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.

SH7763 Group

SH7763 Example of Initialization

Introduction

This application note describes an example of items that must be set when starting up the SH7763 MCU.

Target Device

SH7763

Contents

1. Preface

1.1 Specifications

The clock pulse generator (CPG), local bus state controller (LBSC), DDR-SDRAM interface (DDRIF), and cache are initialized after release from the reset state.

1.2 Modules Used

- Clock pulse generator (CPG)
- Local bus state controller (LBSC)
- DDR-SDRAM interface (DDRIF)
- Cache

1.3 Applicable Conditions

- Evaluation board: SH7763 Solution Engine (type no.: MS7763SE02) from Hitachi ULSI Systems Co., Ltd.
  External memory (area 0): 16-MB NOR-type flash memory: S29GL128M10TDIR90 from Spansion
  (area 2, 3): 128-MB DDR-SDRAM (64 MB × 2): MT46V32M16TG-6T from Micron
- MCU: SH7763 (R5S77631AY266BGV)
- Operating frequency:

CPU clock:	266.66 MHz
SuperHyway bus clock:	133.33 MHz
Bus clock:	66.66 MHz
DDR-SDRAM clock:	133.33 MHz
Peripheral bus clock 0:	66.66 MHz
Peripheral bus clock 1:	33.33 MHz

- Bus width for area 0: 16-bit (with the MD3 pin at the low level, and MD4 pin at the high level)
- Clock operating mode: Mode 0 (with the MD0 to MD2 pin at the low level)
- Endian: Big endian (with the MD5 pin at the low level)
- Toolchain: SuperH RISC engine Standard Toolchain Ver.9.3.0.0 from Renesas Technology
- Compiler options: Default settings of High-performance Embedded Workshop
  (-cpu=sh4a -include="$(PROJDIR)¥inc"
  -object="$(CONFIGDIR)¥$(FILELEAF).obj" -debug -gbr=auto –chgincpath
  -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0
  -del_vacant_loop=0 -struct_alloc=1 -nologo)
- Section locations: The locations of the sections for this sample application are described in table 1.

Table 1 Allocation of Sections

Section Name	Application of Section	Area	Allocation Address (Virtual Address)	
P	Program area (in the case of none specified)	ROM	0x00002000	Area P0 (caching is enabled, MMU addresses can be translated)
C	Constant area	ROM		
C$BSEC	Address structure for non-initialized data area	ROM		
C$DSEC	Address structure for initialized data area	ROM		
D	Initialized data (initial value)	ROM		
B	Non-initialized data area	RAM	0x08000000	
R	Initialized data area	RAM		
S	Stack area	RAM	0x0FFFF9F0	
INTHandler	Exception/interrupt handler	ROM	0x80000800	Area P1 (caching is enabled, MMU addresses cannot be translated)
VECTTBL	Reset vector table Interrupt vector table	ROM		
INTTBL	Interrupt mask table	ROM		
IntPRG	Interrupt function	ROM		
SP_S	Stack area for handler of TLB misses	RAM	0x8FFFFDF0	
RSTHandler	Reset handler	ROM	0xA0000000	Area P2 (caching is disabled, MMU addresses cannot be translated)
PResetPRG	Reset program	ROM		
PnonCACHE	Program area (non-cacheable access)	ROM		

Stack Settings

A stack area is required to run a program; specify the stack size and stack-pointer address. The High-performance Embedded Workshop automatically sets these to the values which have been set when the project was launched. To change the size and address of the stack area, select: **Project (P)** in the High-performance Embedded Workshop menu bar → **Edit Project Configuration (E)** → the **Stack** tab.

2. Description of Sample Application

2.1 Description of the Sample Program

This sample program serves as an initialization program to specify settings in the following source files.

Specify additional settings for table files to be used for exceptions/interrupts, if necessary.

(1) vhandler.src
(2) resetprg.c
(3) cache.c
(4) cache.h

When an exception (reset, general exception, or interrupt) occurs, an exception handler (vhandler.src) is executed. The vhandler.src contains code for handling exceptions and initializing the LBSC and the DDRIF.

When a power-on reset occurs, a reset handler (_Reset_handler) is started. Unlike files automatically generated by High-performance Embedded Workshop, the reset handler used in this application note has the additional functions of invalidating instruction cache and operand cache, setting the CPG, and initializing the LBSC and the DDRIF. Also, the TLB miss handler has been modified.

The resetprg.c has been created based on an initialization function file automatically generated by High-performance Embedded Workshop, and contains the PowerON_Reset() function, which is to be registered in the vecttbl.src.

The PowerON_Reset() function is the first branch target from the reset handler. PowerON_Reset() specifies settings of the VBR (vector base register) and calls the _INITSCT() function, which copies sections, and a cache validation function. After that, PowerON_Reset() calls the main function.

For setting of the DDR-SDRAM interface (DDRIF), also refer to the separate SH7763 Group application note, *Example of DDR-SDRAM Interface Connection* (REJ06B0935).

For details on initial settings, refer to the SH7730 Group application note, *SH7730 Example of Initial Settings* (REJ06B0848).

For setting of caches, refer to the SH7730 Group application notes concerning caches, *Example of Cache Memory Settings* (REJ06B0851) and *Example of Writing Back from the Operand Cache* (REJ06B0853).

Figure 1 shows a flowchart of the processing performed after a power-on reset. Figure 2 shows a flowchart of the processing performed during the time between a power-on reset and a jump to the main() function. Also, figure 3 shows the processing flow of the PowerON_Reset() function.

