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.

SH7780 Group

SH7780 Initialization Example

Introduction

This application note shows examples of settings required when the SH7780 is booted up.

Target Device

SH7780 (Solution Engine MS7780SE03 by Hitachi ULSI Systems)

Contents

1. Preface 2
2. Description of the Sample Application 4
3. List of Sample Programs 11
4. Documents of Reference 33

1. Preface

1.1 Specification

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

1.2 Description of Modules Used

- Local bus state controller (LBSC)
- DDR-SDRAM interface (DDRIF)
- Cache

1.3 Applicable Conditions

•	Microcomputer:	SH7780	
•	Operating frequency:	Internal clock	400 MHz
		SuperHighway clock	200 MHz
		Peripheral clock	33 MHz
		DDR clock	160 MHz
		External bus clock	33 MHz
		PCI bus clock	33 MHz
•	Area 0 bus width:	32 bits (MODE3 pin = High, MODE4 pin = High)	
•	Clock operating mode:	Mode 3 (MODE7 = Low, MODE2 = Low, MODE1 = High, MODE0 = High)	
•	Data alignment:	Little endian	
•	Address mode:	29-bit	
•	C compiler:	Renesas Technology product SuperH RISC engine family C/C++ compiler package Ver. 9.1.0	
•	Section locations:	The locations of the sections for this sample application are described in table 1.	

Table 1 Section Locations

Section	Used As	Address (Virtual Address)	
P	Program area (default)	0x00003000	P0 area (cacheable area with translatable addresses)
C	Constant area		
C$BSEC	Address structure for non-initialized data areas		
C$DSEC	Address structure for initialized data areas		
D	Initialized data (initial value)		
B	Non-initialized data area	0x08000000	
R	Initialized data area		
S	Stack area	0x0FFFF9F0	
INTHandler	Exception/interrupt handler	0x80000800	P1 area (cacheable area with untranslatable addresses)
VECTTBL	Reset vector table or interrupt vector table		
INTTBL	Interrupt mask table		
PIntPRG	Interrupt function		
SP_S	Stack area for the TLB-miss handler	0x8FFFFDF0	
RSTHandler	Reset handler	0xA0000000	P2 area (uncacheable area with untranslatable addresses)
PResetPRG	Reset program		
PnonCache	Program area (cache access invalid)		

2. Description of the Sample Application

2.1 Sample Programs

This initialization program consists of the following eight source files:

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

(1) dbsct.c is a file for initializing the sections. It is automatically created by the High-performance Embedded Workshop. The addresses where the initialized data sections (D and R) and non-initialized data section (B) start and end are defined in this file. Calling of the _INITSCT function by the PowerON_Reset function in resetprg.c clears the B section to 0 and copies the D section to the R section.

When a program is run in RAM using the ROM support function, the address ranges to be transferred should be specified in this file so that the relevant sections are copied by the _INITSCT function.

(2) The interrupt programs which are called by the exception and interrupt handler are written in.intprg.c.

These programs are dummy functions which are void of content for exception sources, external interrupts and peripheral function interrupts. When a peripheral function interrupt is used as in the case where the program for another application note is operating the dummy function should be commented out to create a new function (the vect.inc and vecttbl.src definitions should be changed), or else the program execution should jump from within the dummy function to a separately created function.

Procedures such as the clearing of the interrupt request flags should be written in accord with the precautions given in the hardware manual.

(3) main.c is the main function which is called on completion of intialization. User programs should be written within the main routine. In this program we assume that when a peripheral module is used, the main function calls the function to make settings for operation. It differs in this respect from the source programs automatically created by the High-performance Embdded Workshop (in which case the settings for operation are made in hwsetup.c).

Settings specified at the beginning of the main function are: the exception/interrupt block bits which determine whether or not interrupts are accepted; the processing mode bit which indicates the processing mode; and the interrupt mask level which determines the priority level of interrupts for acceptance. All of these are included in the status register (SR).

(4) the PowerON_Reset function which is registered in the reset vector is written in resetprg.c.

Unlike the file automatically created by the High-performance Embedded Workshop, the resetprg.c used in this program increases the processing speed by executing both the _INITSCT function for section copying and the Purge_Ocbp function for cache writeback while the cache is enabled. In the automatically created file, RAMCR is set before exit from the PowerON_Reset function, however, this setting is not made in this program. Thus, if the on-chip memory is accessed after the processing mode has been changed to the user mode, the access protection feature for the on-chip memory will cause an address error exception. To change to the user mode, set RAMCR (i.e., set the RMD bit to 1).

The PowerON_Reset function is the first to branch from the reset handler, the code it contains makes the initial settings for the cache and initializes the sections. After calling the _INITSCT function, the program code writes back the operand cache. The main function is called once this cache operation is finished.

(5) The reset vector table, exception/interrupt vector table and interrupt mask table are described in cvecttbl.src.

(6) Code for the reset handler, exception/interrupt handler and processing to initialize LBSC and DDRIF is written in vhandler.src The vhandler.src in this program differs from the file that is automatically created by the High-performance Embedded Workshop in that it includes code for invalidating the instruction and operand caches, making the FPSCR settings, initializing the LBSC and DDRIF furthermore the handler for TLB misses has been changed. Since the transfer mode for floating-point operation is 32 bits, change the initial value of FPSCR to suit software specifications.

The TLB miss handler automatically created by the High-performance Embedded Workshop shares a stack area with exception handlers in general and with interrupt handlers other than the TLB miss handler, as well as with all programs other than interrupt and exception handlers. Thus, when the stack area is in the P0 or P3 area (where the addresses are translatable) and the TLB miss handler places data on the stack, a further TLB-miss exception occurs and leads to a power-on reset. The TLB miss handler for this program creates a stack area (with a size of H'200), specially for handling a TLB misses in the P1 area where the addresses are non-translatable. This dedicated stack area remains in use until exit from TLB miss handler, preventing further TLB-miss exceptions during execution of the TLB miss handler.

The stack pointer which indicates addresses on the stack is specified at the beginning of the PowerON_Reset function by a #pragma entry directive. In this sample application, the stack area is placed in DDR-SDRAM which requires initialization. To prevent access to the stack before DDR-SDRAM initialization, LBSC and DDRIF are initialized before the branch to the PowerON_Reset function at the beginning of the reset handler.

Modify each handler in this file accordingly if the interrupt mode switching bit in the CPU operation mode register (CPUOPM) is in use and SR.IMASK for the received interrupt level is automatically set or if multiple-interrupt processing is in use.

(7) 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)

(8) vect.inc declares symbols from external reference to each of the interrupt programs called from the exception and interrupt handlers defined in inteprg.c.

This file need not be changed when the dummy functions in intprg.c are used to describe interrupt processing.

If you create separate interrupt functions by commenting out the dummy functions, change this file so that the function names match.

The sequence of operations from power-on reset to the jump to main function main is outlined in figure 1.

