SH7785 Group

SH7785 Initial Settings Sample Program

R01AN0242EJ0101
Rev.1.01
Mar 17, 2011

Introduction

This application note presents a sample program for setting the initialization items required at SH7785 startup.

Target Device

SH7785

Contents

1. Introduction

1.1 Specifications

After the power-on reset is cleared, the code presented in this application note initializes the local bus state controller (LBSC) and the DDR2 SDRAM interface (DBSC2).

1.2 Functions Used

- Local bus state controller
- DDR2 SDRAM interface (DBSC2)

1.3 Applicable Conditions

Evaluation board:	Renesas R0P7785LC0011RL	
	External memory (Area 0):	NOR flash memory: 64 MB
		Spansion S29GL256P90TFIRI
	(Areas 2 and 3):	DDR2 SDRAM: 128 MB
		(In 32-bit mode: 512 MB)
		Elpida EDE1108ACSE-6E-E (four devices)
Microcontroller:	SH7785	
Operating frequencies:	Internal clock: 600 MHz	
	SuperHyway clock: 300 MHz	
	Peripheral clock: 50 MHz	
	DDR2 clock: 300 MHz	
	External bus clock: 100 MHz	
	PCI bus clock: 33 MHz	
Area 0 bus width:	32 bits (MD5 pin = high, MD6 pin = high)	
Clock operating mode:	Clock mode 16 (MD0 = low, MD1 = low, MD2 = low, MD3 = low, MD4 = high)	
Endian mode:	Little endian (MD8 = high)	
Addressing mode:	29-bit addressing (MD13 = low)	
Tool chain:	Super-H RISC engine Standard Toolchain Version 9.3.2.0	
Compiler options:	Other than the options specified in the include file in the High-Performance Embedded Workshop, the default options are used.	
	-cpu=sh4a -endian=little -include="$(PROJDIR)\inc\drv","$(PROJDIR)\inc" -object="$(CONFIGDIR)\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo	
Assembler options:	-cpu=sh4a -endian=little -round=zero -denormalize=off -include="$(PROJDIR)\inc" -include="$(PROJDIR)\inc\drv" -debug -object="$(CONFIGDIR)\$(FILELEAF).obj" -literal=pool,branch,jump,return -nolist -nologo -chgincpath -errorpath	

Table 1 lists the section allocations used in this sample program.

Table 1 Section Allocations

Section	Section Usage	Area	Allocation Address (Virtual Address)	
P	Program area	ROM	0x00002000	P0 area (Can be cached, MMU address conversion possible)
C	Constant area	ROM		
C$BSEC	Uninitialized data area address structure	ROM		
C$DSEC	Initialized data area address structure	ROM		
D	Initialized data	ROM		
B	Uninitialized data area	RAM	0x0C000000	
R	Initialized data area	RAM		
S	Stack area	RAM	0x0DFF8000	
INTHandler	Exception/interrupt handler	ROM	0x80001000	P1 area (Can be cached, MMU address conversion not possible)
VECTTBL	Reset vector table Interrupt vector table	ROM		
INTTBL	Interrupt mask table	ROM		
PIntPRG	Interrupt function	ROM		
RSTHandler	Reset handler	ROM	0xA0000000	P2 area (Can not be cached, MMU address conversion not possible)
PResetPRG	Reset program	ROM		
PnonCACHE	Program area (Cache invalid access)	ROM		

2. Local Bus State Controller (LBSC)

The functions of the local bus state controller (LBSC) include dividing the external memory space (according to chip select signal settings) and outputting the control signals appropriate for the various types of memory or other devices connected.

2.1 Register Configuration

Table 2 lists the local bus state controller (LBSC) register configuration.

Table 2 LBSC Register Configuration

Register	Symbol	R/W	P4 Address	Access Size	Clock
Memory address map select register	MMSELR	R/W	H'FC40 0020	32	SHck
Bus control register	BCR	R/W	H'FF80 1000	32	Bck
CS0 bus control register	CS0BCR	R/W	H'FF80 2000	32	Bck
CS1 bus control register	CS1BCR	R/W	H'FF80 2010	32	Bck
CS2 bus control register	CS2BCR	R/W	H'FF80 2020	32	Bck
CS3 bus control register	CS3BCR	R/W	H'FF80 2030	32	Bck
CS4 bus control register	CS4BCR	R/W	H'FF80 2040	32	Bck
CS5 bus control register	CS5BCR	R/W	H'FF80 2050	32	Bck
CS6 bus control register	CS6BCR	R/W	H'FF80 2060	32	Bck
CS0 wait control register	CS0WCR	R/W	H'FF80 2008	32	Bck
CS1 wait control register	CS1WCR	R/W	H'FF80 2018	32	Bck
CS2 wait control register	CS2WCR	R/W	H'FF80 2028	32	Bck
CS3 wait control register	CS3WCR	R/W	H'FF80 2038	32	Bck
CS4 wait control register	CS4WCR	R/W	H'FF80 2048	32	Bck
CS5 wait control register	CS5WCR	R/W	H'FF80 2058	32	Bck
CS6 wait control register	CS6WCR	R/W	H'FF80 1068	32	Bck
CS5 PCMCIA control register	CS5PCR	R/W	H'FF80 1070	32	Bck
CS6 PCMCIA control register	CS6PCR	R/W	H'FF80 1080	32	Bck

2.2 The CnCBR Registers

The CnCBR registers set the idle cycles, bus widths, and memory types.

Figure 1 shows the timing chart for idle cycles in the same space and figure 2 shows the timing chart for idle cycles in different spaces. Table 3 lists the timing symbols and their descriptions.

