

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日

ルネサスエレクトロニクス株式会社

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、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。

H8/3064B E6000 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

H8 ファミリ / H8/300H シリーズ

HS3064BEPI61HJ-U2

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注意ください。

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 テクノロジー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 テクノロジー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 テクノロジー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 テクノロジーは、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 テクノロジー半導体製品のご購入に当たりましては、事前にルネサス テクノロジー、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 テクノロジーホームページ (<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 テクノロジー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 テクノロジーは、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 テクノロジー、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 テクノロジー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、その他お気づきの点がございましたらルネサス テクノロジー、ルネサス販売または特約店までご照会ください。

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、株式会社ルネサス テクノロジ（以下、「ルネサス」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）オプションメモリボード、（５）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、ルネサスマイクロコンピュータ（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：

ルネサスは、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。

エミュレータを使う人は：

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：

ルネサスは、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）ルネサス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、ルネサス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ルネサス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：

このユーザー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、すべての権利はルネサスに帰属しています。このユーザー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、ルネサス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：

このユーザーズマニュアルの一部の図は、実物と異なっていることがあります。

予測できる危険の限界：

ルネサスは、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、このユーザー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、ルネサスまたはルネサス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。

はじめに

E6000 エミュレータは、ルネサス 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 ファミリ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。

High-performance Embedded Workshop は、ルネサスのマイクロコンピュータ用に、C/C++言語およびアセンブリ言語で書いたアプリケーションの開発およびデバッグを簡単に行うためのグラフィカルユーザインタフェースを提供します。アプリケーションを実行するエミュレータのアクセス、計測、および変更に関して、High-performance Embedded Workshop は高機能でしかも直観的な手段を提供することを目的としています。

High-performance Embedded Workshop は、ルネサスマイクロコンピュータの組み込み用アプリケーションの開発を強力にサポートするツールです。おもな特徴をまとめると次のようになります。

- 使い勝手の良いインタフェースを活用したコンパイラ、アセンブラ、リンケージエディタなどのオプションが設定できるカスタマイズ可能なプロジェクトビルドシステム。
- プログラムを読みやすくするシンタックス色付け機能を持つ統合化テキストエディタ。
- ユーザ独自のツールを実行するための環境設定。
- 同一アプリケーション内でのビルドおよびデバッグを可能にする統合化デバッグ。
- バージョン管理サポート。

High-performance Embedded Workshop は 2 つの目的で設計されています。一つはユーザに強力な開発ツールを提供すること、そしてもう一つは、それらのツール類を統合して使いやすくすることです。

このマニュアルについて

本マニュアルでは以下の内容を説明しています。

エミュレータデバッグ編 使用前の準備、E6000 エミュレータの機能、デバッグ機能、チュートリアル、E6000 エミュレータのハード仕様、E6000 エミュレータのソフトウェア仕様

High-performance Embedded Workshop の基本的な使い方に関する情報、High-performance Embedded Workshop 環境のカスタマイズ、High-performance Embedded Workshop のビルド機能、および各 High-performance Embedded Workshop 製品で共通なデバッグ機能については、High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

このマニュアルでは C/C++ 言語、アセンブリ言語の書き方や、オペレーティングシステムの使い方、個々のデバイスに適したプログラムの書き方などについては説明していません。それらについては、各々のマニュアルを参照してください。

Microsoft®, MS-DOS, Windows®, Windows NT®は米国 Microsoft 社の米国およびその他の国における登録商標です。

Visual SourceSafe は Microsoft 社の米国およびその他の国における商標です。

IBM は International Business Machines Corporation の登録商標です。

その他、記載されている製品名は各社の商標または登録商標です。

このマニュアルの記号

このマニュアルで使われている記号の意味を説明します。

表 1: 記号一覧

記号	意味
[Menu->Menu Option]	太字と '>' はメニューオプションを示します (例 [File->Save As...])
FILENAME.C	大文字の名前はファイル名を示します
“文字列の入力”	下線は入力する文字列を示します (“ ” を省く)
Key + Key	キー入力を示します。例えば、CTRL+N キーでは CTRL キーと N キーを同時に押します
☞ (「操作方法」マーク)	このマークが左端にあるとき、その右の文章は何かの操作方法を示します
バックスラッシュ文字\	文中および図中で使用しているファイルパス名の文字列中のバックスラッシュ文字は日本語 Windows®では円記号として表示されます。

梱包品の確認

梱包を解いた後、納入品明細書に記入されている梱包品がそろっているか確認してください。確認した結果、梱包品に不足がありましたら、当エミュレータ購入元の営業担当までご連絡ください。

目次

エミュレータデバッグ編	1
1 はじめに	3
1.1 特長	3
1.2 使用上の注意事項	4
1.3 使用環境条件	5
1.4 外形寸法と質量	6
2 使用前の準備	7
2.1 E6000 エミュレータ使用フローチャート	7
2.2 エミュレータソフトウェアのインストール	7
2.3 ユーザシステムへの接続	8
2.3.1 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	8
2.3.2 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	9
2.3.3 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	9
2.4 電源供給	10
2.4.1 AC 電源アダプタ	10
2.4.2 極性	10
2.4.3 電源モニタ回路	10
2.5 オプションメモリボード	10
2.5.1 オプションメモリボードの構成	10
2.6 ハードウェアインタフェース	10
2.6.1 信号保護	10
2.6.2 ユーザインタフェース回路	11
2.6.3 クロック発振器	11
2.6.4 外部プローブ 1 (EXT1) / トリガ出力	11
2.6.5 外部プローブ 2 (EXT2) / トリガ出力	11
2.6.6 電源フォロワ回路	12
2.7 システムチェック	13
2.8 通信不良	18
2.9 その他の起動方法	18
2.10 アンインストール	18
3 E6000 エミュレータ機能	19
3.1 デバッグの特長	19
3.1.1 ブレークポイント	19
3.1.2 トレース	19
3.1.3 実行時間測定	19
3.1.4 パフォーマンスアナリシス	19
3.1.5 バスモニタ	19
3.2 イベント検出システム (CES: Complex Event System)	20
3.2.1 イベントチャンネル	20
3.2.2 範囲チャンネル	20
3.2.3 ブレーク	20
3.2.4 イベント間実行時間測定	20
3.3 ハードウェアの特長	21
3.3.1 メモリ	21
3.3.2 エミュレーションクロック	21
3.3.3 外部プローブ	21

3.4	スタックトレース機能	21
3.5	オンラインヘルプ	21
4	デバッグの準備をする	23
4.1	High-performance Embedded Workshop の起動方法	23
4.1.1	新規にワークスペースを作成する場合(ツールチェイン未使用)	24
4.1.2	新規にワークスペースを作成する場合(ツールチェイン使用)	26
4.1.3	既存のワークスペースを指定する場合	29
4.2	エミュレータの接続	30
4.3	エミュレータの再接続	32
4.4	エミュレータの終了	32
5	デバッグ	33
5.1	エミュレーション環境を設定する	33
5.1.1	Configuration Properties ダイアログボックスを開く	33
5.1.2	MCU 一覧にない MCU を設定する	35
5.1.3	接続するインタフェースを選択する	36
5.1.4	Memory Mapping ダイアログボックスを開く	37
5.1.5	メモリマップ設定を変更する	38
5.2	プログラムをダウンロードする	39
5.2.1	プログラムをダウンロードする	39
5.2.2	ソースコードを表示する	40
5.2.3	アセンブリ言語コードを表示する	42
5.2.4	アセンブリ言語コードを修正する	43
5.2.5	特定のアドレスを見る	43
5.2.6	現在のプログラムカウンタアドレスを見る	43
5.3	現在の状態を表示する	44
5.4	エミュレータの情報を定期的に読み出し表示する	45
5.4.1	[拡張モニタ]ウィンドウを開く	45
5.4.2	表示項目を選択する	45
5.5	リアルタイムにメモリ内容を表示する	46
5.5.1	[モニタ]ウィンドウを開く	46
5.5.2	モニタの設定内容を変更する	48
5.5.3	モニタの更新を一時的に停止する	48
5.5.4	モニタ設定を削除する	48
5.5.5	変数の内容をモニタする	48
5.5.6	モニタウィンドウを非表示にする	48
5.5.7	[モニタ]ウィンドウを管理する	49
5.6	変数の表示	50
5.6.1	ウォッチウィンドウ	50
5.7	イベントポイントを使用する	52
5.7.1	ソフトウェアブレイクポイントとは	52
5.7.2	イベントポイントとは	52
5.7.3	イベント検出システムとは	52
5.7.4	バス状態およびエリア信号について	53
5.7.5	[イベントポイント]ウィンドウを開く	54
5.7.6	ソフトウェアブレイクポイントを設定する	54
5.7.7	イベントポイントを設定する	56
5.7.8	トリガポイントを設定する	62
5.7.9	イベントポイントの編集	63
5.7.10	イベントポイントの設定内容を変更する	63
5.7.11	イベントポイントを有効にする	63
5.7.12	イベントポイントを無効にする	63
5.7.13	イベントポイントを削除する	63
5.7.14	イベントポイントをすべて削除する	63
5.7.15	イベントポイントのソース行を表示する	63
5.8	トレース情報を見る	64
5.8.1	[Trace]ウィンドウを開く	64

5.8.2	トレース情報を取得する	64
5.8.3	トレース情報取得条件を設定する	65
5.8.4	Trace レコードを検索する	73
5.8.5	トレース情報をクリアする	78
5.8.6	トレース情報をファイルに保存する	78
5.8.7	[エディタ]ウィンドウを表示する	79
5.8.8	ソース表示を整形する	79
5.8.9	トレース情報のスナップショットを取得する	79
5.8.10	トレース情報の取得を一時的に停止する	79
5.8.11	トレース情報の取得を再開する	79
5.8.12	取得したトレース情報から必要なレコードを抽出する	79
5.8.13	タイムスタンプの差を計算する	87
5.8.14	統計情報を解析する	88
5.8.15	取得したトレース情報から関数呼び出し箇所を抽出する	89
5.9	パフォーマンスを測定する	90
5.9.1	パフォーマンス解析ウィンドウを開く	91
5.9.2	実行効率測定条件を設定する	92
5.9.3	実行効率測定のアドレス検出方式および分解能を設定する	97
5.9.4	実行効率測定を開始する	98
5.9.5	測定条件を削除する	98
5.9.6	すべての測定条件を削除する	98
6	チュートリアル	99
6.1	はじめに	99
6.2	High-performance Embedded Workshop の起動	99
6.3	チュートリアルプログラムのダウンロード	100
6.3.1	チュートリアルプログラムをダウンロードする	100
6.3.2	ソースプログラムを表示する	101
6.4	ソフトウェアブレークポイントの設定	102
6.5	レジスタ内容の変更	103
6.6	プログラムの実行	104
6.7	ブレークポイントの確認	106
6.8	シンボルの参照	107
6.9	メモリ内容の確認	108
6.10	変数の参照	109
6.11	ローカル変数の表示	112
6.12	プログラムのステップ実行	113
6.12.1	ステップインコマンドの実行	113
6.12.2	ステップアウトコマンドの実行	114
6.12.3	ステップオーバーコマンドの実行	115
6.13	プログラムの強制ブレーク	116
6.14	MCU のリセット	116
6.15	ブレーク機能	117
6.15.1	ソフトウェアブレーク機能	117
6.15.2	イベントポイントによるブレーク機能	121
6.16	トレース機能	125
6.16.1	トレースの表示 (タイムスタンプ無効時)	126
6.16.2	トレースの表示 (タイムスタンプ有効時)	131
6.16.3	統計	135
6.16.4	関数コール	137
6.17	スタックトレース機能	139
6.18	パフォーマンス測定機能	141
6.18.1	指定範囲内時間測定	141

6.19	モニタ機能.....	144
6.20	さてつぎは?	147
7	本製品固有のハードウェア仕様	149
7.1	H8/3064B E6000 エミュレータ仕様.....	149
7.1.1	サポート範囲.....	149
7.1.2	動作電圧および動作周波数.....	150
7.2	ユーザシステムインタフェース回路.....	151
7.2.1	信号保護.....	151
7.2.2	ユーザインタフェース回路.....	151
7.3	MCU と E6000 エミュレータの相違点.....	154
7.3.1	A/D コンバータ、D/A コンバータ	154
8	本製品固有のソフトウェア仕様	155
8.1	H8/3064B E6000 エミュレータソフトウェア仕様	155
8.1.1	対応ハードウェア.....	155
8.1.2	選択可能プラットフォーム.....	155
8.1.3	Configuration Properties ダイアログボックス(General ページ).....	155
8.1.4	Configuration Properties ダイアログボックス(Custom ページ).....	159
8.1.5	Memory Mapping 機能.....	159
8.1.6	ステータスウィンドウ.....	159
8.1.7	拡張モニタ機能.....	160
8.1.8	バス状態およびエリア信号.....	161
8.1.9	モニタ機能.....	161
8.1.10	トリガポイント.....	161
8.1.11	トレース情報.....	162
8.1.12	トレースレコードの検索.....	163
8.1.13	Trace Filter 機能.....	164
8.2	H8/3064B E6000 エミュレータ使用上の注意事項	165
8.2.1	チュートリアルプログラムの実行環境.....	165
8.2.2	I/O レジスタの相違点.....	165
8.2.3	内蔵 RAM 領域を外部アドレスとして使用	165
8.2.4	フラッシュメモリのサポート	165
8.2.5	オプションメモリの取扱い.....	166
8.2.6	H8/3048F-ONE の相違点.....	166
付録 A	I/O ファイルフォーマット	167
A.1	ファイルフォーマット(ビットフィールド非対応).....	167
A.2	ファイルフォーマット(ビットフィールド対応).....	169
付録 B	メニュー一覧	171
付録 C	コマンドライン一覧	175
付録 D	ハードウェア診断プログラムについて	179
D.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.....	179
D.2	テストプログラムによる故障解析	180
D.3	エラー発生時の処理.....	185

エミュレータデバッグ編

1 はじめに

1.1 特長

本エミュレータには、次のような特長があります。

- (1) ブレークポイント・メモリマップ・パフォーマンス・トレースをダイアログボックス上で設定することができます。
また、下記のような特徴を持ちます。
 - 直観的なユーザインタフェース
 - オンラインヘルプ
 - 共通した表示と操作性
- (2) 各種ホストインタフェースをサポート
ホストコンピュータと接続するためにPCIインタフェース、PC Card (PCMCIA) インタフェース、USBインタフェース、LANインタフェースが使用できます。
- (3) リアルタイムエミュレーション
CPUの最高動作周波数でのリアルタイムエミュレーションができます。
- (4) 優れた操作性を実現
High-performance Embedded Workshopの使用により、マウスなどのポインティングデバイスを用いて、ユーザプログラムのデバッグが可能です。また、High-performance Embedded Workshopを使用して、ロードモジュールファイルを高速にダウンロードできます。
- (5) 充実したデバッグ機能
ブレーク、トレース機能の充実によりデバッグ効率が向上します。ブレークポイント、およびブレーク条件を専用のウィンドウで設定したり、トレース情報をウィンドウに表示できます。さらに、豊富なコマンドライン機能を備えています。
- (6) エミュレーション実行中のメモリアクセス機能
エミュレーション実行中にメモリの内容を参照、変更することができます。

1.2 使用上の注意事項

注意

E6000 エミュレータをお使いになる前に、以下の注意事項を必ず確認してください。
誤った使い方は、E6000 エミュレータ、ユーザプログラムおよびユーザシステムの破壊につながります。

- (1) 製品を梱包箱から取り出し、納入品明細書に示されているものがそろっているか、確認してください。
- (2) 製品に重量物を上積みするなどして、無理な力を加えないでください。
- (3) 次の条件を考慮して、E6000エミュレータを設置してください。
直射日光の当たる場所や、暖房機の近く等、高温となる場所に設置しないでください。
「1.3 使用環境条件」を参照してください。
温度や湿度が極端に変化する場所に設置しないでください。
チリやホコリが多い場所に設置しないでください。
振動が多い場所に設置しないでください。「1.3 使用環境条件」を参照してください。
- (4) 製品に過大な物理的衝撃を与えないでください。
- (5) E6000エミュレータに、指定された電圧、電源周波数以外の電源を供給しないでください。
- (6) 設置場所を移動する場合は、本製品に強い振動、衝撃が加わらないように注意してください。
- (7) ケーブルを接続した後は、接続位置が正しいことを再度確認してください。接続方法については、「2 使用前の準備」を参照してください。
すべてのケーブルを接続し終えてから、接続した各装置へ電源を投入してください。電源の投入順序は「2.7 システムチェック」を参照してください。また、電源が入っているときにケーブルの接続および取り外しをしないでください。

1.3 使用環境条件

注意

E6000 エミュレータを使用する場合、表 1-1に示す条件を守ってください。
この条件を満たさない状態で E6000 エミュレータを使用した場合、E6000 エミュレータ、ユーザプログラムおよびユーザシステムが正常に動作しない場合があります。

表1-1 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	振動	動作時 : 最大 2.45m/s ² 非動作時 : 最大 4.9m/s ² 梱包輸送時 : 最大 14.7m/s ²
4	周囲ガス	腐食性ガスのないこと

1.4 外形寸法と質量

表1-2 外形寸法および質量

項番	項目	仕様
1	外形寸法	219 × 170 × 54 (mm)
2	質量	約 1000 (g)

2 使用前の準備

2.1 E6000 エミュレータ使用フローチャート

E6000 エミュレータを使用するにあたって、梱包を解いたあと下記の手順で準備を行ってください。



警告

準備を行う前に図 2-1中のアミのかかっている参照先を全てよく読んで理解してください。誤った使い方は、E6000 エミュレータ、ユーザプログラムおよびユーザシステムの破壊につながります。

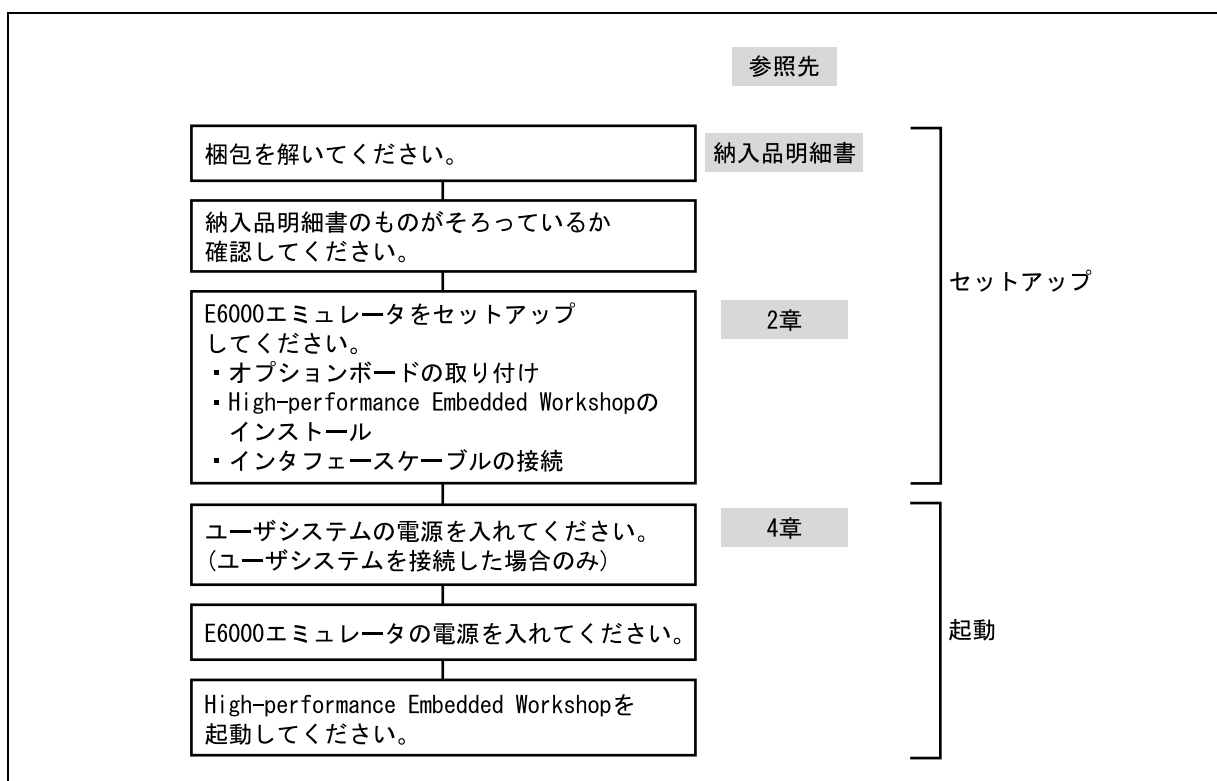


図 2-1 E6000 エミュレータ使用フローチャート

2.2 エミュレータソフトウェアのインストール

エミュレータソフトウェアのインストールにつきましては、同梱の E6000 エミュレータ用セットアップガイドをご参照ください。

2.3 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します。
- 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。
- ケーブル本体部を先端部へ接続する。

これらの手順詳細については、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。

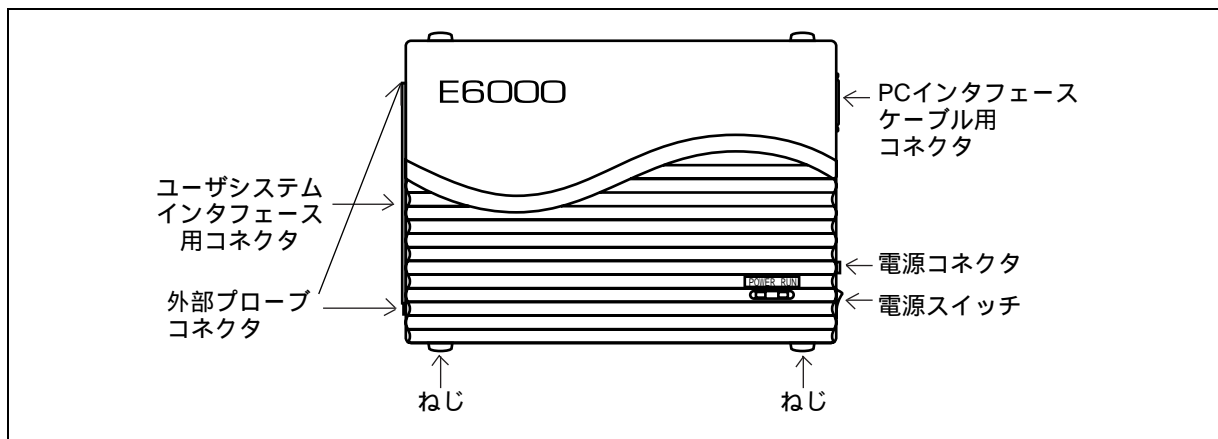


図 2-2 E6000 コネクタの位置

2.3.1 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

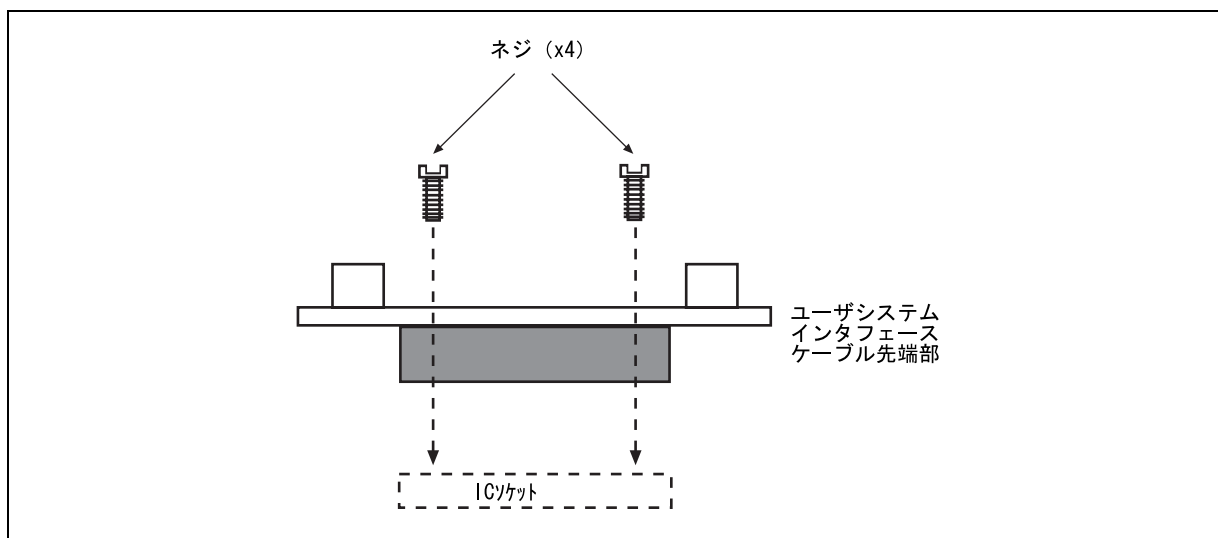


図 2-3 ユーザシステムインタフェースケーブルの接続

- 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】 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 ピンの位置を正しく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。

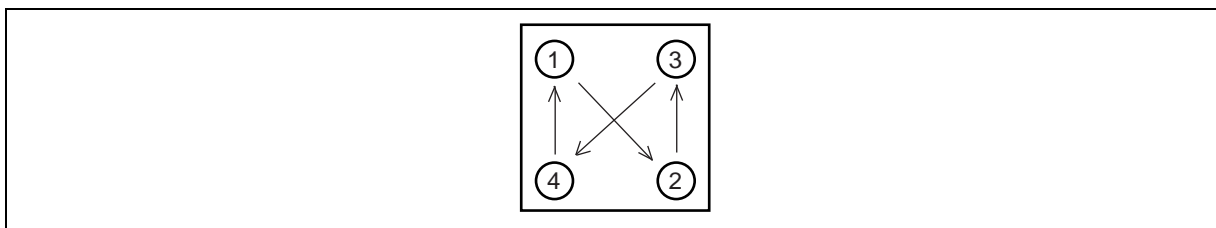


図 2-4 ネジの締めつけ順序

【注】 ネジを締めつ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。

2.3.2 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。

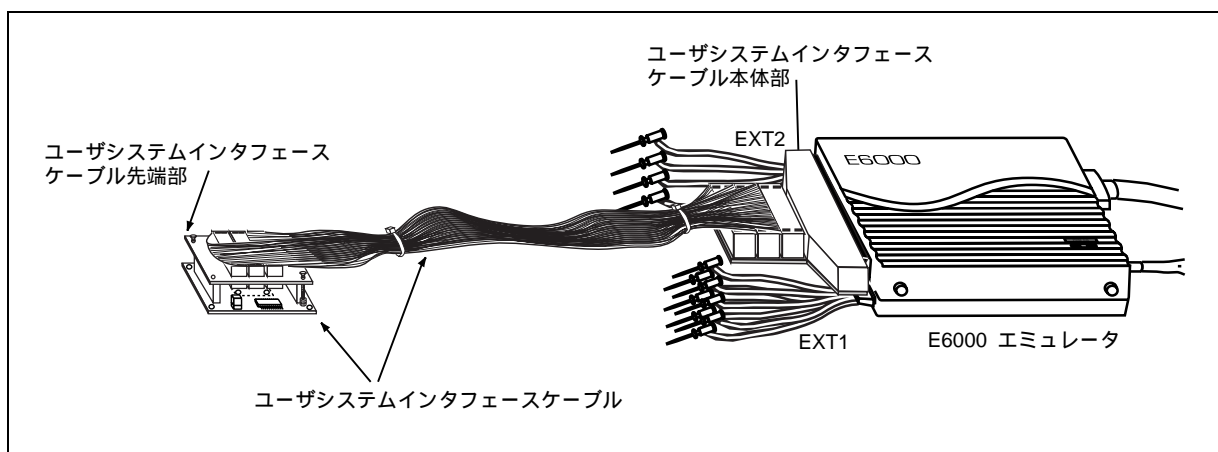


図 2-5 ユーザシステムインタフェースケーブル外観図

2.3.3 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。

2.4 電源供給

2.4.1 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。

2.4.2 極性

以下に電源プラグの極性を示します。

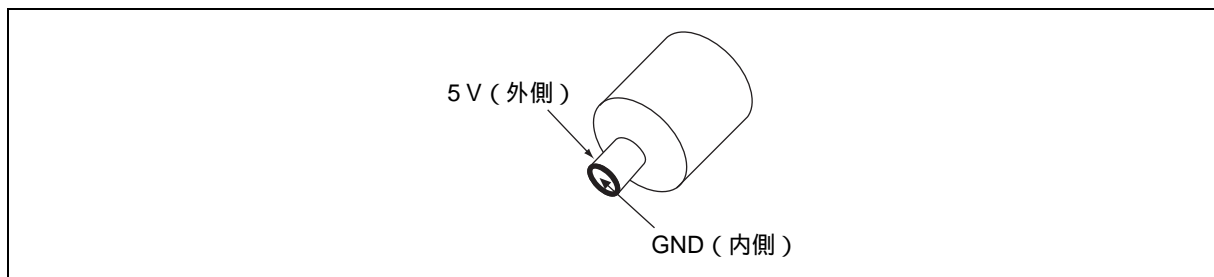


図 2-6 電源プラグ

2.4.3 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。

2.5 オプションメモリボード

E6000 エミュレータに外部領域のエミュレーションメモリとしてオプションメモリボードを接続できます。オプションメモリボードの詳細はオプションメモリボード添付の取扱い説明書を参照してください。オプションメモリボードは 4 つの等しいサイズのバンクを持っており、これらのバンクをユーザエリアに割り当てることができます。

製品によりオプションメモリボードをサポートしていない場合があります。

2.5.1 オプションメモリボードの構成

オプションメモリボードの構成は、マッピング RAM によって制御されます。[ステータス]ウインドウの Memory シートを開くと、インストールされたオプションメモリボードをチェックでき、また Memory Mapping ダイアログボックスより 4 つのバンクを必要なアドレスに再配置できます。

2.6 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、一部の信号を除いてバッファなしに直接エミュレータ上の MCU に接続されています。MCU に入力する前にエミュレータ制御回路が挿入される信号は各エミュレータにより異なります。詳細は「7 本製品固有のハードウェア仕様」を参照してください。

2.6.1 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 Vref には、この保護回路がありません。

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。

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 (AVcc 端子を除く) は、すべて 1 つに接続されています。E6000 エミュレータは、これを監視して、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。

2.6.2 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 8ns 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例を示します。ユーザインタフェースは製品により各々異なります。詳しくは、「7 本製品固有のハードウェア仕様」を参照してください。

2.6.3 クロック発振器

ユーザシステムインタフェースケーブル先端部にはシステムクロック発振回路を実装しています。システムクロック発振回路の詳細については、各ユーザシステムインタフェースケーブル取扱い説明書を参照してください。

2.6.4 外部プローブ 1 (EXT1) / トリガ出力

E6000 エミュレータ筐体側面にある EXT1 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右下) 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。

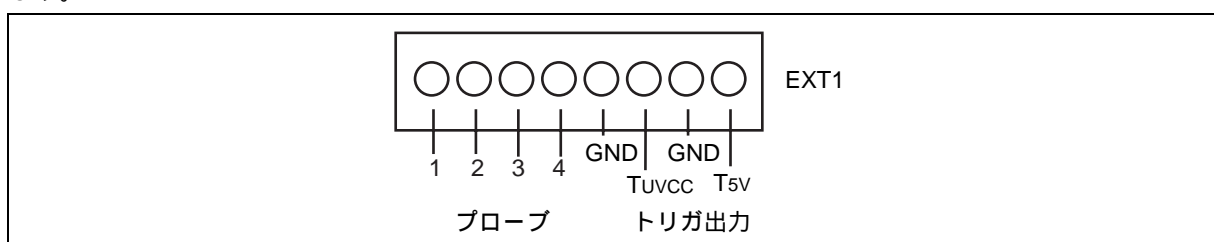


図 2-7 外部プローブ 1 コネクタ

以下に外部プローブ 1 のインタフェース回路を示します。

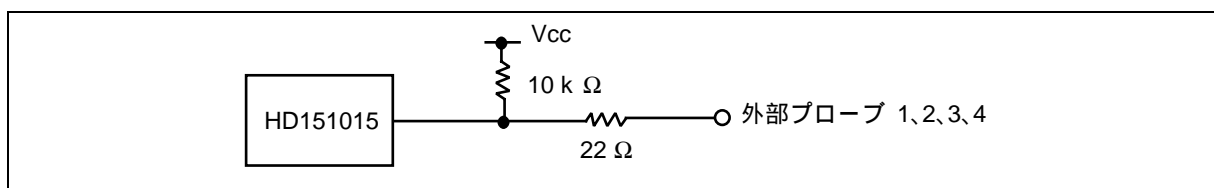


図 2-8 外部プローブ 1 インタフェース回路

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 (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 TU_{vcc} (ユーザシステム電源電圧) レベルの 2 つあります。

2.6.5 外部プローブ 2 (EXT2) / トリガ出力

E6000 エミュレータ筐体側面にある EXT2 のマークが記された 6 ピンコネクタ (ユーザインタフェースコネクタの左下) は、トリガ出力 4 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。

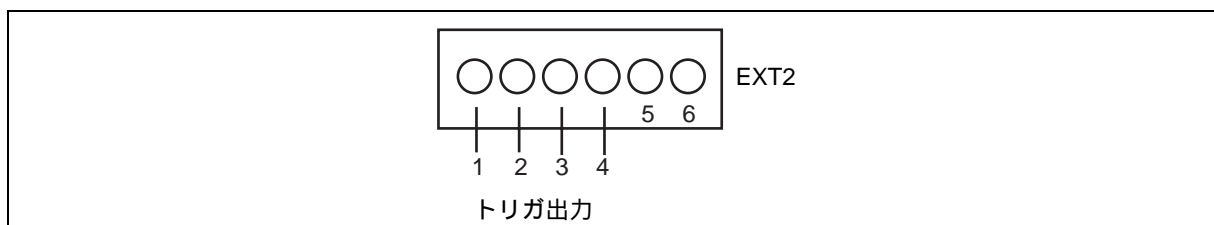


図 2-9 外部プローブ 2 コネクタ

トリガ出力はバスモニタ機能のトリガ設定 (1 ~ 4) の条件が一致した場合にリード / ライトの期間だけ出力されるハイレベル信号です。トリガ出力は V_{cc} (ユーザシステム電源電圧) レベルです。

製品により外部プローブ 2 (EXT2) をサポートしていない場合があります。

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、ユーザシステムインタフェース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い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ユーザシステム Vcc に一致させます。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、[拡張モニタ] ウィンドウの [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。

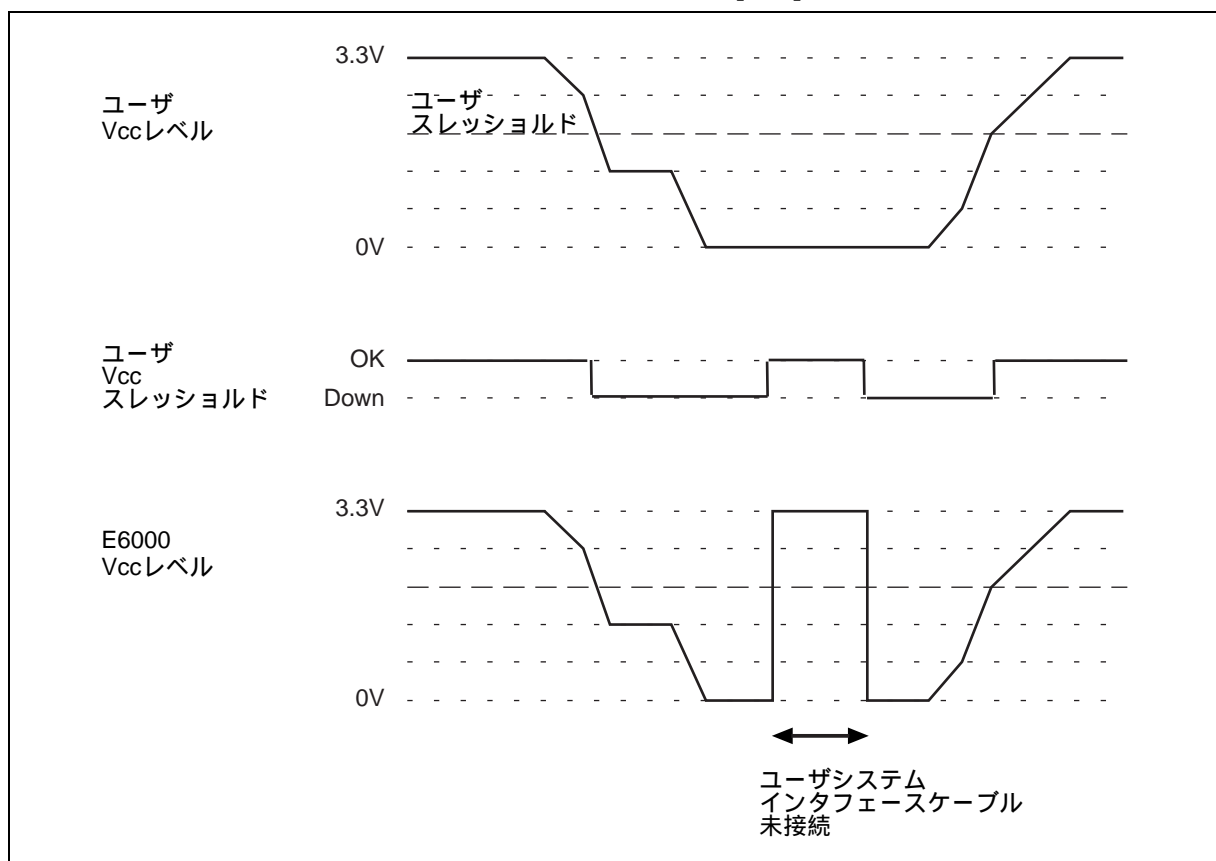


図 2-10 ユーザシステムと E6000 との Vcc の関係 (Vcc=3.3V の例)

2.7 システムチェック

次に、ソフトウェアを実行し、E6000 エミュレータが正しく接続されていることをチェックします。

ここでは、製品に添付のチュートリアル用ワークスペースを使用して起動します。

新規にプロジェクトを作成して起動する方法や、旧バージョンの High-performance Embedded Workshop 用のワークスペースを使用して起動する方法については、「2.9 その他の起動方法」を参照してください。

- (1) ホストコンピュータとE6000エミュレータを接続してください。
- (2) E6000エミュレータのコネクタとユーザインタフェースケーブルを接続します。
- (3) E6000エミュレータの電源を入れてください。
- (4) [スタート]メニューの[プログラム]からHigh-performance Embedded Workshopを起動してください。(図 2-11)



図 2-11 [スタート]メニュー

【留意事項】

[Tools] は、インストール時、“LAN Driver”を選択しなかった場合は表示されません。

2 使用前の準備

- (5) [ようこそ!]ダイアログボックスが表示されます。

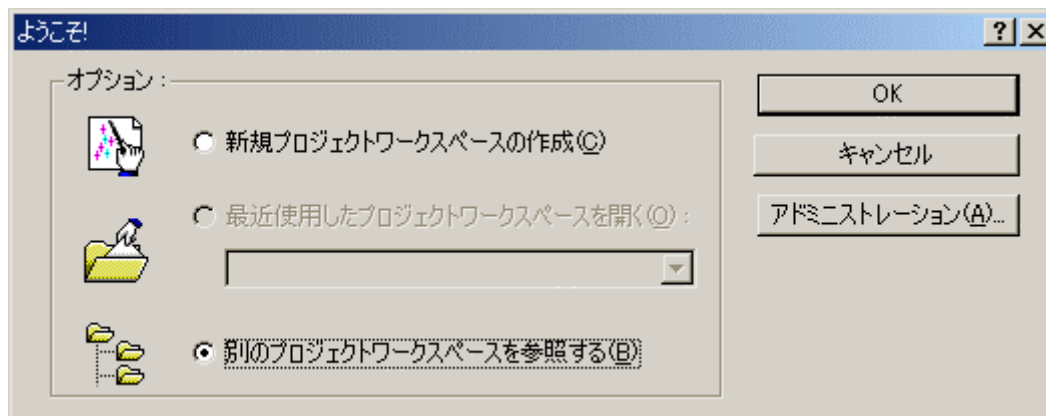


図 2-12 [ようこそ!]ダイアログボックス

ここでは、チュートリアル用ワークスペースを使用するため、[別のプロジェクトワークスペースを参照する]ラジオボタンを選択し、[OK]ボタンをクリックしてください。

[ワークスペースを開く]ダイアログボックスが開きますので、以下のディレクトリを指定してください。
OSインストールドライブ\Workspace\Tutorial\E6000\xxxx

【注】 ソフトウェアのバージョンによっては、上記ディレクトリを指定できない場合があります。その場合は以下のディレクトリを指定してください。

High-performance Embedded Workshop インストール先ディレクトリ
\Tools\Renesas\DebugComp\Platform\E6000\xxxx\Tutorial

ディレクトリの指定後、以下のファイルを選択し[開く]ボタンをクリックしてください。

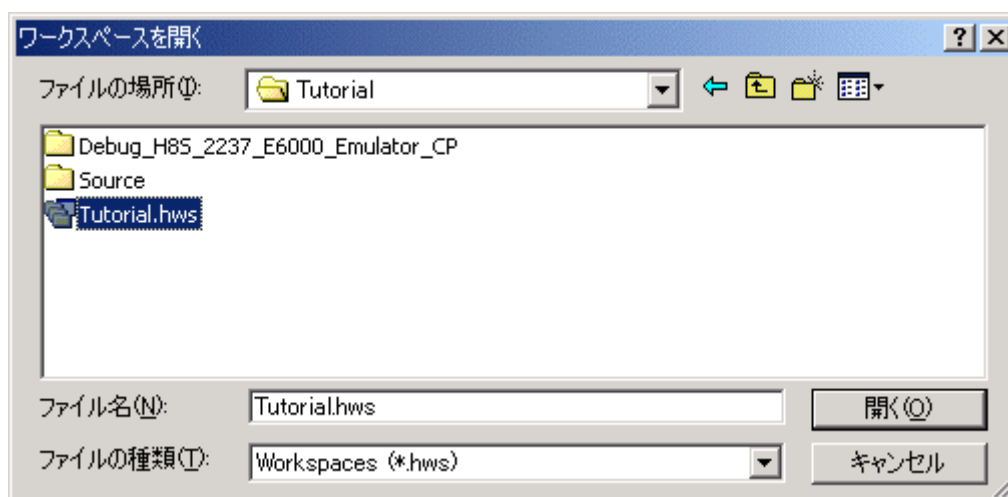


図 2-13 [ワークスペースを開く]ダイアログボックス

コンパイラパッケージがインストールされていない場合、もしくはバージョンの異なるコンパイラパッケージがインストールされている場合は以下のメッセージボックスが表示されます。



図 2-14 メッセージボックス

2 使用前の準備

- (6) [E6000 Driver Details]ダイアログボックスが表示されます。
本ダイアログボックスは、2度目の起動からは表示されません。
また、エミュレータソフトのインストール時にインタフェースドライバを1つのみ選択した場合にも表示されません。

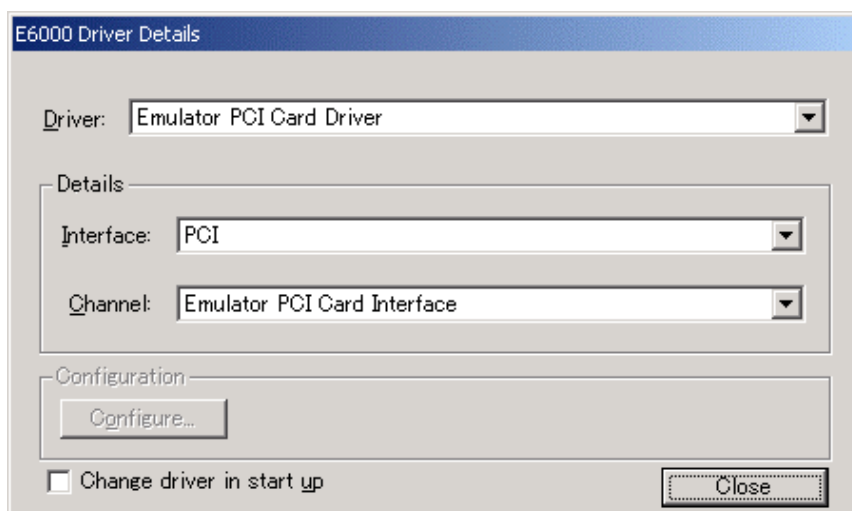


図 2-15 [E6000 Driver Details] ダイアログボックス

- [Driver] コンボボックスで、E6000 エミュレータを接続するドライバの選択を行います。
 - [Interface] には、接続するインタフェース名が表示されます。
 - [Close] ボタンをクリックしてください。
- (7) E6000エミュレータのセットアップを行います。
セットアップ中は、以下のダイアログボックスが表示されます。

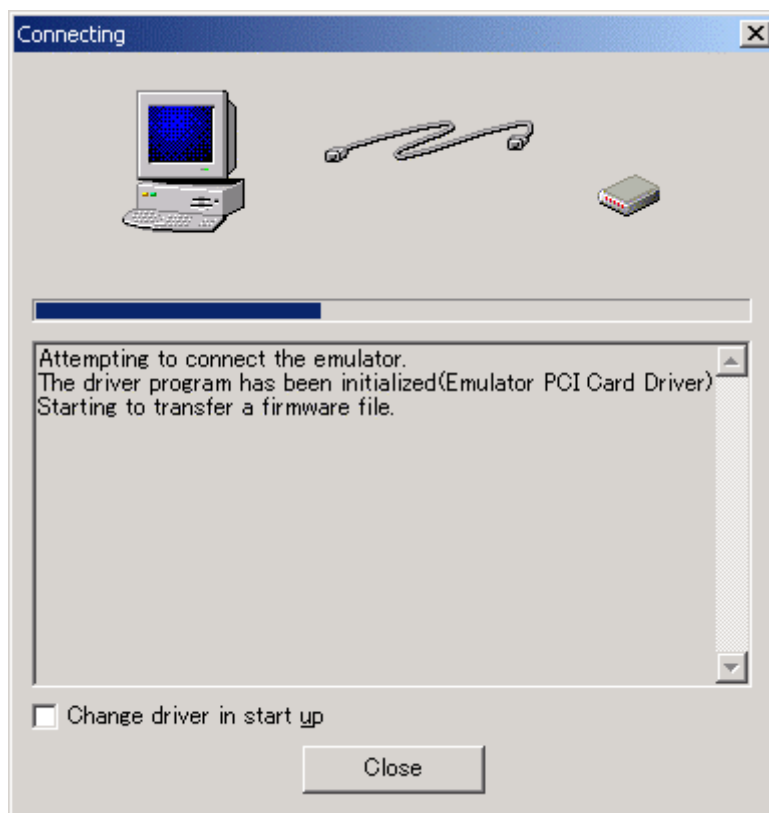


図 2-16 [Connecting] ダイアログボックス

- (8) High-performance Embedded Workshopの[アウトプット]ウィンドウに”Connected”と表示されたら、E6000エミュレータの起動は完了です。

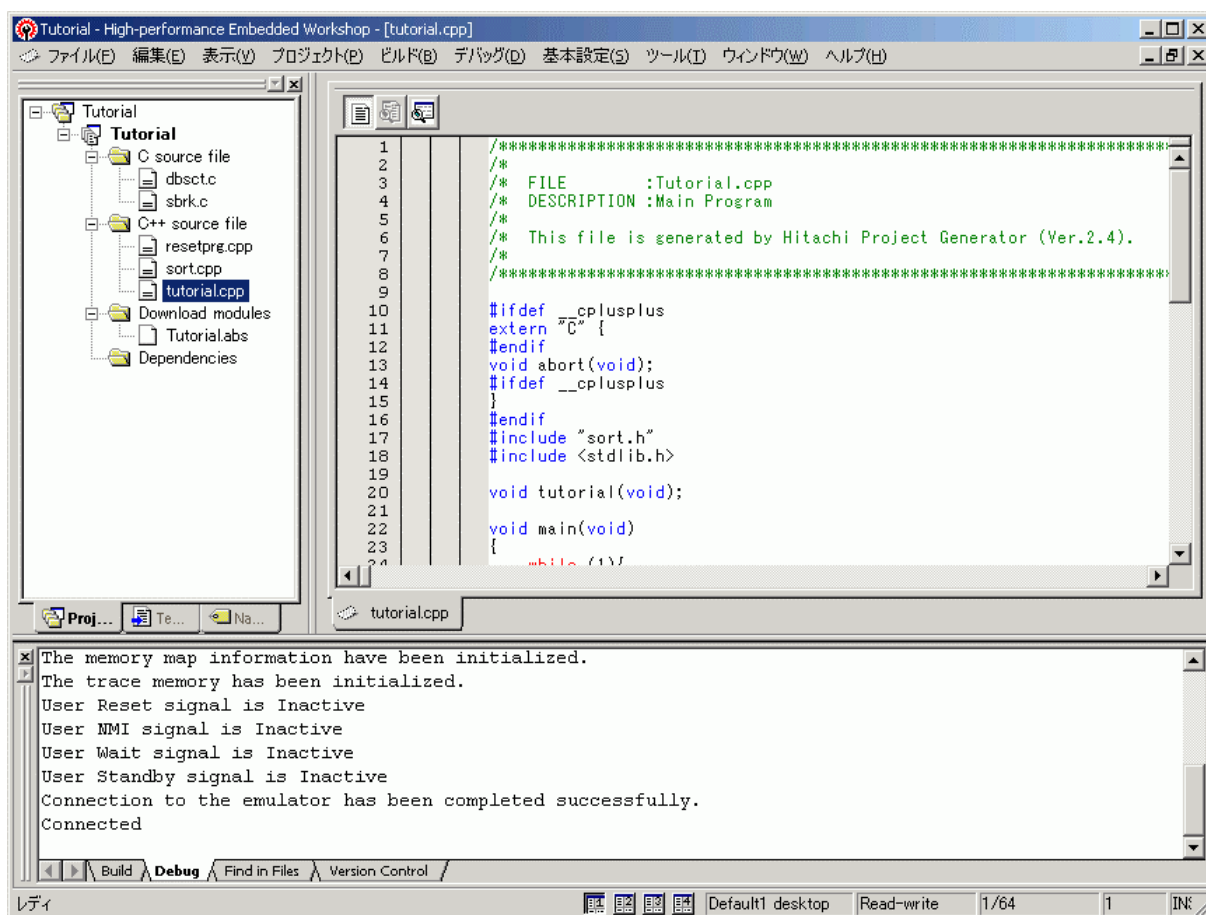


図 2-17 High-performance Embedded Workshop 画面

2.8 通信不良

E6000 エミュレータの電源が OFF の場合、または PC インタフェースケーブルが正しく接続されていない場合は以下のエラーメッセージが表示されます。

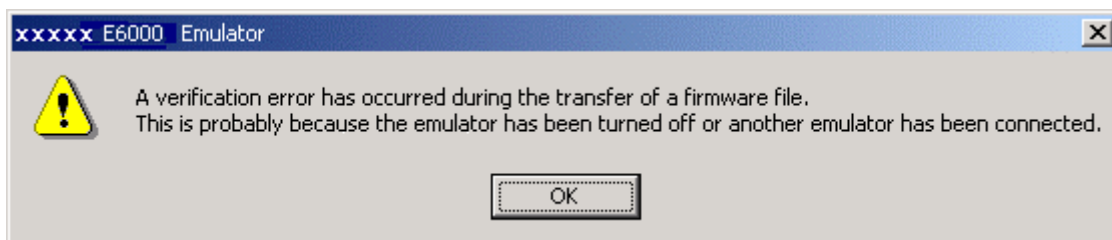


図 2-18 エラーメッセージ

その他のエラーについては、「E6000 エミュレータ用セットアップガイド」を参照してください。

2.9 その他の起動方法

その他の起動方法は、「4 デバッグの準備をする」を参照してください。

2.10 アンインストール

アンインストール方法は、「E6000 エミュレータ用セットアップガイド」を参照してください。

3 E6000 エミュレータ機能

3.1 デバッグの特長

3.1.1 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ソフトウェア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、「3.2 イベント検出システム (CES: Complex Event System)」を参照してください。

ソフトウェアブレークポイント

最大 256 箇所のソフトウェアブレークポイントが設定できます。ソフトウェア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るので、ユーザシステム上の ROM には 1 箇所のみ (On chip Break) 設定できます。

3.1.2 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。

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。

3.1.3 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 で約 6 時間、分解能 16 μ s で約 200 日です。

3.1.4 パフォーマンスアナリシス

E6000 エミュレータは、プログラム実行時間効率の計測機能を備えています。指定した範囲のプログラムの実行効率をヒストグラムまたはパーセントで表示することができます。タイマの分解能は 20ns、40ns、160ns のいずれかの値に設定できます。また、指定した範囲のプログラムの実行回数 (1 ~ 65535) を測定することができます。

3.1.5 バスモニタ

E6000 エミュレータは、バスモニタ機能を備えています。プログラムの実行を中断することなくアクセスのあった領域の値をモニタし、ウィンドウ上に表示することができます。モニタ指定できる領域は 256 バイトで最大 8 ブロックです。また、指定したアドレス (4 箇所まで) に対するアクセスにより、トリガ信号を外部プロ - ブ (EXT2) 出力します。

製品によってはバスモニタ機能をサポートしていない場合があります。

3.2 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。簡単なソフトウェア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。

3.2.1 イベントチャンネル

イベントチャンネルによって、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 (DMAC、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に 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。

3.2.2 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
- 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 (DMAC、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。

3.2.3 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 パスサイクルまで任意に遅らせることができます。

3.2.4 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。

3.3 ハードウェアの特長

3.3.1 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。ただし、選択したデバイスやモードにおいて内蔵 ROM や内蔵 RAM が存在しない場合、これらのエミュレーションメモリは使用できません。従って、外部アドレス空間にプログラムやデータを配置して E6000 エミュレータ単体でデバッグする場合は、別売で用意しているオプションメモリボードを使用してください。

エミュレーションメモリは、MCU アドレス空間の任意のサイズのメモリブロックに割り付けできます。各メモリブロックは、メモリマップ機能を使用して、ユーザシステム上のメモリまたは E6000 エミュレータ上のオプションメモリボードのいずれかに指定でき、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以下に示します。

表3-1 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
ユーザ	ユーザシステム上のメモリ
エミュレータ	オプションメモリボードのメモリ

メモリアドレスの指定されたブロックの内容は、メモリ機能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。なお、プログラム実行中のメモリ内容変更に要する時間は以下ようになります。

- (1) MCU 内蔵 ROM、内蔵 RAM およびオプションメモリボードのメモリ
ユーザプログラムはブレイクすることなく、メモリバスを一時エミュレータ側に切替えてメモリの変更を行います。メモリバスがエミュレータに占有される時間は 256 バイトのリードで最大約 80 μ s (25MHz、内蔵 ROM) です。
- (2) MCU 内蔵 I/O、DTCRAM およびユーザシステム上のメモリ
ユーザプログラム実行をブレイクし、エミュレータがメモリの変更を行います。プログラムが停止する時間は 256 バイトのリードで最大約 2ms (25MHz、エミュレーションメモリ) です。

3.3.2 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 E6000 エミュレータ内蔵クロックと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。選択できるクロックは製品により異なります。詳しくは、「8 本製品固有のソフトウェア仕様」を参照してください。

3.3.3 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 1 (EXT1) および外部プローブ 2 (EXT2) が接続できます。外部プローブ 1 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。また、外部プローブ 2 の信号はバスモニタ機能のトリガ設定 (1~4) の条件が一致した場合にハイレベルを出力しますので、オシロスコープ等のトリガ条件に使用できます。

3.4 スタックトレース機能

E6000 エミュレータでは、スタック情報を用いて、現在の PC がある関数がどの関数からコールされているかを表示します。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。

本機能の使用方法については、「6.17 スタックトレース機能」を参照してください。

3.5 オンラインヘルプ

各機能の操作方法や、コマンドラインウィンドウから入力できるコマンドのシンタックスを記載している、オンラインヘルプ機能があります。

エミュレータ用機能のヘルプを見る場合、[ヘルプ]メニュー [エミュレータヘルプ]を選択してください。

4 デバッグの準備をする

4.1 High-performance Embedded Workshop の起動方法

High-performance Embedded Workshop は以下の手順で起動します。

- (1) ホストコンピュータとE6000エミュレータを接続してください。
- (2) ユーザシステムインタフェースケーブルをご使用の場合は、E6000エミュレータのコネクタとユーザシステムインタフェースケーブルを接続します。ユーザシステムインタフェースケーブル未使用の場合は本手順は不要です。
E6000エミュレータの電源を入れてください。ユーザシステムをご使用の場合は、E6000エミュレータの電源を入れる前に、ユーザシステムの電源を入れてください。
- (3) [スタート]メニューの[プログラム]からHigh-performance Embedded Workshopを起動してください。
- (4) [ようこそ!]ダイアログボックスが表示されます。

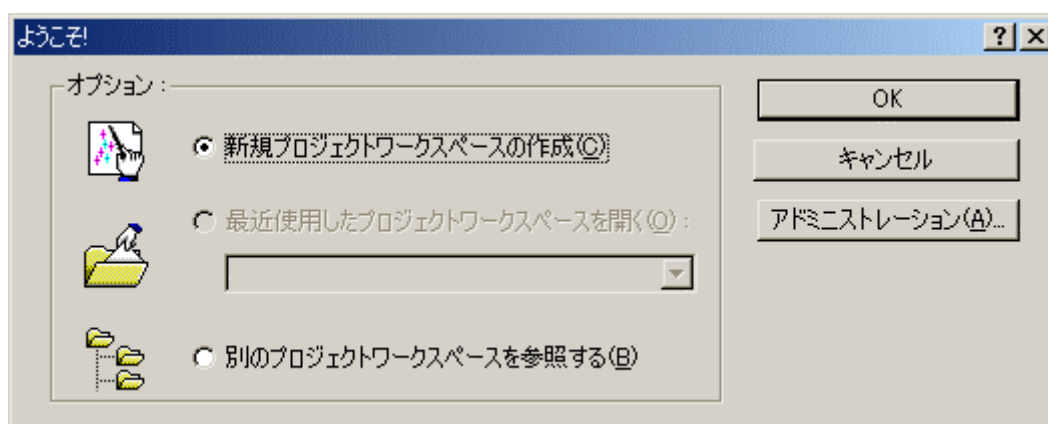


図 4-1 [ようこそ!]ダイアログボックス

- ・[新規プロジェクトワークスペースの作成]ラジオボタン
ワークスペースを新規作成する場合に選択します。
- ・[最近使用したプロジェクトワークスペースを開く]ラジオボタン
既存のワークスペースを使用する場合に選択します。
開いたワークスペースの履歴が表示されます。
- ・[別のプロジェクトワークスペースを参照する]ラジオボタン
既存のワークスペースを使用する場合に選択します。
開いた履歴が残っていない場合に使用します。

ここでは、下記の3つの方法を説明します。

- ・[新規プロジェクトワークスペースの作成] – ツールチェインを使用しない場合
- ・[新規プロジェクトワークスペースの作成] – ツールチェインを使用する場合
- ・[別のプロジェクトワークスペースを参照する]

ツールチェインを使用する場合と使用しない場合では新規プロジェクトワークスペースの作成手順が異なります。本製品には、ツールチェインは含まれていません。ツールチェインは H8S, H8/300 シリーズ C/C++ コンパイラパッケージがインストールされている環境にて使用することができます。

ツールチェインを使用した新規プロジェクトワークスペースの作成についての詳細は、H8S, H8/300 シリーズ C/C++ コンパイラパッケージ付属のマニュアルを参照してください。

4 デバッグの準備をする

4.1.1 新規にワークスペースを作成する場合(ツールチェーン未使用)

- (1) [ようこそ]ダイアログボックスで、[新規プロジェクトワークスペースの作成]ラジオボタンを選択し、[OK]ボタンをクリックしてください。

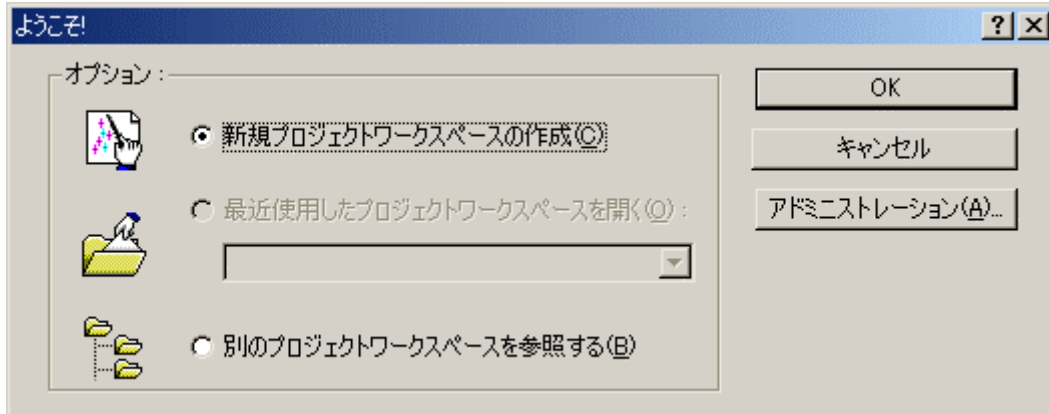


図 4-2 [ようこそ]ダイアログボックス

- (2) 新規プロジェクトワークスペースの作成を開始します。
以下の画面が開きます。

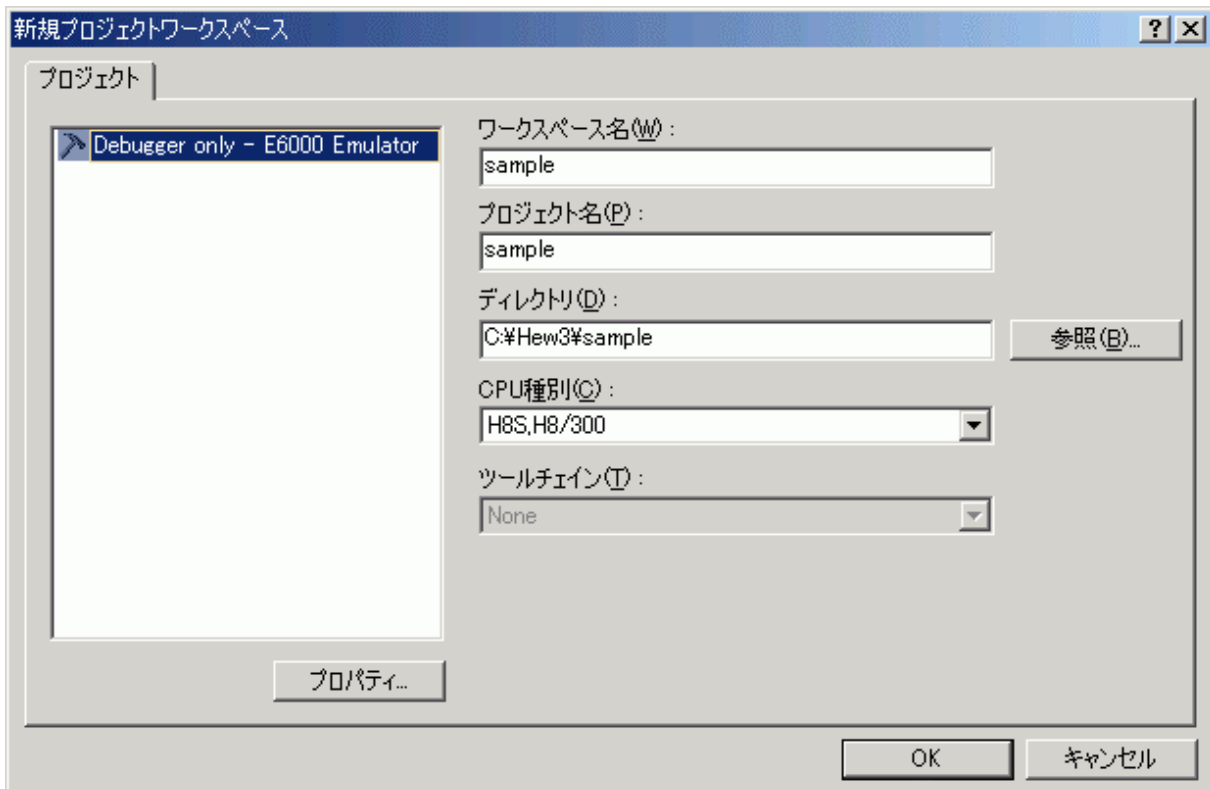


図 4-3 [新規プロジェクトワークスペース]ダイアログボックス

- ・[ワークスペース名]エディットボックス
新規作成するワークスペース名を入力してください。
- ・[プロジェクト名]エディットボックス
プロジェクト名を入力してください。ワークスペース名と同じであれば、入力する必要はありません。
- ・[ディレクトリ]エディットボックス
ワークスペースを作成するディレクトリを入力してください。[参照...]ボタンをクリックしてワークスペースを作成するディレクトリを選択することもできます。
- ・[CPU 種別]ドロップダウンリストボックス
該当する CPU ファミリを選択してください。