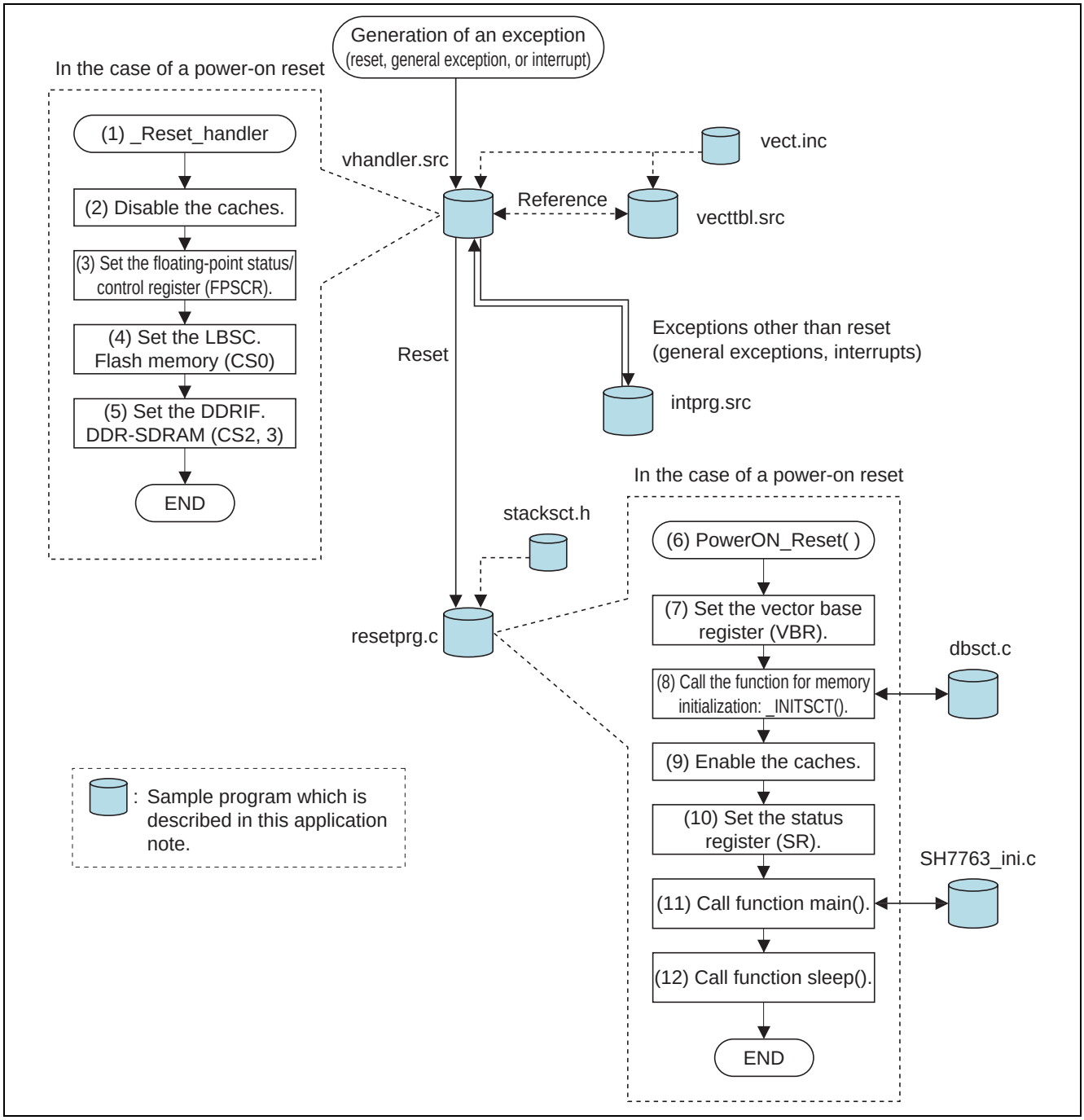

Figure 1 Flow of Processing from Power-On Reset

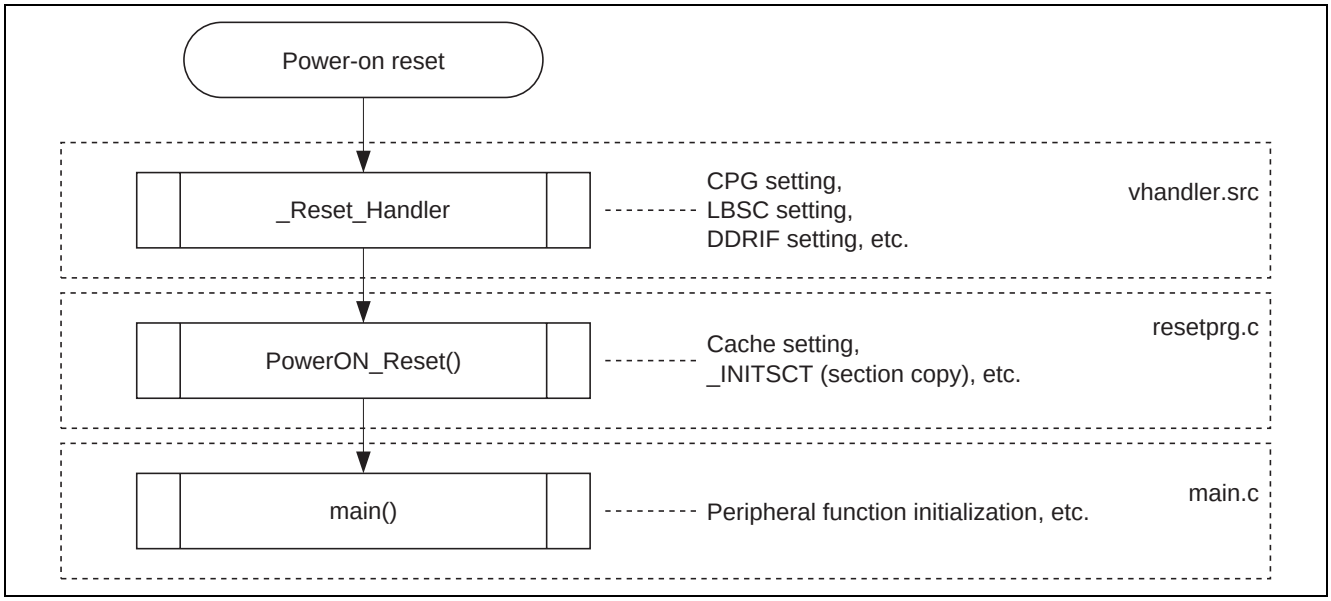

Figure 2 Overview of the Sequence of Operations from Power-on Reset to the Jump to Function main