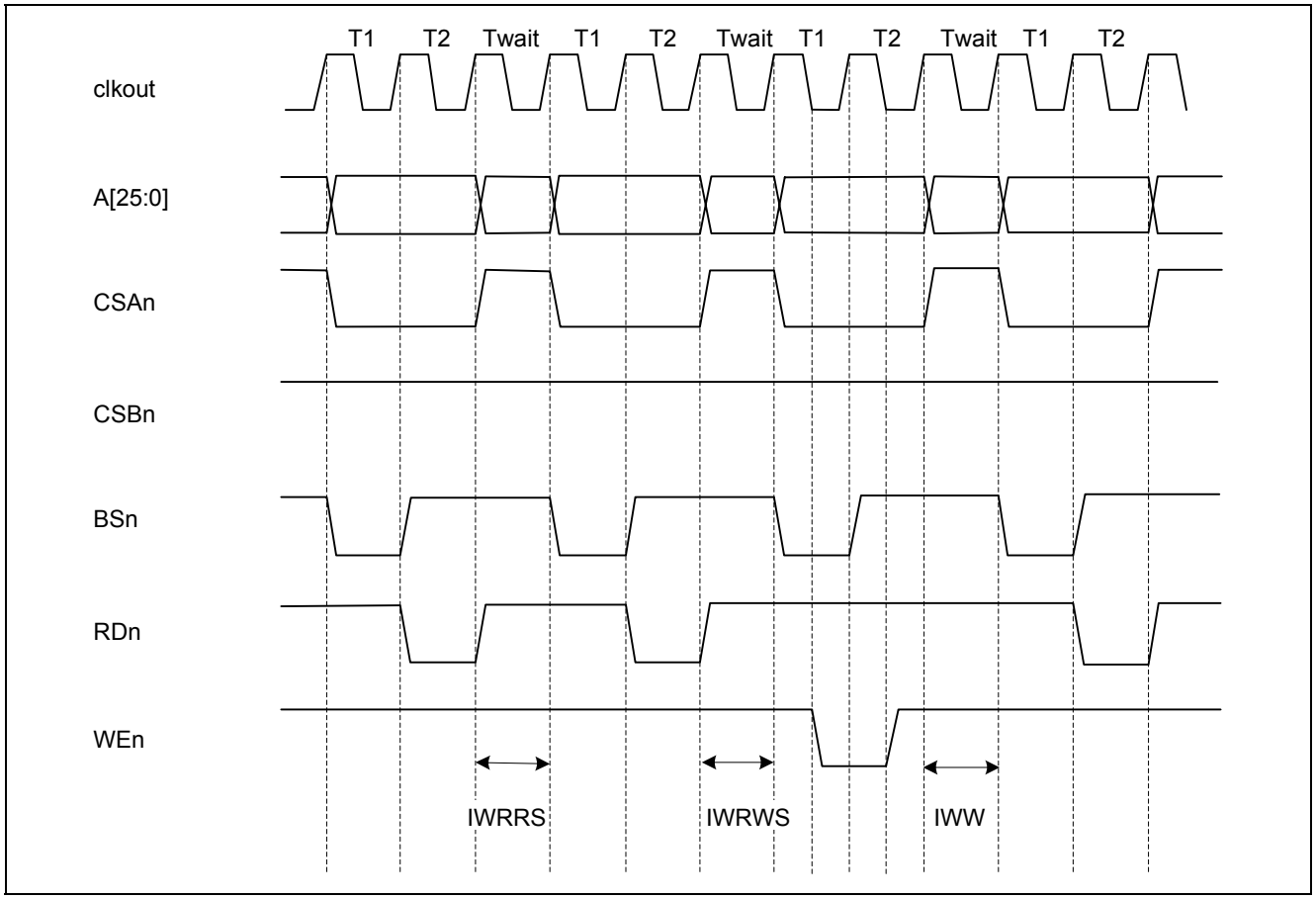


Figure 1 Idle Cycles in the Same Space

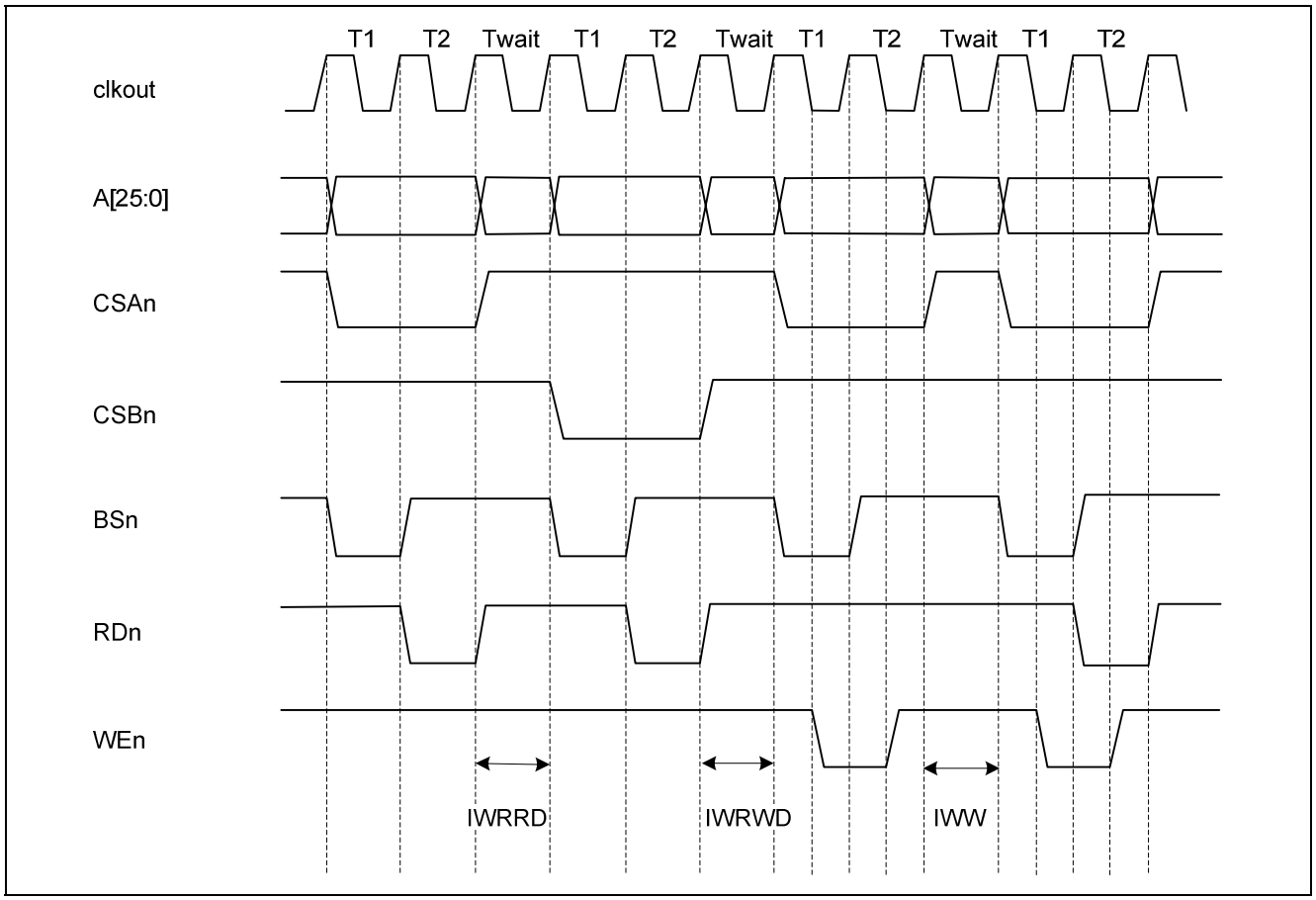


Figure 2 Idle Cycles in Different Spaces

Table 3 Idle Cycle Symbols

Symbol	Description
IWRRS	Idle cycles between read cycles in the same space
IWRWS	Idle cycles between read and write cycles in the same space
IWRRD	Idle cycles between read cycles in different spaces
IWRWD	Idle cycles between read and write cycles in different spaces
IWW	Idle cycles between write and read cycles, Idle cycles between write cycles

2.3 The CnWCR Registers

The CnWCR registers set the setup and hold times and wait cycles for access cycles.

The data output timing during writes differs with the ADS setting.

Figure 3 shows the access cycle timing chart.

Table 4 lists the access cycle symbols and their descriptions.

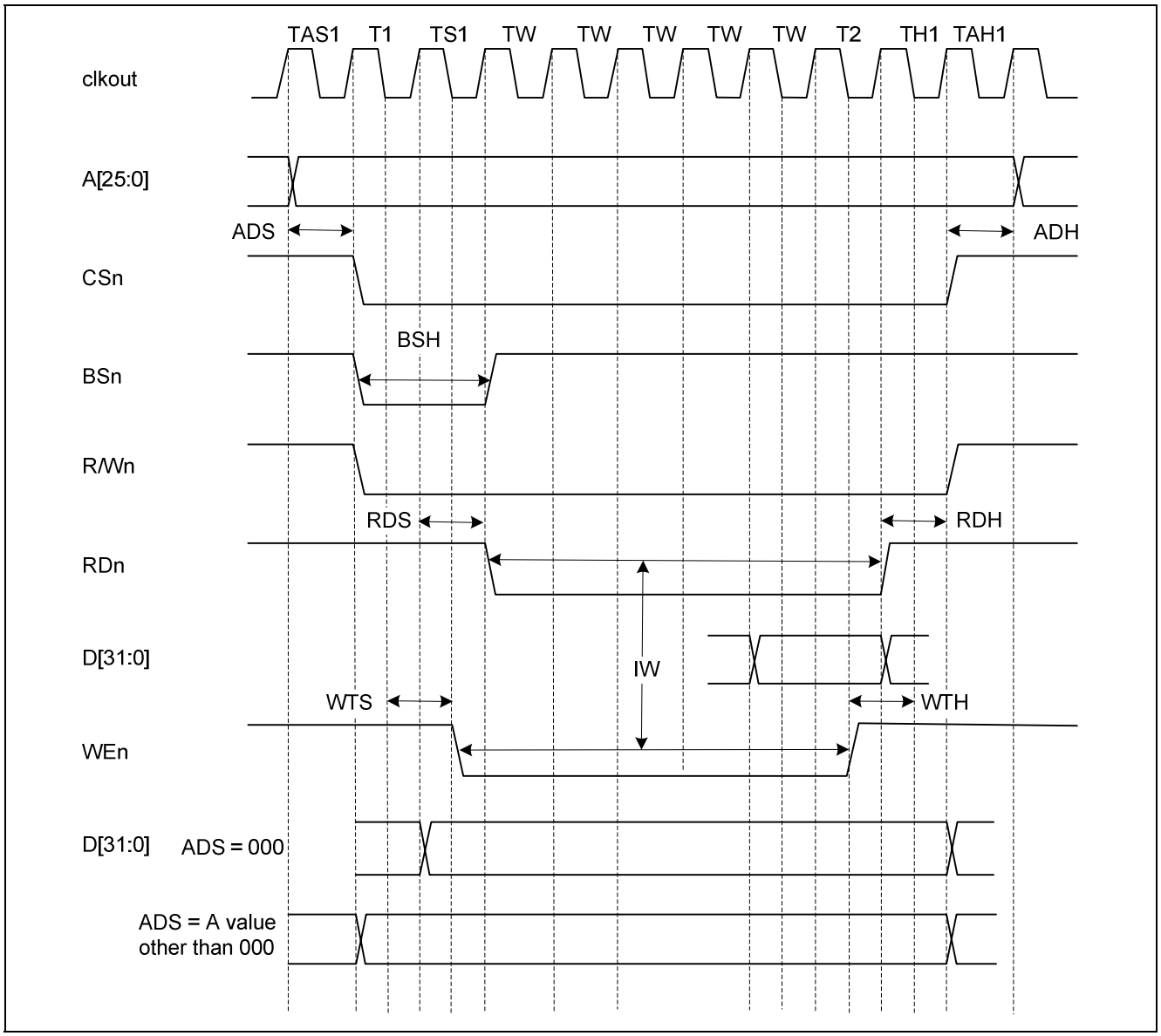


Figure 3 Access Cycle Timing Chart

Table 4 Access Cycle Symbols

Symbol	Description
ADS	Address assert to CS assert delay period
ADH	CS negate to address assert delay period
BSH	BS cycle count
RDS	CS assert to read assert delay period
RDH	Read negate to CS negate delay period
WTS	CS assert to write assert delay period
WTH	Write negate to CS negate delay period
IW	Number of wait cycles in the read and write periods

3. DDR2 SDRAM Interface (DBSC2)

The DDR2 SDRAM interface (DBSC2) arbitrates accesses from the CPU and various modules and outputs control signal required by DDR2 SDRAM allowing DDR2 SDRAM to be connected directly.

- DBSC2 supports 16 and 32-bit bus widths
- DBSC2 supports DDR2-600 (300 MHz) and DDR2-400 (200 MHz) devices
- A burst length of 4 is supported.
- Table 5 lists the memory capacities and configurations that can be used.