その他のリストボックスはツールチェイン設定用ですので、ツールチェインをインストールしていない場合は固定情報が表示されます。

[OK]ボタンをクリックしてください。

- (3) 次に、セッションファイルのターゲットプラットフォームを選択します。
以下の画面が表示されます。

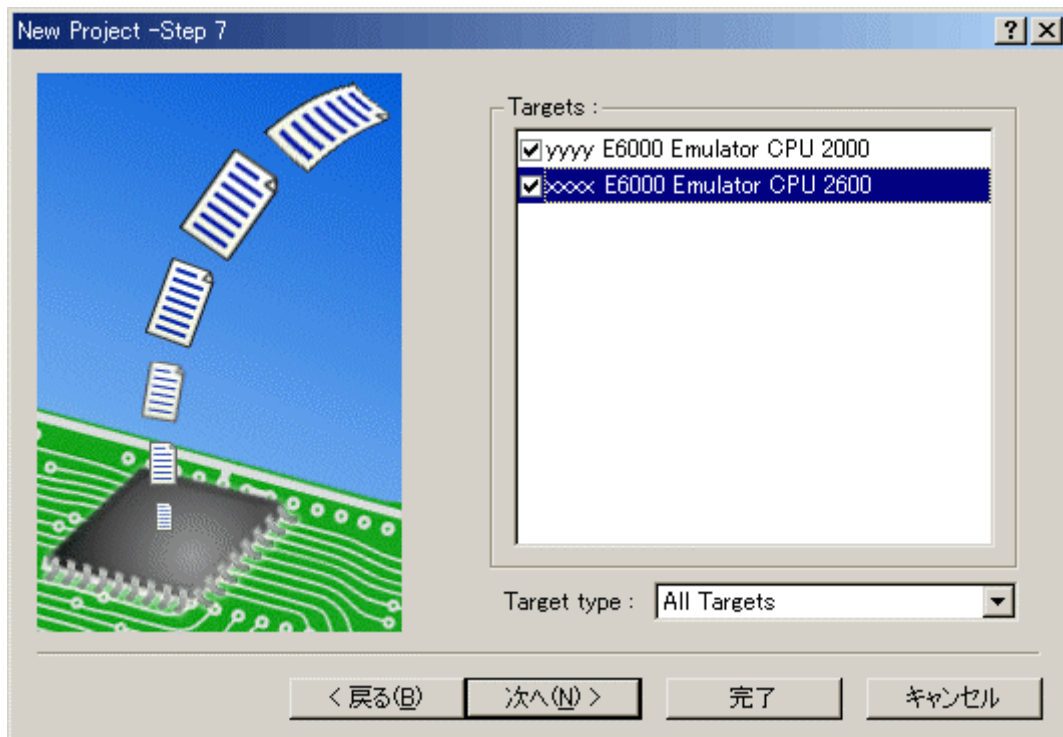


図 4-4 [New Project-Step 7]ダイアログボックス

ここでは、セッションファイルのターゲットプラットフォームを選択します。使用するターゲットプラットフォームにチェックし、[次へ]ボタンをクリックしてください。セッションファイルについての詳細は High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

4 デバッグの準備をする

(4)次に、コンフィグレーションファイル名を設定します。

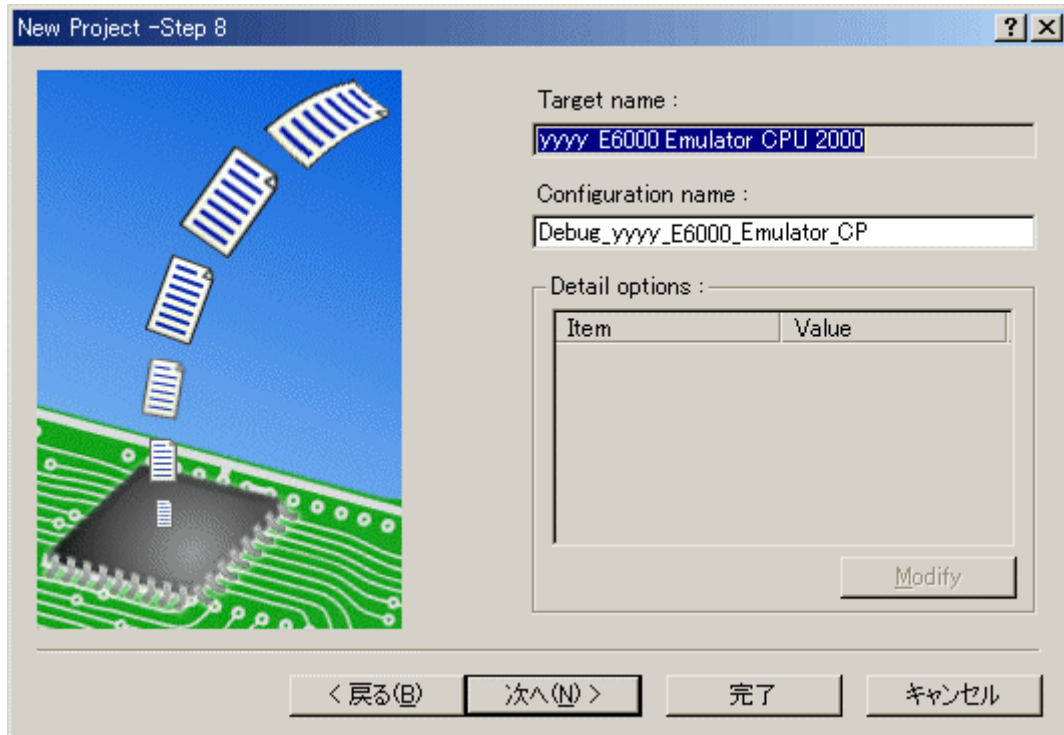


図 4-5 [New Project-Step 8]ダイアログボックス

[New Project-Step 7]ダイアログボックス (図 4-5) で複数のターゲットプラットフォームをチェックした場合には、[次へ]ボタンをクリックし、チェックしたターゲットプラットフォーム毎にコンフィグレーションファイル名を設定します。コンフィグレーションファイル名の設定が完了したら、E6000 エミュレータに関する設定は終了です。

[完了]ボタンをクリックすると[概要]ダイアログボックスが表示されます。[OK]ボタンをクリックすると、High-performance Embedded Workshop が起動します。

(5) 起動後、自動的に E6000 エミュレータが接続されます。

接続が完了すると、[アウトプット]ウィンドウの[Debug]タブに「Connected」と表示されます。

4.1.2 新規にワークスペースを作成する場合(ツールチェーン使用)

(1) [ようこそ!]ダイアログボックスで、[新規プロジェクトワークスペースの作成]ラジオボタンを選択し、[OK]ボタンをクリックしてください。

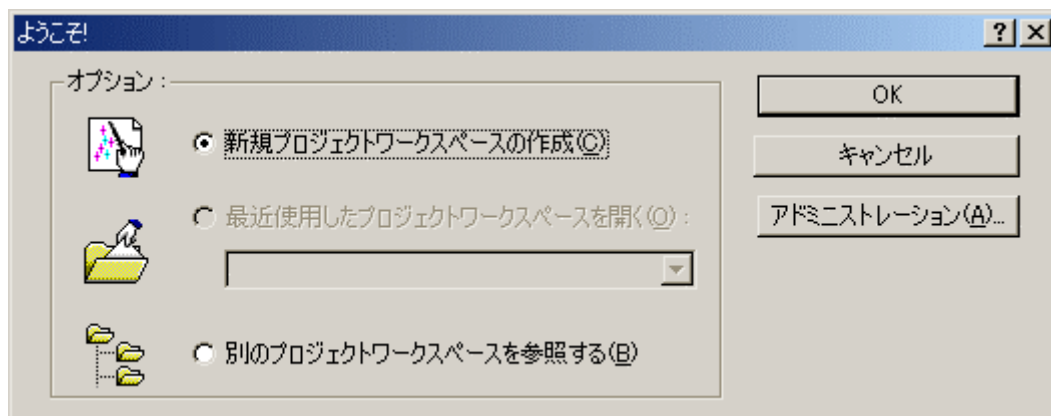


図 4-6 [ようこそ!]ダイアログボックス

- (2)新規プロジェクトワークスペースを作成します。
以下の画面が開きます。

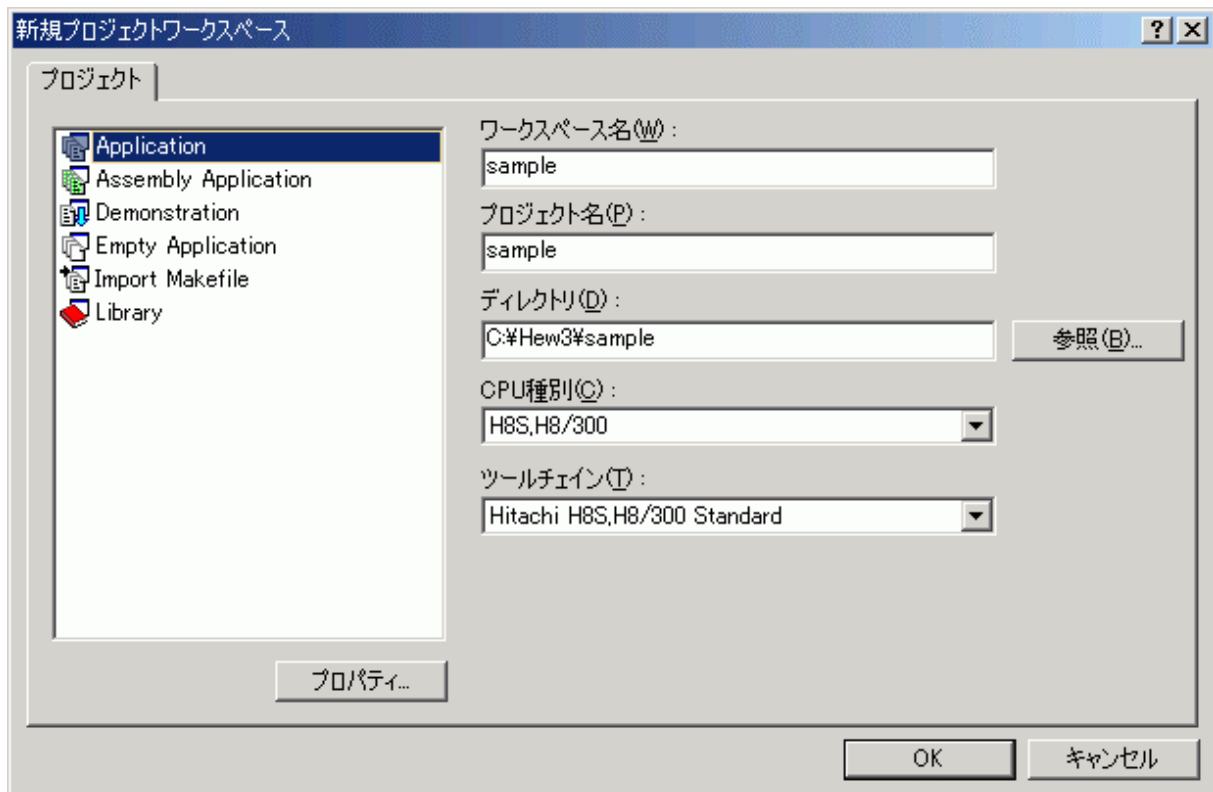


図 4-7 [新規プロジェクトワークスペース]ダイアログボックス

- ・ [ワークスペース名]エディットボックス
新規作成するワークスペース名を入力してください。
- ・ [プロジェクト名]エディットボックス
プロジェクト名を入力してください。ワークスペース名と同じであれば、入力する必要はありません。
- ・ [ディレクトリ]エディットボックス
ワークスペースを作成するディレクトリを入力してください。[参照...]ボタンをクリックしてワークスペースを作成するディレクトリを選択することもできます。
- ・ [CPU 種別]ドロップダウンリストボックス
該当する CPU ファミリを選択してください。
- ・ [ツールチェーン]ドロップダウンリストボックス
ツールチェーンをご使用になる場合、該当するツールチェーン名を選択してください。
使用しない場合、[None]を選択してください。
- ・ [プロジェクトタイプ]リストボックス
使用したいプロジェクトタイプを選択してください。

4 デバッグの準備をする

【留意事項】

1. E6000 エミュレータの場合、[Demonstration]を選択した場合に以下の注意事項があります。

[Demonstration]は H8S, H8/300 コンパイラパッケージ付属のシミュレータ用のプログラムです。生成されたソースファイルを使用する場合、ソースファイル中の“Printf 文”を削除してください。

- (3) 次に、ツールチェーンの設定を行いますので、必要な設定を行ってください。
ツールチェーンの設定が終了したら、以下の画面が表示されます。

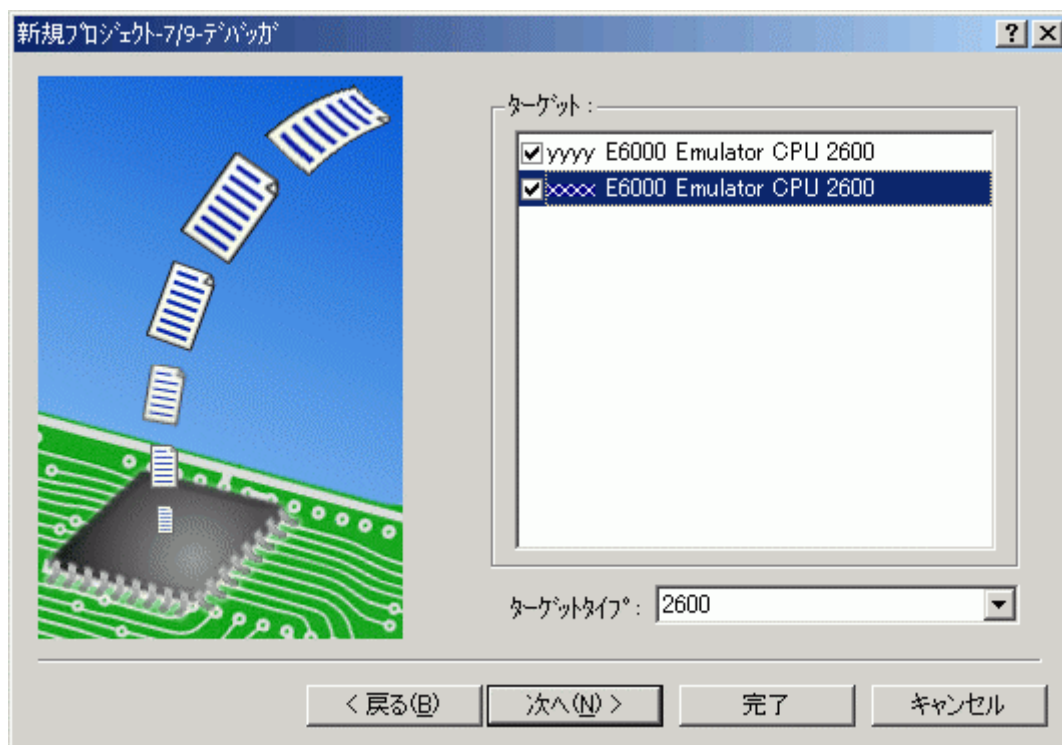


図 4-8 新規プロジェクト-7/9-デバッガダイアログボックス

ここでは、セッションファイルのターゲットプラットフォームを選択します。使用するターゲットプラットフォームにチェックし、[次へ]ボタンをクリックしてください。セッションファイルについての詳細は High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

(4)次に、コンフィグレーションファイル名を設定します。

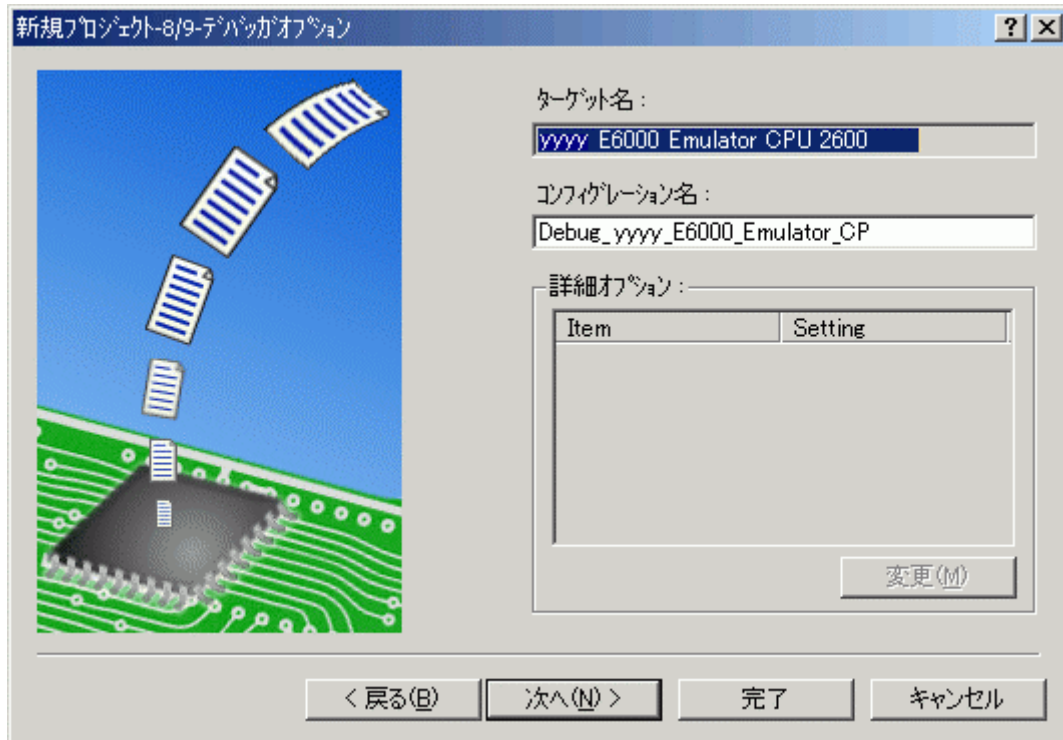


図 4-9 新規プロジェクト-8/9-デバッガオプションダイアログボックス

[新規プロジェクト-7/9-デバッガ]ダイアログボックス (図 4-9) で複数のターゲットプラットフォームをチェックした場合には、[次へ]ボタンをクリックし、チェックしたターゲットプラットフォーム毎にコンフィグレーションファイル名を設定します。コンフィグレーションファイル名の設定が完了しましたら、E6000 エミュレータに関する設定は終了です。

画面の指示に従い、新規ワークスペースの作成を完了してください。High-performance Embedded Workshop が起動します。

(5) 起動後、E6000 エミュレータを接続してください。

E6000 エミュレータは、High-performance Embedded Workshop 起動後すぐに接続する必要はありません。E6000 エミュレータを接続する場合は、E6000 エミュレータ起動時の設定を行ってから接続する方法と E6000 エミュレータ起動時の設定を行わずに簡単に接続する方法があります。エミュレータの接続についての詳細は「4.2 エミュレータの接続」を参照してください。

4.1.3 既存のワークスペースを指定する場合

(1) [ようこそ]ダイアログボックスで、[別のプロジェクトワークスペースを参照する]ラジオボタンを選択し、[OK]ボタンをクリックしてください。

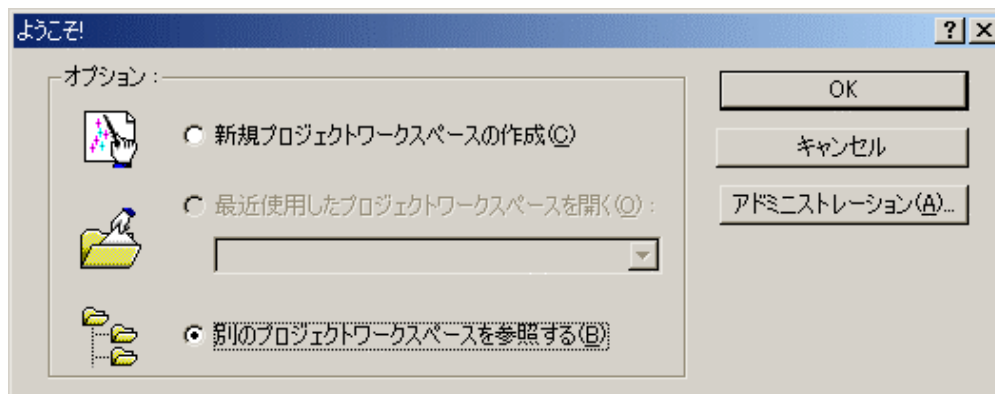


図 4-10 [ようこそ]ダイアログボックス

4 デバッグの準備をする

- (2) [ワークスペースを開く]ダイアログボックスが開きますので、ワークスペースが作成されているディレクトリを指定してください。

ディレクトリの指定後、ワークスペースファイル（拡張子 .hws）を選択し[開く]ボタンを押してください。

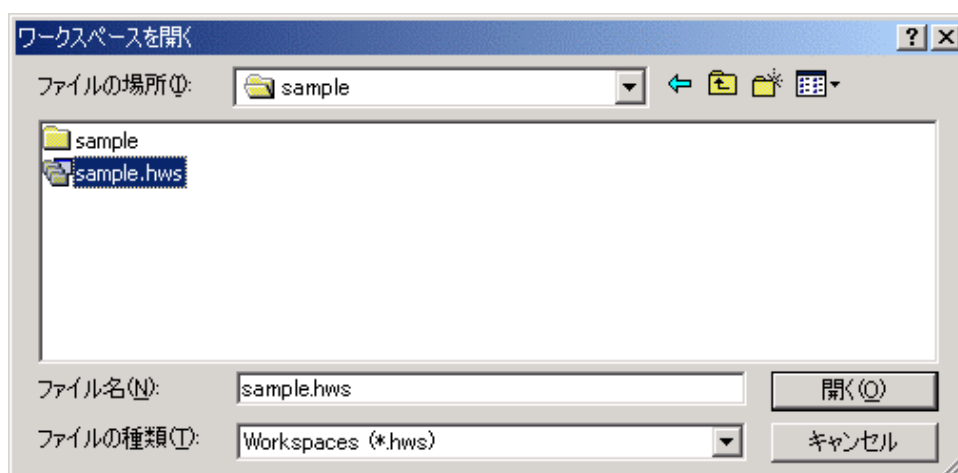


図 4-11 [ワークスペースを開く]ダイアログボックス

- (3) High-performance Embedded Workshop が起動され、指定したワークスペースの保存状態が復元されます。指定したワークスペースがエミュレータに接続された状態を保存していた場合には、エミュレータへの接続が自動で行われます。指定したワークスペースがエミュレータに接続されていない状態を保存していた場合に、エミュレータの接続を行う場合は、「4.2 エミュレータの接続」を参照してください。

4.2 エミュレータの接続

エミュレータの接続には、以下の方法があります。

- (1) E6000 エミュレータ起動時の設定を行ってから接続する方法
[デバッグ -> デバッグの設定...]を選択し、[デバッグの設定]ダイアログボックスを開いてください。ここで、ダウンロードモジュールや起動時に自動的に実行するコマンドチェーンなどを登録することができます。
[デバッグの設定]ダイアログボックスの設定終了後、ダイアログボックスを閉じると、E6000 エミュレータが接続されます。
- (2) E6000 エミュレータ起動時の設定を行わずに簡単に接続する方法
E6000 エミュレータを使用する設定があらかじめ登録されているセッションファイルに切り替えることにより、E6000 エミュレータを簡単に接続できます。

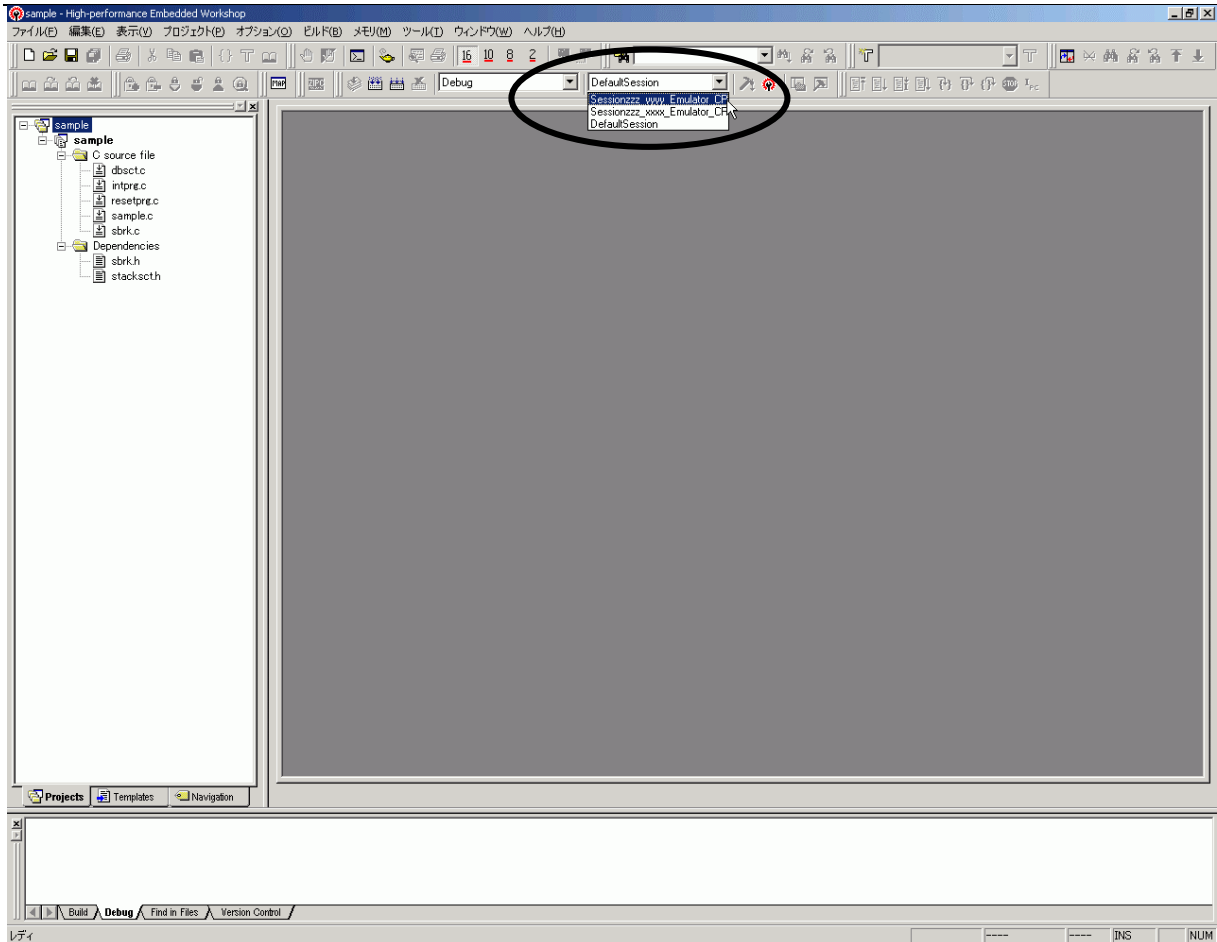


図 4-12 セッションファイルの選択


上記図中の、丸印の中にあるリストボックスから、[新規プロジェクト-8/9-デバッガオプション]ダイアログボックス（図 4-9）の[ターゲット名]テキストボックス内で設定されている文字列を含んだセッションファイル名を選択してください。

このセッションファイルには、E6000 エミュレータを使用する設定が登録されています。

選択終了後、E6000 エミュレータが自動的に接続されます。セッションファイルについての詳細は High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

4.3 エミュレータの再接続

エミュレータ切断状態時に以下の方法で再接続を行うことができます。

[デバッグ -> 接続]を選択するか、接続ツールバーボタン  をクリックしてください。
エミュレータの接続が開始されます。


【注】 エミュレータの再接続を行う場合、あらかじめロードモジュールが High-performance Embedded Workshop に登録されている必要があります。

4.4 エミュレータの終了

エミュレータの終了方法は2通りあります。

- ・起動中のエミュレータの接続を解除する方法
- ・High-performance Embedded Workshop 自体を終了する方法

(1) 起動中のエミュレータの接続を解除する方法

[デバッグ -> 接続解除]を選択するか、接続解除ツールバーボタン  をクリックしてください。

接続解除後、E6000 エミュレータの電源を切ってください。

(2) High-performance Embedded Workshop 自体を終了する方法

[ファイル -> アプリケーションの終了]を選択してください。

メッセージボックスが表示されます。必要なら、[はい]ボタンをクリックし、セッションをセーブしてください。セーブ後、High-performance Embedded Workshop は終了します。不要なら、[いいえ]ボタンをクリックしてください。High-performance Embedded Workshop は終了します。

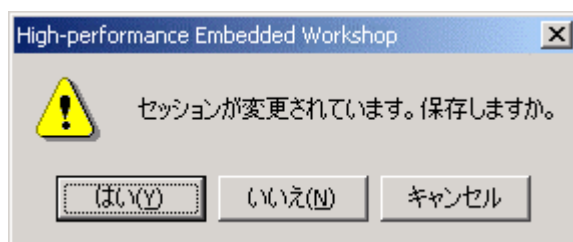


図 4-13 [セッションが変更されています]メッセージボックス

5 デバッグ

デバッグ操作と関連するウィンドウおよびダイアログボックス について説明します。


High-performance Embedded Workshop で共通な下記機能については、High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

- デバッグの準備
- プログラムを表示する
- メモリ内容を参照 / 設定する
- メモリ内容を波形形式で表示する
- メモリ内容を画像形式で表示する
- 変数を参照 / 設定する
- I/O レジスタを参照 / 設定する
- レジスタを参照 / 設定する
- プログラムを実行 / 停止、リセットする
- 関数呼び出し履歴を見る
- コマンドラインインタフェースのデバッグ
- Elf / Dwarf2 のサポート
- ラベルを参照 / 設定する

5.1 エミュレーション環境を設定する

この節では、エミュレーションを行うための環境を設定する方法を説明します。

5.1.1 Configuration Properties ダイアログボックスを開く

[基本設定->エミュレータ->システム...]を選択するか、[Emulator System]ツールバーボタンをクリックすると、[Configuration Properties]ダイアログボックスが開きます。

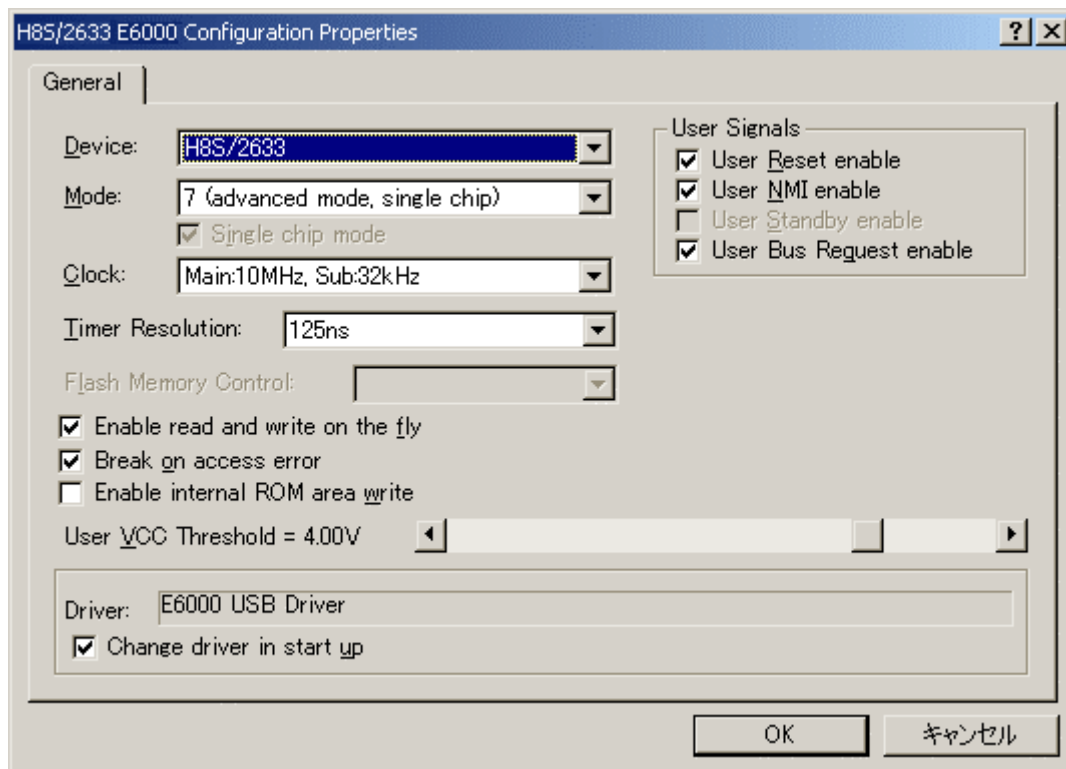


図 5-1 Configuration Properties ダイアログボックス (General ページ)

5 デバッグ

E6000 エミュレータにプログラムをダウンロードする前に、このダイアログボックスで対象 MCU 条件を設定します。

[General]

[Device]	エミュレーションする MCU を指定します。MCU 一覧にない MCU を指定する場合は、Custom を指定し、使用する MCU の機能を設定します。詳細に関しては、それぞれのハードウェアマニュアルを参照してください。
[Mode]	MCU の動作モードを指定します。
[Clock]	MCU のクロック速度とサブクロック速度を指定します。
[Timer Resolution]	実行時間の測定に使用するタイマの分解能を設定します。 分解能は以下のいずれかから選択できます。 20ns, 125ns, 250ns, 500ns, 1us, 2us, 4us, 8us, 16us 実行時間測定用タイマは 40 ビットのカウンタで構成されています。 測定可能な最大時間は分解能 20ns で約 6 時間、分解能 16us で約 200 日です。 カウンタがオーバーフローした場合、オーバーフローしたことを示すプロンプト">"と共に測定可能な最大時間を表示します。
[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。 リアルタイム性はありませんので、リアルタイムでのエミュレーションを行いたい場合には、チェックしないでください。
[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。
[Enable internal ROM area write]	このチェックボックスをチェックすると、ユーザプログラムによる内蔵 ROM エリアへのライトが可能になります。ライトが行われたかどうかは、[拡張モニタ] ウィンドウによって知らされます。
[User VCC Threshold]	ユーザシステム電圧レベルを指定します。
[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、NMI 信号、スタンバイ信号、バスリクエスト信号を有効にします。
[Driver]	現在使用している E6000 ドライバを表示しています。
[Change driver in start up]	このチェックボックスをチェックすると、次回 E6000 接続時にドライバを選択することができます。

【注】 本ダイアログボックスで設定可能な項目はご使用のエミュレータにより異なります。
詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

5.1.2 MCU 一覧にない MCU を設定する

[Configuration Properties]ダイアログボックスの[Device]項目で[Custom]を選択すると、[Configuration Properties]ダイアログボックスに[Custom Device]ページが追加されます。

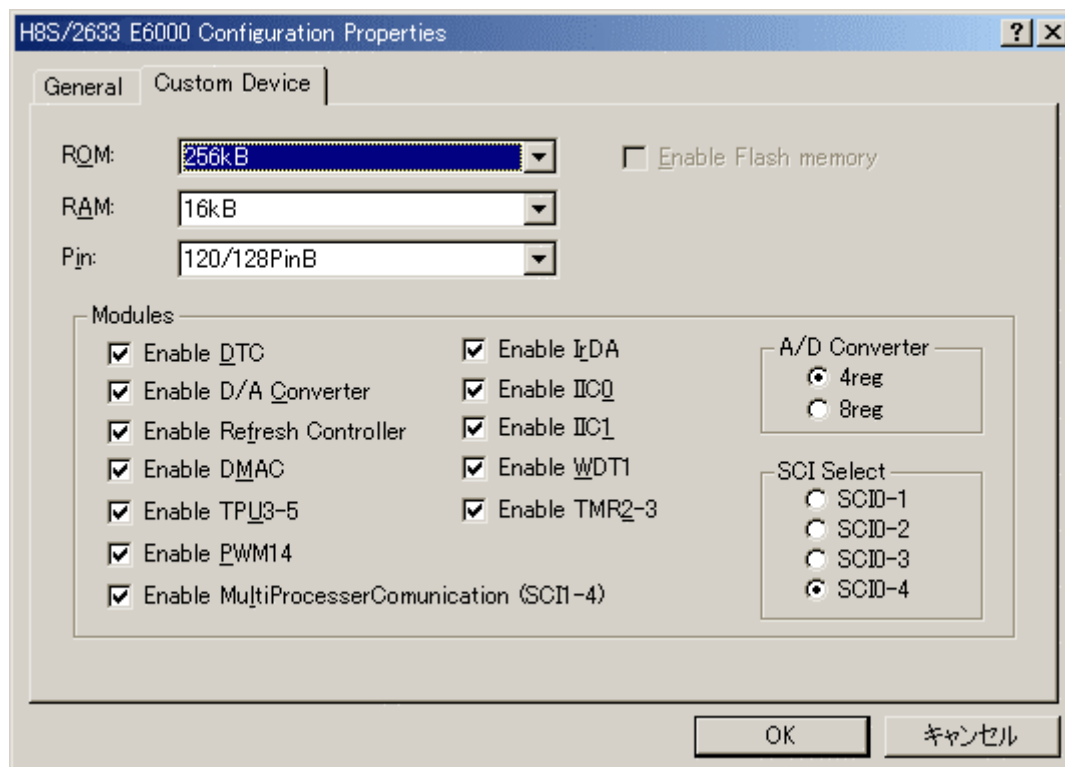


図 5-2 Configuration Properties ダイアログボックス (Custom Device ページ)

ここではDevice指定のMCU一覧にないMCUの機能を指定します。各項目は直前に選択されたデバイスの値が反映されています。

5 デバッグ

[Custom Device]

[ROM]	内蔵 ROM エリアサイズを指定します。
[RAM]	内蔵 RAM エリアサイズを指定します。
[Pin]	製品のパッケージを指定します。
[Modules]	チェックボックスをチェックし、内蔵周辺機能を指定します。

【注】 本ダイアログボックスで設定可能な項目はご使用のエミュレータにより異なります。
また、ご使用のエミュレータによっては[Custom Device]機能をサポートしていない場合があります。
詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

5.1.3 接続するインタフェースを選択する

[Configuration Properties]ダイアログボックスの[Change driver in start up]項目をチェックすると、次回 E6000 接続時にドライバを選択することができます。

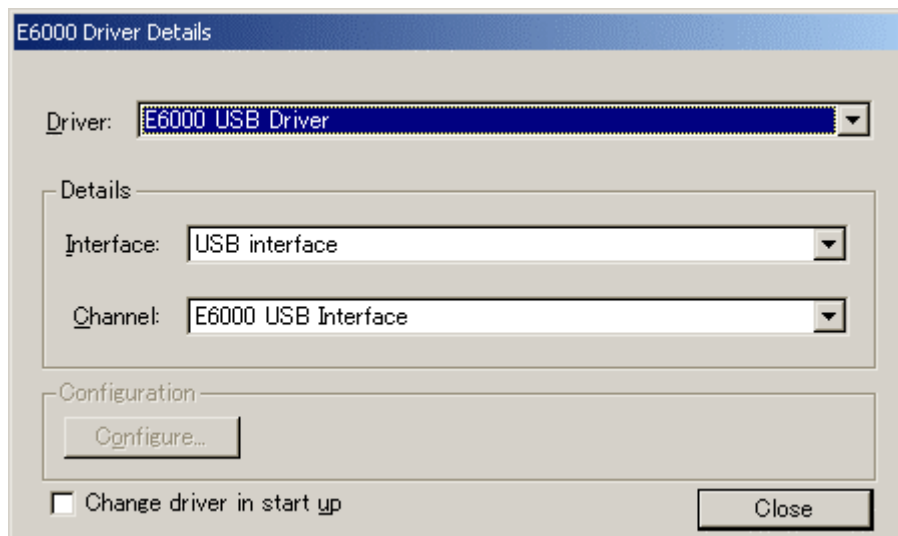



図 5-3 Driver Details ダイアログボックス

[Driver]	High-performance Embedded Workshop と E6000 エミュレータを接続するドライバを選択します。
[Details]	接続するドライバの詳細を設定します。
[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。 本エミュレータでは設定を変更する必要はありません。
[Channel]	選択したインタフェースのチャンネルを選択します。 本エミュレータでは設定を変更する必要はありません。
[Configuration]	ドライバの設定を行います。
[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、設定ダイアログボックスを表示します。(本エミュレータでは使用できません。)
[Change driver in start up]	このチェックボックスをチェックすると、次回 E6000 接続時にドライバを選択することができます。

5.1.4 Memory Mapping ダイアログボックスを開く

[基本設定->エミュレータ->メモリリソース...]を選択するか、[Emulator Memory Resource]ツールバーボタンをクリックすると、[Memory Mapping]ダイアログボックスが開きます。

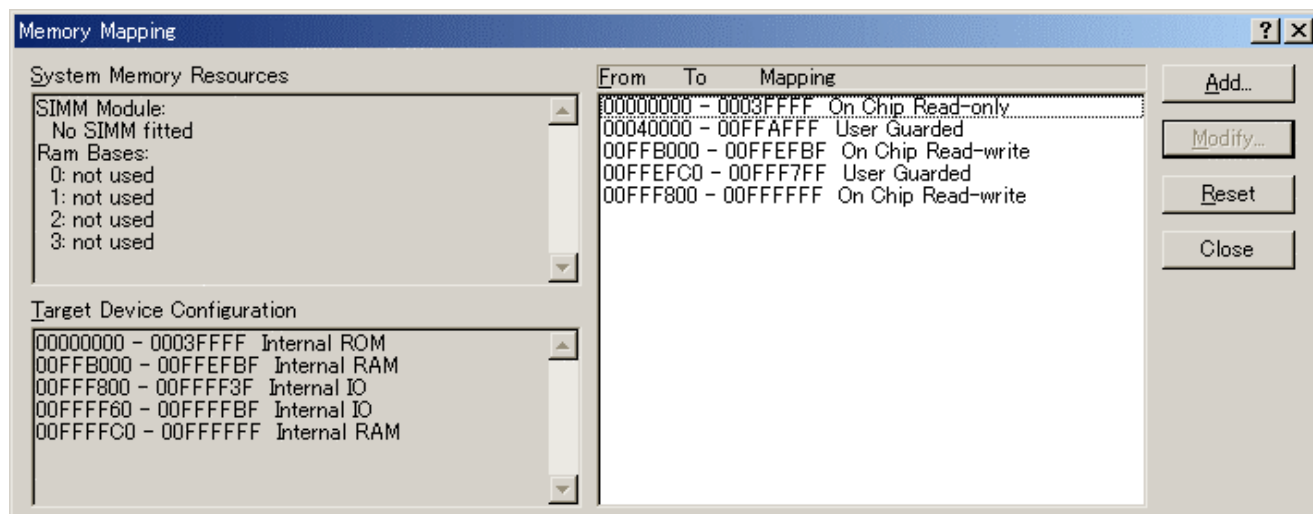


図 5-4 Memory Mapping ダイアログボックス

現在のメモリマップを表示しています。

H8S ファミリオよび H8/300H シリーズ E6000 は 4 ブロックのエミュレーションメモリをサポートしています。これらは実装した SIMM に従い、256kbyte または 1Mbyte 単位の指定ができます。各ブロックは、256kbyte または 1Mbyte 境界のアドレス空間に置くことができます。

メモリマップには、H'40(D'64)バイトのブロックがあります。各 64 バイトのブロックは、内部(エミュレーション)メモリまたは外部メモリ、ガーデッド(アクセス禁止)、書き込み禁止またはリード・ライトに設定することができます。

H8/300 シリーズ E6000 は標準でエミュレーションメモリを実装しています。

メモリマップは 1 バイト単位で内部(エミュレーション)メモリまたは外部メモリ、書き込み禁止またはリード・ライトに設定することができます。

[Add...]	メモリマップのアドレス範囲および属性を変更するため [Edit Memory Mapping] ダイアログボックスを開きます。
[Modify...]	メモリマップのアドレス範囲および属性を変更するため [Edit Memory Mapping] ダイアログボックスを開きます。
[Reset]	メモリマップをデフォルト設定にリセットします。
[Close]	ダイアログボックスを閉じます。

ターゲットマイコンのメモリマップ情報は、[ステイタス]ウィンドウの[Memory]シートに表示されます。

【注】 ご使用のエミュレータによりエミュレーションメモリをサポートしていない場合があります。また、Memory Mapping 機能自体をサポートしていない場合もあります。詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

5.1.5 メモリマップ設定を変更する

[Memory Mapping]ダイアログボックスで[Add...]ボタンをクリックするか、または変更したいメモリマップ設定情報を選択し[Modify...]ボタンをクリックすると[Edit Memory Mapping]ダイアログボックスが開きます。

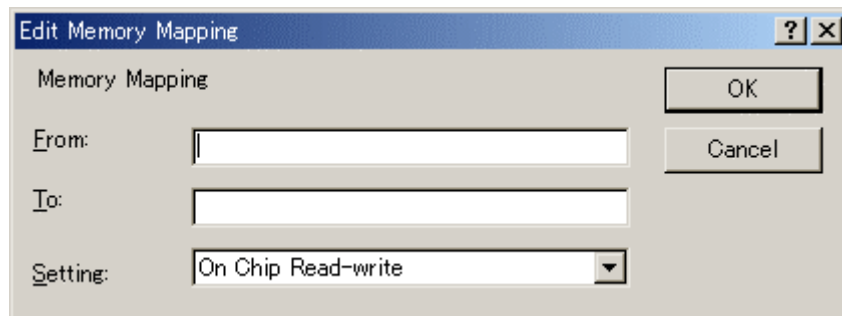


図 5-5 Edit Memory Mapping ダイアログボックス

メモリマップのアドレス範囲および属性を変更します。

[From] 範囲の開始アドレスを入力します。

[To] 範囲の終了アドレスを入力します。

[Setting] マップ設定を選択します。

マップ設定は以下9つの属性があり、User(外部メモリ)および Emulator(エミュレーションメモリ)属性については変更可能です。

On Chip Read-write	(変更できません)
On Chip Read-only	(変更できません)
On Chip Guarded	(変更できません)
User Read-write	(シングルチップモード時は設定できません)
User Read-only	(シングルチップモード時は設定できません)
User Guarded	
Emulator Read-write	
Emulator Read-only	
Emulator Guarded	

5.2 プログラムをダウンロードする

プログラムをダウンロードし、ソースコードおよびアセンブリ言語二モニックとして見る方法を説明します。

- 【注】** ブレークが起こると、High-performance Embedded Workshop はプログラムカウンタ(PC)の場所を表示します。多くの場合、例えば、ELF/DWARF2 をベースにしたプロジェクトが、ビルド時のパスから移動した場合、ソースファイルを自動的に見つけることができない場合があります。この場合、High-performance Embedded Workshop はソースファイルブラウザダイアログボックスを開くので、ユーザは手動でファイルを探することができます。

5.2.1 プログラムをダウンロードする

デバッグするロードモジュールをダウンロードします。

プログラムのダウンロードは、[デバッグ->ダウンロード]からロードモジュールを選択するか、[ワークスペース]ウィンドウの[Download modules]のロードモジュールを右クリックすると表示されるポップアップメニューより[ダウンロード]を選択します。

- 【注】** プログラムをダウンロードする場合、あらかじめロードモジュールとして High-performance Embedded Workshop に登録されている必要があります。

5.2.2 ソースコードを表示する

[ワークスペース]ウィンドウ上のソースファイルをダブルクリックするか、ソースファイル上でマウスの右ボタンをクリックしてポップアップメニューを表示して[開く]を選択すると、[エディタ] ウィンドウが表示されます。

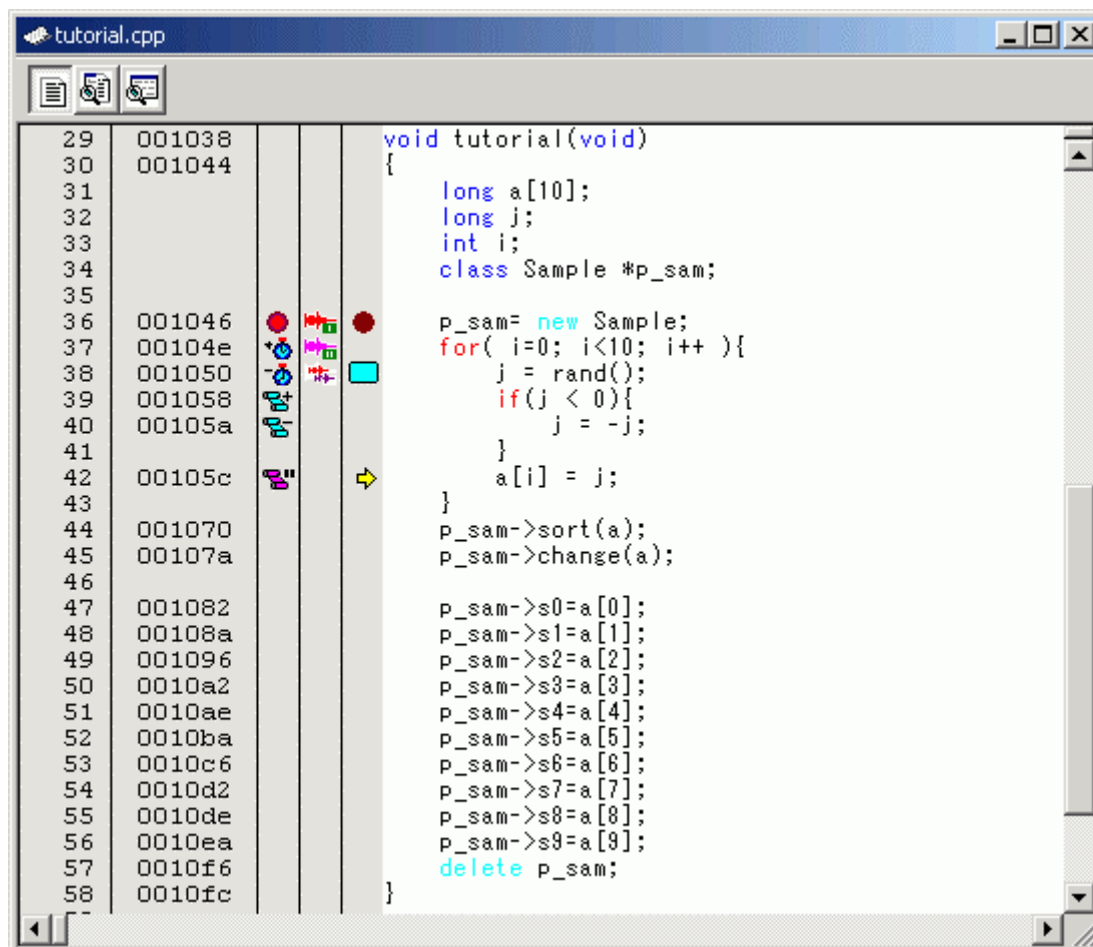


図 5-6 [エディタ]ウィンドウ

本ウィンドウでは左端に行情報として下記を表示します。

- | | | |
|------|---------------------|----------------------|
| 1 列目 | (行番号カラム) | ソースファイルに対応する行番号 |
| 2 列目 | (ソースアドレスカラム) | ソース行に対応するアドレス情報 |
| 3 列目 | (Event カラム) | イベント情報(ブレーク) |
| 4 列目 | (EXT.2 Trigger カラム) | EXT.2 Trigger 情報 |
| 5 列目 | (S/W ブレークポイントカラム) | PC、ブックマーク、ブレークポイント情報 |

右側のソース表示画面をテキスト領域と呼びます。

行番号カラム







ソースファイルに対応する行番号を表示します。

ソースアドレスカラム

プログラムをダウンロードすると、ソースアドレスカラムに現在のソースファイルに対するアドレスを表示します。本機能は PC 値やブレークポイントをどこに設定するかを決めるときに便利です。

Event カラム






Event カラムには下記を表示します。

-  イベントチャンネルまたは範囲チャンネルによるアドレス条件ブレイク
-  イベントチャンネルによる時間計測の開始
-  イベントチャンネルによる時間計測の終了
-  Point to Point の範囲トレースの開始
-  Point to Point の範囲トレースの終了
-  トレース停止

これらの設定は、ポップアップメニューからも可能です。

EXT.2 Trigger カラム




EXT.2. Trigger カラムには下記を表示します。

-  EXT.2-1 トリガの条件
-  EXT.2-2 トリガの条件
-  EXT.2-3 トリガの条件
-  EXT.2-4 トリガの条件
-  EXT.2 トリガの条件が2つ以上ある場合

これらの設定は、ポップアップメニューからも可能です。

S/W ブレークポイントカラム

S/W ブレークポイントカラムには下記を表示します。

-  ブックマークを設定している
-  ソフトウェアブレークを設定している
-  PC 位置

⇒すべてのソースファイルでカラムをオフにするには

1. [エディタ]ウィンドウを右クリックしてください。または、[編集]メニューを選択してください。
2. [表示カラムの設定...]メニュー項目をクリックしてください。
3. [エディタ全体のカラム状態]ダイアログボックスを表示します。
4. チェックボックスは、そのカラムが有効か無効かを示します。チェックしている場合は有効です。チェックボックスがグレー表示の場合、一部のファイルではカラムが有効で、別のファイルでは無効であることを意味します。
オフにしたいカラムのチェックボックスからチェックを外してください。
5. [OK]ボタンをクリックして、新しいカラム設定を有効にしてください。

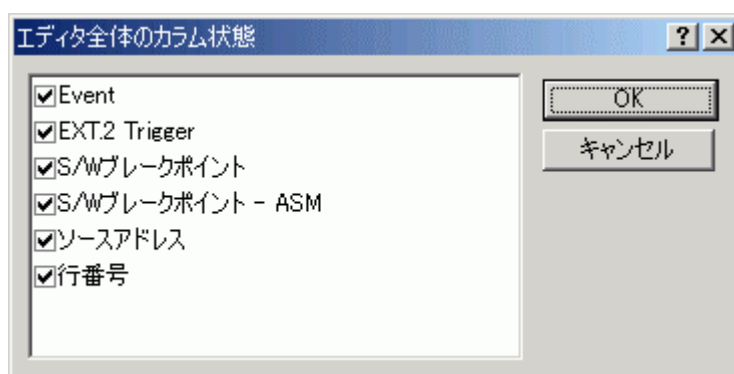


図 5-7 エディタ全体のカラム状態ダイアログボックス

⇒1つのソースファイルでカラムをオフにするには


1. 削除したいカラムのあるソースファイルを開き、[編集]メニューをクリックしてください。
2. [カラム]メニュー項目をクリックしてください。カスケードしたメニュー項目が現れます。各カラムを、このポップアップメニューに表示します。カラムが有効である場合、名前の横にチェックマークがあります。エントリをクリックすると、カラムの表示、非表示を切り替えます。

5.2.3 アセンブリ言語コードを表示する

ソースファイルが開いている場合、[エディタ]ウィンドウ上でマウスの右ボタンをクリックしてポップアップメニューから[逆アセンブリ]を選択すると、[逆アセンブリ]ウィンドウを表示します。[逆アセンブリ]ウィンドウの表示開始アドレスは、[エディタ]ウィンドウのカーソル位置に対応するアドレスとなります。

また、[エディタ]ウィンドウの[逆アセンブリモードで表示]ボタンを使用して、逆アセンブリコードを表示することもできます。

ソースファイルが存在しない場合、次のいずれかの方法でアセンブリ言語レベルでコードを表示できます。

- ・ [逆アセンブリ]ツールバーボタンをクリックする
- ・ [表示 -> 逆アセンブリ...]を選択する
- ・ “Ctrl + D” アクセラレータを使用する

この場合、[逆アセンブリ]ウィンドウは現在の PC の位置で開きます。アドレス、コード(オプション)、逆アセンブルされたニーモニック(可能なときはラベル付きで)を表示します。

オプションで、そのアドレスから開始するソースラインすべてを表示させる混合モード表示もサポートします。混合モードの逆アセンブルを表示するには、[混合モードで表示]ボタンをクリックしてください。

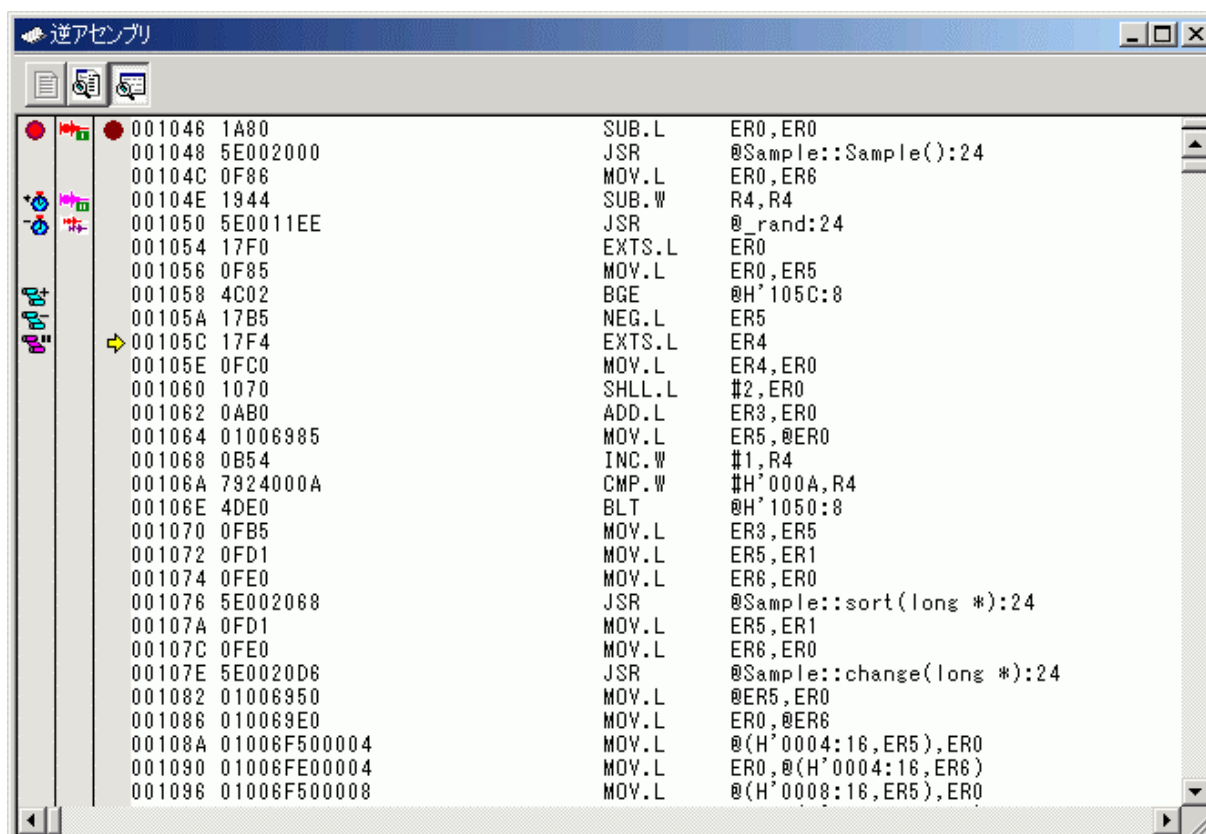


図 5-8 逆アセンブリウィンドウ

本ウィンドウでは左はじに行情報として下記を表示します。

- | | |
|-----------------------------|------------------|
| 1 列目 (Event カラム) | イベント情報(ブレイク) |
| 2 列目 (EXT.2 Trigger カラム) | EXT.2 Trigger 情報 |
| 3 列目 (S/W ブレイクポイント-ASM カラム) | PC、ブレイクポイント情報 |

使用方法はソースコードの表示ウィンドウと同じです。

5.2.4 アセンブリ言語コードを修正する

[逆アセンブリ]ウィンドウで修正したい命令をダブルクリックするか、ポップアップメニューから[編集...]を選択すると、[アセンブル]ダイアログボックスが開き、アセンブリ言語コードを修正することができます。



図 5-9 アセンブルダイアログボックス

アドレス、命令コード、およびニーモニックを表示します。

[ニーモニック]フィールドに新しい命令を入力（または古い命令を編集）します。

'Enter'キーを押すと、メモリ内容を新しい命令コードに書き換えて、次の命令に移ります。

[OK]ボタンをクリックすると、メモリ内容を新しい命令コードに書き換えて、ダイアログボックスを閉じます。

[キャンセル]ボタンをクリックするか"Esc"キーを押すと、メモリ内容を書き換えずに、ダイアログボックスを閉じます。

【注】 アセンブリ言語コードは、現在のメモリ内容から表示しています。メモリ内容を修正すると、[逆アセンブリ]ウィンドウおよび[アセンブル]ダイアログボックスでは、新しいアセンブリ言語コードを表示します。しかし、[エディタ]ウィンドウに表示しているソースファイルは変更しません。これはソースファイルにアセンブラを含む場合も同じです。

5.2.5 特定のアドレスを見る

[逆アセンブリ]ウィンドウを使って作成したプログラムを表示している場合、プログラム内の他の箇所を見たいときがあります。そのような場合、プログラム内のコードをスクロールせずに特定のアドレスにジャンプすることができます。

[逆アセンブリ]ウィンドウでアドレスをダブルクリックするか、ポップアップメニューから[表示アドレス設定...]を選択すると、[アドレス指定]ダイアログボックスを表示します。



図 5-10 アドレス設定ダイアログボックス

[アドレス]エディットボックスにアドレスを入力して、[OK]ボタンをクリックするか'Enter'キーを押します。アドレスは、ラベル名で入力することも可能です。

[逆アセンブリ]ウィンドウを更新して新しいアドレスコードを表示します。オーバーロード関数またはクラス名を入力した場合、[関数選択]ダイアログボックスを開くので、関数を選択してください。


5.2.6 現在のプログラムカウンタアドレスを見る

アドレスまたは値を入力できる場所では、式も入力することができます。先頭に"#"文字をつけたレジスタ名を入力すると、そのレジスタ内容を式の値として使用します。

[アドレス指定]ダイアログボックスで"#pc"という式を入力すると、[エディタ]または[逆アセンブリ]ウィンドウには、現在の PC アドレスを表示します。例えば、"#PC+0x100"といった PC レジスタおよびオフセットの式を入力することにより現在の PC のオフセットを表示することができます。

5.3 現在の状態を表示する

デバッグプラットフォームの現在の状態を知るには[ステイタス]ウィンドウを表示します。

[ステイタス]ウィンドウを開くには、[表示->CPU->ステイタス]を選択するか、[ステイタスの表示]ツールバーボタン  をクリックします。

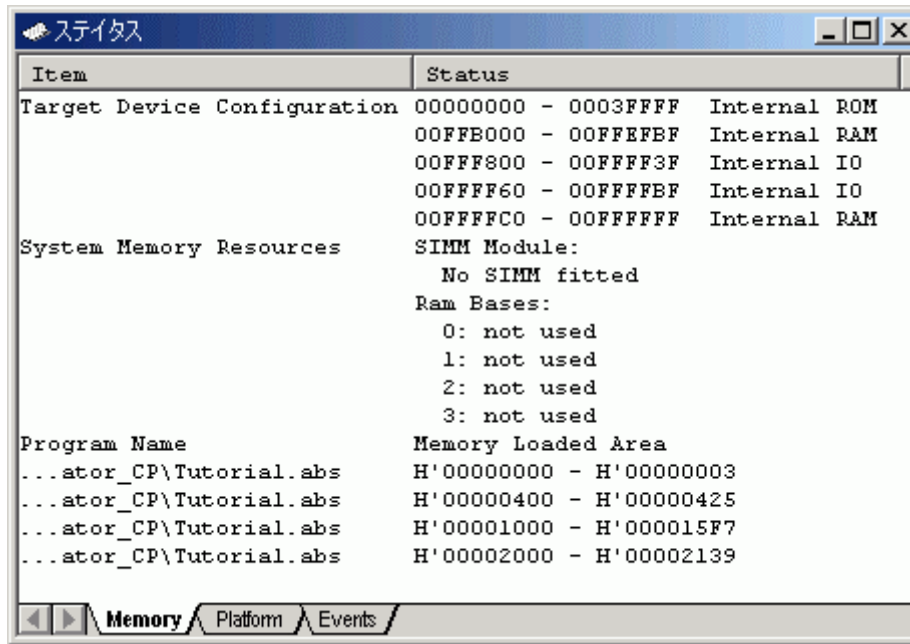


図 5-11 ステイタスウィンドウ

[ステイタス]ウィンドウには、3枚のシートがあります。

[Memory]シート

メモリマッピングおよび現在ロードしたオブジェクト・ファイルが使用するメモリエリアなど、現在のメモリステータスに関する情報を含んでいます。

[Platform]シート

CPU種別および動作モードなど、デバッグプラットフォームのステータス情報、実行状態および実行統計情報を含んでいます。

[Events]シート

リソース情報およびブレークポイント等のイベント情報に関する情報を含んでいます。


【注】 本ウィンドウに表示する項目はご使用のエミュレータにより異なります。
詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

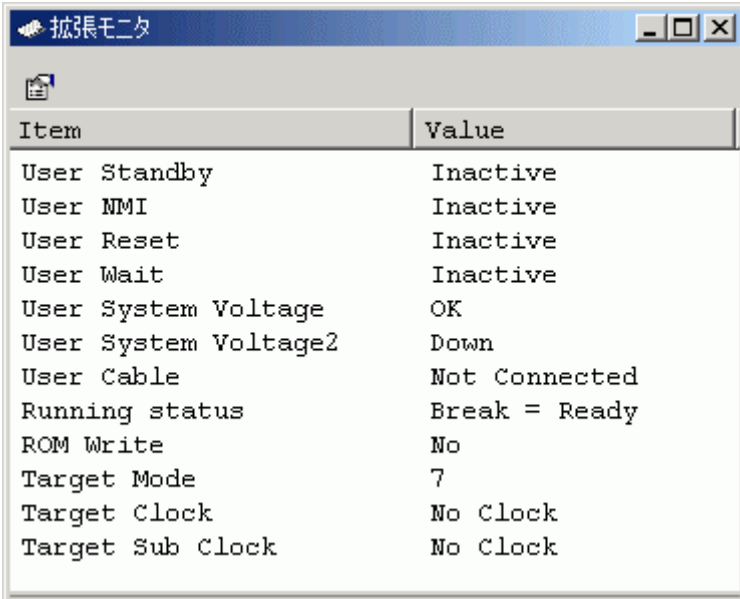
5.4 エミュレータの情報を定期的に読み出し表示する

ユーザプログラム実行中/停止中にかかわらず変化するエミュレータの情報を知るには[拡張モニタ]ウィンドウを使用します。

【注】 拡張モニタ機能は E6000 エミュレータのハードウェア回路によりユーザシステムやエミュレータ内部の MCU から出力される信号をモニタするため、ユーザプログラムの実行に影響を与えることはありません。

5.4.1 [拡張モニタ]ウィンドウを開く

[拡張モニタ]ウィンドウを開くには、[表示->CPU->拡張モニタ]を選択するか、[拡張モニタ]ツールバーボタンをクリックします。表示項目の更新間隔は、ユーザプログラム実行中は約 100ms、ブレーク中は約 1000ms です。



