode ← PROMプログラマの遠隔操作開始メッセージ
* ← PG-2000から出力されたプロンプト
```

(2) 遠隔操作の終了方法

キーボードから[^]Zキーを入力すると遠隔操作が終了します。

例1. PG-1500の遠隔操作の終了

PG> <u>[^]Z</u>	← [^] Zを入力してください
Exit PGM mode (Y/N) <u>Y</u> <u>␣</u>	← PROMプログラマ遠隔操作の 終了確認メッセージ
Termination of PGM mode	← PROMプログラマ遠隔操作の 終了メッセージ
brk:0>	

2. PG-2000の遠隔操作の終了

* <u>[^]Z</u>	← [^] Zを入力してください
Exit PGM mode (Y/N) <u>Y</u> <u>␣</u>	← PROMプログラマ遠隔操作の 終了確認メッセージ
Termination of PGM mode	← PROMプログラマ遠隔操作の 終了メッセージ
brk:0>	

11.2 PROMプログラマのコマンド

PROMプログラマ (PG-1500、2000) のコマンド、オブジェクト・コードのアップ・ロード、ダウン・ロード方法について説明します。

(1) PROMプログラマのコマンド一覧

PROMプログラマ (PG-1500、2000) で使用するコマンドの一覧を次に示します。

(a) PG-1500のコマンド一覧

表11-1 PG-1500のコマンド一覧

コマンド	形 式	機 能
RR	PG>RR P_S_ADR, R_E_ADR, PG_S_ADR, CONV ☺	PROM内容のリード
RS	PG>RS sub ☺ sub=C/R/A	デバイスの選択
RV	PG>RV P_S_ADR, R_E_ADR, PG_S_ADR, CONV ☺	PROMとPG内 メモリとの比較
RW	PG>RW PG_S_ADR, PG_E_ADR, R_S_ADR, CONV ☺	PROMへの書き込み
RZ	PG>RZ ☺	PROM消去チェック
MC	PG>MC PG_S_ADR ☺	PG内メモリ内容の変更
MD	PG>MD PG_S_ADR, PG_E_ADR ☺	PG内メモリ内容の表示
MF	PG>MF PG_S_ADR, PG_E_ADR, INT_DATA ☺	PG内メモリの初期化
LI	PG>LI ☺	IE-78327-Rから PGへのデータ転送
SI	PG>SI PG_S_ADR, PG_E_ADR ☺	PGからIE-78327 -Rへのデータ転送
??	PG>?? ☺	コマンド・ヘルプ

詳細は、PG-1500のユーザーズ・マニュアルを参照してください。

- 備考
- ・ PG_S_ADR : PGスタート・アドレスを表します
 - ・ PG_E_ADR : PGエンド・アドレスを表します
 - ・ R_S_ADR : PROMスタート・アドレスを表します
 - ・ R_E_ADR : PROMエンド・アドレスを表します
 - ・ CONV : アドレス分割指定を表します
 - ・ INT_DATA : 初期化データ表します

(b) PG-2000のコマンド一覧

表11-2 PG-2000コマンド一覧 (1/2)

コマンド	形式	コマンドの機能
A	* <u>A</u> s, e, r <input type="checkbox"/>	パラメータの設定
E	* <u>E</u> r <input type="checkbox"/>	<p>PGのバッファのデータを変更 (形式は次のとおりです)</p> <p>*<u>E</u>r <input type="checkbox"/></p> <p>r <u>XX- YY- XX- YY- XX- <input type="checkbox"/></u></p> ', with the label '入力データ' (input data) to the right."/> <p>データ入力形式</p> <ul style="list-style-type: none"> ・ データを入力する (16進以外を入力するとその時点で '?' を表示) ・ スペース・キーの入力 (データ変更なし)
F	* <u>F</u> r, re, d <input type="checkbox"/>	PGのバッファをdで初期化
I	* <u>I</u> <input type="checkbox"/>	インテリジェント・モード (ディスプレイヘエコー・バックするモード) へ移行
