

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日

ルネサスエレクトロニクス株式会社

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

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

ご注意書き

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

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

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

SH7730 グループ

SH7730 初期設定例

要旨

この資料は、SH7730 の起動時に必要な設定項目の例を示します。

動作確認デバイス

SH7730

目次

1. はじめに.....	3
2. 基本設定事項.....	4
3. 応用例の説明.....	9
4. 参考プログラムリスト.....	16
5. 参考ドキュメント.....	35

本書の位置づけ

本資料は、SH7730 の初期設定の手順を解説したものです。SH7730 関連のアプリケーションノートとして、まず最初に参照されると考えられることから、関連する基礎知識についても簡単な案内を加えるなど、SH-4A コアの SH マイコンを初めて使用される方にも配慮した内容となっています。

本資料の構成を以下に示します。

- 1 章では本アプリケーションノートの仕様、適用条件について記述しています。
- 2 章では本資料を読み進める上で最低限必要となる SH マイコンの開発環境や初期設定に関する基礎知識の案内、補足情報などについてまとめています。すでに SH マイコンに関する一通りの基礎知識や、初期設定に関する知識をお持ちの方は、この章を飛ばしていただいても構いません。
- 3 章では実際の初期設定に関する処理を説明し、初期設定で行うことや、考慮すべき点について記述しています。
- 4 章では 3 章で説明した各処理の参考プログラムを掲載しています。
- 5 章には参考ドキュメントの一覧を記載しています。

SH7730 関連アプリケーションノート

下記アプリケーションノートもご参照ください。各使用機能における各設定の詳細説明を記述しています。

「SH7730 グループ アプリケーションノート BSC SDRAM 設定例 (32 ビットバス) (RJJ06B0867)」

: BSC における外付け SDRAM の初期設定値についての詳細説明を記述しています。

「SH7730 グループ アプリケーションノート BSC NOR 型フラッシュメモリ設定例 (RJJ06B0866)」

: BSC における外付けフラッシュメモリの初期設定値についての詳細説明を記述しています。

「SH7730 グループ アプリケーションノート キャッシュ キャッシュメモリの設定例 (RJJ06B0868)」

: 命令/オペランドキャッシュを ON にする初期設定値についての詳細説明を記述しています。

「SH7730 グループ アプリケーションノート キャッシュ オペランドキャッシュの書き戻し例 (RJJ06B0923)」

: オペランドキャッシュからメモリへの書き戻しについての詳細説明を記述しています。

1. はじめに

1.1 仕様

リセット解除後に、クロックパルス発振器 (CPG)、バスステートコントローラ (BSC)、およびキャッシュの初期設定を行います。

1.2 使用機能

クロックパルス発振器 (CPG)

バスステートコントローラ (BSC)

キャッシュ

1.3 適用条件

評価ボード:	アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
外付けメモリ (エリア 0):	NOR 型フラッシュメモリ 4 M バイト Spansion 製 S29AL032D70TFI04
(エリア 3):	SDR-SDRAM 32 M バイト(16 M バイト × 2 個) Samsung 製 K4S281632F-UC75
マイコン:	SH7730 (R8A77301)
動作周波数	CPU クロック: 266.66 MHz SuperHyway バスクロック: 133.33 MHz バスクロック: 66.66 MHz 周辺クロック: 33.33 MHz
エリア 0 バス幅:	16 ビット固定 (MD3 端子 = Low レベル)
クロック動作モード:	モード 2 (MD0 端子 = Low レベル, MD1 端子 = High レベル)
エンディアン:	ビッグエンディアン (MD5 端子 = Low レベル)
ツールチェーン:	ルネサス テクノロジ製 SuperH RISC engine Standard Toolchain Ver.9.3.0.0
コンパイルオプション:	High-performance Embedded Workshop でのデフォルト設定 -cpu=sh4a -include="\$(PROJDIR)\inc" -object="\$(CONFIGDIR)\\$(FILELEAF).obj" -debug -optimize=0 -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo

2. 基本設定事項

この章では、SH7730 初期設定で必要とされる基礎知識、一般的に行われることや、考慮すべきことを記述します。

2.1 必要となる基礎知識

本資料の利用にあたっては、あらかじめ 5 章に記載したマニュアルを入手の上、開発環境および SH7730 CPU について理解しておいてください。以下にマニュアルを参照する上でのポイントを示します。

(1) 開発環境について

- High-performance Embedded Workshop の設定方法
本資料では、開発環境として High-performance Embedded Workshop を使用します。設定方法については、「High-performance Embedded Workshop スタートアップガイド」および「SuperH RISC engine C/C++ コンパイラパッケージ アプリケーションノート: <導入ガイド> スタートアップルーチンガイド SH-3, SH-4, SH-4A 編 (RJJ06J0016)」を参照ください。使用方法については、High-performance Embedded Workshop のメニューバーにあるヘルプを参照ください。
- フラッシュメモリダウンロード方法
本資料では、E10A-USB エミュレータの「フラッシュメモリダウンロード機能」を使い、外付けフラッシュメモリ領域にユーザプログラムをダウンロードします。使用方法については、「E10A-USB エミュレータ用 フラッシュメモリダウンロードプログラム アプリケーションノート (RJJ10J0957)」を参照ください。

(2) SH7730 CPU について

- セクション
「SuperH RISC engine C/C++ コンパイラ, アセンブラ, 最適化リンケージエディタ コンパイラパッケージ V.9.01 ユーザーズマニュアル (RJJ10J1885)」の「プログラミング」を参照ください。
- レジスタの構成
「SH7730 グループ ハードウェアマニュアル (RJJ09B0339)」の「レジスタ構成」を参照ください。
- 命令セット
「SH7730 グループ ハードウェアマニュアル (RJJ09B0339)」の「命令セット」, 「SuperH RISC engine C/C++ コンパイラ, アセンブラ, 最適化リンケージエディタ コンパイラパッケージ V.9.01 ユーザーズマニュアル (RJJ10J1885)」を参照ください。
- 例外処理
「SH7730 グループ ハードウェアマニュアル (RJJ09B0339)」の「例外処理」を参照ください。
- 仮想アドレス, P0 ~ P4 領域, 物理アドレス
「SH7730 グループ ハードウェアマニュアル (RJJ09B0339)」の「メモリマネジメントユニット (MMU)」を参照ください。
- 空間分割 (エリア 0 ~ エリア 7), シャドウ空間, アドレスマップ
「SH7730 グループ ハードウェアマニュアル (RJJ09B0339)」の「バスステートコントローラ (BSC)」を参照ください。
- キャッシュ
「SH7730 グループ アプリケーションノート キャッシュ キャッシュメモリの設定例 (RJJ06B0868)」, 「SH7730 グループ アプリケーションノート キャッシュ オペランドキャッシュの書き戻し例 (RJJ06B0923)」を参照ください。

- BSC の設定
 - 「SH7730 グループ アプリケーションノート BSC SDRAM 設定例 (32 ビットバス) (RJJ06B0867)」,
 - 「SH7730 グループ アプリケーションノート BSC NOR 型フラッシュメモリ設定例 (RJJ06B0866)」を参照ください。

2.2 開発環境の準備

(1) SH7730 評価ボードの準備

本資料では、アルファプロジェクト製 SH7730 評価ボード (型番 AP-SH4-1A) を使用します。また、プログラムを評価ボード上の外付けフラッシュメモリに書き込みますので、「E10A-USB エミュレータ用 フラッシュメモリダウンロードプログラム アプリケーションノート (RJJ10J0957)」を参照のうえ、E10A-USB エミュレータが接続できる状態にします。

(2) High-performance Embedded Workshop の環境準備

High-performance Embedded Workshop プロジェクトを新規に立ち上げて、以下の設定をします。

- プロジェクト名: sh7730 *任意
- CPU 種別: SuperH RISC engine
- CPU シリーズ: SH-4A
- CPU タイプ: SH7730
- スタックポインタアドレス: 「表 1 セクション配置」のスタック領域の値を設定します。
- ターゲット: SessionSH-4A_E10A-USB_SYSTEM にチェックを入れます。

上記設定で、新規プロジェクトを立ち上げると以下のファイルが自動生成されます。自動生成される内容や、詳細については「SuperH RISC engine C/C++ コンパイラパッケージ アプリケーションノート: <導入ガイド> スタートアップルーチンガイド SH-3, SH-4, SH-4A 編 (RJJ06J0016)」を参照ください。

- sh7730.c
- dbsct.c
- resetprg.c
- sbrk.c
- iodef.h
- sbrk.h
- stacksct.h
- typedef.h
- env.inc
- vect.inc
- intprg.src
- vecttbl.src
- vhandler.src

2.3 設定のポイントと注意点

(1) セクション配置

通常、表 1 に示したようにセクションを配置します。また、開発時のデバッグ等の際には、本来 ROM 領域に書き込むデータ（プログラム）等を ROM 領域でなく、RAM 領域に書き込む場合もあります。セクションのサイズや、そのセクションの配置が、キャッシング可能領域でよいか、またキャッシング不可能領域でよいか等をよく検討してください。

表 1 の設定については、基本的に High-performance Embedded Workshop が自動的に設定しますが、その他のセクションを追加する際は、個別に設定してください (*1, *2 は追加部分)。