Item	Value
User Standby	Inactive
User NMI	Inactive
User Reset	Inactive
User Wait	Inactive
User System Voltage	OK
User System Voltage2	Down
User Cable	Not Connected
Running status	Break = Ready
ROM Write	No
Target Mode	7
Target Clock	No Clock
Target Sub Clock	No Clock

図 5-12 拡張モニタウィンドウ

5.4.2 表示項目を選択する

ポップアップメニューから[プロパティ...]を選択すると[拡張モニタコンフィギュレーション]ダイアログボックスを表示します。

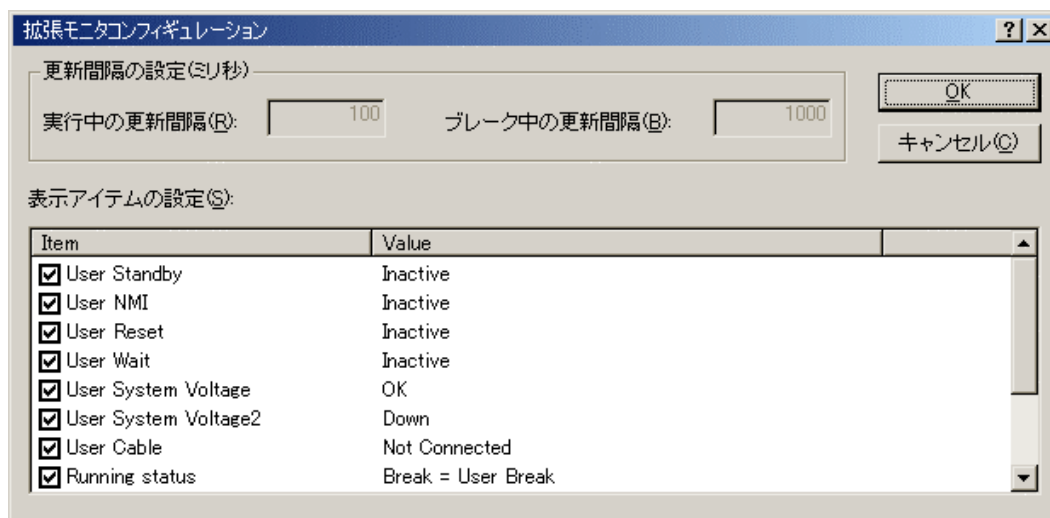


図 5-13 拡張モニタコンフィギュレーションダイアログボックス

[拡張モニタ]ウィンドウに表示する各項を設定できます。

【注】 本ウィンドウに表示する項目はご使用のエミュレータにより異なります。
 詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

5.5 リアルタイムにメモリ内容を表示する

ユーザプログラム実行中にメモリ内容をモニタするには[モニタ]ウィンドウを使用します。

モニタ機能は E6000 エミュレータのバスモニタ回路により MCU 内部のリード/ライト信号をトリガとしてアドレスバスおよびデータバスの値を保持し、該当するメモリの表示内容を更新するためリアルタイム性は損なわれません。


E6000 エミュレータのバスモニタ回路に実装されたモニタチャンネル(8 チャンネル)を使用し、最大 8 ポイント設定できます。

1 ポイントあたりのモニタ可能サイズは 1~256 バイトです。

各ポイントのモニタ範囲の一部または全部が重複する設定も可能です

【注】 MCU に内蔵のタイマカウンタなど、値の更新に MCU 内部のリード/ライト信号が発生しないエリアに対してはモニタできません。

5.5.1 [モニタ]ウィンドウを開く

[モニタ]ウィンドウを開くには、[表示->CPU->モニタ->モニタ設定...]を選択するか、[モニタ]ツールバーボタンをクリックして[Monitor Setting]ダイアログボックスを開きます。

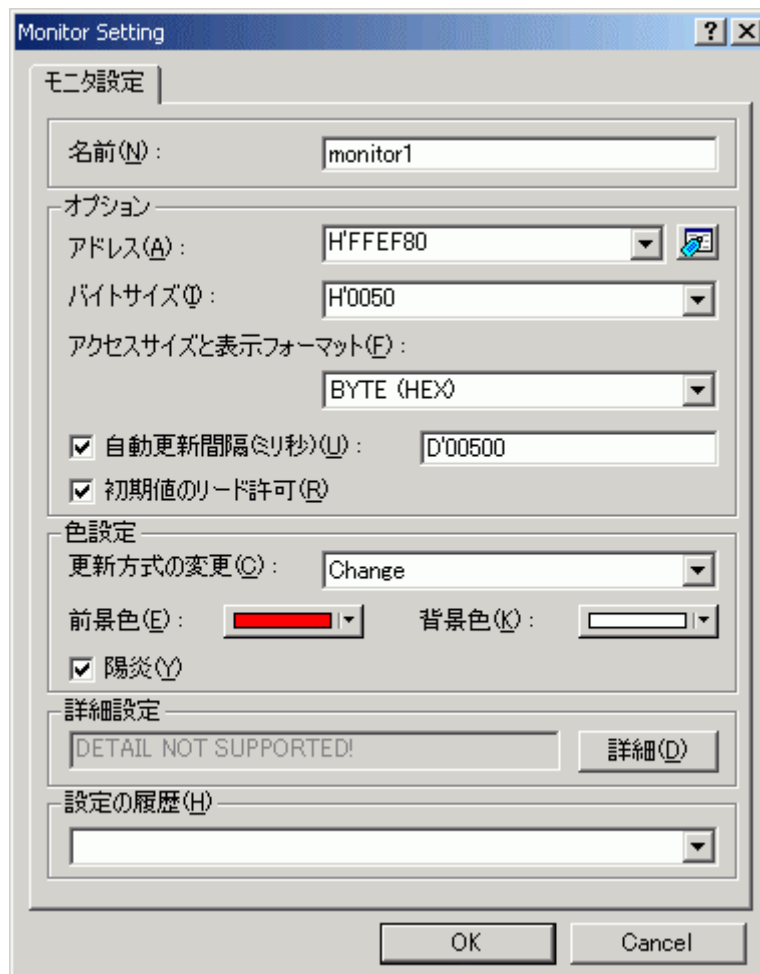


図 5-14 Monitor Setting ダイアログボックス

[名前]	モニタウインドウの名称を設定します。
[オプション]	モニタ条件を設定します。
[アドレス]	モニタを行う先頭アドレスを設定します。
[バイトサイズ]	モニタを行う範囲を設定します。
[アクセスサイズと表示フォーマット]	モニタウインドウに表示するアクセスサイズを設定します。
[自動更新間隔(ミリ秒)]	モニタ取得間隔を設定します。(最小値は 500ms)
[初期値のリード許可]	モニタウインドウ OPEN 時に、モニタ表示エリアの値をリードします。
[色設定]	モニタの更新方法および色属性を設定します
[更新方法の変更]	モニタ中に変更があった値をどのように表示するかを設定します。 ([初期値のリード許可]選択時有効)
	No change: 色の変更は行いません。
	Change: 色を変更します。 色は前景色オプション、背景色オプション、で設定します。
	Gray: 値の変更のないデータを灰色表示します。
	Appear: 値の変更があると表示します。変更なければ表示しません。
[前景色]	表示文字色を設定します。 ([Change]選択時有効)
[背景色]	背景色を設定します。 ([Change]選択時有効)
[陽炎]	チェックボックスにチェックがある場合、一定間隔更新のないデータの色を背景色オプションで設定した色に戻します。一定間隔とは、モニタ取得間隔の一回分です。 ([Change], [Gray], [Appear]選択時有効)
[詳細設定]	エミュレータ固有の項目を設定します。本エミュレータでは設定できません。
[設定の履歴]	前回の設定内容呼び出します。

- 【注】 1. 本エミュレータではモニタを行う先頭アドレスとして奇数アドレスは指定できません。
2. 前景色および背景色の設定はご使用のオペレーティングシステムにより使用できない場合があります。

設定完了後、[OK]ボタンをクリックすると[モニタ]ウインドウが開きます。

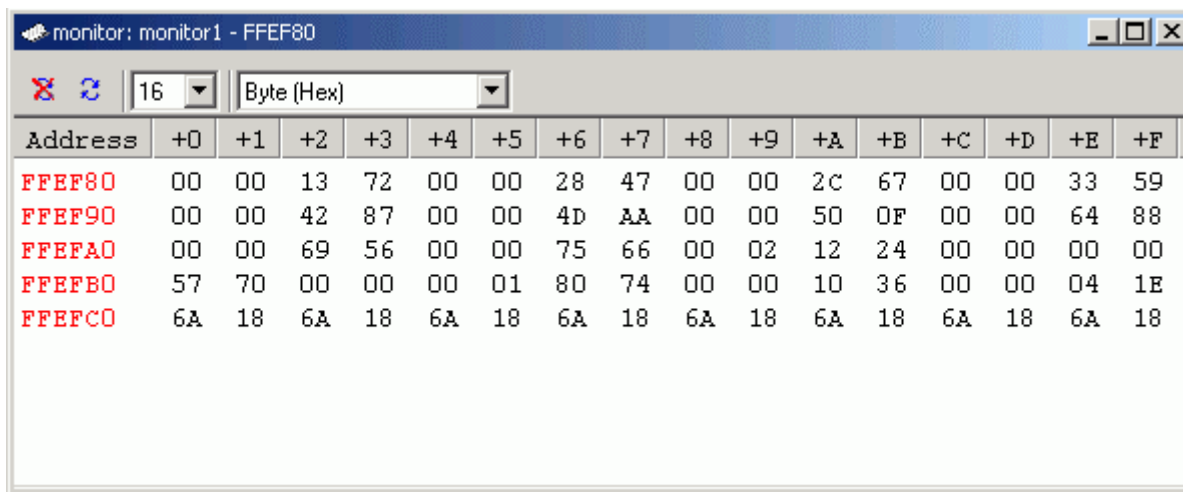


図 5-15 モニタウインドウ

ユーザプログラム実行中、自動更新間隔の設定値に応じて表示を更新します。

- 【注】 アドレス変更時またはメモリ内容変更時、データ内容が正しく表示されない場合は、ポップアップメニューより [最新の情報に更新]を選択してください。

5.5.2 モニタの設定内容を変更する

変更したい[モニタ]ウィンドウのポップアップメニューより[モニタ設定...]を選択すると、[Monitor Setting]ダイアログボックスが開き、設定内容を変更することができます。

また、ポップアップメニューの[色設定]メニューおよび[アクセスサイズと表示フォーマット]メニューより簡単に色設定およびアクセスサイズと表示フォーマットを変更できます。

5.5.3 モニタの更新を一時的に停止する

ユーザプログラム実行中、[モニタ]ウィンドウは設定した自動更新間隔にしたがって自動的に表示を更新します。表示更新を停止させたい[モニタ]ウィンドウのポップアップメニューより[表示固定]を選択してください。

アドレス部の表示文字が黒色となり、表示更新を停止します。

再びポップアップメニューより[表示固定]を選択することにより停止状態は解除できます。

5.5.4 モニタ設定を削除する

削除したい[モニタ]ウィンドウのポップアップメニューより[閉じる]を選択すると、[モニタ]ウィンドウを閉じ、モニタ設定を削除します。

5.5.5 変数の内容をモニタする

任意の変数の値を参照するには、ウォッチウィンドウを使用します。

[ウォッチ]ウィンドウに登録した変数のアドレスが、モニタ機能で設定したモニタ範囲に存在する場合、該当する変数の値をモニタ機能により更新し表示することができます。

この機能によりリアルタイム性を損なわずに変数の内容を確認できます。

5.5.6 モニタウィンドウを非表示にする

モニタ機能を使用し、[ウォッチ]ウィンドウより変数の値をモニタする場合、[モニタ]ウィンドウを非表示にしておくと画面を有効に活用できます。

現在設定しているモニタ情報は[表示->CPU->モニタ]のサブメニューとしてリストされます。

モニタ設定リストは[モニタ]ウィンドウ名およびモニタ開始アドレスで構成されています。

リストの左側にチェックがある場合は該当の[モニタ]ウィンドウが表示されていることを示します。

モニタ設定リストより非表示にしたい[モニタ]ウィンドウ項目を選択すると、該当の[モニタ]ウィンドウが非表示となり、リストの左側にあったチェックマークが消えます。

非表示にした[モニタ]ウィンドウを再び表示するにはモニタ設定リストより非表示にした[モニタ]ウィンドウ項目を選択してください。

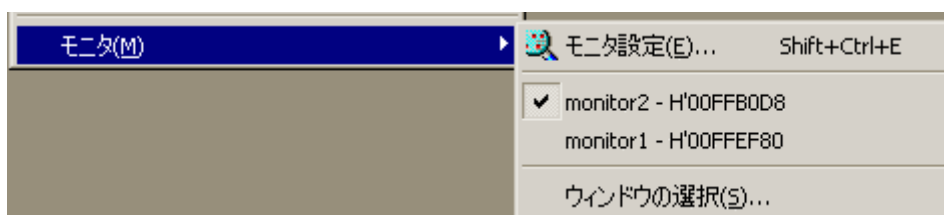


図 5-16 モニタ設定リスト

5.5.7 [モニタ]ウィンドウを管理する

[表示->CPU->モニタ->ウィンドウの選択...]を選択すると表示される、[ウィンドウの選択]ダイアログボックスより、現在設定されているモニタ条件の確認、新規モニタ条件の追加、編集、削除などの操作を連続的に行うことができます。

また、現在設定されているモニタ条件を複数選択することにより、更新の一時停止、非表示、削除を一括して操作できます。

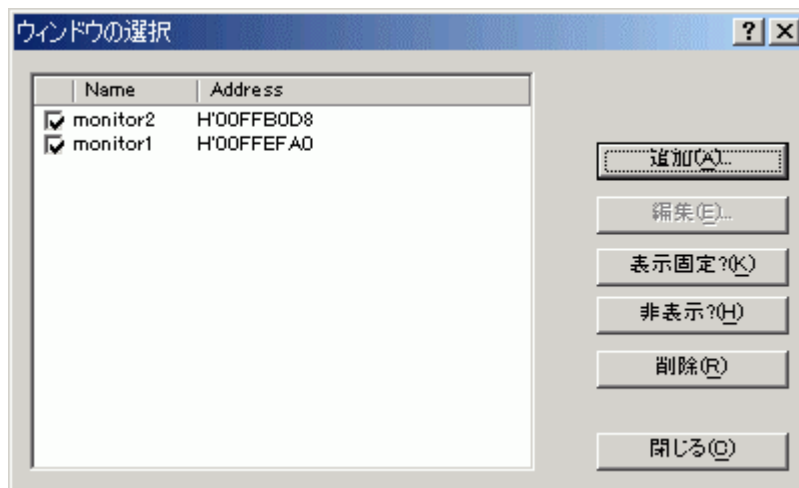


図 5-17 ウィンドウの選択ダイアログボックス

[追加]

新規にモニタ条件を追加します。

[編集]

選択している[モニタ]ウィンドウの設定を変更します。
(複数選択時無効)

[表示固定 / 表示固定解除]

選択している[モニタ]ウィンドウの表示を自動更新または更新停止にします。

[表示 / 非表示]

選択している[モニタ]ウィンドウを表示または非表示にします。

[削除]

選択しているモニタ条件を削除します。

[閉じる]

本ダイアログボックスを閉じます。

5.6 変数の表示

本節では、ソースプログラム上の変数の値を表示する方法について説明します。

5.6.1 ウォッチウィンドウ

[ウォッチ]ウィンドウを開くことにより、任意の変数について値を参照することができます。

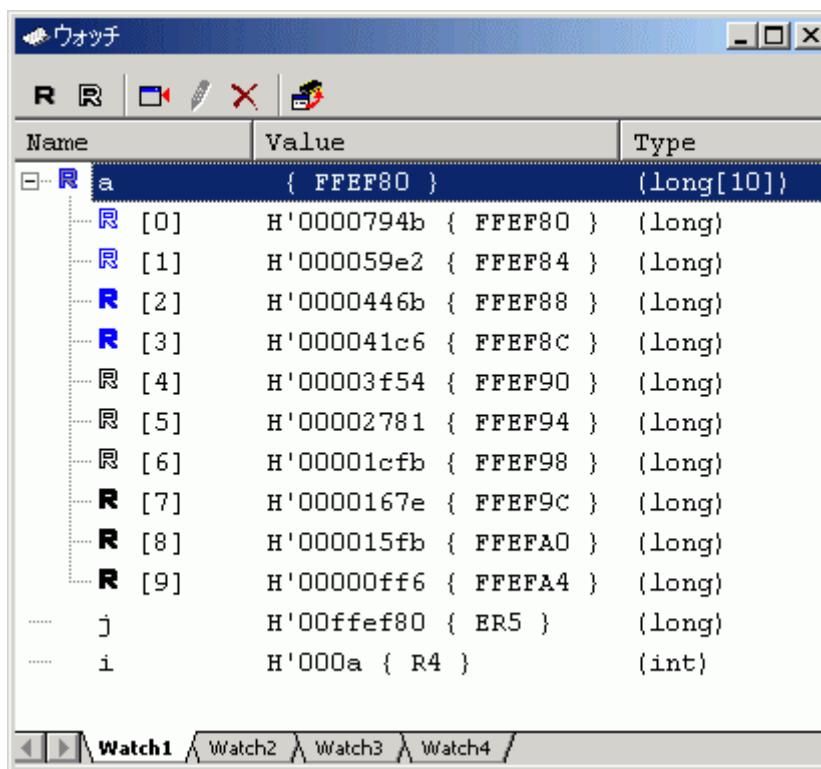


図 5-18 ウォッチウィンドウ

R マークのある変数は、ユーザプログラム実行中に更新できることを示します。

E6000 エミュレータでは[ウォッチ]ウィンドウに登録した変数の内容をユーザプログラム実行中に更新する実現手段として下記 3 種類の方法があります。

1. モニタ機能を使用して、ユーザプログラムを停止せずに値の更新を行う。
MCU内部のリード/ライト信号をトリガとしてアドレスバスおよびデータバスの値を保持し、該当する変数の値を更新します。

【注】 リアルタイム性は損なわれませんが、モニタ可能なサイズおよびポイントには限りがあります。
モニタ機能については「5.5 リアルタイムにメモリ内容を表示する」を参照してください。

2. エミュレータが強制的にバス権を確保することにより、ユーザプログラムを停止せずに、High-performance Embedded Workshopから直接メモリ内容を読み出し値の更新を行う。

【注】 エミュレータがバス権を所有している間、CPU は動作を停止するため、リアルタイム性は損なわれます。
内蔵 ROM、内蔵 RAM、エミュレーションメモリに対するアクセス時のみ適応されます。

ご使用のエミュレータにより、適応されるエリアが異なる、もしくは本アクセス方法が適応できない場合があります。

詳細については「8 本製品固有のソフトウェア仕様」またはオンラインヘルプの[Configuration Properties]ダイアログボックス[General]ページ[Enable read and write on the fly]項目をご参照ください。

3. ユーザプログラムを一時的に停止し、メモリ内容を読み出し値の更新を行う。

【注】 ユーザプログラムを一時的に停止するためリアルタイム性は損なわれます。
上記項番2に該当するエリア以外のエリア(内蔵 I/O、DTCRAM、ユーザメモリ)に対するアクセス時に適応されます。

R マークの色によって、ユーザプログラム実行中における値更新の実現手段が判別できます。

中抜き青色部	変数のアドレスがモニタ機能で設定されているモニタ範囲内で、モニタ機能を使用したデータリードが可能であることを示します。
青色部	モニタ機能を使用したデータリードによる値の更新を行うことを示します。
中抜き黒色部	変数のアドレスがモニタ機能で設定されているモニタ範囲外で、モニタ機能を使用したデータリードが不可能であることを示します。
黒色部	通常のデータリードにより値の更新を行うことを示します。

【注】 1. 本機能は変数ごと、また構造体であれば指定された構造体一括/要素ごとに設定できます。
2. R マークの色は、モニタ設定を変更したときに変わります。
3. レジスタに割り付けられている変数には設定することができません。

5.7 イベントポイントを使用する

E6000 エミュレータは High-performance Embedded Workshop 標準のソフトウェアブレークポイントとは別に、より高度な条件指定によるブレーク、トレース、実行時間測定を行うイベントポイント機能を持っています。

5.7.1 ソフトウェアブレークポイントとは

ソフトウェアブレークポイントは指定アドレスの命令フェッチが行われた場合にユーザプログラムの実行を停止します。

最大 256 ポイントまで設定できます。

ただし、ユーザシステム上に実装された ROM エリアには 1 ポイントのみソフトウェアブレークポイントを設定することが可能です。この特殊なソフトウェアブレークポイントのことを On Chip ブレークポイントと呼びます。

On Chip ブレークポイントは指定アドレスの命令実行後にユーザプログラムを停止します。

複数のソフトウェアブレークポイントをユーザシステム上に実装された ROM エリアに設定する必要がある場合は、このエリアをエミュレーションメモリに割り付け、コードをコピーして、ソフトウェアブレークポイントを設定してください。

5.7.2 イベントポイントとは

イベントポイントは単一アドレス指定以外に、データ条件など、より高度な条件指定が可能なポイントです。

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 ポイント設定できます。

条件成立時の動作としてユーザプログラムの停止以外に実行時間測定の開始/終了、トレース取得の開始/終了条件として利用可能です。

複数のイベントポイントを組み合わせることにより、より複雑な条件設定が可能です。

【注】 イベントポイントはデータの取得、条件の判定、action(ユーザプログラムの停止など)の実施を E6000 エミュレータのハードウェア回路にて行うため、条件成立から action の実施までに数サイクルの遅延が発生します。

5.7.3 イベント検出システムとは

イベントは 8 ポイントのイベントチャンネルと 4 ポイントの範囲チャンネルに割り当てられます。

イベントチャンネルには範囲チャンネル以上の機能があります(シーケンス、カウントなど)。

(1) イベントチャンネル(Ch1 ~ Ch8)

E6000 は 8 ポイントのイベントチャンネルを備えています。

イベントチャンネルは以下の組合せで定義できます。

- 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
- マスク条件指定付きデータ
- バス状態
- エリア
- 4 つの外部プローブ信号の値
- イベント発生回数
- イベント発生後のディレイサイクル数

また、最大 8 ポイントをシーケンスで組み合わせることができます。

それぞれの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。

(2) 範囲チャンネル(Ch9 ~ Ch12)

E6000 は 4 ポイントの範囲チャンネルを備えています。
範囲チャンネルは以下の組合せで定義できます。

- アドレスまたはアドレス範囲
- リード、ライトまたは両方
- マスク条件指定付きデータ
- バス状態
- エリア
- 4 つの外部プローブ信号の値
- イベント発生後のディレイサイクル数

5.7.4 バス状態およびエリア信号について

イベント検出システムではイベント検出の条件として MCU のバス状態およびアクセスしたエリアを示す信号を指定できます。

これらの信号はエミュレータに実装された MCU より出力されるため、ご使用のエミュレータにより取得可能な信号が異なります。

バス状態およびエリア信号はイベントポイントの[Bus/Area]条件設定で使用します。

これらの信号はトレース情報としても取得可能です。

また、バス状態信号についてはトレース非取得条件設定([Suppress]オプション)およびハードウェアパフォーマンス測定領域アクセス回数測定モード([Access Type]オプション)でも使用します。

トレース機能については「5.8 トレース情報を見る」を参照してください。

ハードウェアパフォーマンス機能については「5.9 パフォーマンスを測定する」を参照してください。

例として H8S/2633 E6000 エミュレータで取得可能なバス状態およびエリア信号を下記に示します。

表5-1 H8S/2633 E6000 エミュレータで取得可能なバス状態信号


バス状態	トレース表示(Status)	詳細
CPU Prefetch	PROG	CPU プリフェッチサイクル
CPU Data	DATA	CPU データアクセスサイクル
Refresh	REFRESH	リフレッシュサイクル
DMAC	DMAC	DMAC サイクル
DTC	DTC	DTC サイクル
Other	OTHER	その他

表5-2 H8S/2633 E6000 エミュレータで取得可能なエリア信号

エリア	トレース表示(Area)	詳細
On-chip ROM	ROM	ROM
On-chip RAM	RAM	RAM
On-chip I/O 16bit	I/O-16	16 ビット I/O
On-chip I/O 8bit	I/O-8	8 ビット I/O
External 16bit	EXT-16	16 ビット EXT(外部)
External 8bit	EXT-8	8 ビット EXT(外部)
DTC RAM	RAM/DTC	DTCRAM

【注】 取得可能なバス状態およびエリア信号はご使用のエミュレータにより異なります。
詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

5.7.5 [イベントポイント]ウィンドウを開く

[イベントポイント]ウィンドウを開くには、[表示->コード->イベントポイント]を選択するか、[イベントポイント]ツールバーボタンをクリックします。

[イベントポイント]ウィンドウには、3枚のシートがあります。

[Breakpoint]シート

ソフトウェアブレークポイントの設定内容を表示します。また、ソフトウェアブレークポイントの設定、変更および解除を行うことができます。

[Event]シート

イベントポイントを表示、設定します。

[Trigger]シート

トリガポイントを表示、設定します。

5.7.6 ソフトウェアブレークポイントを設定する

[Breakpoint]シートではソフトウェアブレークポイントの設定内容の表示、変更および追加ができます。

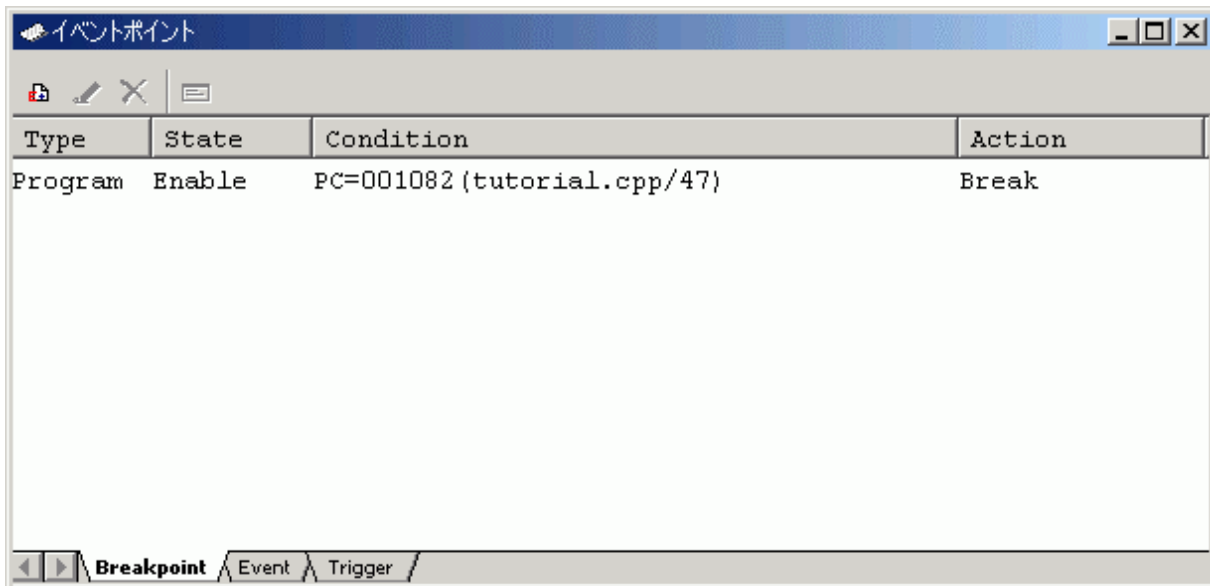


図 5-19 イベントポイントウィンドウ (Breakpoint シート)

ポップアップメニューから[追加...]を選択するか、または本ウィンドウに表示されているソフトウェアブレークポイントを選択しポップアップメニューから[編集...]を選択すると[Breakpoint/Event Properties]ダイアログボックスを表示します。

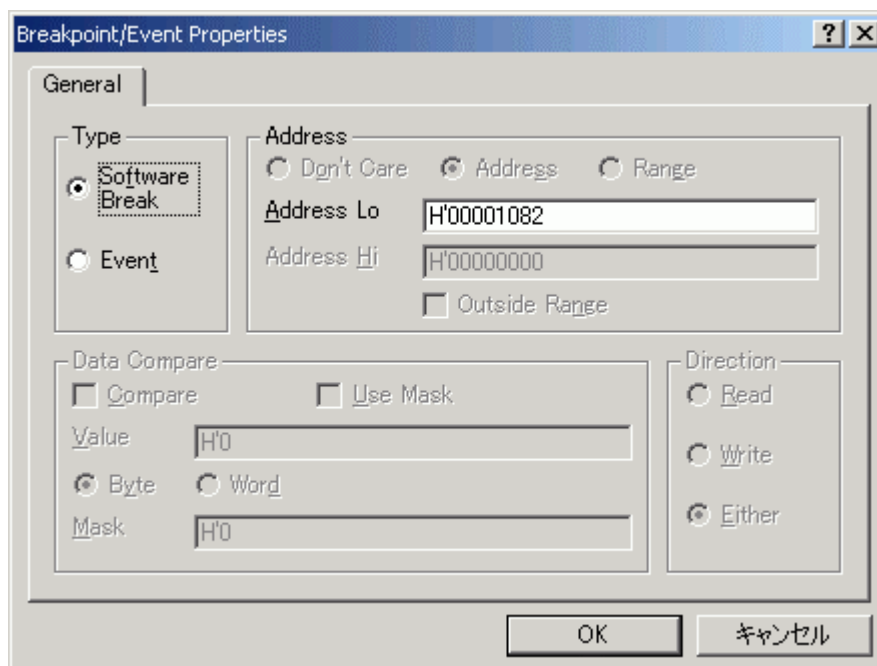


図 5-20 Breakpoint/Event Properties ダイアログボックス (Software Break 設定)

本ダイアログボックスよりソフトウェアブレークポイントを設定するアドレス条件を設定します。

[Type]	ブレークポイントタイプを指定します。 [Breakpoint/Event Properties]ダイアログボックスは、ソフトウェアブレークポイントの設定とイベントポイントの設定とで共用になっています。 指定されたタイプに従い、ダイアログボックスの他の設定可能なオプションが有効表示されます。表示されていないオプションについては設定できません。
[Software Break]	プログラムフェッチによる単一アドレスのみ指定できます。 他のオプションはすべて無効です。
[Event]	[General]ページの他のオプション、[Bus/Area]ページ、[Signals]ページまたは[Action]ページで詳細な検索条件を設定します。
[Address]	アドレス条件を設定します。
[Address Lo]	ソフトウェアブレークポイントを設定する単一アドレスを指定します。

5.7.7 イベントポイントを設定する

[Event]シートではイベントポイントの設定内容の表示、変更および追加ができます。

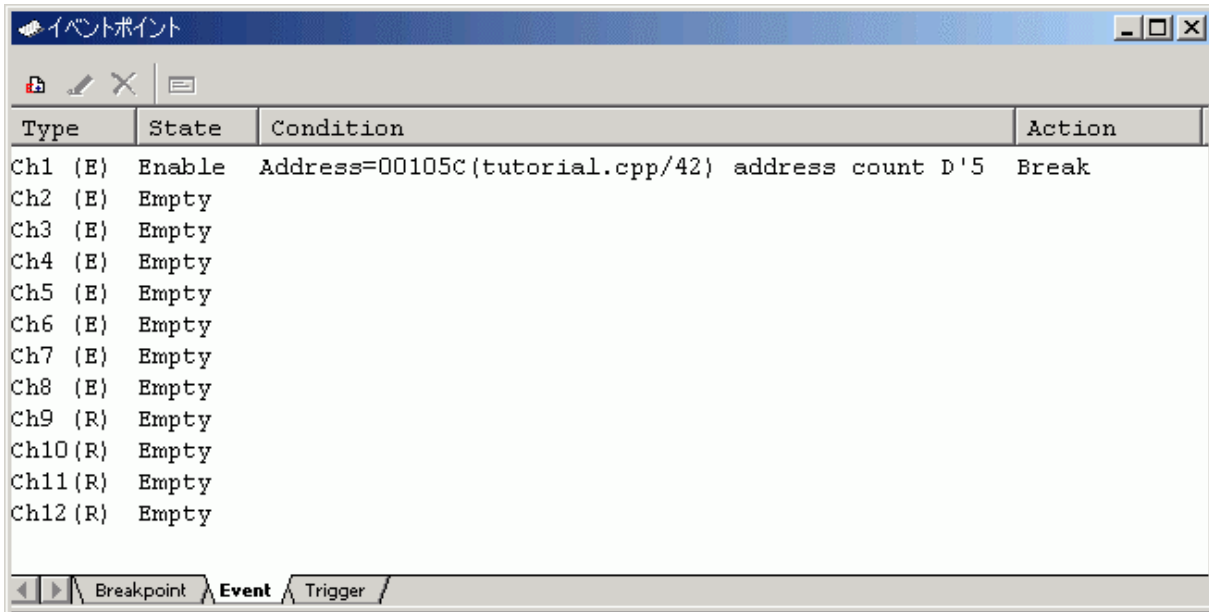


図 5-21 イベントポイントウィンドウ (Event シート)

ポップアップメニューから[追加...]を選択するか、または本ウィンドウに表示されているイベントポイントを選択しポップアップメニューから[編集...]を選択すると[Breakpoint/Event Properties]ダイアログボックスを表示します。

イベントポイントの条件設定は[General]ページ、[Bus/Area]ページ、[Signals]ページ、[Action]ページより構成されています。

各ページで設定した条件を複合したものが イベントポイントの検索条件を設定します。

- 【注】
1. チャンネル 8 はトリガ出力機能を備えています。
チャンネル 8 の条件が成立した場合、外部プローブ 1(EXT1)より 1 バスサイクルの期間だけ Low レベルを出力します。
 2. イベントポイントをトレース情報の取得条件として使用する場合はポップアップメニューより[Trace Acquisition...]を選択してください。
トレース機能については「5.8 トレース情報を見る」を参照してください。
 3. 範囲チャンネル(Ch9 ~ Ch12)編集時に範囲チャンネルでは使用できない条件を設定した場合、設定チャンネルは自動的に未使用のイベントチャンネル(Ch1 ~ Ch8)に変更されます。

表5-3 範囲チャンネルでは使用できない条件

条件	関連オプション
指定アドレス範囲外指定	[General]ページ[Outside Range]項目
実行時間計測開始/終了指定	[Action]ページ[Start Timer], [Stop Timer]項目
イベント発生回数指定 (2 回以上指定時)	[Action]ページ[Required number of event occurrences]項目
シーケンス指定	[Action]ページ[Enable Sequencing]項目

(1) General ページ

アドレス条件およびデータ条件を設定します。

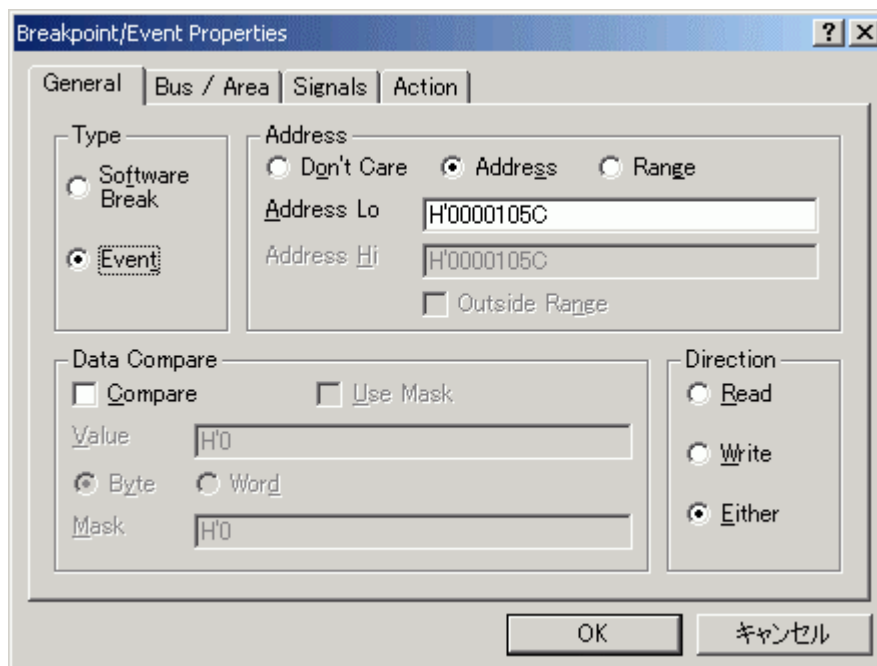


図 5-22 Breakpoint/Event Properties ダイアログボックス (General ページ)

[Type]	ブレイクポイントタイプを指定します。 [Breakpoint/Event Properties]ダイアログボックスは、ソフトウェアブレイクポイントの設定とイベントポイントの設定とで共有になっています。 指定されたタイプに従い、ダイアログボックスの他の設定可能なオプションが有効表示されます。表示されていないオプションについては設定できません。
[Software Break]	プログラムフェッチによる単一アドレスのみ指定できます。 他のオプションはすべて無効です。
[Event]	[General]ページの他のオプション、[Bus/Area]ページ、[Signals]ページまたは[Action]ページで詳細な検索条件を設定します。
[Address]	アドレス条件を設定します。
[Don't Care]	アドレス条件を設定しません。
[Address]	単一アドレスを指定します。
[Range]	範囲アドレスを指定します。
[Address Lo]	単一アドレスまたは範囲アドレスの開始アドレスを指定します。 ([Address/Range]選択時有効)
[Address Hi]	範囲アドレスの終了アドレスを指定します。 ([Range]選択時有効)
[Outside Range]	アドレス範囲を否定します。(例 範囲外のアドレス) ([Range]選択時有効)
[Data Compare]	データ条件を設定します。
[Compare]	チェックすると、データ比較を行います。
[Use Mask]	マスク条件を指定します。 ([Compare]選択時有効)
[Value]	データバスの値を数値で設定します。 また、データのアクセスサイズを選択します。 ([Compare]選択時有効)
[Byte]	バイトアクセスを条件にします。 ([Compare]選択時有効)
[Word]	ワードアクセスを条件にします。 ([Compare]選択時有効)
[Mask]	マスクする値を設定します。 マスク値を設定した場合、データバスの値およびデータ条件の値双方に対しマスク値で AND 演算を行い、その結果をもとにデータの比較を行います。 ([Use Mask]選択時有効)

5 デバッグ

[Direction]	リード、ライトサイクルの条件を選択します。
[Read]	リードサイクルを条件にします。
[Write]	ライトサイクルを条件にします。
[Either]	リード、ライト両方のサイクルを条件にします。

(2) Bus/Area ページ

バス状態およびメモリアクセスエリアを指定します。

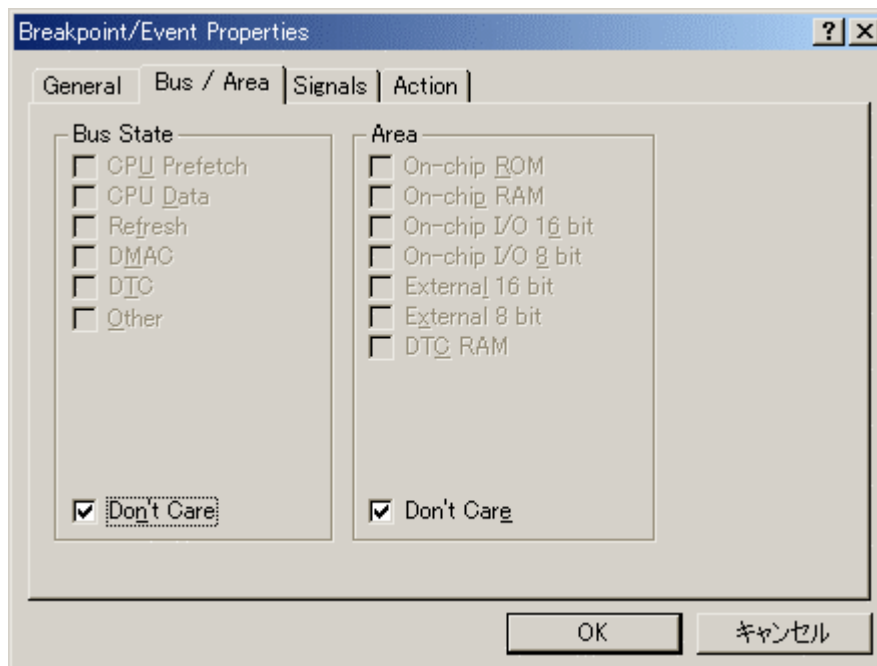


図 5-23 Breakpoint/Event Properties ダイアログボックス (Bus/Area ページ)

[Bus State]	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、すべてのバス状態でイベントが成立します。
[Area]	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、すべてのエリアでイベントが成立します。

【注】 バス状態およびメモリアクセスエリアの設定項目についてはご使用のエミュレータにより異なります。詳細につきましては「5.7.4 バス状態およびエリア信号について」をご参照ください。

- (3) Signals ページ
外部信号を指定します。

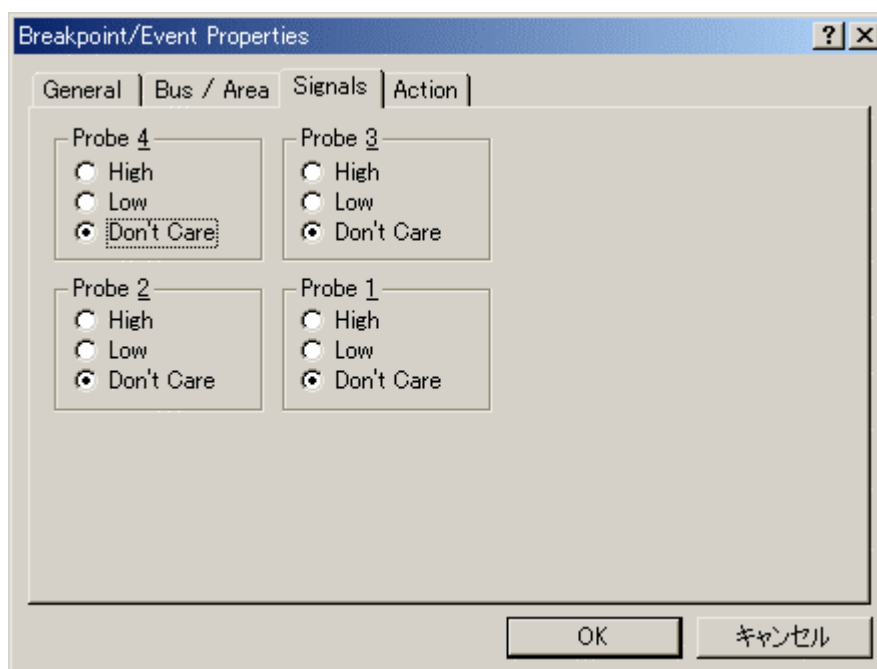


図 5-24 Breakpoint/Event Properties ダイアログボックス (Signals ページ)

[Probe4]	入力プローブ 4 の状態検出
[High]	入力プローブが High の状態を検出します。
[Low]	入力プローブが Low の状態を検出します。
[Don't Care]	入力プローブの状態を検出しません。
[Probe3]	入力プローブ 3 の状態検出
[High]	入力プローブが High の状態を検出します。
[Low]	入力プローブが Low の状態を検出します。
[Don't Care]	入力プローブの状態を検出しません。
[Probe2]	入力プローブ 2 の状態検出
[High]	入力プローブが High の状態を検出します。
[Low]	入力プローブが Low の状態を検出します。
[Don't Care]	入力プローブの状態を検出しません。
[Probe1]	入力プローブ 1 の状態検出
[High]	入力プローブが High の状態を検出します。
[Low]	入力プローブが Low の状態を検出します。
[Don't Care]	入力プローブの状態を検出しません。

5 デバッグ

(4) Action ページ

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。

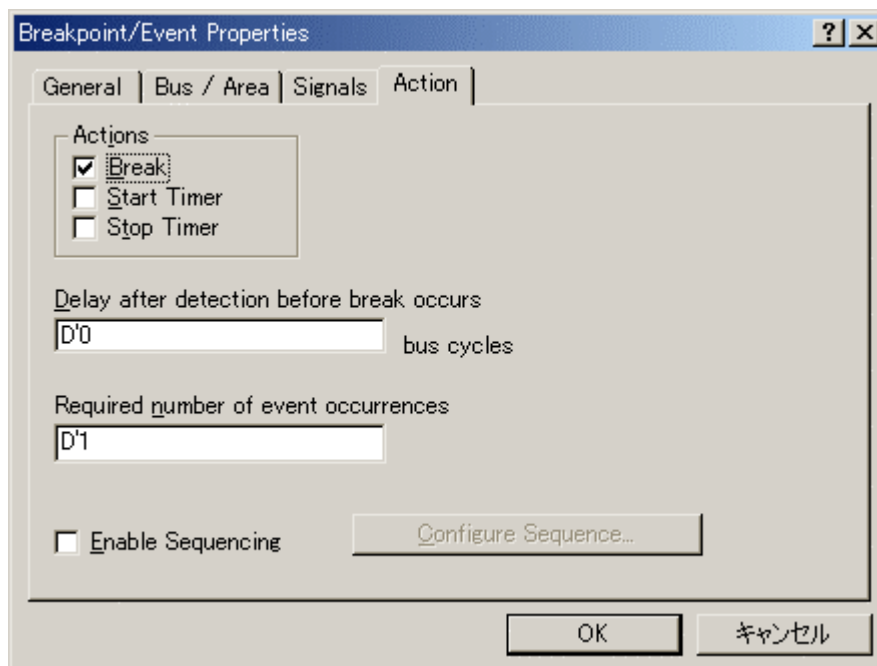


図 5-25 Breakpoint/Event Properties ダイアログボックス (Action ページ)

[Action]	イベント成立時の action を設定します。 トレース情報取得条件として使用するイベントポイントでは設定できません。
[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。
[Start Timer]	実行タイマを始動させます。(実行タイマ値は[ステータス]ウィンドウに表示されます。)
[Stop Timer]	実行タイマを停止させます。(実行タイマ値は[ステータス]ウィンドウに表示されます。)
[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利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。したがって、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。 パラメータの指定可能範囲は D'0 ~ D'65535 です。 ([Break]選択時有効)
[Required number of event occurrences]	16 ビットバ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。 パラメータの指定可能範囲は D'1 ~ D'65535 です。
[Enable Sequencing]	イベントを順位づけします(イベントチャンネルの設定が必要です)。
[Configure Sequence...]	[Event Sequencing]ダイアログボックスを表示し、イベントシーケンスを構成します。 ([Enable Sequencing]選択時有効)

(5) Event Sequencing ダイアログボックス

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、トレース情報取得条件で使用するイベントポイントのみ表示します。

[イベントポイント]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イベントのみ表示します。

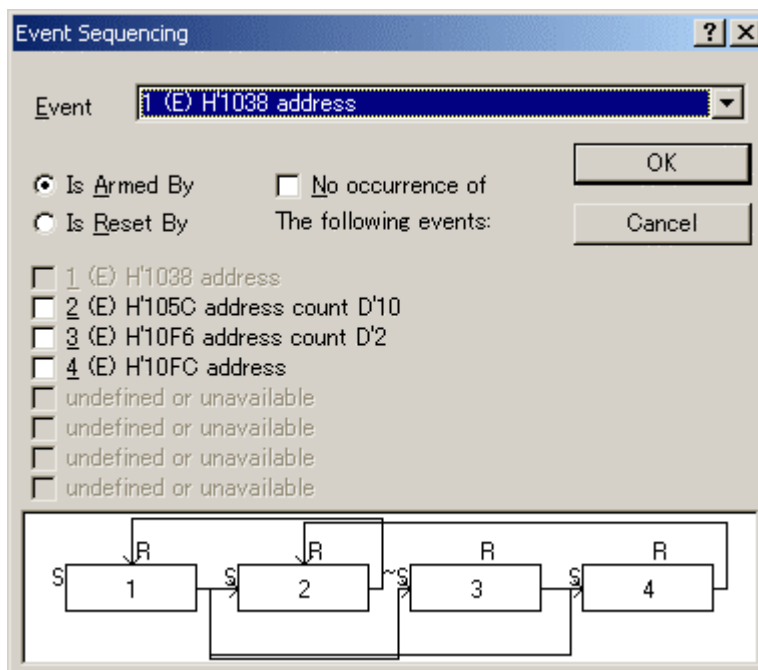


図 5-26 Event Sequencing ダイアログボックス

[Event]	設定の対象となるイベントポイントを選択します。
[Is Armed By]	イベントのシーケンス入力を設定します。
[Is Reset By]	イベントのリセット入力を設定します。
[No occurrence of]	イベントの非発生を条件にします。 ([Is Armed By]選択時有効)

イベントポイントの条件判定はユーザプログラムの実行と共に開始します。ユーザプログラム実行開始直後、イベントポイントは条件非成立状態です。イベントポイントの条件成立により設定された action を実施し条件成立状態に移ります。条件成立状態はユーザプログラムの停止またはイベントポイントのリセットが行われるまで維持されます。イベントポイントが条件成立状態の場合、再び条件が成立しても action は実施されません。条件成立により再度 action を実施したい場合はイベントポイントのリセットを行うことにより条件非成立状態に移らなければならない場合があります。ユーザプログラムの停止によりイベントポイントはすべて条件非成立状態に移ります。

あるイベントポイントの成立条件として別のイベントポイントが条件成立状態または条件非成立状態(No occurrence of 選択時)である必要がある場合、後者のイベントポイントを実用イベントと呼びます。イベントポイントは条件成立により他のイベントポイントまたは自分自身の条件判定状態をリセットできます。このようなイベントポイントを reset イベントと呼びます。reset イベントは条件成立状態のイベントポイント、条件非成立状態のイベントポイントにかかわらずイベントポイントのリセットを行います。(例: カウント途中のパスカウントをリセットするなど)。

[Event]ドロップダウンリストボックスより設定の対象となるイベントポイントを選択してください。選択したイベントポイントに対して arm イベントを設定する場合は、[Is Armed By]を選択し、該当のイベントに対応したチェックボックスをチェックしてください。また、[No occurrence of]チェックボックスをチェックすることにより arm イベントの条件非成立状態を条件とすることができます。選択したイベントポイントに対して reset イベントを設定する場合は、[Is Reset By]を選択し、該当のイベントに対応したチェックボックスをチェックしてください。

画面の下側にイベントの順序付けを示す図があります。(図 5-26)

5 デバッグ

S はシーケンス入力(arm イベント条件成立状態)、R はリセット入力(reset イベント)を示します。~S は非発生のシーケンス入力(arm イベント条件非成立状態)を示します。

図 5-69 を例にすると、Ch1 は Ch2、Ch3、Ch4 に対する arm イベントとなります。Ch3 は Ch4 に対する arm イベントとなります。Ch2 は Ch1 に対する reset イベントとなります。Ch4 は Ch2 に対する reset イベントとなります。

arm イベントを持つイベントポイントの条件成立には、arm イベントが条件成立状態または条件非成立状態(No occurrence of 選択時)である必要があります。

一つのイベントポイントに対し複数の arm イベントが存在する場合、イベントポイントの条件成立には、いずれか一つの arm イベントが条件成立状態または条件非成立状態(No occurrence of 選択時)である必要があります。

一つのイベントポイントに対して arm イベントの条件は、条件成立状態または条件非成立状態のいずれか一方のみ設定できます。

reset イベントによるイベントポイントのリセットは、reset イベントの条件成立により実施されます。

reset イベントとなるイベントポイントが条件成立状態の場合、再び条件が成立してもイベントポイントのリセットは実施されません。

一つのイベントポイントに対し複数の reset イベントが存在する場合、いずれか一つの reset イベントが条件成立した場合にイベントポイントのリセットが実施されます。

5.7.8 トリガポイントを設定する

トリガポイントとは指定アドレスに対してアクセスが行われた際にトリガ出力を行うイベントです。

トリガポイントは E6000 エミュレータのバスモニタ回路に実装されたトリガ出力(4 チャンネル)を使用し、最大 4 ポイント設定できます。

[Trigger]シートではトリガポイントの設定内容の表示および変更ができます。

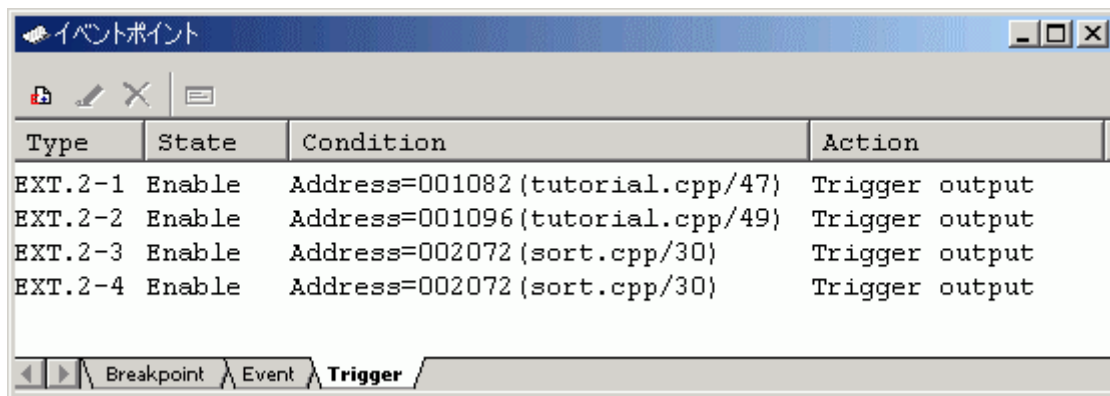


図 5-27 イベントポイントウィンドウ (Trigger シート)

ポップアップメニューから[追加...]を選択するか、または本ウィンドウに表示されているイベントポイントを選択しポップアップメニューから[編集...]を選択すると[Set Address For Trigger]ダイアログボックスを表示します。

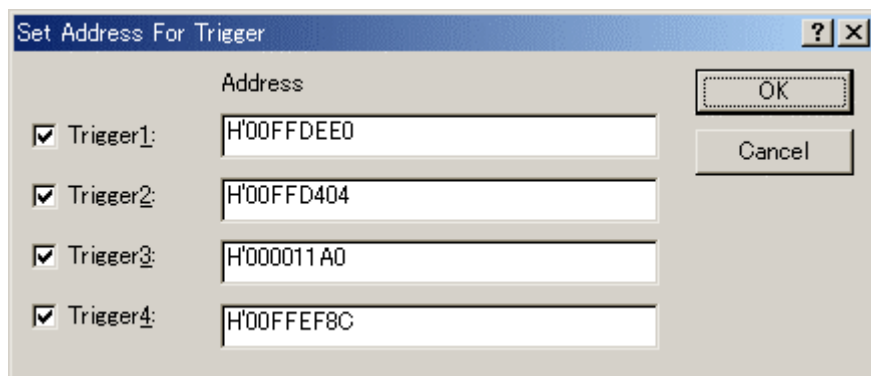


図 5-28 Set Address For Trigger ダイアログボックス

ユーザプログラム実行中のトリガ出力条件として、アクセスするアドレスを指定します。

トリガ出力ポイントの有効/無効は画面左側のチェックボックスで選択してください。

[Trigger1]	トリガチャンネル1の出力を有効にします。
[Trigger2]	トリガチャンネル2の出力を有効にします。
[Trigger3]	トリガチャンネル3の出力を有効にします。
[Trigger4]	トリガチャンネル4の出力を有効にします。
[Address]	該当チャンネルのアドレス条件を設定します。

- 【注】1. トリガ出力はトリガ設定(1~4)の条件が一致した場合に外部プローブ2(EXT2)の該当ピン(1~4)よりリード/ライトの期間だけ High レベルを出力します。
 2. ご使用のエミュレータによりトリガポイントをサポートしていない場合があります。
 詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

5.7.9 イベントポイントの編集

ソフトウェアブレークポイント、イベントポイント、トリガポイントに対する設定以外の操作方法はすべて共通となっています。

以下イベントポイントを例に設定以外の操作方法について説明します。

5.7.10 イベントポイントの設定内容を変更する

変更したいイベントポイントを選択後ポップアップメニューから[編集...]を選択すると、各イベントに対応した設定ダイアログボックスが開き、設定内容を変更することができます。[編集...]メニューはイベントポイントを1個選択しているときのみ有効となります。

5.7.11 イベントポイントを有効にする

イベントポイントを選択後ポップアップメニューから[有効]を選択すると、選択しているイベントポイントを有効にします。

5.7.12 イベントポイントを無効にする

イベントポイントを選択後ポップアップメニューから[無効]を選択すると、選択しているイベントポイントを無効にします。無効にした場合は、イベントポイントはリストには残りますが、指定した条件が一致してもイベントは発生しません。

5.7.13 イベントポイントを削除する

イベントポイントを選択後ポップアップメニューから[削除]を選択すると、選択しているイベントポイントを削除します。イベントポイントを削除しないで、詳細情報は保持したまま、条件が成立してもイベントを発生させないようにするには、[無効]オプションを使用します(「5.7.12 イベントポイントを無効にする」参照)。

【注】 トリガポイントは削除できません。設定を解除する場合は[無効]オプションを使用してください。

5.7.14 イベントポイントをすべて削除する

ポップアップメニューから[すべて削除]を選択すると、すべてのイベントポイントを削除します。

【注】 トリガポイントは削除できません。[すべて削除]を選択した場合、全チャンネルの設定が無効となります。

5.7.15 イベントポイントのソース行を表示する

イベントポイントを選択後ポップアップメニューから[ソースを表示]を選択すると、ブレークポイントのある[エディタ]または[逆アセンブリ]ウィンドウをオープンします。[ソースを表示]メニューは対応するソースファイルを持つイベントポイントを1個選択しているときのみ有効となります。

5.8 トレース情報を見る


E6000 エミュレータでは命令の実行結果をトレース情報としてトレースバッファに取得し、表示することができます。

トレース情報は[Trace]ウィンドウに表示します。トレース情報の取得条件は、[Trace Acquisition]ダイアログボックスで設定します。

トレース機能は E6000 エミュレータのハードウェア回路によりバスサイクル単位にトレース情報を取得し、トレースバッファに保持するためリアルタイム性は損なわれません。

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、バッファがいっぱいになった場合は取得したトレース情報のうち最も古いデータを上書きすることによりトレース情報を取得し続けます。

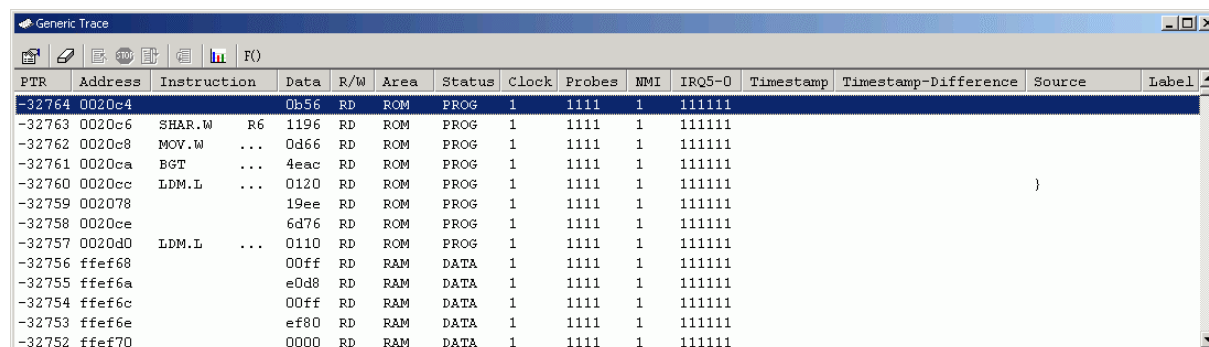
5.8.1 [Trace]ウィンドウを開く

[Trace]ウィンドウを開くには、[表示->コード->トレース]を選択するか、[トレース]ツールバーボタンをクリックします。

5.8.2 トレース情報を取得する

E6000 エミュレータはトレース情報の取得条件を設定しない場合、デフォルトで無条件に全バスサイクルをトレース取得します。(フリートレースモード)

フリートレースモードではユーザプログラムの実行開始と共にトレース取得を開始し、ユーザプログラムの停止によりトレース取得を停止します。取得したトレース情報は[Trace]ウィンドウに表示します。



PTR	Address	Instruction	Data	R/W	Area	Status	Clock	Probes	NMI	IRQ5-0	Timestamp	Timestamp-Difference	Source	Label
-32764	0020c4		0b56	RD	ROM	PROG	1	1111	1	111111				
-32763	0020c6	SHAR.W	R6	1196	RD	ROM	PROG	1	1111	1	111111			
-32762	0020c8	MOV.W	...	0d66	RD	ROM	PROG	1	1111	1	111111			
-32761	0020ca	BGT	...	4eac	RD	ROM	PROG	1	1111	1	111111			
-32760	0020cc	LDM.L	...	0120	RD	ROM	PROG	1	1111	1	111111			}
-32759	002078		19ee	RD	ROM	PROG	1	1111	1	111111				
-32758	0020ce		6d76	RD	ROM	PROG	1	1111	1	111111				
-32757	0020d0	LDM.L	...	0110	RD	ROM	PROG	1	1111	1	111111			
-32756	ffef68		00ff	RD	RAM	DATA	1	1111	1	111111				
-32755	ffef6a		e0d8	RD	RAM	DATA	1	1111	1	111111				
-32754	ffef6c		00ff	RD	RAM	DATA	1	1111	1	111111				
-32753	ffef6e		ef80	RD	RAM	DATA	1	1111	1	111111				
-32752	ffef70		0000	RD	RAM	DATA	1	1111	1	111111				

図 5-29 Trace ウィンドウ

表示する項目は以下の通りです。

[PTR]	トレースバッファ内のサイクル番号。 最後に取得されたサイクルの番号を0とし、古いサイクルにさかのぼって、順に-1、-2と番号が小さくなります。 ディレイカウントが設定されている場合は、トレース停止条件が成立したサイクル番号を0とし、成立後停止するまでに実行されたサイクル(ディレイ期間中のサイクル)には、最後に取得されたサイクルに向かって順に+1、+2と番号が大きくなります。
[Address]	アドレスバスの値(6桁の16進数)。
[Instruction]	実行された命令の逆アセンブルコード。
[Data]	データバスの値。 それぞれ16進の2桁、4桁で表示します。
[R/W]	アクセスサイクルの種類。読み出しはRD、書き込みはWRと表示します。
[Area]	アクセスしたエリア: ROM, RAM, 8 または 16 ビット I/O, 8 または 16 ビット EXT(外部), DTCRAM (タイムスタンプ取得時は無効)
[Status]	バス状態: DTC 動作, PROG(プリフェッチ), Data(CPU データアクセスサイクル), Refresh(リフレッシュサイクル), DMAC(DMAC サイクル) (タイムスタンプ取得時は無効)
[Clock]	バスサイクルのクロック数 1~8 を表示します。それ以上の場合は OVR を表示します。 (タイムスタンプ取得時は無効)
[Probes]	4本のプローブ信号の状態(2進数4桁)。 左から Probe4、Probe3、Probe2、Probe1 の順で表示します。 (タイムスタンプ取得時は無効)
[NMI]	NMI 入力の状態。 (タイムスタンプ取得時は無効)
[IRQ7-0]	8本のIRQ入力の状態。 (タイムスタンプ取得時は無効)
[Timestamp]	バスサイクルのタイムスタンプ。 タイムスタンプは、ユーザプログラム実行を開始するたびに0からカウントを始めます。測定時間の分解能は、Trace Acquisition で設定してください。 (タイムスタンプ取得時のみ有効)
[Source]	ソースコードの表示。
[Label]	アドレスに対応するラベル(ラベルが設定されている場合のみ表示します)。
[Timestamp-Difference]	前の行とのタイムスタンプの差分時間を表示。 (タイムスタンプ取得時のみ有効)

【注】 [PTR], [Address], [Instruction], [Data], [R/W], [Area], [Status], [Probes], [Timestamp], [Source], [Label], [Timestamp-Difference]以外の項目はご使用のエミュレータにより異なります。
詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

[Trace]ウィンドウ内の不要なカラムは非表示にすることができます。

カラムを非表示にする場合はヘッダカラム上で右クリックすると表示されるポップアップメニューより非表示にしたいカラムを選択してください。

カラムを再表示する場合は再度ポップアップメニューより該当のカラムを選択してください。

5.8.3 トレース情報取得条件を設定する

トレースバッファは有限であるため、バッファがいっぱいになった場合は最も古いトレース情報から順に上書きします。トレース情報の取得条件を設定することにより、有用なトレース情報のみを取得し、トレースバッファを有効に活用することができます。

トレース情報の取得条件はイベントポイントにて実現し、トレース情報取得の開始、停止、終了を制御します。イベントポイントについては「5.7 イベントポイントを使用する」を参照してください。

トレース情報の取得条件はポップアップメニューから[設定...]を選択すると表示される[Trace Acquisition]ダイアログボックスで設定します。

[Trace Acquisition]ダイアログボックスは下記ページより構成されています。

表5-4 [Trace Acquisition]ダイアログボックスのページ構成

ページ	設定項目
[General]	トレース情報の取得条件を設定します。
[Stop]	トレース停止条件を設定します。(ディレイ設定なし)
[Delayed Stop]	トレース停止条件を設定します。(ディレイ設定あり)
[1]~[4]	範囲トレースを設定します。 (フリートレースモード無効時のみ有効)