O	* <u>O</u> r, re <input type="checkbox"/>	PGのバッファの内容を表示 表示形式 → r 00 00 00 00 00 00 00
R	* <u>R</u> r, re <input type="checkbox"/>	PROMの内容をPGのバッファへ転送
S	* <u>S</u> <input type="checkbox"/>	PROMの選択
T	* <u>T</u> <input type="checkbox"/>	トランジェント・モード (ディスプレイヘエコー・バックしないモード) へ移行

表11-2 PG-2000コマンド一覧 (2/2)

コマンド	形式	コマンドの機能
V	*Vs,e,r [d]	PROMの内容とPGのバッファの内容を比較する → データがちがうときは'?'が表示される
W	*Ws,e,r [d]	PGのバッファの内容をPROMに書き込む → 書き込みエラー時は'?'が表示される
Y	*Y [d]	現在設定されているパラメータを表示
Z	*Z [d]	PROMにデータが書き込まれているかどうかをチェック → データが書き込まれている場合にのみ'?'が表示される (データが無い場合は、何も表示されない)
L	*Lbias [d] partition= i,ie [d]	IE-78327-R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 番地まで転送する
P	*Pr,re [d] Bias =i [d]	PGのバッファの r 番地から re 番地までのデータを、 IE-78327-Rのマッピングされている r+i 番地 から re+1 番地まで転送する

詳細は、PG-2000のユーザーズ・マニュアルを参照してください。

- 備考
- ・ 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
 - ・ i : IE-78327-R開始番地
 - ・ ie : IE-78327-R終了番地
 - ・ d : データ

注意 エラー発生条件

- s > e
- r > re
- i > ie
- 16進以外の入力

(2) オブジェクト・コードのアップ・ロード、ダウン・ロード方法

(a) PG-1500とオブジェクト・コードのアップ・ロード (LIコマンド)

IE-78327-Rのマッピングされたメモリ空間の内容をPG-1500のバッファへ転送します。

転送範囲を省略した場合は、マッピングされている全メモリ空間を転送します。転送を中止する場合には、<ESC>を入力します。

例 オブジェクト・コードのアップ・ロード

```
PG>LI         
           注1 注2
Partition = YYYY,ZZZZ
PG>
```

- 注1. YYYY には、IE-78327-Rメモリの転送開始アドレスを入力してください。
2. ZZZZ には、IE-78327-Rメモリの転送終了アドレスを入力してください。

(b) PG-1500とオブジェクト・コードのダウン・ロード (SIコマンド)

PG-1500のバッファの内容をIE-78327-Rのマッピングされたメモリ空間に転送します。

IE-78327-Rのロード・バイアスは省略できません。途中で転送を中止する場合には、<ESC>を入力します。

例 オブジェクト・コードのダウン・ロード

```
           注1 注2
PG>SI XXXX,YYYY
           注3
Bias = ZZZZ
      complete
PG>
```

- 注1. XXXX には、PGのバッファの転送開始アドレスを入力してください。
2. YYYY には、PGのバッファの転送終了アドレスを入力してください。
3. ZZZZ には、IE-78327-Rのロード・バイアスを入力してください。

(c) PG-2000とオブジェクト・コードのアップ・ロード (Lコマンド)

IE-78327-Rのマッピングされたメモリ空間の内容をPG-2000のバッファに転送します。

転送範囲を省略した場合は、マッピングされている全メモリ空間を転送します。
途中で転送を中止する場合には、〈ESC〉を入力します。

例 オブジェクト・コードのアップ・ロード

注1
