

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日

ルネサスエレクトロニクス株式会社

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

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

ご注意書き

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

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

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

SH7764 グループ

SH7764 初期設定例

要旨

本アプリケーションノートは、SH7764 を起動する際に必要な設定項目について説明します。

動作確認デバイス

SH7764 (ルネサス テクノロジ製 R0K507764E001BR)

目次

1. はじめに.....	2
2. 応用例の説明.....	3
3. 参考プログラムリスト.....	13
4. 参考ドキュメント.....	32

1. はじめに

1.1 仕様

パワーオンリセット解除後に、メモリコントローラユニット (MCU)、汎用入出力ポート (GPIO)、およびキャッシュの初期設定を行います。

1.2 使用機能

- メモリコントローラユニット (MCU)
- キャッシュ

1.3 適用条件

- マイコン: SH7764
- 動作周波数: CPU クロック: 324MHz
SuperHyway クロック: 108MHz
周辺クロック: 54MHz
バスクロック: 108MHz
- エリア 0 バス幅: 32bit (MODE3 端子 = High, MODE4 端子 = High)
- クロック動作モード: モード 3 (MODE2 = Low, MODE1 = High, MODE0 = High)
- データアライメント: リトルエンディアン
- アドレスモード: 29 ビットアドレスモード
- C コンパイラ: ルネサス テクノロジ製 SuperH RISC engine ファミリ
C/C++ コンパイラパッケージ Ver.9.03 Release 00

表 1 に本アプリケーションノートでの各セクション配置を示します。

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 (指定無しの場合)	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	0x04000000	P1 領域 (キャッシング可能, MMU アドレス変換不可)
R	初期化データ領域	RAM		
S	スタック領域	RAM	0x07FFF9F0	
INTHandler	例外/割り込みハンドラ	ROM	0x80000400	
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		P2 領域 (キャッシング不可, MMU アドレス変換不可)
P_IntPRG	割り込み関数	ROM		
SP_S	TLBmiss ハンドラ用スタック領域	RAM	0x87FFDF0	
RSTHandler	リセットハンドラ	ROM	0xA0000000	
P_ResetPRG	初期化プログラム領域	ROM	0xA0002000	

2. 応用例の説明

本アプリケーションノートでは、SDRAM インタフェース、初期設定プログラムにおける初期設定例を説明します。

SH7764 の各アプリケーションノートでは、本アプリケーションノートで説明する参考プログラム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。

2.1 SDRAM インタフェースの説明

2.1.1 動作概要

SH7764 のメモリコントローラユニット (MCU) は、SDRAM と直結可能な SDRAM インタフェース機能を内蔵しています。

SDRAM は CS 信号によって選択できるため、 \overline{RAS} 等の制御信号を共通に使用して物理空間のエリア 1 とエリア 2 に接続が可能です。

SDRAM を直結するための制御信号は \overline{RAS} , \overline{CAS} , R/\overline{W} , $\overline{CS1}$ または $\overline{CS2}$, \overline{DQMLL} , \overline{DQMLU} , \overline{DQMUL} , \overline{DQMUU} , $\overline{WE0} \sim \overline{WE3}$ (64bit 時) および \overline{CKE} 信号です。 $\overline{CS1}$ または $\overline{CS2}$ を除く信号は各エリア共通であり、 \overline{CKE} を除く信号は $\overline{CS1}$ または $\overline{CS2}$ がアサートされた時のみ有効となり取り込まれます。

本モジュールでは SDRAM の動作モードとして、バーストリード/ライトのモードをサポートしています。データのバス幅は 32 ビットまたは 64 ビットであり、MIM の BW ビットで設定します。

2.1.2 SDRAM の仕様

本アプリケーションノートで使用する SDRAM の仕様を示します。使用する SDRAM は、エルピーダメモリ社製の EDS2516AFTA-75-E です。仕様の詳細はデバイスによって異なりますので、使用する製品のデータシートを確認してください。

表 2 本アプリケーションノートで使用する SDRAM の仕様

項目	仕様
クロック周波数	133MHz (最大)
容量	256M ビット (32M バイト) × 2 個
構成	4 バンク × 16M ワード × 16 ビット構成
CAS レイテンシ	2/3 (プログラマブル)
リフレッシュサイクル	64ms ごとの 8192 リフレッシュサイクル
バースト長	1/2/4/8/フルページ (プログラマブル)
ロウアドレス	A12 ~ A0
コラムアドレス	A8 ~ A0

2.1.3 メモリマップ

SH7764 の CS1 または CS2 空間が SDRAM に接続可能な空間です。内部アドレスの上位 3 ビットにより、キャッシュの有効/無効を制御することができます (MMU オン時)。本応用例では CS1 空間に SDRAM を割り当てます。図 1 にメモリマップを示します。