表 1 セクション配置

セクション名	セクション用途	領域	配置アドレス(仮想アドレス)	
P	プログラム領域 (指定なしの場合)	ROM	0x00003000	P0 領域 (キャッシング可能, MMU アドレス変換可能)
C	定数領域	ROM		
C\$BSEC	未初期化データ領域用アドレス構造体	ROM		
C\$DSEC	初期化データ領域用アドレス構造体	ROM		
D	初期化データ (初期値)	ROM		
B	未初期化データ領域	RAM	0x0C000000	
R	初期化データ領域	RAM		
S	スタック領域	RAM		
INTHandler	例外/割り込みハンドラ	ROM	0x80000800	P1 領域 (キャッシング可能, MMU アドレス変換不可)
VECTTBL	リセットベクタテーブル 割り込みベクタテーブル	ROM		
INTTBL	割り込みマスクテーブル	ROM		
PIntPRG	割り込み関数	ROM		
SP_S* ¹	TLB ミスハンドラ専用スタック領域	RAM		
RSTHandler	リセットハンドラ	ROM	0xA0000000	P2 領域 (キャッシング不可, MMU アドレス変換不可)
PResetPRG	リセットプログラム	ROM		
PnonCache* ²	プログラム領域 (キャッシュ無効アクセス)	ROM		

[参考] セクションの設定方法

High-performance Embedded Workshop のメニューバーのビルド (B) → SuperH RISC engine Standard Toolchain → 最適化リンカ カテゴリ (Y) セクションで下記の画面が表示されます。追加, 編集ボタンを押下してセクションを設定できます。

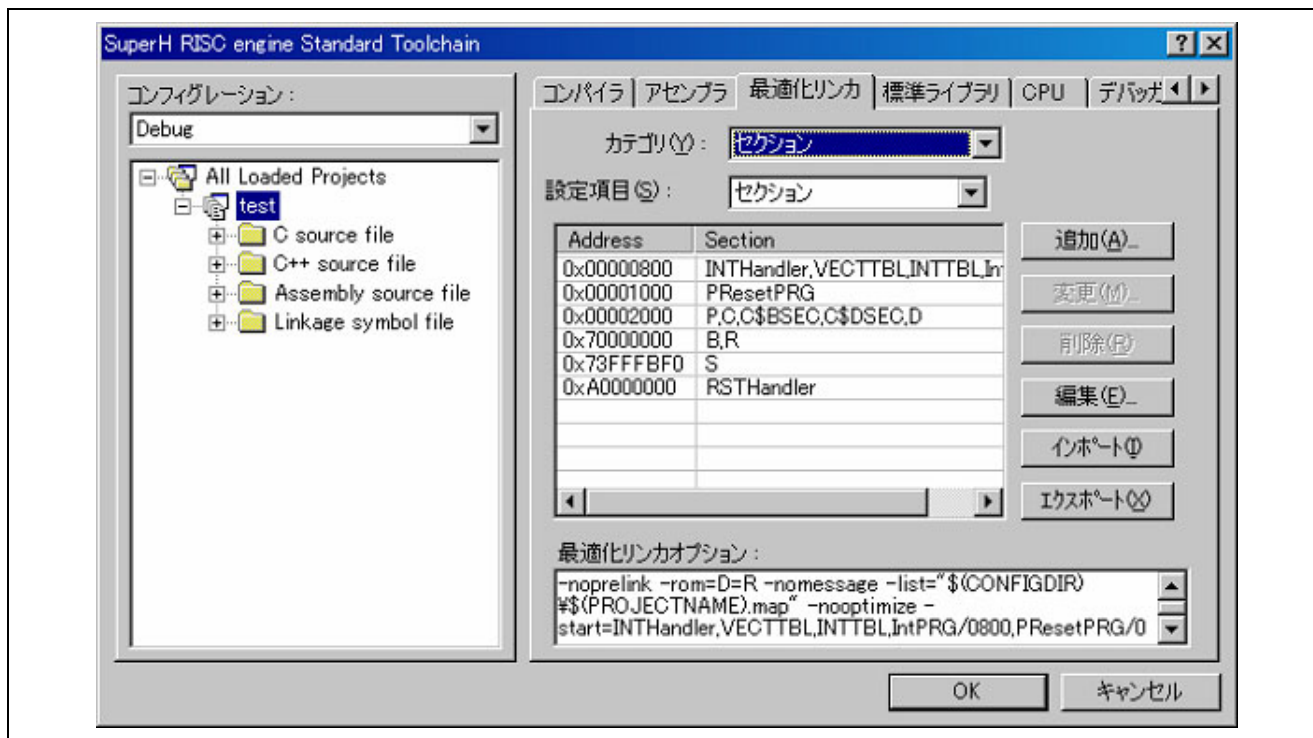


図 1 セクション設定画面

(2) スタックの設定

プログラムを動作させる際には、スタック領域が必要になります。スタックのサイズとスタックポインタアドレスを設定してください。プロジェクト起動時に設定した際の値で、High-performance Embedded Workshop が自動的に設定します。変更する際には、High-performance Embedded Workshop のメニューバーのプロジェクト (P) → 構成の編集 (E) → タブにてスタックを選択すると、スタック領域のアドレスとサイズを変更することができます。

また、MMU を利用する場合は、表 1 の*1 部分の TLB ミスハンドラ専用のスタック領域を考慮する必要があります。詳細は「3.2 章 参考プログラムの説明 (1) vhandler.src」を参照ください。MMU については、「SH7730 グループ ハードウェアマニュアル (RJ09B0339)」の「メモリアネジメントユニット (MMU)」を参照ください。

(3) WDT (ウォッチドッグタイマ) の設定

初期状態でタイマのカウントが始まり、オーバフローすると、内部リセットが発生します。システム起動時には WDT を停止させるか、定期的にカウントをクリアする処理が必要となります。

(4) FPSCR (浮動小数点ステータス/コントロールレジスタ) の設定

浮動少数点命令を単精度/倍精度として実行するかを選択を行います。システムの設計に基づいた設定をしてください。初期状態では、単精度モードになっています。

(5) BSC (バスステートコントローラ) の設定

BSC は、使用する外部メモリの読み出し、書き込みのタイミングスペックに合わせて設定してください。

C 言語で書かれた関数を呼び出す際にスタック領域を使用する可能性があるため、SDRAM のような外部メモリにスタック領域を配置する場合は、あらかじめ BSC の初期化が必要です。参考プログラムでは、BSC の初期化処理を例外ハンドラ (vhandler.src) 内に記述しています。詳細は「3.2 章 参考プログラムの説明 (1) vhandler.src」を参照ください。

(6) VBR (ベクタベースレジスタ) の設定

リセットベクタアドレスは H'A000 0000 に固定されていますが、リセット以外の一般例外、割り込みハンドラ先頭番地は、VBR の値に、決まったオフセット値 (TLB ミス例外は H'400、それ以外の一般例外は H'100、割り込みは H'600) を加えたものになります。

参考プログラムでは、例外ハンドラ (vhandler.src) 内で一般例外のハンドラ (_INTHandlerPRG) の開始番地をエクスポートしておき、リセット処理プログラム (resetprg.c) の PowerON_Reset() 関数の最初で VBR の設定を行っています。

なお、例外ハンドラ (vhandler.src) 内では、一般例外のオフセット (H'100) を基準にして、TLB ミスハンドラ (_TLBmissHandler) の開始番地は「.org H'300」、割り込みハンドラ (_IRQ_Handler) の開始番地は「.org H'500」と記述しています。

(7) メモリの初期化 (_INITSCT)

システム起動時に、初期値のあるグローバル変数は、ROM 領域 (D セクション) に配置しますが、変数として扱うため、RAM 領域 (R セクション) にコピーする必要があります。また、初期値のないグローバル変数は RAM (B セクション) に配置し、システム起動時に初期化する必要があります。High-performance Embedded Workshop が、これらの処理を自動的に行います。詳細は「3.2 章 参考プログラムの説明 (5) dbsct.c」を参照ください。

(8) キャッシュの設定

初期設定で、キャッシング可能領域 (P0, P1, P3) について、どの領域をキャッシュ有効にするか、また、その際のキャッシュメモリへの書き込みモード (ライトスルー/コピーバック) について検討してください。詳細は、「SH7730 グループ アプリケーションノート キャッシュ キャッシュメモリの設定例 (RJJ06B0868)」, 「SH7730 グループ アプリケーションノート キャッシュ オペランドキャッシュの書き戻し例 (RJJ06B0923)」を参照ください。

(9) SR (ステータスレジスタ) の設定

SR は、特権モード/ユーザモードの選択、汎用レジスタバンク指定、例外、割り込みを管理します。システムの設計に基づいた設定をしてください。参考プログラムでは、

- 特権モード
- 汎用レジスタバンク 0 選択
- 例外/割り込みブロックビット解除

を設定しています。

3. 応用例の説明

前章をふまえて、以下では High-performance Embedded Workshop が自動生成するソースコードに修正、追加を加え、SH7730 の初期設定環境を実際に作成していきます。

なお、SH7730 の各アプリケーションノートでは、本資料で説明する例を初期設定プログラムとして使用することを前提としています。

3.1 High-performance Embedded Workshop 自動生成環境の変更点

本資料の環境では、High-performance Embedded Workshop が自動生成した環境に対して以下の部分について、変更・追加・削除します。

- セクション配置:
表 1 に示すように、変更と追加を行います。
- キャッシュ有効化処理:
PowerON_Reset() 関数 (図 2 の(10)) でキャッシュを有効にします。
- FPSCR の初期化設定タイミング:
PowerON_Reset() 関数内で設定されたものを、リセットハンドラ (図 2 の(4)) 内で設定するように変更します。
- CPG の設定処理:
リセットハンドラ (図 2 の(5)) で、1.3 章で示した適用条件のように CPG を設定します。
- BSC の設定:
外部メモリ (フラッシュメモリ、SDRAM) を使用するため、リセットハンドラ (図 2 の(6)) 内に、BSC の初期化処理を追加します。
- RAMCR 設定:
High-performance Embedded Workshop が自動生成する環境では、PowerON_Reset() 関数を抜ける前に RAMCR (内蔵メモリ制御レジスタ) の RMD (内蔵メモリアクセスモードビット) を 1 に設定 (ユーザモードにおける内蔵メモリアクセスを許可) しています。しかし、本プログラムでは以後の処理も特権モードで動作させる前提のため、この設定は省略しています。
- intprg.src:
一般例外、割り込み関数をアセンブラ形式から C 言語形式に変更します。