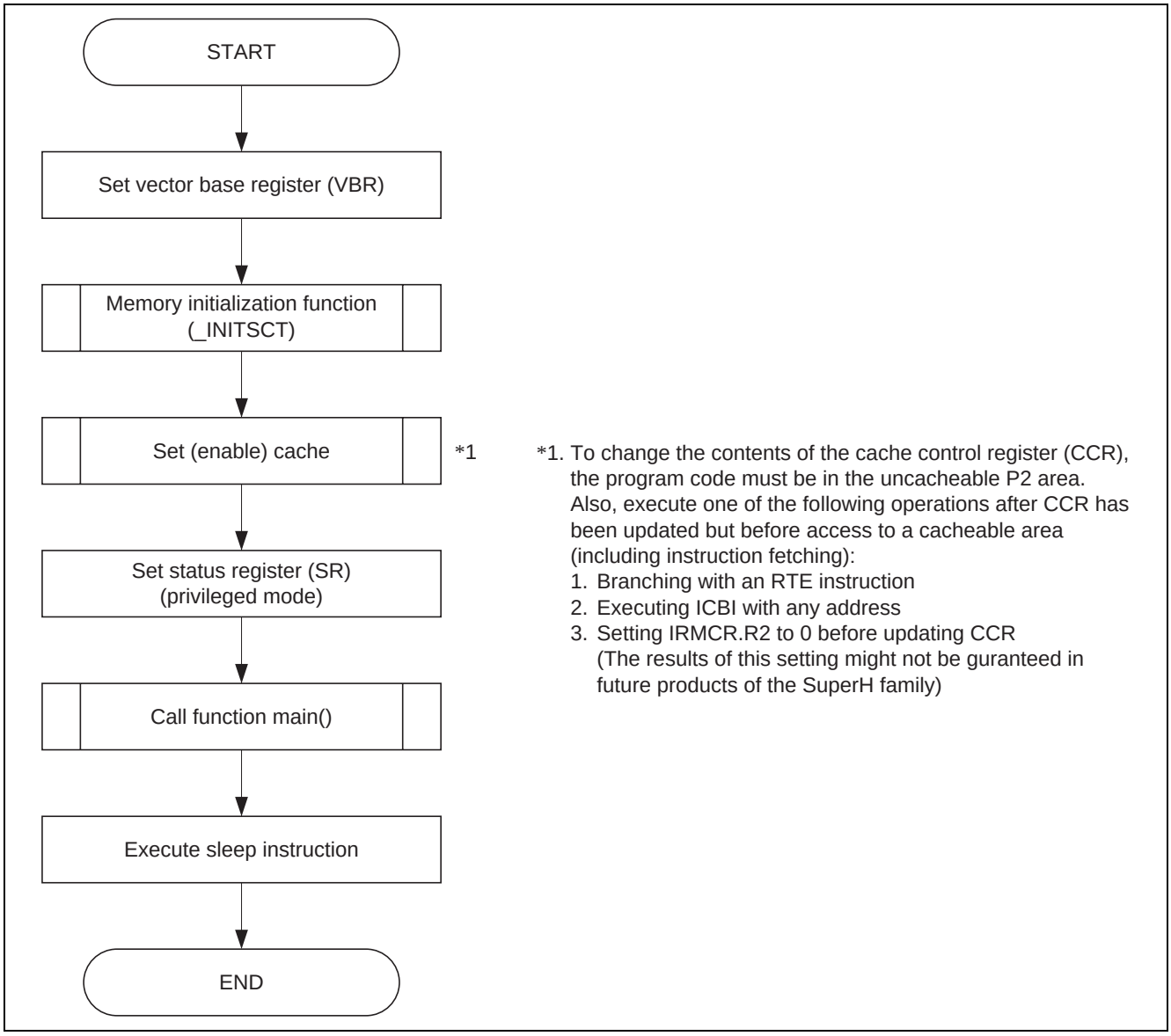

Figure 3 Operations with the PowerON_Reset Function

2.2 Description of Settings in the Sample Program

Table 2 is a list of the settings in the sample program.

Table 2 Settings in the Sample Program

Module	Description
CPG (fixed)	Internal clock: 266.66 MHz SuperHyway bus clock: 133.33 MHz Bus clock: 66.66 MHz DDR-SDRAM clock: 133.33 MHz Peripheral bus clock 0: 66.66 MHz Peripheral bus clock 1: 33.33 MHz
LBSC	CS0: NOR-type flash memory Data bus width: 16-bit (fixed) *1 Cycles of waiting for access: 5 Idle cycles between write-read/write-write: 2 cycles
DDRIF	Connected to CS2 and CS3 Parallel connection of two 512-Mbit (32 M × 16-bits) (Total size 128-Mbyte) Minimum number of cycles from write command to read commands: 4 cycles Minimum number of cycles from read command to write commands: 4 cycles Number of cycles in same bank: 11 cycles PRE/PREALL command issuance cycle: 2 cycles ACT command issuance cycle between banks: 2 cycles Minimum number of cycles between ACT and PRE commands: 6 cycles Auto-refresh/ACT command issuance cycle: 8 cycles CAS latency (CL): 2.5 cycles Number of cycles between RAS and CAS commands: 3 cycles Number of cycles between PRE and ACT commands: 3 cycles
Cache	Instruction/operand cache enabled

Note: *1 Data bus width of area 0 is determined by the level on pins MD4 and MD3.

2.3 Precautions Regarding the Sample Program

In this sample program, the DDR-SDRAM interface (DDRIF) is initialized before the initialization of sections B, and R. This is so that the sections can be allocated to external DDR-SDRAM and then initialized.

For initialization of the sections, the _INITSCT() function that copies data from section D to section R, and relocates symbol to addresses in the R section is used. Therefore, in any function which is executed before the sections are initialized (i.e. before the _INITSCT() function), avoid variables, including global variables, which are to be placed in sections to be initialized by the _INITSCT() function.