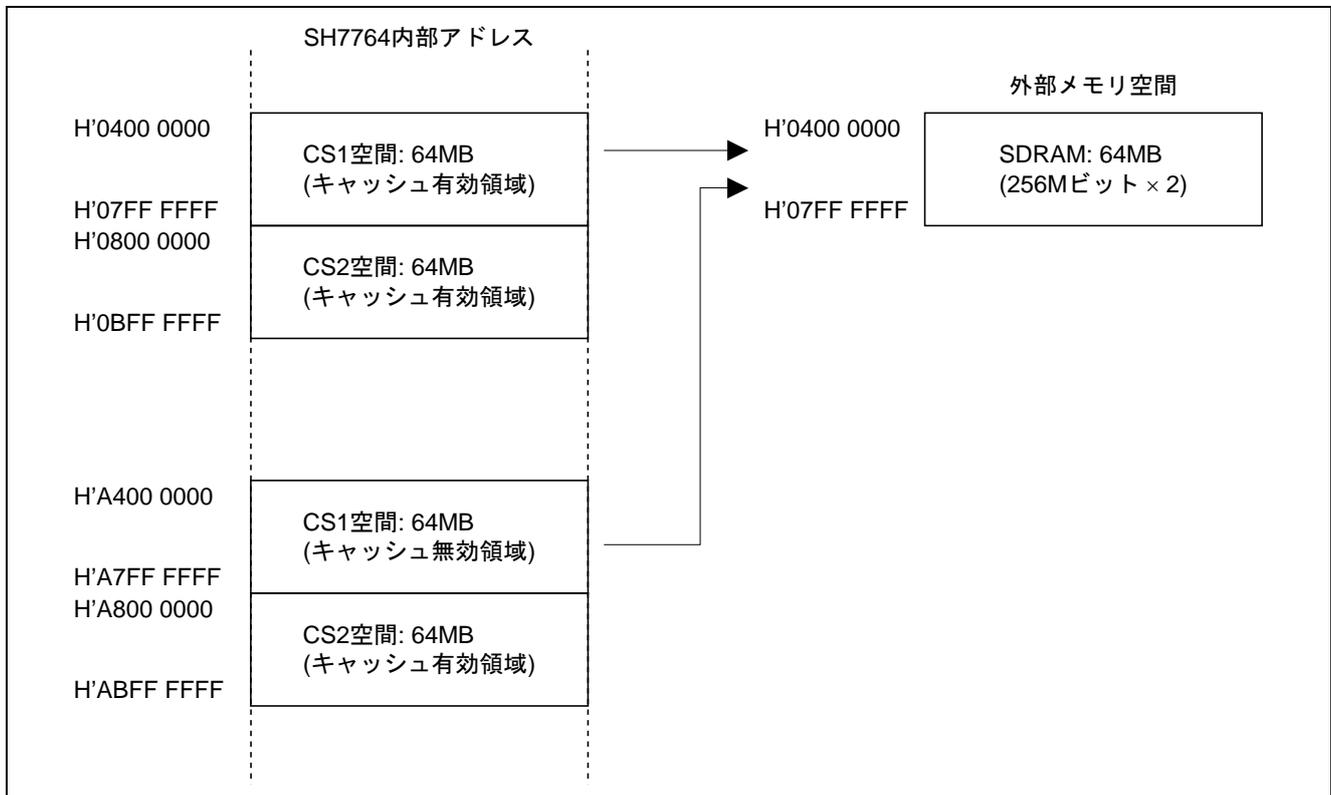


図 1 SDRAM 関連のメモリマップ

2.1.4 SDRAM 接続回路例

図 2 に本アプリケーションノートにおける SDRAM との接続回路例を示します。

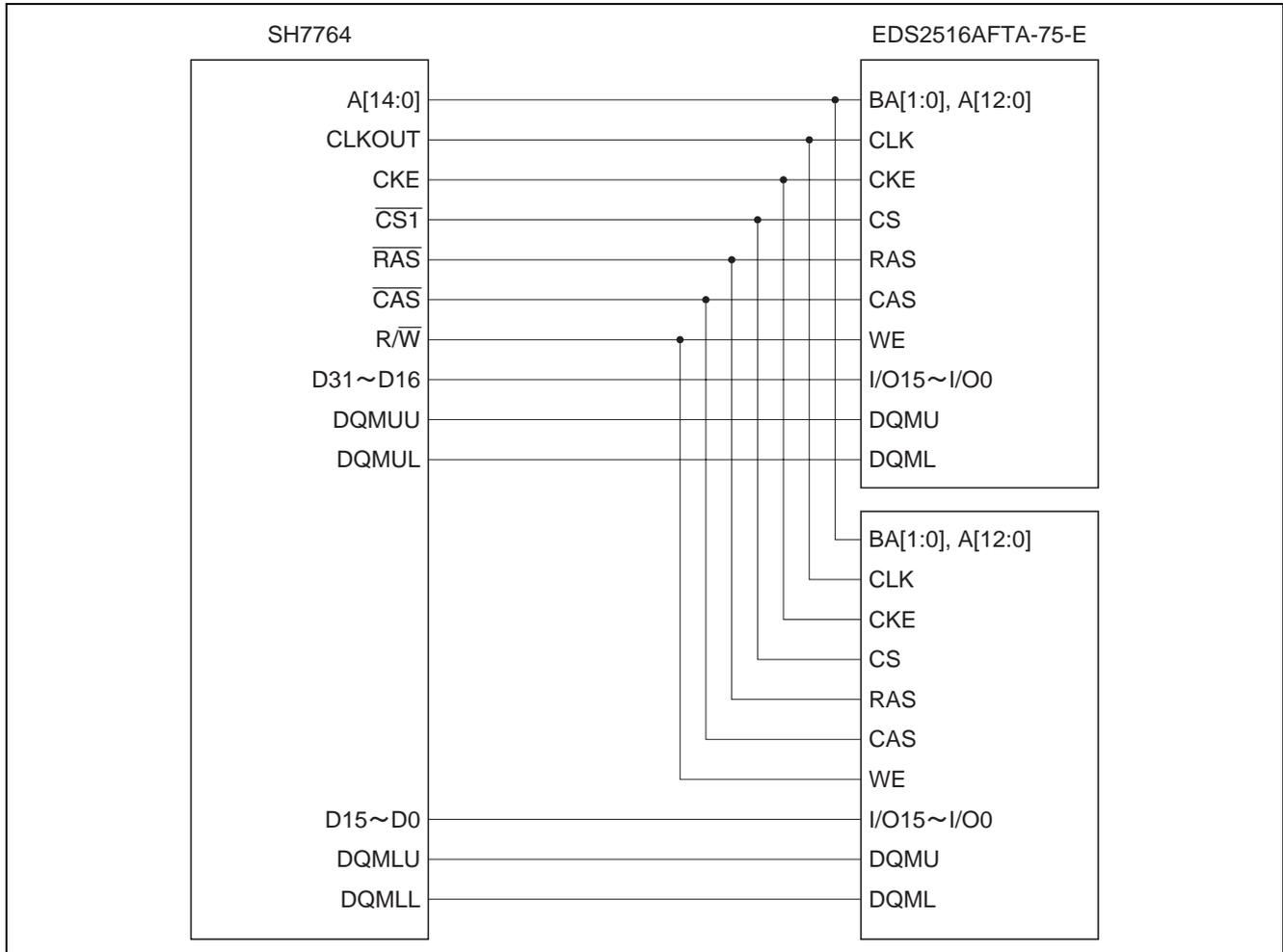


図 2 SDRAM 接続回路例 (256M ビット品 × 2 32 ビットバス)

2.1.5 アドレスマルチプレクス

表 3 に本アプリケーションノートにおけるアドレスマルチプレクス出力端子を示します。

SH7764 では、下記以外のロウ/カラム構成は禁止されておりますので、ご注意ください。

- 12 × 9 (= 8M × 16bit 品 or 8M × 32bit 品)
- 13 × 9 (= 16M × 16bit 品)

表 3 アドレスマルチプレクス

外部バス	SDRAM アドレス	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
32bit	SH7764 アドレス	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
256Mbit (32MB) 16M × 16	Bank (2)	12	13													
	Row (13)			11	25	24	23	22	21	20	19	18	17	16	15	14
	Col (9)							10	9	8	7	6	5	4	3	2