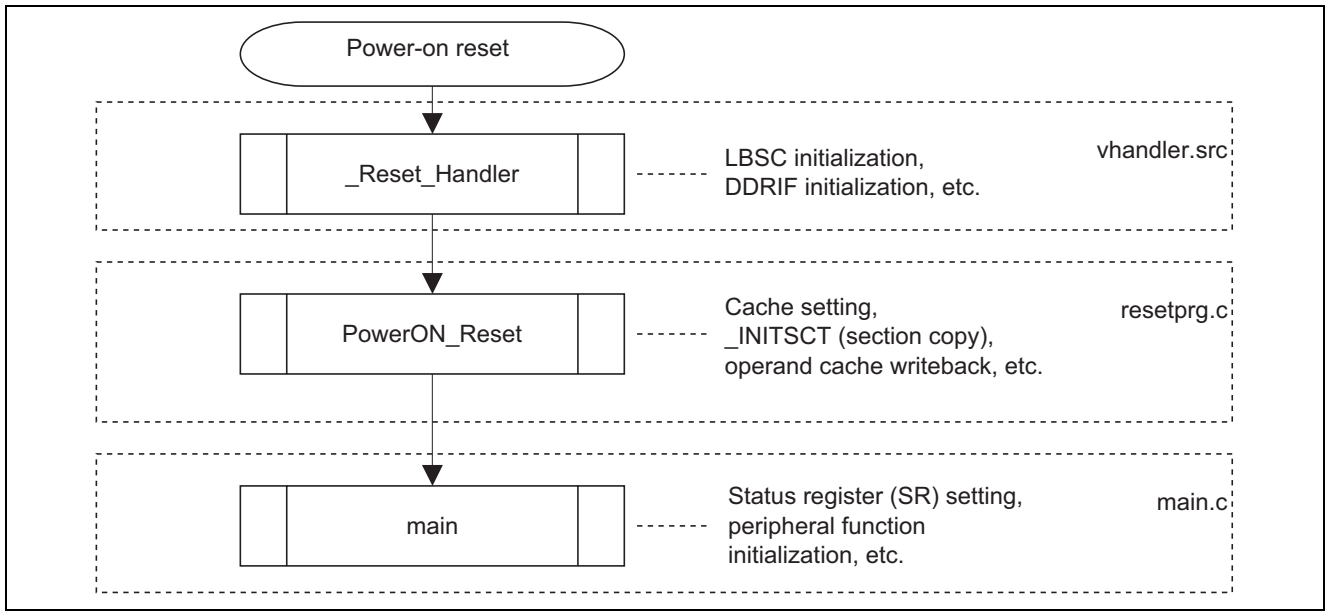

Figure 1 Overview of the Sequence of Operations from Power-on Reset to the Jump to Function main

The DDR-SDRAM initialization sequence is shown in Figure 2.

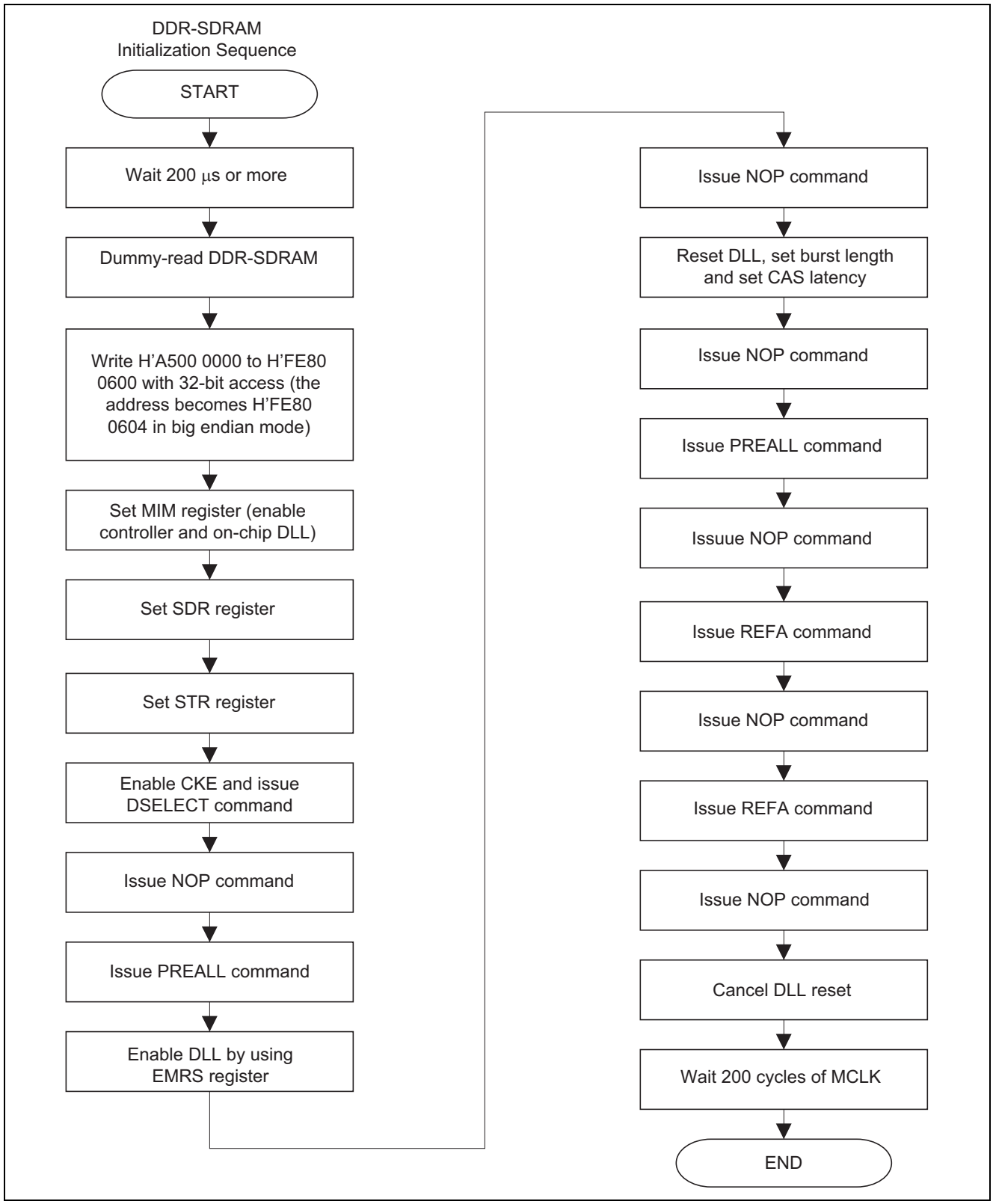

Figure 2 DDR-SDRAM Initialization Sequence

The sequence of operations with the PowerON_Reset function is shown in figure 3.