Table 5 DBSC2 Memory Capacities and Configurations

External Bus width	Memory Capacity	Configuration	Total Capacity
16 bits	256 Mbits (16 M × 16 bits)	Parallel connection (1 chip)	256 Mbits
	256 Mbits (32 M × 8 bits)	Parallel connection (2 chips)	512 Mbits
	512 Mbits (32 M × 16 bits)	Parallel connection (1 chip)	512 Mbits
	512 Mbits (64 M × 8 bits)	Parallel connection (2 chips)	1 Gbit
	1 Gbit (64 M × 16 bits)	Parallel connection (1 chip)	1 Gbit
	1 Gbit (128 M × 8 bits)	Parallel connection (2 chips)	2 Gbits
	2 Gbits (128 M × 16 bits)	Parallel connection (1 chip)	2 Gbits
	2 Gbits (256 M × 8 bits)	Parallel connection (2 chips)	4 Gbits
32 bits	256 Mbits (16 M × 16 bits)	Parallel connection (2 chips)	512 Mbits
	256 Mbits (32 M × 8 bits)	Parallel connection (4 chips)	1 Gbit
	512 Mbits (32 M × 16 bits)	Parallel connection (2 chips)	1 Gbit
	512 Mbits (64 M × 8 bits)	Parallel connection (4 chips)	2 Gbits
	1 Gbit (64 M × 16 bits)	Parallel connection (2 chips)	2 Gbits
	1 Gbit (128 M × 8 bits)	Parallel connection (4 chips)	4 Gbits
	2 Gbits (128 M × 16 bits)	Parallel connection (2 chips)	4 Gbits
	2 Gbits (256 M × 16 bits)	Parallel connection (4 chips)	8 Gbits

3.1 Register Configuration

Table 6 lists the DBSC2 registers.

Table 6 DBSC2 Registers

Register	Symbol	R/W	P4 Address	Access Size	Clock
DBSC2 status register	DBSTATE	R	H'FE80 000C	32	DDRck
SDRAM operation enable register	DBEN	R/W	H'FE80 0010	32	DDRck
SDRAM command control register	DBCMDCNT	R/W	H'FE80 0014	32	DDRck
SDRAM configuration setting register	DBCONF	R/W	H'FE80 0020	32	DDRck
SDRAM timing register 0	DBTR0	R/W	H'FE80 0030	32	DDRck
SDRAM timing register 1	DBTR1	R/W	H'FE80 0034	32	DDRck
SDRAM timing register 2	DBTR2	R/W	H'FE80 0038	32	DDRck
SDRAM refresh control register 0	DBRFCNT0	R/W	H'FE80 0040	32	DDRck
SDRAM refresh control register 1	DBRFCNT1	R/W	H'FE80 0044	32	DDRck
SDRAM refresh control register 2	DBRFCNT2	R/W	H'FE80 0048	32	DDRck
SDRAM refresh status register	DBRFSTS	R/W	H'FE80 004C	32	DDRck
DDRPAD frequency setting register	DBFREQ	R/W	H'FE80 0050	32	DDRck
DDRPAD DIC, DIC, ODT, and ODC setting register	DBDICODTOCD	R/W	H'FE80 0054	32	DDRck
SDRAM mode setting register	DBMRCNT	W	H'FE80 0060	32	DDRck

See section 12.4, Registers, in section 12, DDR2 SDRAM Interface (DBSC2), in the SH7785 Group Hardware Manual (REJ09B0261) for details on these registers.

3.2 Notes on ODT Control Signal Outputs to SDRAM

The following notes apply when outputting control signals from the DBSC2 to SDRAM.

- A minimum of 4 cycles of the DDR clock is required for CAS latency when outputting ODT control signals to SDRAM.
- When extending ODT by outputting the ODT control signals from one DDR clock cycle earlier according to the ODT_EARLY bit in the DBDICODTOCD register, a minimum of 5 cycles of the DDR clock is required for CAS latency. Furthermore, it is necessary to set the value of the RDWR bits in the DBTR2 register to a value 1 larger than that required by the SDRAM specifications.

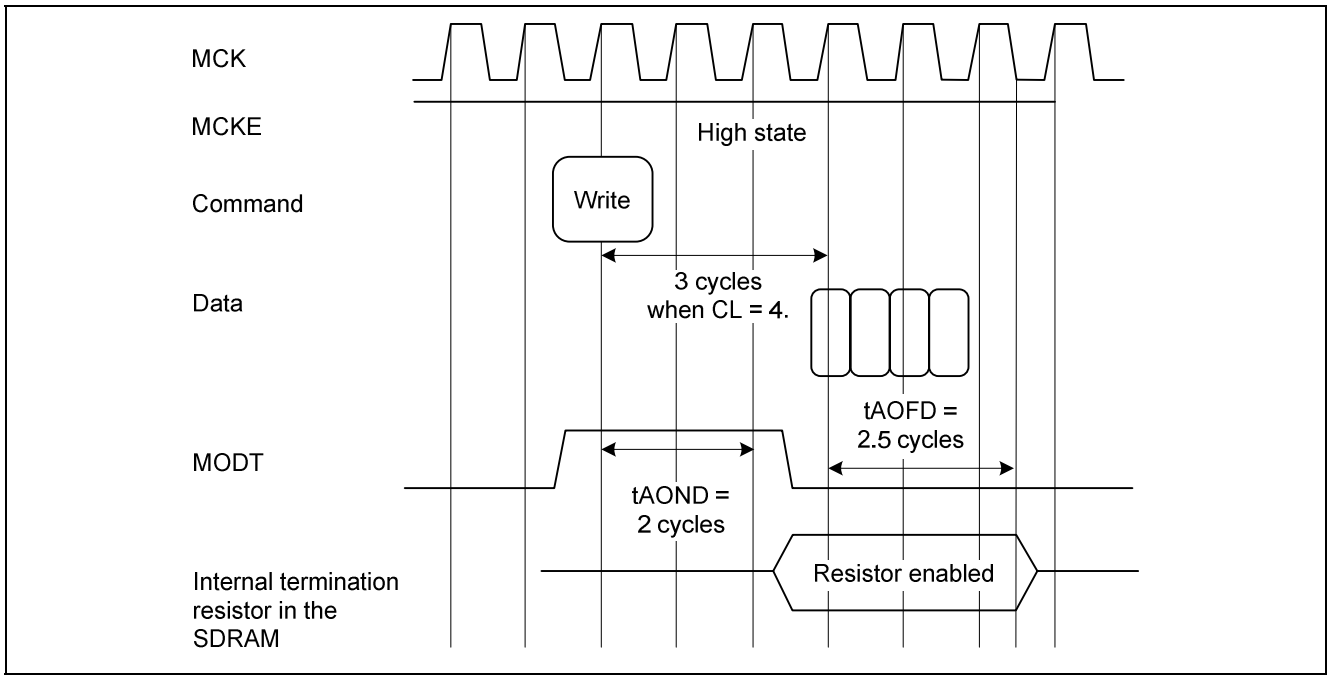


Figure 4 ODT Control Signal when CL is 4

If CL is 4 as in figure 4, assertion of a valid ODT control signal (MODT) to the SDRAM can start with the same timing as that of the write command. If CL is 5 or larger, the assert start occurs with a later timing than the write command. If CL is 3 or smaller, however, it will be necessary to star assertion of the ODT control signal with a timing earlier than that of the write command, which this controller does not support.

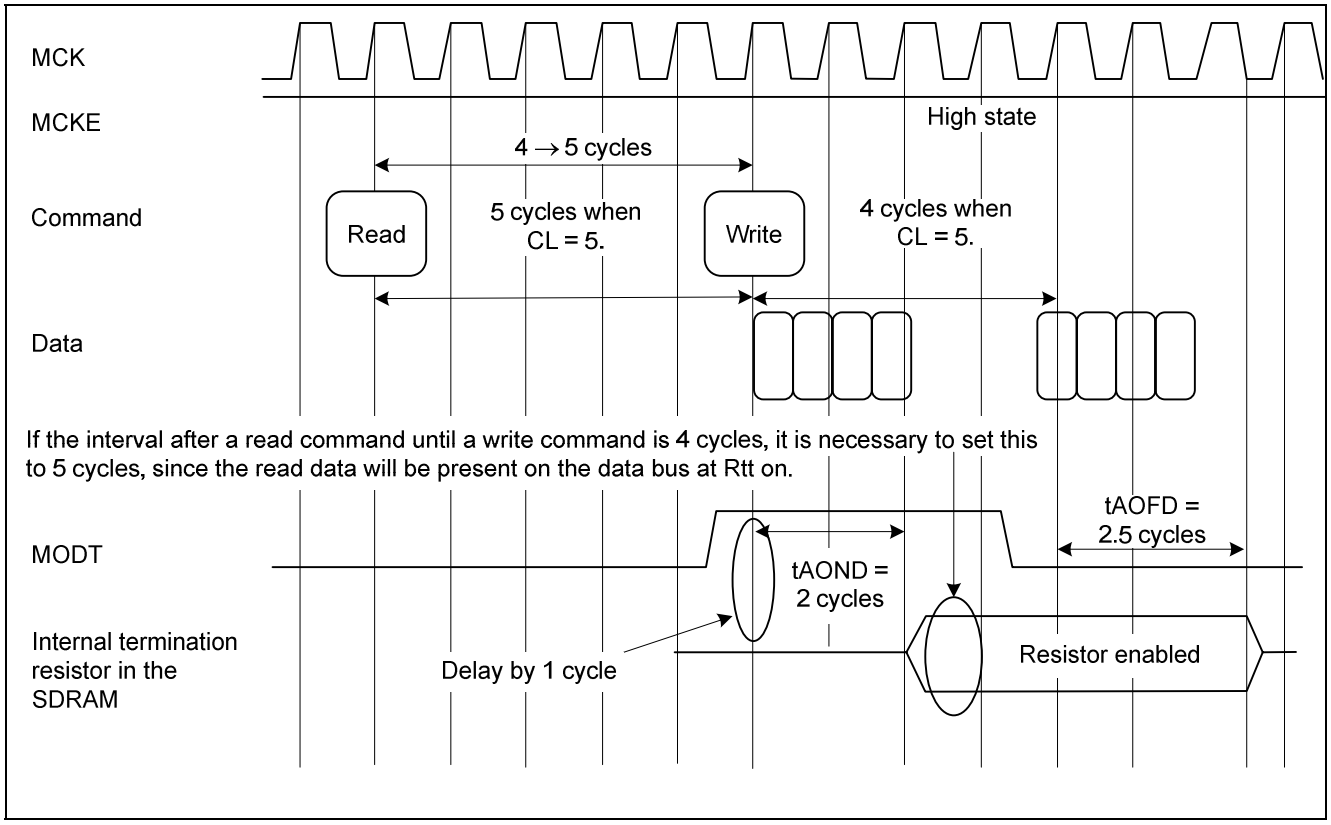


Figure 5 Notes on Using a One-Cycle Delay for the ODT Control Signal

Figure 5 shows an example when CL is 5. Since the ODT control signal (MODT) is delayed by 1 cycle, CL must be 5 or greater. If CL were to used set to 4 or smaller, it would be necessary to assert MODT before the write command, but this controller does not support that.

