

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

ご注意

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ (<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。

E6000 H8/3887シリーズ、H8/3867シリーズ、
H8/3847シリーズ、H8/3827シリーズ、
H8/3847Rシリーズ、H8/3827Rシリーズ用エミュレータ
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

HS388REPI60H

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、（４）オプションボード

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 H8/3887 シリーズ、H8/3867 シリーズ、H8/3847 シリーズ、H8/3827 シリーズ、H8/3847R シリーズ、H8/3827R シリーズ用（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、死亡又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、死亡又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、軽傷又は中程度の傷害を招くこと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、財物損傷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. 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、すべ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を破壊する可能性があります。
3. エミュレータまたはユーザシステムのパワーオン時、エミュレータと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インタフェースケーブルとユーザシステム上の IC ソケット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を破壊する可能性があります。
4. 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上の IC ソケットはピン番号を確かめて正しく接続してください。接続を誤ると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。
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。

まえがき

本書について

本エミュレータ (HS388REPI60H) は、以下のデバイス (MCU) をサポートします。なお、本文中はデバイス名は表記せず、MCUのみ表記します。

サポートデバイス	MCU
H8/3887、3886、3885、3884、3883、3882	H8/3887シリーズ
H8/3867、3866、3865、3864、3863、3862	H8/3867シリーズ
H8/3847、3846、3845、3844、3843、3842	H8/3847シリーズ
H8/3827、3826、3825、3824、3823、3822	H8/3827シリーズ
H8/3847R、3846R、3845R、3844R、3843R、3842R	H8/3847Rシリーズ
H8/3827R、3826R、3825R、3824R、3823R、3822R	H8/3827Rシリーズ

本書は、H8/3802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長を紹介합니다。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 H8/3887シリーズ、H8/3867シリーズ、H8/3847シリーズ、H8/3827シリーズ、H8/3847Rシリーズ、H8/3827Rシリーズ E6000 エミュレータ 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、H8/3887シリーズ、H8/3867シリーズ、H8/3847シリーズ、H8/3827シリーズ、H8/3847Rシリーズ、H8/3827Rシリーズ用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
- ・ユーザシステムインタフェースケーブル取扱い説明書
- ・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
1.1	デバッグの特長	1-1
1.1.1	ブレークポイント	1-1
1.1.2	トレース	1-1
1.1.3	実行時間測定	1-2
1.2	イベント検出システム (CES: Complex Event System)	1-3
1.2.1	イベントチャンネル	1-3
1.2.2	範囲チャンネル	1-4
1.2.3	ブレーク	1-4
1.2.4	イベント間実行時間測定	1-4
1.3	ハードウェアの特長	1-5
1.3.1	メモリ	1-5
1.3.2	動作電圧および動作周波数	1-6
1.3.3	エミュレーションクロック	1-7
1.3.4	外部プローブ	1-7
1.3.5	使用環境条件	1-8
1.3.6	外形寸法と質量	1-8
2	セットアップ	2-1
2.1	パッケージ内容	2-1
2.2	PC インタフェースボードのセットアップ	2-2
2.2.1	Windows [®] 98 での PC インタフェースボードのセットアップ	2-2
2.3	Windows NT [®] 4.0 での PC インタフェースボードの設定	2-6
2.4	HDI のインストール	2-7
2.5	トラブルシューティング	2-8
2.5.1	接続不良	2-8
2.5.2	通信不良	2-9
3	ハードウェア	3-1
3.1	ユーザシステムへの接続	3-1
3.1.1	ユーザシステムインタフェースケーブル先端部と ユーザシステムの接続例	3-2
3.1.2	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	3-3

3.1.3	ユーザシステムインタフェースケーブル本体部と先端部の接続	3-3
3.2	電源供給	3-4
3.2.1	AC 電源アダプタ	3-4
3.2.2	極性	3-4
3.2.3	電源モニタ回路	3-4
3.3	ハードウェアインタフェース	3-5
3.3.1	信号保護	3-5
3.3.2	ユーザシステムインタフェース回路	3-5
3.3.3	クロック発振器	3-7
3.3.4	外部プローブ/トリガ出力	3-7
3.3.5	電源フォロワ回路	3-8
3.4	MCU と E6000 エミュレータの相違点	3-10
3.4.1	A/D コンバータ	3-10
3.4.2	未使用領域のアクセス	3-10
3.4.3	Go Reset コマンドによるプログラム実行	3-10
4	チュートリアル	4-1
4.1	はじめに	4-1
4.2	HDI の起動	4-1
4.2.1	ターゲットプラットフォームの選択	4-2
4.3	E6000 エミュレータのセットアップ	4-4
4.3.1	プラットフォームの構成	4-4
4.3.2	メモリマッピング	4-5
4.4	チュートリアルプログラムのダウンロード	4-8
4.4.1	オブジェクトファイルのダウンロード	4-8
4.4.2	プログラムリストの表示	4-9
4.5	ブレークポイントの使い方	4-11
4.5.1	PC Break の設定	4-11
4.5.2	プログラムの実行	4-12
4.5.3	レジスタ内容の参照	4-14
4.5.4	ブレークポイントの確認	4-15
4.6	メモリと変数の表示	4-16
4.6.1	メモリを表示する	4-16
4.6.2	変数を表示する	4-17
4.7	プログラムのステップ実行	4-19

4.7.1	シングルステップ	4-19
4.7.2	関数全体のステップ実行	4-21
4.7.3	ローカル変数の表示	4-22
4.8	イベント検出システム(Complex Event System)の使用方法	4-24
4.8.1	イベント検出システムによるハードウェアブレークポイントの設定	4-24
4.9	トレースバッファの使い方	4-28
4.9.1	トレースバッファの表示	4-28
4.9.2	トレースフィルタの設定	4-29
4.10	スタックトレース機能	4-31
4.11	セッションの保存	4-32
4.12	さてつぎは？	4-32
5	H8/3887 シリーズ、H8/3867 シリーズ、H8/3847 シリーズ、H8/3827 シリーズ、 H8/3847R シリーズ、H8/3827R シリーズ用 E6000 エミュレータ HDI の機能	5-1
5.1	コンフィグレーションダイアログボックス	5-3
5.2	ブレークポイント	5-5
5.2.1	プログラムブレークポイントを設定する	5-6
5.3	イベント検出システム	5-7
5.3.1	[General]	5-8
5.3.2	[Bus / Area]	5-9
5.3.3	[Signals]	5-10
5.3.4	[Action]	5-11
5.3.5	イベントシーケンス	5-13
5.3.6	イベントの前提条件	5-14
5.3.7	イベントをリセットする	5-15
5.4	メモリマッピングダイアログボックス	5-16
5.5	トレースウインドウ	5-18
5.5.1	[Filter]	5-19
5.5.2	[Find]	5-19
5.5.3	[Cycle]	5-19
5.5.4	[Pattern]	5-20
5.5.5	[General]	5-20
5.5.6	[Bus / Area]	5-21
5.5.7	[Signals]	5-22
5.6	トレース制御	5-23

5.6.1	[General]	5-24
5.6.2	[Stop]	5-25
5.6.3	[Delayed stop]	5-26
5.7	コマンドライン	5-27
6	コマンドライン機能	6-1
6.1	BREAKPOINT / EVENT	6-3
6.1.1	プログラムブレークポイント	6-3
6.1.2	アクセスブレークポイント	6-3
6.1.3	範囲ブレークポイント	6-3
6.1.4	オプション	6-4
6.2	BREAKPOINT_CLEAR / EVENT_CLEAR	6-7
6.3	BREAKPOINT_DISPLAY / EVENT_DISPLAY	6-8
6.4	BREAKPOINT_ENABLE / EVENT_ENABLE	6-9
6.5	BREAKPOINT_SEQUENCE / EVENT_SEQUENCE	6-10
6.6	CLOCK	6-11
6.7	DEVICE_TYPE	6-12
6.8	MAP_SET	6-13
6.9	MODE	6-14
6.10	TEST_EMULATOR	6-15
6.11	TIMER	6-16
6.12	TRACE_ACQUISITION	6-17
6.13	TRACE_COMPARE	6-18
6.14	TRACE_SAVE	6-19
6.15	TRACE_SEARCH	6-20
6.16	USER_SIGNALS	6-21
6.17	REFRESH	6-22
7	故障解析	7-1
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	7-1
7.2	テストプログラムによる故障解析	7-2
7.3	エラー発生時の処理	7-6

目 次

図 2-1	Computer Properties ダイアログボックス(設定前)	2-3
図 2-2	Edit Resource Setting ダイアログボックス	2-4
図 2-3	Computer Properties ダイアログボックス(設定後)	2-5
図 2-4	エラーメッセージ (1)	2-8
図 2-5	エラーメッセージ (2)	2-9
図 3-1	E6000 コネクタの位置	3-1
図 3-2	ユーザシステムインタフェースケーブルの接続	3-2
図 3-3	ネジの締め付け順序	3-2
図 3-4	ユーザシステムインタフェースケーブル外観図	3-3
図 3-5	電源プラグ	3-4
図 3-6	以下に記述のない信号	3-5
図 3-7	OSC1 および X1 信号	3-5
図 3-8	PB0/AN0 ~ PB7/AN7, PC0 ~ PC3, P50/WKP0/SEG1 ~ P57/WKP7/SEG8, P60/SEG9 ~ P67/SEG16, P70/SEG17 ~ P77/SEG24, P80/SEG25 ~ P87/SEG32, P90/SEG33 ~ P97/SEG40 /CL1, PA0/COM1 ~ PA3/COM4 信号	3-6
図 3-9	AVcc, AVss 電源端子	3-6
図 3-10	CVcc, TEST 信号	3-6
図 3-11	V0, V1, V2, V3 電源端子	3-6
図 3-12	P30/PWM ~ P37/AEVL, P40/RCK32 ~ P43/IRQ0 信号	3-7
図 3-13	クロック発振回路	3-7
図 3-14	外部プローブコネクタ	3-7
図 3-15	外部プローブインタフェース回路	3-8
図 4-1	HDI 起動メニュー	4-1
図 4-2	プラットフォームの選択	4-2
図 4-3	HDI ウィンドウ	4-2
図 4-4	Configuration ダイアログボックス	4-4
図 4-5	Memory Mapping ダイアログボックス	4-5
図 4-6	Edit Memory Mapping ダイアログボックス	4-6
図 4-7	System Status ウィンドウ (Memory シート)	4-7
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)	4-8
図 4-9	HDI ダイアログボックス	4-8
図 4-10	Open ダイアログボックス (ソースファイルの選択)	4-9

図 4-11	ソースプログラム画面	4-10
図 4-12	ブレークポイントの設定	4-11
図 4-13	ステートメントの強調表示	4-12
図 4-14	System Status ウィンドウ (Platform シート)	4-13
図 4-15	Registers ウィンドウ	4-14
図 4-16	Register ダイアログボックス	4-14
図 4-17	Breakpoints ウィンドウ	4-15
図 4-18	Open Memory Window の設定	4-16
図 4-19	Memory ウィンドウ(Byte)	4-16
図 4-20	Watch ウィンドウ (変数追加後)	4-17
図 4-21	Watch ウィンドウ (シンボル拡張)	4-18
図 4-22	Add Watch ダイアログボックス	4-18
図 4-23	Watch ウィンドウ (変数の追加)	4-18
図 4-24	Reset Go 実行後の Program ウィンドウ	4-19
図 4-25	Step In 実行後の Program ウィンドウ (1)	4-20
図 4-26	Step Out 実行後の Program ウィンドウ	4-20
図 4-27	Step In 実行後の Program ウィンドウ (2)	4-21
図 4-28	Step Over 実行後の Program ウィンドウ	4-22
図 4-29	Locals ウィンドウ	4-22
図 4-30	Locals ウィンドウ (変数"min"内容変更後)	4-23
図 4-31	Locals ウィンドウ (配列変数"a"ソート後)	4-23
図 4-32	ブレークポイントの追加 (アドレス指定)	4-25
図 4-33	ブレークポイントの追加 (回数指定)	4-26
図 4-34	Breakpoints ウィンドウ (追加後)	4-26
図 4-35	Event ブレークポイントによるプログラムの停止	4-27
図 4-36	Trace ウィンドウ	4-28
図 4-37	Trace Filter ダイアログボックス	4-29
図 4-38	Bus / Area の設定	4-30
図 4-39	Trace ウィンドウ (トレースフィルタ指定)	4-30
図 4-40	Stack Trace ウィンドウ	4-31
図 5-1	Configuration ダイアログボックス	5-3
図 5-2	Breakpoints ウィンドウ	5-5
図 5-3	ブレークポイントの設定	5-6
図 5-4	イベントの設定 (General)	5-8
図 5-5	イベントの設定 (Bus / Area)	5-9

図 5-6	イベントの設定 (Signals)	5-10
図 5-7	イベントの設定 (Action)	5-11
図 5-8	イベントシーケンス画面 (1)	5-13
図 5-9	イベントシーケンス画面 (2)	5-14
図 5-10	イベントシーケンス画面 (3)	5-15
図 5-11	Memory Mapping ダイアログボックス	5-16
図 5-12	メモリマッピングの変更	5-16
図 5-13	Trace ウィンドウ	5-18
図 5-14	Trace Filter ダイアログボックス (General) (1)	5-19
図 5-15	Trace Filter ダイアログボックス (General) (2)	5-20
図 5-16	Trace Filter ダイアログボックス (Bus / Area)	5-21
図 5-17	Trace Filter ダイアログボックス (Signals)	5-22
図 5-18	Trace Acquisition ダイアログボックス (General)	5-23
図 5-19	Trace Acquisition ダイアログボックス (Stop)	5-25
図 5-20	Trace Acquisition ダイアログボックス (Delayed Stop)	5-26
図 5-21	Command Line ウィンドウ	5-27

表 目 次

表 1-1	メモリタイプの定義	1-5
表 1-2	動作電圧および動作周波数	1-6
表 1-3	エミュレーションクロック一覧	1-7
表 1-4	E6000 使用環境条件	1-8
表 1-5	外形寸法および質量	1-8
表 2-1	PC インタフェースボードのメモリ領域	2-3
表 3-1	MCU と E6000 エミュレータの相違	3-10
表 4-1	コンフィグレーションオプションの設定例	4-5
表 4-2	メモリタイプの定義	4-6
表 4-3	メモリタイプオプション	4-6
表 4-4	プログラムステップオプション	4-19
表 5-1	HDI のメニューとマニュアルの対応表	5-1
表 5-2	Configuration ダイアログボックス	5-4
表 5-3	イベントチャンネルとオプションの設定	5-7
表 5-4	イベントアクション	5-11
表 5-5	メモリタイプの定義	5-17
表 5-6	アクセスタイプの定義	5-17
表 6-1	HDI コマンドライン機能とマニュアルの対応表	6-1
表 6-2	MCU バスステータス	6-4
表 6-3	ブレークポイント削除の指定	6-7
表 6-4	ブレークポイント有効 / 無効	6-9
表 6-5	クロックパラメータ	6-11
表 6-6	MCU モードのパラメータ	6-14
表 6-7	タイマコマンド	6-16
表 6-8	ユーザ信号コマンド	6-21

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 H8/3887シリーズ、H8/3867シリーズ、H8/3847シリーズ、H8/3827シリーズ、H8/3847Rシリーズ、H8/3827R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っ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っ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
- ・ マスク条件指定付きデータ
- ・ リードまたはライト
- ・ MCUアクセスタイプ (命令プリフェッチ、データフェッチなど)
- ・ 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できます。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 範囲チャンネル

範囲チャンネルは、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
- ・ マスク条件指定付きデータ
- ・ リードまたはライト
- ・ MCUアクセスタイプ（命令プリフェッチ、データフェッチなど）
- ・ 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ために使われます。

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番目のイベントの発生間のプログラムの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用メモリを標準装備しています。

エミュレーションメモリは、MCUアドレス空間に1バイト単位で割り付けできます。各メモリブロックは、[**Configure Map...**] コマンドを使って、ユーザシステム上のメモリに指定でき、それぞれの場合で、リードライトアクセス、リードオンリアクセス、またはアクセス禁止を指定できます。

エミュレーションメモリの各々のメモリタイプの定義を表1-1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	エミュレーションボード上のメモリ

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、他の関連するウィンドウにすぐに反映されます。

1.3.2 動作電圧および動作周波数

以下に本E6000エミュレータがサポートするMCUの動作電圧および動作周波数仕様について示します。MCUによっては低電圧動作、高周波数動作を保証しないものがありますので注意してください。

表 1-2 動作電圧および動作周波数

No.	MCU型名	動作電圧(V)	動作周波数範囲fosc(MHz)
1	H8/3847Rシリーズ	1.8-5.5	2.0-4.0
	H8/3827Rシリーズ	2.7-5.5	2.0-10.0
		4.5-5.5	2.0-16.0
2	H8/3887シリーズ	1.8-5.5	0.4-1.0
	H8/3867シリーズ	2.2-5.5	0.4-2.0
	H8/3847シリーズ	2.6-5.5	0.4-3.2
	H8/3827シリーズ	3.0-5.5	0.4-4.0
		4.5-5.5	0.4-6.0

1.3.3 エミュレーションクロック

システムクロックとサブクロックは表1-3に示すような周波数に設定できます。

表 1-3 エミュレーションクロック一覧

エミュレータ	エミュレーションクロック	ターゲットMCU	周波数設定値
HS388REPI60 H	システムクロック	H8/3887シリーズ、 H8/3867シリーズ、 H8/3847シリーズ、 H8/3827シリーズ、	2MHz, 0.5MHz およびターゲットクロック ÷ 2
		H8/3847Rシリーズ、 H8/3827Rシリーズ	8MHz, 2MHz, 0.5MHz およびターゲットクロック ÷ 2
	サブクロック	H8/3887シリーズ、 H8/3867シリーズ、 H8/3847シリーズ、 H8/3827シリーズ、 H8/3847Rシリーズ、 H8/3827Rシリーズ	32.768kHz, 38.4kHz, 307.2kHz, およびターゲットサブクロック

1.3.4 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うために、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。

1.3.5 使用環境条件

表 1-4 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	AC入力電源	電圧：100～240V AC 周波数：50/60Hz 電流：Max. 0.6A
5	ユーザVcc (Uvcc)	電圧：1.8～5.5V

1.3.6 外形寸法と質量

表 1-5 外形寸法および質量

項番	項目	仕様
1	外形寸法	219 x 160 x 54 (mm)
2	質量	約970 (g)

2 セットアップ

本章は、PCインタフェースボードを使用したE6000エミュレータのセットアップ方法、およびHDIと共に使うためのE6000エミュレータの準備方法を述べます。

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）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

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
- ・ AC電源アダプタ 5V 6A（ACケーブル付き）
- ・ HDIインストールディスク（HS388REPI60SR）
- ・ 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。

ホストコンピュータにはIBM PCまたは、IBM PC互換機が使用できます。Windows® 98またはWindows NT®が必要です。

2.2 PCインタフェースボードのセットアップ

E6000エミュレータは、PCインタフェースボード（HS6000E1101H）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ドボードであり、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。

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。

2.2.1 Windows[®] 98でのPCインタフェースボードのセットアップ

- Windows[®] 98を起動します。
- [**My Computer**] アイコンをマウス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。

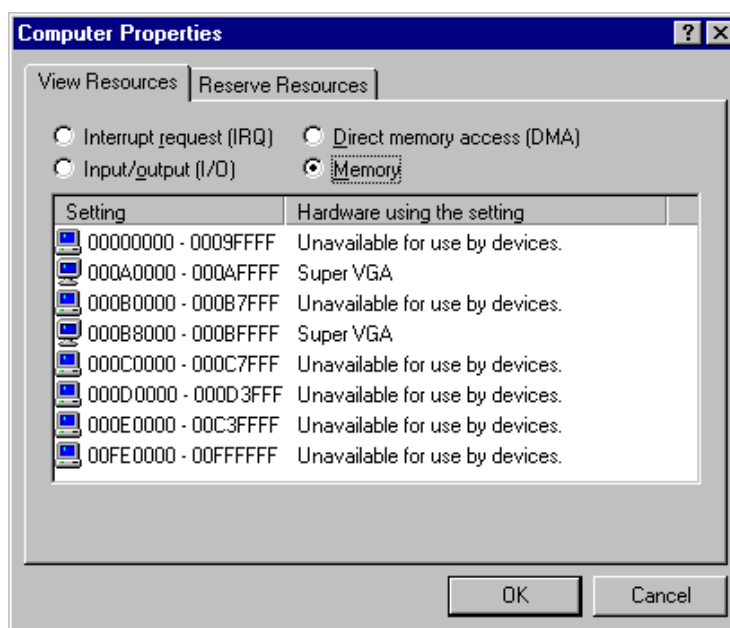


図 2-1 Computer Propertiesダイアログボックス(設定前)

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

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'CEFFF	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® 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。

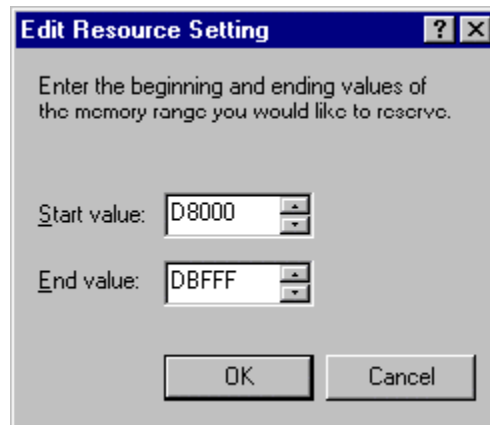


図 2-2 Edit Resource Settingダイアログボックス

- ・ 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。

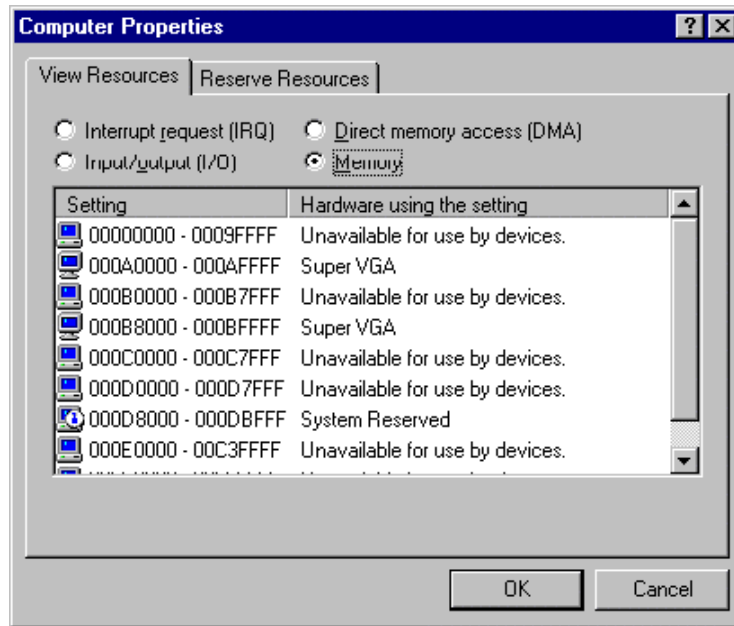


図 2-3 Computer Propertiesダイアログボックス(設定後)

2.3 Windows NT[®] 4.0でのPCインタフェースボードの設定

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	CEFFF	7	DC000	DFFFF	B	EC000	EEFFF

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。

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われた場合、PCインタフェースボードはE6000エミュレータを認識できていません。



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

以下のメッセージが表示されると、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領域と、PCインタフェースボード上のリヤパネルのスイッチ設定が異なります。
- ・ 選択されたメモリ領域が別のアプリケーションで使われています。
- ・ 「2.2 PCインタフェースボードのセットアップ」「2.3 PCインタフェースボードの設定」に従って設定を見直してください。

