To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

SH7764 Group

SH7764 Example of Initial Setting

Introduction

This application note gives an example of configuration items to activate the SH7764 Microcomputers.

Target Device

SH7764 (R0K507764E001BR from Renesas Technology Corp.)

Contents

1. Preface

1.1 Specifications

Configure the memory controller unit (MCU), general purpose I/O (GPIO), and cache after the power-on reset is canceled.

1.2 Module Used

- Memory controller unit (MCU)
- Cache

1.3 Applicable Conditions

- MCU: SH7764
- Operating frequency: CPU clock: 324 MHz; SuperHyway clock: 108 MHz; Peripheral clock: 54 MHz; Bus clock: 108 MHz
- Bus width for area 0: 32 bits (MODE3 pin = High, MODE4 pin = High)
- Clock operating mode: Mode 3 (MODE2 = Low, MODE1 = High, MODE0 = High)
- Data alignment: Little endian
- Address mode: 29-bit
- C compiler: SuperH RISC Engine Family C/C++ Compiler Package Ver.9.03 Release00 from Renesas Technology Corp.

The locations of the sections for this sample application are described in table 1.

Table 1 Section Locations

Section	Used As	Area	Address (Virtual Address)	
P	Program area (default)	ROM	0x00003000	P0 area (cacheable area with translatable MMU addresses)
C	Constant area	ROM		
C$BSEC	Address structure for non-initialized data areas	ROM		
C$DSEC	Address structure for initialized data areas	ROM		
D	Initialized data (initial value)	ROM		
B	Non-initialized data area	RAM	0x04000000	
R	Initialized data area	RAM		
S	Stack area	RAM	0x07FFF9F0	
INTHandler	Exception/interrupt handler	ROM	0x80000400	P1 area (cacheable area with untranslatable MMU addresses)
VECTTBL	Reset vector table or interrupt vector table	ROM		
INTTBL	Interrupt mask table	ROM		
P_IntPRG	Interrupt function	ROM		
SP_S	Stack area for the TLB-miss handler	RAM	0x87FFFDF0	
RSTHandler	Reset handler	ROM	0xA0000000	P2 area (uncacheable area with untranslatable MMU addresses)
P_ResetPRG	Initialized program area	ROM	0xA0002000	

2. Description of the Sample Application

This application note describes the SDRAM interface and an example of a program for making the initial settings.

Execution of the sample program described in this application note is a precondition for the sample programs described in other application notes for the SH7764.

2.1 Description of the SDRAM Interface

2.1.1 Overview

The SH7764 MCU memory controller unit includes the SDRAM interface to connect with the SDRAM directly.

Since SDRAMs can be selected by the CS signals, they can be connected to physical areas 1 and 2 while sharing the control signals such as $\overline{\text{RAS}}$.

The control signals used for direct connection with SDRAM are: $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, R/$\overline{\text{W}}$, $\overline{\text{CS1}}$ or $\overline{\text{CS2}}$, DQMLL, DQMLU, DQMUL, DQMUU, $\overline{\text{WE0}}$ to $\overline{\text{WE3}}$ (for 64-bit bus) and CKE. All these signals except $\overline{\text{CS1}}$ and $\overline{\text{CS2}}$ are common to each section, and those except CKE are enabled and latched only when $\overline{\text{CS1}}$ or $\overline{\text{CS2}}$ is asserted.

This module supports burst read/write mode as the SDRAM operating mode. Data bus width can be 32 or 64 bits depending on the BW bit settings in the MIM register.

2.1.2 SDRAM Specifications

Table 2 lists the specification of the SDRAM to use in this application. The specifications of the SDRAM used for this application note (EDS2516AFTA-75-E, manufactured by Elpida Memory, Inc.) are listed in the table below. As detailed specifications differ with the device, be sure to check the data sheet for the product you will be using.

Table 2 SDRAM Specifications

Item	Specification
Clock frequency	133 MHz (at maximum)
Density	256 Mbits (32 MB): 2
Configuration	4 banks × 16 M words × 16 bits
CAS latency	2, 3 (programmable)
Refresh	8,192 refresh cycles every 64 ms
Burst length	1, 2, 4, 8, or full page (programmable)
Row address	A12 to A0
Column address	A8 to A0

2.1.3 Memory Map

Connect the SH7764 CS1 or CS2 spaces to SDRAM. Cache is enabled or disabled by the top 3 bits of the internal address (when the MMU is on). This application allocates the SDRAM in CS1 space. Figure 1 shows the memory map.