【注】 参考プログラムでは、MMU を利用しませんが、利用する場合には以下の点も考慮します。

- ・ TLB ミスハンドラの追加
- ・ TLB ミス発生時専用のスタック領域の設定追加

3.2 参考プログラムの説明

初期設定プログラムは以下の 8 つのソースファイルから構成されています。

- (1) vhandler.src
- (2) vecttbl.src
- (3) resetprg.c
- (4) stachsct.h
- (5) dbsct.c
- (6) sh7730.c
- (7) intprg.c
- (8) vect.inc

(1) vhandler.src

例外 (リセット, 一般例外, 割り込み) が発生すると, まずこの例外ハンドラ (vhandler.src) が実行されます。vhandler.src には各例外のハンドラ処理と BSC の初期化処理を記述しています。リセットとリセット以外の例外では, ハンドラ内の処理が異なりますが, 詳しくは「SuperH RISC engine C/C++ コンパイラパッケージ アプリケーションノート: <導入ガイド>スタートアップルーチンガイド SH-3, SH-4, SH-4A 編 (RJJ06J0016)」を参照ください。

パワーオンリセットの場合は, リセットハンドラ (_Reset_handler) が起動されます。本アプリケーションノートで使用するリセットハンドラは, High-performance Embedded Workshop が自動生成するファイルとは異なり, 命令キャッシュとオペランドキャッシュの無効化 (図 2 の(3)), FPSCR の設定 (図 2 の(4)), CPG の設定 (図 2 の(5)), BSC の初期化 (図 2 の(6)) を追加しています。また, TLB ミスハンドラも後述する理由により変更しています。

FPSCR の初期値は, 浮動小数点演算の転送モードを 32 ビットとしていますので, アプリケーションの仕様に合わせて変更してください。High-performance Embedded Workshop の自動生成では, PowerON_Reset() 関数内で, set_fpscr (FPSCR_Init) 関数を呼び出して初期化していますが, この処理はパワーオンリセット以外のリセット (マニュアルリセット等) でも実行するため, リセットハンドラ内で処理するように変更しています。

スタック領域は, 初期化処理が必要な外部 SDRAM に設定します。C 言語で書かれた関数を呼び出す時点でスタック領域を使用する可能性があるため, BSC の初期化前にスタック領域へアクセスされることを避けるため, リセットハンドラの前頭部分で BSC を初期化しています。

MMU を利用する場合には, TLB ミスについても考慮する必要があります。High-performance Embedded Workshop が自動生成する TLB ミスハンドラは, TLB ミスハンドラ以外の例外ハンドラと割り込みのハンドラ, また例外/割り込み以外のその他のプログラムとスタック領域を共通に使用しています。そのため, もしスタック領域をアドレス変換可能な P0, もしくは P3 領域に配置していると, TLB ミスハンドラでスタックを積む際に再度 TLB ミスの例外が発生しマニュアルリセットが発生します。

本プログラムの TLB ミスハンドラは, TLB によるアドレス変換不可となる P1 領域に TLB ミス発生時専用のスタック領域 (サイズ H'200) を設けて, TLB ミスハンドラを抜けるまでの間, 専用のスタック領域を使用します。これにより TLB ミスハンドラ中の TLB ミス例外発生を防ぎます。

なお, CPU 動作モードレジスタ (CPUOPM) の割り込み動作モード切り替えビットを使用し, SR.IMASK を受理した割り込みレベルに自動設定する場合, あるいは多重割り込みを使用する場合には, この例外ハンドラ (vhandler.src) の各ハンドラ処理を修正してください。

High-performance Embedded Workshop が自動生成するソースプログラムのイベントハンドラ処理では, ともに例外 (リセット, 一般例外, 割り込み) ベクタテーブル (_INT_Vectors) を参照し, 例外コード (一般例外の場合: 例外事象レジスタ (EXPEVT), 割り込みの場合: 割り込み事象レジスタ (INTEVT)) によって, 起動させる一般例外関数または, 割り込み関数を決定します。一般例外 FPU 抑止例外と DMAC (DEI0), スロット FPU 抑止例外と DMAC (DEI1) の 2 つについては, 例外コードに設定される値が重複するため, High-performance Embedded Workshop が自動生成するイベントハンドラ処理ではこれらについて区別ができません。そのため, 一般例外 FPU 抑止例外とスロット FPU 抑止例外については, High-performance Embedded Workshop が自動生成するソースプログラムのイベントハンドラ処理でも区別できるように回避処理を埋め込んでいます。

High-performance Embedded Workshop が自動生成するソースプログラムの例外ハンドラ処理では、各例外のハンドラ処理前に多重割り込みが行えるように、BL ビットをクリアする処理が行われます。そのため、NMI 割り込み処理中でも、NMI 割り込みを受け付けてしまいます。その回避処理として例外コードが NMI 割り込みの場合は、BL ビットをクリアする処理を行わないようにします。

(2) vecttbl.src

例外（リセット、一般例外、割り込み）ベクタテーブル、割り込みマスクテーブルを記述しています。上記 vhandler.src の処理の中でこのテーブルを参照して、それぞれの例外処理関数 (resetprg.c または intprg.c) に処理を移します。

(3) resetprg.c

リセット処理プログラムとして、PowerON_Reset() 関数を記述しています (図 2 の(7))。本アプリケーションノートで記述する PowerON_Reset() 関数は、High-performance Embedded Workshop が自動生成するファイルとは異なり、HardwareSetup() 関数を呼び出していません。B, R, S セクションを初期化が必要な外部 SDRAM へ配置するため、BSC の初期化はリセットハンドラ (vhandler.src) 内に記述しています。

PowerON_Reset() 関数には、VBR (ベクタベースレジスタ) の設定 (図 2 の(8))、_INITSCT() 関数の呼び出し (図 2 の(9))、キャッシュの有効化関数の呼び出し (図 2 の(10))、SR (ステータスレジスタ) の設定 (図 2 の(11))、メイン関数の呼び出し (図 2 の(12)) などを記述します。

本プログラムでは、周辺モジュールを使用するときに、内部レジスタを設定する関数をメイン関数から呼び出すことを想定しているため、PowerON_Reset() 関数では SR (ステータスレジスタ) を特権モードに設定しています。ユーザモードで使用される場合には、特権モードでのみ使用可能な命令を記述しないでください。

また、High-performance Embedded Workshop が自動生成するファイルでは、PowerON_Reset() 関数を抜ける前に RAMCR を設定しますが、本プログラムでは行っていません。そのため処理モードをユーザモードに変えて、内蔵メモリへアクセスすると、内蔵メモリアクセス保護機能によりアドレスエラー例外が発生します。ユーザモードへ変更する際は RAMCR への設定 (RMD ビット = 1) を行うようにしてください。

(4) stacksct.h

スタック領域のサイズを指定しています (初期値 H'400)。スタック領域のサイズを変更する場合は、このファイルを直接変更することは避けてください (High-performance Embedded Workshop のメニューバーのプロジェクト (P) → 構成の編集 (E) → タブにてスタックを選択すると、スタック領域のアドレスとサイズを変更することができます)。

(5) dbsct.c

High-performance Embedded Workshop が自動生成するセクション初期化用のファイルです。初期化データセクション (D, R セクション) と未初期化データセクション (B セクション) の先頭アドレス、最終アドレスを本ファイルで定義し、resetprg.c の PowerON_Reset() 関数内で _INITSCT() 関数を呼び出すことにより、B セクションの 0 クリア、および D セクションから R セクションへのコピー処理を行います (図 2 の(9))。

ROM 化支援機能を用いて RAM 上でプログラムを動作させる場合は、_INITSCT() 関数で該当のセクションをコピーさせるために、このファイルに対してプログラムを転送するアドレスなどを追加します。

(6) sh7730.c

初期化処理終了後に呼び出されるメイン関数を記述しています (図 2 の (12))。ユーザプログラムなどを記述される場合はメインルーチンの中に記述してください。High-performance Embedded Workshop が自動生成するソースプログラムでは、動作設定を hwsetup.c で行うことになっていますが、本プログラムでは周辺モジュールの動作設定の関数は、本メイン関数から呼び出すことを想定しています。

(7) intprg.c

リセット以外の一般例外と割り込みについて、ハンドラ (vhandler.src) より呼び出される処理プログラム (中身が空の関数のダミー関数) を記述しています。周辺機能割り込みを使用するような場合は、本プログラムを元に、ダミー関数を書き換えて新しく関数を作る (vect.inc, vecttbl.src も関数名に合わせて変更する) か、このダミー関数の中から別途作成された関数を呼び出すようにしてください。

なお、割り込み要求フラグのクリアなどの処理は、ハードウェアマニュアルの記載内容に従って記述してください。

(8) vect.inc

intprg.c で記述したおのこの一般例外、割り込みの処理プログラムを vhandler.src から参照するために、vect.inc に外部参照シンボル宣言を記述しています。intprg.c のダミー関数を書き換えて新しく割り込み関数を作成する場合には関数名に合わせて変更してください。ダミー関数の中から別途作成された関数を呼び出す場合には、このファイルの変更は特に必要ありません。