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

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先端部をユーザシステムへ接続する。
- ・ 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する。
- ・ ケーブル本体部を先端部へ接続する。

これら手順の詳細については、ユーザシステムインタフェースケーブル添付の取扱い説明書を参照してください。

以下に、E6000エミュレータのコネクタを示します。

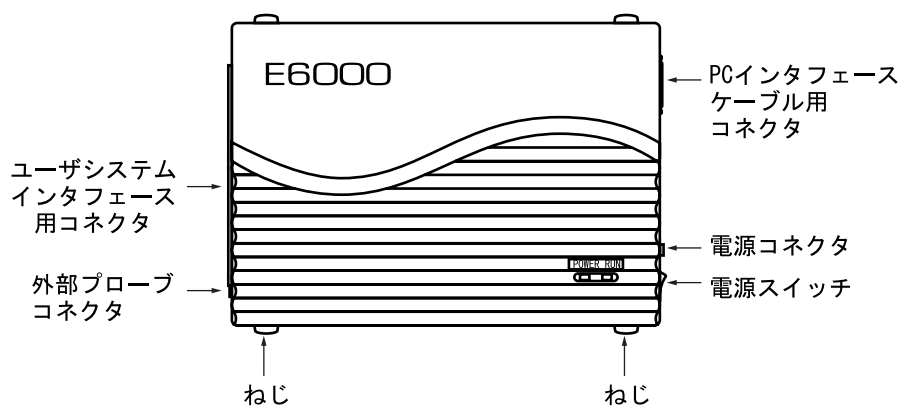


図 3-1 E6000 コネクタの位置

3.1.1 ユーザシステムインタフェースケーブル先端部と ユーザシステムの接続例

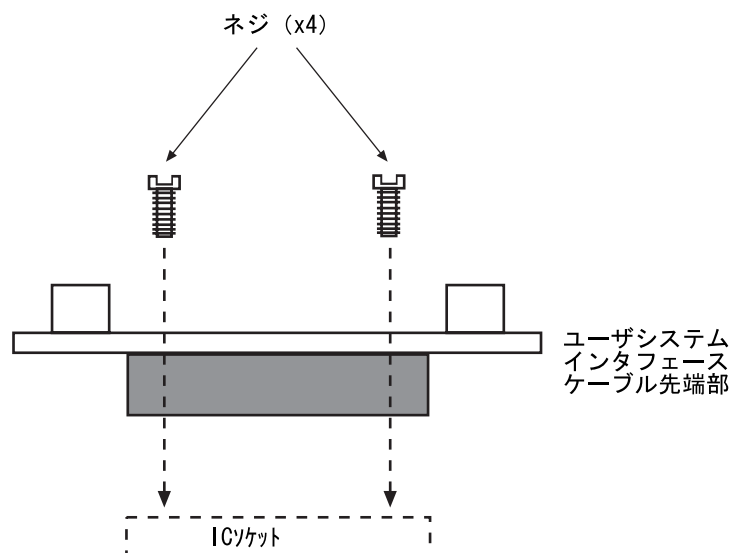


図 3-2 ユーザシステムインタフェースケーブルの接続

- ・ 接続前に、必ず、E6000エミュレータ、ユーザシステムの電源を切ってください。
- ・ 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上のソケットに挿入してください。

注：QFPパッケージによっては、ユーザシステムインタフェースケーブル先端部の向きにかかわらず、ソケットに差し込むことができるものがあります。挿入の際には、E6000エミュレータ側とソケットの1ピンの位置を必ず一致させてください。

- ・ ユーザシステムインタフェースケーブルに付属のネジを使用して、ユーザシステムインタフェースケーブル先端部とソケットをネジ留めしてください。以下に示す順番で、対角に少しずつ締めつけ、1つのネジを集中して締めつけることは絶対に避けてください。

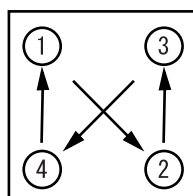


図 3-3 ネジの締め付け順序

注：ネジを締め付け過ぎないように注意してください。ユーザシステムの接続不良やユーザシステムインタフェースケーブル先端部が壊れる原因となります。QFPソケットに半田付け用固定金具が付いている場合は、これを使用して、E6000エミュレータとユーザシステムの接続を強めることができます。

3.1.2 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。

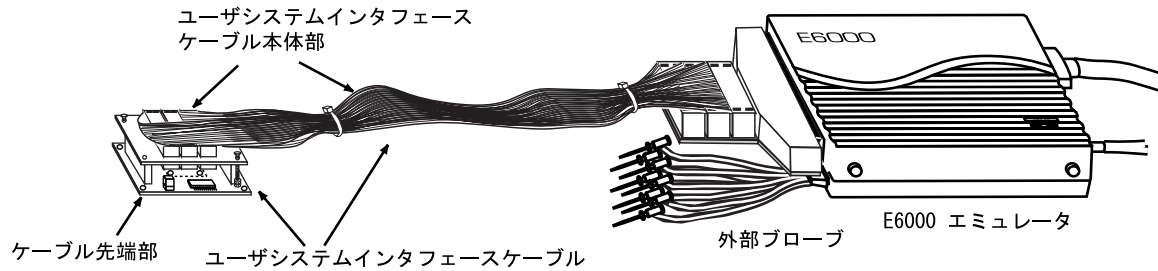


図 3-4 ユーザシステムインタフェースケーブル外観図

3.1.3 ユーザシステムインタフェースケーブル本体部と先端部の接続

ユーザシステムインタフェースケーブル本体部をユーザシステムに接続されている先端部に接続してください。

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。

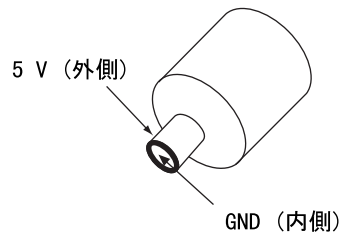


図 3-5 電源プラグ

3.2.3 電源モニタ回路

E6000エミュレータには、ユーザシステムの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が赤く点灯します。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。電源電圧が4.75V未満の場合、E6000エミュレータに必要な電源が供給されません。

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、バッファなしに直接エミュレータ上のエバチップに接続されています。

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/過小電圧から保護されています。ただし、AVccとアナログポートには、この保護回路はありません。

アナログポート以外のポートには、プルアップ抵抗が接続されています。

なお、E6000はユーザシステムインタフェースケーブル先端部の信号を監視することにより、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 ユーザシステム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、プルアップ抵抗が入っており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。また、ユーザシステムインタフェースケーブルによる信号の遅れは約8nsです。

以下にユーザインタフェース信号回路を示します。

(1) 以下に記述のない信号

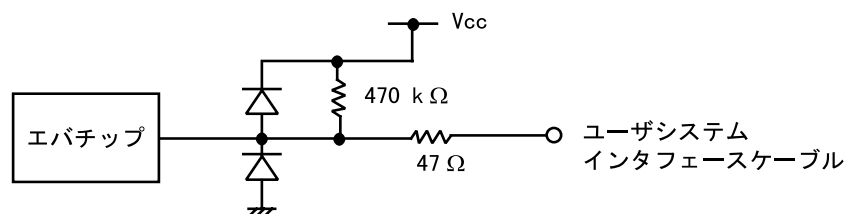


図 3-6 以下に記述のない信号

(2) OSC1, X1

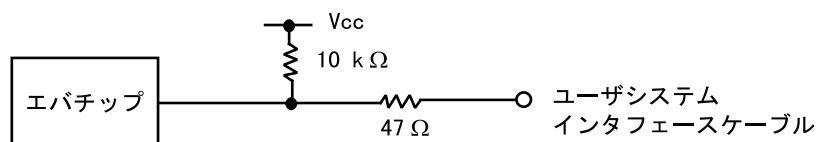


図 3-7 OSC1およびX1信号

- (3) PB0/AN0 ~ PB7/AN7, PC0 ~ PC3,
 P50/WKPO/SEG1 ~ P57/WKP7/SEG8,
 P60/SEG9 ~ P67/SEG16,
 P70/SEG17 ~ P77/SEG24,
 P80/SEG25 ~ P87/SEG32,
 P90/SEG33 ~ P97/SEG40/CL1,
 PA0/COM1 ~ PA3/COM4

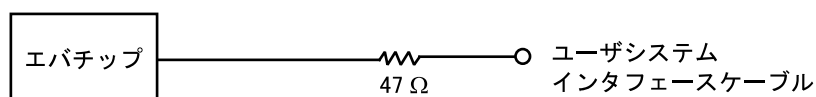


図 3-8 PB0/AN0 ~ PB7/AN7, PC0 ~ PC3, P50/WKPO/SEG1 ~ P57/WKP7/SEG8, P60/SEG9 ~ P67/SEG16, P70/SEG17 ~ P77/SEG24, P80/SEG25 ~ P87/SEG32, P90/SEG33 ~ P97/SEG40/CL1, PA0/COM1 ~ PA3/COM4信号

- (4) AVcc, AVss

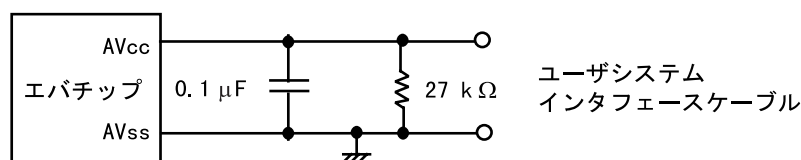


図 3-9 AVcc, AVss電源端子

- (5) CVcc, TEST

CVccがGNDに接続されている場合、またはTESTがVccレベルに接続されている場合は、HDI起動時にメッセージが出力されます。

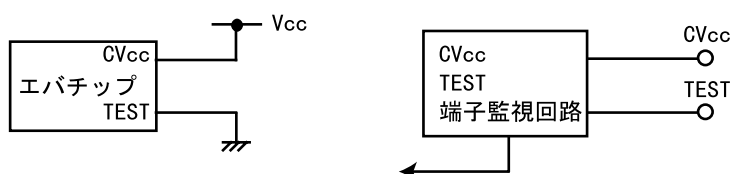


図 3-10 CVcc, TEST信号

- (6) V0, V1, V2, V3

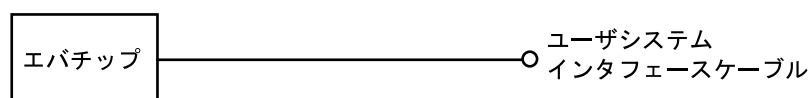


図 3-11 V0, V1, V2, V3電源端子

- (7) P30/PWM ~ P37/AEVL,
P40/RCK32 ~ P43/IRQ0

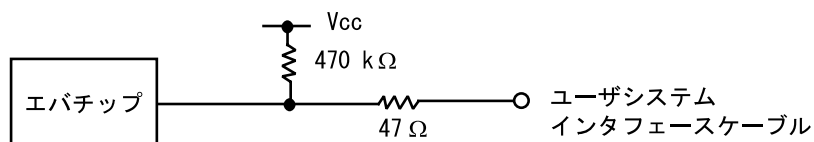


図 3-12 P30/PWM ~ P37/AEVL, P40/RCK32 ~ P43/IRQ0 信号

3.3.3 クロック発振器

以下の図は、ユーザシステムインタフェースケーブル先端部に構成された発振回路を示します。この回路は、水晶発振子を1~16MHzの範囲で発振するように設計されています。詳細については各ユーザシステムインタフェースケーブルマニュアルを参照してください。

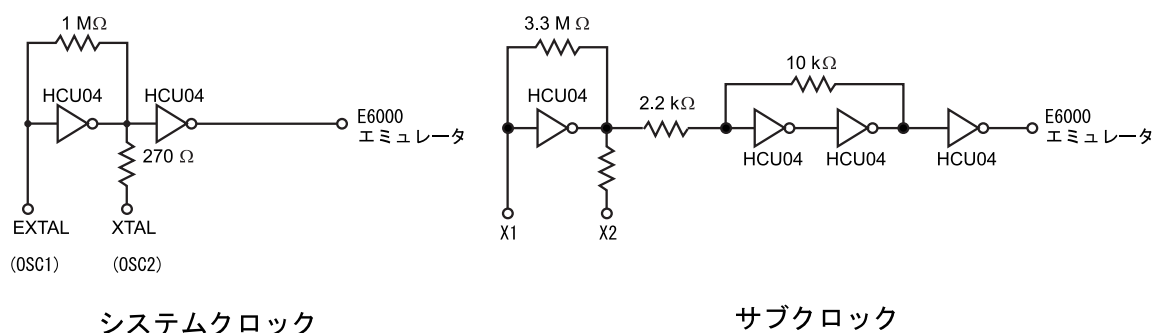


図 3-13 クロック発振回路

3.3.4 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)にE6000エミュレータ付属の外部プローブを接続してください。外部プローブは入力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。

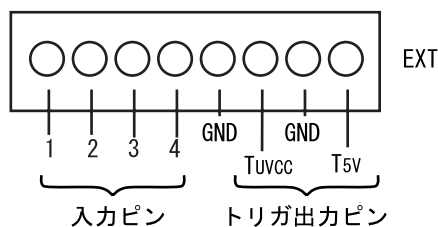


図 3-14 外部プローブコネクタ

以下に外部プローブのインタフェース回路を示します。

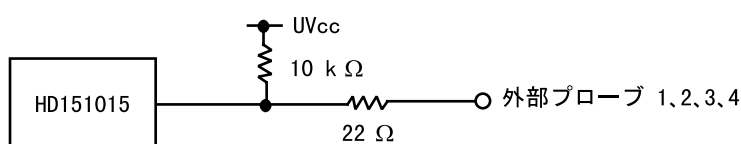


図 3-15 外部プローブインタフェース回路

トリガ出力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(プローブ色：白色、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(プローブ色：黄色、ユーザシステム電源電圧)レベルの2つあります。なお、トリガ出力TUvccについては、ユーザシステム電源電圧1.8Vにての評価は行なえません(2.0V~5.0Vの範囲としてください)。

3.3.5 電源フォロワ回路

注意

1. ユーザシステムをE6000エミュレータに接続しない時は、ユーザケーブルをエミュレータに接続しないでください。
2. ユーザシステムをE6000エミュレータに接続した時は、ユーザシステムの電源を入れてからエミュレータを起動してください。

E6000エミュレータに搭載されてる電圧フォロワ回路は、ユーザシステムの電圧レベルをモニタしています。E6000エミュレータの電源はユーザシステムの電源レベルを生成しE6000エミュレータ内に供給しているため、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。

E6000エミュレータにユーザシステムインタフェースケーブルが接続されていないと、E6000エミュレータ上のMCUは一定電圧レベルで動作し、ユーザシステムインタフェースケーブルが接続されている場合は、ユーザシステムの電源電圧と同レベルの電圧で動作します。ユーザシステムVccがMCUの動作電圧よりも低い場合であっても、E6000エミュレータは供給電圧をUVccに一致させます。したがって5V以下で動作している場合は、エミュレーションクロックの周波数が各Vccにおける最高動作周波数を超えないように注意してください。

Configurationダイアログボックスを使用して、[User VCC Threshold] をVcc max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ウインドウのPlatformシートにある [User System Voltage] には [Down] が表示されます。User VCC Threshold電源レベルよりも高い場合は、[OK] が表示されます。また、ユーザシステムインタフェースケーブル未接続時は、E6000のVccレベルはVcc maxになります。

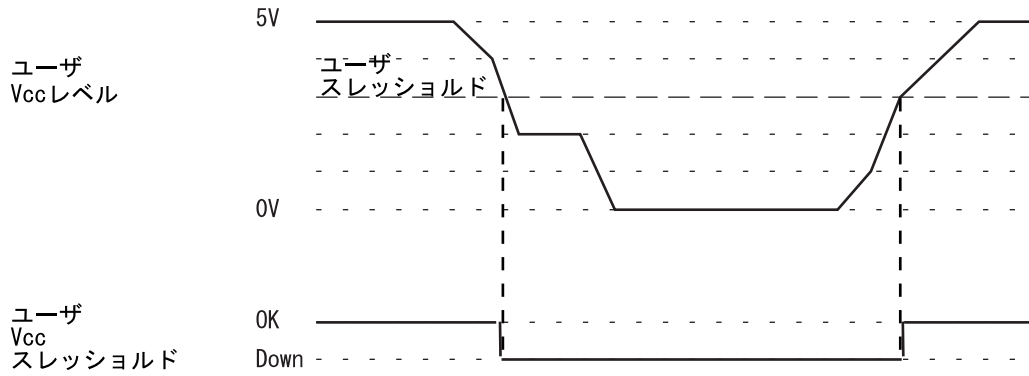


図 3-16 ユーザシステムとE6000とのVccの関係

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の初期値の相違を以下に示します。

表 3-1 MCUとE6000エミュレータの相違

状態	レジスタ	E6000エミュレータ	MCU
電源投入後	PC	不定	リセットベクタ値
	R0 to R6	0000	不定
	R7 (SP)	0010	不定
	CCR	Iマスクは1 その他は不定	Iマスクは1 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)	0010	不定
	CCR	Iマスクは1 その他は不定	Iマスクは1 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の詳細については、「3.3 ハードウェアインタフェース」を参照してください。

3.4.1 A/Dコンバータ

ユーザシステムインタフェースケーブルで接続されているため、A/D変換の精度は、MCUのハードウェアマニュアルに記載の精度より劣下します。

3.4.2 未使用領域のアクセス

未使用領域H'FF80~H'FF8Fはエミュレータシステムで使用しています。MAP設定でエミュレータに割り付けた場合、正常動作できませんので、本領域は使用しないでください。

また、MCUが使用していないレジスタについてはエミュレータでアクセスしないでください。

3.4.3 Go Resetコマンドによるプログラム実行

Go Resetコマンドを用いてプログラムを実行する際、E6000は約500μsのリセット信号をエパチップに入力します。実行時間測定結果には、このリセット信号入力時間が加算されます。

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 H8/3887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_3880R] から [Hitachi Debugging Interface] を選択してください。

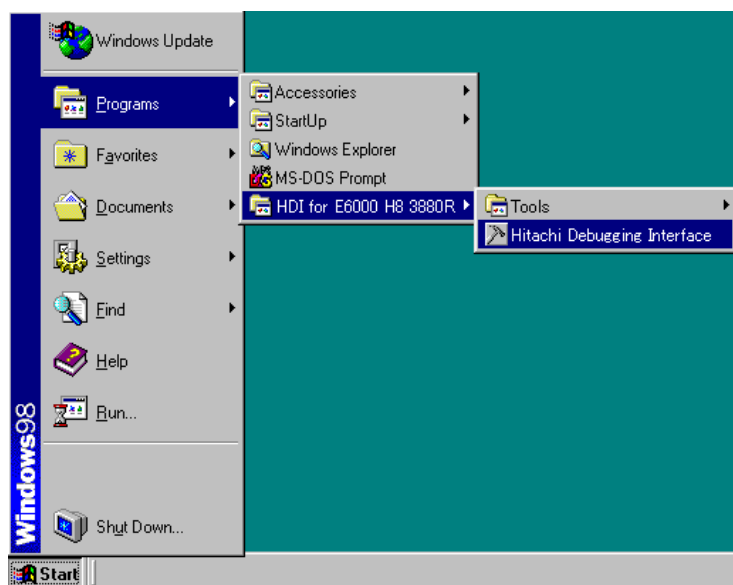


図 4-1 HDI 起動メニュー

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。

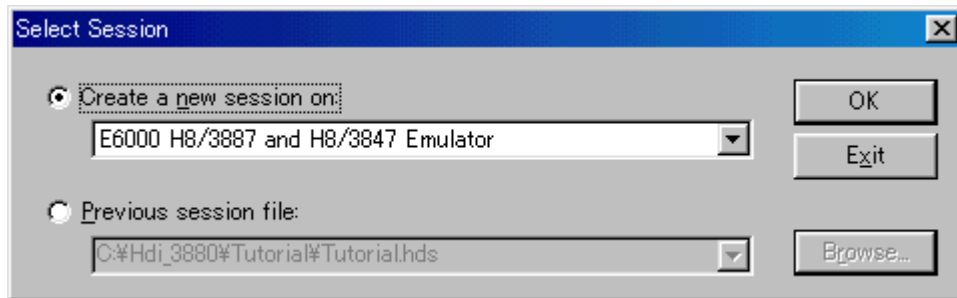


図 4-2 プラットフォームの選択

- ・ このチュートリアルでは、E6000 H8/3887 and H8/3847 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。

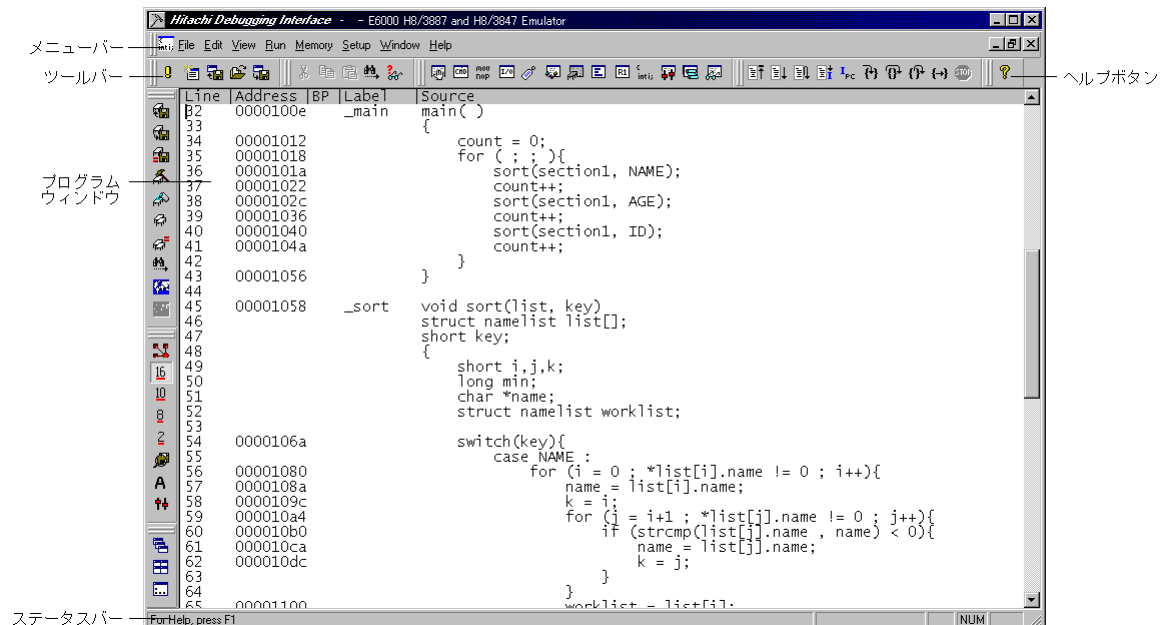


図 4-3 HDIウィンドウ

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
- ・ 動作モード
- ・ 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。