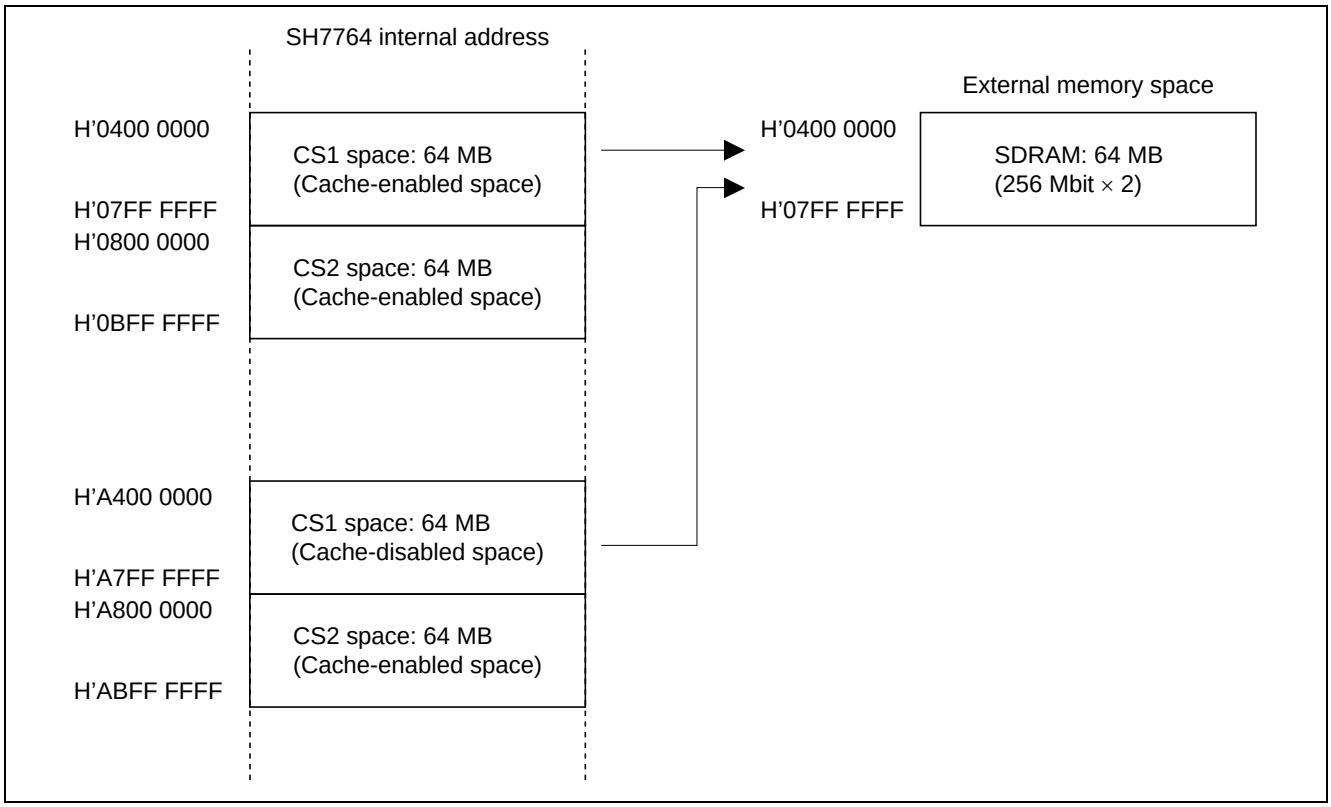

Figure 1 Memory Map Associated with the SDRAM

2.1.4 SDRAM Connection Circuit Example

Figure 2 shows the SDRAM connection circuit example.

