

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日

ルネサスエレクトロニクス株式会社

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

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

ご注意書き

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

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

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

お客様各位

資料中の「日立製作所」、「日立XX」等名称の株式会社ルネサス テクノロジへの変更について

2003年4月1日を以って三菱電機株式会社及び株式会社日立製作所のマイコン、ロジック、アナログ、ディスクリート半導体、及びDRAMを除くメモリ(フラッシュメモリ・SRAM等)を含む半導体事業は株式会社ルネサス テクノロジに承継されました。従いまして、本資料中には「日立製作所」、「株式会社日立製作所」、「日立半導体」、「日立XX」といった表記が残っておりますが、これらの表記は全て「株式会社ルネサス テクノロジ」に変更されておりますのでご理解の程お願い致します。尚、会社商標・ロゴ・コーポレートステートメント以外の内容については一切変更しておりませんので資料としての内容更新ではありません。

ルネサステクノロジ ホームページ (<http://www.renesas.com>)

2003年4月1日
株式会社ルネサス テクノロジ
カスタマサポート部

E6000 H8/3112、H8/3113、
H8/3150、H8/3160、AE-3シリーズ
エミュレータ

HS3160EPI60H ユーザーズマニュアル

ルネサスマイクロコンピュータ開発環境システム

ご注意

1. 本書に記載の製品及び技術のうち「外国為替及び外国貿易法」に基づき安全保障貿易管理関連貨物・技術に該当するものを輸出する場合、または国外に持ち出す場合は日本国政府の許可が必要です。
2. 本書に記載された情報の使用に際して、弊社もしくは第三者の特許権、著作権、商標権、その他の知的所有権等の権利に対する保証または実施権の許諾を行うものではありません。また本書に記載された情報を使用した事により第三者の知的所有権等の権利に関わる問題が生じた場合、弊社はその責を負いませんので予めご了承ください。
3. 製品及び製品仕様は予告無く変更する場合がありますので、最終的な設計、ご購入、ご使用に際しましては、事前に最新の製品規格または仕様書をお求めになりご確認ください。
4. 弊社は品質・信頼性の向上に努めておりますが、宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性が要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途にご使用をお考えのお客様は、事前に弊社営業担当迄ご相談をお願い致します。
5. 設計に際しては、特に最大定格、動作電源電圧範囲、放熱特性、実装条件及びその他諸条件につきましては、弊社保証範囲内でご使用いただきますようお願い致します。
保証値を越えてご使用された場合の故障及び事故につきましては、弊社はその責を負いません。
また保証値内のご使用であっても半導体製品について通常予測される故障発生率、故障モードをご考慮の上、弊社製品の動作が原因でご使用機器が人身事故、火災事故、その他の拡大損害を生じないようにフェールセーフ等のシステム上の対策を講じて頂きますようお願い致します。
6. 本製品は耐放射線設計をしておりません。
7. 本書の一部または全部を弊社の文書による承認なしに転載または複製することを堅くお断り致します。
8. 本書をはじめ弊社半導体についてのお問い合わせ、ご相談は弊社営業担当迄お願い致します。

重要事項

- ・当エミュレータをご使用になる前に、必ずユーザーズマニュアルをよく読んで理解してください。
- ・ユーザーズマニュアルは、必ず保管し、使用上不明な点がある場合は再読してください。

エミュレータとは：

ここでいうエミュレータとは、株式会社日立製作所（以下、「日立」という。）が製作した次の製品を指します。

（１）E6000 エミュレータ本体、（２）ユーザシステムインタフェースケーブル、（３）PC インタフェースボード

お客様のユーザシステム及びホストコンピュータは含みません。

エミュレータの使用目的：

当エミュレータは、日立マイクロコンピュータ H8/3112, H8/3113, H8/3150, H8/3160, AE-3 シリーズ MCU（以下、MCU と略します）を使用したシステムの開発を支援する装置です。ソフトウェアとハードウェアの両面から、システム開発を支援します。

この使用目的に従って、当エミュレータを正しく使用してください。この目的以外に当エミュレータを使用することを堅くお断りします。

使用制限：

当エミュレータは、開発支援用として開発したものです。したがって、機器組み込み用として使用しないでください。また、以下に示す開発用途に対しても使用しないでください。

- 1 ライフサポート関連の医療機器用（人命にかかわる装置用）
- 2 原子力開発機器用
- 3 航空機開発機器用
- 4 宇宙開発機器用

このような目的で当エミュレータの採用をお考えのお客様は、当社営業窓口へ是非ご連絡頂きますようお願い致します。

製品の変更について：

日立は、当エミュレータのデザイン、機能および性能を絶えず改良する方針をとっています。したがって、予告なく仕様、デザイン、およびユーザーズマニュアルを変更することがあります。

エミュレータを使用する人は：

当エミュレータは、ユーザーズマニュアルをよく読み、理解した人のみが使用してください。

特に、当エミュレータを初めて使用する人は、当エミュレータをよく理解し、使い慣れている人から指導を受けることをおすすめします。

保証の範囲：

日立は、お客様が製品をご購入された日から1年間は、無償で故障品を修理、または交換いたします。

- ただし、（１）製品の誤用、濫用、またはその他異常な条件下での使用
- （２）日立以外の者による改造、修理、保守、またはその他の行為
- （３）ユーザシステムの内容、または使用
- （４）火災、地震、またはその他の事故

により、故障が生じた場合はご購入日から1年以内でも有償で修理、または交換を行います。また、日本国内で購入され、かつ、日本国内で使用されるものに限りです。

その他の重要事項：

- 1 本資料に記載された情報、製品または回路の使用に起因する損害または特許権その他権利の侵害に関しては、日立は一切その責任を負いません。
- 2 本資料によって第三者または日立の特許権その他権利の実施権を許諾するものではありません。

著作権所有：

このユーザズマニュアルおよび当エミュレータは著作権で保護されており、すべての権利は日立に帰属しています。このユーザズマニュアルの一部であろうと全部であろうといかなる箇所も、日立の書面による事前の承諾なしに、複写、複製、転載することはできません。

図について：

このユーザズマニュアルの一部の図は、実物と異っていることがあります。

予測できる危険の限界：

日立は、潜在的な危険が存在するおそれのあるすべての起こりうる諸状況や誤使用を予見できません。したがって、このユーザズマニュアルと当エミュレータに貼付されている警告がすべてではありません。お客様の責任で、当エミュレータを正しく安全に使用してください。

安全事項

- ・当エミュレータをご使用になる前に、必ずユーザーズマニュアルをよく読んで理解してください。
- ・ユーザーズマニュアルは、必ず保管し、使用上不明な点がある場合は再読してください。

シグナル・ワードの定義



これは、安全警告記号です。潜在的に、人に危害を与える危険に対し注意を喚起するために用います。起こり得る危害又は死を回避するためにこの記号の後に続くすべての安全メッセージに従ってください。



危険

危険は、回避しないと、死亡または重傷を招く差し迫った危険な状況を示します。ただし、本製品では該当するものではありません。



警告

警告は、回避しないと、死亡または重傷を招く可能性がある潜在的に危険な状況を示します。



注意

注意は、回避しないと、軽傷または中程度の傷害を招く可能性がある潜在的に危険な状況を示します。



注意

安全警告記号の付かない**注意**は、回避しないと財物傷害を引き起こすことがある潜在的に危険な状況を示します。

注、留意事項は、例外的な条件や注意を操作手順や説明記述の中で、ユーザに伝達する場合に使用しています。

警告

- 1 DC 電源を内蔵していますので、触れる場所によっては感電する可能性があります。感電、火災等の危険防止および品質保証のために、お客様ご自身による修理や改造は行なわないでください。故障の際のアフターサービスにつきましては、日立または日立特約店保守担当にお申し付けください。
- 2 エミュレータまたはユーザシステムのパワーオン時、ユーザシステムインタフェースケーブルを除く全てのケーブル類の抜き差しを行なわないでください。抜き差しを行なった場合、エミュレータとユーザシステムの発煙発火の可能性があります。また、デバッグ中のユーザプログラムの破壊の可能性があります。
- 3 ユーザシステムのパワーオン時、ユーザシステムインタフェースケーブルとユーザシステムの抜き差しを行なわないでください。抜き差しを行なった場合、デバッグ中のユーザプログラムの破壊の可能性があります。
- 4 ユーザインタフェースケーブルの表裏をご確認の上、ユーザシステムへ挿入してください。
- 5 電源給電については電源仕様に従って供給してください。使用する電源ケーブルは製品に添付のものを使用してください。仕様以外の電源電圧を加えないでください。ヒューズが切れ、交換する場合、使用するヒューズは指定のものを使用してください。

まえがき

本書について

本書は、H8/3112,H8/3113,H8/3150,H8/3160,AE-3シリーズマイクロコンピュータ用のE6000エミュレータのセットアップと使用方法を説明します。本書はデバッグプラットフォームのマニュアルです。

「1.はじめに」では、E6000エミュレータの主なエミュレーション機能の概要と、E6000エミュレータの制御ソフトウェアである日立デバッグインタフェース（以降、HDIと呼びます）の機能を簡単に紹介します。

「2.セットアップ」は、E6000エミュレータのセットアップ方法と、HDIに接続する方法について記載します。

「3.ハードウェア」は、E6000エミュレータとユーザシステムの接続方法、およびハードウェア詳細について記載します。

「4.チュートリアル」は、簡単なCプログラムのロードとデバッグの方法を示しながら、E6000エミュレータの主な特徴を紹介します。チュートリアルプログラムはディスクで提供されます。したがって、チュートリアルプログラムを実行することによって、システムの動作を直接理解できます。

「5.E6000 H8/3160 HDIの機能」は、本E6000エミュレータ専用のHDIの特長を記載します。

「6.コマンドライン機能」は、H8/3112,H8/3113,H8/3150,H8/3160シリーズ特有のコマンドライン機能について記載します。

「7.故障解析」は、E6000エミュレータ用テストプログラムによる故障解析の手順について記載します。

想定

本書は、読者にMS-DOS[®]およびWindows[®]用アプリケーションの実行および使用の手順に関する知識があるものと想定して話を進めます。

関連マニュアル

- ・日立デバッグインタフェースユーザズマニュアル
- ・PCインタフェースボード取扱い説明書（本共通ユーザズマニュアルでは、以下のいずれかを指します。）

ISAバスインタフェースボード(HS6000E1101HJ)

PCIバスインタフェースボード取扱い説明書(HS6000E1C01HJ, HS6000E1C02HJ)

PCMCIAインタフェースカード取扱い説明書(HS6000E1P01HJ)

LANアダプタ取扱い説明書(HS6000ELN01HJ)

Windows[®], Windows[®] 95, Windows[®] 98, Windows NT[®] 4.0, Windows[®] 2000およびMS-DOS[®]は米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。

IBM PCは米国IBM社の商標です。

本マニュアルは動作環境をIBM PC 上の英語版Microsoft[®] Windows[®] 98として記述しています。

目次

1	はじめに	1
1.1	デバッグの特長	1
1.1.1	ブレークポイント	1
1.1.2	トレース	1
1.1.3	実行時間測定	1
1.2	イベント検出システム (CES: Complex Event System)	2
1.2.1	イベントチャンネル	2
1.2.2	範囲チャンネル	2
1.2.3	ブレーク	2
1.2.4	イベント間実行時間測定	3
1.3	ハードウェアの特長	3
1.3.1	メモリ	3
1.3.2	エミュレーションクロック	3
1.3.3	外部プローブ	3
1.3.4	使用環境条件	4
1.3.5	外形寸法と質量	4
2	セットアップ	5
2.1	パッケージ内容	5
2.2	Windows® 95, Windows® 98 での PC インタフェースボードのセットアップ	5
2.2.1	PC インタフェースボードのセットアップ	5
2.2.2	CONFIG.SYS の変更	8
2.2.3	SYSTEM.INI の変更	8
2.3	Windows NT® 4.0 での PC インタフェースボードのセットアップ	8
2.4	HDI のインストール	10
2.5	トラブルシューティング	10
2.5.1	接続不良	10
2.5.2	通信不良	10
3	ハードウェア	11
3.1	ユーザシステムへの接続	11
3.1.1	ユーザシステムインタフェースケーブル本体部と E6000 エミュレータの接続	11
3.1.2	ユーザシステムインタフェースケーブル先端部の接続	11

3.2	電源供給	12
3.2.1	AC 電源アダプタ	12
3.2.2	極性	12
3.2.3	電源モニタ回路	12
3.3	ハードウェアインタフェース	12
3.3.1	信号保護	12
3.3.2	ユーザインタフェース回路	13
3.3.3	外部プローブ/トリガ出力	14
3.3.4	電源回路	14
3.4	MCU と E6000 エミュレータの相違点	15
3.4.1	EEPROM	16
3.4.2	WDT	16
3.4.3	セキュリティ	16
4	チュートリアル	17
4.1	はじめに	17
4.2	HDI の起動	17
4.2.1	ターゲットプラットフォームの選択	18
4.3	E6000 エミュレータのセットアップ	20
4.3.1	プラットフォームの構成	20
4.3.2	メモリマッピング	21
4.4	チュートリアルプログラムのダウンロード	23
4.4.1	オブジェクトファイルのダウンロード	23
4.4.2	プログラムリストの表示	24
4.5	ブレークポイントの使い方	25
4.5.1	PC Break の設定	25
4.5.2	プログラムの実行	25
4.5.3	レジスタ内容の参照	27
4.5.4	ブレークポイントの確認	28
4.6	メモリと変数の表示	28
4.6.1	メモリを表示する	28
4.6.2	変数を表示する	29
4.7	プログラムのステップ実行	31
4.7.1	シングルステップ	31
4.7.2	関数全体のステップ実行	33

4.7.3 ローカル変数の表示	33
4.8 イベント検出システム(Complex Event System)の使用方式	35
4.8.1 イベント検出システムによる Event の設定	35
4.9 トレースバッファの使い方	38
4.9.1 トレースバッファの表示	38
4.9.2 トレースフィルタの設定	39
4.10 プログラム実行時間効率の計測	40
4.10.1 計測条件の選択	40
4.10.2 計測結果の表示	42
4.11 スタックトレース機能	43
4.12 セッションの保存	44
4.13 さてつぎは?	44
5 E6000 H8/3160 HDI の機能	45
5.1 Configuration ダイアログボックス	47
5.1.1 ドライバ詳細	48
5.2 System Status ウィンドウ	49
5.3 Breakpoints ウィンドウ	53
5.3.1 Range Detectors	57
5.3.2 Event Detectors	57
5.3.3 Event Sequencing	58
5.4 Trace ウィンドウ	59
5.4.1 Trace Acquisition General	60
5.4.2 Trace Acquisition Stop	61
5.4.3 Trace Acquisition Delayed Stop	62
5.4.4 Trace Acquisition Range Example	63
5.4.5 Trace Find	66
5.4.6 Trace Filter	69
5.5 Memory Mapping ダイアログボックス	72
5.6 Performance Analysis	73
5.6.1 Performance Analysis Conditions	74
5.6.2 Performance Analysis Properties	75
6 コマンドライン機能	81
6.1 ANALYSIS	83
6.2 ANALYSIS_RANGE	83

6.2.1	指定範囲内時間測定	83
6.2.2	指定アドレス間時間測定	84
6.2.3	指定アドレス範囲間時間測定	84
6.2.4	領域アクセス回数測定	85
6.2.5	指定範囲内コール回数測定	85
6.2.6	実行効率測定条件の表示	85
6.2.7	アドレス検出モードの設定・表示	86
6.2.8	測定時間最小単位の設定・表示	86
6.3	ANALYSIS_RANGE_DELETE	87
6.4	BREAKPOINT/EVENT	87
6.4.1	プログラムブレークポイント	87
6.4.2	アクセスブレークポイント	87
6.4.3	範囲ブレークポイント	88
6.4.4	オプション	88
6.5	BREAKPOINT_CLEAR / EVENT_CLEAR	89
6.6	BREAKPOINT_DISPLAY / EVENT_DISPLAY	90
6.7	BREAKPOINT_ENABLE / EVENT_ENABLE	90
6.8	BREAKPOINT_SEQUENCE / EVNET_SEQUENCE	90
6.9	CLOCK	91
6.10	COVERAGE	91
6.11	COVERAGE_CLEAR	92
6.12	DEVICE_TYPE	92
6.13	EEPROM_DISPLAY	93
6.14	EEPROM_MAP	93
6.15	MAP_SET	93
6.16	MODE	94
6.17	REFRESH	94
6.18	TEST_EMULATOR	94
6.19	TIMER	94
6.20	TRACE_ACQUISITION	94
6.21	TRACE_COMPARE	95
6.22	TRACE_SAVE	95
6.23	TRACE_SEARCH	96
6.24	USER_SIGNALS	96
7	故障解析	97

7.1	テストプログラムを実行するためのシステムセットアップ	97
7.2	テストプログラムによる故障解析	97
7.3	エラー発生時の処理	102

目 次

図 2-1 Computer Properties ダイアログボックス(設定前)	6
図 2-2 Edit Resource Setting ダイアログボックス	7
図 2-3 Computer Properties ダイアログボックス(設定後)	7
図 2-4 エラーメッセージ (1)	10
図 2-5 エラーメッセージ (2)	10
図 3-1 E6000 コネクタの位置	11
図 3-2 ユーザシステムインタフェースケーブル外観図	11
図 3-3 電源プラグ	12
図 3-4 Vcc 信号回路	13
図 3-5 CLK 信号回路	13
図 3-6 RESET 信号回路	13
図 3-7 I/O-1/IRQ, I/O-2/IRQ 信号回路	13
図 3-8 外部プローブコネクタ	14
図 3-9 外部プローブインタフェース回路	14
図 3-10 ユーザシステムと E6000 との Vcc の関係	15
図 4-1 HDI 起動メニュー	17
図 4-2 プラットホームの選択	18
図 4-3 HDI ウィンドウ	18
図 4-4 Configuration ダイアログボックス	20
図 4-5 Memory Mapping ダイアログボックス	21
図 4-6 Edit Memory Mapping ダイアログボックス	22
図 4-7 System Status ウィンドウ (Memory シート)	22
図 4-8 Open ダイアログボックス (オブジェクトファイルの選択)	23
図 4-9 HDI ダイアログボックス	23
図 4-10 Open ダイアログボックス (ソースファイルの選択)	24
図 4-11 ソースプログラム画面	24
図 4-12 ブレークポイント (PC Break) の設定	25
図 4-13 ステートメントの強調表示	26
図 4-14 System Status ウィンドウ (Platform シート)	26
図 4-15 Registers ウィンドウ	27
図 4-16 Register ダイアログボックス	27
図 4-17 Breakpoints ウィンドウ	28
図 4-18 Open Memory Window の設定	28

図 4-19 Memory ウィンドウ(Byte)	29
図 4-20 Watch ウィンドウ (変数追加後)	29
図 4-21 Watch ウィンドウ (シンボル拡張)	30
図 4-22 Add Watch ダイアログボックス	30
図 4-23 Watch ウィンドウ (変数の追加)	30
図 4-24 Reset Go 実行後の Program ウィンドウ	31
図 4-25 Step In 実行後の Program ウィンドウ (1)	32
図 4-26 Step Out 実行後の Program ウィンドウ	32
図 4-27 Step In 実行後の Program ウィンドウ (2)	33
図 4-28 Step Over 実行後の Program ウィンドウ	33
図 4-29 Locals ウィンドウ	34
図 4-30 Locals ウィンドウ (変数"min"内容変更後)	34
図 4-31 Locals ウィンドウ (配列変数"a"ソート後)	34
図 4-32 ブレークポイントの追加	35
図 4-33 ブレークポイントの追加 (回数指定)	36
図 4-34 Breakpoints ウィンドウ (追加後)	36
図 4-35 Event ブレークポイントによるプログラムの停止	37
図 4-36 Trace ウィンドウ	38
図 4-37 Trace Filter ダイアログボックス	39
図 4-38 Bus / Area の設定	39
図 4-39 Trace ウィンドウ (トレースフィルタ指定)	40
図 4-40 計測条件の選択	40
図 4-41 計測条件の表示	41
図 4-42 計測結果の表示 (1)	42
図 4-43 計測結果の表示 (2)	42
図 4-44 Stack Trace ウィンドウ	43
図 5-1 Configuration ダイアログボックス	47
図 5-2 Driver Details ダイアログボックス	48
図 5-3 System Status ウィンドウ(Session シート)	49
図 5-4 System Status ウィンドウ(Platform シート)	50
図 5-5 System Status ウィンドウ(Memory シート)	51
図 5-6 System Status ウィンドウ(Events シート)	52
図 5-7 Breakpoints ウィンドウ	53
図 5-8 Breakpoint/Event Properties ダイアログボックス(General)	54
図 5-9 Breakpoint/Event Properties ダイアログボックス(Bus / Area)	55

☒ 5-10 Breakpoint/Event Properties ダイアログボックス(Signals)	55
☒ 5-11 Breakpoint/Event Properties ダイアログボックス(Action)	56
☒ 5-12 Event Sequencing ダイアログボックス	58
☒ 5-13 Trace Acquisition ダイアログボックス(General)	60
☒ 5-14 Trace Acquisition ダイアログボックス(Stop)	61
☒ 5-15 Trace Acquisition ダイアログボックス(Delayed Stop)	62
☒ 5-16 Trace Acquisition ダイアログボックス(Point to Point)	63
☒ 5-17 Trace Acquisition ダイアログボックス(Event)	64
☒ 5-18 Trace Acquisition ダイアログボックス(Range)	65
☒ 5-19 Trace Find ダイアログボックス(General)	66
☒ 5-20 Trace Find ダイアログボックス(Bus / Area)	67
☒ 5-21 Trace Find ダイアログボックス(Signals)	67
☒ 5-22 Trace Find ダイアログボックス(Time)	68
☒ 5-23 Trace Filter ダイアログボックス(General)	69
☒ 5-24 Trace Filter ダイアログボックス(Bus / Area)	70
☒ 5-25 Trace Filter ダイアログボックス(Signals)	70
☒ 5-26 Trace Filter ダイアログボックス(Time)	71
☒ 5-27 Memory Mapping ダイアログボックス	72
☒ 5-28 Performance Analysis ダイアログボックス(Graph)	73
☒ 5-29 Performance Analysis ダイアログボックス(Value)	73
☒ 5-30 Performance Analysis Conditions ダイアログボックス	74
☒ 5-31 Performance Analysis Properties ダイアログボックス (1)	75
☒ 5-32 Performance Analysis Properties ダイアログボックス (2)	76
☒ 5-33 Performance Analysis Properties ダイアログボックス (3)	76
☒ 5-34 Performance Analysis Properties ダイアログボックス (4)	77
☒ 5-35 Performance Analysis Properties ダイアログボックス (5)	77

表 目 次

表 1-1 メモリタイプの定義	3
表 1-2 E6000 使用環境条件	4
表 1-3 外形寸法および質量	4
表 2-1 PC インタフェースボードのメモリ領域図	6
表 3-1 MCU と E6000 エミュレータのレジスタ初期値の相違	15
表 4-1 コンフィグレーションオプションの設定例	20
表 4-2 メモリタイプの定義	21
表 4-3 メモリタイプオプション	21
表 4-4 プログラムステップオプション	31
表 5-1 HDI のメニューとマニュアルの対応表	45
表 6-1 HDI コマンドライン機能とマニュアルの対応表	81
表 6-2 ANALYSIS コマンド	83
表 6-3 ANALYSIS_RANGE コマンド (1)	83
表 6-4 ANALYSIS_RANGE コマンド (2)	84
表 6-5 ANALYSIS_RANGE コマンド (3)	84
表 6-6 ANALYSIS_RANGE コマンド (4)	85
表 6-7 ANALYSIS_RANGE コマンド (5)	85
表 6-8 ANALYSIS_RANGE コマンド (6)	85
表 6-9 ANALYSIS_RANGE コマンド (7)	86
表 6-10 ANALYSIS_RANGE コマンド (8)	86
表 6-11 ANALYSIS_RANGE_DELETE コマンド	87
表 6-12 BREAKPOINT_CLEAR / EVENT_CLEAR コマンド	89
表 6-13 BREAKPOINT_ENABLE / EVENT_ENABLE コマンド	90
表 6-14 CLOCK コマンド	91
表 6-15 DEVICE_TYPE コマンド	92
表 6-16 TIMER コマンド	94
表 6-17 USER_SIGNALS コマンド	96

1 はじめに

E6000 エミュレータは、日立 MCU をサポートする高性能リアルタイムインサーキットエミュレータです。本 E6000 エミュレータは H8/3112, 3113, 3150, 3160, AE-3 シリーズマイクロコントローラ用のプログラムの開発とデバッグができます。

E6000 エミュレータは、ソフトウェア開発とデバッグのために単体で、あるいはユーザシステムのデバッグのためにユーザシステムインタフェースケーブルでユーザシステムに接続した状態で使用できます。

E6000 エミュレータは、Windows[®]用アプリケーションである HDI とともに動作します。HDI は、E6000 エミュレータハードウェアを制御し、豊富なコマンドを提供します。

1.1 デバッグの特長

1.1.1 ブレークポイント

E6000 エミュレータは、強力なハードウェアブレークおよびプログラムブレークを備えているので、ソフトウェアとユーザシステムのデバッグを効率よく実行できます。

ハードウェアブレークポイント

イベント検出システムのイベントチャンネルと範囲チャンネルを使って、最大12箇所のブレークポイントが設定できます。ハードウェアブレークポイントに関しては、「1.2 イベント検出システム(CES)」を参照してください。

プログラムブレークポイント(PCブレークポイント)

最大 256 のプログラムブレークポイントが設定できます。プログラムブレークポイントは、ユーザ命令を BREAK 命令で置き換えることによって設定されます。

1.1.2 トレース

E6000 エミュレータは、強力なリアルタイムトレース機能を備えていますので、MCU の動作を詳細に調べることができます。リアルタイムトレースバッファは、32768 までのバスサイクルを保持でき、実行中は常に更新されます。バッファはローリングバッファとして構成され、エミュレーションを中断することなく、トレースを中断しトレース内容を表示することができます。

トレースバッファ内の取得データは、デバッグを容易にするためにソースプログラムおよびアセンブリ言語の両方で表示されます。ただし、トレースフィルタリングが行われた場合は、アセンブリ言語だけが表示されます。

トレースバッファは、すべてのバスサイクルあるいは選択されたサイクルだけを記憶するように制御されます。イベント検出システム を使用して所望のトレース制御を選択します。詳細は、以下の「1.2 イベント検出システム」を参照してください。

すべてのバスサイクルを記憶しておいて、選択されたサイクルだけを見ることも可能です。これをトレースフィルタリングといいます。

1.1.3 実行時間測定

E6000 エミュレータによって、総実行時間の測定、またはイベント検出システムで指定されたイベント間の実行時間の測定ができます。タイマーの分解能は以下のいずれかの値に設定できます。

20ns, 125ns, 250ns, 500ns, 1 μ s, 2 μ s, 4 μ s, 8 μ s, 16 μ s

測定可能な最大時間は、分解能 20ns で約 6 時、分解能 16 μ s で約 200 日間です。

1.2 イベント検出システム (CES: Complex Event System)

実際のデバッグの大部分において、デバッグしようとするプログラムの不具合またはハードウェアの不具合は、限定された状況においてのみ、発生します。たとえば、あるハードウェアエラーは、メモリの特定の領域がアクセスされた時のみ発生します。簡単なプログラムブレークポイントを使用してその問題を調べ上げるのは、非常に困難です。

E6000 エミュレータは、調べたい条件を正確に記述できるシステム (イベント検出システム) を備えています。これによって、MCU 信号の指定された組み合わせのイベントを定義できます。

イベント検出システムは、E6000 エミュレータのトレース、ブレーク、およびイベント間実行時間測定機能を制御します。

1.2.1 イベントチャネル

イベントチャネルによって、指定されたイベントの発生を検出できます。イベントは以下の項目の組み合わせで定義できます。

- ・ アドレスまたはアドレス範囲
- ・ アドレス範囲外
- ・ リード、ライトまたは両方とも
- ・ マスク条件指定付きデータ
- ・ MCUアクセスタイプ (命令プリフェッチなど)
- ・ MCUアクセス領域 (内蔵ROM、内蔵RAMなど)
- ・ 4つの外部プローブ信号の値
- ・ イベントの発生回数
- ・ イベントの発生後のディレイサイクル数

また、最大 8 イベントがシーケンスで組み合わせられます。それぞれのイベントは、シーケンスにおける前のイベントの発生によって起動、あるいは停止します。たとえば、内蔵 RAM の指定された領域がアクセスされた後で I/O レジスタが書き込まれたときというブレーク条件を設定できます。

1.2.2 範囲チャネル

範囲チャネルは、以下の項目の組み合わせで定義できます。

- ・ アドレスまたはアドレス範囲
- ・ リード、ライトまたは両方とも
- ・ マスク条件指定付きデータ
- ・ MCU アクセスタイプ (命令プリフェッチなど)
- ・ MCU アクセス領域 (内蔵 ROM、内蔵 RAM など)
- ・ 4つの外部プローブ信号の値
- ・ イベントの発生後のディレイサイクル数

イベント検出システムは、E6000 エミュレータの以下の機能を制御するのに使われます。

1.2.3 ブレーク

指定されたイベントまたはイベントのシーケンスが発生したときに、プログラム実行を停止します。たとえば、プログラムがあるアドレスからデータ読み出し後、あるアドレスにデータを書き込んだときに実行を停止するように、ブレークを設定できます。また、ブレークは 65535 バスサイクルまで任意に遅らせることができます。

1.2.4 イベント間実行時間測定

2つのイベントを設定し、最初のイベントの発生と2番目のイベント発生間のプログラム実行時間を測定できます。

1.3 ハードウェアの特長

1.3.1 メモリ

E6000 エミュレータは、エミュレーションメモリとして内蔵 ROM/内蔵 RAM 用代替メモリを標準装備しています。また、プログラム開発段階で内蔵 ROM や内蔵 RAM だけではメモリが不足する場合、これらのエミュレーションメモリを一時的に使用することができます。しかし、MCU でもプログラムが動作するよう、プログラムの最終段階での評価ではこの一時使用メモリは使用せずプログラムの正常動作をご確認ください。

エミュレーションメモリは、MCU アドレス空間の任意サイズのメモリブロックに、1バイト単位で割り付けできます。

リザーブ領域は、[Configure Map...] コマンドを使用して、E6000 エミュレータ上のメモリに指定できます。

エミュレーションメモリのメモリタイプ定義を以下に示します。

表 1-1 メモリタイプの定義

メモリタイプ	説明
オンチップ	MCU内蔵メモリ
エミュレータ	一時使用メモリ

メモリアドレスの指定されたブロックの内容は、[Memory...] コマンドを使って表示されます。メモリの内容はいつでも（プログラム実行中であっても）変更でき、その結果は、他の関連するウインドウにすぐに反映されます。

1.3.2 エミュレーションクロック

エミュレーションクロックは 3.5712MHz, 4.9152MHz, 7.1424MHz, 9.8304MHz, ターゲットクロックのいずれかの周波数に設定できます。また、E6000 エミュレータは各 MCU のハードウェアマニュアルに記載されている保証動作範囲内でエミュレーションが可能です。

1.3.3 外部プローブ

ユーザシステム上の任意の信号をブレイクもしくはトレースに使用するために、E6000 エミュレータには外部プローブが接続できます。外部プローブの信号はローまたはハイレベルに応じて、イベント検出システムの条件として設定できます。

1.3.4 使用環境条件

表 1-2 E6000使用環境条件

項番	項目	仕様
1	温度	動作時 : 10 ~ 35 非動作時 : -10 ~ 50
2	湿度	動作時 : 35 ~ 80 %RH (結露なし) 非動作時 : 35 ~ 80 %RH (結露なし)
3	周囲ガス	腐食性ガスのないこと
4	DC入力電源	電圧 : 5V ± 5 % 電流 : Max. 6A
5	ユーザVcc (Uvcc)	電圧 : 2.7 ~ 5.5Vの範囲で各MCUの電源仕様に従う
6	AC入力電源	電圧 : AC 100 ~ 240V 周波数 : 50 / 60Hz 消費電力 : 61 ~ 70VA

1.3.5 外形寸法と質量

表 1-3 外形寸法および質量

項番	項目	仕様
1	外形寸法	219 x 170 x 54 (mm)
2	質量	900 (g)

2 セットアップ

本章は以下の方法について説明します。

- ・ PC インタフェースボード (別売 HS6000E1101H) のセットアップ
- ・ E6000 エミュレータのセットアップ
- ・ HDI のインストールとシステムの動作チェック

PC Card(PCMCIA)などの他のホストシステムインタフェースボードをご使用の場合は、各製品に添付のマニュアルをご覧ください。

E6000 エミュレータは、PC インタフェースボードを使って HDI と通信します。はじめに、PC インタフェースボードを PC に差し込む必要があります。

PC インタフェースボードはメモリマップボードであり、差し込む前に、PC インタフェースボードが使うメモリ領域を確保しなければなりません。これによって、他のプログラムが不用意に PC インタフェースボードを使ってしまうことを防止できます。

PC インタフェースボードに割り当てたメモリ領域が、他のボードに割り当てた領域と重ならないようにします。もしも重なると、PC インタフェースボードと E6000 エミュレータは正しく動作しません。出荷時には、PC インタフェースボードのメモリ領域は H'D0000 から H'D3FFF に割り当ててあります。

Windows® 95 または Windows® 98 をご使用の方は、「2.2 Windows® 95, Windows® 98 での PC インタフェースボードのセットアップ」を、Windows NT® をご使用の方は、「2.3 Windows NT® 4.0 での PC インタフェースボードのセットアップ」をご覧ください。

注：Windows® 2000 では PC インタフェースボードはサポートしていません。

2.1 パッケージ内容

E6000 エミュレータには、以下の構成部品が梱包されています。

- ・ E6000 エミュレータ本体
- ・ AC 電源アダプタ 5V 6A
- ・ HDI インストールディスク (HS3160EPI60SR)
- ・ ユーザインタフェースケーブル
- ・ 外部プローブ
- ・ E6000 用 Hitachi Debugging Interface セットアップガイド