2.1.7 SDRAM 初期化手順例

図 4 に SDRAM 初期化設定手順例を示します。

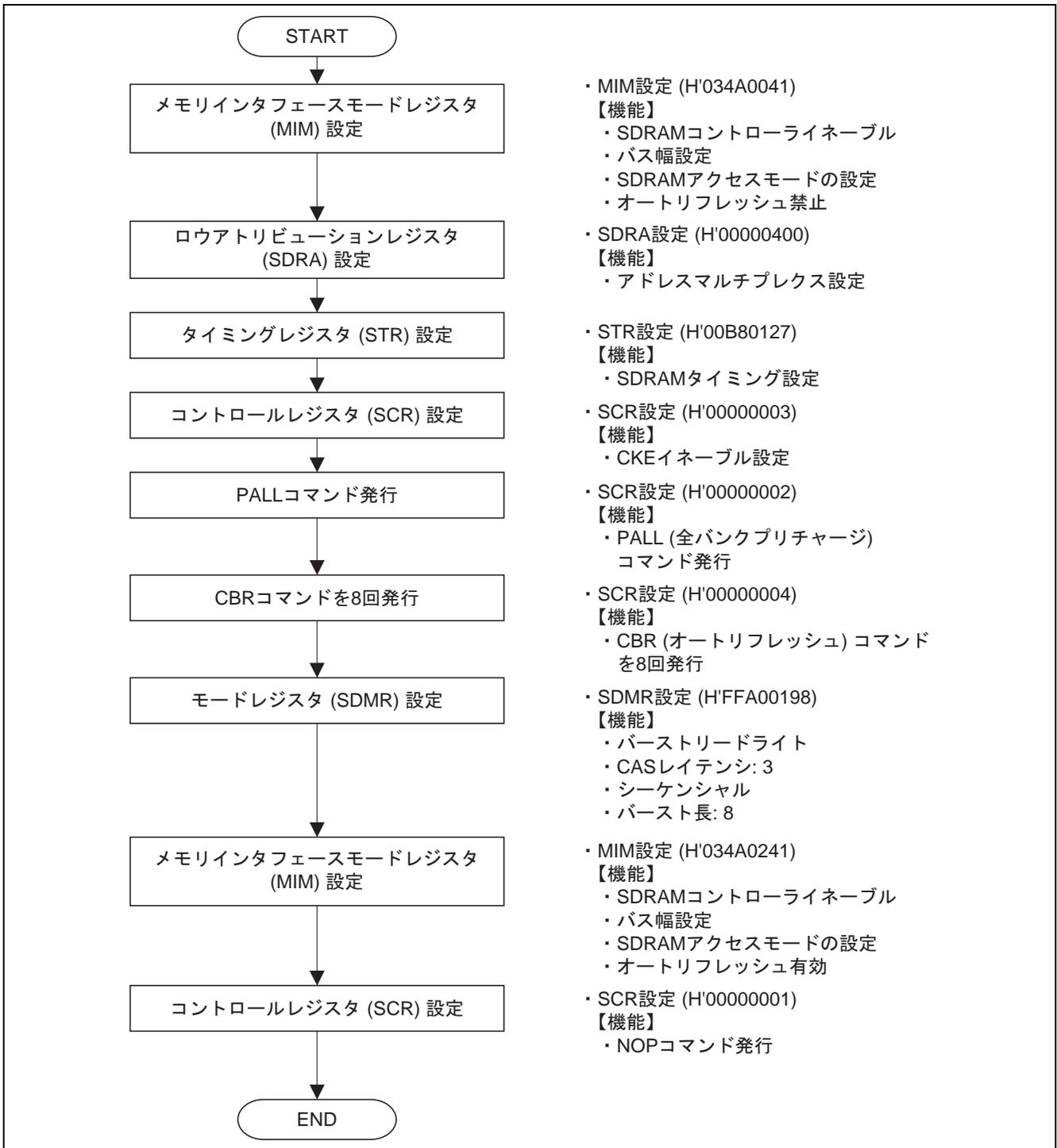


図 4 SDRAM 初期設定手順例

2.2 参考プログラムの説明

初期設定プログラムは、以下の9つのソースプログラムから構成されています。

- (1) vhandler.src
- (2) vecttbl.src
- (3) resetprg.c
- (4) hwsetup.c
- (5) stacksct.h
- (6) dbsct.c
- (7) main.c
- (8) intprg.c
- (9) 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には各例外ハンドラ処理とMCU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」を参照ください。

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。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。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。イベントハンドラ処理では，共に例外（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。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。PowerON_Reset()関数には，VBR（ベクタベースレジスタ）の設定，FPSCRの設定，_INTSCT()関数の呼び出し，HardwareSetup()関数の呼び出し，RAMCDの設定，SR（ステータスレジスタ）の設定，メイン関数の呼び出しなどを記述します。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。

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（ユーザーモードにおける内蔵メモリアクセスを許可）しています。

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，割り込みハンドラの先頭番地は，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export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。

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。

(4) hwsetup.c

周辺モジュールの動作設定を行います。本プログラムでは、キャッシュ、の設定を行います。キャッシュの設定は、命令キャッシュとオペランドキャッシュの有効化、コピーバックモードの選択を行います。また、本プログラムでは使用していませんが、R0K507764E001BR での GPIO 設定を参考資料として記載しています。

(5) 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)。

(6) dbsct.c

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。

(7) main.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。ユーザー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。

(8) intrpg.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。

(9) vect.inc

intrpg.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。intrpg.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 5 にパワーオンリセットからの処理フローを示します。

