

お客様各位

---

## カタログ等資料中の旧社名の扱いについて

---

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

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

2010年4月1日  
ルネサスエレクトロニクス株式会社

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

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

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事事業の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット

高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

特定水準： 航空機器、航空宇宙機器、海中継機器、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が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。

# SuperH™ RISC engine シミュレータ / デバッガ

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



## 本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社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弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証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など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事用途の目的で使用しないでください。また、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、それらの定めるところにより必要な手続を行ってください。
4. 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などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊社営業窓口で最新の情報をご確認頂きますとともに、弊社ホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意下さい。
5. 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のですが、万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独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下さい。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会下さい。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないで下さい。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
  - 1) 生命維持装置。
  - 2) 人体に埋め込み使用するもの。
  - 3) 治療行為（患部切り出し、薬剤投与等）を行なうもの。
  - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）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致します。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断り致します。
13. 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弊社営業窓口までご照会下さい。



---

# はじめに

---

## このマニュアルについて

このマニュアルは、High-performance Embedded Workshop (HEW)に付属しているシミュレータ・デバッガのデバッグ機能について説明します。

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。

HEW は、インストール上、各種言語にカスタマイズされています。このマニュアルでは、HEW アプリケーションの日本語版について説明します。

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
“ <u>文字列の入力</u> ”	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

---

# 目次

---

1.	はじめに	1
2.	シミュレータ・デバッガの機能	3
2.1	特長	3
2.2	デバッグ対象プログラム	3
2.3	シミュレーション範囲	4
2.4	SH-4 シリーズのサポート機能	6
2.4.1	BSC (バスステートコントローラ)	6
2.4.2	DMAC (ダイレクトメモリアクセスコントローラ)	6
2.5	外部/内部クロック比	7
2.6	エンディアン	7
2.7	MMU (メモリマネージメントユニット)	8
2.8	キャッシュ	9
2.8.1	キャッシュの表示	9
2.8.2	キャッシュヒット率	13
2.9	タイマ	14
2.9.1	サポート範囲	14
2.9.2	制御レジスタ	14
2.9.3	クロック	15
2.9.4	タイマを使用する	15
2.9.5	タイマ使用時の注意	15
2.10	制御レジスタ	16
2.11	パイプラインリセット処理	19
2.12	例外処理	19
2.13	メモリ管理	21
2.14	トレース	21
2.15	標準入出力およびファイル入出力処理	22
2.16	ブレーク条件	23
2.17	浮動小数点データ	26
2.18	関数呼び出し履歴の表示	26
2.19	パフォーマンス測定	27
2.19.1	プロファイラ	27
2.19.2	パフォーマンス解析	27
2.20	擬似割込み	28
2.21	カバレッジ	28

3.	デバッグ	29
3.1	ワークスペースの作成	29
3.1.1	ターゲットの選択	29
3.1.2	シミュレータ用ワークスペースの設定	30
3.2	シミュレータ・デバッガの設定を変更する	32
3.2.1	シミュレータシステムの設定を変更する	32
3.2.2	メモリマップおよびメモリアリソースの設定を変更する	33
3.2.3	メモリマップ設定ダイアログボックス	36
3.2.4	ステート設定ダイアログボックス	37
3.2.5	メモリアリソース設定ダイアログボックス	39
3.3	周辺機能シミュレーションを設定する	40
3.3.1	周辺機能シミュレーションモジュールを登録する	40
3.3.2	周辺機能のアドレスを変更する	41
3.3.3	周辺機能の割り込み要因情報を変更する	42
3.3.4	バンク制御レジスタのアドレスを変更する(SH2A-FPUのみ)	43
3.3.5	割り込み優先順位レジスタのメモリアリソースを確保する	43
3.3.6	接続されている周辺機能を確認する	44
3.4	シミュレータ・デバッガのブレークポイントを使用する	44
3.4.1	ブレークポイントを一覧表示する	44
3.4.2	ブレークポイントを設定する	45
3.4.3	ブレークポイントの設定内容を変更する	53
3.4.4	ブレークポイントを有効にする	53
3.4.5	ブレークポイントを無効にする	53
3.4.6	ブレークポイントを削除する	53
3.4.7	ブレークポイントをすべて削除する	53
3.4.8	ブレークポイントのソース行を表示する	53
3.4.9	入出力ファイルを閉じる	53
3.4.10	入出力ファイルをすべて閉じる	53
3.5	トレース情報を見る	54
3.5.1	トレースウィンドウを開く	54
3.5.2	トレース情報取得条件を設定する	54
3.5.3	トレース情報を取得する	55
3.5.4	トレースレコードを検索する	60
3.5.5	トレース情報をクリアする	61
3.5.6	トレース情報をファイルに保存する	61
3.5.7	ソースファイルを表示する	62
3.5.8	ソース表示を整形する	62
3.5.9	統計情報を解析する	63
3.6	プロファイル情報を見る	64
3.6.1	スタック情報ファイル	64
3.6.2	プロファイル情報ファイル	65
3.6.3	スタック情報ファイルのロード	66
3.6.4	プロファイルを有効にする	67
3.6.5	測定方法を指定する	67
3.6.6	ユーザプログラムを実行し結果を確認する	67

3.6.7	List シート	68
3.6.8	Tree シート	69
3.6.9	プロファイルチャートウィンドウ	72
3.6.10	表示データの種類および用途	73
3.6.11	プロファイル情報ファイルを作成する	74
3.6.12	注意事項	74
3.7	パフォーマンスを解析する	75
3.7.1	パフォーマンス解析ウィンドウを開く	75
3.7.2	評価関数を設定する	76
3.7.3	データ収集を開始する	76
3.7.4	データをリセットする	76
3.7.5	評価関数を削除する	76
3.7.6	すべての評価関数を削除する	76
3.7.7	現在の表示内容をセーブする	76
3.8	コードカバレッジを測定する	77
3.8.1	カバレッジウィンドウを開く	77
3.8.2	カバレッジ情報をすべて取得する	79
3.8.3	カバレッジ情報をすべてクリアする	79
3.8.4	ソースファイルを表示する	79
3.8.5	新しいカバレッジ測定範囲を設定する	79
3.8.6	カバレッジ測定範囲を変更する	79
3.8.7	カバレッジ測定範囲を削除する	80
3.8.8	カバレッジ情報を取得する	80
3.8.9	カバレッジ情報をクリアする	80
3.8.10	カバレッジ情報をファイルに保存する	81
3.8.11	カバレッジ情報をファイルからロードする	81
3.8.12	最新の情報に更新する	81
3.8.13	Confirmation Request ダイアログボックス	81
3.8.14	カバレッジ情報を保存ダイアログボックス	82
3.8.15	エディタウィンドウへのカバレッジ結果表示	82
3.8.16	逆アセンブリウィンドウへのカバレッジ結果表示	83
3.9	手動で擬似割り込みを発生させる	84
3.9.1	トリガウィンドウ	84
3.9.2	GUI I/O ウィンドウ	86
3.10	標準入出力およびファイル入出力を行う	88
3.10.1	I/O シミュレーションウィンドウを開く	88
3.10.2	入出力機能	89
3.11	TLB 内容を見る	91
3.11.1	TLB ウィンドウを開く	91
3.11.2	TLB 内容を変更する	96
3.11.3	TLB をフラッシュする	96
3.11.4	TLB 項目を検索する	96
3.11.5	次を検索する	97
3.11.6	現在の表示内容をセーブする	97
3.12	キャッシュ内容を見る	97
3.12.1	キャッシュウィンドウを開く	97

3.12.2	キャッシュ内容を変更する .....	102
3.12.3	キャッシュ内容をフラッシュする .....	102
3.12.4	キャッシュ内容を検索する .....	102
3.12.5	次を検索する .....	103
3.12.6	キャッシュ容量を変更する .....	103
3.12.7	現在の表示内容をセーブする .....	106
3.13	レジスタバンク内容を見る .....	107
3.13.1	レジスタバンクウィンドウを開く .....	107
3.13.2	レジスタバンク内容を変更する .....	108
3.13.3	表示基数を変更する .....	108
3.14	仮想入出力パネルを作成する .....	108
3.14.1	GUI 入出力ウィンドウを開く .....	109
3.14.2	ボタンを作成する .....	110
3.14.3	ラベルを作成する .....	111
3.14.4	LED を作成する .....	113
3.14.5	テキスト文字列を作成する .....	116
3.14.6	アイテムのサイズ/配置を変更する .....	116
3.14.7	アイテムをコピーする .....	117
3.14.8	アイテムを削除する .....	117
3.14.9	グリッド線を表示する .....	117
3.14.10	入出力パネル情報を保存する .....	118
3.14.11	入出力パネル情報を読み込む .....	118
4.	ウィンドウ .....	119
5.	コマンドライン .....	121
5.1	コマンド一覧 (機能別) .....	121
5.1.1	実行関連 .....	121
5.1.2	ダウンロード関連 .....	121
5.1.3	レジスタ操作関連 .....	122
5.1.4	レジスタバンク操作関連 (SH2A-FPU のみ) .....	122
5.1.5	メモリ操作関連 .....	122
5.1.6	アセンブル/逆アセンブル関係 .....	122
5.1.7	ブレーク設定関連 .....	122
5.1.8	トレース関連 .....	123
5.1.9	カバレッジ計測関連 .....	123
5.1.10	パフォーマンス測定関連 .....	123
5.1.11	ウォッチ関連 .....	124
5.1.12	スクリプト/ログファイル関連 .....	124
5.1.13	マップ関連 .....	124
5.1.14	シミュレーションの設定関連 .....	124
5.1.15	標準入出力およびファイル入出力関連 .....	125
5.1.16	ユーティリティ関連 .....	125
5.1.17	プロジェクト/ワークスペース関連 .....	125
5.1.18	テスト支援機能関連 .....	125
5.2	コマンド一覧 (アルファベット順) .....	126
6.	メッセージ一覧 .....	131

6.1	インフォメーションメッセージ.....	131
6.2	エラーメッセージ.....	132
7.	チュートリアル.....	137
7.1	デバッグの準備.....	137
7.1.1	サンプルプログラム.....	137
7.1.2	サンプルプログラムの作成.....	137
7.2	デバッグのための設定.....	138
7.2.1	メモリリソースの確保.....	138
7.2.2	サンプルプログラムのダウンロード.....	139
7.2.3	ソースプログラムの表示.....	140
7.2.4	PC ブレークポイントの設定.....	141
7.2.5	プロファイラの設定.....	142
7.2.6	I/O シミュレーションの設定.....	143
7.2.7	トレース情報取得条件の設定.....	144
7.2.8	スタックポインタ、プログラムカウンタの設定.....	145
7.3	デバッグ開始.....	146
7.3.1	プログラムの実行.....	146
7.3.2	トレースバッファの使い方.....	148
7.3.3	トレース検索の実行.....	150
7.3.4	I/O シミュレーションの確認.....	151
7.3.5	ブレークポイントの確認.....	151
7.3.6	変数の参照.....	152
7.3.7	プログラムのステップ実行.....	153
7.3.8	プロファイラ情報の確認.....	156



---

# 1. はじめに

---

シミュレータ・デバッガは、ルネサステクノロジのマイクロコンピュータのCPUシミュレーション機能およびデバッグ機能を持っています。シミュレータ・デバッガを利用することによって、C/C++言語やアセンブリ言語で作成されたプログラムを効率よくデバッグすることができます。

シミュレータ・デバッガは、High-performance Embedded Workshop (HEW)とともに動作します。HEWは、ルネサステクノロジ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かれ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シミュレータ・デバッガのアクセス、計測、および変更に関して、HEW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。

シミュレータ・デバッガを使用するにあたっては、シミュレータ・デバッガ、およびHEWのヘルプも参照ください。

## 1. はじめに

---

---

## 2. シミュレータ・デバッガの機能

---

本章では、SuperH™ RISC engineシミュレータ・デバッガの機能について説明します。

### 2.1 特長

本シミュレータ・デバッガには、次のような特長があります。

- (1) ホスト計算機上で動作するので、実機がなくてもプログラムのデバッグを開始することができます。システム全体の開発期間を短縮できます。
- (2) 擬似割り込み機能、擬似タイマ機能、およびI/Oシミュレーション機能を持ち、簡易的なシステムレベルシミュレーションを行えます。
- (3) 下記のような機能を持ち、プログラムのテスト、およびデバッグを効率よく進めることができます。
  - SuperH™ RISC engine の各 CPU に対応
  - ユーザプログラムの実行中に異常が発生した場合、異常を無視して続行するか、または停止するかを制御する機能
  - プロファイルデータ取得、および関数単位のパフォーマンス測定
  - 豊富なブレーク機能
  - メモリマップの設定・編集
  - 関数呼び出し履歴の表示
  - C/C++およびアセンブラソースレベルのカバレッジ表示
  - メモリ内容の画像表示、波形表示による視覚的デバッグ機能
- (4) Windows®上で動作し、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。SuperH™ RISC engineマイコンの各々のメモリマップに対応した環境設定もダイアログボックス上で行うことができます。

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

### 2.2 デバッグ対象プログラム

シミュレータ・デバッガでは、ELF/DWARF2 フォーマットのロードモジュールがシンボリックデバッグ可能です。その他のフォーマットのロードモジュールについては、ダウンロードと命令実行はできますが、シンボリックデバッグはできません。詳しくは、「High-performance Embedded Workshop V.4.03 ユーザーズマニュアル」の「17.18.2 Elf/Dwarf2 のサポート」を参照してください。

### 2.3 シミュレーション範囲

- (1) SH-1、SH-2、SH-2E、SH-3、SH-3E、SH2-DSP、SH3-DSP、SH-4、SH2A-FPU、SH-4Aシリーズのシミュレーションをサポートします。
- (2) SH2-DSPシリーズには、キャッシュをサポートしていない SH2-DSP (Core)、SH2-DSP (SH7410)、SH2-DSP (SH7065) と、サポートしている SH2-DSP (SH7612) があります。本マニュアルでは、SH2-DSPシリーズと表記した場合は、SH2-DSP (Core)、SH2-DSP (SH7410)、SH2-DSP (SH7065) と SH2-DSP (SH7612) をあらわします。
- (3) SH3-DSPシリーズには、DSP機能の異なるSH3-DSP (Core) とSH3-DSPがあります。本マニュアルでは、SH3-DSPシリーズと表記した場合は、SH3-DSP (Core)、SH3-DSPをあらわします。
- (4) SH-4シリーズにはキャッシュ仕様の異なるSH-4とSH-4 (SH7750R) があります。さらにSH-4にはシミュレーション機能を一部制限することでシミュレーション速度を向上させた版と高機能版があり、本マニュアルでは、それぞれSH-4とSH-4BSCと記述し区別します。また、SH-4シリーズと表記した場合は、SH-4、SH-4BSC、SH-4 (SH7750R) をあらわします。
- (5) SH-4AシリーズにはDSPを内蔵したSH4AL-DSPとFPUを内蔵したSH-4Aがあります。拡張版はサポートしていません。本マニュアルでは、SH-4Aシリーズと表記した場合は、SH4AL-DSPとSH-4Aをあらわします。
- (6) SH-1、SH-2、SH-2E、SH-3、SH-3E、SH2-DSP、SH3-DSP、SH-4シリーズでは、パイプラインシミュレーションをサポートします。  
SH3-DSP、およびSH-4には、パイプラインシミュレーションをサポートしたデバッガターゲットのほかに命令単位でシミュレーションするデバッガターゲットがあります。デバッガターゲット名に“Functional Simulator”と付いたものが命令単位版です。Functional Simulator の命令実行サイクル数は、1命令1サイクルとして算出しています。
- (7) SH-4Aシリーズ、SH2A-FPUには、パイプラインシミュレーションをサポートしたデバッガターゲットと、命令単位でシミュレーションするデバッガターゲットがあります。デバッガターゲット名に“Cycle Base Simulator”と付いたものがパイプラインサポート版で、“Functional Simulator”と付いたものが命令単位版です。Functional Simulator の命令実行サイクル数は、1命令1サイクルとして算出しています。Cycle Base Simulator の命令実行サイクル数は、命令内部動作を規定したテーブル内容と命令セットレベルシミュレーション結果から得られるレジスタ等のリソース使用状況に基づいて算出しています。但し、実デバイスの結果と完全には一致しない場合があるため、厳密な実行サイクル数(実行時間)の測定にはエミュレータ・評価ボード等の実デバイスをご使用いただくようお願いいたします。
- (8) シミュレータ・デバッガは、SuperH™ RISC engineマイコンの下記機能をサポートしています。
  - 全実行命令
  - 例外処理
  - レジスタ
  - 全アドレス空間
  - 表 2-1に示す周辺機能

表2-1 シミュレータ・デバッガの機能とCPUの対応

デバッグ プラットフォーム名	エンディアン指定	MMU	キャッシュ	制御 レジスタ	BSC	DMAC	タイマ
SH-1							
SH-2							
SH-2E							
SH-3							
SH-3E							
SH3-DSP							
SH3-DSP Functional Simulator							
SH3-DSP(Core)							
SH-4							
SH-4 Functional Simulator							
SH-4BSC							
SH-4(SH7750R)							
SH2-DSP(SH7410)							
SH2-DSP(Core)							
SH2-DSP(SH7065)							
SH2-DSP(SH7612)							
SH2A-FPU							
SH-4A							
SH4AL-DSP							

【注】 はサポート - は未サポート は部分サポート

(9) シミュレータ・デバッガは SuperH™ RISC engineマイコンの下記機能をサポートしていません。下記機能を使用したプログラムは、SuperH™ RISC engine用エミュレータを使用してデバッグしてください。

- シリアルコミュニケーションインタフェース (SCI)
- I/Oポート

(10) SH2A-FPUでは、割り込み処理に伴うレジスタの退避、復帰を高速に行うためのレジスタバンクをサポートしています。サポートしているレジスタバンク数はバンク0からバンク14までの15個です。

## 2.4 SH-4 シリーズのサポート機能

### 2.4.1 BSC (バスステートコントローラ)

SH-4BSC では、BSC に対応したメモリマップの設定、変更ができます。これにより、BSC を利用したプログラムのデバッグが可能になります。

SH-4/SH-4(SH7750R)では、BSC によるバス制御機能を削除して、SRAM、バス幅、およびステート数指定のみをサポートしています。

SH-4BSC、SH-4/SH-4(SH7750R) シミュレータ・デバッガで設定できるメモリ種別を表 2-2 に示します。

表2-2 SH-4BSC、SH-4/SH-4(SH7750R)シミュレータ・デバッガで設定できるメモリ種別

アドレス	設定できるメモリ種別	
	SH-4BSC	SH-4/SH-4(SH7750R)
H'00000000 ~ H'03FFFFFF (エリア0)	通常メモリ、 バースト ROM、MPX	SRAM
H'04000000 ~ H'07FFFFFF (エリア1)	通常メモリ、 バイト制御 SRAM、MPX	
H'08000000 ~ H'0BFFFFFF (エリア2)	通常メモリ、DRAM、 SDRAM、MPX	
H'0C000000 ~ H'0FFFFFFF (エリア3)	通常メモリ、DRAM、 SDRAM、MPX	
H'10000000 ~ H'13FFFFFF (エリア4)	通常メモリ、 バイト制御 SRAM、MPX	
H'14000000 ~ H'17FFFFFF (エリア5)	通常メモリ、 バースト ROM、MPX	
H'18000000 ~ H'1BFFFFFF (エリア6)	通常メモリ、 バースト ROM、MPX	
H'1C000000 ~ H'1FFFFFFF (エリア7)	設定不可	設定不可
H'7C000000 ~ H'7C001FFF	内蔵 RAM (変更不可)	内蔵 RAM (変更不可)
H'E0000000 ~ H'FFFFFFF	I/O (変更不可)	I/O (変更不可)

なお、表 2-2 では、エリア 0 ~ 7 に対するアドレスは、上位 3 ビットを無視して表示しています。つまり、H'00000000 と H'20000000 はともにエリア 0 に入ります。

シミュレータ・デバッガでは、PCMCIA はサポートしていません。

メモリマップの設定については、「3.2.2 メモリマップおよびメモリリソースの設定を変更する」を参照してください。

### 2.4.2 DMAC (ダイレクトメモリアクセスコントローラ)

SH-4BSC では、4 チャンネルの DMAC の動作をシミュレーションします。これにより、DMAC を利用したプログラムのデバッグが可能になります。

SH-4/SH-4(SH7750R)では、DMAC を使用できません。

## 2.5 外部/内部クロック比

外部/内部クロック比は 1:1 です。

SH-4 シリーズのみ[CLOCK\_RATE]コマンドで変更できます。

[CLOCK\_RATE]コマンドについては、シミュレータ・デバッガのヘルプを参照してください。

## 2.6 エンディアン

SH-2、SH-3/SH-3E/SH3-DSP シリーズ/SH-4 シリーズ/SH-4A シリーズ、SH2-DSP(SH7065)およびSH2-DSP(SH7612)では、メモリ上のデータ格納形式としてビッグエンディアンの他にリトルエンディアンのバイト順をサポートしています。これにより、リトルエンディアンで作成したユーザプログラムのシミュレーション、デバッグが可能になります。

デバッグプラットフォーム選択時に、エンディアンも選択します。プラットフォーム名に”(Little endian)”が付いたものがリトルエンディアン、付いていないものがビッグエンディアンです。

エンディアンは外部メモリへのアクセスすべてに有効です。また、SH3-DSP シリーズの X/Y メモリ、SH-4A の L メモリ、SH4AL-DSP の X/Y/U メモリへのアクセスにも有効になります。

ワードサイズまたはロングワードサイズのデータは、指定したバイト順でメモリへ書き込み、指定したバイト順でメモリから読み出します。

**【注】** エンディアンは、外部メモリすべてに対して共通に設定します。SH2-DSP(SH7612)、SH2-DSP(SH7065)における、エリア単位でのエンディアン指定はサポートしていません。

### 2.7 MMU (メモリマネージメントユニット)

SH-3/SH-3E/SH3-DSP シリーズ/SH-4 シリーズ/SH-4A シリーズでは、TLB、アドレス変換機構、MMU 関連例外 (TLB ミス例外、TLB 保護例外、TLB 無効例外、初期ページ書き込み例外) 等 MMU の動作をシミュレーションします。これにより、MMU でのアドレス変換機構を使用しているユーザプログラムのシミュレーション、デバグが可能となります。また本シミュレータ・デバグは、MMU 関連例外のハンドラルーチン作成時にも利用することができます。MMU は CPU によって異なります。

#### (1) SH-3/SH-3E/SH-3DSP シリーズ

32 エントリ 4 ウェイの TLB の内容を操作するために次のウィンドウおよびダイアログボックスをサポートしています。

- TLB ウィンドウ : TLB の内容の表示およびフラッシュ
- TLB 項目の変更ダイアログボックス : TLB 項目の変更
- TLB の検索ダイアログボックス : TLB 項目の検索

詳しくは「3.11 TLB 内容を見る」を参照してください。

TLB は H'F2000000 番地から H'F3FFFFFF 番地にマッピングしています (TLB の各エントリはこのアドレス範囲に割り付けています)。

- 【注】** (1)SH3-DSP Functional Simulator は MMU の動作をシミュレーションしていません。  
(2)SH-3 Functional Simulator は MMU の動作をシミュレーションしていません。

#### (2) SH-4 シリーズ/SH-4A シリーズ

4 エントリの命令 TLB と 64 エントリの共用 TLB の内容を操作するために次のウィンドウおよびダイアログボックスをサポートしています。

- 命令 TLB ウィンドウ : 命令 TLB の内容の表示・フラッシュ
- 共用 TLB ウィンドウ : 共用 TLB の内容の表示・フラッシュ
- TLB 項目の変更ダイアログボックス : TLB 項目の変更
- TLB の検索ダイアログボックス : TLB 項目の検索

さらに、SH-4A では 16 エントリの PMB の内容を操作するために次のウィンドウをサポートしています。

- PMB ウィンドウ : PMB の内容の表示・フラッシュ

詳しくは「3.11 TLB 内容を見る」を参照してください。

シミュレータ・デバグは、命令 TLB・共用 TLB とともにデータアレイ 2 はサポートしていません。

MMU のアドレス変換機構はシミュレーション時だけでなくダイアログボックス、ウィンドウ上のアドレスにも働きます。ユーザはダイアログボックス、ウィンドウ上でのメモリ操作をデバグ対象と同じ論理アドレスで行うことができます。ただし、[メモリマップ]と[システムメモリリソース]のアドレスは、物理アドレスとして扱います。

- 【注】** (1)TLB エントリへの連想ライト時は、[メモリ]ウィンドウ上での入力では正しく変更できないことがあります。このような場合は、ロングワードフォーマットの [メモリ編集]ダイアログボックスを使用してください。ロングワードフォーマットの [メモリ編集]ダイアログボックスは、ロングワードフォーマットで[メモリ]ウィンドウを開き、変更したいデータをダブ

ルクリックすることにより開きます。

(2)製品によってはPMBを内蔵しない場合があります。

(3)SH-4 Functional SimulatorはMMUの動作をシミュレーションしていません。

## 2.8 キャッシュ

SH-3/SH-3E/SH3-DSPシリーズ/SH-4シリーズ/SH2A-FPUシリーズ/SH-4AシリーズおよびSH2-DSP(SH7612)では、キャッシュの動作をシミュレーションし、キャッシュの内容とヒット率を表示します。これにより、ユーザプログラム実行時のキャッシュの動作を確認することができます。キャッシュはCPUによって異なります。

### 2.8.1 キャッシュの表示

(1) SH-3/SH-3E/SH3-DSPシリーズおよびSH2-DSP(SH7612)

キャッシュの内容を操作するために次のウィンドウおよびダイアログボックスをサポートしています。

- キャッシュウィンドウ : キャッシュの内容の表示およびフラッシュ
- キャッシュ項目の変更ダイアログボックス : キャッシュ項目の変更
- キャッシュの検索ダイアログボックス : キャッシュ項目の検索

また、SH-3/SH-3Eシリーズでは、キャッシュ容量ダイアログボックスによりキャッシュ容量を設定することができます。この機能は、本シミュレータ独自の機能です。内蔵RAM指定により、キャッシュの半分を内蔵RAMとして使用します。キャッシュ容量と使用するウェイの関係を表2-3に示します。

表 2-3 SH-3/SH-3E シリーズシミュレータ・デバッガで設定できるキャッシュ容量

キャッシュ容量	使用するウェイ	内蔵RAM指定 (内蔵RAMとして使用するウェイ)
8キロバイト	Way0~3	可能(Way2~3)
4キロバイト	Way0~1	可能(Way1)
2キロバイト	Way0	不可能(内蔵RAM指定を無視します)

詳しくは、「3.12 キャッシュ内容を見る」を参照してください。

キャッシュは、SH-3/SH-3E/SH3-DSPシリーズではH'F0000000番地からH'F1FFFFFF番地に、SH2-DSP(SH7612)ではアドレスアレイはH'60000000番地からH'7FFFFFFF番地に、データアレイはH'C0000000番地からH'C0000FFF番地にマッピングしています。

(2) SH-4/SH-4BSC

8kバイトの命令キャッシュと16kバイトのオペランドキャッシュおよび32バイトから成る2つのSQ(ストアキュー)の動作をシミュレーションします。キャッシュの内容を操作するために次のウィンドウおよびダイアログボックスをサポートしています。

- 命令キャッシュウィンドウ : 命令キャッシュの内容の表示・フラッシュ
- 共用キャッシュウィンドウ : 共用キャッシュの内容の表示・フラッシュ
- キャッシュ項目の変更ダイアログボックス : キャッシュ項目の変更
- キャッシュの検索ダイアログボックス : キャッシュ項目の検索

## 2. シミュレータ・デバッガの機能

---

詳しくは、「3.12 キャッシュ内容を見る」を参照してください。

命令キャッシュは H'F0000000 番地から H'F1FFFFFF 番地に、オペランドキャッシュは H'F4000000 番地から H'F5FFFFFF 番地に、SQ は H'E0000000 番地から H'E3FFFFFF 番地にマッピングしていません。

**【注】** キャッシュエントリへの連想ライト時、あるいはキャッシュアドレスアレイの変更時は、[メモリ]ウィンドウ上での入力では正しく変更できないことがあります。このような場合は、ロングワードフォーマットの[メモリ編集]ダイアログボックスを使用してください。ロングワードフォーマットの [メモリ編集]ダイアログボックスは、ロングワードフォーマットで[メモリ]ウィンドウを開き、変更したいデータをダブルクリックすることにより開きます。

また、キャッシュアドレスアレイに格納するアドレスタグの上位3ビットは本シミュレータ・デバッガでは0にしません。

キャッシュをマッピングしている空間へ [ダウンロードプログラム]ダイアログボックスによってメモリをロードする場合、あるいは [メモリコピー]ダイアログボックスによってメモリをコピーする場合は、MMUCR の AT ビットをオフにし、MMU ディスエーブルにしてください。

### (3) SH-4(SH7750R)

16k バイトの命令キャッシュと 32k バイトのオペランドキャッシュおよび 32 バイトから成る2つの SQ (ストアキュー) の動作をシミュレーションします。キャッシュの内容を操作するために次のウィンドウおよびダイアログボックスをサポートしています。

- 命令キャッシュウィンドウ : 命令キャッシュの内容の表示・フラッシュ
- 共用キャッシュウィンドウ : 共用キャッシュの内容の表示・フラッシュ
- キャッシュ項目の変更ダイアログボックス : キャッシュ項目の変更
- キャッシュの検索ダイアログボックス : キャッシュ項目の検索

詳しくは、「3.12 キャッシュ内容を見る」を参照してください。

命令キャッシュは H'F0000000 番地から H'F1FFFFFF 番地に、オペランドキャッシュは H'F4000000 番地から H'F5FFFFFF 番地に、SQ は H'E0000000 番地から H'E3FFFFFF 番地にマッピングしています。

**【注】** キャッシュエントリへの連想ライト時、あるいはキャッシュアドレスアレイの変更時は、[メモリ]ウィンドウ上での入力では正しく変更できないことがあります。このような場合は、ロングワードフォーマットの [メモリ編集]ダイアログボックスを使用してください。ロングワードフォーマットの [メモリ編集]ダイアログボックスは、ロングワードフォーマットで[メモリ]ウィンドウを開き、変更したいデータをダブルクリックすることにより開きます。

また、キャッシュアドレスアレイに格納するアドレスタグの上位3ビットは本シミュレータ・デバッガでは0にしません。

キャッシュをマッピングしている空間へ [ダウンロードプログラム]ダイアログボックスによってメモリをロードする場合、あるいは [メモリコピー]ダイアログボックスによってメモリをコピーする場合は、MMUCR の AT ビットをオフにし、MMU ディスエーブルにしてください。

## (4) SH2A-FPU シリーズ

命令キャッシュとオペランドキャッシュ、およびROMキャッシュの動作をシミュレーションします。キャッシュの内容を操作するために次のウィンドウおよびダイアログボックスをサポートしています。

- 命令キャッシュウィンドウ : 命令キャッシュの内容の表示・フラッシュ
- オペランドキャッシュウィンドウ : オペランドキャッシュの内容の表示・フラッシュ
- ROM プリフェッチキャッシュウィンドウ : ROM プリフェッチキャッシュの内容の表示・フラッシュ
- ROM プリフェッチミスキャッシュウィンドウ : ROM プリフェッチミスキャッシュの内容の表示・フラッシュ
- ROM オペランドキャッシュウィンドウ : ROM オペランドキャッシュの内容の表示・フラッシュ
- キャッシュ項目の変更ダイアログボックス : キャッシュ項目の変更
- キャッシュの検索ダイアログボックス : キャッシュ項目の検索

また、キャッシュ容量ダイアログボックスによりキャッシュ容量を設定することができます。この機能は、デバイスの機能ではなく、本シミュレータ独自の機能です。キャッシュ種別とキャッシュ容量の関係を表に示します。

表 2-4 SH2A-FPU シミュレータ・デバッガで設定できるキャッシュ容量

キャッシュ種別	キャッシュ容量
命令キャッシュ	8 キロバイト
	16 キロバイト
	32 キロバイト
	64 キロバイト
	128 キロバイト
	256 キロバイト
	512 キロバイト
	1 メガバイト
オペランドキャッシュ	8 キロバイト
	16 キロバイト
	32 キロバイト
	64 キロバイト
	128 キロバイト
	256 キロバイト
	512 キロバイト
	1 メガバイト

詳しくは、「3.12 キャッシュ内容を見る」を参照してください。

## 2. シミュレータ・デバッガの機能

命令キャッシュのアドレスレイはH'F000000番地からH'F07FFFFFF番地に、データレイはH'F100000番地からH'F17FFFFFF番地にマッピングしています。オペランドキャッシュのアドレスレイはH'F080000番地から H'F0FFFFFF番地に、データレイはH'F180000番地からH'F1FFFFFF番地にマッピングしています。

- 【注】**(1) キャッシュエントリへの連想ライト時、あるいはキャッシュアドレスレイの変更時は、[メモリ]ウィンドウ上での入力では正しく変更できないことがあります。このような場合は、ロングワードフォーマットの [メモリ編集]ダイアログボックスを使用してください。ロングワードフォーマットの [メモリ編集]ダイアログボックスは、ロングワードフォーマットで[メモリ]ウィンドウを開き、変更したいデータをダブルクリックすることにより開きます。
- (2) 命令キャッシュの有効なデータを逆アセンブルウィンドウ、メモリウィンドウより変更した場合は、外部メモリのみ更新され、命令キャッシュのデータは更新されません。命令キャッシュの有効なデータを変更する場合は、無効にしてから変更してください。

### (5) SH-4A シリーズ

命令キャッシュとオペランドキャッシュの動作をシミュレーションします。さらに、SH-4Aでは2次キャッシュおよび32バイトから成る2つの SQ (ストアキュー) の動作をシミュレーションします。キャッシュの内容を操作するために次のウィンドウおよびダイアログボックスをサポートしています。

- 命令キャッシュウィンドウ : 命令キャッシュの内容の表示・フラッシュ
- 共用キャッシュウィンドウ : 共用キャッシュの内容の表示・フラッシュ
- 2次キャッシュウィンドウ(SH-4A のみ) : 2次キャッシュの内容の表示・フラッシュ
- キャッシュ項目の変更ダイアログボックス : キャッシュ項目の変更
- キャッシュの検索ダイアログボックス : キャッシュ項目の検索

また、キャッシュ容量ダイアログボックスによりキャッシュ容量を設定することができます。この機能は、デバイスの機能ではなく、本シミュレータ独自の機能です。キャッシュ種別とキャッシュ容量の関係を表に示します。

表 2-5 SH-4A/SH4AL-DSP シミュレータ・デバッガで設定できるキャッシュ容量

キャッシュ種別	キャッシュ容量
命令キャッシュ	8 キロバイト
	16 キロバイト
	32 キロバイト
	64 キロバイト
	128 キロバイト
オペランドキャッシュ	8 キロバイト
	16 キロバイト
	32 キロバイト
	64 キロバイト
	128 キロバイト
2次キャッシュ(SH-4A のみ)	128 キロバイト
	256 キロバイト
	512 キロバイト
	1 メガバイト

詳しくは、「3.12 キャッシュ内容を見る」を参照してください。

命令キャッシュはH'F000000番地からH'F1FFFFFF番地に、オペランドキャッシュはH'F400000番地から H'F5FFFFFF番地にマッピングしています。さらに、SH-4Aでは2次キャッシュをH'F800000番地からH'F9FFFFFF番地に、SQをH'E000000番地からH'E3FFFFFF番地にマッピングしています。

- 【注】** (1) キャッシュエントリへの連想ライト時、あるいはキャッシュアドレスアレイの変更時は、[メモリ]ウィンドウ上での入力では正しく変更できないことがあります。このような場合は、ロングワードフォーマットの [メモリ編集]ダイアログボックスを使用してください。ロングワードフォーマットの [メモリ編集]ダイアログボックスは、ロングワードフォーマットで[メモリ]ウィンドウを開き、変更したいデータをダブルクリックすることにより開きます。
- (2) 製品によっては2次キャッシュを内蔵しない場合があります。

キャッシュをマッピングしている空間へ [ダウンロードプログラム]ダイアログボックスによってメモリをロードする場合、あるいは [メモリコピー]ダイアログボックスによってメモリをコピーする場合は、MMUCRの ATビットをオフにし、MMUディスエーブルにしてください。

## 2.8.2 キャッシュヒット率

### (1) キャッシュヒット率の取得、表示

キャッシュへのアクセス回数に対するヒット回数の割合をキャッシュヒット率として取得します。キャッシュヒット率は [ステータス]ウィンドウの [Platform]シートに 100 分率で表示します。キャッシュヒット率は、キャッシュヒット回数とキャッシュミス回数の和を分母とし、キャッシュヒット回数を分子として算出しています。

また、キャッシュヒット回数とキャッシュミス回数も表示します。

### (2) キャッシュヒット率の初期化

キャッシュヒット率は、起動時、パイプラインリセット時、CCR 制御レジスタ値変更時に初期化します。また、SH3-DSP シリーズでは、CCR2 制御レジスタ値変更時にも初期化します。

## 2.9 タイマ

### 2.9.1 サポート範囲

シミュレータ・デバッガでは各タイマのチャンネル0だけをサポートしています。

また、オーバフロー/アンダフロー/コンペアマッチによる割り込みだけをサポートしており、インプットキャプチャ等の端子入出力をともなう機能はサポートしていません。

### 2.9.2 制御レジスタ

シミュレータ・デバッガではタイマの制御レジスタをサポートしています。シミュレータ・デバッガでサポートしている制御レジスタを、表 2-6に示します。

表中サポート状況の はサポート、 は「2.9.1 サポート範囲」で説明した機能に関するビットのみサポートしています。

タイマ制御レジスタのアドレス、割り込みベクタ番号、および割り込み優先順位レジスタ位置は[周辺モジュールの構成]ダイアログボックスで参照・変更することができます(SH-4 シリーズは参照のみ)。[周辺モジュールの構成]ダイアログボックスの詳細については、「3.3 周辺機能シミュレーションを設定する」を参照してください。

表2-6 シミュレータ・デバッガでサポートするタイマ制御レジスタ

デバッグ プラットフォーム名	タイマ名	サポートする制御レジスタ	サポート状況
SH-1	ITU0	TSTR	
		TCR	
		TIER	
		TSR	
		TCNT	
		GRA	
		GRB	
SH-2 / SH-2E / SH2-DSP(SH7065)/ SH2A-FPU	CMT0	CMSTR	
		CMCSR	
		CMCNT	
		CMCOR	
SH-3 / SH3-DSP / SH3-DSP(Core) / SH-4 / SH-4BSC / SH-4(SH7750R) / SH4AL-DSP	TMU0	TCR	
		TCNT	
		TSTR	
		TCOR	
SH-4A	TMU0	TSTR0	
		TCOR0	
		TCNT0	
		TCR0	
		TSTR1	
		TCOR3	
		TCNT3	
		TCR3	

デバッグプラットフォーム名	タイマ名	サポートする制御レジスタ	サポート状況
SH2-DSP(Core)	FRT0	TIER	
		FTCSR	
		FRC	
SH2-DSP(Core)	FRT0	OCRA	
		OCRB	
		TCR	
		TCOR	

### 2.9.3 クロック

シミュレータ・デバッガでは、メモリアクセスにかかわる内部クロック、周辺機能用クロック、タイマを動かすクロックをサポートします。

メモリマップで指定するサイクル数は内部クロックになります。内部クロックと周辺機能用クロックの比は、[周辺機能シミュレーションの設定]ダイアログボックスで設定してください。

タイマを動かすクロックを作る分周率は、タイマ制御レジスタで指定してください。

### 2.9.4 タイマを使用する

タイマを使用するにはシミュレータ・デバッガ起動時に表示される[周辺機能シミュレーションの設定]ダイアログボックスでタイマモジュールを登録する必要があります。

タイマモジュール登録の詳細については、「3.3 周辺機能シミュレーションを設定する」を参照してください。

### 2.9.5 タイマ使用時の注意

- (1) 0 クリアだけ可能なビットも、1 をライト可能となっています。
- (2) シミュレーション実行中に SH-2DSP(core)で 16 ビットのレジスタをアクセスするときは、テンポラリレジスタを介して行いますが、コマンドアクセス時はテンポラリレジスタを使用せず、直接レジスタをアクセスします。
- (3) タイマ割込み発生時にブレークするか、しないかを選択できます。

[シミュレータシステム]ダイアログボックスまたは EXEC\_STOP\_SET コマンドで設定してください。

## 2.10 制御レジスタ

SH-3/SH-3E/SH3-DSP シリーズ/SH-4 シリーズ/SH-4A シリーズでは、例外処理、MMU、キャッシュの制御で使用するメモリにマッピングした制御レジスタを、さらに SH-4 シリーズでは、BSC、DMAC の制御で使用する制御レジスタをサポートしています。SH2A-FPU では、キャッシュの制御レジスタと割り込みバンク制御レジスタをサポートしています。また、SH2-DSP (SH7612)ではキャッシュの制御で使用する CCR レジスタのみをサポートしています。これにより、例外処理、MMU 制御、キャッシュ制御、BSC 制御 および DMAC 制御を行っているユーザプログラムのシミュレーション、デバッグを行うことができます。タイマ関連の制御レジスタについては、「2.9.2 制御レジスタ」を参照してください。

本シミュレータ・デバッガでサポートしている制御レジスタは以下の通りです。

・ MMU	PTEH	ページテーブルエントリ上位レジスタ	
	PTEL	ページテーブルエントリ下位レジスタ	
	TTB	変換テーブルベースレジスタ	
	TEA	TLB 例外アドレスレジスタ	
	MMUCR	MMU 制御レジスタ	
	PASCR <sup>*3*4</sup>	物理アドレス空間制御レジスタ	
・ 例外処理	TRA	TRAPA 例外レジスタ	
	EXPEVT	例外事象レジスタ	
	INTEVT	割り込み事象レジスタ	
・ INTC	IBCR <sup>*5</sup>	バンクコントロールレジスタ	
	IBNR <sup>*5</sup>	バンク番号レジスタ	
・ INTC2 <sup>*3</sup>	USERIMASK <sup>*3</sup>	ユーザ割り込みマスクレベル設定レジスタ	
	INT2PRI0	割り込み優先順位設定レジスタ 0	
	INT2PRI1	割り込み優先順位設定レジスタ 1	
	INT2A0	割り込み要因レジスタ (マスク状態の影響なし)	
	INT2A1	割り込み要因レジスタ (マスク状態の影響あり)	
	INT2MSKR	割り込みマスクレジスタ	
	INT2MSKCR	割り込みマスククリアレジスタ	
	INT2B0	モジュール別割り込み要因レジスタ	
	・ キャッシュ	CCR	キャッシュ制御レジスタ
		CCR1 <sup>*5</sup>	キャッシュ制御レジスタ 1
CCR2 <sup>*1*5</sup>		キャッシュ制御レジスタ 2	
QACR0,1 <sup>*2*3</sup>		キューアドレス制御レジスタ 0、1	
RCCR <sup>*5</sup>		ROM キャッシュ制御レジスタ	
RCCR2 <sup>*5</sup>		ROM キャッシュ制御レジスタ 2	
RCCR4 <sup>*5</sup>	ROM キャッシュ制御レジスタ 4		

・ BSC	BCR1,2* <sup>2</sup>	バスコントロールレジスタ 1、 2
	WCR1~2* <sup>2</sup>	ウェイトステートコントロールレジスタ 1 ~ 2
	WCR3* <sup>6</sup>	ウェイトステートコントロールレジスタ 3
	MCR* <sup>6</sup>	個別メモリコントロールレジスタ
	RTCSR* <sup>6</sup>	リフレッシュタイマコントロール/ステータスレジスタ
	RTCNT* <sup>6</sup>	リフレッシュタイマカウンタ
	RTCOR* <sup>6</sup>	リフレッシュタイムコンスタントレジスタ
	RFCR* <sup>6</sup>	リフレッシュカウンタレジスタ
	・ DMAC	SAR0~3* <sup>6</sup>
DAR0~3* <sup>6</sup>		DMA デスティネーションアドレスレジスタ 0 ~ 3
DMATCR0~3* <sup>6</sup>		DMA トランスファカウンタレジスタ 0 ~ 3
CHCR0~3* <sup>6</sup>		DMA チャンネルコントロールレジスタ 0 ~ 3
DMAOR* <sup>6</sup>		DMA オペレーションレジスタ
X/Y メモリ	XSA* <sup>3</sup>	X メモリ転送元アドレスレジスタ
	YSA* <sup>3</sup>	Y メモリ転送元アドレスレジスタ
	XDA* <sup>3</sup>	X メモリ転送先アドレスレジスタ
	YDA* <sup>3</sup>	Y メモリ転送先アドレスレジスタ
	XPR* <sup>3</sup>	X バス保護制御レジスタ
	YPR* <sup>3</sup>	Y バス保護制御レジスタ
	XER* <sup>3</sup>	X バス例外制御レジスタ
	YER* <sup>3</sup>	Y バス例外制御レジスタ
L メモリ	LSA0,1* <sup>4</sup>	L メモリ転送元制御レジスタ 0、 1
	LDA0,1* <sup>4</sup>	L メモリ転送先制御レジスタ 0、 1
RAM	RAMACYC* <sup>5</sup>	RAM アクセスサイクル設定レジスタ