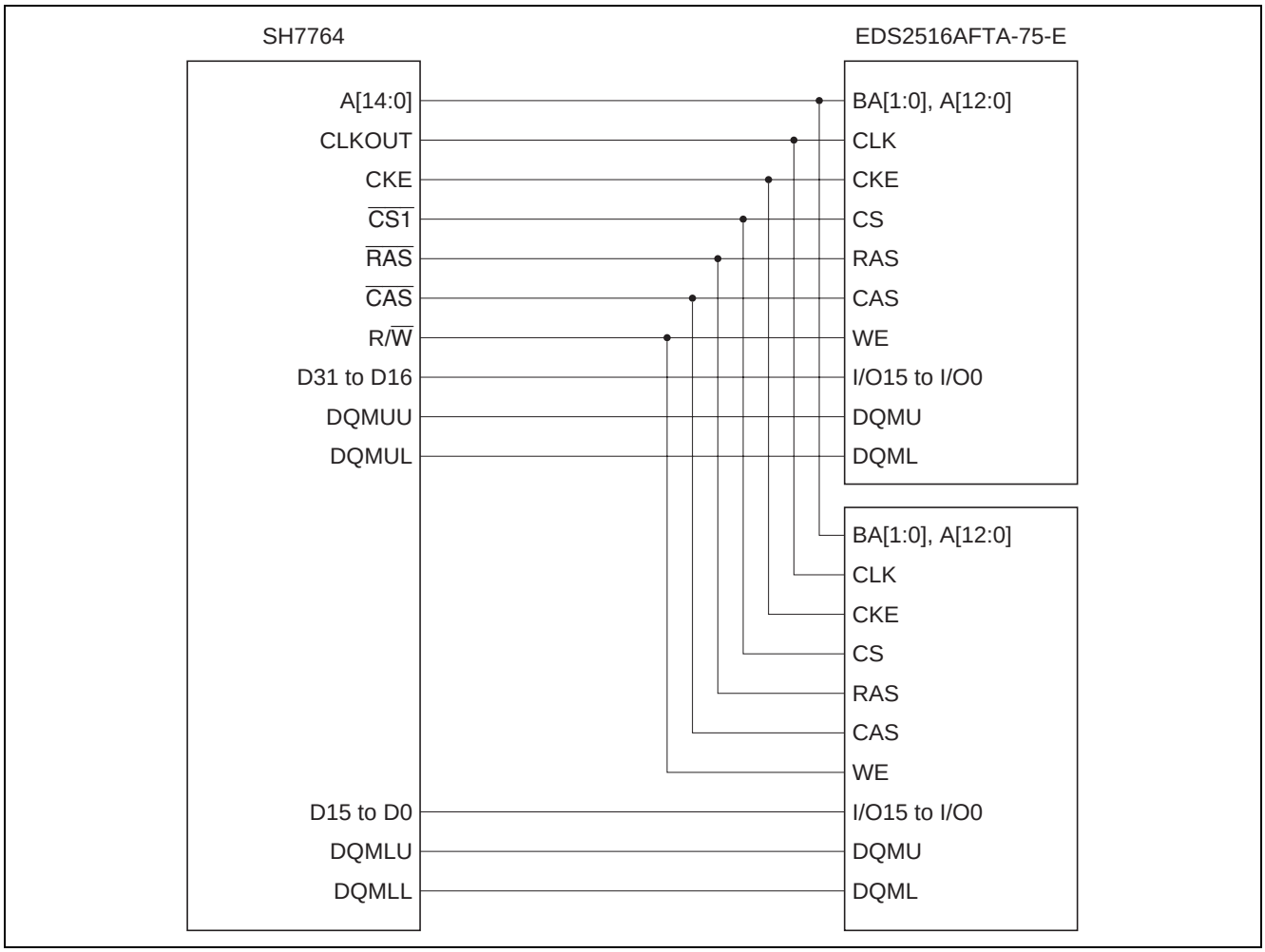

Figure 2 SDRAM Connection Circuit Example (256 Mbit × 2, 32-bit bus)

2.1.5 Address Multiplex Output

Table 3 lists the address-multiplexed output pins used in the sample application.

Note that the SH7764 is not designed to handle row/column configurations other than those listed below.

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

Table 3 Address Multiplex Output

External Bus 32 bits	SDRAM Address	BA1	BA0	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
	SH7764 Address	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
256 Mbits (32 MB) 16 M × 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

2.1.6 Power-on Sequence

To initialize the SDRAM, write the data in the SDRAM mode register after setting the MCU (memory controller unit) register. As the SDRAM requires a certain period of idle cycles, check the specifications of the SDRAM you use, and set the SDRAM mode register after the specified period of idle cycles is elapsed.

Use the arbitrary data in the access address to write in long words (32-bit) in the SDRAM mode register. Specify the access address according to the setting. When writing data in the access address in long words, the MCU provides the set value in the SDRAM mode register.

When other settings than the below are made, correct operation is not guaranteed.

- BL = 4 (When external bus is 64 bits wide) or 8 (When external bus is 32 bits wide)
- BT = Sequential
- LMODE (CL) = 2 or 3
- OPCODE = Burst read & burst write

Figure 3 shows the connection example of the mode register and SDRAM.