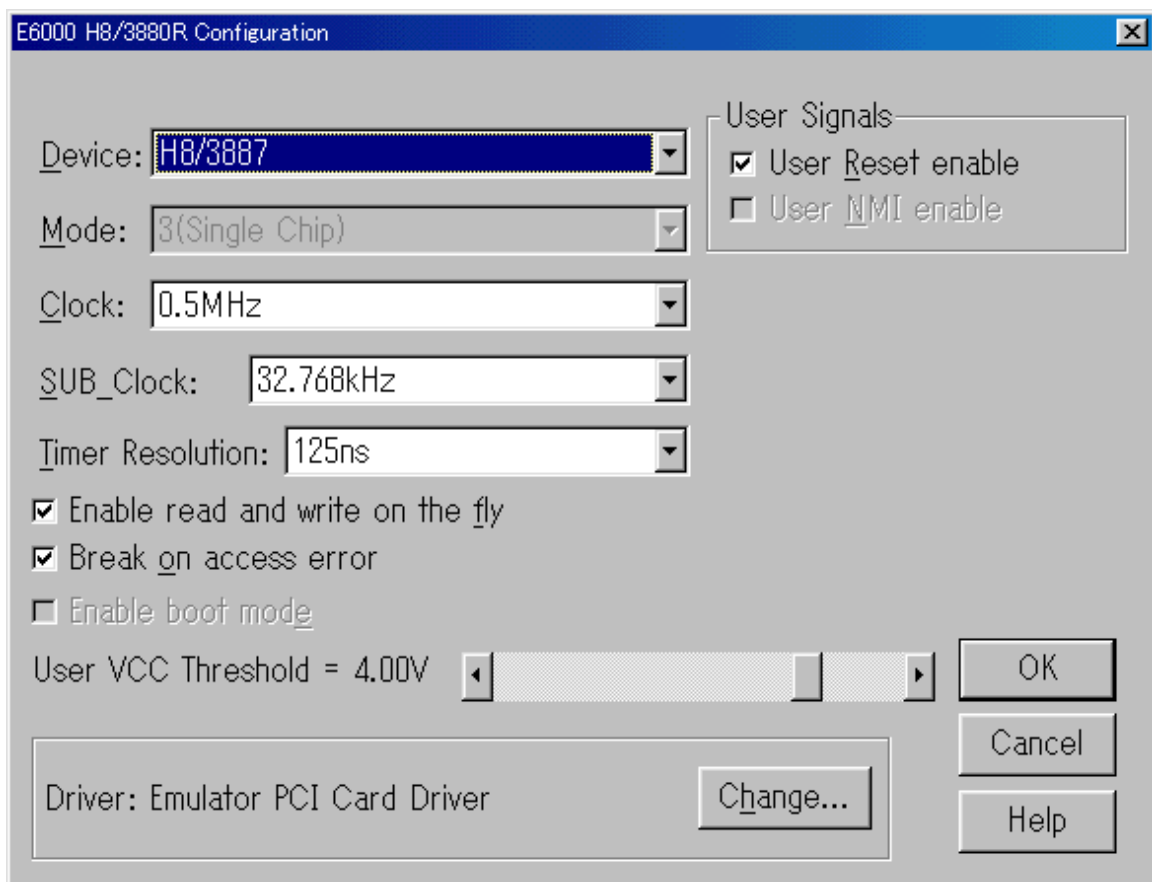


図 4-4 Configurationダイアログボックス

- ・ 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)	H8/3887
モード (Mode)	3 (シングルチップモード) 変更不可
動作クロック (Clock)	0.5MHz
タイマ分解能 (Timer Resolution)	125ns
ユーザシステムの電圧レベル (User Vcc Threshold)	4.00V
その他のオプション	イネーブル

- ・ [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およびモードを選択すると、HDIは自動的に選択したデバイスおよびモード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。

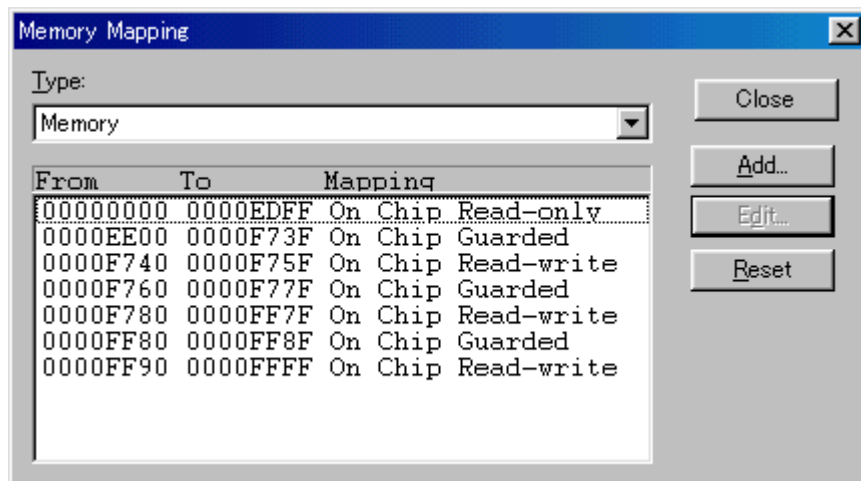


図 4-5 Memory Mappingダイアログボックス

E6000エミュレータメモリは、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)	エミュレーション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
Read-Write	RAM
Read-Only	ROM
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。以下のように割り付け状態を見ることがもできます。

- マッピング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、または対象のマッピング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。

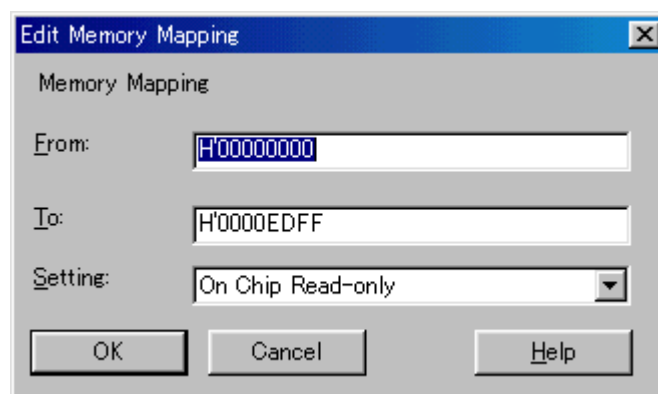



図 4-6 Edit Memory Mappingダイアログボックス

- [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。

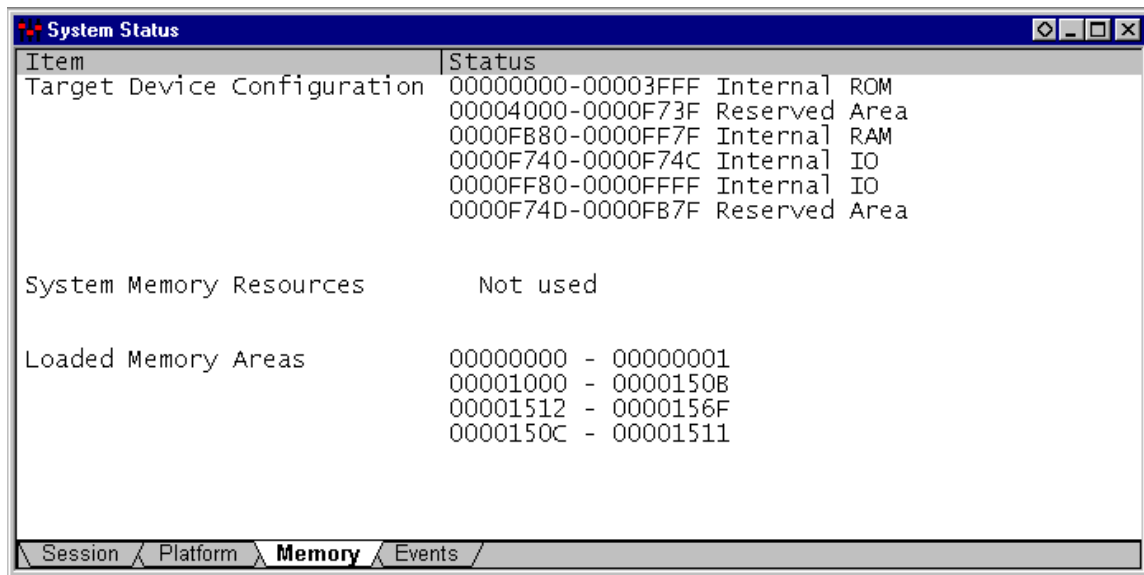


図 4-7 System Statusウィンドウ (Memoryシート)


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
の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。

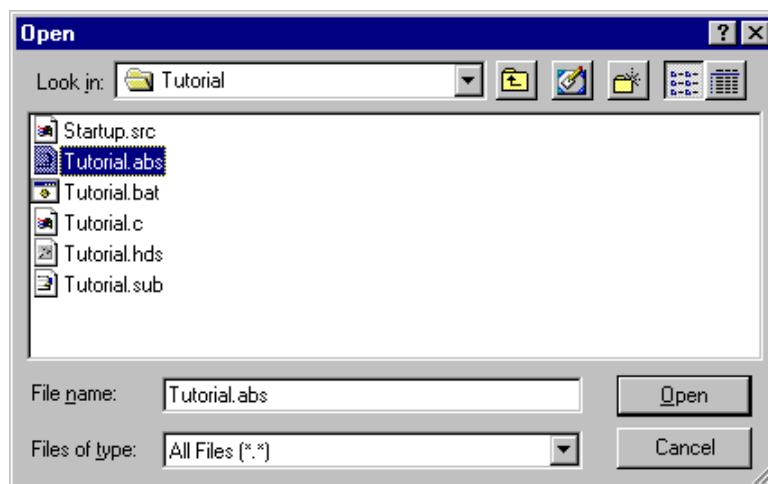


図 4-8 Openダイアログボックス (オブジェクトファイルの選択)

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。

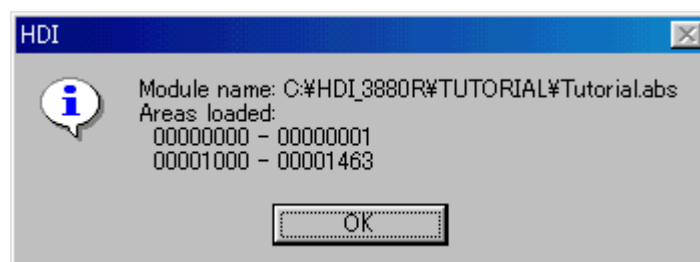



図 4-9 HDIダイアログボックス

- ・ [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。

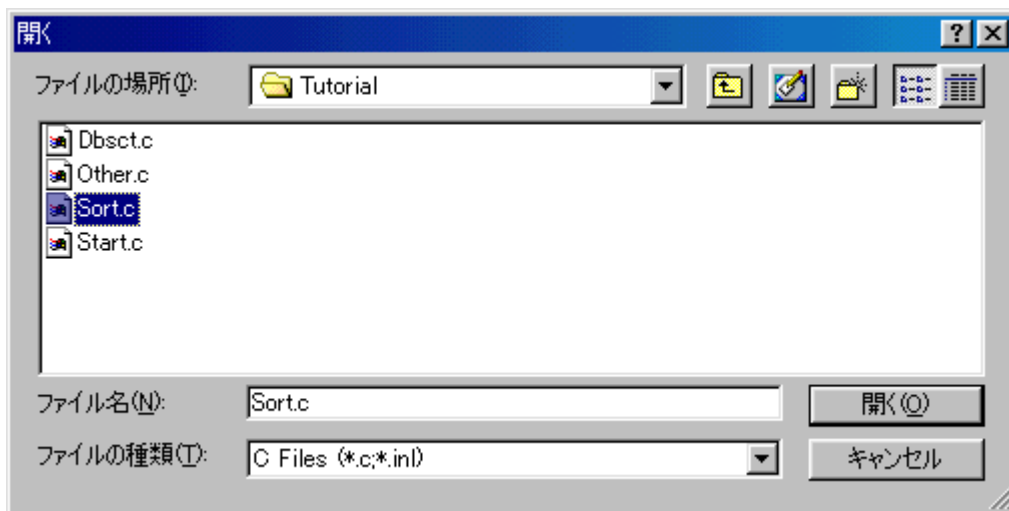
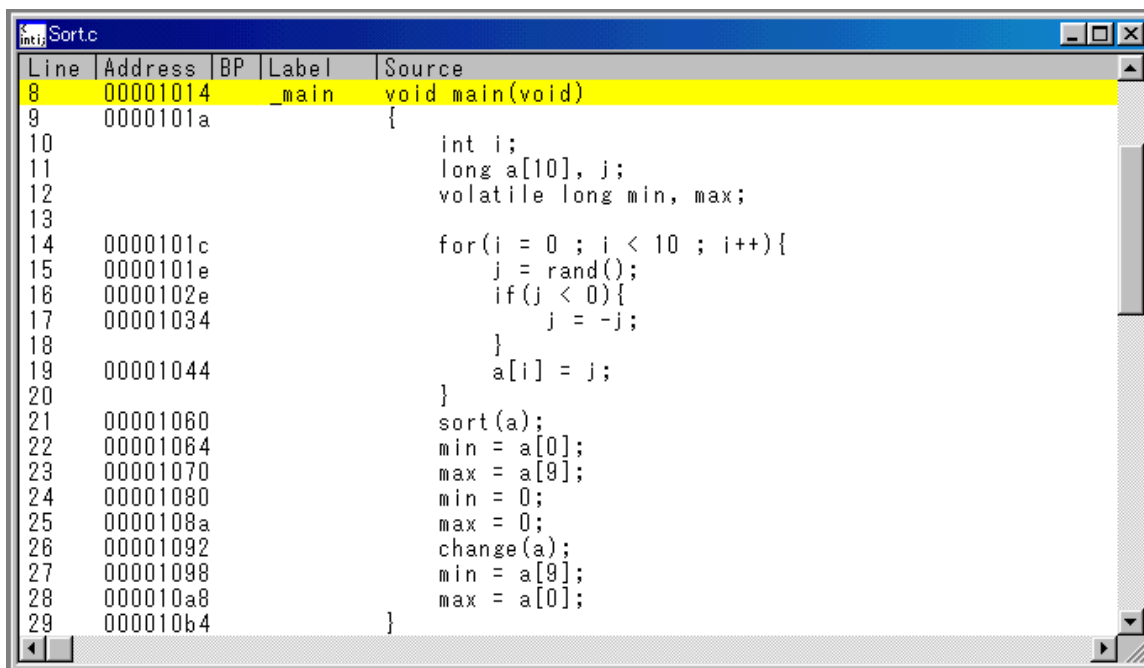


図 4-10 Openダイアログボックス (ソースファイルの選択)

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



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			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 4-11 ソースプログラム画面

- ・ 必要ならば、[Setup]メニューの[Customise]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。

プログラムウィンドウを、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ブレークポイント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にブレークポイントを設定します。

- 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。

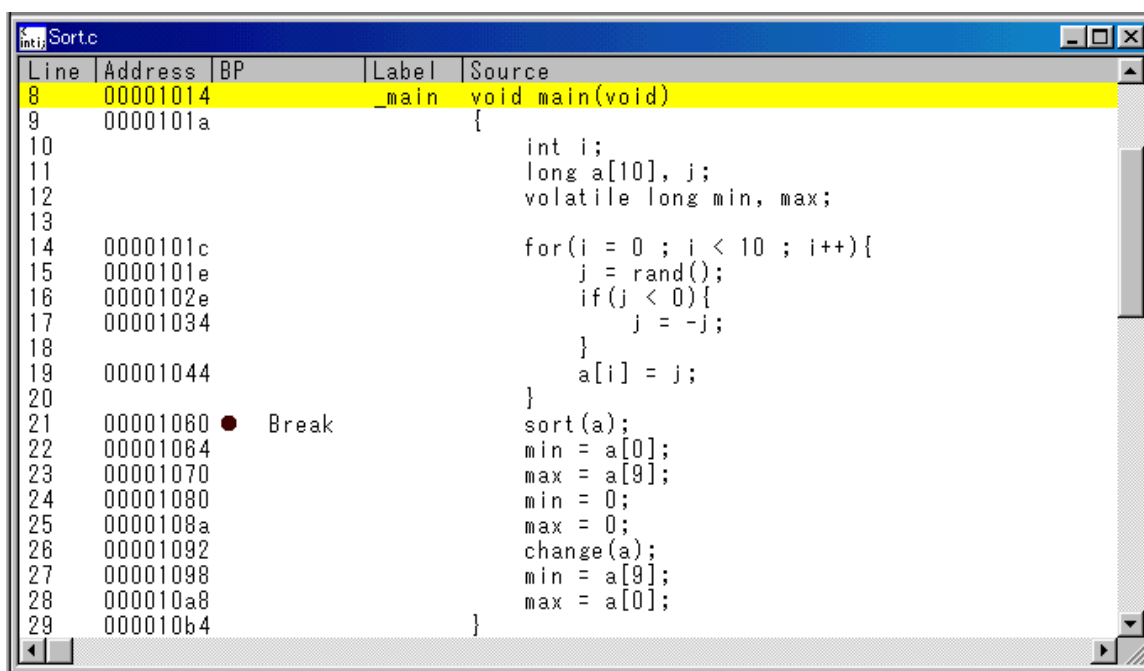


図 4-12 ブレークポイントの設定


その位置に“ Break ”が表示され、そのアドレスにプログラムブレークポイント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のイベント設定 (“ +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 ”
...

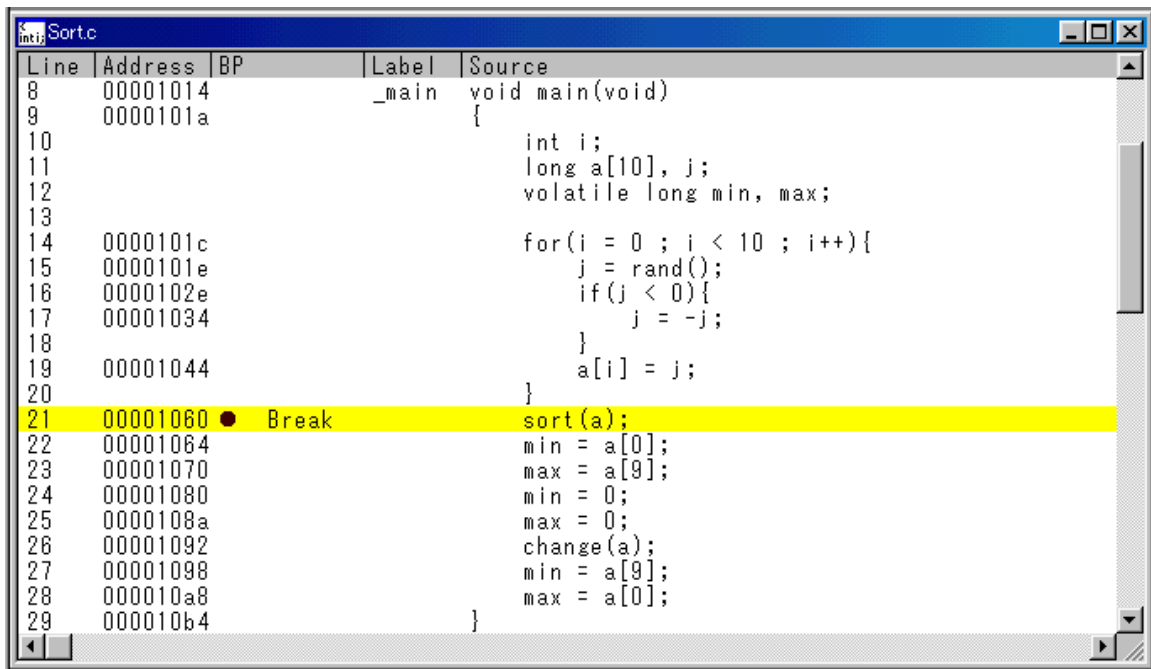
注：“ -Timer ”および“ -Trace ”は対応する“ +Timer ”および“ +Trace ”が設定されているときのみ選択可能となります。

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。

プログラムはブレークポイント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ウインドウ中でステートメントが強調表示されます。




The screenshot shows a window titled 'Sort.c' with a table of code lines. The table has columns for Line, Address, BP, Label, and Source. Line 21 is highlighted in yellow and has a red dot in the BP column, indicating a breakpoint. The source code is as follows:

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 4-13 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ウイ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  をクリックして、System Statusウインドウを開きPlatformシートを選択してください。

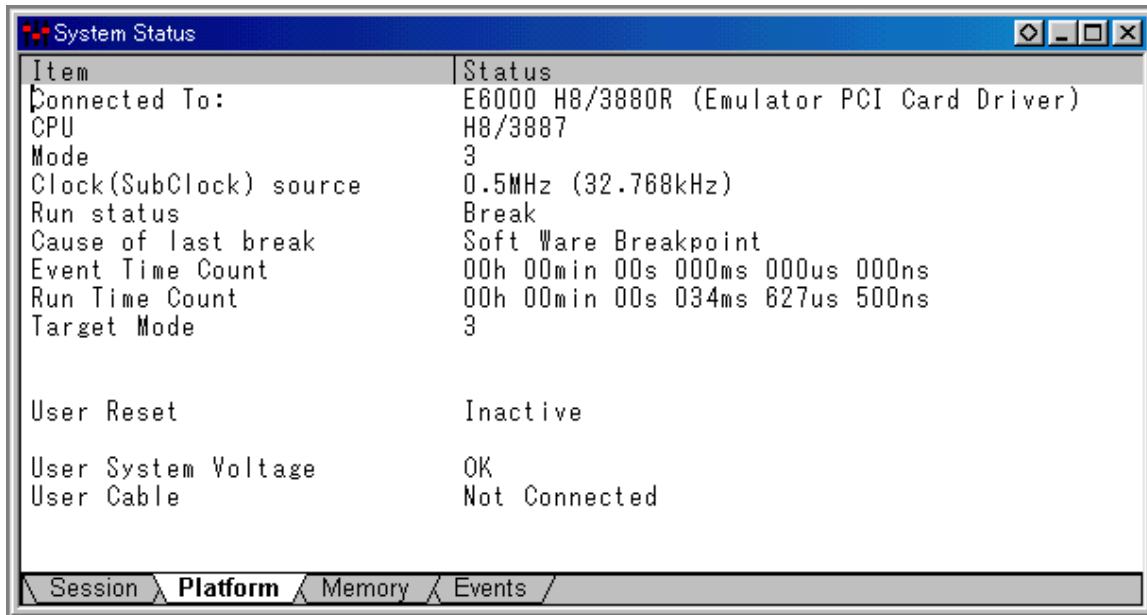



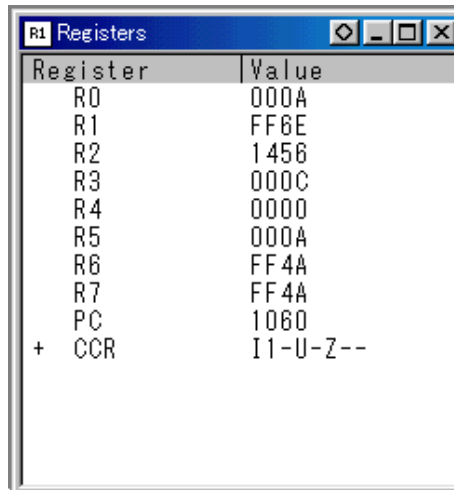
図 4-14 System Statusウインドウ (Platformシート)

[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34ms 624.000 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	FF8E
R2	1456
R3	000C
R4	0000
R5	000A
R6	FF4A
R7	FF4A
PC	1060
+ CCR	I1-U-Z--

図 4-15 Registersウィンドウ

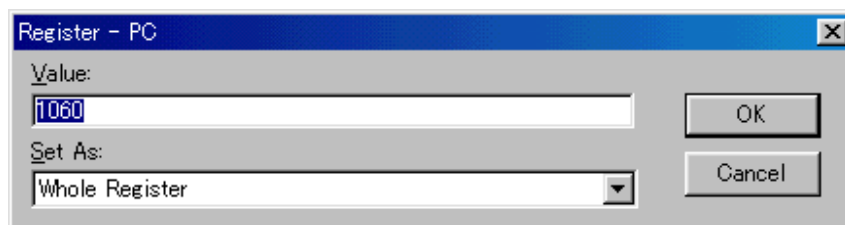
プログラムカウンタ(PC)の値は強調表示されたステートメントH'1060になっています。

(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:


Set As:

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ウイ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。