**【注】** \*<sup>1</sup> は、SH3-DSP シリーズのみサポートしています。

\*<sup>2</sup> は、SH-4 シリーズのみサポートしています。

\*<sup>3</sup> は、SH-4A のみサポートしています。

\*<sup>4</sup> は、SH4AL-DSP のみサポートしています。

\*<sup>5</sup> は、SH2A-FPU のみサポートしています。

\*<sup>6</sup> は、SH-4BSC のみサポートしています。

SH-DSP(SH7612)では CCR レジスタのみサポートしています。

シミュレータ・デバッガでは、PCMCIA インタフェース、シンクロナス DRAM モードレジスタはサポートしていません。

制御レジスタの変更や表示は [IO] ウィンドウをご利用ください。詳しくは、「High-performance Embedded Workshop V.4.03 ユーザーズマニュアル」の「17.6 I/O レジスタを見る」を参照してください。

## 2. シミュレータ・デバッガの機能

---

SH-4/SH-4R では、下記レジスタの一部ビットのみサポートしています。

表2-7 SH-4/SH-4(SH7750R)シミュレータ・デバッガでサポートしているビット

レジスタ名	ビット名
BCR1	ENDIAN
BCR2	A6SZ-A0SZ
WCR1	A6IW-A0IW
WCR2	A6W-A0W

未サポート部分については、[IO]ウィンドウ等を使用して値の参照および変更は行なえますが、シミュレータ・デバッガの実行には影響を与えません。

## 2.11 パイプラインリセット処理

シミュレータ・デバッガでは、パイプラインのシミュレーションを行っていますが、以下の場合にパイプラインをリセットします。

- 命令シミュレーション停止後再実行までにPCを変更した
- 実行開始アドレスを指定したRunコマンドを実行した
- イニシャライズまたはプログラムのロードを行った

パイプラインをリセットすると、すでにフェッチおよびデコードしている内容を捨てて、現在のPCからフェッチおよびデコードをやり直します。また、命令実行数および命令実行サイクル数をゼロクリアします。

## 2.12 例外処理

シミュレータ・デバッガでは、TRAPA 命令、一般不当命令、スロット不当命令、アドレスエラー例外の発生を検出します。さらに、SH-3/SH-3E/SH3-DSP/SH-4/SH-4A シリーズでは MMU 関連の例外( TLB ミス例外、TLB 保護例外、TLB 無効例外、初期ページ書き込み例外) 処理を、SH-2E/SH-3E/SH-4 シリーズ/SH2A-FPU /SH-4A では FPU 例外処理を、SH2A-FPU ではレジスタバンクエラー例外処理をシミュレーションします。これにより、例外発生時のシミュレーションも行うことができます。

例外処理のシミュレーションは、[シミュレータシステム]ダイアログボックスにおける [実行モード]の選択に従って、次の手順で行います。

### (1) SH-1/SH-2/SH-2E/SH2-DSP シリーズ/SH2A-FPU

[続行]を選択した場合（続行モード）

- (a) 命令の実行中に例外の発生を検出します。
- (b) PCとSRを退避します。
- (c) ベクタ番号に対応するベクタアドレスから、スタートアドレスを読み出します。
- (d) スタートアドレスから命令実行を行います。スタートアドレスが0の場合は、例外処理を中止し、例外処理エラーが発生したことを表示した後、シミュレータ・デバッガのコマンド待ち状態に戻ります。

[停止]を選択した場合（停止モード）

前項の (a) (b) (c) を行い停止します。

## 2. シミュレータ・デバッガの機能

---

### (2) SH-3/SH-3E/SH3-DSP シリーズ/SH4AL-DSP

[続行]を選択した場合（続行モード）

- (a) 命令の実行中に例外の発生を検出します。
- (b) SPCとSSRにそれぞれPCとSRを退避します。
- (c) SRのBLビット、RBビット、MDビットを1にセットします。
- (d) 制御レジスタEXPEVTに例外コードを設定します。なお、必要に応じて他の制御レジスタにも適切な値を設定します。
- (e) 例外の要因に応じたベクタアドレスをPCに設定します。  
(SRのBLビットが1のときに例外を検出すると例外の要因にかかわらずリセットのベクタアドレスH'A0000000を設定します。)
- (f) PCに設定したアドレスから命令実行を行います。

[停止]を選択した場合（停止モード）

前項の（a）（b）（c）（d）（e）を行い停止します。

### (3) SH-4 シリーズ/SH-4A

[続行]を選択した場合（続行モード）

- (a) 命令に実行中に例外の発生を検出します。
- (b) SPCとSSRにそれぞれPCとSRを退避します。
- (c) SRのBLビット、RBビット、MDビットを1にセットします。
- (d) リセット時SRのFD（FPUディスエーブル）ビットを0にセットします。
- (e) 制御レジスタEXPEVTに例外コードを設定します。なお、必要に応じて他の制御レジスタにも適切な値を設定します。
- (f) 例外の要因に応じたベクタアドレスをPCに設定します。  
(SRのBLビットが1のときに例外を検出すると例外の要因にかかわらずリセットのベクタアドレスH'A0000000を設定します。)
- (g) PCに設定したアドレスから命令実行を行います。

[停止]を選択した場合（停止モード）

前項の（a）（b）（c）（d）（e）（f）を行い停止します。

## 2.13 メモリ管理

### (1) メモリマップの設定

メモリマップは、シミュレーション時のメモリアクセスサイクル数の計算に使用します。設定できる項目は以下の通りです。

- メモリ種別
- メモリ領域の先頭位置、終了位置
- メモリアクセスのサイクル数
- メモリのデータバス幅

設定できるメモリ種別は、CPUによって異なります。詳細は、「3.2 シミュレータ・デバッガの設定を変更する」を参照してください。なお、ユーザプログラムは、内蔵 I/O 空間を除くすべてのメモリで実行可能です。

### (2) メモリリソースの設定

ユーザプログラムをロードして実行させるためにメモリリソースを設定する必要があります。設定できる項目は以下の通りです。

- 開始アドレス
- 終了アドレス
- アクセス種別

アクセス種別は、読み書き可能、読み出しのみ可能、書き込みのみ可能があります。

ユーザプログラムで、読み出しのみ可能メモリへ書き込みを行う等の不正なアクセスを行ったときはエラーとなるので、誤ったメモリアクセスを検出することができます。

メモリリソース設定の詳細は、「3.2 シミュレータ・デバッガの設定を変更する」を参照してください。

## 2.14 トレース

シミュレータ・デバッガは、実行結果をトレースバッファに書き込みます。トレース情報の取得条件は、[トレース取得]ダイアログボックスで指定します。[トレース取得]ダイアログボックスは、[トレース]ウィンドウ上で右クリックしてポップアップメニューを表示し、[設定...]を選択することによって表示できます。取得したトレース情報は、[トレース]ウィンドウに表示します。[トレース]ウィンドウは、CPUによって異なります。

トレース情報は検索することができます。検索条件は、[トレース検索]ダイアログボックスで設定します。[トレース検索]ダイアログボックスは、[トレース]ウィンドウ上で右クリックしてポップアップメニューを表示し、[検索...]を選択することによって表示できます。

詳しくは、「3.5 トレース情報を見る」を参照してください。

### 2.15 標準入出力およびファイル入出力処理

シミュレータ・デバッガでは、ユーザプログラムから標準入出力およびファイル入出力を行うことができます。入出力機能を利用する場合は、必ず [I/O シミュレーション] ウィンドウをオープンしておいてください。サポートしている入出力処理は、以下の通りです。

表2-8 入出力機能一覧

番号	機能コード	機能名	内容
1	H'21	GETC	標準入力からの1バイト入力
2	H'22	PUTC	標準出力への1バイト出力
3	H'23	GETS	標準入力からの1行入力
4	H'24	PUTS	標準出力への1行出力
5	H'25	FOPEN	ファイルのオープン
6	H'06	FCLOSE	ファイルのクローズ
7	H'27	FGETC	ファイルからの1バイト入力
8	H'28	FPUTC	ファイルへの1バイト出力
9	H'29	FGETS	ファイルからの1行入力
10	H'2A	FPUTS	ファイルへの1行出力
11	H'0B	FEOF	エンドオブファイルのチェック
12	H'0C	FSEEK	ファイルポインタの移動
13	H'0D	FTELL	ファイルポインタの現在位置を得る

入出力機能の詳細は、「3.10 標準入出力およびファイル入出力を行う」を参照してください。

## 2.16 ブレーク条件

ユーザプログラムのシミュレーションを中断する条件として以下のものがあります。

- ブレーク系コマンドの条件成立によるブレーク
- ユーザプログラムの実行時エラー検出によるブレーク
- トレースバッファ満杯によるブレーク
- SLEEP 命令実行によるブレーク
- [停止]ボタンによるブレーク

### (1) ブレーク系コマンドの条件成立によるブレーク

ブレーク条件を設定するコマンドには次の6種類があります。

- BREAKPOINT : 命令実行位置によるブレーク
- BREAK\_ACCESS : メモリ範囲のアクセスによるブレーク
- BREAK\_CYCLE : 実行サイクル数によるブレーク
- BREAK\_DATA : メモリ書き込みデータ値によるブレーク
- BREAK\_REGISTER : レジスタ書き込みデータ値によるブレーク
- BREAK\_SEQUENCE : 実行順序を指定したブレーク

ブレーク条件成立時の動作を[Stop]と指定した場合、そのブレーク条件が成立するとプログラムを中断します。詳しくは、「3.4 シミュレータ・デバッガのブレークポイントを使用する」を参照してください。

ユーザプログラム実行中にブレーク条件が成立しプログラムが中断した場合、ブレークポイントの命令を実行しないで停止するか、実行してから停止するかを表2-9に示します。

表2-9 ブレーク条件成立時の処理

コマンド名	ブレーク条件成立命令	
	実行する	実行しない
BREAKPOINT		
BREAK_ACCESS		
BREAK_CYCLE		
BREAK_DATA		
BREAK_REGISTER		
BREAK_SEQUENCE		

BREAKPOINT、BREAK\_SEQUENCE の場合、実行命令の先頭位置以外にブレークポイントを設定するとブレークを検出できません。

ユーザプログラム実行中にブレーク条件が成立すると、ブレーク条件成立のメッセージを[アウトプット]ウィンドウに表示して、命令実行を中断します。

### (2) ユーザプログラムの実行時エラー検出によるブレーク

シミュレータ・デバッガでは、CPUの例外発生機能では検出できないプログラムの誤りを検出するためにシミュレーションエラーを設けています。これらのエラーが発生した場合に、シミュレーションを停止するか、続行するかを [シミュレータシステム]ダイアログボックスにより選択できます。エラーの種類、エラーメッセージ、エラー発生要因、および続行時のシミュレータ・デバッガの動作

## 2. シミュレータ・デバッガの機能

を表 2-10 に示します。

表2-10 シミュレーションエラー一覧

エラーの種類/メッセージ	エラー発生要因	続行モード時処理
メモリアクセスエラー/ Memory Access Error (Address:H'nnnnnnnn)	<ul style="list-style-type: none"> <li>• 確保していないメモリ領域をアクセスしようとした</li> <li>• 書き込み不可属性を持つメモリへ書き込みを行おうとした</li> <li>• 読み出し不可属性を持つメモリから読み出しを行おうとした</li> <li>• メモリが存在しない領域をアクセスしようとした</li> <li>• 不正なサイズで制御レジスタをアクセスした</li> </ul>	メモリへの書き込み時、何も書き込まない。メモリ読み出し時、全ビット"1"を読み出す
命令実行不正/ Illegal Operation	<ul style="list-style-type: none"> <li>• DIV1 命令でゼロ除算を行おうとした</li> <li>• SETRC 命令でゼロを書き込もうとした</li> </ul>	デバイスと同一動作をする
DSP 命令実行不正/ Illegal DSP Operation	<ul style="list-style-type: none"> <li>• PSHA 命令で 32 ビットを越えるシフトを行おうとした</li> <li>• PSHL 命令で 16 ビットを越えるシフトを行おうとした</li> </ul>	デバイスと同一動作をする
DSP 命令コード不正 Invalid DSP Instruction Code	<ul style="list-style-type: none"> <li>• DSP 命令の命令コードが不正である</li> </ul>	常に停止する
TLB マルチヒット/ TLB Multiple Hit	<ul style="list-style-type: none"> <li>• MMU によるアドレス変換で複数の TLB エントリにヒットした (SH-3/SH-3E/SH-3DSP シリーズのみ)</li> </ul>	動作不定

停止モードの場合、シミュレーションエラーが発生するとシミュレータ・デバッガは、命令実行を中止してエラーメッセージを表示後、コマンド待ち状態に戻ります。メモリアクセスエラーの場合は、エラーが発生したアドレスも表示します。シミュレーションエラー停止後の PC の状態を表 2-11 に示します。なお、シミュレーションエラー停止後 SR の内容は変化しません。

表2-11 シミュレーションエラー停止時のレジスタ

エラーの種類	PC の内容
メモリアクセスエラー	命令読込時： <ul style="list-style-type: none"> <li>• SH2-DSP/SH3-DSP シリーズ エラーが発生した命令の 3 命令前のアドレス</li> <li>• SH-1/SH-2/SH-2E/SH-3/SH-3E/SH-4 シリーズ エラーが発生した命令の前命令のアドレス</li> </ul> ただし、分岐先を読み込んだ時にエラーが発生した時は、スロット位置のアドレス 命令実行時： <ul style="list-style-type: none"> <li>• エラーが発生した命令の次命令のアドレス</li> </ul>
命令実行不正	エラーが発生した命令の次命令のアドレス
DSP 命令実行不正	エラーが発生した命令の 2 命令先のアドレス
DSP 命令コード不正	エラーが発生した命令の 2 命令先のアドレス
TLB マルチヒット	エラーが発生した命令のアドレス

シミュレーションエラーが発生する命令を組み込んだプログラムのデバッグは、次の手順で行ってください。

- (a) 最初は停止モードで実行させて、意図している箇所以外にエラーがないかどうかを確認してください。
- (b) 確認が完了したら、続行モードで実行してください。

**【注】** 停止モードでエラーが発生して停止した状態から、モードを続行モードに変更してシミュレーションを再開すると、正しくシミュレーションできない場合があります。シミュレーションを再開する場合は、レジスタ内容（汎用レジスタ、コントロールレジスタ、システムレジスタ）、メモリの内容をエラー発生前の状態に戻してから再実行してください。

(3) トレースバッファ満杯によるブレーク

[トレース取得]ダイアログボックスの [トレースバッファ満杯時の動作]で [停止]モードを指定し、命令実行中にトレースバッファが満杯になると、シミュレータ・デバッガは、実行を中断します。中断時には以下のメッセージを[アウトプット]ウィンドウに表示します。

Trace Buffer Full

(4) SLEEP 命令実行によるブレーク

命令実行時に、SLEEP 命令を実行すると、シミュレータ・デバッガは実行を中断します。中断時には、以下のメッセージを[アウトプット]ウィンドウに表示します。

Sleep

**【注】** 実行を再開する場合は、PC の値を再開位置の命令アドレスに変更してください。

(5) [停止]ボタンによるブレーク

命令実行中にユーザにより強制的に実行を中断することができます。中断時には以下のメッセージを[アウトプット]ウィンドウに表示します。Go、Step コマンドにより実行を再開できます。

Stop

### 2.17 浮動小数点データ

実数データとして浮動小数点数を指定することができます。これにより、データ値等で浮動小数点を扱う場合の操作が容易になります。浮動小数点を指定できる項目は次の通りです。

- ・[ブレーク種別の選択]ダイアログボックスにおいて、ブレーク種別を [ブレークデータ] や [ブレークレジスタ] と指定したときのデータ
- ・[メモリ]ウィンドウにおけるデータ
- ・[メモリフィル]ダイアログボックスにおけるデータ
- ・[メモリ検索]ダイアログボックスにおけるデータ
- ・レジスタ値編集ダイアログボックスでの入力値

浮動小数点データフォーマットは、ANSI C の浮動小数点フォーマットに準拠しています。

浮動小数点数の 10 進->2 進変換で発生する丸めのモードは下記のようになります。

- SH-3、SH3-DSP シリーズ、SH-4 シリーズ、SH-4A シリーズ：
- ゼロ丸め
- 上記以外：
- 最近値丸め

なお、10 進->2 進変換および 2 進->10 進変換で非正規化数を指定した場合、ゼロ丸めモードでは 0 に変換し、最近値モードでは非正規化数のまま処理します。また、10 進->2 進変換時にオーバフローが発生した場合、ゼロ丸めモードでは浮動小数点数の最大値を、最近値モードでは無限大を設定します。

### 2.18 関数呼び出し履歴の表示

シミュレーションの中断時に、関数の呼び出し履歴を [スタックトレース]ウィンドウに表示します。これにより、プログラムの動作の流れを確認することができます。また、[スタックトレース]ウィンドウ上で関数名を選択することにより、該当するソースプログラムを [エディタ]ウィンドウ上に表示します。これにより、中断している関数の他に、その関数を呼び出した元の関数をチェックすることができます。

関数呼び出し履歴を更新するのは、以下のような場合です。

- 「2.16 ブレーク条件」に示す条件によりシミュレーションが中断した時
- 上記中断した状態で、レジスタの値を変更した時
- シミュレーションをステップ実行している時

詳しくは、「High-performance Embedded Workshop V.4.03 ユーザーズマニュアル」の「17.15 関数呼び出し履歴を見る」を参照してください。

## 2.19 パフォーマンス測定

シミュレータ・デバッガはユーザプログラムのパフォーマンスを測定するためにプロファイラ機能およびパフォーマンス解析機能を提供します。

### 2.19.1 プロファイラ

プロファイラは、ユーザプログラムの全体について関数とグローバル変数のアドレス、サイズ、関数の呼び出し回数、およびプロファイルデータを表示します。プロファイルデータはCPUにより異なります。

プロファイル情報はリスト形式、ツリー形式、チャート形式で表示します。

プロファイル情報を用いることにより、サイズが小さく、呼び出し回数が多い関数をインライン関数にするなどの最適化を検討することができます。

また、ファイルに出力したプロファイル情報を用いて、最適化リンケージエディタで動的情報に基づいた最適化を実行することができます。

詳しくは、「3.6 プロファイル情報を見る」および「最適化リンケージエディタマニュアル 4.2.4 最適化オプション PROfile」を参照してください。

### 2.19.2 パフォーマンス解析

パフォーマンス解析はユーザプログラム内の指定関数について実行サイクル数、呼び出し回数を表示します。指定関数のみについてパフォーマンスデータを取得するため、プロファイラよりも高速なシミュレーションが可能です。詳しくは、「3.7 パフォーマンスを解析する」を参照してください。

### 2.20 擬似割り込み

シミュレータ・デバッガでは、シミュレーション中に擬似割り込みを発生させることができます。擬似割り込みを発生させるには、以下の2通りの方法があります。

(1) ブレーク条件成立時の動作による擬似割り込み発生

ブレーク系コマンドで、ブレーク条件成立時の動作に[Interrupt]を指定することにより、擬似割り込みを発生させることができます。

詳しくは、「3.4 シミュレータ・デバッガのブレークポイントを使用する」を参照してください。

(2) ウィンドウによる擬似割り込み発生

[トリガ]ウィンドウ、または[GUI I/O]ウィンドウのボタンをクリックすることにより、擬似割り込みを発生させることができます。

詳しくは、「3.9 手動で擬似割り込みを発生させる」を参照してください。

なお、擬似割り込みが発生してからその割り込みを受け付けるまでの間に、次の擬似割り込みが発生した場合は、優先順位の高い割り込みだけを受付けます。

(3) 擬似割り込み発生によるブレーク

擬似割り込み発生時にブレークするか、しないかを選択できます。

[シミュレータシステム]ダイアログボックスまたはEXEC\_STOP\_SETコマンドで設定してください。

**【注】** 割り込みを受け付けるかどうかの判定は、選択されたデバッグプラットフォームのCPUの仕様に従います。ただし、優先順位にH'11を指定した場合は、常に割り込みを受け付けます。また、割り込みコントローラはシミュレーションしていません。

### 2.21 カバレッジ

シミュレータ・デバッガでは、ユーザが指定した測定範囲について命令実行中に命令カバレッジ情報を収集できます。

測定範囲は直接アドレスを指定して設定するほかに、ソースファイル名を指定してそのファイルに含まれる全関数を設定することができます。

命令カバレッジ情報を利用することで各命令の実行状態を観察できます。さらにプログラムのどの部分が未実行であるかを容易に特定できます。

収集した命令カバレッジ情報は[カバレッジ]ウィンドウに表示します。

命令カバレッジ情報は命令実行済のソース行に対応するカラムを [エディタ]ウィンドウ上に強調表示します。

また、測定対象のアドレス範囲または関数について、カバレッジ統計情報をパーセント形式で表示します。これによりプログラムがどのくらい実行されているかを定量的に把握できます。

命令カバレッジ情報はファイルへの保存およびファイルからのロードが行えます。ロードできるのは".COV"ファイル形式のみです。

詳しくは、「3.8 コードカバレッジを測定する」を参照してください。

---

## 3. デバッグ

---

この章では、シミュレータ・デバッガ特有の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。

HEWで共通な下記機能については、HEWのヘルプを参照ください。

- ・ 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
- ・ 任意のアドレスのメモリ内容を参照 / 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
- ・ 現在の状態の参照
- ・ 複数デバッグプロットフォームの同期
- ・ 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

### 3.1 ワークスペースの作成

シミュレータ・デバッガを使用するためには、まずHEWでワークスペースを作成する必要があります。ここでは、シミュレータ・デバッガ特有の説明のみをします。ワークスペース作成の詳細は、HEWのヘルプを参照ください。

#### 3.1.1 ターゲットの選択

HEWを起動し、新規プロジェクトワークスペースの作成を行う場合、下記ダイアログボックスを表示します。

ここで、シミュレータ・デバッガのターゲット（CPU種別、エンディアン、シミュレータ種別）を選択してください。

### 3. デバッグ

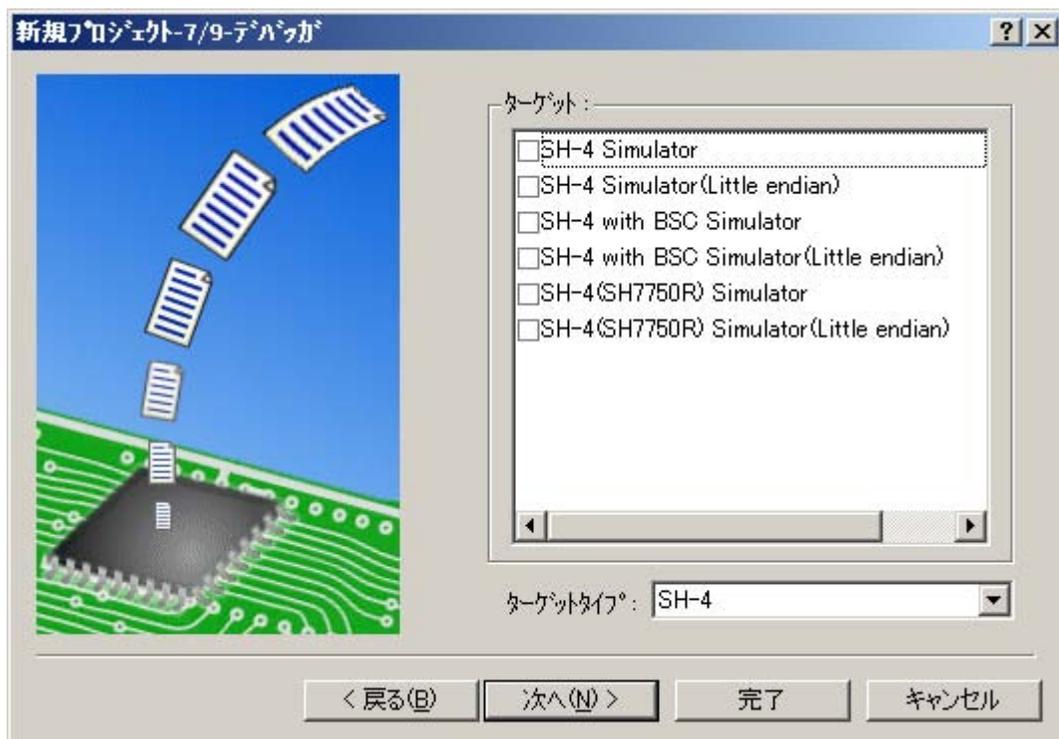


図 3-1 デバッガターゲット設定画面 (7/9)

[ターゲット] デバッガターゲットを設定します。デバッガターゲットを選択(チェック)してください。デバッガターゲットは未選択でも複数選択してもかまいません。

[ターゲットタイプ] [ターゲット]に表示するターゲットの種類を指定します。

【注】新規プロジェクト 2/9 で設定するエンディアンはコンパイラ・アセンブラの設定に反映します。新規プロジェクト 7/9 で選択したデバッガターゲットのエンディアンとは独立しています。

新規プロジェクト 1/9 の [CPU シリーズ] では、FPU 無しの[SH-2A]と、FPU 付きの[SH2A-FPU]を選択できます。しかし、シミュレータ・デバッガは FPU 付きのみをサポートしています。したがって、新規プロジェクト 1/9 で[SH-2A]を選択した場合も、新規プロジェクト 7/9 では、[SH2A-FPU]を選択してください。

#### 3.1.2 シミュレータ用ワークスペースの設定

新規プロジェクト-8/9では、シミュレータ用ワークスペースの設定を行うことができます。

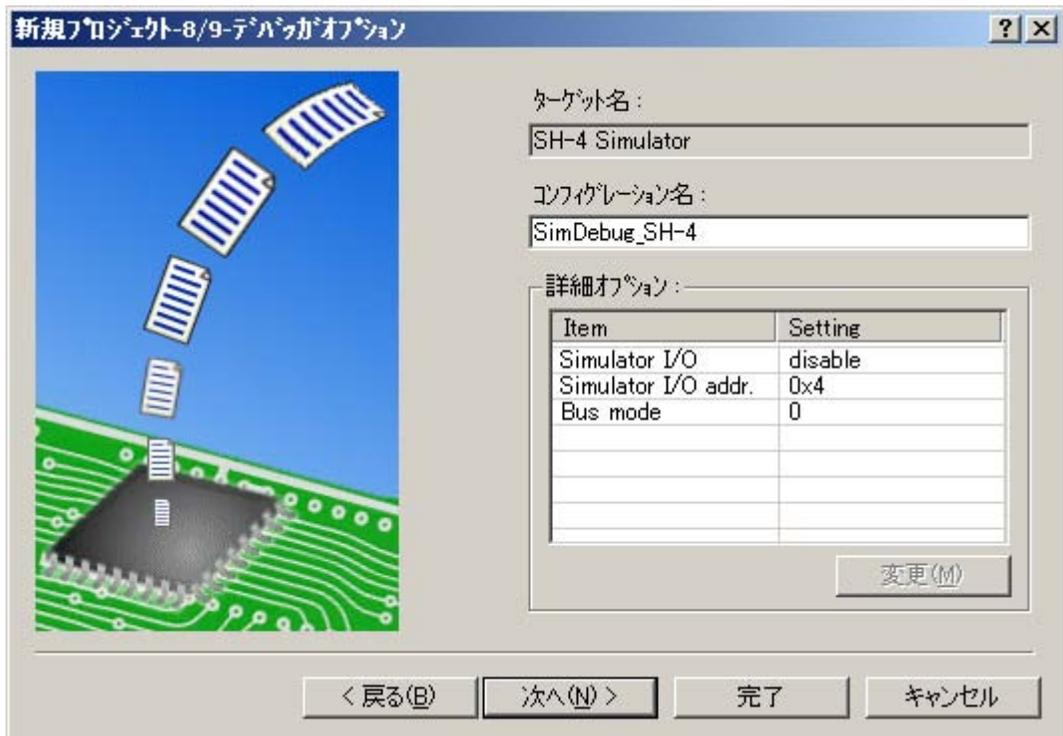


図 3-2 デバッガオプション設定画面 (8/9)

[コンフィギュレーション名]	HEWはデフォルトで[Release]と[Debug]の二つのコンフィグレーションを作成しますが、デバッガターゲットを選択すると選択したターゲット用のコンフィグレーションも作成します(ターゲット名を含んだ略称となります)。このコンフィグレーション名は[コンフィギュレーション名]で変更できます。
[詳細オプション]	デバッガターゲットのオプションを設定します。変更する場合は、[Item]を選択して[変更]をクリックしてください。なお、変更できない項目の場合、[Item]を選択しても[変更]はグレーのままです。
[Simulator I/O]	ユーザプログラムから標準入出力またはファイル入出力を行うI/Oシミュレーションは有効([Enable])または無効([Disable])
[Simulator I/O addr.]	上記I/Oシミュレーション開始アドレス
[On-chip ROM mode]	内蔵ROMの容量(SH2A-FPUのみ) Disable 内蔵ROM無効 Enable (128K byte) 内蔵ROM容量128キロバイト Enable (256K byte) 内蔵ROM容量256キロバイト Enable (512K byte) 内蔵ROM容量512キロバイト Enable (1M byte) 内蔵ROM容量1メガバイト Enable (2M byte) 内蔵ROM容量2メガバイト
[Bus mode]	現状未使用

## 3.2 シミュレータ・デバッグの設定を変更する

本節では、シミュレータ・デバッグ起動後にシミュレータシステム、メモリマップ、およびメモリリソースを変更する方法について説明します。

### 3.2.1 シミュレータシステムの設定を変更する

I/O シミュレーションの開始位置、実行モード等の設定変更は、[シミュレータシステム]ダイアログボックスの[システム]タブで行います。

[シミュレータシステム]ダイアログボックス[システム]タブを開くには、[基本設定->シミュレータ->システム...]を選択するか、[シミュレータシステム]ツールバーボタン  をクリックします。

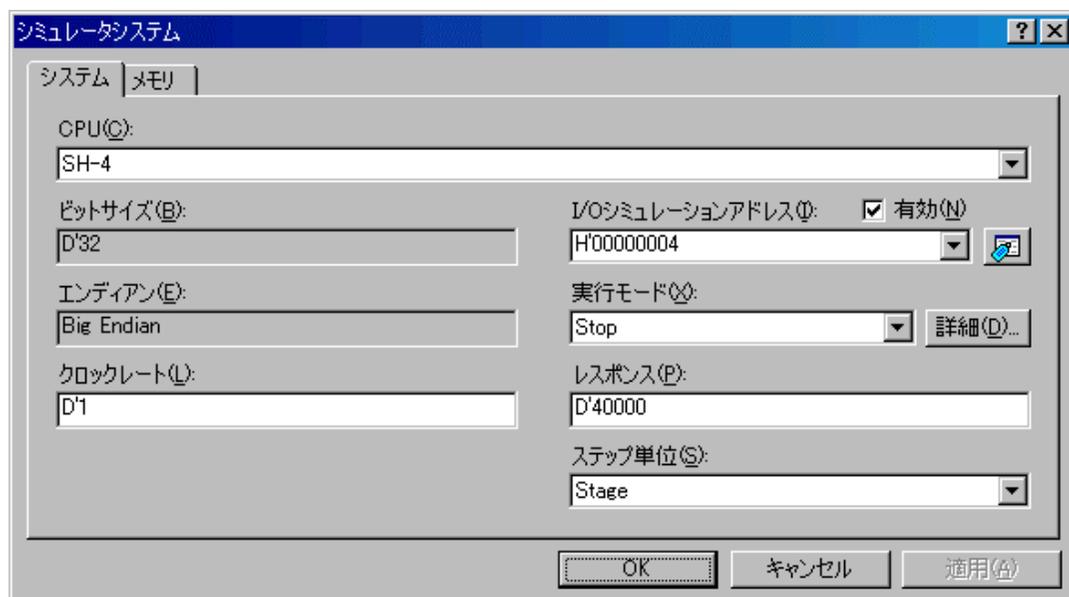


図 3-3 シミュレータシステムダイアログボックス(システムタブ)

本ダイアログボックスでは下記項目を設定または表示します。

[CPU]	現在設定している CPU。 (CPU は、[デバッグの設定]ダイアログボックスにより設定します。) SH2-DSP (SH7410) では、[Internal Mode]/[External Mode]を指定します。 SH2-DSP (SH7065) では、下記を指定します。 [ROM Disable/Fast Mode]      内蔵 ROM 無効/高速アクセスモード [ROM Disable/Slow Mode]      内蔵 ROM 無効/低速アクセスモード [ROM Enable/Fast Mode]      内蔵 ROM 有効/高速アクセスモード [ROM Enable/ Slow Mode]      内蔵 ROM 有効/低速アクセスモード SH-2DSP (Core) では、下記を指定します。 [ROM Disable]      内蔵 ROM 無効モード [ROM Enable]      内蔵 ROM 有効モード
[ビットサイズ]	アドレスバス幅。32 ビット固定です。
[エンディアン]	現在設定している エンディアン。
[クロックレート]	SH-4 シリーズのみ指定可能です。 外部クロックを 1 とした時の内部クロック比を指定します。(1~H'10)
[I/O シミュレーション アドレス]	ユーザプログラムから標準入出力またはファイル入出力を行うための I/O シミュレーションの開始位置を指定します。 [有効]      チェックすると I/O シミュレーションが有効となります。
[レスポンス]	何命令実行ごとにウィンドウをリフレッシュするかを指定します (1~D'2,147,483,647、デフォルトは D'40000)
[実行モード]	シミュレーションエラーが発生した場合のシミュレータ・デバッガ動作を規定します。 割り込みはシミュレーションエラーに含み、本設定に従います。 また、[詳細]ボタンで、割り込み発生時の動作を規定することもできます。 [停止]      シミュレーションを停止します。 [続行]      シミュレーションを続行します。
[ステップ単位]	SH-3、SH3-DSP シリーズ、SH-4 シリーズのみ指定可能 STEP 実行時および PC ブレーク成立時の命令実行状態を規定します。 [ステージ]      命令実行前に停止します パイプラインは乱れません [命令]      命令実行後に停止します パイプラインが乱れ、実行サイクル数も不正確になります

変更内容は、[OK]ボタンまたは[適用]ボタンをクリックすることにより設定します。[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。

### 3.2.2 メモリマップおよびメモリリソースの設定を変更する

メモリマップの設定および変更とメモリリソースの設定および変更は、[シミュレータシステム]ダイアログボックスの[メモリ]タブで行います。

### 3. デバッグ

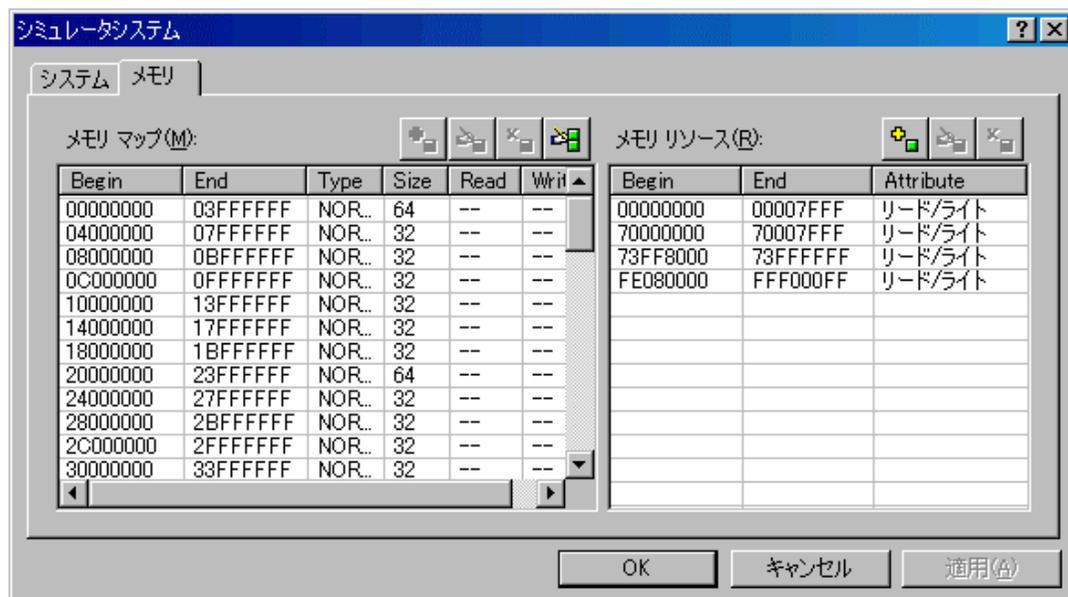


図 3-4 シミュレータシステムダイアログボックス(メモリタブ)

本ダイアログボックスでは以下の項目を設定します。

- [メモリマップ]                      メモリマップ情報として、メモリ種別とその先頭アドレス・終了アドレス、データバス幅、アクセスステート数を表示します。
- [メモリリソース]                    現在設定しているメモリリソースのアクセス種別とその先頭アドレス・終了アドレスを表示します。

[メモリマップ]は、以下の各ボタンにより追加・変更・削除ができます。



[メモリマップ]の項目を追加します。クリックすると、[メモリマップ設定]ダイアログボックス(3.2.3 参照)が開き、設定することができます。



[メモリマップ]の項目を変更します。変更したい項目をリストボックス上で選択後、ボタンをクリックします。クリックすると、[メモリマップ設定]ダイアログボックス(3.2.3 参照)が開き、変更することができます。



[メモリマップ]の項目を削除します。削除したい項目をリストボックス上で選択後、ボタンをクリックします。

[メモリリソース]は次の各ボタンにより追加・変更・削除することができます。



[メモリリソース]の項目を追加します。クリックすることにより、[メモリリソース設定]ダイアログボックスが開き、設定することができます。



[メモリリソース]の項目を変更します。変更したい項目をリストボックス上で選択後、ボタンをクリックします。クリックすることにより、[メモリリソース設定]ダイアログボックスが開き、変更することができます。



[メモリリソース]の項目を削除します。削除したい項目をリストボックス上で選択後、ボタンをクリックします。

[メモリリソース]は、[SuperH RISC engine Standard Toolchain]ダイアログボックスの[デバッグ]シートの[メモリリソース]と同一の設定情報です。おたがいの変更が反映されます。

最適化リンケージエディタが出力するリンケージリストファイル(.map)がある場合、メモリマップおよびリンケージマップ情報に基づきメモリリソースを自動的に確保することができます。

詳しくは「High-performance Embedded Workshop V.4.03 ユーザーズマニュアル」の「13.1.9 メモリリソースを自動的に確保する」を参照してください。

[メモリマップ]には、メモリ情報として、先頭アドレス・終了アドレス、メモリ種別、データバス幅、リードステート数、ライトステート数の順に表示します。表示するメモリ種別は次の通りです。

- SH-1/SH-2/SH-2E/SH2A-FPU /SH-3/SH-3E  
ROM (内蔵ROM)、RAM (内蔵RAM)、EXT (外部メモリ)、I/O (内蔵I/O)
- SH2-DSP(SH7410) / SH3-DSP  
XROM (内蔵XROM)、YROM (内蔵YROM)、XRAM (内蔵XRAM)、YRAM (内蔵YRAM)、EXT (外部メモリ)、I/O (内蔵I/O)
- SH2-DSP(SH7612)  
XRAM (内蔵XRAM)、YRAM (内蔵YRAM)、INTRAM (内蔵RAM)、EXT (外部メモリ)、I/O (内蔵I/O)
- SH2-DSP (SH7065)  
XRAM (内蔵XRAM)、YRAM (内蔵YRAM)、INTROM (内蔵ROM)、EXT (外部メモリ)、I/O (内蔵I/O)
- SH2-DSP(Core)  
XRAM (内蔵XRAM)、YRAM (内蔵YRAM)、INTROM (内蔵ROM)、INTRAM (内蔵RAM)、EXT (外部メモリ)、I/O (内蔵I/O)
- SH3-DSP(Core)  
XRAM (内蔵XRAM)、YRAM (内蔵YRAM)、URAM (ユーザRAM)、EXT (外部メモリ)、I/O (内蔵I/O)
- SH-4/SH-4 (SH7750R)  
NORMAL (通常メモリ)、INTRAM (内蔵RAM)、I/O (内蔵I/O)
- SH-4BSC  
NORMAL (通常メモリ)、MPX (MPX)、BCSRAM (バイト制御SRAM)、BSTROM (バーストROMおよびバースト回数)、DRAM (DRAM)、SDRAM (シンクロナスDRAM)、INTRAM (内蔵RAM)、I/O (内蔵I/O)
- SH-4A  
LRAM(内蔵RAM)、URAM(ユーザRAM)、EXT(外部メモリ)、I/O(内蔵I/O)
- SH4AL-DSP  
XRAM(内蔵XRAM)、YRAM(内蔵YRAM)、URAM(ユーザRAM)、EXT(外部メモリ)、I/O(内蔵I/O)

### 3. デバッグ

**【注】** (1) SH-4BSC では、メモリマップの表示・変更について次の点にご注意ください。

- アクセスステート数は "--"と表示します。
- 新たにメモリマップの設定あるいは削除することはできません。したがって、[メモリマップ]に対しては、[変更]ボタンのみが有効となります。
- 内蔵 RAM、内蔵 I/O に対して、メモリマップの変更ができません。内蔵 RAM の有効/無効の設定は CCR 制御レジスタの ORA ビットにより行います。
- メモリマップは、[メモリマップ変更]ダイアログボックスでの設定の他に、BSC 制御レジスタの設定を反映します。BSC の設定不正により、メモリマップの内容が決定できない場合、本ダイアログボックスでは、メモリ種別を "?????"、バス幅を "??"と表示します。

変更内容は、[OK]ボタンをクリックすることにより設定します。[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。

なお、[メモリマップ]・[メモリアリソース]は、ボタンによりデフォルト値にリセットすることができます。

変更内容は、[OK]ボタンまたは[適用]ボタンをクリックすることにより設定します。[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。

#### 3.2.3 メモリマップ設定ダイアログボックス

[メモリマップ設定]ダイアログボックスでは、対象 CPU のメモリマップを設定します。

各項目に表示する内容は、対象 CPU によって異なります。シミュレータ・デバッグはこれらの値をメモリアクセスサイクル数の算出に使用します。



図 3-5 メモリマップ設定ダイアログボックス

本ダイアログボックスでは以下の項目を設定します。

[メモリ種別]	メモリ種別
[開始アドレス]	メモリ種別に対応するメモリの先頭アドレス
[終了アドレス]	メモリ種別に対応するメモリの終了アドレス
[データバスサイズ]	メモリのデータバス幅
[リードステート数]	メモリのリードアクセスステート数
[ライトステート数]	メモリのライトアクセスステート数

**【注】** SH-4 シリーズでは、[開始アドレス]、[終了アドレス]の変更はできません。

[メモリ種別]、[データバスサイズ]のみ指定してください。

また、SH-4 シリーズでは、[ステート設定...]ボタンをクリックすると[ステート設定]ダイアログボックス(3.2.4 参照)が開き、エリア 0 ~ 7 に対する挿入ウェイトサイクル数を設定することができます。この値は、制御レジスタ WCR1、WCR2 の値に対応します。

**【注】** SH-4A シリーズ、SH2A-FPU のメモリマップについて次の点にご注意ください。

- [メモリ種別]を変更することはできません。
- 外部メモリ以外のメモリ種別では、[開始アドレス]、[終了アドレス]の変更はできません。
- SH-4A シリーズでは全メモリ種別で[ライトステート数]の変更はできません。
- 命令単位シミュレータを使用した場合は、1 命令 1 サイクルで動作するため、[リードステート数]の設定は無効となります。

**【注】** メモリリソースを確保している領域のメモリマップは、削除・変更することができません。あらかじめ、[シミュレータシステム]ダイアログボックスの[メモリ]タブによりメモリリソースを削除してから、メモリマップを削除・変更してください。

変更内容は、[OK]ボタンをクリックすることにより設定します。[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。

### 3.2.4 ステート設定ダイアログボックス

[ステート設定]ダイアログボックスは、各エリアに対する挿入ウェイトサイクル数を設定します。なお、本ダイアログボックスは、SH-4 シリーズのみサポートします。

ただし、SH-4/SH-4 (SH7750R) では通常メモリのみサポートします。

[ステート設定]ダイアログボックスは、エリアに接続しているメモリ種別によって異なります。本ダイアログボックスで設定した値は制御レジスタ WCR1、WCR2 に反映します。

### 3. デバッグ

---

#### (1) 通常メモリ、バースト ROM、バイト制御 SRAM

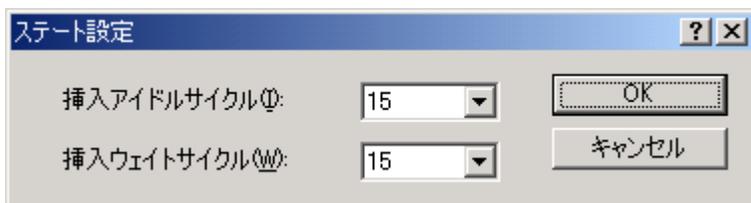


図 3-6 ステート設定ダイアログボックス (通常メモリ)

- [挿入アイドルサイクル] リードアクセスからライトアクセス、または別エリアに切り替わる時に挿入するサイクル数  
(WCR1 レジスタの AnIW に対応)
- [挿入ウェイトサイクル] あらゆるアクセスに挿入するサイクル数  
(WCR2 レジスタの AnW に対応)