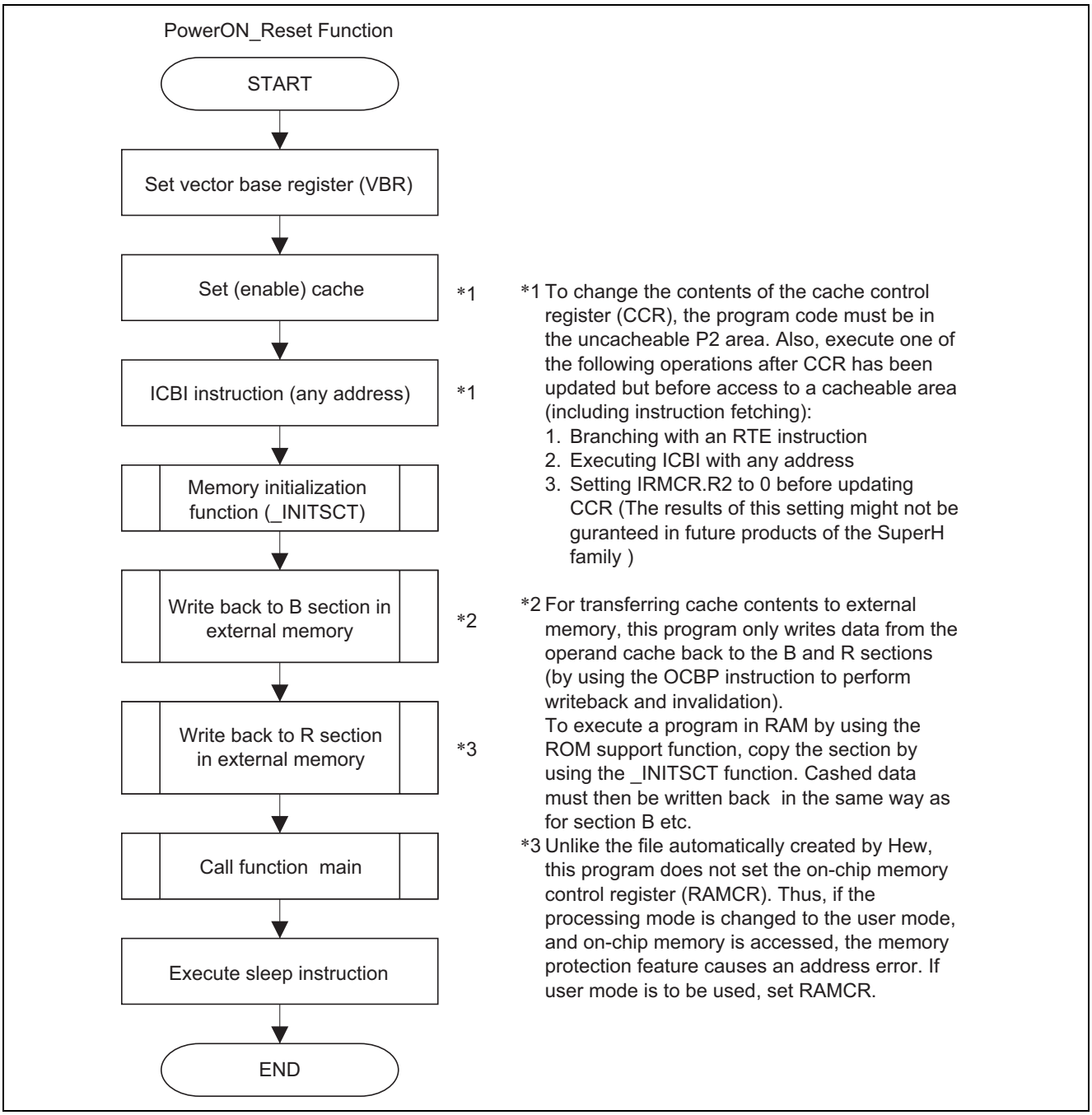

Figure 3 Operations with the PowerON_Reset Function

2.2 Settings for the Sample Programs

Table 2 Settings for the Sample Programs

Module	Settings
LBSC	CS0 space: Flash memory (SRAM) CS2 and CS3 space: DDR-SDRAM memory CS4 space: PCI bus
DDRIF	Access mode: BANK open mode Refresh mode: Auto-refresh Minimum number of cycles for transition from write to read: 3 Minimum number of cycles for transition from read to write: 4 Number of cycles within the same bank: 12 Number of PRE/PALL command issue cycles (tWR): 3 Number of ACT command issue cycles between banks: 2 Minimum number of ACT-PRE command issue cycles: 7 Number of auto-refresh/ACT command issue cycles (tRC): 10 CAS latency (CL): 2.5 Number of RAS-CAS command issue cycles (tRCD): 3 Number of PRE-ACT command issue cycles (tRP): 3 DRAM refresh interval bit: 1240 counts DDR-SDRAM memory configuration specification bit: Row 13 bits × Column 10 bits (32 M × 16 bits product)
Cache	Instruction and operand caches valid P1 area: Write-through mode P0, U0 and P3 areas: Copyback mode

2.3 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 local bus state controller and DDR-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 M 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).

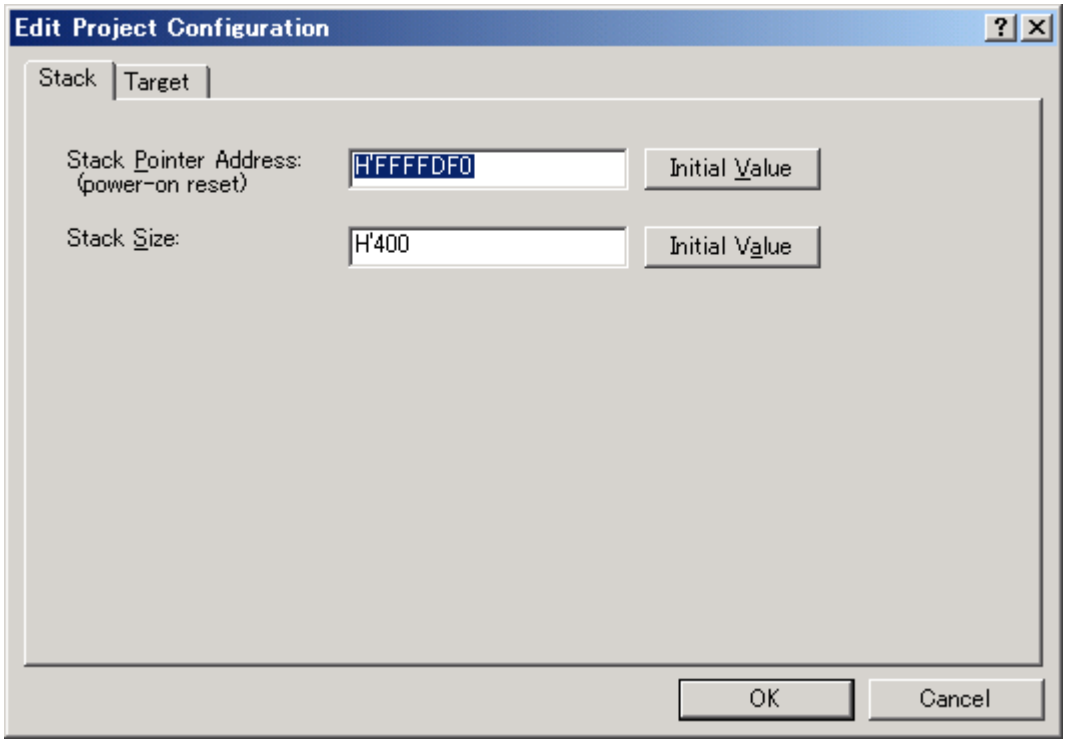

Figure 4 Edit Configuration

3. Listings of Sample Program files

1. Sample Program "dbst.c"

```
/*""FILE COMMENT""*********************************************************************
*     System Name :   SH7780 Sample Program
*     File Name   :   dbsct.c
*     Version     :   1.00.00
*     Contents    :   SH7780 Intialize Program
*     Model       :   Hitachi_ULSI_Systems SolutionEngine MS7780SE03
*     CPU         :   SH7780
*     Compiler    :   SHC.9.1.00
*     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) 2007 Renesas Technology Corp. All Rights Reserved
*
*     History     :   2007/12/26  ver 1.00.00
*
*********************************************************************************/
#pragma section $DSEC
static const struct {
    unsigned char *rom_s;       /* ROM top address of Initialize data section */
    unsigned char *rom_e;       /* ROM end address of Initialize data section */
    unsigned char *ram_s;       /* RAM top address of Initialize data section */
}   DTBL[] = {
    { __sectop("D"), __secend("D"), __sectop("R") }
};

#pragma section $BSEC
static const struct {
    unsigned char *b_s;         /* top address of no Initialize data section */
    unsigned char *b_e;         /* end address of no Initialize data section */
}   BTBL[] = {
    { __sectop("B"), __secend("B") }
};
```