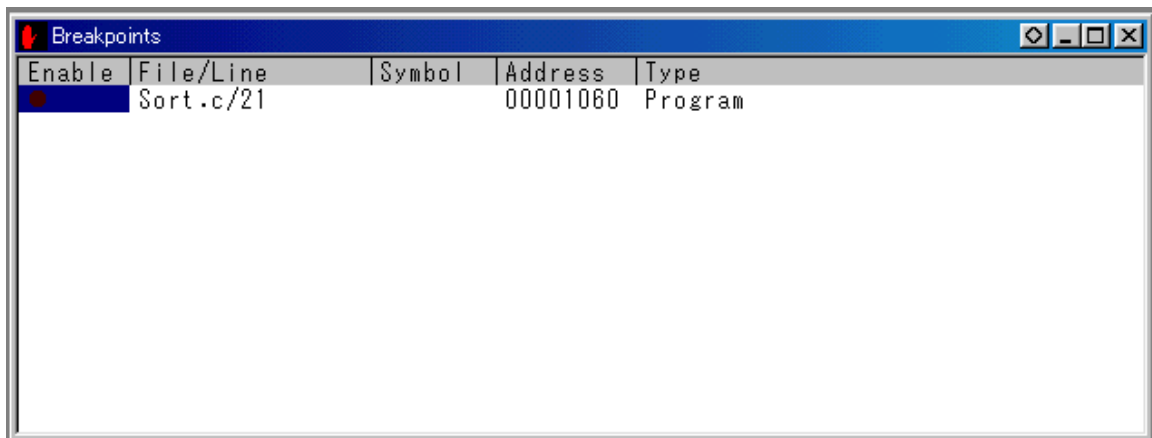


図 4-17 Breakpointsウインドウ


Breakpointsウインドウによって、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ウイ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] エディットボックスにmainを入力し、 [Format] ドロップダウンリストボックスをByteに設定してください。

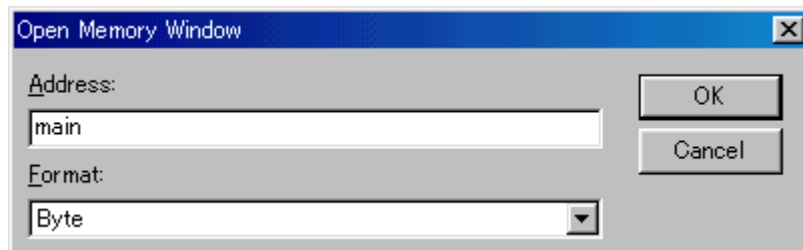


図 4-18 Open Memory Windowの設定

- ・ [OK] をクリックして、指定されたメモリ領域を示すMemoryウイ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。

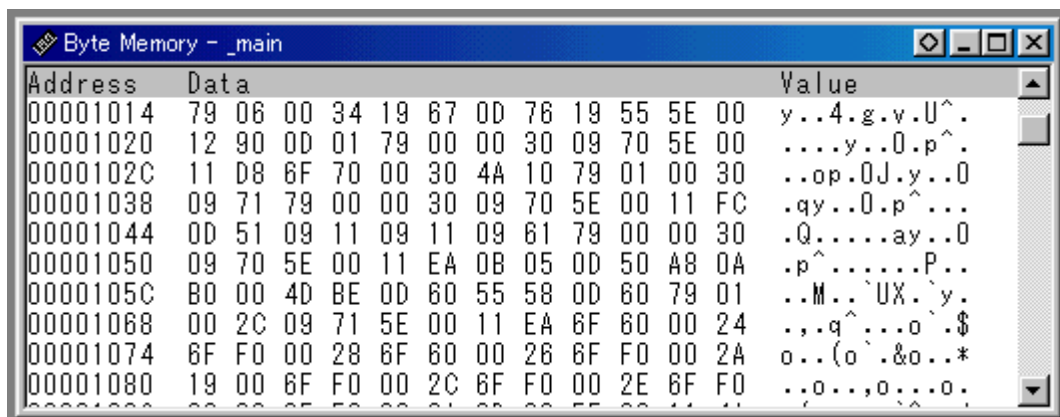


図 4-19 Memoryウインドウ(Byte)

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ウイ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ウイ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ウインドウに変数が表示されます。

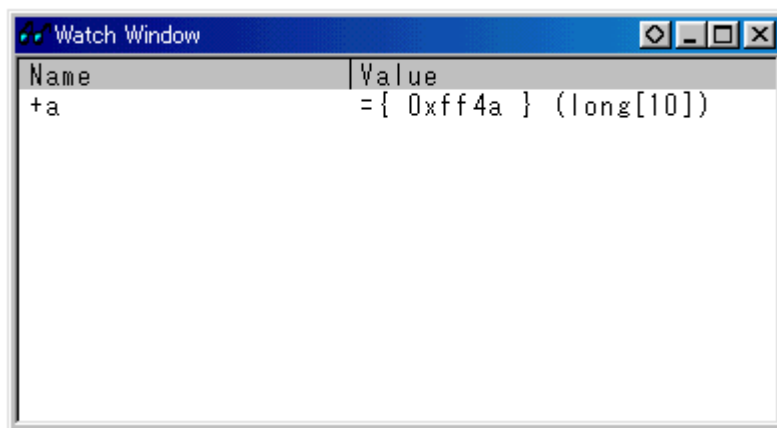



図 4-20 Watchウインドウ（変数追加後）

Watchウインドウのシンボル”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。

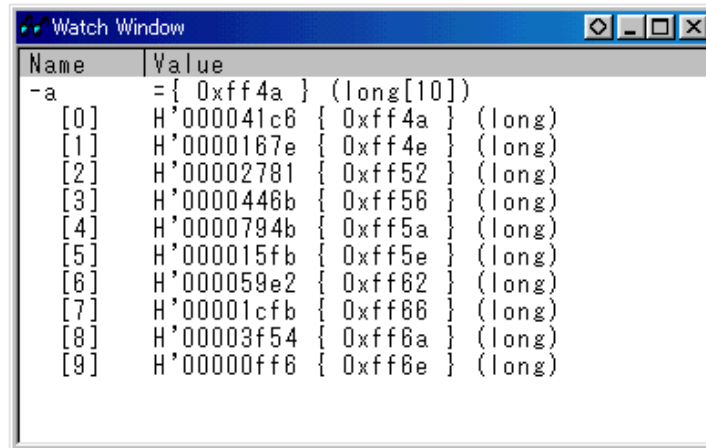


図 4-21 Watchウインドウ (シンボル拡張)

また、変数名を指定して、Watchウイ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ウイ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。

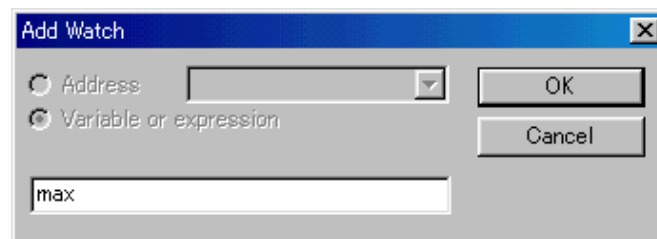


図 4-22 Add Watchダイアログボックス

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

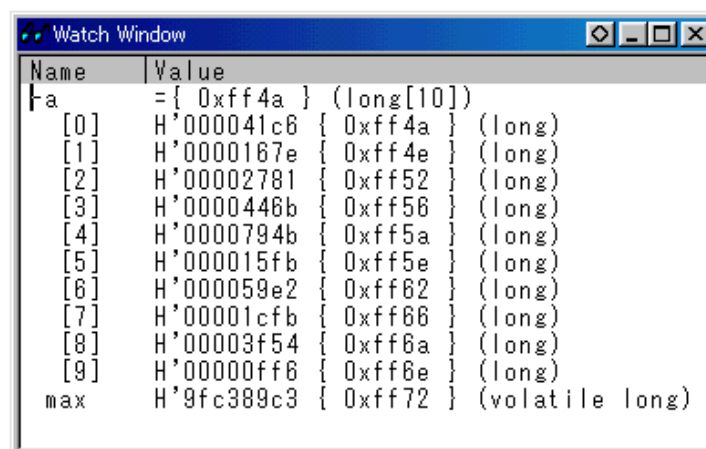


図 4-23 Watchウインドウ (変数の追加)


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。

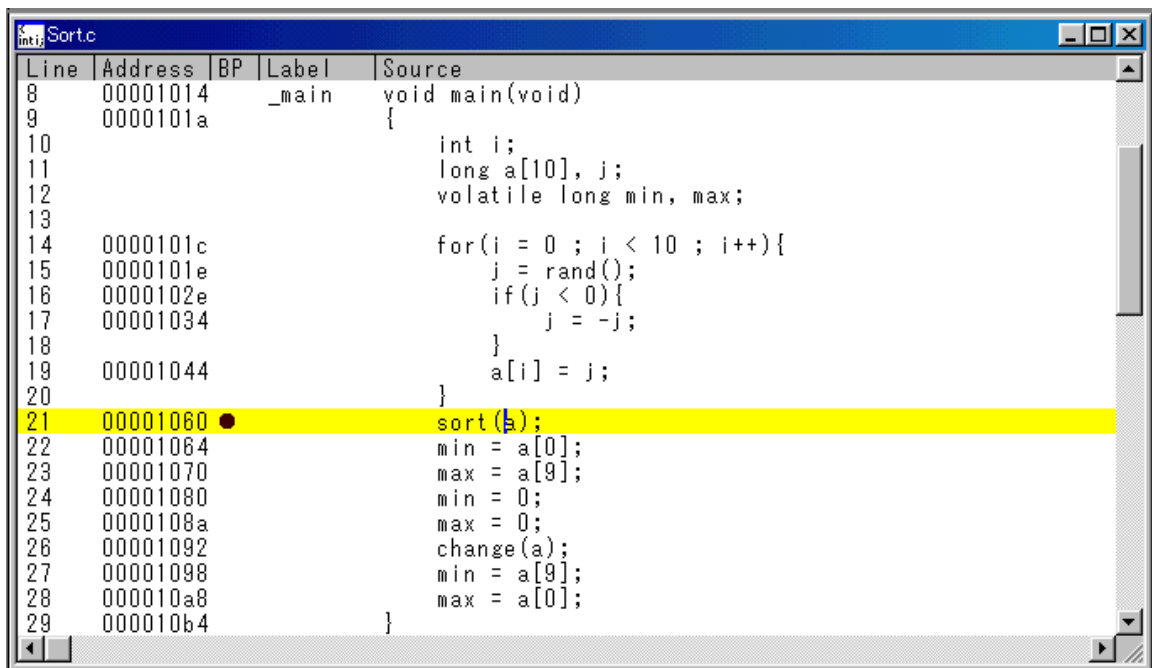

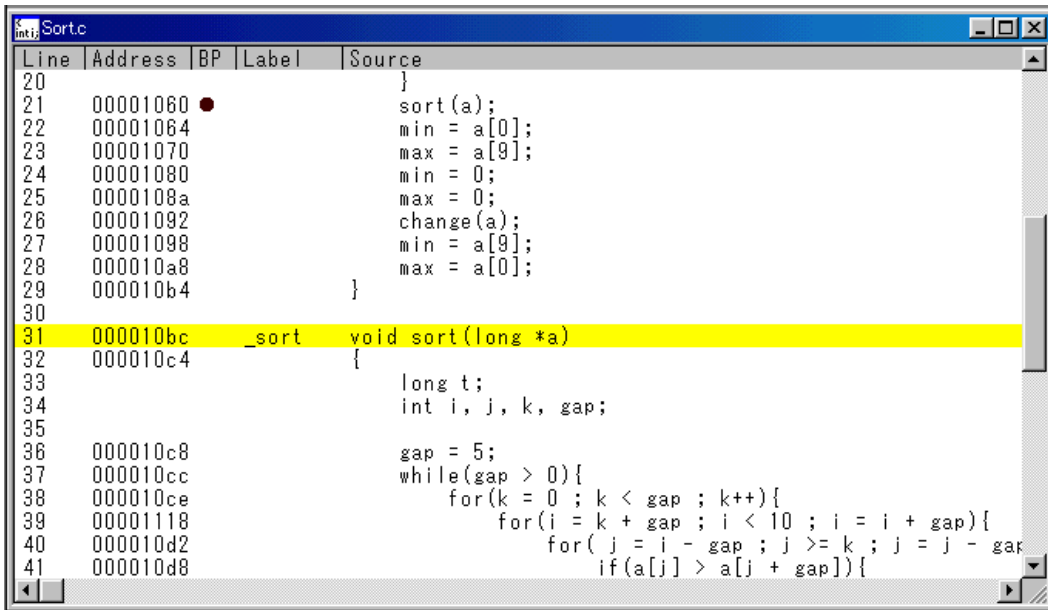



図 4-24 Reset Go実行後のProgramウィンドウ

- ・ 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。

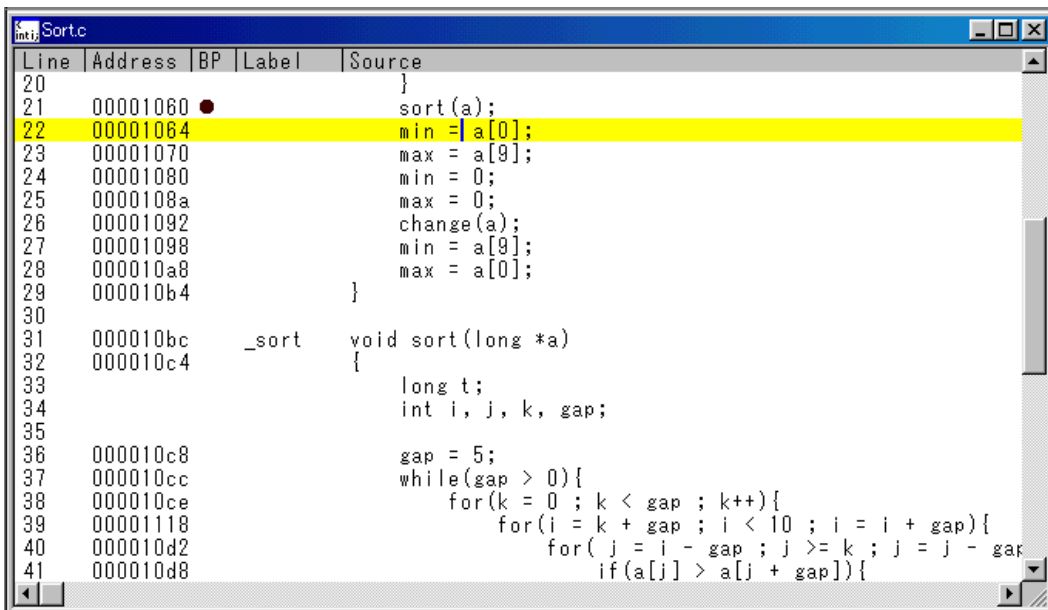


Line	Address	BP	Label	Source
20				}
21	00001060	●		sort(a);
22	00001064			min = a[0];
23	00001070			max = a[9];
24	00001080			min = 0;
25	0000108a			max = 0;
26	00001092			change(a);
27	00001098			min = a[9];
28	000010a8			max = a[0];
29	000010b4			}
30				
31	000010bc		_sort	void sort(long *a)
32	000010c4			{
33				long t;
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]){

図 4-25 Step In実行後のProgramウインドウ (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。



Line	Address	BP	Label	Source
20				}
21	00001060	●		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			}
30				
31	000010bc		_sort	void sort(long *a)
32	000010c4			{
33				long t;
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]){

図 4-26 Step Out実行後のProgramウインドウ

- 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。

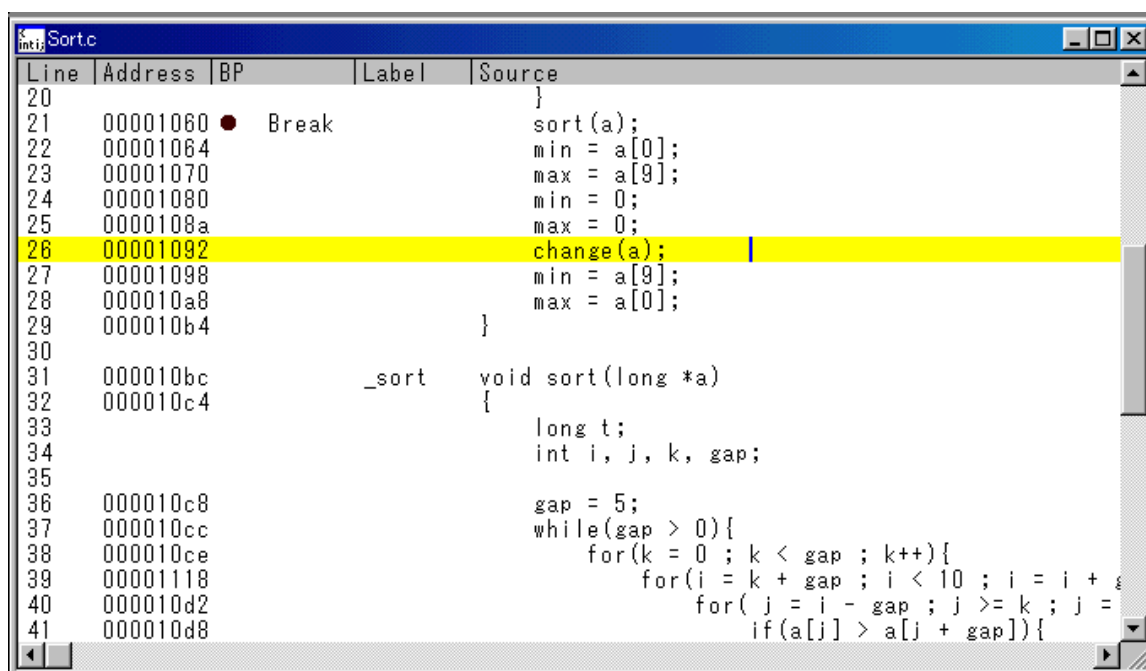



図 4-27 Step In実行後のProgramウィンドウ (2)

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。

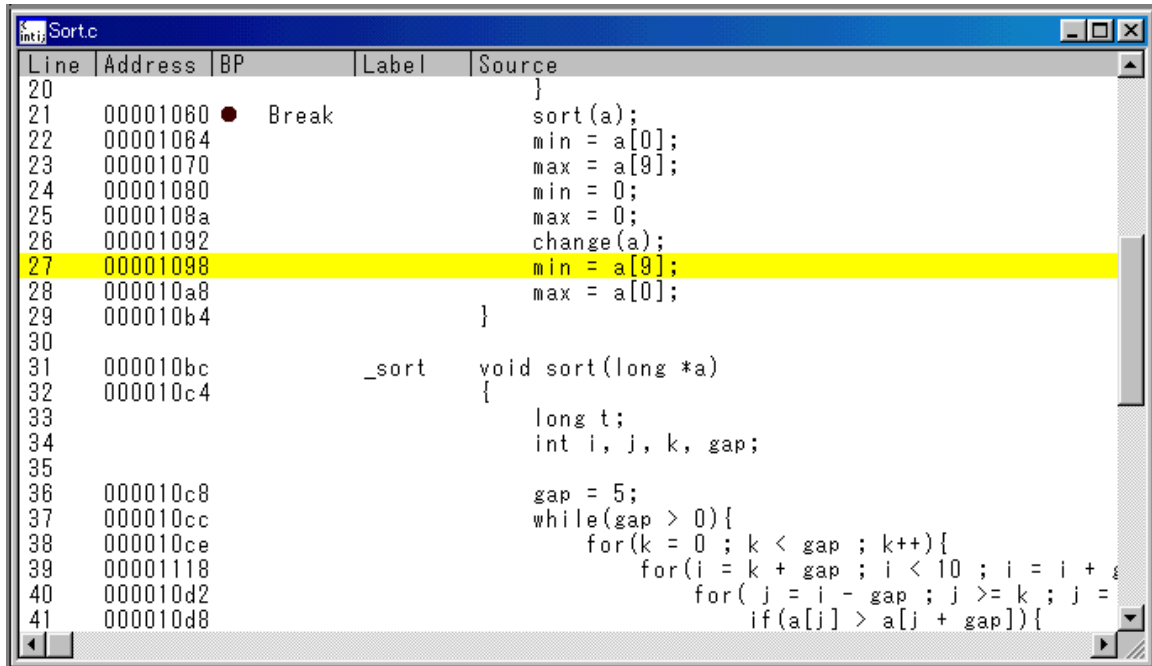



図 4-28 Step Over実行後のProgramウィンドウ

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。

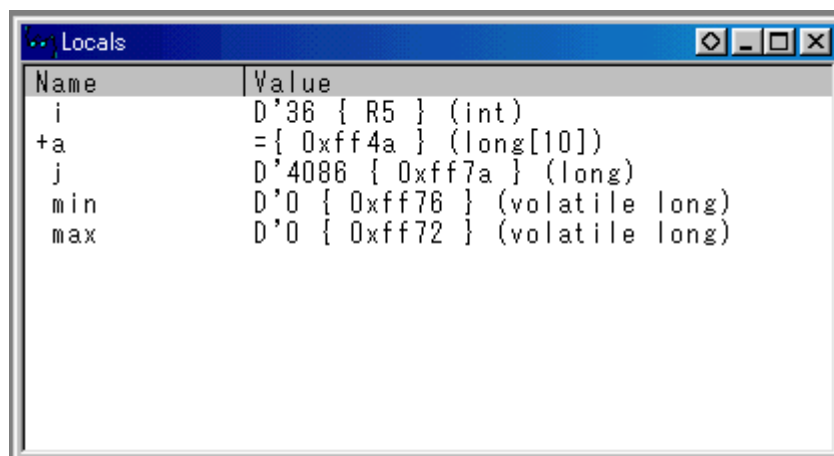



図 4-29 Localsウィンドウ

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。

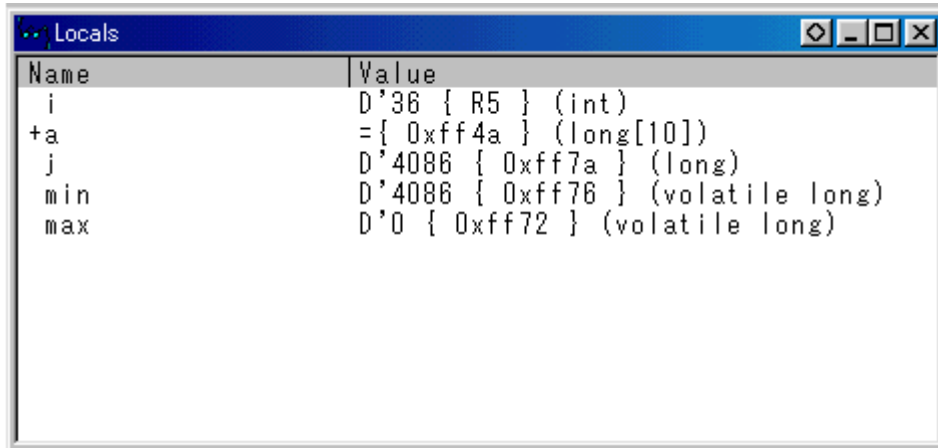


図 4-30 Localsウィンドウ (変数”min”内容変更後)