図 2 に、パワーオンリセットからの処理フローを示します。

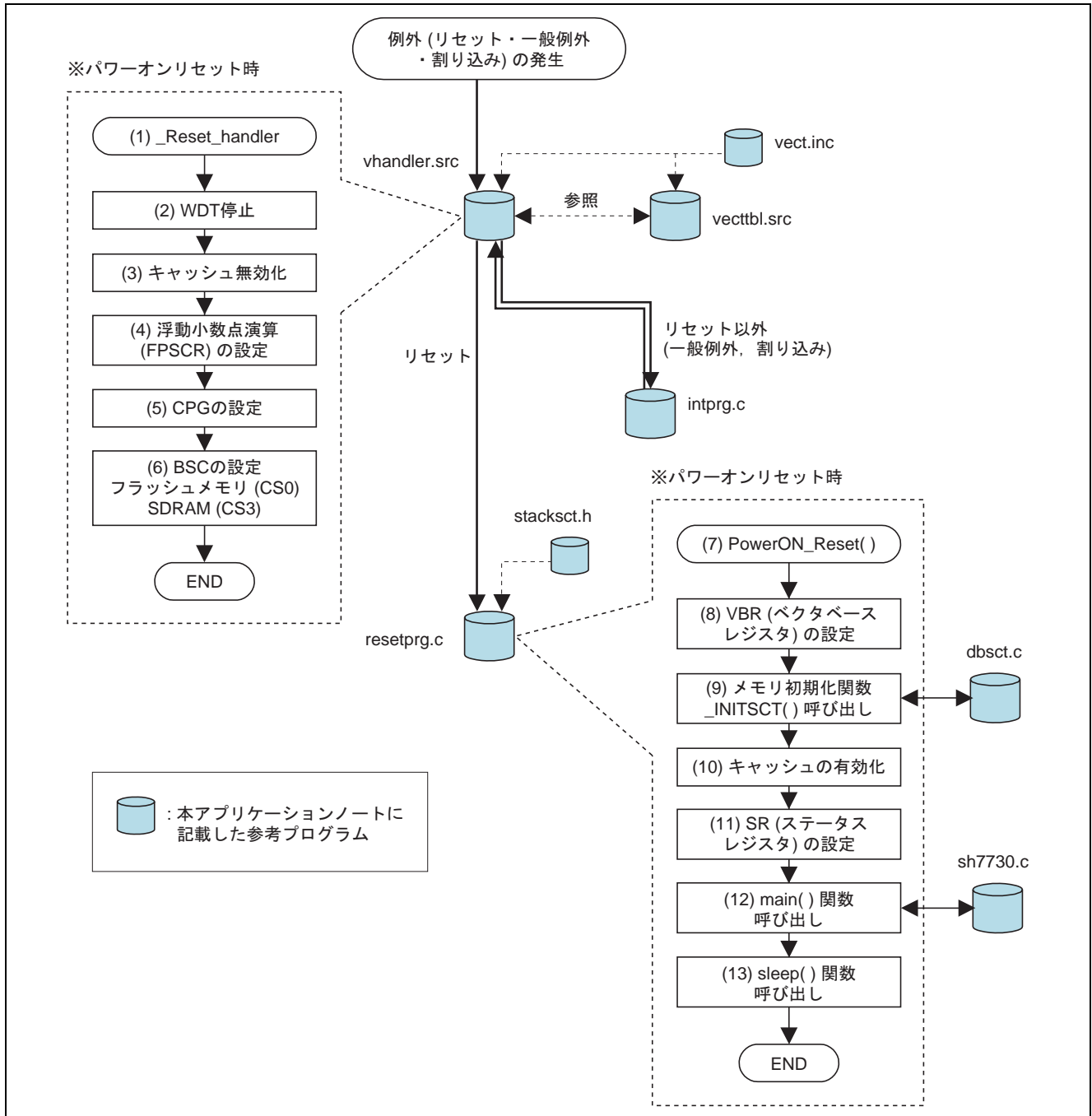


図 2 パワーオンリセットからの処理フロー

3.3 参考プログラムにおける設定内容

表 2 に参考プログラムでの各モジュールの設定を示します。

表 2 参考プログラムでの各モジュールの設定

モジュール	設定内容
CPG	内部クロック: 266.66 MHz SuperHyway バスクロック: 133.33 MHz バスクロック: 66.66 MHz 周辺クロック: 33.33 MHz
BSC	CS0: NOR 型フラッシュメモリ データバス幅: 16 ビット固定* ¹ アドレス, CSn アサート→RD, WEn アサート遅延サイクル数: 1.5 サイクル アクセスウェイトサイクル数: 4 サイクル RD, WEn ネゲート→アドレス, CSn ネゲート遅延サイクル数: 1.5 サイクル CS3: SDRAM データバス幅: 32 ビット ロウアドレスビット: 12 ビット カラムアドレスビット: 9 ビット CAS レイテンシ: 2 サイクル
PFC	マルチプレクス端子はすべて初期設定で利用可能
キャッシュ	命令/オペランドキャッシュ有効

【注】1. エリア 0 のデータバス幅は MD3 端子により決定します。

3.4 参考プログラムの注意点

(1) B, R, S セクションの外部メモリへの割り当てと_INITSCT() 関数によるセクションの初期化

本プログラムでは、外部 SDRAM 上に B, R, S セクションの割り当ておよび初期化を行うことができるように、セクションの初期化前に BSC (バスステートコントローラ) を初期化しています。

セクションの初期化は_INITSCT() 関数を使用し、D セクションから R セクションへのコピー、およびシンボルを R セクションのアドレスでリロケーションしています。そのため、セクション初期化前 (_INITSCT() 関数実行前) に実行する関数において、グローバル変数など_INITSCT() 関数で初期化を行うセクションに配置される変数を使用しないでください。

(2) セクションの初期化の高速化

セクションのコピーを行う_INITSCT() 関数の前でキャッシュを有効にすると、_INITSCT() 関数の実行を高速化できる可能性があります。その場合、セクションのコピー実行後、キャッシュの内容を外部メモリに反映するために、B セクションと R セクションについては、オペランドキャッシュのデータを書き戻す必要があります。

書き戻し処理の詳細については「SH7730 グループ アプリケーションノート キャッシュ オペランド キャッシュの書き戻し例 (RJJ06B0923)」を参照ください。

(3) RAM 上でプログラムを動作させる場合

_INITSCT() 関数と ROM 化支援機能を用いてユーザプログラムを ROM から RAM へ展開し、高速に実行する場合は次のようにしてください。

- 転送元のユーザプログラムを PROM セクションなど明示的な名前でもとめる
- 転送先を PRAM セクションなど明示的な名前とする
- dbsect.c の C\$DSEC に配置される構造体に、PROM セクションと PRAM セクションのアドレスを追加する
- _INITSCT() 関数は P セクションに配置されるので、P セクションは ROM 上に配置すること
- High-performance Embedded Workshop のメニューバーのビルド (B) → SuperH RISC engine Standard Toolchain → 最適化リンカ カテゴリ (Y) 出力 → オプション項目 (S) 「ROM から RAM へマップするセクション」で、ROM セクションに PROM, RAM セクションに PRAM を追加する

(4) スタックポインタのアドレス

#pragma entry でエントリ関数に指定されている PowerON_Reset() 関数の先頭で設定されるスタックポインタのアドレスは、High-performance Embedded Workshop でプロジェクトを生成するときに指定したアドレスが使用されます。

アドレスを変更する場合には、High-performance Embedded Workshop のメニューバーのプロジェクト (P) → 構成の編集 (E) → スタックを選択すると、スタック領域のアドレスとサイズを変更することができます (S セクションのアドレス配置を直接変更することは避けてください。直接変更すると、その後に構成の編集からダイアログが起動しなくなる場合があります)。