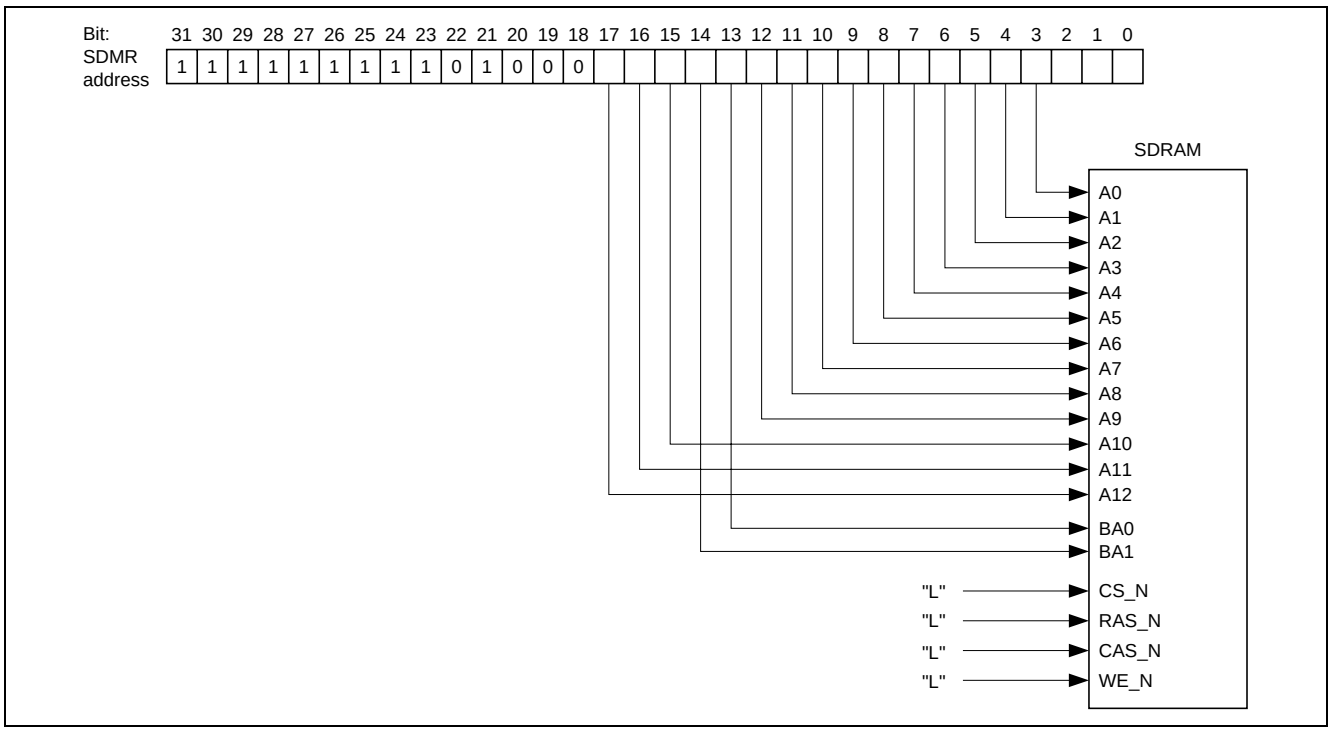

Figure 3 Connection Example of the Mode Register and SDRAM

2.1.7 SDRAM Initialization Procedure (Example)

Figure 4 shows an example of the SDRAM initialization procedure.