(1) General ページ

トレース情報の取得条件を設定します。

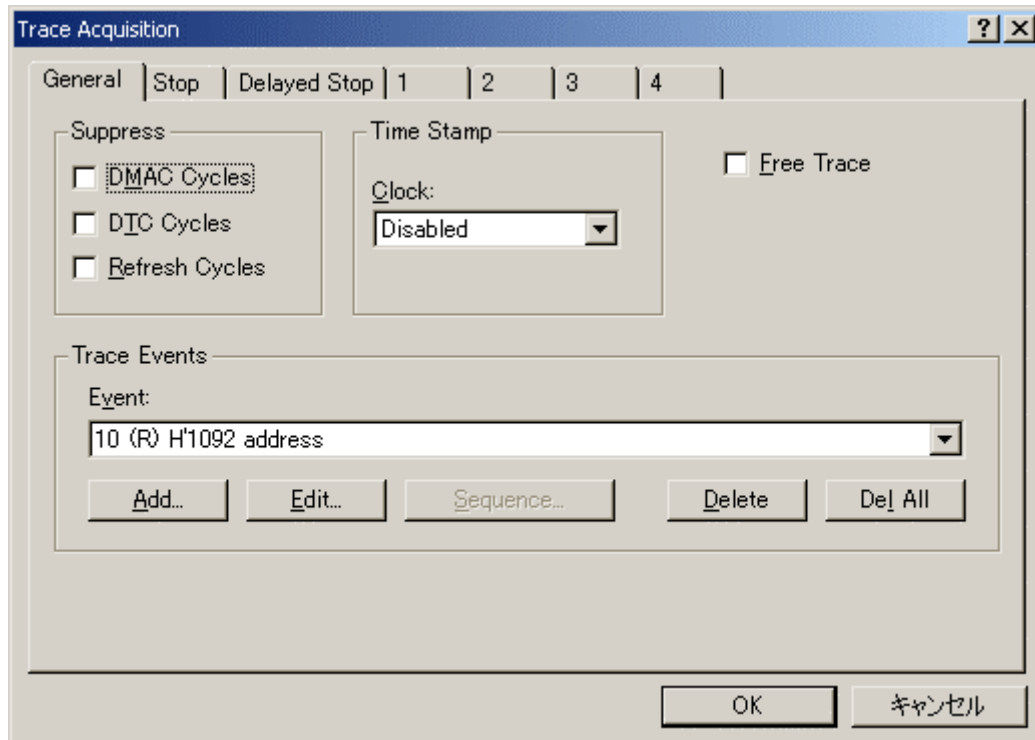


図 5-30 Trace Acquisition ダイアログボックス(General ページ)

[Suppress]	指定したバスサイクルタイプのトレース情報を取得しません。
[Time Stamp]	タイムスタンプの条件を設定します。
[Clock]	タイムスタンプの分解能を設定します。 分解能は以下のいずれかから選択できます。 Disabled, 125ns, 250ns, 500ns, 1us, 2us, 4us, 8us, 16us, 100us タイムスタンプは 32 ビットのカウンタで構成されています。 測定可能な最大時間は分解能 125ns で約 9 分、分解能 100us で約 5 日です。 カウンタがオーバーフローした場合、カウンタをクリアし、カウントアップを続けます。 Disabled を選択した場合はタイムスタンプ情報を取得しません。
[Free Trace]	チェックするとフリートレースモードを有効にします。 フリートレースモード有効の場合 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ではトレース停止条件の設定のみ可能です。範囲トレースは利用不可となり、4 つの範囲トレースページ(1~4)は無効になります。 フリートレースモード無効の場合 トレース取得の開始および停止条件を設定できます。

[Trace Events]	トレース情報取得条件で使用するイベントポイントを設定します。
[Event]	トレース情報取得条件を設定したイベントポイントのリストです。
[Add...]	新しいイベントポイントを追加します。
[Edit...]	Event で選択したイベントポイントの設定を変更します。
[Sequence...]	トレース情報取得条件で使用するイベントポイントに対しイベントシーケンスを構成します。先にイベントを設定していないと、シーケンスのセットアップはできません。
[Delete]	Event で選択したイベントポイントを削除します。
[Del All]	すべてのイベントポイントを削除します。

- 【注】
1. [Suppress]オプションで指定可能なバスサイクルはご使用のエミュレータにより異なります。詳細につきましては「5.7.4 バス状態およびエリア信号について」をご参照ください。
 2. タイムスタンプ情報は一部のトレース情報と共用でトレースバッファを使用します。したがって、タイムスタンプ取得時は PTR, Address, Instruction, Data, R/W, Source, Label, Timestamp および Timestamp-Difference 以外のトレース情報は取得できません。
 3. 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。

(2) Stop ページ

トレース停止条件を設定します。

トレース停止条件はディレイの有無にかかわらず、両方向同時に指定できます。

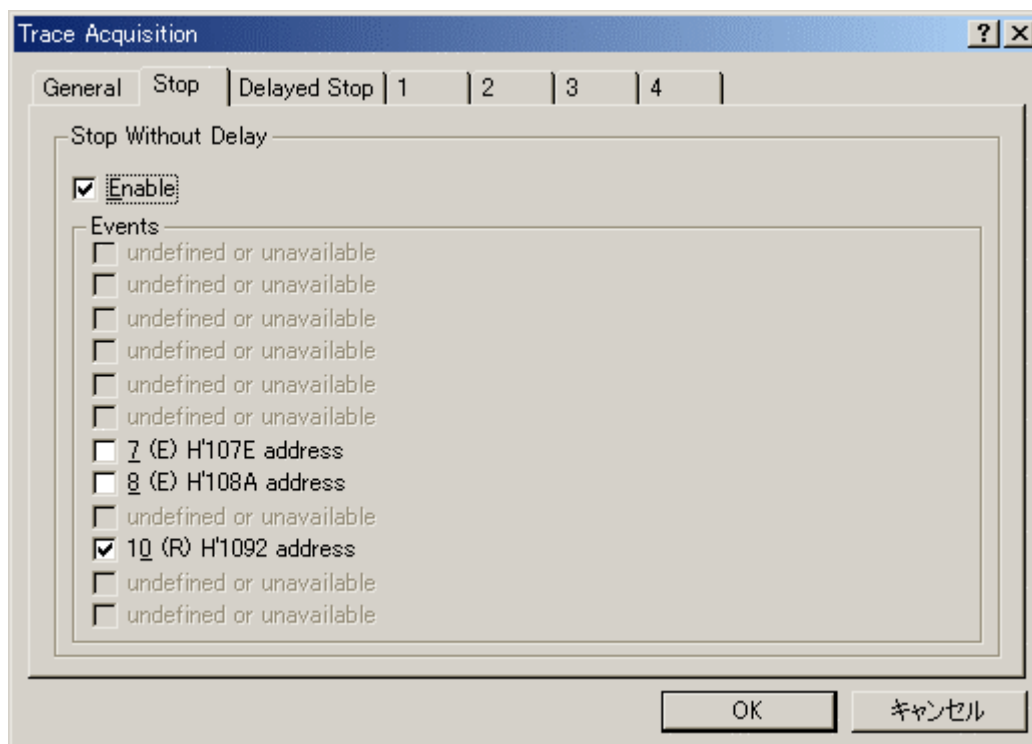


図 5-31 Trace Acquisition ダイアログボックス(Stop ページ)

[Stop Without Delay]	トレース停止条件を設定します。
[Enable]	チェックするとトレース停止を有効にします。
[Events]	トレース情報取得条件を設定したイベントポイントを表示します。イベントポイントに対応したチェックボックスをチェックした場合、該当のイベントが成立したときにトレース情報の取得を停止します。 ([Enable]選択時有効)

5 デバッグ

(3) Delayed Stop ページ

トレース停止条件を設定します。

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。

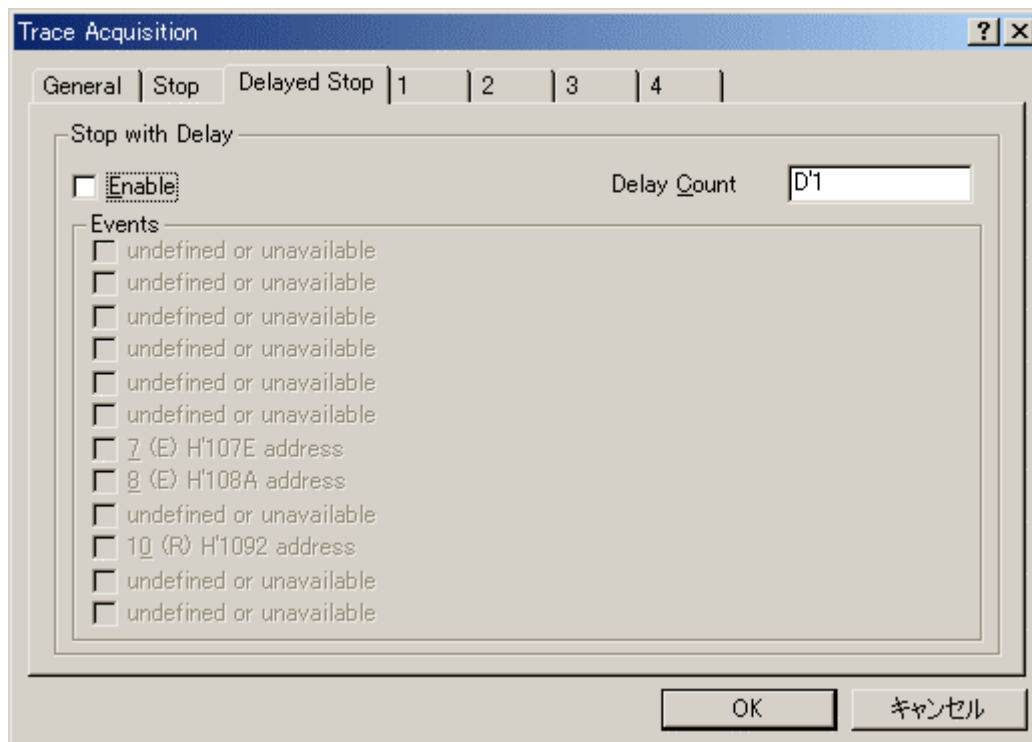


図 5-32 Trace Acquisition ダイアログボックス(Delayed Stop ページ)

[Stop With Delay]	トレース停止条件を設定します。
[Enable]	チェックするとトレース停止を有効にします。
[Delay Count]	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。
[Events]	トレース情報取得条件を設定したイベントポイントを表示します。 イベントポイントに対応したチェックボックスをチェックした場合、該当のイベントが成立したときにトレース情報の取得を停止します。 ([Enable]選択時有効)

(4) 1, 2, 3, 4 ページ

範囲トレースを設定します。フリートレースモード無効時のみ設定可能です。

範囲トレースは[Disabled], [Point to Point], [Range], [Event]の4つのモードを選択できます。

- Disabled

範囲トレースを無効にします。

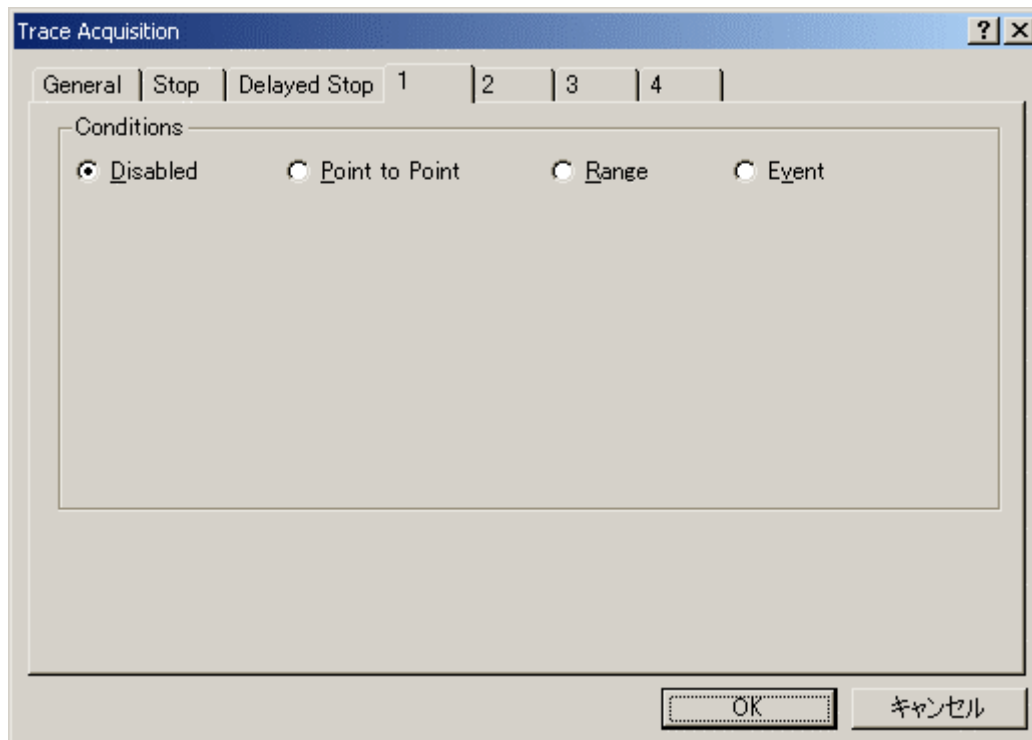


図 5-33 範囲トレース設定(Disabled)

5 デバッグ

・ Point to Point

指定アドレス範囲のトレース情報を取得します。

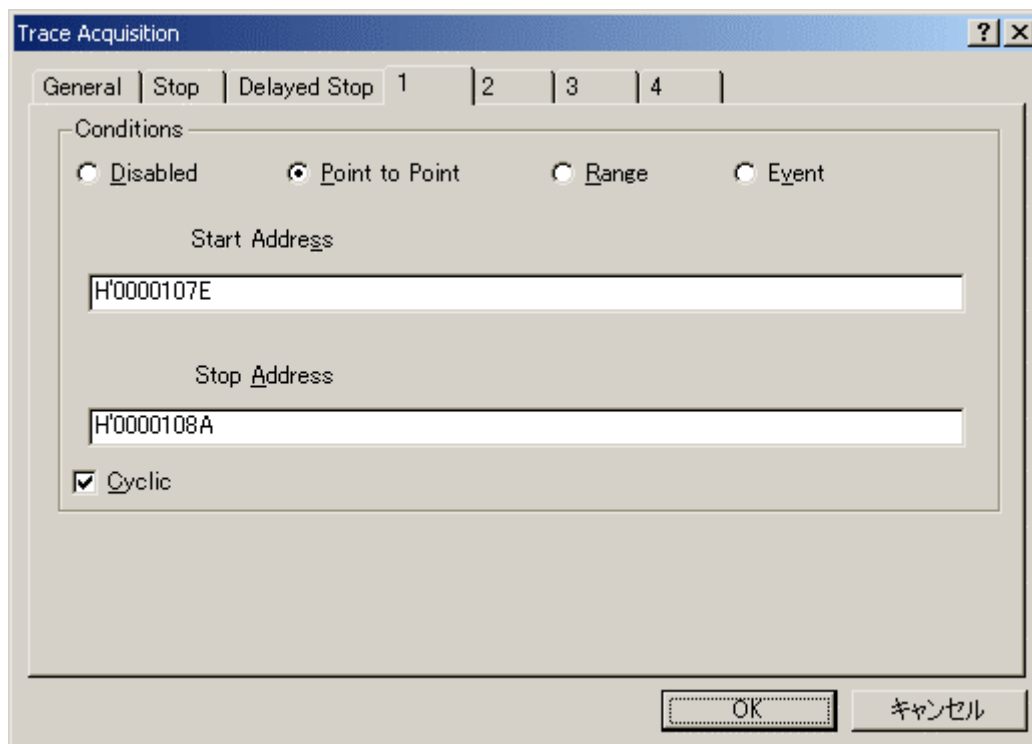


図 5-34 範囲トレース設定(Point to Point)

- [Start Address] トレース情報取得を開始するアドレスを指定します。
[Stop Address] トレース情報取得を停止するアドレスを指定します。
[Cyclic] チェックするとイベントシーケンスを構成し、イベント自身がりセットになり、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。

開始アドレスをアクセスしたときにトレース情報の取得を開始し、停止アドレスをアクセスしたときにトレース情報の取得を停止するために必要なイベントポイントを設定します。

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。

[Cyclic]オプションを設定することにより指定アドレス範囲のトレース情報のみを取得し続けることができます。

- 【注】** 本機能はイベントポイントのシーケンスを自動的に構成しますが、予期しない結果が起きる場合もあります。
この場合、[Event Sequencing]ダイアログボックスにてシーケンス設定を修正してください。

• Range

指定した条件に合うトレース情報のみを取得します。

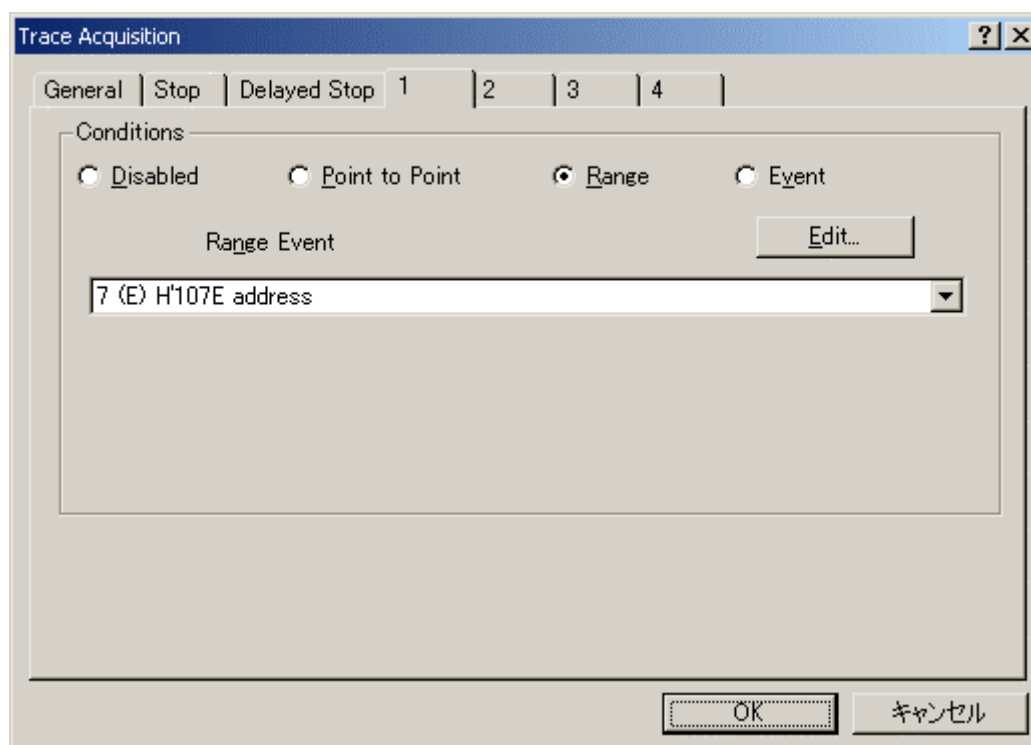


図 5-35 範囲トレース設定(Range)

[Range Event] トレース情報の取得条件を設定したイベントポイントを指定します。
[Edit...] 指定するイベントポイントの設定を変更します。

すべてのバスサイクルから選択したイベントの条件に合うトレース情報のみを取得します。
このモードは1つのイベントチャンネルまたは範囲チャンネルを使用します。

5 デバッグ

・Event

指定した条件によりトレース情報取得の開始と終了を制御しトレース情報を取得します。

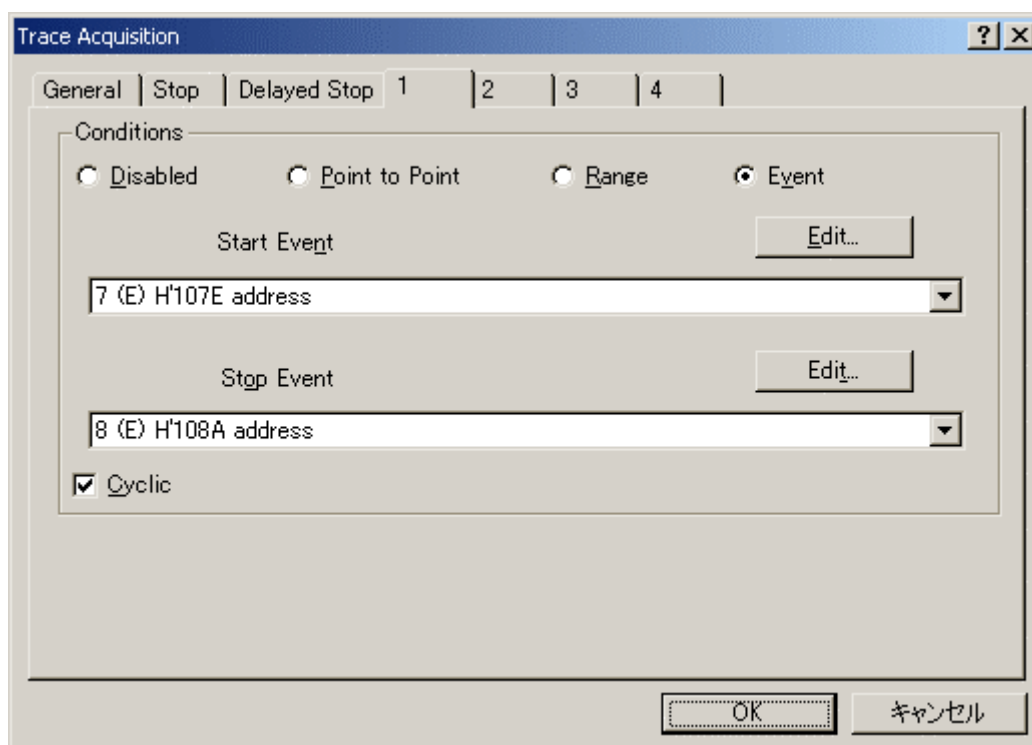


図 5-36 範囲トレース設定(Event)

[Start Event]	トレース情報取得を開始する条件を設定したイベントポイントを指定します。
[Stop Event]	トレース情報取得を停止する条件を設定したイベントポイントを指定します。
[Edit...]	指定するイベントポイントの設定を変更します。
[Cyclic]	チェックするとイベントシーケンスを構成し、イベント自身がりセットになり、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。

開始条件が成立したときにトレース情報の取得を開始し、停止条件が成立したときにトレース情報の取得を停止します。

[Cyclic]オプションを設定することにより指定条件により取得可能なトレース情報のみを取得し続けることができます。

5.8.4 Trace レコードを検索する

トレースレコードを検索するには[Trace Find]ダイアログボックスを使用します。
[Trace Find]ダイアログボックスを開くには、ポップアップメニューの[検索...]を選択します。

[Trace Find]ダイアログボックスは下記ページより構成されています。

表5-5 [Trace Find]ダイアログボックスのページ構成

ページ	設定項目
[General]	検索範囲を指定します。
[Address]	アドレス条件を指定します。
[Data]	データ条件を指定します。
[R/W]	アクセスサイクルの種類を指定します。
[Area]	アクセスしたエリアを指定します。 (タイムスタンプ取得時は無効)
[Status]	バス状態を指定します。 (タイムスタンプ取得時は無効)
[Probes]	4本のプローブ信号の状態を指定します。 (タイムスタンプ取得時は無効)
[IRQ7-0]	8本のIRQ入力の状態を指定します。 (タイムスタンプ取得時は無効)
[Timestamp]	バスサイクルのタイムスタンプを指定します。 (タイムスタンプ取得時のみ有効)

【注】 [General], [Address], [Data], [R/W], [Area], [Status], [Probes], [Timestamp]以外の項目はご使用のエミュレータにより異なります。
詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

各ページで条件を設定し、[OK]ボタンをクリックすることにより、サーチ条件を設定し、検索を開始します。
[キャンセル]ボタンをクリックすると、設定しないでダイアログボックスを閉じます。

検索の結果一致するトレースレコードが見つかった場合は当該レコード行を強調表示します。一致するトレースレコードが見つからなかった場合は、メッセージダイアログボックスを表示します。

トレース情報の検索は各ページで設定した条件がすべて一致するトレース情報のみを検索します。

トレースレコードが検索できた場合は、ポップアップメニューで[次を検索]を選択すると、次のトレースレコードを検索できます。

5 デバッグ

(1) General ページ

検索範囲を指定します。

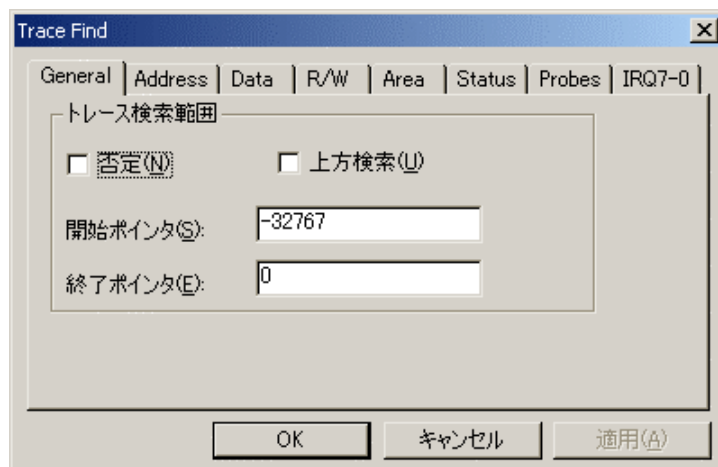


図 5-37 Trace Find ダイアログボックス(General ページ)

[トレース検索範囲]	検索範囲を指定します。
[否定]	チェックすると他のページで設定した項目の否定条件で検索します。
[上方検索]	チェックすると上方検索を行います。
[開始ポインタ]	検索を開始する PTR の値を入力します。
[終了ポインタ]	検索を終了する PTR の値を入力します。

【注】 検索範囲入力時、[開始ポインタ]オプションに検索を終了する PTR の値、[終了ポインタ]オプションに検索を開始する PTR の値を指定することも可能です。

(2) Address ページ

アドレス条件を指定します。

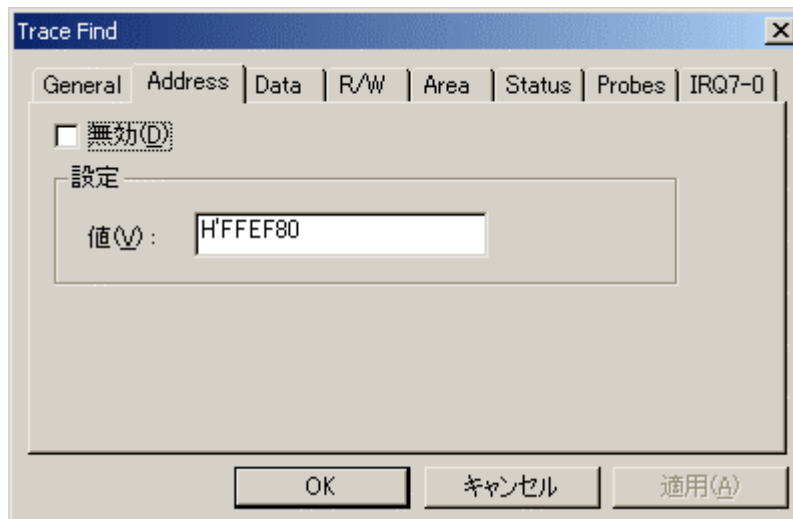


図 5-38 Trace Find ダイアログボックス(Address ページ)

[無効]	チェックすると、アドレスを検出しません。
[設定]	指定したアドレスを検出します。
[値]	アドレス値を入力します。 ([無効]選択時無効)

5 デバッグ

(5) Area ページ

アクセスしたエリアを指定します。
タイムスタンプ取得時は無効です。

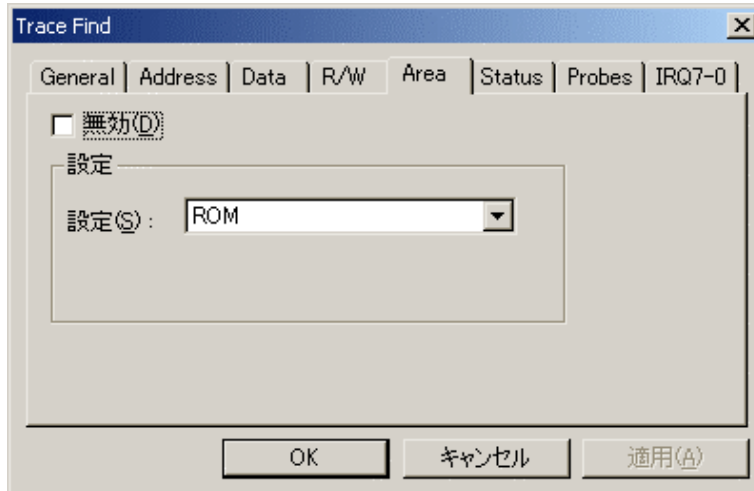


図 5-41 Trace Find ダイアログボックス(Area ページ)

- | | |
|------|-----------------------------|
| [無効] | チェックすると、エリア条件を検出しません。 |
| [設定] | 指定したエリア条件を検出します。 |
| [設定] | エリア条件を選択します。
([無効]選択時無効) |

【注】 指定可能なエリア条件はご使用のエミュレータにより異なります。
詳細につきましては「5.7.4 バス状態およびエリア信号について」をご参照ください。

(6) Status ページ

バス状態を指定します。
タイムスタンプ取得時は無効です。

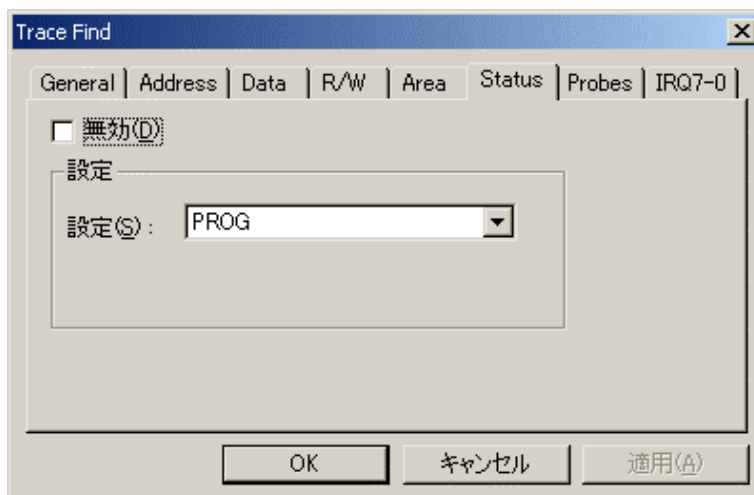


図 5-42 Trace Find ダイアログボックス(Status ページ)

- | | |
|------|----------------------------|
| [無効] | チェックすると、バス条件を検出しません。 |
| [設定] | 指定したバス条件を検出します。 |
| [設定] | バス条件を選択します。
([無効]選択時無効) |

【注】 指定可能なバス条件はご使用のエミュレータにより異なります。
詳細につきましては「5.7.4 バス状態およびエリア信号について」をご参照ください。

(7) Probes ページ

4本のプローブ信号の状態を指定します。
タイムスタンプ取得時は無効です。

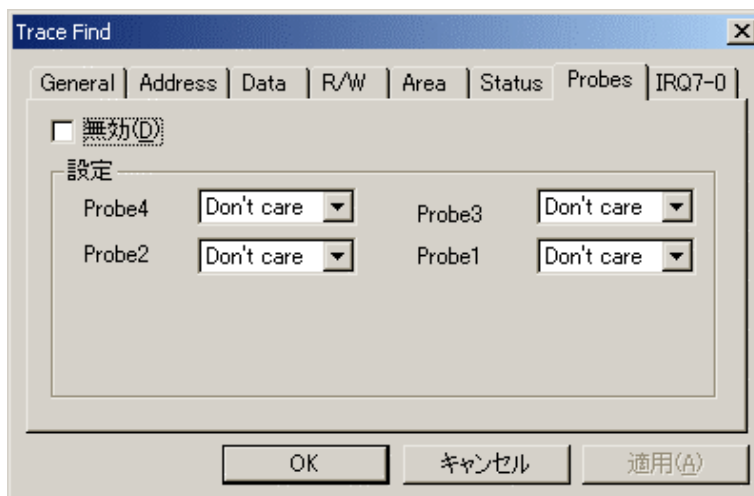


図 5-43 Trace Find ダイアログボックス(Probes ページ)

[無効] チェックすると、プローブ条件を検出しません。

[設定] 指定したプローブ条件を検出します。

[Probe4] ~ [Probe1] プローブ条件を選択します。
([無効]選択時無効)

Don't care: 選択したプローブ条件を検出しません
High: プローブ信号の状態が HIGH
Low: プローブ信号の状態が LOW

(8) IRQ7-0 ページ

IRQ 入力の状態を指定します。
タイムスタンプ取得時は無効です。

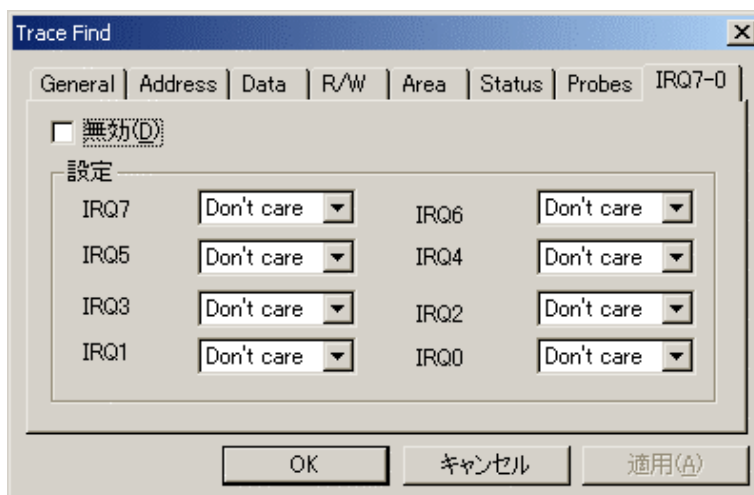


図 5-44 Trace Find ダイアログボックス(IRQ7-0 ページ)

[無効] チェックすると、IRQ 入力の条件を検出しません。

[設定] 指定した IRQ 入力の条件を検出します。

[IRQ7] ~ [IRQ0] IRQ 入力の条件を選択します。
([無効]選択時無効)

Don't care: 選択した IRQ 入力の条件を検出しません
High: IRQ 入力の状態が HIGH
Low: IRQ 入力の状態が LOW

(9) Timestamp ページ

バスサイクルのタイムスタンプを指定します。
タイムスタンプ取得時のみ有効です。

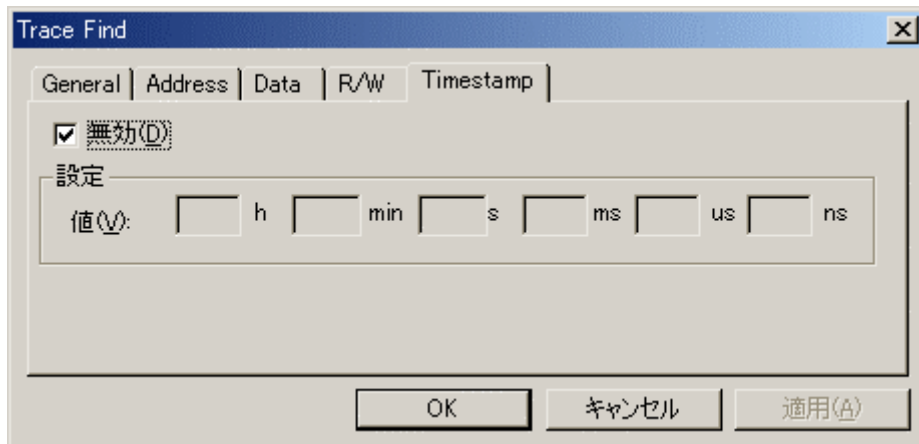


図 5-45 Trace Find ダイアログボックス(Timestamp ページ)

[無効]	チェックすると、タイムスタンプ値を検出しません。
[設定]	指定したタイムスタンプ値を検出します。(全ての桁の入力が必須)
[値]	タイムスタンプ値を入力します。 このフォーマットは以下の通りです。 [時]h [分]min [秒]s [ミリ秒]ms [マイクロ秒]us [ナノ秒]ns ([無効]選択時無効)

5.8.5 トレース情報をクリアする

トレース情報をクリアするには、ポップアップメニューから[クリア]を選択します。トレース情報を保持しているトレースバッファを空にします。複数の[Trace]ウィンドウが開いているときは、それらは同じバッファをアクセスしているため、すべての[Trace]ウィンドウをクリアすることになります。

5.8.6 トレース情報をファイルに保存する

トレース情報をファイルに保存するには、ポップアップメニューから[保存...]を選択します。

[名前を付けて保存]ダイアログボックスを表示します。[Trace]ウィンドウに表示しているトレース情報をテキストファイルとして保存します。保存する範囲を、[PTR]の範囲によって指定することができます(すべてのバッファをセーブするには、数分かかることがあります)。このファイルは保存のみ可能で、[Trace]ウィンドウへの読み込みはできないことに注意してください。

【注】 トレース情報をフィルタリングした場合、保存する範囲の指定はできません。フィルタリングした結果 [Trace]ウィンドウに表示されたトレース情報すべてを保存します。保存する範囲を指定したい場合は [Trace Filter]ダイアログボックスの[General]ページよりフィルタ範囲を指定してください。フィルタ機能については「5.8.12 取得したトレース情報から必要なレコードを抽出する」を参照してください。

5.8.7 [エディタ]ウィンドウを表示する

トレースレコードに対応する[エディタ]ウィンドウを表示するには二通りの方法があります。

- (1) トレースレコードを選択した状態でポップアップメニューから[ソースファイル表示]を選択する
- (2) トレースレコードをダブルクリックする

上記の操作により、[エディタ]ウィンドウあるいは[逆アセンブリ]ウィンドウを開いてソース表示し、選択した行をカーソルで示します。

5.8.8 ソース表示を整形する

ポップアップメニューで[ソーストリム]を選択すると、ソースプログラムの左側の空白を取り除きます。

取り除いた状態にした場合、[ソーストリム]メニューの左にチェックが付きます。チェックありの状態では[ソーストリム]メニューを選択すると取り除いた空白を元に戻します。

5.8.9 トレース情報のスナップショットを取得する

ユーザプログラム実行中、一時的にトレース情報を確認したい場合はスナップショットを取得します。

タイムスタンプの確認や、ブロープの入力チェックなどに便利です。

トレース情報のスナップショットを取得するにはポップアップメニューから[スナップショット]を選択します。

トレース情報の取得を一時中断し、最新のトレース情報 1 レコードを表示した後、トレース情報の取得を再開します。

トレース情報のスナップショットはユーザプログラム実行中のみ取得できます。

5.8.10 トレース情報の取得を一時的に停止する

ユーザプログラム実行中、一時的にトレース情報の取得を停止するにはポップアップメニューから[停止]を選択します。

トレース取得を中止し、トレース表示を更新します。

ユーザプログラムを停止せずにトレース情報の取得のみ停止し、トレース情報を確認する場合などに使用します。

5.8.11 トレース情報の取得を再開する

ユーザプログラム実行中、一時的にトレース情報の取得を停止した場合、再度トレース情報の取得を再開するにはポップアップメニューから[リスタート]を選択します。

5.8.12 取得したトレース情報から必要なレコードを抽出する

取得したトレース情報から必要なレコードのみを抽出するにはフィルタ機能を使用します。

フィルタ機能はハードウェアにより取得したトレース情報をソフトウェアによりフィルタリングします。

取得条件を設定してトレース情報を取得する[Trace Acquisition]設定と異なり、取得したトレース情報に対し何度もフィルタ設定を変更することで必要な情報が簡単に抽出でき、データの分析に役立ちます。

フィルタ機能を使用してもトレースバッファの内容は変更されません。

トレースバッファは有限ですので、[Trace Acquisition]設定により有用なトレース情報をより多く取得することで、より効果的にデータの分析が可能となります。

フィルタ機能を使用するには[Trace Filter]ダイアログボックスを使用します。

[Trace Filter]ダイアログボックスを開くには、ポップアップメニューの[フィルタ...]を選択します。

[Trace Filter]ダイアログボックスは下記ページより構成されています。

表5-6 [Trace Filter]ダイアログボックスのページ構成

ページ	設定項目
[General]	フィルタ範囲を指定します。
[Address]	アドレス条件を指定します。
[Data]	データ条件を指定します。
[R/W]	アクセスサイクルの種類を指定します。
[Area]	アクセスしたエリアを指定します。 (タイムスタンプ取得時は無効)
[Status]	バス状態を指定します。 (タイムスタンプ取得時は無効)
[Probes]	4本のプローブ信号の状態を指定します。 (タイムスタンプ取得時は無効)
[IRQ7-0]	8本のIRQ入力の状態を指定します。 (タイムスタンプ取得時は無効)
[Timestamp]	バスサイクルのタイムスタンプを指定します。 (タイムスタンプ取得時のみ有効)

【注】 [General], [Address], [Data], [R/W], [Area], [Status], [Probes], [Timestamp]以外の項目はご使用のエミュレータにより異なります。
詳細につきましては「8 本製品固有のソフトウェア仕様」またはオンラインヘルプをご参照ください。

各ページでフィルタ条件を設定し、[OK]ボタンをクリックすることにより、フィルタ条件にしたがいフィルタリングを行います。[キャンセル]ボタンをクリックすると、[Trace Filter]ダイアログボックスを開いた時点の設定のままダイアログボックスを閉じます。

フィルタリングは各ページで設定したフィルタ条件が1つ以上一致するトレース情報のみを[Trace]ウィンドウに表示します。

フィルタリングを行ってもトレースバッファの内容は変更されませんので、何度もフィルタ条件を変更しデータの分析ができます。

(1) General ページ

フィルタ範囲を指定します。

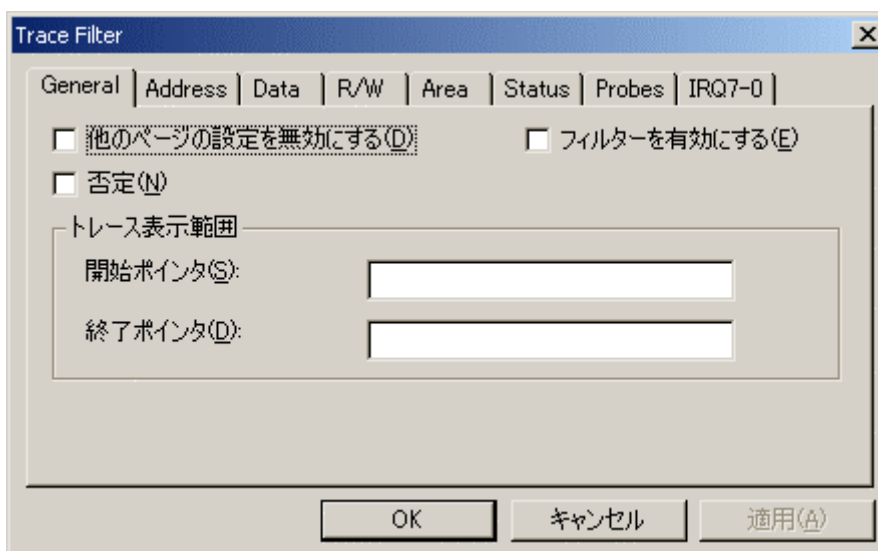


図 5-46 Trace Filter ダイアログボックス(General ページ)

[他のページ設定を無効にする]	チェックすると、サイクル番号のみ指定できます。他のオプションはすべて無効になります。
[フィルターを有効にする]	チェックすると、フィルタを有効にします。
[否定]	チェックすると他のページで設定した項目の否定条件でフィルタリングします。
[トレース表示範囲]	フィルタ範囲を指定します。
[開始ポイント]	フィルタを開始する PTR の値を入力します。
[終了ポイント]	フィルタを終了する PTR の値を入力します。

【注】 フィルタ範囲入力時、[開始ポイント]オプションにフィルタを終了する PTR の値、 [終了ポイント]オプションにフィルタを開始する PTR の値を指定することも可能です。

- (2) Address ページ
アドレス条件を指定します。

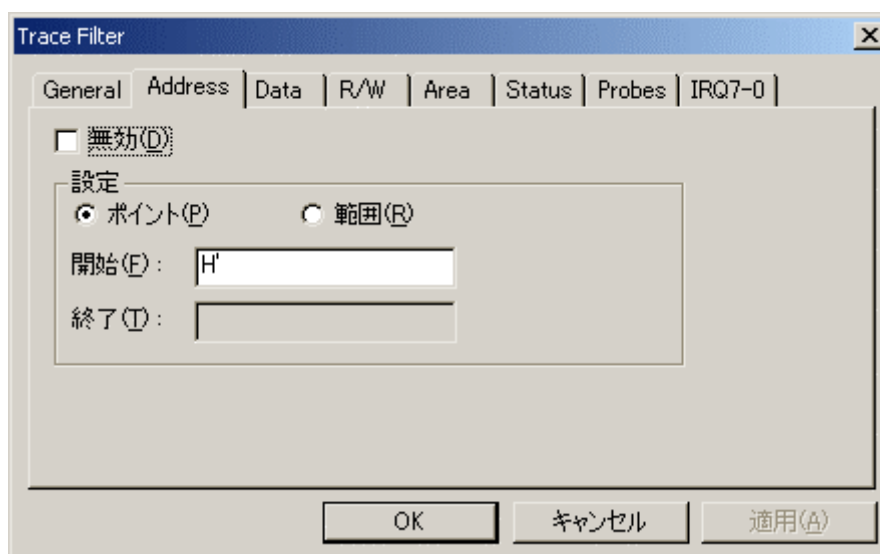


図 5-47 Trace Filter ダイアログボックス(Address ページ)

[無効]	チェックすると、アドレスを検出しません。
[設定]	指定したアドレスを検出します。
[ポイント]	単一アドレスを指定します。 ([無効]選択時無効)
[範囲]	アドレス範囲を指定します。 ([無効]選択時無効)
[開始]	単一アドレスまたはアドレス範囲の開始アドレスを入力します。 ([無効]選択時無効)
[終了]	アドレス範囲の終了アドレスを入力します。 ([範囲]選択時有効)

【注】 アドレス範囲入力時、[開始]オプションにアドレス範囲の終了アドレス、[終了]オプションにアドレス範囲の開始アドレスを指定することも可能です。

5 デバッグ

(3) Data ページ

データ条件を指定します。

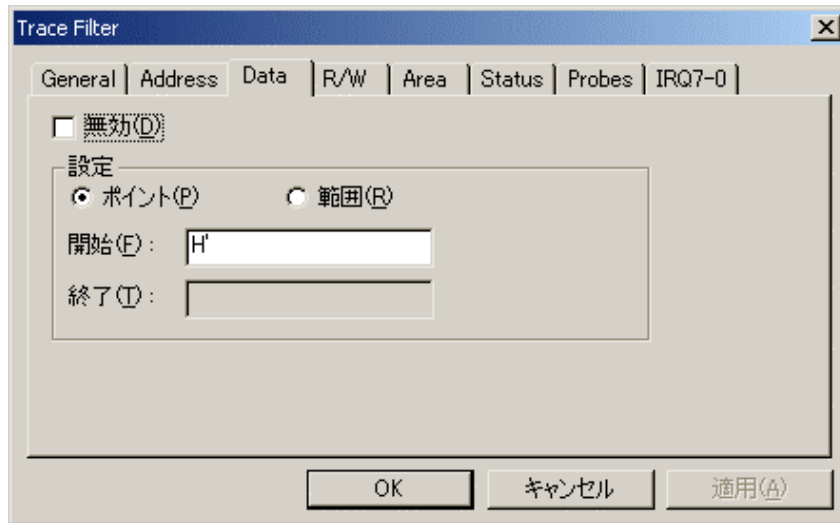


図 5-48 Trace Filter ダイアログボックス(Data ページ)

[無効]	チェックすると、データを検出しません。
[設定]	指定したデータを検出します。
[ポイント]	単一データを指定します。 ([無効]選択時無効)
[範囲]	アドレス範囲を指定します。 ([無効]選択時無効)
[開始]	単一データまたはデータ範囲の最小値を入力します。 ([無効]選択時無効)
[終了]	データ範囲の最大値を入力します。 ([範囲]選択時有効)

【注】 データ範囲入力時、[開始]オプションにデータの最大値、[終了]オプションにデータの最小値を指定することも可能です。

(4) R/W ページ

アクセスサイクルの種類を指定します。

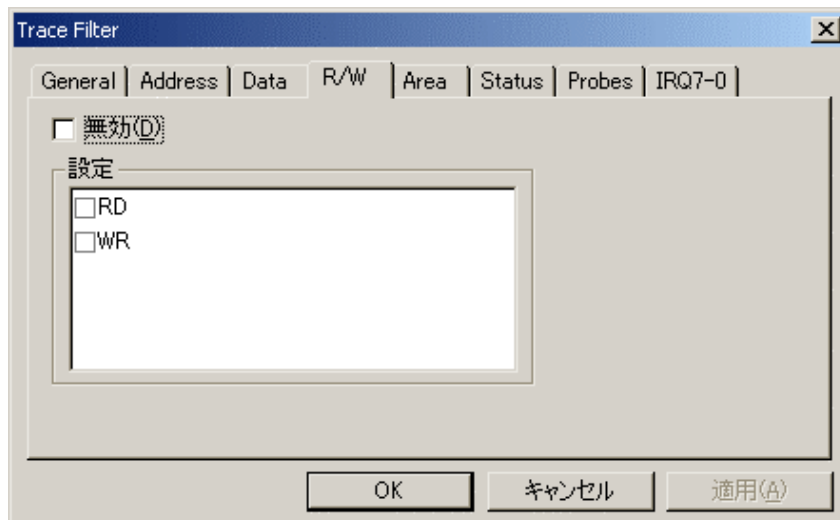


図 5-49 Trace Filter ダイアログボックス(R/W ページ)

[無効]	チェックすると、リード/ライト条件を検出しません。
[設定]	指定したリード/ライト条件を検出します。
[RD]	チェックすると、リードサイクルを検出します。 ([無効]選択時無効)
[WR]	チェックすると、ライトサイクルを検出します。 ([無効]選択時無効)

(5) Area ページ

アクセスしたエリアを指定します。
タイムスタンプ取得時は無効です。

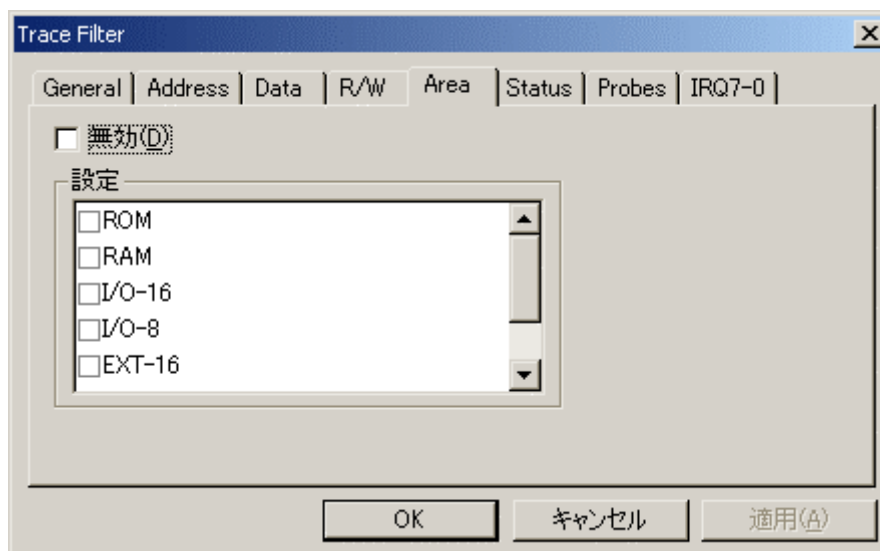


図 5-50 Trace Filter ダイアログボックス(Area ページ)

[無効]	チェックすると、エリア条件を検出しません。
[設定]	指定したエリア条件を検出します。 ([無効]選択時無効)

【注】 指定可能なエリア条件はご使用のエミュレータにより異なります。
詳細につきましては「5.7.4 バス状態およびエリア信号について」をご参照ください。

5 デバッグ

(6) Status ページ

バス状態を指定します。
タイムスタンプ取得時は無効です。

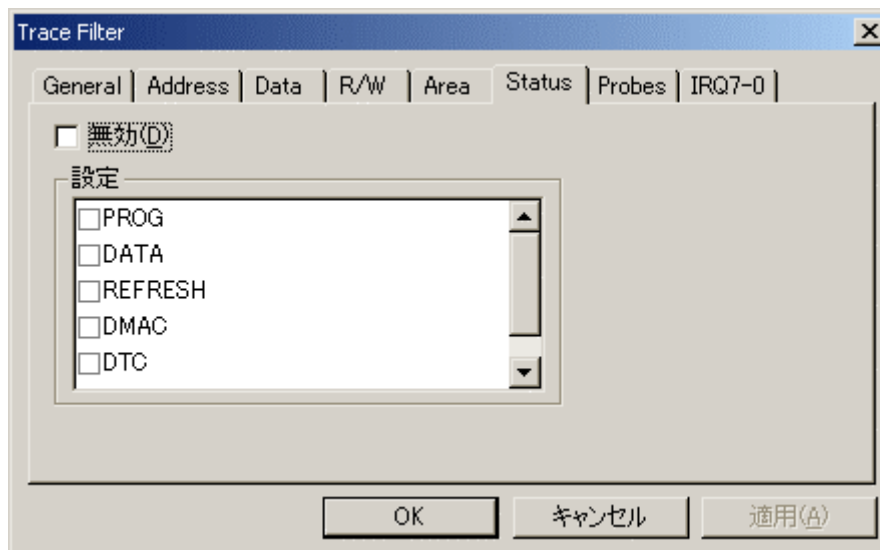


図 5-51 Trace Filter ダイアログボックス(Status ページ)

[無効] チェックすると、バス条件を検出しません。
[設定] 指定したバス条件を検出します。
 (無効)選択時無効)

【注】 指定可能なバス条件はご使用のエミュレータにより異なります。
詳細につきましては「5.7.4 バス状態およびエリア信号について」をご参照ください。

(7) Probes ページ

4本のプローブ信号の状態を指定します。
タイムスタンプ取得時は無効です。

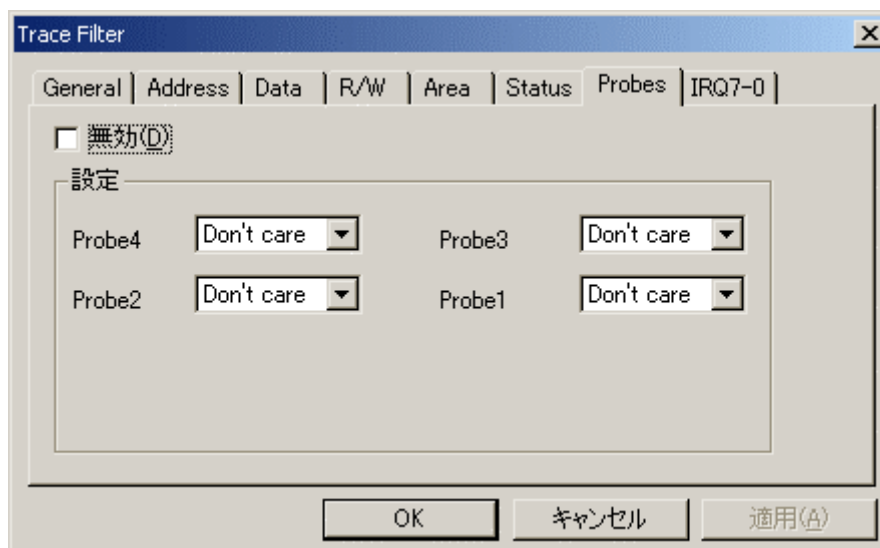


図 5-52 Trace Filter ダイアログボックス(Probes ページ)

- [無効] チェックすると、プローブ条件を検出しません。
 [設定] 指定したプローブ条件を検出します。
 [Probe4] ~ [Probe1] プローブ条件を選択します。
 ([無効]選択時無効)
 Don't care: 選択したプローブ条件を検出しません
 High: プローブ信号の状態が HIGH
 Low: プローブ信号の状態が LOW

(8) IRQ7-0 ページ

IRQ 信号の状態を指定します。
 タイムスタンプ取得時は無効です。

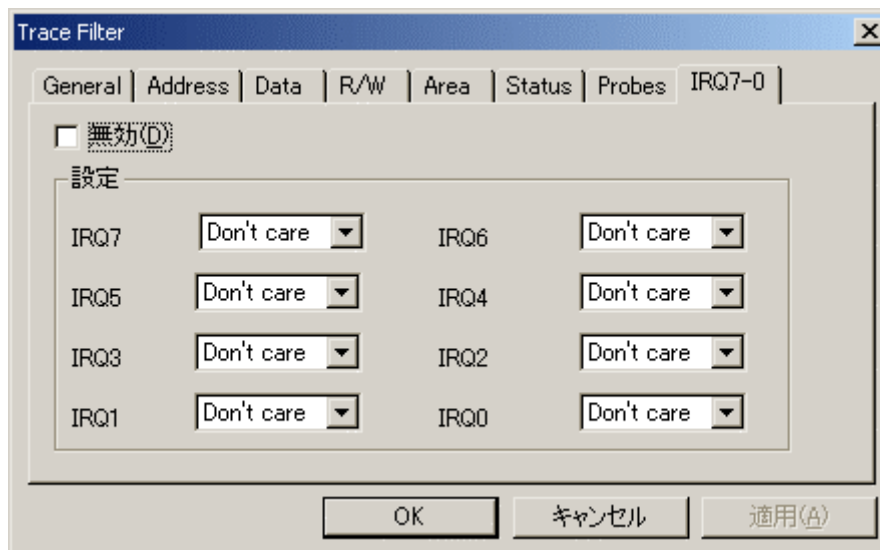


図 5-53 Trace Filter ダイアログボックス(IRQ7-0 ページ)

- [無効] チェックすると、IRQ 入力条件を検出しません。
 [設定] 指定した IRQ 入力条件を検出します。
 [IRQ7] ~ [IRQ0] IRQ 入力条件を選択します。
 ([無効]選択時無効)
 Don't care: 選択した IRQ 入力条件を検出しません
 High: IRQ 入力状態が HIGH
 Low: IRQ 入力状態が LOW

(9) Timestamp ページ

バスサイクルのタイムスタンプを指定します。
タイムスタンプ取得時のみ有効です。

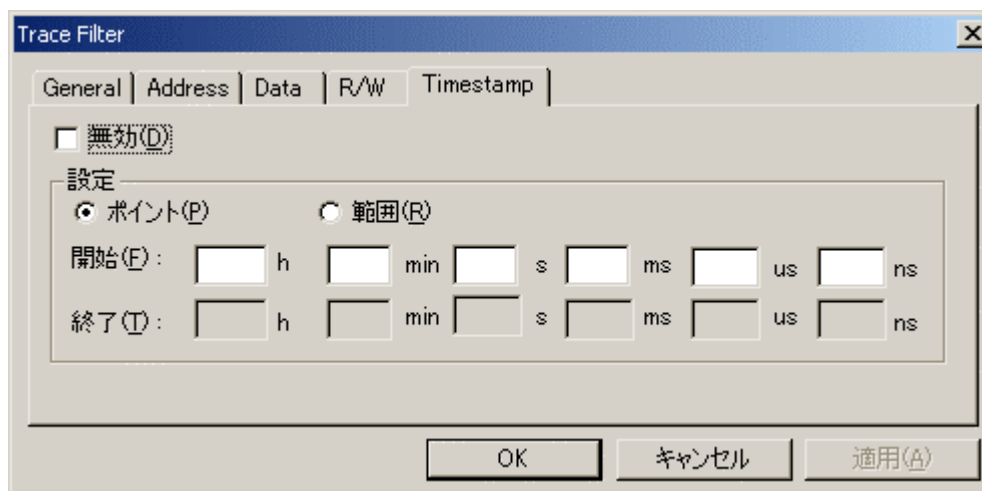


図 5-54 Trace Filter ダイアログボックス(Timestamp ページ)

[無効]	チェックすると、タイムスタンプ値を検出しません。
[設定]	指定したタイムスタンプ値を検出します。
[ポイント]	単一タイムスタンプを指定します。 ([無効]選択時無効)
[範囲]	タイムスタンプ範囲を指定します。 ([無効]選択時無効)
[開始]	単一タイムスタンプまたはタイムスタンプ範囲の最小値を入力します。 このフォーマットは以下の通りです。 [時] h [分] min [秒] s [ミリ秒] ms [マイクロ秒] us [ナノ秒] ns ([無効]選択時無効)
[終了]	タイムスタンプ範囲の最大値を入力します。 このフォーマットは以下の通りです。 [時] h [分] min [秒] s [ミリ秒] ms [マイクロ秒] us [ナノ秒] ns ([範囲]選択時有効)

【注】 タイムスタンプ範囲入力時、[開始]オプションにタイムスタンプの最大値、[終了]オプションにタイムスタンプの最小値を指定することも可能です。

5.8.13 タイムスタンプの差を計算する

タイムスタンプ情報取得時、トレース結果より指定した2点間の時間差を計算するには、ポップアップメニューから[タイムスタンプ差...]を選択します。

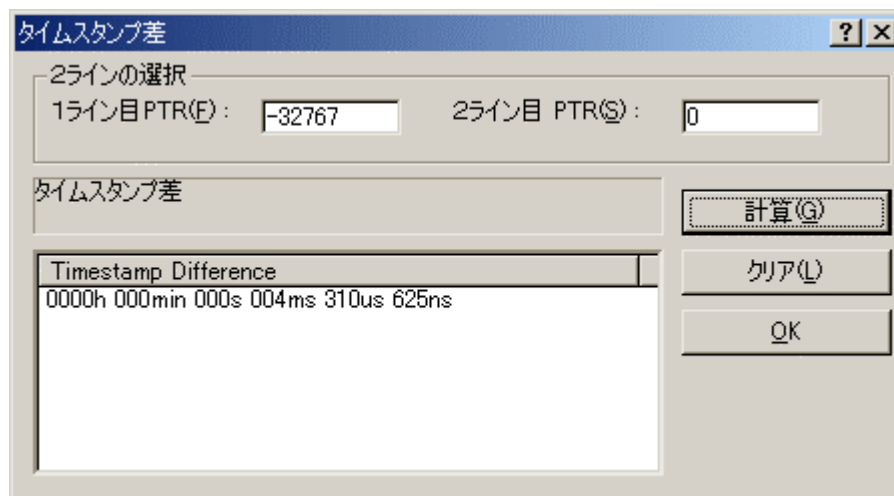


図 5-55 タイムスタンプ差ダイアログボックス

- | | |
|--------------|---|
| [2 ラインの選択] | タイムスタンプの差を計算するトレースレコードを指定します。 |
| [1 ライン目 PTR] | 差分を計測する1点目のポイントを指定します。
デフォルト値は、[Trace]ウィンドウ上で選択されているラインのポイントが表示されます。 |
| [2 ライン目 PTR] | 差分を計測する2点目のポイントを指定します。 |
| [タイムスタンプ差] | 2点間の時間差を計算した結果を表示します。 |
| [計算] | 指定した2点間の差分を計算し、結果を[タイムスタンプ差]リストに表示します。 |
| [クリア] | [タイムスタンプ差]リストのすべての結果を消去します。 |
| [OK] | ダイアログボックスを閉じます。
この時、[タイムスタンプ差]リストのすべての結果は消去されます。 |

5.8.14 統計情報を解析する

指定された条件で統計情報の解析を実行するには、ポップアップメニューから[統計...]を選択します。
[統計]ダイアログボックスが開きます。

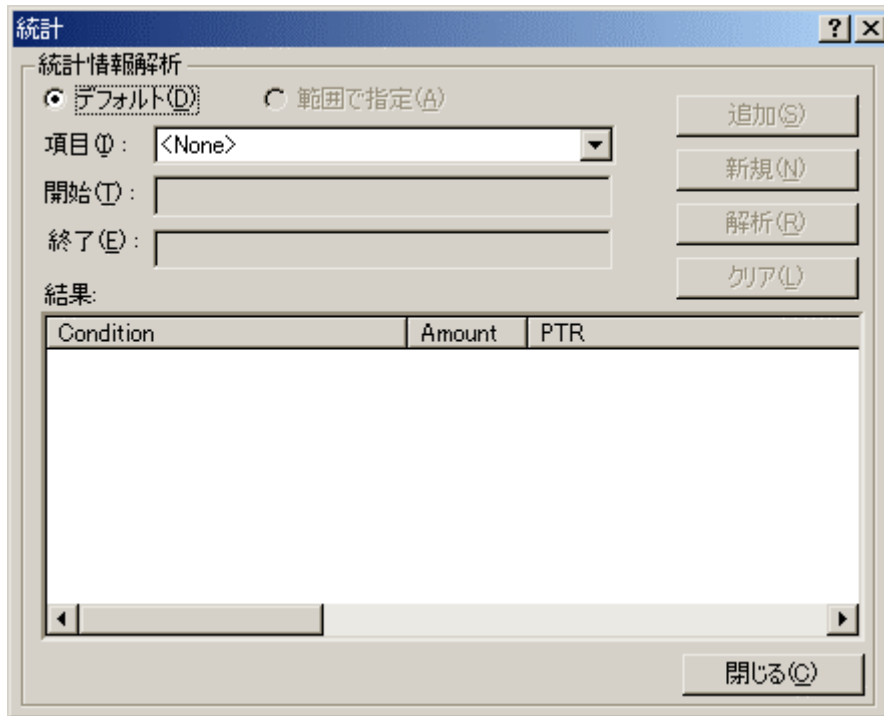


図 5-56 統計ダイアログボックス

[統計情報解析]	統計情報を解析するための設定を行います。
[デフォルト]	単一の入力値または文字列を指定します。
[範囲で指定]	入力値または文字列を範囲で指定します。
[項目]	解析対象項目を指定します。
[開始]	入力値または文字列を指定します。 範囲で指定する場合は開始値を設定します。
[終了]	範囲で指定する場合の終了値を設定します。 ([範囲で指定]選択時有効)
[追加]	現在の条件に追加設定します。
[新規]	新しい条件を指定します。
[解析]	統計情報解析の結果を取得します。
[結果]	すべての条件と統計情報解析結果を削除します。
[閉じる]	ダイアログボックスを閉じます。 この時、[結果]リストのすべての結果は消去されます。