- ・ 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。

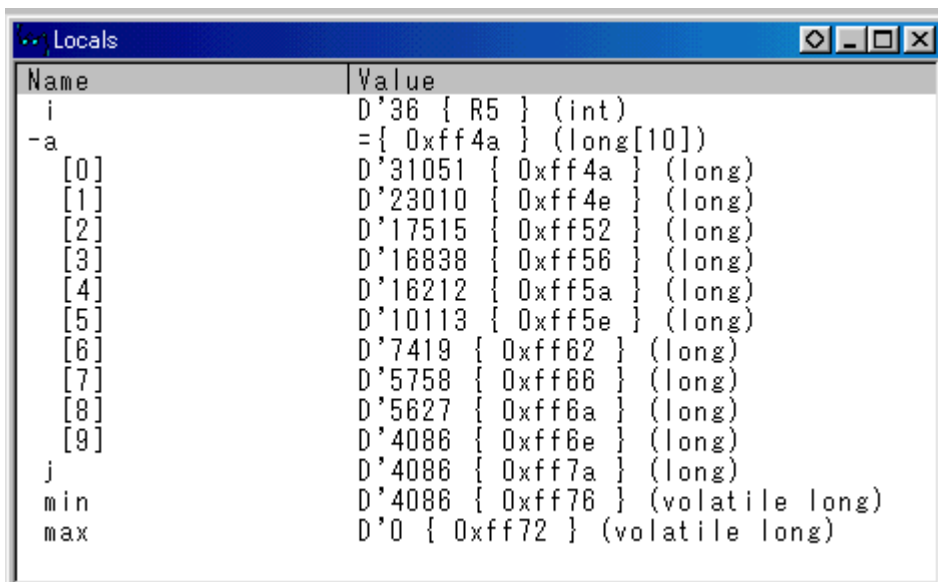


図 4-31 Localsウィンドウ (配列変数”a”ソート後)



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

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ハードウェアブレークポイントの設定

イベント検出システムを使用したハードウェアブレークポイントを設定して、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[Breakpoint]を選択するか、またはツールバーの[Breakpoint]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。

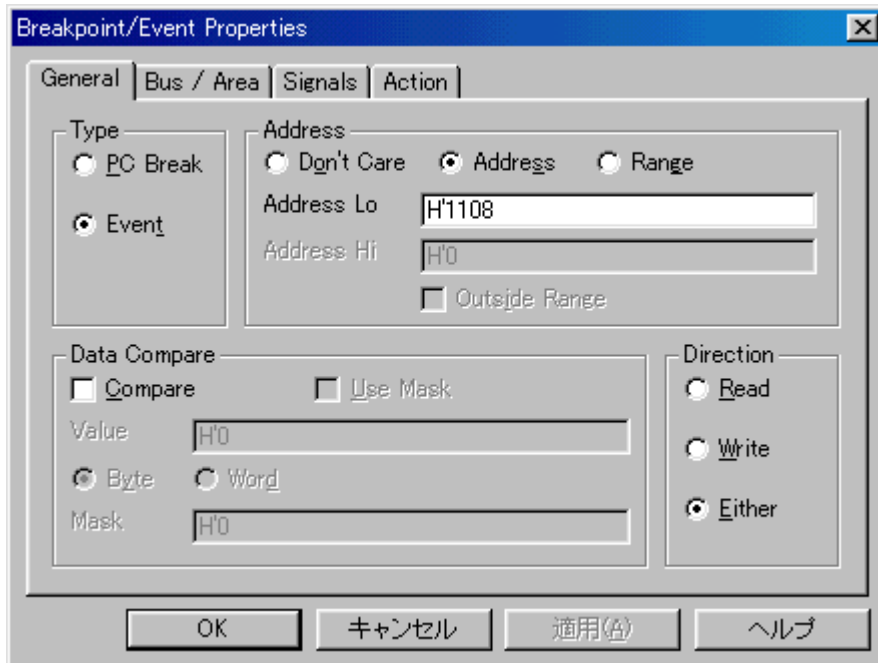


図 4-32 ブレークポイントの追加（アドレス指定）

- ・ [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。

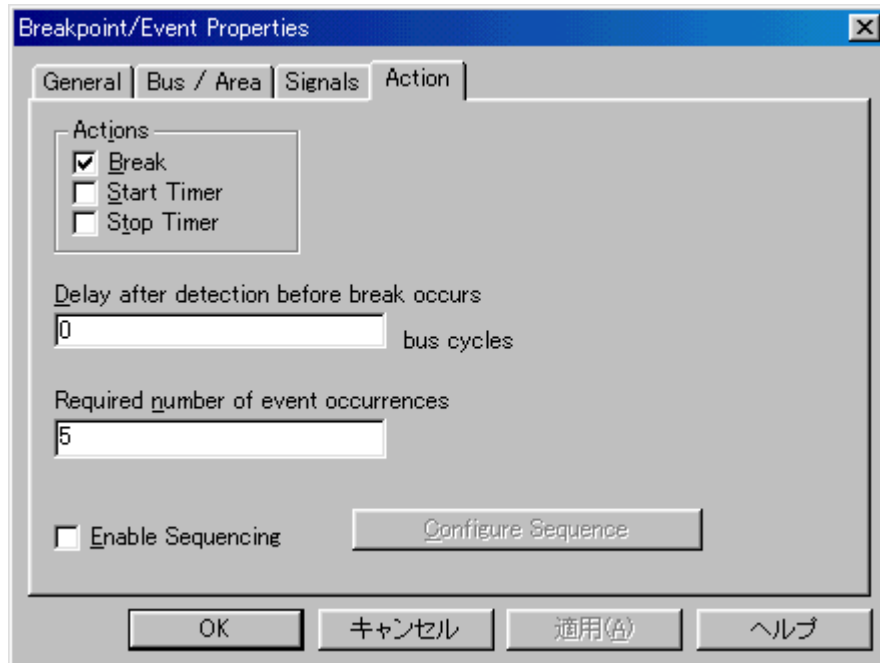


図 4-33 ブレークポイントの追加（回数指定）

- ・ [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。

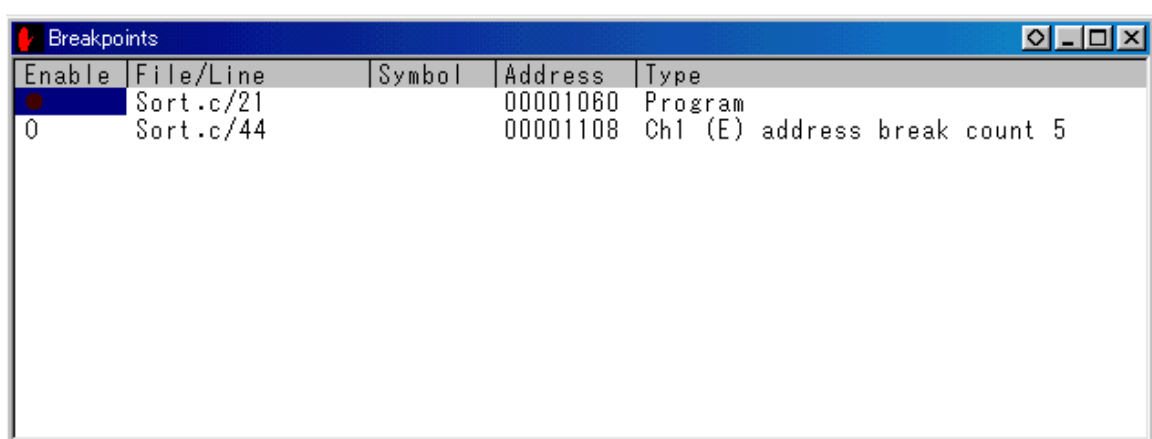




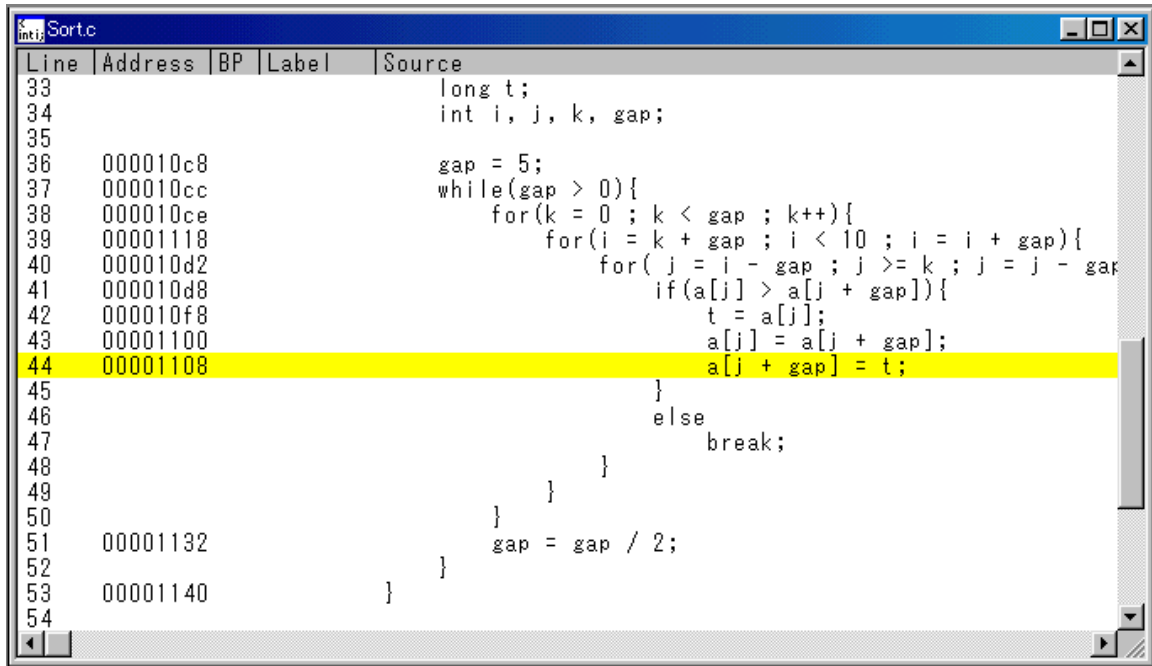
図 4-34 Breakpointsウィンドウ（追加後）

- ・ [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。



Line	Address	BP	Label	Source
33				long t;
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	00001132			gap = gap / 2;
52				}
53	00001140			}
54				

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

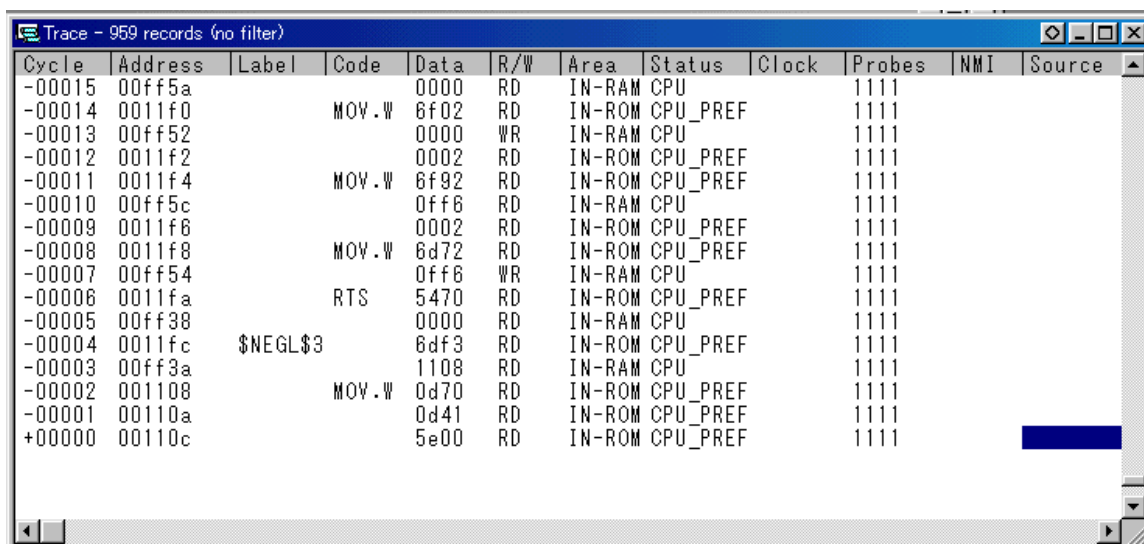
MCUの動作を確認するため、指定されたイベントの直前までのMCUサイクルはトレースバッファに記録されています。

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ウイ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インドウをスクロールダウンしてください。Traceウイ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	NMI	Source
-00015	00ff5a			0000	RD	IN-RAM CPU			1111		
-00014	0011f0		MOV.W	6f02	RD	IN-ROM CPU_PREF			1111		
-00013	00ff52			0000	WR	IN-RAM CPU			1111		
-00012	0011f2			0002	RD	IN-ROM CPU_PREF			1111		
-00011	0011f4		MOV.W	6f92	RD	IN-ROM CPU_PREF			1111		
-00010	00ff5c			0ff6	RD	IN-RAM CPU			1111		
-00009	0011f6			0002	RD	IN-ROM CPU_PREF			1111		
-00008	0011f8		MOV.W	6d72	RD	IN-ROM CPU_PREF			1111		
-00007	00ff54			0ff6	WR	IN-RAM CPU			1111		
-00006	0011fa		RTS	5470	RD	IN-ROM CPU_PREF			1111		
-00005	00ff38			0000	RD	IN-RAM CPU			1111		
-00004	0011fc	\$NEGL\$3		6df3	RD	IN-ROM CPU_PREF			1111		
-00003	00ff3a			1108	RD	IN-RAM CPU			1111		
-00002	001108		MOV.W	0d70	RD	IN-ROM CPU_PREF			1111		
-00001	00110a			0d41	RD	IN-ROM CPU_PREF			1111		
+00000	00110c			5e00	RD	IN-ROM CPU_PREF			1111		

図 4-36 Traceウインドウ

- ・ 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。

注: H8/3887シリーズ、H8/3867シリーズ、H8/3847シリーズ、H8/3827シリーズ、H8/3847Rシリーズ、H8/3827RシリーズではClockカラムには何も表示されません。
ソフトウェアブレークで停止した場合、Codeカラムに "Data"、Dataカラムに "5770" が表示されます。

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] タイプ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。

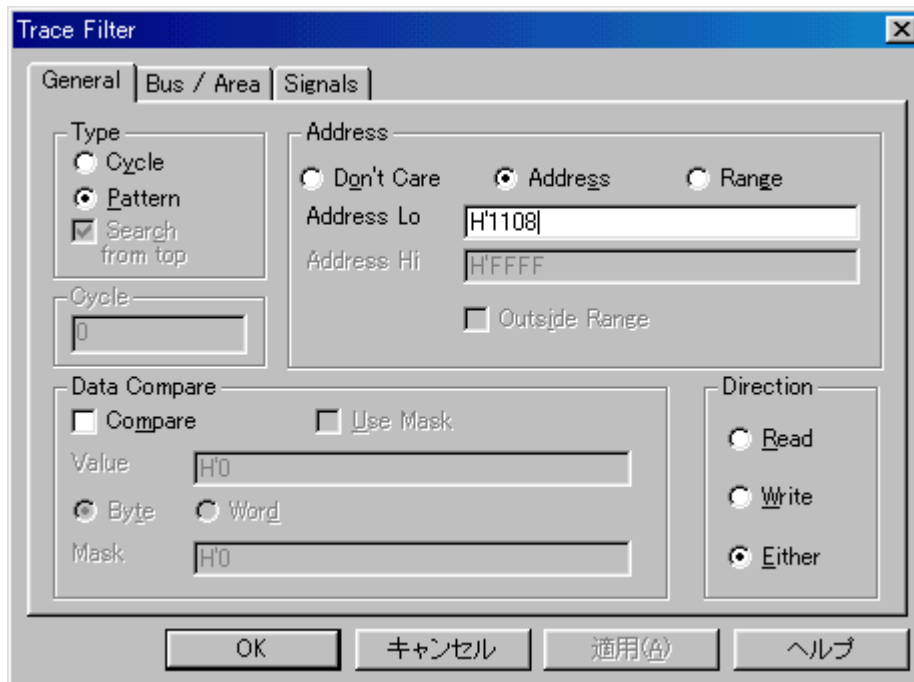


図 4-37 Trace Filterダイアログボックス

- [Bus / Area] タブをクリックし、 [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。

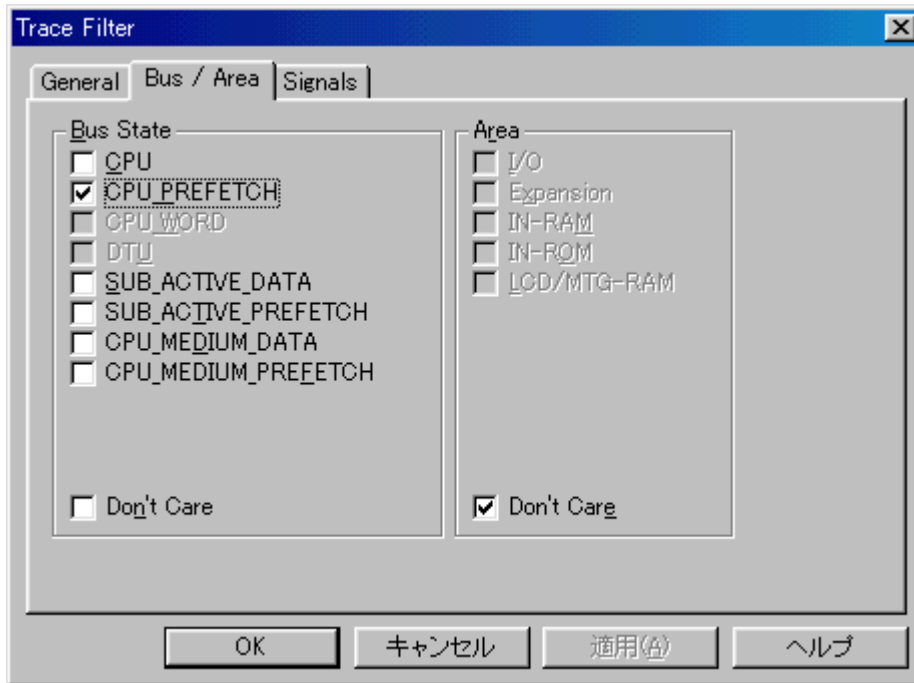


図 4-38 Bus / Areaの設定

- [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。


The screenshot shows the 'Trace' window with 5 records of 959. The records are as follows:

Cycle	Address	Label	Code	Data	R/W	Area	Status	Clock	Probes	NMI	Source
-00815	001108			0d70	RD	IN-ROM	CPU_PREF		1111		
-00588	001108			0d70	RD	IN-ROM	CPU_PREF		1111		
-00409	001108		MOV.W	0d70	RD	IN-ROM	CPU_PREF		1111		
-00250	001108		MOV.W	0d70	RD	IN-ROM	CPU_PREF		1111		
-00002	001108		MOV.W	0d70	RD	IN-ROM	CPU_PREF		1111		

図 4-39 Traceウィンドウ (トレースフィルタ指定)

4.10 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ウインドウから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]ウインドウを開いてください。

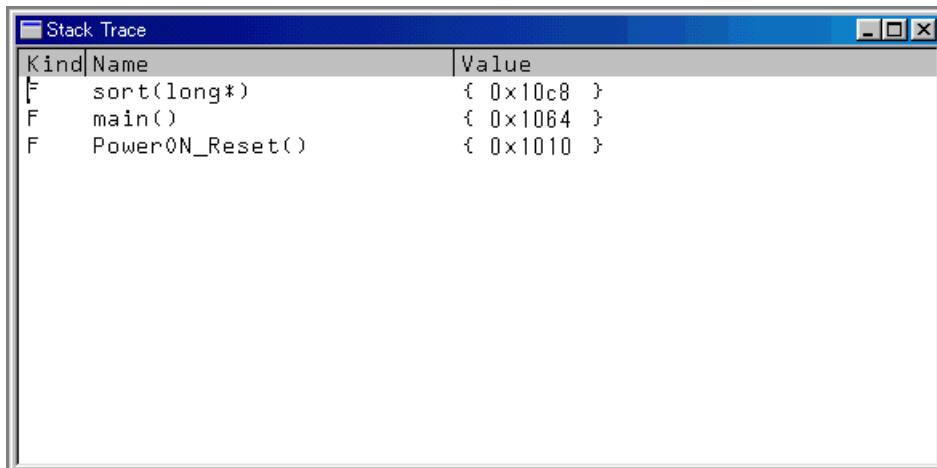


図 4-40 Stack Traceウインドウ

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、各ウインドウで[Help]ボタンまたは[F1]キーを押すと表示されます。

4.11 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。

4.12 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主な特長と、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エミュレーションツール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「日立デバッグインタフェース ユーザーズマニュアル」を参照してください。

5 H8/3887シリーズ、H8/3867シリーズ、H8/3847シリーズ、 H8/3827シリーズ、H8/3847Rシリーズ、H8/3827Rシリーズ用 E6000エミュレータ HDIの機能

本章は、H8/3887シリーズ、H8/3867シリーズ、H8/3847シリーズ、H8/3827シリーズ、H8/3847Rシリーズ、H8/3827Rシリーズ専用の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ーズマニュアル」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ーズマニュアル」(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...		-
	Load Session...		-
	Save Session		4.11
	Save Session As...		-
	Load Program...		4.4
	Initialise		-
	Exit		-
Edit Menu	Cut		-
	Copy		-
	Paste		-
	Find...		-
	Evaluate...		-
View Menu	Breakpoints		4.5.4, 4.8.1, 5.2, 5.3
	Command Line		5.7
	Disassembly...		-
	I/O Area		-
	Labels		-
	Locals		4.7.3
	Memory...		4.6.1
	Performance Analysis		-
	Registers		4.5.3
	Source...		4.4
	Status		4.5.2
	Trace		4.9, 5.5
	Watch		4.6.2

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
Run Menu	Reset CPU		-
	Go		4.5.2
	Reset GO		4.5.2
	Go to Cursor		-
	Set PC To Cursor		-
	Run...		-
	Step In		4.7
	Step Over		4.7
	Step Out		4.7
	Step...		-
	Halt		-
Memory Menu	Refresh		-
	Load...		-
	Save...		-
	Verify...		-
	Test...		-
	File...		-
	Copy...		-
	Compare...		-
	Configure Map...		4.3.2
Configure Overlay...		-	
Setup Menu	Status bar		-
	Options...		-
	Radix		-
	Customise		-
	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 コンフィグレーションダイアログボックス