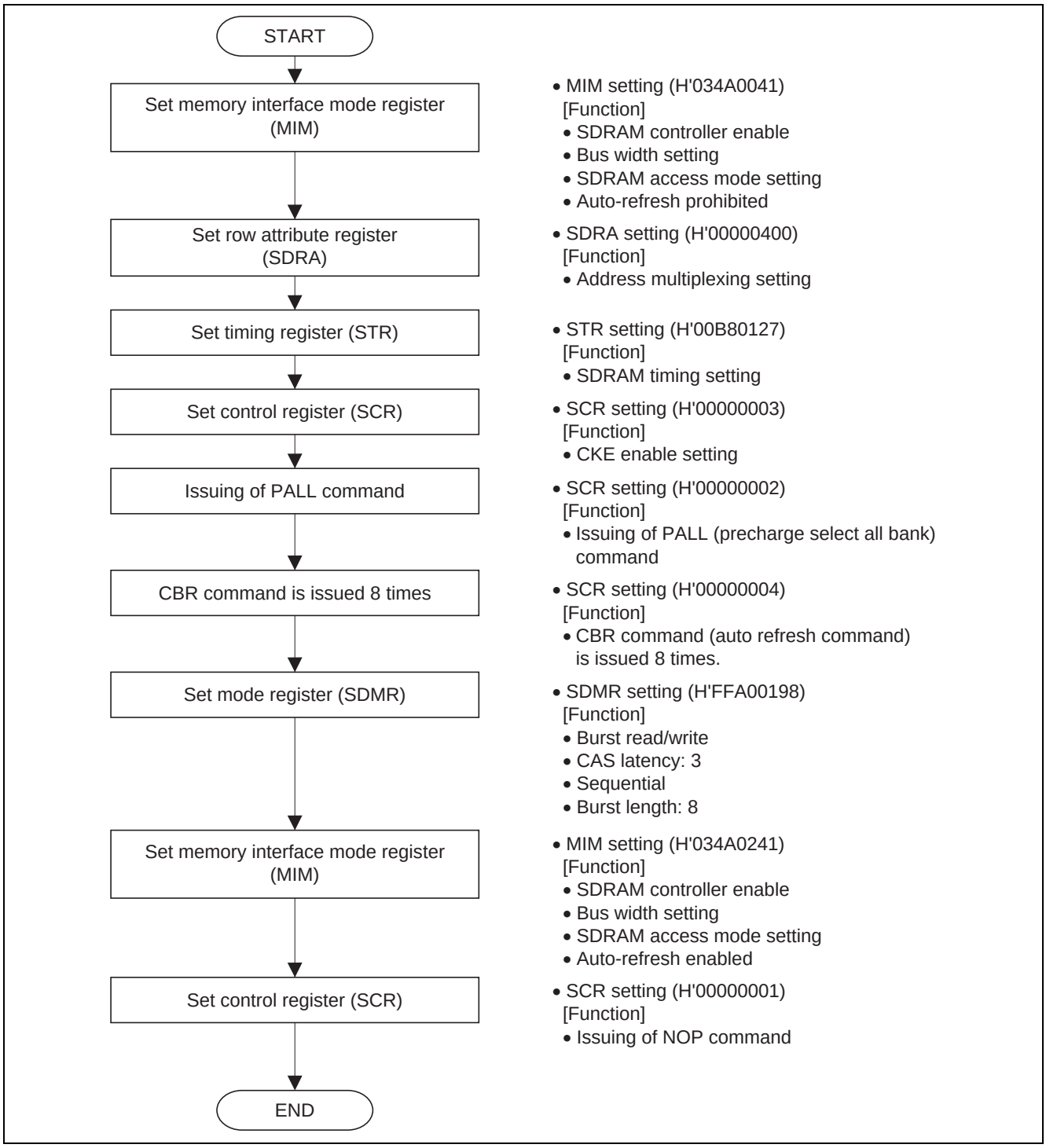

Figure 4 Example of the SDRAM Initialization Procedure

2.2 Description of the Sample Programs

This initialization program consists of the following nine source files:

(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

When an exception (reset, general exception, or interrupt) occurs, the code in the exception handler (vhandler.src) is first to be executed. File vhandler.src contains the code for processing by the handlers for all exceptions and the processing for MCU initialization. Processing in handlers for the reset and for exceptions other than reset is different; for details, see *SuperH RISC engine C/C++ Compiler Package Application Note: [Introduction Guide] Sample FileGuide for SH-3, SH-4, and SH-4A* (REJ06J0012).

If the MMU is to be used, TLB misses must also be taken into consideration. If the stack area is allocated to the P0 or P3 area where address translation by the TLB is enabled, generation of a TLB-miss exception will lead to a further TLB-miss exception every time the TLB-miss handler places a value on the stack, leading to the generation of a manual reset.

A stack area (H'200) for exclusive use in cases where a TLB miss occurs is set up in area P1 where address translation by the TLB is disabled. The TLB-miss handler in this sample program uses the stack area (H'200) for exclusive use until return from the TLB-miss handler. The reset handler (from label _Reset_handler) is activated by a power-on reset. In an event handler processing, a common vector table (_INT_Vectors) for exception handling (resets, general exceptions, and interrupts) is looked up, and the general exception function or interrupt function is determined in accord with the value in the exception event register (EXPEVT) or the interrupt event register (INTEVT), respectively. In an exception handler processing, the BL bit is cleared so that multiple interrupts can be handled before any handler processing of exceptions.

Accordingly, a nonmaskable interrupt (NMI) will be accepted even if a previous NMI is being processed. As a countermeasure against this, processing to clear the BL bit is not executed when the exception code corresponds to the NMI.

(2) vecttbl.src

This file contains definitions for the vector table for exception handling (resets, general exceptions, interrupts) and interrupt mask table. The tables are looked up in processing by the code in vhandler.src (described above), and processing continues in the corresponding exception handling function (resetprg.c or intprg.c).

(3) resetprg.c

This file contains the code for the PowerON_Reset() function, i.e. the reset processing program. The PowerOn_Reset() function contains code that sets the vector base register (VBR), sets the FPSCR, calls the _INITSCT() function, calls the HardwareSetup() function, sets the RAMCD, sets the status register (SR), and calls the main function.

The initial setting of FPSCR selects 32 bits as the transfer size for floating-point instructions. Please change this setting if this is required by the specifications of your application.

The RMD (on-chip memory access mode) bit in the on-chip memory control register (RAMCR) is set to 1, enabling access to on-chip memory in user mode.

The reset vector address is fixed at H'A000 0000. Start addresses for general exceptions and interrupts other than the reset are determined by adding an offset (H'400 for TLB miss exceptions, H'100 for other exceptions in general, and H'600 for interrupts) for the specific event to the vector base address.

In the sample program, the start address of the general exception handler (_INTHandlerPRG) is exported by the exception handler (vhandler.src) and then used to set the VBR in the PowerON_Reset() function, the first to be called in the reset processing program (resetprg.c).