3. Listing of the Sample Program

3.1 Sample Program Listing: "vhandler.src"(1)

```
1     ;/***********************************************************************
2     ;* DISCLAIMER
3     ;
4     ;* This software is supplied by Renesas Technology Corp. and is only
5     ;* intended for use with Renesas products. No other uses are authorized.
6     ;
7     ;* This software is owned by Renesas Technology Corp. and is protected under
8     ;* all applicable laws, including copyright laws.
9     ;
10    ;* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11    ;* REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12    ;* INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13    ;* PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14    ;* DISCLAIMED.
15    ;
16    ;* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17    ;* TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18    ;* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19    ;* FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20    ;* AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21    ;
22    ;* Renesas reserves the right, without notice, to make changes to this
23    ;* software and to discontinue the availability of this software.
24    ;* By using this software, you agree to the additional terms and
25    ;* conditions found by accessing the following link:
26    ;* http://www.renesas.com/disclaimer
27    ;***********************************************************************/
28    ;/* Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved.         */
29    ;/*""FILE COMMENT""*********** Technical reference data ***************
30    ;* System Name  : SH7763 Sample Program
31    ;* File Name    : vhandler.src
32    ;* Abstract     : Sample Program for the SH7763 Initial Setting
33    ;* Version      : Ver 1.00
34    ;* Device       : SH7763
35    ;* Tool-Chain   : High-performance Embedded Workshop (Version 4.05.01.001)
36    ;*              : C/C++ Compiler Package for SuperH Family (V.9.03 release00)
37    ;* OS           : None
38    ;* H/W Platform : MS7763SE02
39    ;* Description  : Sample Program for Setting the SH7763 Initialization
40    ;*              :
41    ;* Operation    :
42    ;* Limitation   :
43    ;*              :
44    ;*********************************************************************
45    ;* History      : 28.July.2009 Ver. 1.00 First Release
46    ;*""FILE COMMENT END""**************************************************/
47
48    ;---------------------------------------------------------------------
49    ;                                                                     |
50    ;   FILE        :vhandler.src                                         |
```

3.2 Sample Program Listing: "vhandler.src"(2)

```
51    ;   DATE        :Fri, Jul 24, 2009                                    |
52    ;   DESCRIPTION :Reset/Interrupt Handler                              |
53    ;   CPU TYPE    :SH7763                                               |
54    ;                                                                     |
55    ;   This file is generated by Renesas Project Generator (Ver.4.13).   |
56    ;                                                                     |
57    ;---------------------------------------------------------------------
58
59
60
61          .include    "env.inc"
62          .include    "vect.inc"
63
64    IMASKclr: .equ    H'FFFFFF0F
65    RBBLclr:  .equ    H'CFFFFFFF
66    MDRBBLset:    .equ    H'70000000
67
68          .import     _RESET_Vectors
69          .import     _INT_Vectors
70          .import     _INT_MASK
71
72    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
73    ;*              macro definition                                    *;
74    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
75                    .macro PUSH_EXP_BASE_REG
76              stc.l   ssr,@-r15               ; save ssr
77              stc.l   spc,@-r15               ; save spc
78              sts.l   pr,@-r15                ; save context registers
79              sts.l   fpscr,@-r15             ; save fpscr registers
80              stc.l   r7_bank,@-r15
81              stc.l   r6_bank,@-r15
82              stc.l   r5_bank,@-r15
83              stc.l   r4_bank,@-r15
84              stc.l   r3_bank,@-r15
85              stc.l   r2_bank,@-r15
86              stc.l   r1_bank,@-r15
87              stc.l   r0_bank,@-r15
88                      .endm
89    ;
90                      .macro POP_EXP_BASE_REG
91              ldc.l   @r15+,r0_bank           ; recover registers
92              ldc.l   @r15+,r1_bank
93              ldc.l   @r15+,r2_bank
94              ldc.l   @r15+,r3_bank
95              ldc.l   @r15+,r4_bank
96              ldc.l   @r15+,r5_bank
97              ldc.l   @r15+,r6_bank
98              ldc.l   @r15+,r7_bank
99              lds.l   @r15+,fpscr
100             lds.l   @r15+,pr
```

3.3 Sample Program Listing: "vhandler.src"(3)

```
101          ldc.l   @r15+,spc
102          ldc.l   @r15+,ssr
103                  .endm
104  ;
105  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
106  ;     reset                                                       ;
107  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
108        .section    RSTHandler,code
109  _ResetHandler:
110
111               mov.l   #H'FF00001C,r0 ;set CCR address
112               mov.l   #H'00000808,r1 ;IC,OC Invalidate
113               mov.l   r1,@r0
114  ;
115               mov.l   #H'00040001,r0 ;set single precision mode
116  ;            mov.l   #H'000C0001,r0 ;set double precision mode
117               lds.l   r0,fpscr
118  ;
119               mov.l   #LBSC_INIT,r0
120               jmp     @r0
121               nop
122  LBSC_INIT_END:
123  ;
124               mov.l   #DDRIF_INIT,r0
125               jmp     @r0
126               nop
127  DDRIF_INIT_END:
128  ;
129               mov.l   #EXPEVT,r0
130               mov.l   @r0,r0
131               shlr2   r0
132               shlr    r0
133               mov.l   #_RESET_Vectors,r1
134               add     r1,r0
135               mov.l   @r0,r0
136               jmp     @r0
137               nop
138  ;
139  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
140  ;     exceptional interrupt                                       ;
141  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
142        .section    INTHandler,code
143        .export     _INTHandlerPRG
144  _INTHandlerPRG:
145  _ExpHandler:
146               PUSH_EXP_BASE_REG
147  ;
148               mov.l   #EXPEVT,r0         ; set event address
149               mov.l   @r0,r1             ; set exception code
150               mov.l   #_INT_Vectors,r0   ; set vector table address
```