Here, we assume that the interval from a read to a write command required by the SDRAM is 4 cycles. However, since the ODT control signal (MODT) is delayed by 1 cycle, if this period remains as is at 4 cycles, there will be read data present on the data bus when the SDRAM internal terminating resistors are turned on. To avoid this, it is necessary to set the interval from a read to a write command to be 5 cycles.

4. Application Example

4.1 Overview of the R0P7785LC0011RL

4.1.1 Device List

Table 7 lists the devices on the R0P7785LC0011RL board used in this application example.

Table 7 R0P7785LC0011RL Peripheral Devices

Item	Description
Control PLD	Altera EPM240T100C5N • 8-bit bus access
USB2.0 controller	Renesas R8A66597FP#F0S • 16-bit bus access
SD controller	D-Broad CG200-V2 • 32-bit bus access
2D graphic controller	Silicon Motion SM107GX04LF0T-AA • 32-bit bus access
I2C controller	NXP Semiconductors PCA9564BS-T • 8-bit bus access

4.1.2 Memory Map

Table 8 lists the R0P7785LC0011RL memory map.

Table 8 R0P7785LC0011RL Memory Map

Area	Address	Connected Device	Bus Width
0	H'0000_0000 H'03FF_FFFF	S29GL256P90TFIRI (64 MB)	32 bits
1	H'0400_0000 H'05FF_FFFF	EPM240T100C5N (16 MB)	8 bits
	H'0600_0000 H'07FF_FFFF	PCA9564BS-T (16 MB)	
2	H'0800_0000 H'0BFF_FFFF	EDE1108ACSE-6E-E (128 MB)	32 bits
3	H'0C00_0000 H'0FFF_FFFF		
4	H'1000_0000 H'13FF_FFFF	SM107GX04LF0T-AA (64 MB)	32 bits
5	H'1400_0000 H'17FF_FFFF	R8A66597FP#F0S (64 MB)	16 bits
6	H'1800_0000 H'17FF_FFFF	CG200-V2 (64 MB)	32 bits

4.2 Sample Program Description

The following source program performs the required settings as an initialization program.

Tables, files, and other data for exceptions and interrupts may be added as required.

- vhandler.src
- resetprg.c
- dbsct.c
- vecttbl.src
- vect.inc
- intprg.c
- lowlevelinit.src
- lowlevelinit.inc

1. vhandler.src
   The exception handled in this file (vhandler.src) is run when and exception (reset, general exception, or interrupt) occurs.
   The exception handlers and jumps to the LBSC and DBSC2 initialization routines are coded in the vhandler.src file. The reset handler (_Reset_handler) is run after the power-on reset. The reset handler used in this application note differs from the one generated automatically by the High-performance Embedded Workshop in that LBSC and DBSC2 initialization processing has been added.

2. resetprg.c
   This file (resetprg.c) is based on the file generated automatically by the High-performance Embedded Workshop and includes the code for the PowerON_Reset() function registered in vecttbl.src.
   The PowerON_Reset() function is the first function that is branched to from the reset handler. It sets the VBR (vector base register), calls the _INITSCR() function, which copies sections, and calls the function that enables the cash. After that, it calls the main function. The status register (SR) selects privileged or user mode, specifies the general-purpose register bank, and manages exceptions and interrupts. This register must be set according to the design of the system.

3. dbsct.c
   This file (dbsct.c) codes copying information used to copy section data from ROM to RAM so that programs can be run from the connected external RAM. The _INITSCT() function copies programs immediately after a reset according to the information coded in this file. See the SuperH RISC Engine C/C++ Compiler Application Note (REJ05B0463) for more information.

4. vecttbl.src
   This file (vecttbl.src) codes registration for external handlers used when an exception (reset, general exception, or interrupt) occurs.

5. vect.inc
   This file (vect.inc) codes function to be registered in vecttbl.src.

6. intprg.c
   This file (intprg.c) interrupt functions used when an exception/interrupt occurs and are registered from the exception handler.

7. lowlevelinit.src
   This file (lowlevelinit.src) codes the LBSC and DBSC2 initial settings. The code in lowlevelinit.src is called from the reset handler (_Reset_handler).

8. lowlevelinit.inc
   This file (lowlevelinit.inc) codes setting values used in lowlevelinit.src.

4.3 Items Set by the Sample Program

The sample program sets the LBSC and DBSC2 so they can access the memory and peripheral devices connected to the R0P7785LC0011RL.

Table 9 lists the settings performed by the sample program.

Table 9 Sample Program Settings

Module	Settings
LBSC	CS0 space: Flash memory (memory type: SRAM), bus width: 32 bits CS1 space: PLD (memory type: SRAM), bus width: 8 bits CS2 and CS3 spaces: DDR2 SDRAM memory CS4 space: Graphic controller (memory type: SRAM), bus width: 32 bits CS5 space: USB controller (memory type: SRAM), bus width: 16 bits CS6 space: SD controller (memory type: SRAM), bus width: 32 bits
DBSC2	Model: EDE1108ACSE-6E-E Capacity (configuration): 1 Gbit (128 M × 8 bits) Devices used: 4 CAS latency: 5 Refresh cycle: 64 ms, average refresh time: 7.8125 µs Burst length: 4 or 8 (programmable) Row address: A13 to A0 Column address: A9 to A0 Precharge: Auto-precharge/all bank precharge controlled by A10 Driver strength: Normal/weak (programmable) ODT: 75Ω

4.4 Sample Program Flowchart

Figure 6 shows the processing flow from the power-on reset until the software jumps to the main() function.

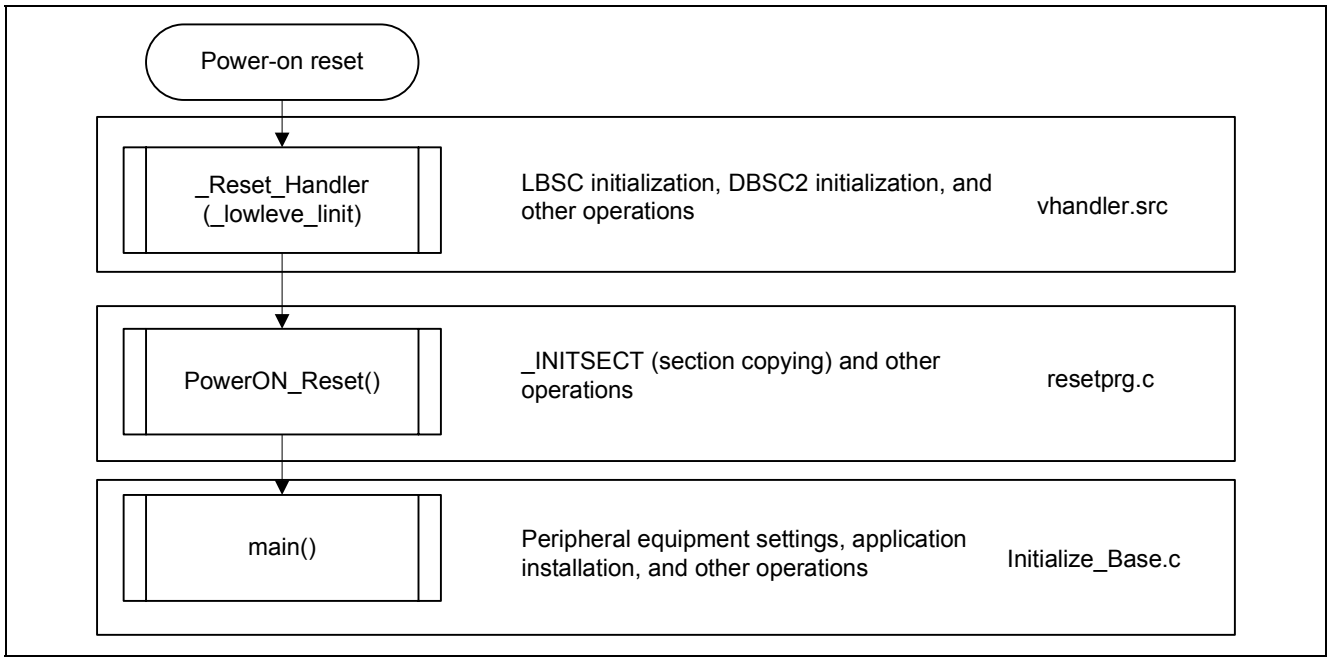


Figure 6 Flowchart from Power-on Reset to the main() Function

Figures 7 to 11 show the processing flow in the _lowlevel_init() function.