In the exception handler (vhandler.src), the start addresses of the TLB miss handler (_TLBmissHandler) and interrupt handler (_IRQ_Handler) are defined by ".org H'300" and ".org H'500", respectively based on the offset (H'100) for other exceptions in general.
If the sample program is extended to include settings for the internal registers of peripheral modules, the main function is intended to be the source of calls to the corresponding functions. Therefore the status register (SR) is set in privileged mode in the PowerON_Reset() function. If peripheral modules are to be used in user mode, be sure to exclude instructions which are only available in privileged mode.

(4) hwsetup.c

This file is used to set up peripheral module operations. In the sample program, the settings are for the cache. The program enables the instruction cache and operand cache, and selects the copy-back mode. Moreover, although they are not used in the program, GPIO settings for the R0K507764E001BR are included for reference.

(5) stacksct.h

stacksct.h specifies the size of the stack area (the initial value is H'400). When changing the size of the stack area is to be changed, avoid doing so by directly changing this file (to change the address and the size of the stack area, click on Project (P) in the menu bar of the High-performance Embedded Workshop, click on the Edit Configuration (E) item and then select the tab for stack settings)

(6) dbsct.c

initialization of sections: specifically, definition of the addresses where the initialized data sections (sections D and R) and non-initialized data section (section B) start and end. Clearing of section B to 0 and copying of data from section D to section R are handled by the call of the _INITSCT() function from within the PowerOn_Reset() function, which is in resetprg.c.

(7) main.c

main.c is the main function which is called on completion of initialization. User programs should be written within the main routine.

(8) intprg.c

The programs (dummy functions) in this file are called by the handler (vhandler.src) for general exceptions and interrupts other than resets. When interrupts for peripheral functions are used, alter the dummy function by creating new functions on the basis of this sample program (and altering vect.inc and vecttbl.src in accord with any changes of function name), or include a call to a separate function within the dummy function.
Processing for tasks such as clearing interrupt request flags should be written in accord with the descriptions in the *SH7764 Group Hardware Manual* (REJ09B0360).

(9) vect.inc

To enable reference from vhandler.src to the individual processing routines for general exceptions and interrupts in intprg.c, declarations of symbols for external reference are made in vect.inc. When a dummy function of intprg.c is rewritten as a new interrupt function, change the function name in the corresponding entry of this file accordingly. If separate functions are called from within the dummy function, changes to this file are not necessary.

Figure 5 shows the flow of processing from a power-on reset.