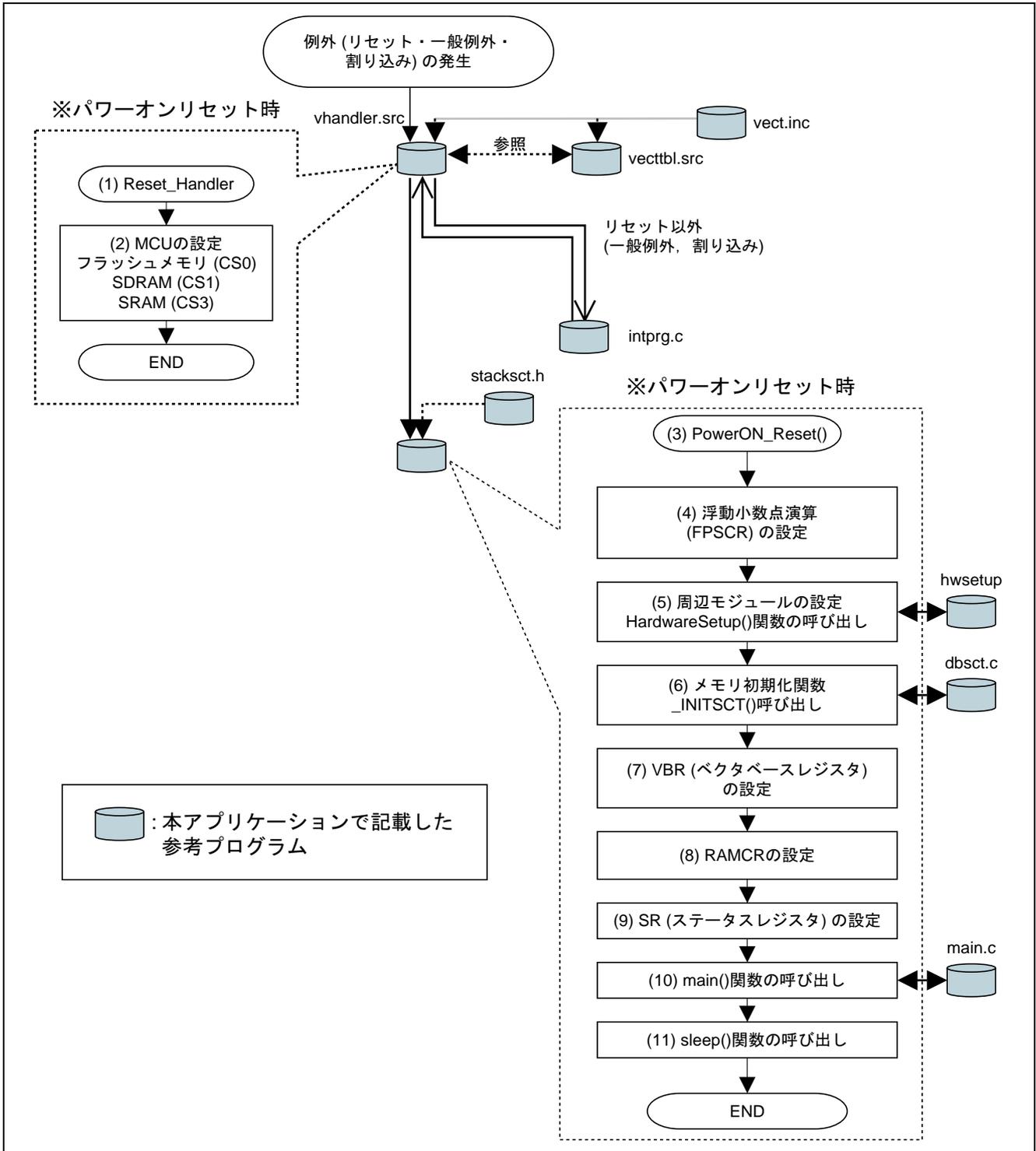


図 5 パワーオンリセットからの処理フロー

2.2.1 参考プログラムにおける設定例

表 4 に参考プログラムにおける設定例を示します。

表 4 参考プログラム設定例

モジュール	設定内容
MCU	CS0: NOR 型フラッシュメモリ データバス幅: 32bit 別空間リード-ライトサイクル間アイドル: 3 サイクル エリア 0 同一空間リード-ライトサイクル間アイドル: 3 サイクル 別空間リード-リードサイクル間アイドル: 3 サイクル エリア 0 同一空間リード-リードサイクル間アイドル: 3 サイクル アドレスホールドサイクル: 1 サイクル CSn 空間アクセス時の挿入ウェイトサイクル: 11 サイクル CS1: SDRAM アクセスモード: BANK オープンモード リフレッシュインターバル: 546 カウント リフレッシュ機能: オートリフレッシュ リトルエンディアン データバス幅: 32bit SDRAM 制御許可 ライト-リード間最低サイクル数: 9 サイクル リード-ライト間最低サイクル数: 12 サイクル PRE/PALL コマンド発行サイクル数 (Twr): 2 サイクル オートリフレッシュ-ACT/オートリフレッシュサイクル数 (Trfc): 8 サイクル ACT-PRE コマンド発行最低サイクル数 (Tras): 7 サイクル PRE-ACT コマンド発行サイクル数 (Trp): 3 サイクル ACT-オートリフレッシュ/ACT-ACT 発行サイクル数 (Trc): 9 サイクル CAS レイテンシ: 3 サイクル RAS-CAS コマンド発行サイクル数 (Trcd): 3 サイクル SDRAM メモリ構成指定ビット: Row13 ビット × Column9 ビット (16M × 16bit 品) CS3: SRAM データバス幅: 32bit ライト-リード/ライト-ライトサイクル間アイドル: 4 アイドル 別空間リード-ライトサイクル間アイドル: 2 アイドル 同一空間リード-ライトサイクル間アイドル: 4 アイドル 別空間リード-リードサイクル間アイドル: 2 アイドル 同一空間リード-リードサイクル間アイドル: 2 アイドル アドレスホールドサイクル: 2 サイクル CSn 空間アクセス時の挿入ウェイトサイクル: 8 サイクル
キャッシュ	命令/オペランドキャッシュ有効 P1 領域: コピーバックモード P0, U0, P3 領域: コピーバックモード

2.2.2 参考プログラム使用時の注意点

- 本プログラムでは、外部メモリ上に 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様に、セクションの初期化前に MCU を初期化しています。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。
- 本プログラムでは、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オペランドキャッシュのデータを書き戻し (書き戻しと無効化を行う `OCBP` 命令を使用) しています。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などでは、`_INITSCT` 関数で該当セクションをコピーした後、B セクションなどと同様にキャッシュのデータを書き戻す必要がありますのでご注意ください。
- `Entry` 関数 (`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 にてプロジェクトを生成する時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) → 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。(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(E) からダイアログが起動しなくなる場合があります。)