3.4 Sample Program Listing: "vhandler.src"(4)

```
151                 add     #-(h'40),r1         ; exception code - h'40
152                 shlr2   r1
153                 shlr    r1
154                 mov.l   @(r0,r1),r3         ; set interrupt function addr
155   ;
156                 mov.l   #_INT_MASK,r0       ; interrupt mask table addr
157                 shlr2   r1
158                 mov.b   @(r0,r1),r1         ; interrupt mask
159                 extu.b  r1,r1
160   ;
161                 stc     sr,r0               ; save sr
162                 mov.l   #(RBBLclr&IMASKclr),r2     ; RB,BL,mask clear data
163                 and     r2,r0               ; clear mask data
164                 or      r1,r0               ; set interrupt mask
165                 ldc     r0,ssr              ; set current status
166   ;
167                 ldc.l   r3,spc
168                 mov.l   #__int_term,r0      ; set interrupt terminate
169                 lds     r0,pr
170   ;
171                 rte
172                 nop
173   ;
174                 .pool
175   ;
176   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
177   ;     Interrupt terminate                                             ;
178   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
179                 .align  4
180   __int_term:
181                 mov.l   #MDRBBLset,r0  ; set MD,BL,RB
182                 ldc.l   r0,sr               ;
183                 POP_EXP_BASE_REG
184                 rte                 ; return
185                 nop
186   ;
187                 .pool
188   ;
189   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
190   ;     TLB miss interrupt                                          ;
191   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
192         .org    H'300
193   _TLBmissHandler:
194                 PUSH_EXP_BASE_REG
195   ;
196                 mov.l   #EXPEVT,r0          ; set event address
197                 mov.l   @r0,r1              ; set exception code
198                 mov.l   #_INT_Vectors,r0    ; set vector table address
199                 add     #-(h'40),r1         ; exception code - h'40
200                 shlr2   r1
```

3.5 Sample Program Listing: "vhandler.src"(5)

```
201                 shlr    r1
202                 mov.l   @(r0,r1),r3         ; set interrupt function addr
203   ;
204                 mov.l   #_INT_MASK,r0       ; interrupt mask table addr
205                 shlr2   r1
206                 mov.b   @(r0,r1),r1         ; interrupt mask
207                 extu.b  r1,r1
208   ;
209                 stc     sr,r0               ; save sr
210                 mov.l   #(RBBLclr&IMASKclr),r2     ; RB,BL,mask clear data
211                 and     r2,r0               ; clear mask data
212                 or      r1,r0               ; set interrupt mask
213                 ldc     r0,ssr              ; set current status
214   ;
215                 ldc.l   r3,spc
216                 mov.l   #__int_term,r0      ; set interrupt terminate
217                 lds     r0,pr
218   ;
219                 rte
220                 nop
221   ;
222                 .pool
223   ;
224   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
225   ;     IRQ                                                           ;
226   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
227         .org    H'500
228   _IRQHandler:
229                 PUSH_EXP_BASE_REG
230   ;
231                 mov.l   #INTEVT,r0          ; set event address
232                 mov.l   @r0,r1              ; set exception code
233                 mov.l   #_INT_Vectors,r0    ; set vector table address
234                 add     #-(h'40),r1         ; exception code - h'40
235                 shlr2   r1
236                 shlr    r1
237                 mov.l   @(r0,r1),r3         ; set interrupt function addr
238   ;
239                 mov.l   #_INT_MASK,r0       ; interrupt mask table addr
240                 shlr2   r1
241                 mov.b   @(r0,r1),r1         ; interrupt mask
242                 extu.b  r1,r1
243   ;
244                 stc     sr,r0               ; save sr
245                 mov.l   #(RBBLclr&IMASKclr),r2     ; RB,BL,mask clear data
246                 and     r2,r0               ; clear mask data
247                 or      r1,r0               ; set interrupt mask
248                 ldc     r0,ssr              ; set current status
249   ;
250                 ldc.l   r3,spc
```

3.6 Sample Program Listing: "vhandler.src"(6)

```
251                 mov.l   #__int_term,r0      ; set interrupt terminate
252                 lds     r0,pr
253   ;
254                 rte
255                 nop
256   ;
257                 .pool
258   ;
259   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
260   ;     LBSC_INIT                                                 ;
261   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
262   LBSC_INIT:
263                 mov.l   #H'FE600020,r0      ;set MMSELR address
264                 mov.l   #H'A5A50002,r1      ;set CS2,3 DDRIF
265                 synco
266                 mov.l   r1,@r0
267                 mov.l   @r0,r2              ;dummy read
268                 mov.l   @r0,r2              ;dummy read
269                 synco
270   ;
271                 mov.l   #H'FF801000,r0      ;set BCR address
272                 mov.l   #H'00000000,r1      ;set BUS condition
273                 mov.l   r1,@r0
274   ;
275                 mov.l   #H'FF802000,r0      ;set CS0BCR address
276                 mov.l   #H'20000200,r1      ;set for FLASHROM(spansion S29GL128M10TDIR90)
277                 mov.l   r1,@r0
278   ;
279                 mov.l   #H'FF802008,r0      ;set CS0WCR address
280                 mov.l   #H'00000005,r1      ;set for CS0 wait
281                 mov.l   r1,@r0
282   ;
283                 mov.l   #LBSC_INIT_END,r0
284                 jmp     @r0
285                 nop
286   ;
287                 .pool
288   ;
289   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
290   ;     DDRIF_INIT                                                ;
291   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
292   DDRIF_INIT:
293                 mov.l   #H'00007000,r0
294   LOOP1:
295                 dt      r0
296                 bf      LOOP1                   ;200µs wait
297                 nop
298                 nop
299   ;
300                 mov.l   #H'A8000000,r0          ;set dummy read access
```