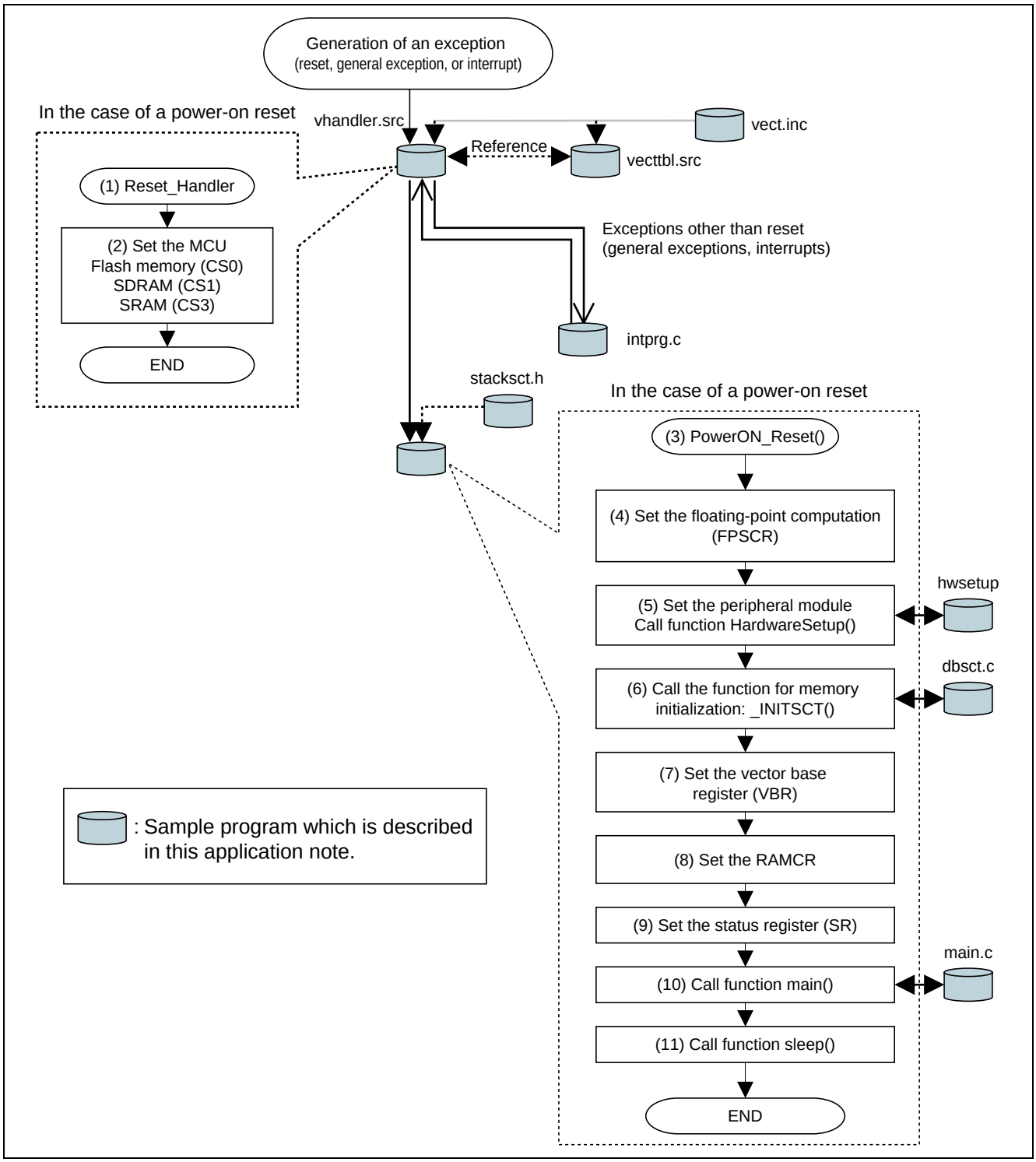

Figure 5 Flow of Processing from Power-On Reset

2.2.1 Example of Settings in the Sample Program

Table 4 is a list of the settings in the sample program.

Table 4 Settings in the Sample Program

Module	Description	
MCU	CS0	NOR-type flash memory Data bus width: 32-bit Idle cycles between read and write access cycles to different areas: 3 Idle cycles between read and write access cycles to same area (area 0): 3 Idle cycles between read and read access cycles to different areas: 3 Idle cycles between read and read access cycles to same area (area 0): 3 Address hold cycles: 1 Specify the number of wait cycles to be inserted during an access to the CSn space: 11
	CS1	SDRAM Access mode: BANK open mode Refresh interval: 546 counts Refresh mode: Auto-refresh Little endian Data bus width: 32-bit Enables SDRAM control Minimum number of cycles for transition from write to read: 9 Minimum number of cycles for transition from read to write: 12 Number of PRE/PALL command issue cycles (Twr): 2 Number of auto-refresh-ACT/auto-refresh issue cycles (Trfc): 8 Minimum number of ACT-PRE command issue cycles (Tras): 7 Number of PRE-ACT command issue cycles (Trp): 3 Number of ACT-auto-refresh/ACT-ACT command issue cycles (Trc): 9 CAS latency: 3 cycles Number of RAS-CAS command issue cycles (Trcd): 3 SDRAM memory configuration specification bit: Row 13 bits × Column 9 bits (16 M × 16 bits product)
	CS3	SRAM Data bus width: 32-bit Idle cycles between write and read/write and write access cycles: 4 Idle cycles between read and write access cycles to different areas: 2 Idle cycles between read and write access cycles to same area: 4 Idle cycles between read and read access cycles to different areas: 2 Idle cycles between read and read access cycles to same area: 2 Address hold cycles: 2 Specify the number of wait cycles to be inserted during an access to the CSn space: 8
Cache	Instruction/operand cache enabled P1 area: Copyback mode P0, U0 and P3 areas: Copyback mode	

2.2.2 Precautions in Using the Sample Programs

- To permit allocation of the B, R and S sections to external memory and initialization, this program initializes the memory controller unit (MCU) and SDRAM interface before section initialization Thus, a function to be executed before section initialization (_INITSCT function execution) should not use global variables which are placed in a section to be initialized by the _INITSCT function.
- This program writes data from the operand cache back to the B and R sections (by using the OCBP instruction for writeback and invalidation) to transfer the cache contents to the external memory. To execute a program in RAM by using the ROM support function, copy the section by using the _INITSCT function. Cached data must then be written back. in the same way as for the B section, etc.
- The stack pointer address set at the beginning of the Entry function (PowerON_Reset function) is the address specified in project generation by the High-performance Embedded Workshop. To change the address and size of the stack area, click on Project (P) in the menu bar for the High-performance Embedded Workshop, click on the Edit Configuration (E) item and then select the the tab for the stack (do not directly change the address of the S section directly as doing so might prevent starting of the dialog dialogue box by clicking on Edit Configuration (E).