3. 参考プログラムリスト

3.1 サンプルプログラムリスト "vhandler.src"

```

1  ;/*****"FILE COMMENT"*****
2  ;*      System Name :   SH7764 Sample Program
3  ;*      File Name   :   vhandler.src
4  ;*      Version    :   1.00.00
5  ;*      Contents   :   SH7764 Intialize Program
6  ;*      Model      :   Renesas SH7764 Board R0K507764E001BR
7  ;*      CPU        :   SH7764
8  ;*      Compiler   :   SHC.9.03.00
9  ;*      OS         :   none
10 ;*
11 ;*      note       :   < Caution >
12 ;*                  This sample program is provided simply as a reference and
13 ;*                  its operation is not guaranteed.
14 ;*                  Use this sample program as a technical reference when
15 ;*                  developing software.
16 ;*
17 ;*                  Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
18 ;*
19 ;*      History    :   2009/03/01 ver 1.00.00
20 ;*
21 ;*****/
22         .include    "vect.inc"
23
24         .import     _INT_VECTORS
25         .import     _RESET_VECTORS
26         .import     _INT_MASK
27
28     EXPEVT     .equ    H'FF000024
29     INTEVT     .equ    H'FF000028
30
31     IMASKclr:   .equ    H'FFFFFF0F
32     RBBLclr:   .equ    H'FFFFFFF
33     MDRBBLset .equ    H'70000000
34     MDRBset   .equ    H'60000000
35     RBclr:    .equ    H'DFFFFFFF
36
37     ;;;;;;;;;;;;;;
38     ;          MACRO DEFINITON
39     ;;;;;;;;;;;;;;
40
41         .macro PUSH_EXP_BASE_REG
42             STC.L    SSR,@-R15
43             STC.L    SPC,@-R15
44             STS.L    PR,@-R15
45             STS.L    FPSCR,@-R15
46             STC.L    R7_BANK,@-R15
47             STC.L    R6_BANK,@-R15
48             STC.L    R5_BANK,@-R15
49             STC.L    R4_BANK,@-R15
50             STC.L    R3_BANK,@-R15
51             STC.L    R2_BANK,@-R15
52             STC.L    R1_BANK,@-R15
53             STC.L    R0_BANK,@-R15
54         .endm
55
56         .macro POP_EXP_BASE_REG
57             LDC.L    @R15+,R0_BANK
58             LDC.L    @R15+,R1_BANK
    
```

```

59         LDC.L      @R15+,R2_BANK
60         LDC.L      @R15+,R3_BANK
61         LDC.L      @R15+,R4_BANK
62         LDC.L      @R15+,R5_BANK
63         LDC.L      @R15+,R6_BANK
64         LDC.L      @R15+,R7_BANK
65         LDS.L      @R15+,FPSCR
66         LDS.L      @R15+,PR
67         LDC.L      @R15+,SPC
68         LDC.L      @R15+,SSR
69         .endm
70
71
72         ;;;;;;;;;;;;;;
73         ;          Reset_Handler
74         ;;;;;;;;;;;;;;
75         .section   RSTHandler,code
76         _Reset_Handler:
77             MOV.L   #BSC_INIT,R0
78             JMP     @R0
79             NOP
80         BSC_INIT_END:
81
82             MOV.L   #SDRAM_INIT,R0
83             JMP     @R0
84             NOP
85         SDRAM_INIT_END:
86
87             MOV.L   #EXPEVT,R0
88             MOV.L   @R0,R0
89             SHLR2   R0
90             SHLR    R0
91             MOV.L   #_RESET_VECTORS,R1
92             MOV.L   @(R0,R1),R0
93             JMP     @R0
94             NOP
95
96         .pool
97
98         ;;;;;;;;;;;;;;
99         ;          exceptional interrupt
100        ;;;;;;;;;;;;;;
101        .section   INTHandler,code
102        .export    _INTHandlerPRG
103
104        _INTHandlerPRG:
105            PUSH_EXP_BASE_REG
106
107            MOV.L   #EXPEVT,R0
108            MOV.L   @R0,R1
109
110            MOV.L   #_INT_VECTORS,R0
111            ADD     #-(H'40),R1
112            SHLR2   R1
113            SHLR    R1
114            MOV.L   @(R0,R1),R3
115
116            LDC.L   R3,SPC
117            MOV.L   #__INT_TERM,R0
118            LDS.L   R0,PR
119
120            RTE
121            NOP
    
```

```

122
123         .pool
124
125  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
126  ;           TLB miss interrupt
127  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
128         .org           H'300
129  _TLBmissHandler:
130
131         MOV.L          #(SP_STACK+H'200),R15
132         STC.L          SGR,@-R15
133
134         PUSH_EXP_BASE_REG
135
136         MOV.L          #EXPEVT,R0
137         MOV.L          @R0,R1
138         MOV.L          #_INT_VECTORS,R0
139         ADD             #-(H'40),R1
140         SHLR2          R1
141         SHLR           R1
142         MOV.L          @(R0,R1),R3
143
144         MOV.L          #_INT_MASK,R0
145         SHLR2          R1
146         MOV.B          @(R0,R1),R1
147         EXTU.B         R1,R1
148
149         STC            SR,R0
150         MOV.L          #(RBclr&IMASKclr),R2
151         AND            R2,R0
152         OR             R1,R0
153         LDC            R0,SSR
154
155         LDC.L          R3,SPC
156         MOV.L          #__TLBMISS_INT_TERM,R0
157         LDS.L          R0,PR
158
159         RTE
160         NOP
161
162         .align         4
163
164  __TLBMISS_INT_TERM:
165         MOV.L          #MDRBBLset,R0
166         LDC.L          R0,SR
167
168         PUSH_EXP_BASE_REG
169
170         LDC.L          @R15+,SGR
171         STC.L          SGR,R15
172         RTE
173         NOP
174
175         .pool
176
177  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
178  ;           IRQ
179  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
180         .org           H'500
181  _IRQ_Handler:
182         PUSH_EXP_BASE_REG
183
184         MOV.L          #INTEVT,R0
    
```

```

185      MOV.L      @R0,R1
186
187      MOV.L      #_INT_VECTORS,R0
188      ADD        #-(H'40),R1
189      SHLR2      R1
190      SHLR      R1
191      MOV.L      @(R0,R1),R3
192
193      MOV.L      #_INT_MASK,R0
194      SHLR2      R1
195      MOV.B      @(R0,R1),R1
196      EXTU.B     R1,R1
197
198      STC        SR,R0
199      MOV.L      #(RBBLClr&IMASKClr),R2
200      AND        R2,R0
201      OR         R1,R0
202      LDC        R0,SSR
203
204      LDC.L      R3,SPC
205      MOV.L      #__INT_TERM,R0
206      LDS.L      R0,PR
207
208      RTE
209      NOP
210
211      .align     4
212
213  __INT_TERM:
214      MOV.L      #MDRBBLset,R0
215      LDC.L      R0,SR
216
217      POP_EXP_BASE_REG
218
219      RTE
220      NOP
221
222      .pool
223
224  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
225  ;          BSC INIT
226  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
227  BSC_INIT:
228      MOV.L      #H'FF801000,R0
229      MOV.L      #H'00000000,R1
230      MOV.L      R1,@R0
231      MOV.L      #H'FF801004,R0
232      MOV.L      R1,@R0
233      SYNCO
234
235      MOV.L      #H'FF802000,R0
236      MOV.L      #H'00000000,R1
237      MOV.L      R1,@R0
238      MOV.L      #H'FF802004,R0
239      MOV.L      #H'03333300,R1
240      MOV.L      R1,@R0
241      SYNCO
242
243      MOV.L      #H'FF802008,R0
244      MOV.L      #H'00000000,R1
245      MOV.L      R1,@R0
246      MOV.L      #H'FF80200C,R0
247      MOV.L      #H'0100000A,R1
    
```

```

248      MOV.L      R1,@R0
249      SYNCO
250
251      MOV.L      #H'FF802030,R0
252      MOV.L      #H'00000000,R1
253      MOV.L      R1,@R0
254      MOV.L      #H'FF802034,R0
255      MOV.L      #H'42424300,R1
256      MOV.L      R1,@R0
257      SYNCO
258
259      MOV.L      #H'FF802038,R0
260      MOV.L      #H'00000000,R1
261      MOV.L      R1,@R0
262      MOV.L      #H'FF80203C,R0
263      MOV.L      #H'02000008,R1
264      MOV.L      R1,@R0
265      SYNCO
266
267      MOV.L      #BSC_INIT_END,R0
268      JMP        @R0
269      NOP
270
271      .pool
272
273      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
274      ;          SDRAM INIT
275      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
276      SDRAM_INIT:
277      MOV.L      #H'FF800008,R0
278      MOV.L      #H'00000000,R1
279      MOV.L      R1,@R0
280      MOV.L      #H'FF80000C,R0
281      MOV.L      #H'034A0041,R1
282      MOV.L      R1,@R0
283      SYNCO
284
285      MOV.L      #H'FF800030,R0
286      MOV.L      #H'00000000,R1
287      MOV.L      R1,@R0
288      MOV.L      #H'FF800034,R0
289      MOV.L      #H'00000400,R1
290      MOV.L      R1,@R0
291      SYNCO
292
293      MOV.L      #H'FF800018,R0
294      MOV.L      #H'00000000,R1
295      MOV.L      R1,@R0
296      MOV.L      #H'FF80001C,R0
297      MOV.L      #H'00B80127,R1
298      MOV.L      R1,@R0
299      SYNCO
300
301      MOV.L      #H'FF800014,R0
302      MOV.L      #H'00000003,R1
303      MOV.L      R1,@R0
304      SYNCO
305
306      MOV.L      #H'00000002,R1
307      MOV.L      R1,@R0
308      SYNCO
309
310      MOV.L      R1,@R0
    
```

```

311          SYNCO
312
313          MOV.L      #H'00000004,R1
314          MOV.L      R1,@R0
315          SYNCO
316
317          MOV.L      R1,@R0
318          SYNCO
319
320          MOV.L      R1,@R0
321          SYNCO
322
323          MOV.L      R1,@R0
324          SYNCO
325
326          MOV.L      R1,@R0
327          SYNCO
328
329          MOV.L      R1,@R0
330          SYNCO
331
332          MOV.L      R1,@R0
333          SYNCO
334
335          MOV.L      R1,@R0
336          SYNCO
337
338          MOV.L      #H'FFA00198,R0
339          MOV.L      #H'00000000,R1
340          MOV.L      R1,@R0
341          SYNCO
342
343          MOV.L      #H'FF800008,R0
344          MOV.L      #H'00000000,R1
345          MOV.L      R1,@R0
346          MOV.L      #H'FF80000C,R0
347          MOV.L      #H'034A0241,R1
348          MOV.L      R1,@R0
349          SYNCO
350
351          MOV.L      #H'FF800014,R0
352          MOV.L      #H'00000001,R1
353          MOV.L      R1,@R0
354          SYNCO
355
356          MOV.L      #SDRAM_INIT_END,R0
357          JMP        @R0
358          NOP
359
360          .pool
361
362          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
363          ;          SPECIAL STACK(for TLBmiss Handler)
364          ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
365          .section    SP_S,data
366          SP_STACK:
367          .res.b      H'200
368
369          .END
    
```