セットアップの前に、上記の構成部品がすべてそろっていることを確認してください。

2.2 Windows® 95, Windows® 98 での PC インタフェースボードのセットアップ

2.2.1 PC インタフェースボードのセットアップ

- ・ Windows® 95 または Windows® 98 を起動します。
- ・ [**My Computer**] アイコンをマウスの右ボタンでクリックし、ポップアップメニューから [**Properties**] を選択します。
System Properties ダイアログボックスが表示されます。
- ・ Device Manager パネルの [**Computer**] アイコンをダブルクリックし、Computer Properties ダイアログボックスを開きます。
- ・ View Resources パネルの [**Memory**] をクリックし、メモリのリソースを表示します。

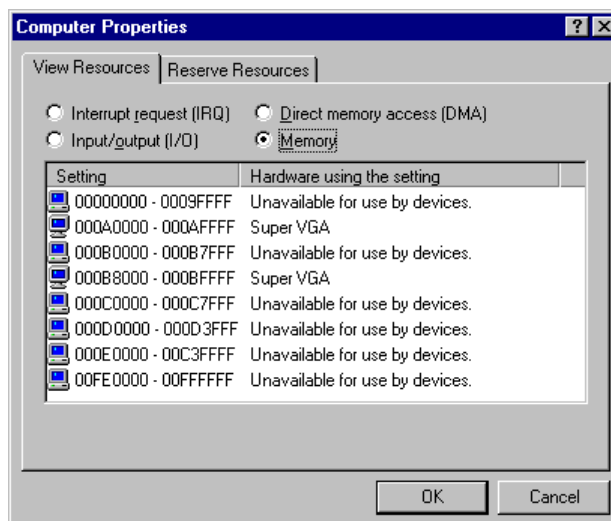


図 2-1 Computer Properties ダイアログボックス(設定前)

ここにリストされていないメモリ領域が、PC インタフェースボード用に使用できます。下の表は、PC インタフェースボードのリアパネルのスイッチによって指定されるアドレスを示しています。これらのメモリ領域の中で、Computer Properties ダイアログボックスでリストされていないメモリ領域を選択してください。たとえば、H'D8000 から H'DBFFF の領域を選択すると、対応するスイッチ番号は6になります。

表 2-1 PC インタフェースボードのメモリ領域図

メモリ領域	スイッチ
H'C0000 ~ H'C3FFF	0
H'C4000 ~ H'C7FFF	1
H'C8000 ~ H'CBFFF	2
H'CC000 ~ H'CFFFF	3
H'D0000 ~ H'D3FFF (出荷時の設定)	4
H'D4000 ~ H'D7FFF	5
H'D8000 ~ H'DBFFF	6
H'DC000 ~ H'DFFFF	7
H'E0000 ~ H'E3FFF	8
H'E4000 ~ H'E7FFF	9
H'E8000 ~ H'EBFFF	A
H'EC000 ~ H'EFFFF	B

選択したメモリ領域を Windows® 95 または Windows® 98 が使用しないよう、以下の手順で登録します。

- Reserve Resources パネルの [**Memory**] をクリックし、[**Add**] をクリックします。

Edit Resource Setting ダイアログボックスが表示されます。

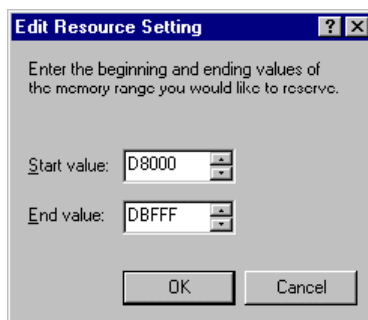


図 2-2 Edit Resource Setting ダイアログボックス

選択したメモリ領域の [Start value] [End value] を入力してください。

PC をリスタートせずシャットダウンし、電源スイッチを切ってください。

小型のマイナスドライバを使って、PC インタフェースボードのリアパネルのスイッチを回し、選択したメモリ領域に対応するスイッチ番号を矢印が差すようにしてください。

- PC のカバーを取り外し、未使用の ISA バススロットに PC インタフェースボードを差し込んでください。
- PC のカバーを取り付けてください。
- PC インタフェースボードと E6000 エミュレータの “ PC IF ” コネクタの間に PC インタフェースケーブルを接続してください。各プラグはカチッと音がするまでしっかりと差し込んでください。
- PC の電源スイッチを入れてください。
- Computer Properties ダイアログボックスで選択したメモリ領域が、System Reserved とリストされていることを確認してください。

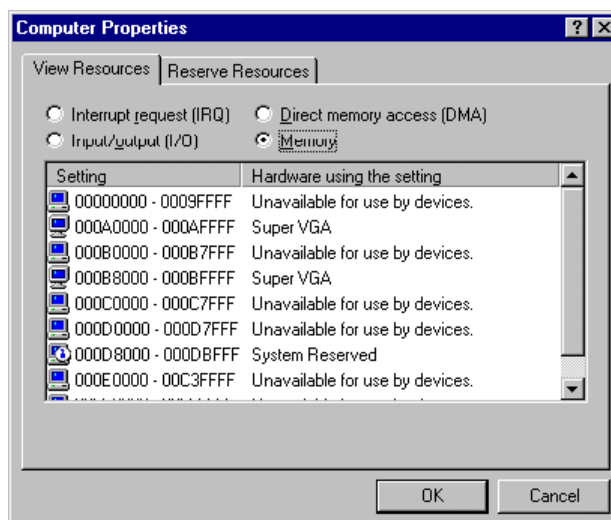


図 2-3 Computer Properties ダイアログボックス(設定後)

2.2.2 CONFIG.SYS の変更

次のステップは、PC インタフェースボードが使用するメモリ領域を、他のプログラムが使ってしまうことを防止します。

- ・ [**Start**] メニューから [**Run**] を選択してください。
- ・ “ SYSEDIT ” とタイプし、 [**OK**] をクリックしてください。

CONFIG.SYS ファイル中で EMM386.EXE を使用している場合は、以下の変更を行う必要があります。CONFIG.SYS ファイルを使用していない場合、または CONFIG.SYS ファイルを使用している場合、その中で EMM386.EXE を使用していない場合は、「2.2.3 SYSTEM.INI の変更」に進んでください。

- ・ CONFIG.SYS ファイルの下記の行にラインカーソルを移動してください。
DEVICE=C:¥WINDOWS¥EMM386.EXE

- ・ この行を以下のように変更してください。
DEVICE=C:¥WINDOWS¥EMM386.EXE X=aaaa-bbbb

- ・ aaaa は Start value、bbbb は End value のそれぞれ最下位を取った値です。たとえば、メモリ領域 H'D8000 ~ H'DBFFF、スイッチが 6 に設定されていれば、この行を以下のように設定します。

DEVICE=C:¥WINDOWS¥EMM386.EXE X=D800-DBFF

CONFIG.SYS ファイルをセーブしてください。

2.2.3 SYSTEM.INI の変更

- ・ SYSTEM.INI ファイル中にある [386Enh] セクションに以下の行を追加してください。
EMMExclude=aaaa-bbbb

- ・ aaaa は Start value、bbbb は End value のそれぞれ最下位を取った値です。たとえば、メモリ領域 H'D8000 ~ H'DBFFF、スイッチが 6 に設定されていれば、この行を以下のように設定します。

EMMExclude=D800-DBFF

- ・ SYSTEM.INI ファイルをセーブし、SYSEDIT を終了させてください。
- ・ PC を再起動してください。

これによって、Windows®はこのメモリ領域を使いません。これで E6000 エミュレータを接続し、HDI を実行して E6000 エミュレータの通信状態をチェックする準備が整いました。

2.3 Windows NT® 4.0 での PC インタフェースボードのセットアップ

PC インタフェースボードは ISA バススロットを使用しますので、ISA バススロットをサポートしていない PC では使用することはできません。

ISA ボードのインストール方法については、お使いの PC に付属のマニュアルを参照してください。ここでは一般的な方法を述べます。

(1) Windows NT®実行

- ・ Start/Programs/Administrative Tools (Common)/Windows NT Diagnostics を実行してください。
- ・ 'Resource' タブの 'Memory' ボタンをクリックし、使用されている上位メモリ領域を以下に記録してください。

#	Start	End	#	Start	End	#	Start	End
0			4			8		
1			5			9		
2			6			A		
3			7			B		

- Windows NT®をシャットダウンしてください。

(2) PC をセットアップモードで起動

セットアップモードについては、お使いの PC に付属のマニュアルを参照してください。

- 使用されている上位メモリ領域をチェックしてください。

#	Start	End	#	Start	End	#	Start	End
0			4			8		
1			5			9		
2			6			A		
3			7			B		

(これは、上で得た Windows NT®の値と同じはずです)

- PC インタフェースボードの設定を登録してください。使用するメモリ領域は、他のデバイスにいつも使用されず、かつ PC インタフェースボードのスイッチ位置の一つと同じにする必要があります。

スイッチ位置：

#	Start	End	#	Start	End	#	Start	End
0	C0000	C3FFF	4	D0000	D3FFF	8	E0000	E3FFF
1	C4000	C7FFF	5	D4000	D7FFF	9	E4000	E7FFF
2	C8000	CBFFF	6	D8000	DBFFF	A	E8000	EBFFF
3	CC000	FFFFF	7	DC000	DFFFF	B	EC000	FFFFF

ご使用の PC に Intel P&P BIOS ディスクが付属の場合は以下の手順で設定します。

- Intel P&P BIOS ディスクで PC を起動します。
- 'View/System Resources' で使用している上位メモリ領域をチェックします。
- 'Configure/Add Card/Others...' で 'Unlisted Card' を追加します。
- .CFG ファイルがありませんので、次のダイアログボックスでは No と答えます。
- 'Configure Unlisted Card' ダイアログボックスで、'Memory [hex]' リストボックスに移動します。
- 'Add Memory...' ボタンを使用し、'Specify Memory' ダイアログボックスを表示します。
- 他のデバイスにいつも使用されず、かつ PC インタフェースボードのスイッチ位置の一つと同じメモリ領域を入力します。

- ファイルをセーブします。
- 終了します。
- PC をリスタートせずシャットダウンし、電源スイッチを切ってください。
- 小型のマイナスイコライバを使って、PC インタフェースボードのリアパネルのスイッチを回し、選択したメモリ領域に対応するスイッチ番号を矢印が差すようにしてください。
- PC のカバーを取り外し、未使用の ISA バススロットに PC インタフェースボードを差し込んでください。
- PC のカバーを取り付けてください。

2. セットアップ

- ・ PC インタフェースボードと E6000 エミュレータの“ PC IF ”コネクタの間に PC インタフェースケーブルを接続してください。各プラグはカチッと音がするまでしっかりと差し込んでください。
- ・ PC の電源スイッチを入れてください。

2.4 HDI のインストール

HDI のインストールについては、添付の「E6000 用 Hitachi Debugging Interface セットアップガイド」を参照してください。

2.5 トラブルシューティング

2.5.1 接続不良

イニシャライズ中に以下のメッセージボックスが現れた場合 (ISA インタフェースボードでの例)、PC インタフェースボードは E6000 エミュレータを検出できていません。

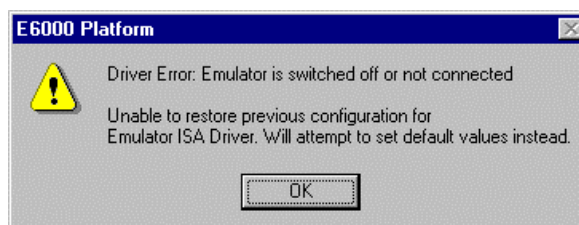


図 2-4 エラーメッセージ (1)

考えられる原因としては以下のようなものがあります。

- ・ 添付の AC 電源アダプタが E6000 エミュレータに接続されていないか、または E6000 エミュレータの電源スイッチが入っていません。E6000 エミュレータのパワーLEDを確認してください。
- ・ PC インタフェースケーブルが、PC インタフェースボードと E6000 エミュレータの間で正しく接続されていません。

2.5.2 通信不良

以下のメッセージボックスが表示されると (ISA インタフェースボードでの例)、HDI が E6000 エミュレータを正しくセットアップできていません。



図 2-5 エラーメッセージ (2)

考えられる原因としては以下のようなものがあります。

- ・ CONFIG.SYS ファイルに確保されたメモリエリアと PC インタフェースボード上のリヤパネルスイッチの設定が異なります。
- ・ 選択されたメモリエリアが別のアプリケーションで使用されています。

3 ハードウェア

本章は、E6000エミュレータをユーザシステムに接続する方法を説明します。

3.1 ユーザシステムへの接続

本E6000エミュレータはICカードと同一形状の先端部を有したユーザシステムインタフェースケーブルを同梱しています。このケーブルによりエミュレータをICカードと同様に使用できる為、ユーザシステムのICカードリーダー/ライターへの挿抜を容易に行うことができます。

E6000エミュレータをユーザシステムへ接続するには、以下の手順に従ってください。

- ・ ユーザシステムインタフェースケーブルのケーブル本体部をE6000エミュレータへ接続します。接続前に必ず、E6000エミュレータの電源を切ってください。
- ・ ケーブル先端部をユーザシステムに挿入します。
- ・ E6000エミュレータを起動します。

以下に、E6000エミュレータのコネクタを示します。

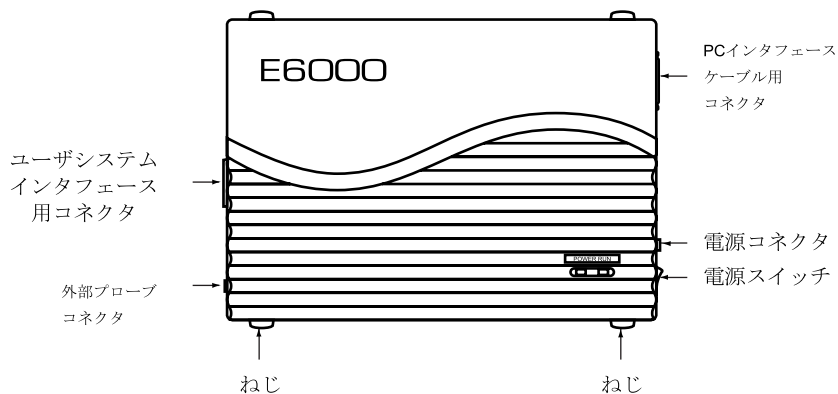


図 3-1 E6000 コネクタの位置

3.1.1 ユーザシステムインタフェースケーブル本体部とE6000エミュレータの接続

ユーザシステムインタフェースケーブル本体部ケーブルをE6000エミュレータに接続してください。ケーブルのコネクタは、まっすぐに、確実に接続されるまで押し込んでください。

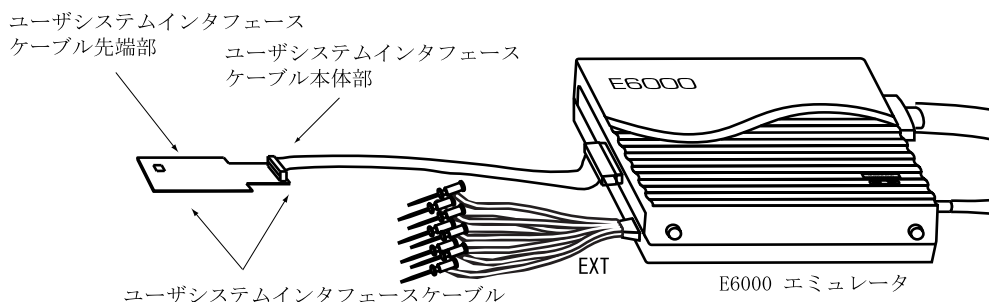


図 3-2 ユーザシステムインタフェースケーブル外観図

3.1.2 ユーザシステムインタフェースケーブル先端部の接続

エミュレータの起動後、ユーザシステムインタフェースケーブル先端部をユーザシステムに挿入してください。挿入の際には下記にご注意ください。

- ・ ICカードリーダーの挿入口に対して、ケーブル先端部の表裏がまっているか確認してください。
- ・ ICカードが自動挿入・排出されるICカードリーダーの場合はICカードリーダーのカードストロークなどの距離を考慮して、ユーザケーブルとエミュレータの間の距離に余裕を持たせてください。
- ・ ユーザケーブル先端部のコンタクトパッドに汚れ、ホコリ、ゴミなどが付着、蓄積するとコンタクト部の電氣的接触が悪くなり、正常に機能しなくなる可能性があります。その際は、コンタクト部を乾いた布などでクリーニングしてください。

3.2 電源供給

3.2.1 AC電源アダプタ

E6000エミュレータに付属のAC電源アダプタを常に使用してください。

3.2.2 極性

以下に電源プラグの極性を示します。

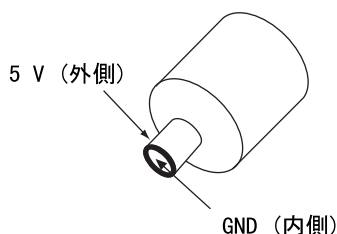


図 3-3 電源プラグ

3.2.3 電源モニタ回路

E6000エミュレータには、電源モニタ回路があり、4.75V以上の電源が供給されているとパワーLEDが緑色に点灯します。パワーLEDが消えている場合は、E6000エミュレータの電源レベルをチェックしてください。電源が4.75V未満の場合、E6000エミュレータに必要な電流が供給されません。

注：必ずE6000エミュレータに付属のAC電源アダプタを使用してください。

3.3 ハードウェアインタフェース

E6000エミュレータのユーザシステムインタフェース信号は、エミュレータ上のMCUに入力する前に、制御回路が挿入されています。

- ・ RESET
- ・ CLK
- ・ I/O-1/IRQ, I/O-2/IRQ

3.3.1 信号保護

ユーザシステムインタフェース信号は、ダイオードによって、過大/逆電圧から保護されています。また、プルアップ抵抗が接続されています。

E6000エミュレータは、ユーザシステムインタフェースケーブル先端部のVcc端子を監視して、ユーザシステムが接続されているかどうかを判断しています。

3.3.2 ユーザインタフェース回路

E6000エミュレータのユーザインタフェースには、ケーブルにより約3nsの信号の遅れが生じます。また、プルアップ抵抗により信号がハイインピーダンス状態でもハイレベルになります。このことを考慮してユーザシステムのハードウェアを調整してください。

以下にユーザインタフェース信号回路を示します。

(1)Vcc

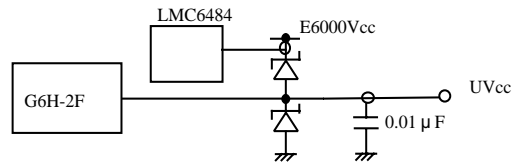


図 3-4 Vcc信号回路

(2)CLK

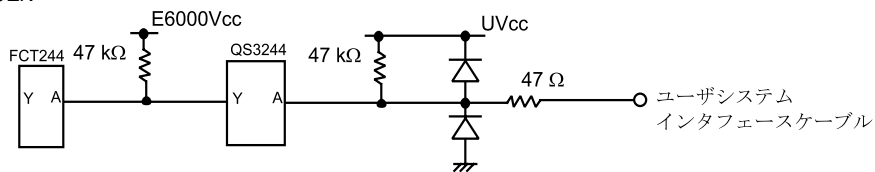


図 3-5 CLK 信号回路

(3)RESET

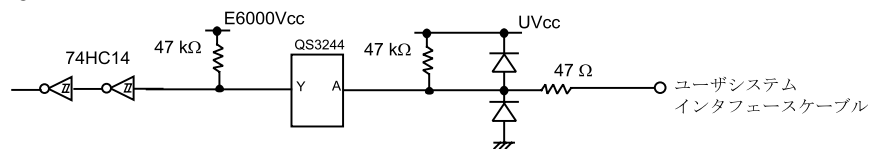


図 3-6 RESET 信号回路

(4)I/O-1/IRQ, I/O-2/IRQ

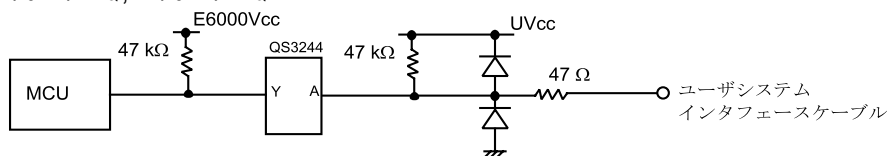


図 3-7 I/O-1/IRQ,I/O-2/IRQ 信号回路

3.3.3 外部プローブ/トリガ出力

E6000エミュレータ筐体側面にあるEXTのマークが記された8ピンコネクタ（ユーザインタフェースコネクタの横）は、外部プローブ入力4本とトリガ出力2本を備えています。以下にこのコネクタのピン配置を示します。

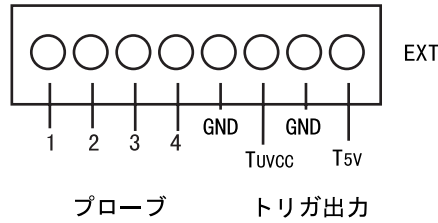


図 3-8 外部プローブコネクタ

以下に外部プローブのインタフェース回路を示します。

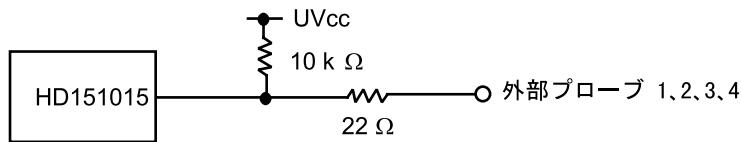


図 3-9 外部プローブインタフェース回路

トリガ出力（赤色ケーブル）はイベントチャンネル8によって出力されるローレベル信号です。トリガ出力はT5V（白色ケーブル：2.5V～5.0Vの範囲でユーザシステムの電圧レベルに依存しません）、またはTUVCC（黄色ケーブル：ユーザシステム電源電圧）レベルの2つあります。GNDは黒色ケーブルです。

3.3.4 電源回路

注意

- 1 ユーザシステムをE6000エミュレータに接続した時は、エミュレータを起動した後ユーザシステムの電源のON/OFFをしてください。
エミュレータの電源がOFFのままユーザシステムの電源を入れた場合、エミュレータホストコンピュータの発煙発火の可能性があります。

E6000エミュレータに搭載されているMCUは常にE6000の5V電源で動作しています。MCU電源がユーザシステムから供給されることはありません。MCUとユーザシステムとのI/F回路には電圧レベルシフタを使用し、低電圧対応しています。よって、ユーザシステムが5V以下で動作していてもエミュレータはMCUの最高動作周波数で動作できてしまいますので注意してください。

Configurationダイアログボックスを使用して、[User LowVCC Threshold] を5Vから0Vの範囲で設定できます。ユーザVccがその値よりも下がった場合、System StatusウィンドウのPlatformシートにある[User System LowVoltage]には[Down]が表示されます。User LowVCC Threshold電源レベルよりも高い場合は、[OK]が表示されます。ユーザシステムインタフェースケーブル未接続の場合、本ユーザ電源電圧監視機能は働きません。

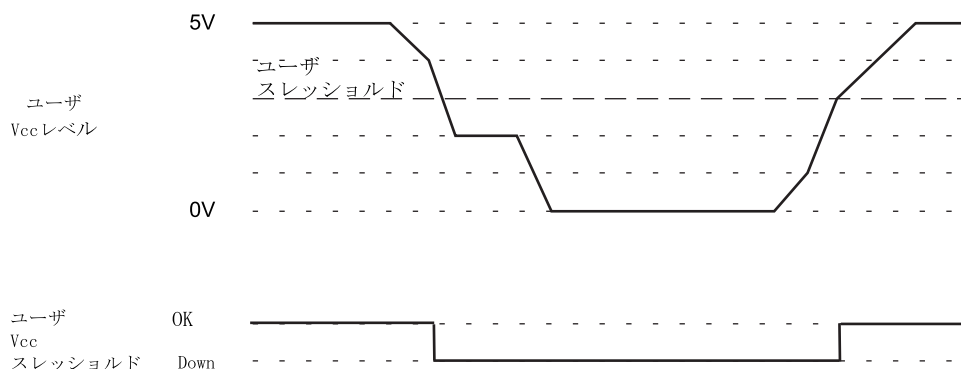


図 3-10 ユーザシステムとE6000とのVccの関係

3.4 MCUとE6000エミュレータの相違点

E6000エミュレータの電源投入後およびコマンドリセット後の、MCUとE6000エミュレータのレジスタ初期値の相違を以下に示します。

表 3-1 MCUとE6000エミュレータのレジスタ初期値の相違

状態	レジスタ	E6000エミュレータ	MCU
電源投入後	PC	リセットベクタ値	リセットベクタ値
	R0 to R6	H'0000	不定
	R7 (SP)	H'0010	不定
	CCR	Iマスクは1	Iマスクは1
		その他は不定	その他は不定
リセットコマンド後	PC	リセットベクタ値	リセットベクタ値
	R0 to R6	不定	不定
	R7 (SP)	H'0010	不定
	CCR	Iマスクは1	Iマスクは1
		その他は不定	その他は不定

E6000エミュレータのI/Oポート上の保護回路詳細については、「3.3.2 ユーザインタフェース回路」を参照してください。

表 3-2 MCUとE6000エミュレータのレジスタアクセスの相違

レジスタ	MCU	H8/3160 E6000
バイトサイズレジスタへのワードアクセス	各MCUのハードウェアマニュアルをご参照ください。	バイトサイズレジスタへワードアクセスした場合、その結果は保証されません。また、エミュレータとMCUでは結果が異なります。
SYSCRレジスタのCPUCS0ビット	各MCUのハードウェアマニュアルをご参照ください。	H8/3164, AE350:GO RESETコマンドによりプログラムを実行した場合およびGO中にリセット信号を入力した後に1回のみ書き換え可能です。

3.4.1 EEPROM

表 3-3 EEPROMに関する相違点

詳細項目	H8/3112	H8/3113, 3150シリーズ	H8/3160, AE-3シリーズ	H8/3160 E6000
書き込み時間	15ms(max)	10ms(max)	4ms(max)	常にMCUのmax時間で実行されます
重ね書き時間	15ms(max)	5ms(max)	2ms(max)	常にMCUのmax時間で実行されます
消去時間	15ms(max)	5ms(max)	2ms(max)	常にMCUのmax時間で実行されます
レジスタ設定	MCUではレジスタ設定に下記制限があります。 ソースレジスタ(R5) : RAM デスティネーションレジスタ(R6) : EEPROM バイトカウント(R4L) : ページサイズまで この制限を違反した場合はEEPMOV命令の結果は保証されません。			左記レジスタ設定の制限を違反した場合はEEPMOV命令の結果は保証されません。また、MCUと結果が異なります。特にデスティネーションにROMを指定した場合ROMの内容が書き換わりますのでご注意ください。

3.4.2 WDT

表 3-4 WDTに関する相違点

詳細項目	H8/3113,3150,3160,AE-3シリーズ MCU	H8/3160 E6000
EWE割込み	下記例では2)の命令実行後に割込み処理を開始します。 1)MOV.B ROH,@TCSR 2)MOV.B ROL,@ECR 3)EEPMOV	左記の命令列に対してブレーク,またはステップで実行するとEWE割込み例外の発生がMCUと異なります。1)直後にUDFが発生した場合、EWE割込みも1)直後に入ります。
UDF割込み	EWE割込み処理ルーチンのRTE命令と次の命令の間にUDF割込み処理が入ることはありません。	左記個所でブレーク,ステップをするとUDF割込みが入ります
TCWAレジスタ	各MCUのハードウェアマニュアルをご参照ください。	E6000では下記の場合にTCWA機能が働きません。(TCNTレジスタのリロードはできません) H8/3158: TCWA=H'8E-H'BE H8/3164,AE350: TCWA=H'6E

3.4.3 セキュリティ

表 3-5 セキュリティに関する相違点

詳細項目	H8/3113,3150,3160,AE-3シリーズ MCU	H8/3160 E6000
HVD	高電圧でリセット	サポートしていません

4 チュートリアル

本章では、E6000 エミュレータの主な特長を HDI の操作例に従って説明します。チュートリアルでは、E6000 エミュレータ上のエミュレーションメモリを使用して実行しますので、E6000 エミュレータをユーザシステムに接続する必要はありません。

例といたしまして、E6000 H8/3153 を使用した手順を説明します。他の製品をご使用の方は、ファイル名、ディレクトリ名などをご使用のものに読みかえてください。

4.1 はじめに

このチュートリアルは、簡単な C プログラムで作成されています。本章を読む前に、以下のことを行ってください。

- ・ 「2 セットアップ」に従って、E6000 エミュレータを HDI で起動してください。
このチュートリアルを使用するために E6000 エミュレータをユーザシステムに接続する必要はありません。
- ・ MCU のアーキテクチャと命令セットについてよく理解してください。詳しくは、ご使用になる MCU のハードウェアマニュアルとプログラミングマニュアルを参照してください。

チュートリアルは、10 個のランダムデータを昇順、降順に並び換えるプログラムです。ソースプログラム (Sort.c) および ELF/DWARF2 フォーマットのオブジェクトファイル (Tutorial.abs) は、HDI のインストールディスク中に用意されています。

4.2 HDI の起動

HDI を起動するには、[Start]メニューの[HDI for E6000 H8_3160]から[Hitachi Debugging Interface] アイコンを選択してください。

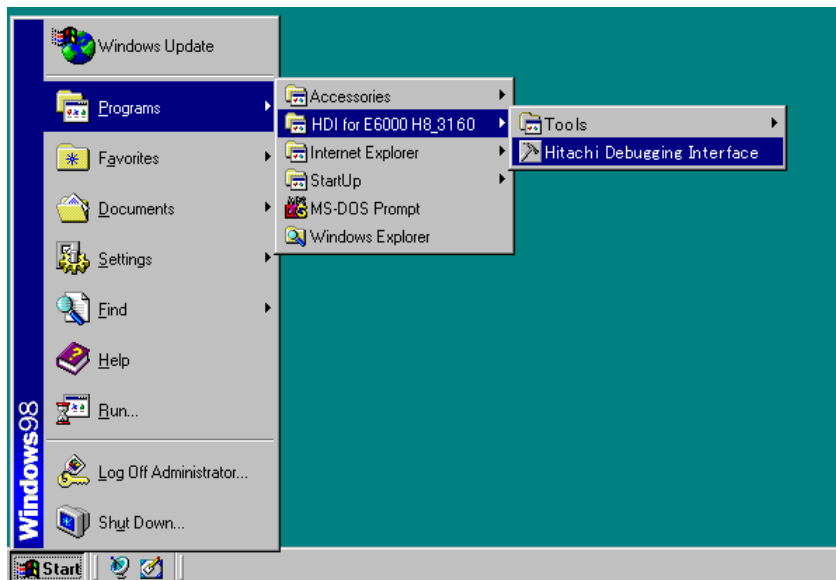


図 4-1 HDI 起動メニュー

4.2.1 ターゲットプラットフォームの選択

HDI は複数のターゲットプラットフォームをサポートする拡張機能があります。複数のプラットフォーム用にシステムがセットアップされると、使用するプラットフォームを選択する必要があります。

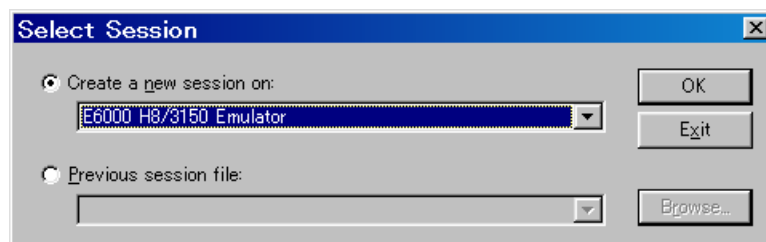


図 4-2 プラットホームの選択

- このチュートリアルでは、E6000 H8/3150 Emulator を選択し、[OK] をクリックしてください。

[File] メニューから [New Session...] を選択すれば、いつでもターゲットプラットフォームを変更できます。

E6000 エミュレータが正しくセットアップされていれば、ステータスバーの [Link up] メッセージと共に、HDI ウィンドウが表示されます。以下にウィンドウの主な機能を示します。

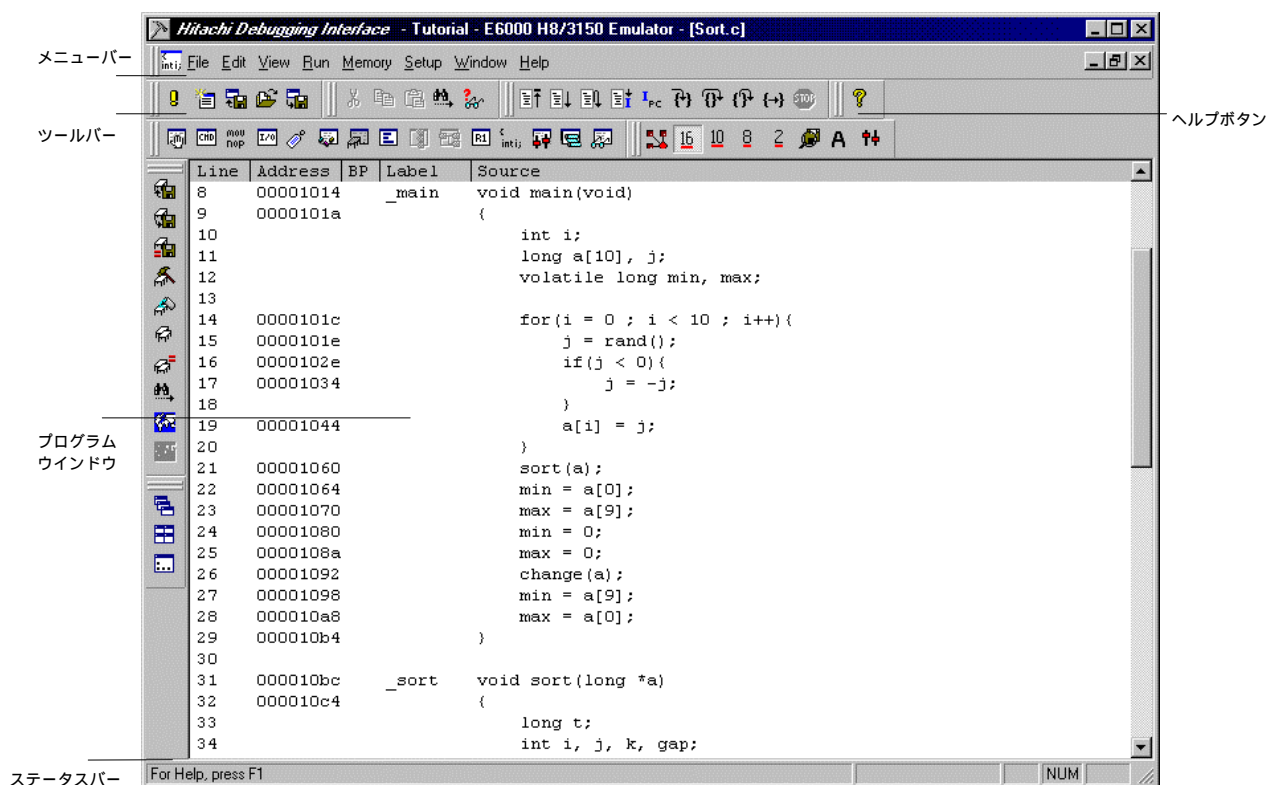


図 4-3 HDI ウィンドウ

HDI の主な機能については、「日立デバッグインタフェースユーザズマニュアル」をご覧ください。E6000 エミュレータに特有な機能は、オンラインヘルプを参照してください。