3.7 Sample Program Listing: "vhandler.src"(7)

```
301                mov.l   @r0,r1
302    ;
303                mov.l   #H'FE80000C,r0          ;set MIM(31-0bit) address
304                mov.l   #H'02EE0109,r1          ;DRAM refresh disable,DLL enable,DDRIF enable
305                mov.l   r1,@r0
306    ;
307                mov.l   #H'02EE0309,r1          ;Refresh enable
308                mov.l   r1,@r0
309    ;
310                mov.l   #H'FE80001C,r0          ;set STR(31-0bit) address
311                mov.l   #H'00050040,r1
312                mov.l   r1,@r0
313    ;
314                mov.l   #H'FE800034,r2          ;set SDR(31-0bit) address
315                mov.l   #H'00000400,r1          ;32Mx16bit
316                mov.l   r1,@r2
317    ;
318                mov.l   #H'FE800014,r2          ;set SCR(31-0bit) address
319                mov.l   #H'00000003,r1          ;SCR M_CKE enable
320                mov.l   r1,@r2
321                mov.l   #H'00000001,r1          ;SCR NOP
322                mov.l   r1,@r2
323                mov.l   #H'00000002,r1          ;SCR PREALL
324                mov.l   r1,@r2
325    ;
326                mov.l   #H'FE902000,r3          ;EMRS DLL enable
327                mov.l   #H'00000000,r4
328                mov.l   r4,@r3
329    ;
330                mov.l   #H'00000001,r1          ;SCR NOP
331                mov.l   r1,@r2
332    ;
333                mov.l   #H'FE900B08,r3          ;MRS DLL reset,CAS Latency=2.5,burstlength=2
334                mov.l   r4,@r3
335    ;
336                mov.l   #H'00000001,r1          ;SCR NOP
337                mov.l   r1,@r2
338                mov.l   #H'00000002,r1          ;SCR PREALL
339                mov.l   r1,@r2
340                mov.l   #H'00000001,r1          ;SCR NOP
341                mov.l   r1,@r2
342                mov.l   #H'00000004,r1          ;SCR REFA
343                mov.l   r1,@r2
344                mov.l   #H'00000001,r1          ;SCR NOP
345                mov.l   r1,@r2
346                mov.l   #H'00000004,r1          ;SCR REFA
347                mov.l   r1,@r2
348    ;
349                mov.l   #H'FE900308,r3          ;MRS Reset Cancel
350                mov.l   r4,@r3
```

3.8 Sample Program Listing: "vhandler.src"(8)

```
;
                mov.l   #H'00001000,r0
LOOP2:
                dt          r0
                bf          LOOP2                   ;more then 200 MCLK wait
                nop
                nop
;
                mov.l   #DDRIF_INIT_END,r0
                jmp     @r0
                nop
;
                .pool
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; SPECIAL STACK(for TLBmiss Handler)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                .section SP_S,data
SP_STACK:
                .res.b  H'200
;
                .end
```

3.9 Sample Program Listing: "resetprg.c"(1)

```
1    /***********************************************************************
2    * DISCLAIMER
3
4    * This software is supplied by Renesas Technology Corp. and is only
5    * intended for use with Renesas products. No other uses are authorized.
6
7    * This software is owned by Renesas Technology Corp. and is protected under
8    * all applicable laws, including copyright laws.
9
10   * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11   * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12   * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13   * PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14   * DISCLAIMED.
15
16   * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17   * TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18   * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19   * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20   * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21
22   * Renesas reserves the right, without notice, to make changes to this
23   * software and to discontinue the availability of this software.
24   * By using this software, you agree to the additional terms and
25   * conditions found by accessing the following link:
26   * http://www.renesas.com/disclaimer
27   ************************************************************************/
28   /* Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved.         */
29   /*""FILE COMMENT""*********** Technical reference data ***************
30   * System Name  : SH7763 Sample Program
31   * File Name    : resetprg.c
32   * Abstract     : Sample Program for the SH7763 PowerON Reset
33   * Version      : Ver 1.00
34   * Device       : SH7763
35   * Tool-Chain   : High-performance Embedded Workshop (Version 4.05.01.001)
36   *              : C/C++ Compiler Package for SuperH Family (V.9.03 release00)
37   * OS           : None
38   * H/W Platform : MS7763SE02
39   * Description  : Sample Program for Setting the SH7763 Initialization
40   *              :
41   * Operation    :
42   * Limitation   :
43   *              :
44   ***********************************************************************
45   * History      : 28.July.2009 Ver. 1.00 First Release
46   *""FILE COMMENT END""*************************************************/
47
48   /*********************************************************************/
49   /*                                                                   */
50   /*  FILE        :resetprg.c                                          */
```

3.10 Sample Program Listing: "resetprg.c"(2)

```
51    /*  DATE        :Fri, Jul 24, 2009                                  */
52    /*  DESCRIPTION :Reset Program                                      */
53    /*  CPU TYPE    :SH7763                                             */
54    /*                                                                  */
55    /*  This file is generated by Renesas Project Generator (Ver.4.13).    */
56    /*                                                                  */
57    /******************************************************************/
58
59
60
61    #include <machine.h>
62    #include <_h_c_lib.h>
63    //#include   <stddef.h>               // Remove the comment when you use errno
64    //#include   <stdlib.h>               // Remove the comment when you use rand()
65    #include "typedefine.h"
66    #include "stacksct.h"
67    #include "cache.h"
68
69    #define SR_Init    0x40000000
70    #ifdef _FPD  // when -fpu=double is specified
71    #define FPSCR_Init 0x000C0001
72    #else
73    #define FPSCR_Init 0x00040001
74    #endif
75    #define INT_OFFSET 0x100UL
76
77    #define RAMCR_ADDRESS       0xff000074
78    #define RAMCR_INIT_VALUE    0x00000200
79
80    #ifdef __cplusplus
81    extern "C" {
82    #endif
83    extern void INTHandlerPRG(void);
84    void PowerON_Reset(void);
85    void Manual_Reset(void);
86    void main(void);
87    #ifdef __cplusplus
88    }
89    #endif
90
91    //#ifdef __cplusplus             // Enable I/O in the application(both SIM I/O and hardware I/O)
92    //extern "C" {
93    //#endif
94    //extern void _INIT_IOLIB(void);
95    //extern void _CLOSEALL(void);
96    //#ifdef __cplusplus
97    //}
98    //#endif
99
100   //extern void srand(_UINT);     // Remove the comment when you use rand()
```