このダイアログボックスではE6000 H8/388Rエミュレータの各種パラメータを設定します。

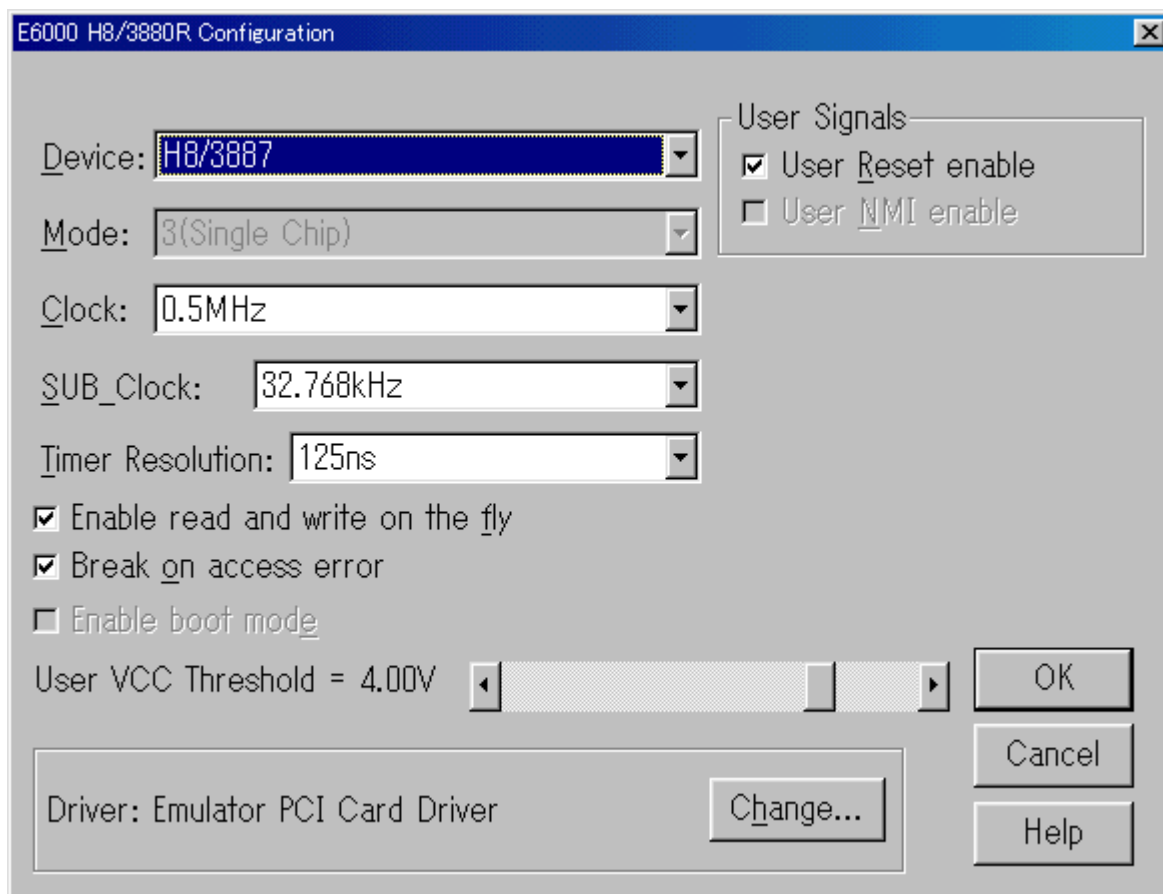


図 5-1 Configurationダイアログボックス

Configurationダイアログボックスを表示するには、[Setup]メニューから [Configure Platform...] を選択してください。

以下の表は、Configurationダイアログボックスで表わされるオプションについて説明します。

表 5-2 Configurationダイアログボックス

オプション	説明
デバイス (Device)	エミュレーションするMCUを指定します。
モード (Mode)	MCUの動作モードを指定します。動作モードは3に固定されています。
エミュレーションクロック (Clock)	MCUクロックの速度を指定します。H8/3887シリーズ、H8/3867シリーズ、H8/3847シリーズ、H8/3827シリーズ、H8/3847Rシリーズ、H8/3827Rシリーズは、8MHz、2MHz、0.5MHz、またはTarget/2が選択できます。MCUのサブクロックを指定します。32.768kHz、38.4kHz、307.2kHz、またはTargetが選択できます。
タイマ分解能 (timer Resolution)	実行時間の測定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 のいずれかの値に設定できます。
ユーザ信号 (User Signals)	ユーザシステムからのリセット信号を有効または無効にできます。ボックスをチェックすると、信号が有効になります。
プログラム実行中のリードライトの許可 (Enable read and write on the fly)	プログラム実行中でのユーザメモリのリードライトを有効または無効にできます。
ブートモード (Enable boot mode)	MCUのフラッシュメモリへのブートプログラミング動作を有効にします。(H8/3887シリーズ、H8/3867シリーズ、H8/3847シリーズ、H8/3827シリーズ、H8/3847Rシリーズ、H8/3827Rシリーズは使用できません。)
アクセスエラーでのブレーク (Break on access error)	不当なアクセス (ROM領域へのライト等) ブレークを有効または無効にできます。チェックしなければ、ROMへのあらゆる書き込みおよびアクセス禁止された領域へのアクセスが無視されます。
ユーザVccスレッシュホールド (User VCC Threshold = x.xxV)	ユーザシステム電圧レベルを監視します。スレッシュホールドによって設定された値より低くなると、ユーザVccが低いことをSystem Statusウィンドウによってユーザに知らせます。

5.2 ブレークポイント

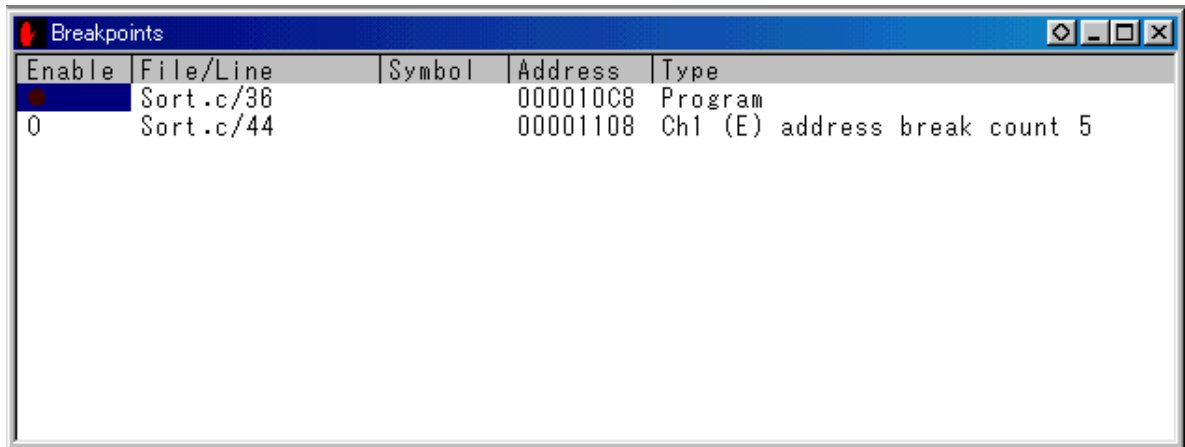



図 5-2 Breakpoints ウィンドウ

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。

Breakpointsウィンドウを表示するには、[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。

現状のブレークポイントを変更するには、それをダブルクリックするか、あるいは[Breakpoints]リストの中から選択した後、ポップアップメニューより[Edit...]を選択してください。

ブレークポイントを許可または禁止するには、[Breakpoints]リストの中から選択した後、ポップアップメニューより[Disable/Enable]を選択してください。ブレークポイントが許可されると、[Enable]コラムの中に表示されます。

ブレークポイントを削除するには、ブレークポイントリストの中から選択した後、ポップアップメニューより[Delete]を選択してください。全てのブレークポイントを削除するには、[Delete All]を選択してください。

新しいブレークポイントを設定するには、ポップアップメニューより[Add...]を選択して、Breakpoint/Event Propertiesダイアログボックスを表示してください。そして、付け加えたいブレークポイントの条件を設定してください。

Breakpoints/Event Propertiesダイアログボックスの詳細に関しては、以下の「5.3 イベント検出システム」を参照してください。

5.2.1 プログラムブレークポイントを設定する

プログラムブレークポイントを設定するには、[Type] を [PC Break] に設定し、[Address Lo] フィールドにブレークポイントのアドレスを入力してください。

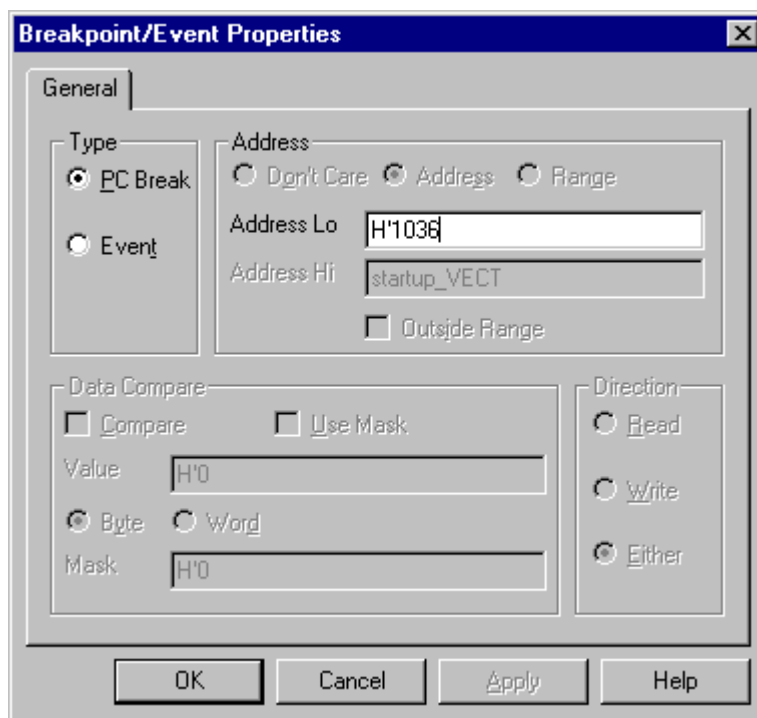


図 5-3 ブレークポイントの設定

あるいは、プログラムウィンドウの [Break] カラムをダブルクリックしてください。

5.3 イベント検出システム

イベント検出システムに、MCU信号の状態を指定したイベントを設定できます。このイベントは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。

イベント検出システムはイベントチャンネルおよび範囲チャンネルで構成し、指定されたイベントがいつ発生したかを検出できます。また、最大8イベントによるシーケンス指定も可能です。この場合、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。

以下の表は、イベントチャンネルおよび範囲チャンネルで指定できる条件を示します。

表 5-3 イベントチャンネルとオプションの設定

オプション	イベントチャンネル	範囲チャンネル
アドレスまたはアドレス範囲内		
アドレス範囲外		
データの値 (マスク機能付)		
リードまたはライト		
アクセスタイプ (命令プリフェッチなど)		
アクセス領域 (内蔵ROM、内蔵RAMなど)		
外部プローブの値		
イベントの一致回数		
シーケンスでの組み合わせが可能		

Breakpoint/Event Propertiesダイアログボックスによって、各イベントをブレーク、トレース、もしくはイベント間実行時間測定の内、どれに使用するが設定します。

イベントをブレークに設定するには、[**Type**] を [**Event**] に設定してください。すると、Breakpoint/Event Propertiesダイアログボックスは、4つのオプションパネルを表わし、ブレークポイントで使うイベントの全ての条件 (General, Bus / Area, Signals, Action) を設定できます。

5.3.1 [General]

[General] パネルによって、イベントチャンネルのアドレスとデータアクセス条件を設定できます。

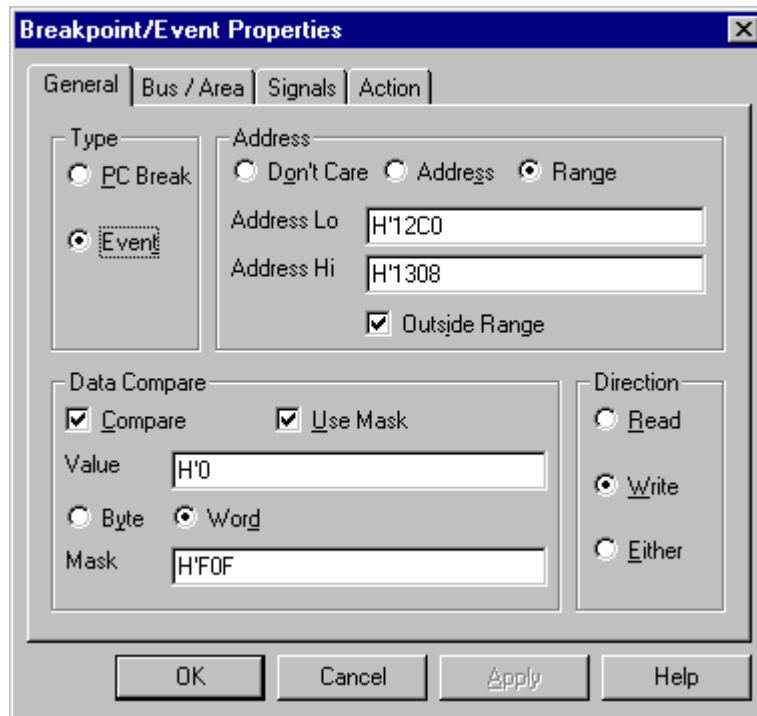


図 5-4 イベントの設定 (General)

アドレス

アドレスまたはアドレスの範囲がアクセスされたときに、チャンネルを起動できます。指定された範囲外のアドレスへアクセスするとチャンネルを起動するように指定するには、[Outside Range] を選択してください。

データコンペア

設定した値とデータバスの値が一致したときにチャンネルを起動できます。データの特定のビットをマスクする場合は、[Use Mask] を選択してください。

ディレクション

リード、ライト、またはリードライトアクセス時にチャンネルが起動できます。

5.3.2 [Bus / Area]

特定のバスステートまたはアクセスされたメモリ領域でチャンネルを起動できます。

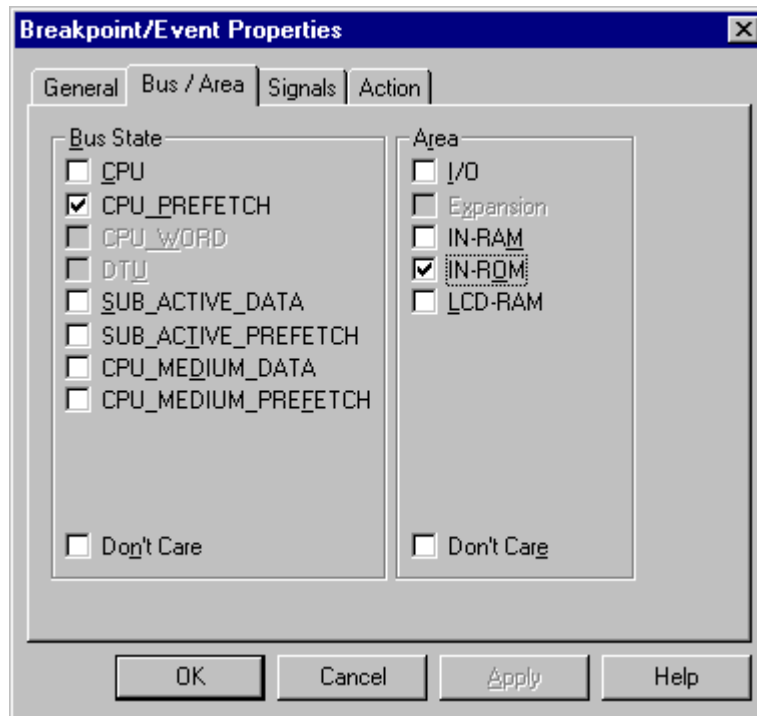


図 5-5 イベントの設定 (Bus / Area)

5.3.3 [Signals]

4つの外部プローブ信号の特定の組み合わせでイベントを起動するように指定します。

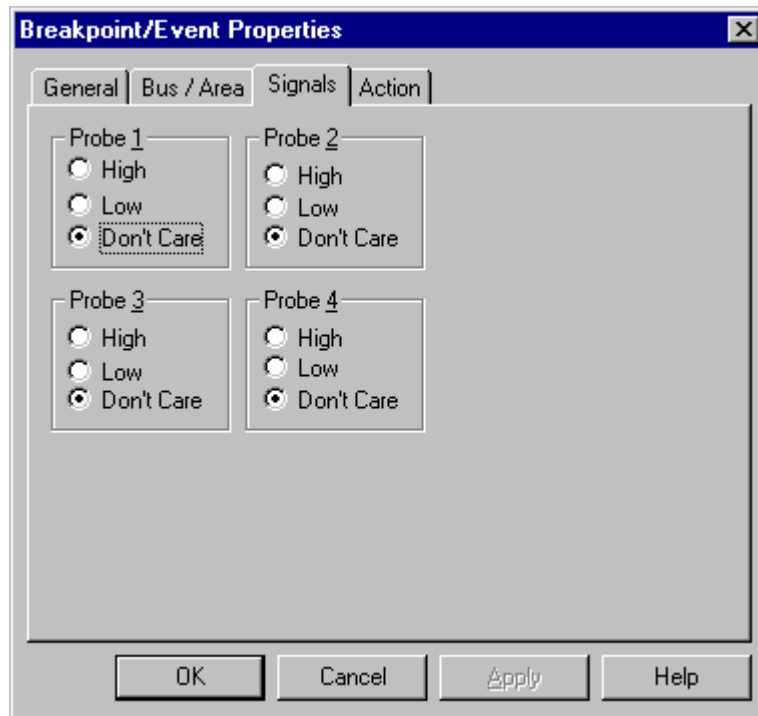


図 5-6 イベントの設定 (Signals)

5.3.4 [Action]

イベントが起動されるときアクションを指定します。

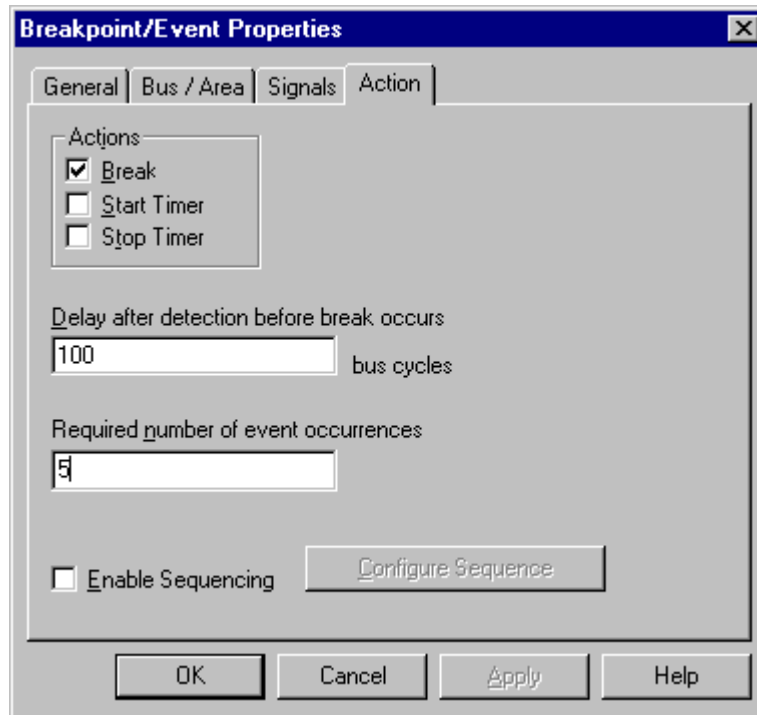


図 5-7 イベントの設定 (Action)

表5-4のアクションが指定できます。

表 5-4 イベントアクション

アクション	説明
ブレーク	プログラムの実行を停止します。
タイマ開始	イベント間実行時間測定タイマを開始します。タイマ分解能については、「5.1 コンフィグレーションダイアログボックス」を参照してください。
タイマ停止	イベント間実行時間測定タイマを停止します。

イベントが起動された後、指定された数のバスサイクルだけチャンネルの起動を遅らせるには、
[**Delay after detection before break occurs**] フィールドにバスサイクル数を入力してください。

指定された回数だけイベントが起動されるまでチャンネルの起動を遅らせるには、[**Required number of event occurrences**] フィールドに必要なイベント発生数を入力してください。

イベントのシーケンスを生成するには、シーケンスの一部を形成する全てのイベントの [**Enable Sequencing**] オプションを選択してください。

5.3.5 イベントシーケンス

シーケンスを構成するには、[Action] パネルで[Enable Sequencing]チェックボックスをチェックし、[Configure Sequence] ボタンをクリックしてください。Event Sequencingダイアログボックスが表示されます。

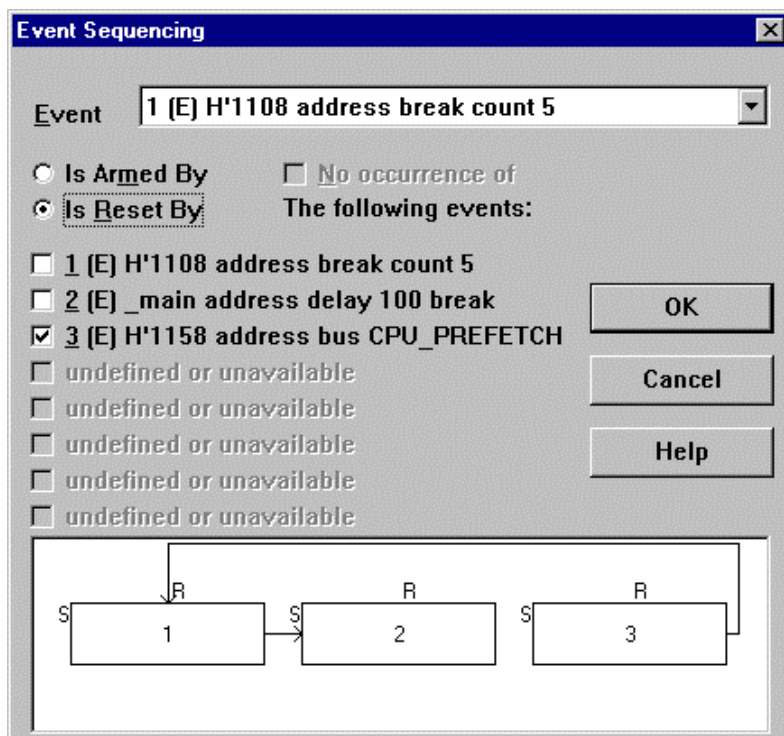


図 5-8 イベントシーケンス画面 (1)

このダイアログボックスによって、シーケンスの各イベントにおいて、イベントの前提条件となるイベント、またはリセットするイベントを1つ以上指定できます。

[Event] ドロップダウンリストボックスから、構成したいイベントを選択してください。これによって、[Enable Sequencing] で指定されたあらゆるイベントが選択できます。

現在選択されているイベントにおいて、[Is Armed By] をクリックして、イベントの前提条件となるイベントを設定してください。

同様に、[Is Reset By] をクリックして、イベントをリセットするイベントを設定してください。

5.3.6 イベントの前提条件

たとえば、4つのアドレスのシーケンスが発生した時にだけ起動されるイベントシーケンスを定義するには、以下のように設定します。

- 4 is armed by 3.
- 3 is armed by 2.
- 2 is armed by 1.

Event Sequencingダイアログボックスは、定義したシーケンスを図式表示します。

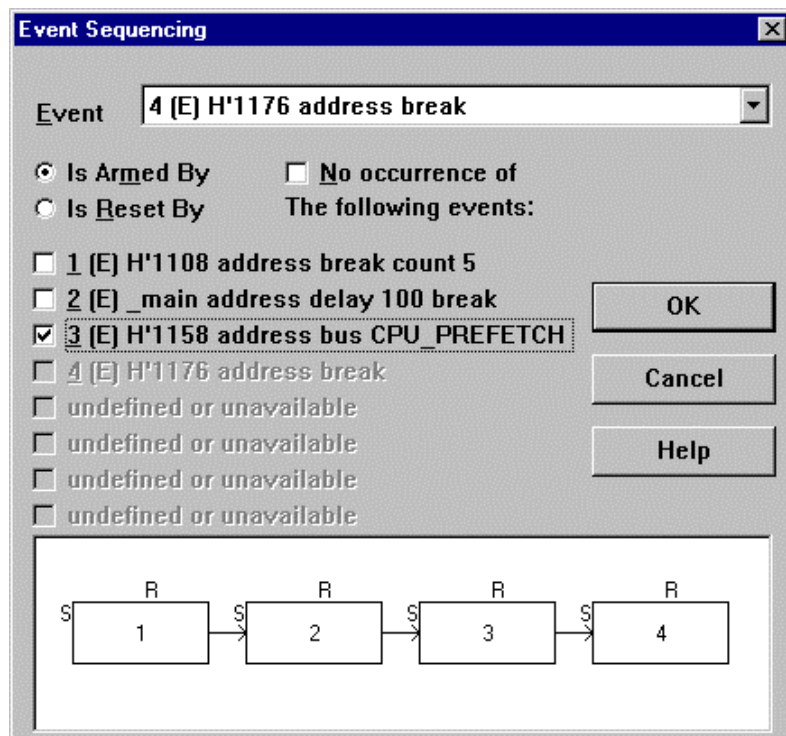


図 5-9 イベントシーケンス画面 (2)

シーケンスを定義するときは、シーケンスの最後のイベントだけを [Break] として定義してください。

5.3.7 イベントをリセットする

イベントが、シーケンスの中の別のイベントによってリセットされるように、指定することもできます。たとえば、イベント2の後にイベント3が続き、その後にイベント4が続いて、イベント1はその間には発生しない(もし、この間にイベント1が発生した場合は、前提条件のイベント発生がクリアされる場合)という条件でブレークが発生するようにするには、イベントシーケンスを以下のように設定してください。

4 is armed by 3 and reset by 1.