3.2 サンプルプログラムリスト "vecttbl.src"

```

1  ;/*****"FILE COMMENT"*****/
2  ;*      System Name :   SH7764 Sample Program
3  ;*      File Name   :   vecttbl.src
4  ;*      Version    :   1.00.00
5  ;*      Contents   :   SH7764 Intialize Program
6  ;*      Model      :   Renesas SH7764 Board R0K507764E001BR
7  ;*      CPU        :   SH7764
8  ;*      Compiler   :   SHC.9.03.00
9  ;*      OS         :   none
10 ;*
11 ;*      note       :   < Caution >
12 ;*                  This sample program is provided simply as a reference and
13 ;*                  its operation is not guaranteed.
14 ;*                  Use this sample program as a technical reference when
15 ;*                  developing software.
16 ;*
17 ;*                  Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
18 ;*
19 ;*      History    :   2009/03/01 ver 1.00.00
20 ;*
21 ;*****/
22
23         .include   "vect.inc"
24
25         .section   VECTTBL,data
26         .export    _RESET_VECTORS
27
28 _RESET_VECTORS:
29         ;H'000      Power On Reset(Hitachi-UDI RESET)
30         .data.l    _PowerON_Reset
31
32         ;H'020      Manual Reset
33         .data.l    _Manual_Reset
34
35         ;H'040-120  Reserved
36         .datab.l   8,H'00000000
37
38         ;H'140 TLB Reset(DATA TLB Reset)
39         .data.l    _TLB_Reset
40
41         .section   INTTBL, data
42         .export    _INT_VECTORS
43
44 _INT_VECTORS:
45         ;H'040 Data TLB miss exception(read)
46         .data.l    _INT_TLB_MISS_READ_EXP
47
48         ;H'060 Data TLB miss exception(write)
49         .data.l    _INT_TLB_MISS_WRITE_EXP
50
51         ;H'080 Initial page write exception
52         .data.l    _INT_TLB_INIT_PAGE_EXP
53
54         ;H'0A0 Data TLB protection violation exception (read)
55         .data.l    _INT_TLB_PROTECT_READ_EXP
56
57         ;H'0C0 Data TLB protection violation exception (write)
58         .data.l    _INT_TLB_PROTECT_WRITE_EXP
59
60         ;H'0E0 Data address error(read)
    
```

```

61      .data.l   _INT_ADR_ERROR_READ
62
63      ;H'100 Data address error(write)
64      .data.l   _INT_ADR_ERROR_WRITE
65
66      ;H'120 FPU exception
67      .data.l   _INT_FPU_EXP
68
69      ;H'140 Instruction TLB multiple-hit exception
70      .data.l   _TLB_Reset
71
72      ;H'160 Unconditional trap(TRAPA)
73      .data.l   _INT_TRAP
74
75      ;H'180 General illegal instruction exception
76      .data.l   _INT_ILLEGAL_INST_EXP
77
78      ;H'1A0 Slot illegal instruction exception
79      .data.l   _INT_ILLEGAL_SLOT_EXP
80
81      ;===== EXTERNAL INTERRUPT =====
82      ;H'1C0 NMI
83      .data.l   _INT_NMI
84
85      ;H'1E0 USER_BREAK
86      .data.l   _INT_USER_BREAK
87
88      ;H'200 RESERVED
89      ;H'220 RESERVED
90      .datab.l   2,H'00000000
91
92      ;H'240 Interrupt IRQ0
93      .data.l   _INT_IRQ0
94
95      ;H'260 RESERVED
96      .data.l   H'00000000
97
98      ;H'280 Interrupt IRQ1
99      .data.l   _INT_IRQ1
100
101     ;H'2A0 ~ H'540 RESERVED
102     .datab.l   22,H'00000000
103
104     ;===== WDT =====
105     ;H'560 WDT interval timer interrupt
106     .data.l   _INT_WDT_ITI
107
108     ;===== TMU(ch0-ch2) =====
109     ;H'580 TMU ch-0 underflow interrupt
110     .data.l   _INT_TMU0_TUNI0
111
112     ;H'5A0 TMU ch-1 underflow interrupt
113     .data.l   _INT_TMU1_TUNI1
114
115     ;H'5C0 TMU ch-2 underflow interrupt
116     .data.l   _INT_TMU2_TNUI2
117
118     ;H'5E0 TMU ch-2 inputcapture interrupt
119     .data.l   _INT_TMU2_TICPI2
120
121     ;===== H-UDI =====
122     ;H'600 H-UDI interrupt
123     .data.l   _INT_H_UDII
    
```

```

124
125         ;===== LCDC =====
126         ;H'620 LCDC interrupt
127         .data.l     _INT_LCDC
128
129         ;===== DMAC(ch0-ch3) =====
130         ;H'640 ch-0 DMA transmit end or halfend interrupt
131         .data.l     _INT_DMAC_DMINT0
132
133         ;H'660 ch-1 DMA transmit end or halfend interrupt
134         .data.l     _INT_DMAC_DMINT1
135
136         ;H'680 ch-2 DMA transmit end or halfend interrupt
137         .data.l     _INT_DMAC_DMINT2
138
139         ;H'6A0 ch-3 DMA transmit end or halfend interrupt
140         .data.l     _INT_DMAC_DMINT3
141
142         ;H'6C0 ch0-5,ch6-11 DMA address error interrupt
143         .data.l     _INT_DMAC_DMAE
144
145         ;H'6E0 RESERVED
146         .data.l     H'00000000
147
148         ;===== SCIF(ch0) =====
149         ;H'700 SCIF ch-0 receive error interrupt
150         .data.l     _INT_SCIF0_ERI0
    
```

...省略

3.3 サンプルプログラムリスト "resetprg.c"

```

1  /*"FILE COMMENT"*****
2  *      System Name :   SH7764 Sample Program
3  *      File Name   :   resetprg.c
4  *      Version    :   1.00.00
5  *      Contents   :   SH7764 Intialize Program
6  *      Model      :   Renesas SH7764 Board R0K507764E001BR
7  *      CPU        :   SH7764
8  *      Compiler   :   SHC.9.03.00
9  *      OS         :   none
10 *
11 *      note       :   < Caution >
12 *                  This sample program is provided simply as a reference and
13 *                  its operation is not guaranteed.
14 *                  Use this sample program as a technical reference when
15 *                  developing software.
16 *
17 *                  Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
18 *
19 *      History    :   2009/03/01 ver 1.00.00
20 *
21 *****/
22
23
24 #include <machine.h>
25 #include <_h_c_lib.h>
26
27 #include "iodefine.h"
28 #include "stacksct.h"
29
30 static void Purge_OCBP(unsigned long *,unsigned long *);
31
32 #define SR_Default 0x700000F0          // MD/RB/BL Set, IMASK Level15
33 #define SR_Init 0x400000F0           // MD/RB/BL Clear, IMASK Level15
34 #ifdef _FPD // when -fpu=double is specified
35 #define FPSCR_Init 0x000C0001
36 #else
37 #define FPSCR_Init 0x00040001
38 #endif
39 #define INT_OFFSET 0x100UL
40
41 #define RAMCR_ADDRESS 0xff000074
42 #define RAMCR_INIT_VALUE 0x00000200
43
44 #ifdef __cplusplus
45 extern "C" {
46 #endif
47 extern void INTHandlerPRG(void);
48 void PowerON_Reset(void);
49 void Manual_Reset(void);
50 void main(void);
51 #ifdef __cplusplus
52 }
53 #endif
54
55 #ifdef __cplusplus // Use Hardware Setup
56 extern "C" {
57 #endif
58 extern void HardwareSetup(void);
59 #ifdef __cplusplus
60 }

```

```

61  #endif
62
63  #pragma section _ResetPRG
64  #pragma entry PowerON_Reset
65
66  void PowerON_Reset(void)
67  {
68      _UDWORD* ramcr_address;
69
70      /* ===== FPU Setup ===== */
71      set_fpscr(FPSCR_Init);
72
73      /* ===== IMASK Set ===== */
74      set_imask(15);          // IMASK Level15
75
76      /* ===== Module Setup Function ===== */
77      HardwareSetup();       // Use Hardware Setup
78
79      /* ===== B D Section Initial ===== */
80      _INITSCT();
81
82      Purge_OCBP(__sectop("B"), __secend("B"));  /* Purge Bsection */
83      Purge_OCBP(__sectop("R"), __secend("R"));  /* Purge Rsection */
84
85      /* ===== VBR Setup ===== */
86      set_vbr((void *)((_UINT)INTHandlerPRG - INT_OFFSET));
87
88
89      // _CALL_INIT();          // Remove the comment when you use global class object
90
91      // _INIT_IOLIB();        // Enable I/O in the application(both SIM I/O and hardware I/O)
92
93      CACHE_RAMCR = RAMCR_INIT_VALUE;
94
95      /* ===== SR Init ===== */
96      set_cr(SR_Init);
97
98      /* ===== IMASK Set ===== */
99      // set_imask(0);          // IMASK Level0
100
101      /* ===== Main Function ===== */
102      main();
103
104      // _CLOSEALL();          // Close I/O in the application(both SIM I/O and hardware I/O)
105
106      // _CALL_END();          // Remove the comment when you use global class object
107
108      sleep();
109  }
110
111  void Manual_Reset(void)
112  {
113  }
114
115  static void Purge_OCBP(unsigned long *start, unsigned long *end)
116  {
117      long  addr_length;
118      unsigned long *start1;
119
120      addr_length = (unsigned long)end - (unsigned long)start;
121
122      start1 = start +8;
123