3.11 Sample Program Listing: "resetprg.c"(3)

```
101  //extern _SBYTE *_s1ptr;            // Remove the comment when you use strtok()
102
103  //#ifdef __cplusplus            // Use Hardware Setup
104  //extern "C" {
105  //#endif
106  //extern void HardwareSetup(void);
107  //#ifdef __cplusplus
108  //}
109  //#endif
110
111  //#ifdef __cplusplus        // Remove the comment when you use global class object
112  //extern "C" {                  // Sections C$INIT and C$END will be generated
113  //#endif
114  //extern void _CALL_INIT(void);
115  //extern void _CALL_END(void);
116  //#ifdef __cplusplus
117  //}
118  //#endif
119
120  /* = = = = Changing section name to ResetPRG = = = = */
121  #pragma section ResetPRG
122
123  #pragma entry PowerON_Reset
124  /*""FUNC COMMENT""*************************************************
125  * ID                    :
126  * Outline               : Function for CPU Initialization
127  * Include               :
128  * Declaration           : void PowerON_Reset(void)
129  * Description           : CPU initialization routine. Its address is registered in
130  *                       : the vector table entry for power-on reset exception handling.
131  *                       : This is the first function executed after a power-on reset.
132  *                       : Enable processing which has been commented out as required.
133  * Argument              : none
134  * Return Value          : none
135  * Calling Functions     :
136  *""FUNC COMMENT END""*********************************************/
137  void PowerON_Reset(void)
138  {
139      _UDWORD* ramcr_address;
140
141      set_vbr((void *)((_UINT)INTHandlerPRG - INT_OFFSET));
142
143      set_fpscr(FPSCR_Init);
144
145      /* = = = = Initialization of sections B and D = = = = */
146      _INITSCT();
147
148  //  _CALL_INIT();                 // Remove the comment when you use global class object
149
150  //  _INIT_IOLIB();                // Enable I/O in the application(both SIM I/O and hardware I/O)
```

3.12 Sample Program Listing: "resetprg.c"(4)

```
151
152   //  errno=0;                    // Remove the comment when you use errno
153   //  srand((_UINT)1);                // Remove the comment when you use rand()
154   //  _s1ptr=NULL;                // Remove the comment when you use strtok()
155
156   //  HardwareSetup();            // Use Hardware Setup
157
158       /* ==== Cache setting ==== */
159       cache_set_ccr(D_CACHE_I_ON | D_CACHE_O_ON );
160
161       ramcr_address = (_UDWORD*)RAMCR_ADDRESS;
162       *ramcr_address = RAMCR_INIT_VALUE;
163
164       /* ==== Setting the status register (privileged mode) ==== */
165       set_cr(SR_Init);
166
167       main();
168
169   //  _CLOSEALL();                // Close I/O in the application(both SIM I/O and hardware I/O)
170
171   //  _CALL_END();                // Remove the comment when you use global class object
172
173       sleep();
174   }
175
176   //#pragma entry Manual_Reset            // Remove the comment when you use Manual Reset
177   /*""FUNC COMMENT""************************************************
178   * ID                        :
179   * Outline                   : Manual reset processing
180   * Include                   :
181   * Declaration               : void Manual_Reset_PC (void)
182   * Description               : The address of this function is registered in the vector
183   *                           : table entry for manual reset exception handling.
184   *                           : No processing is defined in this sample program.
185   *                           : Add processing as required.
186   * Argument                  : none
187   * Return Value              : none
188   * Calling Functions         :
189   *""FUNC COMMENT END""*******************************************/
190   void Manual_Reset(void)
191   {
192   }
193   /* END of File */
```

3.13 Sample Program Listing: "cache.c"(1)

```
1    /***********************************************************************
2    * DISCLAIMER
3
4    * This software is supplied by Renesas Technology Corp. and is only
5    * intended for use with Renesas products. No other uses are authorized.
6
7    * This software is owned by Renesas Technology Corp. and is protected under
8    * all applicable laws, including copyright laws.
9
10   * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11   * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12   * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13   * PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14   * DISCLAIMED.
15
16   * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17   * TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18   * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19   * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20   * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21
22   * Renesas reserves the right, without notice, to make changes to this
23   * software and to discontinue the availability of this software.
24   * By using this software, you agree to the additional terms and
25   * conditions found by accessing the following link:
26   * http://www.renesas.com/disclaimer
27   ***********************************************************************/
28   /* Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved.         */
29   /*""FILE COMMENT""************ Technical reference data ***************
30   * System Name  : SH7763 Sample Program
31   * File Name    : cache.c
32   * Abstract     : Sample Program for Setting the SH7763 Cache
33   * Version      : Ver 1.00
34   * Device       : SH7763
35   * Tool-Chain   : High-performance Embedded Workshop (Version 4.05.01.001)
36   *              : C/C++ Compiler Package for SuperH Family (V.9.03 release00)
37   * OS           : None
38   * H/W Platform : MS7763SE02
39   * Description  : Sample Program for Setting the SH7763 Initialization
40   *              :
41   * Operation    :
42   * Limitation   :
43   *              :
44   ***********************************************************************
45   * History      : 28.July.2009 Ver. 1.00 First Release
46   *""FILE COMMENT END""*************************************************/
47   #include <machine.h>
48   #include "iodefine.h"
49   #include "cache.h"
50
```

3.14 Sample Program Listing: "cache.c"(2)

```
51    #pragma section nonCACHE /* Allocation to the CS0 cache disabled space*/
52    /*""FUNC COMMENT""*************************************************
53    * ID                    :
54    * Outline               : Cache Setting
55    * Include               :
56    * Declaration           : void cache_set_ccr(unsigned int i_mode)
57    * Description           : Cache registers are set.
58    *                       :
59    * Argument              : unsigned int i_mode
60    *                       : Following modes are set by logical OR.
61    *                       : D_CACHE_I_INVALID    : IC invalidation
62    *                       : D_CACHE_I_ON         : IC enabled
63    *                       : D_CACHE_O_INVALID    : OC invalidation
64    *                       : D_CACHE_O_ON         : OC enabled
65    *                       : D_CACHE_IO_ON        : IC/OC enabled
66    *                       : D_CACHE_O_WT         : Write-through mode
67    *                       : D_CACHE_OFF          : IC/OC disabled
68    * Return Value          : none
69    * Calling Functions     :
70    *""FUNC COMMENT END""*********************************************/
71    void cache_set_ccr(unsigned int i_mode)
72    {
73        /* ==== Setting the exception/interrupt block bit (BL) ==== */
74        set_cr(get_cr() | 0x10000000);
75
76        /* ==== Setting cache registers ==== */
77        CACHE.CCR.LONG = i_mode;
78
79        /* ==== Issuing the ICBI instruction to enable cache ==== */
80        icbi(0);
81
82        /* ==== Releasing the exception/interrupt block bit (BL) ==== */
83        set_cr(get_cr() & ~(0x10000000));
84
85    }
86
87    #pragma section
88    /*""FUNC COMMENT""*************************************************
89    * ID                    :
90    * Outline               : Processing for operand cache purge
91    * Include               :
92    * Declaration           : void  cache_Purge_OCBP
93    *                       :  (unsigned long *i_start,
94    *                       :   unsigned long *i_end)
95    * Description           : Operand cache purge is executed.
96    *                       :
97    *
98    * Argument              : unsigned long *i_pstart :
99    *                       :   Start address for purge
100   *                       : unsigned long *i_pend   :
```