3 is armed by 2 and reset by 1.

2 is reset by 1.

1 is reset by 1.

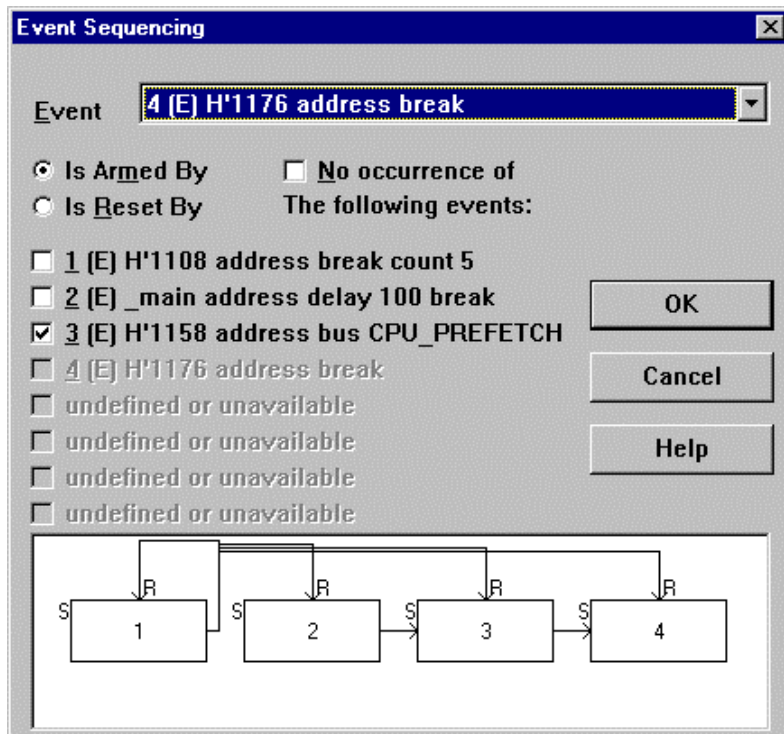



図 5-10 イベントシーケンス画面 (3)

5.4 メモリマッピングダイアログボックス

Memory Mappingダイアログボックスは、エミュレーションメモリのマッピングの表示および編集ができます。

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。

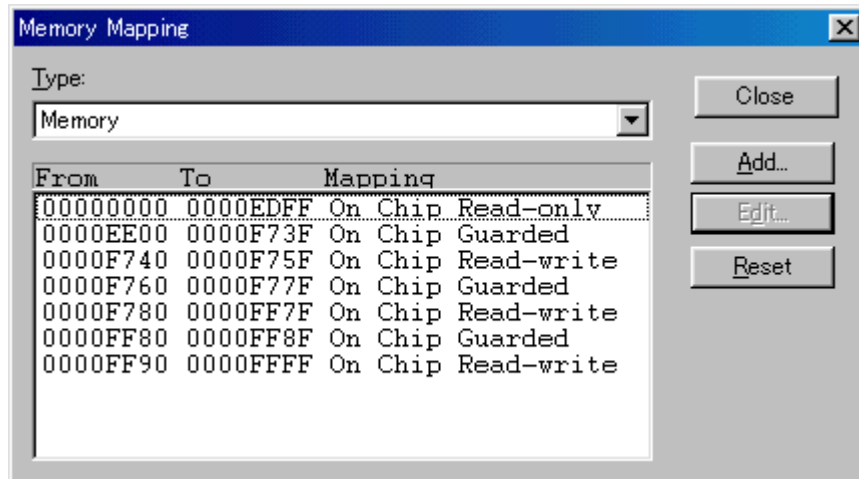


図 5-11 Memory Mappingダイアログボックス

メモリマッピングを編集するには、該当メモリブロックをダブルクリックしてください。あるいは、メモリマッピングリストの中から選択し、[Edit]をクリックしてください。以下のEdit Memory Mappingダイアログボックスがメモリブロックの現在の設定を示します。

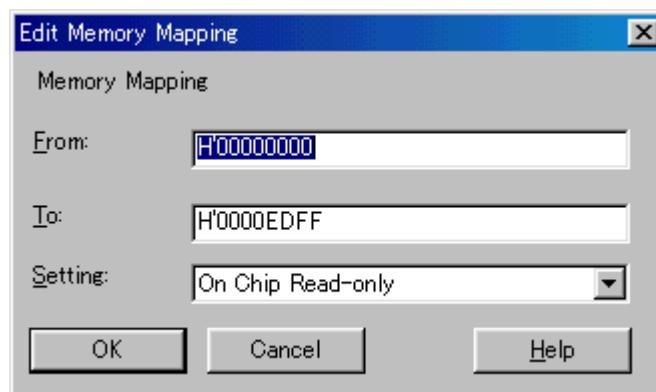


図 5-12 メモリマッピングの変更

[From]フィールドと[To]フィールドに、メモリブロックのアドレスの範囲を指定してください。そして、[Setting]ドロップダウンリストボックスで、メモリのタイプを選択してください。

以下のオプションが選択できます。

表 5-5 メモリタイプの定義

メモリ	説明
オンチップ	MCU内蔵メモリをアクセスします。
エミュレータ	エミュレーションメモリをアクセスします。

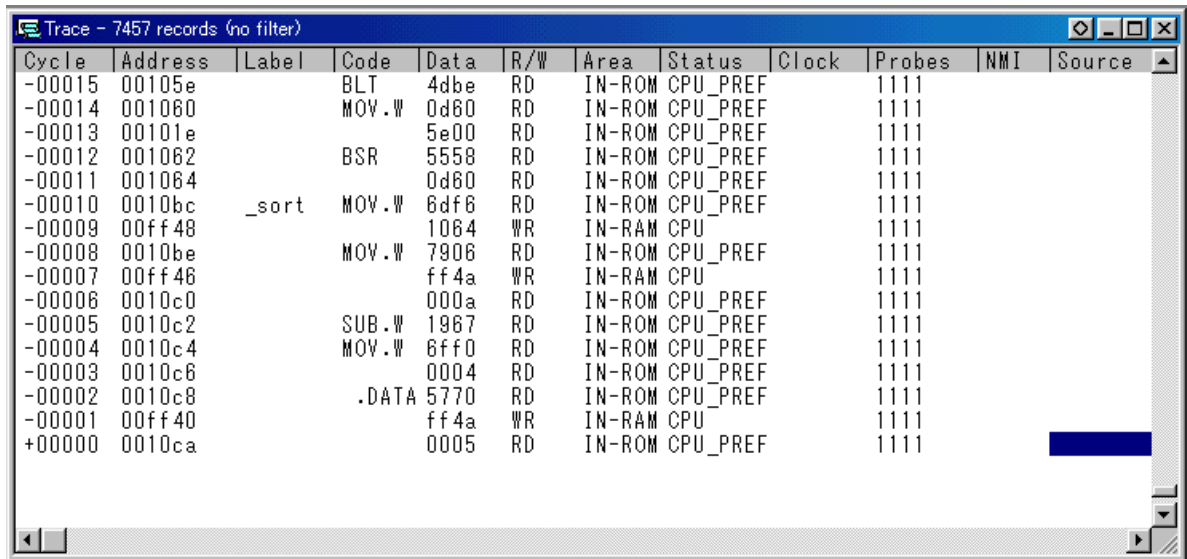
オプションにおいて、以下の3つのアクセスタイプから1つを指定できます。

表 5-6 アクセスタイプの定義

アクセスタイプ	説明
Read-Write	RAM
Read-Only	ROM
Guarded	アクセス不可

選択したMCUタイプとモードのデフォルトマッピングにメモリマッピングをリセットするには、**[Reset]** ボタンをクリックしてください。

5.5 トレースウインドウ




The screenshot shows a window titled "Trace - 7457 records (no filter)". It contains a table with the following columns: Cycle, Address, Label, Code, Data, R/W, Area, Status, Clock, Probes, NMI, and Source. The data rows are as follows:

Cycle	Address	Label	Code	Data	R/W	Area	Status	Clock	Probes	NMI	Source
-00015	00105e		BLT	4dbe	RD	IN-ROM	CPU_PREF		1111		
-00014	001060		MOV.W	0d60	RD	IN-ROM	CPU_PREF		1111		
-00013	00101e			5e00	RD	IN-ROM	CPU_PREF		1111		
-00012	001062		BSR	5558	RD	IN-ROM	CPU_PREF		1111		
-00011	001064			0d60	RD	IN-ROM	CPU_PREF		1111		
-00010	0010bc	_sort	MOV.W	6df6	RD	IN-ROM	CPU_PREF		1111		
-00009	00ff48			1064	WR	IN-RAM	CPU_PREF		1111		
-00008	0010be		MOV.W	7906	RD	IN-ROM	CPU_PREF		1111		
-00007	00ff48			ff4a	WR	IN-RAM	CPU_PREF		1111		
-00006	0010c0			000a	RD	IN-ROM	CPU_PREF		1111		
-00005	0010c2		SUB.W	1967	RD	IN-ROM	CPU_PREF		1111		
-00004	0010c4		MOV.W	6ff0	RD	IN-ROM	CPU_PREF		1111		
-00003	0010c6			0004	RD	IN-ROM	CPU_PREF		1111		
-00002	0010c8		.DATA	5770	RD	IN-ROM	CPU_PREF		1111		
-00001	00ff40			ff4a	WR	IN-RAM	CPU_PREF		1111		
+00000	0010ca			0005	RD	IN-ROM	CPU_PREF		1111		

図 5-13 Traceウインドウ

Traceウインドウは、トレースバッファの内容を表示します。

Traceウインドウを表示するには、[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  をクリックしてください。

トレースバッファの記憶データは、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使われた場合は、アセンブリ言語だけが表示されます。

本エミュレータでは、何も表示しません。

4本の外部プローブの値をProbesカラムに表示します。右から順に外部プローブ番号1、2、3、4となり、Highを1、Lowを0と表示します。

[NMI]カラムは、本エミュレータでは、何も表示しません。

ポップアップメニューより[Clear]を選択することによりトレースバッファを削除できます。また[Save]をクリックして、トレースバッファの内容をファイルに保存できます。

デフォルトでは、トレースバッファはあらゆる実行サイクルを収集し、最後の32768サイクルを保存します。また、指定されたサイクルだけを表示するフィルタ機能を設定できます。

5.5.1 [Filter]

フィルタを定義するには、Traceウインドウのポップアップメニューより [Filter...] を選択してください。

5.5.2 [Find]

トレースバッファの特定のトレースデータを検索するには、ポップアップメニューより [Find...] を選択してください。見つけたいトレースを指定するため、同じダイアログボックスが現われます。

5.5.3 [Cycle]

フィルタとして特定のサイクルを指定するには、[Type] を [Cycle] に設定し、[Cycle] ボックスにサイクル番号を入力してください。

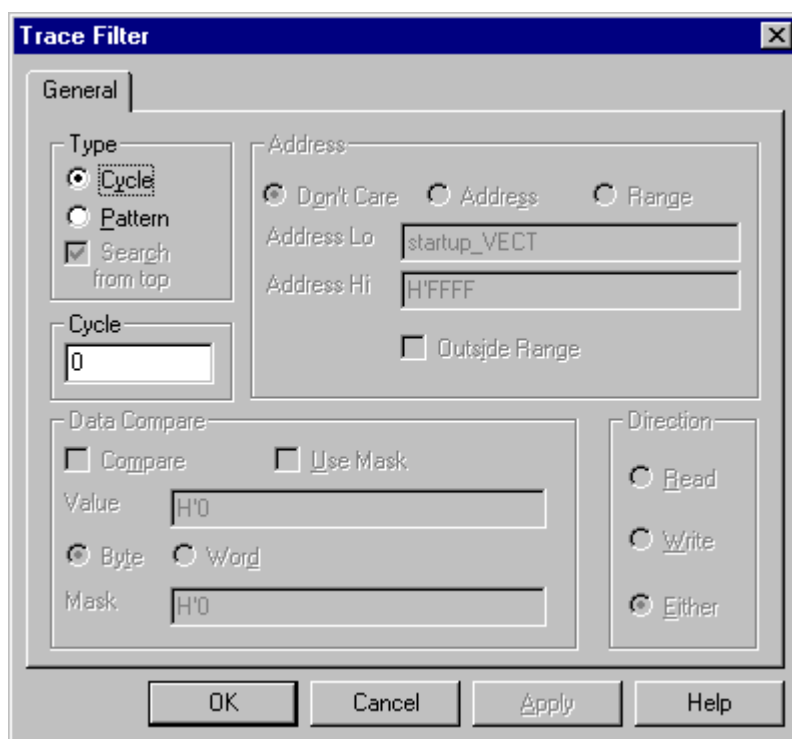


図 5-14 Trace Filterダイアログボックス (General) (1)

5.5.4 [Pattern]

フィルタパターンを入力するには、[Type]を[Pattern]に設定し、必要な値を指定してください。

Trace Filterダイアログボックスが3つのオプションパネルを示しますので、[General]、[Bus / Area]、および[Signals]のうちどのサイクルを表示するかを指定できます。

5.5.5 [General]

[General]パネルによって、表示されるサイクルのアドレスとデータのアクセス条件を設定できます。

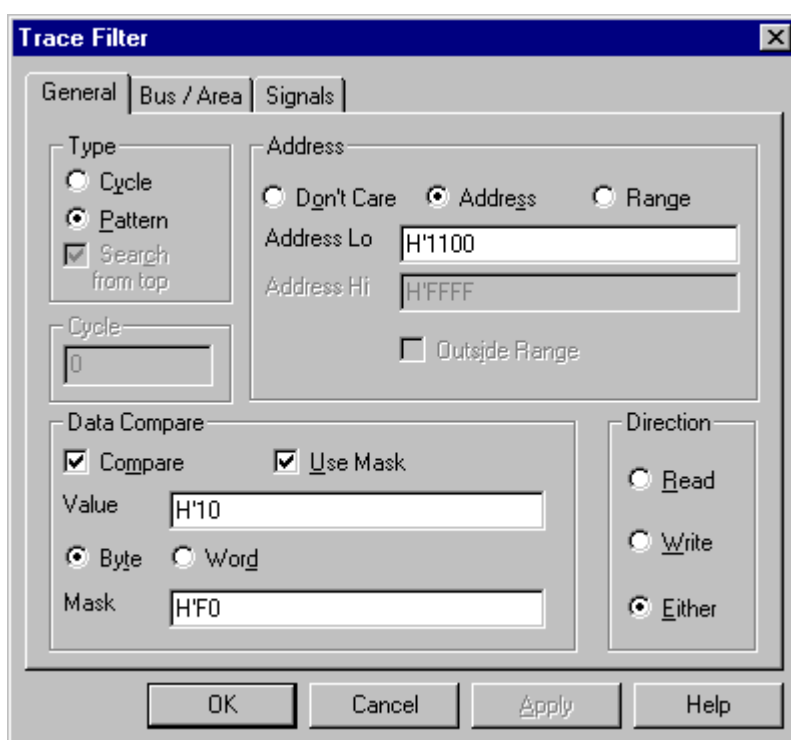


図 5-15 Trace Filterダイアログボックス (General) (2)

5.5.6 [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。

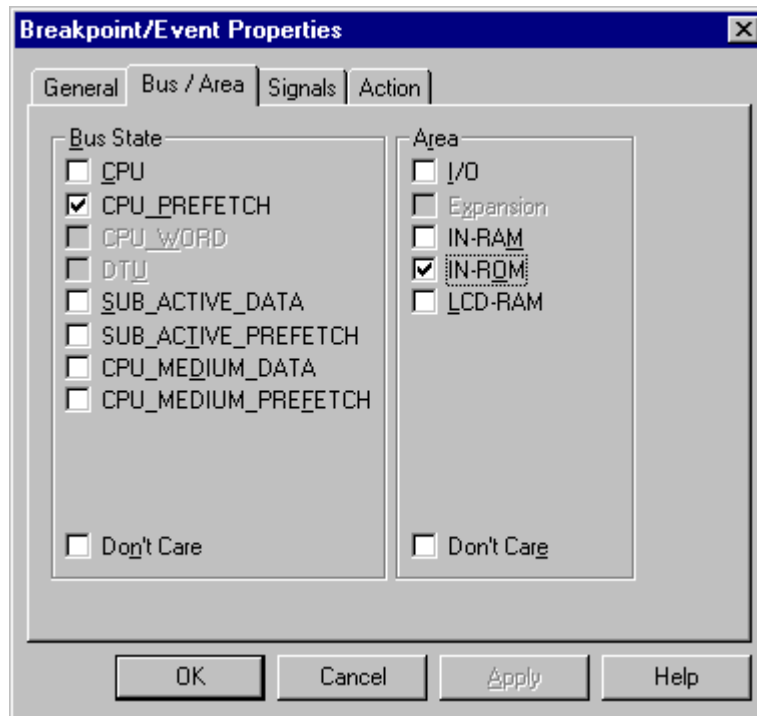


図 5-16 Trace Filterダイアログボックス (Bus / Area)

5.5.7 [Signals]

[Signals] パネルでは、フィルタする外部信号を指定します。

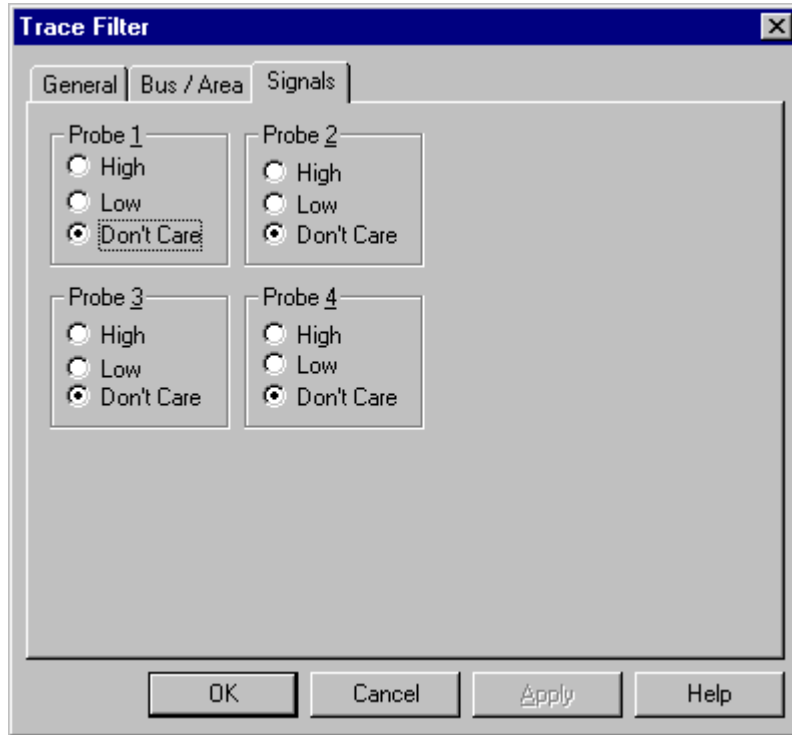


図 5-17 Trace Filterダイアログボックス (Signals)

5.6 トレース制御

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ためにセットアップされます。トレース制御を指定するには、Traceウインドウのポップアップメニューより [Acquisition...] を選択してください。

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。

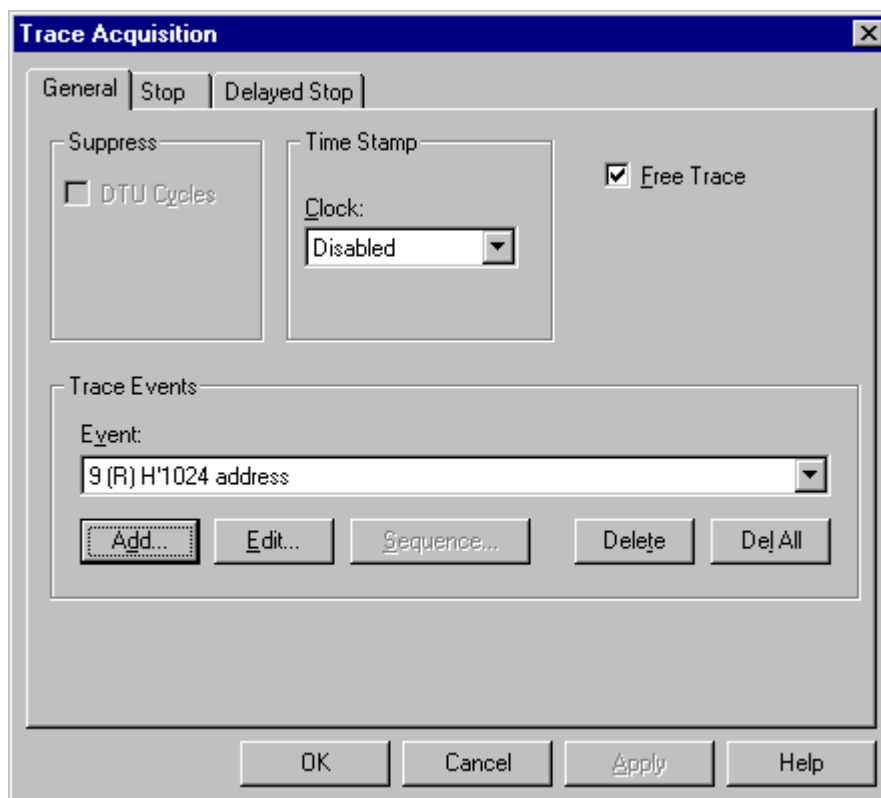


図 5-18 Trace Acquisitionダイアログボックス (General)

5.6.1 [General]

[**Suppress**] ボックスの [**DTU Cycles**] チェックボックスは、無効となります。

[**Time Stamp**] でクロックを指定することで、MCUの動作サイクルに対応した実行時間をトレースバッファに表示させることができます。ただし、実行時間表示を指定した場合は、以下の情報は表示されません。

・Area, ・Status, ・Probes

また、乗除算命令の実行も正しく表示されません。

[**Free Trace**] チェックボックスをチェックして、すべてのトレース制御条件を禁止してください。これは、条件を削除することなくトレース制御を一時的に禁止します。[**Free Trace**] をチェックすると、トレースストップ、トレースディレイストップおよび [**Suppress**] セクションで指定された条件を除くすべてのバスサイクルを取得できます。

[**General**] パネルの [**Trace Events**] セクションによって、トレース制御のイベント、およびイベントシーケンスを定義できます。

[**Event**] ドロップダウンリストボックスが、現在設定されているすべてのイベントを表示します。

新しいイベントを加えるには、[**Add...**] をクリックし、Breakpoint/Event Propertiesダイアログボックスにイベントの内容を入力してください。利用できるオプションの詳細については、「5.3 イベント検出システム」を参照してください。

イベントを編集するには、[**Event**] ドロップダウンリストボックス中のイベントを選択し、[**Edit...**] をクリックしてください。

イベントのシーケンスを設定するには、[**Sequence...**] をクリックしてください。このオプションは、一つ以上のイベントが設定され、[**Enable Sequencing**] が選択されているときのみ使えます。

イベントを削除するには、[**Event**] ドロップダウンリストボックス中のイベントを選択し、[**Delete**] をクリックしてください。また、すべてのトレースイベントを削除するには、[**Del All**] をクリックしてください。

5.6.2 [Stop]