*LXXX □
注2 注3
Partition = YYYYY, ZZZZ □
*

- 注1. XXXX には、PGのロード・バイアスを入力してください。
- 2. YYYYY には、IE-78327-Rのメモリの転送開始アドレスを入力してください。
- 3. ZZZZ には、IE-78327-Rのメモリの転送終了アドレスを入力してください。

(d) PG-2000とオブジェクト・コードのダウン・ロード (Pコマンド)

PG-2000のバッファの内容をIE-78327-Rのマッピングされたメモリ空間に転送します。

IE-78327-Rのロード・バイアスは省略できません。
途中で転送を中止する場合には、〈ESC〉を入力します。

例 オブジェクト・コードのダウン・ロード

注1 注2
*PXXXX, YYYYY □
注3
Bias = ZZZZ □
complete
*

- 注1. XXXX には、PGのバッファの転送開始アドレスを入力してください。
- 2. YYYYY には、PGのバッファの転送終了アドレスを入力してください。
- 3. ZZZZ には、IE-78327-Rのロード・バイアスを入力してください。

第12章 使用上の注意事項

この章では、IE-78327-Rおよび開発対象製品である μ PD78320シリーズ、 μ PD78327シリーズのデバイスの使用上の注意事項について説明します。

応用製品の開発前には、必ずこの章をお読みください。

1 2 . 1 I E - 7 8 3 2 7 - R に関する注意事項

このマニュアルの各章で述べた I E - 7 8 3 2 7 - R の使用に関する注意事項をまとめて記述します。

(1) 「第 1 章 概説」に関する注意事項

1. ホスト・マシン (P C - 9 8 0 0 シリーズ) の内蔵メモリ容量は 6 4 0 K 以上必要です。 (p. 7)
2. コントロール・プログラムの 8 インチ F D (2 D) による供給は廃止致しました。なお、現在 8 インチ F D をご使用のユーザへは、バージョン・アップの際に 5 インチ F D で供給致します。ご了承ください。 (p. 8)

(2) 「第 2 章 I E - 7 8 3 2 7 - R の起動方法」に関する注意事項

1. デバイス・ドライバのインストールを行ったあとは、必ずホスト・マシンをリセットして、MS - D O S または P C D O S を再起動してください。 (p. 10)
2. コントロール・プログラムの起動後、ターゲット・システムの電源要求メッセージの入力において、N と入力すると、同じメッセージを出力し続け、次のステップに進みません。 (p. 14)

(3) 「第 3 章 機能概要」に関する注意事項

1. リアルタイム・トレース機能で、内部 R A M (0 F E 0 0 H - 0 F E F F H) を、ショート・ダイレクト・アドレッシング以外のアドレッシングでリードした場合、トレース・データの値は不定となります。ただし、命令 (ターゲット・プログラム) の実行は正常に行われます。 (p. 67、p. 344、p. 352)
2. トリガ信号外部出力機能の注意事項
 - ① I E - 7 8 3 2 7 - R は、外部から入力するデータをトレース、イベント検出するため 8 本の外部センス・クリップ (No. 1 - 8) を持っています。8 本の外部センス・クリップは通常入力ラインとなっていますが、O U T O N コマンドにより、N o . 1 のみをトリガ信号出力にすることができます。 (p. 76)

- ② O U T O N コマンドにより、外部センス・クリップの N o . 1 をトリガ信号出力とする場合は、この外部センス・クリップをターゲット・システムの信号を出力するラインに接続しないでください。ターゲット・システムや I E - 7 8 3 2 7 - R を破損する原因になります。(p.76)
- 3. メモリまたはレジスタの操作実行後、エミュレーション C P U は実行を再開します。しかし、コマンド実行によりプログラムの実行を一時中断したあとの動作となっていますので注意してください(リアルタイム実行にはなりません)。(p.93)
- 4. ノンリアルタイム実行機能(R U N T コマンド)で H A L T または S T O P 命令を実行してもエミュレーション C P U はスタンバイ・モードにはなりません。(p.94)