2. Sample Program File "vhandler.src" (1)

```
;*""FILE COMMENT""*********************************************************************
;     System Name :   SH7780 Sample Program
;     File Name   :   vhandler.src
;     Version     :   1.00.00
;     Contents    :   SH7780 Intialize Program
;     Model       :   Hitachi_ULSI_Systems SolutionEngine MS7780SE03
;     CPU         :   SH7780
;     Compiler    :   SHC.9.1.00
;     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) 2007 Renesas Technology Corp. All Rights Reserved
;
;     History     :   2007/12/26 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

```

3. Sample Program File "vhandler.src" (2)

```
57                          .macro POP_EXP_BASE_REG
58              LDC.L       @R15+,R0_BANK
59              LDC.L       @R15+,R1_BANK
60              LDC.L       @R15+,R2_BANK
61              LDC.L       @R15+,R3_BANK
62              LDC.L       @R15+,R4_BANK
63              LDC.L       @R15+,R5_BANK
64              LDC.L       @R15+,R6_BANK
65              LDC.L       @R15+,R7_BANK
66              LDS.L       @R15+,FPSCR
67              LDS.L       @R15+,PR
68              LDC.L       @R15+,SPC
69              LDC.L       @R15+,SSR
70                          .endm
71
72  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
73  ;           Reset_Handler
74  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
75
76              .section    RSTHandler,code
77  _Reset_Handler:
78
79              MOV.L       #H'FF00001C,R0          ;set CCR address
80              MOV.L       #H'00000808,R1          ;IC,OC Invalidate
81              MOV.L       R1,@R0
82
83              MOV.L       #H'00040001,R0          ;set single precision mode
84  ;           MOV.L       #H'000C0001,R0          ;set double precision mode
85              LDS.L       R0,FPSCR
86
87              MOV.L       #LBSC_INIT,R0
88              JMP         @R0
89              NOP
90  LBSC_INIT_END:
91              MOV.L       #DDRIF_INIT,R0
92              JMP         @R0
93              NOP
94  DDRIF_INIT_END:
95
96              MOV.L       #EXPEVT,R0              ;set event address
97              MOV.L       @R0,R0
98              SHLR2       R0
99              SHLR        R0
100             MOV.L       #_RESET_VECTORS,R1
101             MOV.L       @(R0,R1),R0             ;set Reset function address
102             JMP         @R0
103             NOP
104
105 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
106 ;           exceptional interrupt
107 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
108             .section    INTHandler,code
109             .export     _INTHandlerPRG
110
111 _INTHandlerPRG:
112             PUSH_EXP_BASE_REG
```

4. Sample Program File "vhandler.src" (3)

```
113
114             MOV.L       #EXPEVT,R0              ;set event address
115             MOV.L       @R0,R1                  ;set exception code
116
117             MOV.L       #_INT_VECTORS,R0        ;set vector table address
118             ADD         #-(H'40),R1             ;exception code - h'40
119             SHLR2       R1
120             SHLR        R1
121             MOV.L       @(R0,R1),R3             ;set interrupt function address
122
123             LDC.L       R3,SPC
124             MOV.L       #__INT_TERM,R0          ;set interrupt terminate
125             LDS.L       R0,PR
126
127             RTE
128             NOP
129
130             .pool
131
132 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
133 ;           TLB miss interrupt
134 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
135             .org        H'300
136 _TLBmissHandler:
137
138             MOV.L       #(SP_STACK+H'200),R15  ;set SP_STACK(for only TLBmiss) pointer
139             STC.L       SGR,@-R15
140
141             PUSH_EXP_BASE_REG
142
143             MOV.L       #EXPEVT,R0              ;set event address
144             MOV.L       @R0,R1                  ;set exception code
145             MOV.L       #_INT_VECTORS,R0        ;set vectors table address
146             ADD         #-(h'40),R1             ;exception code - h'40
147             SHLR2       R1
148             SHLR        R1
149             MOV.L       @(R0,R1),R3             ;set interrupt function
150
151             MOV.L       #_INT_MASK,R0           ;interrupt mask table address
152             SHLR2       R1
153             MOV.B       @(R0,R1),R1             ;interrupt mask
154             EXTU.B      R1,R1
155
156             STC         SR,R0                   ;save SR
157             MOV.L       #(RBclr&IMASKclr),R2    ;RB,mask clear data(BL="1")
158             AND         R2,R0                   ;clear mask data
159             OR          R1,R0                   ;set interrupt mask
160             LDC         R0,SSR                  ;set current status
161
162             LDC.L       R3,SPC
163             MOV.L       #__TLBMISS_INT_TERM,R0 ;set interrupt terminate
164             LDS.L       R0,PR
165
166             RTE
167             NOP
168
```

5. Sample Program File "vhandler.src" (4)

```
169             .align  4
170
171 __TLBMISS_INT_TERM:
172             MOV.L       #MDRBBLset,R0          ;set MD,BL,RB
173             LDC.L       R0,SR
174
175             POP_EXP_BASE_REG
176
177             LDC.L       @R15+,SGR
178             STC.L       SGR,R15
179             RTE
180             NOP
181
182             .pool
183
184 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
185 ;           IRQ
186 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
187
188             .org        H'500
189 _IRQ_Handler:
190             PUSH_EXP_BASE_REG
191
192             MOV.L       #INTEVT,R0             ;set event address
193             MOV.L       @R0,R1                 ;set exception code
194
195             MOV.L       #_INT_VECTORS,R0       ;set vector table address
196             ADD         #-(H'40),R1            ;exception code - h'40
197             SHLR2       R1
198             SHLR        R1
199             MOV.L       @(R0,R1),R3            ;set interrupt function address
200
201             MOV.L       #_INT_MASK,R0          ;interrupt mask table address
202             SHLR2       R1
203             MOV.B       @(R0,R1),R1            ;interrupt mask
204             EXTU.B      R1,R1
205
206             STC         SR,R0                  ;save SR
207             MOV.L       #(RBBLclr&IMASKclr),R2 ;RB,BL,mask clear data
208             AND         R2,R0                  ;clear mask data
209             OR          R1,R0                  ;set interrupt mask
210             LDC         R0,SSR                 ;set current status
211
212             LDC.L       R3,SPC
213             MOV.L       #__INT_TERM,R0         ;set interrupt terminate
214             LDS.L       R0,PR
215
216             RTE
217             NOP
218
219             .align  4
220 __INT_TERM:
221             MOV.L       #MDRBBLset,R0          ;set MD,BL,RB
222             LDC.L       R0,SR
223             POP_EXP_BASE_REG
224             RTE
```