メニューバー

メニューバーには、E6000 エミュレータの環境設定または HDI のデバッグ機能を使用するためのコマンドがあります。

ツールバー

頻繁に使うメニューコマンドのショートカットとして便利なボタンです。

プログラムウインドウ

デバッグしているソースプログラムなどを表示します。

ステータスバー

E6000 エミュレータの状態、例えばダウンロードの進捗状況や実行モードにおけるアドレスバスの状態を示します。

ヘルプボタン

HDI の使い方やコマンド構成についてのヘルプ画面を表示します。

4.3 E6000 エミュレータのセットアップ

E6000 エミュレータにプログラムをダウンロードする前に、対象 MCU 条件を設定しなければなりません。以下の項目を設定する必要があります。

- ・ デバイスタイプ
- ・ 動作クロック
- ・ ユーザ信号
- ・ メモリマップ

以下に、チュートリアルプログラム用に E6000 エミュレータを設定する方法について述べます。

4.3.1 プラットフォームの構成

- ・ 選択したプラットフォームに固有の設定をするために、[**Setup**] メニューから [**Configure Platform...**] を選択してください。
以下のダイアログボックスが表示されます。

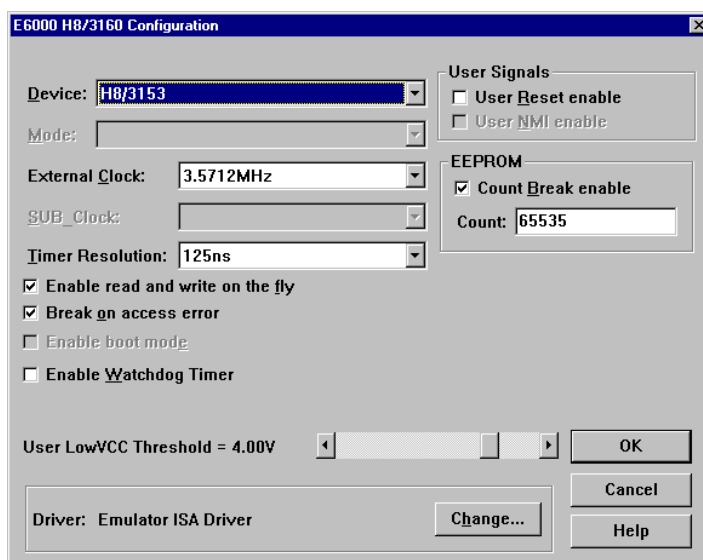


図 4-4 Configuration ダイアログボックス

- ・ オプションを以下のように設定してください。


表 4-1 コンフィグレーションオプションの設定例

オプション	設定値
デバイス (Device)	H8/3153
外部入力クロック (Clock)	3.5712MHz
タイマ分解能 (Timer Resolution)	125ns
ユーザシステムの電圧監視レベル (User LowVcc Threshold)	4.00V
ドライバ (Driver)	Emulator ISA Driver
Enable Watchdog Timer, User Reset enable	無効
その他のオプション	有効
Break Count	65535

- ・ [**OK**] をクリックして対象 MCU 条件を設定してください。

4.3.2 メモリマッピング

Configuration ダイアログボックスでデバイスを選択すると、HDI は自動的に選択したデバイスに合わせたマップの割り付けを行います。

- 現在のメモリマップを表示するには、[Memory] メニューから [Configure Map...] を選択するか、またはツールバーの [Memory Map] ボタン  をクリックしてください。Memory Mapping ダイアログボックスが以下のように表示されます。

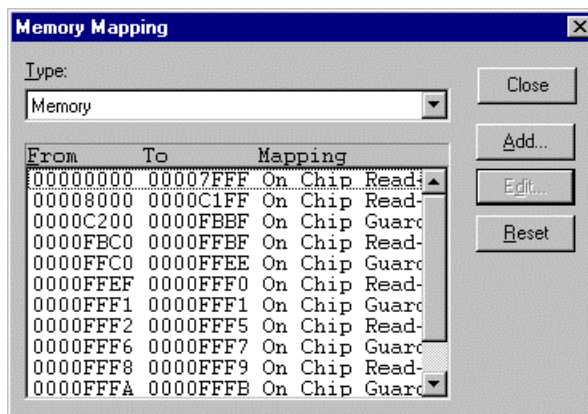


図 4-5 Memory Mapping ダイアログボックス

E6000 エミュレータのメモリには、以下のタイプがあります。

表 4-2 メモリタイプの定義

メモリタイプ	説明
オンチップ (On Chip)	MCU 内蔵メモリをアクセスします。
エミュレータ (Emulator)	一時使用メモリをアクセスします。

また、アクセス制限については以下の3つのタイプがあります。

表 4-3 メモリタイプオプション

アクセスタイプ	説明
リードライト (Read-Write)	RAM
リードオンリ (Read-Only)	ROM
ガード (Guarded)	アクセス不可

本チュートリアルでは、デフォルトのマッピングを使用しますが、以下のようにメモリの割り付け状態を見ることもできます。

- マップ設定を変更する場合は、対象の設定値を選択して [Edit...] ボタンをクリックするか、または対象のマップ設定行をダブルクリックしてください。
- ここでは Memory Mapping ダイアログボックスの On Chip Read-only の箇所をダブルクリックしてください。

Edit Memory Mapping ダイアログボックスが表示されます。

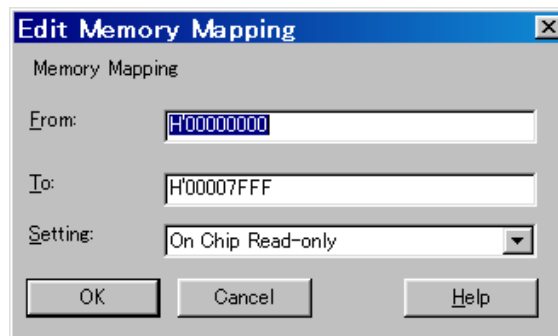



図 4-6 Edit Memory Mapping ダイアログボックス

- ・ [OK] をクリックして、ダイアログボックスを閉じてください。
- ・ デバイスのマップ情報を表示するには [View] メニューから [Status] を選択するか、またはツールバーの [Status] ボタン  をクリックし System Status ウィンドウを開き、Memory シートを選択してください。デバイスのマップ情報が以下のように表示されます。

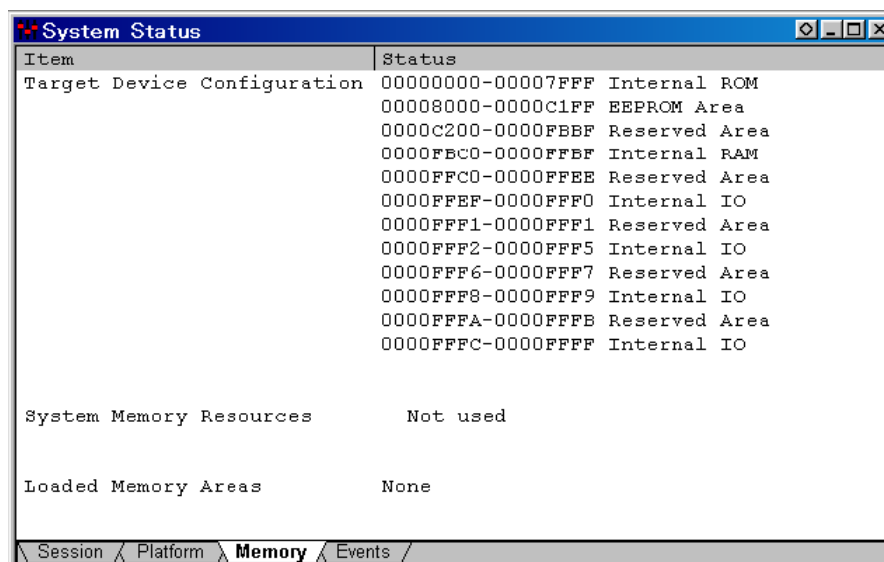


図 4-7 System Status ウィンドウ (Memory シート)


注: メモリマップは対象 MCU によって異なります。

4.4 チュートリアルプログラムのダウンロード

E6000 エミュレータを上記のようにセットアップした後、デバッグしたいオブジェクトプログラムをダウンロードします。

4.4.1 オブジェクトファイルのダウンロード

最初に、以下のように ELF/DWARF2 フォーマットオブジェクトファイルをロードしてください。

- ・ [File] メニューから [Load Program...] を選択するか、またはツールバーの [Load Program] ボタン  をクリックしてください。

Load Program ダイアログボックスが開きます。

- ・ [Browse...] ボタンをクリックし、Open ダイアログボックスより Tutorial ディレクトリの下で Tutorial.abs ファイルを選択した後、[Open] ボタンをクリックしてください。

Load Program ダイアログボックスに戻りますので、さらに [Open] ボタンをクリックしファイルのダウンロードを開始してください。

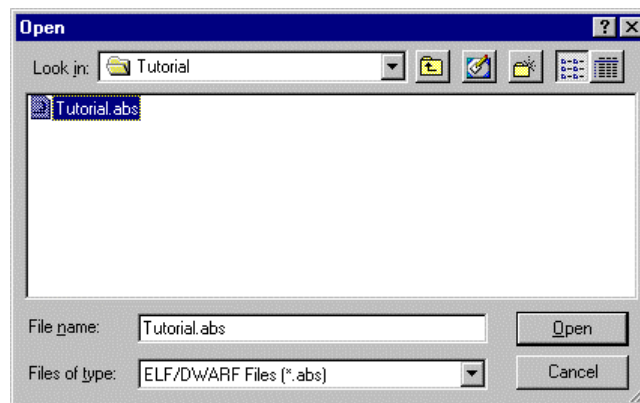


図 4-8 Open ダイアログボックス (オブジェクトファイルの選択)

ファイルがロードされると、以下のダイアログボックスにプログラムコードが書き込まれたメモリエリアに関する情報が表示されます。

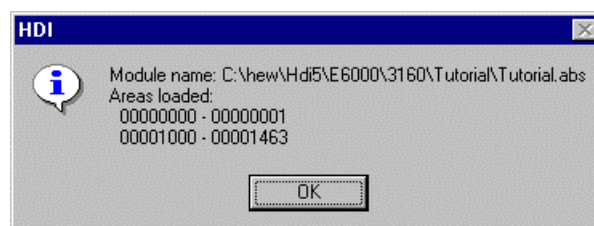



図 4-9 HDI ダイアログボックス

- ・ [OK] をクリックしてください。
プログラムは内蔵 ROM 領域にロードされました。

4.4.2 プログラムリストの表示

HDI では、ソースレベルでプログラムをデバッグできます。

- ・ [View]メニューから[Source...]を選択するか、またはツールバーの [Program Source] ボタン  をクリックしてください。

ロードしたオブジェクトファイルに対応する C ソースファイルを選ぶ必要があります。

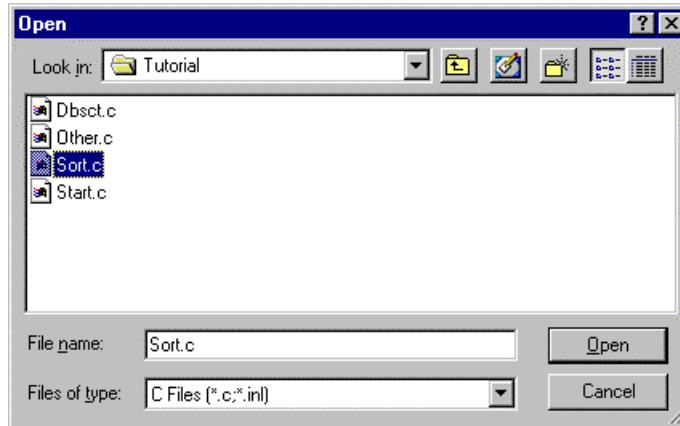


図 4-10 Open ダイアログボックス (ソースファイルの選択)

- ・ Sort.c を選択し、[Open] をクリックしてください。Program ウィンドウが表示されます。

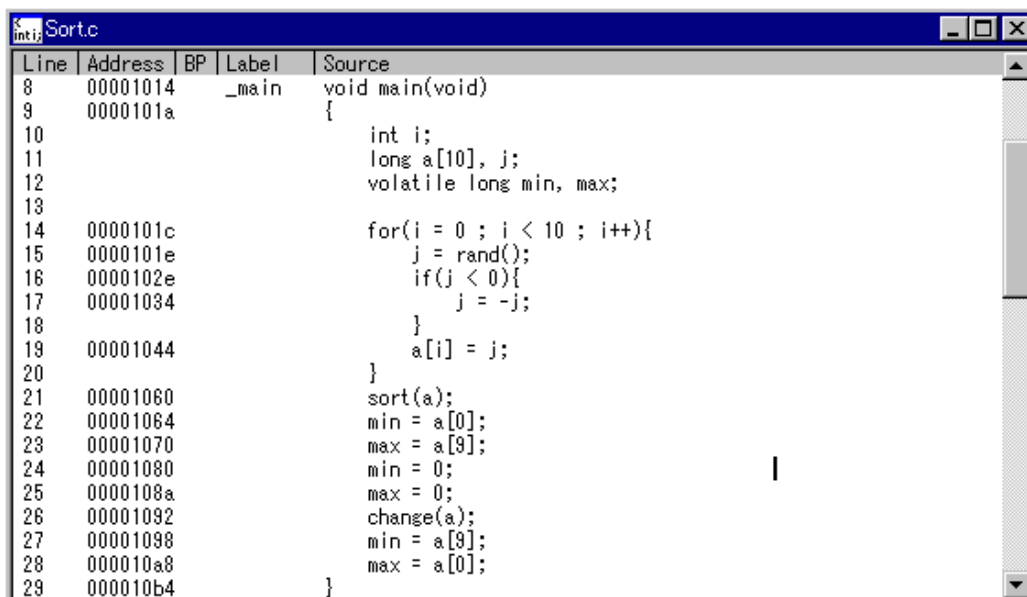


図 4-11 ソースプログラム画面

- ・ 必要ならば、[Setup]メニューの [Customize] サブメニューから [Font...] オプションを選択し、ホスト PC に合ったフォントとサイズを選択してください。

Program ウィンドウを、最初に開いたときはメインプログラムの先頭を示しますが、スクロールバーを使って他の部分を見ることができます。

4.5 ブレークポイントの使い方

最も簡単なデバッグ機能のひとつにプログラムの特定の箇所に達したときに実行を停止できる PC Break があります。この機能を使用することによりプログラムが停止した時の MCU やメモリの状態を調べることができます。

4.5.1 PC Break の設定

Program ウィンドウによって、プログラムのあらゆるポイントに PC Break を簡単に設定できます。たとえば、以下のようにしてアドレス H'1060 に PC Break を設定します。

- ・ H'1060 番地を含むラインの [BP] カラムをダブルクリックしてください。

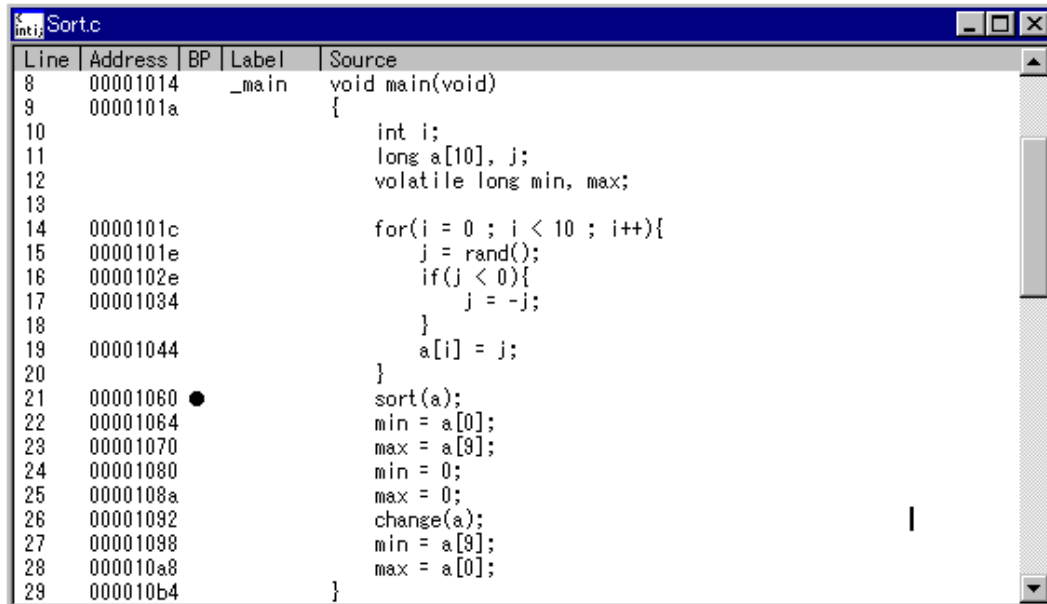


図 4-12 ブレークポイント (PC Break) の設定


その位置に “ Break ” が表示され、そのアドレスに PC Break が設定されたことを示します。また、本章では実行しませんが、さらにダブルクリックしていくことによりイベント間実行測定イベント設定 (“ +Timer ” で測定開始、“ -Timer ” で測定終了)、Point to Point トレース制御の設定 (“ +Trace ” でトレース開始、“ -Trace ” でトレース停止)およびトレースストップの設定 (“ TrStop ” でトレースストップ)ができます。これらはダブルクリックすることにより、以下のような順序でサイクリックに設定できます。

“Blank” “Break” “+Timer” “-Timer” “+Trace” “-Trace” “TrStop”
“Blank” ...

注：“-Timer” および “-Trace” は対応する “+Timer” および “+Trace” が設定されているときのみ選択可能となります。

4.5.2 プログラムの実行

リセットベクタで指定されているアドレスからプログラムを実行するには、

- ・ [Run] メニューから [Reset Go] を選択するか、またはツールバーの [Reset Go] ボタン  をクリックしてください。

プログラムは PC Break を設定したところまで実行し、プログラムが停止した位置を示すために Program ウィンドウ中でステートメントが強調表示されます。

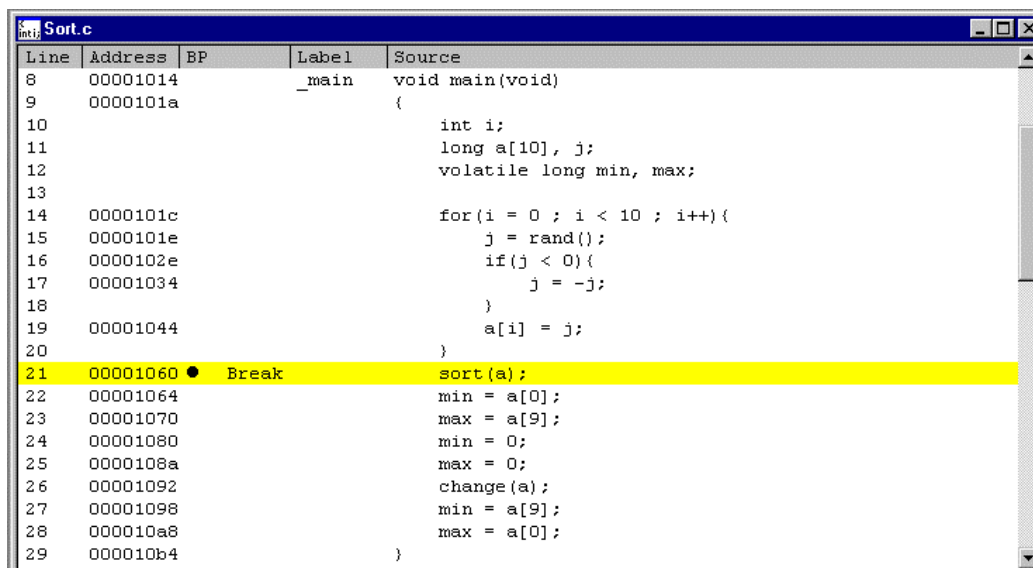



図 4-13 ステートメントの強調表示

[Break = Soft Ware Breakpoint] メッセージがステータスバーに表示され、ブレークの原因を示します。また System Status ウィンドウでも最後のブレークの原因が確認できます。

・ [View]メニューから [Status] を選択するか、またはツールバーの [Status] ボタン  をクリックし System Status ウィンドウを開き Platform シートを選択してください。

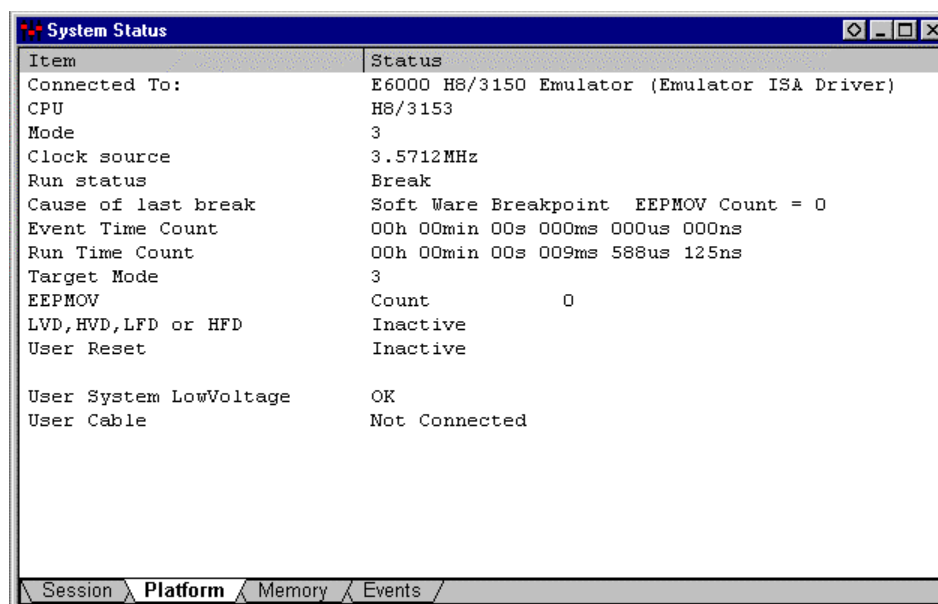



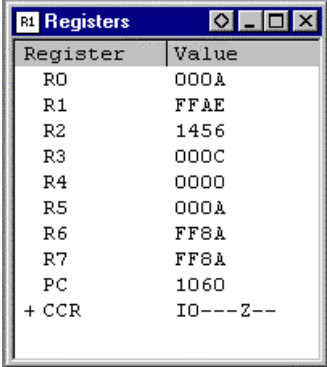
図 4-14 System Status ウィンドウ (Platform シート)

[Cause of last break] のラインは、ブレークの原因が PC Break であることを示しています。また [Run Time Count] のラインは、プログラムが実行してから停止するまでの時間 (実行時間) が 9ms 588.125 μ s だったことを示しています。イベント間実行時間測定 [Event Time Count] (+Timer, -Timer で設定) および実行時間に使用するタイマの分解能は、Configuration ダイアログボックスの [Timer Resolution] により設定されます。長時間の時間計測に分解能 20ns のような小さい値を使用すると、誤差が大きくなることがあります。計測時間の長さに合わせて分解能を調整してください。

4.5.3 レジスタ内容の参照

プログラムが停止している間に、MCU レジスタの内容を参照できます。それらは Registers ウィンドウに表示されます。

- ・ [View]メニューから [Registers] を選択するか、またはツールバーの [CPU Registers] ボタン  をクリックしてください。



Register	Value
R0	000A
R1	FFAE
R2	1456
R3	000C
R4	0000
R5	000A
R6	FF8A
R7	FF8A
PC	1060
+ CCR	I0---Z--

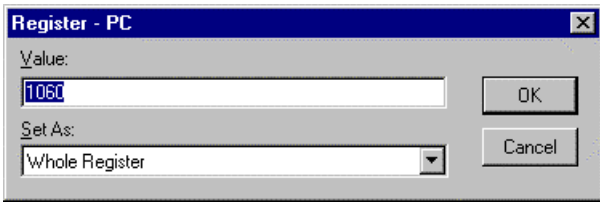
図 4-15 Registers ウィンドウ

プログラムカウンタ(PC)の値は強調表示されたステートメント H'1060 を示しています。(注：その他のレジスタの値は上に示すものとは異なることがあります。)

レジスタの値は Registers ウィンドウで変更できます。

- ・ PC の値を変えるには、Registers ウィンドウで PC に対応する [Value] カラムをダブルクリックしてください。

以下のダイアログボックスによって値を編集できます。



Register - PC

Value:
1060


Set As:
Whole Register

OK Cancel

図 4-16 Register ダイアログボックス


- ・ 値を H'1014 (メインプログラムの先頭アドレス) に変更し、[OK] をクリックしてください。

強調表示されたバーがメインプログラムの先頭に移動し、新しいプログラムカウンタの値を示します。

- ・ [Run]メニューから [Go] を選択するか、またはツールバーの [Go] ボタン  をクリックし、ブレークポイントまでの実行を再開してください。

4.5.4 ブレークポイントの確認

プログラムに設定したすべてのブレークポイントの一覧を Breakpoints ウィンドウで見ることができます。

- ・ [View]メニューから [Breakpoints] を選択するか、またはツールバーの [Breakpoints] ボタン  をクリックしてください。

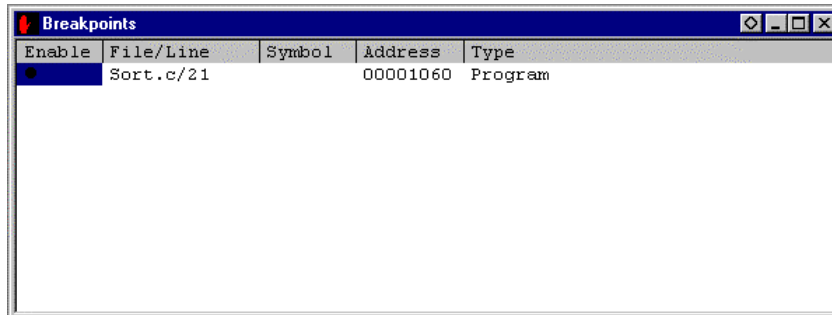


図 4-17 Breakpoints ウィンドウ


Breakpoints ウィンドウによって、ブレークポイントの有効または無効、新しいブレークポイントの設定、およびブレークポイントの削除ができます。

4.6 メモリと変数の表示

メモリ領域の内容を参照することにより、またはプログラム中で使われる変数の値を表示することによって、プログラムの動作をモニタできます。

4.6.1 メモリを表示する

メモリブロックの内容を Memory ウィンドウで見ることができます。
たとえば、Byte で main に対応したメモリを見る場合：

- ・ [View]メニューから [Memory...] を選択するか、またはツールバーの [Memory] ボタン  をクリックしてください。
- ・ [Address] に main を入力し、[Format] を Byte に設定してください。

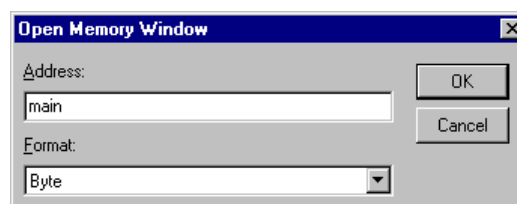


図 4-18 Open Memory Window の設定

- ・ [OK] をクリックして、指定されたメモリ領域を示す Memory ウィンドウを開くことにより、メモリブロックの内容が確認できます。

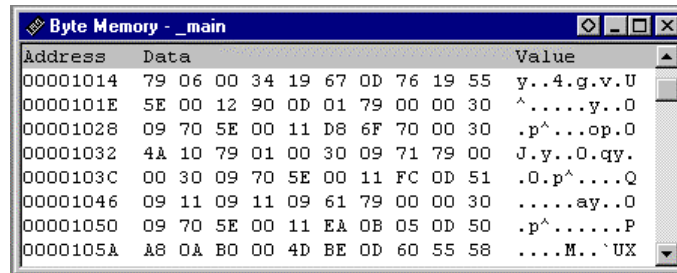


図 4-19 Memory ウィンドウ(Byte)

4.6.2 変数を表示する

プログラムをステップ処理するとき、プログラムで使用される変数の値を見ることができ、期待したようにそれらが変化することを確認できます。

たとえば以下の手順で、プログラムの始めに宣言した long 型の配列変数”a”を見ることができます。

- Program ウィンドウに表示されている配列変数”a”の左側をクリックし、カーソルを置いてください。
- マウスの右ボタンで Program ウィンドウをクリックし、ポップアップメニューより、[Add Watch] を選択してください。

Watch ウィンドウに変数が表示されます。

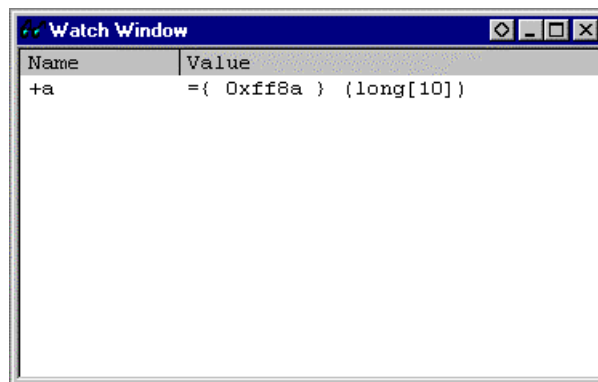



図 4-20 Watch ウィンドウ (変数追加後)

Watch ウィンドウのシンボル”a”の左にある”+”をダブルクリックし、シンボルを拡張して各配列の要素を見ることができます。

必要ならば、[Setup] メニューの [Radix] サブメニューから [Decimal] を選択するか、あるいはツールバーの [Radix = Decimal] ボタン  を押し、10 進数表示にしてください。

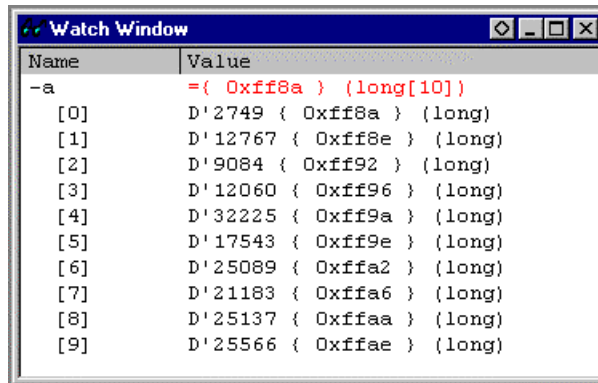


図 4-21 Watch ウィンドウ (シンボル拡張)

また、変数名を指定して、Watch ウィンドウに変数を追加することもできます。

- マウスの右ボタンで Watch ウィンドウをクリックし、ポップアップメニューから [Add Watch...] を選択してください。
- 変数名"max"を入力し、[OK] ボタンをクリックしてください。

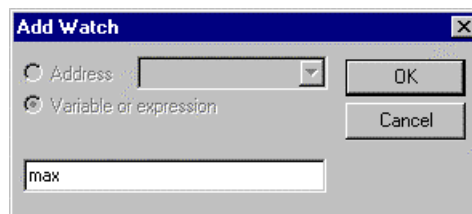


図 4-22 Add Watch ダイアログボックス

Watch ウィンドウに volatile long 型の変数"max"が追加されます。

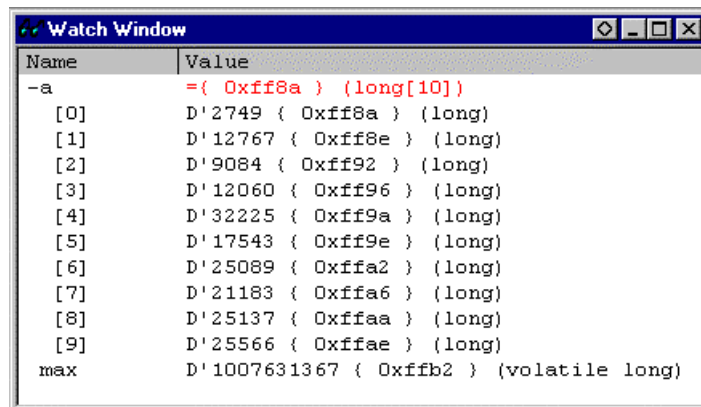


図 4-23 Watch ウィンドウ (変数の追加)

4.7 プログラムのステップ実行


E6000 エミュレータは、プログラムのシングルステップにおけるオプションを備えており、命令やステートメントを一度に実行します。以下に示すようなステップオプションがあります。

表 4-4 プログラムステップオプション

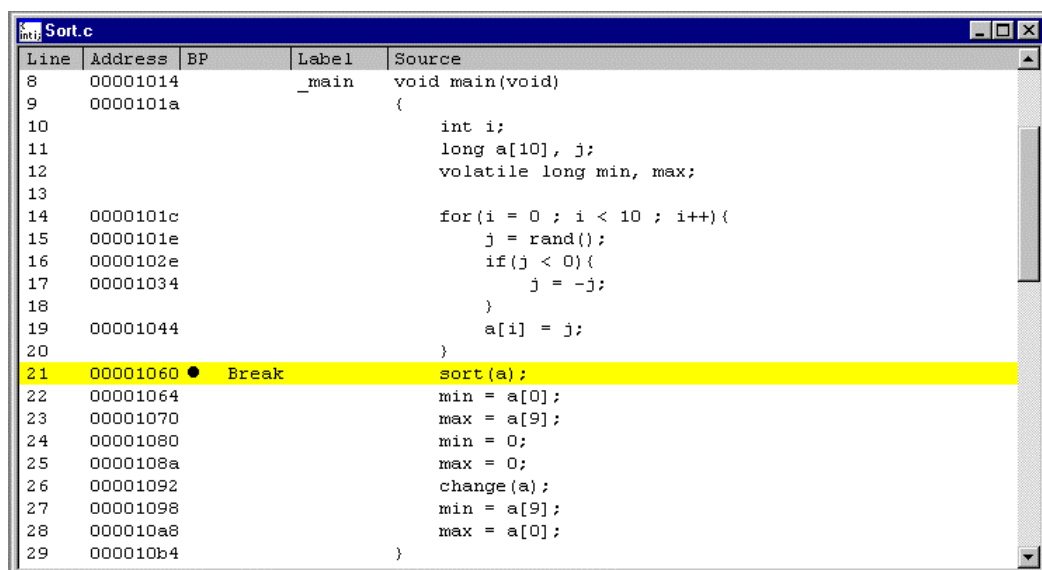
コマンド	説明
Step In	各ステートメントを実行します (関数内のステートメントを含む)。
Step Over	呼び出された関数の全ステートメントを実行します。
Step Out	関数を抜け出し、関数を呼び出したプログラムにおける次のステートメントで停止します。
Step...	指定したステートメント数ステップ実行します。