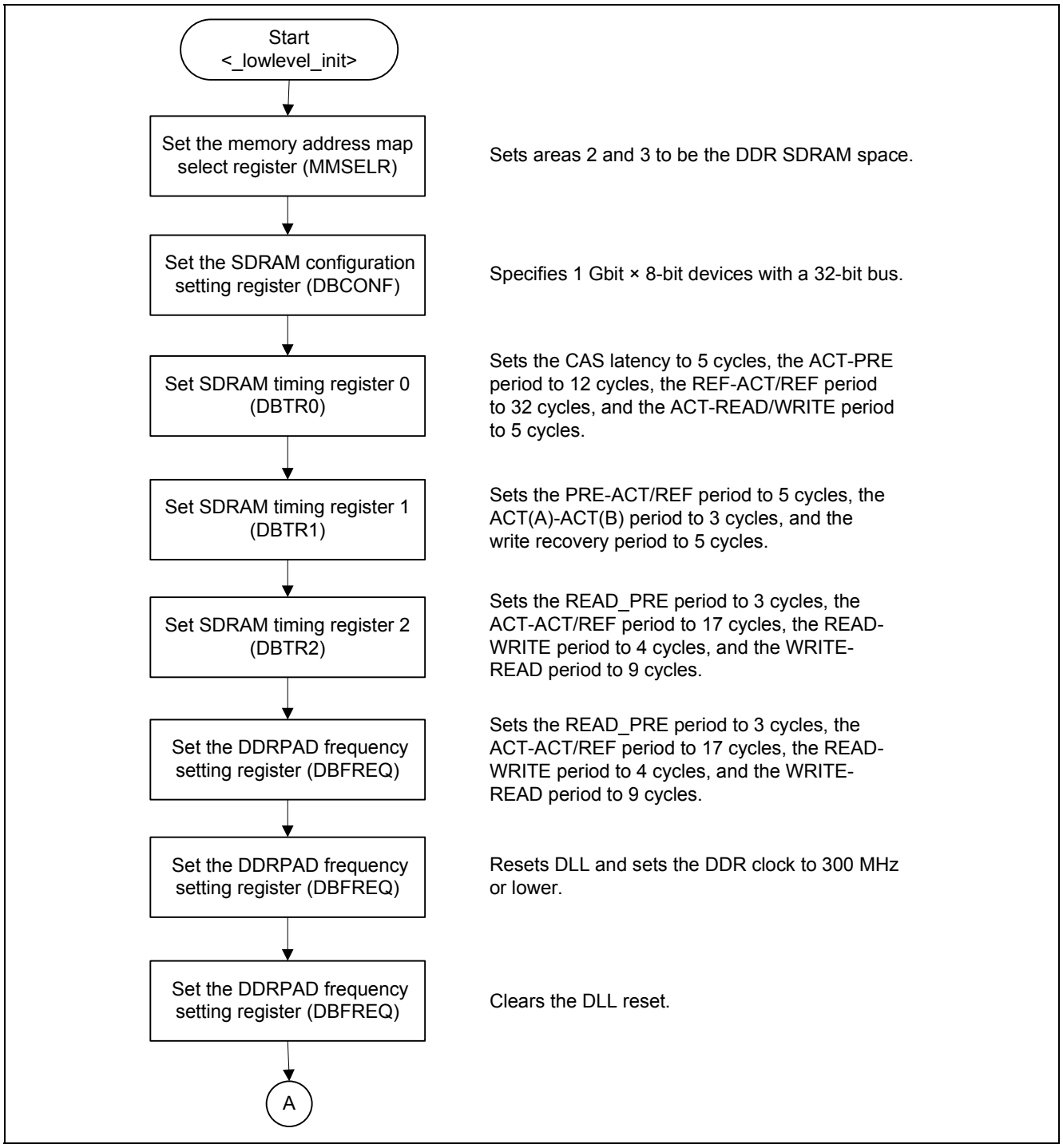


Figure 7 Flowchart for the _lowlevel_init() Function 1

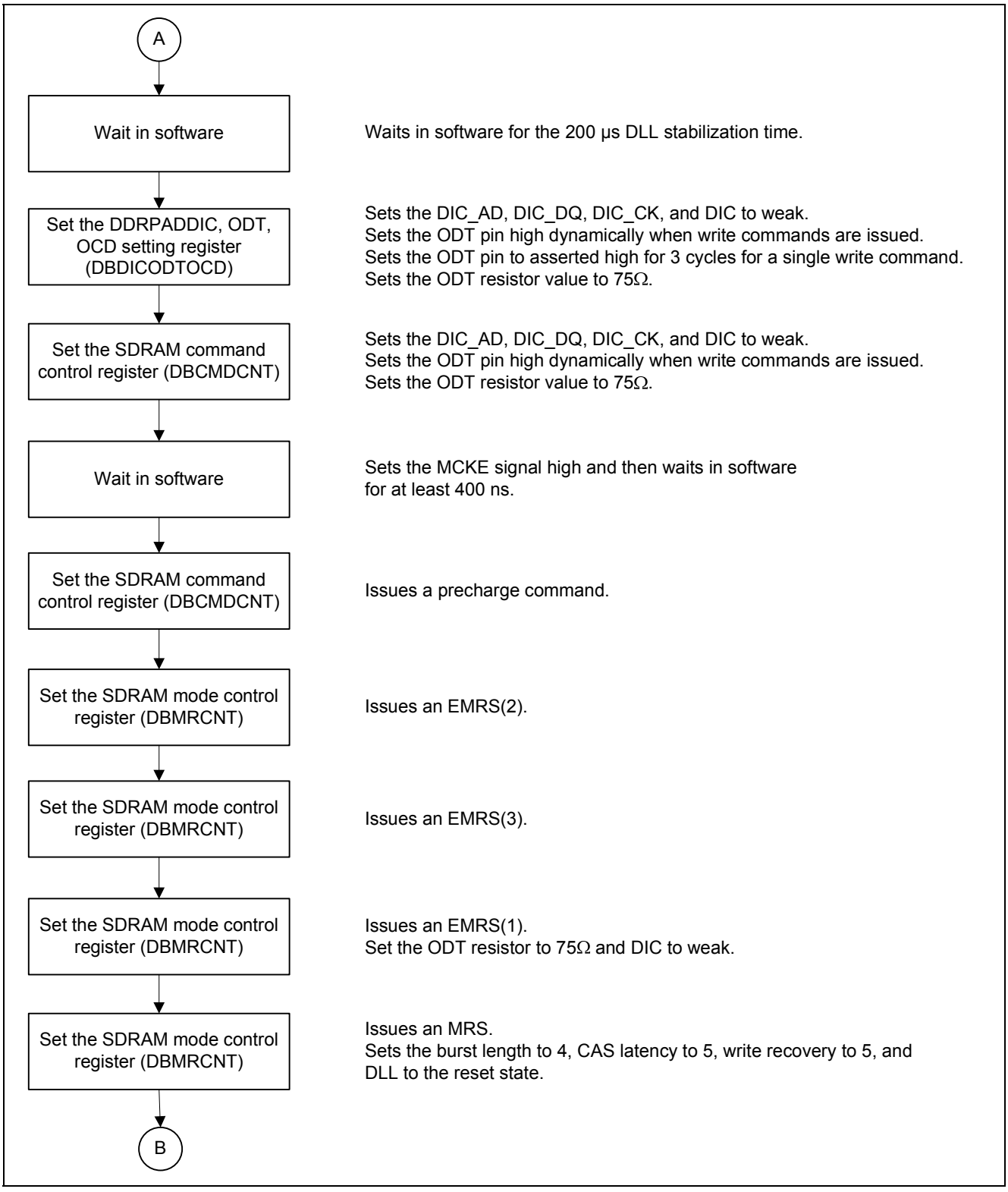


Figure 8 Flowchart for the _lowlevel_init() Function 2

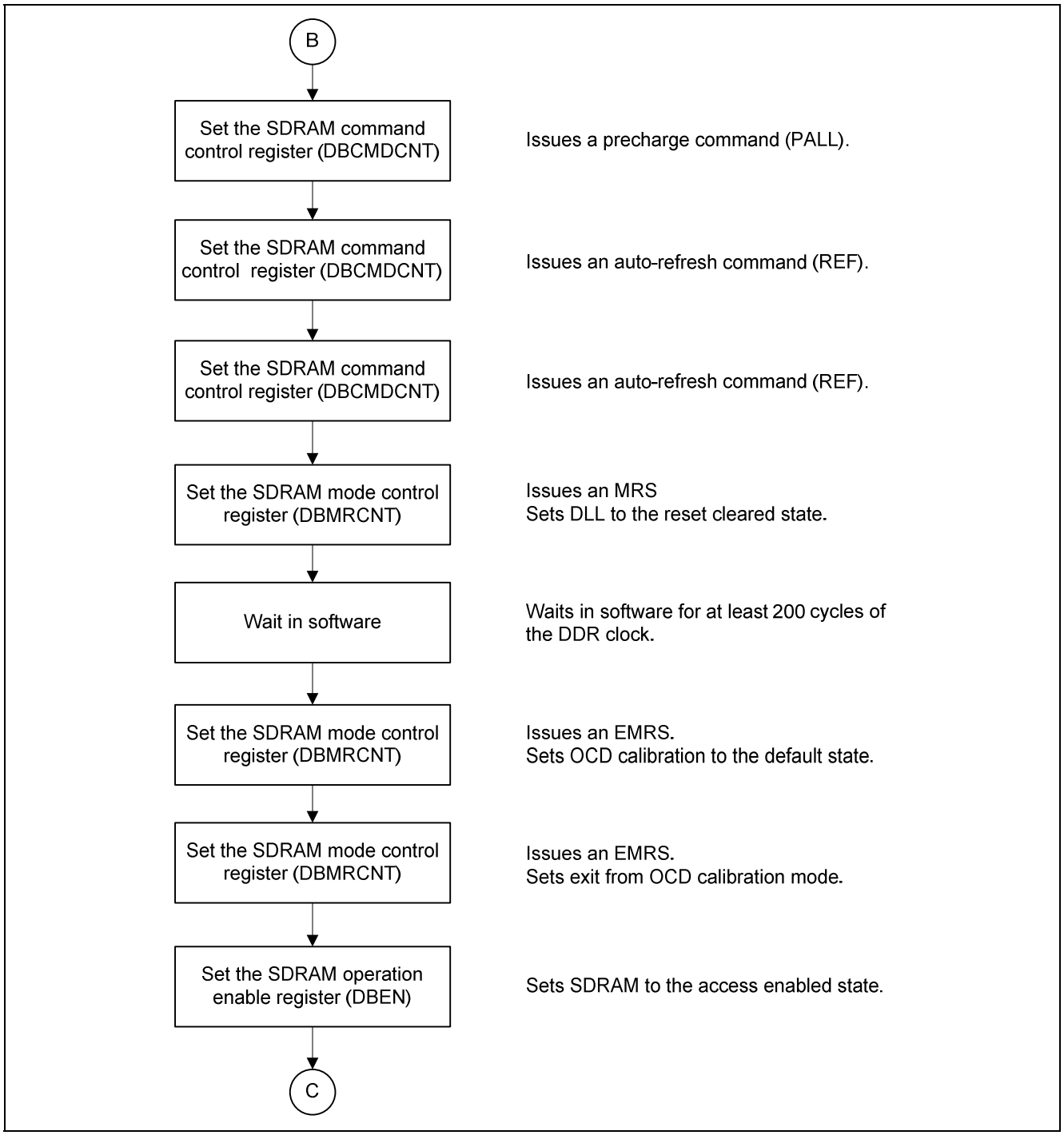
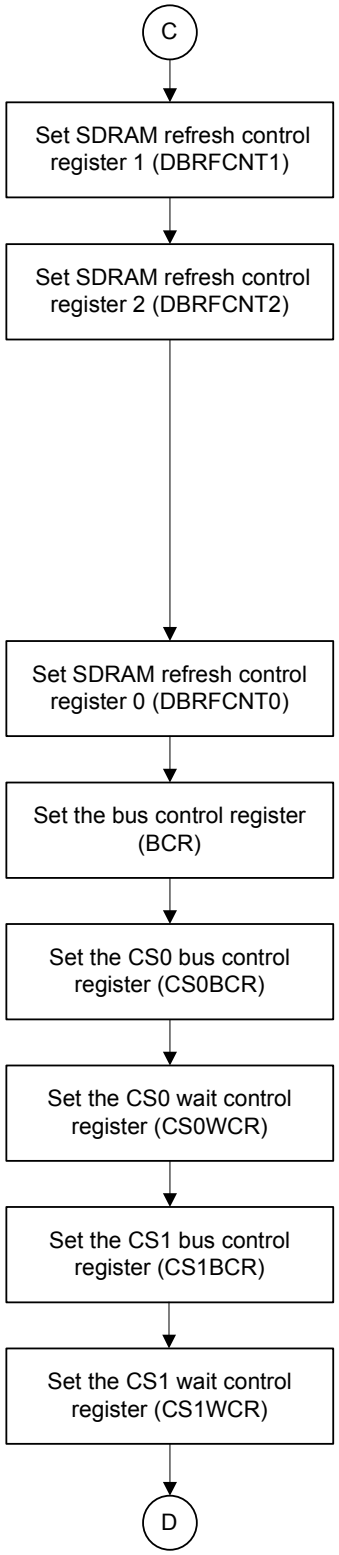


Figure 9 Flowchart for the _lowlevel_init() Function 3



Average refresh cycle: 7.8125 µs
DDR clock period at 300 MHz: 3.3333 ns
7.8125µs/3.3333ns = 2343
The average refresh cycle is set to 2343.
LV0 threshold: Threshold for performing an auto refresh between SHwy commands.
LV1 threshold: Threshold for performing an auto refresh when there is a success request opening due to the SHwy bus.
The LV1 threshold is set to 255 and the LV0 threshold is set to 255.

Note : Relationship between the average refresh cycle and the LV1 threshold:
When auto-refresh commands are issued periodically, the sum of the TREFI bits in the DBRFCNT1 register and the LV1TH bits in the DBRFCNT1 register will be the maximum interval between refresh commands.
The LV1TH bits must be set so that this sum falls within the maximum value for the ACT-PRE command interval in the specifications for the SDRAM devices used.

Enables auto-issuing of auto-refresh commands to.

Sets the RDY and BREQ pins to asynchronous input.

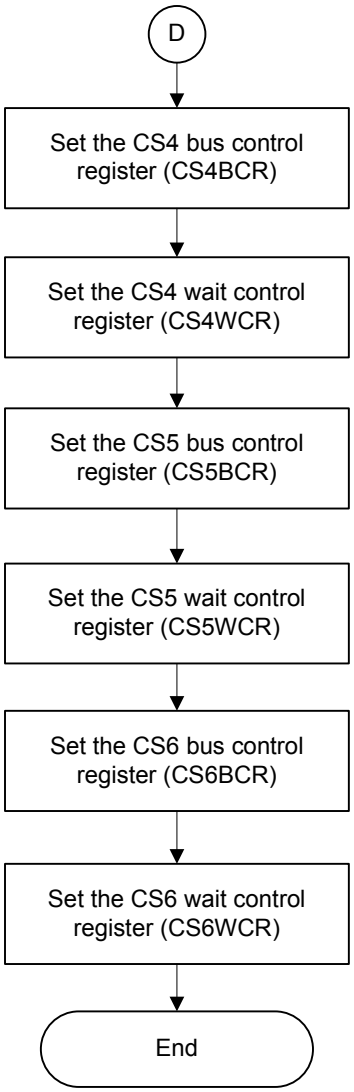
Sets the WE-RD/WE-WE intervals to 2 cycles, the different space RD-WE interval to 2 cycles, the same space RD-WE interval to 2 cycles, the different space RD-RD interval to 2 cycles, the same space RD-RD interval to 2 cycles, the same space RD-RD interval to 2 cycles, the bus width to 32 bits, the burst pitch to 4 wait states, and the memory type to SRAM.

Sets the CS-RD assert delay to 1 cycle, the RD-CS negate delay to 1 cycle, the CS-WE assert delay to 1 cycle, the WE-CS negate delay to 1 cycle, the BS assert period to 2, and the wait to 8 cycles.