3. Listings of Sample Program Files

3.1 Sample Program "vhandler.src"

```
;/*""FILE COMMENT""*********************************************************************
;*        System Name :   SH7764 Sample Program
;*        File Name   :   vhandler.src
;*        Version     :   1.00.00
;*        Contents    :   SH7764 Intialize Program
;*        Model       :   Renesas SH7764 Board R0K507764E001BR
;*        CPU         :   SH7764
;*        Compiler    :   SHC.9.03.00
;*        OS          :   none
;*
;*        note        :   < Caution >
;*                        This sample program is provided simply as a reference and
;*                        its operation is not guaranteed.
;*                        Use this sample program as a technical reference when
;*                        developing software.
;*
;*                        Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
;*
;*       History      :   2009/03/01  ver 1.00.00
;*
;************************************************************************************/
                .include    "vect.inc"

                .import     _INT_VECTORS
                .import     _RESET_VECTORS
                .import     _INT_MASK

EXPEVT          .equ        H'FF000024
INTEVT          .equ        H'FF000028

IMASKclr:       .equ        H'FFFFFF0F
RBBLclr:        .equ        H'CFFFFFFF
MDRBBLset       .equ        H'70000000
MDRBset         .equ        H'60000000
RBclr:          .equ        H'DFFFFFFF

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;             MACRO DEFINITON
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                            .macro PUSH_EXP_BASE_REG
                STC.L       SSR,@-R15
                STC.L       SPC,@-R15
                STS.L       PR,@-R15
                STS.L       FPSCR,@-R15
                STC.L       R7_BANK,@-R15
                STC.L       R6_BANK,@-R15
                STC.L       R5_BANK,@-R15
                STC.L       R4_BANK,@-R15
                STC.L       R3_BANK,@-R15
                STC.L       R2_BANK,@-R15
                STC.L       R1_BANK,@-R15
                STC.L       R0_BANK,@-R15
                            .endm

                            .macro POP_EXP_BASE_REG
                LDC.L       @R15+,R0_BANK
                LDC.L       @R15+,R1_BANK
                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 Sample Program "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 to 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

      …Omitted
```

3.3 Sample Program "resetprg.c"

```
/*""FILE COMMENT""*********************************************************************
*       System Name :   SH7764 Sample Program
*       File Name   :   resetprg.c
*       Version     :   1.00.00
*       Contents    :   SH7764 Intialize Program
*       Model       :   Renesas SH7764 Board R0K507764E001BR
*       CPU         :   SH7764
*       Compiler    :   SHC.9.03.00
*       OS          :   none
*
*       note        :   < Caution >
*                       This sample program is provided simply as a reference and
*                       its operation is not guaranteed.
*                       Use this sample program as a technical reference when
*                       developing software.
*
*                       Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
*
*      History      :   2009/03/01  ver 1.00.00
*
*********************************************************************************/


#include <machine.h>
#include <_h_c_lib.h>

#include "iodefine.h"
#include "stacksct.h"

static void Purge_OCBP(unsigned long *,unsigned long *);

#define SR_Default 0x700000F0           // MD/RB/BL Set, IMASK Level15
#define SR_Init 0x400000F0          // MD/RB/BL Clear, IMASK Level15
#ifdef _FPD  // when -fpu=double is specified
#define FPSCR_Init 0x000C0001
#else
#define FPSCR_Init 0x00040001
#endif
#define INT_OFFSET 0x100UL

#define RAMCR_ADDRESS       0xff000074
#define RAMCR_INIT_VALUE    0x00000200

#ifdef __cplusplus
extern "C" {
#endif
extern void INTHandlerPRG(void);
void PowerON_Reset(void);
void Manual_Reset(void);
void main(void);
#ifdef __cplusplus
}
#endif

#ifdef __cplusplus              // Use Hardware Setup
extern "C" {
#endif
extern void HardwareSetup(void);
#ifdef __cplusplus
}
#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 Sample Program "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 &= ~(0xfffC);
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

      …Omitted
```

3.5 Sample Program "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 Sample Program "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;       /* ROM top address of Initialize data section */
29        _UBYTE *rom_e;       /* ROM end address of Initialize data section */
30        _UBYTE *ram_s;       /* RAM top address of Initialize data section */
31    }   DTBL[] = {
32        { __sectop("D"), __secend("D"), __sectop("R") }
33    };
34    #pragma section $BSEC
35    static const struct {
36        _UBYTE *b_s;         /* top address of no Initialize data section */
37        _UBYTE *b_e;         /* end address of no Initialize data section */
38    }   BTBL[] = {
39        { __sectop("B"), __secend("B") }
40    };
```

3.7 Sample Program "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 Sample Program "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    }

      …Omitted
```

3.9 Sample Program "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

      …Omitted
```

4. Documents for Reference

- Hardware Manual
  SH7764 Group Hardware Manual (REJ09B0360)
  The most up-to-date version of this document is available on the Renesas Technology Website.

- Software Manual (REJ09B0003)
  SH-4A Software Manual
  The most up-to-date version of this document is available on the Renesas Technology Website.

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Sep.28.09	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.