4.7.1 シングルステップ

- ・ H'1060 に PC Break が設定されていることを確認してください。
- ・ 次に [Run] メニューから [Reset Go] を選択するか、あるいはツールバーの [Reset Go]

ボタン  を押してください。

設定した PC Break により H'1060 で停止し、sort(a); のステートメントが強調表示されます。




```

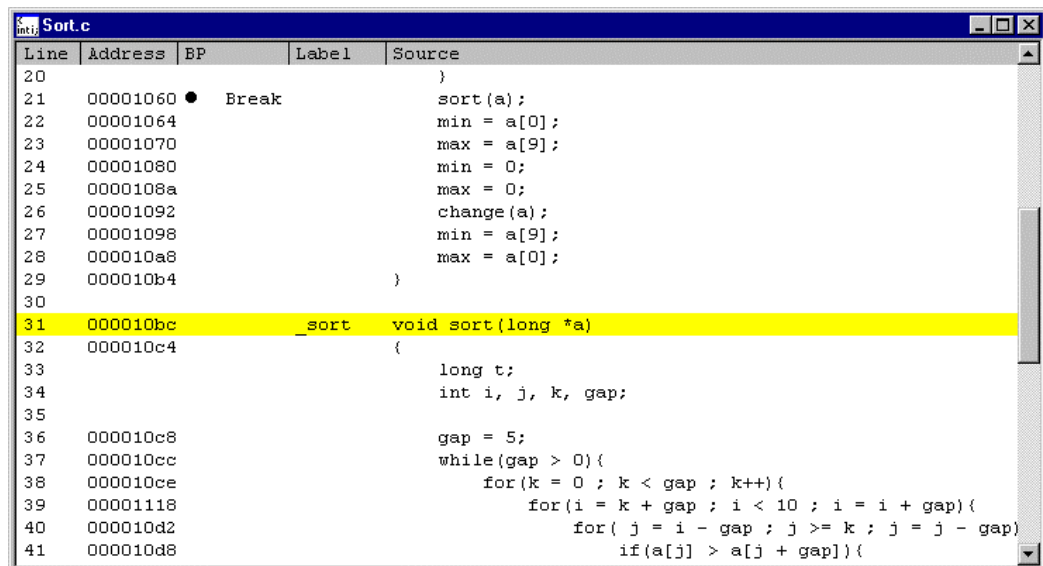
Sort.c
Line Address BP Label Source
8 00001014 _main void main(void)
9 0000101a {
10 int i;
11 long a[10], j;
12 volatile long min, max;
13
14 0000101c for(i = 0 ; i < 10 ; i++){
15 0000101e j = rand();
16 0000102e if(j < 0){
17 00001034 j = -j;
18 }
19 00001044 a[i] = j;
20 }
21 00001060 ● Break sort(a);
22 00001064 min = a[0];
23 00001070 max = a[9];
24 00001080 min = 0;
25 0000108a max = 0;
26 00001092 change(a);
27 00001098 min = a[9];
28 000010a8 max = a[0];
29 000010b4 }

```

図 4-24 Reset Go 実行後の Program ウィンドウ


- ・ sort 中のステートメントをステップ実行するために [Run] メニューから [Step In] を選択するか、またはツールバーの [Step In] ボタン  をクリックしてください。

4.チュートリアル

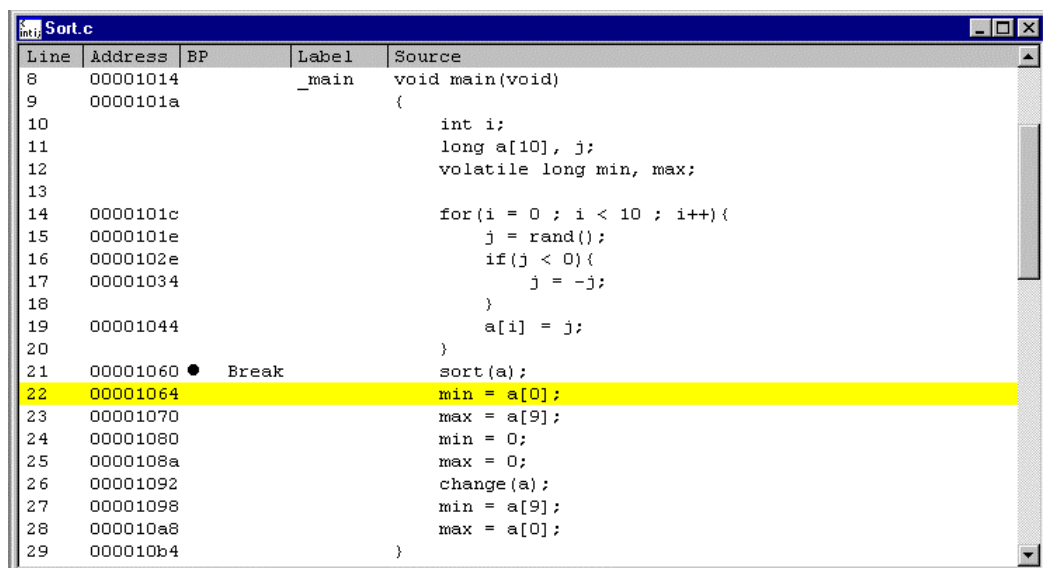


```
Sort.c
Line Address BP Label Source
20      00001060      Break      sort(a);
21      00001064      min = a[0];
22      00001070      max = a[9];
23      00001080      min = 0;
24      0000108a      max = 0;
25      00001092      change(a);
26      00001098      min = a[9];
27      000010a8      max = a[0];
28      000010b4      }
29
30
31      000010bc      _sort      void sort(long *a)
32      000010c4      {
33      long t;
34      int i, j, k, gap;
35
36      gap = 5;
37      while(gap > 0){
38          for(k = 0 ; k < gap ; k++){
39              for(i = k + gap ; i < 10 ; i = i + gap){
40                  for( j = i - gap ; j >= k ; j = j - gap)
41                      if(a[j] > a[j + gap]){
```

図 4-25 Step In 実行後の Program ウィンドウ (1)

- ・ [Run] メニューから [Step Out] を選択するか、またはツールバーの [Step Out] ボタン  をクリックして関数を抜け出し、メイン関数内の次のステートメントに戻ってください。

アドレス H'1064 が強調表示され、関数から抜け出したことが判ります。



```
Sort.c
Line Address BP Label Source
8      00001014      _main      void main(void)
9      0000101a      {
10     int i;
11     long a[10], j;
12     volatile long min, max;
13
14     for(i = 0 ; i < 10 ; i++){
15         j = rand();
16         if(j < 0){
17             j = -j;
18         }
19         a[i] = j;
20     }
21     00001060      Break      sort(a);
22     00001064      min = a[0];
23     00001070      max = a[9];
24     00001080      min = 0;
25     0000108a      max = 0;
26     00001092      change(a);
27     00001098      min = a[9];
28     000010a8      max = a[0];
29     000010b4      }
```

図 4-26 Step Out 実行後の Program ウィンドウ

- ・ さらに [Step In] コマンドによって change ファンクションコールまでプログラムを実行してください。

注：ステップ実行時 C/C++ ライブラリ関数または実行時ルーチン内にステップインした場合、自動的に Disassembly ウィンドウが開きます。この状態でステップ実行を行なった場合、アセンブラレベルでステップ実行が行なわれます。C/C++ 言語ソースレベルでステップ実行する場合は、[Step Out] コマンドにより C/C++ ライブラリ関数または実行時ルーチンを抜けた後、Disassembly ウィンドウを閉じて下さい。


Line	Address	BP	Label	Source
8	00001014		_main	void main(void)
9	0000101a			{
10				int i;
11				long a[10], j;
12				volatile long min, max;
13				
14	0000101c			for(i = 0 ; i < 10 ; i++){
15	0000101e			j = rand();
16	0000102e			if(j < 0){
17	00001034			j = -j;
18				}
19	00001044			a[i] = j;
20				}
21	00001060	●	Break	sort(a);
22	00001064			min = a[0];
23	00001070			max = a[9];
24	00001080			min = 0;
25	0000108a			max = 0;
26	00001092			change(a);
27	00001098			min = a[9];
28	000010a8			max = a[0];
29	000010b4			}

図 4-27 Step In 実行後の Program ウィンドウ (2)

4.7.2 関数全体のステップ実行

[Step Over] コマンドは、関数本体をシングルステップすることなく実行し、メインプログラムの中の次のステートメントで停止します。

- ・ [Run] メニューから [Step Over] を選択するか、またはツールバーの [Step Over]

ボタン  をクリックしてください。

プログラムは change 関数を実行し、次のアドレス H'1098 で停止します。


Line	Address	BP	Label	Source
8	00001014		_main	void main(void)
9	0000101a			{
10				int i;
11				long a[10], j;
12				volatile long min, max;
13				
14	0000101c			for(i = 0 ; i < 10 ; i++){
15	0000101e			j = rand();
16	0000102e			if(j < 0){
17	00001034			j = -j;
18				}
19	00001044			a[i] = j;
20				}
21	00001060	●	Break	sort(a);
22	00001064			min = a[0];
23	00001070			max = a[9];
24	00001080			min = 0;
25	0000108a			max = 0;
26	00001092			change(a);
27	00001098			min = a[9];
28	000010a8			max = a[0];
29	000010b4			}

図 4-28 Step Over 実行後の Program ウィンドウ

4.7.3 ローカル変数の表示

Locals ウィンドウを使って関数内のローカル変数を表示させることができます。例として、main 関数のローカル変数を調べます。

この関数は、5 つのローカル変数 a, j, i, min, max を宣言しています。

- ・ [View] メニューから [Locals] を選択するか、またはツールバーの [Locals] ボタン  をクリックしてください。

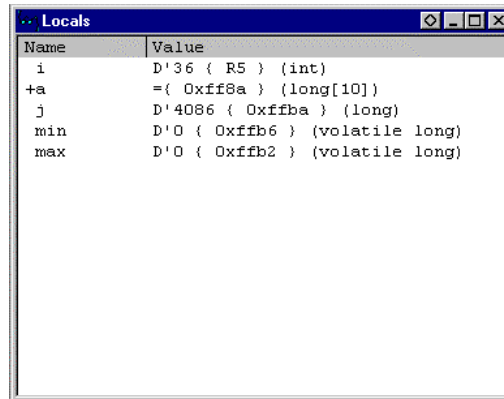



図 4-29 Locals ウィンドウ

ローカル変数が存在しない場合は Locals ウィンドウに何も表示されません。

・ [Run] メニューから [Step In] を選択するか、またはツールバーの [Step] ボタン  をクリックして、1ステップ実行してください。

変数”min”の内容が変更され、その値が表示されます。

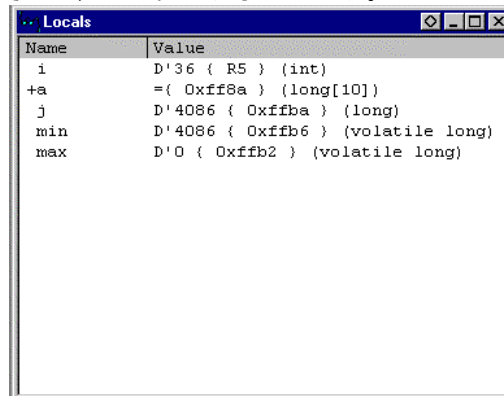


図 4-30 Locals ウィンドウ (変数”min”内容変更後)

- ・ Locals ウィンドウのシンボル”a”の左にある”+”をダブルクリックし、シンボルを拡張して各配列の要素を表示させてください。
- ・ sort 関数実行前の配列変数”a”の要素を参照し、ランダムデータが降順にソートされていることを確認してください。

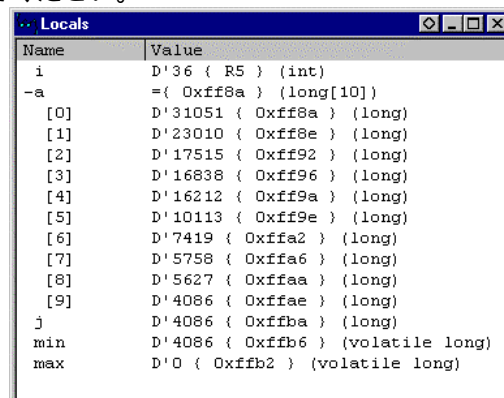


図 4-31 Locals ウィンドウ (配列変数”a”ソート後)

4.8 イベント検出システム(Complex Event System)の使用法

本チュートリアルでは、Memory ウィンドウでメモリ領域の内容を見ること、あるいは Watch ウィンドウおよび Locals ウィンドウで変数の値を見ることによって、プログラムの動作をモニタしてきました。

しかしプログラムの動作は非常に複雑なため、メモリ領域をモニタしたり、変数を見たりできないことがあります。そこで、E6000 エミュレータのイベント検出システムを使用することにより、たとえば、プログラムが 5 回 H'1108 をアクセスした時を検出することができます。

4.8.1 イベント検出システムによる Event の設定

イベント検出システムを使用した Event を設定して、以下のようにプログラムの一部をモニタしてください。



- ・ [Setup] メニューの [Radix] サブメニューから [Hexadecimal] を選択するか、あるいはツールバーの [Radix = Hex] ボタン  を押し、16 進数表示にしてください。これにより 16 進数入力時に基数の接頭部(H')を省略することができます。
 - ・ [View] メニューから [Breakpoints] を選択するか、またはツールバーの [Breakpoints] ボタン  をクリックして、Breakpoints ウィンドウを表示してください。
 - ・ 新しいブレークポイントを設定するため Breakpoints ウィンドウ内でマウスの右ボタンをクリックし [Add...] を選択してください。
- 以下のダイアログボックスが現われ、ブレークポイントの属性を設定できます。
- ・ [Type] 選択を [Event] にし、条件として [Address Lo] ボックスにアドレス H'1108 を入力してください。



図 4-32 ブレークポイントの追加

- ・ [Action] タブをクリックして [Action] パネルを表示してください。

- 5回アクセスした時にブレークさせるため、[**Required number of event occurrences**] エディットボックスに5を入力してください。

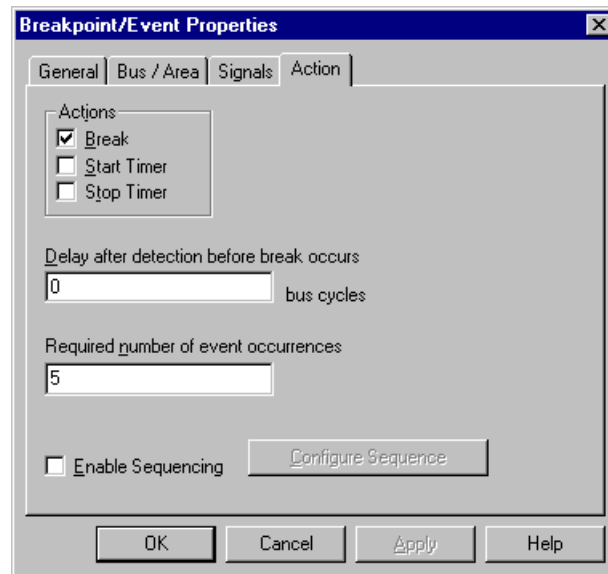


図 4-33 ブレークポイントの追加（回数指定）

- [**OK**] をクリックしてブレークポイントを設定してください。
これによって、アドレス H'1108 が5回アクセス（読み出しまたは書き込み）されたときにブレークします。
Breakpoints ウィンドウは、設定された新しい Event を表示します。

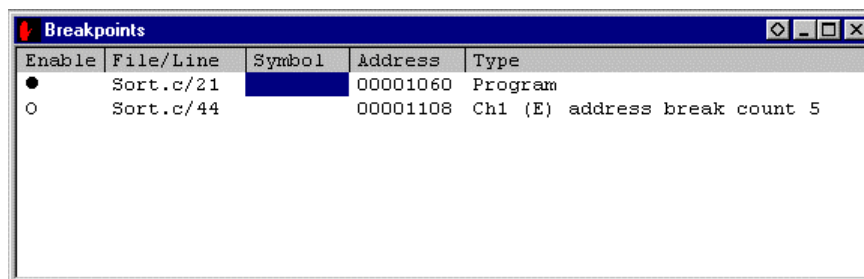


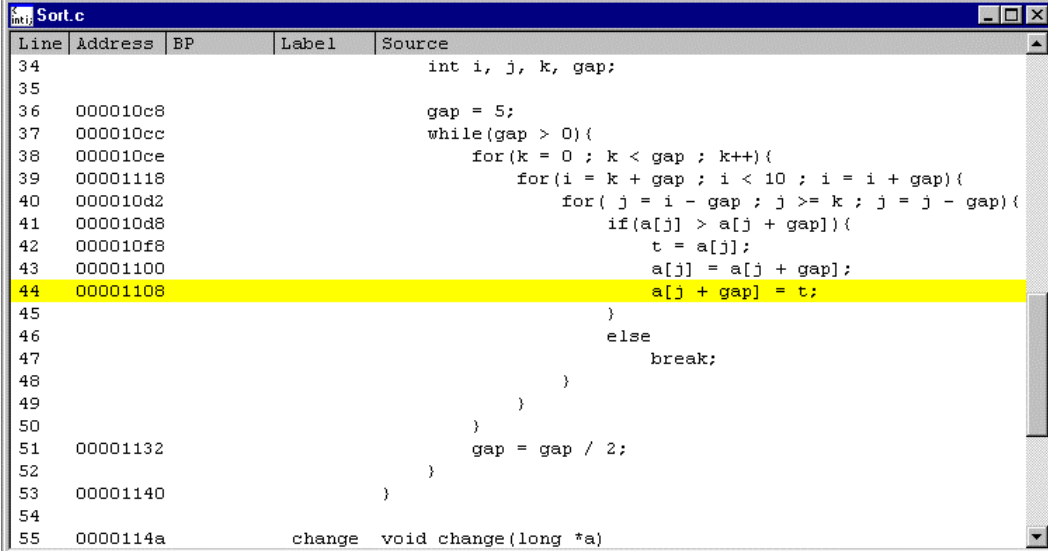


図 4-34 Breakpoints ウィンドウ（追加後）

- [**Run**] メニューから [**Reset Go**] を選択するか、あるいはツールバーの [**Reset Go**] ボタン  をクリックしてください。
アドレス H'1060 に設定された PC Break で停止します。
- さらに [**Run**] メニューから [**Go**] を選択するか、あるいはツールバーの [**Go**] ボタン  をクリックして、プログラムを現在の位置から実行してください。
アドレス H'1108 への5回のアクセスにより実行が停止します。



```
Sort.c
Line Address BP Label Source
34      int i, j, k, gap;
35
36 000010c8      gap = 5;
37 000010cc      while(gap > 0){
38 000010ce          for(k = 0 ; k < gap ; k++){
39 00001118              for(i = k + gap ; i < 10 ; i = i + gap){
40 000010d2                  for( j = i - gap ; j >= k ; j = j - gap){
41 000010d8                      if(a[j] > a[j + gap]){
42 000010f8                          t = a[j];
43 00001100                              a[j] = a[j + gap];
44 00001108                                  a[j + gap] = t;
45                                          }
46                                          else
47                                          break;
48                                  }
49                              }
50                          }
51                      gap = gap / 2;
52                  }
53              }
54
55 0000114a      change void change(long *a)
```

図 4-35 Event ブレークポイントによるプログラムの停止

また、ステータスバーには“Break = Event Break”と表示し、Event に設定した条件の一致によってブレークが発生したことを示します。

4.9 トレースバッファの使い方

MCUの動作を確認するため、指定されたイベントの直前までのMCUサイクルはトレースバッファに記録されています。

4.9.1 トレースバッファの表示

プログラムのアクセスアドレスを指定し、トレースバッファ内のMCUサイクルを調べることによって、どのようなアクセスが起こったかを知ることができます。

- ・ [View]メニューから [Trace] を選択するか、あるいはツールバーの [Trace] ボタン



をクリックして、Trace ウィンドウを開いてください。

必要ならば、最後の数サイクルが見えるようにウィンドウをスクロールダウンしてください。Trace ウィンドウが以下のように表示されます。

Cycle	Address	Label	Code	Data	R/W	Area	Status	Clock	Probes	IRQ1	IRQ2	Source
-00015	00ff9a			0000	RD	IN-RAM	CPU	2	1111	HIGH	HIGH	
-00014	0011f0		MOV.W	6f02	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
-00013	00ff92			0000	WR	IN-RAM	CPU	2	1111	HIGH	HIGH	
-00012	0011f2			0002	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
-00011	0011f4		MOV.W	6f92	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
-00010	00ff9c			0ff6	RD	IN-RAM	CPU	2	1111	HIGH	HIGH	
-00009	0011f6			0002	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
-00008	0011f8		MOV.W	6d72	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
-00007	00ff94			0ff6	WR	IN-RAM	CPU	2	1111	HIGH	HIGH	
-00006	0011fa		RTS	5470	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
-00005	00ff78			0000	RD	IN-RAM	CPU	2	1111	HIGH	HIGH	
-00004	0011fc	\$NEGL\$3		6df3	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
-00003	00ff7a			1108	RD	IN-RAM	CPU	2	1111	HIGH	HIGH	
-00002	001108		MOV.W	0d70	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
-00001	00110a			0d41	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
+00000	00110c			5e00	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	

図 4-36 Trace ウィンドウ

- ・ 必要ならば、タイトルバーのすぐ下のラベルの横にあるカラムディバイダをドラッグして、カラムの幅を調節してください。

Cycle カラムの -00002 を見ると、アドレス H'1108 がアクセスされていることを確認できます。

4.9.2 トレースフィルタの設定

現在 Trace ウィンドウは、すべての MCU サイクルを表示しています。

- マウスの右ボタンで Trace ウィンドウをクリックし、ポップアップメニューより [Filter...] を選択して、Trace Filter ダイアログボックスを表示してください。これによって、トレースバッファに表示されるサイクルを限定するためのフィルタ条件を設定できます
- 必要ならば、[General] をクリックして、[General] パネルを表示してください。
- [Type] セクションで [Pattern] を選択してください。
- [Address] セクションで、[Address] をクリックし、[Address Lo] フィールドに H'1108 と入力してください。

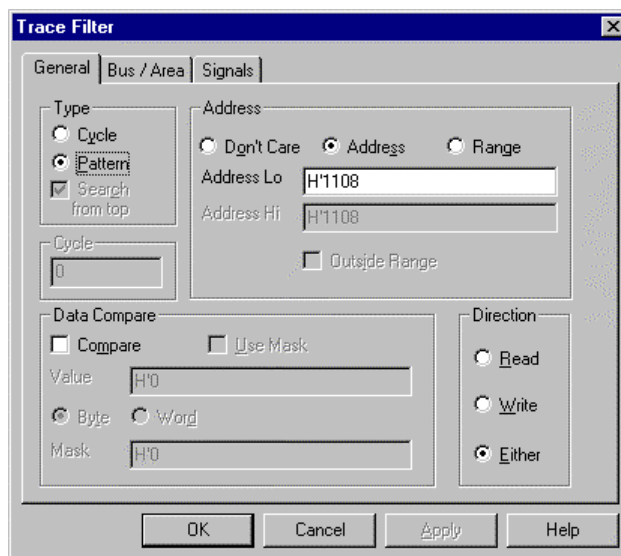


図 4-37 Trace Filter ダイアログボックス

- [Bus / Area] をクリックし、[Bus / Area] パネルを表示してください。
- [Bus State] を [CPU Prefetch] に設定してください。

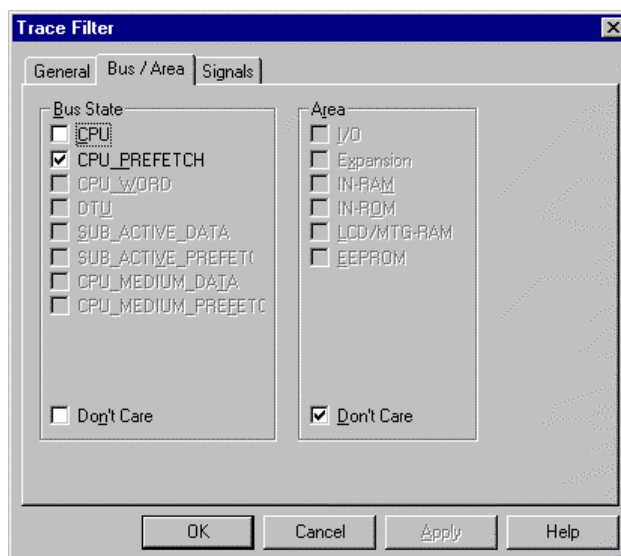
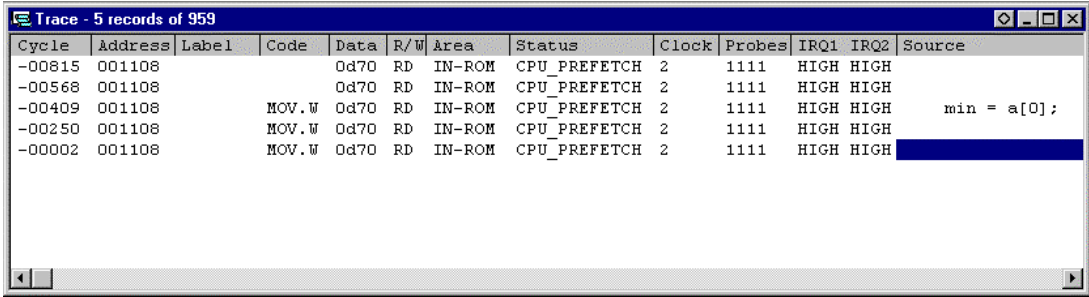


図 4-38 Bus / Area の設定

- [OK] をクリックして、トレースフィルタを保存してください。

Trace ウィンドウには、MCU が H'1108 番地をアクセスしたサイクルだけが表示されます。これにより H'1108 への 5 回のアクセスによりプログラムが停止したことがわかります。



Cycle	Address	Label	Code	Data	R/W	Area	Status	Clock	Probes	IRQ1	IRQ2	Source
-00815	001108			0d70	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
-00568	001108			0d70	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
-00409	001108		MOV.W	0d70	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	min = a[0];
-00250	001108		MOV.W	0d70	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	
-00002	001108		MOV.W	0d70	RD	IN-ROM	CPU_PREFETCH	2	1111	HIGH	HIGH	

図 4-39 Trace ウィンドウ (トレースフィルタ指定)

4.10 プログラム実行時間効率の計測

HDI では Performance Analysis 機能を使用することにより、プログラムが効率よく実行されているかを計測することができます。計測結果はヒストグラムまたはパーセント表示で確認できます。

4.10.1 計測条件の選択

計測を行う為に条件を選択してください。

- ・ [View] メニューから [Performance Analysis] を選択するか、あるいはツールバーの [PA] ボタン  をクリックして、Performance Analysis ダイアログボックスを開いてください。
- ・ [Conditions] ボタンをクリックして、Performance Analysis Conditions ダイアログボックスを開いてください。
- ・ [Performance Analysis Conditions] の "No 1" をクリックした後、[Edit] ボタンをクリックして、Performance Analysis Properties ダイアログボックスを開いてください。

以下のダイアログボックスが現われ、計測条件を設定できます。

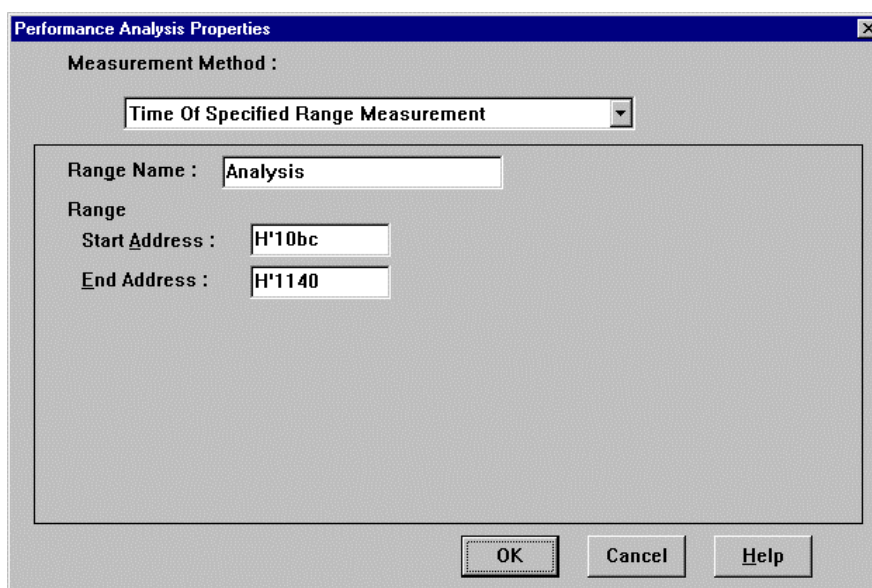


図 4-40 計測条件の選択

- ・ 指定範囲内でのプログラム実行時間効率を計測するため [**Measurement Method**] から [**Time Of Specified Range Measurement**] を選択してください。
- ・ [**Range Name**] に、"Analysis" を入力してください。
- ・ 指定範囲として [**Start Address**] にアドレス H'10bc、 [**End Address**] にアドレス H'1140 を入力してください。
- ・ [**OK**] をクリックして、計測条件を設定してください。
これにより、"No 1"に対する計測条件が設定されました。

Performance Analysis Conditions ダイアログボックスは、設定条件を表示しています。

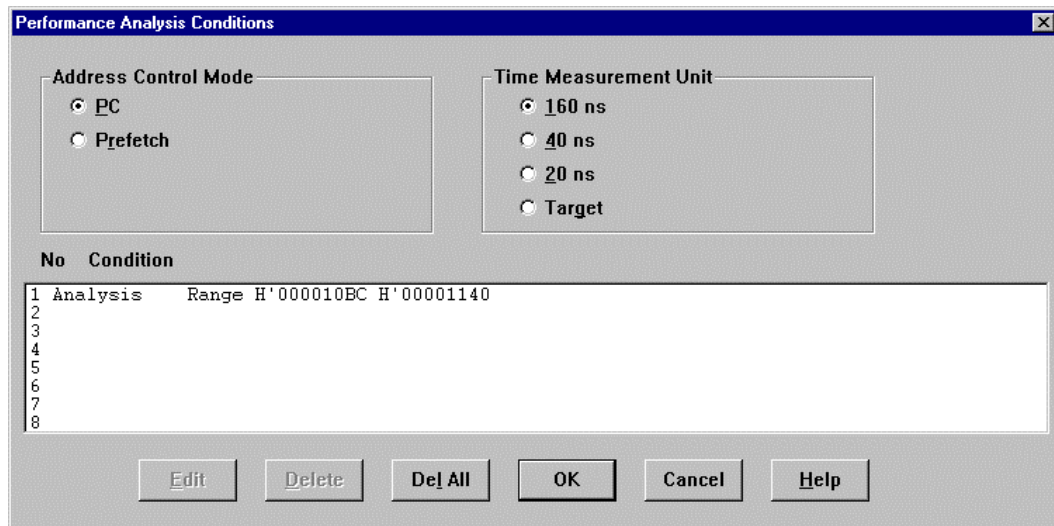




図 4-41 計測条件の表示

- ・ [**OK**] をクリックして、計測条件を設定してください。
これにより、プログラムがアドレス H'10bc と H'1140 の範囲を実行した時の実行効率が計測できます。
- ・ [**Close**] をクリックして、Performance Analysis ダイアログボックスを閉じてください。
- ・ Program ウィンドウから H'10b4 番地を含むラインの [**BP**] カラムをダブルクリックして PC Break を設定してください。
- ・ [**Run**] メニューから [**Reset Go**] を選択するか、あるいはツールバーの [**Reset Go**] ボタン  をクリックして、プログラムを最初の位置から実行してください。
アドレス H'10b4 で実行が停止します。

4.10.2 計測結果の表示

プログラム実行時間効率の計測結果をヒストグラムまたはパーセントで表示します。

- ・ [View] メニューから [Performance Analysis] を選択するか、あるいはツールバーの [PA] ボタン  をクリックして、Performance Analysis ダイアログボックスを開いてください。

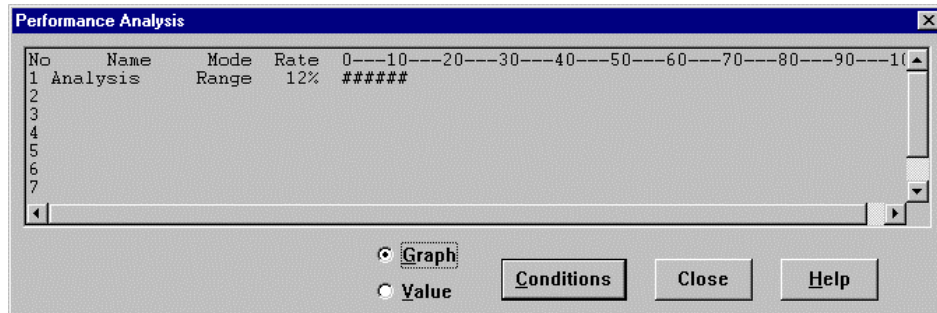


図 4-42 計測結果の表示 (1)