本ダイアログボックスは、トレース情報の統計情報解析に使用します。[項目]オプションで解析対象項目を指定し、[開始] オプションおよび[終了]オプションで入力値または文字列を指定します。
[新規]ボタンまたは[追加]ボタンにより条件を設定し[解析]ボタンをクリックすると、統計情報を解析し[結果]リストに解析結果を表示します。

【注】 本エミュレータでは[PTR]項目のみ範囲で指定可能です。それ以外の項目は単一の文字列で指定してください。

統計情報の解析における文字列の判定は[Trace]ウィンドウに表示される文字列と比較し、完全一致したもののだけをカウントします。ただし、大文字小文字は区別しません。また、空白の数も考慮しません。

5.8.15 取得したトレース情報から関数呼び出し箇所を抽出する

取得したトレース情報から関数呼び出し箇所のみを抽出するには、ポップアップメニューから[関数コール...]を選択します。

[関数コール箇所の表示]ダイアログボックスが開きます。

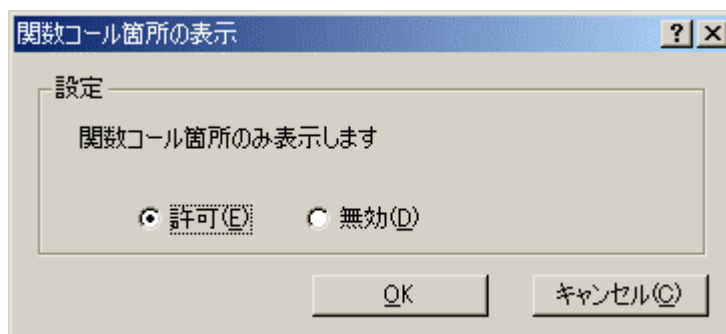


図 5-57 関数コール箇所の表示ダイアログボックス

[設定]	関数呼び出し箇所の抽出を行うかどうか設定します。
[許可]	関数呼び出し箇所の抽出を行います。
[無効]	関数呼び出し箇所の抽出を行いません。

[許可]オプションを選択した場合、取得したトレース情報より関数呼び出しを行っているサイクルのみを抽出し表示します。関数呼び出し箇所の抽出を行ってもトレースバッファの内容は変更されません。

フリートレースのトレース結果または、関数の呼び出しを含んだトレース情報に対して本機能を使用することにより、関数の呼び出し順序を調べることができます。

5.9 パフォーマンスを測定する

ユーザプログラムの実行効率を測定するにはパフォーマンス解析機能を使用します。

パフォーマンス解析機能は E6000 エミュレータのハードウェアパフォーマンス測定回路により指定範囲の実行効率を測定するため、リアルタイム性は損なわれません。

実行効率測定の設定は測定用途に応じて下記 5 つのモードより選択できます。

表5-7 実行効率測定条件で設定可能な測定モード

測定モード	測定内容	測定用途
指定範囲内時間測定	指定した範囲の実行時間および実行回数を測定します。	関数の処理時間のうち関数内から呼び出す子関数の処理時間を除いた処理時間を測定する場合などに使用します。
指定アドレス間時間測定	指定したアドレス間の実行時間および実行回数を測定します。	関数の処理時間を測定する場合などに使用します。
指定アドレス範囲間時間測定	指定範囲から別の指定範囲までの実行時間を測定します。	アセンブリプログラムなど、サブルーチンを連続して配置したプログラムにおいて、連続するサブルーチンのうちのいずれかが呼ばれてから別の連続するサブルーチンのいずれかが呼ばれるまでの実行時間を測定する場合などに使用します。
領域アクセス回数測定	指定した範囲から別の指定した範囲に対するアクセス回数を測定します。	特定の関数からのグローバル変数に対するアクセス回数を測定する場合などに使用します。
指定範囲内コール回数測定	指定した範囲から別の指定した範囲をコールした回数を測定します。	特定の関数からの関数コール回数を測定する場合などに使用します。

実行効率測定の設定は E6000 ハードウェアパフォーマンス測定回路に実装されたパフォーマンスチャネル (8 チャネル) を使用し、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設定では 1 条件設定あたり連続した 2 ポイントを使用するため、これらの測定モードを使用する場合、設定可能ポイント数は最大 4 ポイントとなります。


表5-8 実行効率測定条件のモード設定

測定モード	ポイント							
	1	2	3	4	5	6	7	8
指定範囲内時間測定								
指定アドレス間時間測定								
指定アドレス範囲間時間測定		x		x		x		x
領域アクセス回数測定		x		x		x		x
指定範囲内コール回数測定		x		x		x		x

: 設定可, x: 設定不可

【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・領域アクセス回数測定および指定範囲内コール回数測定は連続した 2 ポイントを使用します。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。

5.9.1 パフォーマンス解析ウィンドウを開く

[パフォーマンス解析]ウィンドウを開くには、[表示->パフォーマンス->パフォーマンス解析]を選択するか、[パフォーマンス解析]ツールバーボタンをクリックしてください。[パフォーマンス解析方式の選択]ダイアログボックスを表示します。

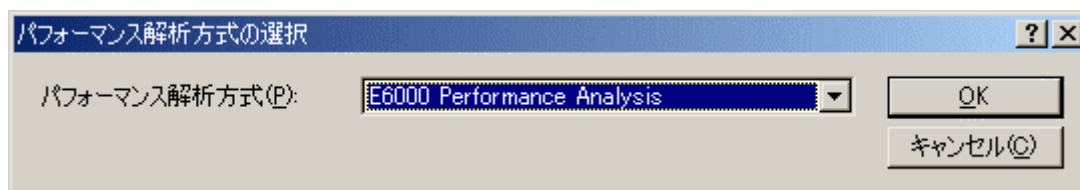


図 5-58 パフォーマンス解析方式の選択

[E6000 Performance Analysis]を選択し[OK]ボタンをクリックすると[パフォーマンス解析]ウィンドウが開きます。

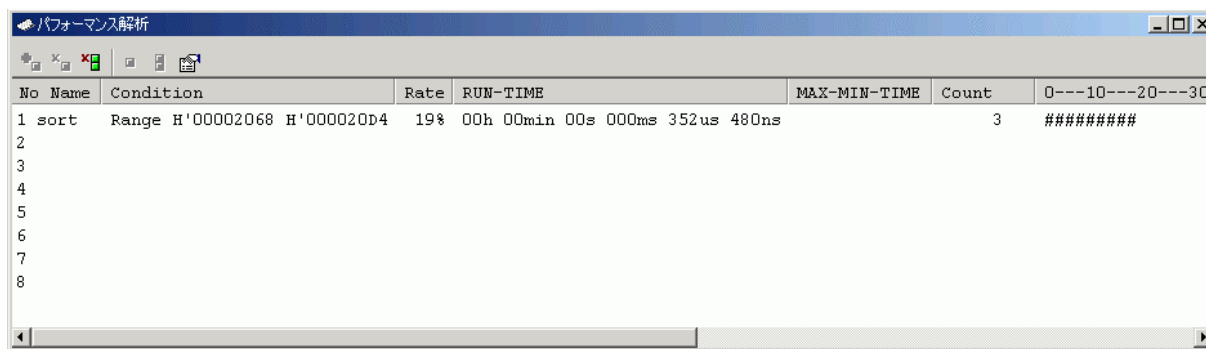


図 5-59 パフォーマンス解析ウィンドウ

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。

[パフォーマンス解析]ウィンドウ内の不要なカラムは非表示にすることができます。カラムを非表示にする場合はヘッダカラム上で右クリックすると表示されるポップアップメニューより非表示にしたいカラムを選択してください。カラムを再表示する場合は再度ポップアップメニューより該当のカラムを選択してください。

5.9.2 実行効率測定条件を設定する

[パフォーマンス解析]ウィンドウでは測定条件の設定内容の表示および変更ができます。

条件を設定するポイントを選択し、ポップアップメニューから[設定...]を選択すると[Performance Analysis Properties]ダイアログボックスを表示します。

実行効率測定条件は[Measurement Method]オプションにより5つのモードから選択可能です。

表5-9 実行効率測定条件(Measurement Method)

[Measurement Method]オプション	測定モード
Time Of Specified Range Measurement	指定範囲内時間測定
Start Point To End Point Measurement	指定アドレス間時間測定
Start Range To End Range Measurement	指定アドレス範囲時間測定
Access Count Of Specified Range Measurement	領域アクセス回数測定
Called Count Of Specified Range Measurement	指定範囲内コール回数測定

使用するモードに応じて実行効率測定条件を設定します。

各モードにより設定するパラメータが異なります。

[Performance Analysis Properties]ダイアログボックスにはアドレス範囲指定時、関数名を入力することにより関数のアドレス範囲を自動的に入力する入力補助機能があります。

[Performance Analysis Properties]ダイアログボックス上にある[...]ボタンをクリックすると表示される[Input Function Range]ダイアログボックスより、関数名を指定すると自動的に関数のアドレス範囲を入力することができます。

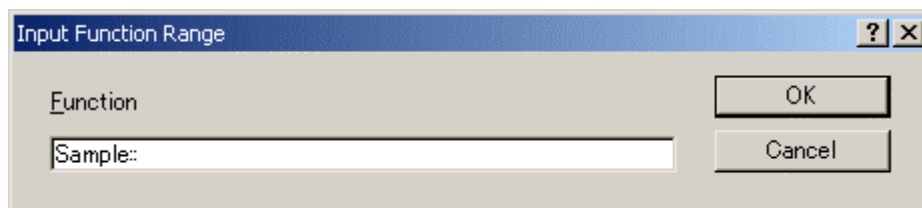


図 5-60 Input Function Range ダイアログボックス

- 【注】
1. オーバーロード関数またはクラス名を入力した場合、[関数選択]ダイアログボックスが開きますので、関数を選択してください。
 2. 算出されるアドレスは参考値です。場合により関数の終了アドレスが異なる場合があります。[逆アセンブリ]ウィンドウにより関数の最終命令を確認し、[End Address]の設定値を最終命令のアドレスに補正してください。
(一般的に関数の最終命令は RTS 命令となります。)
アドレス入力ではアドレス値以外にラベル名および式の指定も可能です。

(1) Time Of Specified Range Measurement

指定範囲内時間測定

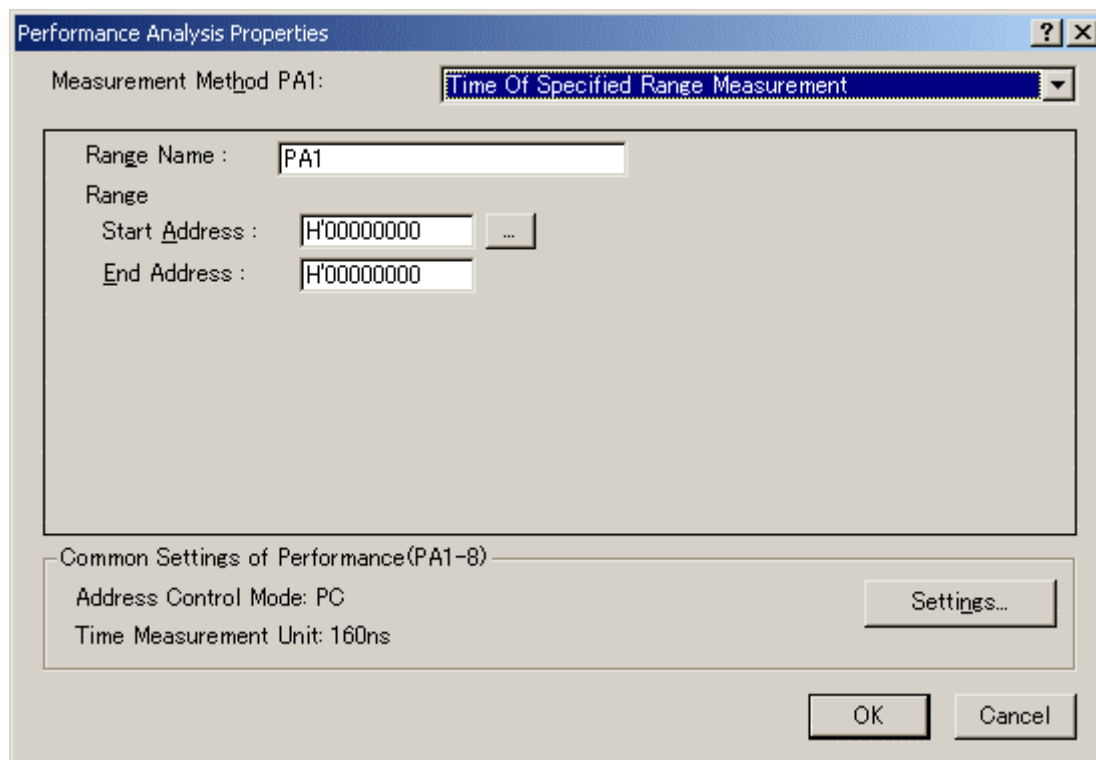


図 5-61 Time Of Specified Range Measurement 設定

[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。

<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。

(2) Start Point To End Point Measurement

指定アドレス間時間測定

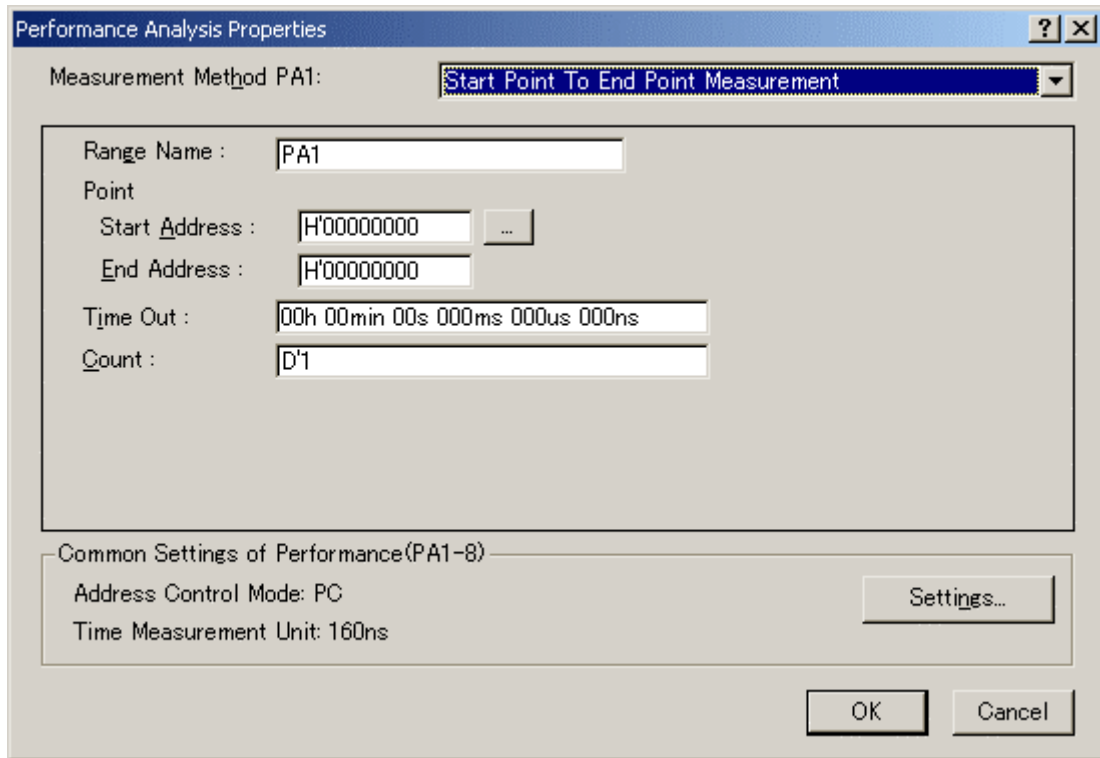


図 5-62 Start Point To End Point Measurement 設定

[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。 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ときにブレイクします(トータル時間ではありません)。 (チャンネル 1 のみ設定可能)
[Count]	実行回数測定カウントアップ値を指定します。 実行回数がカウントアップ値を超えたときにブレイクします。 (チャンネル 1 のみ設定可能)

<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。

- 【注】 1. 指定アドレス間時間測定モードの[Time Out]オプションを設定した場合、実行時間は正しく計測されません。
2. [Time Out]オプションおよび[Count]オプションは、設定した条件のうちいずれか一方が成立した場合にユーザプログラムを停止します。(Performance Break)
3. [Time Out]オプションおよび[Count]オプションはチャンネル 1 のみ設定可能です。
指定アドレス間時間測定モードで[Time Out]オプションおよび[Count]オプションを使用したくない場合はチャンネル 1 以外を使用してください。

(3) Start Range To End Range Measurement

指定アドレス範囲間時間測定

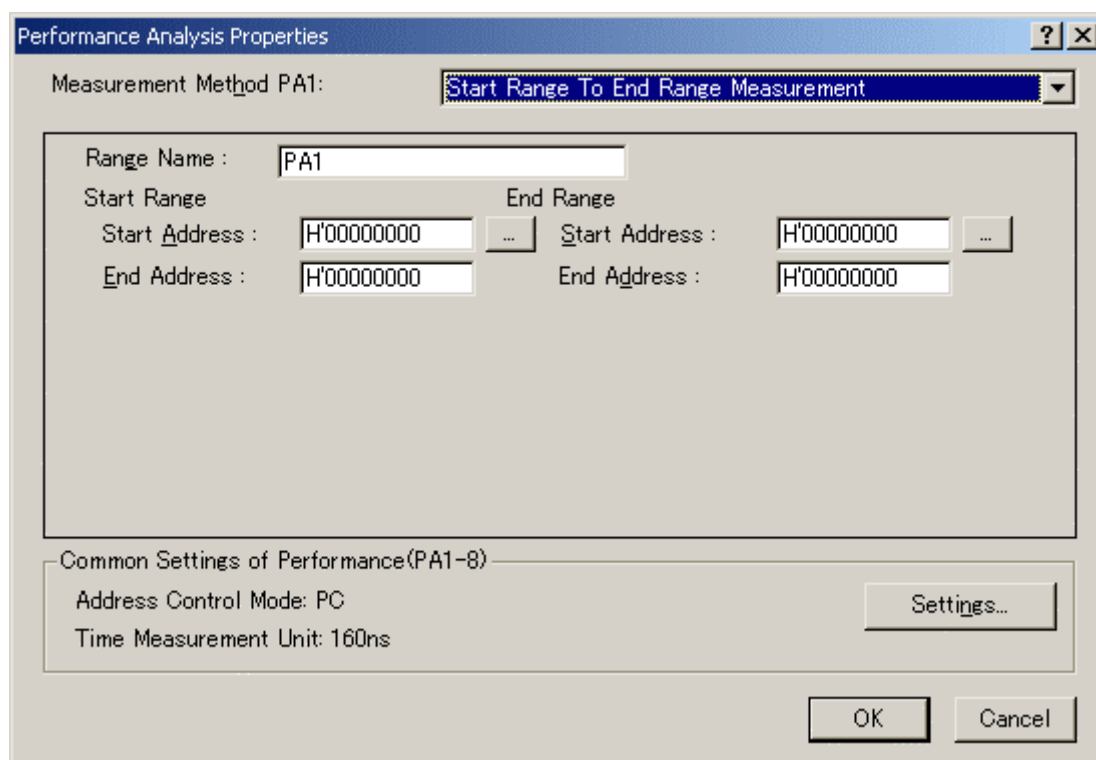


図 5-63 Start Range To End Range Measurement 設定

[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。

<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。

5 デバッグ

(4) Access Count Of Specified Range Measurement

領域アクセス回数測定

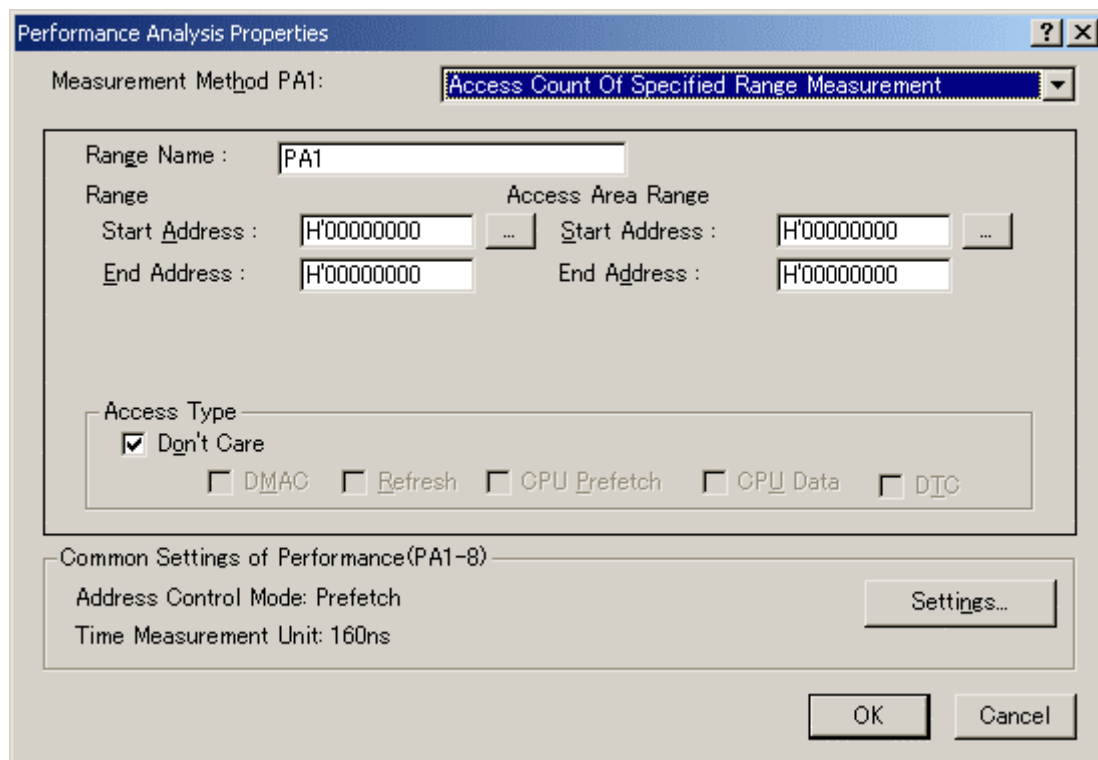


図 5-64 Access Count Of Specified Range Measurement 設定

[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。

<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。

【注】 指定可能なアクセス領域のバスサイクル条件はご使用のエミュレータにより異なります。詳細につきましては「5.7.4 バス状態およびエリア信号について」をご参照ください。

(5) Called Count Of Specified Range Measurement

指定範囲内コール回数測定

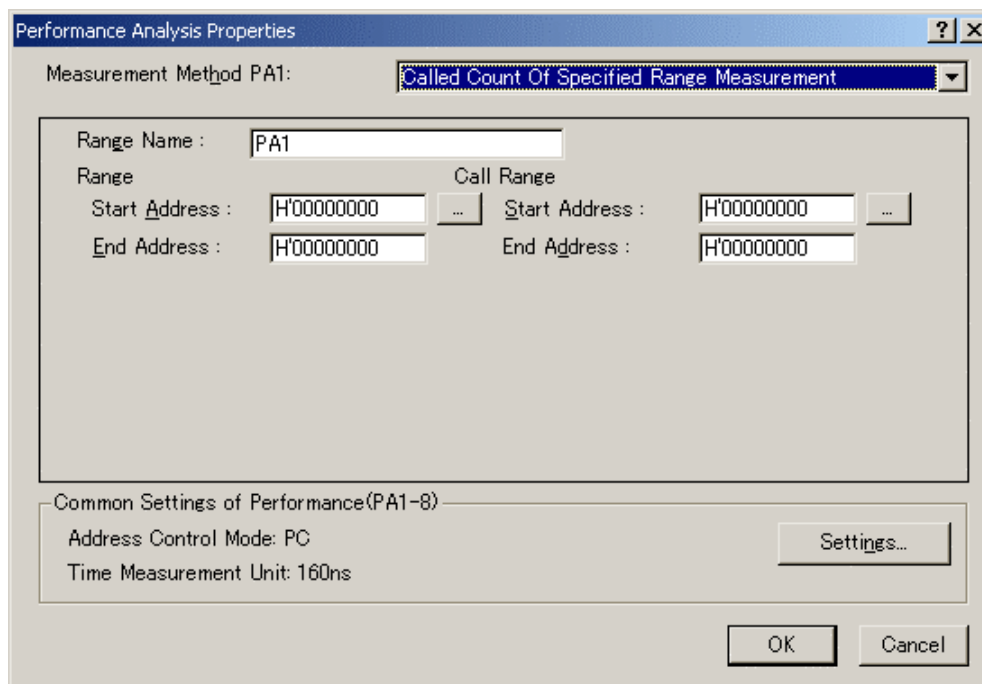


図 5-65 Called Count Of Specified Range Measurement 設定

[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。 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。
[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。

<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。

5.9.3 実行効率測定のアドレス検出方式および分解能を設定する

E6000 エミュレータのハードウェアパフォーマンス測定では、プログラムプリフェッチによる測定方式と実行PCによる測定方式の2通りのアドレス検出方式を使用します。

これらのアドレス検出方式は使用する測定モードにより適切に設定する必要があります。

また、E6000 エミュレータのハードウェアパフォーマンス測定では、実行時間測定の分解能も設定できます。

アドレス検出方式および分解能を設定するには[Performance Analysis Properties]ダイアログボックス上にある[Settings...]ボタンをクリックしてください。

[Common Settings of Performance(PA1-8)]ダイアログボックスが開きます。

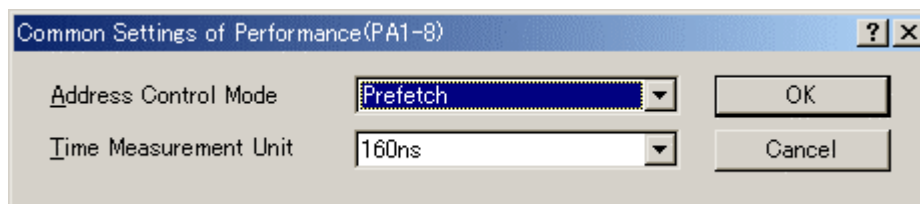


図 5-66 Common Settings of Performance(PA1-8)ダイアログボックス

[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。 Prefetch: プリフェッチアドレス検出モード PC: PC アドレス検出モード
[Time Measurement Unit]	時間測定に使用するタイマの分解能を設定します。 分解能は以下のいずれかから選択できます。 20ns, 40ns, 160ns, Target 実行時間測定用タイマは 40 ビットのカウンタで構成されています。 測定可能な最大時間は分解能 20ns で約 6 時、分解能 160us で約 2 日です。 カウンタがオーバーフローした場合、測定結果は Timer Overflow と表示します。 Target 選択時は入力クロックによりカウントアップします。 測定結果は 10 桁の 16 進数で表示します。

領域アクセス回数測定モード(Access Count Of Specified Range Measurement)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。
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。

5.9.4 実行効率測定を開始する

ユーザプログラムを実行すると前回の測定結果をクリアした後、設定した実行効率測定条件にしたがい自動的に実行効率測定を開始します。

ユーザプログラムを停止すると、測定結果を[パフォーマンス解析]ウィンドウに表示します。

5.9.5 測定条件を削除する

測定条件を選択した状態で、ポップアップメニューから[リセット]を選択すると、選択された測定条件を削除します。

5.9.6 すべての測定条件を削除する

ポップアップメニューから[全てリセット]を選択すると、設定している測定条件をすべて削除します。

6 チュートリアル

6.1 はじめに

E6000 エミュレータの主な機能を紹介するために、チュートリアルプログラムを提供しています。このプログラムを用いて説明します。

このチュートリアルプログラムは、C++言語で書かれており、10個のランダムデータを昇順/降順にソートします。チュートリアルプログラムでは、以下の処理を行います。

- main 関数ではソート処理を繰り返し実行するため tutorial 関数の呼び出しを繰り返します。
- tutorial 関数ではソートするランダムデータを生成し、sort 関数および change 関数を順に呼び出します。
- sort 関数では tutorial 関数で生成したランダムデータを格納した配列を入力し、昇順にソートします。
- change 関数では sort 関数で昇順にソートした配列を入力し、降順にソートします。

チュートリアルプログラムは、tutorial.cpp ファイルで提供しています。コンパイルされたロードモジュールは、Tutorial.abs ファイルとして Dwarf2 フォーマットで提供しています。

- 【注】
1. 再コンパイルを行った場合、本章で説明しているアドレスとずれることがあります。
 2. 本章は、一般的な E6000 エミュレータの使用例です。各製品の仕様については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。
 3. 各製品に添付される Tutorial.abs の動作アドレスは、製品によって異なります。本章で使用するアドレスについては、対応するソース行より各製品上で該当のアドレスを確認し、適宜読み替えて操作してください。
 4. 本チュートリアルでは H8S/2646 E6000 エミュレータを例に説明しています。ファイルのパス情報や、図の画面は各製品により異なりますので適宜読み替えて操作してください。

6.2 High-performance Embedded Workshop の起動

「4.1.3 既存のワークスペースを指定する場合」の手順に従ってワークスペースを開きます。

ディレクトリは以下を指定してください。

OS インストールドライブ\Workspace\Tutorial\E6000\2646

- 【注】 パス情報は製品によって異なります。「8.2.4 チュートリアルプログラムの実行環境」を参照してください。

ファイルは以下を指定してください。

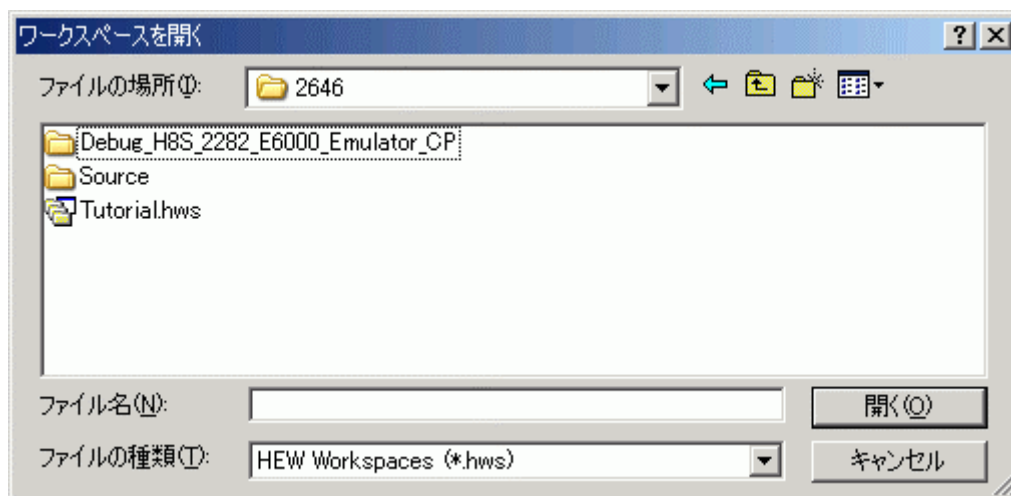


図 6-1 [ワークスペースを開く]ダイアログボックス

ワークスペースが開かれると、自動でエミュレータへの接続を行います。

6.3 チュートリアルプログラムのダウンロード

6.3.1 チュートリアルプログラムをダウンロードする

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できます。

[Download modules]の[Tutorial.abs]を右クリックすると表示されるポップアップメニューより[ダウンロード]を選択します。

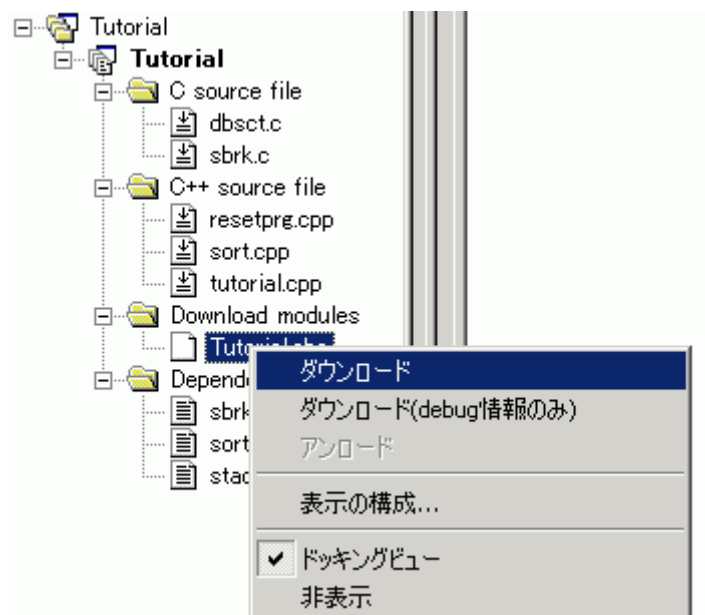


図 6-2 チュートリアルプログラムのダウンロード

6.3.2 ソースプログラムを表示する

High-performance Embedded Workshop では、ソースレベルでプログラムをデバッグできます。

[C++ source file]の[tutorial.cpp]をダブルクリックします。

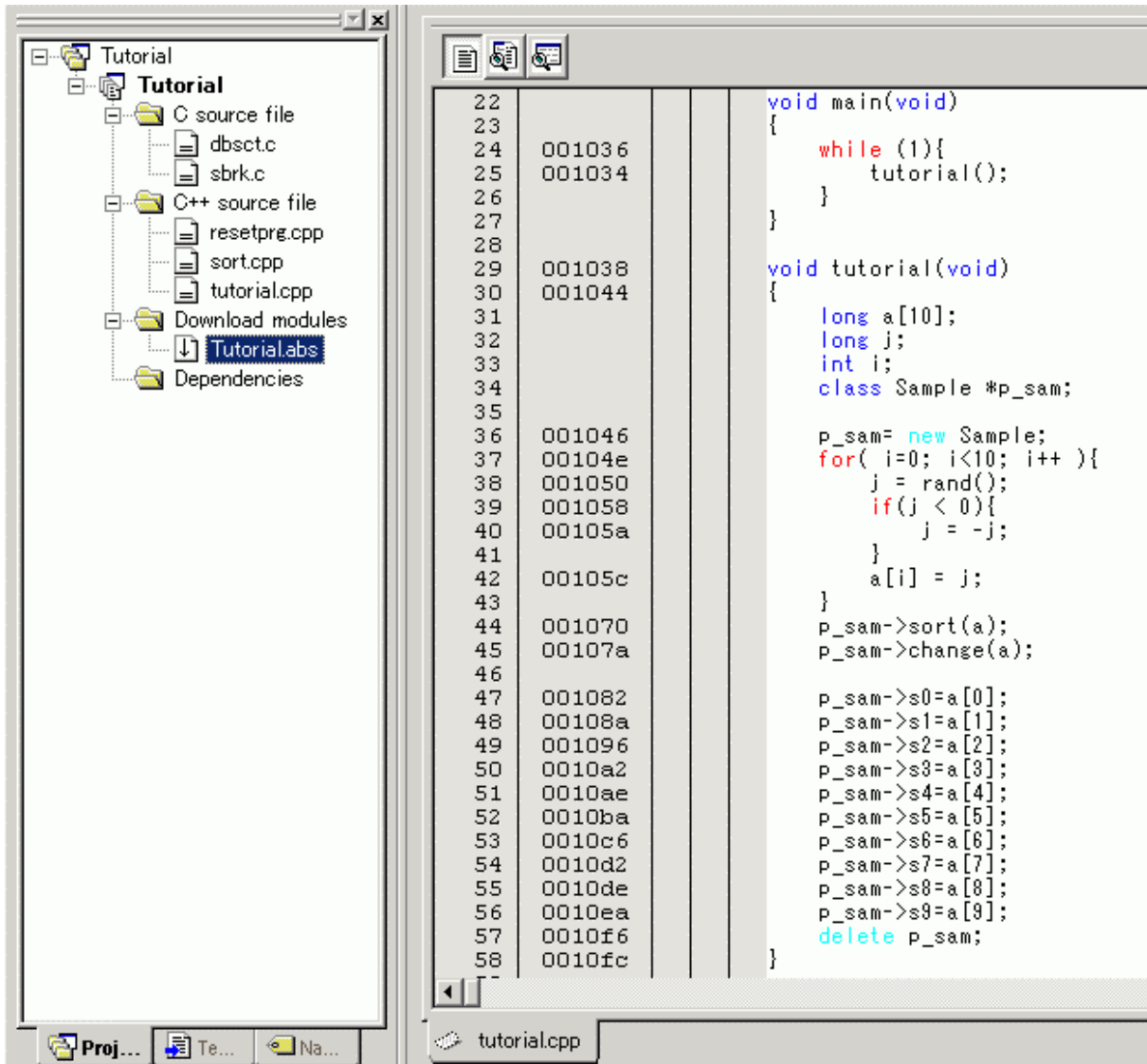


図 6-3 [エディタ]ウィンドウ (ソースプログラムの表示)

必要であれば、見やすいフォントとサイズに変更することも可能です。変更方法については High-performance Embedded Workshop ユーザーズマニュアルを参照してください。

[エディタ]ウィンドウは、最初は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。

6.4 ソフトウェアブレイクポイントの設定

簡単なデバッグ機能の1つにソフトウェアブレイクポイントがあります。

[エディタ]ウィンドウにおいて、ソフトウェアブレイクポイントを簡単に設定できます。たとえば、sort 関数のコール箇所にソフトウェアブレイクポイントを設定します。

sort 関数コールを含む行の[S/W ブレイクポイント]カラムをダブルクリックしてください。

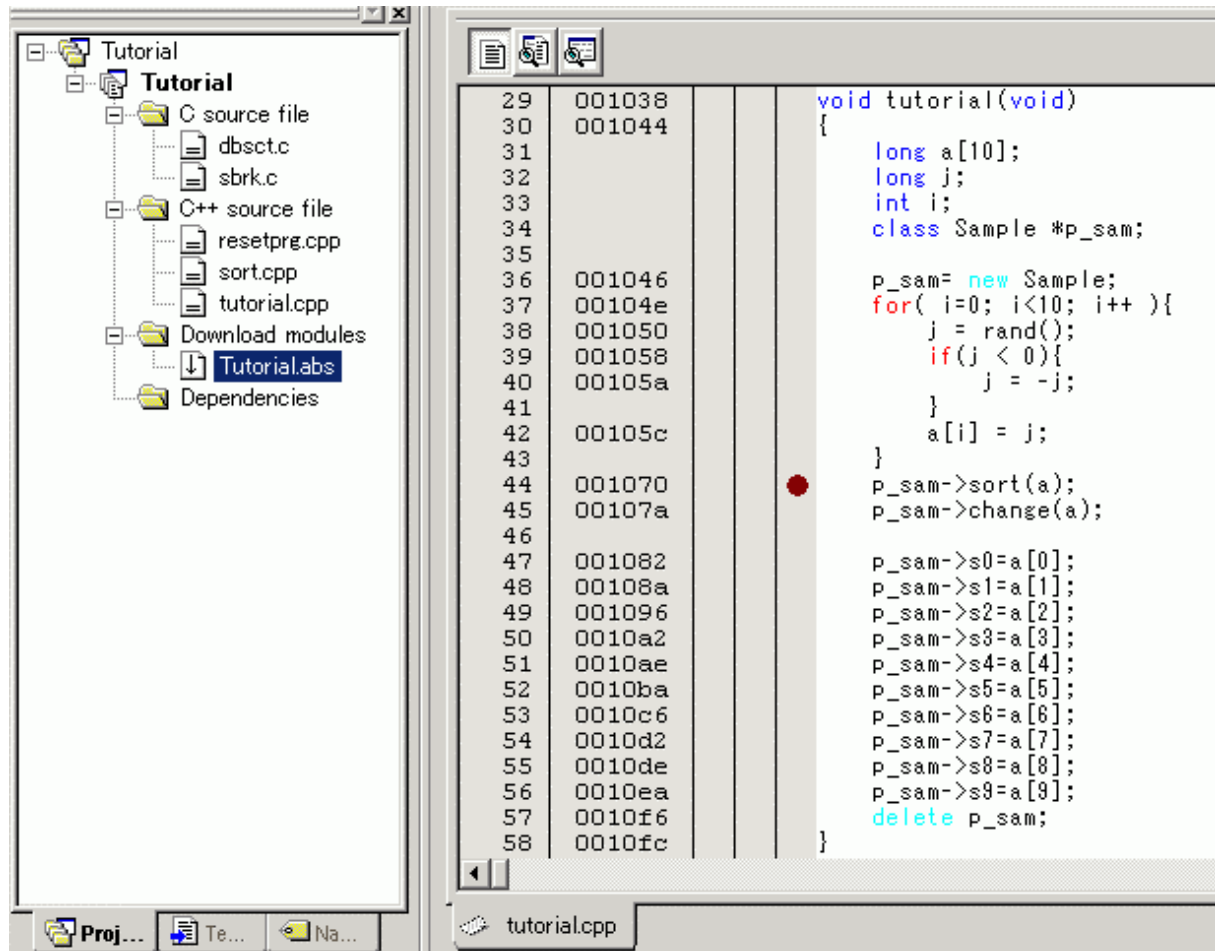


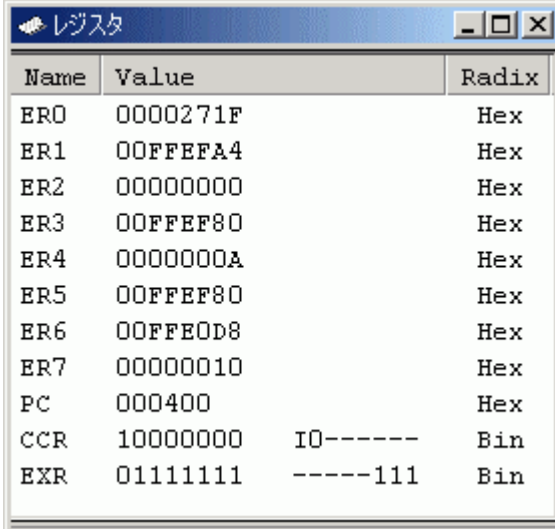
図 6-4 [エディタ]ウィンドウ (ソフトウェアブレイクポイントの設定)

sort 関数を含む行に”•”と表示されます。この表示によりソフトウェアブレイクポイントが設定されたことを示します。

6.5 レジスタ内容の変更

プログラムを実行する前に、プログラムカウンタの値を設定してください。

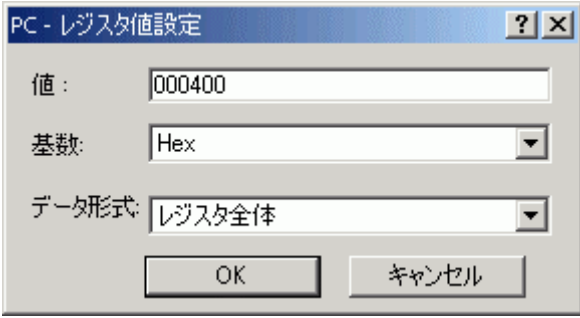
[表示]メニューから[CPU]サブメニューを選択し、[レジスタ]を選択してください。[レジスタ]ウィンドウが表示されます。



Name	Value	Radix
ER0	0000271F	Hex
ER1	00FFFEFA4	Hex
ER2	00000000	Hex
ER3	00FFFEF80	Hex
ER4	0000000A	Hex
ER5	00FFFEF80	Hex
ER6	00FFE0D8	Hex
ER7	00000010	Hex
PC	000400	Hex
CCR	10000000	I0----- Bin
EXR	01111111	-----111 Bin

図 6-5 [レジスタ]ウィンドウ

プログラムカウンタ(PC)を変更する場合には、[レジスタ]ウィンドウで[PC]の数値エリアをマウスでダブルクリックすると、以下のダイアログボックスが表示され、値の変更が可能です。本チュートリアルプログラムでは、H'00000400 を設定し、[OK]ボタンをクリックしてください。



PC - レジスタ値設定

値:

基数:

データ形式:

OK キャンセル

図 6-6 [レジスタ]ダイアログボックス(PC)

6.6 プログラムの実行

プログラムの実行方法について説明します。

プログラムを実行する場合は、[デバッグ]メニューから[実行]を選択するか、ツールバー上の[実行]ボタンを選択してください。



図 6-7 [実行]ボタン

実行を開始すると、ステータスバーに現在のアドレスバスの値、および MCU の動作状態を表示します。

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されます。プログラムが停止した位置を示すために [エディタ]ウィンドウの [S/W ブレークポイント] カラムに矢印が表示されます。また、[Break = Software Break] メッセージがステータスバーに表示されます。

【注】 ブレーク後にソースファイルを表示する際に、ソースファイルパスを問い合わせる場合があります。ソースファイルの場所は以下です。

OS インストールドライブ\Workspace\Tutorial\E6000\2646\Source

ソフトウェアのバージョンによっては、上記ディレクトリを指定できない場合があります。その場合は以下のディレクトリを指定してください。

High-performance Embedded Workshop インストール先ディレクトリ
 \Tools\Renesas\DebugComp\Platform\E6000\2646\Tutorial\Source

バス情報は各製品により異なりますので「\2646」の部分を変更してください。

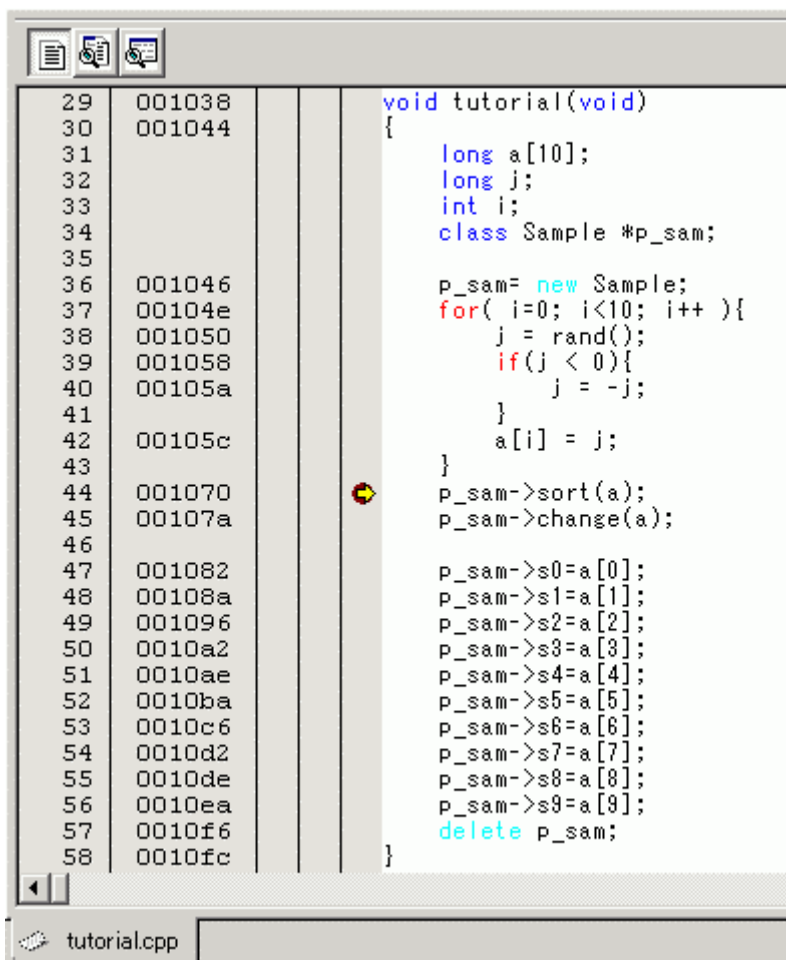


図 6-8 [エディタ]ウィンドウ (ブレーク状態)

[ステータス]ウィンドウで最後に発生したブレークの要因が確認できます。

[表示]メニューから[CPU]サブメニューを選択し、[ステータス]を選択してください。
[ステータス]ウィンドウが表示されますので、[Platform]シートを開いて Cause of last break の Status を確認してください。

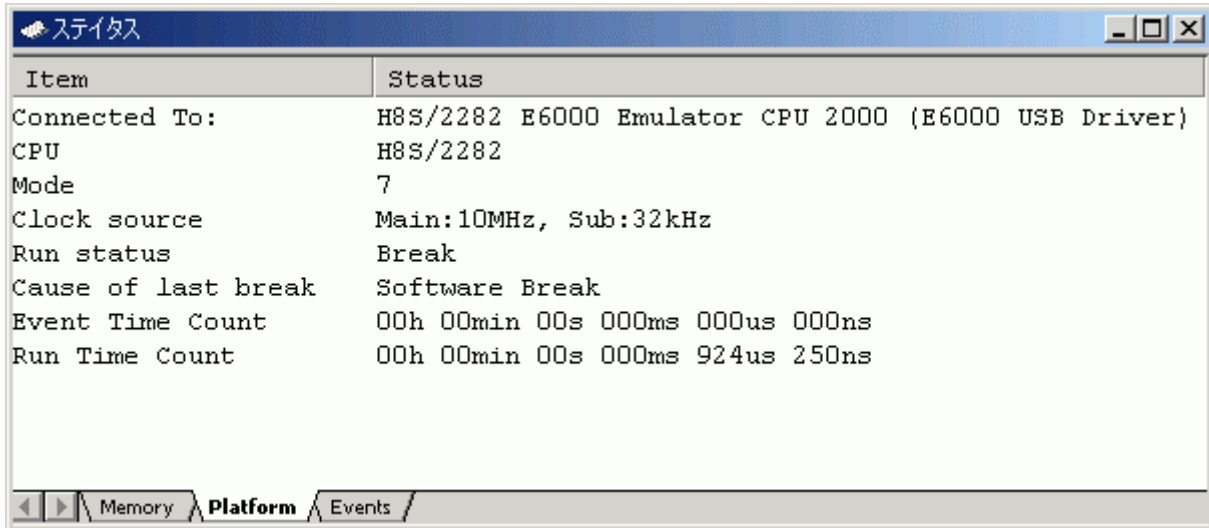


図 6-9 [ステータス]ウィンドウ

【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

6.7 ブレークポイントの確認

設定した全てのブレークポイントは、[イベントポイント]ウィンドウで確認することができます。

[表示]メニューから[コード]サブメニューを選択し、[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。[Breakpoint]シートを開きます。

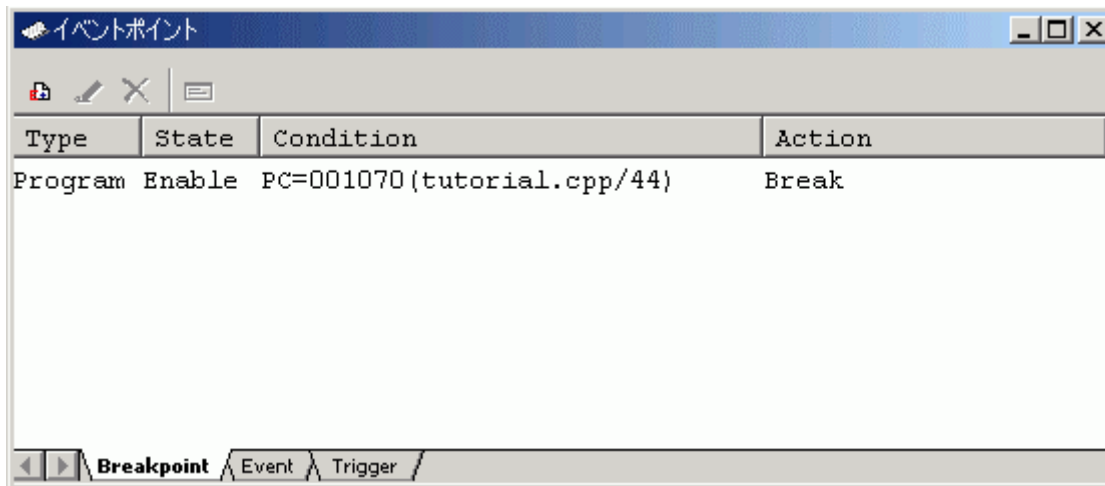


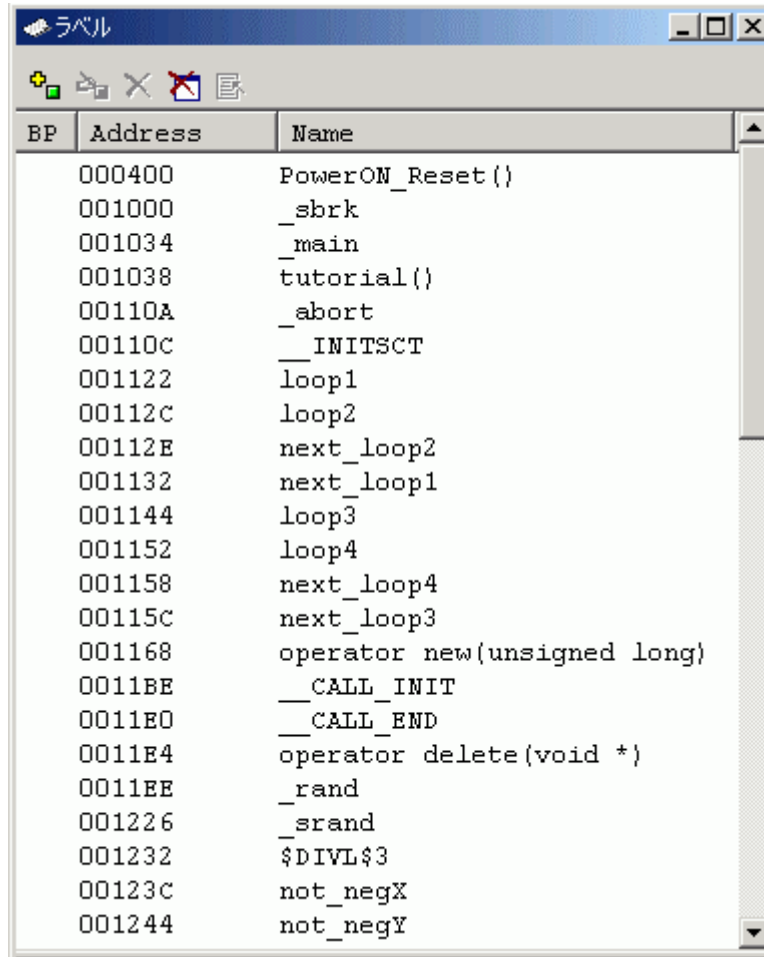
図 6-10 [イベントポイント]ウィンドウ

マウスの右ボタンで[イベントポイント]ウィンドウをクリックすると開くポップアップメニューにより、ブレークポイントの設定 / 変更、新しいブレークポイントの定義、およびブレークポイントの削除、有効 / 無効の選択ができます。

6.8 シンボルの参照

[ラベル]ウィンドウを使ってモジュール内のシンボル情報を表示させることができます。

[表示]メニューから[シンボル]サブメニューを選択し、[ラベル]を選択してください。[ラベル]ウィンドウが表示され、モジュール内のシンボル情報が参照できます。




The screenshot shows a window titled 'ラベル' (Labels) with a table of symbols. The table has three columns: 'BP', 'Address', and 'Name'. The symbols listed include various functions and variables such as PowerON_Reset, _sbrk, _main, tutorial, _abort, __INIT\$CT, loop1, loop2, next_loop2, next_loop1, loop3, loop4, next_loop4, next_loop3, operator new, __CALL__INIT, __CALL__END, operator delete, _rand, _srand, \$DIVL\$3, not_negX, and not_negY.

BP	Address	Name
	000400	PowerON_Reset()
	001000	_sbrk
	001034	_main
	001038	tutorial()
	00110A	_abort
	00110C	__INIT\$CT
	001122	loop1
	00112C	loop2
	00112E	next_loop2
	001132	next_loop1
	001144	loop3
	001152	loop4
	001158	next_loop4
	00115C	next_loop3
	001168	operator new(unsigned long)
	0011BE	__CALL__INIT
	0011E0	__CALL__END
	0011E4	operator delete(void *)
	0011EE	_rand
	001226	_srand
	001232	\$DIVL\$3
	00123C	not_negX
	001244	not_negY

図 6-11 [ラベル]ウィンドウ

6.9 メモリ内容の確認

ラベル名を指定することによって、ラベルが登録されているメモリの内容を[メモリ]ウィンドウで確認することができます。たとえば、以下のように、バイトサイズで_mainに対応するメモリ内容を確認します。

[表示 -> CPU -> メモリ...]を選択するか、[メモリ]ツールバーボタンをクリックして、[表示開始アドレス]ダイアログボックスを開いてください。

[表示開始アドレス]エディットボックスに"_main"を入力してください。



図 6-12 [表示開始アドレス]ダイアログボックス

[OK]ボタンをクリックしてください。指定されたメモリ領域を示す[メモリ]ウィンドウが表示されます。

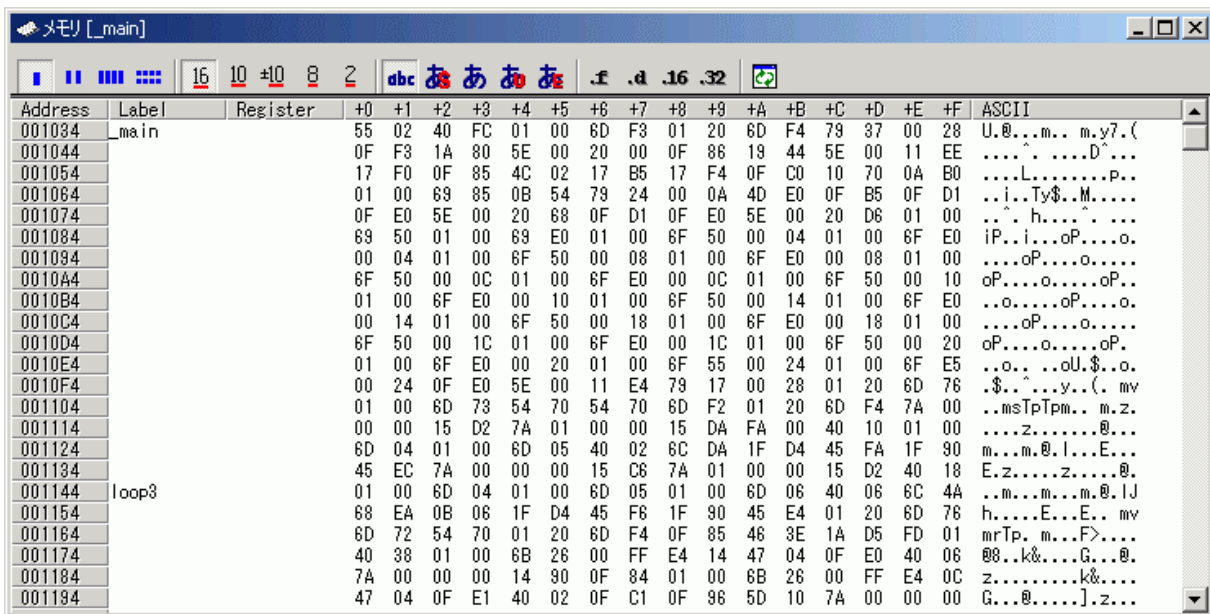


図 6-13 [メモリ]ウィンドウ

6.10 変数の参照

プログラムをステップ処理するとき、プログラムで使われる変数の値が変化することを確認できます。たとえば、以下の手順で、プログラムの初めに宣言した long 型の配列 a を見ることができます。

[エディタ]ウィンドウに表示されている配列 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[インスタントウォッチ]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。

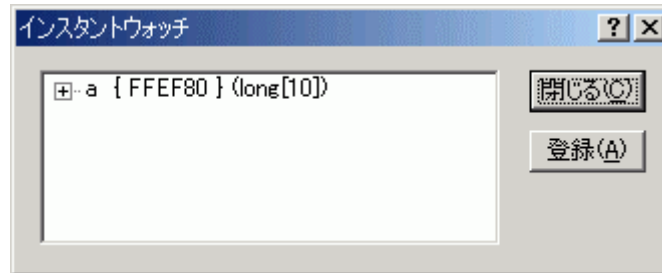


図 6-14 [インスタントウォッチ]ダイアログボックス

[登録]ボタンをクリックして、[ウォッチ]ウィンドウに変数を加えてください。

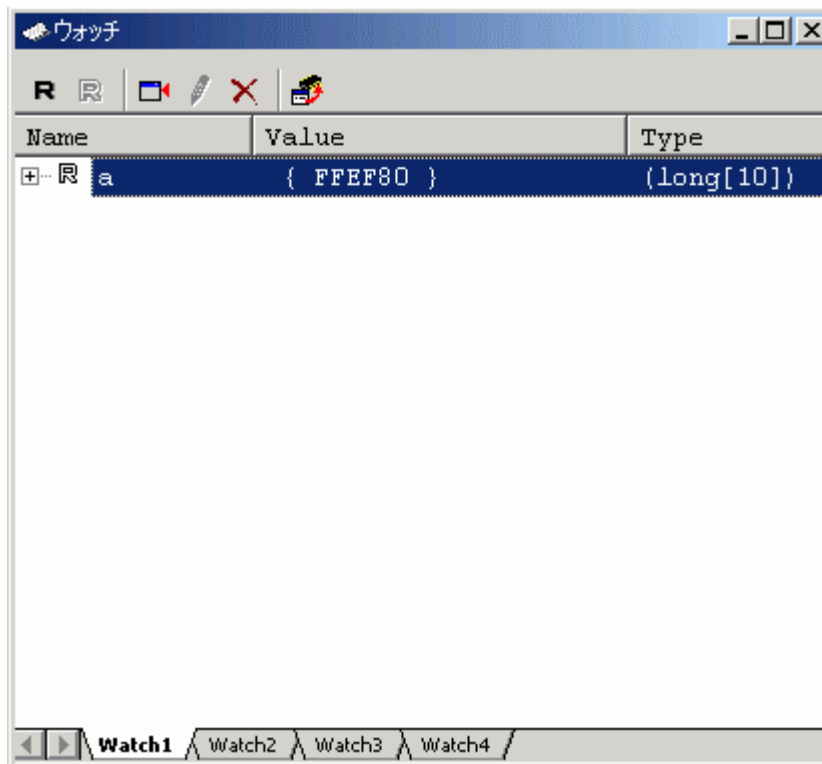


図 6-15 [ウォッチ]ウィンドウ (配列の表示)

6 チュートリアル

また、変数名を指定して、[ウォッチ]ウィンドウに変数を加えることもできます。

マウスの右ボタンで[ウォッチ]ウィンドウをクリックし、ポップアップメニューから[シンボル登録]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 6-16 [シンボル登録]ダイアログボックス

[変数または式]エディットボックスに変数 `i` を入力し、[OK]ボタンをクリックします。

[ウォッチ]ウィンドウに、`int` 型の変数 `i` が表示されます。

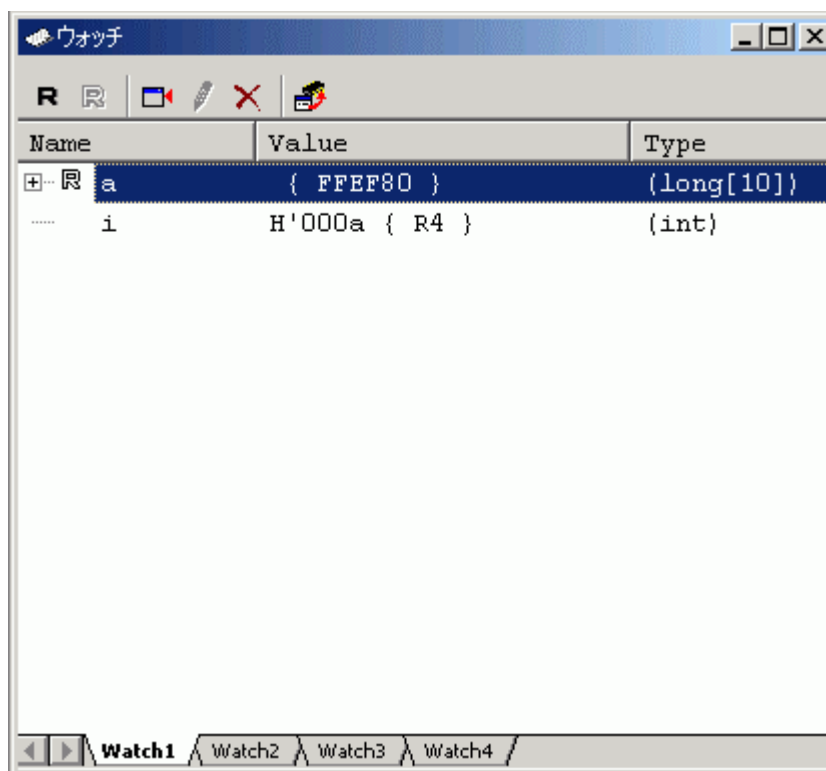


図 6-17 [ウォッチ]ウィンドウ (変数の表示)

[ウォッチ]ウィンドウの配列 a の左側にある”+”マークをクリックし、配列 a の各要素を参照することができます。

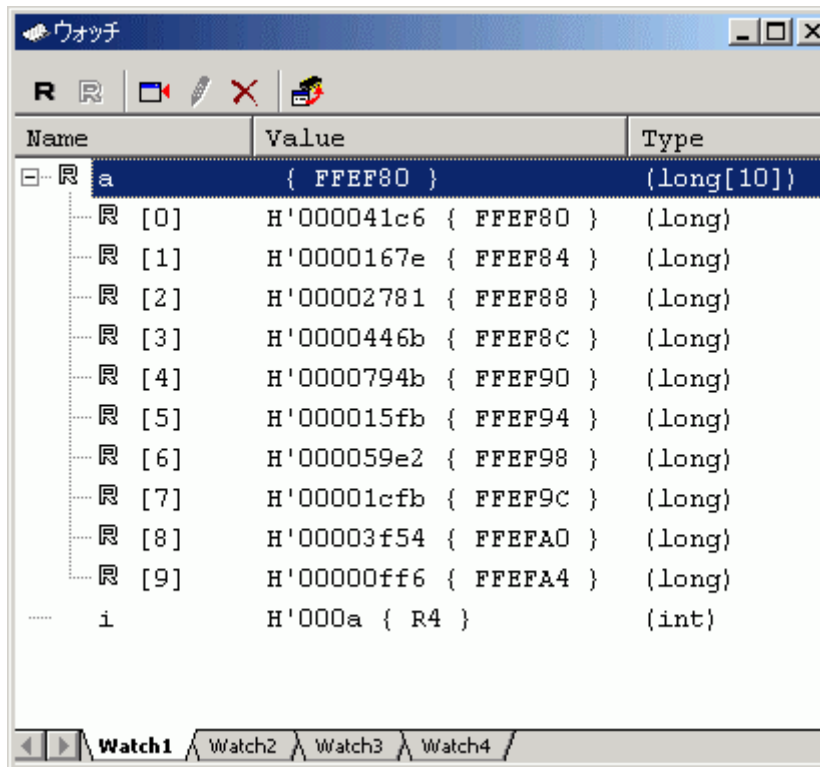


図 6-18 [ウォッチ]ウィンドウ (配列要素の表示)