Sets the WE-RD/WE-WE intervals to 1 cycle, the different space RD-WE interval to 1 cycle, the same space RD-WE interval to 1 cycle, the different space RD-RD interval to 1 cycle, the same space RD-RD interval to 1 cycle, the bus width to 8 bits, and the memory type to SRAM.

Sets the address to CS assert delay to 3, the CS to address negate delay to 3, the CS-RD assert delay to 3 cycles, the RD-CS negate delay to 3 cycles, the CS-WE assert delay to 3 cycles, the WE-CS negate delay to 3 cycles, the BS assert period to 1, and the wait to 3 cycles.

Figure 10 Flowchart for the _lowlevel_init() Function 4



Sets the WE-RD/WE-WE intervals to 1 cycle, the different space RD-WE interval to 1 cycle, the same space RD-WE interval to 1 cycle, the different space RD-RD interval to 1 cycle, the same space RD-RD interval to 1 cycle, the bus width to 32 bits, and the memory type to SRAM.

Sets the CS-RD assert delay to 1 cycle, the CS-WE assert delay to 1 cycle, the BS assert period to 1, and the wait to 2 cycles.

Sets the WE-RD/WE-WE intervals to 1 cycle, the different space RD-WE interval to 1 cycle, the same space RD-WE interval to 1 cycle, the different space RD-RD interval to 1 cycle, the same space RD-RD interval to 1 cycle, the bus width to 16 bits, and the memory type to SRAM.

Sets the RD-CS negate delay to 2 cycles, the BS assert period to 1, and the wait to 4 cycles.

Sets the bus width to 32 bits and the memory type to SRAM.

Sets the RD-CS negate delay to 3 cycles, the WE-CS negate delay to 1 cycle, the BS assert period to 1, and the wait to 8 cycles.

Figure 11 Flowchart for the _lowlevel_init() Function 5

Figure 12 shows the processing flow in the PowerON_Reset() function.