プログラム実行時間効率の計測結果をヒストグラムとパーセントで表示します。

- ・ [Value] をクリックしてください。

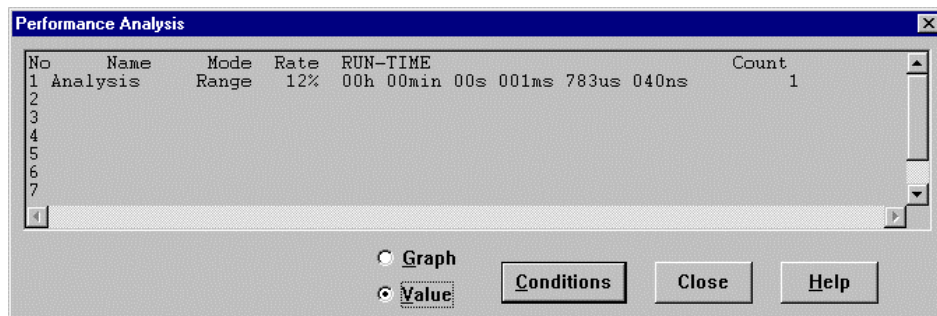



図 4-43 計測結果の表示 (2)

プログラム実行時間効率の計測結果をパーセントと計測時間で表示します。

4.11 スタックトレース機能

スタックトレース機能を使用すると、ユーザプログラム停止時に関数の呼び出し履歴を確認できます。

- ・ Program ウィンドウから H*10c8 番地を含むラインの [BP] カラムをダブルクリックして PC Break を設定してください。
- ・ [Run] メニューから [Reset Go] を選択するか、あるいはツールバーの [Reset Go] ボタン  をクリックして、プログラムを最初の位置から実行してください。設定した PC Break によりアドレス H*10c8 で実行が停止します。
- ・ [View]メニューから [Stack Trace]を選択し [Stack Trace] ウィンドウを開いてください。

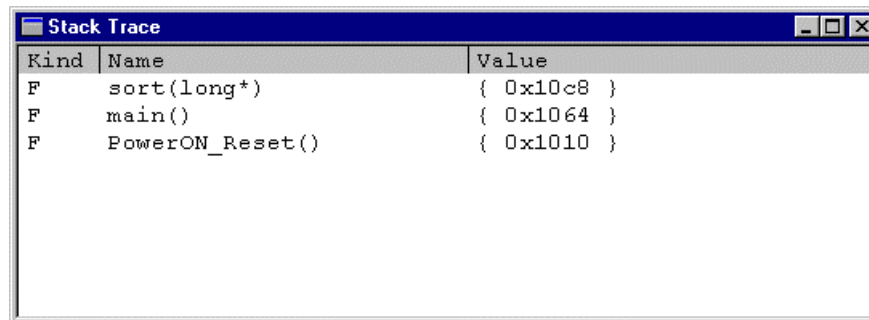


図 4-44 Stack Trace ウィンドウ

現在 PC が sort 関数内にあり、main 関数から呼び出されていることがわかります。

注：本機能は、Dwarf2 形式のデバッグ情報を持ったロードモジュールをロードした場合のみ使用できます。

これまで説明いたしました各機能の詳細とその他の機能につきましては、オンラインヘルプをご覧ください。オンラインヘルプは、各ウィンドウで [Help] ボタンまたは [F1] キーを押すと表示されます。

4.12 セッションの保存

終了する前に、次回のデバッグセッションで同じ E6000 エミュレータと HDI コンフィグレーションを使用して再開できるように、セッションを保存しておくとい良いでしょう。

- ・ [File] メニューから [Save Session] を選択してください。
- ・ [File] メニューから [Exit] を選択して、HDI を終了してください。

4.13 さてつぎは？

このチュートリアルは、E6000 エミュレータのいくつかの特長と、HDI の使用方法を紹介しました。E6000 エミュレータで提供される機能を組み合わせることによって、非常に高度なデバッグを行うことができます。それによって、ハードウェアとソフトウェアの問題が発生する条件を正確に分離し、識別することにより、それらの問題点を効果的に調査することができます。

HDI の使用方法に関する詳細については、別に発行されている「日立デバッグングインタフェースユーザズマニュアル」を参照してください。

5 E6000 H8/3160 HDI の機能

本章は、E6000 H8/3160 専用の HDI の特徴に関する情報について述べます。あらゆるターゲットに共通する HDI の一般的な特長に関しては、別に発行されている「日立デバッグインタフェースユーザズマニュアル」を参照してください。以下に HDI メニューと「日立デバッグインタフェースユーザズマニュアル」(HDI マニュアル) および本マニュアルに記載する項目の対応表を示します。

表 5-1 HDI のメニューとマニュアルの対応表

メニューバー	プルダウンメニュー	HDI マニュアル	本マニュアル
File Menu	New Session...		4.2.1
	Load Session...		
	Save Session		4.12
	Save Session As...		
	Load Program...		4.4.1
	Initialize		
	Exit		
Edit Menu	Cut		
	Copy		
	Paste		
	Find...		
	Evaluate...		
View Menu	Breakpoints		4.5.4, 4.8.1, 5.3
	Command Line		
	Disassembly...		
	I/O Area		
	Labels		
	Locals		4.7.3
	Memory...		4.6.1
	Performance Analysis		4.10.1, 4.10.2, 5.6
	Registers		4.5.3
	Source...		4.4.2
	Status		4.3.2, 4.5.2, 5.2
	Trace		4.9.1, 5.4
	Watch		4.6.2
Stack Trace		4.11	
Run Menu	Reset CPU		
	Go		4.5.3, 4.8.1
	Reset Go		4.5.2, 4.8.1
	Go To Cursor		
	Set PC To Cursor		
	Run...		
	Step In		4.7.1, 4.7.3
	Step Over		4.7.2, 4.7.3
	Step Out		4.7.1
	Step...		
	Halt		

表 5-1 HDI のメニューとマニュアルの対応表 (つづき)

メニューバー	プルダウンメニュー	HDIマニュアル	本マニュアル
Memory Menu	Refresh		
	Load...		
	Save...		
	Verify...		
	Test...		
	Fill...		
	Copy...		
	Compare...		
	Configure Map...		4.3.2,5.5
	Configure Overlay...		
Setup Menu	Status bar		4.2.1
	Options...		
	Radix		4.6.2
	Customize		4.4.2
	Configure Platform...		4.3.1,5.1
Window Menu	Cascade		
	Tile		
	Arrange Icons		
	Close All		
Help Menu	Index		
	Using Help		
	Search for Help on		
	About HDI		

5.1 Configuration ダイアログボックス

このダイアログボックスでは E6000 H8/3160 エミュレータの各種パラメータを設定します。

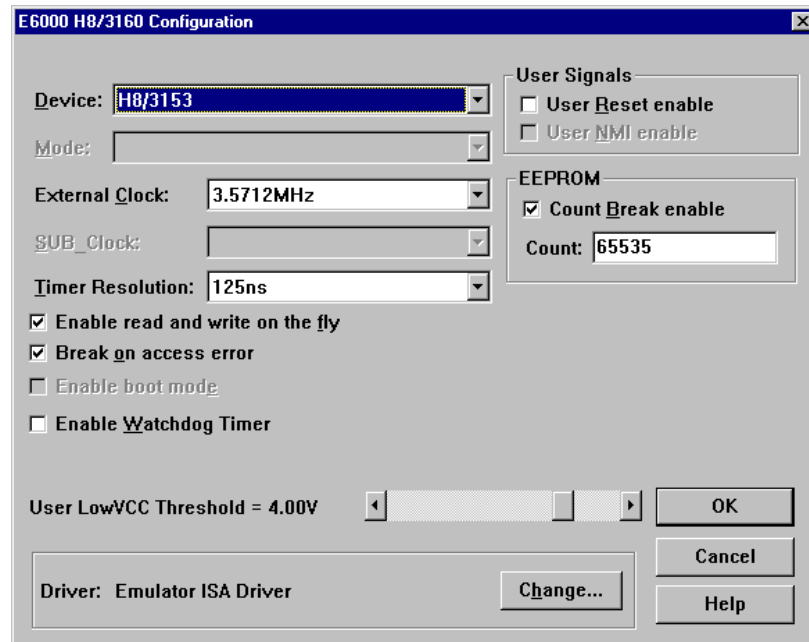


図 5-1 Configuration ダイアログボックス

Device	エミュレーションする H8/3160 ファミリの MCU を指定します。各デバイスによって ROM, RAM, EEPROM 容量が異なります。詳細に関してはハードウェアマニュアルを参照してください。
External Clock	CPU クロック速度を指定します。3.5712MHz、4.9152MHz、7.1424MHz、9.8304MHz、ターゲットのオプションがあります。
Timer Resolution	実行時間の計測に使用する最小時間を指定します。20ns、125ns、250ns、500ns、1 μ s、2 μ s、4 μ s、8 μ s、16 μ s のいずれかの値に指定できます。
Enable read and write on the fly	このチェックボックスをチェックすると、プログラム実行中にメモリにアクセスすることが可能です。ただし、メモリアクセスが行われている間、プログラム実行は停止するため、リアルタイム性はなくなります。
Break on access error	このチェックボックスをチェックすると、プログラムでアクセス禁止エリアにアクセス、または書き込み禁止エリアに書き込みが生じると、ブレーク（ユーザプログラム停止）します。 注意：[Break on access error]チェックボックスのチェックの有無にかかわらず EEPMOV 命令のディスティネーションに ROM を指定した場合は ROM の内容が書き換わりますのでご注意ください。
Enable Watchdog Timer	このチェックボックスをチェックすると Watchdog Timer は、有効になります。
User LowVCC Threshold	ユーザシステム電圧レベルを指定します。指定された値よりユーザ VCC が低くなると、System Status ウィンドウによって知らされます。
User Signals	このチェックボックスをチェックすると、ユーザシステムからのリセット信号を有効にします。

EEPROM	このボックスの[Count Break enable]チェックボックスをチェックすると、EEPMOV 命令実行回数が、Count 値と一致した場合に Break が発生します。 EEPMOV 命令実行回数は、System Status ウィンドウの Platform シートの[EEPMOV Count]に表示され、Go コマンドを実行するごとにクリアされません。
Driver... Change	E6000 エミュレータのドライバを指定します。本ダイアログボックスを開くと設定を変更しない場合でも HDI を初期化します。

5.1.1 ドライバ詳細

E6000 エミュレータへのインタフェースで使用する、ドライバのソフトウェアを選択できます。このダイアログボックスは、終了時に E6000 を再初期化するため、キャンセルすることができません。

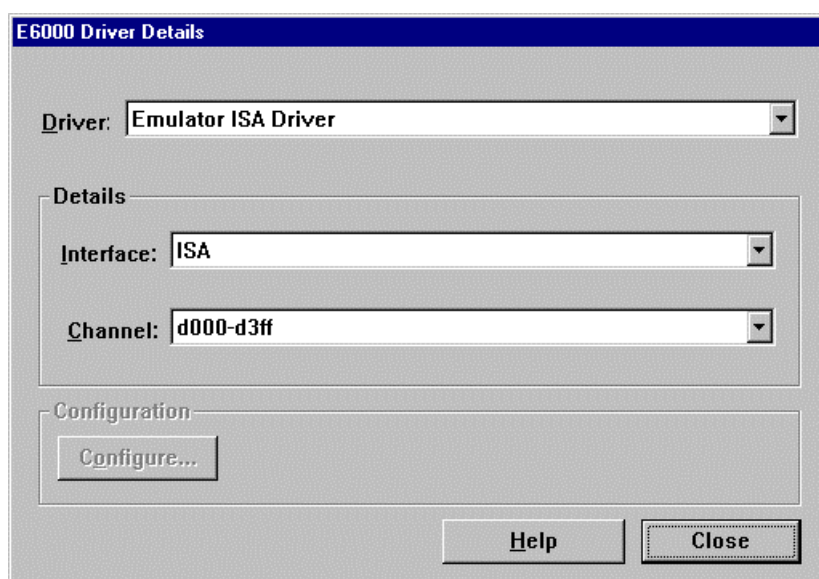


図 5-2 Driver Details ダイアログボックス

Driver	インストールされている E6000 ドライバを選択します。
Interface	選択したドライバがサポートしているインタフェースを選択します。
Channel	選択したインタフェースのチャンネルを選択します (ISA Driver で、E6000 インタフェースカードが見つけられたハードウェアアドレス)。
Configure...	ドライバが、コンフィグレーションダイアログをサポートしている場合、このボタンを押すと表示されます。

5.2 System Status ウィンドウ

このウィンドウでは、セッション情報、デバッグプラットフォームの現在のステータス、デバイスのメモリ構成及び エミュレータハードウェアのメモリ資源 及び ブレークポイントの種類とリソースを表示します。

Session シート:

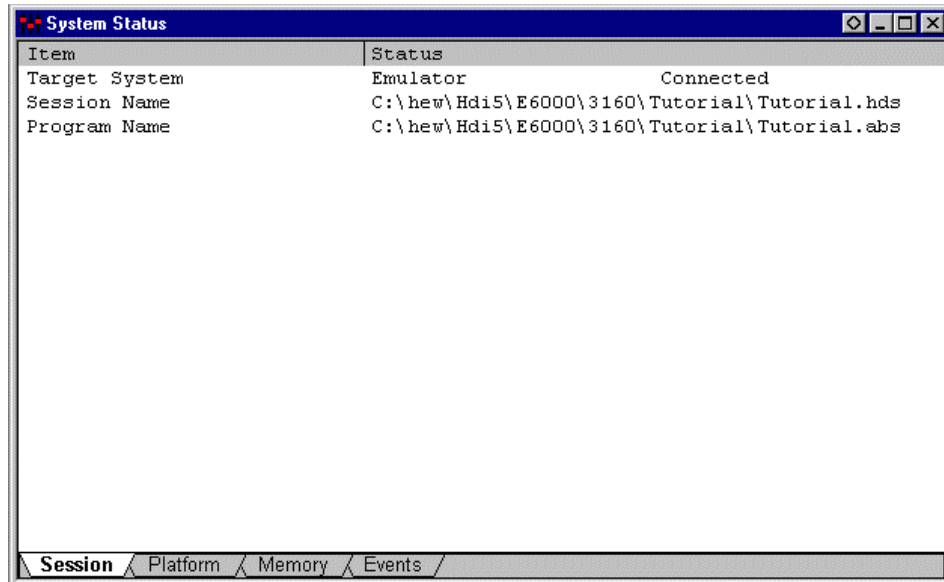


図 5-3 System Status ウィンドウ(Session シート)

各項目について説明します。

Target System	ターゲットシステムが接続されているかを表示
Session Name	セッションファイル名を表示
Program Name	プログラムファイル名を表示

Platform シート:

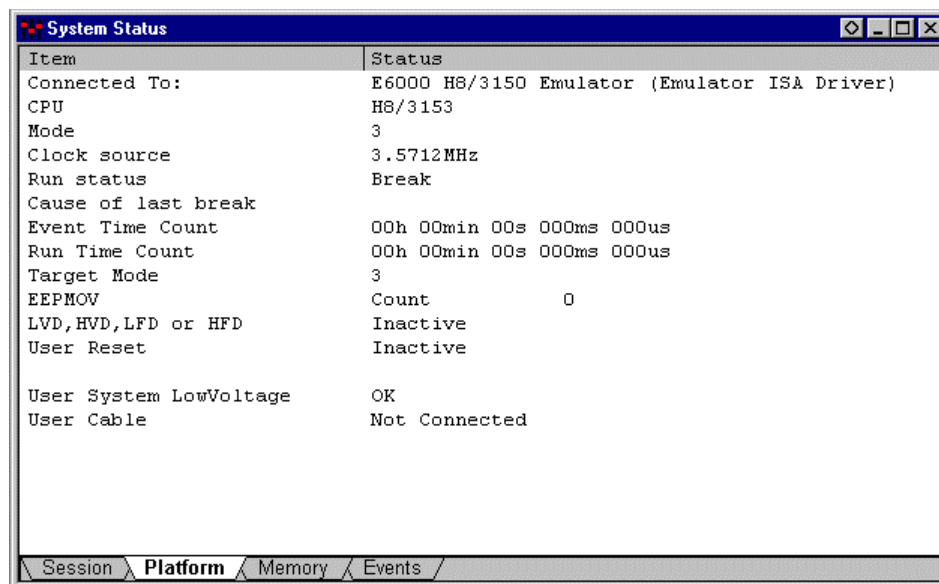


図 5-4 System Status ウィンドウ(Platform シート)

各項目について説明します。

Connected To:	エミュレータの品名を表示(使用しているドライバ名)
CPU	選択しているデバイス名を表示
Mode	常に”3”を表示
Clock source	選択しているクロックを表示
Run status	Break: ブレーク中 Running: プログラム実行中
Cause of last break	プログラムのブレーク要因を表示 User Break: ユーザによる停止 Soft Ware Breakpoint: プログラムブレークポイントによる停止 Hard Ware PC Breakpoint: ハードウェア PC Break による停止 Event Break: イベント検出システムによる停止 Performance Break: Performance Analysis による停止 EEPMOV Instruction Count Break: EEPMOV 命令指定回数実行による停止 Stepping Completed: ステップ実行の完了 Stepping Aborted: ステップの異常終了 ROM Write Access Break: ROM にライトした Write-protect Access Break: リードオンリメモリにライトした Unused Area Access Break: アクセス禁止メモリにアクセスした Stack In I/O Area: I/O エリアへのスタックが発生した。 Invalid breakpoint: PC Break 以外のブレーク命令による停止
Event Time Count	イベント間タイマの測定結果を表示(+Timer から-Timer まで)
Run Time Count	プログラムの総実行時間を表示
Target Mode	常に”3”を表示
EEPMOV Count	EEPMOV 命令の実行回数を表示
LVD,HVD,LFD or HFD	LVD,HVD,LFD,HFD 信号の状態を表示
User Reset	Reset 信号の状態を表示
User System Low Voltage	ユーザ VCC が Configuration ダイアログボックスの[User LowVCC Threshold]で指定している値を満たしているかを表示
User Cable	ユーザケーブルが接続されているかを表示

Memory シート:

Item	Status
Target Device Configuration	00000000-00007FFF Internal ROM
	00008000-0000C1FF EEPROM Area
	0000C200-0000FBBF Reserved Area
	0000FBC0-0000FFBF Internal RAM
	0000FFC0-0000FFEE Reserved Area
	0000FFEF-0000FFF0 Internal IO
	0000FFF1-0000FFF1 Reserved Area
	0000FFF2-0000FFF5 Internal IO
	0000FFF6-0000FFF7 Reserved Area
	0000FFF8-0000FFF9 Internal IO
	0000FFFA-0000FFFB Reserved Area
	0000FFFC-0000FFFF Internal IO
System Memory Resources	Not used
Loaded Memory Areas	00000000 - 00000001
	00001000 - 00001463

図 5-5 System Status ウィンドウ (Memory シート)

各項目について説明します。

Target Device Configuration	デバイスのメモリ構成を表示
System Memory Resources	エミュレータハードウェアのメモリ資源を表示 メモリ資源が全く使用されていない場合 Not used を表示します。
Load Memory Areas	現在ロードしたオブジェクトファイルによって使用されるメモリエリアを表示

Events シート:

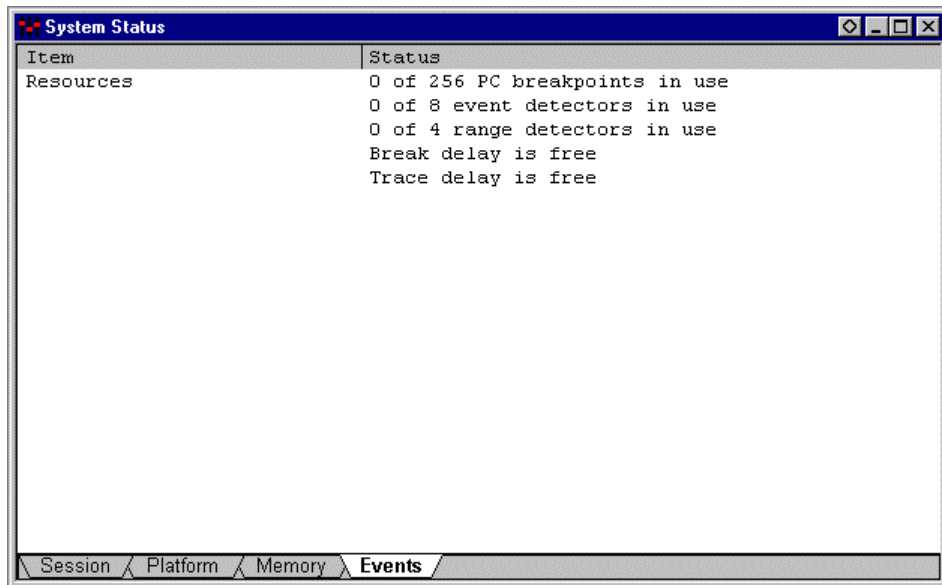


図 5-6 System Status ウィンドウ (Events シート)

各項目について説明します。

Resources | ブレークポイントの種類とリソースを表示

各ページ内でマウスの右ボタンをクリックするとポップアップメニューが表示されます。

Update	表示データを最新情報に更新します。
Copy	反転表示されたテキストを Windows クリップボードにコピーし、他のアプリケーションに貼り付けられるようにします。テキストブロックが反転されている場合のみ使用できます。

5.3 Breakpoints ウィンドウ

ブレークポイントの追加、削除、変更または有効 / 無効について説明します。ブレークポイントには、PC ブレークポイントとイベントの 2 つのタイプがあります。ブレークポイントとイベントがこのドキュメントで互換性を持って使用されることに注意してください。特に、ブレークポイントは必ずプログラム実行を停止するわけではありません。

PC ブレークポイントは指定されたアドレスの命令を実行しないで、ユーザプログラムを停止します。PC ブレークは 256 ポイントまで設定できます。

イベントは、条件の検出、ブレーク、時間計測およびトレース取得など様々なトリガとして、広範囲の目的に使用します。このウィンドウではリストボックスに含まれる全てのイベントおよび PC ブレークポイントの設定が可能です。E6000 H8/3160 は広範囲のイベントをサポートしたシステムであり、洗練された条件の設定が可能です。条件としては、タイマの開始または停止、トレース取得の on/off およびユーザプログラム実行の停止 (ブレークポイント) を一度に定義することができます。

E6000 ハードウェアが含む機能:

event detectors.	8 ポイント
range detectors.	4 ポイント

このウィンドウの例を以下に示します。

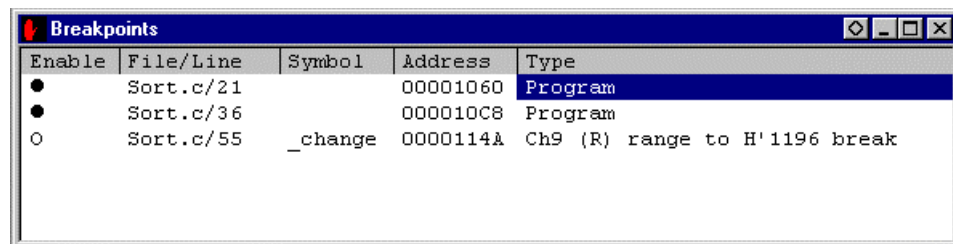


図 5-7 Breakpoints ウィンドウ

ポップアップメニューでブレークポイントの追加 / 編集 / 削除 / 有効 / 無効 を行います。

ポップアップメニュー

Add	新しいブレークポイントを追加します。
Edit	既存のブレークポイントを編集します。
Enable/Disable	ブレークポイントを有効または無効にします。
Delete	選択したブレークポイントを削除します。
Delete All	ブレークポイントを全て削除します。
Go to Source	ブレークポイントのある Source または Disassembly ウィンドウを表示します。

ポップアップメニューから[Add...]を選択すると、以下のダイアログボックスを表示します。

[General]パネル:

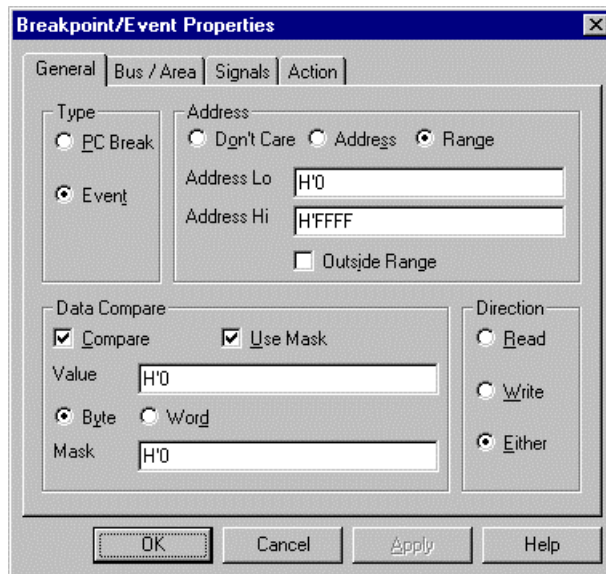


図 5-8 Breakpoint/Event Properties ダイアログボックス(General)

特定タイプのブレークポイントを有効または無効に選択し、他のパネルとダイアログボックスのオプションに従って、そのタイプのブレークポイントを利用可能にします。

PC ブレークポイントを選択すると、[Address Lo]エディットボックスのみ有効になります。

イベントブレークポイントを選択すると、[General]パネルでアドレス、バス方向およびデータ比較オプションを設定します。同様に他のアクセス条件を[Bus / Area]、[Action]および[Signals]パネルで設定します。

ダイアログボックスは、最も一般的にアクセスするオプションが最初([General]パネル)に表示されています。

Address	単一アドレス、範囲アドレスを選択します。
Outside Range	アドレス範囲を否定します。(例 範囲外にアドレスがあるときイベントが成立する)
Direction	バス方向を設定します。
Data Compare	バスデータを指定します。データ値はマスクすることができるので、指定されたビットだけ比較します。データサイズ(ByteかWord)は比較を正しく実行するためにソフトウェアによってデータをシフトおよびマスクして使用します(アドレス最下位ビットに伴う)。[Compare]チェックボックスをチェックしてデータ比較を切り換えます。

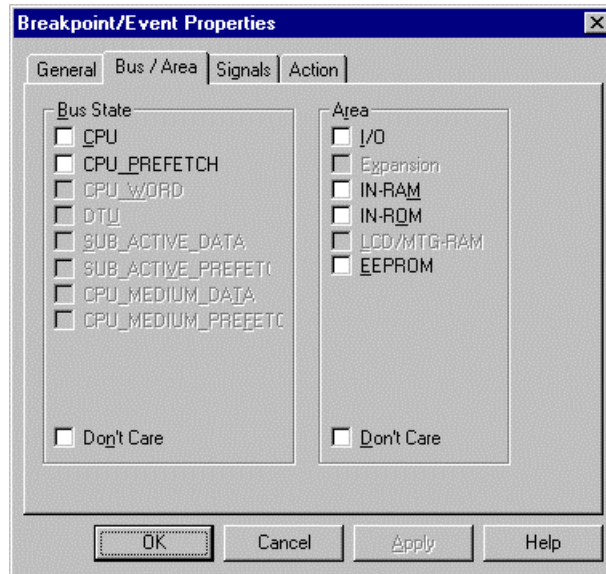
[Bus / Area]パネル:

図 5-9 Breakpoint/Event Properties ダイアログボックス(Bus / Area)

[Bus / Area]パネルはバス状態とアクセスするメモリエリアを指定します。

[Bus State]ボックスには複数のチェックボックスがあり、検索するバス状態を指定します。
[Don't Care]チェックボックスをチェックすると、すべてのバス状態を検索します。

[Area]ボックスには複数のチェックボックスがあり、検索するエリアを指定します。
[Don't Care]チェックボックスをチェックすると、すべてのエリアを検索します。

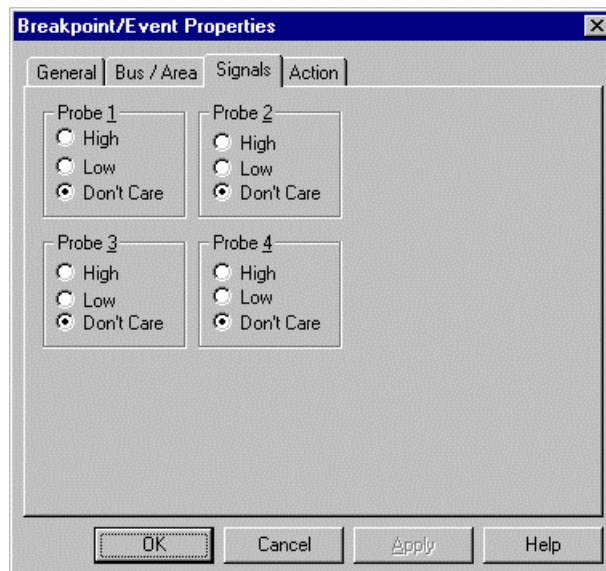
[Signals]パネル:

図 5-10 Breakpoint/Event Properties ダイアログボックス(Signals)

[Signals]パネルは、外部信号をイベントに入力します。**Don't Care**を設定しない場合は、イベントが成立する値を指定します。

[Action]パネル:

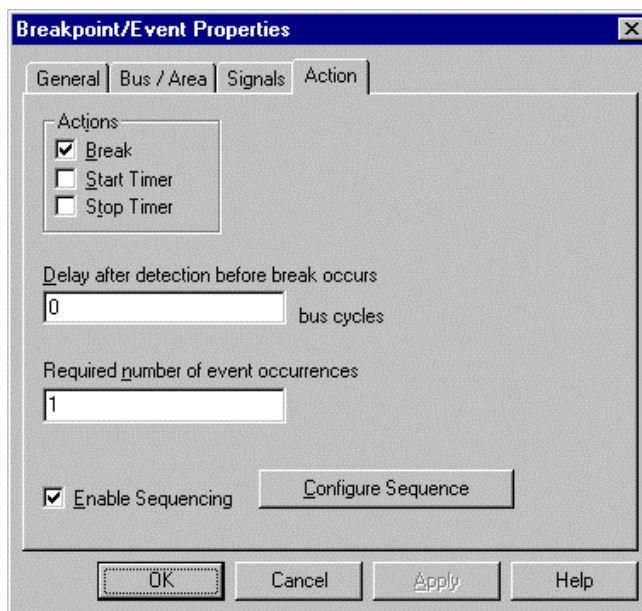


図 5-11 Breakpoint/Event Properties ダイアログボックス(Action)

[Action]パネルは、前のパネルで定義したイベントが成立すると、エミュレータがどんな action をするかを決定します。

Break	イベントが成立したとき、ユーザプログラムを中断(停止)します。これはデフォルト action です。
Timer Start	実行タイマを始動させます。(実行タイマ値は System Status ウィンドウの Platform シートに表示されます。)
Timer Stop	実行タイマを停止させます。(実行タイマ値は System Status ウィンドウの Platform シートに表示されます。)
Delay after detection before break occurs	イベント成立後の action が発生するまでの 16 ビットディレイ (バスサイクル) を設定します。ディレイはイベントブレイクのみで使用できるハードウェア上の 1 つだけのディレイカウンタです。従って、1 つのブレイクポイントのみディレイを指定することができます。
Required number of event occurrences	16 ビットパスカウントを設定します。指定した回数イベント成立後、action が発生します。event detector の設定が 1 つ必要です。
Enable Sequencing	イベントを順位づけします (event detector の設定が必要です)。[Enable Sequencing] チェックボックスをチェックすると [Configure Sequence] ボタンが有効になります。
Configure Sequence	event sequencing を指定するダイアログボックスを表示します。

5.3.1 Range Detectors

E6000 には、4 ポイントの独立した range detector があります。range detector は、以下の様々な組み合わせの指定により、イベントを定義します：

- アドレスまたはアドレス範囲のアクセス。
- メモリアクセスの特定のバス方向 (read、write、either)。
- データバス上に現れる値 (任意のマスク)。
- 1 つまたは複数の外部 probe の状態。4 つの probe は high、low または Don't Care を指定することができます。
- バス状態。以下の状態のどのような組み合わせも指定することができます：

CPU	CPU 命令サイクル
CPU_PERFETCH	CPU 命令プリフェッチサイクル
- エリアアクセス。以下のどのような組み合わせも指定することができます：

I/O, IN-RAM, IN-ROM, EEPROM

range detector は breakpoint、または trace acquisition stop、trace acquisition delayed stop、trace acquisition range example の指定に使用します

5.3.2 Event Detectors

E6000 には 4 ポイントの range detectors に加えて、8 ポイントの event detector があります。これらは range detectors と同じ設定条件の他に下記指定が可能です。

- アドレスの範囲内か範囲外かを指定します。
- イベントカウンタに、イベント以外の条件がイベントトリガの前に成立する回数を指定します。
- イベント順位付けすることができます。これは、他のイベントでイベントを使用可能な状態にすることができることを意味します (他のイベントがトリガとなり、その後に成立する)。また、別のイベントによりイベントカウンタをリセットすることができます (イベント自身が絶えず別のイベントのトリガとなる)。順序づけは、Event Sequencing ダイアログボックスを使用して指定します。

event detector は range detector と同じ指定の他に、trace acquisition event モードに使用されます。

5.3.3 Event Sequencing

Event Sequencing ダイアログボックスは、イベントが他のイベントのトリガとなることを定義します。Trace Acquisition からこのダイアログをアクセス（直接または間接）すると、トレースのサブシステムに割当てることができるイベントのみ表示します。Breakpoints ウィンドウからアクセスすると、ブレークポイントまたはタイマイベントのみ表示します。

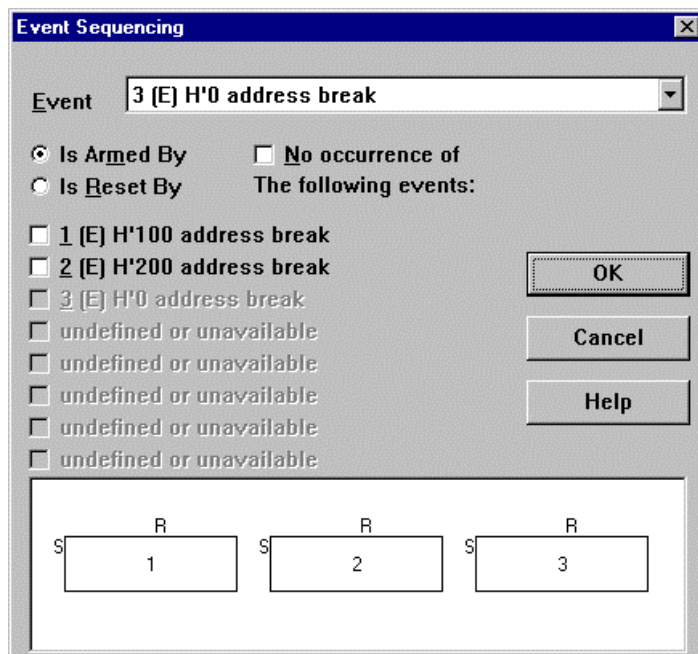


図 5-12 Event Sequencing ダイアログボックス

本ダイアログボックスにより、設定された各イベントは、1 イベントまたは複数イベント発生、非発生イベント等の条件でシーケンス制御することが可能です。

また、1 イベントまたは複数イベント発生によりイベントをリセット（例 カレントのパスカウントをリセットしてゼロにする）することができます。イベントはイベント自身をリセットすることができます。