#### (2) DRAM

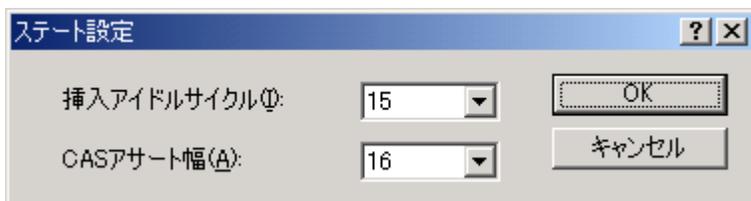


図 3-7 ステート設定ダイアログボックス (DRAM)

- [挿入アイドルサイクル] リードアクセスからライトアクセス、または別エリアに切り替わる時に挿入するサイクル数  
(WCR1 レジスタの AnIW に対応)
- [CAS アサート幅] CAS アサート幅  
(WCR2 レジスタの AnW に対応)

#### (3) SDRAM

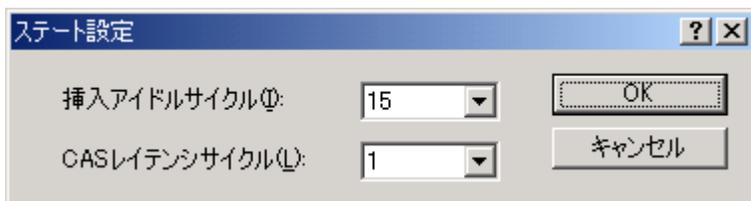


図 3-8 ステート設定ダイアログボックス (SDRAM)

- [挿入アイドルサイクル] リードアクセスからライトアクセス、または別エリアに切り替わる時に挿入するサイクル数  
(WCR1 レジスタの AnIW に対応)
- [CAS レイテンシサイクル] CAS レイテンシサイクル数  
(WCR2 レジスタの AnW に対応)

## (4) MPX

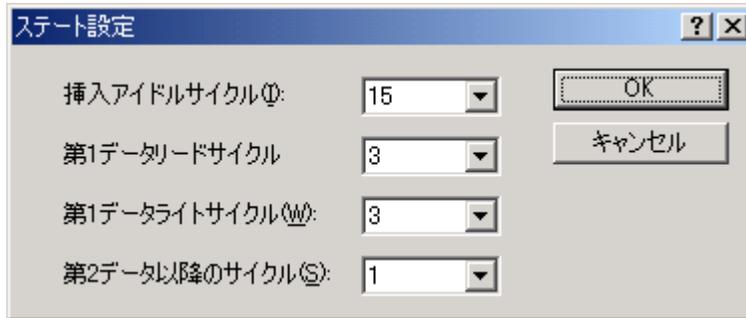


図 3-9 ステート設定ダイアログボックス (MPX)

- |                 |   |
|-----------------|---|
| [挿入アイドルサイクル]    | リードアクセスからライトアクセス、または別エリアに切り替わるときに挿入するサイクル数<br>(WCR1 レジスタの AnIW に対応) |
| [第1データリードサイクル]* | リードアクセスで第1サイクルに挿入するサイクル数  |
| [第1データライトサイクル]* | ライトアクセスで第1サイクルに挿入するサイクル数  |
| [第2データ以降のサイクル]* | バースト転送で第2サイクル以降に挿入するサイクル数   |
- 【注】 \*は、WCR2 レジスタの AnW に対応します。

## 3.2.5 メモリリソース設定ダイアログボックス

[メモリリソース設定]ダイアログボックスでは、メモリリソースの設定・変更を行います。

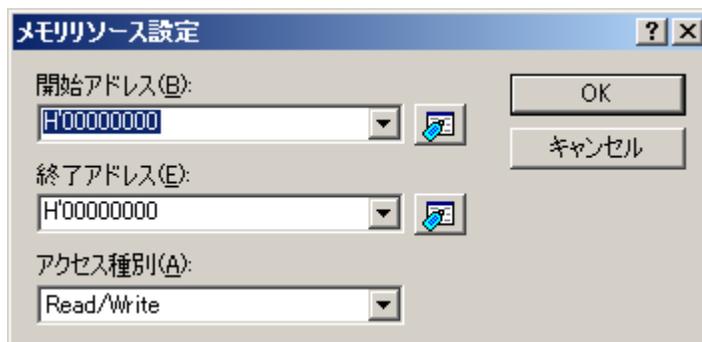


図 3-10 メモリリソース設定ダイアログボックス

本ダイアログボックスでは以下の項目を設定します。

- |              |                  |
|--------------|------------------|
| [開始アドレス]     | 確保するメモリ領域の先頭アドレス |
| [終了アドレス]     | 確保するメモリ領域の終了アドレス |
| [アクセス種別]     | アクセス種別           |
| [Read]       | 読み出しのみ可能         |
| [Write]      | 書き込みのみ可能         |
| [Read/Write] | 読み書き可能           |

### 3. デバッグ

各項目を指定後、[OK]ボタンをクリックすることによりメモリリソースの設定、変更を行います。  
[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。

【注】1.メモリリソースの設定すると、PCのメモリを使用します。したがって、メモリリソースを大きく取りすぎると、PCの動作が極端に遅くなる場合があります。

2.SH2A-FPU、SH-4A シリーズのメモリリソースについて次の点にご注意ください。

- メモリリソースは、64kB 境界でのみ設定可能です。64kB 境界以外での設定は、設定したメモリリソースを含む 64kB 境界に補正されます。また、アクセス種別に関しても 64kB 境界となります。なお、XRAM、YRAM に対しても 64kB 境界に設定されますので、64kB 以下で使用する場合は、ハードウェアマニュアルに沿った範囲内のメモリを使用してください。
- I/O 空間にデフォルトで確保されているメモリリソースは削除しないでください。削除すると、MMU、キャッシュ等の動作が不正になります。

## 3.3 周辺機能シミュレーションを設定する

シミュレータ・デバッグは周辺機能シミュレーションを DLL 形式のモジュールで実現しています。

ここでは周辺機能シミュレーションを有効にするための周辺機能シミュレーションモジュールの登録方法および周辺機能シミュレーション構成の設定方法を説明します。

### 3.3.1 周辺機能シミュレーションモジュールを登録する

周辺機能シミュレーションモジュールの登録は[周辺機能シミュレーションの設定]ダイアログボックスで行います。[周辺機能シミュレーションの設定]ダイアログボックスはシミュレータ・デバッグ起動時に表示されます。

本ダイアログボックスで周辺機能シミュレーションモジュールを登録すると、当該周辺機能シミュレーションモジュールが提供する周辺機能シミュレーションを利用可能となります。シミュレータ・デバッグ起動後は、周辺機能シミュレーションモジュール登録内容を変更することはできません。利用する周辺機能シミュレーションモジュールを変更する場合はシミュレータ・デバッグを再起動し、本ダイアログボックスを表示させてください。



図 3-11 周辺機能シミュレーションの設定ダイアログボックス

本ダイアログボックスでは以下の項目を設定します。

- [周辺機能] 周辺機能シミュレーションモジュールの情報を表示します。
- [Module Name] シミュレーションする周辺機能名
- [File Name] 周辺機能シミュレーションモジュールファイル名
- [Module name]欄のチェックボックスをチェックした周辺機能シミュレーションモジュールが登録されて、利用可能となります。
- [すべて登録] すべての周辺機能を有効にします。
- [すべて無効] すべての周辺機能を無効にします。
- [詳細...] 周辺機能情報の表示、周辺機能の開始アドレス、および割り込み要因情報の変更を行うための[周辺モジュールの構成]ダイアログボックスを表示します。
- [周辺クロックレート] 周辺クロックと内部クロックの比(周辺1クロックが内部クロックいくつに相当するか)を指定します。  
(1,2,3,4,6,8,12,16,24,32から選択)

各項目を指定後、[OK]ボタンをクリックすることにより周辺機能シミュレーションの設定、変更を行います。[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。

[このダイアログを表示しない]チェックボタンをチェックすると、次回以降シミュレータ・デバッグ起動時に本ダイアログボックスを表示しなくなります。

### 3.3.2 周辺機能のアドレスを変更する

周辺機能のアドレス変更は[周辺モジュールの構成]ダイアログボックス[アドレス]タブで行います。

[周辺モジュールの構成]ダイアログボックスを開くには、[周辺機能シミュレーションの設定]ダイアログボックスの[周辺機能]欄で周辺機能を選択して、[詳細...]ボタンをクリックします。

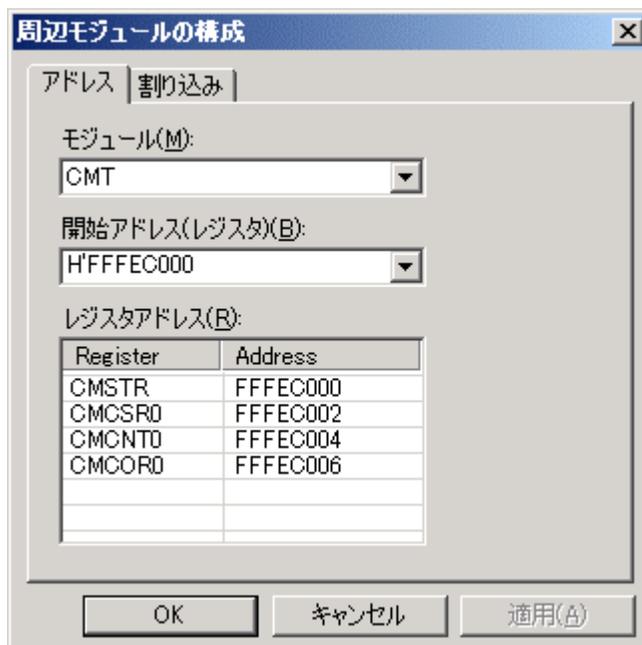


図 3-12 周辺モジュールの構成ダイアログボックス (アドレスタブ)

### 3. デバッグ

本ダイアログボックスでは以下の項目を表示、設定します。

- [モジュール] 選択した周辺機能シミュレーションモジュールでサポートしている周辺機能名
- [開始アドレス] [モジュール]で選択した周辺機能の開始アドレス
- [レジスタアドレス] [モジュール]で指定した周辺機能のレジスタ名、レジスタアドレスを表示します。個々のレジスタアドレスは変更できません。

各項目を指定後、[OK]ボタンをクリックすることにより周辺機能のアドレスを設定します。[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。

#### 3.3.3 周辺機能の割り込み要因情報を変更する

周辺機能の割り込み要因情報は[周辺モジュールの構成]ダイアログボックス[割り込み]タブで参照できます。

[周辺モジュールの構成]ダイアログボックスを開くには、[周辺機能シミュレーションの設定]ダイアログボックスの[周辺機能]欄で周辺機能を選択して、[詳細...]ボタンをクリックします。



図 3-13 周辺モジュールの構成ダイアログボックス (割り込みタブ)

本ダイアログボックスでは以下の項目を表示します。

- 割り込み要因情報
- |  |                                     |
|--|-------------------------------------|
| Interrupt Source                             | 周辺機能でサポートしている割り込み要因名                |
| INTEVT                                       | INTEVT 値： SH-3、SH-4、SH-4A シリーズ      |
| INTEVT/INTEVT2                               | INTEVT 値、および NTEVT2 値： SH3-DSP シリーズ |
| Vector Number                                | 割り込みベクタ番号： その他の CPU                 |
| Priority Register Address/Bit Field Position | 割り込み優先順位レジスタアドレスとレジスタ内ビット位置         |

割り込み要因情報を変更する場合は、変更したい割り込み要因をダブルクリックします。[割り込み要因情報の設定]ダイアログボックスが表示されます(SH-4 シリーズを除く)。

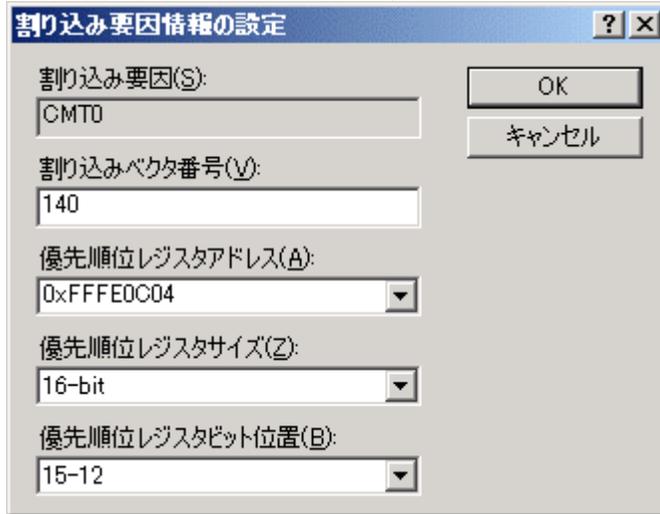


図 3-14 割り込み要因情報の設定ダイアログボックス

本ダイアログボックスでは以下の項目を表示、設定します。

割り込み要因	割り込み要因名
INTEVT	INTEVT 値 : SH-3、SH-4A、SH3-DSP シリーズ (接頭辞省略時は 16 進入力、16 進表示)
INTEVT2	INTEVT2 値 : SH3-DSP シリーズのみ (接頭辞省略時は 16 進入力、16 進表示)
割り込みベクタ番号	割り込みベクタ番号 : その他の CPU (接頭辞省略時は 10 進入力、10 進表示)
優先順位レジスタアドレス	割り込み優先順位レジスタのアドレス
優先順位レジスタサイズ	割り込み優先順位レジスタのサイズ
優先順位レジスタビット位置	割り込み優先順位レジスタ内のビット位置

各項目を指定後、[OK]ボタンをクリックすることにより割り込み要因情報を設定します。[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。

### 3.3.4 バンク制御レジスタのアドレスを変更する(SH2A-FPU のみ)

SH2A-FPU シリーズのバンクコントロールレジスタ(BCR)、およびバンク番号レジスタ(IBNR)アドレスの表示/変更は下記コマンドで行います。

REGISTER\_ADDRESS\_GET コマンド: バンク制御レジスタアドレスの表示  
REGISTER\_ADDRESS\_SET コマンド: バンク制御レジスタアドレスの設定

コマンドの詳細はシミュレータヘルプを参照してください。

### 3.3.5 割り込み優先順位レジスタのメモリリソースを確保する

周辺機能シミュレーションモジュールは割り込み優先順位レジスタ(IPR)メモリ領域を確保させ

### 3. デバッグ

---

ん(SH-4 シリーズを除く)。割り込み優先順位レジスタ(IPR)領域は、ユーザ操作でメモリリソースを確保してください。メモリリソース設定の詳細は、「3.2.2 メモリマップおよびメモリリソースの設定を変更する」を参照してください。

#### 3.3.6 接続されている周辺機能を確認する

シミュレータ・デバッガ起動後は、[ステータス]ウィンドウの[Platform]シートの[Peripheral Modules]項目に接続されている周辺機能名を表示します。

### 3.4 シミュレータ・デバッガのブレイクポイントを使用する

シミュレータ・デバッガでは HEW 標準の PC ブレイクポイントとは別により高度なブレイクポイント機能を持っています。

これらブレイクポイントについて、ブレイク条件の設定、ブレイク条件成立時の動作、および設定されているブレイクポイントの表示が行えます。

#### 3.4.1 ブレイクポイントを一覧表示する

現在設定されているブレイクポイントを一覧表示するには[イベントポイント]ウィンドウを開きます。

[イベントポイント]ウィンドウは[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。

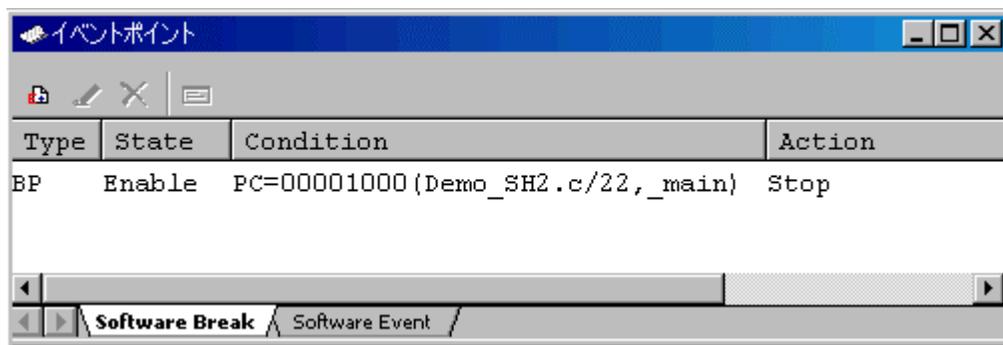


図 3-15 イベントポイントウィンドウ

表示する項目は以下の通りです。

[Type]	ブレイク種別を表示します。
	[BP] : PC ブレイク
	[BA] : ブレイクアクセス
	[BD] : ブレイクデータ
	[BR] : ブレイクレジスタ
	[BS] : ブレイクシーケンス
	[BCY] : ブレイクサイクル
[State]	該当ブレイクポイントの有効/無効を示します。
	[Enable] : 有効
	[Disable] : 無効
[Condition]	Break が成立する条件を表示します。表示内容はブレイク種別により異なります。 ブレイク種別が BR の時はレジスタ名を、BCY の時はサイクル数を表示します。
	BP 時 : PC=プログラムカウンタ (対応するファイル名 / 行、シンボル名)
	BA 時 : Address=アドレス (シンボル名)
	BD 時 : Address=アドレス (シンボル名)
	BR 時 : Register=レジスタ名
	BS 時 : PC=プログラムカウンタ (対応するファイル名 / 行、シンボル名)
	BCY 時 : Cycle=サイクル数 (16 進表示)
[Action]	ブレイク条件成立時の動作を表示します。
	[Stop] : 実行停止
	[File Input] (ファイル名) [ファイルの状態] : ファイルからのメモリデータ読みこみ
	[File Output] (ファイル名) [ファイルの状態] : ファイルへメモリデータ書きこみ
	[Interrupt] (割り込み種別 / 優先順位) : 割り込み処理
	SH3-DSP のみ (割り込み種別 1、 割り込み種別 2 / 優先順位) と表示

[Action]の設定で、[Stop]を指定した条件を[Software Break]タブに、[Stop]以外を指定した条件を[Software Event]タブに表示します。

### 3.4.2 ブレイクポイントを設定する

[イベントポイント]ウィンドウのポップアップメニューで[設定...]を選択すると、[ブレイク種別の選択]ダイアログボックスが開きブレイクポイントを設定できます。

[ブレイク種別の選択]ダイアログボックスからブレイク条件を設定する[条件の設定]ダイアログボックスとブレイク成立時の動作を設定する[動作の設定]ダイアログボックスが開きます。[動作種別]として、[停止]を選択する場合は[Software Break]タブで、[停止]以外を選択する場合は[Software Event]タブでポップアップメニューを選択します。

### 3. デバッグ

#### (1) ブレーク種別を選択する

[イベントポイント]ウィンドウのポップアップメニューで[設定]を選択すると、[ブレーク種別の選択]ダイアログボックスが開きます。

ブレーク種別は[ブレーク種別の選択]ダイアログボックスの[ブレーク種別]欄で選択します。

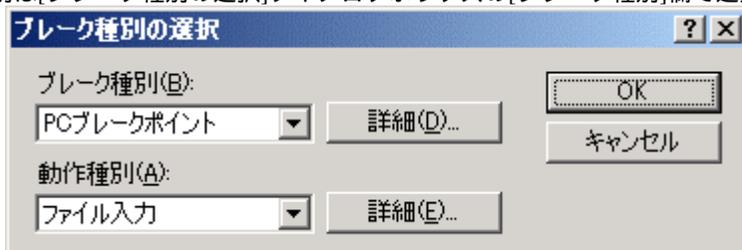


図 3-16 ブレーク種別の選択ダイアログボックス

選択内容を以下に示します。

ブレーク種別	内容
[PC ブレークポイント]	実行命令位置によるブレークポイント
[ブレークアクセス]	メモリ範囲のアクセスによるブレーク
[ブレークデータ]	メモリのデータ値によるブレーク
[ブレークレジスタ]	レジスタのデータ値によるブレーク
[ブレークシーケンス]	実行順序を指定したブレークポイント
[ブレークサイクル]	サイクル数によるブレーク

#### (2) ブレーク条件を設定する

[ブレーク種別の選択]ダイアログボックスでブレーク種別を選択後[詳細]をクリックすると、各ブレーク種別の条件を設定するダイアログボックスを表示します。

##### (a) PC ブレークポイント



図 3-17 PC ブレークポイント条件の設定ダイアログボックス

PC ブレークポイント条件を設定します。

[PC ブレークポイント]	1,024 個まで指定可能
[アドレス]	ブレークする命令の位置
[回数]	指定位置の命令をフェッチする回数 (接頭辞省略時は 10 進入力、10 進表示) (1~16,383、省略すると 1 となります)

## (b) ブレークアクセス

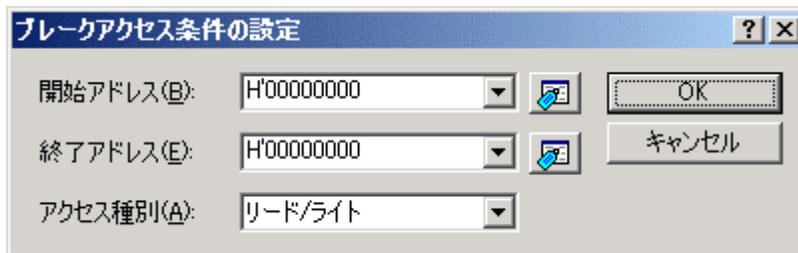


図 3-18 ブレークアクセス条件の設定ダイアログボックス

ブレークアクセス条件を設定します。

[ブレークアクセス]	1,024 個まで指定可能
[開始アドレス]	アクセスするとブレークするメモリの開始位置
[終了アドレス]	アクセスするとブレークするメモリの終了位置 (省略すると開始位置のみが範囲となります)
[アクセス種別]	アクセス種別

## (c) ブレークレジスタ

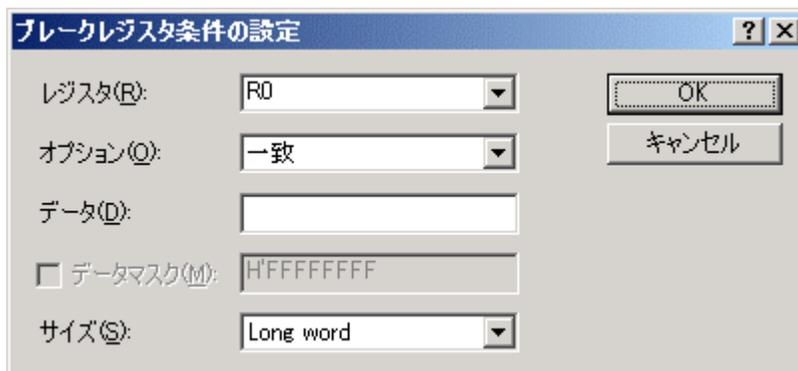


図 3-19 ブレークレジスタ条件の設定ダイアログボックス

ブレークレジスタ条件を設定します。

[ブレークレジスタ]	1,024 個まで指定可能
[レジスタ]	ブレーク条件を設定するレジスタ名
[オプション]	データの一致/不一致
[データ]	ブレーク条件となるデータ値 (省略するとレジスタへ書き込みたびにブレークします)
[データマスク]	マスク条件(0 を指定したビットがマスクされます)
[サイズ]	データのサイズ

### 3. デバッグ

#### (d) ブレークシーケンス

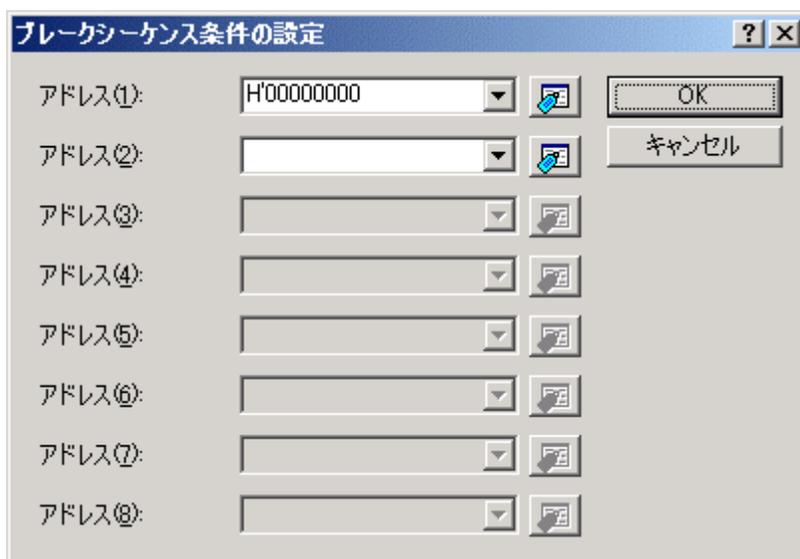


図 3-20 ブレークシーケンス条件の設定ダイアログボックス

ブレークシーケンス条件を設定します。

[ブレークシーケンス] 1組のみ指定可能

[アドレス(1)]~[アドレス(8)]

ブレークの発生条件となる通過アドレス

(8ポイントすべてを設定する必要はありません)

#### (e) ブレークサイクル



図 3-21 ブレークサイクル条件の設定ダイアログボックス

ブレークサイクル条件を設定します。

[ブレークサイクル]	1,024 個まで指定可能 [サイクル]	ブレーク判定を行うサイクル数(1~H'FFFFFFFF) [サイクル]×n のサイクルで条件が一致します ただし、指定したサイクルと実際に条件が一致する サイクルはずれることがあります
[回数]	ブレークが成立する回数 [すべて]	条件が一致するごとにブレークが成立します
	[回数指定]	条件一致した回数が[回数指定]以下の時だけブ レークが成立します (接頭辞省略時は 16 進入力、16 進表示)(1~65,535)

## (f) ブレークデータ

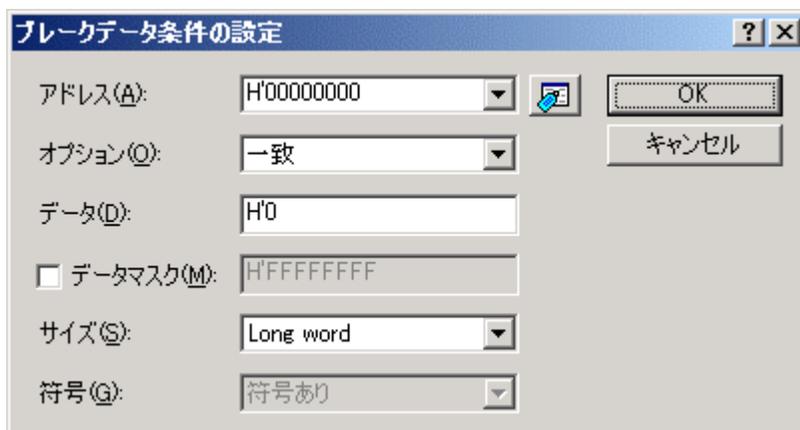


図 3-22 ブレークデータ条件の設定ダイアログボックス

ブレークデータ条件を設定します。

[ブレークデータ]	1,024 個まで指定可能
[アドレス]	ブレーク判定を行うメモリの位置
[オプション]	データの比較方法(一致/不一致/符号反転/差分)
	一致      メモリ書き込み値と指定値([データ])が一致で成 立
	不一致    メモリ書き込み値と指定値([データ])が不一致で 成立
	符号反転 <sup>*1</sup> 前回メモリ書き込み値と今回メモリ書き込み値 で符号が反転した場合に成立
	差分 <sup>*1</sup> 前回メモリ書き込み値と今回メモリ書き込み値 の差分が指定値([データ])を越えた場合に成立
[データ]	ブレーク条件となるデータ値
[データマスク]	マスク条件(0 を指定したビットがマスクされます) データの一致/不一致オプション時のみ有効です
[サイズ]	データのサイズ
[符号]	データの符号 データの差分オプション時のみ有効です

\*1 : [符号反転]、および[差分]は前回書き込み値と比較するため、リセット後、およびブレーク成立  
後 1 回目の判定は常に不成立となります。

### 3. デバッグ

#### (g) 留意事項

SH3-DSP シリーズにおいて、MOVX、MOVY 命令の X/Y メモリアクセスに対する [ブレイクアクセス] の [開始アドレス]、[終了アドレス] および [ブレイクデータ] の [アドレス] には、H'A5000000 ~ H'A501FFFF (X/Y メモリの論理アドレス。物理アドレスの H'05000000 ~ H'0501FFFF に該当。) を設定してください。

#### (3) 動作種別を選択する

各ブレイク条件設定ダイアログボックスで条件設定後[OK]をクリックすると、再度[ブレイク種別の選択]ダイアログボックスが開きます。

[ブレイク種別の選択]ダイアログボックスの[動作種別]欄で動作種別を選択します。

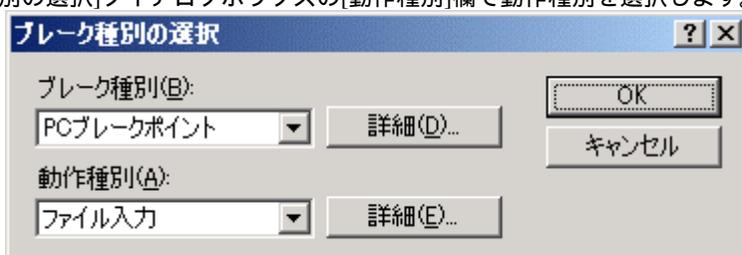


図 3-23 ブレイク種別の選択ダイアログボックス

選択内容を以下に示します。

動作種別	内容
[停止]	条件成立時にユーザプログラムの実行を停止します
[ファイル入力]	条件成立時に指定ファイルから読み込んだデータを指定メモリへ書き込みます
[ファイル出力]	条件成立時に指定メモリの内容を指定ファイルへ書き込みます
[割り込み]	条件成立時に割り込み処理を行います

#### (4) 動作内容を設定する

[ブレイク種別の選択]ダイアログボックスで動作種別を選択後[詳細]をクリックすると[停止]を除く各動作種別の内容を設定するダイアログボックスを表示します。

#### (a) ファイル入力

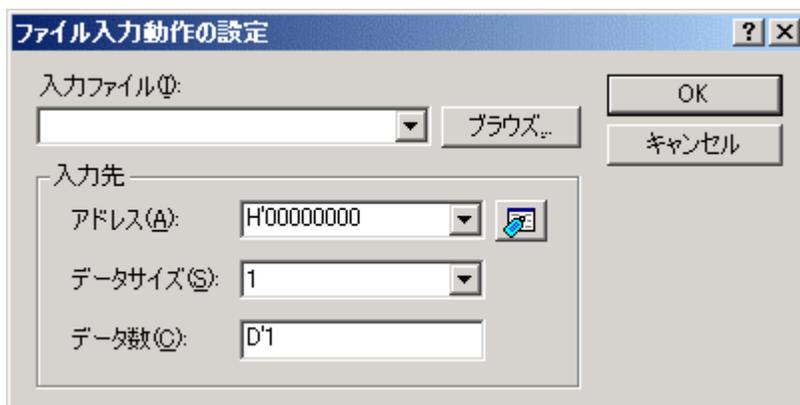


図 3-24 ファイル入力動作の設定ダイアログボックス

ファイル入力動作内容を設定します。

- [ファイル入力] 条件成立時に指定ファイルから読み込んだデータを指定メモリへ書き込みます
- [入力ファイル] 読み込むデータファイルを指定します  
ファイルの終端まで読み込んだら先頭から繰り返して読み込みます
- [アドレス] データを書き込むメモリのアドレスを指定します
- [データサイズ] 読み込むデータ 1 個のサイズ(バイト数)を指定します(1/2/4/8)
- [データ数] 読み込むデータの個数を指定します  
(接頭辞省略時は 10 進入力、10 進表示)(1~H'FFFFFFFF)

(b) ファイル出力

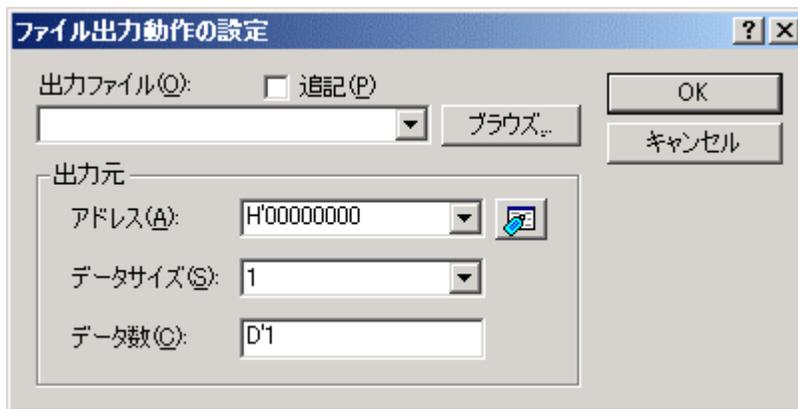


図 3-25 ファイル出力動作の設定ダイアログボックス

ファイル出力動作内容を設定します。

- [ファイル出力] 条件成立時に指定メモリの内容を指定ファイルへ書き込みます
- [出力ファイル] 書き込むデータファイルを指定します
- [追記] 既存のファイルを[出力ファイル]で指定した場合にファイルの最後に追加出力するかを指定します
- [アドレス] データを読み出すメモリのアドレスを指定します
- [データサイズ] 書き込むデータ 1 個のサイズ(バイト数)を指定します (1/2/4/8)
- [データ数] 書き込むデータの個数を指定します  
(接頭辞省略時は 10 進入力、10 進表示)(1~H'FFFFFFFF)

(c) 割り込み

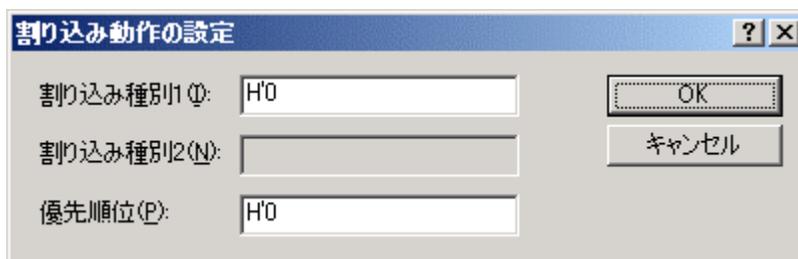


図 3-26 割り込み動作の設定ダイアログボックス

### 3. デバッグ

---

割り込み動作内容を設定します。

- [割り込み] 条件成立時に割り込み処理を行います。詳細は「2.20 擬似割り込み」を参照してください。
- [割り込み種別 1] CPU ごとに下記を指定します。(接頭辞省略時は 16 進入力、16 進表示)
- SH-1、SH-2、SH2-DSP、SH2A-FPU シリーズ  
割り込みベクタ番号
  - SH-3、SH-4、SH3-DSP シリーズ  
INTEVT 値 (0~H'FFF)
  - SH-4A シリーズ  
INTEVT 値 (0~H'3FFF)
- [割り込み種別 2] SH3-DSP シリーズのみ下記を指定可能です。(接頭辞省略時は 16 進入力、16 進表示)
- INTEVT2 値 (0~H'FFF)
- [優先順位] 割り込み優先順位を指定します (接頭辞省略時は 16 進入力、16 進表示) (0~H'11)
- H'10 指定時は SR の I ビットによらず割り込みが発生しますが、SR の BL ビットによってマスクされます。
- H'11 指定時は SR の I ビット、および BL ビットによらず割り込みが発生します。

#### (d) 留意事項

複数の[ファイル入力]で同一ファイルを指定した場合、ブレイク成立順にファイルからデータを読み込みます。複数の[ファイル出力]で同一ファイルを指定した場合、ブレイク成立順にファイルヘータを書き込みます。ただし、[ファイル入力]と[ファイル出力]で同一ファイルを指定した場合は、最初に成立した動作のみが有効となります。

### 3.4.3 ブレークポイントの設定内容を変更する

変更したいブレーク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、[ブレーク種別の選択]ダイアログボックスが開き、ブレーク条件を変更することができます。[編集...]メニューはブレークポイントを1個選択しているときのみ有効となります。

### 3.4.4 ブレークポイントを有効にする

ブレーク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、選択しているブレークポイントを有効にします。

### 3.4.5 ブレークポイントを無効にする

ブレーク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、選択しているブレークポイントを無効にします。無効にした場合は、ブレーク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ブレークは成立しません。

### 3.4.6 ブレークポイントを削除する

ブレーク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、選択しているブレークポイントを削除します。ブレークポイントを削除しないで、詳細情報は保持したまま、条件が一致してもブレークを成立させないようにするには、[無効]オプションを使用します(「3.4.5 ブレークポイントを無効にする」参照)。

### 3.4.7 ブレーク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、すべてのブレークポイントを削除します。

### 3.4.8 ブレークポイントのソース行を表示する

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

### 3.4.9 入出力ファイルを閉じる

ブレークポイントを選択後ポップアップメニューから[ファイルを閉じる]を選択すると、選択した[ファイル入力]または[ファイル出力]のデータファイルを閉じ、ファイル読み出し位置をリセットします。

### 3.4.10 入出力ファイルをすべて閉じる

ポップアップメニューから[すべてのファイルを閉じる]を選択すると、すべての[ファイル入力]および[ファイル出力]のデータファイルを閉じ、ファイル読み出し位置をリセットします。

## 3.5 トレース情報を見る

シミュレータ・デバッガでは命令の実行結果をトレース情報として取得および表示することができます。

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

### 3.5.1 トレースウィンドウを開く

[トレース]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。

### 3.5.2 トレース情報取得条件を設定する

[トレース]ウィンドウを開いたら、トレース情報取得条件を設定します。

トレース情報取得条件は、[トレース取得]ダイアログボックスで設定します。

[トレース取得]ダイアログボックスを開くには、ポップアップメニューから[設定...]を選択します。

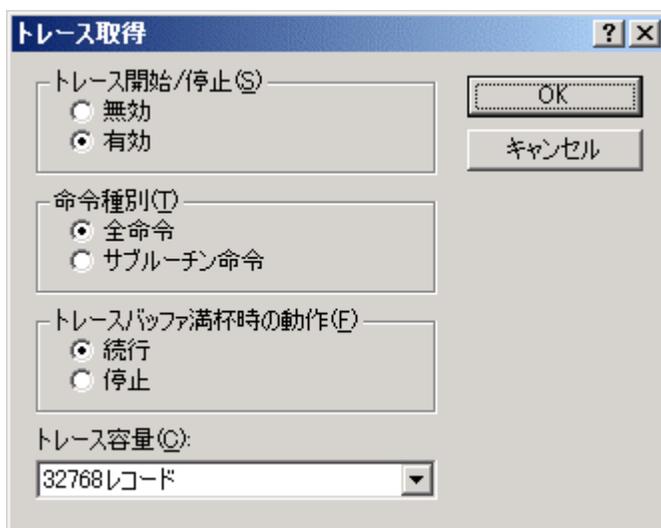


図 3-27 トレース取得ダイアログボックス

本ダイアログボックスでは、トレース情報の取得条件を設定します。

[トレース開始/停止]

[無効]                   トレース情報の取得停止

[有効]                   トレース情報の取得開始

[命令種別]

[全命令]                全命令のトレース情報を取得

[サブルーチン命令]    サブルーチン命令のみトレース情報を取得

[トレースバッファ満杯時の動作]

[続行]                 トレース情報取得バッファが満杯になっても取得を続行

[停止]                 トレース情報取得バッファが満杯になった場合、実行停止

また、トレースバッファの大きさは、[トレース容量]で 1K(1,024)、4K(4,096)、16K(16,384)、32K(32,768)、64K(65,536)、128K(131,072)、または 256K(262,144)レコードの中から選択します。

指定した内容は、[OK]ボタンをクリックすることにより設定します。[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。

### 3.5.3     トレース情報を取得する

トレース情報の取得を開始した状態で命令を実行すると、トレースを取得できます。

取得したトレース情報は[トレース]ウィンドウに表示します。表示内容は CPU によって異なります。お使いになる CPU に合わせてお読みください。

### 3. デバッグ

#### (1) SH-1/SH-2/SH-2E/SH2-DSP シリーズ

PTR	Cycle	Address	Pipeline	Instruction	Access Data	Source
-02260	0000247637	00003A24	FF<<-D>EMW	MOV.B @(R0, R14), R4	R4<-01	
-02259	0000247639	00001314	FFD<E>MM	MOV.L R13, @-R15	07FFFE34<-07FFFEA6	int write(...)
-02258	0000247641	00001316	f<D>E>MM>>	MOV.L R14, @-R15	07FFFE30<-0600001C	
-02257	0000247645	00001318	FFD><<E>MM	STS.L PR, @-R15	07FFFE2C<-00003A26	
-02256	0000247647	0000131A	f><<D>E>	ADD #F4, R15	R15<-07FFFE20	
-02255	0000247649	0000131C	FFD>E>MM	MOV.L R6, @(00000000, R15)	07FFFE28<-00000001	if (f1mod[f...
-02254	0000247651	0000131E	f>D>E>	MOV R5, R14	R14<-06000199	
-02253	0000247653	00001320	FFD>E>MMW	MOV.L @(00000158, PC), R5	R5<-060001AC	
-02252	0000247655	00001322	f>D>E>	MOV R4, R0	R0<-00000001	
-02251	0000247657	00001324	FFD>E>MMW	MOV.B @(R0, R5), R2	R2<-1A	
-02250	0000247660	00001326	f>D><E	EXTS.B R2, R0	R0<-0000001A	
-02249	0000247661	00001328	FF<DE>	TST #02, R0	T<- (0)	
-02248	0000247663	0000132A	fD>E>	BF 00001338	T(0), Pc<-00001338	
-02247	0000247665	0000132C	FFD			
-02246	0000247665	0000132E	f			
-02245	0000247667	00001338	FPDE>	MOV R6, R13	R13<-00000001	for(i=count...

図 3-28 トレースウィンドウ (SH-1/SH-2/SH-2E/SH2-DSP シリーズ)

表示する項目は以下の通りです。

[PTR] トレースバッファ内ポインタ (最後に実行した命令が 0 となります)

[Cycle] 累計命令実行サイクル数  
(パイプラインリセットによりクリアします)

[Address] 命令アドレス

[Pipeline] パイプラインの実行状況  
各記号の意味は以下の通りです。

- F : 命令フェッチ (メモリアクセスあり)
- f : 命令フェッチ (メモリアクセスなし)
- D : 命令デコード
- E : 命令実行
- M : メモリアクセス
- W : ライトバック
- P : DSP
- m : 乗算器実行
- : 命令固有のストール
- > : スプリット
- < : 競合によるストール

パイプライン動作の詳細については、各デバイスのプログラミングマニュアルを参照してください。

[Instruction] 命令二モニック

[Access Data] データアクセス (転送先<-転送データの形式で表示)

[Source] C/C++またはアセンブラソース

[Label] ラベル

## (2) SH-3/SH-3E シリーズ

PTR	Cycle	Address	Data Bus	Code	No	Instruction	IF	DE	EX	Access Data	Source	Label
-01308	0000265879	0000213E	52F962F2	0009	--	----	D0	--	----	(CC):R4<-...	void cha... _c.	
-01307	0000265881	00002140	00092F96	2F96	D0	MOV.L R9,@-	D1	D0	CE...	(CD):PR<-...		
-01306	0000265883	00002142	7FD4E200	7FD4	D1	ADD #D4,R15...	[D2]	D1	D0...		for (i=0;...	
-01305	0000265884	7FFFFFFAC	7FD4E200	E200	D2	MOV #000000...	--	D2	D1...	(D0):7FFF...		
-01304	0000265887	00002144	00000000	E200	--	----	D4	--	D2...	(D0):R15<...		
-01303	0000265889	00002148	E10A3213	E10A	D4	MOV #000000...	[D5]	D4	--...	(D1):R15<...		
-01302	0000265890	00002148	E10A3213	3213	D5	CMP/GE R1,R...	D6	D5	D4...	(D2):R2<-...		
-01301	0000265892	0000214A	890E6523	890E	D6	BT 00002168...	[D7]	D6	D5...		tmp[i] =...	
-01300	0000265893	0000214C	890E6523	6523	D7	MOV R2,R5 ...	D8	D7	D6...	(D4):R1<-...		
-01299	0000265895	0000214E	66534608	6653	D8	MOV R5,R6 ...	[D9]	D8	D7...	(D5):T<-0		
-01298	0000265896	00002150	66534608	4608	D9	SHLL2 R6 ...	DA	D9	D8...			
-01297	0000265898	00002152	6063074E	6063	DA	MOV R6,R0 ...	[DE]	DA	D9...	(D7):R5<-...		
-01296	0000265899	00002154	6063074E	074E	DB	MOV.L @{R0},...	DC	DB	DA...	(D8):R6<-...		
-01295	0000265901	00002156	65F36123	65F3	DC	MOV R15,R5 ...	[DD]	DC	DB...	(D9):R6<-...		
-01294	0000265902	7FFFFFFB0	65F36123	6123	DD	MOV R2,R1 ...	--	DD	DC...	(DA):R0<-...		
-01293	0000265904	00002158	00000000	6123	--	----	DF	--	DD...	(DB):R7<-...		