指定したイベントが発生した時にトレース取得を停止できます。

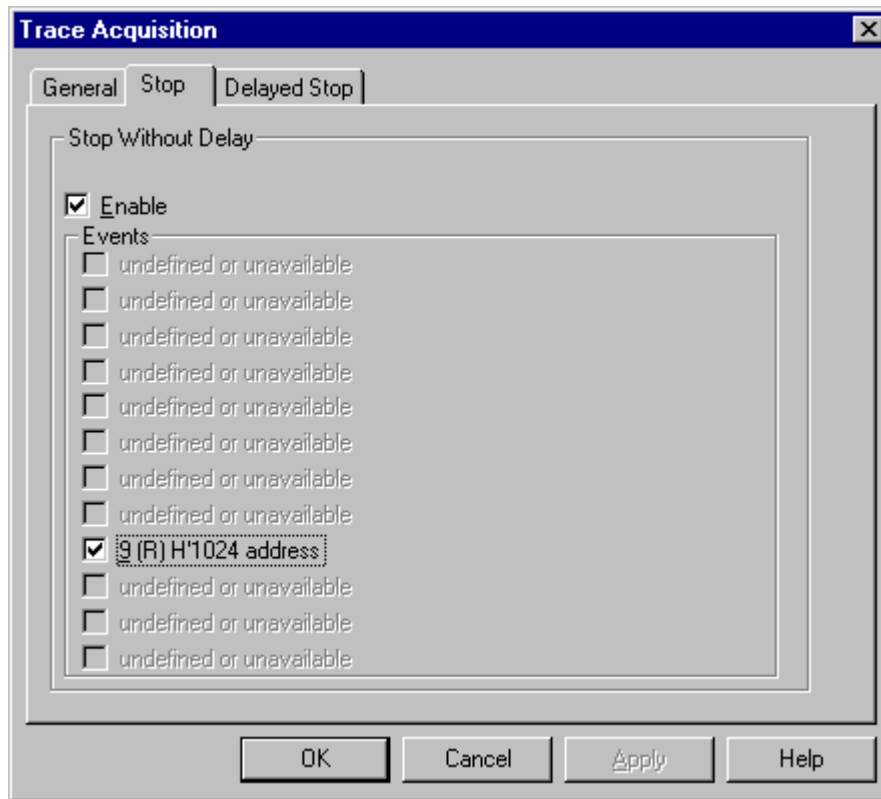


図 5-19 Trace Acquisitionダイアログボックス (Stop)

5.6.3 [Delayed stop]

指定したイベントが発生した後、指定されたサイクル数後にトレース制御を停止できます。

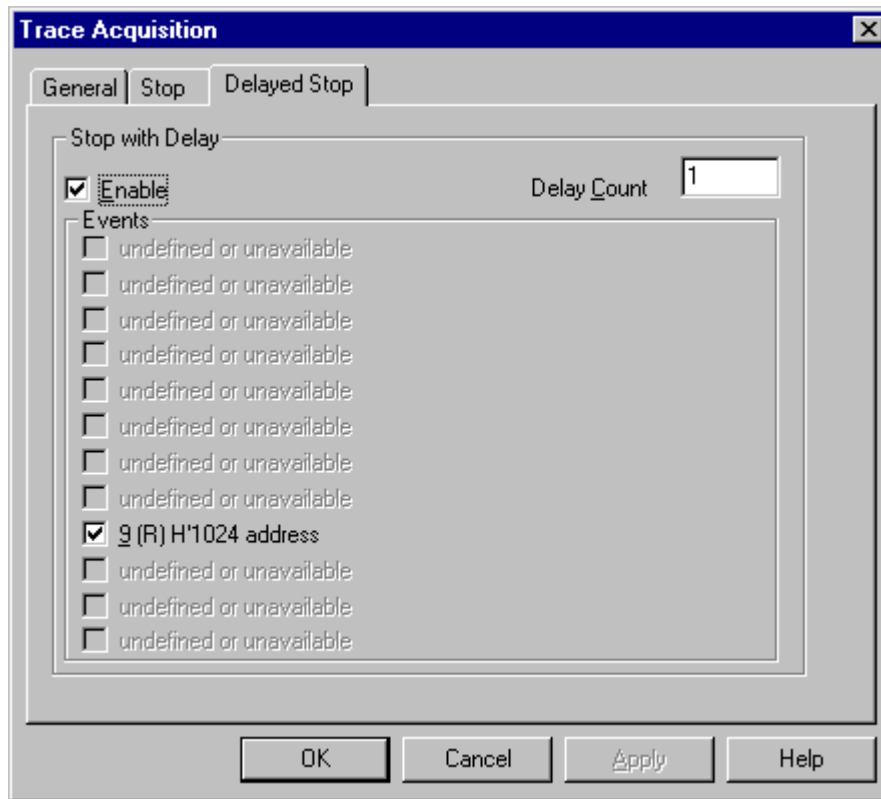


図 5-20 Trace Acquisitionダイアログボックス (Delayed Stop)

5.7 コマンドライン

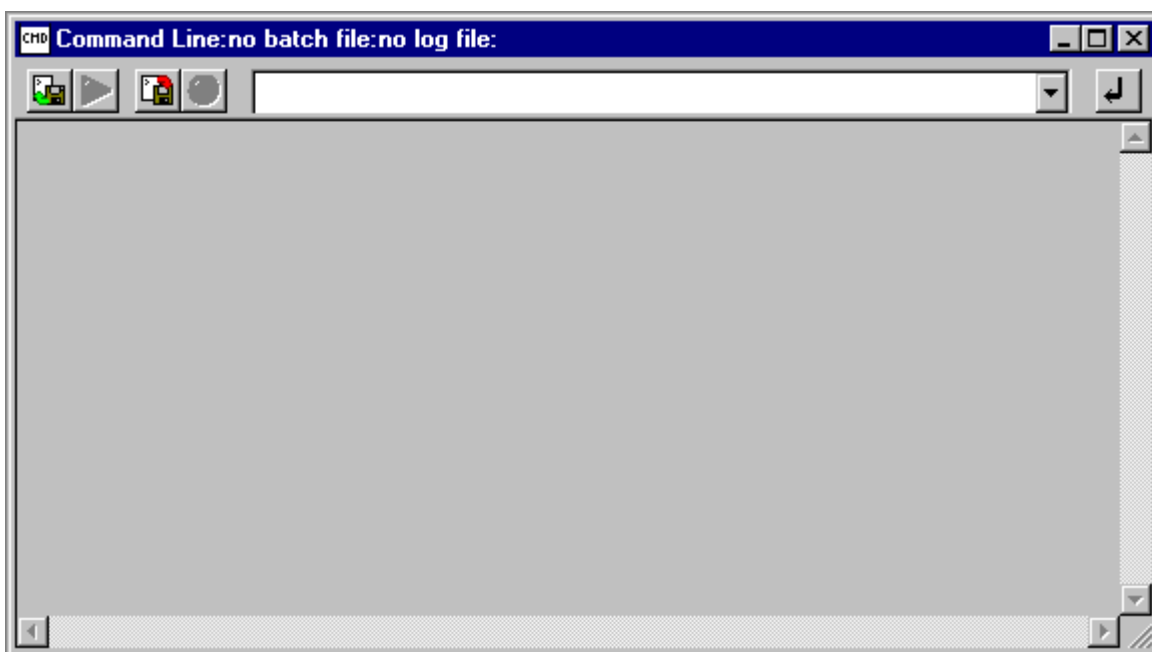


図 5-21 Command Lineウィンドウ

効率よくデバッグを行うために、Command Lineウィンドウを使用してコマンドの入力と実行ができます。Command Lineウィンドウを表示するには、[View]メニューから[Command Line]を選択してください。

MCU専用のコマンドライン機能の詳細については、「6 コマンドライン機能」を参照してください。

6 コマンドライン機能

本章では、MCU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_RANGE	AR		-	性能分析範囲の設定と表示
ANALYSIS_RANGE_DELETE	AD		-	性能分析範囲の解除
ANALYSIS	AN		-	性能分析機能の有効化 / 無効化
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	-	6.1	ブレークポイント / イベントの設定
BREAKPOINT_CLEAR EVENT_CLEAR	BC EC	-	6.2	ブレークポイント / イベントの解除
BREAKPOINT_DISPLAY EVENT_DISPLAY	BD ED	-	6.3	ブレークポイント / イベントの表示
BREAKPOINT_ENABLE EVENT_ENABLE	BE EE	-	6.4	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	-	6.5	シーケンスの定義および解除
CLOCK	CK	-	6.6	エミュレータのCPUクロック時間の設定
DEVICE_TYPE	DE	-	6.7	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
ERASE	ER		-	コマンドウィンドウの内容のクリア
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		-	ユーザープログラムの実行
GO_RESET	GR		-	リセットベクタからのユーザープログラムの実行
GO_TILL	GT		-	テンポラリブレークポイントまでのユーザープログラムの実行
HALT	HA		-	ユーザープログラムの停止
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

表 6-1 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
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エ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	-	6.8	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	-	6.9	CPU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	メモリ関連ウィンドウの更新
REGISTER_DISPLAY	RD		-	CPUレジスタ値の表示
REGISTER_SET	RS		-	CPUレジスタ値の設定
RESET	RE		-	CPU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_OVER	SO		-	ステップオーバー実行
STEP_RATE	SR		-	複数ステップ実行
STEP_OUT	SP		-	PC位置の関数を終了するまでのステップ実行
SUBMIT	SU		-	エ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	-	6.10	エミュレータハードウェアのテスト
TIMER	TI	-	6.11	実行時間測定タイマ分解能の表示、設定
TRACE	TR		-	トレース情報の表示
TRACE_ACQUISITION	TA	-	6.12	トレース取得情報の設定と表示
TRACE_COMPARE	TC	-	6.13	トレース情報の比較
TRACE_SAVE	TV	-	6.14	トレース情報の保存
TRACE_SEARCH	TS	-	6.15	トレース情報の検索
USER_SIGNALS	US	-	6.16	ユーザーシグナル情報の有効化 / 無効化

6.1 BREAKPOINT / EVENT

省略形： BP, EN

ブレークポイントを設定します。ブレークポイントには以下の3種類があり、それぞれ別のフォーマットが用意されています。

- ・ 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

6.1.1 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*

プログラムブレークポイントはアドレスを指定します。

6.1.2 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 ! sadata ! sapre! cpumdata ! cpumpre)+
<areaopts> = area (io!iram!irom!lcdram)+
<actionopts> = action (trace ! none ! break!(timer (start!stop)))+
<channelno> = 1..12

アクセスブレークポイントは、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。

6.1.3 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、アクセスブレークポイント(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。

6.1.4 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 | sadata | sapre| cpumdata | cpumpre) +

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。

表 6-2 MCUバスステータス

イベント	バスステータス
cpu	CPUデータアクセスサイクル(高速モード)
cpupre	CPU命令プリフェッチサイクル(高速モード)
sadata	サブアクティブデータアクセスサイクル
sapre	サブアクティブ命令プリフェッチサイクル
cpumdata	CPUデータアクセスサイクル(中速モード)
cpumpre	CPU命令プリフェッチサイクル(中速モード)

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 | lcdram)+

busオプションと同様に、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または内部RAMがアクセスされている場合にイベントが成立します。

lcdramはMCUのLCDRAM領域のアクセスを示します。

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り当てます（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>

イベントパ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。チャンネル1~8はイベント検出用、チャンネル9~12は範囲検出用チャンネルです。

使用例

en access 100	アクセスブレークポイントをアドレス100に設定
bp p 110	プログラムブレークポイントをアドレス110に設定
en access 100 data 55 byte	アクセスブレークポイントをアドレス100、データ55に設定
bp range 12 45	範囲ブレークポイントをアドレス12からアドレス45の範囲に設定
bp range outside 60 89	範囲ブレークポイントを、アドレス60からアドレス89以外の範囲がアクセスされたときにブレークするように設定
bp a 200 read	アクセスブレークポイントをアドレス200のリードサイクルに設定
bp a 500 write	アクセスブレークポイントをアドレス500のライトサイクルに設定
bp a 100 read channel 8	アクセスブレークポイントをアドレス100のリードサイクルに設定。チャンネル8を指定しているため、イベントが発生した時に外部プローブからトリガ信号が出力されます。

6.2 BREAKPOINT_CLEAR / EVENT_CLEAR

省略形： BC, EC

このコマンドは登録済みのブレークポイントを削除します。

表 6-3 ブレークポイント削除の指定

種類 (キーワード)	説明 (ブレークポイントの種類)
program <address>	指定したプログラムブレークポイントの削除
access <address> <options>	指定したアクセスブレークポイントの削除
range <address> <options>	指定した範囲ブレークポイントの削除
all	全ブレークポイントの削除
all trace	全トレースイベントの削除
channel 1..12	指定したチャンネル番号のイベントの削除

<options>は、BREAKPOINT / EVENTコマンドと同様です。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り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（デフォルトはブレークポイントサブシステムからのイベント削除です）。

使用例

bc p 256 アドレス256のプログラムブレークポイントを削除

event_clear chan 5 チャンネル番号を指定してイベントを削除

bc all 全ブレークポイントを削除

6.3 BREAKPOINT_DISPLAY / EVENT_DISPLAY

省略形： BD, ED

現在設定されているブレークポイントと、その有効 / 無効を表示します。トレースイベントは “ trace ” と表示します。

使用例

bd 設定されている全ブレークポイントと、その有効 / 無効状態を表示します。

6.4 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ブレークポイントを有効 / 無効にします。

表 6-4 ブレークポイント有効 / 無効

パラメータ	種類 (キーワード)	説明
1	true false	ブレークポイント有効 ブレークポイント無効
2	all program <address> access <address> <options> range <address1> <address 2> <options> channel 1..12	全てのブレークポイント 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 指定した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BREAKPOINT / EVENT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。

使用例

be true all 全ブレークポイントを有効

be false all 全ブレークポイントを無効

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

be true access 12 アドレス12のアクセスブレークポイントを有効

be false chan 1 チャンネル1のイベント検出を無効

6.5 BREAKPOINT_SEQUENCE / EVENT_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イクポイントイベントに使用可能です。ただし、ブレイ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。

使用例

```
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。

6.6 CLOCK

省略形：CK

システムクロック()およびサブクロック(w)の指定、表示を行ないます。パラメータを省略すると、現在のシステムクロックを表示します。システムクロックを変更すると、E6000エミュレータはリセットされます。MCUのシステムクロック()は、(OSC1,OSC2) 入力、1/2の周波数になります。

表 6-5 クロックパラメータ

パラメータ	種類 (キーワード)	説明 (エミュレーションクロック)
1	05	0.5 MHz内部クロック
	2	2MHz内部クロック
	8	8MHz内部クロック
	t2	ターゲット/2
2	sub 32k	32.768kHz 内部サブクロック(w)
	sub 38k	38.4kHz 内部サブクロック(w)
	sub 307k	307.2kHz 内部サブクロック(w)
	sub t	ターゲットサブクロック

ユーザシステムのクロックは、ユーザシステムのVccが供給されているときだけ選択可能です。

ck 現在のエミュレーションクロックを表示

ck 2 sub 32k システムクロックとして2MHz内部クロック、サブクロックとして32.768kHz内部クロックを指定

本エミュレータでサブクロック(32.768kHzおよび38.4kHz)動作中にブレークが検出された場合、そのクロックにて動作、表示反応してしまうため、サブクロック評価時は、38.4kHzの8倍周波数307.2kHzを選択していただき評価することをお勧めします。

- 【注意】
1. ターゲットシステムクロックはユーザシステムのVccが供給されている場合にのみ選択可能です。
 2. 8MHz内部クロックはH8/3847Rシリーズ、H8/3827Rシリーズでのみ選択可能です。
 3. ターゲットMCU (H8/3887シリーズ、H8/3867シリーズ、H8/3847シリーズ、H8/3827シリーズ、H8/3847Rシリーズ、H8/3827Rシリーズ) は、内部サブクロック(w)として、307.2KHzを選択することはできません。

6.7 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。

使用例

de MCUタイプの表示

de h8/3887 H8/3887を指定

6.8 MAP_SET

省略形： MS

エミュレーションメモリマップを設定します。

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メモリの設定が可能です。

注：リザーブエリア（H'F74D-H'F75F）はInternalに変更しないで下さい。

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。

使用例

```
ms 8000 F73F internal none      8000からF73Fまでを内部のアクセス可能メモリとして設定  
                                します。
```

6.9 MODE

省略形：M0

MCUモードの設定、表示を行ないます。

表 6-6 MCUモードのパラメータ

パラメータ	種類 (キーワード)	説明 (モード)
1	3	3 (シングルチップモード)

MCUではモード3に固定です。

使用例

mode 現在のモードを表示

mode 3 モード3に設定、メモリマップを再設定

6.10 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）

使用例

te エミュレータテストを実行します。

6.11 TIMER

省略形： TI

実行時間測定タイマの分解能の表示、変更を行いません。これにより、実行時間測定およびイベント間実行時間測定のタイマの分解能が設定できます。

表 6-7 タイマコマンド

コマンド	説明
ti	タイマ分解能の表示
ti <timer resolution>	タイマ分解能の設定

タイマ分解能： 20ns、125ns、250ns、500ns、1 μ s、2 μ s、4 μ s、8 μ s、16 μ s

使用例

ti 20 タイマ分解能を20nsに設定

ti 250ns タイマ分解能を250nsに設定

ti 8 タイマ分解能を8 μ sに設定

ti 16us タイマ分解能を16 μ sに設定

6.12 TRACE_ACQUISITION

省略形： TA

トレース取得オプションの設定、表示を行いません。

フォーマット

```
TA [<suppress>] [<freetrace>] [<timestamp>] [<stop>] [<stopdelay>]
   [<range>][<default>]
```

<suppress> = suppress dtu (true|false) (MCUでは使用できません。)

<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

<suppress>オプションは、H8/388Rでは使用できません。

使用例

ta	全てのトレース取得オプションを表示します。
ta stop event 1 2	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。
ta stopdelay event 1 2 count 100	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。
ta timestamp 500ns	トレースのタイムスタンプ機能を有効にし、タイムスタンプ機能の分解能を500nSに設定します。
ta range 2 event 4 5 cyclic	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。

6.13 TRACE_COMPARE

省略形 : TC

[**trace_save**] で保存したトレースファイルと現在のトレース結果を比較します。

フォーマット

trace_compare <filename>

6.14 TRACE_SAVE

省略形： TV

トレースした情報を、バイナリ形式のファイルに保存します。保存したデータは、別のトレース結果と [**trace_compare**] コマンドで比較することができます。

フォーマット

`trace_save <filename>`

6.15 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 | sadata | sapre! cpumdata! cpumpre )+
<areaopts>    = area ( io | iram | irom | lcdram )+
<directionopts> = dir (read | write | either)
<timestampopts> = time <start> [ to <stop>]
                 <start> and <stop> should be in format 0s 000ms 000us 000ns
<fromopts>    = from <record>
```

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。

使用例

```
ts address 104 data 55aa w   アドレスが104番地、データが55aaのワードアクセスのトレースサイクルを検索する
```

```
ts area irom                ROMエリアアクセスのトレースサイクルを検索する
```

6.16 USER_SIGNALS

省略形： US

ユーザ信号 (Reset) の入力を有効 / 無効にします。パラメータを省略すると、Reset信号の有効 / 無効状態を表示します。

表 6-8 ユーザ信号コマンド

種類	説明
us	ユーザ信号状態を表示
us enable reset	指定された信号を有効
us disable reset	指定された信号を無効

6.17 REFRESH

省略形：RF

メモリ関連ウィンドウを更新します。

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時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を接続しないでください。
 - ・ E6000エミュレータ (HS388REPI60H)
 - ・ PC
 - ・ E6000 PCインタフェースボード (本マニュアルでは、以下のいずれかを指します。PCのインタフェース仕様に合わせて以下のインタフェースボードのいずれか一枚をご用意ください。)
 - ISAバスインタフェースボード (HS6000EI101H)
 - PCIバスインタフェースボード (HS6000EIC01HまたはHS6000EIC02H)
 - PCMCIAインタフェースカード (HS6000EIP01H)
- (2) PCにE6000PC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E6000エミュレータ本体に接続してください。
- (4) E6000エミュレータ本体に、付属のACアダプタを接続してください。
- (5) PCを起動し、DOSプロンプトのコマンド入力待ち状態にしてください。
- (6) E6000エミュレータ本体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 (HS388REPI60SR)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。

- (1) ISAバスインタフェースボード (HS6000EI101H)
>TM388R -ISA (RET)
- (2) PCIバスインタフェースボード (HS6000EIC01HまたはHS6000EIC02H)
>TM388R -PCI (RET)
- (3) PCMCIAインタフェースカード (HS6000EIP01H)
>TM388R -PCCD (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¥TM388R -ISA (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、>TM388R -ISA -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.11までのテストを繰り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-R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。

テストが実行されているときに表示されるメッセージとテスト内容は次のようになります。テストはNo.1からNo.11までです (PC 166MHz, Windows®95 PCMCIAインタフェースカード使用時にテスト時間は約6分です)。

<p>E6000 H8/3880(16MHZ) EMULATION BOARD Tests V1.1 Hitachi Ltd (1999)</p>	<p>テストプログラムのスタートメッセージです。x.xはバージョン番号です。</p>
<p>Searching for interface cardOK</p>	<p>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、ISAバスインタフェースボードを接続している場合は、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 (PCI, PCMCIAではアドレスは表示しません)</p>
<p>Checking emulator is connectedOK</p>	<p>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。</p>
<p>Emulator Board Information: Main Board ID H'1 Emulation Board ID H'1a</p>	<p>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1aを示します。</p>
<p>SIMM No SIMM module inserted</p>	<p>オプションのメモリボードの有無を示します。</p>
<p>01) Testing Rgister : IDR RegisterOK BOC control registerOK MODERE registerOK MODERO registerOK</p>	<p>E6000エミュレータ上のレジスタのチェック結果(正常終了)を示します。</p>
<p>02) Testing Dual-Port RAM : Decode TestOK Marching TestOK</p>	<p>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。</p>

- 03) Testing Firmware RAM :
- Decode Test. page range H'700 - H'71fOK 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。
- Marching Test. page range H'700 - H'71fOK E6000エミュレータ上の Firm RAMのマーチングテストチェック結果(正常終了)を示します。
- 04) Testing Trace RAM :
- Decode Test. page range H'000 - H'04fOK 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。
- Marching Test. page range H'000 - H'04fOK E6000エミュレータ上の Trace RAMのマーチングテストチェック結果(正常終了)を示します。
- 05) Testing Mapping RAM :
- Decode Test. page range H'200 - H'21fOK E6000エミュレータ上の Mapping RAMのデコードテストチェック結果(正常終了)を示します。
- Marching Test. page range H'200 - H'21fOK E6000エミュレータ上の Mapping RAMのマーチングテストチェック結果(正常終了)を示します。
- 06) Testing Emulation RAM :
- Decode TestOK 内蔵ROMおよび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。
- Marching TestOK

07) Testing STEP Operation :		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。
Single Step Operation	OK	
Step Into Operation	OK	
08) 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	
09) Testing Emulation RAM Hardware Break :		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。
GRD Break	OK	
WPT Break	OK	
IN Rom	OK	
EML RAM Write	OK	
Access Error Break	OK	
10) Testing Go Reset	OK	内蔵ROM領域に対する書き込み禁止制御回路のチェック結果(正常終了)を示します。
11) Tsetting Double Stack, I/O Stack Test :		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。
Double Stack	OK	
I/O Stack	OK	
0 total errors		エラー発生数の合計を示します。
Tests passed, emulator functioning correctly		テストプログラムにより正常動作が確認されたことを示します。

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(HS388REPI60H) 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(HS388REPI60SR) 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： H8/ _____
- 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

3) エラー発生状況

a~c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

(テスト番号 : _____ 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 : _____、ピン番号 : _____

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 H8/3887 シリーズ、H8/3867 シリーズ、H8/3847 シリーズ、
H8/3827 シリーズ、H8/3847R シリーズ、H8/3827R シリーズ用エミュレータ
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