(4) 「第 8 章 コマンドの詳細」に関する注意事項

- 1. トレース (trc:0>) およびエミュレーション (emu:0>) 時にメモリの内容を操作するコマンドを入力すると、エミュレーション C P U の実行は一時中断されます。
- 2. P G M コマンド使用時の注意事項 (p.280)
 - ① チャンネル 2 に N E C 製 P R O M プログラマ (P G - 1 5 0 0 、 2 0 0 0) 以外の外部装置を接続して、 I E - 7 8 3 2 7 - R を端末として使用する場合は、次の制御キャラクタが使用できません。

^ A ^ B ^ C ^ D ^ E ^ F ^ H ^ I ^ J
 ^ K ^ L ^ M ^ N ^ Q ^ S ^ W ^ Z

上記の制御キャラクタを使用する場合、P G M C ㊸と入力して制御キャラクタの使用制限解除や、一部の制御キャラクタの変更を行ってください

- ② 制御キャラクタの変更は対話形式で行います。特定のキャラクタとして、次の 1 6 種類のキャラクタを使用できます。

^ A ^ B ^ E ^ F ^ B ^ N ^ O ^ P
 ^ R ^ T ^ U ^ V ^ W ^ X ^ Y ^ Z

同一のキャラクタは重複して設定できません。

D E L キーまたは ^ H を入力すると初期値に戻ります。

<ESC> 入力による変更中断の場合、それまでに変更したキャラクタを無効にします。

- ③ 制御キャラクタの使用制限解除の変更を行うと、自動的に端末モードとなります。

3. PSD コマンド使用時の注意事項 (p. 285)
 - ① PSA コマンドでサンプル・アドレスの設定終了後、ライト動作が一度行われるまでは、サンプル・データは不定値となります。
 - ② サンプル・データは再リアルタイム実行によりクリアされます。
 - ③ シングル・ステップ実行中またはプロシージャ実行中では、内部 RAM データ・サンプリングは停止します。
4. レジスタの操作（読み出し、書き込み）を、トレース (trc:0>) またはエミュレーション (emu:0>) 時に行うと、エミュレーション CPU の実行は一時中断されます。 (p. 290)
5. レジスタ、メモリ、SFR の内容の操作を行うコマンドをトレース (trc:0>) およびエミュレーション (emu:0>) 時に入力すると、エミュレーション CPU の実行は一時中断されます。 (p. 298)
6. ステップ実行中の割り込みはすべて保留されます。 (p. 305)
7. プロシージャ実行中の割り込みはすべて保留されます。 (p. 309)
8. emu:0> (プロンプト) において SFR の操作を行う場合、SFR の読み出し、書き込み時に実行が一時中断されます。 (p. 316)
9. TRD F コマンド使用時の注意事項 (p. 345)
 - ① 先頭フレームのタイム・タグは不定値となるため “xxx” と表示されます。
 - ② タイム・タグ値が、255 システム・クロック以上の場合、すべて “xxx” と表示されます。
10. TRD I コマンド使用時の注意事項 (p. 353)
 - ① 先頭フレームのタイム・タグは不定値となるため “xxx” と表示されます。
 - ② タイム・タグ値が、255 システム・クロック以上の場合、すべて “xxx” と表示されます。
 - ③ 命令に対するリードまたはライト・アクセスを表示するので、実際のエミュレーション CPU の動作とトレース・データの表示の順番が異なる場合があります。

- ④ クオリファイ・トレース・モード (TRM TRX) でトレースされたデータは、インストラクション・トレース表示 (TRD I) を行っても、フレーム表示となります
- ⑤ 区間トレース・モード (TRM SEC) でトレースされたデータの先頭フレームと最終フレームは、インストラクション・トレース表示 (TRD I) を行うと不定値となる場合があります。

(5) 「第 1 1 章 PROM プログラム (PG-1500、2000) の操作方法」
に関する注意事項

PROM プログラムのコマンド入力時の、エラー発生要因 (P. 438)

s > e
r > r e
i > i e
16 進以外の入力