6. Sample Program File "vhandler.src" (5)

```
225             NOP
226
227             .pool
228
229 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
230 ;           LBSC INIT
231 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
232 LBSC_INIT:
233             MOV.L       #H'FF400020,R0          ;set MMSELR address
234             MOV.L       #H'A5A50003,R1          ;set CS2,3 DDRIF CS4 PCIC
235             MOV.L       R1,@R0
236             MOV.L       @R0,R2
237             MOV.L       @R0,R2
238             SYNCO
239
240             MOV.L       #H'FF802000,R0          ;set CS0BCR address
241             MOV.L       #H'00000000,R1          ;set for FLASHROM(spansion MS29J964H70TFI000D)
242             MOV.L       R1,@R0
243
244             MOV.L       #H'FF802060,R0          ;set CS6BCR address
245             MOV.L       #H'77777670,R1          ;set for Peripheral Device Control
246             MOV.L       R1,@R0
247
248             MOV.L       #H'FF802008,R0          ;set CS0WCR address
249             MOV.L       #H'00000002,R1
250             MOV.L       R1,@R0
251
252             MOV.L       #H'FF802068,R0          ;set CS6WCR address
253             MOV.L       #H'7777770f,R1
254             MOV.L       R1,@R0
255
256             MOV.L       #LBSC_INIT_END,R0
257             JMP         @R0
258             NOP
259
260             .pool
261
262 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
263 ;           DDRIF INIT
264 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
265
266 DDRIF_INIT:
267             MOV.L       #H'00007d00,R0
268 LOOP1:
269             DT          R0
270             BF          LOOP1                   ;200µs wait
271             NOP
272             NOP
273
274             MOV.L       #H'AC000000,R0          ;set dummy read access
275             MOV.L       @R0,R1
276
277 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
278 ;Big Endian
279 ;           MOV.L       #H'FE800604,R0
```

7. Sample Program File "vhandler.src" (6)

```
;            MOV.L      #H'A5000000,R1
;            MOV.L      R1,@R0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;Little Endian
             MOV.L      #H'FE800600,R0
             MOV.L      #H'A5000000,R1
             MOV.L      R1,@R0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;BANK CLOSE MODE
;            MOV.L      #H'FE800008,R0
;            MOV.L      #H'00004000,R1
;            MOV.L      R1,@R0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

             MOV.L      #H'FE80000C,R0        ;set MIM(31-0bit) address
             MOV.L      #H'04D80109,R1        ;DRAM refresh disable,DLL enable,DDRIF enable
             MOV.L      R1,@R0
             MOV.L      #H'04D80309,R1        ;DRAM refresh enable
             MOV.L      R1,@R0

             MOV.L      #H'FE800034,R2        ;set SDR(31-0bit) address
             MOV.L      #H'00000400,R1        ;32M*16bit
             MOV.L      R1,@R2

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;DDR320
             MOV.L      #H'FE80001C,R0        ;set STR(31-0bit) address
             MOV.L      #H'00013180,R1
             MOV.L      R1,@R0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;DDR266
;            MOV.L      #H'FE80001C,R0        ;set STR(31-0bit) address
;            MOV.L      #H'00010040,R1
;            MOV.L      R1,@R0
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

             MOV.L      #H'FE800014,R2        ;set SCR(31-0bit) address
             MOV.L      #H'00000003,R1
             MOV.L      R1,@R2                ;SCR CKE enable
             MOV.L      #H'00000001,R1
             MOV.L      R1,@R2                ;SCR NOP
             MOV.L      #H'00000002,R1
             MOV.L      R1,@R2                ;SCR PREALL

             MOV.L      #H'FEC02000,R3        ;EMRS DLL enable
             MOV.L      #H'00000000,R4
             MOV.L      R4,@R3
             MOV.L      #H'00000001,R1        ;SCR NOP
             MOV.L      R1,@R2

             MOV.L      #H'FEC00B08,R3        ;MRS DLL reset,CAS Latency=2.5,burstlength=2
             MOV.L      R4,@R3

             MOV.L      R1,@R2                ;SCR NOP
             MOV.L      #H'00000002,R1
             MOV.L      R1,@R2                ;SCR PREALL
```

8. Sample Program File "vhandler.src" (7)

```
336             MOV.L       #H'00000001,R1
337             MOV.L       R1,@R2                  ;SCR NOP
338             MOV.L       #H'00000004,R1
339             MOV.L       R1,@R2                  ;SCR REFA
340             MOV.L       #H'00000001,R1
341             MOV.L       R1,@R2                  ;SCR NOP
335             MOV.L       #H'00000004,R1
336             MOV.L       R1,@R2                  ;SCR REFA
337             MOV.L       #H'00000001,R1
338             MOV.L       R1,@R2                  ;SCR NOP
339             MOV.L       #H'FEC00308,R3
340             MOV.L       R4,@R3                  ;MRS Reset Cancel
341
342             MOV.L       #H'00001000,R0
343 LOOP2:
344             DT          R0
345             BF          LOOP2                   ;more then 200 MCLK wait
346             NOP
347             NOP
348
349             MOV.L       #DDRIF_INIT_END,R0
350             JMP         @R0
351             NOP
352
353             .pool
354
355 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
356 ;           SPECIAL STACK(for TLBmiss Handler)
357 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
358             .section    SP_S,data
359 SP_STACK:
360             .datab.b    H'200,H'00
361
362             .END
```

9. Sample Program File "resetprg.c" (1)

```
1   /*""FILE COMMENT""*****************************************************************
2   *     System Name :   SH7780 Sample Program
3   *     File Name   :   resetprg.c
4   *     Version     :   1.00.00
5   *     Contents    :   SH7780 Initialize Program
6   *     Model       :   Hitachi_ULSI_Systems SolutionEngine MS7780SE03
7   *     CPU         :   SH7780
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) 2007 Renesas Technology Corp. All Rights Reserved
18  *
19  *     History     :   2007/12/26  ver 1.00.00
20  *
21  ******************************************************************************/
22  #include <machine.h>
23  #include <_h_c_lib.h>
24  #include "stacksct.h"
25
26  /* --- Function Definition(internal) --- */
27  void PowerON_Reset(void);
28  void Manual_Reset(void);
29
30  static void Purge_OCBP(unsigned long *,unsigned long *);
31
32  /* --- Function Definition(external) --- */
33  extern void INTHandlerPRG(void);
34
35  /* --- Symbol Definition --- */
36  #define      INT_OFFSET     (0x100)
37  #define      DDR_TOP_ADR    ((void *)0x08000000)
38  #define      CCR            (*(volatile unsigned int    *)0xFF00001C) /* CCR Address*/
39
40  #pragma section ResetPRG
41
42  /***** Function Comment ***************************************
43  * Outline    : PowerON_Reset
44  *-------------------------------------------------------------
45  * Declaration    : void PowerON_Reset(void)
46  *-------------------------------------------------------------
47  * Functional description:
48  *              Call Hardware Initialize Function and
49  *              Call main Function
50  *-------------------------------------------------------------
51  * Return Value   : -
52  * Argument       : -
53  *-------------------------------------------------------------
54  * Input          : -
55  * Output     : -
56  *-------------------------------------------------------------
```

10. Sample Program File "resetprg.c" (2)

```
57  * Notes          : -
58  ****** Function Comment End *********************************/
59  #pragma entry PowerON_Reset
60  void PowerON_Reset(void)
61  {
62    set_vbr((void *)((unsigned int)INTHandlerPRG - INT_OFFSET));    /* set VBR */
63
64    CCR = 0x0909;                 /* ICache and OCache disable */
65    CCR = 0x0101;                 /* ICache and OCache enable */
66    icbi(DDR_TOP_ADR);            /* cache disable (optional address) */
67
68    _INITSCT();
69
70    Purge_OCBP(__sectop("B"),__secend("B"));  /* Purge Bsection */
71    Purge_OCBP(__sectop("R"),__secend("R"));  /* Purge Rsection */
72
73    main();
74    sleep();
75  }
76
77  /***** Function Comment ***************************************
78  * Outline    : Manual_Reset
79  *-----------------------------------------------------------------
80  * Declaration    : void Manual_Reset(void)
81  *-----------------------------------------------------------------
82  * Functional description:
83  *
84  *
85  *-----------------------------------------------------------------
86  * Return Value   : -
87  * Argument       : -
88  *-----------------------------------------------------------------
89  * Input          : -
90  * Output     : -
91  *-----------------------------------------------------------------
92  * Notes          : -
93  ****** Function Comment End *********************************/
94  void Manual_Reset(void)
95  {
96  }
97
```