図 3-29 トレースウィンドウ (SH-3/SH-3E シリーズ)

表示する項目は以下の通りです。

[PTR]	トレースバッファ内ポインタ (最後に実行した命令が 0 となります)
[Cycle]	累計命令実行サイクル数 (パイプラインリセットによりクリアします)
[Address Bus]	アドレスバス上のデータ
[Data Bus]	データバス上のデータ
[Code]	命令コード
[No]	命令の番号 (各ステージの実行番号と対応しています)
[Instruction]	命令ニモニク
[IF]	フェッチした命令の番号 (メモリアクセスを行わないフェッチは命令番号を[ ]で囲んで表示)
[DE]	デコードした命令の番号
[EX]	実行した命令の番号
[MA]	メモリアクセスした命令の番号
[SW]	ライトバックした命令の番号
[Access Data]	データアクセスの内容 (転送先<-転送データの形式で表示)
[Source]	C/C++またはアセンブラソース
[Label]	ラベル

### 3. デバッグ

#### (3) SH3-DSP シリーズ

PTR	Cycle	Address	Code	IF_DE_EX...	No	Instruction	Access_Data	Source	Label
-01320	0000271616	0000213E	2F96	AF AE AC...	AE	MOV.L R9,@...	(AB):PR<-0000...	void chan...	_change
-01319	0000271618	00002140	7FD4	[B0]AF AE...	AF	ADD #D4,R15...			
-01318	0000271619	00002142	E200	[B1]B0 AF...	B0	MOV #000000...	(AE):05018FAC...	for(i=0; ...	
-01317	0000271621	00002144	E10A	B2 B1 B0...	B1	MOV #000000...	(AE):R15<-050...		
-01316	0000271623	00002146	3213	B3 B2 B1...	B2	CMP/GE R1,R...	(AF):R15<-050...		
-01315	0000271625	00002148	890E	B4 B3 B2...	B3	BT 00002168...	(B0):R2<-0000...		
-01314	0000271627	0000214A	6523	[B5]B4 B3...	B4	MOV R2,R5 ...	(B1):R1<-0000...	tmp[i] = ...	
-01313	0000271628	0000214C	6653	B6 B5 B4...	B5	MOV R5,R6 ...	(B2):T<-0		
-01312	0000271630	0000214E	4608	[E7]B6 B5...	B6	SHLL2 R6 ...			
-01311	0000271631	00002150	6063	B8 B7 B6...	B7	MOV R6,R0 ...	(B4):R5<-0000...		
-01310	0000271633	00002152	074E	[B9]B8 B7...	B8	MOV.L @(R0,...	(B5):R6<-0000...		
-01309	0000271634	00002154	65F3	BA B9 B8...	B9	MOV R15,R5 ...	(B6):R6<-0000...		
-01308	0000271636	00002156	6123	[BB]BA B9...	BA	MOV R2,R1 ...	(B7):R0<-0000...		
-01307	0000271637	00002158	6613	BC BB BA...	BB	MOV R1,R6 ...	(B8):R7<-0000...		
-01306	0000271639	0000215A	4608	[BD]BC BB...	BC	SHLL2 R6 ...	(B9):R5<-0501...		

図 3-30 トレースウィンドウ (SH3-DSP シリーズ)

表示する項目は以下の通りです。

[PTR]	トレースバッファ内ポインタ (最後に実行した命令が 0 となります)
[Cycle]	累計命令実行サイクル数 (パイプラインリセットによりクリアします)
[Address]	プログラムカウンタ値
[Code]	命令のコード
[IF]	フェッチした命令の番号 (メモリアクセスを行わないフェッチは命令番号を[ ]で囲んで表示)
[DE]	デコードした命令の番号
[EX]	実行した命令の番号
[MA]	メモリアクセスした命令の番号
[SW]	ライトバックした命令の番号
[No]	命令の番号 (各ステージの実行番号と対応しています)
[Instruction]	命令二モニク
[Access Data]	データアクセスの内容 (転送先<転送データの形式で表示)
[Source]	C/C++またはアセンブラソース
[Label]	ラベル

## (4) SH-4 シリーズ

PTR	Cycle	Address...	Code1	Code2	EX_EAS	LS_EAS	BR_EAS	N	Address	Code	Instruction	Access_Data
-01210	0003142320	00002174	*2F96	7FD4	9 x 6	x x 7	x 8 x	B	0000216C	2F96	MOV.L R...	R2<-000...
-01209	0003142321	00002178	E200	*7FD4	x 9 x	B x x	x x 8	C	0000216E	7FD4	ADD #D4...	PR<-000...
-01208	0003142375	00002178	*E200	E10A	C x 9	x B x	x x x	D	00002170	E200	MOV #00...	73FFFFAc...
-01207	0003142413	0000217C	3213	*E10A	D C x	x x B	x x x	E	00002172	E10A	MOV #0A...	R15<-73...
-01206	0003142414	00002180	*3213	890E	E D C	x x x	x x x	F	00002174	3213	CMP/GE ...	R15<-73...
-01205	0003142415	00002180	6523	*890E	F E D	x x x	x x x	0	00002176	890E	BT 0000...	R2<-000...
-01204	0003142415	00002180	6523	*890E	F E D	x x x	x x x	1	00002178	6523	MOV R2, R5	
-01203	0003142436	00002180	4608	*6653	1 F E	x x x	0 x x	2	0000217A	6653	MOV R5, R6	R1<-000...
-01202	0003142436	00002180	4608	*6653	1 F E	x x x	0 x x	3	0000217C	4608	SHLL2 R6	
-01201	0003142459	00002184	xxxx	*6063	3 1 F	2 x x	x 0 x	4	0000217E	6063	MOV R6, R0	SR<-000...
-01200	0003142482	00002188	074E	*65F3	4 3 1	x 2 x	x x 0	5	00002180	074E	MOV.L @...	R5<-000...
-01199	0003142482	00002188	074E	*65F3	4 3 1	x 2 x	x x 0	6	00002182	65F3	MOV R15...	
-01198	0003142505	0000218C	*6123	6613	6 4 3	5 x 2	x x x	7	00002184	6123	MOV R2, R1	R6<-000...
-01197	0003142505	0000218C	*6123	6613	6 4 3	5 x 2	x x x	8	00002186	6613	MOV R1, R6	
-01196	0003142560	00002190	*4608	6063	7 6 4	8 5 x	x x x	9	00002188	4608	SHLL2 R6	R0<-000...
-01195	0003142561	00002194	0576	*6063	9 7 6	x 8 5	x x x	A	0000218A	6063	MOV R6, R0	R5<-73F...

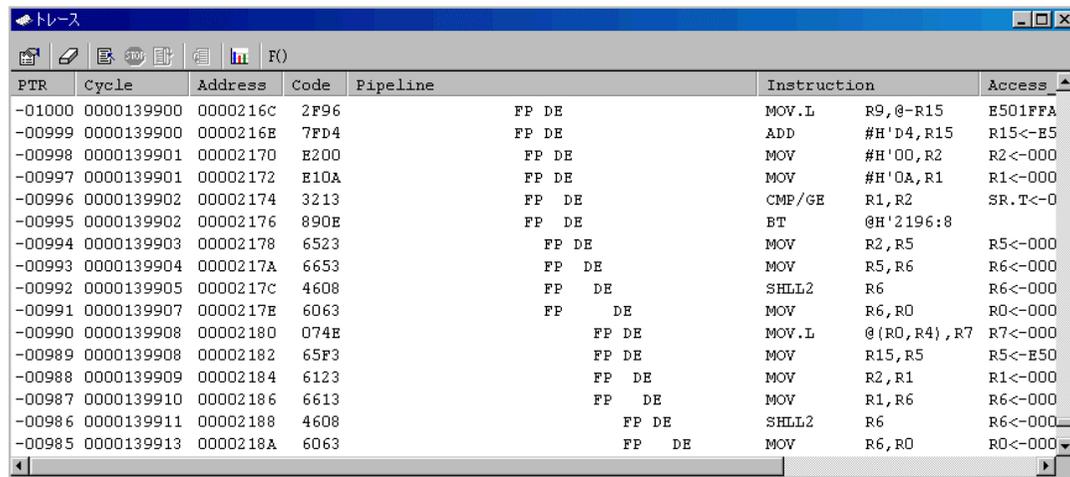
図 3-31 トレースウィンドウ (SH-4 シリーズ)

表示する項目は以下の通りです。

[PTR]	トレースバッファ内ポインタ (最後に実行した命令が 0 となります)
[Cycle]	累計命令実行サイクル数。パイプラインリセットによりクリアします。 サイクル数は、CPU 内部クロックでのサイクル数です。外部クロックと内部クロックの比は Clock_Rate コマンドで設定できます。
[Address Bus]	プログラムカウンタ値
[Code1]	フェッチした命令のコード 1
[Code2]	フェッチした命令のコード 2 (現在デコードしている側に "*" を表示します。なお、並列実行時は若いアドレスの側に表示します。)
[EX-EAS]	EX パイプラインで実行 (E)、メモリアクセス (A)、ライトバック (S) した命令の番号
[LS-EAS]	LS パイプラインで実行、メモリアクセス、ライトバックした命令の番号
[BR-EAS]	BR パイプラインで実行、メモリアクセス、ライトバックした命令の番号
[FP-EXASD]	FP パイプラインで実行、メモリアクセス、ライトバックした命令の番号 (X ステージは、FSCA、FSRRA、FIPR、FTRV 命令、D ステージは、FDIV、FSQRT 命令のみ使用します。)
[No]	命令の番号 (各ステージの実行番号と対応しています)
[Address]	実行した命令アドレス
[Code]	実行した命令コード
[Instruction]	実行した命令二モニク
[Access Data]	データアクセスの内容 (転送先<-転送データの形式で表示)
[Source]	C/C++またはアセンブラソース
[Label]	ラベル

### 3. デバッグ

#### (5) SH2A-FPU、SH-4A シリーズ



PTR	Cycle	Address	Code	Pipeline	Instruction	Access
-01000	0000139900	0000216C	2F96		MOV.L R9, @-R15	E501FFA
-00999	0000139900	0000216E	7FD4	FP DE	ADD #H'D4, R15	R15<-E5
-00998	0000139901	00002170	E200	FP DE	MOV #H'00, R2	R2<-000
-00997	0000139901	00002172	E10A	FP DE	MOV #H'0A, R1	R1<-000
-00996	0000139902	00002174	3213	FP DE	CMP/GE R1, R2	SR.T<-0
-00995	0000139902	00002176	890E	FP DE	BT @H'2196:8	
-00994	0000139903	00002178	6523	FP DE	MOV R2, R5	R5<-000
-00993	0000139904	0000217A	6653	FP DE	MOV R5, R6	R6<-000
-00992	0000139905	0000217C	4608	FP DE	SHLL2 R6	R6<-000
-00991	0000139907	0000217E	6063	FP DE	MOV R6, R0	R0<-000
-00990	0000139908	00002180	074E	FP DE	MOV.L @(R0, R4), R7	R7<-000
-00989	0000139908	00002182	65F3	FP DE	MOV R15, R5	R5<-E50
-00988	0000139909	00002184	6123	FP DE	MOV R2, R1	R1<-000
-00987	0000139910	00002186	6613	FP DE	MOV R1, R6	R6<-000
-00986	0000139911	00002188	4608	FP DE	SHLL2 R6	R6<-000
-00985	0000139913	0000218A	6063	FP DE	MOV R6, R0	R0<-000

図 3-32 トレースウィンドウ (SH-4A シリーズ)

表示する項目は以下の通りです。

[PTR]	トレースバッファ内ポインタ (最後に実行した命令が 0 となります)
[Cycle]	累計命令実行サイクル数 (パイプラインリセットによりクリアします)
[Address]	プログラムカウンタ値
[Code]	命令のコード
[Pipeline]	パイプラインの実行状況

各記号の意味は以下の通りです。

- F : 命令フェッチ
- P : 命令プリデコード (SH-4A シリーズのみ)
- D : 命令デコード
- E : 命令実行
- M : メモリアクセス
- W : ライトバック

パイプライン動作の詳細については、各デバイスのプログラミングマニュアルを参照してください。

命令単位シミュレータの場合は、パイプライン情報を持っていないため"x"を表示します。パイプラインが表示範囲(パイプライン表示幅の右端)を越えた場合は、先頭位置(パイプライン表示幅の左端)から続けて表示します。

[Instruction]	命令ニーモニック
[Access Data]	データアクセスの内容 (転送先<-転送データの形式で表示)
[Source]	C/C++またはアセンブラソース
[Label]	ラベル

#### 3.5.4 トレースレコードを検索する

トレースレコードを検索するには[トレース検索]ダイアログボックスを使用します。

[トレース検索]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。

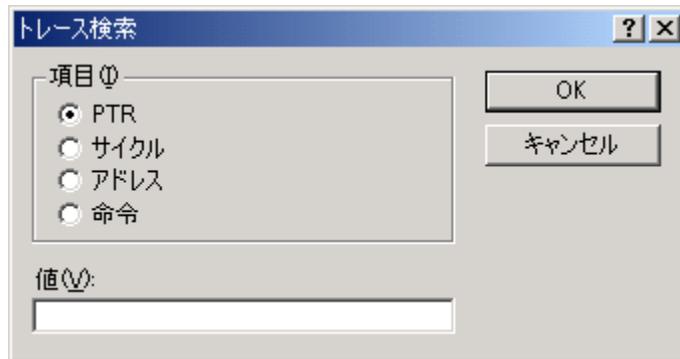


図 3-33 トレース検索ダイアログボックス

本ダイアログボックスは、トレース情報の検索条件を設定します。[項目]で検索対象項目を指定し、[値]で指定した内容を検索します。

[PTR]	トレースバッファ内ポインタ。最後に実行した命令が0となります。 -nmn の形式で指定してください。
[サイクル]	累計命令実行サイクル数
[アドレス]	命令アドレス
[命令]	命令モニック

[OK]ボタンをクリックすることにより、検索条件を設定し、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青色で表示します。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。

### 3.5.5 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[トレース]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[トレース]ウィンドウをクリアすることになります。

### 3.5.6 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。トレースバッファの内容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。このファイルはトレースバッファに再ロードできないことに注意してください。

#### 3.5.7 ソースファイルを表示する

トレースレコードに対応するソースファイルを[エディタ]ウィンドウに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、選択した行をカーソルで示します。

#### 3.5.8 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。

取り除いた状態だと[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[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。

### 3.5.9 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。

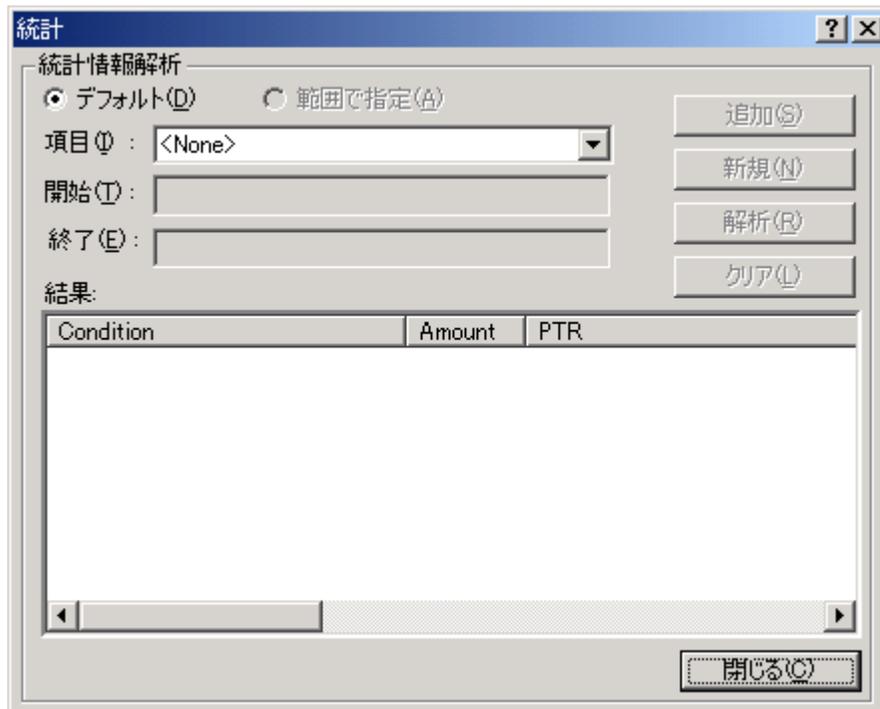


図 3-34 統計ダイアログボックス

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]で解析対象項目を指定し、[開始]および[終了]で入力値または文字列を指定します。

[デフォルト]を選択すると入力値または文字列を範囲で指定することはできません。範囲で指定する場合は[範囲で指定]を選択してください。

- |       |                       |
|-------|-----------------------|
| [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       |
| [クリア] | すべての条件と統計情報解析結果を削除します |

[閉じる]ボタンをクリックすると、ダイアログボックスを閉じます。

## 3.6 プロファイル情報を見る

プロファイル機能は、アプリケーションプログラムの実行パフォーマンスを関数単位に測定します。アプリケーションプログラム中の性能劣化の原因となっている場所および要因を調査することができます。

HEWはプロファイルデータの参照方法、参照目的に応じて、3つのウィンドウでプロファイル測定結果を表示します。

### 3.6.1 スタック情報ファイル

プロファイル機能は、最適化リンカ(Ver.7.0以降)が出力するスタック情報ファイル(拡張子".SNI")を読み込むことができます。このファイルには、ソースファイル上の(静的な)関数呼び出し関係の情報が入っています。HEWがスタック情報ファイルを読み込むことで、ユーザアプリケーションが未実行(プロファイルデータの測定を行う前)でも、関数の呼び出し関係を表示できるようになります。(但し、[プロファイル]ウィンドウのポップアップメニューで[表示設定->未実行関数を表示しない]をチェックしている場合を除きます。)

HEWがスタック情報ファイルを読み込まない場合、プロファイル機能で表示するデータは、プロファイルデータ測定中に実行した関数についてのみになります。

リンカでスタック情報ファイルを生成するには、[ビルド-> SuperH Risc engine Standard Toolchain...]を選択して、[Standard Toolchain]ダイアログボックスを開き、[最適化リンカ]タブで[カテゴリ]リストボックスを[その他]に指定し、[スタック情報ファイル(sni)出力]チェックボックスをチェックしてください。

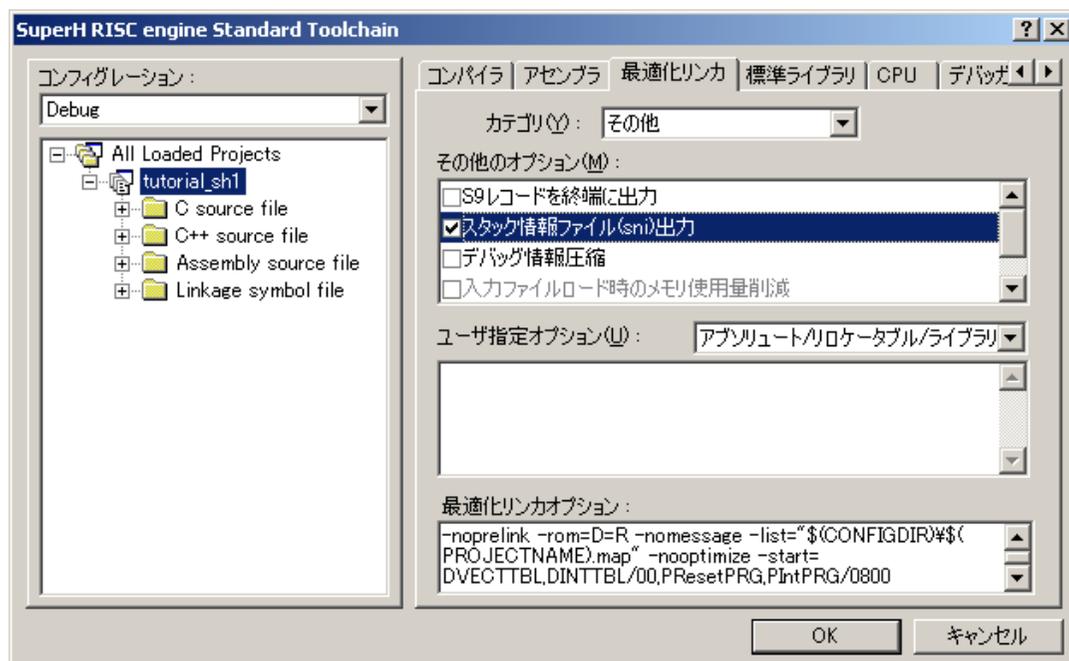


図 3-35 Standard Toolchain ダイアログボックス(1)

### 3.6.2 プロファイル情報ファイル

プロファイル情報ファイルを作成するためには、アプリケーションプログラムのプロファイルデータを測定後に、[プロファイル]ウィンドウのPop-upメニューで[プロファイル情報の保存...]メニューオプションを選択し、ファイル名を指定します。

プロファイル情報ファイルには、関数の呼び出し回数とグローバル変数のアクセス回数の情報が入っています。最適化リンカ(Ver.7.0以降)は、プロファイル情報ファイルを読み込み、関数および変数の配置を実際のプログラム動作状況に合わせた配置に最適化する機能を持っています。

プロファイル情報ファイルをリンカに入力するには、[Standard Toolchain]ダイアログボックスの[最適化リンカ]タブで[カテゴリ]リストボックスを[最適化]に指定し、[プロファイル情報]チェックボックスをチェックして、プロファイル情報ファイル名を指定してください。



図 3-36 Standard Toolchain ダイアログボックス(2)

なお、[プロファイル情報]チェックボックスを有効にするには、[最適化]リストボックスを[最適化なし]以外に設定する必要があります。

### 3.6.3 スタック情報ファイルのロード

スタック情報ファイルを読み込むかどうかは、ロードモジュールロード時に表示する、確認のメッセージボックスで指定できます。メッセージボックスの[OK]ボタンをクリックするとスタック情報ファイルをロードします。

確認のメッセージボックスは、次の場合に表示します。

- スタック情報ファイルが存在する時
- [オプション]ダイアログボックス(メインメニューの[基本設定->オプション...])を選択すると開きます)の[確認]タブ(図 3-37)で[スタック情報ファイルをロードします (SNI ファイル)]チェックボックスをチェックしている場合

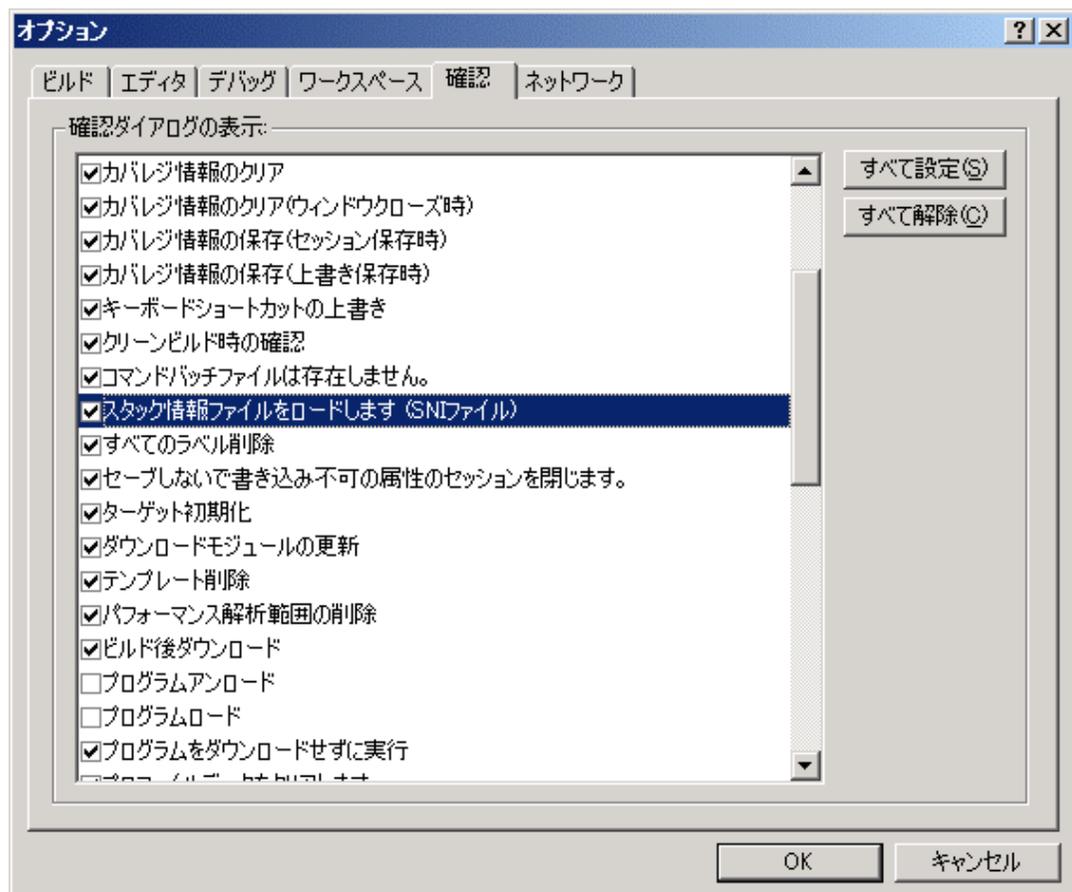


図 3-37 オプションダイアログボックス

### 3.6.4 プロファイルを有効にする

[表示->パフォーマンス->プロファイル]を選択し、[プロファイル]ウィンドウをオープンします。

[プロファイル]ウィンドウのポップアップメニューで[有効]メニューオプションを選択します（メニューにチェックマークが付きます）。

### 3.6.5 測定方法を指定する

プロファイルデータの測定時に、関数呼び出しをトレースするかどうかを指定できます。関数呼び出しをトレースすると、ユーザプログラム実行時の関数呼び出し関係をツリー形式で表示できるようになります。関数呼び出しをトレースしないと、関数呼び出し関係を表示できませんが、Profileデータの測定時間を短縮することができます。

関数呼び出しをトレースしないようにするためには、[プロファイル]ウィンドウのポップアップメニュー[関数呼び出しをトレースしない]を選択します。（メニューにチェックマークが付きます。）

また、OSによるタスクスイッチなど、通常の方法以外で関数を呼び出しているプログラムの場合、関数呼び出しを正しく表示できない場合がありますので、関数呼び出しをトレースせずにプロファイルデータを測定してください。

### 3.6.6 ユーザプログラムを実行し結果を確認する

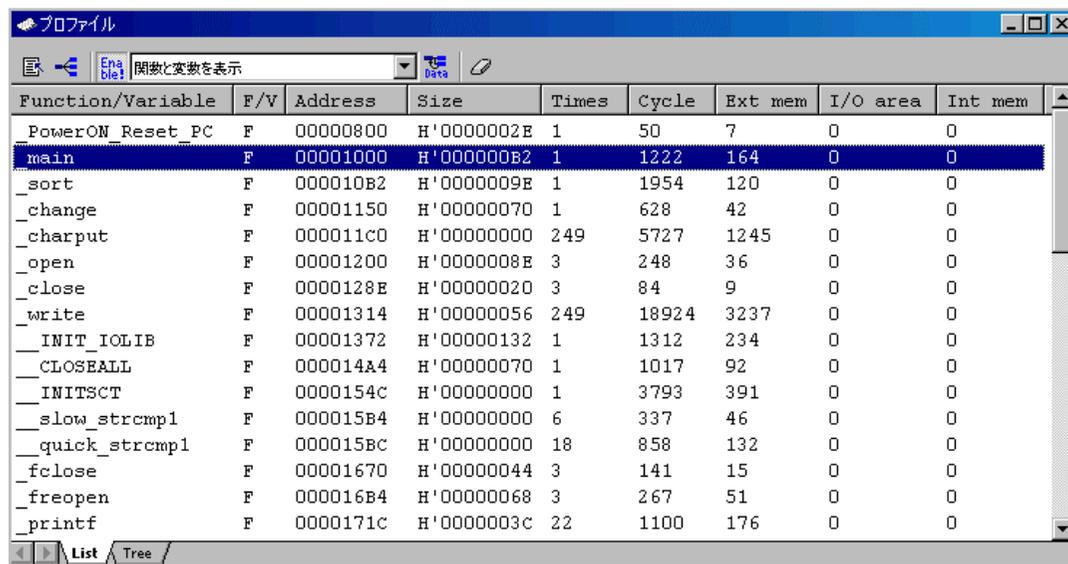
ユーザプログラムを実行し、停止すると[プロファイル]ウィンドウに測定結果を表示します。

[プロファイル]ウィンドウには、[List]シートと[Tree]シートがあります。

### 3. デバッグ

#### 3.6.7 List シート

関数とグローバル変数をリスト表示し、各関数/変数のプロファイルデータを表示します。



The screenshot shows a window titled 'プロファイル' (Profile) with a table of function/variable data. The table has columns for Function/Variable, F/V, Address, Size, Times, Cycle, Ext mem, I/O area, and Int mem. The 'main' function is highlighted in blue.

Function/Variable	F/V	Address	Size	Times	Cycle	Ext mem	I/O area	Int mem
PowerON_Reset_PC	F	00000800	H'0000002E	1	50	7	0	0
main	F	00001000	H'000000B2	1	1222	164	0	0
_sort	F	000010B2	H'0000009E	1	1954	120	0	0
_change	F	00001150	H'00000070	1	628	42	0	0
_charput	F	000011C0	H'00000000	249	5727	1245	0	0
_open	F	00001200	H'0000008E	3	248	36	0	0
_close	F	0000128E	H'00000020	3	84	9	0	0
_write	F	00001314	H'00000056	249	18924	3237	0	0
_INIT_IOLIB	F	00001372	H'00000032	1	1312	234	0	0
_CLOSEALL	F	000014A4	H'00000070	1	1017	92	0	0
_INIT_SCT	F	0000154C	H'00000000	1	3793	391	0	0
_slow_strncmp1	F	000015B4	H'00000000	6	337	46	0	0
_quick_strncmp1	F	000015BC	H'00000000	18	858	132	0	0
_fclose	F	00001670	H'00000044	3	141	15	0	0
_freopen	F	000016B4	H'00000068	3	267	51	0	0
_printf	F	0000171C	H'0000003C	22	1100	176	0	0

図 3-38 List シート

カラムヘッダをクリックすると、アルファベットまたは数値の昇降順にソートして表示します。

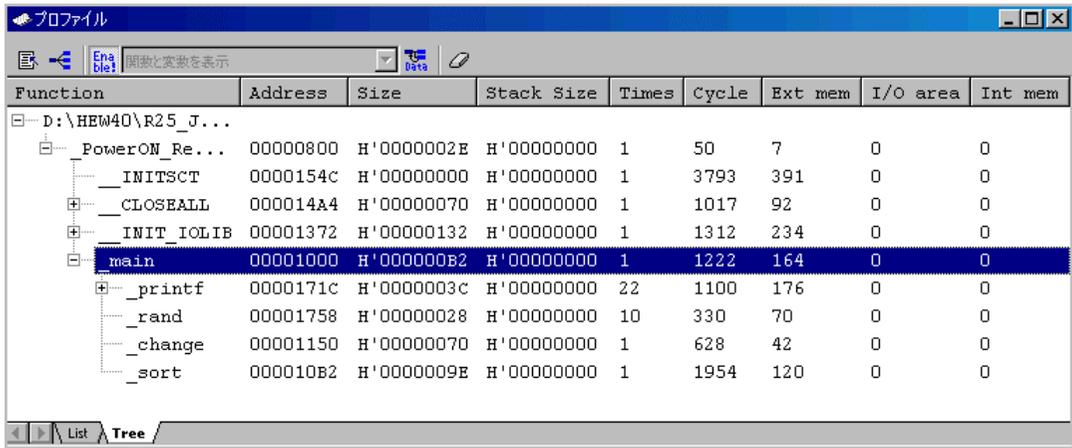
[Function/Variable]列または[Address]列をクリックすると、該当するアドレスに対応したソースプログラムを表示します。

ウィンドウ内でマウスの右ボタンをクリックするとポップアップメニューを表示します。このポップアップメニューについては「3.6.8 Tree シート」を参照してください。

### 3.6.8 Tree シート

関数の呼び出し関係を表示し、各呼び出し位置におけるプロファイルデータを表示します。

[Tree]シートは、[プロファイル]ウィンドウのポップアップメニュー[関数呼び出しをトレースしない]をチェックしていない時のみ有効です。



Function	Address	Size	Stack Size	Times	Cycle	Ext mem	I/O area	Int mem
PowerON_Re...	00000800	H'0000002E	H'00000000	1	50	7	0	0
__INITSC	0000154C	H'00000000	H'00000000	1	3793	391	0	0
__CLOSEALL	000014A4	H'00000070	H'00000000	1	1017	92	0	0
__INIT_IOLIB	00001372	H'00000132	H'00000000	1	1312	234	0	0
main	00001000	H'000000B2	H'00000000	1	1222	164	0	0
__printf	0000171C	H'0000003C	H'00000000	22	1100	176	0	0
__rand	00001758	H'00000028	H'00000000	10	330	70	0	0
__change	00001150	H'00000070	H'00000000	1	628	42	0	0
__sort	000010B2	H'0000009E	H'00000000	1	1954	120	0	0

図 3-39 Tree シート

[Function]列の関数をダブルクリックすると、ツリー構造を拡張または収縮表示します。また、'+/-'キーでも拡張/収縮表示することができます。[Address]列をダブルクリックすると、該当するアドレスに対応したソースプログラムを表示します。

ウィンドウ内でマウスの右ボタンをクリックするとポップアップメニューを表示します。このメニューは以下のオプションを含みます。

(1) ソースファイル表示

選択している行の該当アドレスに対応したソースプログラムまたは逆アセンブルを表示します。

(2) チャート表示

選択している行の関数に着目した[プロファイルチャート]ウィンドウを表示します。

(3) 有効

プロファイルデータ測定のオン・オフを切り替えます。プロファイルデータ測定がオンのとき、メニューテキストの左にチェックマークを表示します。

(4) 関数呼び出しをトレースしない

本メニューをチェックすると、プロファイルデータ測定時に関数呼び出しをトレースしません。例えば、OSのタスクスイッチのように通常の方法以外で関数が呼び出されるプログラムのデータを測定する場合に使用します。

[プロファイル]ウィンドウの[Tree]シートで関数呼び出し関係を表示するためには、本メニューをチェックせずにプロファイルデータを測定してください。また、測定結果のプロファイル情報ファイルを使用して、最適化リンケージエディタによる最適化を行う場合も、本メニューをチェックしないでください。

### 3. デバッグ

---

#### (5) 検索...

[Function]列の文字列を検索する[テキスト検索]ダイアログボックスを表示します。検索したい文字列をエディットボックスに入力し、[次を検索]ボタンまたは、'Enter'キーを入力すると、検索を開始します。

#### (6) データ検索...

[データ検索]ダイアログボックスを表示します。



図 3-40 データ検索ダイアログボックス

[カラム]コンボボックスで検索カラムを、[検索データ]グループで検索方向を設定し、[次を検索]ボタンまたは、'Enter'キーを入力すると、検索を開始します。また、連続して[次を検索]ボタンまたは'Enter'キーを入力すると、次に大きいデータ（最小値の場合は小さいデータ）を検索します。

#### (7) データクリア

関数呼び出し回数のカウントおよびプロファイルデータをクリアします。[プロファイル]ウィンドウの[List]シートおよび[プロファイルチャート]ウィンドウのデータもクリアします。

#### (8) プロファイル情報の保存...

[プロファイル情報の保存]ダイアログボックスを表示します。プロファイル結果をプロファイル情報ファイル（拡張子は".pro"）に保存します。最適化リンケージエディタは、プロファイル情報を元に、ユーザプログラムの最適化を行うことが出来ます。プロファイル情報を使用した最適化についての詳細は、最適化リンケージエディタのマニュアルを参照してください。

**【注】** [関数呼び出しをトレースしない]メニューをチェックして測定した結果のプロファイル情報では、最適化リンケージエディタによる最適化は行えません。

#### (9) テキスト形式で保存...

[プロファイルデータをテキスト形式で保存]ダイアログボックスを表示します。表示している状態をテキストファイルに保存します。

## (10) 表示設定

このメニューには下記サブメニューがあります。(以下の説明には[List]シートのためのメニューも含まれます)

## (a) 関数と変数を表示

[Function/Variable]列で、関数およびグローバル変数の両方表示します。

## (b) 関数を表示

[Function/Variable]列で、関数のみを表示します。

## (c) 変数を表示

[Function/Variable]列で、グローバル変数のみを表示します。

## (d) 未実行関数を表示しない

実行した関数のみ表示することができます。最適化リンケージエディタが出力するスタック使用量情報ファイル(拡張子: sni)がロードモジュールと同一ディレクトリに存在しない場合、このチェックボックスの設定に関わらず、実行関数のみ表示します。

## (e) 子関数の実行結果を含んで表示

表示するプロファイルデータに、関数内で呼び出した子関数のプロファイルデータを含めるかどうかを設定します。

## (11) プロパティ...

本シミュレータ・デバッガでは使用できません。

### 3.6.9 プロファイルチャートウィンドウ

[プロファイルチャート]ウィンドウは、特定の関数に着目した関数の呼び出し関係を表示します。本ウィンドウは、着目する関数を中心に表示し、その左側には着目した関数を呼び出した関数、右側には、着目している関数が呼び出した関数を、それぞれ表示します。また、各呼び出しを行った回数も表示します。

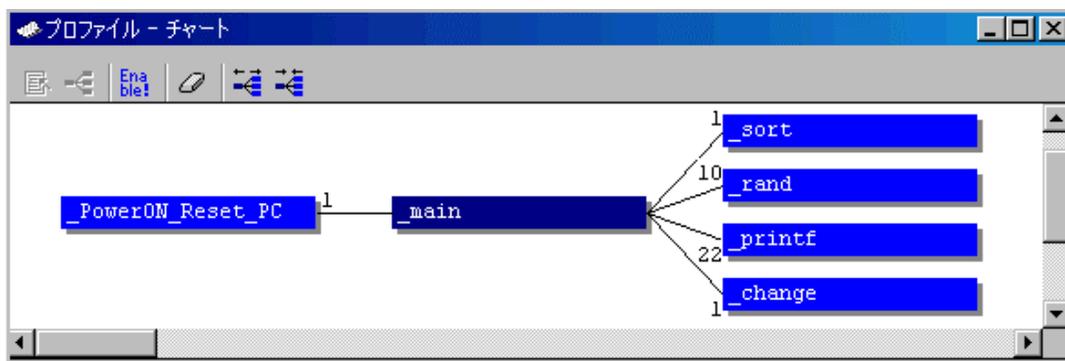


図 3-41 プロファイルチャートウィンドウ

ウィンドウ内でマウスの右ボタンをクリックするとポップアップメニューを表示します。このメニューは以下のオプションを含みます。

#### (1) ソースファイル表示

マウスの右ボタンをクリックしたときの位置にある関数の該当アドレスに対応したソースプログラムまたは逆アセンブルを表示します。マウスの右ボタンをクリックしたときの位置が関数ではない場合、このメニューオプションはグレー表示となります。

#### (2) チャート表示

マウスの右ボタンをクリックしたときの位置にある関数に着目した[プロファイル-チャート]ウィンドウを表示します。マウスの右ボタンをクリックしたときの位置が関数ではない場合、このメニューオプションはグレー表示となります。

#### (3) 有効

プロファイルデータ収集の有効/無効を切り替えます。プロファイルデータ測定が有効のとき、メニューテキストの左にチェックマークを表示します。

#### (4) データクリア

関数呼び出し回数のカウントをクリアします。[プロファイル]ウィンドウの[List]シートおよび[Tree]シートのデータもクリアします。

#### (5) チャートウィンドウを複数開く

[プロファイル-チャート]ウィンドウを表示する際、既に[プロファイル-チャート]ウィンドウが開いているとき、別のウィンドウを開くか同ウィンドウに表示するかを設定します。メニューテキストの左にチェックマークを表示していれば、別のウィンドウを開きます。

#### (6) プロファイル情報の保存...

[プロファイル情報の保存]ダイアログボックスを表示します。プロファイル結果をプロファイル情報ファイル(拡張子は".pro")に保存します。最適化リンケージエディタは、プロファイル情報を元に、ユーザプログラムの最適化を行うことが出来ます。プロファイル情報を使用した最適化についての詳細は、最適化リンケージエディタのマニュアルを参照してください。

#### (7) 拡大

各関数の間隔を広げて表示します。また、'+キーでも広げて表示することができます。

#### (8) 縮小

各関数の間隔を縮めて表示します。また、'-キーでも縮めて表示することができます。

### 3.6.10 表示データの種類および用途

プロファイル機能から下記情報を得ることができます。

#### (1) Address

関数を配置しているメモリ上の位置を知ることができます。アドレス順にソート表示することにより、メモリ上の配置イメージで関数とグローバル変数を並べることができます。

#### (2) Size

サイズ順にソート表示すれば、サイズが小さくて頻繁に呼び出している関数を見つけることができます。そのような関数があればinline関数にすることで、関数呼び出しのオーバーヘッドを減らせる場合があります。

また、キャッシュ内蔵マイコンをご使用の場合、サイズの大きい関数を実行すると、更新するキャッシュのサイズが大きくなります。このような、キャッシュミスの原因となり得る関数を頻繁に呼び出していないかを容易に確認できます。

#### (3) Stack Size

関数呼び出しのネストが深い場合、関数呼び出し経路をたどり、その経路上の全関数のスタックサイズを合計することで、おおよそのスタック使用量を見積もれます。

#### (4) Times

呼び出し(アクセス)回数順にソート表示すれば、頻繁に呼び出している関数や頻繁にアクセスしている変数を容易に調べることができます。

#### (5) プロファイルデータ

CPUにより、さまざまなデータを測定できます。各CPUで測定するデータは以下の通りです。

- SH-1/SH-2/SH-2E シリーズ、SH2-DSP(SH7410)、SH2-DSP(Core)、およびSH2-DSP(SH7065) [Cycle] (実行サイクル数)、

### 3. デバッグ

[Ext\_mem] (外部メモリアクセス回数)、[I/O\_area] (内蔵I/Oアクセス回数)、  
[Int\_mem] (内部メモリアクセス回数)

- SH-3/SH-3E/SH3-DSP シリーズ、SH2-DSP(SH7612)

[Cycle] (実行サイクル数)、

[Cache miss] (キャッシュミス数)、[Ext\_mem] (外部メモリアクセス回数)、  
[I/O\_area] (内蔵I/Oアクセス回数)、[Int\_mem] (内部メモリアクセス回数)

- SH-4 シリーズ、SH2A-FPU、SH-4A シリーズ

[Cycle] (実行サイクル数)、

[ICache miss] (命令キャッシュミス数)、[OCache miss] (オペランドキャッシュミス数)、

[Ext\_mem] (外部メモリアクセス回数)、[I/O\_area] (内蔵I/Oアクセス回数)、

[Int\_mem] (内部メモリアクセス回数)

実行サイクル数、およびキャッシュミス数は、当該関数コール命令実行時の累計実行サイクル数 (キャッシュミス数) と当該関数からのリターン命令実行時の累計実行サイクル数 (キャッシュミス数) の差から求めています。

#### 3.6.11 プロファイル情報ファイルを作成する

プロファイル情報ファイルを作成する場合は、Pop-upメニューの[プロファイル情報の保存...]メニューオプションを選択します。[プロファイル情報の保存]ダイアログボックスを表示します。ファイル名を選択して[保存]ボタンを押すと、選択したファイルにプロファイル情報を書きこみます。[全て保存]ボタンを押すと、全てのファイルにプロファイル情報を書きこみます。

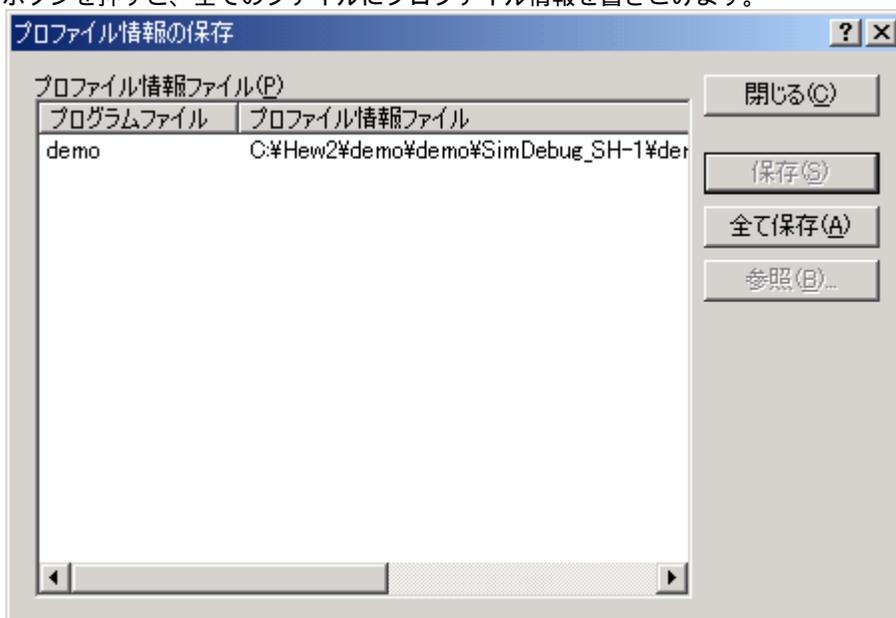


図 3-42 プロファイル情報の保存ダイアログボックス

#### 3.6.12 注意事項

- (1) アプリケーションプログラムの実行サイクル数をプロファイル機能で測定した場合のデータには誤差があります。プロファイル機能では、アプリケーションプログラム全体の中で各