1 2. 2 開発対象製品（ μ PD78320シリーズ、 μ PD78327シリーズ）に関する注意事項

1 2. 2. 1 “内部制御、外部制御機能”に関する注意事項

(1) STOPモードをNMIで解除する場合

1. NMIの入力によりSTOPモードが解除されると、STOPモードに設定する命令(MOV STBC #byte)に続く1命令を実行してからNMIの割り込み処理ルーチンに分岐しますので、STOPモードに設定する命令の次の命令はNOPにしてください。
2. STOPモードに設定する命令の実行中に割り込み要求が発生すると、STOPモード解除後にその割り込み処理を実行して、そのあとでNMIの割り込み処理ルーチンに分岐します。したがって、STOPモードに設定する命令の前に、すべてのマスカブル割り込みをマスクしてください。
なお、発生した割り込みの処理形態によって、STOPモード解除後の動作が異なります。
 - ① ベクタ割り込み処理に設定したマスカブル割り込み要求が発生した場合、マスカブル割り込み処理ルーチンに分岐し、1命令実行したあとNMIの割り込み処理ルーチンに分岐します。
 - ② マクロ・サービス処理に設定したマスカブル割り込み要求が発生した場合、マクロ・サービスを実行したあとでNMIの割り込み処理ルーチンに分岐します。
なお、マクロ・サービスが終了するまでNMIの処理は保留されます。
3. STOPモードに設定する命令を実行中に、ノンマスカブル割り込みのウォッチドッグ・タイマ割り込み要求が発生した場合、NMI割り込みの優先順位が高くてもウォッチドッグ・タイマ割り込み処理ルーチンに分岐し、1命令実行後にNMIの割り込み処理ルーチンに分岐しますのでウォッチドッグ・タイマ割り込み処理ルーチンの先頭にNOPを入れてください。
また、このとき割り込みのネスティングは2レベル使用することを考慮してスタック・エリアを設定してください。

(2) 高速フェッチと通常フェッチが同一メモリ・マップに混在する場合

1. 通常フェッチ・サイクル・モード領域から高速フェッチ・サイクル・モード領域へ連続して命令を実行した場合、通常フェッチ・サイクル領域の最後の1バイトはフェッチされないので、分岐命令（BR、CALL命令）を用いて移動するようにしてください。
2. 高速フェッチ・サイクル・モード領域から通常フェッチ・サイクルモードへ連続して命令を実行した場合、2つのモードの境界のバイト前の命令コードがフェッチされません。したがって、異なるフェッチ・サイクル・モード領域の境界の2バイト前には命令コードを置かないでください。
3. 命令実行中の領域を通常フェッチ・サイクル・モードから高速フェッチ・サイクル・モードに設定を変更するため、フェッチ・サイクル制御レジスタ（FCC）へ書き込みを行う場合は次の注意事項を守ってください。
 - ① FCCレジスタへの書き込みは下記の命令を用いて、メモリ・アクセスによる書き込みを行わないようにしてください。
 - ② リセット解除からFCCレジスタの設定までの間、メモリ・アクセスによる特殊機能レジスタ（SFR）の操作は基本的にはしないでください。メモリ・アクセスによるSFRの操作を行う場合は、FCCレジスタの設定を行う直前に、SFRアクセス命令またはショート・ダイレクト・メモリ・アクセス命令によるダミーのSFR操作命令を挿入してください。

12.2.2 “命令セット”に関する注意事項

(1) “ADDC/SUBC sfr,#byte 命令”使用時の注意事項

第1オペランドで、次の特殊機能レジスタを指定した場合、演算結果が不正になりますので第1オペランドには記述しないでください。

●対象となる特殊機能レジスタ：P4、P5、PM5、MM、外部SFR

(2) “ADDC/SUBC saddr,#byte 命令”使用時の注意事項

第1オペランドにショート・ダイレクト・メモリとして、次の特殊機能レジスタを指定した場合、演算結果が不正になりますので第1オペランドには記述しないでください。

●対象となる特殊機能レジスタ：P4、P5

(3) “ADDC/SUBC saddr, saddr 命令” 使用時の注意事項

第 1 オペランドにショート・ダイレクト・メモリとして、次の特殊機能レジスタを指定した場合、演算結果が不正になりますので第 1 オペランドには記述しないでください。

- 対象となる特殊機能レジスタ：P 4、P 5

アンケート記入のお願い