11. Sample Program File "resetprg.c" (3)

```
#pragma section
/***** Function Comment ***************************************
* Outline    : Purge_OCBP
*-------------------------------------------------------------
* Declaration    : static void Purge_OCBP()
*-------------------------------------------------------------
* Functional description:
*              WriteBack Operand Cache and
*              Purge Operand Cache
*-------------------------------------------------------------
* Return Value   : -
* Argument       : Purge section Start Address : *start
*            : Purge section End Address       : *end
*-------------------------------------------------------------
* Input          : -
* Output     : -
*-------------------------------------------------------------
* Notes          : -
****** Function Comment End *********************************/
static void Purge_OCBP(unsigned long *start, unsigned long *end)
{
  long    addr_length;
  unsigned long *start1;

  addr_length = (unsigned long)end - (unsigned long)start;

  start1 = start +8;

  /* If purge cache is bigger than 4 entry,then use OCBP_LOOP_1*/
  if(addr_length > 32)          /* OCBP_LOOP_1 */
  {
      do{
          ocbp(start);
          start += 16;
          ocbp(start1);
          start1 += 16;
      }while(start1 < (end + 8));
  }
  else                          /* OCBP_LOOP_2 */
  {
      while(start < end)
      {
          ocbp(start);
          start = start + 8;
      }
  }
}
```

12. Sample Program File "vecttbl.src" (1)

```
1   ;*""FILE COMMENT""*********************************************************************
2   ;     System Name :   SH7780 Sample Program
3   ;     File Name   :   vecttbl.src
4   ;     Version     :   1.00.00
5   ;     Contents    :   SH7780 Initialize Program
6   ;     Model       :   Hitachi_ULSI_Systems SolutionEngine MS7780SE03
7   ;     CPU         :   SH7780
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) 2007 Renesas Technology Corp. All Rights Reserved
18  ;
19  ;     History     :   2007/12/26 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                ;H'020      Manual Reset
32                .data.l     _Manual_Reset
33                ;H'040-120  Reserved
34                .datab.l    8,H'00000000
35                ;H'140      TLB Reset(DATA TLB Reset)
36                .data.l     _TLB_Reset
37
38                .section    INTTBL,data
39                .export     _INT_VECTORS
40
41  _INT_VECTORS:
42                ; H'040     TLB miss read error
43                .data.l     _INT_TLB_MISS_READ
44                ; H'060     TLB miss write error
45                .data.l     _INT_TLB_MISS_WRITE
46                ; H'080     TLB 1stpage error
47                .data.l     _INT_TLB_1ST_PAGE
48                ; H'0A0     TLB protect write error
49                .data.l     _INT_TLB_PROTECT_WRITE
50                ; H'0C0     TLB protect read error
51                .data.l     _INT_TLB_PROTECT_READ
52                ; H'0E0     adr_error_read
53                .data.l     _INT_ADR_ERROR_READ
54                ; H'100     adr_error_write
55                .data.l     _INT_ADR_ERROR_WRITE
56                ; H'120     FPU error
```

13. Sample Program File "vecttbl.src" (2)

```
57                .data.l     _INT_FPU_ERROR
58                ; H'140     TLB reset
59                .data.l     _TLB_Reset
60                ;H'160      TRAPA (EXP_TRAPA)
61                .data.l     _INT_TRAP
62                ;H'180      ILLEGAL_INST
63                .data.l     _INT_ILLEGAL_INST
64                ;H'1A0      ILLEGAL_SLOT
65                .data.l     _INT_ILLEGAL_SLOT
66  ;EXTERNAL INTERRUPT
67                ;H'1C0      NMI
68                .data.l     _INT_NMI
69                ;H'1E0      USER_BREAK
70                .data.l     _INT_USER_BREAK
71                ; H'200     IRQ15
72                .data.l     _INT_IRQ15
73                ; H'220     IRQ14
74                .data.l     _INT_IRQ14
75                ; H'240     IRQ13
76                .data.l     _INT_IRQ13
77                ; H'260     IRQ12
78                .data.l     _INT_IRQ12
79                ; H'280     IRQ11
80                .data.l     _INT_IRQ11
81                ; H'2A0     IRQ10
82                .data.l     _INT_IRQ10
83                ; H'2C0     IRQ9
84                .data.l     _INT_IRQ9
85                ; H'2E0     IRQ8
86                .data.l     _INT_IRQ8
87                ; H'300     IRQ7
88                .data.l     _INT_IRQ7
89                ; H'320     IRQ6
90                .data.l     _INT_IRQ6
91                ; H'340     IRQ5
92                .data.l     _INT_IRQ5
93                ; H'360     IRQ4
94                .data.l     _INT_IRQ4
95                ; H'380     IRQ3
96                .data.l     _INT_IRQ3
97                ; H'3A0     IRQ2
98                .data.l     _INT_IRQ2
99                ; H'3C0     IRQ1
100               .data.l     _INT_IRQ1
101               ;H'3E0-460  Reserved
102               .datab.l    5,H'00000000
103 ;RTC
104               ;H'480      RTC ATI
105               .data.l     _INT_RTC_ATI
106               ;H'4A0      RTC PRI
107               .data.l     _INT_RTC_PRI
108               ;H'4C0      RTC CUI
109               .data.l     _INT_RTC_CUI
110               ;H'4E0-540  Reserved
111               .datab.l    4,H'00000000
```

14. Sample Program File "vecttbl.src" (3)

```
112 ;WDT
113             ;H'560      ITI
114             .data.l     _INT_WDT_ITI
115 ;TMU-ch0
116             ;H'580      TMU_TUNI0
117             .data.l     _INT_TMU0_TUNI0
118 ;TMU-ch1
119             ;H'5A0      TMU_TUNI1
120             .data.l     _INT_TMU1_TUNI1
121 ;TMU-ch2
122             ;H'5C0      TMU_TUNI2
123             .data.l     _INT_TMU2_TNUI2
124             ;H'5E0      TMU_TICPI2
125             .data.l     _INT_TMU2_TICPI2
126 ;H-UDI
127             ;H'600      H-UDII
128             .data.l     _INT_H_UDII
129             ;H'620      Reserved
130             .data.l     H'00000000
131 ;DMAC(0)
132             ;H'640      DMINT0
133             .data.l     _INT_DMAC0_DMINT0
134             ;H'660      DMINT1
135             .data.l     _INT_DMAC0_DMINT1
136             ;H'680      DMINT2
137             .data.l     _INT_DMAC0_DMINT2
138             ;H'6A0      DMINT3
139             .data.l     _INT_DMAC0_DMINT3
140             ;H'6C0      DMAE
141             .data.l     _INT_DMAC0_DMAE
142             ;H'6E0      Reserved
143             .data.l     H'00000000
144 ;SCIF-ch0
145             ;H'700      ERI0
146             .data.l     _INT_SCIF0_ERI0
147             ;H'720      RXI0
148             .data.l     _INT_SCIF0_RXI0
149             ;H'740      BRI0
150             .data.l     _INT_SCIF0_BRI0
151             ;H'760      TXI0
152             .data.l     _INT_SCIF0_TXI0
153 ;DMAC(0)
154             ;H'780      DMINT4
155             .data.l     _INT_DMAC0_DMINT4
156             ;H'7A0      DMINT5
157             .data.l     _INT_DMAC0_DMINT5
158 ;DMAC(1)
159             ;H'7C0      DMINT6
160             .data.l     _INT_DMAC1_DMINT6
161             ;H'7E0      DMINT7
162             .data.l     _INT_DMAC1_DMINT7
163             ;H'800-8E0  Reserved
164             .datab.l    8,H'00000000
165 ;CMT
166             ;H'900      CMTI
167             .data.l     _INT_CMT_CMTI
```