関数が占める実行時間の比率を調べることができますが、より厳密に関数の実行サイクル数を測定したい場合には、パフォーマンス解析機能を使用してください。

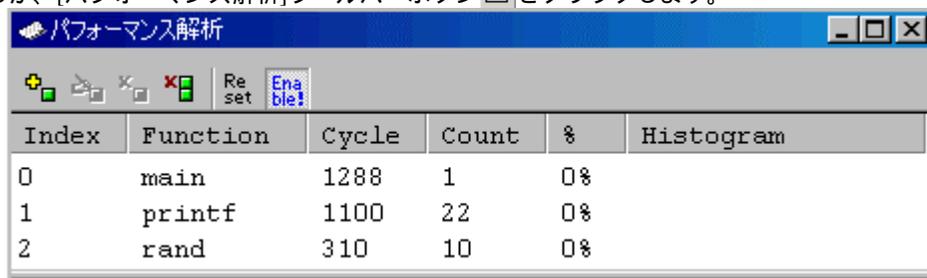
- (2) デバッグ情報が無いロードモジュールでプロファイル情報の測定を行った場合、関数名を表示しない場合があります。
- (3) スタック情報ファイル(拡張子".SNI")はロードモジュールファイル(拡張子".ABS")と同一のディレクトリに置いてある必要があります。
- (4) 測定結果の蓄積はできません。
- (5) 測定結果の編集はできません。

## 3.7 パフォーマンスを解析する

関数名を指定してパフォーマンスを解析する場合は、[パフォーマンス解析]ウィンドウを使用します。

### 3.7.1 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  をクリックします。



Index	Function	Cycle	Count	%	Histogram
0	main	1288	1	0%	
1	printf	1100	22	0%	
2	rand	310	10	0%	

図 3-43 パフォーマンス解析ウィンドウ

本ウィンドウは、指定関数ごとの実行サイクル数を表示します。  
実行サイクル数は、以下の計算で求めています。

実行サイクル数 = 指定関数からのリターン命令実行時累計実行サイクル数 - 指定関数コール命令実行時累計実行サイクル数

表示する項目は以下の通りです。

[Index]	設定条件のインデックス番号
[Function]	測定対象の関数名（または関数の開始アドレス）
[Cycle]	当該関数の累計実行サイクル数
[Count]	当該関数の累計呼び出し回数
[%]	プログラム全体の実行サイクル数に占める当該関数の実行サイクル数の割合
[Histogram]	上記割合のヒストグラム表示

#### 3.7.2 評価関数を設定する

[パフォーマンス解析]ウィンドウを開いたら、評価する関数を設定します。ポップアップメニューから[範囲の追加...]を選択すると、[パフォーマンスオプション]ダイアログボックスが開きます。また、'Insert'キーでも[パフォーマンスオプション]ダイアログボックスを開くことができます。



図 3-44 パフォーマンスオプションダイアログボックス

本ダイアログボックスでは、性能評価する関数（ラベルも可）を設定します。評価関数は255個まで設定可能です。

[OK]ボタンをクリックすることにより、評価関数を設定します。[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。

設定した評価関数を選択後、ポップアップメニューの[範囲の編集]を選択すると[パフォーマンスオプション]ダイアログボックスを表示して、評価関数を変更できます。また、'Enter'キーでも[パフォーマンスオプション]ダイアログボックスを開くことができます。

#### 3.7.3 データ収集を開始する

ポップアップメニューの[有効]をチェックして、プログラムを実行するとパフォーマンスデータを収集できます。

#### 3.7.4 データをリセットする

ポップアップメニューから[リセット]を選択すると、現在のプログラムのパフォーマンスデータをクリアします。

#### 3.7.5 評価関数を削除する

評価関数を選択した状態で、ポップアップメニューから[範囲の削除]を選択すると、選択された評価関数を削除し、他の範囲のデータを再計算します。また、'Delete'キーでも評価関数を削除することができます。

#### 3.7.6 すべての評価関数を削除する

ポップアップメニューから[全ての範囲を削除]を選択すると、現在の評価関数をすべて削除し、パフォーマンスデータをクリアします。

#### 3.7.7 現在の表示内容をセーブする

現在ウィンドウに表示している内容をテキストファイルにセーブすることが出来ます。ポップアップメニューから[ファイルに保存...]を選択してください。

## 3.8 コードカバレッジを測定する

[カバレッジ]ウィンドウは、ユーザが指定したアドレス範囲についてコードカバレッジ情報(C0 カバレッジおよび C1 カバレッジ)を収集し、結果を表示します。

### 3.8.1 カバレッジウィンドウを開く

[表示->コード->カバレッジ...]を選択するか[カバレッジ]ツールバーボタンをクリックすると、[カバレッジの設定]ダイアログボックスが開きます。

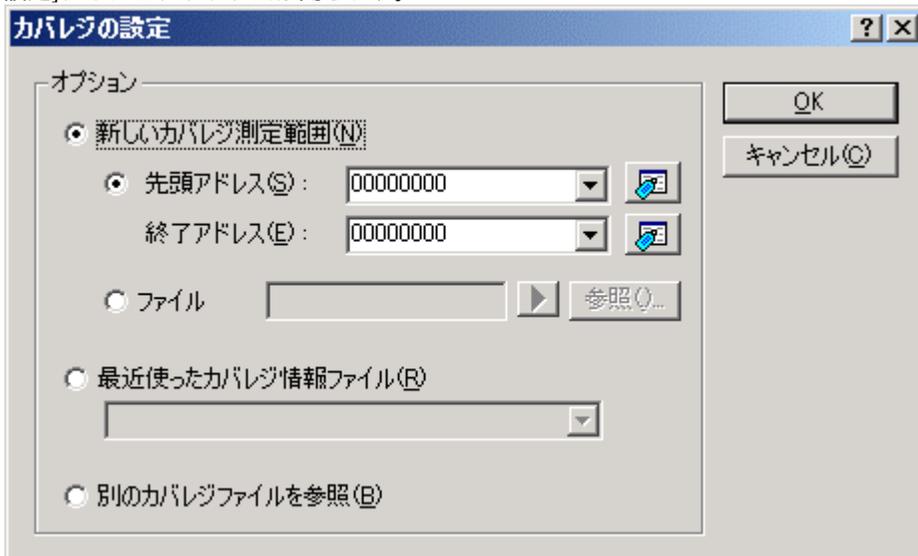


図 3-45 カバレッジの設定ダイアログボックス

[カバレッジの設定]ダイアログボックスでは、カバレッジ測定範囲を指定します。新たなカバレッジ範囲を設定するには、以下に示す 2 通りの方法があります。

- [新しいカバレッジ測定範囲]で開始アドレスおよび終了アドレスを指定する  
指定項目は以下のとおりです。

[先頭アドレス]      カバレッジ情報表示の開始アドレスを指定します。(接頭辞省略時は 16 進で入力)  
[終了アドレス]      カバレッジ情報表示の終了アドレスを指定します。(接頭辞省略時は 16 進で入力)

- [新しいカバレッジ測定範囲]でファイルを指定する  
指定項目は以下のとおりです。

[ファイル]      現在のプロジェクト中の“.C”または“.CPP”を拡張子にもつソースファイルを指定します。これにより指定ファイル内に存在する関数をカバレッジ範囲として設定します。拡張子を省略した場合は、“.C”を補います。“.C”または“.CPP”以外の拡張子を持つファイルは指定できません。ブレースホルダまたは[参照...]ボタンが利用できます。

### 3. デバッグ

---

カバレッジ情報ファイルに保存した内容を再設定する場合は、[最近使ったカバレッジ情報ファイル]から選択するか、[別のカバレッジファイルを参照]でファイルオープンダイアログボックスを表示して選択します。

[最近使ったカバレッジ情報ファイル]では最近保存されたファイルを4個まで表示します。

[OK]ボタンをクリックすると[カバレッジ]ウィンドウを表示します。

アドレス指定で[カバレッジ]ウィンドウがすでに表示されている場合は、そのウィンドウに設定が追加されます。

#### (1) カバレッジウィンドウ (アドレス指定)

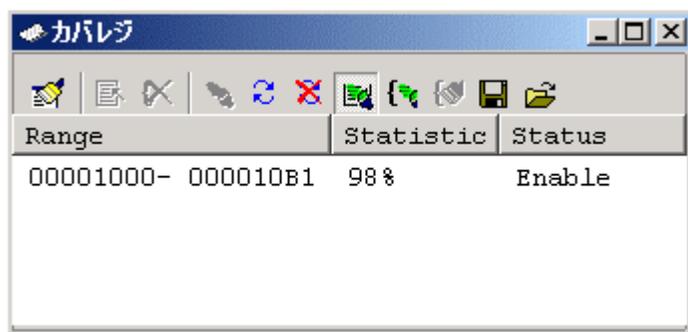


図 3-46 カバレッジウィンドウ

カバレッジ範囲とカバレッジ統計情報を表示します。  
表示する項目は以下の通りです。

[Range]	アドレス範囲
[Statistic]	C0 カバレッジ値(パーセント表示)
[Status]	各カバレッジ範囲の有効/無効状態

[カバレッジ]ウィンドウをクローズすると取得したカバレッジ情報と取得条件の設定をクリアします。

#### (2) カバレッジウィンドウ (ソースファイル指定)

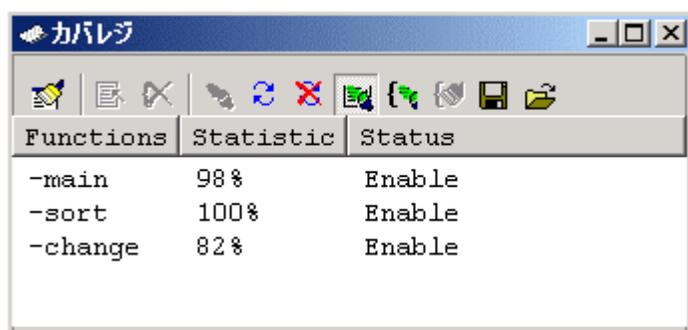


図 3-47 カバレッジウィンドウ

カバレッジ範囲とカバレッジ統計情報を表示します。

表示する項目は以下の通りです。

[Functions]	カバレジ対象の関数
[Statistic]	C0 カバレジ値(パーセント表示)
[Status]	各関数の有効/無効状態

- カラムタブをクリックすることで関数名または C0 カバレジ値の降順または昇順に並べ替えることができます。

[カバレジ]ウィンドウをクローズすると取得したカバレジ情報と取得条件の設定をクリアします。

### 3.8.2 カバレジ情報をすべて取得する

ポップアップメニューで[すべて有効]をチェック後、命令を実行するとすべてのカバレジ情報を取得します。デフォルトで[すべて有効]はチェックされています。

### 3.8.3 カバレジ情報をすべてクリアする

ポップアップメニューから[すべてクリア]を選択すると、取得したすべてのカバレジ情報をクリアします。

### 3.8.4 ソースファイルを表示する

ポップアップメニューから[ソースファイル表示]を選択すると、[エディタ]ウィンドウを開いて、[カバレジ]ウィンドウ上のカーソル位置のアドレスに対応するソースファイルを表示します。

### 3.8.5 新しいカバレジ測定範囲を設定する

ポップアップメニューで[測定範囲追加...]を選択すると、[カバレジの設定]ダイアログボックス(図 3-45)を表示します。[カバレジの設定]ダイアログボックスについては、「3.8.1 カバレジウィンドウを開く」を参照してください。

### 3.8.6 カバレジ測定範囲を変更する

- (1) カバレジ測定範囲をアドレスで指定した場合

カバレジ測定範囲を選択し、ポップアップメニューで[測定範囲編集...]を選択すると、[カバレジ測定範囲]ダイアログボックスを表示します。

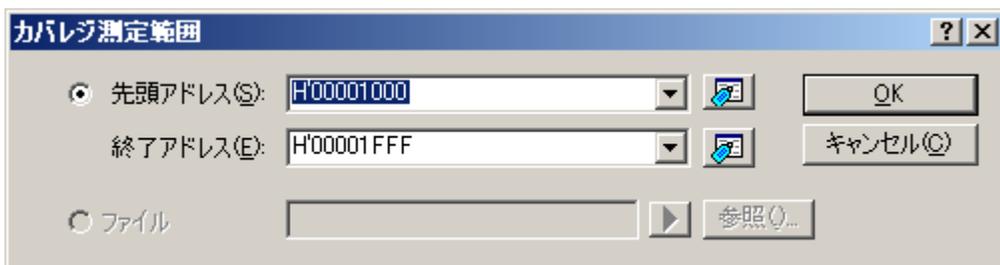


図 3-48 カバレジ測定範囲ダイアログボックス(アドレス指定)

### 3. デバッグ

本ダイアログボックスで命令実行情報取得の条件を変更します。下記項目を指定できます。

[先頭アドレス] 先頭アドレス（接頭辞省略時は 16 進で入力）

[終了アドレス] 終了アドレス（接頭辞省略時は 16 進で入力）

[OK]ボタンをクリックするとカバレッジ測定範囲を変更します。

#### (2) カバレッジ測定範囲をソースファイルで指定した場合

ポップアップメニューで[測定範囲編集...]を選択すると、[カバレッジ測定範囲]ダイアログボックスを表示します。



図 3-49 カバレッジ測定範囲ダイアログボックス(ファイル指定)

本ダイアログボックスで命令実行情報取得の条件を変更します。下記項目を指定できます。

[ファイル] 現在のプロジェクト中の".C"または".CPP"を型名にもつソースファイルを指定します。これにより指定ファイル内に存在する関数をカバレッジ範囲として設定します。ファイル型名を省略した場合は、".C"を補います。".C"または".CPP"以外のファイル型名を持つファイルは指定できません。ブレースホルダまたは[参照...]ボタンが利用できます。

[OK]ボタンをクリックするとカバレッジ測定範囲を変更します。

#### 3.8.7 カバレッジ測定範囲を削除する

カバレッジ測定範囲を選択し、ポップアップメニューから[カバレッジ測定範囲削除]を選択すると、選択したカバレッジ測定範囲を削除します。

#### 3.8.8 カバレッジ情報を取得する

カバレッジ測定範囲を選択し、ポップアップメニューから[有効]をチェック後、命令を実行するとカバレッジ情報を取得します。デフォルトで[有効]はチェックされています。

#### 3.8.9 カバレッジ情報をクリアする

カバレッジ測定範囲を選択し、ポップアップメニューから[クリア]を選択すると、取得したカバレッジ情報をクリアします。

### 3.8.10 カバレジ情報をファイルに保存する

ポップアップメニューから[保存...]を選択すると、カバレジ情報をファイルに保存するための[カバレジ情報を保存]ダイアログボックスを表示します。



図 3-50 カバレジ情報を保存ダイアログボックス

保存するカバレジ情報ファイルの場所と名前を指定します。プレースホルダまたは[参照...]ボタンが使用できます。

ファイル拡張子の入力を省略すると、ファイル拡張子として".COV"を自動的に付加します。ファイル拡張子として、".COV"または".TXT"以外を入力するとエラーメッセージを出力します。

### 3.8.11 カバレジ情報をファイルからロードする

ポップアップメニューから[ロード...]を選択すると、カバレジ情報をファイルからロードするための[カバレジ情報ロード]ダイアログボックスを表示します。

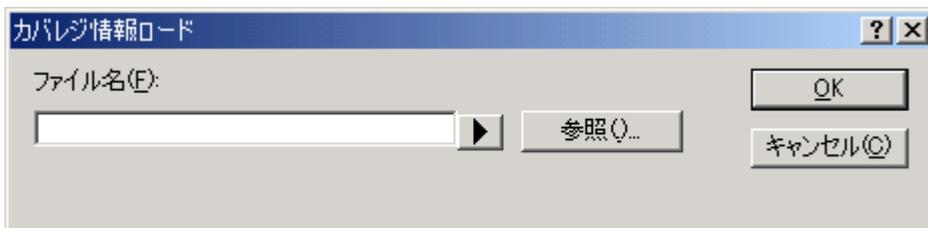


図 3-51 カバレジ情報ロードダイアログボックス

ロードするカバレジ情報ファイルの場所と名前を指定します。プレースホルダまたは[参照...]ボタンが使用できます。

ロードできるファイル拡張子は".COV"のみです。その他のファイル拡張子を入力するとエラーメッセージを出力します。

### 3.8.12 最新の情報に更新する

ポップアップメニューから[最新の情報に更新]を選択すると、[カバレジ]ウィンドウ内容を最新に更新します。

### 3.8.13 Confirmation Request ダイアログボックス

[すべてクリア]、[クリア]、[測定範囲編集...]、[測定範囲削除]をクリックするか、[カバレジ]ウィンドウを閉じようとする、確認のダイアログボックスを表示します。



図 3-52 Confirmation Request ダイアログボックス

[OK]ボタンをクリックすると、カバレジデータをクリアします。  
[カバレジ情報を保存]をチェックすると、「図 3-50」に示す[カバレジ情報を保存]ダイアログボックスを表示し、クリアする前にカバレジデータをファイルに保存できます。

#### 3.8.14 カバレジ情報を保存ダイアログボックス

[ファイル->セッションの保存]メニューオプションをクリックすると、[カバレジ情報を保存]ダイアログボックスを表示します。[カバレジ]ウィンドウのデータを別々またはまとめて保存することができます。

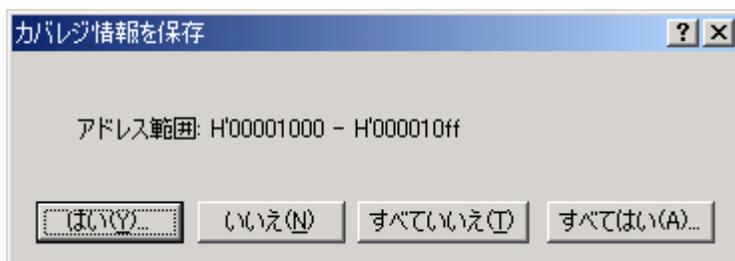


図 3-53 カバレジ情報を保存ダイアログボックス

- [カバレジ]ウィンドウが複数開いているとウィンドウ数分の[カバレジ情報を保存]ダイアログボックスが開きます。
- [すべていいえ]ボタンをクリックすると、すべてのカバレジ情報を保存しないでダイアログボックスを閉じます。
- [すべてはい]ボタンをクリックすると、すべての[カバレジ]ウィンドウデータを1個のファイルに保存します。

【注】カバレジ範囲としてファイルを指定した場合は、すべての[カバレジ]ウィンドウデータを1個のファイルに保存することはできません。

#### 3.8.15 エディタウィンドウへのカバレジ結果表示

命令実行済のソース行に対応するカバレジカラムを強調表示することで[エディタ]ウィンドウにもカバレジ結果を表示します。[カバレジ]ウィンドウでカバレジに関する設定を変更すると、対応するカバレジカラムの表示も更新します。

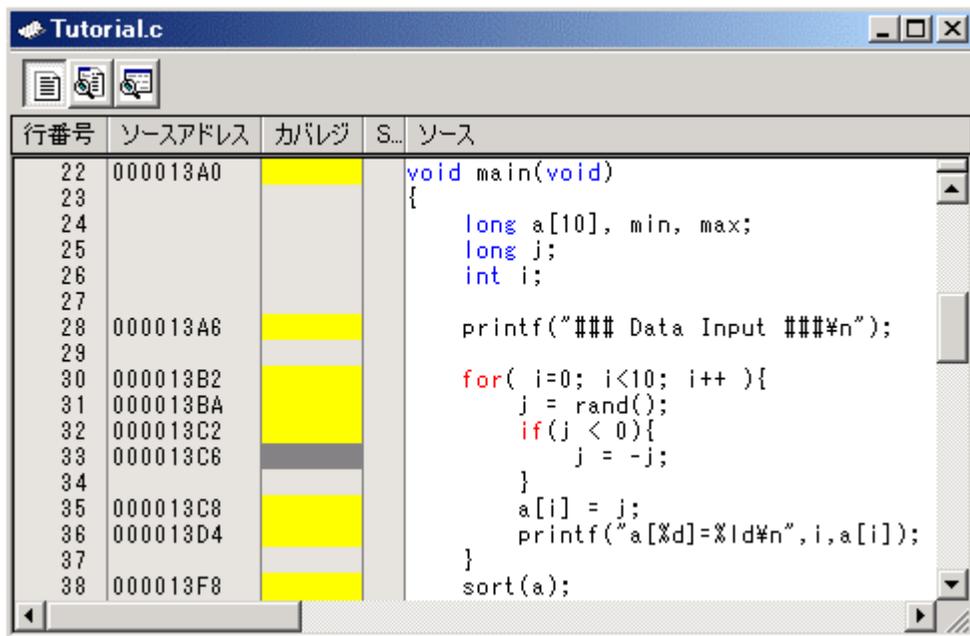


図 3-54 カバレジカラム (ソース)

### 3.8.16 逆アセンブリウィンドウへのカバレジ結果表示

命令実行済の逆アセンブリ行に対応するカバレジ-ASM カラムを強調表示することで[逆アセンブリ]ウィンドウにもカバレジ結果を表示します。[カバレジ]ウィンドウでカバレジに関する設定を変更すると、対応するカバレジ-ASM カラムの表示も更新します。



図 3-55 カバレジカラム (逆アセンブリ)

## 3.9 手動で擬似割込みを発生させる

ウィンドウ上のボタンを押下して手動で擬似割込みを発生させるには、[トリガ]ウィンドウ、または[GUI I/O]ウィンドウを使用します。

### 3.9.1 トリガウィンドウ

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



図 3-56 トリガウィンドウ

本ウィンドウは、手動で擬似割込みを発生させるためのトリガボタンを表示します。トリガボタン押下時に発生する擬似割込みの内容は、[トリガ設定]ダイアログボックスで設定します。

トリガボタンは最大 256 個まで設定できます。

シミュレータ・デバッガの擬似割込み処理については、「2.20

「擬似割り込み」を参照してください。

(1) トリガボタンを設定する

各トリガボタン押下時に発生する擬似割り込みの内容を設定するには[トリガ設定]ダイアログボックスを使用します。

[トリガ設定]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[設定...]を選択します。

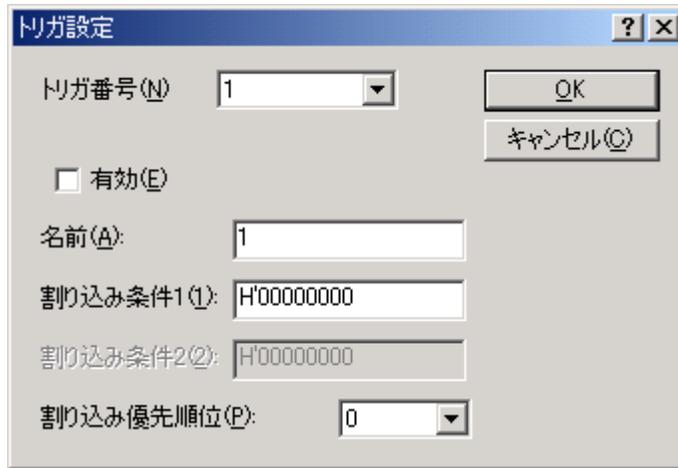


図 3-57 トリガ設定ダイアログボックス

本ダイアログボックスでは、トリガボタン押下時に発生する割り込みの内容を設定します。

- |            |  |
|------------|--|
| [トリガ番号]    | 設定するトリガボタンを選択します。  |
| [名前]       | [トリガ]ウィンドウに表示するトリガボタンの名前を指定します。  |
| [有効]       | チェックするとトリガボタンが有効になります。   |
| [割り込み条件 1] | CPU ごとに下記を指定します。<br><ul style="list-style-type: none"> <li>・ SH-1、SH-2、SH2-DSP、SH2A-FPU シリーズ<br/>割り込みベクタ番号</li> <li>・ SH-3、SH-4、SH3-DSP シリーズ<br/>INTEVT 値 (0-H'FFF)</li> <li>・ SH-4A シリーズ<br/>INTEVT 値 (0-H'3FFF)</li> </ul> |
| [割り込み条件 2] | SH3-DSP シリーズのみ下記を指定可能です。<br>INTEVT2 値 (0-H'FFF)  |
| [割り込み優先順位] | 割り込み優先順位を指定します。(接頭辞省略時は 16 進入力、16 進表示) (0~H'11)<br>H'10 指定時は、SR の I ビットによらず割り込みが発生しますが、SR の BL によってマスクします。<br>H'11 指定時は、SR の I ビット、BL ビットによらず、割り込みが発生します。  |

### 3. デバッグ

---

指定した内容は、[OK]ボタンをクリックすることにより設定します。[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。

**【注】** 複数のトリガボタンの設定を変更した後に[キャンセル]ボタンをクリックすると、すべてのトリガボタンの設定変更が無効になります。

#### (2) トリガボタンの数を変える

[トリガ]ウィンドウに表示するトリガボタンの数を変えるには、ポップアップメニューから[ボタンの数]のサブメニューで[4]、[16]、[64]、[256]を選択します。

#### (3) トリガボタンのサイズを変える

[トリガ]ウィンドウに表示するトリガボタンのサイズを変えるには、ポップアップメニューから[サイズ]のサブメニューで[大]、[中]、[小]を選択します。

## 3.9.2 GUI I/O ウィンドウ

[GUI I/O]ウィンドウを開くには、[表示->グラフィック->GUI I/O]を選択するか、[GUI I/O]ツールバーボタンをクリックします。

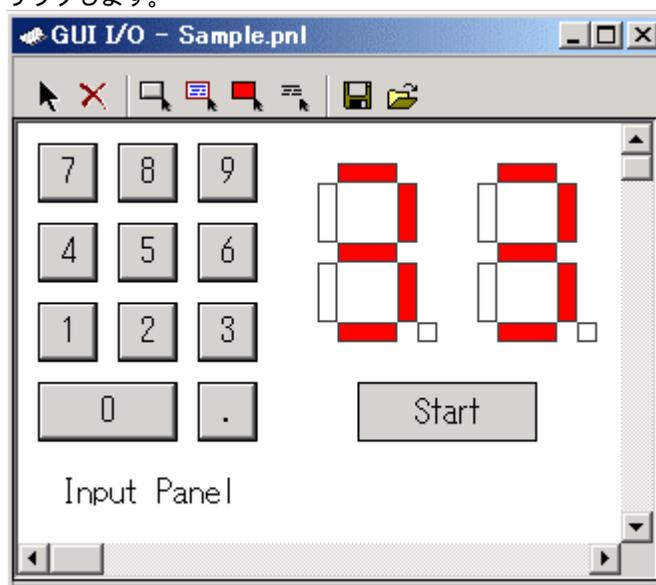


図 3-58 GUI I/O ウィンドウ

本ウィンドウは、手動で擬似割り込みを発生させるためのボタンを設定します。ボタン押下時に発生する擬似割り込みの内容は、[ボタンの設定]ダイアログボックスで設定します。

シミュレータ・デバッガの擬似割り込み処理については、「2.20

「擬似割り込み」を参照してください。

#### (1) ボタンを設定する

擬似割り込みを発生させるボタンを設定するには[ボタンの設定]ダイアログボックスを使用します。まずポップアップメニューの[ボタンの作成]を選択するか、ツールバーの[ボタンの作成]ボタンをクリックしてください。マウスカソールの形状が" + "に変化します。この状態で作成するボタンの左上位置から右下位置までドラッグしてください。

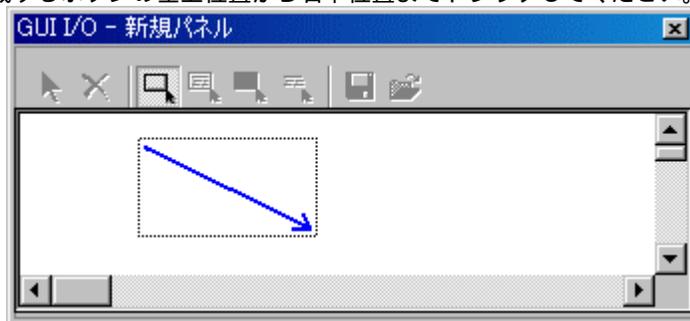


図 3-59 GUI I/O ウィンドウ (ボタンの作成)

次にこのボタンをダブルクリックして[ボタンの設定]ダイアログボックスを開きます。



図 3-60 ボタンの設定ダイアログボックス

### 3. デバッグ

---

本ダイアログボックスでは、ボタン押下時に発生する割り込みの内容を設定します。

[ボタン名]	[GUI I/O]ウィンドウに表示するボタンの名前を指定します。
[ボタン種別の選択]	[割り込み用]、または[入力と割り込みの両用]を選択します。
[割り込み]    [割り込み条件 1]	CPU ごとに下記を指定します。 <ul style="list-style-type: none"><li>・ SH-1、SH-2、SH2-DSP、SH2A-FPU シリーズ   割り込みベクタ番号</li><li>・ SH-3、SH-4、SH3-DSP シリーズ   INTEVT 値 (0~H'FFF)</li><li>・ SH-4A シリーズ   INTEVT 値 (0~H'3FFF)</li></ul>
[割り込み条件 2]	SH3-DSP シリーズのみ下記を指定可能です。 INTEVT2 値 (0~H'FFF)
[割り込み優先順位]	割り込み優先順位を指定します。(接頭辞省略時は 16 進入力、16 進表示) (0~H'11) H'10 指定時は、SR の I ビットによらず割り込みが発生しますが、SR の BL によってマスクします。 H'11 指定時は、SR の I ビット、BL ビットによらず、割り込みが発生します。

## 3.10 標準入出力およびファイル入出力を行う

ユーザプログラムから標準入出力およびファイル入出力の I/O シミュレーションを行うには、[I/O シミュレーション]ウィンドウを利用します。

### 3.10.1 I/O シミュレーションウィンドウを開く

[I/O シミュレーション]ウィンドウを開くには、[表示->CPU->I/O シミュレーション]を選択するか、[I/O シミュレーション]ツールバーボタン  をクリックします。

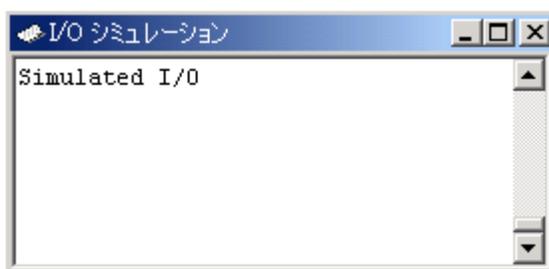


図 3-61 I/O シミュレーションウィンドウ

ユーザプログラムからの標準出力は本ウィンドウに出力されます。本ウィンドウ上でのキー入力がユーザプログラムへの標準入力となります。

### 3.10.2 入出力機能

サポートしている入出力処理は、以下の通りです。

表3-1 入出力機能一覧

番号	機能コード	機能名	内容
1	H'21	GETC	標準入力からの1バイト入力
2	H'22	PUTC	標準出力への1バイト出力
3	H'23	GETS	標準入力からの1行入力
4	H'24	PUTS	標準出力への1行出力
5	H'25	FOPEN	ファイルのオープン
6	H'06	FCLOSE	ファイルのクローズ
7	H'27	FGETC	ファイルからの1バイト入力
8	H'28	FPUTC	ファイルへの1バイト出力
9	H'29	FGETS	ファイルからの1行入力
10	H'2A	FPUTS	ファイルへの1行出力
11	H'0B	FEOF	エンドオブファイルのチェック
12	H'0C	FSEEK	ファイルポインタの移動
13	H'0D	FTELL	ファイルポインタの現在位置を得る

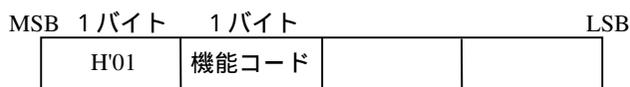
### 3. デバッグ

---

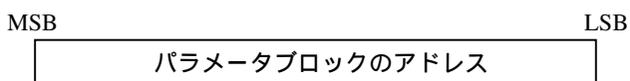
この機能を実現するためには、まず入出力用の特定の位置を[シミュレータシステム]ダイアログボックス(3.2.1 参照)の [I/O シミュレーションアドレス]で指定し、[有効]をチェック後、ユーザプログラムを実行します。シミュレータ・デバッガでは、ユーザプログラムの命令を実行中に、指定した位置へのサブルーチン分岐命令 (BSR、JSR、BSRF) すなわち I/O シミュレーション命令を検出すると、R0、R1 の内容をパラメータとして入出力処理を行います。

したがって、I/O シミュレーションを行う前にユーザプログラムの中で次の設定をしておきます。

- ・ R0 レジスタ：表 3-1に示す機能コード



- ・ R1 レジスタ：パラメータブロックのアドレス  
(パラメータブロックの内容は各機能の説明を参照してください。)



- ・ パラメータブロックおよび入出力バッファ領域の確保

なお、パラメータブロックの各パラメータにアクセスする場合は、該当するパラメータのサイズでアクセスしてください。

入出力処理が終了すると、I/O シミュレーション命令の次の命令からシミュレーションを再開します。

**【注】** JSR、BSR、BSRF 命令を I/O シミュレーション命令として実行すると、JSR、BSR、BSRF の次命令はスロット命令ではなく、通常の命令として実行します。このため、スロット命令と通常命令で実行結果に違いが発生する命令を I/O シミュレーション用 JSR、BSR、BSRF 命令の次命令には、記述しないでください。

各入出力機能の詳細は、シミュレータヘルプを参照ください。

以下に標準入力 (キーボード) から 1 文字入力する例を示します。

```
MOV.L   PAR_ADR, R1
MOV.L   REQ_COD, R0
MOV.L   CALL_ADR, R3
JSR     @R3
NOP
STOP    NOP
SYS_CALL  NOP
.ALIGN  4
CALL_ADR .DATA.L SYS_CALL
REQ_COD  .DATA.L  H'01210000
PAR_ADR  .DATA.L  PARM
PARM     .DATA.L  INBUF
INBUF    .RES.B   2
.END
```

## 3.11 TLB 内容を見る

MMU(Memory Management Unit)機能を持つ CPU で、TLB 情報を表示するには[TLB]ウィンドウを使用します。[TLB]ウィンドウは CPU によって異なります。お使いの CPU に合わせてお読みください。

### 3.11.1 TLB ウィンドウを開く

#### (1) SH-3/SH-3E/SH3-DSP シリーズ

[表示->CPU->TLB...]を選択するか、[TLB]ツールバーボタンをクリックすると、[TLB]ウィンドウが開きます。

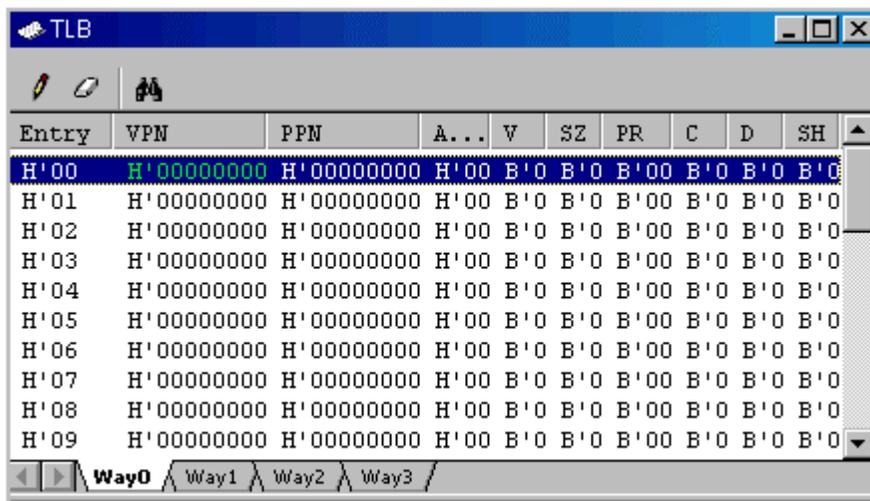


図 3-62 TLB ウィンドウ(SH-3/SH-3E/SH3-DSP シリーズ)

本ウィンドウは、TLB 情報を表示します。選択されている項目を緑色で表示します。各ウェイごとに1つのシートを持っており、表示したいウェイのタブをクリックすると、その内容を表示します。現在無効になっているウェイがある場合、その内容はグレーで表示します。

ウィンドウに表示する項目は以下の通りです。

[Entry]	TLB のエントリ番号(H'00 ~ H'FF)
[VPN]	論理ページ番号(ロングワードサイズ)を表示します。ビット 31~10 が有効となります。
[PPN]	物理ページ番号(ロングワードサイズ)を表示します。ビット 31~10 が有効となります。
[ASID]	アドレス空間識別子を表示します。
[V]	有効ビットを表示します。
[SZ]	ページサイズビットを表示します。
[PR]	保護キーデータを表示します。
[C]	キャッシング可能ビットを表示します。
[D]	ダーティビットを表示します。
[SH]	共有状態ビットを表示します。

### 3. デバッグ

---

#### (2) SH-4 シリーズ、SH-4A シリーズ

SH-4 シリーズ、SH-4A シリーズには、命令 TLB 用、共用 TLB 用の 2 種類の TLB ウィンドウがあります。さらに、SH-4A には、PMB 用の TLB ウィンドウがあります。

[表示->CPU->TLB...]を選択するか、[TLB]ツールバーボタンをクリックすると、表示する TLB を選択するための[TLB の選択]ダイアログボックス(図 3-63参照)を表示します。



図 3-63 TLB の選択ダイアログボックス

本ダイアログボックスでは、表示したい TLB を選択します。以下の TLB 種別より選択します。

[命令 TLB]	命令 TLB ウィンドウを開きます。
[共用 TLB]	共用 TLB ウィンドウを開きます。
[PMB]	PMB ウィンドウを開きます。(SH-4A のみ)

[OK]ボタンをクリックすることにより、選択した TLB のウィンドウが開きます。

[キャンセル]ボタンは、[TLB]ウィンドウを開かずにダイアログボックスを閉じます。

## (a) 命令 TLB ウィンドウ

Entry	VPN	PPN	ASID	V	SZ	PR	C	SH
H'0	H'00000000	H'00000000	H'00	B'0	B'0	B'0	B'0	B'0
H'1	H'00000000	H'00000000	H'00	B'0	B'0	B'0	B'0	B'0
H'2	H'00000000	H'00000000	H'00	B'0	B'0	B'0	B'0	B'0
H'3	H'00000000	H'00000000	H'00	B'0	B'0	B'0	B'0	B'0

図 3-64 命令 TLB ウィンドウ(SH-4 シリーズ)

本ウィンドウは、命令 TLB 情報を表示します。選択されている項目を緑色で表示します。ウィンドウに表示する項目は以下の通りです。

[Entry]	命令 TLB のエントリ番号(H'0~H'3)
[VPN]	論理ページ番号(ロングワードサイズ)を表示します。ビット 31~10 が有効となります。
[PPN]	物理ページ番号(ロングワードサイズ)を表示します。ビット 31~10 が有効となります。
[ASID]	アドレス空間識別子を表示します。
[V]	有効ビットを表示します。
[SZ]	ページサイズビットを表示します。
[PR]	保護キーデータを表示します。
[C]	キャッシング可能ビットを表示します。
[SH]	共有状態ビットを表示します。

## (b) 共用 TLB ウィンドウ

Entry	VPN	PPN	ASID	V	SZ	PR	C	D	SH	WT
H'00	H'00000000	H'00000000	H'00	B'0						
H'01	H'00000000	H'00000000	H'00	B'0						
H'02	H'00000000	H'00000000	H'00	B'0						
H'03	H'00000000	H'00000000	H'00	B'0						
H'04	H'00000000	H'00000000	H'00	B'0						
H'05	H'00000000	H'00000000	H'00	B'0						
H'06	H'00000000	H'00000000	H'00	B'0						
H'07	H'00000000	H'00000000	H'00	B'0						
H'08	H'00000000	H'00000000	H'00	B'0						
H'09	H'00000000	H'00000000	H'00	B'0						
H'0A	H'00000000	H'00000000	H'00	B'0						

図 3-65 共用 TLB ウィンドウ(SH-4 シリーズ)

本ウィンドウは、共用 TLB 情報を表示します。選択されている項目を緑色で表示します。ウィンドウに表示する項目は以下の通りです。

### 3. デバッグ

---

[Entry]	共用 TLB のエントリ番号(H'00 ~ H'3F)
[VPN]	論理ページ番号(ロングワードサイズ)を表示します。ビット 31~10 が有効となります。
[PPN]	物理ページ番号(ロングワードサイズ)を表示します。ビット 31~10 が有効となります。
[ASID]	アドレス空間識別子を表示します。
[V]	有効ビットを表示します。
[SZ]	ページサイズビットを表示します。
[PR]	保護キーデータを表示します。
[C]	キャッシング可能ビットを表示します。
[D]	ダーティビットを表示します。
[SH]	共有状態ビットを表示します。
[WT]	ライトスルービットを表示します。

## (c) PMB ウィンドウ

Entry	VPN	PPN	V	SZ	C	WT	UB
H'0	H'00000000	H'00000000	B'0	B'00	B'0	B'0	B'0
H'1	H'00000000	H'00000000	B'0	B'00	B'0	B'0	B'0
H'2	H'00000000	H'00000000	B'0	B'00	B'0	B'0	B'0
H'3	H'00000000	H'00000000	B'0	B'00	B'0	B'0	B'0
H'4	H'00000000	H'00000000	B'0	B'00	B'0	B'0	B'0
H'5	H'00000000	H'00000000	B'0	B'00	B'0	B'0	B'0
H'6	H'00000000	H'00000000	B'0	B'00	B'0	B'0	B'0
H'7	H'00000000	H'00000000	B'0	B'00	B'0	B'0	B'0
H'8	H'00000000	H'00000000	B'0	B'00	B'0	B'0	B'0
H'9	H'00000000	H'00000000	B'0	B'00	B'0	B'0	B'0
H'A	H'00000000	H'00000000	B'0	B'00	B'0	B'0	B'0

図 3-66 PMB ウィンドウ(SH-4A)

SH-4Aのみサポートしている本ウィンドウは、PMB情報を表示します。選択されている項目を緑色で表示します。

ウィンドウに表示する項目は以下の通りです。

[Entry]	PMBのエントリ番号(H'0~H'F)
[VPN]	論理ページ番号(ロングワードサイズ)を表示します。ビット31~24が有効となります。
[PPN]	物理ページ番号(ロングワードサイズ)を表示します。ビット31~24が有効となります。
[V]	有効ビットを表示します。
[SZ]	ページサイズビットを表示します。
[C]	キャッシング可能ビットを表示します。
[WT]	ライトスルービットを表示します。
[UB]	バッファドライトするかどうかを表示します。

【注】製品によってはPMBを内蔵しない場合があります。

#### 3.11.2 TLB 内容を変更する

TLB 項目を選択した状態でポップアップメニューから[変更...]を選択すると、TLB 内容を変更するためのダイアログボックスが開きます。

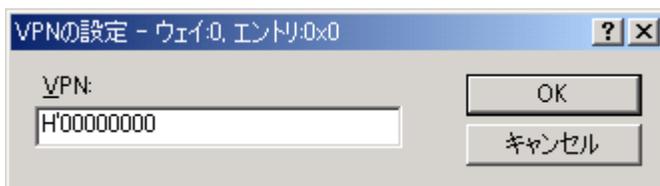


図 3-67 TLB 項目の設定ダイアログボックス(SH-3 で VPN を選択した場合)

本ダイアログボックスでは、現在選択されている TLB 項目を変更できます。ダイアログボックスのキャプションには、選択した項目名とエントリ番号を表示します。ウェイを持つ TLB の場合は、ウェイ番号も表示します。ダイアログボックス内の項目名には、選択した内容の項目名を表示します。

[OK]ボタンをクリックすることにより、設定した内容を TLB に反映します。[キャンセル]ボタンをクリックすると、TLB に反映しないでダイアログボックスを閉じます。

また、ウィンドウに対して値を直接入力することもできます。

#### 3.11.3 TLB をフラッシュする

ポップアップメニューから[フラッシュ]を選択すると、TLB をフラッシュします。すべての Valid bit が 0 になり、全 TLB を無効化できます。

なお、命令実行中は[フラッシュ]メニューが無効になり、TLB のフラッシュはできません。

#### 3.11.4 TLB 項目を検索する

ポップアップメニューから[検索...]を選択すると、[TLB の検索]ダイアログボックスが開き、コラム単位で TLB 項目を検索できます。

なお、命令実行中は[検索...]メニューが無効になり、TLB 項目の検索はできません。

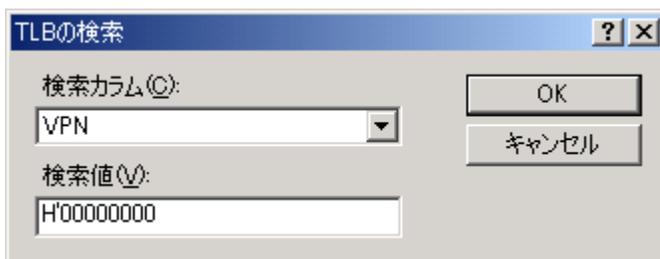


図 3-68 TLB の検索ダイアログボックス

本ダイアログボックスは、TLB の検索を行います。検索する条件を次のように設定します。

[検索カラム]      検索する項目を指定します。  
[検索値]            検索する値を指定します。

条件を設定後、[OK]ボタンをクリックすることにより検索を行います。検索した結果、一致するエントリを強調表示します。

[キャンセル]ボタンをクリックすると検索を実行しないでダイアログボックスを閉じます。

### 3.11.5 次を検索する

前回の検索条件で、次に一致する TLB 項目を検索できます。前回の検索で一致する TLB 内容が見つかった状態で、ポップアップメニューから[次を検索]を選択します。