```

```

124     /* If purge cache is bigger than 4 entry, then use OCBP_LOOP_1*/
125     if(addr_length > 32)          /* OCBP_LOOP_1 */
126     {
127         do{
128             ocbp(start);
129             start += 16;
130             ocbp(start1);
131             start1 += 16;
132         }while(start1 < (end + 8));
133     }
134     else                          /* OCBP_LOOP_2 */
135     {
136         while(start < end)
137         {
138             ocbp(start);
139             start = start + 8;
140         }
141     }
142 }
    
```

3.4 サンプルプログラムリスト "hwsetup.c"

```

1  /*"FILE COMMENT"*****
2  *   System Name :   SH7764 Sample Program
3  *   File Name   :   hwsetup.c
4  *   Version    :   1.00.00
5  *   Contents   :   SH7764 Intialize Program
6  *   Model      :   Renesas SH7764 Board R0K507764E001BR
7  *   CPU        :   SH7764
8  *   Compiler   :   SHC.9.03.00
9  *   OS         :   none
10 *
11 *   note       :   < Caution >
12 *               This sample program is provided simply as a reference and
13 *               its operation is not guaranteed.
14 *               Use this sample program as a technical reference when
15 *               developing software.
16 *
17 *               Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
18 *
19 *   History    :   2009/03/01 ver 1.00.00
20 *
21 *****/
22
23
24 #include <machine.h>
25 #include "iodefine.h"
26 #ifdef __cplusplus
27 extern "C" {
28 #endif
29 extern void HardwareSetup(void);
30 #ifdef __cplusplus
31 }
32 #endif
33
34 void cache_setup(void);
35 void gpio_initial(void);
36
37 #define CCR_VALUE 0x00000105 /* cashe copy back mode */
38
39 void HardwareSetup(void)
40 {
41     /* ===== Cache Setup ===== */
42     cache_setup();
43
44     /* ===== GPIO Initial ===== */
45     gpio_initial();
46 }
47
48 void cache_setup(void)
49 {
50     CACHE.CCR = (CCR_VALUE | 0x00000808);
51     icbi((void *)cache_setup);
52 }
53
54 void gpio_initial(void)
55 {
56     #if 0
57     /* ----- IRQ Select ----- */
58     GPIO.PTSEL_K.BIT._PTSEL_K7 = 1; // IRQ1
59     GPIO.PTSEL_S.BIT._PTSEL_S15 = 0; // IRQ0
60

```

```

61     /* ----- ATAPI Select ----- */
62     GPIO.PTSEL_R.WORD = 0xffff;
63     GPIO.PTSEL_S.WORD |= 0x3ffe;
64
65     /* ----- SCIF0 Select ----- */
66     GPIO.PTSEL_K.BIT._PTSEL_K6 = 0; // SCK0
67     GPIO.PTSEL_P.BIT._PTSEL_P11 = 0; // RXD0
68     GPIO.PTSEL_P.BIT._PTSEL_P10 = 0; // TXD0
69
70     /* ----- FLCTL Select ----- */
71     GPIO.PTSEL_A.BIT._PTSEL_A5 = 0; // FCE
72     GPIO.PTSEL_A.BIT._PTSEL_A4 = 0; // FRE
73     GPIO.PTSEL_A.BIT._PTSEL_A3 = 0; // FWE
74     GPIO.PTSEL_C.BIT._PTSEL_C0 = 0; // FALE
75     GPIO.PTSEL_K.BIT._PTSEL_K6 = 2; // FCLE
76     GPIO.PTSEL_K.BIT._PTSEL_K5 = 1; // FR/B
77
78     #if 1 // VDC2 Select
79     /* ----- VDC2 Select ----- */
80     GPIO.PTSEL_G.WORD = 0x5555;
81     GPIO.PTSEL_H.BIT._PTSEL_H3 = 1; // DE_V
82     GPIO.PTSEL_H.BIT._PTSEL_H2 = 1; // DCLKOUT
83     GPIO.PTSEL_H.BIT._PTSEL_H1 = 1; // DR4
84     GPIO.PTSEL_H.BIT._PTSEL_H0 = 1; // DR5
85     GPIO.PTSEL_I.BIT._PTSEL_I7 = 1; // DB3
86     GPIO.PTSEL_I.BIT._PTSEL_I6 = 1; // DB2
87     GPIO.PTSEL_I.BIT._PTSEL_I5 = 1; // DB1
88     GPIO.PTSEL_I.BIT._PTSEL_I4 = 1; // DG1
89     GPIO.PTSEL_I.BIT._PTSEL_I3 = 1; // DG0
90     GPIO.PTSEL_I.BIT._PTSEL_I2 = 1; // DB5
91     GPIO.PTSEL_I.BIT._PTSEL_I1 = 1; // DB4
92     GPIO.PTSEL_I.BIT._PTSEL_I0 = 1; // COM/CDE
93     GPIO.PTSEL_K.BIT._PTSEL_K4 = 1; // DB0
94     GPIO.PTSEL_K.BIT._PTSEL_K3 = 1; // HSYNC
95     GPIO.PTSEL_K.BIT._PTSEL_K2 = 1; // DCLKIN
96     GPIO.PTSEL_K.BIT._PTSEL_K1 = 1; // VSYNC
97     GPIO.PTSEL_K.BIT._PTSEL_K0 = 1; // DE_C/DE_H
98     #else
99     /* ----- LCDC Select ----- */
100    GPIO.PTSEL_G.WORD = 0x0000;
101    GPIO.PTSEL_H.WORD &= ~(0x00ff);
102    GPIO.PTSEL_I.WORD &= ~(0xffff);
103    GPIO.PTSEL_K.WORD &= ~(0x03ff);
104    #endif
105
106    /* ----- ETHER Select ----- */
107    GPIO.PTSEL_D.WORD = 0x0000;
108    GPIO.PTSEL_E.WORD = 0x0000;
109    GPIO.PTSEL_F.BIT._PTSEL_F4 = 0; // EXOUT
110    GPIO.PTSEL_F.BIT._PTSEL_F3 = 0; // LNKSTA
111    GPIO.PTSEL_F.BIT._PTSEL_F2 = 0; // WOL
112    GPIO.PTSEL_F.BIT._PTSEL_F1 = 0; // MDIO
113    GPIO.PTSEL_F.BIT._PTSEL_F0 = 0; // MDC
    
```