4. 参考プログラムリスト

(1) サンプルプログラムリスト"vhandler.src"

```

1  ;-----
2  ;
3  ; FILE      :vhandler.src
4  ; DATE      :Tue, Oct 07, 2008
5  ; DESCRIPTION :Reset/Interrupt Handler
6  ; CPU TYPE  :SH7730
7  ;
8  ; This file is generated by Renesas Project Generator (Ver.4.9).
9  ;
10 ;-----
11 ;/*****
12 ;*
13 ;* Device    : SH-4A/SH7730
14 ;*
15 ;* File Name : vhandler.src
16 ;*
17 ;* Abstract  : Reset/Interrupt Handler.
18 ;*
19 ;* History   : 1.00 (2008-10-01) [Hardware Manual Revision : 1.00]
20 ;*
21 ;* Copyright(c) 2008 Renesas Technology Corp.
22 ;*           And Renesas Solutions Corp.,All Rights Reserved.
23 ;*
24 ;*****/
25
26     .include    "env.inc"
27     .include    "vect.inc"
28
29 ;
30 ILLEGALFPU_CODE:      .equ    H'800
31 DUMMY_ILLEGALFPU_CODE: .equ    H'880
32 ILLEGALSLOTFPU_CODE: .equ    H'820
33 DUMMY_ILLEGALSLOTFPU_CODE: .equ H'8A0
34 INT_NMI_CODE:        .equ    H'1C0
35 ;
36
37 IMASKclr: .equ    H'FFFFFF0F
38 RBBLclr:  .equ    H'CFFFFFFF
39 MDRBBLset: .equ    H'70000000
40 MDRBset:  .equ    H'60000000
41 RBclr:    .equ    H'DFFFFFFF
42
43     .import     _RESET_Vectors
44     .import     _INT_Vectors
45     .import     _INT_MASK
46
47 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
48 ; macro definition
49 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
50     .macro     PUSH_EXP_BASE_REG
51         stc.l   ssr,@-r15      ; save ssr
52         stc.l   spc,@-r15      ; save spc
53         sts.l   pr,@-r15       ; save context registers
54         sts.l   fpscr,@-r15    ; save fpscr registers
55         stc.l   r7_bank,@-r15
56         stc.l   r6_bank,@-r15
57         stc.l   r5_bank,@-r15
58         stc.l   r4_bank,@-r15
59         stc.l   r3_bank,@-r15
60         stc.l   r2_bank,@-r15
61         stc.l   r1_bank,@-r15
62         stc.l   r0_bank,@-r15
    
```

```

63         .endm
64     ;
65         .macro POP_EXP_BASE_REG
66         ldc.l @r15+,r0_bank          ; recover registers
67         ldc.l @r15+,r1_bank
68         ldc.l @r15+,r2_bank
69         ldc.l @r15+,r3_bank
70         ldc.l @r15+,r4_bank
71         ldc.l @r15+,r5_bank
72         ldc.l @r15+,r6_bank
73         ldc.l @r15+,r7_bank
74         lds.l @r15+,fpscr
75         lds.l @r15+,pr
76         ldc.l @r15+,spc
77         ldc.l @r15+,ssr
78         .endm
79     ;
80     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
81     ; reset ;
82     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
83     .section RSTHandler,code
84     _ResetHandler:
85         mov.l #H'A4520004,r0 ;set RWTCsr address
86         mov.l #H'0000A507,r1 ;RWDT disable
87         mov.w r1,@r0
88
89         mov.l #H'FF00001C,r0 ;set CCR address
90         mov.l #H'00000808,r1 ;IC,OC Invalidate
91         mov.l r1,@r0
92
93         mov.l #H'00040001,r0 ;set single precision mode
94         ;mov.l #H'000C0001,r0 ;set double precision mode
95         lds.l r0,fpscr
96
97         mov.l #H'A4150000,r0 ;set CPG address
98         mov.l #H'07002508,r1 ; * Clockin = 33.333MHz, CKIO = 66.6MHz
99         ; * I Clock = 266MHz, B Clock = 66.6MHz,
100        ; * P Clock = 33.3MHz
101        mov.l r1,@r0
102
103        mov.l #CS0_INIT,r0
104        jmp @r0
105        nop
106
107    CS0_INIT_END:
108        mov.l #SDRAM_INIT,r0
109        jmp @r0
110        nop
111
112    SDRAM_INIT_END:
113        mov.l #EXPEVT,r0
114        mov.l @r0,r0
115        shlr2 r0
116        shlr r0
117        mov.l #_RESET_Vectors,r1
118        mov.l @(r0,r1),r0 ;set Reset function address
119        jmp @r0
120        nop
121    ;
122    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
123    ; exceptional interrupt ;
124    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
125    .section INTHandler,code
126    .export _INTHandlerPRG
127    _INTHandlerPRG:
128    _ExpHandler:
    
```

```

129         PUSH_EXP_BASE_REG
130     ;
131         mov.l #EXPEVT,r0           ; set event address
132         mov.l @r0,r1              ; set exception code
133     ;
134         mov.l #ILLEGALFPU_CODE,r2          ; H'800
135         cmp/eq r1,r2
136         bf     exp_01
137         mov.l #DUMMY_ILLEGALFPU_CODE,r1    ; H'800 -> H'880
138         bra   exp_10
139         nop
140     exp_01:
141         mov.l #ILLEGALSLOTFPU_CODE,r2     ; H'820
142         cmp/eq r1,r2
143         bf     exp_10
144         mov.l #DUMMY_ILLEGALSLOTFPU_CODE,r1 ; H'820 -> H'8A0
145     exp_10:
146     ;
147         mov.l #_INT_Vectors,r0           ; set vector table address
148         add  #- (h'40),r1                ; exception code - h'40
149         shlr2 r1
150         shlr r1
151         mov.l @(r0,r1),r3                ; set interrupt function addr
152     ;
153         mov.l #_INT_MASK,r0              ; interrupt mask table addr
154         shlr2 r1
155         mov.b @(r0,r1),r1                ; interrupt mask
156         extu.b r1,r1
157     ;
158         stc  sr,r0                       ; save sr
159         mov.l #(RBBLclr&IMASKclr),r2     ; RB,BL,mask clear data
160         and  r2,r0                       ; clear mask data
161         or   r1,r0                       ; set interrupt mask
162         ldc  r0,ssr                      ; set current status
163     ;
164         ldc.l r3,spc
165         mov.l #__int_term,r0             ; set interrupt terminate
166         lds  r0,pr
167     ;
168         rte
169         nop
170     ;
171         .pool
172     ;
173     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
174     ; Interrupt terminate ;
175     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
176         .align 4
177     __int_term:
178         mov.l #MDRBBLset,r0              ; set MD,BL,RB
179         ldc.l r0,sr                      ;
180         POP_EXP_BASE_REG
181         rte                               ; return
182         nop
183     ;
184         .pool
185     ;
186     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
187     ; TLB miss interrupt ;
188     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
189         .org H'300
190     _TLBmissHandler:
191
192         mov.l #(SP_STACK+H'200),r15 ;set SP_STACK(for only TLBmiss) pointer
193         stc.l sgr,@-r15
194
    
```

```

195         PUSH_EXP_BASE_REG
196     ;
197         mov.l  #EXPEVT,r0          ; set event address
198         mov.l  @r0,r1             ; set exception code
199         mov.l  #_INT_Vectors,r0   ; set vector table address
200         add    #-(h'40),r1        ; exception code - h'40
201         shlr2  r1
202         shlr   r1
203         mov.l  @(r0,r1),r3        ; set interrupt function addr
204     ;
205         mov.l  #_INT_MASK,r0      ; interrupt mask table addr
206         shlr2  r1
207         mov.b  @(r0,r1),r1        ; interrupt mask
208         extu.b r1,r1
209     ;
210         stc    sr,r0              ; save sr
211         mov.l  #(RBBLclr&IMASKclr),r2 ; RB,BL,mask clear data
212         and    r2,r0              ; clear mask data
213         or     r1,r0              ; set interrupt mask
214         ldc    r0,ssr             ; set current status
215     ;
216         ldc.l  r3,spc
217         mov.l  #__TLBMISS_INT_TERM,R0 ;set interrupt terminate
218         lds    r0,pr
219     ;
220         rte
221         nop
222
223         .align 4
224
225     __TLBMISS_INT_TERM:
226         mov.l  #MDRBBLset,r0      ;set MD,BL,RB
227         ldc.l  r0,sr
228
229         POP_EXP_BASE_REG
230
231         ldc.l  @r15+,sgr
232         stc.l  sgr,r15
233         rte
234         nop
235     ;
236         .pool
237     ;
238     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
239     ;   IRQ                                     ;
240     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
241         .org   H'500
242     _IRQHandler:
243         PUSH_EXP_BASE_REG
244     ;
245         mov.l  #INTEVT,r0          ; set event address
246         mov.l  @r0,r1             ; set exception code
247
248         mov.l  #INT_NMI_CODE,r2    ; H'1C0
249         cmp/eq r1,r2
250         bf     no_nmi
251
252     ;add NMI CASE =====
253
254         mov.l  #_INT_NMI,r3
255         jsr   @r3
256         nop
257
258         POP_EXP_BASE_REG
259
260         rte
    
```

```

261         nop
262     no_nmi:
263     ;add end NMI CASE =====
264
265         mov.l   #_INT_Vectors,r0    ; set vector table address
266         add     #-(h'40),r1         ; exception code - h'40
267         shlr2   r1
268         shlr    r1
269         mov.l   @(r0,r1),r3         ; set interrupt function addr
270     ;
271         mov.l   #_INT_MASK,r0       ; interrupt mask table addr
272         shlr2   r1
273         mov.b   @(r0,r1),r1         ; interrupt mask
274         extu.b  r1,r1
275     ;
276         stc     sr,r0               ; save sr
277         mov.l   #(RBBLclr&IMASKclr),r2 ; RB,BL,mask clear data
278         and     r2,r0               ; clear mask data
279         or      r1,r0               ; set interrupt mask
280         ldc     r0,ssr              ; set current status
281     ;
282         ldc.l   r3,spc
283         mov.l   #__int_term,r0      ; set interrupt terminate
284         lds     r0,pr
285     ;
286         rte
287         nop
288     ;
289         .pool
290
291     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
292     ; CS0 INIT
293     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
294     CS0_INIT:
295         mov.l   #H'FEC10004,r0      ;set CS0BCR address
296         mov.l   #H'10480400,r1      ;set for FLASHROM(spanion S29AL032D70TFI04)
297         mov.l   r1,@r0
298
299         mov.l   #H'FEC10024,r0      ;set CS0WCR address
300         mov.l   #H'00000A41,r1
301         mov.l   r1,@r0
302
303         mov.l   #CS0_INIT_END,r0
304         jmp     @r0
305         nop
306
307         .pool
308
309     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
310     ; SDRAM INIT
311     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
312     SDRAM_INIT:
313         mov.l   #H'FEC1000C,r0      ;set CS3BCR address
314         mov.l   #H'10004600,r1      ;set for SDRAM(Samsung K4S281632F-UC75)
315                                     ;32bit bus-width, IWW 1cyc
316         mov.l   r1,@r0
317
318         mov.l   #H'FEC1002C,r0 ;set CS3WCR address
319         mov.l   #H'00002492,r1 ;tRP 2cyc
320                                     ;tRCD 2cyc
321                                     ;A3CL 2cyc
322                                     ;tRWL 2cyc
323                                     ;tRC 6cyc
324         mov.l   r1,@r0
325
326         mov.l   #H'FEC10044,r0 ;set SDCR address
    
```

```

327         mov.l   #H'00000809,r1 ;auto refresh mode, row 12bit, column 9bit
328         mov.l   r1,@r0
329
330         mov.l   #H'FEC10050,r0 ;set RTCOR address
331         mov.l   #H'a55a003E,r1 ;refresh rate
332         mov.l   r1,@r0
333
334
335         mov.l   #H'000030d4,r0
336 LOOP1:
337         dt      r0
338         bf      LOOP1 ;200µs wait
339         nop
340         nop
341
342         mov.l   #H'FEC10048,r0 ;set RTCSR address
343         mov.l   #H'a55a0010,r1
344         mov.l   r1,@r0
345
346         mov.l   #H'FEC15080,r0 ;set SDMR3(32bit bus-width, CL=2, burstR/W(burst length=1))
347         mov.l   #H'00000000,r1
348         mov.w   r1,@r0
349
350         mov.l   #SDRAM_INIT_END,r0
351         jmp     @r0
352         nop
353
354         .pool
355
356         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
357         ; SPECIAL STACK(for TLBmiss Handler)
358         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
359         .section SP_S,data
360 SP_STACK:
361         .res.b  H'200
362
363         .end
    
```