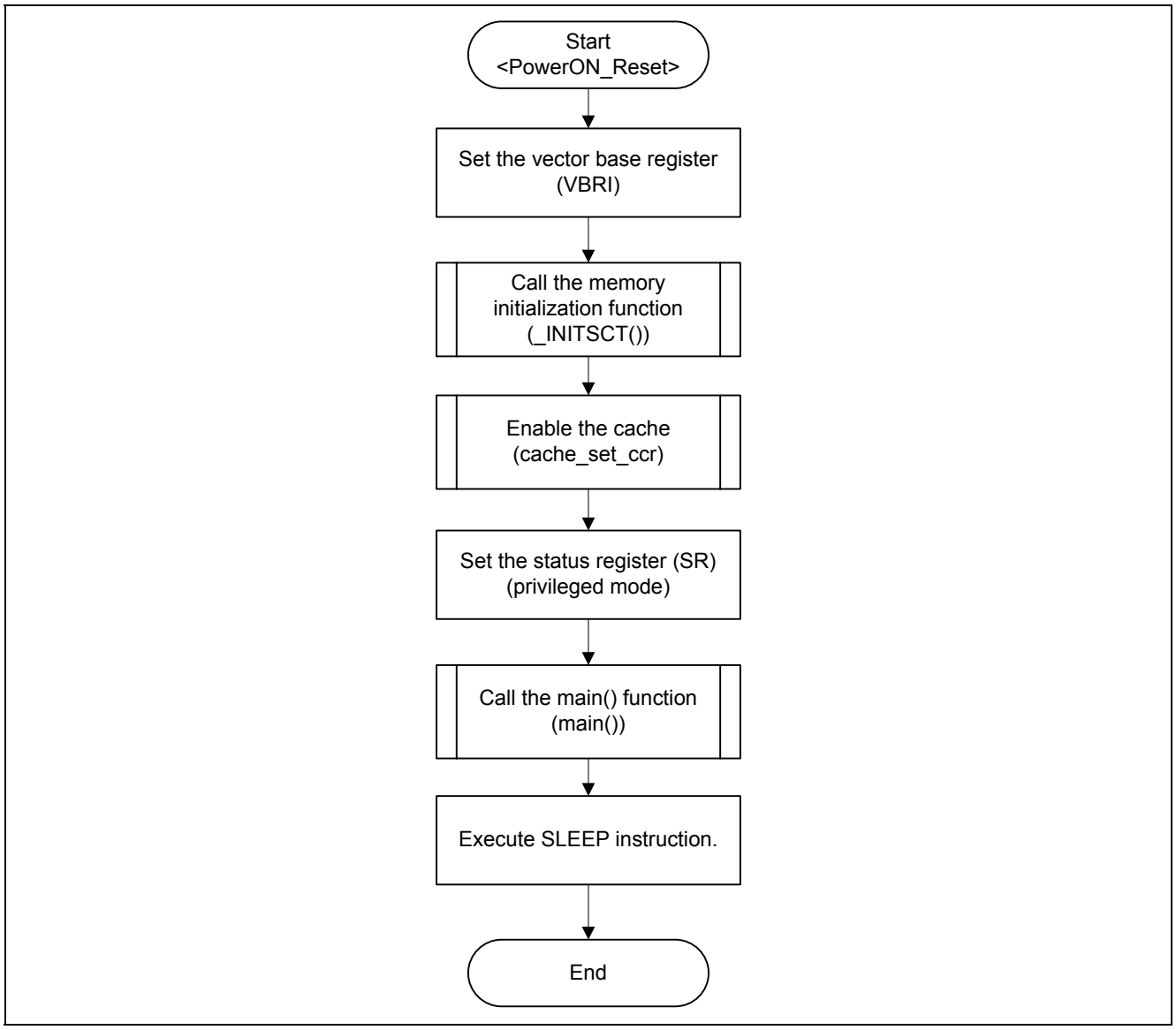


Figure 12 Flowchart for the PowerON_Reset() Function

5. Sample Program

Sample program listing: vhandler.src

The code in this file performs exception handling.

```
001 ;/***********************************************************************
002 ;* DISCLAIMER
003 ;
004 ;* This software is supplied by Renesas Electronics Corporation. and is only
005 ;* intended for use with Renesas products. No other uses are authorized.
006 ;
007 ;* This software is owned by Renesas Electronics Corporation. and is protected
008 ;* under all applicable laws, including copyright laws.
009 ;
010 ;* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
011 ;* REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
012 ;* INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
013 ;* PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
014 ;* DISCLAIMED.
015 ;
016 ;* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
017 ;* ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
018 ;* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
019 ;* FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
020 ;* AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
021 ;
022 ;* Renesas reserves the right, without notice, to make changes to this
023 ;* software and to discontinue the availability of this software.
024 ;* By using this software, you agree to the additional terms and
025 ;* conditions found by accessing the following link:
026 ;* http://www.renesas.com/disclaimer
027 ;*********************************************************** ***************/
028 ;/* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.*/
029 ;/*""FILE COMMENT""************ Technical reference data ****************
030 ;* System Name  : SH7785 Sample Program
031 ;* File Name    : vhandler.src
032 ;* Abstract     : SH7785 Initial Settings Sample Program
033 ;* Version      : Ver 1.00
034 ;* Device       : SH7785
035 ;* Tool-Chain   : High-performance Embedded Workshop (Version 4.07.00.007)
036 ;*              : C/C++ Compiler Package for SuperH Family (V.9.3.2.0)
037 ;* OS           : None
038 ;* H/W Platform : This is a sample program with examples of initialization for
039 ;* Description  : the SH7785 on the Renesas SH-4A board, model number R0P7785LC0011RL.
040 ;*              :
041 ;*              :
042 ;* Operation    :
043 ;* Limitation   :
044 ;*              :
045 ;*********************************************************************
046 ;* History      : 30.SEP.2010 Ver. 1.00 First Release
047 ;*""FILE COMMENT END""*************************************************/
048 ;---------------------------------------------------------------------
049 ;                                                                     |
050 ;   FILE        :vhandler.src                                         |
051 ;   DATE        :Thu, May 13, 2010                                    |
052 ;   DESCRIPTION :Reset/Interrupt Handler                               |
```

```
053 ;   CPU TYPE    :Other                                                    |
054 ;                                                                         |
055 ;   This file is generated by Renesas Project Generator (Ver.4.16).      |
056 ;                                                                         |
057 ;-------------------------------------------------------------------------
058                  .macro PUSH_EXP_BASE_REGA a, b
059         stc.l   ssr,@-r15               ; save ssr
060         stc.l   spc,@-r15               ; save spc
061         sts.l   pr,@-r15                ; save context registers
062         sts.l   fpscr,@-r15             ; save fpscr registers
063         stc.l   r7_bank,@-r15
064         stc.l   r6_bank,@-r15
065         stc.l   r5_bank,@-r15
066         stc.l   r4_bank,@-r15
067         stc.l   r3_bank,@-r15
068         stc.l   r2_bank,@-r15
069         stc.l   r1_bank,@-r15
070         stc.l   r0_bank,@-r15
071                 .endm
072 ;
073
074
075      .include "env.inc"
076      .include "vect.inc"
077
078 ;
079 ILLEGALFPU_CODE:           .equ  H'800
080 DUMMY_ILLEGALFPU_CODE:     .equ  H'880
081 ILLEGALSLOTFPU_CODE:       .equ  H'820
082 DUMMY_ILLEGALSLOTFPU_CODE: .equ  H'8A0
083 INT_NMI_CODE:              .equ  H'1C0
084 ;
085
086 IMASKclr:  .equ  H'FFFFFF0F
087 RBBLclr:   .equ  H'CFFFFFFF
088 MDRBBLset: .equ  H'70000000
089
090      .import     _RESET_Vectors
091      .import     _INT_Vectors
092      .import     _INT_MASK
093      .import     _lowlevel_init
094
095 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
096 ;*              macro definition                                        *;
097 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
098                  .macro PUSH_EXP_BASE_REG
099         stc.l   ssr,@-r15               ; save ssr
100         stc.l   spc,@-r15               ; save spc
101         sts.l   pr,@-r15                ; save context registers
102         sts.l   fpscr,@-r15             ; save fpscr registers
103         stc.l   r7_bank,@-r15
104         stc.l   r6_bank,@-r15
105         stc.l   r5_bank,@-r15
106         stc.l   r4_bank,@-r15
107         stc.l   r3_bank,@-r15
108         stc.l   r2_bank,@-r15
109         stc.l   r1_bank,@-r15
```

```
110         stc.l   r0_bank,@-r15
111                 .endm
112 ;
113                 .macro POP_EXP_BASE_REG
114         ldc.l   @r15+,r0_bank           ; recover registers
115         ldc.l   @r15+,r1_bank
116         ldc.l   @r15+,r2_bank
117         ldc.l   @r15+,r3_bank
118         ldc.l   @r15+,r4_bank
119         ldc.l   @r15+,r5_bank
120         ldc.l   @r15+,r6_bank
121         ldc.l   @r15+,r7_bank
122         lds.l   @r15+,fpscr
123         lds.l   @r15+,pr
124         ldc.l   @r15+,spc
125         ldc.l   @r15+,ssr
126                 .endm
127 ;
128 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
129 ;    reset                                                ;
130 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
131      .section RSTHandler,code
132 _ResetHandler:
133
134
135            mov.l #_lowlevel_init,r0
136            jsr   @r0
137            nop
138
139            mov.l #EXPEVT,r0
140            mov.l @r0,r0
141            shlr2 r0
142            shlr  r0
143            mov.l #_RESET_Vectors,r1
144            add   r1,r0
145            mov.l @r0,r0
146            jmp   @r0
147            nop
148 ;
149 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
150 ;    exceptional interrupt                              ;
151 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
152      .section INTHandler,code
153      .export     _INTHandlerPRG
154 _INTHandlerPRG:
155 _ExpHandler:
156            PUSH_EXP_BASE_REG
157 ;
158            mov.l #EXPEVT,r0        ; set event address
159            mov.l @r0,r1         ; set exception code
160
161 ;
162            mov.l #ILLEGALFPU_CODE,r2           ; H'800
163            cmp/eqr1,r2
164            bf    exp_01
165            mov.l #DUMMY_ILLEGALFPU_CODE,r1     ; H'800 -> H'880
166            bra   exp_10
```

```
167             nop
168 exp_01:
169             mov.l #ILLEGALSLOTFPU_CODE,r2       ; H'820
170             cmp/eqr1,r2
171             bf    exp_10
172             mov.l #DUMMY_ILLEGALSLOTFPU_CODE,r1 ; H'820 -> H'8A0
173 exp_10:
174
175 ;           mov.l #_INT_Vectors,r0  ; set vector table address
176             add   #-(h'40),r1       ; exception code - h'40
177             shlr2 r1
178             shlr  r1
179             mov.l @(r0,r1),r3       ; set interrupt function addr
180 ;
181             mov.l #_INT_MASK,r0     ; interrupt mask table addr
182             shlr2 r1
183             mov.b @(r0,r1),r1       ; interrupt mask
184             extu.br1,r1
185 ;
186             stc   sr,r0          ; save sr
187             mov.l #(RBBLclr&IMASKclr),r2     ; RB,BL,mask clear data
188             and   r2,r0          ; clear mask data
189             or    r1,r0          ; set interrupt mask
190             ldc   r0,ssr         ; set current status
191 ;
192             ldc.l r3,spc
193             mov.l #__int_term,r0    ; set interrupt terminate
194             lds   r0,pr
195 ;
196             rte
197             nop
198 ;
199             .pool
200 ;
201 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
202 ;    Interrupt terminate                                    ;
203 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
204             .align4
205 __int_term:
206             mov.l   #MDRBBLset,r0   ; set MD,BL,RB
207             ldc.l r0,sr          ;
208             POP_EXP_BASE_REG
209             rte            ; return
210             nop
211 ;
212             .pool
213 ;
214 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
215 ;    TLB miss interrupt                                ;
216 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
217      .org  H'300
218 _TLBmissHandler:
219             PUSH_EXP_BASE_REG
220 ;
221             mov.l #EXPEVT,r0        ; set event address
222             mov.l @r0,r1         ; set exception code
223             mov.l #_INT_Vectors,r0  ; set vector table address
```

```
224             add   #-(h'40),r1       ; exception code - h'40
225             shlr2 r1
226             shlr  r1
227             mov.l @(r0,r1),r3       ; set interrupt function addr
228 ;
229             mov.l #_INT_MASK,r0     ; interrupt mask table addr
230             shlr2 r1
231             mov.b @(r0,r1),r1       ; interrupt mask
232             extu.br1,r1
233 ;
234             stc   sr,r0          ; save sr
235             mov.l #(RBBLclr&IMASKclr),r2     ; RB,BL,mask clear data
236             and   r2,r0          ; clear mask data
237             or    r1,r0          ; set interrupt mask
238             ldc   r0,ssr         ; set current status
239 ;
240             ldc.l r3,spc
241             mov.l #__int_term,r0    ; set interrupt terminate
242             lds   r0,pr
243 ;
244             rte
245             nop
246 ;
247             .pool
248 ;
249 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
250 ;    IRQ                                                ;
251 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
252       .org  H'500
253 _IRQHandler:
254             PUSH_EXP_BASE_REG
255 ;
256             mov.l #INTEVT,r0        ; set event address
257             mov.l @r0,r1         ; set exception code
258             mov.l #_INT_Vectors,r0  ; set vector table address
259             add   #-(h'40),r1       ; exception code - h'40
260             shlr2 r1
261             shlr  r1
262             mov.l @(r0,r1),r3       ; set interrupt function addr
263 ;
264             mov.l #_INT_MASK,r0     ; interrupt mask table addr
265             shlr2 r1
266             mov.b @(r0,r1),r1       ; interrupt mask
267             extu.br1,r1
268 ;
269             stc   sr,r0          ; save sr
270             mov.l #(RBBLclr&IMASKclr),r2     ; RB,BL,mask clear data
271             and   r2,r0          ; clear mask data
272             or    r1,r0          ; set interrupt mask
273             ldc   r0,ssr         ; set current status
274 ;
275             ldc.l r3,spc
276             mov.l #__int_term,r0    ; set interrupt terminate
277             lds   r0,pr
278 ;
279             rte
280             nop
```

```
281 ;
282             .pool
283             .end
284
285
```