なお、命令実行中は[次を検索]メニューが無効になり、次に一致する TLB 項目の検索はできません。

### 3.11.6 現在の表示内容をセーブする

現在ウィンドウに表示している内容をテキストファイルにセーブすることが出来ます。ポップアップメニューから[ファイルに保存...]を選択してください。

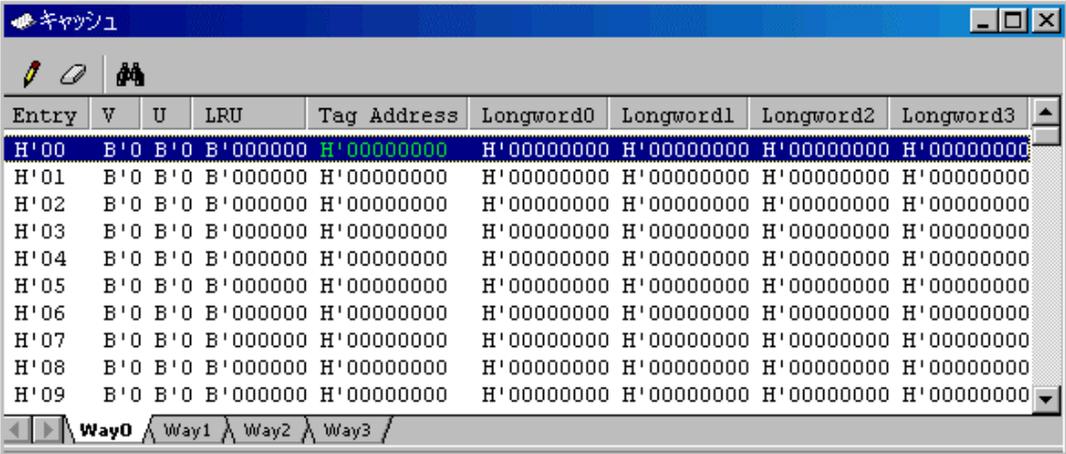
## 3.12 キャッシュ内容を見る

キャッシュ機能を持つ CPU で、キャッシュ内容を見るにはキャッシュウィンドウを使用します。キャッシュウィンドウの内容は CPU により異なります。お使いの CPU に合わせてお読みください。

### 3.12.1 キャッシュウィンドウを開く

(1) SH-3/SH-3E/SH3-DSP/SH2-DSP(SH7612)シリーズ

[表示->CPU->キャッシュ...]を選択するか、[キャッシュ]ツールバーボタンをクリックすると、キャッシュウィンドウが開きます。



Entry	V	U	LRU	Tag Address	Longword0	Longword1	Longword2	Longword3
H'00	B'0	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'01	B'0	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'02	B'0	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'03	B'0	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'04	B'0	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'05	B'0	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'06	B'0	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'07	B'0	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'08	B'0	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'09	B'0	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000

図 3-69 キャッシュウィンドウ (SH-3/SH-3E、SH2-DSP(SH7612)シリーズ)

### 3. デバッグ

本ウィンドウは、キャッシュ情報を表示します。選択されている項目を緑色で表示します。各ウェイトごとに1つのシートをもっており、表示したいウェイトのタブをクリックすると、その内容を表示します。現在無効になっているウェイトがある場合、その内容はグレーで表示します。

キャッシュウィンドウに表示する項目は以下の通りです。

[Entry]	キャッシュのエントリ番号。エントリ数はCPUにより異なります。 ・ SH-3/SH-3E シリーズ H'00~H'7F ・ SH3-DSP シリーズ H'00~H'FF ・ SH2-DSP(SH7612) H'00~H'3F
[V]	有効ビット。1 のとき有効です。
[U]	書き込みビット。1 のときは、書き込んだことを示します
[LRU]	キャッシュミス発生時にどのウェイトのエントリを入れ替えるかを決定する数字列（同一エントリに対する LRU 値は各ウェイト共通です。）
[Tag Address]	タグアドレス
[Longword0] ~ [Longword3]	キャッシュエントリに設定しているロングワードデータ 0 ~ 3

#### (2) SH-4 シリーズ、SH-4A シリーズ、SH2A-FPU

SH2A-FPU、SH-4 シリーズ、SH-4A シリーズには、命令キャッシュ用、オペランドキャッシュ用の2種類のキャッシュウィンドウがあります。さらに SH-4A には、2次キャッシュ用のキャッシュウィンドウ、SH2A-FPU には ROM キャッシュ用のキャッシュウィンドウがあります。

[表示->CPU->キャッシュ...]を選択するか、[キャッシュ]ツールバーボタンをクリックすると、表示するキャッシュを選択するための[キャッシュの選択]ダイアログボックス(図 3-70参照)を表示します。



図 3-70 キャッシュの選択ダイアログボックス

本ダイアログボックスでは、表示したいキャッシュを選択します。

以下のキャッシュ種別より選択します。

[命令キャッシュ]	命令キャッシュウィンドウを開きます。 <sup>1</sup>
[オペランドキャッシュ]	オペランドキャッシュウィンドウを開きます。 <sup>1</sup>
[2次キャッシュ]	2次キャッシュウィンドウを開きます。(SH-4Aのみ)
[ROMプリフェッチキャッシュ]	ROMプリフェッチキャッシュウィンドウを開きます。(SH2A-FPUのみ) <sup>2</sup>
[ROMプリフェッチミスキャッシュ]	ROMプリフェッチミスキャッシュウィンドウを開きます。(SH2A-FPUのみ) <sup>2</sup>
[ROMオペランドキャッシュ]	ROMオペランドキャッシュウィンドウを開きます。(SH2A-FPUのみ) <sup>2</sup>

\*1:SH2A-FPUシリーズでは内蔵ROM無効時のみ表示します。

\*2:内蔵ROM有効時のみ表示します。

SH2A-FPUの内蔵ROM有効/無効指定方法については、「3.1.2 シミュレータ用ワークスペースの設定」を参照してください。

[OK]ボタンをクリックすることにより、選択したキャッシュのウィンドウが開きます。

[キャンセル]ボタンをクリックすると、キャッシュウィンドウを開かないでダイアログボックスを閉じます。

#### (a) 命令キャッシュウィンドウ

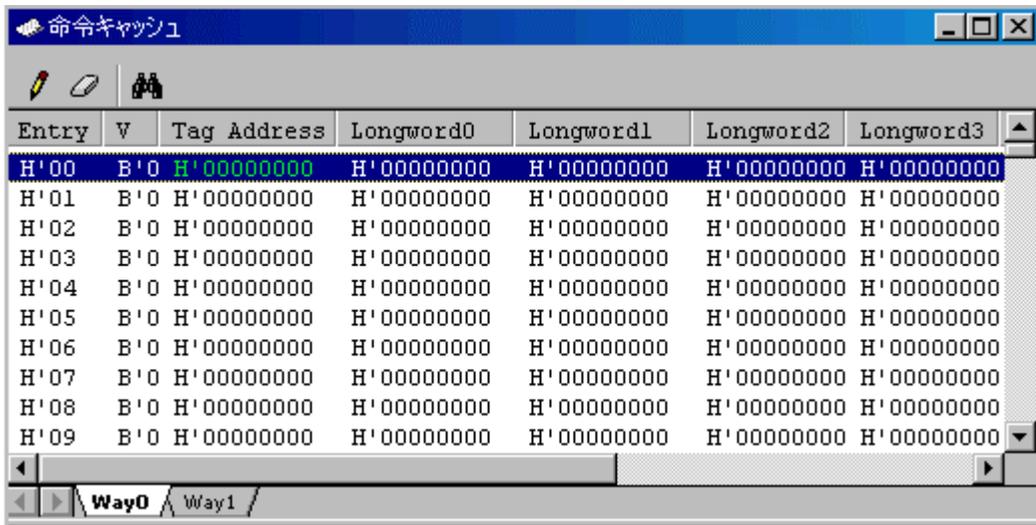


図 3-71 命令キャッシュウィンドウ(SH-4(SH7750R))

本ウィンドウは、命令キャッシュ情報を表示します。選択されている項目を緑色で表示します。各ウェイごとに1つのシートをもっており、表示したいウェイのタブをクリックすると、その内容を表示します。現在無効になっているウェイがある場合、その内容はグレーで表示します。

(SH-4、SH-4BSCは1ウェイのみです)

命令キャッシュウィンドウに表示する項目は以下の通りです。

[Entry]	命令キャッシュのエントリ番号(キャッシュ容量によって異なります)
[V]	有効ビット。1のとき有効です。
[LRU]	キャッシュミス発生時にどのウェイのエントリを入れ替えるかを決定する数字列(同一エントリに対するLRU値は各ウェイ共通です。)
[Tag Address]	タグアドレス
[Longword0]~ [Longword7]	命令キャッシュエントリに設定しているロングワードデータ0~7

【注】SH2A-FPUは、[Longword0]~[Longword3]

### 3. デバッグ

#### (b) オペランドキャッシュウィンドウ

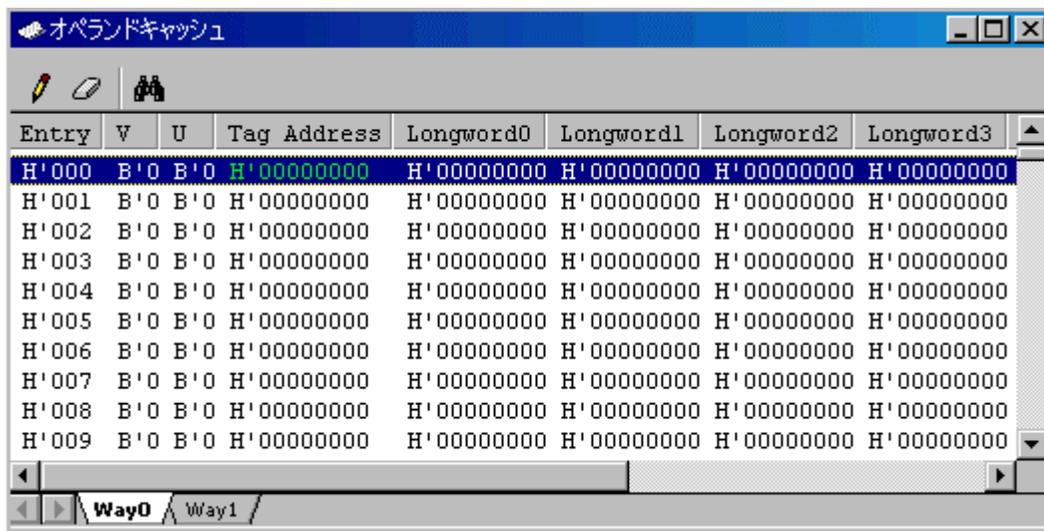


図 3-72 オペランドキャッシュウィンドウ(SH-4(SH7750R))

本ウィンドウは、オペランドキャッシュ情報を表示します。選択されている項目を緑色で表示します。各ウェイごとに1つのシートをもっており、表示したいウェイのタブをクリックすると、その内容を表示します。現在無効になっているウェイがある場合、その内容はグレーで表示します。(SH-4、SH-4BSCは1ウェイのみです)

オペランドキャッシュウィンドウに表示する項目は以下の通りです。

- [Entry] オペランドキャッシュのエントリ番号 (キャッシュ容量によって異なります)
  - [V] 有効ビット。1のとき有効です。
  - [U] 書き込みビット。1のときは、書き込んだことを示します。
  - [LRU] キャッシュミス発生時にどのウェイのエントリを入れ替えるかを決定する数字列 (同一エントリに対するLRU値は各ウェイ共通です。)
  - [Tag Address] タグアドレス。
  - [Longword0] ~ [Longword7] オペランドキャッシュエントリに設定しているロングワードデータ0~7
- 【注】 SH2A-FPUは、[Longword0] ~ [Longword3]

## (c) 2次キャッシュウィンドウ

Entry	V	LRU	Tag Address	Longword0	Longword1	Longword2	Longword3
H'0000	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'0001	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'0002	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'0003	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'0004	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'0005	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'0006	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'0007	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'0008	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'0009	B'0	B'000000	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000

図 3-73 2次キャッシュウィンドウ (SH-4A)

SH-4Aのみサポートしている本ウィンドウは、2次キャッシュ情報を表示します。選択されている項目を緑色で表示します。

2次キャッシュウィンドウに表示する項目は以下の通りです。

[Entry]	2次キャッシュのエントリ番号。(キャッシュ容量によって異なります)
[V]	有効ビット。1のとき有効です。
[LRU]	キャッシュミス発生時にどのウェイのエントリを入れ替えるかを決定する数字列(同一エントリに対するLRU値は各ウェイ共通です。)
[Tag Address]	タグアドレス。
[Longword0]~ [Longword7]	エントリに設定しているロングワードデータ0~7

【注】製品によっては2次キャッシュを内蔵しない場合があります。

## (d) ROM キャッシュウィンドウ

Entry	V	LRU	Tag Address	Longword0	Longword1	Longword2	Longword3
H'0	B'0	B'111111	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000
H'1	B'0	B'111111	H'00000000	H'00000000	H'00000000	H'00000000	H'00000000

図 3-74 ROM キャッシュウィンドウ(プリフェッチキャッシュ)

SH2A-FPUのみサポートしている本ウィンドウは、ROMキャッシュ情報を表示します。ROMキャッ

### 3. デバッグ

---

シュウインドウにはROMプリフェッチキャッシュウインドウ、ROMプリフェッチミスウインドウ、およびROMオペランドキャッシュウインドウの3種類があります。各ROMキャッシュウインドウの表示項目は同じです。選択されている項目を緑色で表示します。

ROMキャッシュウインドウに表示する項目は以下の通りです。

[Entry]	ROM キャッシュのエントリ番号。
[V]	有効ビット。1 のとき有効です。
[LRU]	キャッシュミス発生時にどのウェイのエントリを入れ替えるかを決定する数字列（同一エントリに対する LRU 値は各ウェイ共通です。）
[Tag Address]	タグアドレス。
[Longword0]~ [Longword3]	エントリに設定しているロングワードデータ 0~3

#### 3.12.2 キャッシュ内容を変更する

キャッシュ項目を選択した状態で、ポップアップメニューから[変更...]を選択するとキャッシュ項目の変更ダイアログボックスが開き、キャッシュ項目の内容を変更することができます。

なお、命令実行中は[変更...]メニューが無効になり、キャッシュ項目の変更はできません。

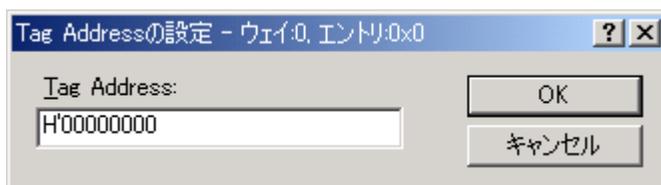


図 3-75 キャッシュ項目の設定ダイアログボックス(SH-3 で Tag Address を選択した場合)

本ダイアログボックスでは、現在選択されているキャッシュ項目を変更できます。ダイアログボックスのキャプションには、選択した項目名とエントリ番号を表示します。ウェイを持つキャッシュの場合は、ウェイ番号も表示します。ダイアログボックス内の項目名には、選択した内容の項目名を表示します。

[OK]ボタンをクリックすることにより、設定した内容をキャッシュに反映します。[キャンセル]ボタンをクリックすると、キャッシュに反映しないでダイアログボックスを閉じます。

また、ウインドウに対して値を直接入力することもできます。

#### 3.12.3 キャッシュ内容をフラッシュする

ポップアップメニューから[フラッシュ]を選択すると、キャッシュをフラッシュします。すべての V、U、LRU が 0 になり、全キャッシュを無効化できます。

なお、命令実行中は[フラッシュ]メニューが無効になり、キャッシュのフラッシュはできません。

#### 3.12.4 キャッシュ内容を検索する

ポップアップメニューから[検索]を選択すると、[キャッシュの検索]ダイアログボックスが開き、

カラム単位でキャッシュ項目を検索できます。

なお、命令実行中は[検索]メニューが無効になり、キャッシュ項目の検索はできません。

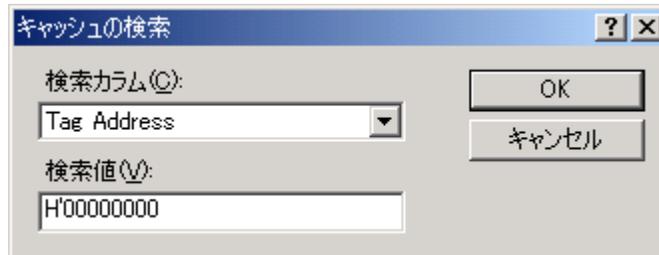


図 3-76 キャッシュの検索ダイアログボックス

本ダイアログボックスは、キャッシュの検索を行います。検索する条件を次のように設定します。

- [検索カラム] 検索する項目を指定します。
- [検索値] 検索する値を指定します。

条件を設定後、[OK]ボタンをクリックすることにより検索を行います。検索した結果、一致するエントリを強調表示します。

[キャンセル]ボタンをクリックすると検索を実行しないでダイアログボックスを閉じます。

### 3.12.5 次を検索する

前回の検索条件で、次に一致するキャッシュ項目を検索できます。前回の検索で一致するキャッシュ内容が見つかった状態で、ポップアップメニューから[次を検索]を選択します。

なお、命令実行中は[次を検索]メニューが無効になり、次に一致するキャッシュ項目の検索はできません。

### 3.12.6 キャッシュ容量を変更する

本機能は、SH-3/SH-3E シリーズ、SH-4A シリーズでのみ利用できます。ポップアップメニューから[キャッシュ容量...]を選択すると、[キャッシュ容量]ダイアログボックスが開き、キャッシュ容量を設定できます。

なお、命令実行中は[キャッシュ容量...]メニューが無効になり、キャッシュ容量の設定はできません。

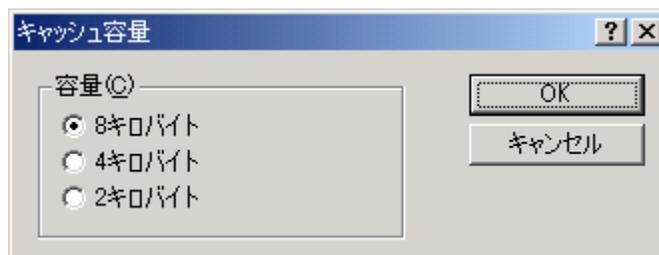


図 3-77 キャッシュ容量ダイアログボックス (SH-3/SH-3E シリーズ)

本ダイアログボックスでは、キャッシュ容量を設定します。

### 3. デバッグ

---

以下のキャッシュ容量より選択します。

- [8 キロバイト]    キャッシュ容量を 8 キロバイトに設定します(デフォルト)。  
Way0 ~ Way3 の 4 ウェイすべて(内蔵 RAM 指定時は Way0 ~ Way1 の 2 ウェイ)が有効です。
- [4 キロバイト]    キャッシュ容量を 4 キロバイトに設定します。  
Way0 ~ Way1 の 2 ウェイ (内蔵 RAM 指定時は Way0 の 1 ウェイ)が有効です。
- [2 キロバイト]    キャッシュ容量を 2 キロバイトに設定します。  
Way0 の 1 ウェイが有効です。内蔵 RAM 指定は無視されます。

[OK]ボタンをクリックすることにより、キャッシュ容量を変更します。

[キャンセル]ボタンをクリックすると、キャッシュ容量を変更しないでダイアログボックスを閉じます。

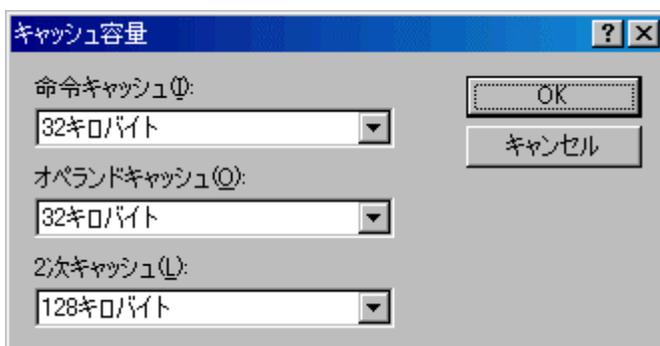


図 3-78 キャッシュ容量ダイアログボックス(SH-4A)

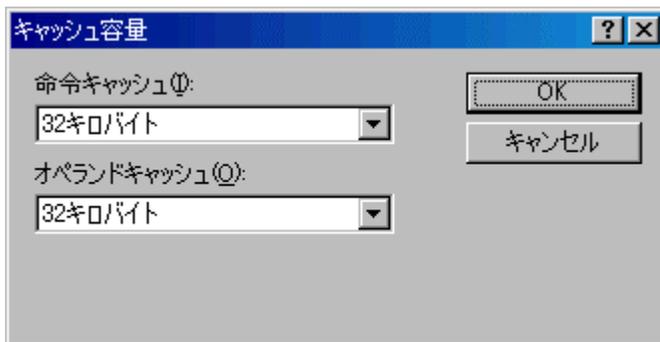


図 3-79 キャッシュ容量ダイアログボックス(SH2A-FPU、SH4AL-DSP)

本ダイアログボックスでは、キャッシュ容量を設定します。  
キャッシュ種別毎に以下のキャッシュ容量より選択します。

## SH4Aシリーズ

## [命令キャッシュ]

命令キャッシュの容量を設定します。

Way0～Way3の4ウェイすべてが有効で、エントリ数が増えます。(各ウェイのエントリ数は同一です。)

- [8 キロバイト]      8 キロバイトに設定します。  
(1 ウェイ：2 キロバイト、エントリ数 H'000～H'03F)
- [16 キロバイト]    16 キロバイトに設定します。  
(1 ウェイ：4 キロバイト、エントリ数 H'000～H'07F)
- [32 キロバイト]    32 キロバイトに設定します。(デフォルト)  
(1 ウェイ：8 キロバイト、エントリ数 H'000～H'0FF)
- [64 キロバイト]    64 キロバイトに設定します。  
(1 ウェイ：16 キロバイト、エントリ数 H'000～H'1FF)
- [128 キロバイト]   128 キロバイトに設定します。  
(1 ウェイ：32 キロバイト、エントリ数 H'000～H'3FF)

## [オペランドキャッシュ]

オペランドキャッシュの容量を設定します。

Way0～Way3の4ウェイすべてが有効で、エントリ数が増えます。(各ウェイのエントリ数は同一です。)

- [8 キロバイト]      8 キロバイトに設定します。  
(1 ウェイ：2 キロバイト、エントリ数 H'000～H'03F)
- [16 キロバイト]    16 キロバイトに設定します。  
(1 ウェイ：4 キロバイト、エントリ数 H'000～H'07F)
- [32 キロバイト]    32 キロバイトに設定します。(デフォルト)  
(1 ウェイ：8 キロバイト、エントリ数 H'000～H'0FF)
- [64 キロバイト]    64 キロバイトに設定します。  
(1 ウェイ：16 キロバイト、エントリ数 H'000～H'1FF)
- [128 キロバイト]   128 キロバイトに設定します。  
(1 ウェイ：32 キロバイト、エントリ数 H'000～H'3FF)

## [2次キャッシュ]

(SH-4Aのみ)

2次キャッシュの容量を設定します。

Way0～Way3の4ウェイすべてが有効で、エントリ数が増えます。(各ウェイのエントリ数は同一です。)

- [128 キロバイト]   128 キロバイトに設定します。(デフォルト)  
(1 ウェイ：32 キロバイト、エントリ数 H'0000～H'03FF)
- [256 キロバイト]   256 キロバイトに設定します。  
(1 ウェイ：64 キロバイト、エントリ数 H'0000～H'07FF)
- [512 キロバイト]   512 キロバイトに設定します。  
(1 ウェイ：128 キロバイト、エントリ数 H'0000～H'0FFF)
- [1 メガバイト]     1 メガバイトに設定します。  
(1 ウェイ：256 キロバイト、エントリ数 H'0000～H'1FFF)

### 3. デバッグ

---

#### SH2A-FPU

##### [命令キャッシュ]

命令キャッシュの容量を設定します。

Way0~Way3の4ウェイすべてが有効で、エントリ数が増えます。(各ウェイのエントリ数は同一です。)

[8 キロバイト]	8 キロバイトに設定します。 (1 ウェイ : 2 キロバイト、 エントリ数 : H'007F)
[16 キロバイト]	16 キロバイトに設定します。 (1 ウェイ : 4 キロバイト、 エントリ数 : H'00FF)
[32 キロバイト]	32 キロバイトに設定します。(デフォルト) (1 ウェイ : 8 キロバイト、 エントリ数 : H'01FF)
[64 キロバイト]	64 キロバイトに設定します。 (1 ウェイ : 16 キロバイト、 エントリ数 : H'03FF)
[128 キロバイト]	128 キロバイトに設定します。 (1 ウェイ : 32 キロバイト、 エントリ数 : H'07FF)
[256 キロバイト]	256 キロバイトに設定します。 (1 ウェイ : 64 キロバイト、 エントリ数 : H'0FFF)
[512 キロバイト]	512 キロバイトに設定します。 (1 ウェイ : 128 キロバイト、 エントリ数 : H'1FFF)
[1 メガバイト]	1 メガバイトに設定します。 (1 ウェイ : 256 キロバイト、 エントリ数 : H'3FFF)

##### [オペランドキャッシュ]

オペランドキャッシュの容量を設定します。

Way0~Way3の4ウェイすべてが有効で、エントリ数が増えます。(各ウェイのエントリ数は同一です。)

[8 キロバイト]	8 キロバイトに設定します。 (1 ウェイ : 2 キロバイト、 エントリ数 : H'007F)
[16 キロバイト]	16 キロバイトに設定します。 (1 ウェイ : 4 キロバイト、 エントリ数 : H'00FF)
[32 キロバイト]	32 キロバイトに設定します。(デフォルト) (1 ウェイ : 8 キロバイト、 エントリ数 : H'01FF)
[64 キロバイト]	64 キロバイトに設定します。 (1 ウェイ : 16 キロバイト、 エントリ数 : H'03FF)
[128 キロバイト]	128 キロバイトに設定します。 (1 ウェイ : 32 キロバイト、 エントリ数 : H'07FF)
[256 キロバイト]	256 キロバイトに設定します。 (1 ウェイ : 64 キロバイト、 エントリ数 : H'0FFF)
[512 キロバイト]	512 キロバイトに設定します。 (1 ウェイ : 128 キロバイト、 エントリ数 : H'1FFF)
[1 メガバイト]	1 メガバイトに設定します。 (1 ウェイ : 256 キロバイト、 エントリ数 : H'3FFF)

[OK]ボタンをクリックすることにより、キャッシュ容量を変更します。

[キャンセル]ボタンをクリックすると、キャッシュ容量を変更しないでダイアログボックスを閉じます。

#### 3.12.7 現在の表示内容をセーブする

現在ウィンドウに表示している内容をテキストファイルにセーブすることが出来ます。ポップアップメニューから[ファイルに保存...]を選択してください。

### 3.13 レジスタバンク内容を見る

SH2A-FPU でレジスタバンク内容を見るにはレジスタバンクウィンドウを使用します。

#### 3.13.1 レジスタバンクウィンドウを開く

[表示->CPU->レジスタバンク]を選択するか、[レジスタバンク]ツールバーボタンをクリックすると、レジスタバンクウィンドウが開きます。

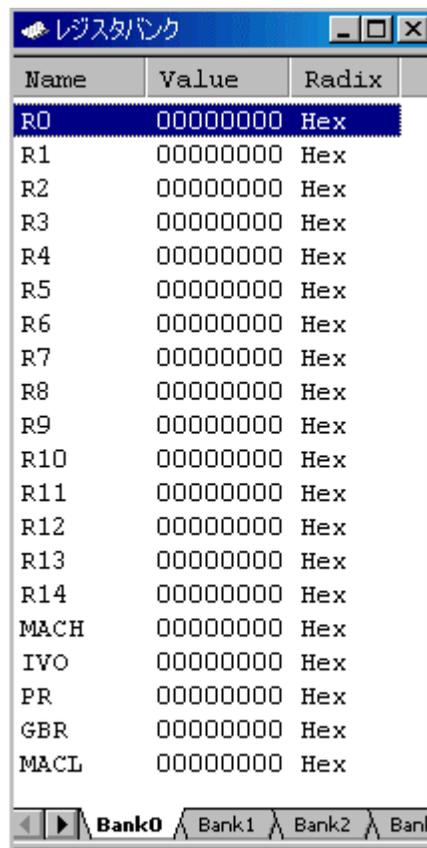


図 3-80 レジスタバンクウィンドウ

本ウィンドウはレジスタバンク情報を表示します。各バンクごとに1つのシートを持っており、表示したいバンクのタブをクリックすると、その内容を表示します。現在無効になっているバンクは、グレーで表示します。

レジスタバンクウィンドウに表示する項目は以下の通りです。

[Name]	レジスタ名
[Value]	値
[Radix]	表示基数

#### 3.13.2 レジスタバンク内容を変更する

レジスタバンク項目を選択した状態で、ポップアップメニューから[編集...]を選択するとレジスタバンク項目の変更ダイアログボックスが開き、レジスタバンク項目の内容を変更することができます。

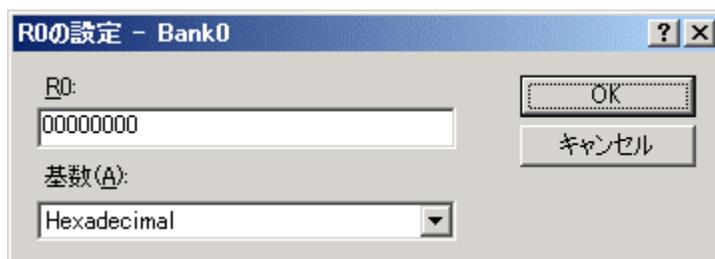


図 3-81 レジスタ値設定ダイアログボックス

本ダイアログボックスでは、現在選択されているレジスタバンク項目を変更できます。ダイアログボックスのキャプションには、選択した、項目名とバンク番号を表示します。ダイアログボックスの項目名には、選択した項目名を表示します。

[OK]ボタンをクリックすることにより、設定した内容をレジスタバンクに反映します。[キャンセル]ボタンをクリックすると、レジスタバンクに反映しないでダイアログボックスを閉じます。

また、ウィンドウに対して値を直接入力することもできます。

#### 3.13.3 表示基数を変更する

レジスタバンク項目を選択した状態で、ポップアップメニューから[基数]を選択するとレジスタバンク項目の表示基数を変更することができます。

### 3.14 仮想入出力パネルを作成する

シミュレータ・デバッガにはユーザーターゲットシステムの簡単なキー入力パネルおよび出力パネルをウィンドウ上で模擬する GUI 入出力機能があります。この仮想的な入出力パネルを作成するには GUI 入出力ウィンドウを使用します。GUI 入出力ウィンドウでは、ウィンドウ上に仮想のボタンを配置してデータを入力することや仮想の LED を配置してデータを出力することができます。

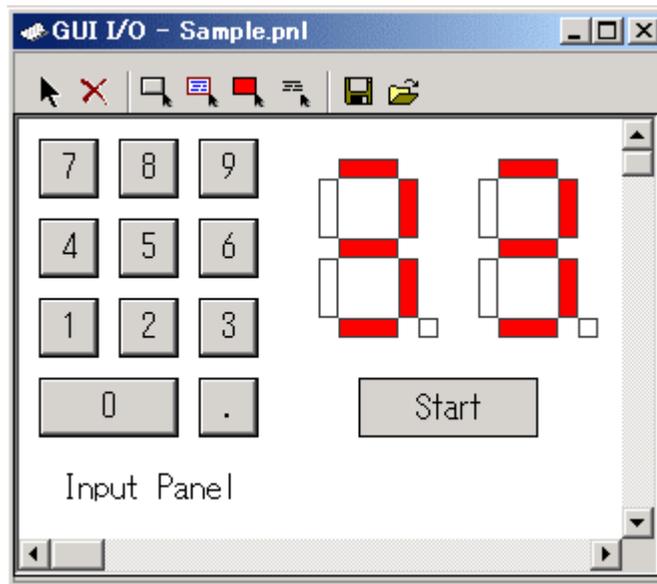


図 3-82 GUI I/O ウィンドウ例

### 3.14.1 GUI 入出力ウィンドウを開く

[表示->グラフィック->GUI I/O]を選択するか、[GUI I/O]ツールバーボタンをクリックすると、GUI I/O ウィンドウが開きます。



図 3-83 GUI I/O ウィンドウ

GUI 入出力ウィンドウでは、以下のアイテムが配置できます。

- ボタン  
ボタンを押下することにより、仮想ポートへのデータ入力や、仮想割り込みを発生させます。
- ラベル  
指定したアドレス(もしくはビット)に指定した値が書き込まれた際に、文字列を表示/消去します。
- LED  
指定したアドレス(もしくはビット)に指定した値が書き込まれた際に、指定した色で表示

### 3. デバッグ

---

- テキスト  
テキスト文字列を表示します。

#### 3.14.2 ボタンを作成する

ツールバーの  ボタンをクリックするかポップアップメニューの[ボタンの作成]を選択してください。マウスマウスカーソルの形状が "+" に変化します。

この状態で作成するボタンの左上位置から右下位置までドラッグしてください。

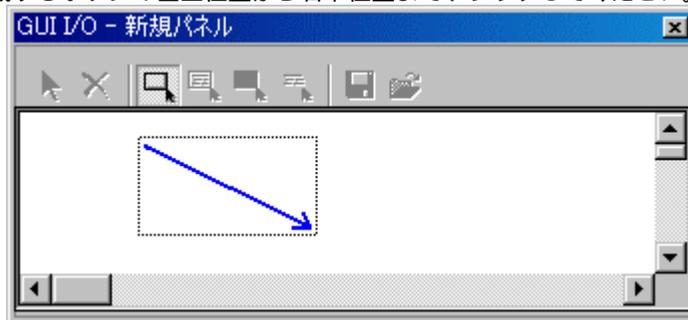


図 3-84 GUI I/O ウィンドウ (ボタン作成)

##### (1) ボタンクリック時のイベントを指定する

ツールバーの  ボタンをクリックした状態で、作成したボタンをダブルクリックしてください。以下のダイアログボックスが開きます。



図 3-85 ボタンの設定ダイアログボックス

ボタン名、入力ポートアドレスおよび入力データを指定してください。  
ボタン名に空白文字を使用することはできません。

### 3.14.3 ラベルを作成する

ツールバーの  ボタンをクリックするか、ポップアップメニューの[ラベルの作成]を選択してください。マウスカースルの形状が "+" に変化します。

この状態で作成する位置の左上位置から右下位置までドラッグしてください。ラベル枠を表示します。

### 3. デバッグ

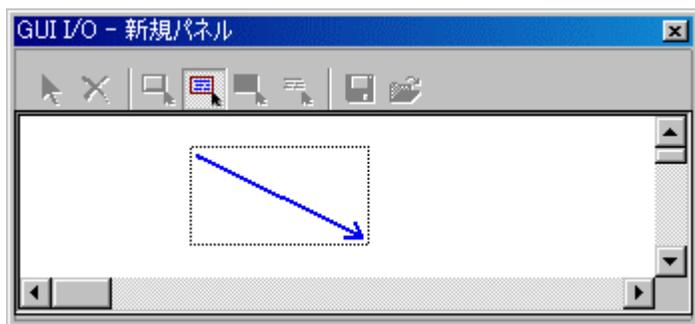


図 3-86 GUI I/O ウィンドウ (ラベル作成)

ツールバーの  ボタンをクリックするか、ポップアップメニューの[アイテムの選択]を選択後、ラベル枠をダブルクリックしてください。ラベル内容を設定するダイアログボックスが開きます。イベント発生時の動作を指定してください。ラベル名に空白文字を使用することはできません。

#### (1) 指定ビットへの任意の値書き込みの場合

アドレス 0x3E0 のビット 3 が 0 であれば"印刷中"、1 であれば"印刷可"と表示する設定を以下に示します。



図 3-87 ラベルの設定ダイアログボックス (ビット指定)

(2) 指定アドレスへの任意の値書き込みの場合

アドレス 0x3E0 にデータ 0x10 が書き込まれた時に"印刷中"、データ 0x20 が書き込まれた時に"印刷可"と表示する設定を以下に示します。

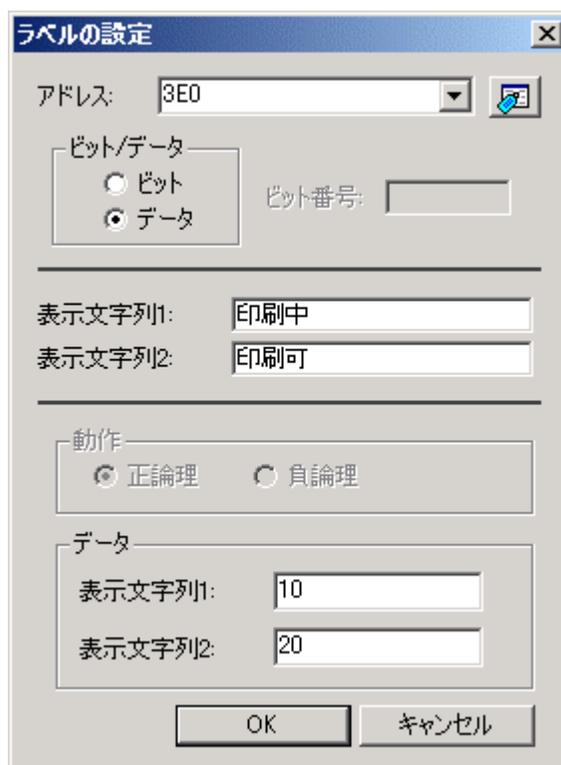


図 3-88 ラベルの設定ダイアログボックス (データ指定)

### 3.14.4 LED を作成する

ツールバーの  ボタンをクリックするか、ポップアップメニューの[LED の作成]を選択してください。マウスカーソルの形状が "+" に変化します。

この状態で作成する位置の左上位置から右下位置までドラッグしてください。LED 出力枠を表示します。

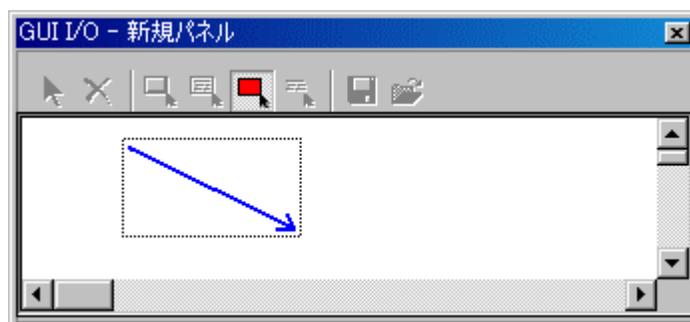


図 3-89 GUI I/O ウィンドウ (LED 作成)

### 3. デバッグ

---

ツールバーの  ボタンをクリックするか、ポップアップメニューの[アイテムの選択]を選択後、LED 出力枠をダブルクリックしてください。LED 出力を設定するダイアログボックスが開きます。イベント発生時の動作を指定してください。

(1) 指定ビットへの任意の値書き込みの場合

アドレス 0x3E0 のビット 2 が 0 であれば緑色、1 であれば赤色表示する設定を以下に示します。



図 3-90 LED の設定ダイアログボックス (ビット指定)

(2) 指定アドレスへの任意の値書き込みの場合

アドレス 0x3E0 にデータ 0x10 が書き込まれた時に緑色表示、データ 0x20 が書き込まれた時に赤色表示する設定を以下に示します。



図 3-91 LED の設定ダイアログボックス (データ指定)

表示色 1、表示色 2 ボタンをクリックするとカラー選択ダイアログボックスが開きます。

#### 3.14.5 テキスト文字列を作成する

ツールバーの  ボタンをクリックするか、ポップアップメニューの[テキストの作成]を選択してください。マウスカーソルの形状が"+"に変化します。

この状態で作成するテキスト文字列の左上位置から右下位置までドラッグしてください。

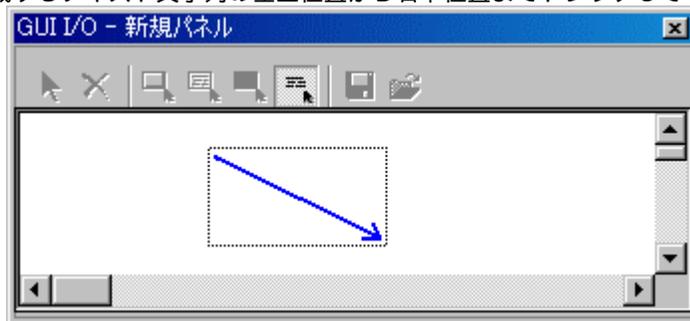


図 3-92 GUI I/O ウィンドウ (文字列作成)

##### (1) テキスト文字列の表示形式を指定する

ツールバーの  ボタンをクリックした状態で、作成したテキスト文字列をダブルクリックしてください。

以下のテキスト文字列の表示形式を設定するためのダイアログボックスが開きます。



図 3-93 テキストの設定ダイアログボックス

[フォント]ボタンを押して、表示するテキスト文字列のフォントとサイズを設定してください。

[テキスト]ボタンを押して、表示するテキスト文字列の文字色を設定してください。

[背景]ボタンを押して、表示するテキスト文字列の背景色を設定してください。

#### 3.14.6 アイテムのサイズ/配置を変更する

ツールバーの  ボタンをクリックした状態で、変更するアイテムをクリックしてください。以下のようにアイテムが選択された状態となります。

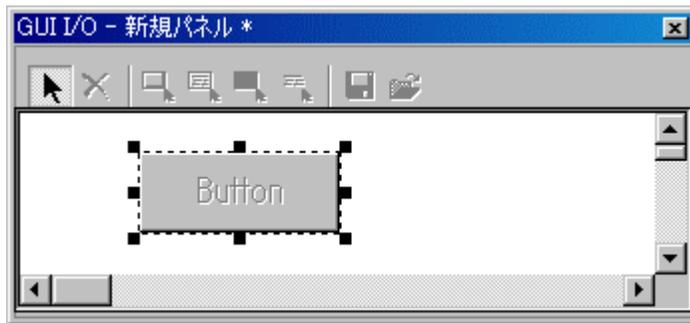


図 3-94 GUI I/O ウィンドウ (アイテムの選択)

この状態でドラッグすることにより、アイテムサイズの変更や移動ができます。

### 3.14.7 アイテムをコピーする

ツールバーの  ボタンをクリックするか、ポップアップメニューの[コピー]を選択してください。マウскарソルの形状が "+" になります。

この状態でコピー元のアイテムをクリックしてください。ツールバーの  ボタンをクリックするか、ポップアップメニューの[貼り付け]を選択してください。コピー元と同サイズのアイテムが作成されます。

### 3.14.8 アイテムを削除する

ツールバーの  ボタンをクリックするか、ポップアップメニューの[削除]を選択してください。マウскарソルの形状が "+" になります。

この状態で削除するアイテムをクリックしてください。

### 3.14.9 グリッド線を表示する

ツールバーの  ボタンをクリックするか、ポップアップメニューの[グリッドの表示]を選択してください。背景にグリッドを表示します。

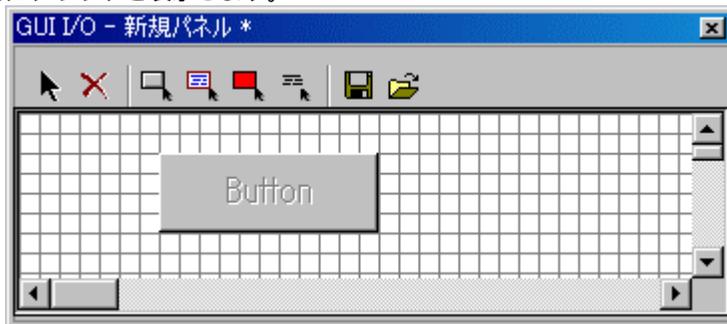


図 3-95 GUI I/O ウィンドウ (グリッドの表示)

再度  ボタンをクリックすることにより、グリッドを非表示にします。

#### 3.14.10 入出力パネル情報を保存する

作成した入出力パネルは、その情報をファイルに保存することにより、再利用することができます。ツールバーの  ボタンをクリックするか、ポップアップメニューの[保存]を選択してください。[GUI I/O パネルファイルの保存]ダイアログボックスが開きます。保存先ディレクトリとファイル名を指定してください。