...省略

3.5 サンプルプログラムリスト "stacksct.h"

```

1  /*"FILE COMMENT"*****
2  *      System Name :   SH7764 Sample Program
3  *      File Name   :   stacksct.h
4  *      Version    :   1.00.00
5  *      Contents   :   SH7764 Intialize Program
6  *      Model     :   Renesas SH7764 Board R0K507764E001BR
7  *      CPU       :   SH7764
8  *      Compiler  :   SHC.9.03.00
9  *      OS        :   none
10 *
11 *      note       :   < Caution >
12 *                  This sample program is provided simply as a reference and
13 *                  its operation is not guaranteed.
14 *                  Use this sample program as a technical reference when
15 *                  developing software.
16 *
17 *                  Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
18 *
19 *      History    :   2009/03/01 ver 1.00.00
20 *
21 *****/
22 #pragma stacksize 0x400      /* Do not modify this line. */
    
```

3.6 サンプルプログラムリスト "dbsct.c"

```

1  /*"FILE COMMENT"*****
2  *   System Name :   SH7764 Sample Program
3  *   File Name   :   dbsct.c
4  *   Version     :   1.00.00
5  *   Contents    :   SH7764 Intialize Program
6  *   Model       :   Renesas SH7764 Board R0K507764E001BR
7  *   CPU         :   SH7764
8  *   Compiler    :   SHC.9.03.00
9  *   OS          :   none
10 *
11 *   note        :   < Caution >
12 *                   This sample program is provided simply as a reference and
13 *                   its operation is not guaranteed.
14 *                   Use this sample program as a technical reference when
15 *                   developing software.
16 *
17 *                   Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
18 *
19 *   History      :   2009/03/01 ver 1.00.00
20 *
21 *****/
22
23
24 #include "typedefine.h"
25
26 #pragma section $DSEC
27 static const struct {
28     _UBYTE *rom_s;      /* 初期化データセクションの ROM 上の先頭アドレス */
29     _UBYTE *rom_e;      /* 初期化データセクションの ROM 上の最終アドレス */
30     _UBYTE *ram_s;      /* 初期化データセクションの RAM 上の先頭アドレス */
31 } DTBL[] = {
32     { __sectop("D"), __secend("D"), __sectop("R") }
33 };
34 #pragma section $BSEC
35 static const struct {
36     _UBYTE *b_s;        /* 未初期化データセクションの先頭アドレス */
37     _UBYTE *b_e;        /* 未初期化データセクションの最終アドレス */
38 } BTBL[] = {
39     { __sectop("B"), __secend("B") }
40 };
    
```

3.7 サンプルプログラムリスト "main.c"

```

1  /*"FILE COMMENT"*****
2  *   System Name :   SH7764 Sample Program
3  *   File Name   :   sh7764.c
4  *   Version    :   1.00.00
5  *   Contents   :   SH7764 Intialize Program
6  *   Model      :   Renesas SH7764 Board R0K507764E001BR
7  *   CPU        :   SH7764
8  *   Compiler   :   SHC.9.03.00
9  *   OS         :   none
10 *
11 *   note       :   < Caution >
12 *               This sample program is provided simply as a reference and
13 *               its operation is not guaranteed.
14 *               Use this sample program as a technical reference when
15 *               developing software.
16 *
17 *               Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
18 *
19 *   History    :   2009/03/01 ver 1.00.00
20 *
21 *****/
22
23
24 // #include "typedefine.h"
25 #ifdef __cplusplus
26 // #include <ios> // Remove the comment when you use ios
27 // _SINT ios_base::Init::init_cnt; // Remove the comment when you use ios
28 #endif
29
30 void main(void);
31 #ifdef __cplusplus
32 extern "C" {
33 void abort(void);
34 }
35 #endif
36
37 void main(void)
38 {
39 }
40
41 #ifdef __cplusplus
42 void abort(void)
43 {
44 }
45 }
46 #endif
    
```

3.8 サンプルプログラムリスト "intprg.c"

```

1  /*"FILE COMMENT"*****
2  *   System Name :   SH7764 Sample Program
3  *   File Name   :   intprg.c
4  *   Version    :   1.00.00
5  *   Contents   :   SH7764 Intialize Program
6  *   Model      :   Renesas SH7764 Board R0K507764E001BR
7  *   CPU        :   SH7764
8  *   Compiler   :   SHC.9.03.00
9  *   OS         :   none
10 *
11 *   note       :   < Caution >
12 *               This sample program is provided simply as a reference and
13 *               its operation is not guaranteed.
14 *               Use this sample program as a technical reference when
15 *               developing software.
16 *
17 *               Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
18 *
19 *   History    :   2009/03/01 ver 1.00.00
20 *
21 *****/
22
23
24 #include <machine.h>
25 #include "iodefine.h"
26
27 #pragma section _IntPRG
28
29 //;H'040 Data TLB miss exception(read)
30 void INT_TLB_MISS_READ_EXP(void)
31 {
32     /* sleep(); */
33 }
34
35 //;H'060 Data TLB miss exception(write)
36 void INT_TLB_MISS_WRITE_EXP(void)
37 {
38     /* sleep(); */
39 }
40
41 //;H'080 Initial page write exception
42 void INT_TLB_INIT_PAGE_EXP(void)
43 {
44     /* sleep(); */
45 }
46
47 //;H'0A0 Data TLB protection violation exception (read)
48 void INT_TLB_PROTECT_READ_EXP(void)
49 {
50     /* sleep(); */
51 }

```

...省略

3.9 サンプルプログラムリスト "vect.inc"

```

1  ;*"FILE COMMENT"*****
2  ;   System Name :   SH7764 Sample Program
3  ;   File Name   :   vect.inc
4  ;   Version    :   1.00.00
5  ;   Contents   :   SH7764 Intialize Program
6  ;   Model      :   Renesas SH7764 Board R0K507764E001BR
7  ;   CPU        :   SH7764
8  ;   Compiler   :   SHC.9.1.00
9  ;   OS         :   none
10 ;
11 ;   note       :   < Caution >
12 ;               This sample program is provided simply as a reference and
13 ;               its operation is not guaranteed.
14 ;               Use this sample program as a technical reference when
15 ;               developing software.
16 ;
17 ;   Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
18 ;
19 ;   History    :   2009/03/01 ver 1.00.00
20 ;
21 ;*****/
22 ;<<VECTOR DATA START (POWER ON RESET)>>
23 ;H'000 Power On Reset(H-UDI RESET)
24 .import    _PowerON_Reset
25 ;<<VECTOR DATA END (POWER ON RESET)>>
26
27 ;<< VECTOR DATA START (MANUAL RESET) >>
28 ;H'020 Manual Reset (Dummy)
29 .import    _Manual_Reset
30 ;<<VECTOR DATA END (MANUAL RESET)>>
31
32 ; H'040   Data TLB miss exception(read)
33 .import    _INT_TLB_MISS_READ_EXP
34
35 ; H'060   Data TLB miss exception(write)
36 .import    _INT_TLB_MISS_WRITE_EXP
37
38 ;H'080   Initial page write exception
39 .import    _INT_TLB_INIT_PAGE_EXP
40
41 ;H'0A0   Data TLB protection violation exception (read)
42 .import    _INT_TLB_PROTECT_READ_EXP
43
44 ;H'0C0   Data TLB protection violation exception (write)
45 .import    _INT_TLB_PROTECT_WRITE_EXP
46
47 ;H'0E0   Data address error(read)
48 .import    _INT_ADR_ERROR_READ
49
50 ;H'100   Data address error(write)
51 .import    _INT_ADR_ERROR_WRITE
52
53 ;H'120   FPU exception
54 .import    _INT_FPU_EXP

```

...省略

4. 参考ドキュメント

- ハードウェアマニュアル
SH7764 グループ ハードウェアマニュアル (RJJ09B0395)
(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)
- 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)

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	2009.09.10	—	初版発行

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事務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。
 - 1 1. 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。
 - 1 2. 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。
 - 1 3. 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