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

[ドキュメント名] IE-78327-R ユーザーズ・マニュアル ソフトウェア編

(EEU-720B (第3版) , November 1991 P)

[お名前など] (さしつかえのない範囲で)

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

1. ご評価(各欄に○をご記入ください)

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

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

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

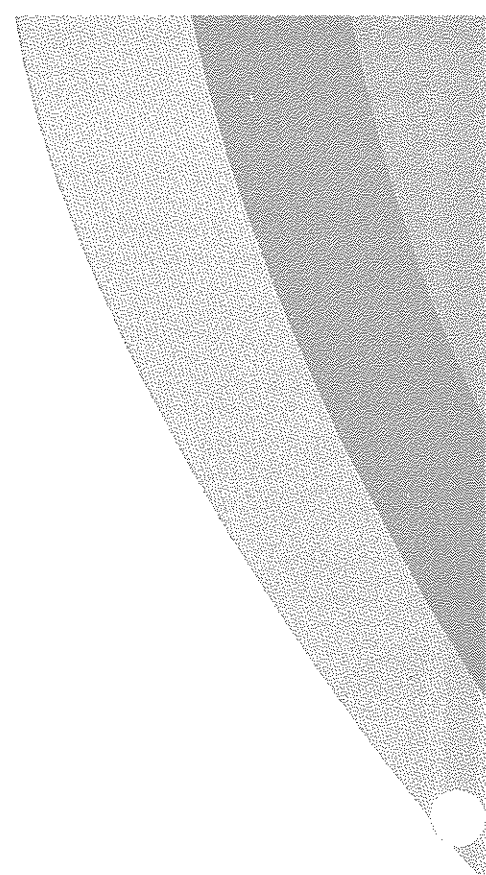
4. ご意見、ご要望

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

ご協力ありがとうございました。

下記あてにFAXで送信いただくか、最寄りの販売員にコピーをお渡ししてください。

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



— NEC 日本電気株式会社 —

本社	〒108-01 東京都港区芝五丁目7番1号(日本電気本社ビル)
半導体第一、第二販売事業部	〒108-01 東京都港区芝五丁目7番1号(日本電気本社ビル) 東京(03)3454-1111
関西支社半導体販売部	〒540 大阪市中央区城見一丁目4番24号(日本電気関西ビル) 大阪(06)945-3178 大阪(06)945-3200
中部支社半導体販売部	〒460 名古屋市中区栄四丁目14番5号(松下中日ビル) 名古屋(052)242-2755

北海道支社	札幌(011)231-0161	立川支社	立川(0425)26-0911
東北支社	仙台(022)261-5511	千葉支社	千葉(0472)27-5441
岩手支社	盛岡(0196)51-4344	茨城支社	水戸(0559)63-4455
山形支社	山形(0236)23-5511	群馬支社	高崎(053)452-2711
都府支社	いわき(0249)23-5511	栃木支社	宇都宮(0762)23-1621
新潟支社	新潟(0258)36-2155	富山支社	富山(0776)22-1866
長岡支社	長岡(0292)26-1717	山梨支社	山梨(0764)31-8461
水戸支社	水戸(045)324-5511	長野支社	長野(075)221-8511
神奈川支社	横浜(045)324-5511	東京支社	東京(078)332-3311
群馬支社	高崎(0273)26-1255	神奈川支社	横浜(082)242-5504
茨城支社	水戸(0276)46-4011	山梨支社	山梨(0857)27-5311
栃木支社	宇都宮(0286)21-2281	長野支社	長野(0862)25-4455
群馬支社	高崎(0285)24-5011	東京支社	東京(0878)36-1200
山梨支社	山梨(0262)35-1444	神奈川支社	横浜(0897)32-5001
長野支社	長野(0263)35-1666	山梨支社	山梨(0899)45-4111
新潟支社	新潟(0266)53-5350	長野支社	長野(092)271-7700
富山支社	富山(0552)24-4141	東京支社	東京(093)541-2887
石川支社	金沢(048)641-1411	千葉支社	千葉(093)541-2887

(技術お問い合わせ先)

半導体応用技術本部 第一応用システム技術部	〒108-01 東京都港区芝五丁目7番1号(日本電気本社ビル)	東京(03)3798-6105
半導体応用技術本部 第二応用システム技術部	〒540 大阪市中央区城見一丁目4番24号(日本電気関西ビル)	大阪(06)945-3383
半導体応用技術本部 第三応用システム技術部	〒460 名古屋市中区栄四丁目14番5号(松下中日ビル)	名古屋(052)242-2762
半導体応用技術本部 マイクロコンピュータ技術部	〒210 川崎市幸区塚越三丁目484番地	川崎(044)548-8890

インフォメーションセンター
FAX(044)548-7900