#### 3.14.11 入出力パネル情報を読み込む

ツールバーの  ボタンをクリックするか、ポップアップメニューの[読み込み]を選択してください。[GUI I/O パネルファイルの読み込み]ダイアログボックスが開きます。読み込むファイルを指定してください。読み込む前のパネル情報は削除されます。

---

## 4. ウィンドウ

---

ウィンドウ一覧を表 4-1に示します。

ツールバーボタンについては、シミュレータ・デバッガヘルプを参照ください。

表 4-1 ウィンドウ一覧

ウィンドウ名	機能
IO	I/O レジスタを見る
I/O シミュレーション	標準入出力およびファイル入出力を行う
イベントポイント	シミュレータ・デバッガのブレークポイントを使用する
ウォッチ	変数を表示する（任意の変数）
エディタ	ソースコードを表示する
画像	メモリ内容を画像形式で表示する
カバレジ	コードカバレジを測定する
逆アセンブリ	アセンブリ言語コードを表示する
キャッシュ	キャッシュの内容を見る
コマンドライン	コマンドラインインタフェースでデバッグ
スタックトレース	関数呼び出し履歴を見る
ステイタス	現在の状態の参照
TLB	TLB 内容を見る
トリガ	手動で擬似割り込みを発生させる
トレース	トレース情報を見る
波形	メモリ内容を波形形式で表示する
パフォーマンス解析	パフォーマンスを解析する
プロファイル/ プロファイルチャート	プロファイル情報を見る
メモリ	メモリ領域を見る
ラベル	ラベルを参照 / 設定する
レジスタ	レジスタ内容を見る
ローカル	変数を表示する（ローカル変数）
レジスタバンク	レジスタバンク内容を見る
GUI I/O	仮想入出力パネルを作成する



---

## 5. コマンドライン

---

### 5.1 コマンド一覧 (機能別)

機能別のコマンド一覧を示します。  
各コマンドのシンタックスは、ヘルプを参照ください。

#### 5.1.1 実行関連

コマンド名	短縮形	説明
GO	GO	ユーザプログラムの実行
GO_RESET	GR	リセットベクタからのユーザプログラムの実行
GO_TILL	GT	テンポラリブレークポイントまでのユーザプログラムの実行
HALT	HA	ユーザプログラムの停止
RESET	RE	CPUのリセット
STEP	ST	ステップ実行(命令単位またはソース行単位)
STEP_MODE	SM	ステップモードの設定
STEP_OUT	SP	PC位置の関数を終了するまでのステップ実行
STEP_OVER	SO	ステップオーバ実行
STEP_RATE	SR	ステップ実行速度の設定、表示
STEP_UNIT	SN	実行単位の設定

#### 5.1.2 ダウンロード関連

コマンド名	短縮形	説明
BUILD	BU	カレントプロジェクトのビルド
BUILD_ALL	BL	カレントプロジェクトのすべてをビルド
BUILD_FILE	BF	ファイルのコンパイル
BUILD_MULTIPLE	BM	複数プロジェクトのビルド
CLEAN	CL	ビルドの中間ファイルおよび出力ファイルの削除
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
FILE_LOAD	FL	オブジェクト(プログラム)ファイルのロード
FILE_LOAD_ALL	LA	すべてのオブジェクト(プログラム)ファイルのロード
FILE_SAVE	FS	メモリ内容のファイルセーブ
FILE_UNLOAD	FU	オブジェクト(プログラム)ファイルのアンロード
FILE_UNLOAD_ALL	UA	すべてのオブジェクト(プログラム)ファイルのアンロード
FILE_VERIFY	FV	ファイル内容とメモリ内容の比較
GENERATE_MAKE_FILE	GM	カレントワークスペースのmakeファイル作成

## 5. コマンドライン

### 5.1.3 レジスタ操作関連

コマンド名	短縮形	説明
REGISTER_DISPLAY	RD	レジスタ値の表示
REGISTER_SET	RS	レジスタ値の設定

### 5.1.4 レジスタバンク操作関連 (SH2A-FPU のみ)

コマンド名	短縮形	説明
REGISTER_ADDRESS_GET	RAG	バンク制御レジスタアドレスの表示
REGISTER_ADDRESS_SET	RAS	バンク制御レジスタアドレスの設定
REGISTER_BANK_DISPLAY	RBD	レジスタバンク値の表示
REGISTER_BANK_SET	RBS	レジスタバンク値の設定

### 5.1.5 メモリ操作関連

コマンド名	短縮形	説明
CACHE	-	メモリキャッシュの有効化 / 無効化
MEMORY_COMPARE	MC	メモリ内容の比較
MEMORY_DISPLAY	MD	メモリ内容の表示
MEMORY_EDIT	ME	メモリ内容の変更
MEMORY_FILL	MF	指定データによるメモリ内容の一括変更
MEMORY_FIND	MI	メモリ内容の検索
MEMORY_MOVE	MV	メモリブロックの移動
MEMORY_TEST	MT	メモリブロックのテスト

### 5.1.6 アセンブル/逆アセンブル関係

コマンド名	短縮形	説明
ASSEMBLE	AS	アセンブルの実行
DISASSEMBLE	DA	逆アセンブル表示
SYMBOL_ADD	SA	シンボルの設定
SYMBOL_CLEAR	SC	シンボルの削除
SYMBOL_LOAD	SL	シンボル情報ファイルのロード
SYMBOL_SAVE	SS	シンボル情報のファイルセーブ
SYMBOL_VIEW	SV	シンボルの表示

### 5.1.7 ブレーク設定関連

コマンド名	短縮形	説明
BREAKPOINT	BP	実行命令位置によるブレークポイントの設定
BREAK_ACCESS	BA	メモリ範囲のアクセスによるブレーク条件の設定
BREAK_CLEAR	BC	ブレークポイントの削除
BREAK_CYCLE	BCY	サイクルによるブレーク条件の設定
BREAK_DATA	BD	メモリのデータ値によるブレーク条件の設定

コマンド名	短縮形	説明
BREAK_DATA_DIFFERENCE	BDD	メモリのデータ値の変化量(差分)によるブレイク条件の設定
BREAK_DATA_INVERSE	BDI	メモリのデータ値の符号反転によるブレイク条件の設定
BREAK_DISPLAY	BI	ブレイクポイント一覧の表示
BREAK_ENABLE	BE	ブレイクポイントの有効/無効の切り換え
BREAK_REGISTER	BR	レジスタのデータ値によるブレイク条件の設定
BREAK_SEQUENCE	BS	実行順序を指定したブレイクポイントの設定
SET_DISASSEMBLY_SOFT_BREAK	SDB	逆アセンブリレベルのソフトウェアブレイクポイントの設定または解除
SET_SOURCE_SOFT_BREAK	SSB	ソースレベルのソフトウェアブレイクポイントの設定または解除
STATE_DISASSEMBLY_SOFT_BREAK	TDB	逆アセンブリレベルのソフトウェアブレイクポイントの有効化または無効化
STATE_SOURCE_SOFT_BREAK	TSB	ソースレベルのソフトウェアブレイクポイントの有効化または無効化

### 5.1.8 トレース関連

コマンド名	短縮形	説明
TRACE	TR	トレース情報の表示
TRACE_ACQUISITION	TA	トレース情報取得の有効/無効の切り換え
TRACE_SAVE	TV	トレース情報をファイルへ出力
TRACE_STATISTIC	TST	統計情報解析の実行

### 5.1.9 カバレッジ計測関連

コマンド名	短縮形	説明
COVERAGE	CV	カバレッジ測定の有効化/無効化
COVERAGE_DISPLAY	CVD	カバレッジ情報の表示
COVERAGE_LOAD	CVL	カバレッジ情報のロード
COVERAGE_RANGE	CVR	カバレッジ範囲の設定
COVERAGE_SAVE	CVS	カバレッジ情報のセーブ

### 5.1.10 パフォーマンス測定関連

コマンド名	短縮形	説明
ANALYSIS	AN	性能分析機能の有効化/無効化
ANALYSIS_RANGE	AR	性能評価関数の設定、表示
ANALYSIS_RANGE_DELETE	AD	性能分析範囲の削除
PROFILE	PR	プロファイルの有効化/無効化
PROFILE_DISPLAY	PD	プロファイル情報の表示
PROFILE_SAVE	PS	プロファイル情報のファイル出力

## 5. コマンドライン

---

### 5.1.11 ウォッチ関連

コマンド名	短縮形	説明
WATCH_ADD	WA	Watch アイテムの追加
WATCH_AUTO_UPDATE	WU	Watch アイテムの自動更新の設定または解除
WATCH_DELETE	WD	Watch アイテムの削除
WATCH_DISPLAY	WI	ウォッチウィンドウの内容の表示
WATCH_EDIT	WE	Watch アイテムの値の編集
WATCH_EXPAND	WX	Watch アイテムの展開または縮小
WATCH_RADIX	WR	Watch アイテムの表示基数の変更
WATCH_RECORD	WO	Watch アイテムの値更新履歴をファイルに出力
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

### 5.1.12 スクリプト/ログファイル関連

コマンド名	短縮形	説明
!	-	コメント
ASSERT	-	コンディションのチェック
AUTO_COMPLETE	AC	オートコンプリート機能の有効化 / 無効化
ERASE	ER	コマンドラインウィンドウの内容のクリア
EVALUATE	EV	式の計算
LOG	LO	ロギングファイルの操作
SLEEP	-	コマンド実行の遅延
SUBMIT	SU	コマンドファイルの実行
TCL	-	TCL の情報を表示

### 5.1.13 マップ関連

コマンド名	短縮形	説明
MAP_DISPLAY	MA	メモリリソース設定の表示
MAP_SET	MS	メモリリソースの設定

### 5.1.14 シミュレーションの設定関連

コマンド名	短縮形	説明
CLOCK_RATE	CKR	クロック比の設定
EXEC_MODE	EM	実行モードの設定、表示
EXEC_STOP_SET	ESS	割り込み発生時の実行モードの設定、表示

## 5.1.15 標準入出力およびファイル入出力関連

コマンド名	短縮形	説明
SIMULATEDIO_CLEAR	SIOC	I/O シミュレーションウィンドウの内容のクリア
TRAP_ADDRESS	TP	I/O シミュレーションアドレスの設定
TRAP_ADDRESS_DISPLAY	TD	I/O シミュレーションアドレス設定の表示
TRAP_ADDRESS_ENABLE	TE	I/O シミュレーション有効化 / 無効化

## 5.1.16 ユーティリティ関連

コマンド名	短縮形	説明
HELP	HE	コマンドラインヘルプの表示
INITIALIZE	IN	デバッグプラットフォームの初期化
QUIT	QU	HEW の終了
RADIX	RA	入力ラディックス(基数)の設定
RESPONSE	RP	ウィンドウリフレッシュ間隔の設定
STATUS	STA	デバッグプラットフォームの状況表示
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

## 5.1.17 プロジェクト/ワークスペース関連

コマンド名	短縮形	説明
ADD_FILE	AF	カレントプロジェクトへのファイル追加
CHANGE_CONFIGURATION	CC	コンフィギュレーションの設定
CHANGE_PROJECT	CP	プロジェクトの設定
CHANGE_SESSION	CS	セッションの設定
CHANGE_SUB_SESSION	CB	同期デバッグが有効な場合のアクティブセッションの変更
CLEAR_OUTPUT_WINDOW	COW	アウトプットウィンドウの各タブの表示内容クリア
CLOSE_WORKSPACE	CW	ワークスペースを閉じる
OPEN_WORKSPACE	OW	ワークスペースのオープン
REFRESH_SESSION	RSE	セッション情報の再読み込み
REMOVE_FILE	REM	カレントプロジェクトからのファイル削除
SAVE_SESSION	SE	現在のセッションをセーブ
SAVE_WORKSPACE	SW	ワークスペースをセーブ
UPDATE_ALL_DEPENDENCIES	UD	カレントプロジェクトのすべての依存関係更新

## 5.1.18 テスト支援機能関連

コマンド名	短縮形	説明
CLOSE_TEST_SUITE	CTS	テストスイートを閉じる
COMPARE_TEST_DATA	CTD	テストデータの比較
OPEN_TEST_SUITE	OTS	テストスイートを開く
RUN_TEST	RT	テストの実行

## 5.2 コマンド一覧 (アルファベット順)

アルファベット順コマンド一覧を示します。  
各コマンドのシンタックスは、ヘルプを参照ください。

表 5-1 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へのファイル追加
3	ANALYSIS	AN	性能分析機能の有効化 / 無効化
4	ANALYSIS_RANGE	AR	性能評価関数の設定、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	BREAK_ACCESS	BA	メモリ範囲のアクセスによるブレーク条件の設定
11	BREAK_CLEAR	BC	ブレークポイントの削除
12	BREAK_CYCLE	BCY	サイクルによるブレーク条件の設定
13	BREAK_DATA	BD	メモリのデータ値によるブレーク条件の設定
14	BREAK_DATA_DIFFERENCE	BDD	メモリのデータ値の変化量(差分)によるブレーク条件の設定
15	BREAK_DATA_INVERSE	BDI	メモリのデータ値の符号反転によるブレーク条件の設定
16	BREAK_DISPLAY	BI	ブレークポイント一覧の表示
17	BREAK_ENABLE	BE	ブレークポイントの有効/無効の切替
18	BREAK_REGISTER	BR	レジスタのデータ値によるブレーク条件の設定
19	BREAK_SEQUENCE	BS	実行順序を指定したブレークポイントの設定
20	BUILD	BU	カレントプロジェクトのビルド
21	BUILD_ALL	BL	カレントプロジェクトのすべてのビルド
22	BUILD_FILE	BF	ファイルのコンパイル
23	BUILD_MULTIPLE	BM	複数プロジェクトのビルド
24	CACHE	-	メモリアクセスの有効化 / 無効化
25	CHANGE_CONFIGURATION	CC	コンフィギュレーションの設定
26	CHANGE_PROJECT	CP	プロジェクトの設定
27	CHANGE_SESSION	CS	セッションの設定
28	CHANGE_SUB_SESSION	CB	同期デバッグが有効な場合のアクティブセッションの変更
29	CLEAN	CL	ビルドの中間ファイルおよび出力ファイルの削除
30	CLEAR_OUTPUT_WINDOW	COW	アウトプットウィンドウの各タブの表示内容クリア
31	CLOCK_RATE	CKR	クロック比の設定
32	CLOSE_TEST_SUITE	CTS	テストスイートを閉じる
33	CLOSE_WORKSPACE	CW	ワークスペースを閉じる
34	COMPARE_TEST_DATA	CTD	テストデータの比較
35	COVERAGE	CV	カバレッジ測定の有効化 / 無効化
36	COVERAGE_DISPLAY	CVD	カバレッジ情報の表示
37	COVERAGE_LOAD	CVL	カバレッジ情報のロード

## 5. コマンドライン

項番	コマンド名	短縮形	説明
38	COVERAGE_RANGE	CVR	カバレッジ範囲の設定
39	COVERAGE_SAVE	CVS	カバレッジ情報のセーブ
40	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
41	DISASSEMBLE	DA	逆アセンブル表示
42	ERASE	ER	[コマンドライン]ウィンドウの内容のクリア
43	EVALUATE	EV	式の計算
44	EXEC_MODE	EM	実行モードの設定、表示
45	EXEC_STOP_SET	ESS	割込み発生時の実行モードの設定、表示
46	FILE_LOAD	FL	オブジェクト(プログラム)ファイルのロード
47	FILE_LOAD_ALL	LA	すべてのオブジェクト(プログラム)ファイルのロード
48	FILE_SAVE	FS	メモリ内容のファイルセーブ
49	FILE_UNLOAD	FU	オブジェクト(プログラム)ファイルのアンロード
50	FILE_UNLOAD_ALL	UA	すべてのオブジェクト(プログラム)ファイルのアンロード
51	FILE_VERIFY	FV	ファイル内容とメモリ内容の比較
52	GENERATE_MAKE_FILE	GM	カレントワークスペースの make ファイル作成
53	GO	GO	ユーザプログラムの実行
54	GO_RESET	GR	リセットベクタからのユーザプログラムの実行
55	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
56	HALT	HA	ユーザプログラムの停止
57	HELP	HE	コマンドラインのヘルプ表示
58	INITIALIZE	IN	デバッグプラットフォームの初期化
59	LOG	LO	ロギングファイルの操作
60	MAP_DISPLAY	MA	メモリリソース設定の表示
61	MAP_SET	MS	メモリリソースの設定
62	MEMORY_COMPARE	MC	メモリ内容の比較
63	MEMORY_DISPLAY	MD	メモリ内容の表示
64	MEMORY_EDIT	ME	メモリ内容の変更
65	MEMORY_FILL	MF	指定データによるメモリ内容の一括変更
66	MEMORY_FIND	MI	メモリ内容の検索
67	MEMORY_MOVE	MV	メモリブロックの移動
68	MEMORY_TEST	MT	メモリブロックのテスト
69	OPEN_TEST_SUITE	OTS	テストスイートを開く
70	OPEN_WORKSPACE	OW	ワークスペースのオープン
71	PROFILE	PR	プロファイルの有効化/無効化
72	PROFILE_DISPLAY	PD	プロファイル情報の表示
73	PROFILE_SAVE	PS	プロファイル情報のファイル出力
74	QUIT	QU	HEW の終了
75	RADIX	RA	入力ラディックス(基数)の設定
76	REFRESH_SESSION	RSE	セッション情報の再読み込み
77	REGISTER_ADDRESS_GET	RAG	バンク制御レジスタアドレスの表示
78	REGISTER_ADDRESS_SET	RAS	バンク制御レジスタアドレスの設定
79	REGISTER_BANK_DISPLAY	RBD	レジスタバンク値の表示

## 5. コマンドライン

項番	コマンド名	短縮形	説明
80	REGISTER_BANK_SET	RBS	レジスタバンク値の設定
81	REGISTER_DISPLAY	RD	レジスタ値の表示
82	REGISTER_SET	RS	レジスタ値の設定
83	REMOVE_FILE	REM	カレントプロジェクトからのファイル削除
84	RESET	RE	CPUのリセット
85	RESPONSE	RP	ウィンドウリフレッシュ間隔の設定
86	RUN_TEST	RT	テストの実行
87	SLEEP	-	コマンド実行の遅延
88	SAVE_SESSION	SE	現在のセッションをセーブ
89	SAVE_WORKSPACE	SW	ワークスペースの保存
90	SET_DISASSEMBLY_SOFT_BREAK	SDB	逆アセンブリレベルのソフトウェアブレークポイントの設定または解除
91	SET_SOURCE_SOFT_BREAK	SSB	ソースレベルのソフトウェアブレークポイントの設定または解除
92	SIMULATEDIO_CLEAR	SIOC	I/O シミュレーションウィンドウの内容のクリア
93	STATE_DISASSEMBLY_SOFT_BREAK	TDB	逆アセンブリレベルのソフトウェアブレークポイントの有効化または無効化
94	STATE_SOURCE_SOFT_BREAK	TSB	ソースレベルのソフトウェアブレークポイントの有効化または無効化
95	STATUS	STA	デバッグプラットフォームの状況表示
96	STEP	ST	ステップ実行(命令単位またはソース行単位)
97	STEP_MODE	SM	ステップモードの設定
98	STEP_OUT	SP	PC 位置の関数を終了するまでのステップ実行
99	STEP_OVER	SO	ステップオーバー実行
100	STEP_RATE	SR	ステップ実行速度の設定、表示
101	STEP_UNIT	SN	実行単位の設定
102	SUBMIT	SU	コマンドファイルの実行
103	SYMBOL_ADD	SA	シンボルの設定
104	SYMBOL_CLEAR	SC	シンボルの削除
105	SYMBOL_LOAD	SL	シンボル情報ファイルのロード
106	SYMBOL_SAVE	SS	シンボル情報のファイルセーブ
107	SYMBOL_VIEW	SV	シンボルの表示
108	TCL	-	TCL の有効化 / 無効化
109	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
110	TRACE	TR	トレース情報の表示
111	TRACE_ACQUISITION	TA	トレース情報取得の有効/無効の切替
112	TRACE_SAVE	TV	トレース情報をファイルへ出力
113	TRACE_STATISTIC	TST	統計情報解析の実行
114	TRAP_ADDRESS	TP	I/O シミュレーションアドレスの設定
115	TRAP_ADDRESS_DISPLAY	TD	I/O シミュレーションアドレス設定の表示
116	TRAP_ADDRESS_ENABLE	TE	I/O シミュレーション有効化 / 無効化
117	UPDATE_ALL_DEPENDENCIES	UD	カレントプロジェクトのすべての依存関係更新
118	WATCH_ADD	WA	Watch アイテムの追加
119	WATCH_AUTO_UPDATE	WU	Watch アイテムの自動更新の設定または解除
120	WATCH_DELETE	WD	Watch アイテムの削除

---

項番	コマンド名	短縮形	説明
121	WATCH_DISPLAY	WI	ウォッチウィンドウの内容の表示
122	WATCH_EDIT	WE	Watch アイテムの値の編集
123	WATCH_EXPAND	WX	Watch アイテムの展開または縮小
124	WATCH_RADIX	WR	Watch アイテムの表示基数の変更
125	WATCH_RECORD	WO	Watch アイテムの値更新履歴をファイルに出力
126	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



---

## 6. メッセージ一覧

---

### 6.1 インフォメーションメッセージ

シミュレータ・デバッガは実行経過をユーザに知らせるため、インフォメーションメッセージを出力します。シミュレータ・デバッガの出力するインフォメーションメッセージを表 6-1に示します。

表 6-1 インフォメーションメッセージ一覧

メッセージ	内容
Break Access (Access Address:H'nnnnnnnn, Type:xxxx, Access Size:yyyy)	ブレイクアクセス条件が成立して実行を中断しました。付加情報として成立したブレイクアクセス条件(アクセスアドレス(Access Address)、アクセス種別(Type)、およびアクセスサイズ(Access Size))を表示します。
Break Cycle (Cycle:H'nnnnnnnn)	ブレイクサイクル条件が成立して実行を中断しました。付加情報として成立したブレイクサイクル条件(サイクル数(Cycle))を表示します。
Break Data (Access Address:H'nnnnnnnn, Data:H'mmmmmmmm)	ブレイクデータ条件(一致、または不一致)が成立して実行を中断しました。付加情報として成立したブレイクデータ条件(アクセスアドレス(Access Address)、および値(Data))を表示します。
Break Data (Access Address:H'nnnnnnnn, Previous Data:H'mmmmmmmm, Current Data:H'mmmmmmmm)	ブレイクデータ条件(符号反転、または差分)が成立して実行を中断しました。付加情報として成立したブレイクデータ条件(アクセスアドレス(Access Address)、前回値(Previous Data)、および今回値(Current Data))を表示します。
Break Register (Register:XX, Value:H'mmmmmmmm)	ブレイクレジスタ条件が成立して実行を中断しました。付加情報として成立したブレイクレジスタ条件(レジスタ名(Register)、および値(Value))を表示します。
Break Sequence (PC:H'nnnnnnnn)	ブレイクシーケンス条件が成立して実行を中断しました。付加情報として成立したブレイクシーケンス条件(最後の命令アドレス(PC))を表示します。
I/O DLL Stop	周辺機能が停止しました。
PC Breakpoint (PC:H'nnnnnnnn)	ブレイクポイント条件が成立して実行を中断しました。付加情報として成立したブレイクポイント条件(命令アドレス(PC))を表示します。
Sleep	SLEEP 命令により実行を中断しました。
Step Normal End	ステップ実行が正常に終了しました。
Stop	[STOP]ボタンにより実行を中断しました。
Trace Buffer Full	[トレース取得]ダイアログボックスの[トレースバッファ満杯時の動作]で[停止]モードを選択しており、かつトレースバッファが満杯となったので実行を中断しました。

## 6.2 エラーメッセージ

シミュレータ・デバッガはユーザプログラムや操作の誤りをユーザに知らせるため、エラーメッセージを出力します。シミュレータ・デバッガの出力するエラーメッセージを表 6-2に示します。

表 6-2 エラーメッセージ一覧

メッセージ	内容・対策
Address Error	以下のいずれかの状態になりました。 (1) PC 値が奇数である (2) 内蔵 I/O 空間から命令読み出しを行おうとした (3) ワードデータを (2n) 番地以外からアクセスしようとした (4) ロングワードデータを (4n) 番地以外からアクセスしようとした (5) VBR,SP が 4 の倍数以外である (6) アドレスエラーの例外処理でエラーが発生した エラーが発生しないようにユーザプログラムを修正してください。
Exception Error	例外処理でエラーが発生しました。 エラーが発生しないようにユーザプログラムを修正してください。
File Open Error	Break の FileInput/Output アクションでファイルオープンに失敗しました。 ファイル指定を見直してください。
File Input Error	Break の File Input アクションでファイル読み込みに失敗しました。 ファイル指定を見直してください。
File Output Error	Break の File Output アクションでファイル書き出しに失敗しました。 ファイル指定を見直してください。
FPU Disable	FPU が使用できない状態 (SR.FD=1) で FPU 命令を実行しようとした。 エラーが発生しないようにプログラムを修正してください。
FPU Error	浮動小数点演算で、以下のいずれかの状態になりました。 (1) FPU エラー発生 (2) 無効演算発生 (3) ゼロによる除算発生 (4) オーバフロー発生 (5) アンダフロー発生 (6) 不正確発生 エラーが発生しないようにユーザプログラムを修正してください。
General Invalid Instruction	以下のいずれかの状態になりました。 (1) 命令ではないコードを実行しようとした (2) 一般不当命令の例外処理でエラーが発生した エラーが発生しないようにユーザプログラムを修正してください。
Illegal CCR2 Set	CCR2 の値が不正です。 設定を確認してください。
Illegal Combination BSC Register	BSC レジスタの設定が不正なエリアへアクセスしようとした。 エラーが発生しないようにユーザプログラムを修正してください。
Illegal DSP Operation	以下のいずれかの状態になりました。 (1) PSHA 命令で 32 ビットを越えるシフトを行おうとした (2) PSHL 命令で 16 ビットを越えるシフトを行おうとした エラーが発生しないようにユーザプログラムを修正してください。

メッセージ	内容・対策	
Illegal LRU Set	キャッシュのLRUの値が不正です。 設定を確認してください。	
Illegal Operation	以下のいずれかの状態になりました。 (1) DIV1 命令でゼロ除算が発生した (2) SETRC 命令でゼロを書き込もうとした エラーが発生しないようにユーザプログラムを修正してください。	
Illegal PR bit	FPSCR の PR ビットが不正な状態で FPU 命令を実行しようとした。 エラーが発生しないようにユーザプログラムを修正してください。	
Initial Page Write	シミュレーション実行中に初期ページ書き込み例外が発生しました。 TLB の内容を更新する等必要な処置をとってください。	
Instruction TLB Illegal LRU	命令 TLB の LRU の値が不正です。 設定を確認してください。	
Instruction TLB Miss	メモリアクセスで 命令 TLB ミスが発生しました。 命令 TLB 内容を更新する等必要な処置をとってください。	
Instruction TLB Multiple Hit	シミュレーション実行中またはコマンド実行中にアクセスした論理アドレスが 命令 TLB の複数エンTRIES にヒットしました。 命令 TLB を適正に設定していません。 命令 TLB 内容を修正するとともにプログラム (ハンドルーチン) を修正してください。	
Instruction TLB Protection Violation	メモリアクセスで 命令 TLB 保護例外が発生しました。 命令 TLB 内容を更新する等必要な処置をとってください。	
Interrupt Exception	割り込み例外が発生して実行を中断しました。	
Invalid DSP Instruction Code	DSP パラレル命令で不正な命令コードを検出しました。 エラーが発生しないようにユーザプログラムを修正してください。	
Invalid Slot Instruction	以下のいずれかの状態になりました。 (1) 遅延分岐命令直後の PC を変える命令 (分岐命令) を実行した (2) スロット不当命令の例外処理でエラーが発生した エラーが発生しないようにユーザプログラムを修正してください。	
Invalid Register Bank Number	STBANK 命令、LDBANK 命令で存在しないバンク、ENTRIES を指定しました。正しいバンクおよびENTRIES を指定してください。	
I/O area not exist	I/O 領域を削除しようとした。 I/O 領域は必ず設定してください。	
I/O DLL Illegal Interrupt Information (errNum=2xx)	割り込み情報が不正です。詳細情報を[errNum]に示します。割り込み情報を修正してください。	
	errNum	説明
	200	指定したベクタが範囲外です
	201	指定した優先度が範囲外です
I/O DLL Memory Access Error (errNum=0xx, Address=0xXXXXXXXX)	周辺機能へのメモリアクセスでエラーが発生しました。詳細内容を[errNum]、エラー発生アドレスを[Address]に示します。エラー情報に従ってプログラムを修正してください。	
	errNum	説明
	001	指定したアドレスが範囲外です
	002	指定した領域にメモリがありません
	003	必要なメモリが確保できません
	004	指定したデータサイズが範囲外です
005	指定したアドレスにアクセスできません	

## 6. メッセージ一覧

メッセージ	内容・対策	
I/O DLL Register Access Error (errNum=1xx, RegisterName=xxxx)	周辺機能へのレジスタアクセスでエラーが発生しました。詳細内容を [errNum]、エラー発生レジスタを[RegisterName]に示します。エラー情報に従ってプログラムを修正してください。	
	errNum	説明
	100	レジスタの記述に誤りがあります
	101	指定したデータ値が不正です
Memory Access Error (Address:H'nnnnnnnn)	以下のいずれかの状態になりました。付加情報としてエラーの発生したアドレス(Address)を表示します。 (1) 確保していないメモリ領域をアクセスしようとした (2) 書き込み不可属性を持つメモリへの書き込みを行おうとした (3) 読み出し不可属性を持つメモリからの読み出しを行おうとした (4) メモリが存在しない領域をアクセスしようとした メモリの確保、属性変更を行うか、当該メモリアccessが発生しないようにユーザプログラムを修正してください。	
Multiple Exception	多重例外が発生しました。 エラーが発生しないようにプログラムを修正してください。	
PMB Multiple Hit	シミュレーション実行中またはコマンド実行中にアクセスした論理アドレスが PMB の複数エントリにヒットしました。 PMB を適正に設定していません。 PMB 内容を修正するとともにプログラム(ハンドラルーチン)を修正してください。	
PMB Miss	メモリアccessで PMB ミスが発生しました。 PMB 内容を更新する等必要な処置をとってください。	
Register Bank Overflow	レジスタバンクオーバーフローを受け付けるように設定されており、レジスタバンクの全ての領域に退避が行われている状態で、レジスタバンクを使用する割り込みが発生しました。	
Register Bank Underflow	レジスタバンクに退避が行われていないときに、RESBANK 命令を実行しました。	
Slot FPU Disable	FPU が使用できない状態(SR.FD=1)で遅延スロットにある FPU 命令を実行しようとした。 エラーが発生しないようにプログラムを修正してください。	
System Call Error	I/O シミュレーションエラーが発生しました。 レジスタ R0,R1 およびパラメータブロックの内容の誤りを修正してください。	
The memory resource has not been set up	メモリマップ範囲外にメモリリソースを設定しました。 エラーが発生しないようにメモリリソース設定を修正してください。	
TLB Invalid	シミュレーション実行中またはコマンド実行中に TLB 無効例外が発生しました。 TLB の内容を更新する等必要な処置をとってください。	
TLB Miss	シミュレーション実行中またはコマンド実行中に TLB ミス例外が発生しました。 TLB の内容を更新する等必要な処置をとってください。	
TLB Multiple Hit	シミュレーション実行中またはコマンド実行中にアクセスした論理アドレスが TLB の複数エントリにヒットしました。 TLB を適正に設定していません。 TLB の内容を修正するとともにプログラム(ハンドラルーチン)を修正してください。	
TLB Protection Violation	シミュレーション実行中に TLB 保護違反例外が発生しました。 TLB の内容を更新する等必要な処置をとってください。	

メッセージ	内容・対策
Unified TLB Miss	メモリアクセスで 共用 TLB ミスが発生しました。 共用 TLB 内容を更新する等必要な処置をとってください。
Unified TLB Multiple Hit	シミュレーション実行中またはコマンド実行中にアクセスした論理アドレスが 共用 TLB の複数エントリにヒットしました。 共用 TLB を適正に設定していません。 共用 TLB 内容を修正するとともにプログラム（ハンドラルーチン）を修正してください。
Unified TLB Protection Violation	メモリアクセスで 共用 TLB 保護例外が発生しました。 共用 TLB 内容を更新する等必要な処置をとってください。



---

## 7. チュートリアル

---

### 7.1 デバッグの準備

シミュレータ・デバッガの主な特徴をサンプルプログラムを用いて説明します。

#### 注意

本章の使用例(図)の内容は、ご使用になるコンパイラのバージョンによって変わってくる場合がありますのでご了承ください。

#### 7.1.1 サンプルプログラム

サンプルプログラムは、10個のランダムデータを昇順にソートした後、降順にソートするCプログラムに基づいています。

サンプルプログラムでは、以下の処理を行います。

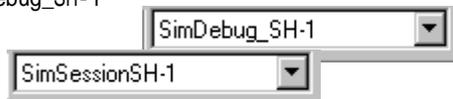
- main関数でソートするランダムデータを生成します。
- sort関数ではmain関数で生成したランダムデータを格納した配列を入力し、昇順にソートします。
- change関数では、sort関数で生成した配列を入力し、降順にソートします。
- printf関数を用いて、ランダムデータ、ソートしたデータを表示します。

サンプルプログラムは、HEWのデモンストレーションプログラムを用います。

#### 7.1.2 サンプルプログラムの作成

下記に注意してHEWのデモンストレーションプログラムを作成してください。

- 「新規ワークスペースを作成する」では[Project Type]に"Demonstration"を指定してください。
- [CPUシリーズ:]は"SH-1"を、[ターゲット:]は"SH-1 Simulator"を選択してください。
- コンフィギュレーションは、ツールバーより"SimDebug\_SH-1"をビルド前に選択してください。
- セッションは、ツールバーより"SimSessionSH-1"を選択してください。
- 本デモンストレーションプログラムでは周辺機能(タイマ)は利用しないので、セッション切換え時に表示される[周辺機能シミュレーションの設定]ダイアログボックスは[このダイアログを表示しない]チェックボックスチェック後に[OK]ボタンをクリックしてください。



デバッグ機能の説明のため、Demonstrationは最適化なしの設定になっています。最適化の設定は変更しないでください。

## 7.2 デバッグのための設定

### 7.2.1 メモリリソースの確保

開発しているアプリケーションを動作させるためにメモリリソースの確保が必要です。デモンストレーションプロジェクトでは、自動的にメモリリソースを確保しますので、設定を確認してください。

- [基本設定]メニューから [シミュレータ -> メモリリソース...]を選択し、現在のメモリリソースを表示してください。

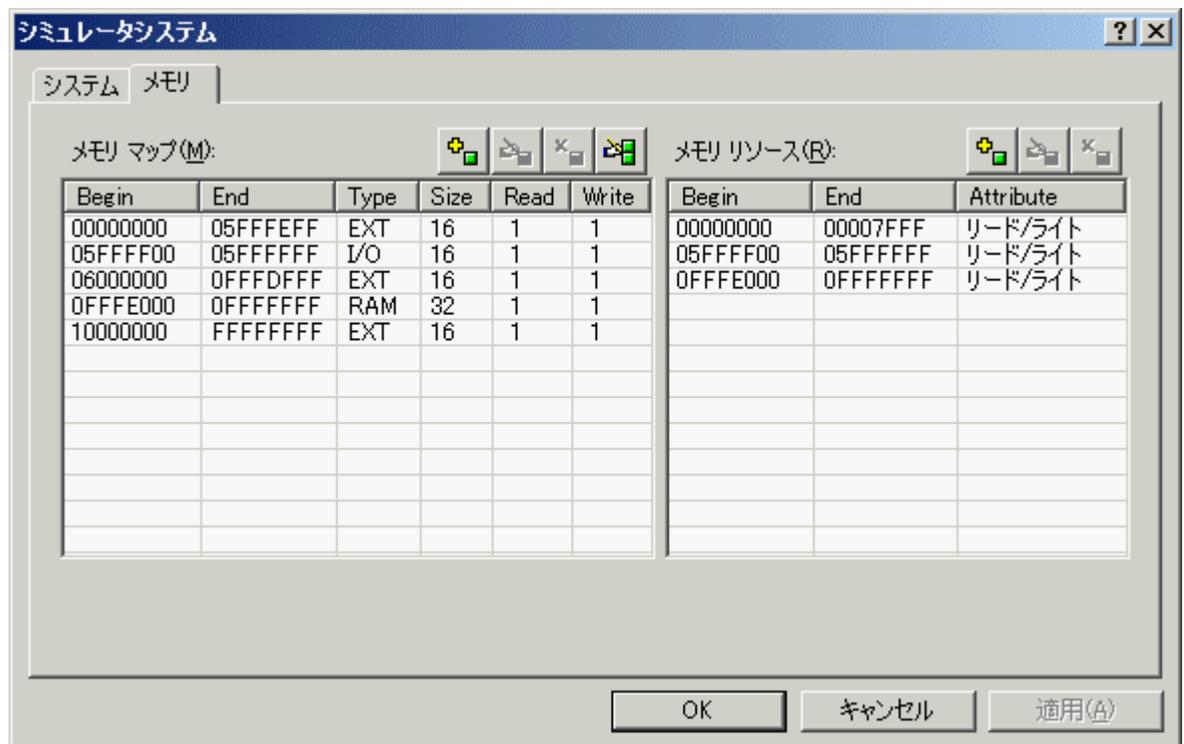


図 7-1 シミュレータシステムダイアログボックス (メモリページ)

プログラム領域として H'00000000 から H'00007FFF を、スタック領域として H'0FFFE000 から H'0FFFFFFF を読み出し/書き込み可能領域として確保しています。

- [OK]ボタンをクリックしてダイアログボックスを閉じてください。

メモリリソースは、[SuperH RISC engine Standard Toolchain]ダイアログボックスの[デバッガ]ページでも参照 / 変更ができます。お互いの変更は反映されます。

## 7.2.2 サンプルプログラムのダウンロード

デモンストレーションプロジェクトでは、ダウンロードするサンプルプログラムを自動的に設定しますので、設定を確認してください。

- [デバッグ]メニューから [デバッグの設定...]を選択して、[デバッグの設定]ダイアログボックスを開いてください。

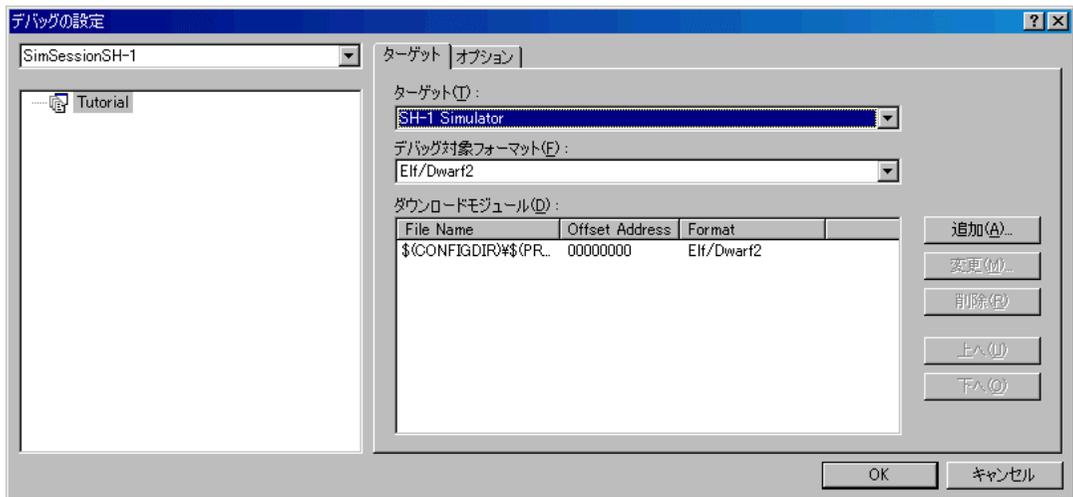


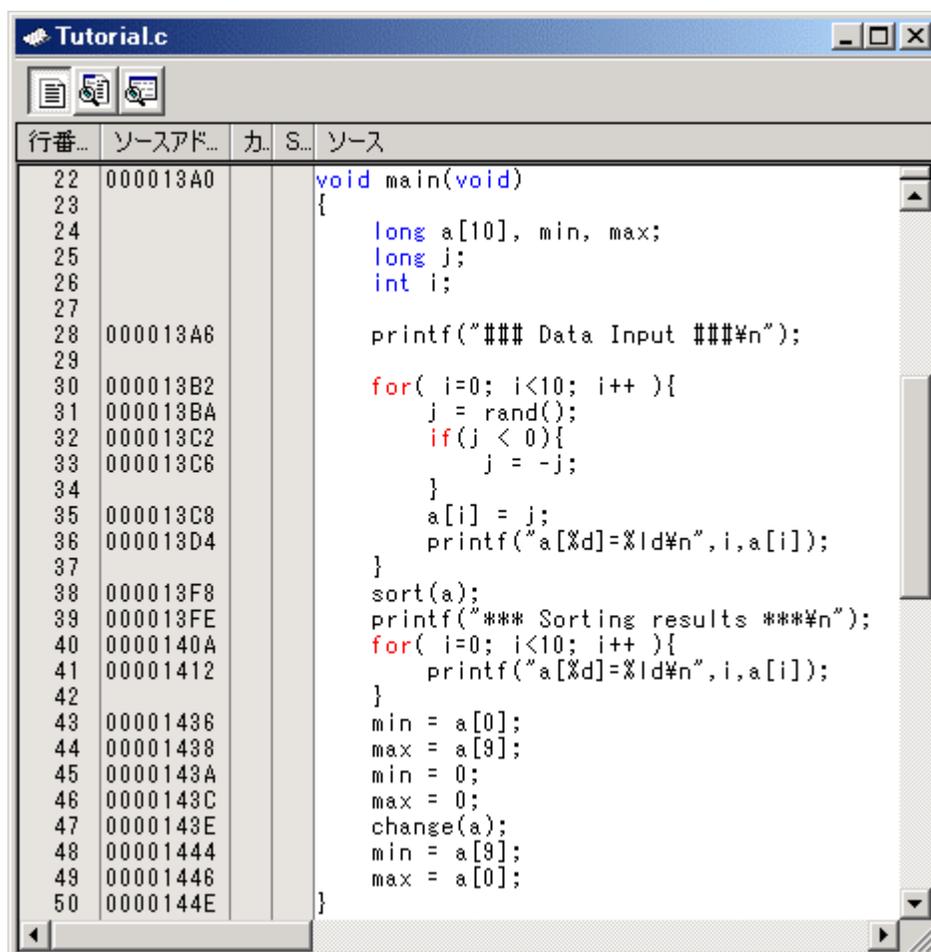
図 7-2 デバッグの設定ダイアログボックス

- [ダウンロードモジュール]に設定しているファイルがダウンロードするファイルです。
- [OK]ボタンをクリックして[デバッグの設定]ダイアログボックスを閉じてください。
- [デバッグ]メニューから [ダウンロード -> All Download Modules]を選択して、サンプルプログラムをダウンロードしてください。

## 7.2.3 ソースプログラムの表示

HEW では、ソースレベルでプログラムをデバッグできます。ソースプログラム ("Tutorial.c") を表示してください。

- [ワークスペース]ウィンドウの Tutorial.c をダブルクリックして[ソース]ウィンドウを開いてください。



```
void main(void)
{
    long a[10], min, max;
    long j;
    int i;

    printf("### Data Input ###\n");

    for( i=0; i<10; i++ ){
        j = rand();
        if(j < 0){
            j = -j;
        }
        a[i] = j;
        printf("a[%d]=%ld\n", i, a[i]);
    }
    sort(a);
    printf("### Sorting results ###\n");
    for( i=0; i<10; i++ ){
        printf("a[%d]=%ld\n", i, a[i]);
    }
    min = a[0];
    max = a[9];
    min = 0;
    max = 0;
    change(a);
    min = a[9];
    max = a[0];
}
```

図 7-3 ソースウィンドウ (ソースプログラムの表示)