15. Sample Program File "vecttbl.src" (4)

```
168              ;H'920-960  Reserved
169              .datab.l    3,H'00000000
170 ;HAC
171              ;H'980      HACI
172              .data.l     _INT_HAC_HACI
173              ;H'9A0-9E0  Reserved
174              .datab.l    3,H'00000000
175 ;PCIC(0)
176              ;H'A00      PCISERR
177              .data.l     _INT_PCIC_PCISERR
178 ;PCIC(1)
179              ;H'A20      PCIINTA
180              .data.l     _INT_PCIC_PCIINTA
181 ;PCIC(2)
182              ;H'A40      PCIINTB
183              .data.l     _INT_PCIC_PCIINTB
184 ;PCIC(3)
185              ;H'A60      PCIINTC
186              .data.l     _INT_PCIC_PCIINTC
187 ;PCIC(4)
188              ;H'A80      PCIINTD
189              .data.l     _INT_PCIC_PCIINTD
190 ;PCIC(5)
191              ;H'AA0      PCIERR
192              .data.l     _INT_PCIC_PCIINTERR
193              ;H'AC0      PCIPWD3
194              .data.l     _INT_PCIC_PCIPWD3
195              ;H'AE0      PCIPWD2
196              .data.l     _INT_PCIC_PCIPWD2
197              ;H'B00      PCIPWD1
198              .data.l     _INT_PCIC_PCIPWD1
199              ;H'B20      PCIPWD0
200              .data.l     _INT_PCIC_PCIPWD0
201              ;H'B40-B60  Reserved
202              .datab.l    2,H'00000000
203 ;SCIF-ch1
204              ;H'B80      ERI1
205              .data.l     _INT_SCIF1_ERI1
206              ;H'BA0      RXI1
207              .data.l     _INT_SCIF1_RXI1
208              ;H'BC0      BRI1
209              .data.l     _INT_SCIF1_BRI1
210              ;H'BE0      TXI1
211              .data.l     _INT_SCIF1_TXI1
212 ;SIOF
213              ;H'C00      SIOFI
214              .data.l     _INT_SIOF_SIOFI
215              ;H'C20-C60  Reserved
216              .datab.l    3,H'00000000
217 ;HSPI
218              ;H'C80      HSPI
219              .data.l     _INT_HSPI_SPII
220              ;H'CA0-CE0  Reserved
221              .datab.l    3,H'00000000
```

16. Sample Program File "vecttbl.src" (5)

```
222 ;MMCIF
223             ;H'D00     FSTAT
224             .data.l    _INT_MMCIF_FSTAT
225             ;H'D20     TRAN
226             .data.l    _INT_MMCIF_TRAN
227             ;H'D40     ERR
228             .data.l    _INT_MMCIF_ERR
229             ;H'D60     FRDY
230             .data.l    _INT_MMCIF_FRDY
231 ;DMAC(1)
232             ;H'D80     DMINT8
233             .data.l    _INT_DMAC1_DMINT8
234             ;H'DA0     DMINT9
235             .data.l    _INT_DMAC1_DMINT9
236             ;H'DC0     DMINT10
237             .data.l    _INT_DMAC1_DMINT10
238             ;H'DE0     DMINT11
239             .data.l    _INT_DMAC1_DMINT11
240 ;TMU-ch3
241             ;H'E00     TUNI3
242             .data.l    _INT_TMU3_TUNI3
243 ;TMU-ch4
244             ;H'E20     TUNI4
245             .data.l    _INT_TMU4_TUNI4
246 ;TMU-ch5
247             ;H'E40     TUNI5
248             .data.l    _INT_TMU5_TUNI5
249             ;H'E60     Reserved
250             .data.l    H'00000000
251 ;SSI
252             ;H'E80     SSII
253             .data.l    _INT_SSI_SSII
254             ;H'EA0-EE0 Reserved
255             .datab.l   3,H'00000000
256 ;FLCTL
257             ;H'F00     FLSTE
258             .data.l    _INT_FLCTL_FLSTE
259             ;H'F20     FLTEND
260             .data.l    _INT_FLCTL_FLTEND
261             ;H'F40     FLTRQ0
262             .data.l    _INT_FLCTL_FLTRQ0
263             ;H'F60     FLTRQ1
264             .data.l    _INT_FLCTL_FLTRQ1
265 ;GPIO
266             ;H'F80     GPIOI0
267             .data.l    _INT_GPIO_GPIOI0
268             ;H'FA0     GPIOI1
269             .data.l    _INT_GPIO_GPIOI1
270             ;H'FC0     GPIOI2
271             .data.l    _INT_GPIO_GPIOI2
272             ;H'FE0     GPIOI3
273             .data.l    _INT_GPIO_GPIOI3
274
275             .export     _INT_MASK
276 _INT_MASK:
277             ; interrupt priority mask revel(31~0)
```

17. Sample Program File "vecttbl.src" (6)

```
278
279              ;H'040-140  Reserved
280               .datab.b    9,H'00
281               ;H'160     TRAPA
282               .data.b     H'00
283               ;H'180     ILLEGAL_INST
284               .data.b     H'00
285               ;H'1A0     ILLEGAL_SLOT
286               .data.b     H'00
287 ;EXTERNAL INTERRUPT
288               ;H'1c0     NMI
289               .data.b     H'00
290               ;H'1E0     USER_BREAK
291               .data.b     H'00
292               ;H'200-3c0     IRL
293               .datab.b    15,H'00
294               ;H'3e0-460     reserve
295               .datab.b    5,H'00
296 ;RTC
297               ;H'480     RTC ATI
298               .data.b     H'00
299               ;H'4A0     RTC PRI
300               .data.b     H'00
301               ;H'4C0     RTC CUI
302               .data.b     H'00
303               ;H'4E0-560     reserve
304               .datab.b    5,H'00
305 ;TMU-ch0
306               ;H'580     TMU_TUNI0
307               .data.b     H'00
308 ;TMU-ch1
309               ;H'5A0     TMU_TUNI1
310               .data.b     H'00
311 ;TMU-ch2
312               ;H'5C0     TMU_TUNI2
313               .data.b     H'00
314               ;H'5E0     TMU_TICPI2
315                .data.b    H'00
316               ;H'600     _INT_H_UDII
317                .data.b    H'00
318               ;H'620     reserve
319                .data.b    H'00
320 ;DMAC(0)
321               ;H'640     DMINT0
322                .data.b    H'00
323               ;H'660     DMINT1
324                .data.b    H'00
325               ;H'680     DMINT2
326                .data.b    H'00
327               ;H'6A0     DMINT3
328                .data.b    H'00
329               ;H'6C0     DMAE
330                .data.b    H'00
331               ;H'6E0     reserve
332               .data.b    H'00
```