(2) サンプルプログラムリスト"vecttbl.src"

```

1  ;-----
2  ;
3  ; FILE      :vecttbl.src
4  ; DATE      :Fri, Aug 01, 2008
5  ; DESCRIPTION :Initialize of Vector Table
6  ; CPU TYPE   :SH7730
7  ;
8  ; This file is generated by Renesas Project Generator (Ver.4.9).
9  ;
10 ;-----
11 ;/*****
12 ;*
13 ;* Device    : SH-4A/SH7730
14 ;*
15 ;* File Name : vecttbl.src
16 ;*
17 ;* Abstract  : Initialize of Vector Table.
18 ;*
19 ;* History   : 1.00 (2008-10-01) [Hardware Manual Revision : 1.00]
20 ;*
21 ;* Copyright(c) 2008 Renesas Technology Corp.
22 ;*           And Renesas Solutions Corp.,All Rights Reserved.
23 ;*
24 ;*****/
25
26         .include    "vect.inc"
27
28         .section    VECTTBL,data
29         .export     _RESET_Vectors
30
31 _RESET_Vectors:
32 ;<<VECTOR DATA START (POWER ON RESET)>>
33         ;H'000      Power On Reset(H-UDI RESET)
34         .data.l     _PowerON_Reset
35 ;<<VECTOR DATA END (POWER ON RESET)>>
36 ;<<VECTOR DATA START (MANUAL RESET)>>
37         ;H'020      Manual Reset
38         .data.l     _Manual_Reset
39 ;<<VECTOR DATA END (MANUAL RESET)>>
40         ;H'040-120  Reserved
41         .datab.l    8,H'00000000
42         ;H'140      TLB Reset(DATA TLB Reset)
43         .data.l     _TLB_Reset
44
45         .section    INTTBL,data
46         .export     _INT_Vectors
47
48 _INT_Vectors:
49         ; H'040      Data TLB miss exception(read)
50         .data.l     _INT_TLB_MISS_READ_EXP
51         ; H'060      Data TLB miss exception(write)
52         .data.l     _INT_TLB_MISS_WRITE_EXP
53         ; H'080      Initial page write exception
54         .data.l     _INT_TLB_INIT_PAGE_EXP
55         ; H'0A0      Data TLB protection violation exception (read)
56         .data.l     _INT_TLB_PROTECT_READ_EXP
57         ; H'0C0      Data TLB protection violation exception (write)
58         .data.l     _INT_TLB_PROTECT_WRITE_EXP
59         ; H'0E0      Data address error(read)
60         .data.l     _INT_ADR_ERROR_READ
61         ; H'100      Data address error(write)
    
```


62	.data.l	_INT_ADR_ERROR_WRITE
63	; H'120	FPU exception
64	.data.l	_INT_FPU_EXP
65	; H'140	Instruction TLB multiple-hit exception
66	.data.l	_TLB_Reset
67	; H'160	Unconditional trap(TRAPA)
68	.data.l	_INT_TRAP
69	; H'180	General illegal instruction exception
70	.data.l	_INT_ILLEGAL_INST_EXP
71	; H'1A0	Slot illegal instruction exception
72	.data.l	_INT_ILLEGAL_SLOT_EXP
73	;EXTERNAL INTERRUPT	
74	; H'1C0	NMI
75	.data.l	_INT_NMI
76	; H'1E0	USER_BREAK
77	.data.l	_INT_USER_BREAK
78	; H'200	IRL_LEVEL15
79	.data.l	_INT_IRL_LEVEL15
80	; H'220	IRL_LEVEL14
81	.data.l	_INT_IRL_LEVEL14
82	; H'240	IRL_LEVEL13
83	.data.l	_INT_IRL_LEVEL13
84	; H'260	IRL_LEVEL12
85	.data.l	_INT_IRL_LEVEL12
86	; H'280	IRL_LEVEL11
87	.data.l	_INT_IRL_LEVEL11
88	; H'2A0	IRL_LEVEL10
89	.data.l	_INT_IRL_LEVEL10
90	; H'2C0	IRL_LEVEL9
91	.data.l	_INT_IRL_LEVEL9
92	; H'2E0	IRL_LEVEL8
93	.data.l	_INT_IRL_LEVEL8
94	; H'300	IRL_LEVEL7
95	.data.l	_INT_IRL_LEVEL7
96	; H'320	IRL_LEVEL6
97	.data.l	_INT_IRL_LEVEL6
98	; H'340	IRL_LEVEL5
99	.data.l	_INT_IRL_LEVEL5
100	; H'360	IRL_LEVEL4
101	.data.l	_INT_IRL_LEVEL4
102	; H'380	IRL_LEVEL3
103	.data.l	_INT_IRL_LEVEL3
104	; H'3A0	IRL_LEVEL2
105	.data.l	_INT_IRL_LEVEL2
106	; H'3C0	IRL_LEVEL1
107	.data.l	_INT_IRL_LEVEL1
108	;H'3E0	Reserved
109	.data.l	H'00000000
110	;TMU-ch0	
111	;H'400	TMU_TUNI0
112	.data.l	_INT_TMU0_TUNI0
113	;TMU-ch1	
114	;H'420	TMU_TUNI1
115	.data.l	_INT_TMU1_TUNI1
116	;TMU-ch2	
117	;H'440	TMU_TUNI2
118	.data.l	_INT_TMU2_TUNI2
119	;H'460	Reserved
120	.data.l	H'00000000
121	;RTC	
122	;H'480	RTC ATI
123	.data.l	_INT_RTC_ATI
124	;H'4A0	RTC PRI

```

125         .data.l    _INT_RTC_PRI
126         ;H'4C0     RTC CUI
127         .data.l    _INT_RTC_CUI
128 ;PINT
129         ;H'4E0 PINT PINTA
130         .data.l    _INT_PINT_PINTA
131         ;H'500 PINT PINTB
132         .data.l    _INT_PINT_PINTB
133         ;H'520-5C0 Reserved
134         .datab.l   6,H'00000000
135 ;H-UDI
136         ;H'5E0     H-UDII
137         .data.l    _INT_H_UDII
138 ;IRQ
139         ;H'600     IRQ IRQ0
140         .data.l    _INT_IRQ_IRQ0
141         ;H'620     IRQ IRQ1
142         .data.l    _INT_IRQ_IRQ1
143         ;H'640     IRQ IRQ2
144         .data.l    _INT_IRQ_IRQ2
145         ;H'660     IRQ IRQ3
146         .data.l    _INT_IRQ_IRQ3
147         ;H'680     IRQ IRQ4
148         .data.l    _INT_IRQ_IRQ4
149         ;H'6A0     IRQ IRQ5
150         .data.l    _INT_IRQ_IRQ5
151         ;H'6C0     IRQ IRQ6
152         .data.l    _INT_IRQ_IRQ6
153         ;H'6E0     IRQ IRQ7
154         .data.l    _INT_IRQ_IRQ7
155 ;SIM
156         ;H'700     SIM ERI
157         .data.l    _INT_SIM_ERI
158         ;H'720     SIM RXI
159         .data.l    _INT_SIM_RXI
160         ;H'740     SIM TXI
161         .data.l    _INT_SIM_TXI
162         ;H'760     SIM TEI
163         .data.l    _INT_SIM_TEI
164         ;H'780-7C0 Reserved
165         .datab.l   3,H'00000000
166 ;IIC1
167         ;H'7E0     IIC1 IIC11
168         .data.l    _INT_IIC1_IIC11
169 ;DMAC(1)
170         ;H'800     DMAC DEI0 ; Illegal FPU -> Dummy Code H'880
171         .data.l    _INT_DMAC_DEI0
172         ;H'820     DMAC DEI1 ; Illegal slot FPU -> Dummy Code H'8A0
173         .data.l    _INT_DMAC_DEI1
174         ;H'840     DMAC DEI2
175         .data.l    _INT_DMAC_DEI2
176         ;H'860     DMAC DEI3
177         .data.l    _INT_DMAC_DEI3
178 ;Illegal FPU
179         ;H'880     Reserved -> Used as Illegal FPU
180         .data.l    _INT_ILLEGAL_FPU
181         ;H'8A0     Reserved -> Used as Illegal slot FPU
182         .data.l    _INT_ILLEGAL_SLOT_FPU
183         ;H'8C0-920 Reserved
184         .datab.l   4,H'00000000
185 ;IRDA
186         ;H'940     IRDA IRDAI0
187         .data.l    _INT_IRDA_IRDAI0
    
```

```

188             ;H'960 IRDA IRDAI1
189             .data.l    _INT_IRDA_IRDAI1
190 ;ADC
191             ;H'980 ADC ADI
192             .data.l    _INT_ADC_ADI
193 ;TPU
194             ;H'9A0 TPU TPUI0
195             .data.l    _INT_TPU_TPUI0
196             ;H'9C0 TPU TPUI1
197             .data.l    _INT_TPU_TPUI1
198             ;H'9E0-B60 Reserved
199             .datab.l   13,H'00000000
200 ;DMAC(2)
201             ;H'B80     DMAC DEI4
202             .data.l    _INT_DMACEI4
203             ;H'BA0     DMAC DEI5
204             .data.l    _INT_DMACEI5
205             ;H'BC0     DMAC DADERR
206             .data.l    _INT_DMACEI5
207             ;H'BE0     Reserved
208             .data.l    H'00000000
209 ;SCIF
210             ;H'C00     SCIF SCIFI0
211             .data.l    _INT_SCIF_SCIFI0
212             ;H'C20     SCIF SCIFI1
213             .data.l    _INT_SCIF_SCIFI1
214             ;H'C40     SCIF SCIFI2
215             .data.l    _INT_SCIF_SCIFI2
216             ;H'C60     SCIF SCIFI3
217             .data.l    _INT_SCIF_SCIFI3
218             ;H'C80     SCIFA SCIFI4
219             .data.l    _INT_SCIFA_SCIFI4
220             ;H'CA0     SCIFA SCIFI5
221             .data.l    _INT_SCIFA_SCIFI5
222             ;H'CC0-E40 Reserved
223             .datab.l   13,H'00000000
224 ;IIC0
225             ;H'E60     IIC0 IICIO
226             .data.l    _INT_IIC0_IICIO
227             ;H'E80-EE0 Reserved
228             .datab.l   4,H'00000000
229 ;CMT
230             ;H'F00     CMTI
231             .data.l    _INT_CMT_CMTI
232 ;SIOF
233             ;H'F20     SIOFI
234             .data.l    _INT_SIOF_SIOFI
235             ;H'F40-FE0 Reserved
236             .datab.l   6,H'00000000
237
238             .export    _INT_MASK
239 _INT_MASK:
240             ; interrupt priority mask level(31~0)
241
242             ;H'040     Data TLB miss exception(read)
243             .data.b    H'00
244             ;H'060     Data TLB miss exception(write)
245             .data.b    H'00
246             ;H'080     Initial page write exception
247             .data.b    H'00
248             ;H'0A0     Data TLB protection violation exception (read)
249             .data.b    H'00
250             ;H'0C0     Data TLB protection violation exception (write)
    
```