## 7.2.4 PCブレークポイントの設定

[ソース]ウィンドウによって、ブレークポイントを簡単に設定できます。以下のようにして sort 関数のコール箇所にブレークポイントを設定します。

- sort 関数コールを含む行にカーソルを移動して右クリックし、ポップアップメニューの[ブレークポイントの挿入/削除]を選択してください。

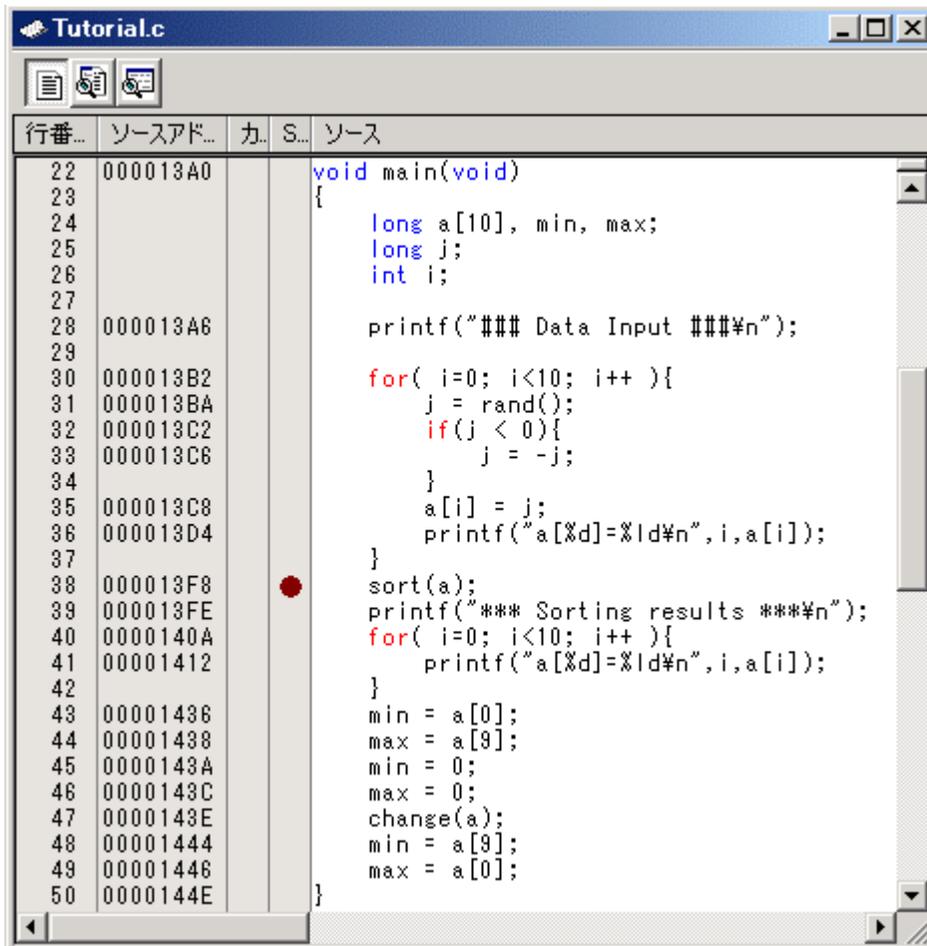


図 7-4 ソースウィンドウ (ブレークポイントの設定)

sort 関数コールを含む行に  を表示し、そのアドレスに PC ブレークポイントを設定したことを示します。

### 7.2.5 プロファイラの設定

- [表示]メニューから [パフォーマンス -> プロファイル]を選択し、[プロファイル]ウィンドウを開いてください。

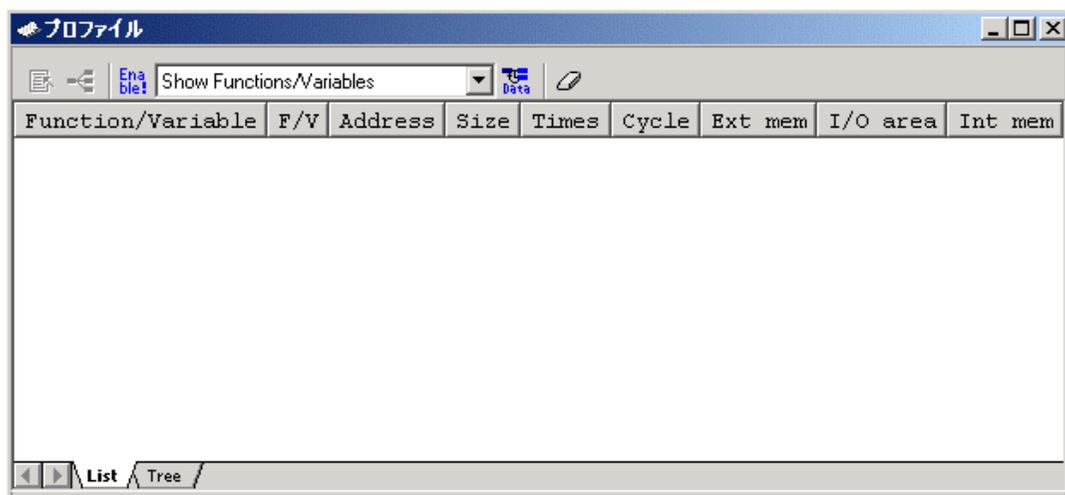


図 7-5 プロファイルウィンドウ

- [プロファイル]ウィンドウ上で右クリックしてポップアップメニューを表示し、[有効]を選択してプロファイラ情報取得を有効にしてください。

## 7.2.6 I/Oシミュレーションの設定

デモンストレーションプロジェクトでは、自動的にI/Oシミュレーションを設定しますので、設定を確認してください。

- [基本設定]メニューから [シミュレータ -> システム]を選択して、[シミュレータシステム]ダイアログボックスを開いてください。

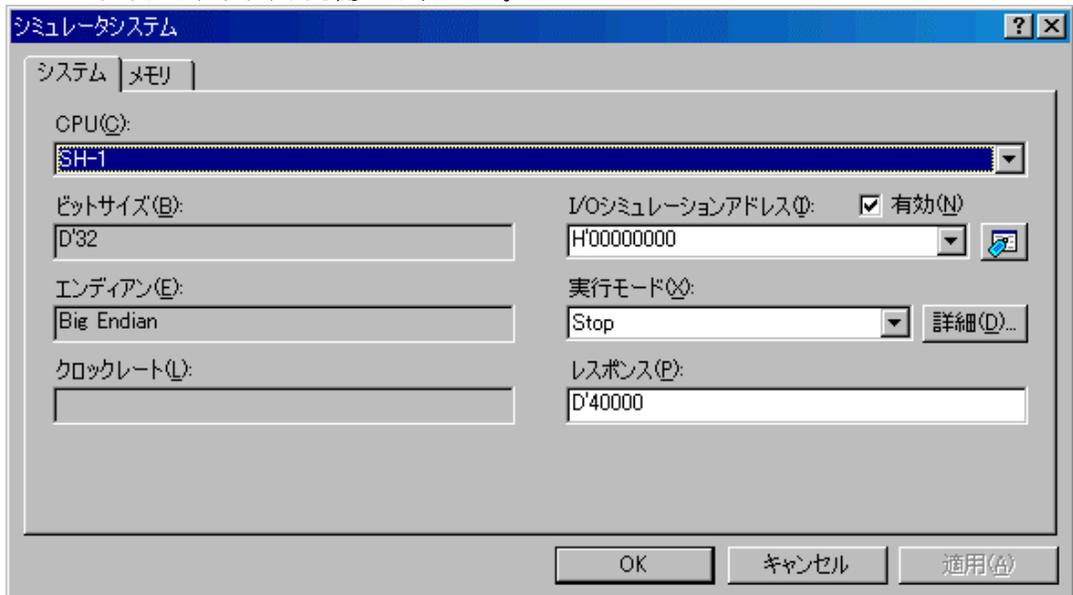


図 7-6 シミュレータシステムダイアログボックス (システムページ)

- [I/Oシミュレーションアドレス]の [有効]にチェックがあることを確認してください。
- [OK]ボタンをクリックして I/Oシミュレーションを有効にしてください。
- [表示]メニューから [CPU -> I/Oシミュレーション]を選択し、[I/Oシミュレーション]ウィンドウを開いてください。[I/Oシミュレーション]ウィンドウを開かなければ、I/Oシミュレーションが有効になりません。



図 7-7 I/O シミュレーションウィンドウ

### 7.2.7 トレース情報取得条件の設定

- [表示]メニューから [コード -> トレース]を選択し、[トレース]ウィンドウを開いてください。さらに、[トレース]ウィンドウ上で右クリックしてポップアップメニューを表示し、[設定...]を選択してください。

以下の[トレース取得]ダイアログボックスを表示します。

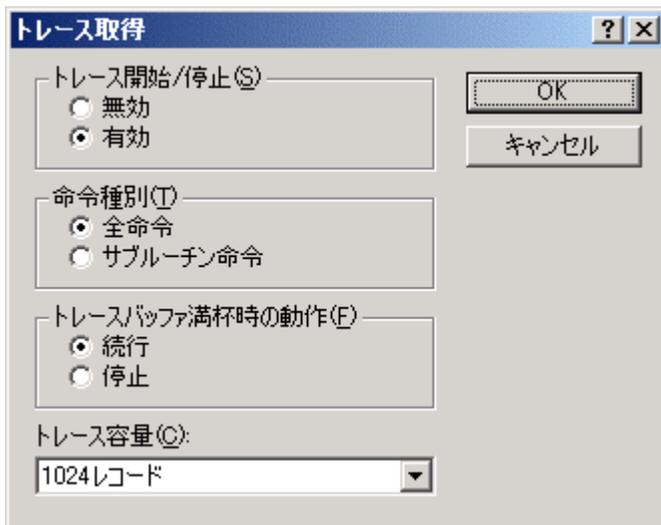


図 7-8 トレース取得ダイアログボックス

- [トレース取得]ダイアログボックスの [トレース開始/停止]を [有効]に設定し、[OK]ボタンをクリックしてトレース情報取得を有効にしてください。

## 7.2.8 スタックポインタ、プログラムカウンタの設定

プログラムを実行するために、プログラムカウンタおよびスタックポインタをリセットベクタから設定してください。サンプルプログラムのリセットベクタには、PC 値として H'800 が、SP 値として H'0FFFFFF0 が書いてあります。

- [デバッグ]メニューから [CPU のリセット]を選択するか、またはツールバー上の [CPU リセット]ボタンをクリックしてください。

リセットベクタからプログラムカウンタに H'800 を、スタックポインタに H'0FFFFFF0 を設定します。



図 7-9 CPU リセットボタン

## 7.3 デバッグ開始

### 7.3.1 プログラムの実行

- プログラムを実行するには、[デバッグ]メニューから [実行]を選択するか、またはツールバー上の[実行]ボタンをクリックしてください。



図 7-10 実行ボタン

プログラムはブレークポイントを設定したところまで実行します。プログラムが停止した位置を示すために [ソース]ウィンドウに を表示します。また 停止要因として[アウトプット]ウィンドウに”PC Breakpoint (PC:H’000013F8)”を表示します。

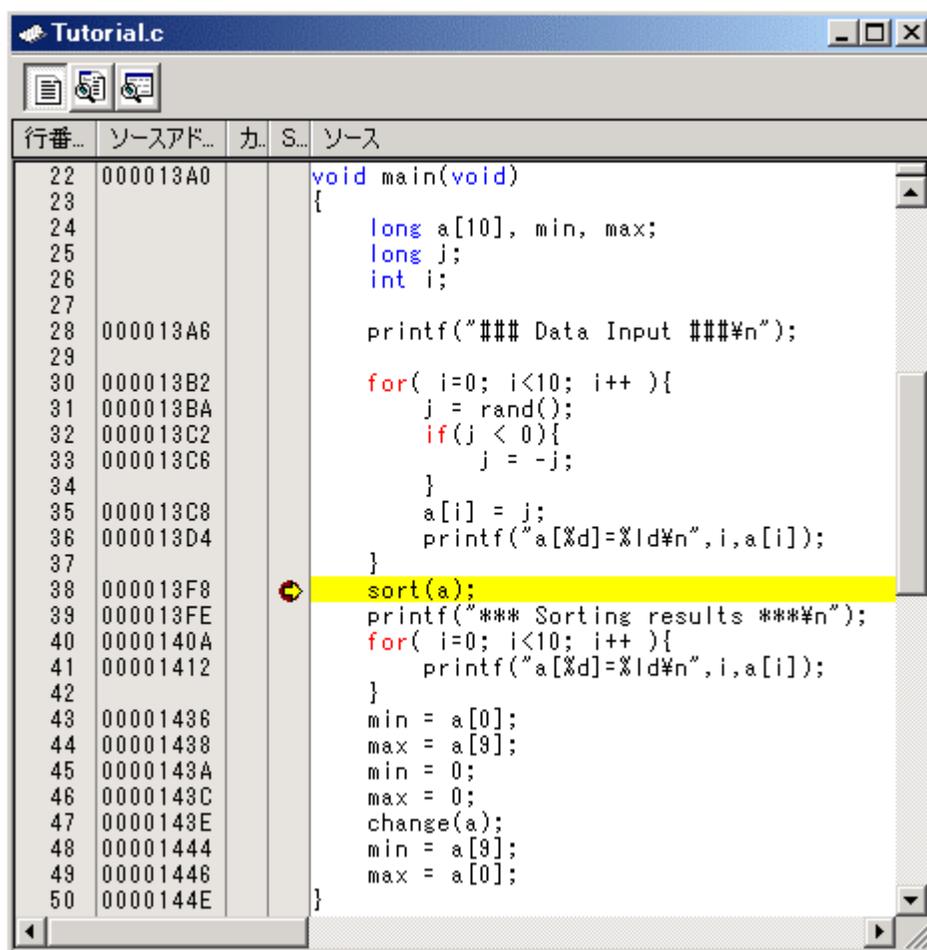
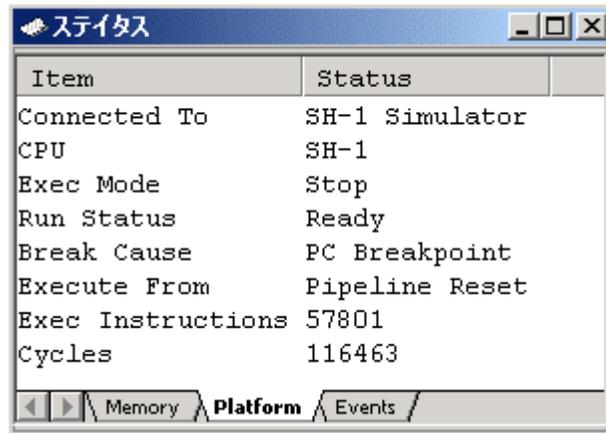


図 7-11 ソースウィンドウ (ブレーク状態)

[ステータス]ウィンドウで停止要因が確認できます。

- [表示]メニューから [CPU -> ステータス]を選択し、[ステータス]ウィンドウを開いてください。さらに、[ステータス]ウィンドウのうち [Platform]シートを表示してください。



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

Item	Status
Connected To	SH-1 Simulator
CPU	SH-1
Exec Mode	Stop
Run Status	Ready
Break Cause	PC Breakpoint
Execute From	Pipeline Reset
Exec Instructions	57801
Cycles	116463

At the bottom of the window, there are three tabs: 'Memory', 'Platform' (which is selected), and 'Events'.

図 7-12 ステータスウィンドウ

これは、以下の実行内容を表示します。

ブレークの原因は PCブレーク

パイプラインリセットからの実行

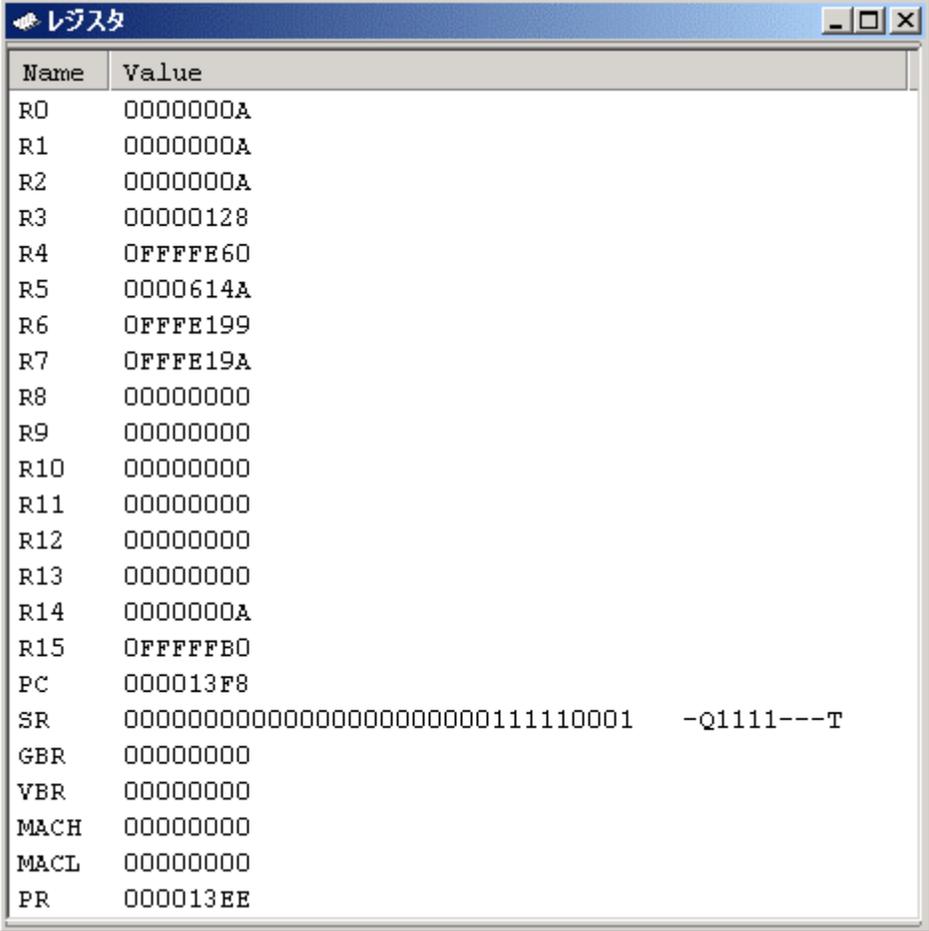
GOコマンドでの実行命令数は57,801命令

パイプラインリセットからの実行サイクル数は116,463サイクル

## 7. チュートリアル

[レジスタ]ウィンドウでレジスタの値が確認できます。

- [表示]メニューから [CPU -> レジスタ]を選択してください。



The screenshot shows a window titled 'レジスタ' (Registers) with a table of registers and their values. The registers listed are R0 through R15, PC, SR, GBR, VBR, MACH, MACL, and PR. The values are in hexadecimal format. The SR register has a unique value with a label '-Q1111---T'.

Name	Value
R0	0000000A
R1	0000000A
R2	0000000A
R3	00000128
R4	0FFFFFFE60
R5	0000614A
R6	0FFFE199
R7	0FFFE19A
R8	00000000
R9	00000000
R10	00000000
R11	00000000
R12	00000000
R13	00000000
R14	0000000A
R15	0FFFFFFB0
PC	000013F8
SR	00000000000000000000000000000000111110001 -Q1111---T
GBR	00000000
VBR	00000000
MACH	00000000
MACL	00000000
PR	000013EE

図 7-13 レジスタウィンドウ

プログラム停止時の各レジスタの値を確認することができます。

### 7.3.2 トレースバッファの使い方

トレースバッファを使って、命令実行の履歴を知ることができます。

- [表示]メニューから [コード -> トレース]を選択し、[トレース]ウィンドウを開いてください。ウィンドウの最上部までスクロールアップしてください。

PTR	Cycle	Address	Pipeline	Instruction	Access_Data	Source
-32767	0000057043	0000100E	fD>E	RTS	PC<-0000118C	RTS
-32766	0000057046	00001010	FF-D>E	NOP		NOP
-32765	0000057047	0000118C	FFDE>	ADD #FF, R13	R13<-00000000	
-32764	0000057049	0000118E	fD>E	MOV #00, R1	R1<-00000000	
-32763	0000057050	00001190	FFDE>	CMP/HI R1, R13	T<-(0)	
-32762	0000057052	00001192	fD>E	BT 00001180	T(0)	
-32761	0000057053	00001194	FFDE>MM	MOV.L @(00000008, R15), R6	R6<-00000001	return count;
-32760	0000057055	00001196	fD>E	BRA 0000119C	PC<-0000119C	
-32759	0000057058	00001198	FF-D>E	NOP		
-32758	0000057059	0000119C	FFDE>	MOV R6, R0	R0<-00000001	
-32757	0000057061	0000119E	fD>E	ADD #0C, R15	R15<-0FFFFE28	
-32756	0000057062	000011A0	FFDE>MM	LDS.L @R15+, PR	PR<-00003B98	
-32755	0000057064	000011A2	fD>EMM	MOV.L @R15+, R14	R14<-0FFFE01C	
-32754	0000057066	000011A4	FFD<E>MM	MOV.L @R15+, R13	R13<-00006146	
-32753	0000057068	000011A6	f<D>E	RTS	PC<-00003B98	}

図 7-14 トレースウィンドウ (トレース情報の表示)

## 7.3.3 トレース検索の実行

最初に、[トレース]ウィンドウ上で右クリックしてポップアップメニューを表示し、[トレース検索...]を選択して、[トレース検索]ダイアログボックスを開いてください。

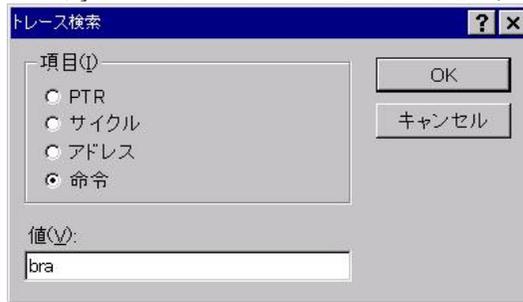


図 7-15 トレース検索ダイアログボックス

検索項目 [項目]と検索内容 [値]を設定して [OK]ボタンをクリックすると、トレース検索を実行します。該当するトレース情報があった場合、該当する最初の行を強調表示します。同じ検索内容 [値]でトレース検索を続ける場合は、[トレース]ウィンドウ上で右クリックしてポップアップメニューを表示し、[次を検索]を選択してください。次に該当する行を強調表示します。

PTR	Cycle	Address	Pipeline	Instruction	Access Data	Source
-32767	0000057043	0000100E	fD>E	RTS	PC<-0000118C	RTS
-32766	0000057046	00001010	FF-D>E	NOP		NOP
-32765	0000057047	0000118C	FFDE>	ADD #FF, R13	R13<-00000000	
-32764	0000057049	0000118E	fD>E	MOV #00, R1	R1<-00000000	
-32763	0000057050	00001190	FFDE>	CMP/HI R1, R13	T<-(0)	
-32762	0000057052	00001192	fD>E	BT 00001180	T(0)	
-32761	0000057053	00001194	FFDE>MW	MOV.L @(00000008, R15), R6	R6<-00000001	return count;
-32760	0000057055	00001196	fD>E	<b>BRA 0000119C</b>	<b>PC&lt;-0000119C</b>	
-32759	0000057058	00001198	FF-D>E	NOP		
-32758	0000057059	0000119C	FFDE>	MOV R6, R0	R0<-00000001	
-32757	0000057061	0000119E	fD>E	ADD #0C, R15	R15<-0FFFFFFE28	
-32756	0000057062	000011A0	FFDE>MW	LDS.L @R15+, PR	PR<-00003B98	
-32755	0000057064	000011A2	fD>EMW	MOV.L @R15+, R14	R14<-0FFFFFFE01C	
-32754	0000057066	000011A4	FFD<E>MW	MOV.L @R15+, R13	R13<-00006146	
-32753	0000057068	000011A6	f<D>E	RTS	PC<-00003B98	}

図 7-16 トレースウィンドウ (検索結果)

### 7.3.4 I/Oシミュレーションの確認

printf 関数で表示したランダムデータを[I/O シミュレーション]ウィンドウで確認することができます。

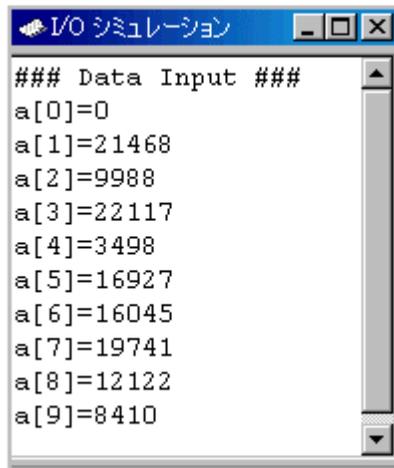


図 7-17 I/O シミュレーションウィンドウ

- [I/Oシミュレーション]ウィンドウは閉じないでください。

### 7.3.5 ブレークポイントの確認

プログラムに設定した全てのブレークポイントのリストを[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 -> イベントポイント]を選択してください。

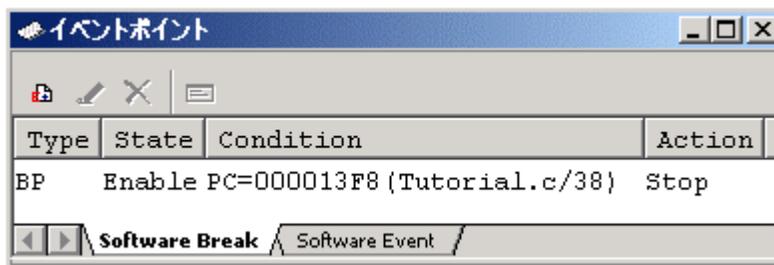


図 7-18 イベントポイントウィンドウ

[イベントポイント]ウィンドウによって、ブレークポイントの設定、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ができます。

- [イベントポイント]ウィンドウを閉じてください。

## 7.3.6 変数の参照

プログラムで使用する変数の値を[ウォッチ]ウィンドウで確認することができます。たとえば、プログラムの始めに宣言した long 型の配列 a を見る場合：

- [表示]メニューから [シンボル -> ウォッチ]を選択し[ウォッチ]ウィンドウを表示してください。さらに、[ウォッチ]ウィンドウ上で右クリックしてポップアップメニューを表示し、[シンボル登録...]を選択してください。

以下の [シンボル登録]ダイアログボックスを表示します。



図 7-19 シンボル登録ダイアログボックス

- 配列 “a” をタイプし、[OK]ボタンをクリックします。

[ウォッチ]ウィンドウに、long 型の配列 a を表示します。

配列 a の前にあるシンボル+をクリックすると、配列を拡張して表示します。

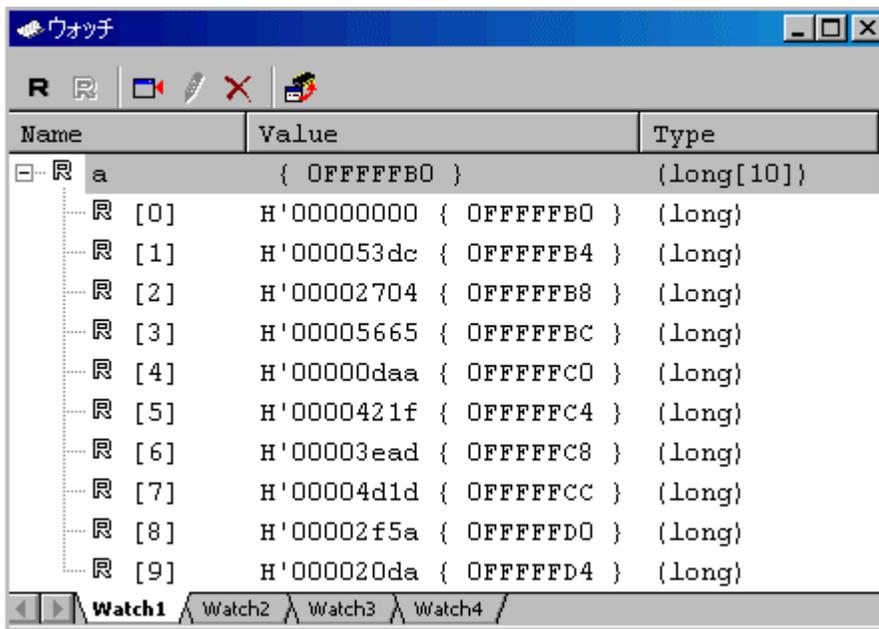


図 7-20 ウォッチウィンドウ

- [ウォッチ]ウィンドウを閉じてください。

### 7.3.7 プログラムのステップ実行

シミュレータ・デバッガは、プログラムのデバッグに有効な各種のステップメニューを備えています。

メニュー	説明
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）。
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。
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における次のステートメントで停止します。
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。

#### (1) [ステップイン]の実行

[ステップイン]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 7-21 ステップインボタン

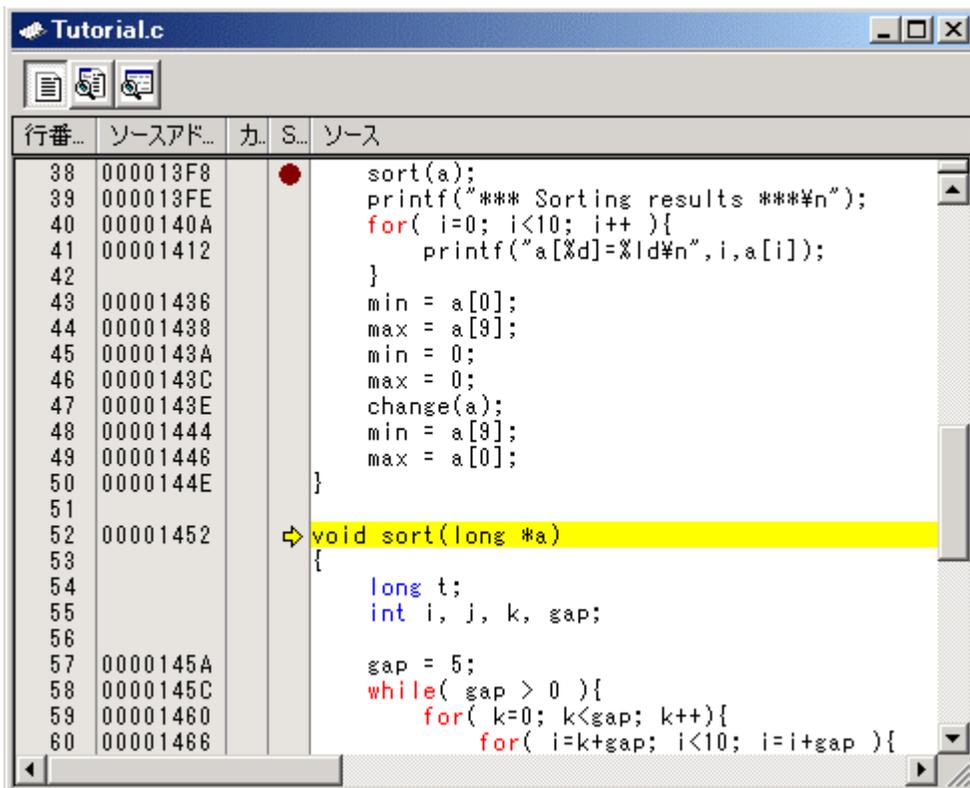


図 7-22 ソースウィンドウ (ステップイン)

## 7. チュートリアル

- [ソース]ウィンドウの PC 位置表示 ( ) が、sort 関数の先頭のステートメントに移動します。

### (2) [ステップアウト]の実行

[ステップアウト]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 7-23 ステップアウトボタン

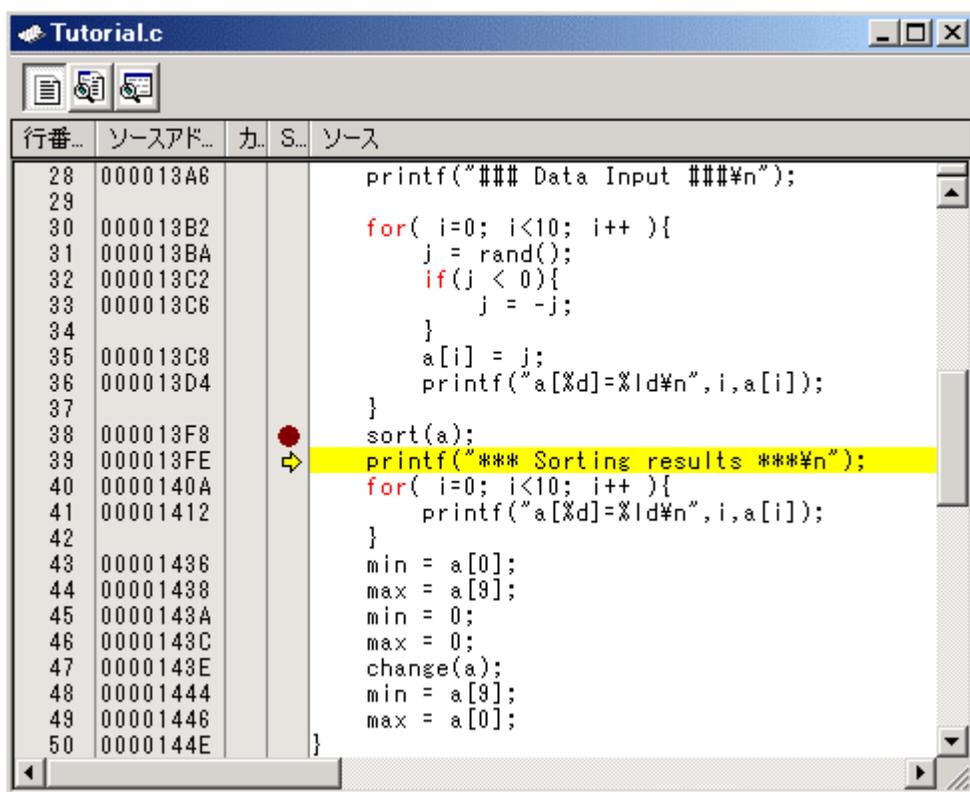


図 7-24 ソースウィンドウ (ステップアウト)

## (3) [ステップオーバー]の実行

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

Printf 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 7-25 ステップオーバーボタン

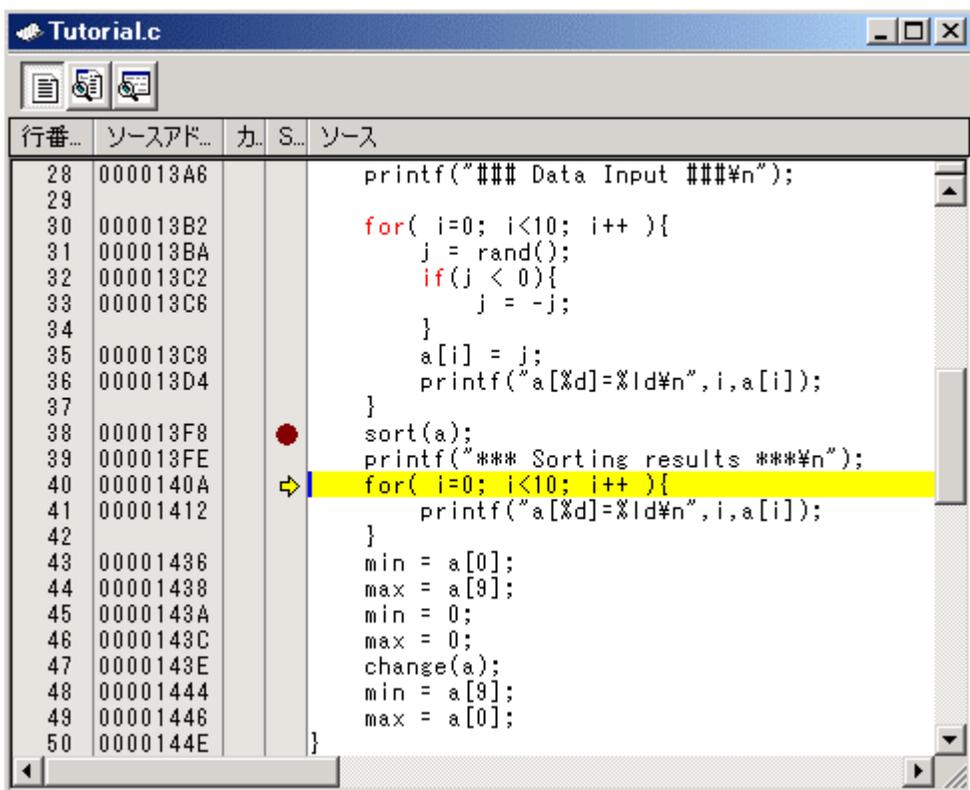


図 7-26 ソースウィンドウ (ステップオーバー)

printf 関数を実行すると、[I/O シミュレーション]ウィンドウに\*\*\* Sorting results \*\*\*を表示します。

## 7.3.8 プロファイラ情報の確認

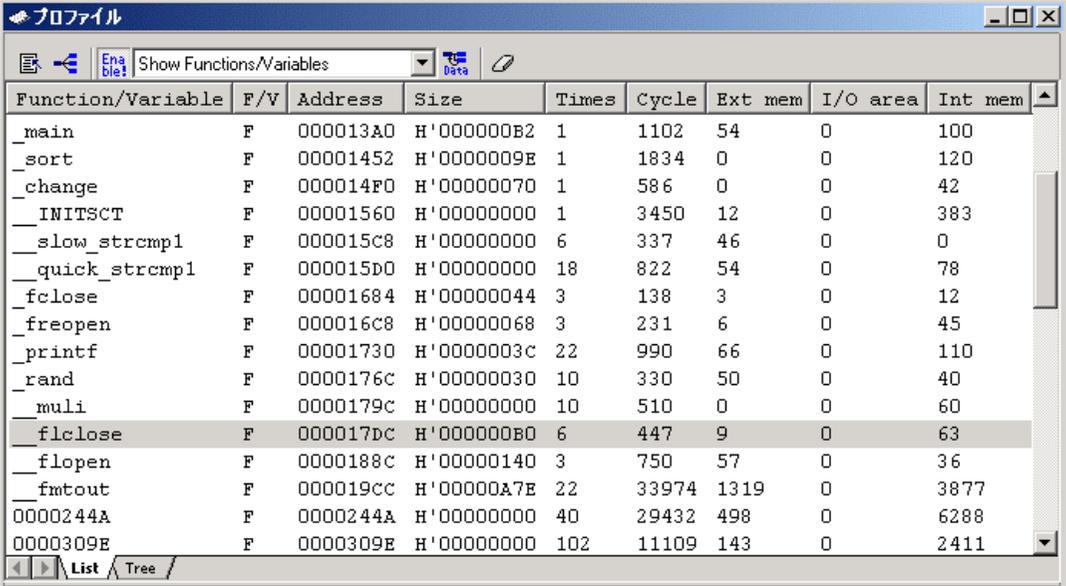
プロファイラ情報を[プロファイル]ウィンドウで確認することができます。

- [実行]ボタンをクリックして現在の PC から継続実行すると、SLEEP 命令を実行して停止します。

## (1) [List]シート

プロファイラ情報をリスト形式で表示します。

- [表示]メニューから [パフォーマンス -> プロファイル]を選択し、[プロファイル]ウィンドウを開いてください。[List]シートを表示します。



Function/Variable	F/V	Address	Size	Times	Cycle	Ext mem	I/O area	Int mem
_main	F	000013A0	H'000000B2	1	1102	54	0	100
_sort	F	00001452	H'0000009E	1	1834	0	0	120
_change	F	000014F0	H'00000070	1	586	0	0	42
__INIT\$CT	F	00001560	H'00000000	1	3450	12	0	383
__slow_strncmp1	F	000015C8	H'00000000	6	337	46	0	0
__quick_strncmp1	F	000015D0	H'00000000	18	822	54	0	78
_fclose	F	00001684	H'00000044	3	138	3	0	12
_freopen	F	000016C8	H'00000068	3	231	6	0	45
_printf	F	00001730	H'0000003C	22	990	66	0	110
_rand	F	0000176C	H'00000030	10	330	50	0	40
_mul	F	0000179C	H'00000000	10	510	0	0	60
fclose	F	000017DC	H'000000B0	6	447	9	0	63
_flopen	F	0000188C	H'00000140	3	750	57	0	36
_fmtout	F	000019CC	H'00000A7E	22	33974	1319	0	3877
0000244A	F	0000244A	H'00000000	40	29432	498	0	6288
0000309E	F	0000309E	H'00000000	102	11109	143	0	2411

図 7-27 List シート (プロファイルウィンドウ)

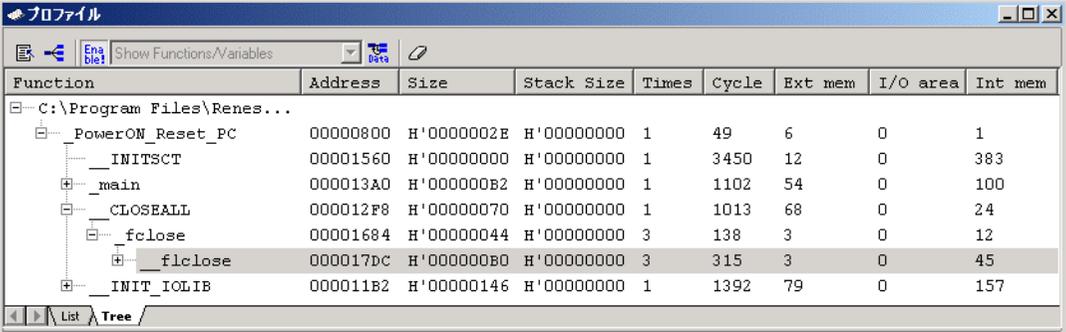
fclose 関数を 6 回コールし、実行サイクルは 447 サイクル、外部メモリを 9 回アクセスし、内部メモリを 63 回アクセスしたことがわかります。

コール回数が多い関数や遅いメモリを多くアクセスする関数など、プログラム性能のクリティカルパスを探すことができます。

## (2) [Tree]シート

プロファイラ情報をツリー形式で表示します。

- [Tree]シートを選択してください。関数名をダブルクリックすることによりツリー構造を拡張または縮小します。



Function	Address	Size	Stack Size	Times	Cycle	Ext mem	I/O area	Int mem
PowerON_Reset_PC	00000800	H'0000002E	H'00000000	1	49	6	0	1
__INIT_SCT	00001560	H'00000000	H'00000000	1	3450	12	0	383
__main	000013A0	H'000000E2	H'00000000	1	1102	54	0	100
__CLOSEALL	000012F8	H'00000070	H'00000000	1	1013	68	0	24
__fclose	00001684	H'00000044	H'00000000	3	138	3	0	12
__fclose	000017DC	H'000000B0	H'00000000	3	315	3	0	45
__INIT_IOLIB	000011B2	H'00000146	H'00000000	1	1392	79	0	157

図 7-28 Tree シート (プロファイルウィンドウ)

\_\_fclose 関数が \_\_fclose 関数から 3 回コールされ、その時の実行サイクルは 315 サイクル、外部メモリを 3 回アクセスし、内部メモリを 45 回アクセスしたことがわかります。

## 7. チュートリアル

---

### (3) プロファイル - チャートウィンドウ

[プロファイル - チャート]ウィンドウで関数の呼び出し関係を表示します。

- [プロファイル]ウィンドウ上で `_fclose` 関数を選択してから、右クリックしてポップアップメニューを表示し、 [チャート表示] を選択して [プロファイル - チャート]ウィンドウを表示してください。

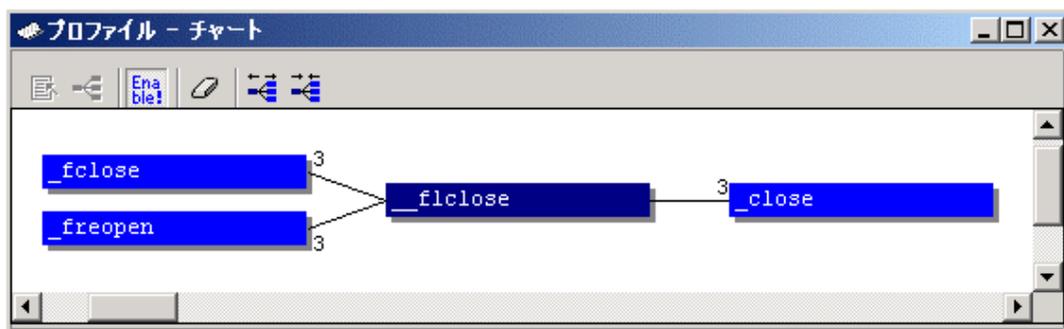


図 7-29 プロファイル - チャートウィンドウ

`_fclose` 関数が `_fclose` 関数から 3 回、`_freopen` 関数から 3 回コールされたことがわかります。また、`_close` 関数を 3 回コールしたことがわかります。

以上で、シミュレータ・デバッガを使用したチュートリアルは終了です。

---

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  
SuperH™ RISC engine シミュレータ / デバッガ

発行年月日 2007年7月25日 Rev.4.00

発行 株式会社ルネサス テクノロジ 営業統括部  
〒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

株式会社ルネサス テクノロジ 営業統括部 〒100-0004 東京都千代田区大手町2-6-2 日本ビル

営業お問合せ窓口  
株式会社ルネサス販売

# RENESAS

<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	浜	支	社	〒212-0058	川崎市幸区鹿島田890-12 (新川崎三井ビル)	(044) 549-1662
西	東	京	支	〒190-0023	立川市柴崎町2-2-23 (第二高島ビル2F)	(042) 524-8701
東	北	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	わ	き	支	〒970-8026	いわき市平小太郎町4-9 (平小太郎ビル)	(0246) 22-3222
茨	城	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	潟	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	本	支	社	〒390-0815	松本市深志1-2-11 (昭和ビル7F)	(0263) 33-6622
中	部	支	社	〒460-0008	名古屋市中区栄4-2-29 (名古屋広小路プレイス)	(052) 249-3330
関	西	支	社	〒541-0044	大阪府中央区伏見町4-1-1 (明治安田生命大阪御堂筋ビル)	(06) 6233-9500
北	陸	支	社	〒920-0031	金沢市広岡3-1-1 (金沢パークビル8F)	(076) 233-5980
広	島	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
島	取	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	州	支	社	〒812-0011	福岡市博多区博多駅前2-17-1 (博多プレステージ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：コンタクトセンター E-Mail: [csc@renesas.com](mailto:csc@renesas.com)



SuperH™ RISC engine シミュレータ／デバッガ  
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

RJJ10B0218-0400