[Event] ドロップダウンリストボックスを使用してイベントを選択してください。これをイベント A とします。選択したイベント A の成立条件を *Is Armed By* で、また、リセット条件を *Is Reset By* でシーケンス制御することができます。イベントの中のチェックボックスをチェックし、これをイベント B とすると、*Is Armed By* が選択されている場合イベント A の成立判定はイベント B の成立後開始されます。*Is Reset By* が選択されている場合イベント A はイベント B の成立によってリセットされます。[No occurrence of] チェックボックスを使用して *Is Armed By* を選択すると、イベント A の成立判定はイベント B 不成立後に開始されます。

スクリーンの下に、イベントのカレントの順序づけを示す図があります。S はシーケンス入力を（使用可能）、R はリセット入力を示します。~S が示すのは入力したイベント（使用可能）の非発生を意味します。

5.4 Trace ウィンドウ

Trace ウィンドウはトレースバッファの内容を表示します。トレースバッファは直前のプログラム実行において最大 32k バスサイクル取得します。**Cycle** は格納したトレース番号を示します。

Trace ウィンドウは、未加工のトレースデータ、逆アセンブルおよび C ソースコード（使用している場合）を表示します。

以下のカラムを表示します：

Cycle	トレースメモリ内のトレース番号。
Address	プロセッサバス上のアドレス。
Label	アドレスと関連づけられるラベル（ある場合）。
Code	アドレスの命令コード。
Data	バスに関するデータ。これは 1 バイトまたは 2 バイトです。
R/W	バスサイクルがリードまたはライトを示します。
Area	アクセスしたエリア。I/O、IN-RAM、IN-ROM、EEPROM
Status	このサイクルのデバイス状態。CPU、CPU_PREFETCH
Clock	バスサイクルのクロック数。
Probes	信号入力の状態。
IRQ1 IRQ2	IRQ1, IRQ2 の状態。
Source	ソースコードの表示。

Time Stamp が無効の場合、上のカラムを表示します。有効な場合は、**area**、**status**、**clock**、**probes** および **IRQ1**、**IRQ2** カラムは **timestamp** と **delta time** カラムに取り替えられます。**timestamp** は各サイクルの時間で、取得パラメータによって定義される単位で測定します。**delta time** はカレントサイクルと前サイクルとの時間間隔です。詳細に関しては Trace Acquisition General を参照してください。

Trace ウィンドウがアクティブなとき、ポップアップメニューの項目 Find を使用して特定のトレース検索ができます。

Trace Acquisition 項目はトレース構成をするためにダイアログボックスを表示します。これは 4 つの範囲トレースページとその上の 3 つのセクションに分かれます。ヘルプは以下のページで利用可能です：

- Trace Acquisition - General
- Trace Acquisition - Stop
- Trace Acquisition - Delayed Stop
- Trace Acquisition - Range example (“1”~”4”のタブ)

Trace ウィンドウのカラム幅は、カラム見出し間の垂直な分離ラインをクリックおよびドラッグすることにより調整することができます。ウィンドウを閉じると、新しいカラム幅は自動的にセーブされます。

ポップアップメニュー

Find	指定したパターンに合うトレースデータサイクルを検索します。
Find Next	次に合うサイクルを検索します。
Filter	定義した規準に合うサイクルだけを表示するためにフィルタを創ります。ハードウェアの制御によりトレースデータを取得した後、基準に合うトレースサイクルを表示します。
Acquisition	ハードウェアのトレース取得パラメータを設定します。
Halt	トレース取得を中止します。
Restart	トレース取得を再開します。
Snapshot	カレントトレース記録の"snapshot"を読みます。
Clear	全てのトレース記録をクリアします。
Save	表示しているトレースデータを全てファイルに書き込みます。
View Source	該当アドレスに対応したソースコードを表示します。
Trim Source	ソースコードの左側の空白を取り除きます。

5.4.1 Trace Acquisition General

以下に Trace Acquisition ダイアログボックスの[General]パネルの例を示します。

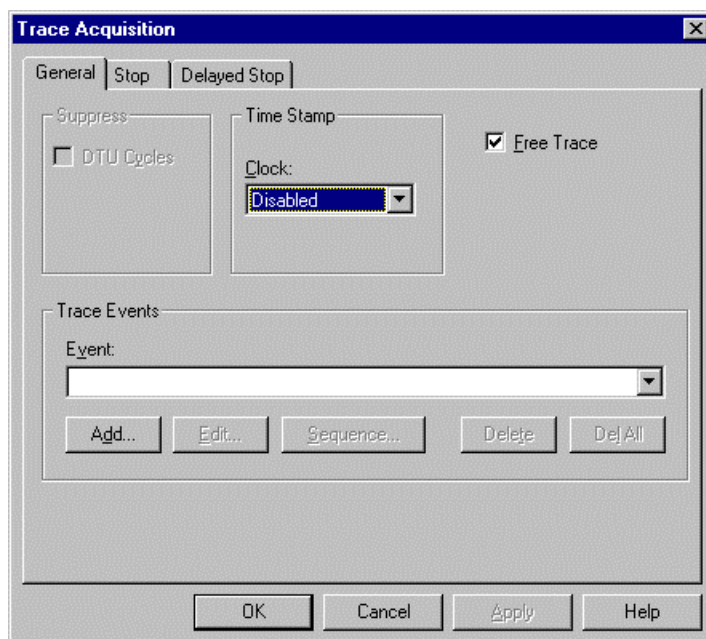


図 5-13 Trace Acquisition ダイアログボックス(General)

[Free Trace]チェックボックスはフリートレースモードを有効にします。このモードで範囲トレースは利用不可であり、4つの範囲トレースページ(ラベル1~4)は無効になります。フリートレースモードはプログラム実行開始直後にデータ取得を開始します。フリートレースモードを無効にすると、トレース取得の開始および停止条件の設定となります。フリートレースモードではトレース停止条件の設定のみ可能です。

[Time Stamp]ボックスは、トレース取得の時間計測を有効にします。[Time Stamp]ボックスを Disabled とした場合、制御信号がトレースバッファに収納されます。[Time Stamp]ボックスを Disabled としない場合、timestamp および delta time を取得します。利用可能な計測クロックは; 125ns, 250ns, 500ns, 1us, 2us, 4us, 8us, 16us および 100us です。Disabled とした場合は、制御信号 area, status, clock, probes および IRQ1, IRQ2 を取得します。

[Trace Events]ボックスはトレースのサブシステムにより使用する E6000 Complex Events 操作の 1 つです。[Add...]ボタンは新しいイベントを追加、[Edit...]ボタンはイベントを変更、[Delete]および[Del All]ボタンはイベントを削除します。[Sequence...]は各イベントを arm または reset に順序づけるダイアログボックスのショートカットです。先にイベントを設定していないと、セットアップはできません。

イベントは 8 ポイントの event detector (1E~8E) と 4 ポイントの range detector (9R~12R) に割り当てられます。但し、ハードウェアブレークを設定しているとその分は除かれます。

event detector には range detector 以上の機能があります。(シーケンス、カウントなど)。イベントがこのダイアログボックスからトレースのサブシステムに割り当てられるとき、ブレークポイントまたは Breakpoints ウィンドウのタイムイベントとしては利用不可です。範囲トレースまたはトレース停止で使用しているイベントが削除されると、範囲トレースまたはトレース停止機能は無効になります。

イベントのチャンネル割り当て順序は、シーケンス、カウントを使用していない場合、チャンネル 9R~12R、1E~8E の順になります。

5.4.2 Trace Acquisition Stop

以下に Trace Acquisition ダイアログボックスの[Stop]パネルの例を示します。

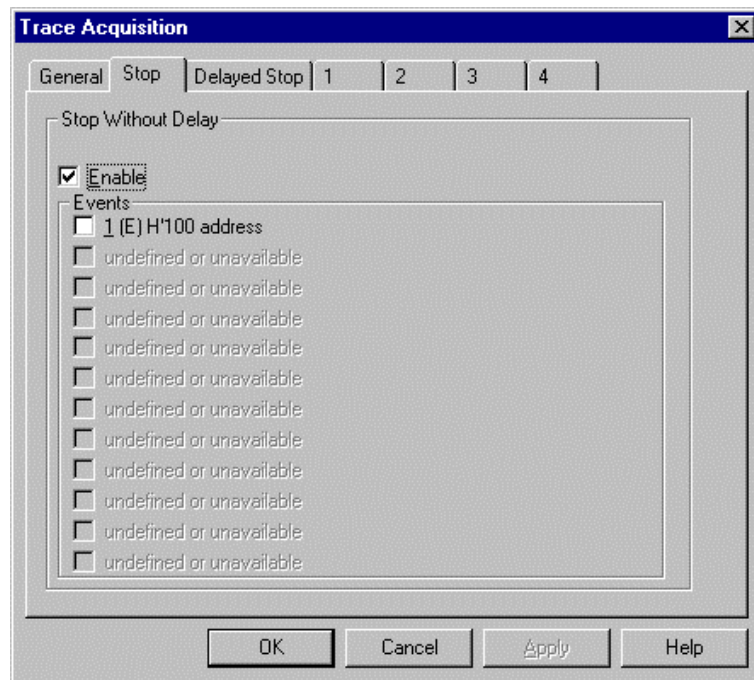


図 5-14 Trace Acquisition ダイアログボックス(Stop)

Trace Stop Without Delay はトレース停止条件を定義します。トレース停止条件はディレイの有無にかかわらず、両方同時に指定できます。[Enable]チェックボックスはトレース停止を有効にし、[Events]ボックスは定義されたトレースイベントを表示します。イベントチェックボックスをチェックした場合、イベントが成立したときトレース停止します。[Enable]チェックボックスをチェックした場合は、少なくとも 1 イベントを選択してください。

5.4.3 Trace Acquisition Delayed Stop

以下に Trace Acquisition ダイアログボックスの[Delayed Stop]パネルの例を示します。

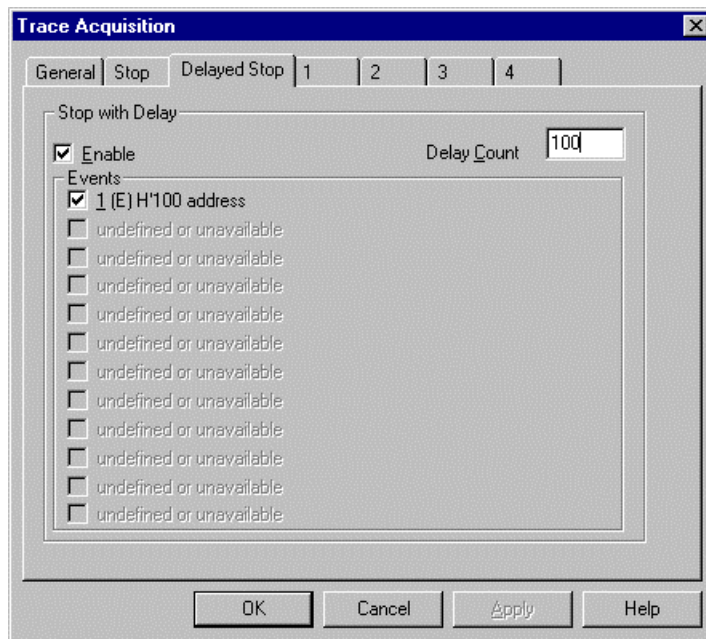


図 5-15 Trace Acquisition ダイアログボックス(Delayed Stop)

*Trace Stop with Delay*はトレース停止条件を定義します。トレース停止条件はディレイの有無にかかわらず、両方同時に指定できます。[Enable]チェックボックスはトレース停止を有効にし、[Events]ボックスは定義されたトレースイベントを表示します。イベントチェックボックスをチェックした場合、イベントが成立したときトレース停止します。[Enable]チェックボックスをチェックした場合は、少なくとも1イベントを選択してください。

[Delay Count]エディットボックスはディレイカウント(バスサイクル、1~65535の範囲)を設定します。この機能により、イベントが成立した後にいくらかのトレースサイクルを取得します。

5.4.4 Trace Acquisition Range Example

各 *Range Trace* ページでは、**Disabled**、**Point to Point**、**Range** および **Event** ラジオボタンで範囲トレースモードを選択します。

Disabled を選択した場合、範囲トレースは使用されず、また他のどんな制御も表示しません。

Point to Point モード:

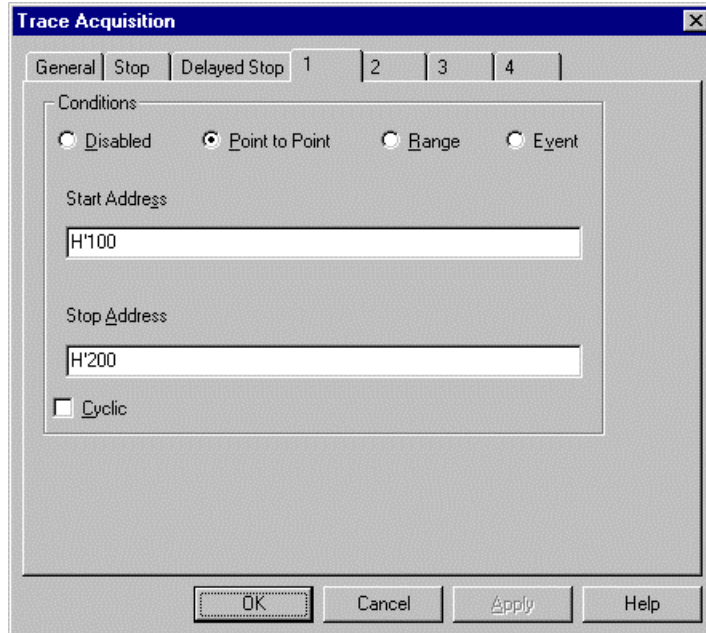


図 5-16 Trace Acquisition ダイアログボックス(Point to Point)

Point to Point モードは**開始**および**終了アドレス**のエディットボックスを表示します。開始アドレスをアクセスしたときトレースを開始し、停止アドレスをアクセスしたときトレースを停止するために必要なイベントをソフトウェアに設定します。Point to point モードは Event モードをセットアップする簡単な方法で、開始および停止イベントは単一アドレスをアクセスするイベントです。

Event モード:

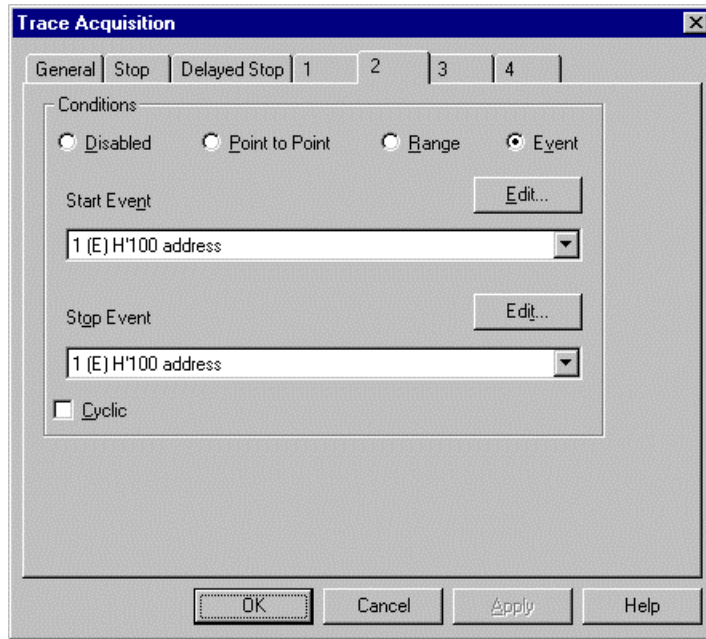


図 5-17 Trace Acquisition ダイアログボックス(Event)

Event モードは、定義されたイベントリストからトレース停止および開始に使用するイベントを選択します。(event detectors は Event モードおよび Point to point モードの場合、イベントを定義すると強制的に Breakpoint/Event Properties ダイアログボックスの **Action** ページの中の **Use Sequencing** ボックスがチェックされます。) Event モードは、より複雑なトレースの開始および停止条件の定義とトレースを開始するためのイベントシーケンスで構成されます。[**Cyclic**] チェックボックスをチェックするとイベントシーケンスを構成し、イベント自身がリセットになり、停止イベントの後に開始イベントが成立すると再度トレースを開始します。

Range モード:

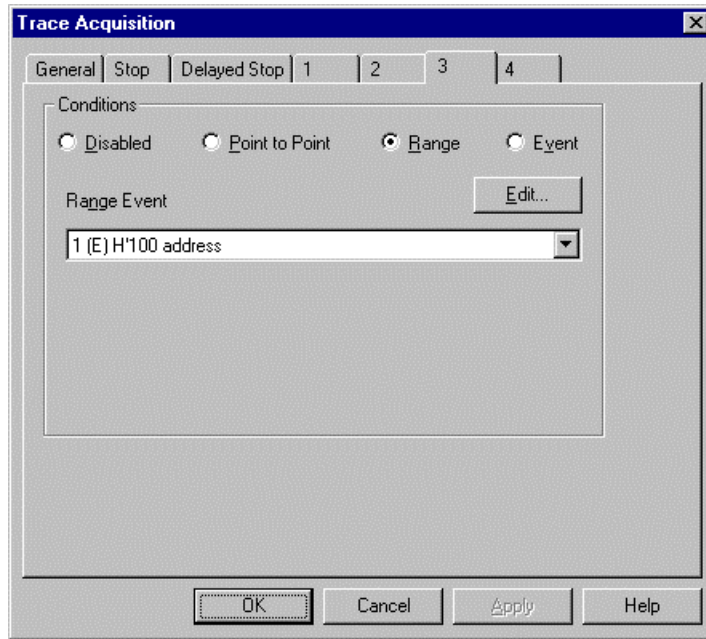


図 5-18 Trace Acquisition ダイアログボックス(Range)

Range トレースモードは、すべてのバスサイクルから選択したイベント条件に合うものをトレース取得します。このモードは1つの event detector または range detector を使用します。

注意: 範囲を定義したとき、トレース範囲の適切な機能を順位づけしセットアップします。予期しない結果が起きるかもしれませんが、Event Sequencing ダイアログボックスで順序を変更するのは可能です。一般に、トレース開始の範囲イベントのみ、イベントシーケンスの定義が可能です。

5.4.5 Trace Find

このオプションは Trace ウィンドウのデータを検索するダイアログボックスを表示します。指定した検索標準の次を検索するにはポップアップメニューから [Find Next] を選択してください。

以下のダイアログボックスを表示します：

[General] パネル：

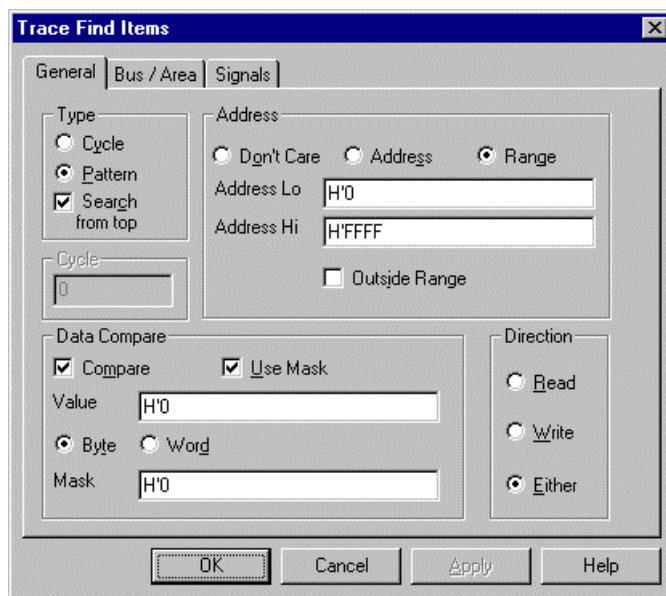


図 5-19 Trace Find ダイアログボックス(General)

サイクルまたはパターン検索タイプのどちらかを選択し、そのタイプで使用できるオプションに従って、ダイアログボックスの様々な部分を有効および無効にします。

サイクル検索を選択した場合、サイクル番号以外のすべてのオプションが無効になります。

パターン検索を選択した場合、[General] パネルでアドレス範囲、データ値およびアクセスしたバス方向を指定します。他のパネルではバス状態およびエリア、外部信号、タイムスタンプ値のオプションを使用して一般的な指定ができます。

[General] パネルで利用可能なオプション：

Address	単一アドレス、範囲アドレス、アドレスを無視、により検索します。
Outside Range	アドレス範囲を否定します。(例 範囲外のアドレス)
Direction	バス方向を設定します。
Data Compare	バスデータを指定します。データ値はマスクすることができるので、指定されたビットだけ比較します。データサイズ (Byte か Word) は比較を正しく実行するために、ソフトウェアによってデータをシフトおよびマスクして使用します (アドレス最下位ビットに伴う)。[Compare] チェックボックスをチェックしてデータ比較を切り換えます。

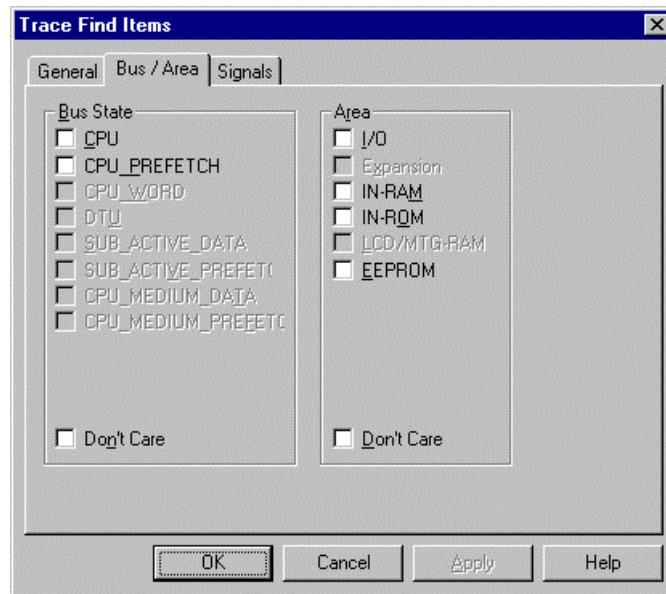
[Bus / Area]パネル:

図 5-20 Trace Find ダイアログボックス(Bus / Area)

[Bus / Area]パネルでは検索するバス状態およびメモリエリアアクセスを指定します。(このパネルはTrace Acquisition General 設定でタイムスタンプが無効の場合にのみ利用可能です。)

[Bus State]ボックスには複数のチェックボックスがあり、検索するバス状態を指定します。**[Don't Care]**チェックボックスをチェックすると、すべてのバス状態を検索します。

[Area]ボックスには複数のチェックボックスがあり、検索するエリアを指定します。**[Don't Care]**チェックボックスをチェックすると、すべてのエリアを検索します。

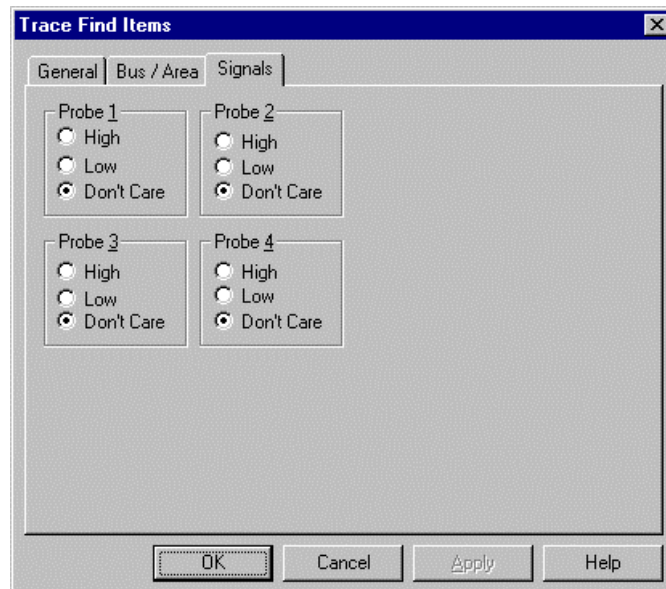
[Signals]パネル:

図 5-21 Trace Find ダイアログボックス(Signals)

[Signals]パネルでは検索する外部信号を指定します。[Don't Care]を設定しない場合は、検索する信号のトレースデータを指定します。（このパネルはTrace Acquisition General 設定でタイムスタンプが無効の場合にのみ利用可能です。）

[Time]パネル:

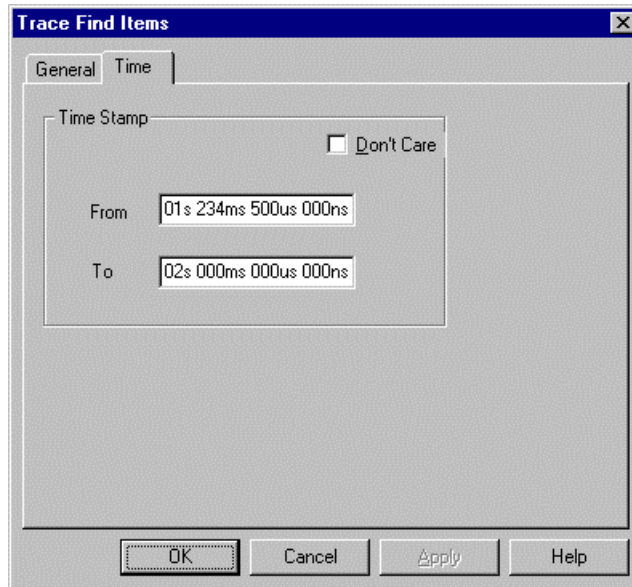


図 5-22 Trace Find ダイアログボックス(Time)

[Time]パネルでは検索するトレースのタイムスタンプ値または範囲を指定します。
[Don't Care]チェックボックスをチェックすると、タイムスタンプはトレース検索項目に含まれません。タイムスタンプ値は"1s 1ms 260us 500ns"などのように検索するタイムスタンプ値の範囲を指定します。タイムスタンプ値はTrace ウィンドウのタイムスタンプ表示のフォーマットと同じです。このフォーマットは以下の通りです:

秒 *S* ミリ秒 *ms* マイクロ秒 *us* ナノ秒 *ns*

例:

2s 123ms 400us 125ns

0s 000ms 100us 000ns

(このパネルはTrace Acquisition General 設定でタイムスタンプが有効の場合にのみ利用可能です。)

5.4.6 Trace Filter

Trace ウィンドウに表示するトレース情報を決定するフィルタを創ります。フィルタは Trace Acquisition パラメータと違い、ハードウェアにより取得したトレースデータをソフトウェアによりフィルタリングします。それに対して、Trace Acquisition はハードウェアによりトレースデータ取得を制御します。従って、1つのウィンドウではすべてのデータアクセスを表示し、もう1つのウィンドウではすべてのプログラムアクセスを表示するというように、異なった Trace ウィンドウには異なったトレースフィルタを設定することが可能です。ウィンドウでトレース表示のフィルタを止めるには、ダイアログボックスのすべてのオプションを Don't Care に設定してください。

以下のダイアログボックスを表示します。

[General]パネル:

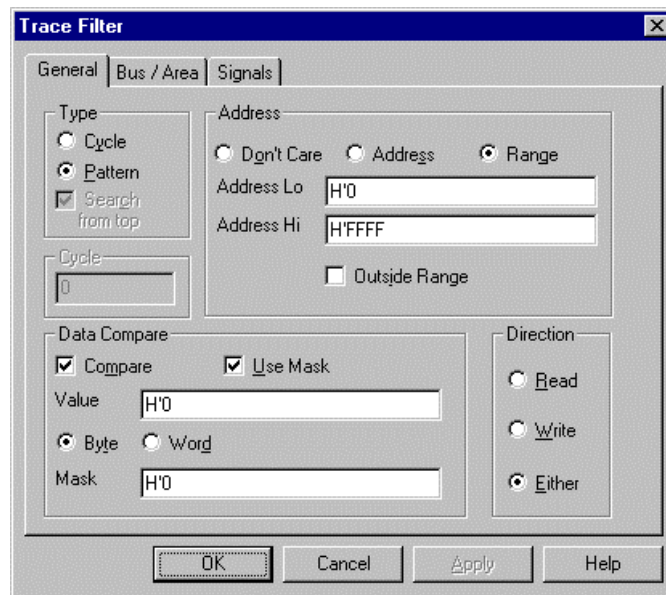


図 5-23 Trace Filter ダイアログボックス(General)

サイクルまたはパターンタイプのどちらかを選択し、そのタイプで使用できるオプションに従って、ダイアログボックスの様々な部分を有効および無効にします。

サイクルフィルタを選択した場合、サイクル番号以外のすべてのオプションが無効になります。

パターンフィルタを選択した場合、[General]パネルでアドレス範囲、データ値およびアクセスしたバス方向を指定します。他のパネルではバス状態およびエリア、外部信号、タイムスタンプ値のオプションを使用して一般的な指定ができます。

[General]パネルで利用可能なオプション:

Address	単一アドレス、範囲アドレス、アドレスを無視、によりフィルタリングします。
Outside Range	アドレス範囲を否定します。(例 範囲外のアドレス)
Direction	バス方向を設定します。
Data Compare	バスデータを指定します。データ値はマスクすることができるので、指定されたビットだけ比較します。データサイズ(ByteかWord)は比較を正しく実行するために、ソフトウェアによってデータをシフトおよびマスクして使用します(アドレス最下位ビットに伴う)。[Compare]チェックボックスをチェックしてデータ比較を切り替えます。

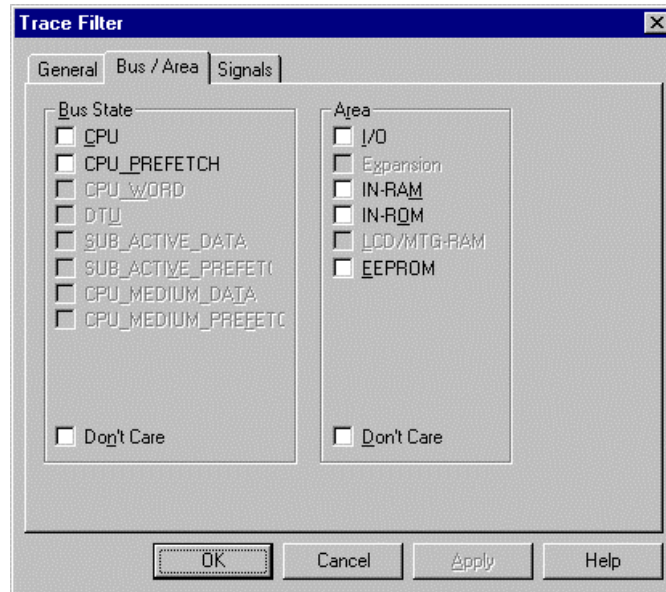
[Bus / Area]パネル:

図 5-24 Trace Filter ダイアログボックス(Bus / Area)

[Bus / Area]パネルではフィルタするバス状態およびメモリエリアアクセスを指定します。(このパネルはTrace Acquisition General 設定でタイムスタンプが無効の場合にのみ利用可能です。)

[Bus State]ボックスには複数のチェックボックスがあり、フィルタするバス状態を指定します。**[Don't Care]**チェックボックスをチェックすると、すべてのバス状態がトレース記録に含まれます。

[Area]ボックスは複数のチェックボックスがあり、フィルタするエリアを指定します。**[Don't Care]**チェックボックスをチェックすると、すべてのエリアがトレース記録に含まれます。

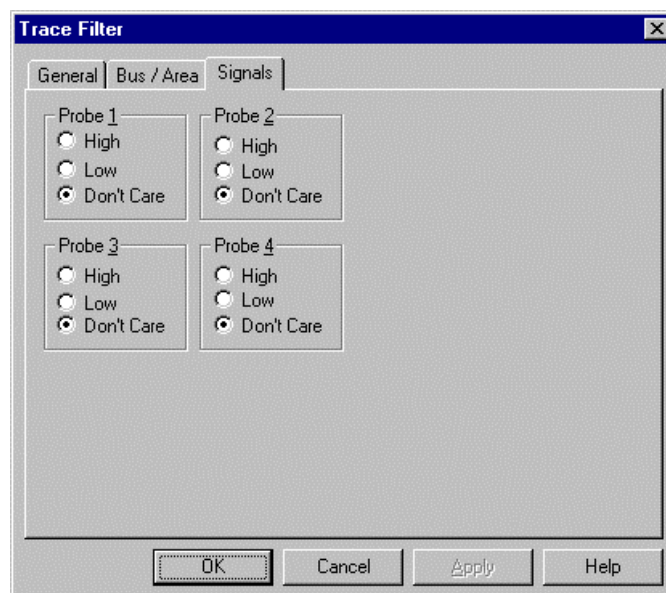
[Signals]パネル:

図 5-25 Trace Filter ダイアログボックス(Signals)

[Signals]パネルでは、フィルタする外部信号を指定します。[Don't Care]を設定しない場合は、フィルタするトレースデータを指定します。(このパネルは Trace Acquisition General 設定でタイムスタンプが無効の場合にのみ利用可能です。)

[Time]パネル:

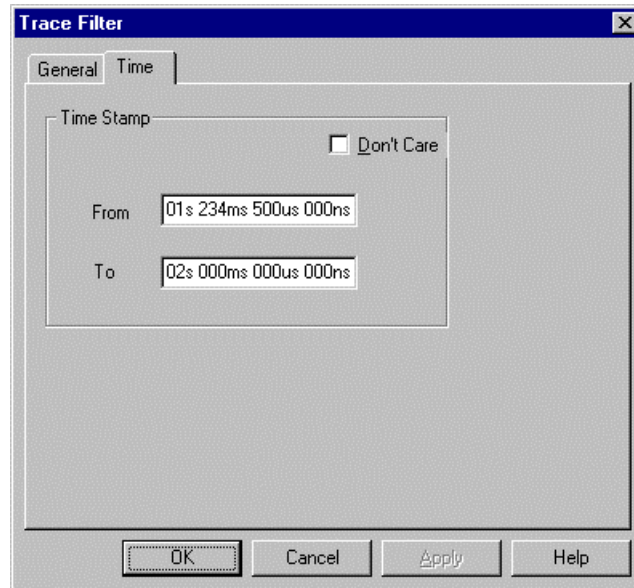


図 5-26 Trace Filter ダイアログボックス(Time)