251	.data.b	H'00
252	;H'0E0	Data address error(read)
253	.data.b	H'00
254	;H'100	Data address error(write)
255	.data.b	H'00
256	;H'120	FPU exception
257	.data.b	H'00
258	;H'140	Instruction TLB multiple-hit exception
259	.data.b	H'00
260	;H'160	TRAPA
261	.data.b	H'00
262	;H'180	ILLEGAL_INST
263	.data.b	H'00
264	;H'1A0	ILLEGAL_SLOT
265	.data.b	H'00
266	;EXTERNAL INTERRUPT	
267	;H'1c0	NMI
268	.data.b	H'00
269	;H'1E0	USER_BREAK
270	.data.b	H'00
271	;H'200-3c0	IRL
272	.datab.b	15,H'00
273	;H'3e0	Reserved
274	.data.b	H'00
275	;TMU	
276	;H'400	TMU TUNI0
277	.data.b	H'00
278	;H'420	TMU TUNI1
279	.data.b	H'00
280	;H'440	TMU TUNI2
281	.data.b	H'00
282	;H'460	Reserved
283	.data.b	H'00
284	;RTC	
285	;H'480	RTC ATI
286	.data.b	H'00
287	;H'4A0	RTC PRI
288	.data.b	H'00
289	;H'4C0	RTC CUI
290	.data.b	H'00
291	;PINT	
292	;H'4E0	PINT PINTA
293	.data.b	H'00
294	;H'500	PINT PINTB
295	.data.b	H'00
296	;H'520-5C0	Reserved
297	.datab.b	6,H'00
298	;HUDI	
299	;H'5E0	__INT_H_UDII
300	.data.b	H'00
301	;IRQ	
302	;H'600	IRQ IRQ0
303	.data.b	H'00
304	;H'620	IRQ IRQ1
305	.data.b	H'00
306	;H'640	IRQ IRQ2
307	.data.b	H'00
308	;H'660	IRQ IRQ3
309	.data.b	H'00
310	;H'680	IRQ IRQ4
311	.data.b	H'00
312	;H'6A0	IRQ IRQ5
313	.data.b	H'00

```

314 ;H'6C0      IRQ IRQ6
315 .data.b    H'00
316 ;H'6E0      IRQ IRQ7
317 .data.b    H'00
318 ;SIM
319 ;H'700      SIM ERI
320 .data.b    H'00
321 ;H'720      SIM RXI
322 .data.b    H'00
323 ;H'740      SIM TXI
324 .data.b    H'00
325 ;H'760      SIM TEI
326 .data.b    H'00
327 ;H'780-7C0 Reserved
328 .datab.b   3,H'00
329 ;IIC1
330 ;H'7E0      IIC1 IIC11
331 .data.b    H'00
332 ;DMAC(1)
333 ;H'800      DMAC DEI0
334 .data.b    H'00
335 ;H'820      DMAC DEI1
336 .data.b    H'00
337 ;H'840      DMAC DEI2
338 .data.b    H'00
339 ;H'860      DMAC DEI3
340 .data.b    H'00
341 ;H'880      Reserved -> Used as Illegal FPU
342 .data.b    H'00
343 ;H'8A0      Reserved -> Used as Illegal slot FPU
344 .data.b    H'00
345 ;H'8C0-920 Reserved
346 .datab.b   4,H'00
347 ;IRDA
348 ;H'940      IRDA IRDAI0
349 .data.b    H'00
350 ;H'960      IRDA IRDAI1
351 .data.b    H'00
352 ;ADC
353 ;H'980      ADC ADI
354 .data.b    H'00
355 ;TPU
356 ;H'9A0      TPU TPUI0
357 .data.b    H'00
358 ;H'9C0      TPU TPUI1
359 .data.b    H'00
360 ;H'9E0-B60 Reserved
361 .datab.b   13,H'00
362 ;DMAC
363 ;H'B80      DMAC DEI4
364 .data.b    H'00
365 ;H'BA0      DMAC DEI5
366 .data.b    H'00
367 ;H'BC0      DMAC DADERR
368 .data.b    H'00
369 ;H'BE0      Reserved
370 .data.b    H'00
371 ;SCIF
372 ;H'C00      SCIF SCIFI0
373 .data.b    H'00
374 ;H'C20      SCIF SCIFI1
375 .data.b    H'00
376 ;H'C40      SCIF SCIFI2
    
```

```

377         .data.b      H'00
378         ;H'C60      SCIF SCIFI3
379         .data.b      H'00
380         ;H'C80      SCIFA SCIFI4
381         .data.b      H'00
382         ;H'CA0      SCIFA SCIFI5
383         .data.b      H'00
384         ;H'CC0-E40  Reserved
385         .datab.b     13,H'00
386 ;IIC0
387         ;H'E60 IIC0 IICIO
388         .data.b      H'00
389         ;H'E80-EE0  Reserved
390         .datab.b     4,H'00
391 ;CMT
392         ;H'F00      CMT
393         .data.b      H'00
394 ;SIOF
395         ;H'F20      SIOFI
396         .data.b      H'00
397         ;H'F40-FE0  Reserved
398         .datab.b     6,H'00
399
400         .end
    
```