18. Sample Program File "vecttbl.src" (7)

```
333 ;SCIF-ch0
334               ;H'700     ERI0
335                .data.b    H'00
336               ;H'720     RXI0
337                .data.b    H'00
338               ;H'740     BRI0
339                .data.b    H'00
340               ;H'760     TXI0
341                .data.b    H'00
342 ;DMAC(0)
343               ;H'780     DMINT4
344                .data.b    H'00
345               ;H'7A0     DMINT5
346                .data.b    H'00
347 ;DMAC(1)
348               ;H'7C0     DMINT6
349                .data.b    H'00
350               ;H'7E0     DMINT7
351                .data.b    H'00
352               ;H'800-8E0     reserve
353               .datab.b   8,H'00
354 ;CMT
355               ;H'900     CMT
356               .data.b     H'00
357               ;H'920-960 reserve
358               .datab.b    3,H'00
359               ;H'980     HAC
360               .data.b     H'00
361               ;H'9A0-9E0     reserve
362               .datab.b    3,H'00
363 ;PCIC(0)
364               ;H'A00     PCISERR
365               .data.b     H'00
366 ;PCIC(1)
367               ;H'A20     PCIINTA
368               .data.b     H'00
369 ;PCIC(2)
370               ;H'A40     PCIINTB
371               .data.b     H'00
372 ;PCIC(3)
373               ;H'A60     PCIINTC
374               .data.b     H'00
375 ;PCIC(4)
376               ;H'A80     PCIINTD
377               .data.b     H'00
378 ;PCIC(5)
379               ;H'AA0     PCIERR
380               .data.b     H'00
381               ;H'AC0     PCIPWD3
382               .data.b     H'00
383               ;H'AE0     PCIPWD2
384               .data.b     H'00
385               ;H'B00     PCIPWD1
386               .data.b     H'00
387               ;H'B20     PCIPWD0
388               .data.b     H'00
```

19. Sample Program File "vecttbl.src" (8)

```
389              ;H'B40-B60     reserve
390              .datab.b    2,H'00
391 ;SCIF-ch1
392              ;H'B80     ERI1
393              .data.b     H'00
394              ;H'BA0     RXI1
395              .data.b     H'00
396              ;H'BC0     BRI1
397              .data.b     H'00
398              ;H'BE0     TXI1
399              .data.b     H'00
400 ;SIOF
401              ;H'C00     SIOFI
402              .data.b     H'00
403              ;H'C20-C60     reserve
404              .datab.b    3,H'00
405 ;HSPI
406              ;H'C80     HSPI
407              .data.b     H'00
408              ;H'CA0-CE0     reserve
409              .datab.b    3,H'00
410 ;MMCIF
411              ;H'D00     FSTAT
412              .data.b     H'00
413              ;H'D20     TRAN
414              .data.b     H'00
415              ;H'D40     ERR
416              .data.b     H'00
417              ;H'D60     FRDY
418              .data.b     H'00
419 ;DMAC(1)
420              ;H'D80     DMINT8
421              .data.b     H'00
422              ;H'DA0     DMINT9
423              .data.b     H'00
424              ;H'DC0     DMINT10
425              .data.b     H'00
426              ;H'DE0     DMINT11
427              .data.b     H'00
428 ;TMU-ch3
429              ;H'E00     TUNI3
430              .data.b     H'00
431 ;TMU-ch4
432              ;H'E20     TUNI4
433              .data.b     H'00
434 ;TMU-ch5
435              ;H'E40     TUNI5
436              .data.b     H'00
437              ;H'E60     reserve
438              .data.b     H'00
439 ;SSI
440              ;H'E80     SSII
441              .data.b     H'00
442              ;H'EA0-EE0     reserve
443              .datab.b    3,H'00
```

20. Sample Program File "vecttbl.src" (9)

```
444 ;FLCTL
445                 ;H'F00      FLSTE
446                 .data.b      H'00
447                 ;H'F20      FLTEND
448                 .data.b      H'00
449                 ;H'F40      FLTRQ0
450                 .data.b      H'00
451                 ;H'F60      FLTRQ1
452                 .data.b      H'00
453 ;GPIO
454                 ;H'F80      GPIOI0
455                 .data.b      H'00
456                 ;H'FA0      GPIOI1
457                 .data.b      H'00
458                 ;H'FC0      GPIOI2
459                 .data.b      H'00
460                 ;H'FE0      GPIOI3
461                 .data.b      H'00
462
463                .end
```

21. Sample Program File "intprg.c"

```
1   /*""FILE COMMENT""*********************************************************************
2 *   System Name :   SH7780 Sample Program
3 *   File Name   :   intprg.c
4 *   Version     :   1.00.00
5 *   Contents    :   SH7780 Initialize Program
6 *   Model       :   Hitachi_ULSI_Systems SolutionEngine MS7780SE03
7 *   CPU         :   SH7780
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) 2007 Renesas Technology Corp. All Rights Reserved
18  *
19  *     History     :   2007/12/26 ver 1.00.00
20  *
21  ***********************************************************************************/
22  #include <machine.h>
23
24  /* --- Function Definition(internal) --- */
25  static void int_responstime_wait(unsigned int wait_time);
26
27  /* --- Symbol Definition --- */
28  #define INTC_RESPONSEWAIT   (0x00000014)       /* INT response wait Pck 5cycle
29                                                 H'14 = (1/Pck*5cyc) / (1/Ick*3cyc) */
30
31  /* --- RAM allocation variable declaration --- */
32
33  #pragma section IntPRG
34  /* H'040 Data TLB miss exception(read) */
35  void INT_TLB_MISS_READ_EXP(void)
36  {
37  }
                Omitted
591 /* H'FC0 GPIO interrupt(port H0,H1,J0,K4) */
592 void INT_GPIO_GPIOI2(void)
593 {
594 }
595 /* H'FE0 GPIO interrupt(port K5,E6) */
596 void INT_GPIO_GPIOI3(void)
597 {
598 }
599
600 #pragma inline_asm(int_responstime_wait)
601 static void int_responstime_wait(unsigned int wait_time)
602 {
603 ?0001:
604             DT          R4
605             BF          ?0001
606             NOP
607   }
```

22. Sample Program File "main.c"

```
/*""FILE COMMENT""*********************************************************************
*   System Name :   SH7780 Sample Program
*   File Name   :   main.c
*   Version     :   1.00.00
*   Contents    :   SH7780 Initialize Program
*   Model       :   Hitachi_ULSI_Systems SolutionEngine MS7780SE03
*   CPU         :   SH7780
*   Compiler    :   SHC.9.1.00
*   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) 2007 Renesas Technology Corp. All Rights Reserved
*
*     History     :   2007/12/26 ver 1.00.00
*
*********************************************************************************************/
#include <machine.h>

/* --- Function Definition(internal) --- */

/* --- Symbol Definition --- */
#define  SR_Init        0x400000e0         /* Privileged mode, RB,BL=0, IMASK level 14 */

/* --- RAM allocation variable declaration --- */

/***** Function Comment ***************************************
* Outline    : main
*-------------------------------------------------------------
* Declaration    : void main(void)
*-------------------------------------------------------------
* Functional description:
*              main function
*-------------------------------------------------------------
* Return Value   : -
* Argument       : -
*-------------------------------------------------------------
* Input          : -
* Output     : -
*-------------------------------------------------------------
* Notes          : -
****** Function Comment End *********************************/
void main(void)
{
  set_cr(SR_Init);          /* Set SR "Privileged mode, RB,BL=0, IMASK level 14" */
                            /* Call from here on-chip module initialize function */
  }
```

4. Documents for Reference

- Hardware Manual
  SH7780 Hardware Manual
  The most up-to-date version of this document is available on the Renesas Technology Website.
- Software Manual
  SH-4A Software Manual
  The most up-to-date information 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	Feb.18.08	—	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.