[Time]パネルではフィルタするトレースのタイムスタンプ値または範囲値を指定します。[Don't Care]チェックボックスをチェックすると、タイムスタンプはトレースフィルタ項目に含まれません。タイムスタンプ値は"1s 1ms 260us 500ns"などのようにフィルタするタイムスタンプ値の範囲を指定します。タイムスタンプ値はTrace ウィンドウのタイムスタンプ表示のフォーマットと同じです。このフォーマットは以下の通りです:

秒 s ミリ秒 ms マイクロ秒 us ナノ秒 ns

例:

2s 123ms 400us 125ns

0s 000ms 100us 000ns

(このパネルは Trace Acquisition General 設定でタイムスタンプが有効の場合にのみ利用可能です。)

5.5 Memory Mapping ダイアログボックス

H8/3160 の Memory Mapping ダイアログボックスの例を以下に示します：
メモリマップは、リザーブエリア（アクセス禁止）を 1 バイト単位で設定することができます。デバイス内蔵メモリの内、Internal ROM、RAM、I/O の属性変更はできません。EEPROM のみ属性変更が可能です。

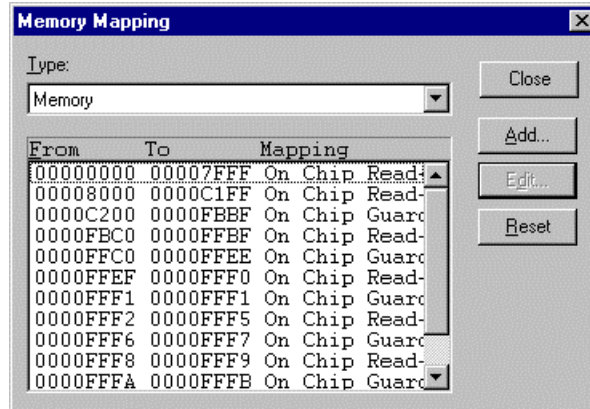


図 5-27 Memory Mapping ダイアログボックス

マップ設定を変更する場合は、対象の設定値を選択して [Edit...] ボタンをクリックするか、または対象のマップ設定行をダブルクリックしてください。

Edit Memory Mapping ダイアログボックスが表示され、アドレス範囲および属性を変更することができます。

EEPROM は、User Guarded への変更のみが可能です。

また、リザーブエリアは、Emulator Read-Write への変更のみが可能です。

デバイスのメモリ構成および エミュレータハードウェアのメモリ資源は、System Status ウィンドウの Memory シートに表示されます。

[Add...] ボタンを選択すると、メモリマップの追加、変更をするために前述したダイアログボックスを表示します。新しい割り付け位置の詳細を記入することができます。[Reset] ボタンを選択すると、メモリマップはデフォルト設定にリセットされます。

5.6 Performance Analysis

このダイアログボックスは、直前のプログラム実行でユーザが選択したエリアの実行時間比率をパーセント、ヒストグラムおよび数値で表示します。

このダイアログボックスは、ユーザプログラムの実行中は表示できません。表示するには、必ずユーザプログラムの実行を中止してください。

[Graph]をチェックすることにより、実行比率時間のパーセントとヒストグラムを表示します。“#”マーク1個につき、2%です。

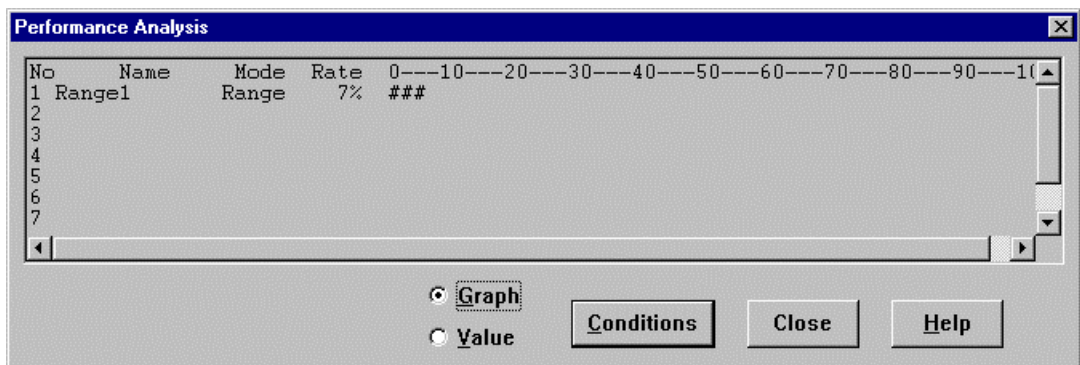


図 5-28 Performance Analysis ダイアログボックス(Graph)

ユーザプログラムを実行するごとに、前回の計測結果はクリアされます。

[Value]をチェックすることにより、実行比率時間のパーセント、実行時間および実行回数を表示します。

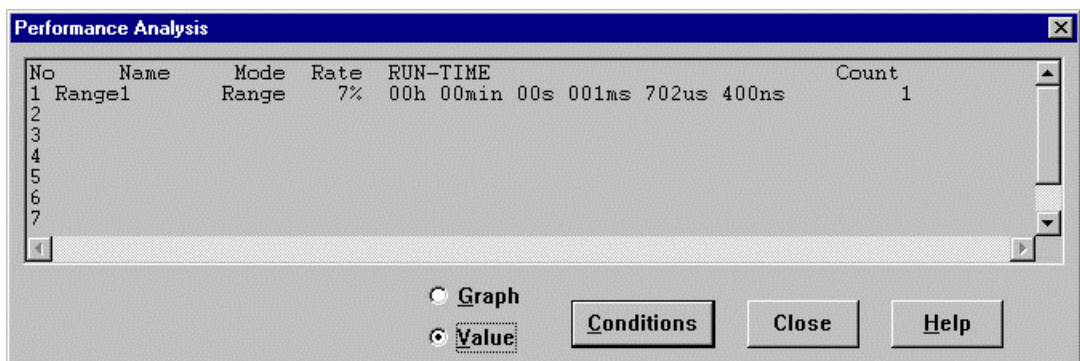


図 5-29 Performance Analysis ダイアログボックス(Value)

Graph	実行比率時間のパーセント、ヒストグラムを表示
Value	実行比率時間のパーセント、実行時間および実行回数を表示
Conditions	実行効率測定結果表示、アドレス検出モードおよび測定時間最小単位の設定を行います。(Performance Analysis Conditions ダイアログボックスを開きます。)
Close	このダイアログボックスを閉じます。

5.6.1 Performance Analysis Conditions

このダイアログボックスは、アドレス検出モード・測定時間最小単位の設定および実行効率測定条件の内容を表示します。

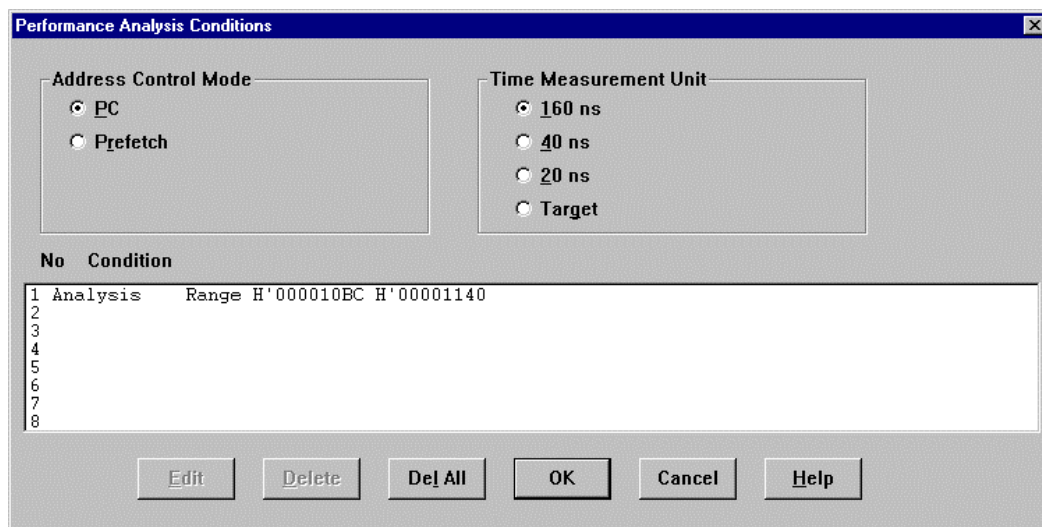


図 5-30 Performance Analysis Conditions ダイアログボックス

Address Control Mode	実行効率測定に使用するアドレス検出方式を設定します。[PC]をチェックすると PC アドレス検出モードとなり、[Prefetch] をチェックするとプリフェッチアドレス検出モードとなります。 領域アクセス回数測定モードは、プリフェッチアドレス検出モードに、それ以外の測定モードでは、PC アドレス検出モードに設定してください。上記の設定に従わない場合、その測定モードの測定値は不正になります。
Time Measurement Unit	測定時間の最小単位を指定します。160ns、40ns、20ns およびターゲットクロックが指定できます。
Condition	実行効率測定条件の内容を表示します。設定は最大 8 ポイントまで設定できます。ただし、指定アドレス範囲間時間測定、領域アクセス回数測定、指定範囲内コール回数測定の場合は、最大 4 ポイントとなります。設定内容に関しては Performance Analysis を参照してください。
Edit	選択したポイントに実行効率測定条件を設定します。(Performance Analysis Properties ダイアログボックスを開きます。)
Delete	選択したポイントの実行効率測定条件を削除します。
Del All	すべての実行効率測定条件を削除します。
OK	設定した内容を登録し、このダイアログボックスを閉じます。(Performance Analysis ダイアログボックスに戻ります。)
Cancel	設定した内容を登録せずに、このダイアログボックスを閉じます。(Performance Analysis ダイアログボックスに戻ります。)

注：指定範囲内時間測定および指定アドレス間時間測定は 1 ポイントを使用し、指定アドレス範囲間時間測定・領域アクセス回数および指定範囲内コール回数測定は連続した 2 ポイントを指定します。2 ポイントを使用するモードから 1 ポイントを使用するモードに変更した時、また、1 ポイントを使用するモードから 2 ポイントを使用するモードに変更した時、設定されていた条件は削除されます。

5.6.2 Performance Analysis Properties

このダイアログボックスは、実行効率測定の設定を行います。

実行効率測定条件は、5種類のモード(Measurement Method)があります。

領域アクセス回数測定モードは、アドレス検出モードをプリフェッチアドレス検出モードに、それ以外の測定モードでは、PC アドレス検出モードに設定してください。

このダイアログボックスは、実行効率測定の設定を行います。

実行効率測定条件は、5種類のモード(Measurement Method)があります。

領域アクセス回数測定モードは、アドレス検出モードをプリフェッチアドレス検出モードに、それ以外の測定モードでは、PC アドレス検出モードに設定してください。

1. 指定範囲内時間測定 : Time Of Specified Range Measurement

<開始アドレス>と<終了アドレス>で設定された範囲の実行時間および実行回数を測定します。測定時間は<開始アドレス>と<終了アドレス>の範囲のプログラムプリフェッチで測定を開始し、範囲以外のプログラムプリフェッチで測定を中断します。再度、設定範囲のプログラムプリフェッチで測定を開始します。実行回数は、設定範囲の<終了アドレス>のプログラムをフェッチするたびにカウントします。測定結果には、設定範囲内から呼び出された処理の実行時間は含まれません。

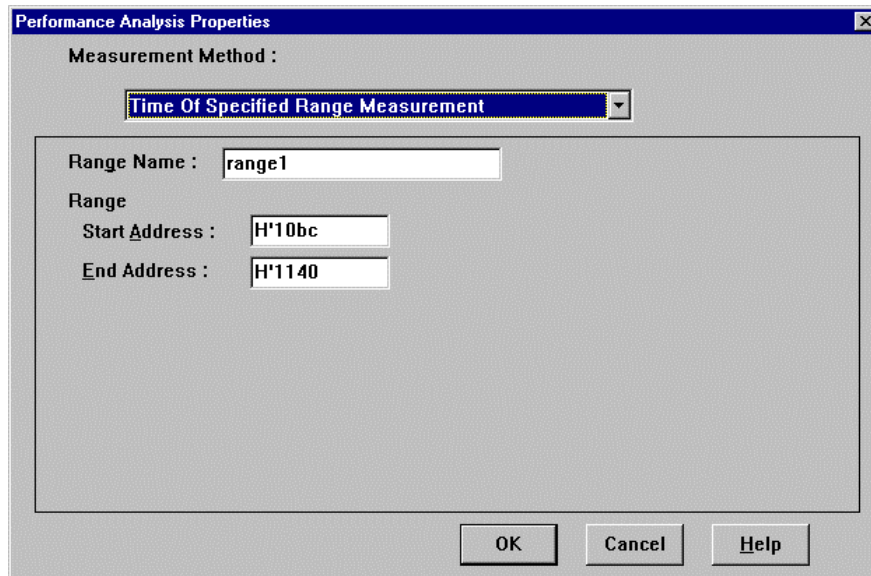


図 5-31 Performance Analysis Properties ダイアログボックス (1)

- 2 . 指定アドレス間時間測定 : Start Point To End Point Measurement
 <開始アドレス>と<終了アドレス>で設定された範囲の実行時間および実行回数を測定します。測定時間は、<開始アドレス>のプログラムプリフェッチで測定を開始し、<終了アドレス>のプログラムプリフェッチで測定を中断します。実行回数は、設定範囲の<終了アドレス>のプログラムをプリフェッチするたびにカウントします。測定結果には、設定範囲内から呼び出された処理の実行時間を含みます。ポイント1~4の場合に、設定された範囲の最大、最小時間を測定します。

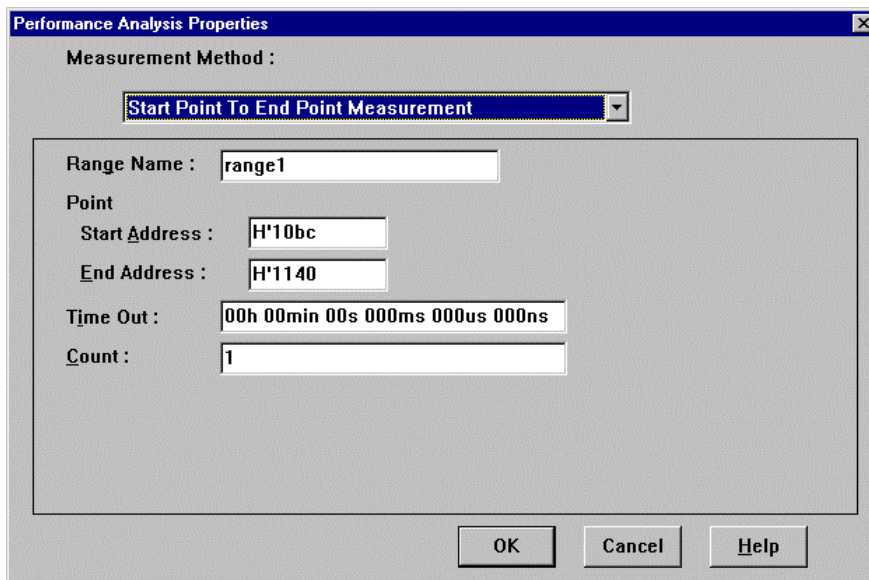


図 5-32 Performance Analysis Properties ダイアログボックス (2)

- 3 . 指定アドレス範囲間時間測定 : Start Range To End Range Measurement
 <開始アドレス範囲>のプリフェッチサイクルで時間測定を開始し、<終了アドレス範囲>のプログラムプリフェッチサイクルで測定を中断します。また、実行回数は、<終了アドレス範囲>を通過するたびにカウントアップします。

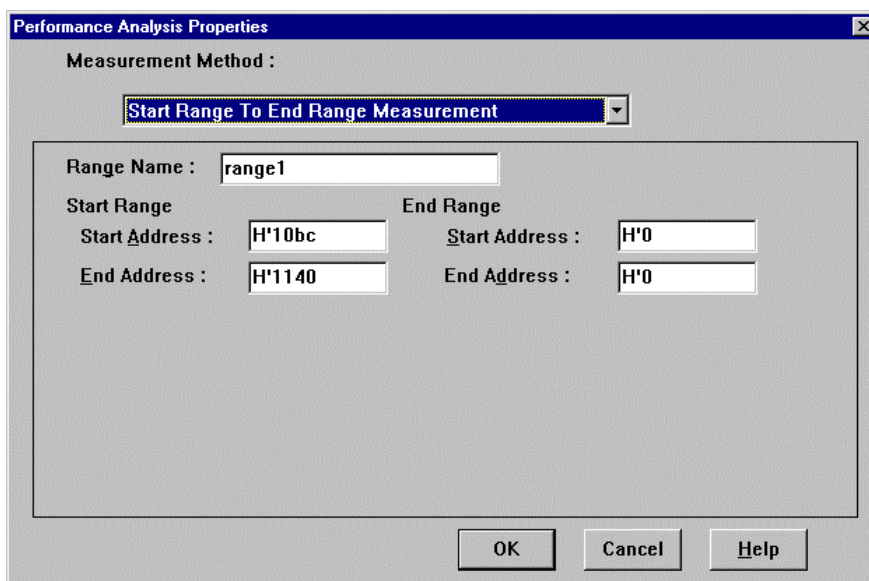


図 5-33 Performance Analysis Properties ダイアログボックス (3)

4. 領域アクセス回数測定 : Access Count Of Specified Range Measurement
 <開始アドレス>と<終了アドレス>で設定されている範囲から
 <アクセス領域アドレス範囲>で設定されている領域をアクセスした回数を測定します。
 また、範囲内の実行時間は、指定範囲内時間測定を用いて測定します。

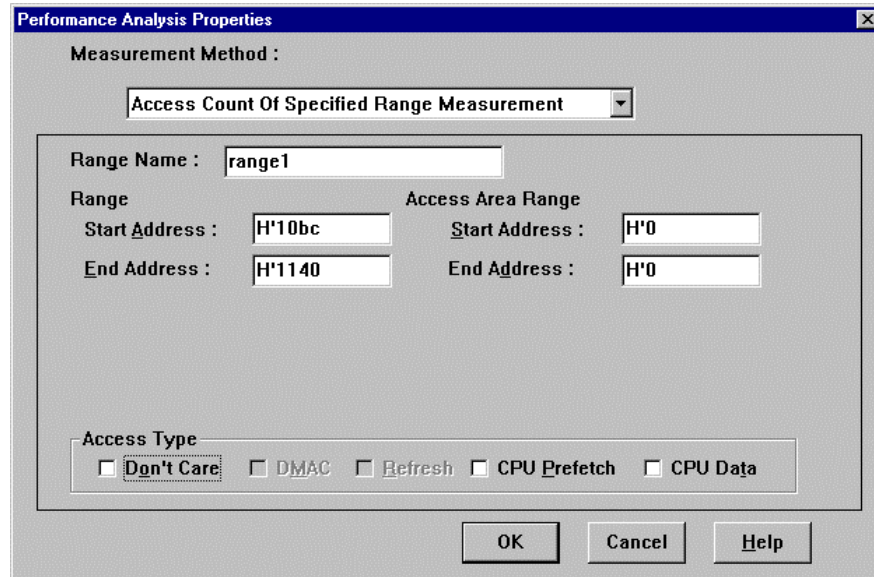


図 5-34 Performance Analysis Properties ダイアログボックス (4)

5. 指定範囲内コール回数測定 : Called Count Of Specified Range Measurement
 <開始アドレス>、<終了アドレス>で設定されている範囲から<コール範囲>で
 設定されている範囲をコールした回数を測定します。また、範囲内の実行時間は、
 指定範囲内時間測定を用いて測定します。
 <コール範囲>は、指定サブルーチンの開始アドレスと終了アドレスを指定して
 ください。

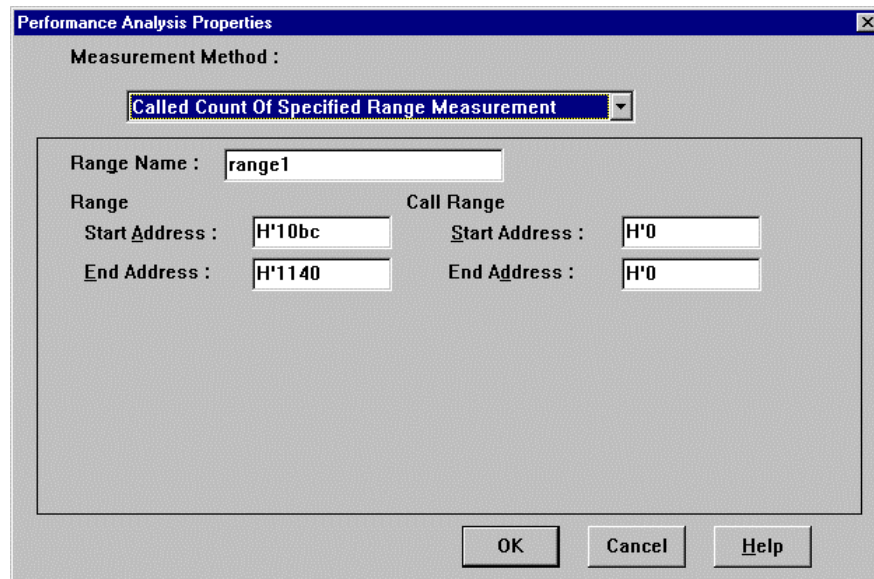


図 5-35 Performance Analysis Properties ダイアログボックス (5)

また、モード毎に設定するパラメータが異なります。

1．指定範囲内時間測定

Range Name	範囲の名称を指定します。
Range	指定範囲内時間測定を行う範囲を指定します。 Start Address : 開始アドレスを指定します。 End Address : 終了アドレスを指定します。

2．指定アドレス間時間測定

Range Name	範囲の名称を指定します。
Point	指定アドレス間時間測定を行う範囲を指定します。 Start Address : 開始アドレスを指定します。 End Address : 終了アドレスを指定します。
Time Out	実行時間測定タイムアウト値を指定します (チャンネル1使用時のみ有)。 測定時間の最小単位が 160ns、40ns、20ns の指定は、時 h 分 min 秒 s ミリ秒 ms マイクロ秒 us ナノ秒 ns (例: 1h 2min 3s 123ms 456us 789ns) で入力し、ターゲットクロック指定は、16進で10桁(例: 123456789A) と入力します。 開始、終了アドレス間の1回ごとの計測値がタイムアウト値を超えた時にブレイクします (トータル時間ではありません)。
Count	実行回数測定カウントアップ値を指定します (チャンネル1使用時のみ有)。

3．指定アドレス範囲間時間測定

Range Name	範囲の名称を指定します。
Start Range	指定アドレス範囲間時間測定を行う開始範囲を指定します。 Start Address : 先頭アドレスを指定します。 End Address : 最終アドレスを指定します。
End Range	指定アドレス範囲間時間測定を行う終了範囲を指定します。 Start Address : 先頭アドレスを指定します。 End Address : 最終アドレスを指定します。

4．領域アクセス回数測定

Range Name	範囲の名称を指定します。
Range	領域アクセス回数測定を行う範囲を指定します。 Start Address : 先頭アドレスを指定します。 End Address : 最終アドレスを指定します。
Access Area Range	領域アクセス回数測定を行うアクセス領域アドレス範囲を指定します。 Start Address : 先頭アドレスを指定します。 End Address : 最終アドレスを指定します。
Access Type	アクセス領域のバスサイクルを選択します。 Don't Care : 全アクセス CPU Prefetch : CPU プリフェッチサイクル CPU : CPU データサイクル

5 . 指定範囲内コール回数測定

Range Name	範囲の名称を指定します。
Range	指定範囲内コール回数測定を行う範囲を指定します。 Start Address : 先頭アドレスを指定します。 End Address : 最終アドレスを指定します。
Call Range	指定範囲内コール回数測定を行うコール範囲を指定します。 Start Address : 先頭アドレスを指定します。 End Address : 最終アドレスを指定します。 コール範囲は、指定サブルーチンの開始アドレスと終了アドレスを指定してください。

OK	設定した実行効率測定条件を登録し、このダイアログボックスを閉じます。 (Performance Analysis Conditions ダイアログボックスに戻ります。)
Cancel	設定した実行効率測定条件を登録せずに、このダイアログボックスを閉じます。 (Performance Analysis Conditions ダイアログボックスに戻ります。)

6 コマンドライン機能

本章では、E6000 H8/3160専用のコマンドライン機能について説明します。その他のコマンドライン機能については、「日立デバッグインタフェースユーザズマニュアル」を参照してください。以下にコマンドライン機能と「日立デバッグインタフェースユーザズマニュアル」(HDIマニュアル)および本マニュアルに記載する項目の対応表を示します。

表 6-1 HDI コマンドライン機能とマニュアルの対応表

コマンド名	短縮形	HDI マニュアル	本 マニュアル	説明
!	-		-	コメント
ACCESS	AC		-	不当アクセスに対する動作の設定
ANALYSIS	AN	-	6.1	性能分析機能の有効化/無効化
ANALYSIS_RANGE	AR	-	6.2	性能分析範囲の設定と表示
ANALYSIS_RANGE_DELETE	AD	-	6.3	性能分析範囲の解除
ASSEMBLE	AS		-	アセンブルの実行
ASSERT	-		-	コンディションのチェック
BREAKPOINT / EVENT	BP, EN	-	6.4	ブレークポイント/イベントの設定
BREAKPOINT_CLEAR EVENT_CLEAR	BC EC	-	6.5	ブレークポイント/イベントの解除
BREAKPOINT_DISPLAY EVENT_DISPLAY	BD ED	-	6.6	ブレークポイント/イベントの表示
BREAKPOINT_ENABLE EVENT_ENABLE	BE EE	-	6.7	ブレークポイント/イベントの有効化/ 無効化
BREAKPOINT_SEQUENCE EVENT_SEQUENCE	BS ES	-	6.8	シーケンスの定義および解除
CLOCK	CK	-	6.9	エミュレータのMCUクロック時間の設定
CVERAGE	CV	-	6.10	カバレッジトレース情報の表示
CVERAGE_CLEAR	CC	-	6.11	カバレッジトレース情報のクリア
DEVICE_TYPE	DE	-	6.12	エミュレータのデバイスタイプの選択
DISASSEMBLE	DA		-	逆アセンブル表示
EEPROM_DISPLAY	EP	-	6.13	EEPROMのプロテクト領域の表示
EEPROM_MAP	EM	-	6.14	EEPROMのプロテクト領域の設定/解除
ERASE	ER		-	コマンドインポートの内容のクリア
EVALUATE	EV		-	式の計算
FILE_LOAD	FL		-	オブジェクト(プログラム)ファイルのロード
FILE_SAVE	FS		-	メモリ内容のファイルセーブ
FILE_VERIFY	FV		-	ファイル内容とメモリ内容の比較
GO	GO		-	ユーザプログラムの実行
GO_RESET	GR		-	リセットベクタからのユーザプログラムの実行
GO_TILL	GT		-	テンポラリブレークポイントまでユーザプログラムの実行
HALT	HA		-	ユーザプログラムの停止

HDI コマンドライン機能とマニュアルの対応表 (つづき)

コマンド名	短縮形	HDI マニュアル	本 マニュアル	説明
HELP	HE		-	コマンドラインまたはコマンドに対するヘルプ表示
INITIALISE	IN		-	プラットフォームの初期化
INTERRUPTS	IR		-	プラットフォームの割り込み処理の有効化/無効化 (E6000 Eミュレータではサポートしません)
LOG	LO		-	ロギングファイルの操作
MAP_DISPLAY	MA		-	メモリマッピング情報の表示
MAP_SET	MS	-	6.15	メモリマッピングの設定
MEMORY_DISPLAY	MD		-	メモリ内容の表示
MEMORY_EDIT	ME		-	メモリ内容の変更
MEMORY_FILL	MF		-	指定データによるメモリ内容の一括変更
MEMORY_MOVE	MV		-	メモリブロックの移動
MEMORY_TEST	MT		-	メモリブロックのテスト
MODE	MO	-	6.16	MCU モードの設定と表示
QUIT	QU		-	HDIの終了
RADIX	RA		-	入力ラディックスの設定
REFRESH	RF	-	6.17	メモリ関連ウインドウの更新
REGISTER_DISPLAY	RD		-	MCU レジスタ値の表示
REGISTER_SET	RS		-	MCU レジスタ値の設定
RESET	RE		-	MCUのリセット
SLEEP	-		-	コマンド実行の遅延
STEP	ST		-	ステップ実行(命令単位またはソース行単位)
STEP_OUT	SP		-	PC位置の関数を終了するまでのステップ実行
STEP_OVER	SO		-	ステップオーバー実行
STEP_RATE	SR		-	ステップ速度の設定
SUBMIT	SU		-	Eミュレータコマンドファイルの実行
SYMBOL_ADD	SA		-	シンボルの追加
SYMBOL_CLEAR	SC		-	シンボルの削除
SYMBOL_LOAD	SL		-	シンボル情報ファイルのロード
SYMBOL_SAVE	SS		-	シンボル情報のファイルセーブ
SYMBOL_VIEW	SV		-	シンボルの表示
TEST_EMULATOR	TE	-	6.18	Eミュレータハードウェアのテスト
TIMER	TI	-	6.19	実行時間測定タイマ分解能の表示、設定
TRACE_ALL	TL		-	トレース情報の表示
TRACE_ACQUISITION	TA	-	6.20	トレース取得情報の設定と表示
TRACE_COMPARE	TC	-	6.21	トレース情報の比較
TRACE_SAVE	TV	-	6.22	トレース情報の保存
TRACE_SEARCH	TS	-	6.23	トレース情報の検索
USER_SIGNALS	US	-	6.24	ユーザーシグナル情報の有効化/無効化

6.1 ANALYSIS

省略形： AN

実行効率測定結果の表示をします。

フォーマット

AN

表 6-2 ANALYSIS コマンド

パラメータ	型 (キーワード)	説明
1	v	実行時間のパーセント、実行時間および実行回数の表示
	省略	実行時間のパーセント、ヒストグラムの表示

使用例

analysis v 実行結果をパーセント、時間、回数を数値で表示します。

an 実行結果をパーセント、ヒストグラムで表示します。

6.2 ANALYSIS_RANGE

省略形： AR

実行効率測定条件、アドレス検出モード、測定時間最少単位を設定する。また、設定内容を表示します。指定可能な実行効率測定条件は5種類です。

- ・ 指定範囲内時間測定
- ・ 指定アドレス間測定
- ・ 指定アドレス範囲間時間測定
- ・ 領域アクセス回数測定
- ・ 指定範囲内コール回数測定

全ての測定を指定範囲内時間測定または指定アドレス間測定を用いて行う場合は、最大8ポイントまで設定できます。ただし、指定アドレス範囲間時間測定、領域アクセス回数測定、指定範囲内コール回数測定の場合は、最大4ポイントまで設定できます。また、各ポイントNo.における測定条件設定に関する制限事項は、「5.5 Performance Analysisダイアログボックス」を参照してください。

6.2.1 指定範囲内時間測定

フォーマット

ARn Range <Name> <StartAddress> <EndAddress>

表 6-3 ANALYSIS_RANGE コマンド (1)

パラメータ	型 (キーワード)	説明
1 n	数値	設定するポイント番号 (n=1~8)
2 Range	Range	指定範囲内時間測定の設定
3<Name>	文字列	範囲の名称
4<StartAddress>	数値	開始アドレス
5<EndAddress>	数値	終了アドレス

6.2.2 指定アドレス間時間測定

フォーマット

ARn PtoP <Name> <StartAddress> <EndAddress> [<timeOpts>] [<CountOpts>]

表 6-4 ANALYSIS_RANGE コマンド (2)

パラメータ	型 (キーワード)	説明
1 n	数値	設定するポイント番号 (n=1~8)
2 PtoP	PtoP	指定アドレス間時間測定の設定
3 <Name>	文字列	範囲の名称
4 <StartAddress>	数値	開始アドレス
5 <EndAddress>	数値	終了アドレス
6 [<timeOpts>]	数値	実行時間測定タイムアウト値測定時間の最小単位が160ns、40ns、20nsの指定は、h時 分min 秒s ミリ秒ms マイクロ秒us ナノ秒ns (例: time 1h 2min 3s 123ms 456us 789ns) で入力し、ターゲットクロック指定は、16進で10桁 (例: H123456789A) と入力します。
7 [<CountOpts>]	数値	実行回数測定カウントアップ値 : 1~65535 (例) count 10

6.2.3 指定アドレス範囲間時間測定

フォーマット

ARn RtoR <Name> <StartRange> <EndRange>

表 6-5 ANALYSIS_RANGE コマンド (3)

パラメータ	型 (キーワード)	説明
1 n	数値	設定するポイント番号 (n=1,3,5,7)
2 RtoR	RtoR	指定アドレス範囲間時間測定の設定
3 <Name>	文字列	範囲の名称
4 <StartRange>	数値	開始アドレス範囲 <StartAddress>:<EndAddress> <StartAddress>:先頭アドレス <EndAddress> :最終アドレス
5 <EndRange>	数値	終了アドレス範囲 <StartAddress>:<EndAddress> <StartAddress>:先頭アドレス <EndAddress> :最終アドレス

6.2.4 領域アクセス回数測定

フォーマット

ARn AC <Name> <Range> <AccessRange> [bus<TypeOpts>] [<SizeOpts>]

表 6-6 ANALYSIS_RANGE コマンド (4)

パラメータ	型 (キーワード)	説明
1 n	数値	設定するポイント番号 (n=1,3,5,7)
2 AC	AC	領域アクセス回数測定の設定
3<Name>	文字列	範囲の名称
4<Range>	数値	アドレス範囲 <StartAddress>:<EndAddress> <StartAddress>:先頭アドレス <EndAddress> :最終アドレス
5<AccessRange>	数値	アドレス範囲 <StartAddress>:<EndAddress> <StartAddress>:先頭アドレス <EndAddress> :最終アドレス
6[bus<TypeOpts>]	cpupre cpu <default>	CPUプリフェッチバスサイクル CPUデータバスサイクル 全バスサイクル