(3) サンプルプログラムリスト"resetprg.c"

```

1  /*****
2  /*
3  /* FILE      :resetprg.c
4  /* DATE      :Wed, Dec 24, 2008
5  /* DESCRIPTION :Reset Program
6  /* CPU TYPE   :SH7730
7  /*
8  /* This file is generated by Renesas Project Generator (Ver.4.9).
9  /*
10 /*****
11 /"FILE COMMENT"***** Technical reference data *****
12 * System Name : SH7730 Sample Program
13 * File Name   : resetprg.c
14 * Abstract    : SH7730 初期設定例 Sample Program
15 * Version     : Ver 1.00
16 * Device      : SH7730
17 * Tool-Chain  : High-performance Embedded Workshop (Version 4.04.01.001)
18 *             : C/C++ Compiler Package for SuperH Family (V.9.02release00)
19 * OS          : None
20 * H/W Platform : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
21 * Description  : SH7730 初期設定例のサンプルプログラムです。
22 *
23 * Operation   :
24 * Limitation  :
25 *
26 *****
27 * History     : 27.May.2008 Ver. 1.00 First Release
28 *"FILE COMMENT END"*****
29 #include <machine.h>
30 #include <_h_c_lib.h>
31 #include "typedefine.h"
32 #include "stacksct.h"
33 #include "iodefine.h"
34 #include "cache.h" /* キャッシュ機能 追加 */
35
36
37 #define SR_Init 0x40000000
38 #define INT_OFFSET 0x100UL
39
40 #ifdef __cplusplus
41 extern "C" {
42 #endif
43 extern void INTHandlerPRG(void);
44 void PowerON_Reset(void);
45 void Manual_Reset(void);
46 void main(void);
47 #ifdef __cplusplus
48 }
49 #endif
50
51 // #ifdef __cplusplus // Enable I/O in the application(both SIM I/O and hardware I/O)
52 // extern "C" {
53 // #endif
54 // extern void _INIT_IOLIB(void);
55 // extern void _CLOSEALL(void);
56 // #ifdef __cplusplus
57 // }
58 // #endif
59
60 // extern void srand(_UINT); // Remove the comment when you use rand()
61 // extern _SBYTE *_slptr; // Remove the comment when you use strtok()
    
```

```

62
63 #ifdef __cplusplus          // Use Hardware Setup
64 extern "C" {
65 #endif
66 extern void HardwareSetup(void);
67 #ifdef __cplusplus
68 }
69 #endif
70
71 // #ifdef __cplusplus      // Remove the comment when you use global class object
72 // extern "C" {           // Sections C$INIT and C$END will be generated
73 // #endif
74 // extern void _CALL_INIT(void);
75 // extern void _CALL_END(void);
76 // #ifdef __cplusplus
77 // }
78 // #endif
79
80
81 /* = = = = セクション名を ResetPRG に切り替え = = = = */
82 #pragma section ResetPRG
83
84 /* = = = = エントリ関数の指定 = = = = */
85 #pragma entry PowerON_Reset
86 /* "FUNC COMMENT" "*****"
87 * ID :
88 * Outline : CPU 初期化関数
89 * Include :
90 * Declaration : void PowerON_Reset(void)
91 * Description : パワーオンリセット例外ベクタテーブルに
92 * : 登録する CPU 初期化処理です。
93 * : パワーオンリセット後本関数を最初に実行します。
94 * :
95 * Limitation : コメントアウトしている処理は必要に応じて有効にしてください。
96 * Argument : none
97 * Return Value : none
98 * Calling Functions :
99 * "FUNC COMMENT END" "*****"/
100 void PowerON_Reset(void)
101 {
102
103     set_vbr((void *)((_UINT)INTHandlerPRG - INT_OFFSET));
104
105     /* = = = = B,D セクションの初期化 = = = = */
106     _INITSCT();
107
108     // errno=0; // Remove the comment when you use errno
109     // srand((_UINT)1); // Remove the comment when you use rand()
110     // _slptr=NULL; // Remove the comment when you use strtok()
111
112
113     /* ===== キャッシュの設定 ===== */
114     /* ===== こちらの関数の詳細は、SH7730 アプリケーションノート キャッシュ 設定例を参照 ===== */
115     cache_set_ccr(D_CACHE_I_ON | D_CACHE_O_ON );
116
117     /* ===== ステータスレジスタの設定(特権モード) ===== */
118     set_cr(SR_Init);
119
120     main();
121
122     // _CLOSEALL(); // Close I/O in the application(both SIM I/O and hardware I/O)
123
124     // _CALL_END(); // Remove the comment when you use global class object
    
```



```

125
126     sleep();
127 }
128
129 // #pragma entry Manual_Reset           // Remove the comment when you use Manual_Reset
130 /*"FUNC COMMENT"*****
131 * ID                                     :
132 * Outline                               : マニュアルリセット処理
133 * Include                               :
134 * Declaration                           : void Manual_Reset_PC (void)
135 * Description                           : マニュアルリセット例外ベクタテーブルに
136 *                                       : 登録する関数です。
137 *                                       :
138 * Limitation                            : 参考プログラムでは処理を定義していません。
139 *                                       : 必要に応じて処理を追加してください。
140 * Argument                              : none
141 * Return Value                          : none
142 * Calling Functions                     :
143 *"FUNC COMMENT END"*****/
144 void Manual_Reset (void)
145 {
146     /* NOP */
147 }
148 /* END of File */
    
```

(4) サンプルプログラムリスト"dbst.c"

```

1  /*****
2  /*
3  /* FILE           :dbst.c
4  /* DATE           :Wed, Dec 24, 2008
5  /* DESCRIPTION    :Setting of B,R Section
6  /* CPU TYPE       :SH7730
7  /*
8  /* This file is generated by Renesas Project Generator (Ver.4.9).
9  /*
10 /*****
11
12
13
14 #include "typedefine.h"
15
16 #pragma section $DSEC
17 static const struct {
18     _UBYTE *rom_s;           /* 初期化データセクションのROM 上の先頭アドレス */
19     _UBYTE *rom_e;           /* 初期化データセクションのROM 上の最終アドレス */
20     _UBYTE *ram_s;           /* 初期化データセクションのRAM 上の先頭アドレス */
21 } DTBL[] = {
22     { __sectop("D"), __secend("D"), __sectop("R") }
23 };
24 #pragma section $BSEC
25 static const struct {
26     _UBYTE *b_s;             /* 未初期化データセクションの先頭アドレス */
27     _UBYTE *b_e;             /* 未初期化データセクションの最終アドレス */
28 } BTBL[] = {
29     { __sectop("B"), __secend("B") }
30 };
    
```

(5) サンプルプログラムリスト"sh7730.c"

```

1  /*"FILE COMMENT"***** Technical reference data *****
2  * System Name      : SH7730 Sample Program
3  * File Name       : sh7730.c
4  * Abstract        : SH7730 初期設定例 Sample Program
5  * Version         : Ver 1.00
6  * Device          : SH7730
7  * Tool-Chain      : SuperH RISC engine Standard Toolchain Ver.9.1.1.0
8  * OS              : None
9  * H/W Platform    : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
10 * Description      : SH7730 初期設定例のサンプルプログラムです。
11 *                :
12 * Operation       :
13 * Disclaimer      :
14 *                :
15 * Copyright (C) 2008. Renesas Technology Corp., All Rights Reserved.
16 *
17 *****
18 * History          : 27.May.2008 Ver. 1.00 First Release
19 *"FILE COMMENT END"*****/
20 #include <machine.h>
21 #include "iodefine.h"
22
23 // #include "typedefine.h"
24 #ifdef __cplusplus
25 // #include <ios> // Remove the comment when you use ios
26 // _SINT ios_base::Init::init_cnt; // Remove the comment when you use ios
27 #endif
28
29 void main(void);
30
31 #ifdef __cplusplus
32 extern "C" {
33 void abort(void);
34 }
35 #endif
36
37 /*"FUNC COMMENT"*****
38 * ID                :
39 * Outline           : メイン関数
40 * Include           :
41 * Declaration       : void main(void)
42 * Description       : サンプルプログラムのメイン関数です。
43 *                  :
44 * Argument          : none
45 * Return Value      : none
46 * Calling Functions :
47 *"FUNC COMMENT END"*****/
48 void main(void)
49 {
50
51 }
52
53 #ifdef __cplusplus
54 void abort(void)
55 {
56
57 }
58 #endif
    
```

(6) サンプルプログラムリスト"intprg.c"

```

1  /*"FILE COMMENT"***** Technical reference data *****
2  * System Name      : SH7730 Sample Program
3  * File Name       : intprg.c
4  * Abstract        : SH7730 初期設定例 Sample Program
5  * Version         : Ver 1.00
6  * Device          : SH7730
7  * Tool-Chain      : SuperH RISC engine Standard Toolchain Ver.9.1.1.0
8  * OS              : None
9  * H/W Platform    : アルファプロジェクト製 SH-4A ボード 型番 AP-SH4A-1A
10 * Description      : SH7730 初期設定例のサンプルプログラムです。
11 *                  : intprg.src を C 言語形式に変更した。
12 *                  :
13 * Operation       :
14 * Disclaimer      :
15 *                  :
16 * Copyright (C) 2008. Renesas Technology Corp., All Rights Reserved.
17 *
18 *****
19 * History          : 27.May.2008 Ver. 1.00 First Release
20 *"FILE COMMENT END"*****/
21
22 #include <machine.h>
23 #include "iodefine.h"
24
25 /* --- RAM allocation variable declaration --- */
26
27 #pragma section IntPRG
28 /* H'040 Data TLB miss exception(read) */
29 void INT_TLB_MISS_READ_EXP(void)
30 {
31 }

```

...途中省略...

```

/* H'1C0 NMI */
void INT_NMI(void)
{
}

```

...途中省略...

5. 参考ドキュメント

- ソフトウェアマニュアル
SH-4A ソフトウェアマニュアル (RJJ09B0090)
(最新版をルネサス テクノロジホームページから入手してください。)
- ハードウェアマニュアル
SH7730 グループ ハードウェアマニュアル (RJJ09B0339)
(最新版をルネサス テクノロジホームページから入手してください。)
- High-performance Embedded Workshop スタートアップガイド
(最新版をルネサス テクノロジホームページから入手してください。)
- アプリケーションノート
SuperH RISC engine C/C++ コンパイラパッケージ アプリケーションノート: <導入ガイド>スタートアップ
ブルーチンガイド SH-3,SH-4,SH-4A 編 (RJJ06J0016)
(最新版をルネサス テクノロジホームページから入手してください。)
- 開発環境マニュアル
E10A-USB エミュレータ用 フラッシュメモリダウンロードプログラム アプリケーションノート
(RJJ10J0957)
(最新版をルネサス テクノロジホームページから入手してください。)

SuperH RISC engine C/C++コンパイラ, アセンブラ, 最適化リンケージエディタ コンパイラパッケージ
V.9.01 ユーザーズマニュアル (RJJ10J1885)
(最新版をルネサス テクノロジホームページから入手してください。)

ホームページとサポート窓口

ルネサス テクノロジホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2009.02.09	—	初版発行
1.01	2009.03.30	20, 21	ソースファイルを改訂
2.00	2009.12.24	3	ツールチェーンのバージョンおよびコンパイルオプションの内容を修正
		6	配置アドレスを修正

すべての商標および登録商標は、それぞれの所有者に帰属します。

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情報の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」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いただきますとともに、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ください。
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ください。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ください。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ください。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いたします。
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いたします。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いたします。
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ください。

D039444