6.11 ローカル変数の表示

[ローカル]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、tutorial 関数のローカル変数を調べます。

この関数は、4 つのローカル変数 a, j, i, p_sam を宣言します。

[表示]メニューから[シンボル]サブメニューを選択し、[ローカル]を選択してください。[ローカル]ウィンドウが表示されます。

[ローカル]ウィンドウには、現在のプログラムカウンタ (PC) が指している関数のローカル変数とその値が表示されます。

関数内にローカル変数が存在しない場合、[ローカル]ウィンドウに何も表示されません。

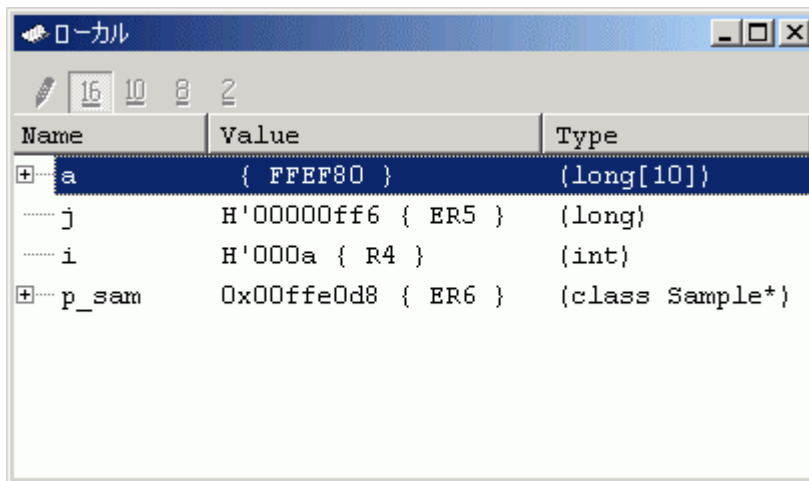


図 6-19 [ローカル]ウィンドウ

[ローカル]ウィンドウの配列 a の左側にある”+”マークをクリックし、配列 a の構成要素を表示させてください。sort 関数実行前と実行後の配列 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。

6.12 プログラムのステップ実行

High-performance Embedded Workshop は、プログラムのデバッグに有効な各種のステップコマンドを備えています。

表6-1 ステップオプション

項番	コマンド	説明
1	ステップイン	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）。
2	ステップオーバ	関数コールを1ステップとして、ステップ実行します。
3	ステップアウト	関数を抜け出し、関数を呼び出したプログラムの次のステートメントで停止します。
4	ステップ...	指定した速度で指定回数分ステップ実行します。

6.12.1 ステップインコマンドの実行

[ステップイン]コマンドはコール関数の中に入り、コール関数の先頭のステートメントで停止します。

sort 関数の中に入るために、[デバッグ]メニューから[ステップイン]を選択するか、またはツールバーの[ステップイン]ボタンをクリックしてください。



図 6-20 [ステップイン]ボタン

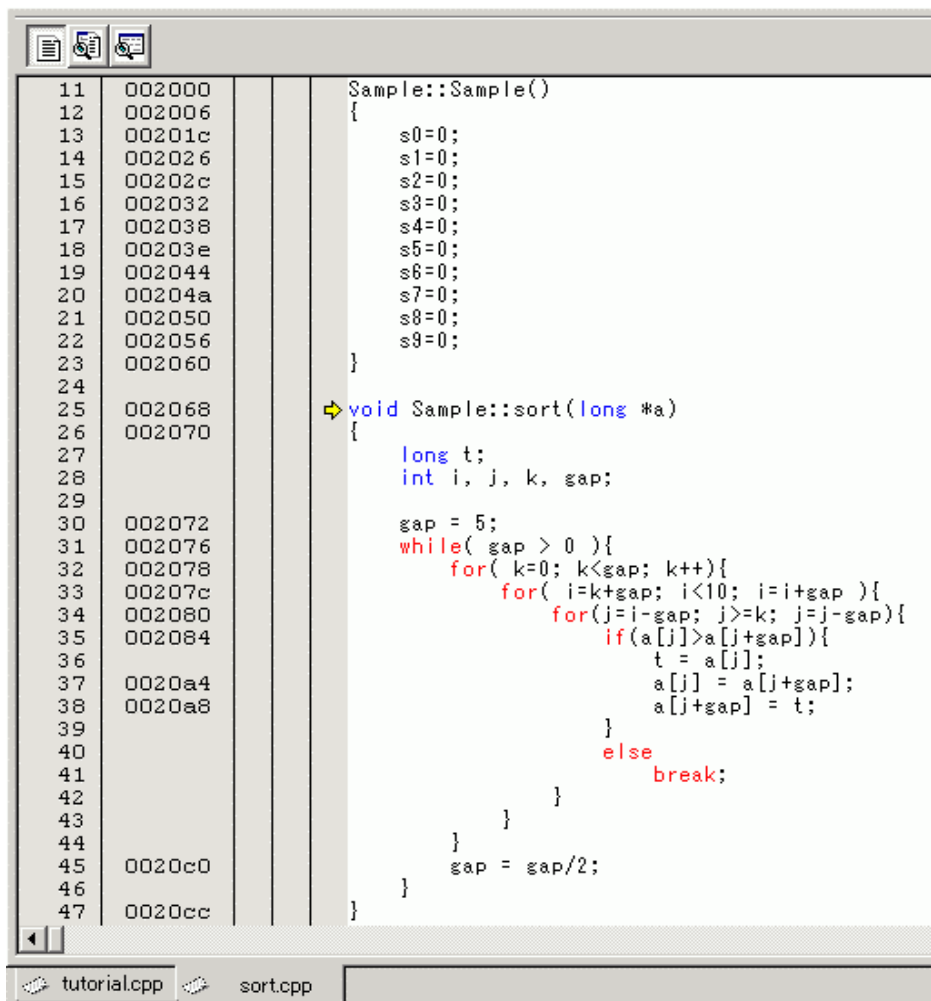


図 6-21 [エディタ]ウィンドウ (ステップイン)

[エディタ]ウィンドウの強調表示が、sort 関数の先頭のステートメントに移動します。

6.12.2 ステップアウトコマンドの実行

[ステップアウト]コマンドはコール関数の中から抜け出し、コール元プログラムの次のステートメントで停止します。

sort 関数の中から抜け出すために、[デバッグ]メニューから[ステップアウト]を選択するか、またはツールバーの[ステップアウト]ボタンをクリックしてください。



図 6-22 [ステップアウト]ボタン

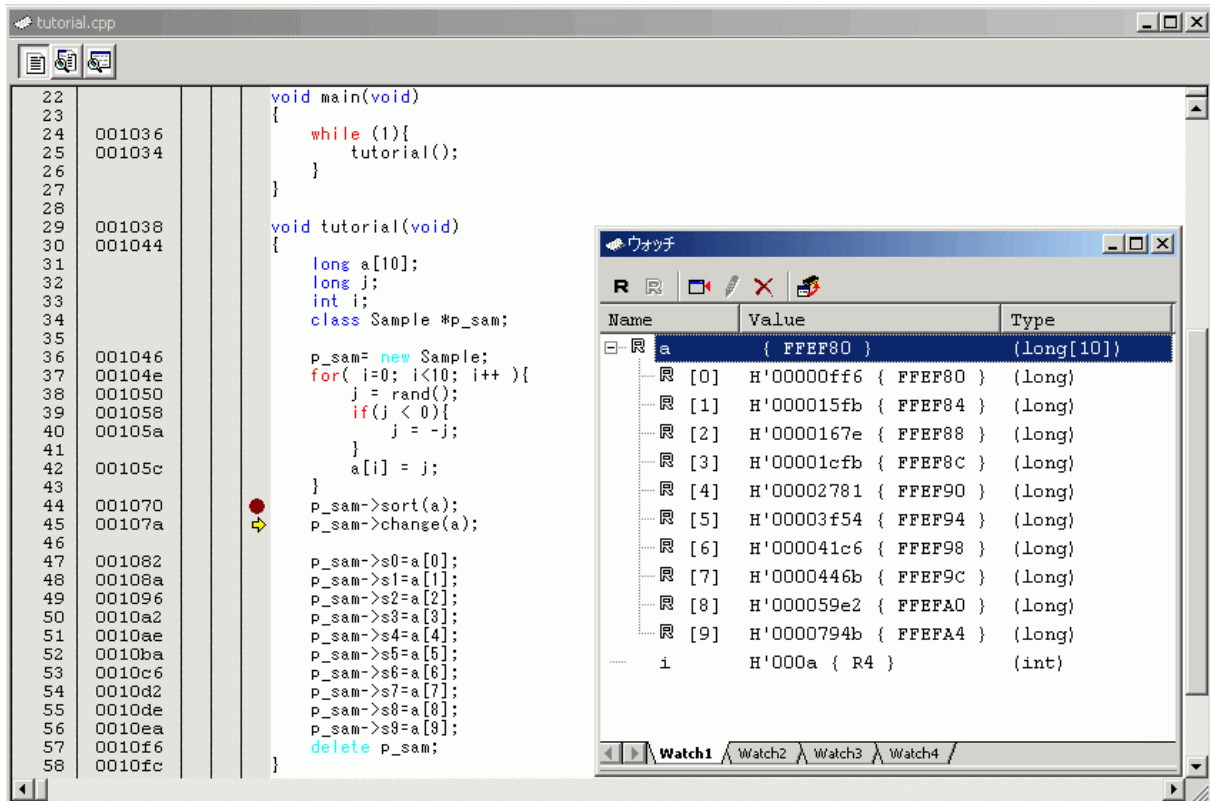


図 6-23 [High-performance Embedded Workshop]ウィンドウ (ステップアウト)

[ウォッチ]ウィンドウに表示された変数 a のデータが昇順にソートされます。

6.12.3 ステップオーバーコマンドの実行

[ステップオーバー]コマンドは関数コールを1ステップとして実行して、メインプログラムの次のステートメントで停止します。

change 関数中のステートメントを一度にステップ実行するために、[デバッグ]メニューから[ステップオーバー]を選択するか、またはツールバーの[ステップオーバー]ボタンをクリックしてください。



図 6-24 [ステップオーバー]ボタン

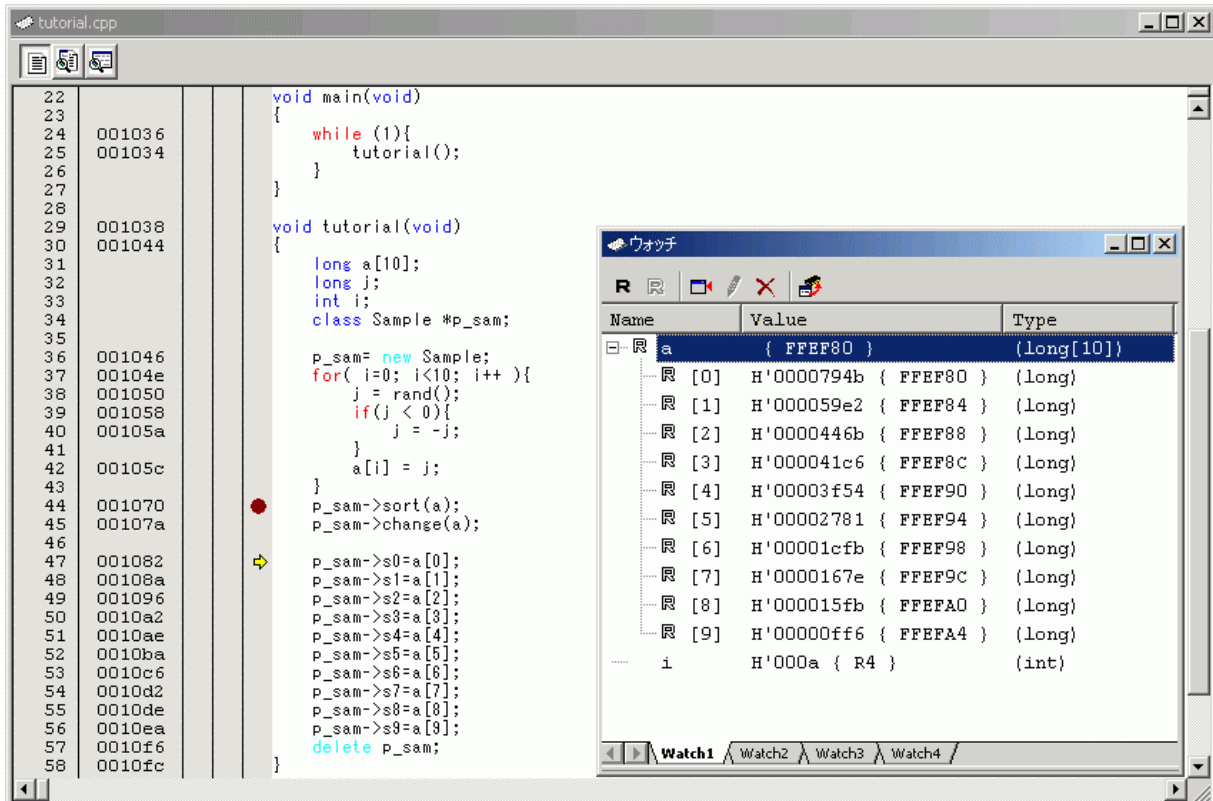


図 6-25 [High-performance Embedded Workshop]ウィンドウ (ステップオーバー)

6.13 プログラムの強制ブレーク

High-performance Embedded Workshop は、プログラムを強制的にブレークすることができます。

ブレークポイントを全て解除してください。

tutorial 関数の残り部分を実行するために、[デバッグ]メニューから[実行]を選択するか、ツールバー上の[実行]ボタンを選択してください。



図 6-26 [実行]ボタン

プログラムは無限ループ処理を実行していますので、強制ブレークするために、[デバッグ]メニューから[プログラムの停止]を選択するか、ツールバー上の[停止]ボタンを実行してください。



図 6-27 [停止]ボタン

6.14 MCUのリセット

MCU をリセットすることにより、内蔵 I/O レジスタの初期化、およびリセットベクタに設定されたアドレスを PC に設定します。

MCU のリセットを行うには、[デバッグ]メニューから[CPUのリセット]を選択するか、ツールバー上の[CPU リセット]ボタンを選択してください。



図 6-28 CPU リセットボタン

また、リセット直後からプログラムを実行する場合には、[デバッグ]メニューから[リセット後実行]を選択するか、ツールバー上の[リセット後実行]ボタンを選択してください。



図 6-29 リセット後実行ボタン

【注】 本チュートリアルプログラムは、リセットベクタからプログラムを実行しても問題なく実行できるように作成しています。

6.15 ブレーク機能

E6000 エミュレータは、ソフトウェアブレーク機能とイベントポイントによるブレーク機能を持っています。ソフトウェアブレークポイントの設定、およびイベントポイントの設定は[イベントポイント]ウィンドウでそれぞれ行うことができます。

以下にブレーク機能の概要と設定方法について説明します。

6.15.1 ソフトウェアブレーク機能

E6000 エミュレータは、最大 256 ポイントまでソフトウェアブレークポイントを設定することができます。

[表示]メニューから[コード]サブメニューを選択し、[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。

[Breakpoint]シートを開きます。

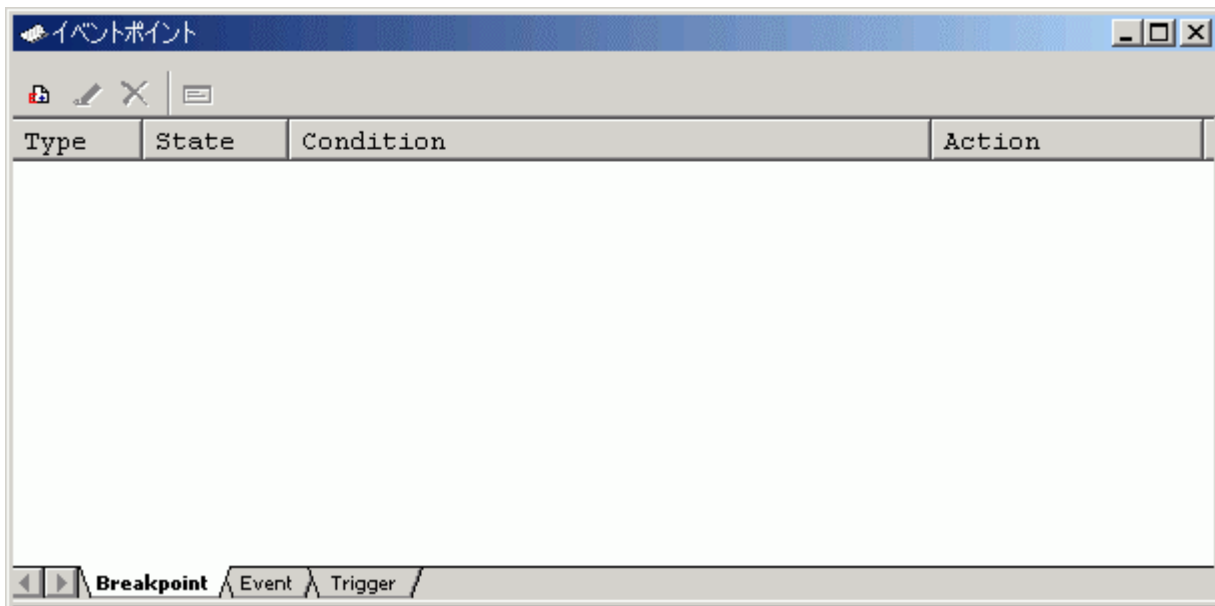


図 6-30 [イベントポイント]ウィンドウ (ソフトウェアブレーク設定前)

6 チュートリアル

マウスの右ボタンで[イベントポイント]ウィンドウをクリックし、ポップアップメニューから[追加...]を選択してください。

[Breakpoint/Event Properties]ダイアログボックスが表示されます。

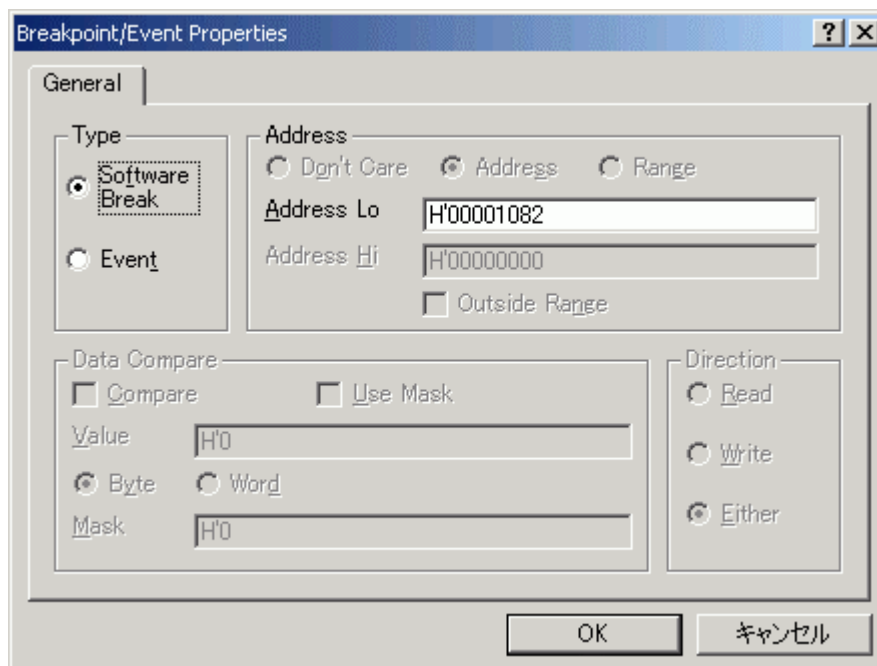


図 6-31 [Breakpoint/Event Properties]ダイアログボックス

[Type]グループボックスの[Software Break]ラジオボタンをチェックします。

[Address]グループボックスの[Address Lo]エディットボックスにtutorial関数内の「p_sam->s0=a[0];」と記述されている行のアドレスを[エディタ]ウィンドウで参照し入力してください。本例ではH'00001082を入力します。

【注】 本ダイアログボックスは、製品ごとに異なります。各製品の内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

[OK]ボタンをクリックしてください。

[イベントポイント]ウィンドウには、設定されたソフトウェアブレイクポイントが表示されます。

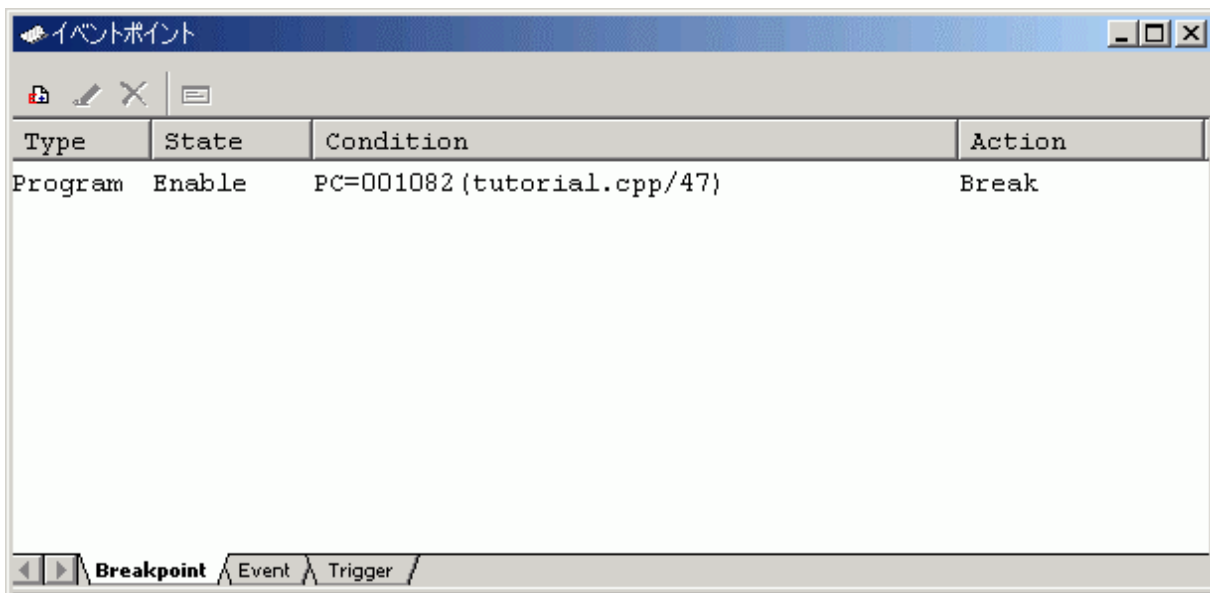


図 6-32 [イベントポイント]ウィンドウ (ソフトウェアブレイク設定時)

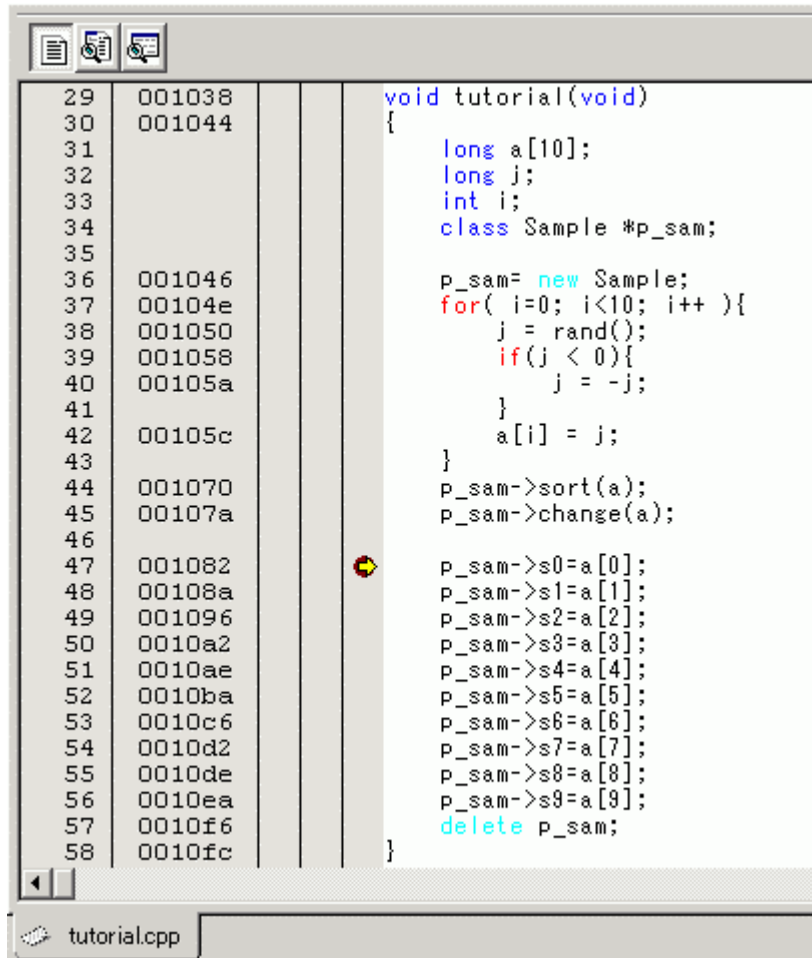
【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

[イベントポイント]ウィンドウを閉じてください。

チュートリアルプログラムをブレイクポイントで停止させるため、[デバッグ]メニューから[リセット後実行]を選択してください。

6 チュートリアル

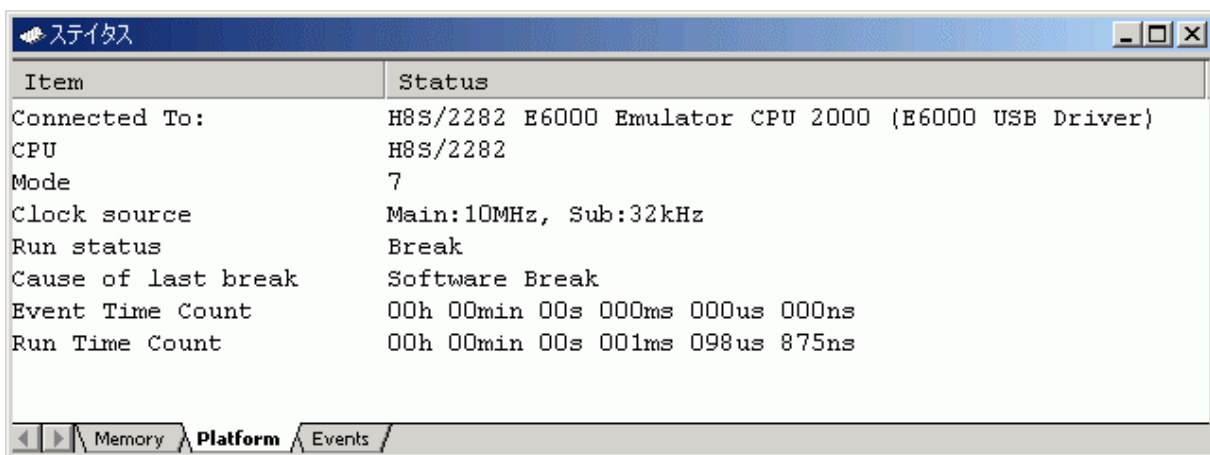
設定したブレークポイントまで、プログラムを実行して停止します。



```
29 001038 void tutorial(void)
30 001044 {
31     long a[10];
32     long j;
33     int i;
34     class Sample *p_sam;
35
36 001046     p_sam= new Sample;
37 00104e     for( i=0; i<10; i++ ){
38 001050         j = rand();
39 001058         if(j < 0){
40 00105a             j = -j;
41         }
42 00105c         a[i] = j;
43     }
44 001070     p_sam->sort(a);
45 00107a     p_sam->change(a);
46
47 001082     p_sam->s0=a[0];
48 00108a     p_sam->s1=a[1];
49 001096     p_sam->s2=a[2];
50 0010a2     p_sam->s3=a[3];
51 0010ae     p_sam->s4=a[4];
52 0010ba     p_sam->s5=a[5];
53 0010c6     p_sam->s6=a[6];
54 0010d2     p_sam->s7=a[7];
55 0010de     p_sam->s8=a[8];
56 0010ea     p_sam->s9=a[9];
57 0010f6     delete p_sam;
58 0010fc }
```

図 6-33 実行停止時の[エディタ]ウィンドウ (ソフトウェアブレーク)

[ステータス]ウィンドウの表示内容は、以下のようになります。



Item	Status
Connected To:	H8S/2282 E6000 Emulator CPU 2000 (E6000 USB Driver)
CPU	H8S/2282
Mode	7
Clock source	Main:10MHz, Sub:32kHz
Run status	Break
Cause of last break	Software Break
Event Time Count	00h 00min 00s 000ms 000us 000ns
Run Time Count	00h 00min 00s 001ms 098us 875ns

図 6-34 [ステータス]ウィンドウの表示内容 (ソフトウェアブレーク)

【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

6.15.2 イベントポイントによるブレーク機能

イベントポイントのイベントチャンネル1 (Ch1) を用いて、指定したイベントが5回成立した場合にブレークする方法を説明します。

[表示]メニューから[コード]サブメニューを選択し、[イベントポイント]を選択してください。[イベントポイント]ウィンドウが表示されます。

先ほど設定したブレークポイントを削除します。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているブレークポイントをすべて解除してください。

次は Event を設定します。[Event]タブをクリックしてください。

イベントポイントは、イベントチャンネル8箇所、範囲チャンネル4箇所の最大12箇所まで独立に条件を設定することができます。ここでは、イベントチャンネル1 (Ch1) を設定します。

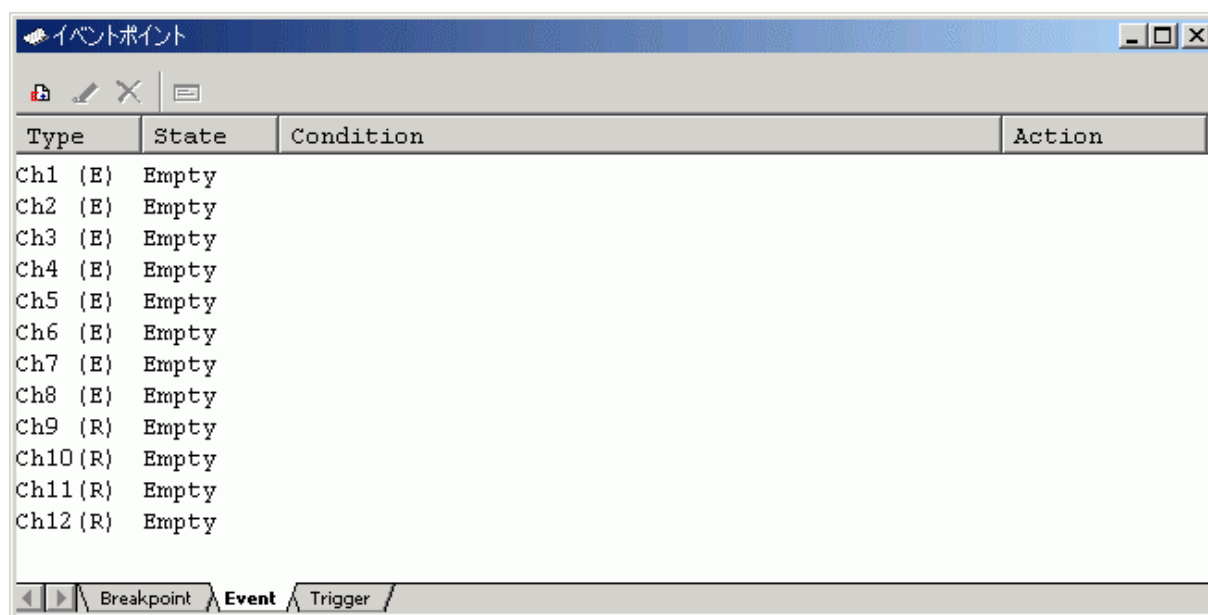


図 6-35 [イベントポイント]ウィンドウ (イベントチャンネル[Ch1])

[イベントポイント]ウィンドウ内の Ch1 行を選択してください。Ch1 行が強調表示されますので、ダブルクリックしてください。[Breakpoint/Event Properties]ダイアログボックスが表示されます。

6 チュートリアル

[General]ページの各グループボックスを以下のように設定します。

- ・ [Type]グループボックスの[Event]ラジオボタンを選択します。
- ・ [Address]グループボックスの[Address]ラジオボタンを選択し、[Address Lo]エディットボックスに tutorial 関数内の「a[i]=j;」と記述されている行のアドレスを[エディタ]ウィンドウで参照し入力してください。本例では H'0000105c を入力します。

[Action]ページの[Required number of event occurrences]エディットボックスにイベント成立回数として D'5 を入力してください。

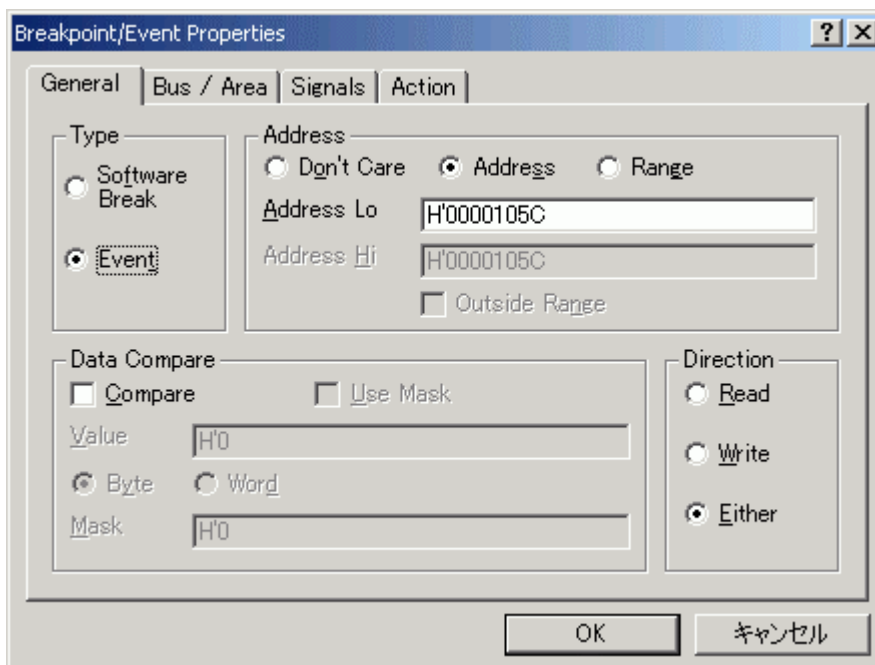


図 6-36 [General]ページ ([Breakpoint/Event Properties]ダイアログボックス)

[OK]ボタンをクリックしてください。

[イベントポイント]ウィンドウに以下が表示されます。

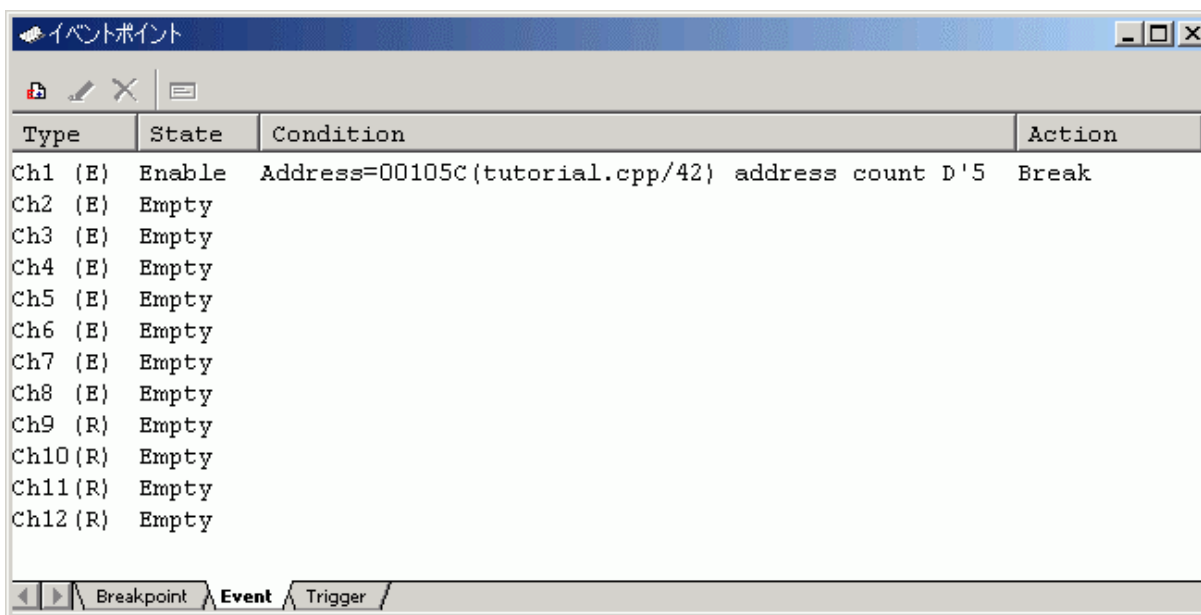



図 6-37 [イベントポイント]ウィンドウ (設定完了時)

【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

チュートリアルプログラムをブレークポイントで停止させるため、[デバッグ]メニューから[リセット後実行]を選択してください。

Ch1 の条件まで、プログラムを実行して停止します。

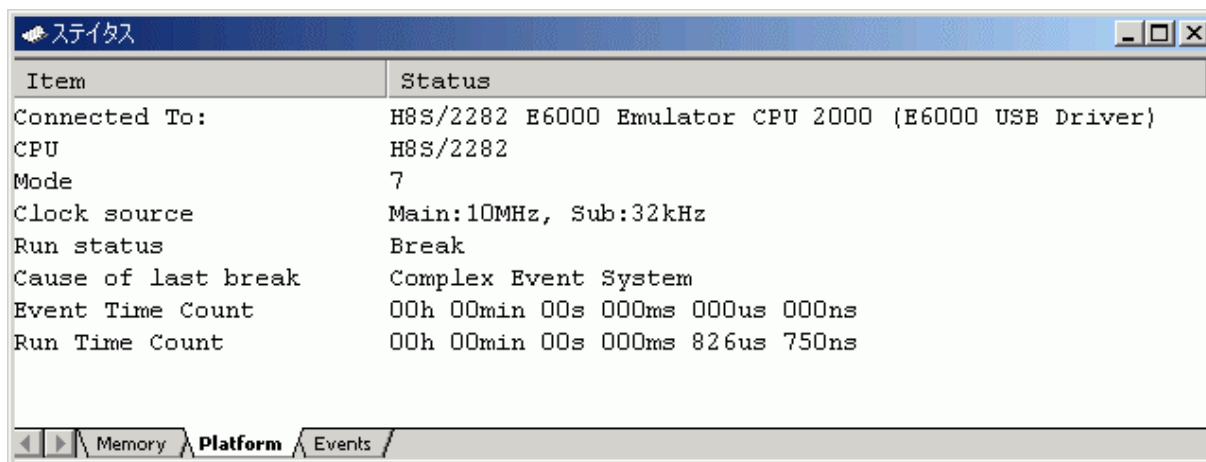


```
29 001038 void tutorial(void)
30 001044 {
31
32 long a[10];
33 long j;
34 int i;
35 class Sample *p_sam;
36
37 001046 p_sam= new Sample;
38 00104e for( i=0; i<10; i++){
39 001050     j = rand();
40 00105a     if(j < 0){
41         j = -j;
42 00105c     }
43         a[i] = j;
44     }
45 001070 p_sam->sort(a);
46 00107a p_sam->change(a);
47
48 001082 p_sam->s0=a[0];
49 00108a p_sam->s1=a[1];
50 001096 p_sam->s2=a[2];
51 0010a2 p_sam->s3=a[3];
52 0010ae p_sam->s4=a[4];
53 0010ba p_sam->s5=a[5];
54 0010c6 p_sam->s6=a[6];
55 0010d2 p_sam->s7=a[7];
56 0010de p_sam->s8=a[8];
57 0010ea p_sam->s9=a[9];
58 0010f6 delete p_sam;
59 0010fc }
```

図 6-38 実行停止時の[エディタ]ウィンドウ

6 チュートリアル

[ステイタス]ウィンドウの表示内容は、以下のようになります。



The screenshot shows a window titled 'ステイタス' (Status) with a table of system information. The table has two columns: 'Item' and 'Status'. The data is as follows:

Item	Status
Connected To:	H8S/2282 E6000 Emulator CPU 2000 (E6000 USB Driver)
CPU	H8S/2282
Mode	7
Clock source	Main:10MHz, Sub:32kHz
Run status	Break
Cause of last break	Complex Event System
Event Time Count	00h 00min 00s 000ms 000us 000ns
Run Time Count	00h 00min 00s 000ms 826us 750ns

At the bottom of the window, there are three tabs: 'Memory', 'Platform', and 'Events'. The 'Platform' tab is currently selected.

図 6-39 [ステイタス]ウィンドウの表示内容

[ウォッチ]ウィンドウから変数*i*の値を参照します。変数*i*の値は4となっており、イベントが5回成立した事によってブレークが発生したことがわかります。

【注】 本ウィンドウで表示される内容は、製品ごとに異なります。各製品の表示内容については、「8 本製品固有のソフトウェア仕様」またはオンラインヘルプを参照してください。

設定したイベントポイントを解除します。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除してください。

6.16 トレース機能

E6000 エミュレータのトレース機能がもつリアルタイムトレースバッファでは、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。リアルタイムトレースバッファの内容は[Trace]ウィンドウに表示されません。

[表示]メニューから[コード]サブメニューを選択し、[トレース]を選択してください。[Trace]ウィンドウが表示されます。

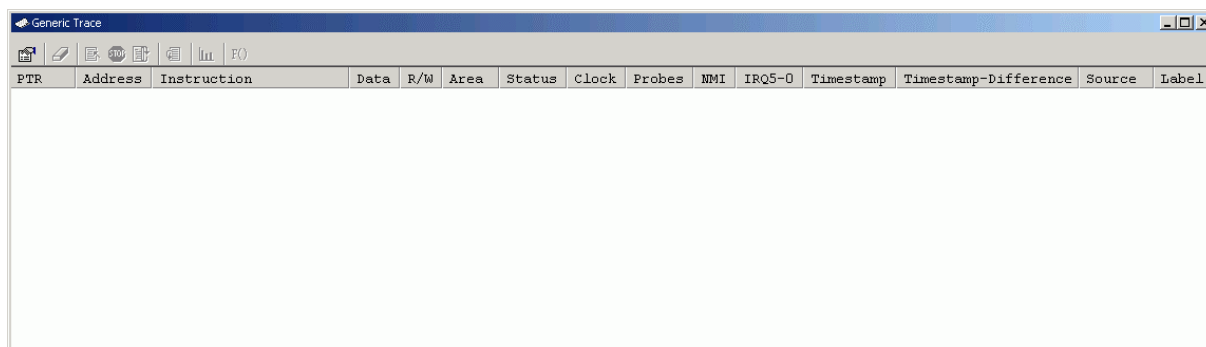


図 6-40 [Trace]ウィンドウ

[Trace]ウィンドウにトレース情報が表示されている場合は、マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。

次にトレース機能の概要と設定方法について説明します。

6.16.1 トレースの表示（タイムスタンプ無効時）

指定したアドレスに対するリード/ライトサイクルをトレース表示します。

- (1)マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[設定...]を選択してください。[Trace Acquisition]ダイアログボックスが表示されます。

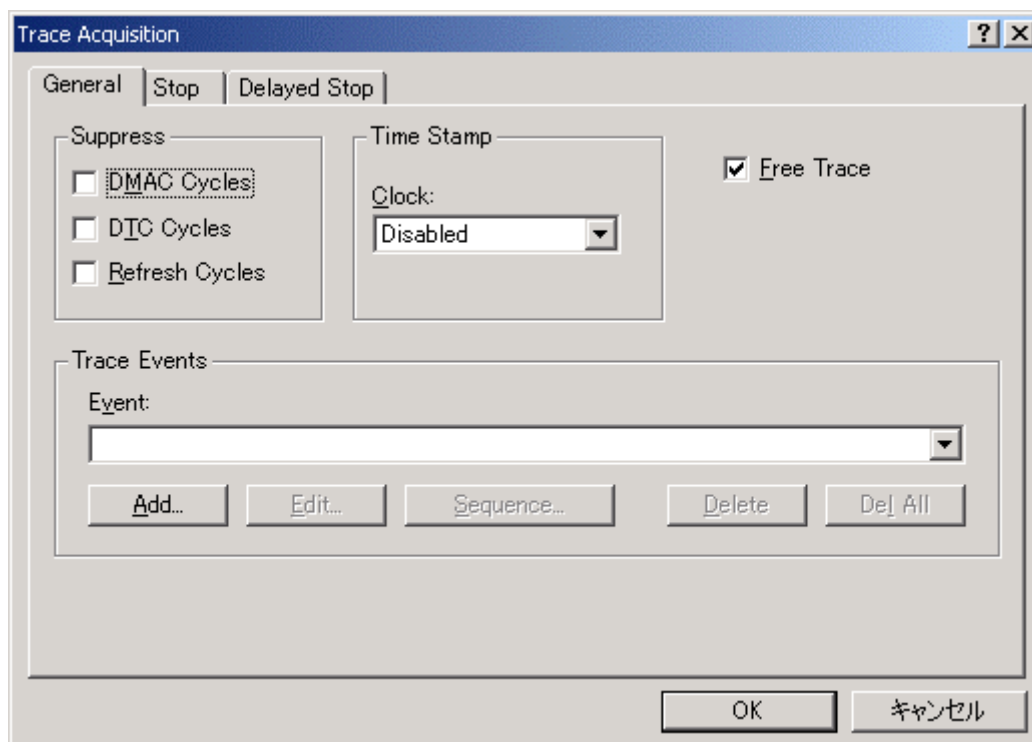


図 6-41 [Trace Acquisition]ダイアログボックス

- (2) トレースを取得するアドレスを指定する為に、イベントを登録します。[General]ページの[Trace Events]グループボックスにある[Add...]ボタンをクリックして[Breakpoint/Event Properties]ダイアログボックスを表示します。

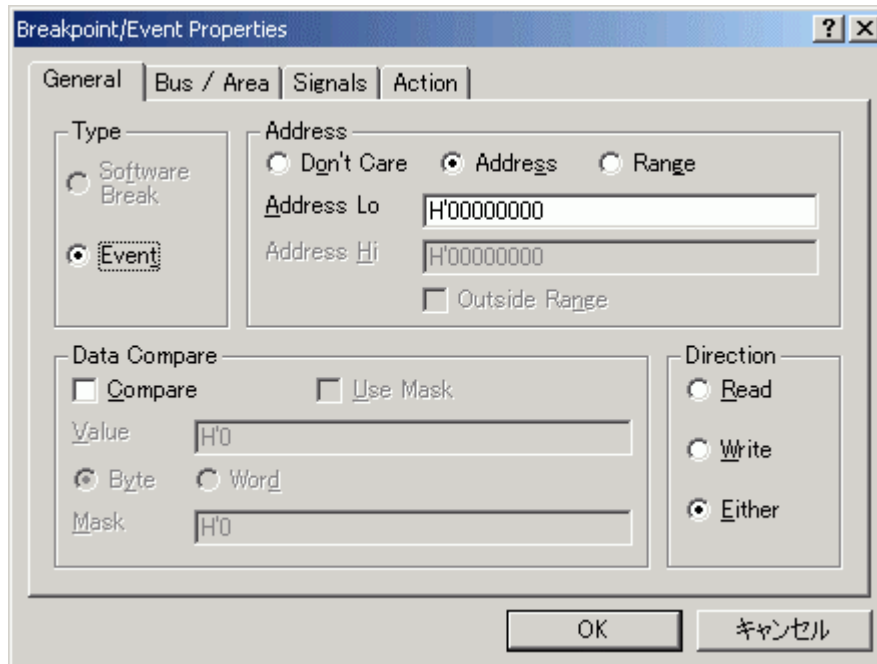


図 6-42 [Breakpoint/Event Properties]ダイアログボックス

- (3) [Breakpoint/Event Properties]ダイアログボックス-[General]ページの[Address]グループボックスの[Address Lo]エディットボックスに tutorial 関数内の「a[i]=j;」と記述されている行のアドレスを[エディタ]ウィンドウで参照し入力してください。本例では H'0000105c を入力します。これでアドレスの指定は終了です。[OK]ボタンをクリックし、[Breakpoint/Event Properties]ダイアログボックスを閉じます。

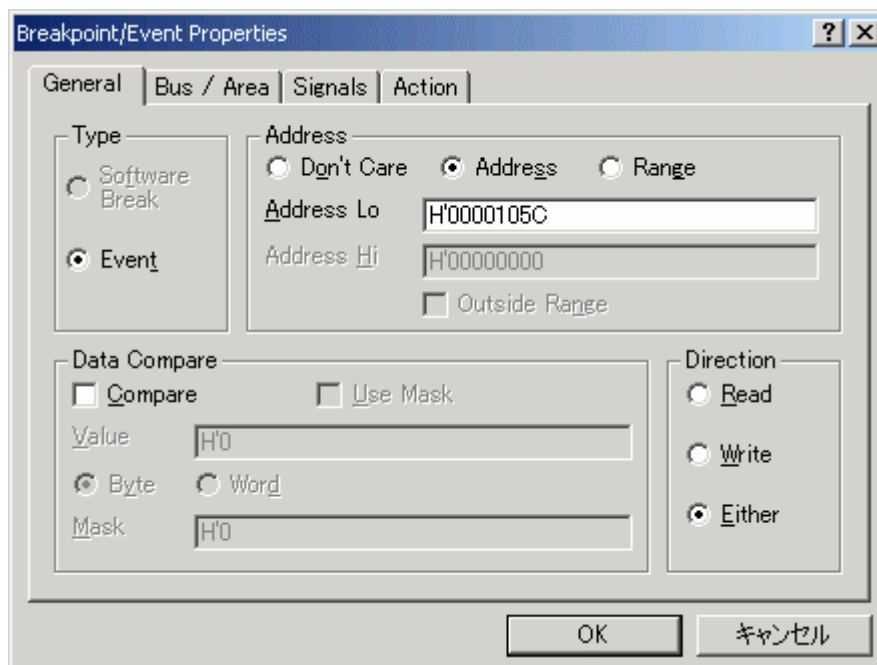


図 6-43 [Breakpoint/Event Properties]ダイアログボックス (イベント設定後)

6 チュートリアル

(4)[Trace Acquisition]ダイアログボックス-[General]ページの[Trace Events]グループボックスにある[Event]ドロップダウンリストボックスに設定したイベントの内容が表示されます。

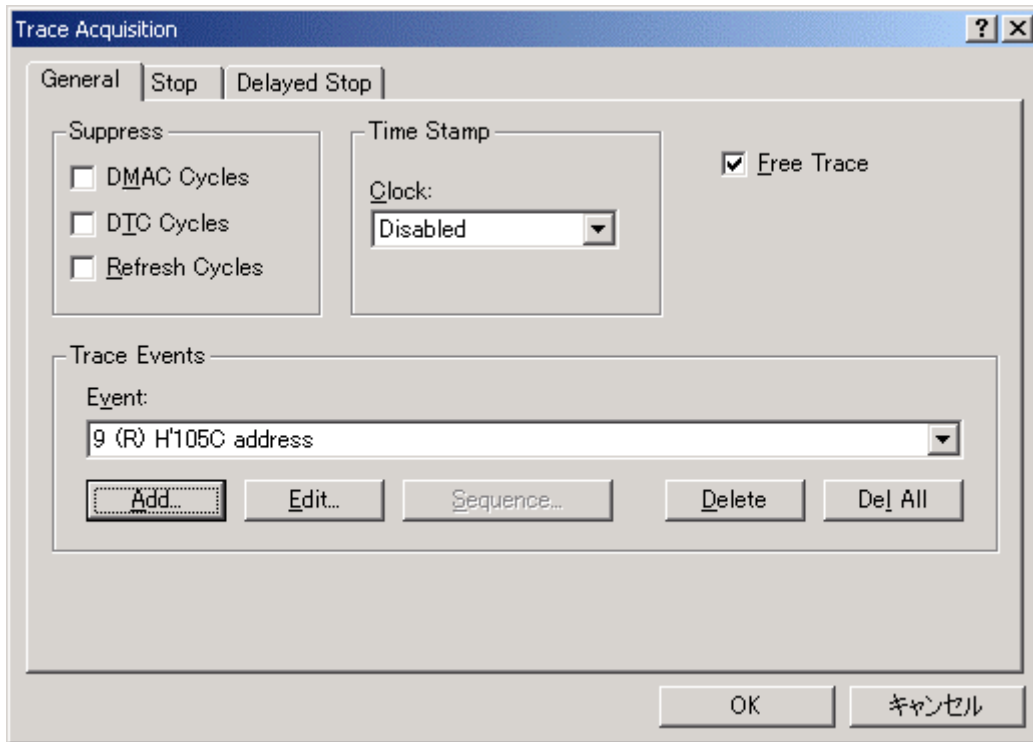


図 6-44 [Trace Acquisition]ダイアログボックス (イベント追加)

(5)設定したイベントを有効とする為に、[General]ページの[Free Trace]チェックボックスのチェックを解除します。チェックを解除すると新たに[1]～[4]のページが追加されます。

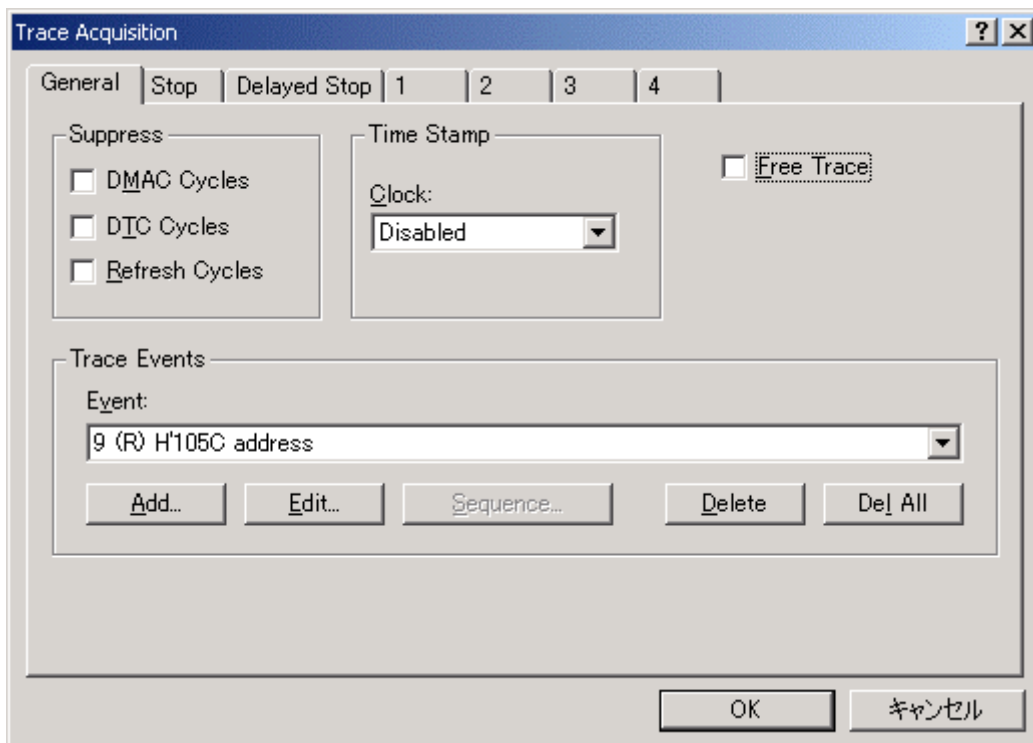


図 6-45 [Trace Acquisition]ダイアログボックス (ページ追加)

- (6)[1]ページを選択し、[Conditions]グループボックスの[Range]ラジオボタンをクリックします。クリックと同時に[Range Event]ドロップダウンリストボックスと[Edit...]ボタンが表示されます。

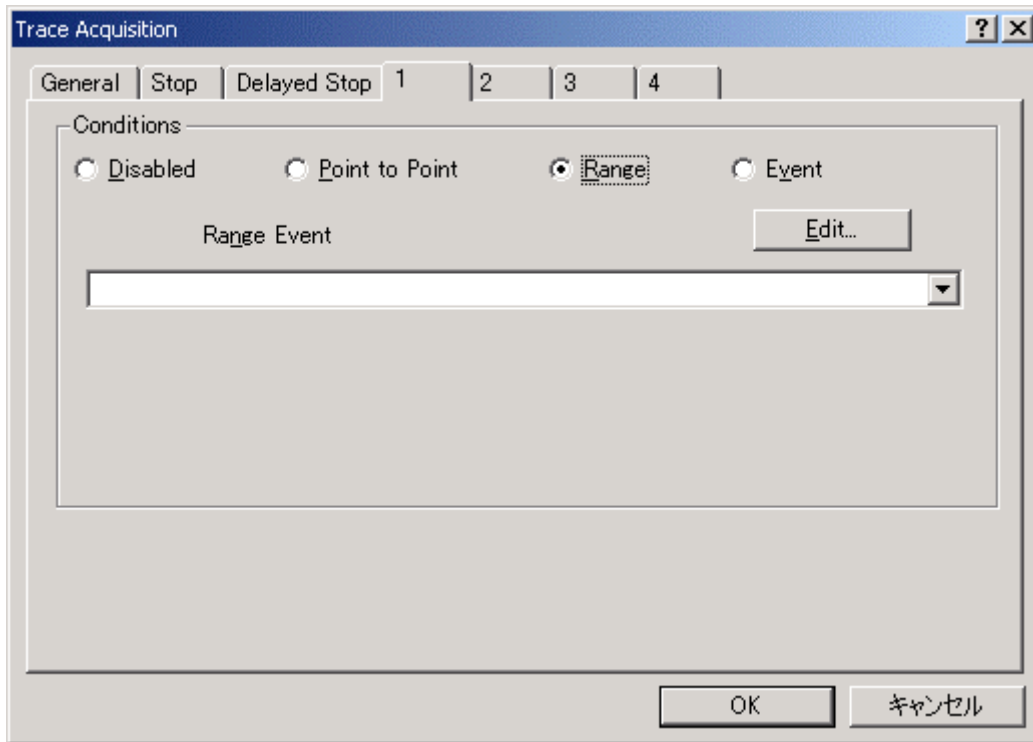


図 6-46 [Trace Acquisition]ダイアログボックス (1 ページ表示)

- (7)[Range Event]ドロップダウンリストボックスから、先ほど登録したイベントを選択します。これでイベントが有効となりました。[OK]ボタンをクリックし、トレースの設定は終了です。

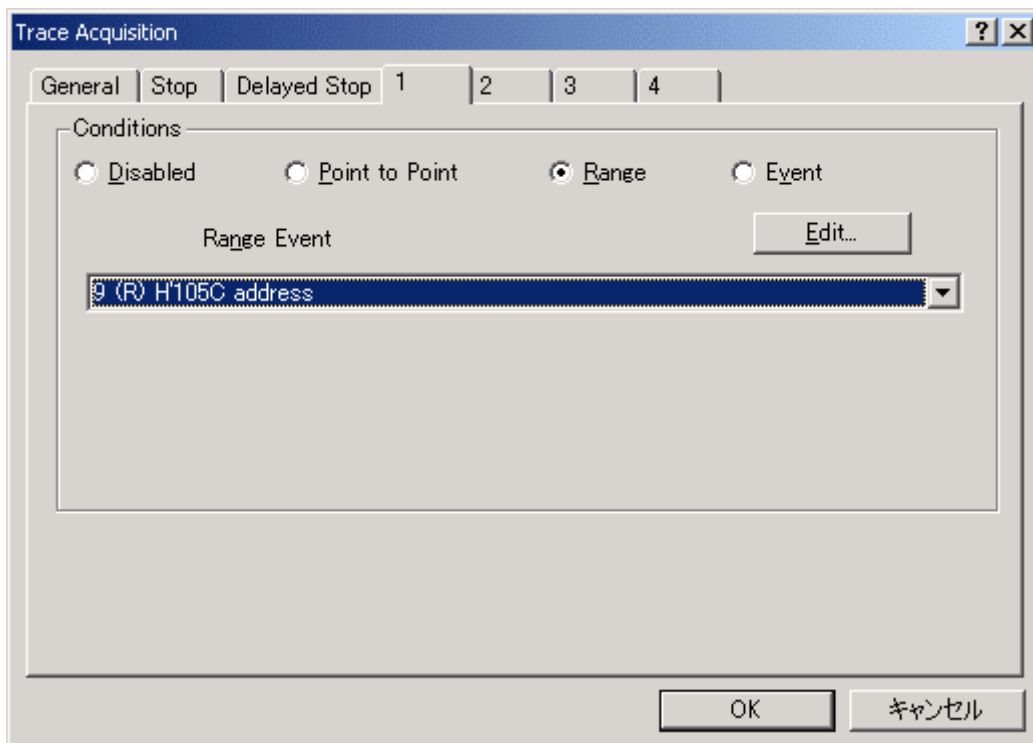
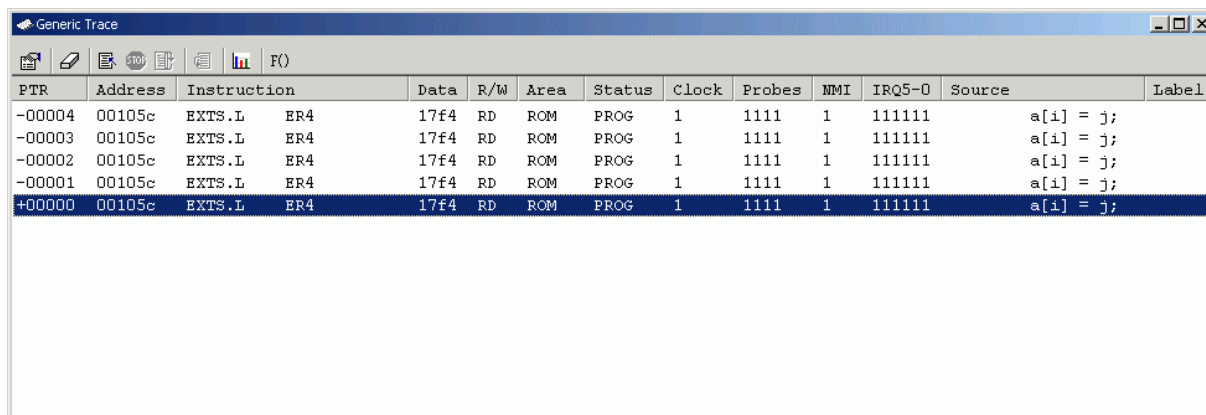


図 6-47 [Trace Acquisition]ダイアログボックス (設定完了時)

6 チュートリアル

(8)指定した tutorial 関数内の「a[i]=j;」と記述されている行のアドレス（本例ではH'0000105c）が5回実行された時点でブレークするように設定します。（「6.15.2 イベントポイントによるブレーク機能」参照）

(9)[デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[Trace]ウィンドウに以下の内容が表示されます。



PTR	Address	Instruction	Data	R/W	Area	Status	Clock	Probes	NMI	IRQ5-0	Source	Label
-00004	00105c	EXTS.L	ER4	17f4	RD	ROM	PROG	1	1111	1	111111	a[i] = j;
-00003	00105c	EXTS.L	ER4	17f4	RD	ROM	PROG	1	1111	1	111111	a[i] = j;
-00002	00105c	EXTS.L	ER4	17f4	RD	ROM	PROG	1	1111	1	111111	a[i] = j;
-00001	00105c	EXTS.L	ER4	17f4	RD	ROM	PROG	1	1111	1	111111	a[i] = j;
+00000	00105c	EXTS.L	ER4	17f4	RD	ROM	PROG	1	1111	1	111111	a[i] = j;

図 6-48 [Trace]ウィンドウ（結果表示）

必要ならば、タイトルバーの下のヘッダバーをドラッグして、コラムの幅を調節してください。

(10)設定したイベントポイントを解除し、トレース情報をクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。

6.16.2 トレースの表示（タイムスタンプ有効時）

指定した範囲のメモリに対するライトサイクルを、タイムスタンプを設定し、トレース表示します。

- (1) マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[設定...]を選択してください。[Trace Acquisition]ダイアログボックスが表示されます。（図 6-41 [Trace Acquisition]ダイアログボックス）
- (2) トレースを取得するメモリの範囲を指定する為に、イベントを登録します。[General]ページの[Trace Events]グループボックスにある[Add...]ボタンをクリックして[Breakpoint/Event Properties]ダイアログボックスを表示します。（図 6-42 [Breakpoint/Event Properties]ダイアログボックス）
- (3) [Breakpoint/Event Properties]ダイアログボックス-[General]ページの[Address]グループボックスの[Range]ラジオボタンをクリックし、[Address Lo]エディットボックスに tutorial 関数で定義している変数 a が割りついているアドレスを[ローカル]ウィンドウで参照し入力してください。本例では H'00FFEF80 を入力します。また、[Address Hi]エディットボックスには[Address Lo]エディットボックスに入力したアドレスに H'27 を加算したアドレスを入力してください。本例では、H'00FFEFA7 を入力します。この設定により tutorial 関数の変数 a のメモリ領域を指定したことになります。
- (4) 同様に[Direction]グループボックスの[Write]ラジオボタンをクリックし、設定した範囲に対するライトサイクルを指定します。これでメモリの範囲指定は終了です。[OK]ボタンをクリックし、[Breakpoint/Event Properties]ダイアログボックスを閉じます。