3.15 Sample Program Listing: "cache.c"(3)

```
101   *                         :   End addres for purge
102   * Return Value            : none
103   * Calling Functions       :
104   *""FUNC COMMENT END""*****************************************/
105   void cache_Purge_OCBP(unsigned long *i_pstart, unsigned long *i_pend)
106   {
107       unsigned long addr_length;
108       unsigned long *pStart1;
109
110       addr_length = (unsigned long)i_pend - (unsigned long)i_pstart;
111
112       pStart1 = i_pstart +8;
113
114       /* If purge cache is bigger than 4 entry,then use OCBP_LOOP_1*/
115       if(addr_length > 32) /* OCBP_LOOP_1 */
116       {
117
118         do
119         {
120             ocbp(i_pstart);
121             i_pstart += 16;
122             ocbp(pStart1);
123             pStart1 += 16;
124
125         }while(pStart1 < (i_pend + 8));
126
127       }
128       else /* OCBP_LOOP_2 */
129       {
130
131         while(i_pstart < i_pend)
132         {
133             ocbp(i_pstart);
134             i_pstart = i_pstart + 8;
135         }
136       }
137
138   }
139
140   /* End of File */
```

3.16 Sample Program Listing: "cache.h"(1)

```
1     /***********************************************************************
2     * DISCLAIMER
3
4     * This software is supplied by Renesas Technology Corp. and is only
5     * intended for use with Renesas products. No other uses are authorized.
6
7     * This software is owned by Renesas Technology Corp. and is protected under
8     * all applicable laws, including copyright laws.
9
10    * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11    * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12    * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13    * PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14    * DISCLAIMED.
15
16    * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17    * TECHNOLOGY CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18    * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19    * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
20    * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21
22    * Renesas reserves the right, without notice, to make changes to this
23    * software and to discontinue the availability of this software.
24    * By using this software, you agree to the additional terms and
25    * conditions found by accessing the following link:
26    * http://www.renesas.com/disclaimer
27    ***********************************************************************/
28    /* Copyright (C) 2009. Renesas Technology Corp., All Rights Reserved.         */
29    /*""FILE COMMENT""************ Technical reference data ***************
30    * System Name  : SH7763 Sample Program
31    * File Name    : cache.h
32    * Abstract     : Sample Program for Setting the SH7763 Cache
33    * Version      : Ver 1.00
34    * Device       : SH7763
35    * Tool-Chain   : High-performance Embedded Workshop (Version 4.05.01.001)
36    *              : C/C++ Compiler Package for SuperH Family (V.9.03 release00)
37    * OS           : None
38    * H/W Platform : MS7763SE02
39    * Description  : Sample Program for Setting the SH7763 Initialization
40    *              :
41    * Operation    :
42    * Limitation   :
43    *              :
44    ***********************************************************************
45    * History      : 28.July.2009 Ver. 1.00 First Release
46    *""FILE COMMENT END""*************************************************/
47
48    #ifndef  __CACHE_DEF_H__
49    #define __CACHE_DEF_H__
50
```

3.17 Sample Program Listing: "cache.h"(2)

```
51    /* ==== Macro definition ==== */
52    /* ---- Cache setting ---- */
53    #define D_CACHE_OFF          0x0000u
54    #define D_CACHE_I_INVALID    0x0800u
55    #define D_CACHE_I_ON     0x0100u
56    #define D_CACHE_O_INVALID    0x0008u
57    #define D_CACHE_O_ON     0x0001u
58    #define D_CACHE_IO_ON        (CACHE_I_ON | CACHE_O_ON)
59    #define D_CACHE_O_WT     0x0002u
60
61    /* ---- Area to be purged ---- */
62    #define D_CACHE_PURGE_SDRAM_START   *( volatile unsigned long  *)(0x08000000)
63    #define D_CACHE_PURGE_SDRAM_END     *( volatile unsigned long  *)(0x0FFFFFFF)
64
65    /* ==== Function declaration ==== */
66    void cache_set_ccr(unsigned int i_mode);
67    void cache_Purge_OCBP(unsigned long *i_start, unsigned long *i_end);
68
69    #endif /* __CACHE_DEF_H__ */
```

4. Documents for Reference

- Software Manual
  SH-4A Software Manual (REJ09B0003)
  (The most up-to-date versions of the documents are available on the Renesas Technology Website.)

- Hardware Manual
  SH7763 Group Hardware Manual (REJ09B0256)
  (The most up-to-date versions of the documents are available on the Renesas Technology Website.)

- Application Note
  SuperH RISC engine C/C++ Compiler Package Application notes: [Introduction guide] Sample file Guide for SH-3, SH-4, and SH-4A (REJ06J0012)
  (The most up-to-date versions of the documents are available on the Renesas Technology Website.)

- Development Tool Manual
  Flash Memory Download Program for the E10A-USB Emulator Application Note (REJ10J1221)
  (The most up-to-date versions of the documents are available on the Renesas Technology Website.)

  SuperH RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package V.9.01 User's Manual (REJ10J1571)
  (The most up-to-date versions of the documents are 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	Dec.03.09	—	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.