6.2.5 指定範囲内コール回数測定

フォーマット

ARn RC <Name> <Range> <CallRange>

表 6-7 ANALYSIS_RANGE コマンド (5)

パラメータ	型 (キーワード)	説明
1 n	数値	設定するポイント番号 (n=1,3,5,7)
2 RC	RC	指定範囲内コール回数測定の設定
3<Name>	文字列	範囲の名称
4<Range>	数値	開始アドレス範囲 <StartAddress>:<EndAddress> <StartAddress> :先頭アドレス <EndAddress> :最終アドレス
5<CallRange>	数値	終了アドレス範囲 <StartAddress>:<EndAddress> <StartAddress> :先頭アドレス <EndAddress> :最終アドレス

6.2.6 実行効率測定条件の表示

フォーマット

ARn

表 6-8 ANALYSIS_RANGE コマンド (6)

パラメータ	型 (キーワード)	説明
1 n	数値	表示するポイント番号 (n=1~8)
	省略	全ポイントの設定条件を表示

6.2.7 アドレス検出モードの設定・表示

フォーマット

AR mode (p|pc)

表 6-9 ANALYSIS_RANGE コマンド (7)

パラメータ	型 (キーワード)	説明
1 mode	mode	アドレス検出モードの設定・表示
2 p	p	プリフェッチアドレス検出モードの設定
pc	pc	PCアドレス検出モードの設定
	省略	アドレス検出モードの表示

6.2.8 測定時間最小単位の設定・表示

フォーマット

AR time (160|40|20|t)

表 6-10 ANALYSIS_RANGE コマンド (8)

パラメータ	型 (キーワード)	説明
1 time	time	測定時間最小単位の設定・表示
2 160	160	160nsの設定
40	40	40nsの設定
20	20	20nsの設定
t	t	ターゲットの設定
	省略	測定時間最小単位の表示

使用例

ar1 range RAM 100 200

アドレス100から200に指定範囲内時間測定の実行効率条件を設定します。

ar1 PtoP RAM 200 2ff time
1h 2min 3s 0ms 0us 0ns

アドレス200から2ff、実行時間測定タイムアウト値を1時間2分3秒で指定アドレス間時間測定の実行効率条件を設定します。

ar3 RtoR RAM 100:1ff 200:
2ff

開始アドレス範囲100から1ff、終了アドレス範囲200から2ffで指定アドレス範囲間時間測定の実行効率条件を設定します

ar5 AC RAM 100:1ff 200:2ff
bus cpu

アドレス範囲100から1ff、アクセス領域200から2ff、CPUデータバスサイクルで、領域アクセス回数測定の実行効率条件を設定します。

ar7 RC RAM 100:11f 120:12f

アドレス範囲100から11f、コールするアドレス120から12fの範囲で、指定範囲内コール回数測定の実行効率条件を設定します。

ar mode p

アドレス検出モードをプリフェッチアドレス検出モードに設定します。

ar time

測定時間最小単位を表示します。

6.3 ANALYSIS_RANGE_DELETE

省略形： AD

指定した実行効率測定条件、又は全ての実行効率測定条件を削除します。

フォーマット

ADn

表 6-11 ANALYSIS_RANGE_DELETE コマンド

パラメータ	型 (キーワード)	説明
1	数値	削除するポイント番号 (n=1 ~ 8)
	省略	全てのポイントを削除する

使用例

```
analysis_range_delete3   ポイント番号 3 の実行効率条件を削除します。
ad4                       ポイント番号 4 の実行効率条件を削除します。
ad                         全ての実行効率条件を削除します。
```

6.4 BREAKPOINT/EVENT

省略形： BP, EN

ブレークポイントを設定します。このコマンドには、異なったタイプのブレークポイントが設定可能であり、いくつかのフォーマットがあります。

指定可能なブレークポイントは下記 3 種類です：

- プログラムブレークポイント
- アクセスブレークポイント
- 範囲ブレークポイント

6.4.1 プログラムブレークポイント

フォーマット : bp program address
 : bp p address

PC Breakpoint はアドレスを指定します。

6.4.2 アクセスブレークポイント

フォーマット : bp access address [options]
 : bp a address [options]

オプション：

```
<options> = [<dataopts>] [read | write] [<signalopts>] [<busopts>] [<areaopts>]
           [<actionopts>] [count <countval>] [delay <delayval>] [channel
           <channelno>]
<dataopts> = data <data> [mask <mask>] [byte | word]
<signalopts> = signal ((1 | 2 | 3 | 4) (high | low))+
<busopts> = bus (cpu | cpupre )+
<areaopts> = area (io | iram | irom | eeprom )+
<actionopts> = action (break | (timer (start | stop)) | none | trace)+
<channelno> = 1..12
```

bus オプションで指定できるパラメータです。

cpu、cpupre

area オプションで指定できるパラメータです。

io、iram、irom、eeprom

アクセブ레이크ポイントは、CPU が指定された方法で指定されたアドレスにアクセスした時ブレークします。

6.4.3 範囲ブレークポイント

フォーマット: bp range [outside] <address low> <address hi> [<options>]

<options>の指定方法は、アクセブ레이크ポイント（前述）と同様です。

このコマンドでは CPU が前述のアドレス範囲に対し、指定されるアドレスの範囲内または範囲外のどちらかのアクセスで成立するブレークポイントを設定します。

6.4.4 オプション

data <data> [mask <mask>] [byte | word]

データ比較条件を指定します。マスクを指定すると、マスク値の 0 を設定したビットの比較を行いません。例: *data h' 20 mask h'fff0 word*. これはデータバスの上位 12 ビットが h'002 のときイベントが成立します。

デフォルトはデータを比較しません。

signal ((1 | 2 | 3 | 4) (high | low)) +

このオプションで指定された信号が指定した状態になったときイベントが成立します。

例: *signal 1 high 3 low*. これは信号 1 が high、信号 3 が low の場合にイベントが成立します。（他の信号の値はチェックされません）

デフォルトは全ての信号を無視します。

bus (cpu | cpupre) +

CPU バスが指定された状態の内の 1 つの状態になったときイベントが成立します。例: *bus cpupre*. これはバス状態が CPU プリフェッチの場合にイベントが成立します。

デフォルトはバスサイクルタイプを無視します。

area (io | iram | irom | eeprom) +

'bus...' と同様にこのオプションは指定されたエリアがアクセスされている場合にイベントが成立します。例: *area irom iram*. これは内部 ROM か内部 RAM がアクセスされている場合にイベントが成立します。

デフォルトはあらゆるメモリです。

action (break | (timer (start | stop)) | none | trace) +

イベントが成立したときに action が起こるように定義します。

デフォルトの action はブレークになっています。他のオプションは、イベントタイマの開始および停止、action なし（イベントシーケンスの一部を定義するのに役に立ちます）およびイベントをトレースのサブシステムに割当てます（trace_acquisition コマンドで使用できます）。トレースを指定すると、他の action のいずれも使用することはできません。

count <countval>

イベントバスカウントを設定します（event detector の指定が必要です）。バスサイクル単位でカウントします（10 進指定）。

delay <delayval>

イベント成立後の指定された action が発生するまでのディレイカウントを設定します。バスサイクル単位でディレイをカウントします（10進指定）。

channel 1..12

イベントシステムのチャンネル番号を設定します。チャンネル番号の指定によりイベントの順位づけが可能となり、イベントシーケンスのセットアップに役立ちます。（event_sequence 参照）チャンネル 1～8 は event detectors、9～12 は range detectors です。

例:

en access 100	アドレス 100 にアクセスブレイクポイントを設定します。
bp p 110	アドレス 110 にプログラムブレイクポイントを設定します。
en access 100 data 55 byte	アドレス 100、データ 55 にアクセスブレイクポイントを設定します。
bp range 12 45	アドレス 12～45 に範囲ブレイクポイントを設定します。
bp range outside 60 89	アドレス 60～89 の範囲外に範囲ブレイクポイントを設定します。
bp a 200 read	アドレス 200 にリードのアクセスブレイクポイントを設定します。
bp a 500 write	アドレス 500 にライトのアクセスブレイクポイントを設定します。
en a 500 write action trace	アドレス 500 にライトでイベントが成立するようトレースのサブシステムに割当てます。

6.5 BREAKPOINT_CLEAR / EVENT_CLEAR

省略形: BC, EC

このコマンドは登録済みのブレイクポイントを削除します。

表 6-12 BREAKPOINT_CLEAR / EVENT_CLEAR コマンド

パラメータ	型 (キーワード)	説明 (ブレイクポイントタイプ)
1 to 2	program <address>	プログラムブレイクポイントをクリアします
	access <address> <options>	アクセスブレイクポイントをクリアします
	range <address> <options>	範囲ブレイクポイントをクリアします
	all	全てのブレイクポイントを削除します
	all trace	全てのトレースイベントを削除します
	channel 1..12	チャンネル番号のイベントを削除します

<options>は event コマンドの指定と同様です。イベントの指定にはアドレスおよびイベントが識別できる最小限の固有オプションの指定が必要です。トレースのサブシステムに割当てられるイベントを削除するためには、"action trace"を指定しなければなりません（デフォルトはブレイクポイントサブシステムからのイベント削除です）。

例:

bc p 256	アドレス 256 のプログラムブレイクポイントをクリアします。
event_clear chan 5	チャンネル番号を使用してイベントを削除します。
bc all	全てのブレイクポイントをクリアします。
ec all trace	全てのトレースイベントをクリアします。
ec access h'100 action trace	H'100 のトレースのアクセスイベントをクリアします。

6.6 BREAKPOINT_DISPLAY / EVENT_DISPLAY

省略形: BD, ED

カレントのブレークポイント表示します。このコマンドはトレースに割当てられるイベントおよびブレークポイントサブシステムを表示します。トレースイベントは"trace"と表示します。

例:

```
bd 全てのブレークポイント設定を表示します。
```

6.7 BREAKPOINT_ENABLE / EVENT_ENABLE

省略形: BE, EE

指定したブレークポイントまたは全てのブレークポイントを有効または無効にします。

表 6-13 BREAKPOINT_ENABLE / EVENT_ENABLE コマンド

パラメータ	型 (キーワード)	説明
1	true	ブレークポイント有効
	false	ブレークポイント無効
2	all	全てのブレークポイント
	program <address>	プログラムブレークポイント
	access <address> <options>	アクセスブレークポイント
	range <address1> <address2> <options>	範囲ブレークポイント
	channel 1..12	チャンネル番号のイベントを有効または無効にします

<options>は event コマンドの指定と同様です。イベントの指定にはアドレスおよびイベントが識別できる最小限の固有オプションの指定が必要です。トレースのサブシステムイベントを無効にするためには、"action trace"を指定しなければなりません。

例:

```
be true all      全てのブレークポイントを有効にします。
be false all     全てのブレークポイントを無効にします。
be false p 256   アドレス 256 のプログラムブレークポイントを無効にします。
be true access 12 12 のアクセスブレークポイントを有効にします。
be false chan 1  event detector チャンネル 1 を無効にします。
```

6.8 BREAKPOINT_SEQUENCE / EVNET_SEQUENCE

省略形: BS, ES

フォーマット:

```
bs <channel> [armed_by [not] <chan1> <chan2> ...]
             [armed_by off]
             [reset_by <chan1> <chan2> ...]
             [reset_by off]
```

定義済みのイベントを arm イベントまたは reset イベントに設定します。このコマンドはシーケンストレースとブレークポイントイベントに使用可能です。ただし、ブレークポイントとトレースとを組み合わせたシーケンシャル条件を設定することはできません。

例:

bs 1 armed_by 2 3

イベント 2 または 3 イベント 1 の順序でイベントが検出されたとき、シーケンスが成立します。番号は event detectors のチャンネル番号で、event コマンドの channel オプションが 1~8 に設定されたイベントを使用できます。

bs 2 reset_by 4

イベント 4 イベント 2 の順序でイベントが検出されたとき、シーケンスがリセットされます。Off を使用すると、指定イベントの arm イベントおよび reset イベントを無効にし、イベントを独立させます。

6.9 CLOCK

省略形: CK

外部入力クロック、レートを選択または表示します。パラメータがなければクロックとレートを表示します。クロックかレートが変更されると、エミュレータシステムはリセットされます。

表 6-14 CLOCK コマンド

パラメータ	型 (キーワード)	説明 (選択クロック)
1	3	3.5712MHz 内部クロック
	4	4.9152MHz 内部クロック
	7	7.1424MHz 内部クロック
	9	9.8304MHz 内部クロック
	t	ターゲット

注: ターゲットシステムクロックはターゲットシステムの Vcc が供給されている場合にのみ選択可能です。

6.10 COVERAGE

省略形: CV

フォーマット:

cv <StartAddress> <EndAddress> <DisplayOption> [not]

<StartAddress> = カバレッジトレース表示の開始アドレス

<EndAddress> = カバレッジトレース表示の終了アドレス

<DisplayOption> = (a | d)

a: アドレス値表示

d: ダンプ表示

not = 実行されなかったアドレスの表示指定 (DisplayOption が a の時のみ有効)

ユーザプログラム実行中に実行されたプログラムアドレスのカバレッジトレース情報をアドレス値またはダンプ表示します。対象となるのはプログラム領域のみです。

カバレッジトレース情報の記録は起動時から行います。カバレッジトレース情報のクリアは、coverage_clear または初期化(HDI メニュー File/Initialise)により行うことができます。ダンプ表示する場合、指定した開始アドレスは 16 の倍数、終了アドレスは 16 の倍数-1 に補正されます。

注: 4 バイト命令の 3, 4 バイト目のアドレスは、カバレッジトレース情報として取得されません。

アドレス値表示のフォーマット例

<S-ADDR> <E-ADDR> <S-ADDR> <E-ADDR> <S-ADDR> <E-ADDR>
 xxxxxxxx-xxxxxxx xxxxxxxx-xxxxxxx xxxxxxxx-xxxxxxx

実行したアドレスを範囲で表示します。
 「not」を指定した場合は、指定したアドレス範囲で実行されなかったアドレスを表示します。

ダンプ表示のフォーマット例

<ADDRESS> <DATA>
 xxxxxxxx yy yy yy yy yy yy yy yy - yy yy yy yy yy yy yy yy

xxxxxxx アドレス
 yy 実行アドレス情報を 16 進数 (00-FF) で表示します。1bit が 2
 アドレスに対応しており、実行したアドレスは '1' として表
 示します。

(例) 00001000 8F 00
 先頭の 8F というデータは、1000-1, 1008-9, 100A-B, 100C-D,
 100E-F を実行したことを示します。

例:

cv 400 7FFF a 実行アドレスの 400 から 7FFF の範囲をアドレス値で表示します。
 cv 1000 6FFF d 実行アドレスの 1000 から 6FFF の範囲をダンプ表示します。
 cv 100 7FFF a not 実行アドレスの 100 から 7FFF の範囲で実行されなかったアドレ
 スを表示します。

6.11 COVERAGE_CLEAR

省略形: CC

カバレッジトレース情報をクリアします。
 カバレッジトレース情報の記録は起動時から行います。カバレッジトレース情報のクリアは、
 本コマンドまたは初期化(HDI メニューFile/Initialise)によって行うことができます。

6.12 DEVICE_TYPE

省略形: DE

デバイスタイプのセットアップまたはカレントのデバイスタイプを表示します。

表 6-15 DEVICE_TYPE コマンド

コマンド	説明
de	デバイスタイプを表示します。
de デバイスタイプ	デバイスタイプで指定したデバイスを設定します。

例:

de H8/3113 H8/3113 を設定します。
 de 3153 H8/3153 を設定します。

6.13 EEPROM_DISPLAY

省略形: EP

EEPROM のプロテクト領域の表示を行ないます。表示するアドレス領域は、ページ単位です。

6.14 EEPROM_MAP

省略形: EM

EEPROM のプロテクト領域の設定、解除を行ないます。

フォーマット:

EM <start address> <end address> <protection>

<protection> = p | n

p: プロテクト設定

n: プロテクト解除

例:

em 8000 807f p

アドレス H'8000 から H'807f に含まれるページをプロテクトに設定します。

6.15 MAP_SET

省略形: MS

このオプションによりエミュレータのメモリマップを変更します。

フォーマット:

ms <start address> <end address> <where> <protection>

<where> = on-chip | internal

<protection> = none | read-only | guarded

On-chip メモリ (例 内蔵 RAM、ROM、I/O、EEPROM またはリザーブエリア) はデバイス内蔵のメモリです。基本的には、メモリマップの属性変更はできません。ただし、リザーブエリアのみ、Internal を指定することで、emulation メモリ (プロテクトなし) の設定が可能です。

Internal メモリはエミュレータ内部のメモリです。ただし、デバイス内蔵のメモリではありません。

例:

ms C200 EFFF internal none

C200 から EFFF までを内部のアクセス可能メモリとして設定します。

6.16 MODE

省略形: MO

CPU モードを設定または表示します。

例:

mode カレントのモードを表示します。

6.17 REFRESH

省略形: RF

メモリ関連ウィンドウを更新します。

6.18 TEST_EMULATOR

省略形: TE

エミュレータハードウェアおよびエミュレータメモリエリアをテストします。このコマンドを実行後、エミュレータシステムを必ず再初期化してください。(HDI メニュー File/Initialise)

例:

te エミュレータテストを実行します。

6.19 TIMER

省略形: TI

時間計測の最小時間を表示または変更します。これは実行時間とイベントタイミングに影響します。

表 6-16 TIMER コマンド

コマンド	説明
ti	時間計測の最小時間を表示します。
ti <timer resolution>	時間計測の最小時間を設定します。

時間計測の最小時間は以下の通りです:

20ns 125ns 250ns 500ns 1us 2us 4us 8us 16us

6.20 TRACE_ACQUISITION

省略形: TA

Trace Acquisition オプションを設定または表示します。

フォーマット:

ta [<freetrace>] [<timestamp>] [<stop>] [<stopdelay>] [<range>]

<freetrace> = freetrace (true|false)

<timestamp> = timestamp (disable | 125ns | 250ns | 500ns | 1us | 2us | 4us | 8us | 16us | 100us)

<stop> = stop (disable | event <1~12>)

<stopdelay> = stopdelay (disable | event <1~12> [count <count>])

<range> = range <1~4> (disable | ptop <startaddr> <stopaddr> [cyclic] |

```

range <1 ~ 12> | event <1 ~ 8> <1 ~ 8> [cyclic] )
<default> = default

```

例:

<pre> ta ta stop event 1 2 ta stopdelay event 1 2 count 100 ta timestamp 500ns ta range 1 ptop H'100 H'10a ta range 1 range 3 ta range 2 event 4 5 ta range 2 event 4 5 cyclic </pre>	<p>全てのトレース取得オプションを表示します。</p> <p>チャンネル1またはチャンネル2のどちらかのイベントが成立したとき、トレース取得を停止します。</p> <p>チャンネル1またはチャンネル2のイベントが成立後に100バスサイクルトレース取得し停止します。</p> <p>トレースのタイムスタンプを有効にし、時間計測の最小時間を500nsに設定します。</p> <p>アドレスH'100をアクセスしたときトレース取得を開始し、アドレスH'10aをアクセスしたときトレース取得を停止します。</p> <p>トレース取得条件1にイベント3をrangeモードで設定します。</p> <p>トレース取得条件2にイベント4成立でトレース取得を開始し、イベント5成立でトレース取得を停止するeventモード条件を設定します(cyclicではありません)。</p> <p>トレース取得条件2にイベント4成立でトレース取得を開始し、イベント5成立でトレース取得を停止するeventモード条件をcyclicで設定します。cyclic指定を行なった場合はトレース取得条件が成立する毎にトレースを取得します。</p>
--	--

6.21 TRACE_COMPARE

省略形: TC

カレントのトレースデータと、セーブしたトレースファイル(trace_save 参照)を比較します。

フォーマット:

```
trace_compare <filename>
```

6.22 TRACE_SAVE

省略形: TV

トレースデータをバイナリフォーマットでファイルにセーブします。trace_compare コマンドを使用するとトレースとセーブしたデータを比較することができます。

フォーマット:

```
trace_save <filename>
```

6.23 TRACE_SEARCH

省略形: TS

トレースを検索します。指定条件に合うトレース取得サイクルを検索します。

フォーマット:

```
ts [<address>] [<dataopts>] [<signalopts>] [<busopts>] [<areaopts>]
    [<directionopts>] [<timestampopts>] [<fromopts>]
<address>    = address <address> [to <address>]
<dataopts>   = data <data> [mask <mask>] (byte | word)
<signalopts> = signal <sig><sig><sig><sig>
               <sig> = (1|0|x)      1 = high, 0 = low, x = don't care
<busopts>    = bus (cpu | cpupre)+
<areaopts>   = area (io | iram | irom | eeprom)+
<directionopts> = dir (read | write | either)
<timestampopts> = time <start> [ to <stop>]
<fromopts>   = from <record>
```

bus オプションで指定できるパラメータです。

cpu、cpupre

area オプションで指定できるパラメータです。

io、iram、irom、eeprom

timestamp オプションの<start>および<stop>の値には検索するタイムスタンプ値の範囲を指定します。タイムスタンプ値のフォーマットは以下の通りです:

秒 *S* ミリ秒 *ms* マイクロ秒 *us* ナノ秒 *ns*

例:

2s 123ms 400us 125ns

0s 000ms 100us 000ns

“time”オプションは、Trace Acquisition のタイムスタンプが有効なトレースサイクルの場合にだけ指定可能であることに注意してください。タイムスタンプが有効のときは“area”、“signal”および“bus”オプションは指定できません。

6.24 USER_SIGNALS

省略形: US

ユーザ信号 (Reset) の有効または無効を設定します。パラメータがなければ、Reset の有効 / 無効の状態を表示します。

表 6-17 USER_SIGNALS コマンド

型 (キーワード)	説明
us	ユーザ信号の状態を表示します。
us enable reset	信号を有効にします。
us disable reset	信号を無効にします。

7 故障解析

本章では、E6000エミュレータ用テストプログラムによる故障解析の手順について示します。

7.1 テストプログラムを実行するためのシステムセットアップ

- (1) テストプログラムを実行するためには、以下に示す機器が必要です。なお、本テストプログラムの実行にはユーザシステムインタフェースケーブルおよびユーザシステムは不要です。
 - ・ H8/3160用E6000エミュレータ(HS3160EPI60H)
 - ・ E6000 PCインタフェースボード(HS6000EII01H)
 - ・ ISAバス仕様ホストコンピュータ(MS-DOS環境)
- (2) ホストPCにE6000 PCインタフェースボードを挿入し、付属のPCインタフェースケーブルを接続してください。
- (3) PCインタフェースケーブルをH8/3160用E6000エミュレータに接続してください。
- (4) H8/3160用E6000エミュレータに、付属のACアダプタを接続してください。
- (5) ホストPCを起動し、MS-DOSのコマンド入力待ち状態にしてください。
- (6) H8/3160用E6000エミュレータの電源をオンにしてください。

7.2 テストプログラムによる故障解析

E6000エミュレータに添付されているCD-R(HS3160EPI60SR)をShiftキーを押しながらPCのCD-ROMドライブに挿入し、コマンドプロンプトでカレントディレクトリを <ドライブ>:¥Diagフォルダに移動した後、使用しているPCインタフェースボードの種類に従い、下記コマンドを入力すると直ちにテストプログラムが起動します。なお、テストプログラムの実行に際しては、HDIがインストールされていることを前提としています。

```
>TM3160(RET)
```

カレントディレクトリを <ドライブ>:¥Diagフォルダに移動しない状態で

```
> <ドライブ>:¥Diag¥TM3160(RET)
```

のように他のカレントディレクトリからテストプログラムを起動した場合はテストプログラムが正しく動作しません。必ず <ドライブ>:¥Diagフォルダにカレントディレクトリを移動してテストプログラムを実行してください。

なお、> TM3160 -S (RET) のように、-S をコマンドラインに追加すると、No.1からNo.21までのテストを繰返し実行することができます。途中でテストを中断する場合は Q を入力してください。

注：Shiftキーを押さずにCD-RをCD-ROMドライブに挿入した場合、HDIインストールウィザードが自動的に起動します。

HDIインストールウィザードが自動起動した場合はHDIインストールウィザードを終了させてください。

注：<ドライブ>はCD-ROMドライブのドライブ文字です。

注：テストプログラム実行中はCD-ROMドライブからCD-Rを取り出さないでください。

7.故障解析

```
E6000 H8/3160 EMULATION BOARD Tests Vn.m
Hitachi Ltd (1998,1999)

Searching for interface card.....OK, card at H'd0000000

Checking emulator is connected .....OK

Emulator Board Information:
Main Board ID      H'1

Emulation Board ID H'c

Revision           H'x

SIMM               No SIMM module inserted

Downloading firmware .....

01) Testing Register :
  Emulation Board ID .....H'c
  BOC control register .....OK
  MODER register .....OK
  EEMAX register .....OK
  CES GA register .....OK
02) Testing Dual-Port RAM :
  Decode Test .....OK
  Marching Test .....OK
03) Testing Firmware RAM :
  Decode Test.  page range H'700 - H'70f .....OK
  Marching Test. page range H'700 - H'70f .....OK
04) Testing Trace RAM :
  Decode Test.  page range H'000 - H'04f .....OK
  Marching Test. page range H'000 - H'04f .....OK
05) Testing Map Control RAM :
  Decode Test.  page range H'200 - H'21f .....OK
  Marching Test. page range H'200 - H'21f .....OK
06) Testing Coverage RAM :
  Decode Test.  page range H'400 - H'41f ...OK
07) Testing PERF GA :
  PERF GA register .....OK
08) Testing Emulation RAM :
```

テストプログラムのスタートメッセージです。Vn.mはバージョン番号です。

ホストPCにPCインタフェースボードが正しく接続されていることを示します。また、このときのアドレスを表示します。値はアドレス設定値によって変わります。

ホストPCとE6000エミュレータが正しく接続されていることを示します。

E6000エミュレータ（下基板）のID番号で、常に1を示します。
E6000エミュレータ（上基板）のID番号で、常にcを示します。
E6000エミュレータ（上基板）のレビジョン番号をxで示します。
E6000エミュレータに実装されているエミュレーションメモリの容量を示します。

テスト用のプログラムをロードしていることを示します。

E6000エミュレータ上のレジスタのテストチェック結果（正常終了）を示します。

E6000エミュレータ上のDual-Port RAMのデコードテスト、マーチングテストチェック結果（正常終了）を示します。

E6000エミュレータ上のFirm RAMのデコードテストチェック結果（正常終了）を示します。
マーチングテストチェック結果（正常終了）を示します。

E6000エミュレータ上のTrace RAMのデコードテストチェック結果（正常終了）を示します。
マーチングテストチェック結果（正常終了）を示します。

E6000エミュレータ上のMap Control RAMのデコードテストチェック結果（正常終了）を示します。
マーチングテストチェック結果（正常終了）を示します。

E6000エミュレータ上のCoverage RAMのデコードテストチェック結果（正常終了）を示します。

実行時間測定回路のレジスタチェック結果（正常終了）を示します。
内蔵ROMおよび内蔵RAMのデコード

Normal Window:Decode Test	OK	
Normal Window:Marching Test	OK	
09) Testing STEP Operation :		テスト、マーチングテストチェック結果（正常終了）を示します。
Step Operation	OK	ステップ実行制御回路のチェック結果（正常終了）を示します。
10) Testing Key Break :		強制ブレーク制御回路のチェック結果（正常終了）を示します。
Key Break	OK	不当アクセスブレーク制御回路のチェック結果（正常終了）を示します。
11) Testing Emulation RAM Hardware Break :		内蔵のROMのデコートテスト、マーチングテストチェック結果、制御回路テストチェック結果（正常終了）を示します。
GRD Break	OK	
WPT Break	OK	
12) Testing Internal ROM :		
Write-Protect Test.....	OK	
ROM Bank0 Test		
Normal Window:Decode Test	OK	
Normal Window:Marching Test	OK	
ROM Bank1 Test		
Normal Window:Decode Test	OK	
Normal Window:Marching Test	OK	
ROM Bank alternation Test	OK	
ROM Bank function Test.....	OK	
Register Test.....	OK	
13) Testing EEPROM :		内蔵EEPROMへの書き込み / 重ね書き / 消去とその時間、EEMAXブレーク制御回路のチェック結果（正常終了）を示します。
A) EEPROM TEST 1	OK	
B) EEPROM TEST 2	OK	
C) EEPROM TEST 3	OK	
D) EEPROM DELETE TEST	OK	
E) RESERVED	OK	
F) EEPROM PROTECT TEST	OK	
G) EEMAX TEST 1	OK	
H) EEMAX TEST 2	OK	
I) EEMOV ON EEPROM TEST	OK	
J) EEPMOV TIME MEASUREMENT TEST.....	OK	
14) Testing Mapping :		マッピング制御回路のチェック結果（正常終了）を示します。
A)ACCESS ERROR TEST	OK	
B)WRITE PROTECT ERROR TEST	OK	
C)ROM WRITE PROTECT ERROR TEST	OK	
15) Testing Trace :		トレース制御回路のチェック結果（正常終了）を示します。
Free Trace Test	OK	
Range Trace Test	OK	
Point to Point Trace Test	OK	
Start and Stop Event Trace Test	OK	
16) Testing Hardware Break :		ハードウェアブレーク制御回路のチェック結果（正常終了）を示します。
A)Break Point Intialised	OK	
B)Event Detectors CES channel 1-12	OK	
C)CES Event Sequencing Test	OK	
D)Range Break Point Test	OK	
E)Range Break Point Test for DATA	OK	
F)Compare break either Test	OK	
17) Testing PC_CVR :		カバレジ制御回路のチェック結果（正常終了）を示します。
A)ALL Coverage Test	OK	
B)Area Coverage Test 1	OK	
C)Area Coverage Test 2	OK	
18) Testing PERM_GA :		実行時間測定回路のチェック結果（正常終了）を示します。
A)Time Measure Test	OK	
B)PERM_POINT TO POINT Time Measure Test	OK	

7.故障解析

- C)PERM_SUBROUTINE Time Measure TestOK
- D)PERM Time Out Bit Test
 - Time Out Test 1.....OK
 - Time Out Test 2.....OK
- 19) Testing Double Stack,I/O Stack :
 - A)Double Stack TestOK
 - B)I/O Stack Test 1OK
 - C)I/O Stack Test 2OK
 - D)I/O Stack Test 3OK
- 20) Testing Emulation RAM Time Measurement:
 - Testing Internal Clock = 9.8304MHzOK
 - Testing Internal Clock = 7.1424MHzOK
 - Testing Internal Clock = 4.9152MHzOK
 - Testing Internal Clock = 3.5712MHzOK
- 21) Testing Emulation Monitor :
 - A)A15 to A0 (MONIT00, MONIT1E, MONIT10)OK
 - B)INST bit (MONIT20:D1).....OK
 - C)ASE STATUS BIT(MONIT2E:D7-D4)OK
 - D)HLT BIT(MONIT20:D6).....OK
 - E)CNN BIT(MONIT2E:D1).....OK
 - F)ASE BRKACK BIT(MONIT20:D0).....OK
 - G)EVAGDD BIT(MONIT20:D5).....OK

ブレーク時のスタックポインタ不正検出回路のチェック結果（正常終了）を示します。

各エミュレータ内部クロックでの実行時間測定回路のチェック結果（正常終了）を示します。

モニタ制御回路のチェック結果（正常終了）を示します。

-S をコマンドラインに追加した場合は、No.21のテスト終了後、再びNo.1のテストから繰返し実行されます。

本テストプログラムは不具合を検出するとERRORを表示してプログラムの実行を中止します。この場合、エミュレータハードウェアの故障が考えられます。発生したエラー内容の詳細を当社の購入営業担当までご連絡ください。

7.3 エラー発生時の処理

E6000エミュレータをご使用中に動作エラーが発生した場合は、お手数ですが下記故障症状調査書に症状をご記入のうえ、担当営業までFAXでご連絡いただくようお願い申し上げます。

故障症状調査書

ご購入営業担当行

お客様ご芳名 会社名 _____
 ご担当者名 _____ 様
 TEL _____
 FAX _____

- 1) 不具合発生製品型名およびシステム構成
 - a) E6000エミュレータ(HS3160EP160H)
 シリアルNo. _____、レビジョン _____
 (ケース裏面に表示しています：シリアルNo.は数字4桁、レビジョンはそれに続くアルファベットです)
 - b) PCインタフェースボード
 型式 HS _____ H、シリアルNo. _____、レビジョン _____
 (基板上に捺印表示しています)
 - c) ユーザシステムインタフェースケーブル
 型式 HS _____ H、シリアルNo. _____、レビジョン _____
 (基板上に捺印表示しています)
 - d) HDI(HS3160EP160SR) バージョンV _____
 (CD-RにVx.xxと表示しています)
 - e) ご使用になっているPC
 メーカー名 _____、型式 _____
 使用OS (Windows®95, Windows®98, WindowsNT®4.0, Windows®2000 いずれかに)
- 2) ターゲットシステムの使用条件
 - a) デバッグ対象マイコン型名： _____
 - b) ターゲットシステム電圧： _____ V
 - c) 使用クロック：(貸出しクロック、外部クロック入力 いずれかに)
 - d) 動作周波数： _____ MHz

7.故障解析

3) エラー発生状況

a~bのいずれかに をつけ、内容を記載してください。

a) HDIがLink upしない

(エラーメッセージ: _____)

b) テストプログラムでエラーが発生

(エラーの発生したテストNo. _____)

(エラーメッセージ: _____)

c) デバッグ中にエラーが発生

下記の各項目についてエラー内容を記載してください。

4) メモリのデータ化けは発生していますか? (はい、 いいえ どちらかに)

a) データ化けを起こしている箇所のMemory Mapping設定

(ROM, RAM, I/O, Emulator, その他 _____ どちらかに)

b) データ化けを起こしている箇所はMemory Windowでリード/ライトできますか?

(はい、 いいえ どちらかに)

5) ターゲットシステムに対して入出力できない特定の信号はありますか?

(はい、 いいえ どちらかに)

a) 信号名: _____

b) 信号レベル異常: (High固定、Low固定、中間レベル どちらかに)

6) 上記以外のエラーについては、下記に症状を記載いただくようお願いいたします。

--

E6000 H8/3112、H8/3113、H8/3150、H8/3160、AE-3 シリーズ
エミュレータ
HS3160EPI60H ユーザーズマニュアル



ルネサスエレクトロニクス株式会社
神奈川県川崎市中原区下沼部1753 〒211-8668