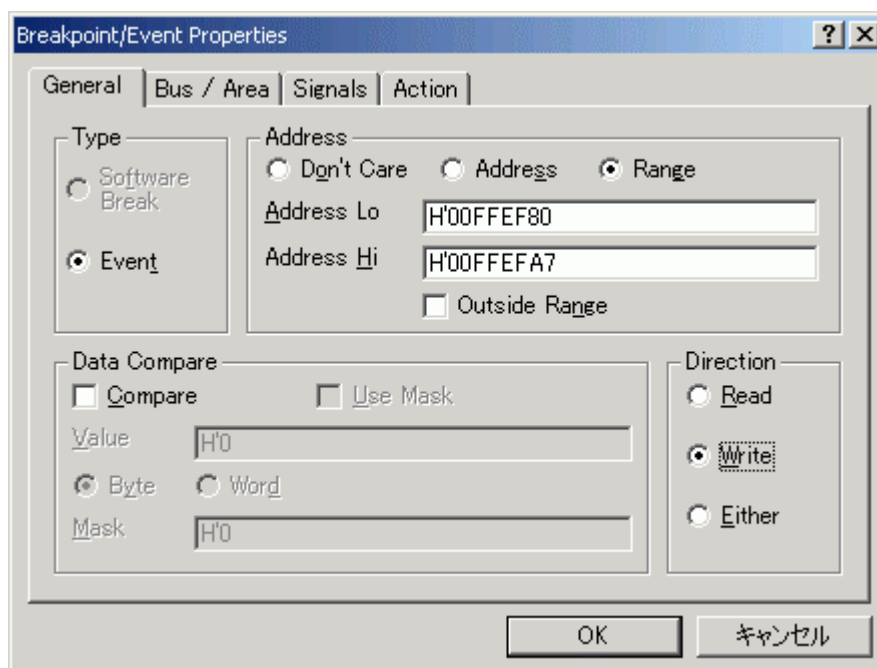


図 6-49 [Breakpoint/Event Properties]ダイアログボックス（イベント設定後）

6 チュートリアル

- (5)[Trace Acquisition]ダイアログボックス-[General]ページの[Trace Events]グループボックスにある[Event]ドロップダウンリストボックスに設定したイベントの内容が表示されます。

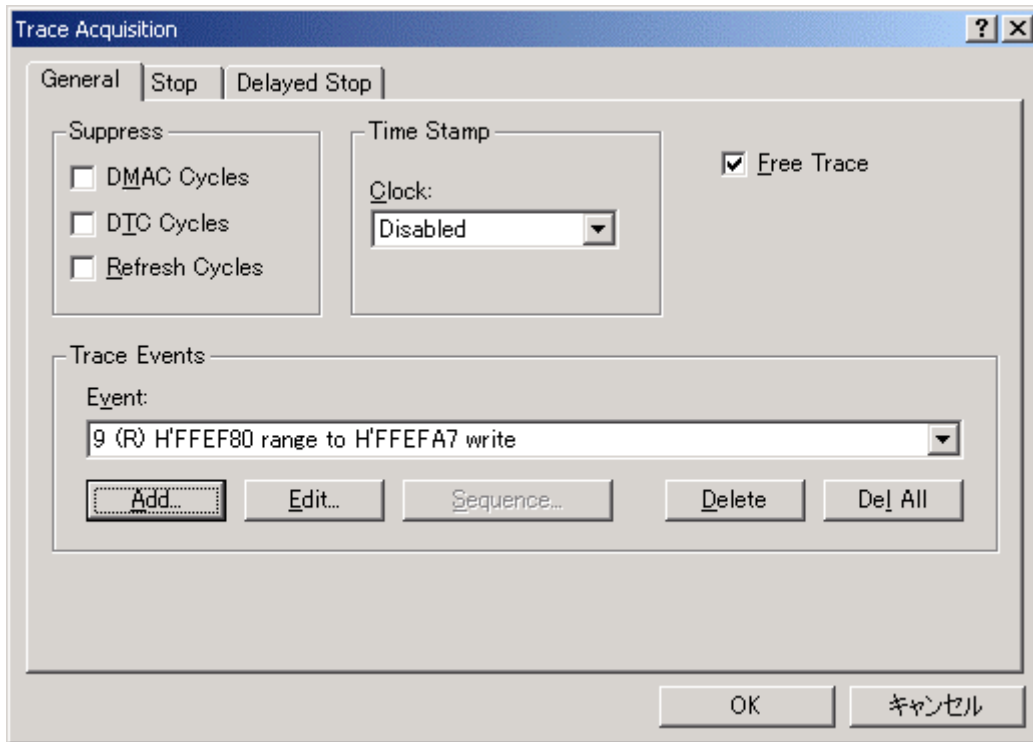


図 6-50 [Trace Acquisition]ダイアログボックス (イベント追加時)

- (6)タイムスタンプを有効とする為に、[Time Stamp]グループボックスの[Clock]ドロップダウンリストボックスから 125ns を選択します。

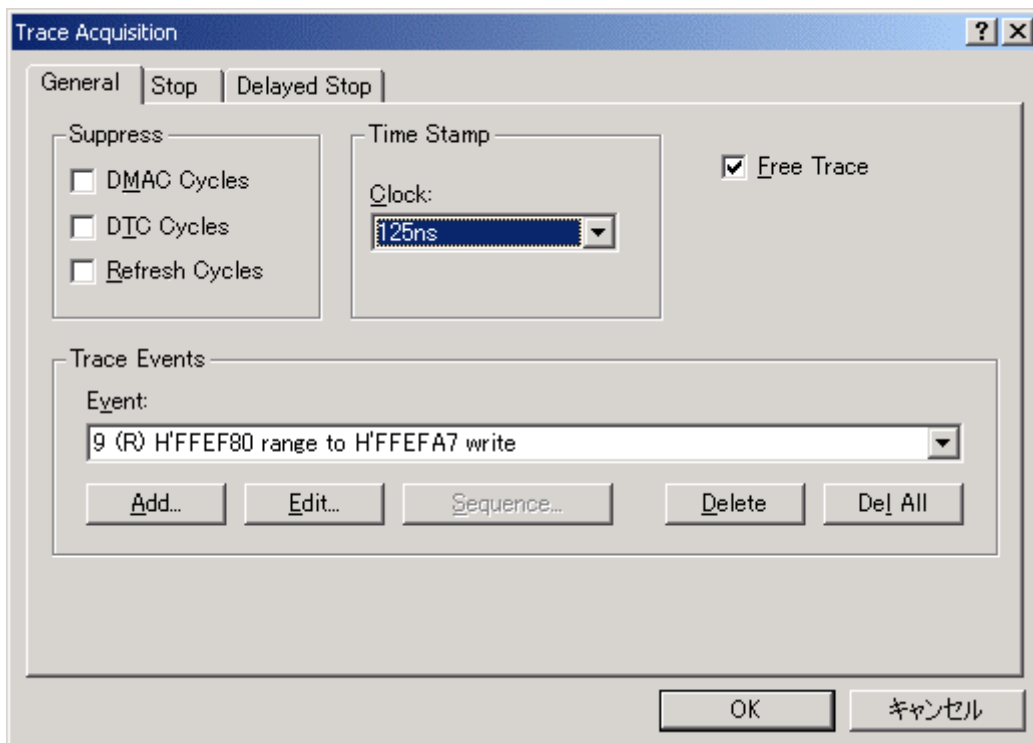


図 6-51 [Trace Acquisition]ダイアログボックス (タイムスタンプ有効時)

- (7)設定したイベントを有効とする為に、[General]ページの[Free Trace]チェックボックスのチェックを解除します。チェックを解除すると新たに[1]～[4]のページが表示されます。（図 6-45 [Trace Acquisition]ダイアログボックス）
- (8)[1]ページを選択し、[Conditions]グループボックスの[Range]ラジオボタンをクリックします。クリックと同時に[Range Event]ドロップダウンリストボックスと[Edit...]ボタンが表示されます。（図 6-46 [Trace Acquisition]ダイアログボックス）
- (9)[Range Event]ドロップダウンリストボックスをクリックし、先ほど登録したイベントを選択します。これでイベントが有効となりました。[OK]ボタンをクリックし、トレースの設定は終了です。

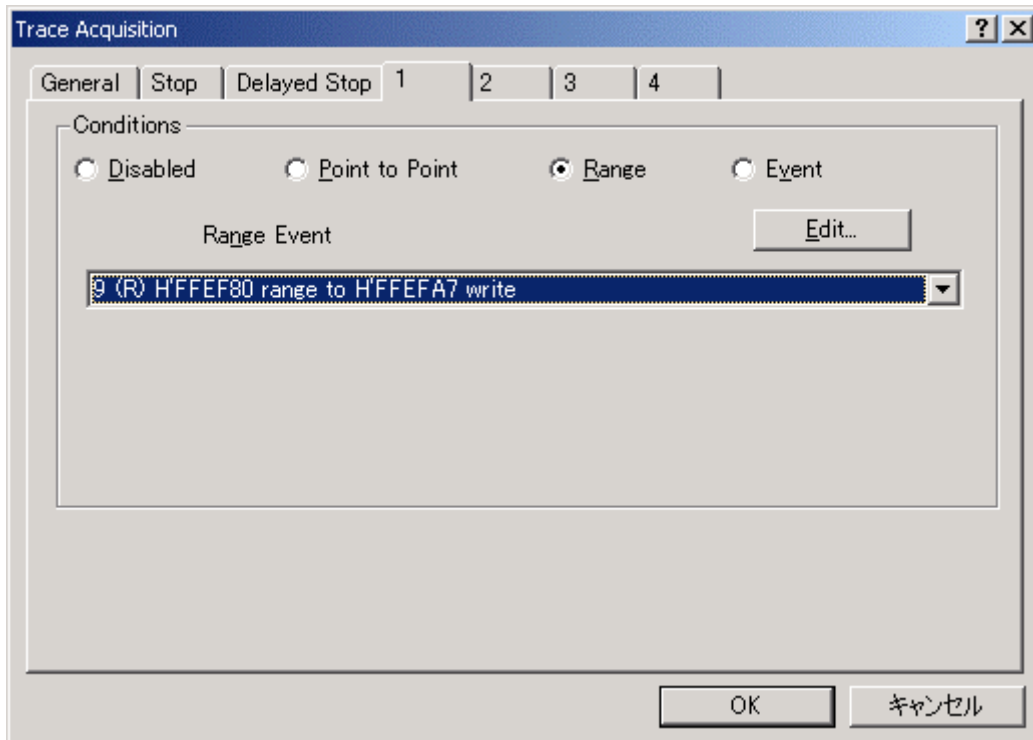
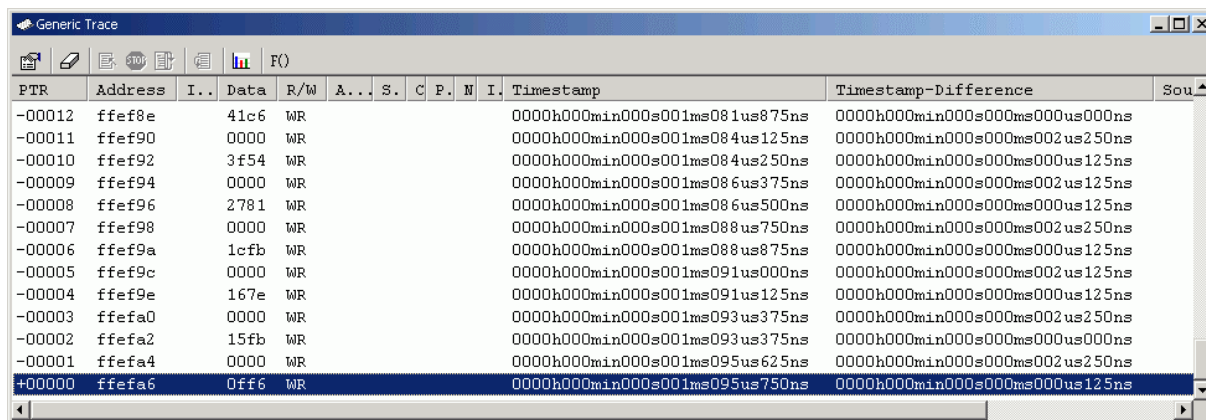


図 6-52 [Trace Acquisition]ダイアログボックス（設定完了時）

- (10)tutorial 関数内の「p_sam->s0=a[0];」と記述されている行のアドレス（本例では H'00001082）でブレークするように設定します。（「6.15.1 ソフトウェアブレーク機能」参照）

6 チュートリアル

(11)[デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[Trace]ウィンドウに以下の内容が表示されます。



PTR	Address	I.	Data	R/W	A...	S.	C	P.	N	I.	Timestamp	Timestamp-Difference	Sou
-00012	ffef8e		41c6	WR							0000h000min000s001ms081us875ns	0000h000min000s000ms000us000ns	
-00011	ffef90		0000	WR							0000h000min000s001ms084us125ns	0000h000min000s000ms002us250ns	
-00010	ffef92		3f54	WR							0000h000min000s001ms084us250ns	0000h000min000s000ms000us125ns	
-00009	ffef94		0000	WR							0000h000min000s001ms086us375ns	0000h000min000s000ms002us125ns	
-00008	ffef96		2781	WR							0000h000min000s001ms086us500ns	0000h000min000s000ms000us125ns	
-00007	ffef98		0000	WR							0000h000min000s001ms088us750ns	0000h000min000s000ms002us250ns	
-00006	ffef9a		1cfb	WR							0000h000min000s001ms088us875ns	0000h000min000s000ms000us125ns	
-00005	ffef9c		0000	WR							0000h000min000s001ms091us000ns	0000h000min000s000ms002us125ns	
-00004	ffef9e		167e	WR							0000h000min000s001ms091us125ns	0000h000min000s000ms000us125ns	
-00003	ffefa0		0000	WR							0000h000min000s001ms093us375ns	0000h000min000s000ms002us250ns	
-00002	ffefa2		15fb	WR							0000h000min000s001ms093us375ns	0000h000min000s000ms000us000ns	
-00001	ffefa4		0000	WR							0000h000min000s001ms095us625ns	0000h000min000s000ms002us250ns	
+00000	ffefa6		0ff6	WR							0000h000min000s001ms095us750ns	0000h000min000s000ms000us125ns	

図 6-53 [Trace]ウィンドウ（結果表示）

必要ならば、タイトルバーの下のヘッダバーをドラッグして、カラムの幅を調節してください。

(12)設定したイベントポイントを解除し、トレース情報をクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。また、タイムスタンプを無効とする場合には、[Trace Acquisition]ダイアログボックス-[General]ページの[Time Stamp]グループボックスの[Clock]ドロップダウンリストボックスからDisabledを選択します。

6.16.3 統計

取得したトレース情報から、内蔵 RAM に対する書き込み回数を収集します。

- (1) tutorial 関数内の「p_sam->s0=a[0];」と記述されている行のアドレス（本例では H'00001082）でブレークするように設定します。（「6.15.1 ソフトウェアブレーク機能」参照）
- (2) [デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[Trace]ウィンドウにトレース情報が表示されます。
- (3) マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[統計...]を選択してください。トレースデータのロードを示すダイアログが表示された後に、[統計]ダイアログボックスが表示されます。

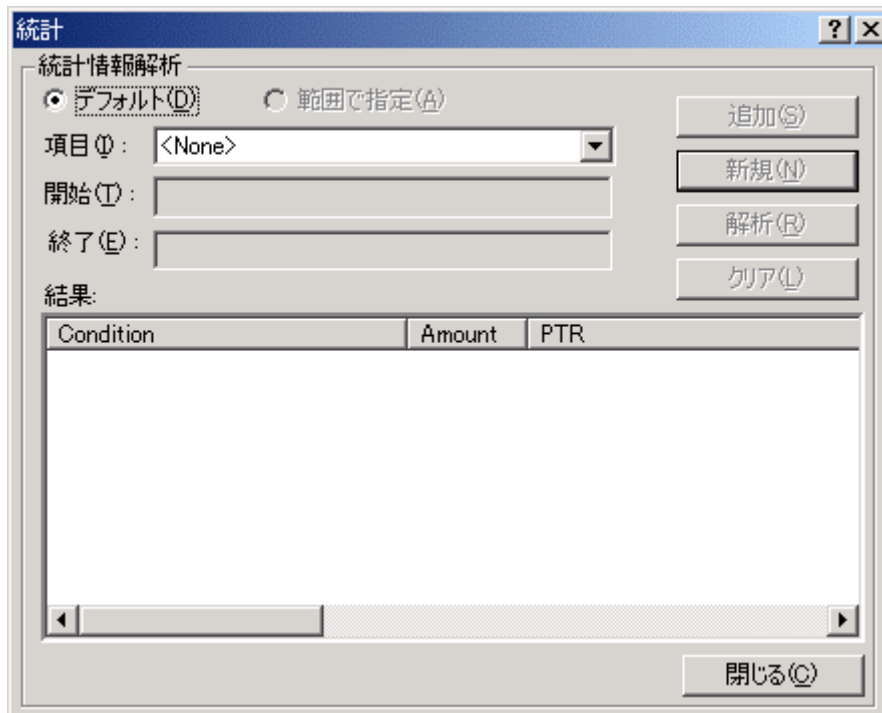


図 6-54 [統計]ダイアログボックス

6 チュートリアル

- (4)[項目]ドロップダウンリストボックスから R/W を選択し、[開始]エディットボックスに WR を入力します。入力終了後、[新規]ボタンをクリックすると[結果]リストボックスの[Condition]列に”R/W=WR”と表示されます。

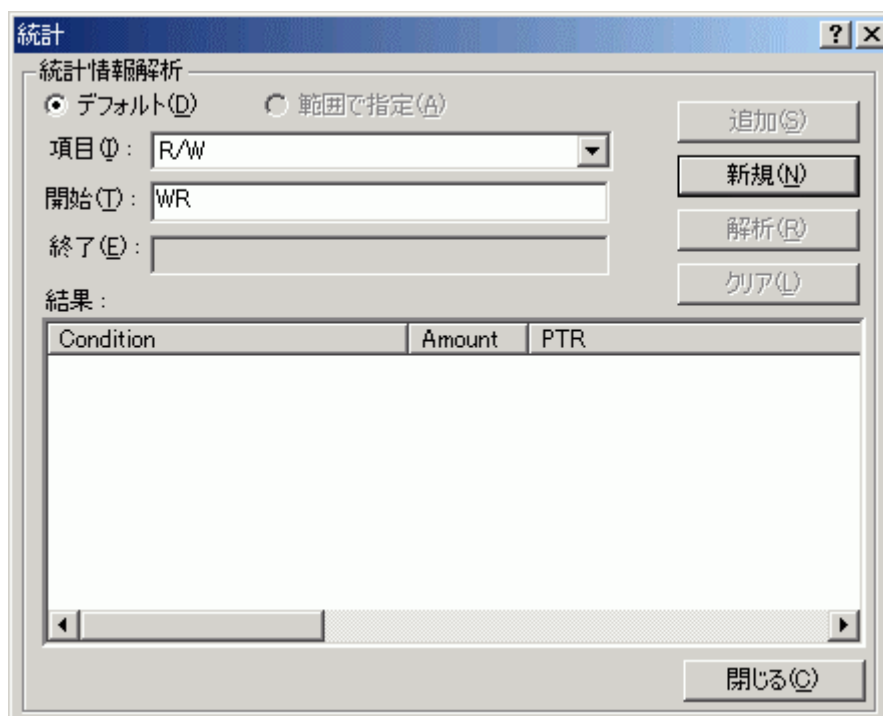


図 6-55 [統計]ダイアログボックス (新規条件)

- (5)続いて[項目]ドロップダウンリストボックスから Area を選択し、[開始]エディットボックスに RAM を入力します。入力終了後、[追加]ボタンをクリックすると[結果]リストボックスの[Condition]列に表示されていた”R/W=WR”に条件が追加され、”R/W=WR & Area=RAM”と表示されます。これで条件の入力は終了です。

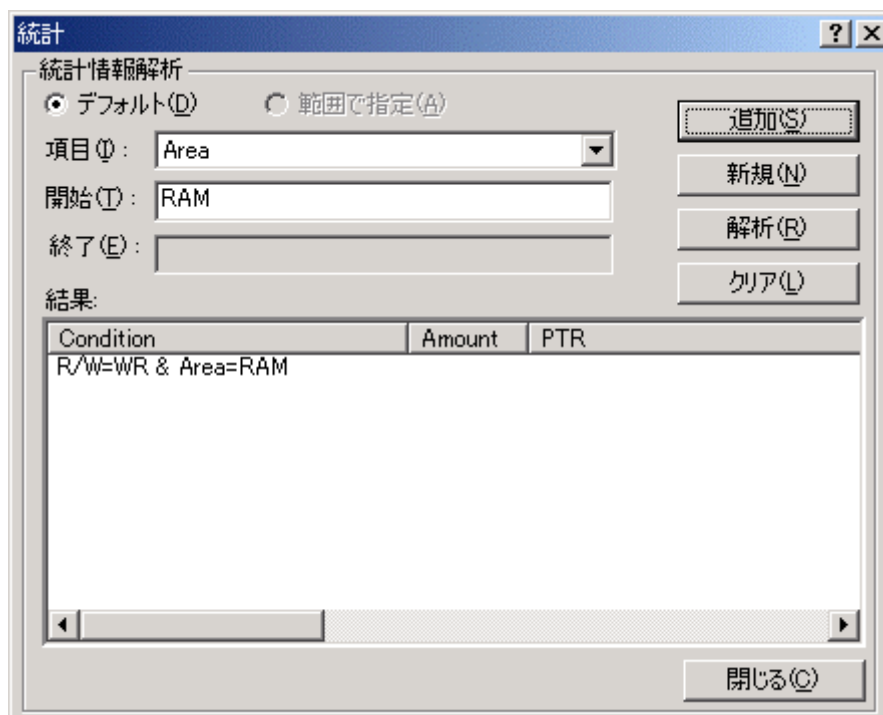


図 6-56 [統計]ダイアログボックス (条件追加)

(6)入力した条件による解析を行います。[解析]ボタンをクリックしてください。条件に該当する件数とその PTR が表示されます。

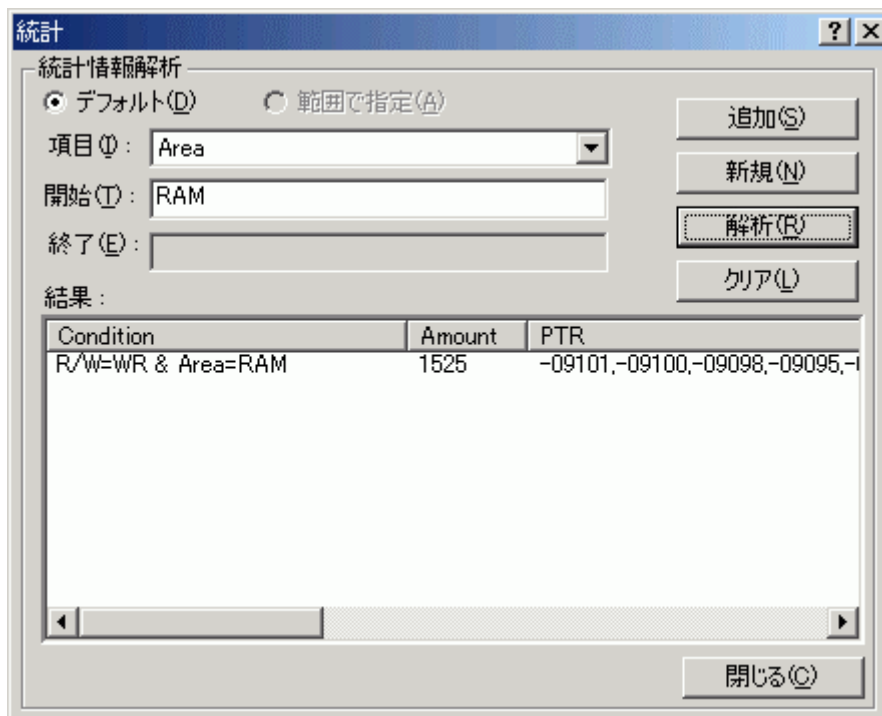


図 6-57 [統計]ダイアログボックス (解析結果)

(7)[閉じる]ボタンをクリックし、[統計]ダイアログボックスを閉じます。

(8)設定したイベントポイントを解除し、トレース情報をクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。

6.16.4 関数コール

取得したトレース情報から、関数をコールしている情報のみを収集します。

(1) tutorial 関数内の「p_sam->s0=a[0];」と記述されている行のアドレス (本例では H'00001082) でブレークするように設定します。(「6.15.1 ソフトウェアブレーク機能」参照)

(2)[デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[Trace]ウィンドウにトレース情報が表示されます。

(3)マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[関数コール...]を選択してください。[関数コール箇所の表示]ダイアログボックスが表示されます。

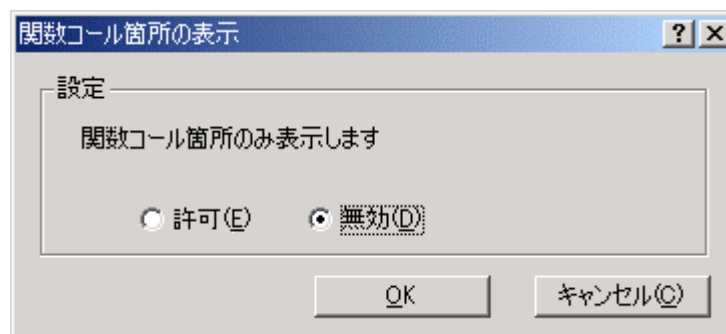
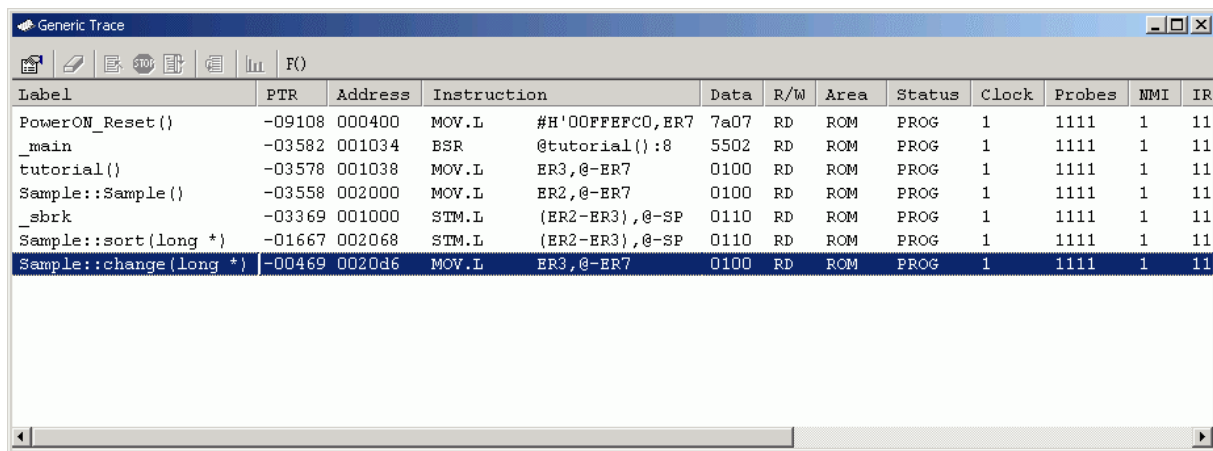


図 6-58 [関数コール箇所の表示]ダイアログボックス

6 チュートリアル

- (4)[許可]ラジオボタンをクリックし、[OK]ボタンをクリックしてください。[Trace]ウィンドウの表示が関数をコールしている情報のみに変更されます。(関数コールを確認する為、[Label]列を[Trace]ウィンドウの左端に移動しています)



Label	PTR	Address	Instruction	Data	R/W	Area	Status	Clock	Probes	NMI	IR
PowerON_Reset()	-09108	000400	MOV.L #H'00FFEFCD,ER7	7a07	RD	ROM	PROG	1	1111	1	11
_main	-03582	001034	BSR @tutorial():8	5502	RD	ROM	PROG	1	1111	1	11
tutorial()	-03578	001038	MOV.L ER3,@-ER7	0100	RD	ROM	PROG	1	1111	1	11
Sample::Sample()	-03558	002000	MOV.L ER2,@-ER7	0100	RD	ROM	PROG	1	1111	1	11
_sbrk	-03369	001000	STM.L (ER2-ER3),@-SP	0110	RD	ROM	PROG	1	1111	1	11
Sample::sort(long *)	-01667	002068	STM.L (ER2-ER3),@-SP	0110	RD	ROM	PROG	1	1111	1	11
Sample::change(long *)	-00469	0020d6	MOV.L ER3,@-ER7	0100	RD	ROM	PROG	1	1111	1	11

図 6-59 [Trace]ウィンドウ(関数コール)

- (5)[Trace]ウィンドウの表示を元の状態に戻します。(3)の手順にて[関数コール箇所の表示]ダイアログボックスを表示し、[無効]ラジオボタンをクリック、その後[OK]ボタンをクリックします。
- (6)設定したイベントポイントを解除し、トレース情報をクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。マウスの右ボタンで[Trace]ウィンドウをクリックすることによって開くポップアップメニューから[クリア]を選択し、トレース情報をクリアします。

6.17 スタックトレース機能

E6000 エミュレータでは、スタック情報を用いて、関数呼び出し履歴を表示します。

- 【注】1. 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。Dwarf2 形式のデバッグ情報を持ったロードモジュールは、H8S,H8/300 C/C++コンパイラ V4.0 以降でサポートしています。
2. 本機能の詳細については、オンラインヘルプを参照してください。

sort 関数内の行の[Editor]カラムをダブルクリックして、ソフトウェアブレークポイントを設定してください。

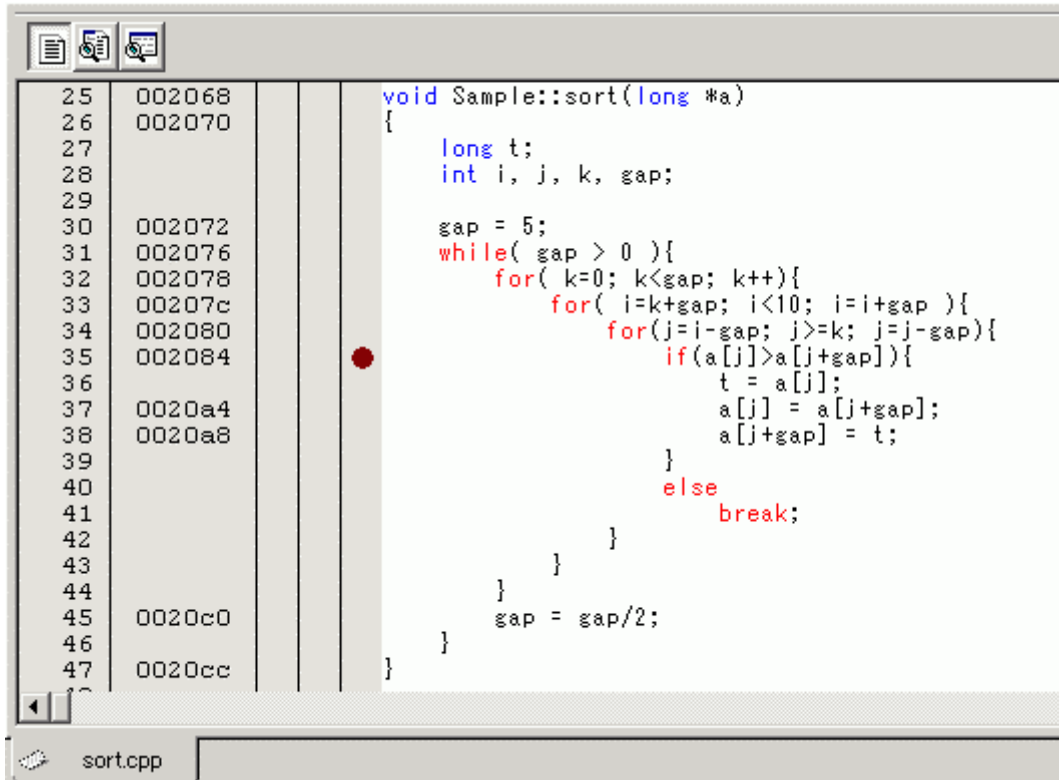
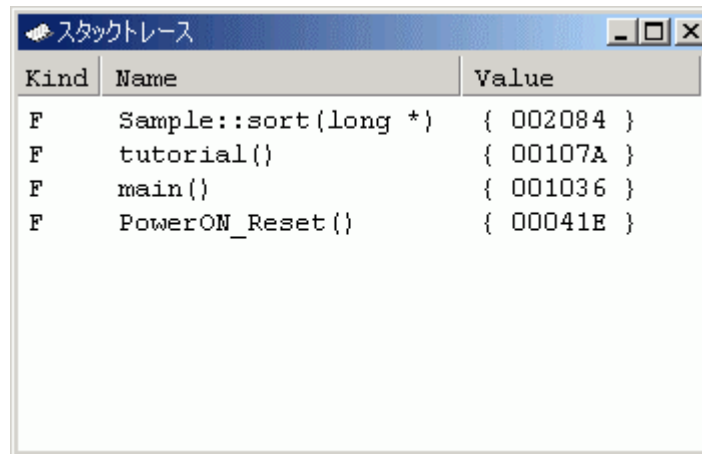


図 6-60 [エディタ]ウィンドウ (ソフトウェアブレークポイントの設定)

[デバッグ]メニューから[リセット後実行]を選択してください。

6 チュートリアル

ブレーク後、[表示]メニューから[コード]サブメニューを選択し、[スタックトレース]を選択し[スタックトレース]ウィンドウを開いてください。



Kind	Name	Value
F	Sample::sort(long *)	{ 002084 }
F	tutorial()	{ 00107A }
F	main()	{ 001036 }
F	PowerON_Reset()	{ 00041E }

図 6-61 [スタックトレース]ウィンドウ

現在 PC が sort()関数内にあり、sort()関数は、tutorial()関数からコールされていることがわかります。

sort 関数内の行の[Editor]カラムを再度ダブルクリックして、ソフトウェアブレークポイントを解除します。

6.18 パフォーマンス測定機能

E6000 エミュレータには、チップのパフォーマンスを測定する機能として、以下に示すモードがあります。

- ・指定範囲内時間測定
- ・指定アドレス間時間測定
- ・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

本チュートリアルでは「指定範囲内時間測定」の設定方法について説明します。

6.18.1 指定範囲内時間測定

(1)[表示]メニューから[パフォーマンス]サブメニューを選択し、[パフォーマンス解析]を選択します。[パフォーマンス解析方式の選択]ダイアログボックスが表示されます。

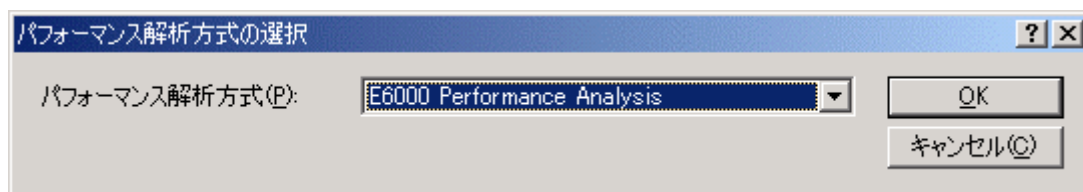


図 6-62 [パフォーマンス解析方式の選択]ダイアログボックス

(2)[パフォーマンス解析方式の選択]ダイアログボックスの[パフォーマンス解析方式]ドロップダウンリストボックスから"E6000 Performance Analysis"を選択し、[OK]ボタンをクリックします。[パフォーマンス解析]ウィンドウが表示されます。

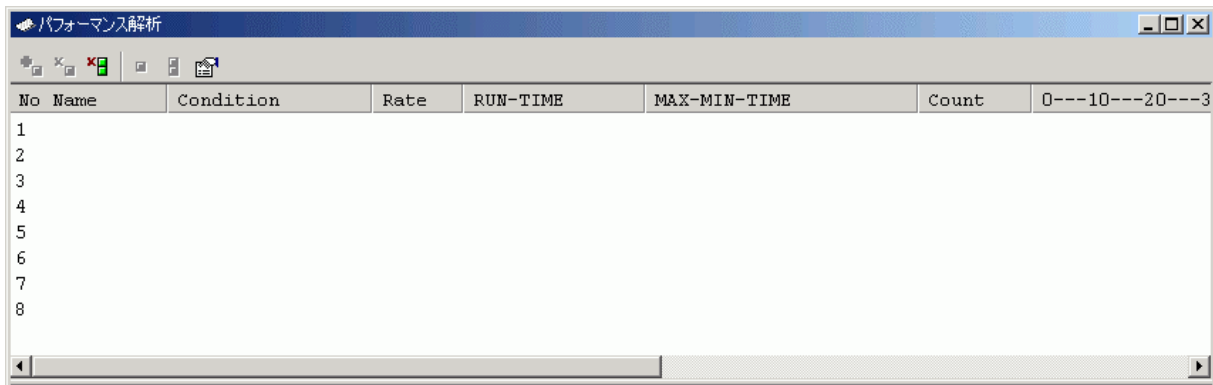


図 6-63 [パフォーマンス解析]ウィンドウ

6 チュートリアル

- (3)[パフォーマンス解析]ウィンドウの[No]列が“1”の行を選択し、マウスの右ボタンをクリックすることによって開くポップアップメニューから[設定...]を選択してください。[Performance Analysis Properties]ダイアログボックスが表示されます。

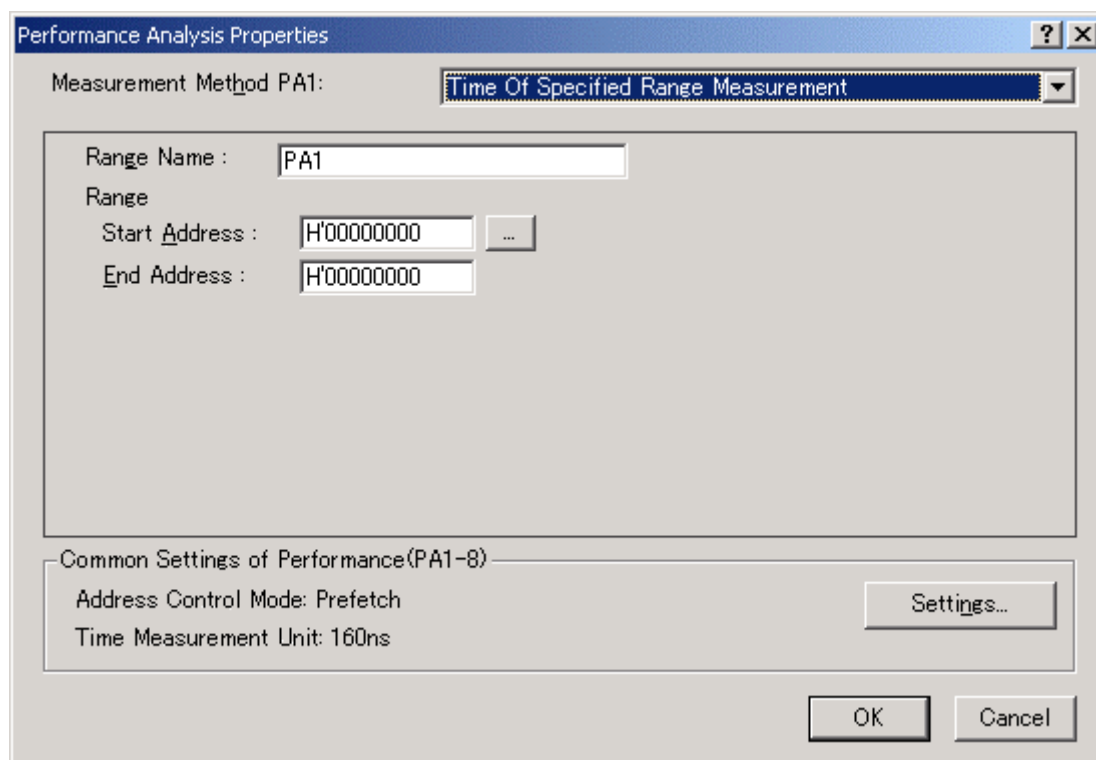


図 6-64 [Performance Analysis Properties]ダイアログボックス

- (4)[Measurement Method PA1]ドロップダウンリストボックスから、モードとして Time Of Specified Range Measurement を選択します。
- (5)パラメータの設定は以下とします。
- [Range Name]エディットボックスに sort と入力します。
 - [Start Address]エディットボックスの右側の[...]ボタンをクリックし、[Input Function Range]ダイアログボックスを表示します。[Input Function Range]ダイアログボックスの[Function]エディットボックスに関数名” Sample::sort(long *)”を入力し、[OK]ボタンをクリックします。[Start Address]エディットボックスと[End Address]エディットボックスに指定した関数のアドレスが設定されます。



図 6-65 [Input Function Range]ダイアログボックス

- 【注】 [Input Function Range]ダイアログボックスにより算出されるアドレスは参考値です。場合により関数の終了アドレスが異なる場合があります。
[逆アセンブリ]ウィンドウにより関数の最終命令を確認し、[End Address]の設定値を最終命令のアドレスに補正してください。(一般的に関数の最終命令は RTS 命令となります。)
アドレス入力ではアドレス値以外にラベル名および式の指定も可能です。

(6)[Common Setting of Performance(PA1-8)]グループボックスの[Settings...]ボタンをクリックし、[Common Setting of Performance(PA1-8)]ダイアログボックスを表示します。[Address Control Mode]ドロップダウンリストボックスから PC を選択し、[OK]ボタンをクリックします。[Common Setting of Performance(PA1-8)]グループボックスの[Address Control Mode]テキストフィールドの表示が PC となります。

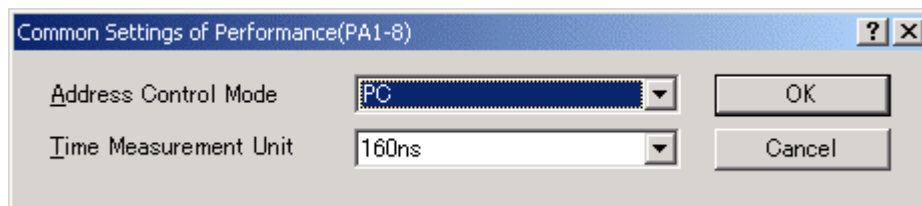


図 6-66 [Common Setting of Performance(PA1-8)]ダイアログボックス

(7)[OK]ボタンをクリックします。[パフォーマンス解析]ウィンドウの[No]列 1 に設定した内容が表示されます。これで指定範囲内時間測定の設定は終了です。

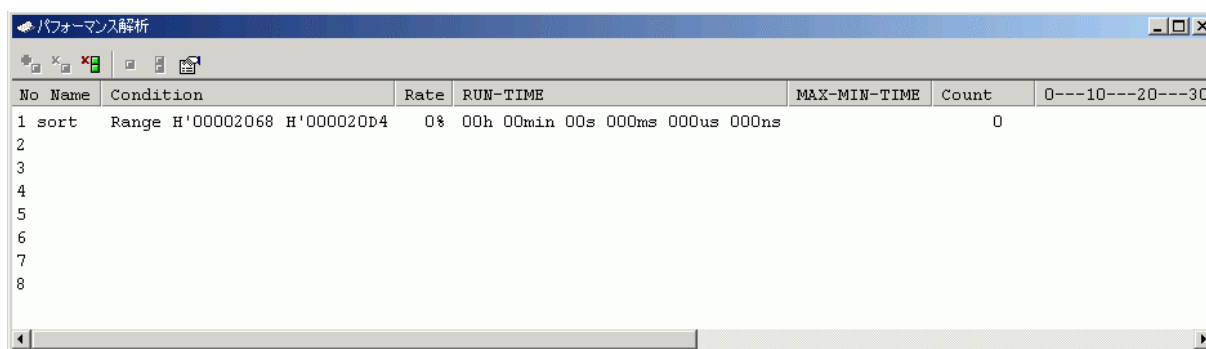


図 6-67 [パフォーマンス解析]ウィンドウ（設定完了時）

(8)指定した sort 関数を 3 回実行した後にブレークするように tutorial 関数内の「p_sam->change(a);」と記述されている行のアドレス(本例では H'0000107a)にイベントポイントによるブレークを設定します。(「6.15.2 イベントポイントによるブレーク機能」参照)

(9)[デバッグ]メニューから[リセット後実行]を選択してください。ブレークにより処理が停止し、[パフォーマンス解析]ウィンドウに以下の内容が表示されます。[Count]列の値が 3 となっており、sort 関数が 3 回実行された事と、実行時間が確認できます。

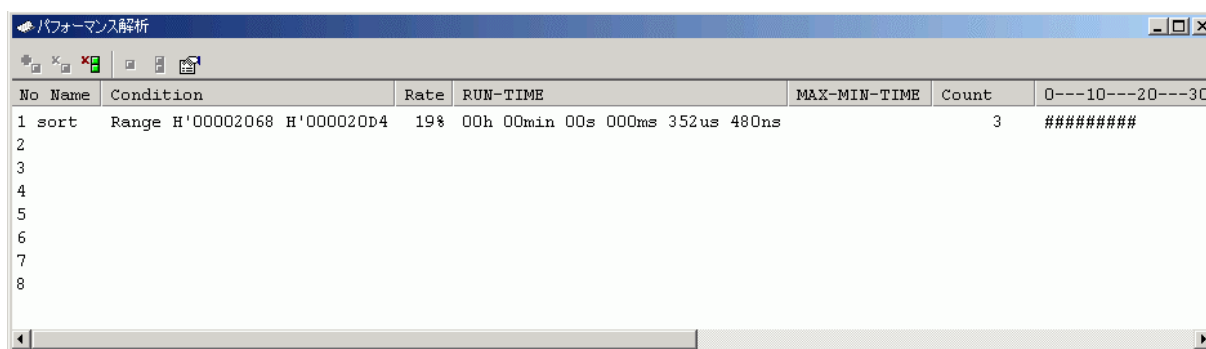


図 6-68 [パフォーマンス解析]ウィンドウ（結果表示）

(10)パフォーマンスの設定をクリアし、イベントポイントを解除します。マウスの右ボタンで[パフォーマンス解析]ウィンドウをクリックすることによって開くポップアップメニューから[全てリセット]を選択し、設定されている設定をすべてクリアします。マウスの右ボタンで[イベントポイント]ウィンドウをクリックすることによって開くポップアップメニューから[すべてを削除]を選択し、設定されているイベントポイントをすべて解除します。

6.19 モニタ機能

E6000 エミュレータには、指定したアドレスのメモリ内容を、ユーザプログラム実行中にモニタすることができます。

ここでは tutorial 関数の変数 a が割り当てられたアドレス近辺の内容をモニタします。

- (1)[表示]メニューから[CPU]サブメニューを選択、さらに[モニタ]サブメニューを選択し、[モニタ設定...]を選択します。[Monitor Setting]ダイアログボックスが表示されます。

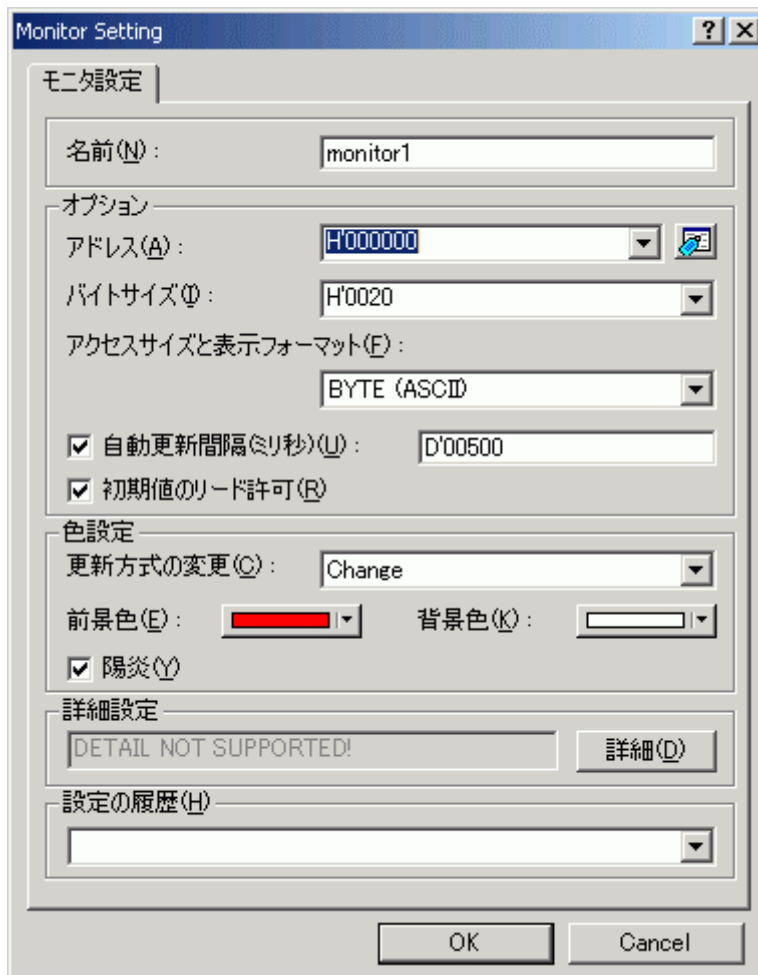


図 6-69 [Monitor Setting]ダイアログボックス

- (2)[モニタ設定]ページの内容を以下のように設定します。

- ・ [名前]エディットボックスに monitor1 と入力します。
- ・ [オプション]グループボックスの各パラメータを以下の設定とします。
 - [アドレス]エディットボックスに tutorial 関数で定義している変数 a が割りついているアドレスを [ローカル]ウィンドウで参照し入力してください。本例では H'00FFEF80 を入力します。
 - [バイトサイズ]ドロップダウンリストボックスに H'50 と入力します。
 - [アクセスサイズと表示フォーマット]ドロップダウンリストボックスから BYTE (HEX) を選択します。
 - [自動更新間隔(ミリ秒)]のチェックボックスをチェックし、エディットボックスに D'00500 と入力します。
 - [初期値のリード許可]チェックボックスをチェックします。

- ・ [色指定]グループボックスの各パラメータを以下の設定とします。
[更新方式の変更]ドロップダウンリストボックスから Change を選択します。
[前景色]ドロップダウンリストボックスを赤に、[背景色]ドロップダウンリストボックスを白に選択します。
[陽炎]チェックボックスをチェックします。

【注】前景色および背景色の設定はご使用のオペレーティングシステムにより使用できない場合があります。

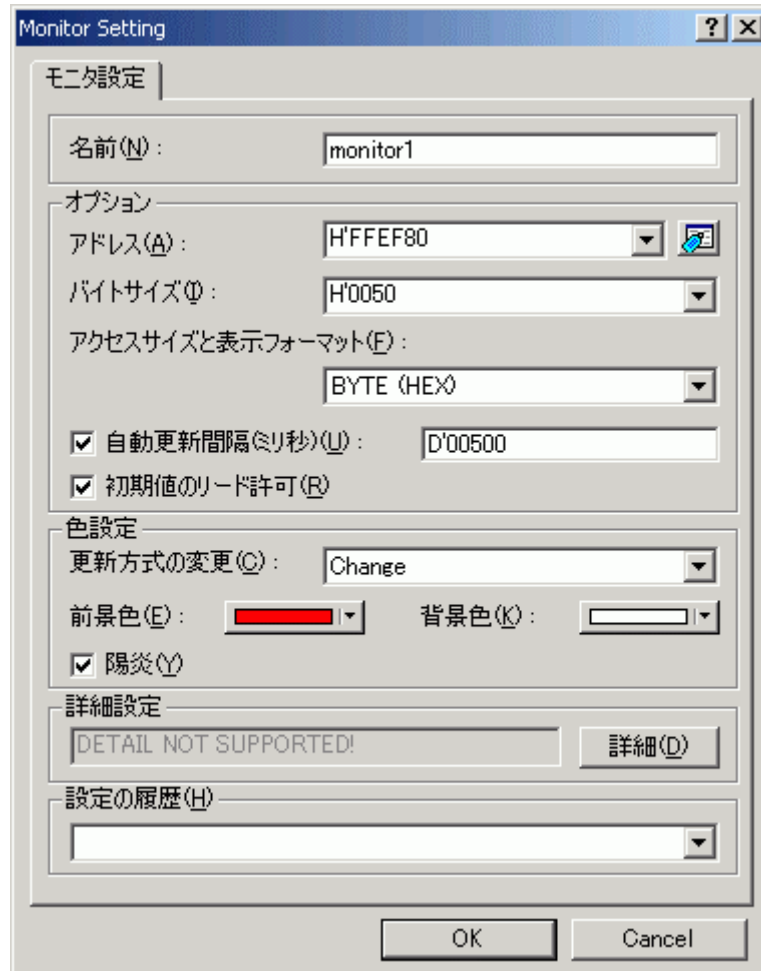
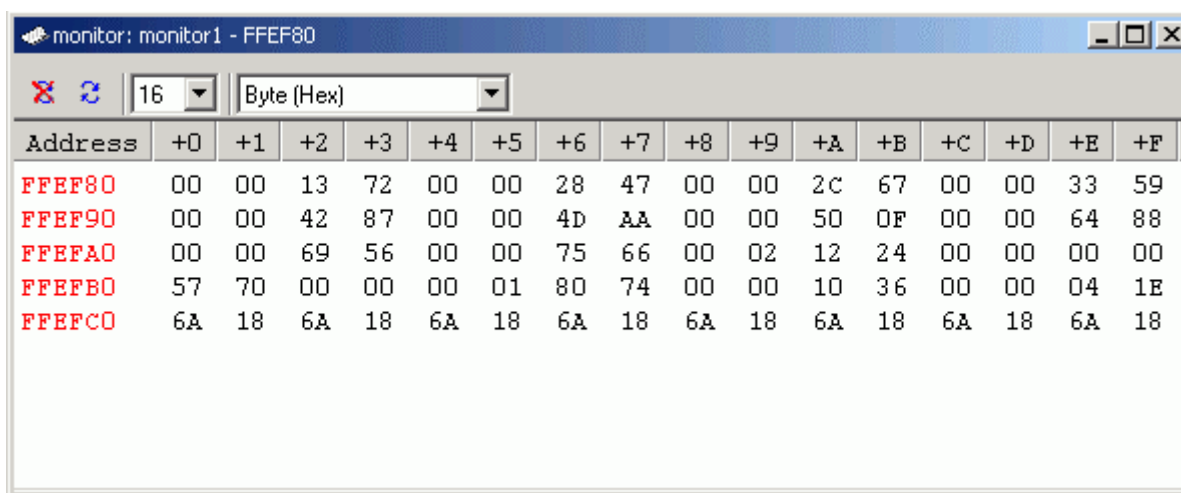


図 6-70 [Monitor Setting]ダイアログボックス (設定完了時)

6 チュートリアル

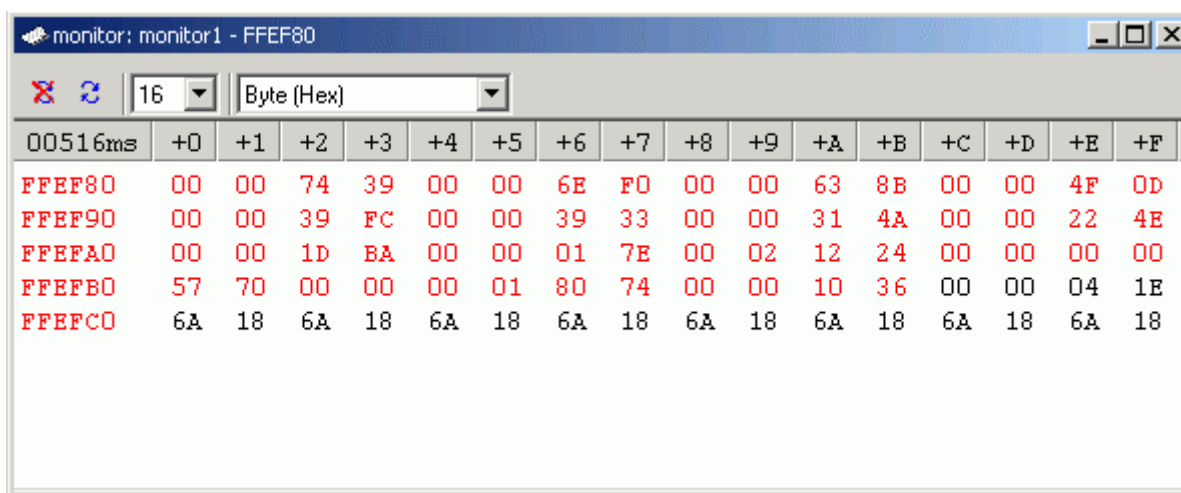
(3)[OK]ボタンをクリックします。[モニタ]ウィンドウが開きます。



Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
FFEF80	00	00	13	72	00	00	28	47	00	00	2C	67	00	00	33	59
FFEF90	00	00	42	87	00	00	4D	AA	00	00	50	0F	00	00	64	88
FFEFA0	00	00	69	56	00	00	75	66	00	02	12	24	00	00	00	00
FFEFB0	57	70	00	00	00	01	80	74	00	00	10	36	00	00	04	1E
FFEFC0	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18

図 6-71 [モニタ]ウィンドウ

(4)[デバッグ]メニューから[リセット後実行]を選択してください。処理が実行されることによりメモリの内容が更新された場合は、更新されたメモリの内容が赤（[モニタ設定]ページの[前景色]ドロップダウンリストボックスと[背景色]ドロップダウンリストボックスで設定した色）に変更されます。また、更新が行われないメモリの内容および、更新後一定間隔更新が行われなかったメモリの内容は黒で表示されます。



00516ms	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
FFEF80	00	00	74	39	00	00	6E	F0	00	00	63	8B	00	00	4F	0D
FFEF90	00	00	39	FC	00	00	39	33	00	00	31	4A	00	00	22	4E
FFEFA0	00	00	1D	BA	00	00	01	7E	00	02	12	24	00	00	00	00
FFEFB0	57	70	00	00	00	01	80	74	00	00	10	36	00	00	04	1E
FFEFC0	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18	6A	18

図 6-72 [モニタ]ウィンドウ（実行中）

(5)[モニタ]ウィンドウの表示状況が確認できましたら、[デバッグ]メニューから[プログラムの停止]を選択し、実行を停止します。

6.20 さてつぎは？

このチュートリアルでは、E6000 エミュレータのいくつかの主な特徴と、High-performance Embedded Workshop の使い方を紹介しました。

E6000 エミュレータで提供されるエミュレーション機能を使用することによって、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、識別すると、それらの問題点を効果的に調査することができます。

7 本製品固有のハードウェア仕様

この章では、H8/3064B E6000 エミュレータに関するハードウェア仕様について説明します。

7.1 H8/3064B E6000 エミュレータ仕様

H8/3064B E6000 エミュレータは下記のマイクロコンピュータを使用したシステムの開発をサポートします。

サポートデバイス

- H8/3052
- H8/3048B グループ
- H8/3048 グループ
- H8/3067 グループ
- H8/3069
- H8/3068
- H8/3062 グループ
- H8/3029 グループ
- H8/3028 グループ
- H8/3024 グループ
- H8/3008
- H8/3007
- H8/3006

7.1.1 サポート範囲

本 E6000 エミュレータがサポートする MCU 型名と対応するパッケージ、E6000 ユーザシステムインタフェースケーブルおよびオプションボードの組み合わせにつきましては、開発環境カタログを参照してください。

7.1.2 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 の保証する動作電圧・動作周波数を越えた状態で E6000 エミュレータを使用した場合、E6000 エミュレータは正常に動作しないため注意してください。

また、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7-1 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 ϕ (MHz)
1	H8/3052	4.5-5.5	2-25
2	H8/3048B グループ	3.0-3.6	2-25
		4.5-5.5	2-25
3	H8/3048 グループ	2.7-5.5	1-8
		3.15-5.5	1-13
		4.5-5.5	1-18
4	H8/3067 グループ	2.7-5.5	1-10
		3.0-5.5	1-13
		4.5-5.5	1-20
5	H8/3069	4.5-5.5	2-25
6	H8/3068	4.5-5.5	2-25
7	H8/3062 グループ	2.7-5.5	1-10
		3.0-5.5	1-13
		4.5-5.5	1-20
8	H8/3029 グループ	3.0-3.6	2-25
9	H8/3028 グループ	3.0-3.6	2-25
10	H8/3024 グループ	3.0-3.6	2-25
11	H8/3008	3.0-3.6	2-25
		4.5-5.5	2-25
12	H8/3007	2.7-5.5	1-10
		3.0-5.5	1-13
		4.5-5.5	1-20
13	H8/3006	2.7-5.5	1-10
		3.0-5.5	1-13
		4.5-5.5	1-20

留意事項

動作電圧および動作周波数範囲の詳細については、各MCUのハードウェアマニュアルでご確認ください。

7.2 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 MCU に接続されています。ただし、以下の信号は、MCU に入力する前に、制御回路が挿入されています。

- ・ NMI
- ・ /RES
- ・ MD2, MD1, MD0
- ・ /STBY
- ・ XTAL
- ・ EXTAL

7.2.1 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、 V_{CC} と V_{ref} には、この保護回路がありません。

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。

ユーザシステムインタフェースケーブル先端部の V_{CC} 端子 (AV_{CC} 端子を除く) は、すべて 1 つに接続されています。E6000 エミュレータは、 V_{CC} 端子の電圧レベルを監視して [拡張モニタ] ウィンドウに電源状態を表示します。ただし、 V_{CL} 端子または V_{CC} 端子となる 1 ピンについては未接続です。

なお、ユーザシステムインタフェースケーブル未接続の場合、ユーザシステム側の V_{CC} 端子は 3.3V となります。

7.2.2 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 8ns 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。

- (1) 以下に記述のない信号

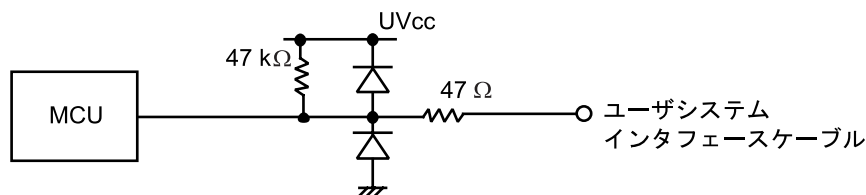


図 7-1 ユーザインタフェース信号回路

- (2) モード端子 (MD2, MD1, MD0)

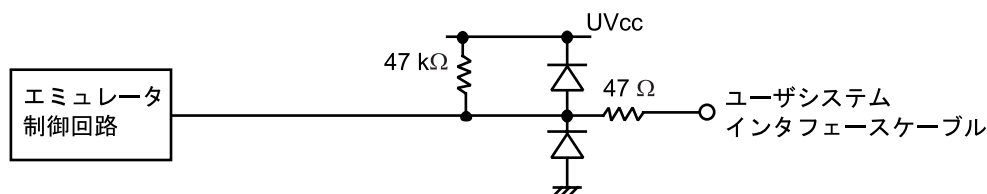


図 7-2 MD2, MD1, MD0 端子回路

- (3) /STBY

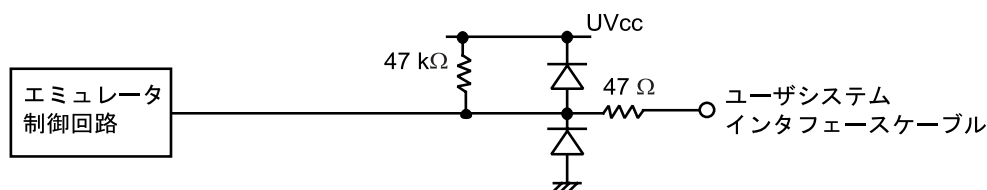


図 7-3 /STBY 信号回路

7 本製品固有のハードウェア仕様

(4) /RES、NMI

NMI 信号はエミュレータ制御回路を経由して MCU に入力されます。したがって、信号の立ち上がり / 立ち下がり時間は 8ns/V 以下にしてください。

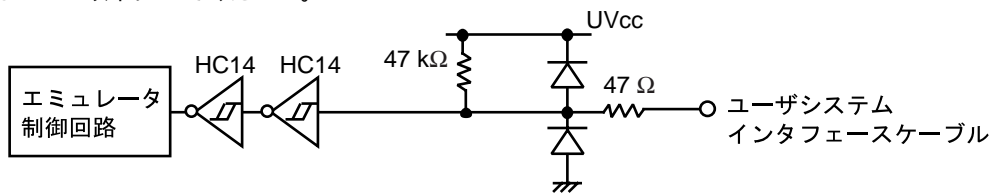


図 7-4 /RES 信号、NMI 信号回路

(5) /RESO、FWE

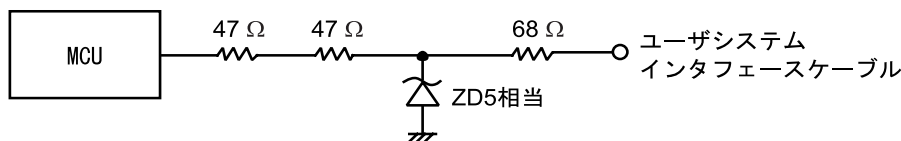


図 7-5 /RESO、FWE 信号回路

【注】 /RESO、FWEはピン配置では同じピン位置であり、MCUにより機能が異なります。本エミュレータではMCU内蔵のフラッシュメモリ機能のエミュレーションはサポートしておらず、FWE 入力はできません。/RESO出力のみ可能です。なお、インタフェースケーブル上にショートピンがあり、/RESO出力を利用する場合にのみインタフェースケーブルのP4ショートピンを1ピン側にしてください。FWEを使用するユーザーシステムに接続する場合など、/RESO出力を使用しない場合はP4ショートピンを必ず3ピン側にしてください。詳細はH8/3064B E6000ユーザーシステムインタフェースケーブル取扱い説明書(HS3064BECH61HJ)を参照ください。

(6) P70-P77 (AN0-AN7、DA0-DA1 兼用端子)、Avcc、Avss、Vref

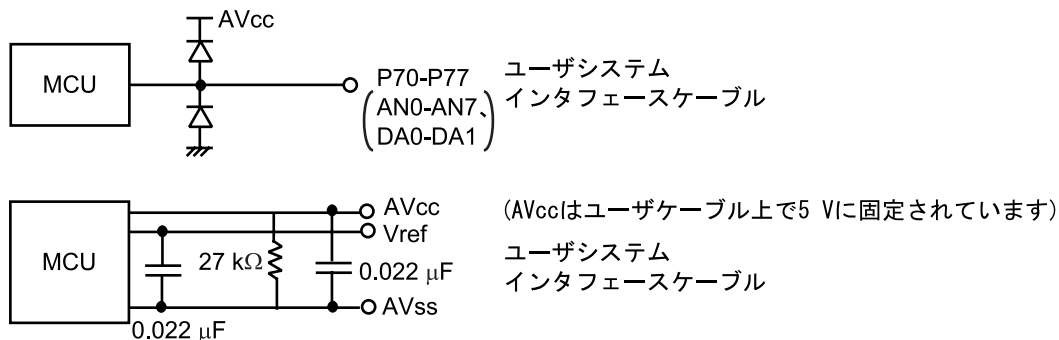


図 7-6 P70-P77、Avcc、Avss、Vref 信号回路

【注】 A/D変換器を使用しない場合、AVcc=Vcc、Vref=Vcc、AVss=Vssとしてください。

(7) /IRQ0-/IRQ5、/WAIT

/IRQ0-/IRQ5、/WAIT 信号は MCU への入力と同時にトレース取得用の回路にも入力されます。したがって、これらの信号の立ち上がり / 立ち下がり時間は 8ns/V 以下にしてください。

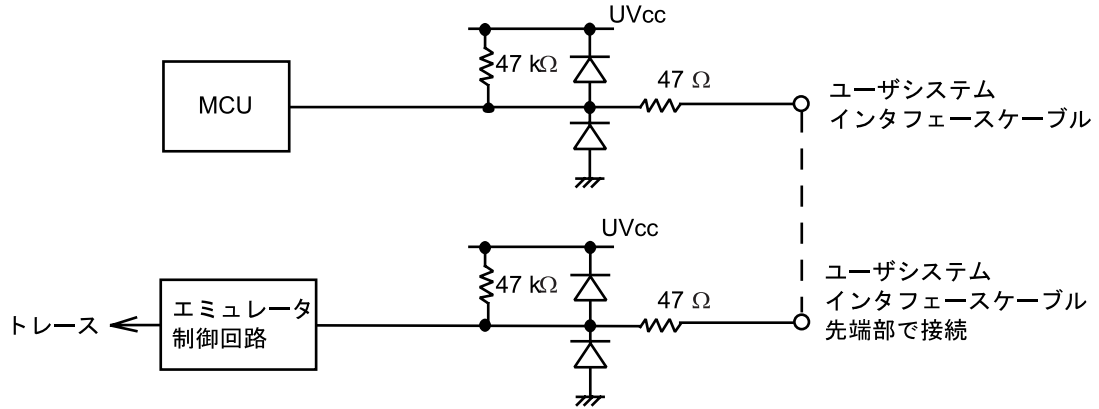


図 7-7 /IRQ0-/IRQ5、/WAIT 信号回路

7.3 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、初期化後およびリセット後の、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7-2 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/初期化後	PC	リセットベクタ値	リセットベクタ値
	ER0 to ER6	不定	不定
	ER7 (SP)	H'10	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
リセット後	PC	リセットベクタ値	リセットベクタ値
	ER0 to ER6	不定	不定
	ER7 (SP)	H'10	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

7.3.1 A/D コンバータ、D/A 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 変換と D/A 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。

8 本製品固有のソフトウェア仕様

この章では、H8/3064B E6000 エミュレータに関するソフトウェア仕様について説明します。

8.1 H8/3064B E6000 エミュレータソフトウェア仕様

本エミュレータ固有の情報を以下に示します。

8.1.1 対応ハードウェア

本エミュレータソフトウェアは H8/3064B E6000 エミュレータ(HS3064BEPI61H)に対応したソフトウェアです。

8.1.2 選択可能プラットフォーム

本エミュレータで選択可能なデバッグプラットフォームは以下の通りです。

選択したデバッグプラットフォームによりエミュレーション可能なデバイスが異なります。

表8-1 選択可能ターゲット

デバッグプラットフォーム	備考
H8/3048 E6000 Emulator CPU 300H	H8/300H CPU をコアとし、以下のデバイスのエミュレーションを行います。 H8/3048, H8/3047, H8/3045, H8/3044, H8/3052
H8/3067 E6000 Emulator CPU 300H	H8/300H CPU をコアとし、以下のデバイスのエミュレーションを行います。 H8/3067, H8/3066, H8/3065, H8/3068, H8/3069, H8/3062, H8/3061, H8/3060, H8/3064, H8/3007, H8/3006, H8/3008, H8/3024, H8/3026, H8/3028, H8/3029

8.1.3 Configuration Properties ダイアログボックス(General ページ)

本ダイアログボックスで設定可能な項目は以下の通りです。

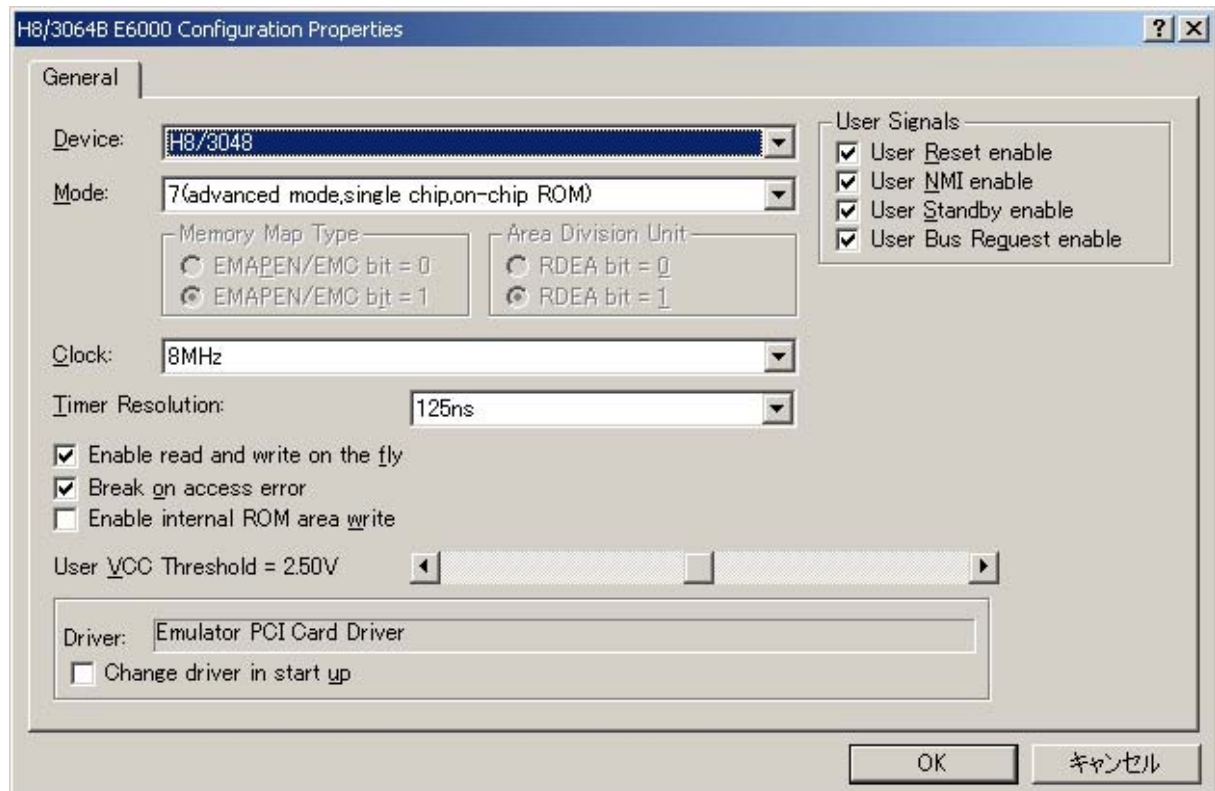


図 8-1 Configuration Properties ダイアログボックス (General ページ)

8 本製品固有のソフトウェア仕様

[General]

[Device]	エミュレーションする MCU を指定します。詳細に関しては、それぞれのハードウェアマニュアルを参照してください。
[Mode]	MCU の動作モードを指定します。
[Memory Map Type]	使用するメモリマップの種類を選択します。 この指定は E6000 の MAP 制御で使用します。レジスタには値を設定しません。 本オプションは EMAPEN bit(EMCR) または EMC bit(BCR)を持つ MCU 選択時のみ有効です。
[Area Division Unit]	使用するエリア分割単位を選択します。 この指定は E6000 の MAP 制御で使用します。レジスタには値を設定しません。 本オプションは RDEA bit(BCR)を持つ MCU 選択時でかつ、モード 3、4、5、選択時のみ有効です。
[Clock]	MCU のクロック速度を指定します。
[Timer Resolution]	実行時間の測定に使用するタイマの分解能を設定します。 分解能は以下のいずれかから選択できます。 20ns, 125ns, 250ns, 500ns, 1us, 2us, 4us, 8us, 16us 実行時間測定用タイマは 40 ビットのカウンタで構成されています。 測定可能な最大時間は分解能 20ns で約 6 時間、分解能 16us で約 200 日です。 カウンタがオーバーフローした場合、オーバーフローしたことを示すプロンプト">"と共に測定可能な最大時間を表示します。
[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。 リアルタイム性はありませんので、リアルタイムでのエミュレーションを行いたい場合には、チェックしないでください。 <ul style="list-style-type: none"> • アクセスするメモリが内蔵 ROM、エミュレーションメモリの場合 ユーザプログラムをブレイクせずに、エミュレータが強制的にバス権を確保することにより、High-performance Embedded Workshop から直接メモリにアクセスします。1 回のアクセスにつき MCU は 8 クロック停止します。 • アクセスするメモリが内蔵 RAM、内蔵 I/O、ユーザメモリの場合 ユーザプログラムをブレイクして行います。停止時間は、25MHz 動作時で約 400us です。 内蔵 RAM 無効時、プログラム実行中にこの領域に対してのアクセスは不可となります。

- 【注】** ユーザプログラム実行中、メモリ内容を変更した場合（例：[メモリ]ウィンドウ、MEMORY_EDIT コマンド等による変更）、High-performance Embedded Workshop は変更した値を表示するためメモリ内容のリードを行います。
また[メモリ]ウィンドウのポップアップメニューから[最新の情報に更新(R)]等のメモリ内容を更新する操作を行った場合も、メモリ内容のリードを行います。
このとき、メモリ内容を表示している各ウィンドウに対し、それぞれメモリ内容のリードを行い表示内容を更新します。
ユーザプログラム実行中に不要なメモリ内容のリードを抑止する場合はメモリ内容を表示しているウィンドウ（[メモリ]ウィンドウ、[逆アセンブリ]ウィンドウ等）を閉じるか、または更新を行わない設定としてください。
ただし、[モニタ]ウィンドウまたは下記条件を満たす[ウォッチ]ウィンドウはメモリ内容を表示していますが、メモリ内容更新の実現方法が異なるため、ウィンドウを開いていてもリアルタイム性を阻害しません。

[条件]

1. 登録したシンボルが汎用レジスタにのみ割りついている
2. 登録したシンボルが[モニタ]機能により設定したモニタ範囲にのみ割りついている（R マークが青色となります）
3. 登録したシンボルが上記 1 および 2 で構成されている

[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、ブレイク（ユーザプログラム停止）します。
[Enable internal ROM area write]	このチェックボックスをチェックすると、ユーザプログラムによる内蔵 ROM エリアへのライトが可能になります。ライトが行われたかどうかは、[拡張モニタ] ウィンドウによって知らされます。
[User VCC Threshold]	ユーザシステム電圧レベルを指定します。 ユーザ VCC が指定値よりも下がった場合、[拡張モニタ]ウィンドウの[User System Voltage]に[Down]が表示されます。
[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、NMI 信号、スタンバイ信号、バスリクエストを有効にします。
[Driver]	現在使用している E6000 ドライバを表示しています。
[Change driver in start up]	このチェックボックスをチェックすると、次回 E6000 接続時にドライバを選択することができます。

[Device]オプションで選択可能なデバイスおよびデバイスに依存したオプションは以下の通りです。
 拡張ハードウェア欄に記載があるデバイスのエミュレーションを行う場合は必ず該当の拡張ハードウェアを接続してください。

表8-2 H8/3048 E6000 Emulator CPU 300H デバッグプラットフォーム環境

[Device]オプション	[Mode]オプション	[Clock]オプション	拡張ハードウェア
H8/3048	1 (Internal ROM invalidity and expansion, 1Mbyte memory)	Main: 8MHz	-
H8/3052	2 (Internal ROM invalidity and expansion, 1Mbyte memory)	Main: 10MHz	
	3 (Internal ROM invalidity and expansion, 16Mbyte memory)	Main: 13MHz	
	4 (Internal ROM invalidity and expansion, 16Mbyte memory)	Main: 16MHz	
	5 (Internal ROM effective and expansion, 1Mbyte memory)	Main: 17MHz	
	6 (Internal ROM effective and expansion, 16Mbyte memory)	Main: 18MHz	
	7 (Advanced mode, single chip)	Main: 20MHz	
	Target	Main: 25MHz	
H8/3047 H8/3045 H8/3044		Main: Target	
		Main: Target/2	
		Main: 8MHz	
		Main: 10MHz	
		Main: 13MHz	
		Main: 16MHz	
	Main: 17MHz		
	Main: 18MHz		
	Main: Target		
	Main: Target/2		

- 【注】 1. [Mode]オプションの Target はユーザシステム接続時のみ選択可能です。
 2. [Clock]オプションの Target および Target/2 はユーザシステム接続時のみ選択可能です。

8 本製品固有のソフトウェア仕様

表8-3 H8/3067 E6000 Emulator CPU 300H デバッグプラットフォーム環境

[Device]オプション	[Mode]オプション	[Clock]オプション	拡張ハードウェア
H8/3068	1 (Internal ROM invalidity and expansion, 1Mbyte memory)	Main: 8MHz	-
H8/3064	2 (Internal ROM invalidity and expansion, 1Mbyte memory)	Main: 10MHz	
H8/3062	3 (Internal ROM invalidity and expansion, 16Mbyte memory)	Main: 13MHz	
H8/3028	4 (Internal ROM invalidity and expansion, 16Mbyte memory)	Main: 16MHz	
H8/3026	5 (Internal ROM effective and expansion, 1Mbyte memory)	Main: 17MHz	
H8/3024	6 (Internal ROM effective and expansion, 16Mbyte memory)	Main: 18MHz	
	7 (Advanced mode, single chip)	Main: 20MHz	
	Target	Main: 25MHz	
		Main: Target	
		Main: Target/2	
H8/3067		Main: 8MHz	
H8/3066		Main: 10MHz	
H8/3065		Main: 13MHz	
H8/3061		Main: 16MHz	
H8/3060		Main: 17MHz	
		Main: 18MHz	
		Main: 20MHz	
		Main: Target	
		Main: Target/2	
H8/3069	1 (Internal ROM invalidity and expansion, 1Mbyte memory)	Main: 8MHz	
H8/3029	2 (Internal ROM invalidity and expansion, 1Mbyte memory)	Main: 10MHz	
	3 (Internal ROM invalidity and expansion, 16Mbyte memory)	Main: 13MHz	
	4 (Internal ROM invalidity and expansion, 16Mbyte memory)	Main: 16MHz	
	5 (Internal ROM effective and expansion, 1Mbyte memory)	Main: 17MHz	
	7 (Advanced mode, single chip)	Main: 18MHz	
	Target	Main: 20MHz	
	Main: 25MHz		
	Main: Target		
		Main: Target/2	
H8/3008	1 (Internal ROM invalidity and expansion, 1Mbyte memory)		
	2 (Internal ROM invalidity and expansion, 1Mbyte memory)		
	3 (Internal ROM invalidity and expansion, 16Mbyte memory)		
	4 (Internal ROM invalidity and expansion, 16Mbyte memory)		
	Target		
H8/3007		Main: 8MHz	
	H8/3006	Main: 10MHz	
		Main: 13MHz	
		Main: 16MHz	
		Main: 17MHz	
		Main: 18MHz	
Main: 20MHz			
	Main: Target		
	Main: Target/2		

- 【注】 1. [Mode]オプションの Target はユーザシステム接続時のみ選択可能です。
 2. [Clock]オプションの Target および Target/2 はユーザシステム接続時のみ選択可能です。

8 本製品固有のソフトウェア仕様

(3) Events シート

[ステータス]ウィンドウの[Events]タブを選択すると表示されます。表示内容は以下となります。

表8-6 Eventsシート内容一覧

Item カラムの項目	Status カラムの内容
Resources	ブレイクポイントの種類とリソースを表示

8.1.7 拡張モニタ機能

[表示 -> CPU -> 拡張モニタ] を選択するか、[拡張モニタ]ツールバーボタンをクリックすると[拡張モニタ]ウィンドウを表示します。

本エミュレータでは下記項目を表示します。

表8-7 [拡張モニタ]ウィンドウ内容一覧

Item カラムの項目	Value カラムの内容												
User Standby	スタンバイ端子の状態を表示												
User NMI	NMI 端子の状態を表示												
User Reset	リセット端子の状態を表示												
User Wait	ウェイト端子の状態を表示 (該当端子が存在しない場合は常に"Inactive"と表示されます。)												
User System Voltage	ユーザ VCC が[Configuration Properties]ダイアログボックス-[General]ページの [User VCC 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												
Running status	ユーザプログラム実行中は MCU 内部のアドレスバスの値および CPU ステータスを表示 ユーザプログラム停止中はユーザプログラムの停止要因を表示 Break = <ブレイク要因> ブレイク要因を表示します。 Address = <アドレスバス値> 実行中のアドレスバスの値を表示します。 Status = <CPU のステータス> 実行中の CPU ステータスを表示します。 <table style="margin-left: 20px; border: none;"> <tr> <td>PREFETCH</td> <td>CPU 命令プリフェッチサイクル</td> </tr> <tr> <td>DATA</td> <td>CPU データアクセスサイクル</td> </tr> <tr> <td>DMAC</td> <td>DMAC 動作</td> </tr> <tr> <td>SLEEP</td> <td>スリープモード</td> </tr> <tr> <td>STANDBY</td> <td>スタンバイモード</td> </tr> <tr> <td>REFRESH</td> <td>リフレッシュサイクル</td> </tr> </table>	PREFETCH	CPU 命令プリフェッチサイクル	DATA	CPU データアクセスサイクル	DMAC	DMAC 動作	SLEEP	スリープモード	STANDBY	スタンバイモード	REFRESH	リフレッシュサイクル
PREFETCH	CPU 命令プリフェッチサイクル												
DATA	CPU データアクセスサイクル												
DMAC	DMAC 動作												
SLEEP	スリープモード												
STANDBY	スタンバイモード												
REFRESH	リフレッシュサイクル												
ROM Write	ユーザプログラム実行中、ROM にライトアクセスしたかを表示 1度ライトが行われると Configure Platform を再設定するまでライト状態が保持されます。												
Target Mode	ユーザシステムから入力されるモードを表示												
Target Clock	ユーザシステムから入力されるクロックの有無を表示												

【注】1. 本エミュレータでは[拡張モニタ]ウィンドウの更新間隔の設定および変更はできません。

8.1.8 バス状態およびエリア信号

本エミュレータで取得可能なバス状態およびエリア信号を下記に示します。

表8-8 本エミュレータで取得可能なバス状態信号

バス状態	トレース表示(Status)	詳細
CPU Prefetch	PROG	CPU プリフェッチサイクル
CPU Data	DATA	CPU データアクセスサイクル
Refresh	REFRESH	リフレッシュサイクル
DMAC	DMAC	DMAC サイクル

表8-9 本エミュレータで取得可能なエリア信号

エリア	トレース表示(Area)	詳細
On-chip ROM	ROM	ROM
On-chip RAM	RAM	RAM
On-chip I/O 16bit	I/O-16	16 ビット I/O
On-chip I/O 8bit	I/O-8	8 ビット I/O
External 16bit normal	EXT16	16 ビット EXT(外部) ノーマル
External 16bit DRAM	EXT16DRAM	16 ビット EXT(外部) DRAM
External 8bit normal	EXT8	8 ビット EXT(外部) ノーマル
External 8bit DRAM	EXT8DRAM	8 ビット EXT(外部) DRAM
On-chip Overlap RAM	RAM/OVERLAP	オーバーラップ RAM

- 【注】** バス状態およびエリア信号はイベントポイントの[Bus/Area]条件設定で使します。
 これらの信号はトレース情報としても取得可能です。
 また、バス状態信号についてはトレース非取得条件設定([Suppress]オプション)およびハードウェアパフォーマンス測定の領域アクセス回数測定モード([Access Type]オプション)でも使します。

8.1.9 モニタ機能

本エミュレータは標準でバスモニタ回路を実装しています。
 したがって、リアルタイム性を損なわずにメモリの表示内容を更新するモニタ機能を使用できます。

8.1.10 トリガポイント

本エミュレータは標準でバスモニタ回路を実装しています。
 したがって、[イベントポイント]ウィンドウの[Trigger]シートで設定可能なトリガポイントを使用できます。

8.1.11 トレース情報

[表示 -> コード -> トレース] を選択するか、[トレース]ツールバーボタンをクリックすると[Trace]ウィンドウを表示します。

本エミュレータで取得可能なトレース情報、及び表示する項目は以下の通りです。

[PTR]	トレースバッファ内のサイクル番号。 最後に取得されたサイクルの番号を 0 とし、古いサイクルにさかのぼって、順に-1、-2 と番号が小さくなります。 ディレイカウンタが設定されている場合は、トレース停止条件が成立したサイクル番号を 0 とし、成立後停止するまでに実行されたサイクル(ディレイ期間中のサイクル)には、最後に取得されたサイクルに向かって順に+1、+2 と番号が大きくなります。
[Address]	アドレスバスの値(6 桁の 16 進数)。
[Instruction]	実行された命令の逆アセンブルコード。
[Data]	データバスの値。 それぞれ 16 進の 2 桁、4 桁で表示します。
[R/W]	アクセスサイクルの種類。読み出しは RD、書き込みは WR と表示します。
[Area]	アクセスしたエリア: ROM, RAM, 8 または 16 ビット I/O, 8 または 16 ビット EXT(外部, ノーマルと DRAM), RAM/OVERLAP (タイムスタンプ取得時は無効)
[Status]	バス状態: PROG(プリフェッチ), Data(CPU データアクセスサイクル), Refresh(リフレッシュサイクル), DMAC(DMAC サイクル) (タイムスタンプ取得時は無効)
[Clock]	バスサイクルのクロック数 1~8 を表示します。それ以上の場合は OVR を表示します。 (タイムスタンプ取得時は無効)
[Probes]	4 本のプローブ信号の状態(2 進数 4 桁)。 左から Probe4、Probe3、Probe2、Probe1 の順で表示します。 (タイムスタンプ取得時は無効)
[NMI]	NMI 入力の状態。 (タイムスタンプ取得時は無効)
[IRQ5-0]	6 本の IRQ 入力の状態。 (タイムスタンプ取得時は無効)
[Timestamp]	バスサイクルのタイムスタンプ。 タイムスタンプは、ユーザプログラム実行を開始するたびに 0 からカウントを始めます。測定時間の分解能は、Trace Acquisition で設定してください。 (タイムスタンプ取得時のみ有効)
[Source]	ソースコードの表示。
[Label]	アドレスに対応するラベル(ラベルが設定されている場合のみ表示します)。
[Timestamp-Difference]	前の行とのタイムスタンプの差分時間を表示。 (タイムスタンプ取得時のみ有効)

8.1.12 トレースレコードの検索

本エミュレータ使用時、[Trace Find]ダイアログボックスは下記ページより構成されます。

表8-10 [Trace Find]ダイアログボックスのページ構成

ページ	設定項目
[General]	検索範囲を指定します。
[Address]	アドレス条件を指定します。
[Data]	データ条件を指定します。
[R/W]	アクセスサイクルの種類を指定します。
[Area]	アクセスしたエリアを指定します。 (タイムスタンプ取得時は無効)
[Status]	バス状態を指定します。 (タイムスタンプ取得時は無効)
[Probes]	4本のプローブ信号の状態を指定します。 (タイムスタンプ取得時は無効)
[IRQ5-0]	6本のIRQ入力の状態を指定します。 (タイムスタンプ取得時は無効)
[Timestamp]	バスサイクルのタイムスタンプを指定します。 (タイムスタンプ取得時のみ有効)

[IRQ5-0]ページは本エミュレータ特有の設定項目です。
以下に本エミュレータ特有のページについて説明します。

(1) IRQ5-0 ページ

IRQ 入力の状態を指定します。
タイムスタンプ取得時は無効です。

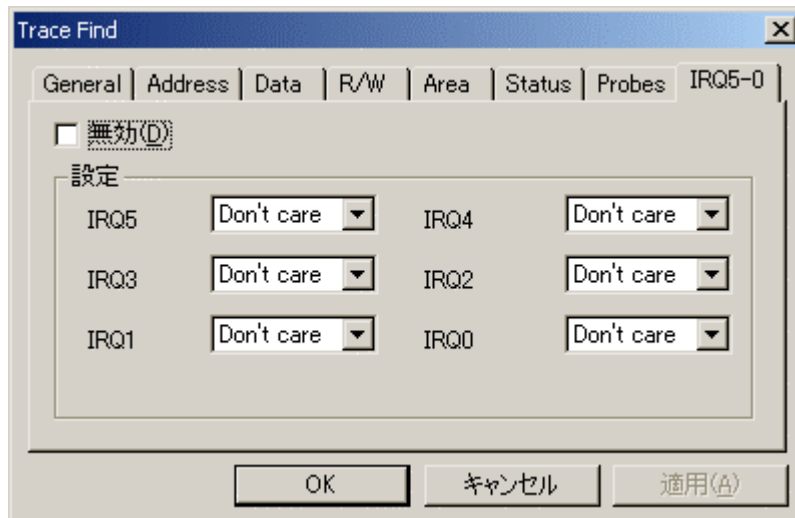


図 8-2 Trace Find ダイアログボックス(IRQ5-0 ページ)

[無効]

チェックすると、IRQ 入力の条件を検出しません。

[設定]

指定した IRQ 入力の条件を検出します。

[IRQ5] ~ [IRQ0]

IRQ 入力の条件を選択します。

([無効]選択時無効)

Don't care: 選択した IRQ 入力の条件を検出しません

High: IRQ 入力の状態が HIGH

Low: IRQ 入力の状態が LOW

8.1.13 Trace Filter 機能

本エミュレータ使用時、[Trace Filter]ダイアログボックスは下記ページより構成されます。

表8-11 [Trace Filter]ダイアログボックスのページ構成

ページ	設定項目
[General]	フィルタ範囲を指定します。
[Address]	アドレス条件を指定します。
[Data]	データ条件を指定します。
[R/W]	アクセスサイクルの種類を指定します。
[Area]	アクセスしたエリアを指定します。 (タイムスタンプ取得時は無効)
[Status]	バス状態を指定します。 (タイムスタンプ取得時は無効)
[Probes]	4本のプローブ信号の状態を指定します。 (タイムスタンプ取得時は無効)
[IRQ5-0]	6本のIRQ入力の状態を指定します。 (タイムスタンプ取得時は無効)
[Timestamp]	バスサイクルのタイムスタンプを指定します。 (タイムスタンプ取得時のみ有効)

[IRQ5-0]ページは本エミュレータ特有の設定項目です。
以下に本エミュレータ特有のページについて説明します。

(1) IRQ5-0 ページ

IRQ 入力の状態を指定します。
タイムスタンプ取得時は無効です。

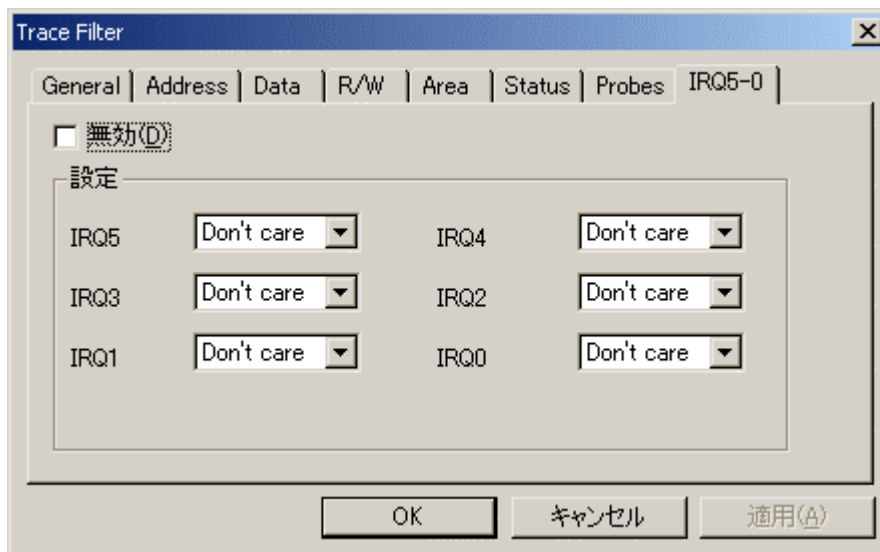


図 8-3 Trace Filter ダイアログボックス(IRQ5-0 ページ)

- [無効] チェックすると、IRQ 入力の条件を検出しません。
- [設定] 指定した IRQ 入力の条件を検出します。
- [IRQ5] ~ [IRQ0] IRQ 入力の条件を選択します。
 ([無効]選択時無効)
- Don't care: 選択した IRQ 入力の条件を検出しません
- High: IRQ 入力の状態が HIGH
- Low: IRQ 入力の状態が LOW

8.2 H8/3064B E6000 エミュレータ使用上の注意事項

本エミュレータを使用するにあたり以下の注意事項があります。

8.2.1 チュートリアルプログラムの実行環境

チュートリアルプログラムを実行する場合は、以下のディレクトリに格納されている「Tutorial.hws」を指定して下さい。

OS インストールドライブ\Workspace\Tutorial\E6000\3064B

ソフトウェアのバージョンによっては、上記ディレクトリを指定できない場合があります。その場合は以下のディレクトリを指定してください。

High-performance Embedded Workshop インストール先ディレクトリ
\\Tools\Renesas\DebugComp\Platform\E6000\3064B\Tutorial

8.2.2 I/O レジスタの相違点

E6000 エミュレータでは一つのエバリュエーションチップで複数のターゲット MCU のエミュレーションを行っているため、ターゲット MCU の I/O レジスタと E6000 エミュレータの間には以下に示すような相違点があります。I/O レジスタをアクセスする場合は注意してください。

I/O ポートは初期状態で入力になっており、エミュレータのポート端子の状態がそのまま I/O レジスタの内容に反映されています。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状態では E6000 エミュレータ上のプルアップ抵抗により、リード値は"1"になります。

本エミュレータではフラッシュメモリ制御に関する下記レジスタのアクセスは無効です。

RAM コントロールレジスタ(RAMCR)

フラッシュメモリコントロールレジスタ(FLMCR)

消去ブロック指定レジスタ 1 (EBR1)

消去ブロック指定レジスタ 2 (EBR2)

また、フラッシュメモリがあるターゲット MCU のエミュレーションの場合にはウォッチドッグタイマの/RESO 出力許可は設定をしないで下さい。

- 【注】
1. アドレスは下位 16 ビットを示しています。
 2. ウォッチドッグタイマの制御レジスタのリセット出力イネーブルビットでリセット信号の外部への出力を許可 / 禁止できますが、許可しないでください。内部はリセットされます。端子リセット時、上記リセット出力イネーブルビットは禁止状態に初期化されます。

8.2.3 内蔵 RAM 領域を外部アドレスとして使用

SYSCR の RAME ビットを"0"にした場合は内蔵 RAM 領域を外部アドレスとして使用することができます。ただし、外部アドレスとしてアクセスできるのは User(外部メモリ)のみであり、Emulator(エミュレーションメモリ)はアクセスできません。この場合、Memory Mapping は On Chip Read-write(Internal RAM)の設定となります。

8.2.4 フラッシュメモリのサポート

本エミュレータでは、フラッシュメモリ内蔵 MCU のエミュレーションはサポートしていません。

8.2.5 オプションメモリの取扱い

オプションメモリボードが実装された E6000 エミュレータでは Memory Mapping によりユーザエリアにオプションメモリを 4 バンクまで配置が可能です。

また、オプションメモリには下記の制限事項がありますのでご注意ください。

表8-12 オプションメモリの制限事項

No	項目	制限内容
1	動作周波数	20MHz 以下の場合、動作周波数による制限はありません。 20MHz を超える場合、オプションメモリは 3 ステート空間で使用してください。 本エミュレータでは、バスコントローラ内のアクセスステート制御およびウェイト制御がオプションメモリでも有効となります。 また、ウェイト制御も周波数に応じて設定してください。 リセット時、3 ステート+3 ウェイト。本エミュレータでは 25MHz 以下の場合、3 ステート+0 ウェイトで、オプションメモリのアクセスが可能です。
2	シングルチップ	シングルチップモードでは外部アクセス空間がありません。メモリマップ上はリード/ライトアクセスが禁止されているリザーブ領域となります。 本エミュレータでは、リザーブ領域にオプションメモリを配置可能ですが、下記のリザーブ領域での制限を参照のこと。
3	リザーブ領域	本エミュレータでは、リザーブ領域にオプションメモリを配置した場合、バスコントローラの設定に関わらず 3 ステート+0 ウェイトとなります。25MHz 以下の場合にアクセスできます。 実デバイスではリザーブ領域にアクセスした場合、正常動作が保証されないので注意してください。
4	メモリマップ	オプションメモリのあるエリア(0-7)に配置した場合、メモリマップ上のオプションメモリを含むエリア全体が外部アドレス空間としてアクセスできなくなります。 ユーザメモリとしてアクセスするエリアがある場合には、そのエリア内にオプションメモリは配置しないでください。

8.2.6 H8/3048F-ONE の相違点

ターゲット MCU が H8/3048F-ONE の場合、本 E6000 エミュレータとの間には以下に示すような相違点がありますので注意してください。

本 E6000 エミュレータでは下記レジスタは H8/3048F の仕様となります。詳細は、「H8/3048 シリーズ、H8/3048F-ZTAT™ ハードウェアマニュアルの 1.4.3 章 H8/3048F と H8/3048F-ONE の相違点」を参照してください。

- ・ A/D の A/D コントロールレジスタ (ADCR)
- ・ WDT のリセットコントロール/ステータスレジスタ (RSTCSR)
- ・ システムコントロールレジスタ (SYSCR) の STS2-0

付録 A I/O ファイルフォーマット

High-performance Embedded Workshopは、I/Oレジスタ定義ファイルで取得する情報に基づいて、[IO]ウィンドウをフォーマットします。デバッグプラットフォームを選択すると、High-performance Embedded Workshopは、選択したデバイスに対応する“<device>.IO”ファイルを検索し、存在する場合にはこのファイルをロードします。これは、I/Oモジュール、およびそのレジスタのアドレスやサイズを記述するフォーマット済みテキストファイルです。ユーザはテキストエディタでこのファイルを編集し、ユーザアプリケーションに特有のメモリマップレジスタや周辺レジスタ(例えば、マイコンのアドレス空間にマップしたASICデバイスのレジスタ)のサポートを追加することができます。“<device>.IO”ファイルには、ビットフィールド対応/非対応により二種類のフォーマットがあります。それぞれのフォーマットについて説明します。

A.1 ファイルフォーマット(ビットフィールド非対応)

各モジュール名を[Modules]定義セクションで定義し、モジュールの番号を、順番に付けていなければなりません。各モジュールはレジスタ定義セクションに対応しており、セクション内のエントリは、I/Oレジスタを定義します。

“BaseAddress”はデバイスのための定義であり、そのデバイスでは、CPUモードによってアドレス空間のI/Oレジスタの場所が移動します。この場合、“BaseAddress”値は、ある特有モードのI/Oレジスタのベースアドレスです。また、レジスタ定義で使用するアドレスは、同じモードにおけるレジスタのアドレス位置です。I/Oレジスタファイルを実際を使用する場合、定義したレジスタアドレスから“BaseAddress”値を引き、その結果のオフセットを選択したモードのベースアドレスに加算します。

[Register]定義エントリは、<name>=<address> [<size>[<absolute>]]のフォーマットで入力します。

1. <name>は表示するレジスタ名です。
2. <address>はレジスタのアドレスです。
3. <size>は、Bがバイトサイズ、Wがワードサイズ、Lがロングワードサイズを意味します(デフォルトはバイトです)。
4. <absolute>は、レジスタが絶対アドレスにある場合、Aと設定します。これは、異なるモードのCPUによってI/O空間アドレス範囲が移動する場合のみ関連します。レジスタが絶対アドレスにあると定義すると、ベースアドレスオフセットは計算せず、指定したアドレスを直接使用します。

コメント行を入れる場合、“;”で始めなければなりません。

次に例を示します。

例:

コメント ; H8S/2655 Series I/O Register Definitions File

モジュールの定義

```
[Modules]
BaseAddress=0
Module1=Power_Down_Mode_Registers
Module2=DMA_Channel_Common
Module3=DMA_Channel_0
...
Module42=Bus_Controller
Module43=System_Control
Module44=Interrupt_Controller
```

レジスタの定義

```
[DMA_Channel_Common]
DMAWER=0xffff00 B A
DMATCR=0xffff01 B A
DMACR0A=0xffff02 B A
DMACR0B=0xffff03 B A
DMACR1A=0xffff04 B A
DMACR1B=0xffff05 B A
DMABCRH=0xffff06 B A
DMABCRL=0xffff07 B A
```

レジスタ名

```
[DMA_Channel_0]
MAR0AH=0xfffee0 W A
MAR0AL=0xfffee2 W A
IOAR0A=0xfffee4 W A
ETCR0A=0xfffee6 W A
MAR0BH=0xfffee8 W A
MAR0BL=0xfffeea W A
IOAR0B=0xfffeec W A
ETCR0B=0xfffeee W A
```

アドレス

サイズ

絶対アドレスラグ

A.2 ファイルフォーマット(ビットフィールド対応)

各モジュール名を[Modules]定義セクションで定義し、モジュールの番号を、順番に付けていなければなりません。各モジュールはレジスタ定義セクションに対応しており、セクション内のエントリは、I/Oレジスタを定義します。

このセクションの最初で、"FileVersion=2"と宣言する必要があります。これは、このI/Oレジスタファイルがビットフィールド対応のバージョンで記述されていることを示します。

"BaseAddress"はデバイスのための定義であり、そのデバイスでは、CPUモードによってアドレス空間のI/Oレジスタの場所が移動します。この場合、"BaseAddress"値は、ある特有モードのI/Oレジスタのベースアドレスです。また、レジスタ定義で使用するアドレスは、同じモードにおけるレジスタのアドレス位置です。I/Oレジスタファイルを実際に使用する場合、定義したレジスタアドレスから"BaseAddress"値を引き、その結果のオフセットを選択したモードのベースアドレスに加算します。

各モジュールにはセクションがあり、オプションの依存性によって形成するレジスタを定義します。依存性は、モジュールがイネーブルかどうかを確認するためにチェックします。各レジスタ名をセクションで定義し、レジスタの番号を、順番に付けていなければなりません。依存性は、dep=<reg> <bit> <value>のようにセクションに入力します。

1. <reg>は依存性のレジスタ ID です。
2. <bit>はレジスタのビット位置です。
3. <value>は値で、ビットは、イネーブルであるモジュールに使用しなければなりません。

[Register]定義エントリは、id=<name> <address> [<size>[<absolute>[<format>[<bitfields>]]]]のフォーマットで入力します。

1. <name>は表示するレジスタ名です。
2. <address>はレジスタのアドレスです。
3. <size>は、B がバイトサイズ、W がワードサイズ、L がロングワードサイズを意味します(デフォルトはバイトです)。
4. <absolute>は、レジスタが絶対アドレスにある場合、A と設定します。これは、異なるモードのCPUによってI/O空間アドレス範囲が移動する場合のみ関連します。レジスタが絶対アドレスにあると定義すると、ベースアドレスオフセットは計算せず、指定したアドレスを直接使用します。
5. <format>はレジスタを出力するためのフォーマットです。有効な値は、16進数の場合はH、10進数はD、2進数はBです。
6. <bitfields>セクションは、レジスタのビットを定義します。

ビットフィールドセクションは、各エントリがbit<no>=<name>タイプのレジスタ内のビットを定義します。

1. <no>はビット番号です。
2. <name>はビットのシンボル名です。

コメント行を入れる場合、";"で始めなければなりません。

次に例を示します。

例:

コメント ; H8S/2655 Series I/O Register Definitions File

モジュール [Modules]
 FileVersion=2
 BaseAddress=0
 Module1=Power_Down_Mode_Registers
 Module2=DMA_Channel_Common
 Module3=DMA_Channel_0
 ...
 Module42=Bus_Controller
 Module43=System_Control
 Module44=Interrupt_Controller

モジュールの定義 [DMA_Channel_Common]
 reg0=regDMAWER
 reg1=regDMATCR
 reg2=regDMACR0A
 reg3=regDMACR0B
 reg4=regDMACR1A
 reg5=regDMACR1B
 reg6=regDMABCRH
 reg7=regDMABCRL
 dep= regMSTPCRH 7 0

レジスタ名
 ビット
 値

レジスタの定義 [regDMAWER]
 id=DMAWER 0xffff00 B A H dmawer_bitfields



レジスタ名
 アドレス
 サイズ
 絶対アドレスフラグ
 フォーマット
 ビットフィールド

ビットフィールドの定義 [dmawer_bitfields]
 bit3=WE1B
 bit2=WE1A
 bit1=WE0B
 bit0=WE0A

付録 B メニュー一覧

GUI メニューの一覧を表 B-1 に示します。

表 B-1 GUI メニュー一覧

メニュー	メニューオプション	ショートカットキー	ツールバーボタン	備考	
表示	差分			[差分]ウィンドウを表示します	
	コマンドライン	Ctrl+L		[コマンドライン]ウィンドウを表示します	
	TCL ツールキット	Ctrl+Shift+L		[Console]ウィンドウを表示します	
	ワークスペース	Alt+K		[ワークスペース]ウィンドウを表示します	
	アウトプット	Alt+U		[アウトプット]ウィンドウを表示します	
	逆アセンブリ	Ctrl+D		[逆アセンブリ]ウィンドウを表示します	
	CPU	レジスタ	Ctrl+R		[レジスタ]ウィンドウを表示します
		メモリ...	Ctrl+M		[メモリ]ウィンドウを表示します
		IO	Ctrl+I		[IO]ウィンドウを表示します
		ステータス	Ctrl+U		[ステータス]ウィンドウを表示します
		拡張モニタ			[拡張モニタ]ウィンドウを表示します
		モニタ 設定...	Shift+Ctrl+E		[モニタ]ウィンドウを表示します
		ウィンドウの選択...			[モニタ]ウィンドウの一覧表示、および追加 / 編集等を行う[ウィンドウの選択]ダイアログボックスを表示します
	シンボル	ラベル	Shift+Ctrl+A		[ラベル]ウィンドウを表示します
ウォッチ		Ctrl+W		[ウォッチ]ウィンドウを表示します	
ローカル		Shift+Ctrl+W		[ローカル]ウィンドウを表示します	
コード	イベントポイント	Ctrl+E		[イベントポイント]ウィンドウを表示します	
	トレース	Ctrl+T		[Trace]ウィンドウを表示します	
	スタックトレース	Ctrl+K		[スタックトレース]ウィンドウを表示します	
グラフィック	画像...	Shift+Ctrl+G		[画像]ウィンドウを表示します	
	波形...	Shift+Ctrl+V		[波形]ウィンドウを表示します	
パフォーマンス	パフォーマンス解析	Shift+Ctrl+P		[パフォーマンス解析]ウィンドウを表示します	

メニュー	メニューオプション	ショートカットキー	ツールバーボタン	備考	
デバッグ	デバッグセッション...			デバッグセッションの一覧表示、および追加/削除等が可能な[デバッグセッション]ダイアログボックスを表示します	
	デバッグの設定...			デバッグ時の条件やダウンロードモジュール等の設定を行う[デバッグの設定]ダイアログボックスを表示します	
	CPUのリセット			ターゲットハードウェアをリセットし、PC をリセットベクタアドレスに設定します	
	実行	F5		現在の PC からユーザプログラムを実行します	
	リセット後実行	Shift+F5		ターゲットハードウェアをリセットし、リセットベクタアドレスからユーザプログラムを実行します	
	カーソル位置まで実行			現在の PC からテキストカーソルの位置までユーザプログラムを実行します	
	カーソル位置を PC 値に設定			テキストカーソルの位置に PC を設定します	
	ラン...			実行時の PC や PC ブレークポイントの設定が可能な[プログラム実行]ダイアログボックスを表示します。	
	PC 位置表示	Shift+Ctrl+Y		現在の PC がある[エディタ]または[逆アセンブリ]ウィンドウをオープンします	
	ステップイン	F11		ユーザプログラムの 1 ブロックを実行して停止します	
	ステップオーバ	F10		ユーザプログラムの 1 ブロックを実行して停止しますが、サブルーチン呼び出しの場合は、サブルーチンには入りません	
	ステップアウト	Shift+F11		現在の関数の終わりに到達するまでユーザプログラムを実行します	
	ステップ...			ステップ動作の設定が可能な[プログラムステップ]ダイアログボックスを表示します	
	ステップモード	自動			[ソース]ウィンドウがアクティブの場合はソースライン一行だけをステップ実行します、[逆アセンブリ]ウィンドウがアクティブの場合はアセンブリ言語命令単位にステップ実行します
		アセンブリ			アセンブリ言語命令単位にステップ実行します
		ソース			ソースライン一行だけをステップ実行します
	プログラムの停止	Esc		ユーザプログラムの実行を停止します	
	初期化			デバッグプラットフォームを切断し、再接続します	
	接続			デバッグプラットフォームを接続します	
	接続解除			デバッグプラットフォームを切断します	
	メモリの保存			任意のアドレス領域を保存します	
メモリのペリファイ			アドレス領域を検証します		
オーバーレイの構成...			オーバーレイ使用時の優先セクショングループの設定を行います		
ダウンロード			オブジェクトプログラムをロードします		
アンロード			オブジェクトプログラムをアンロードします		

メニュー	メニューオプション	ショートカットキー	ツールバーボタン	備考	
基本設定	カスタマイズ...			High-performance Embedded Workshop アプリケーションの設定をカスタマイズします	
	オプション...			High-performance Embedded Workshop アプリケーションのオプションを設定します	
	表示の形式...			ウィンドウの表示色、フォント、キーワードなどを設定します	
	基数	16 進数			数値の表示 / 入力時の基数のデフォルト設定を 16 進数とします
		10 進数			数値の表示 / 入力時の基数のデフォルト設定を 10 進数とします
		8 進数			数値の表示 / 入力時の基数のデフォルト設定を 8 進数とします
		2 進数			数値の表示 / 入力時の基数のデフォルト設定を 2 進数とします
エミュレータ	システム...			デバッグプラットフォームの設定を行う [Configuration Properties] ダイアログボックスを表示します	
	メモリリソース...			デバッグプラットフォームのメモリマップの表示、および設定を行う [Memory Mapping] ダイアログボックスを表示します	

付録 C コマンドライン一覧

コマンド一覧を表 C-1 に示します。

表 C-1 コマンド一覧

項番	コマンド名	短縮形	説明
1	!	-	コメント
2	ADD_FILE	AF	カレントプロジェクトにファイルを追加
3	ANALYSIS	AN	性能分析機能の有効化/無効化
4	ANALYSIS_RANGE	AR	性能分析範囲の設定, 表示
5	ANALYSIS_RANGE_DELETE	AD	性能分析範囲の削除
6	ASSEMBLE	AS	アセンブルの実行
7	ASSERT	-	コンディションのチェック
8	AUTO_COMPLETE	AC	オートコンプリート機能の有効化/無効化
9	BREAKPOINT	BP	実行命令位置によるブレークポイントの設定
10	BREAKPOINT_CLEAR	BC	ブレークポイントの削除
11	BREAKPOINT_DISPLAY	BD	ブレークポイント一覧の表示
12	BREAKPOINT_ENABLE	BE	ブレークポイントの有効/無効の切換え
13	BREAKPOINT_SEQUENCE	BS	実行順序を指定したブレークポイントの設定
14	BUILD	BU	カレントプロジェクトのビルド処理を開始
15	BUILD_ALL	BL	カレントプロジェクトのすべてのビルド処理を開始
16	CHANGE_CONFIGURATION	CC	コンフィギュレーションの設定
17	CHANGE_PROJECT	CP	プロジェクトの設定
18	CHANGE_SESSION	CS	セッションの設定
19	CLOCK	CK	エミュレータの CPU のクロック時間の設定
20	CONFIGURE_PLATFORM	CPF	エミュレータのデバッグ環境の設定
21	CLOSE_WORKSPACE	CW	ワークスペースを閉じる
22	DEFAULT_OBJECT_FORMAT	DO	デフォルトオブジェクト(プログラム)フォーマットの設定
23	DEVICE_TYPE	DE	デバイスタイプの選択
24	DISASSEMBLE	DA	逆アセンブル表示
25	ERASE	ER	[コマンドライン]ウィンドウの内容のクリア
26	EVALUATE	EV	式の計算
27	EXMONITOR_DISPLAY	EXMD	拡張モニタの内容表示
28	EXMONITOR_SET	EXMS	拡張モニタ項目の表示/非表示の切換え
29	EXMONITOR_SETRATE	EXMSR	実行中およびブレーク中の拡張モニタ更新時間の設定
30	FILE_LOAD	FL	オブジェクト(プログラム)ファイルのロード
31	FILE_SAVE	FS	メモリ内容のファイルセーブ
32	FILE_UNLOAD	FU	ファイルのアンロード
33	FILE_VERIFY	FV	ファイル内容とメモリ内容の比較
34	GENERATE_MAKE_FILE	GM	High-performance Embedded Workshop 外でビルドするための make ファイルを生成
35	GO	GO	ユーザプログラムの実行
36	GO_RESET	GR	リセットベクタからのユーザプログラムの実行
37	GO_TILL	GT	テンポラリブレークポイントまでのユーザプログラムの実行
38	HALT	HA	ユーザプログラムの停止
39	HELP	HE	コマンドラインのヘルプ表示
40	INITIALIZE	IN	デバッグプラットフォームの初期化

項番	コマンド名	短縮形	説明
41	LOG	LO	ロギングファイルの操作
42	MAP_DISPLAY	MA	メモリマッピング情報の表示
43	MAP_SET	MS	メモリマッピングの設定
44	MEMORY_COMPARE	MC	メモリ内容の比較
45	MEMORY_DISPLAY	MD	メモリ内容の表示
46	MEMORY_EDIT	ME	メモリ内容の変更
47	MEMORY_FILL	MF	指定データによるメモリ内容の一括変更
48	MEMORY_FIND	MI	メモリ範囲内でデータを検索
49	MEMORY_MOVE	MV	メモリブロックの移動
50	MEMORY_TEST	MT	メモリブロックのテスト
51	MODE	MO	CPU モードの設定, 表示
52	MODULES	MU	内蔵周辺機能の設定, 表示
53	MONITOR_CLEAR	MOC	モニタポイントの削除
54	MONITOR_DISPLAY	MOD	モニタ内容の表示
55	MONITOR_REFRESH	MOR	モニタ内容の自動更新制御
56	MONITOR_SET	MOS	モニタポイントの設定と表示
57	OPEN_WORKSPACE	OW	ワークスペースのオープン
58	QUIT	QU	High-performance Embedded Workshop の終了
59	RADIX	RA	入カラディックス(基数)の設定
60	REFRESH	RF	メモリ関連ウィンドウの更新
61	REGISTER_DISPLAY	RD	CPU レジスタ値の表示
62	REGISTER_SET	RS	CPU レジスタ値の設定
63	REMOVE_FILE	REM	カレントプロジェクトからのファイル削除
64	RESET	RE	CPU のリセット
65	SAVE_SESSION	SE	現在のセッションをセーブ
66	SLEEP	-	コマンド実行の遅延
67	STATUS	STS	[ステータス]ウィンドウの内容表示
68	STEP	ST	ステップ実行(命令単位またはソース行単位)
69	STEP_MODE	SM	ステップモードの設定
70	STEP_OUT	SP	PC 位置の関数を終了するまでのステップ実行
71	STEP_OVER	SO	ステップオーバー実行
72	STEP_RATE	SR	ステップ実行速度の設定, 表示
73	SUBMIT	SU	コマンドファイルの実行
74	SYMBOL_ADD	SA	シンボルの設定
75	SYMBOL_CLEAR	SC	シンボルの削除
76	SYMBOL_LOAD	SL	シンボル情報ファイルのロード
77	SYMBOL_SAVE	SS	シンボル情報のファイルセーブ
78	SYMBOL_VIEW	SV	シンボルの表示
79	SAVE_WORKSPACE	SW	現在のワークスペースの保存
80	TCL	-	TCL の有効/無効の切換え
81	TIMER	TI	実行時間測定タイマの分解能の設定, 表示
82	TOOL_INFORMATION	TO	現在登録されているツールの情報をファイルへ出力
83	TRACE	TR	トレース情報の表示
84	TRACE_ACQUISITION	TA	トレース取得条件の設定, 表示
85	TRACE_BINARY_COMPARE	TBC	トレースバイナリファイルと現在のトレース情報の比較
86	TRACE_BINARY_SAVE	TBV	トレース情報をバイナリファイルに保存
87	TRACE_FILTER	TF	トレース情報のフィルタ
88	TRACE_STATISTIC	TST	統計情報解析の実行
89	TRACE_SAVE	TV	トレース情報をテキストファイルに保存

項番	コマンド名	短縮形	説明
90	TRIGGER_CLEAR	TGC	EXT.2 のトリガ出力条件の削除
91	TRIGGER_DISPLAY	TGD	EXT.2 のトリガ出力条件の表示
92	TRIGGER_SET	TGS	EXT.2 のトリガ出力条件の設定
93	UPDATE_ALL_DEPENDENCIES	UD	カレントプロジェクトの依存関係を更新
94	USER_SIGNALS	US	ユーザシグナル情報の有効/無効の切り換え
95	WATCH_ADD	WA	Watch アイテムの追加
96	WATCH_AUTO_UPDATE	WU	Watch アイテムの自動更新の設定または解除
97	WATCH_DELETE	WD	Watch アイテムの削除
98	WATCH_DISPLAY	WI	[ウォッチ]ウィンドウの内容の表示
99	WATCH_EDIT	WE	Watch アイテムの値の編集
100	WATCH_EXPAND	WX	Watch アイテムの展開または縮小
101	WATCH_RADIX	WR	Watch アイテムの表示基数の変更
102	WATCH_SAVE	WS	[ウォッチ]ウィンドウの表示内容をファイルに保存

各コマンドのシンタックスについてはオンラインヘルプを参照ください。

付録 D ハードウェア診断プログラムについて

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。

D.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

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(HS3064BEPI61H)
- PC
- E6000 PC インタフェースボード(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。)
 - 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)
 - LANアダプタ (HS6000ELN01H)
 - USBインタフェースアダプタ (HS6000EIU01HまたはHS6000EIU02H)

- (1) 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。
- (2) 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。
- (3) 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。
- (4) 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。
- (5) E6000エミュレータ本体の電源をオンにしてください。

D.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(HS3064BEPI61SR)を 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、High-performance Embedded Workshop がインストールされていることを前提としています。

- (1) PCIバスインタフェースボード (HS6000EIC01HまたはHS6000EIC02H)
> TM3064B -PCI (Enter)
- (2) PCMCIAインタフェースカード (HS6000EIP01H)
> TM3064B -PCCD (Enter)
- (3) LANアダプタ (HS6000ELN01H)
> TM3064B -ELN (Enter)
- (4) USBインタフェースアダプタ (HS6000EIU01HまたはHS6000EIU02H)
> TM3064B -USB (Enter)

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\TM3064B -PCI (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、> TM3064B -PCI -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.18 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】
1. <ドライブ>は CD-ROM ドライブのドライブ文字です。
 2. 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.18までです。

<p>E6000 H8/3064B Emulator Tests Vx.x</p> <p>Option memory board fitted? (1. None 2. 1MB 3. 4MB) : 1</p> <p>Loading driverOK (Use PCI)</p> <p>Initializing driverOK</p> <p>Searching for interface cardOK</p> <p>Checking emulator is connectedOK</p> <p>Emulator board information: Main board ID: H'5 Emulation board ID: H'1b</p> <p>COM board ID: None Option memory board: None</p> <p>01) Testing Register :</p> <p>A) IDRO RegisterOK</p> <p>B) PAGE RegisterOK</p> <p>C) TRACE G/A RegisterOK</p> <p>D) PERFM G/A RegisterOK</p> <p>E) CES G/A RegisterOK</p> <p>F) IDR1 RegisterOK</p> <p>G) IDR2 RegisterOK</p> <p>02) Testing DPRAM :</p> <p>A) Decode TestOK</p> <p>B) Marching TestOK</p> <p>03) Testing Firmware RAM :</p> <p>A) Decode Test page[H'700 - H'71f]OK</p> <p>B) Marching test page[H'700 - H'71f]OK</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>オプションメモリは使用しません。“1”を入力してEnterキーを押してください。</p> <p>ドライバソフトウェアが正しくロードできたことを示します。</p> <p>ドライバソフトウェアが正しく初期化できたことを示します。</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。</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>E6000エミュレータのID番号で、常に下基板は5、上基板は1bを示します。</p> <p>COMボードのID番号、オプションメモリボードの有無を示します。</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>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> <p>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。</p> <p>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。</p>
---	---

付録D ハードウェア診断プログラムについて

04) Testing Trace memory : A) Decode Test page[H'000 - H'04f](Lower 32K)OK	E6000エミュレータ上のTrace RAM(前半)のデコードテストチェック結果(正常終了)を示します。
B) Marching test page[H'000 - H'04f](Lower 32K)OK	E6000エミュレータ上のTrace RAM(前半)のマーチングテストチェック結果(正常終了)を示します。
C) Decode Test page[H'000 - H'04f](Upper 32K)OK	E6000エミュレータ上のTrace RAM(後半)のデコードテストチェック結果(正常終了)を示します。
D) Marching test page[H'000 - H'04f](Upper 32K)OK	E6000エミュレータ上のTrace RAM(後半)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 Map control memory :	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。
A) Decode Test page[H'200 - H'27f]OK	
B) Marching test page[H'200 - H'27f]OK	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 Internal ROM and RAM :	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。
A) Decode Test (Internal ROM)OK	
B) Marching test (Internal ROM)OK	
C) Decode Test (Internal RAM)OK	
D) Marching Test (Internal RAM)OK	
07) Testing Option RAM : No option memory board fitted - test skipped	オプションメモリボードのチェック結果(未実装)を示します。
08) Testing Emulation RAM 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。
A) Step OperationOK	
09) Testing Keybreak :	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。
A) Key BreakOK	
10) 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。
A) GRD BreakOK	
B) WPT BreakOK	
C) WPT(ROM) BreakOK	
11) Testing Internal ROM Write-Protect :	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。
A) Write-ProtectOK	

- 12) Testing Hardware Break :
- A) Break Point InitializedOK
 - B) Event Detectors CES channel 1-12OK
 - C) Test Sequencing 1OK
 - D) Check Range BreakOK
 - E) Range Break Test for DataOK
 - F) Check Compare EitherOK
- ハードウェアブレイク制御回路のチェック結果(正常終了)を示します。
- 13) Testing Emulation RAM Trace :
- A) Free TraceOK
 - B) Range TraceOK
 - C) Point to Point TraceOK
 - D) Start and Stop Event TraceOK
 - E) Trace memory OverflowOK
 - F) Time STAMP Trace (35MHz)OK
 - G) Time STAMP Trace (25MHz)OK
 - H) Time STAMP Trace (12MHz)OK
- トレース制御回路のチェック結果(正常終了)を示します。
- 14) Testing Runtime Counter :
- Testing Internal Clock = 16.00MHzOK
 - Testing Internal Clock = 8.00MHzOK
 - Testing CLK Generate = 34.00MHzOK
 - Testing CLK Generate = 16.18MHzOK
 - Testing CLK Generate = 11.46MHzOK
 - Testing CLK Generate = 13.56MHzOK
 - Testing CLK Generate = 22.47MHzOK
- 実行時間測定カウンタのチェック結果(正常終了)を示します。
- 15) Testing Emulation Monitor :
- A) ACST3-ACST0OK
 - B) EMA23-EMAOOK
 - C) ASEST3-ASEST0OK
 - D) ASEBRKACK (MONITOE)OK
 - E) CNNOK
 - F) NOCLKOK
- エミュレーションモニタ制御回路のチェック結果(正常終了)を示します。
- 16) Testing PERFM G/A :
- A) Time Measurement (Unit 20ns)OK
 - B) Time Measurement (Unit Target)OK
 - C) Subroutine Count MeasurementOK
 - D) Timeout Function (TIMOT Bit)OK
 - E) Timeout Function (TIMOP Bit)OK
- パフォーマンス測定制御回路のチェック結果(正常終了)を示します。
- 17) Testing Bus Monitor :
- A) RegisterOK
 - B) Parallel RAMOK
 - C) SPRSEL2OK
 - D) RAM MonitorOK
- バスモニタ制御回路のチェック結果(正常終了)を示します。
- 18) Testing Parallel Access :
- A) Internal ROM Parallel Read Access(WORD)OK
 - B) Internal ROM Parallel Write Access(WORD)OK
 - C) Internal ROM Parallel Write Access(High Byte)OK
 - D) Internal ROM Parallel Write Access(Low Byte)OK
 - E) Option RAM Parallel Read Access(WORD)SKIP
 - F) Option RAM Parallel Write Access(WORD)SKIP
 - G) Option RAM Parallel Write Access(High Byte)SKIP
 - H) Option RAM Parallel Write Access(Low Byte)SKIP
- パラレルアクセス制御回路のチェック結果(正常終了)を示します。

付録D ハードウェア診断プログラムについて

Tests run for xH:xM:xS

テスト時間を示します。

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。

-S をコマンドラインに追加した場合は、No.18 のテスト終了後、再び No.1 のテストから繰り返し実行されます。

D.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(HS3064BEPI61H)
 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) 拡張 I/O ボード
 型式 HS6000EIO _____ H、シリアルNo. _____、レビジョン _____
 (基板上に捺印表示しています)
 - e) オプションボード
 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)
 - f) High-performance Embedded Workshop (HS3064BEPI61SR) バージョンV _____
 (CD-RにV.x.xx Release xxと表示しています)
 - g) ご使用になっているPC
 メーカー名 _____、型式 _____
 使用OS(Windows®98SE, WindowsNT®4.0, Windows®Me, Windows®2000, Windows®XP 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) 動作モード：モード _____
 - c) ターゲットシステム電圧： _____ V
 - d) 使用クロック：(貸出しクロック、Xtal発振、外部クロック入力 いずれかに)
 - e) 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) High-performance Embedded Workshopがエミュレータと接続できない
 (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。

付録D ハードウェア診断プログラムについて

- 4) メモリのデータ化けは発生していますか？(はい、いいえ いずれかに)
- 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_____ いずれかに)
- b) データ化けを起こしている箇所はMemoryウィンドウでリード/ライトできますか？
(はい、いいえ いずれかに)
- 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 いずれかに)
- 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 いずれかに)
- 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。

ルネサスマイクロコンピュータ開発環境システム
ユーザーズマニュアル
H8/3064B E6000エミュレータ

発行年月日 2006年3月22日 Rev.2.00

発行 株式会社ルネサス テクノロジ 営業企画統括部
〒100-0004 東京都千代田区大手町 2-6-2

編集 株式会社ルネサスソリューションズ
グローバルストラテジックコミュニケーション本部
カスタマサポート部

営業お問合せ窓口
株式会社ルネサス販売



<http://www.renesas.com>

本			社	〒100-0004	千代田区大手町2-6-2 (日本ビル)	(03) 5201-5350
京			社	〒212-0058	川崎市幸区鹿島田890-12 (新川崎三井ビル)	(044) 549-1662
西	浜	支	社	〒190-0023	立川市柴崎町2-2-23 (第二高島ビル2F)	(042) 524-8701
東	東	支	社	〒980-0013	仙台市青葉区花京院1-1-20 (花京院スクエア13F)	(022) 221-1351
い	北	支	店	〒970-8026	いわき市平小太郎町4-9 (平小太郎ビル)	(0246) 22-3222
茨	わ	支	店	〒312-0034	ひたちなか市堀口832-2 (日立システムプラザ勝田1F)	(029) 271-9411
新	城	支	店	〒950-0087	新潟市東大通1-4-2 (新潟三井物産ビル3F)	(025) 241-4361
松	潟	支	社	〒390-0815	松本市深志1-2-11 (昭和ビル7F)	(0263) 33-6622
中	本	支	社	〒460-0008	名古屋市中区栄4-2-29 (名古屋広小路ブレイス)	(052) 249-3330
関	部	支	社	〒541-0044	大阪市中央区伏見町4-1-1 (明治安田生命大阪御堂筋ビル)	(06) 6233-9500
北	西	支	社	〒920-0031	金沢市広岡3-1-1 (金沢パークビル8F)	(076) 233-5980
広	陸	支	店	〒730-0036	広島市中区袋町5-25 (広島袋町ビルディング8F)	(082) 244-2570
鳥	島	支	店	〒680-0822	鳥取市今町2-251 (日本生命鳥取駅前ビル)	(0857) 21-1915
九	取	支	社	〒812-0011	福岡市博多区博多駅前2-17-1 (ヒロカネビル本館5F)	(092) 481-7695

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：コンタクトセンタ E-Mail: csc@renesas.com

H8/3064B E6000 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

RJ10B0077-0200