Sample program listing: vhandler.src

This is the power-on reset function.

```
001/***********************************************************************
002 * DISCLAIMER
003
004 * This software is supplied by Renesas Electronics Corporation. and is only
005 * intended for use with Renesas products. No other uses are authorized.
006
007 * This software is owned by Renesas Electronics Corporation. and is protected
008 * under all applicable laws, including copyright laws.
009
010 * THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
011 * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
012 * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
013 * PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
014 * DISCLAIMED.
015
016 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
017 * ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
018 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
019 * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
020 * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
021
022 * Renesas reserves the right, without notice, to make changes to this
023 * software and to discontinue the availability of this software.
024 * By using this software, you agree to the additional terms and
025 * conditions found by accessing the following link:
026 * http://www.renesas.com/disclaimer
027 ***********************************************************************/
028 /* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.*/
029 /*""FILE COMMENT""************ Technical reference data ****************
030 * System Name  : SH7785 Sample Program
031 * File Name    : Initialize_Base.c
032 * Abstract     : SH7785 Initial Settings Sample Program
033 * Version      : Ver 1.00
034 * Device       : SH7785
035 * Tool-Chain   : High-performance Embedded Workshop (Version 4.07.00.007)
036 *              : C/C++ Compiler Package for SuperH Family (V.9.3.2.0)
037 * OS           : None
038 * H/W Platform : This is a sample program with examples of initialization for
039 * Description  : the SH7785 on the Renesas SH-4A board, model number R0P7785LC0011RL.
040 *              :
041 *              :
042 * Operation    :
043 * Limitation   :
044 *              :
045 ***********************************************************************
046 * History      : 30.SEP.2010 Ver. 1.00 First Release
047 *""FILE COMMENT END""*****************************************************/
048 /*********************************************************************/
049 /*                                                                   */
050 /*  FILE        :resetprg.c                                          */
051 /*  DATE        :Thu, May 13, 2010                                   */
052 /*  DESCRIPTION :Reset Program                                       */
053 /*  CPU TYPE    :Other                                               */
054 /*                                                                   */
```

```
055 /*  This file is generated by Renesas Project Generator (Ver.4.16).    */
056 /*                                                                      */
057 /************************************************************************/
058
059
060
061 #include<machine.h>
062 #include<_h_c_lib.h>
063 //#include <stddef.h>           // Remove the comment when you use errno
064 //#include <stdlib.h>           // Remove the comment when you use rand()
065 #include"typedefine.h"
066 #include"stacksct.h"
067 #include"cache.h"
068
069
070 #define SR_Init    0x40000000
071 #define INT_OFFSET 0x100UL
072
073 #ifdef __cplusplus
074 extern "C" {
075 #endif
076 extern void INTHandlerPRG(void);
077 void PowerON_Reset(void);
078 void Manual_Reset(void);
079 void main(void);
080 #ifdef __cplusplus
081 }
082 #endif
083
084 #pragma section ResetPRG
085
086
087 #pragma entry PowerON_Reset
088
089 void PowerON_Reset(void)
090 {
091   set_vbr((void *)((_UINT)INTHandlerPRG - INT_OFFSET));
092
093
094   _INITSCT();
095
096 //   errno=0;                   // Remove the comment when you use errno
097 //   srand((_UINT)1);           // Remove the comment when you use rand()
098 //   _s1ptr=NULL;             // Remove the comment when you use strtok()
099   cache_set_ccr(D_CACHE_I_ON | D_CACHE_O_ON);
100
101   set_cr(SR_Init);
102
103   main();
104
105 //   _CLOSEALL();   // Close I/O in the application(both SIM I/O and hardware I/O)
106
107 //   _CALL_END();   // Remove the comment when you use global class object
108
109   sleep();
110 }
111
```

```
112 void Manual_Reset(void)
113 {
114 }
```

Sample program listing: dbsct.c

The code in this file performs memory initialization.

```
01 /***********************************************************************/
02 /*                                                                     */
03 /*  FILE        :dbsct.c                                               */
04 /*  DATE        :Thu, May 13, 2010                                     */
05 /*  DESCRIPTION :Setting of B,R Section                                */
06 /*  CPU TYPE    :Other                                                 */
07 /*                                                                     */
08 /*  This file is generated by Renesas Project Generator (Ver.4.16).    */
09 /*                                                                     */
10 /***********************************************************************/
11
12
13
14 #include "typedefine.h"
15
16
17 #pragma section $DSEC
18 static const struct {
19     _UBYTE *rom_s;       /* Start address in ROM for the initialization data section */
20     _UBYTE *rom_e;       /* End address in ROM for the initialization data section */
21     _UBYTE *ram_s;       /* Start address in RAM for the initialization data section */
22 }   DTBL[] = {
23     { __sectop("D"), __secend("D"), __sectop("R") }
24 };
25 #pragma section $BSEC
26 static const struct {
27     _UBYTE *b_s;         /* Start address of the uninitialized data section */
28     _UBYTE *b_e;         /* End address of the uninitialized data section */
29 }   BTBL[] = {
30     { __sectop("B"), __secend("B") }
31 };
```

Sample program listing: vecttbl.src

The code in this file sets the interrupt priorities used during interrupt handling.

```
001 ;/***********************************************************************
002 ;* DISCLAIMER
003 ;
004 ;* This software is supplied by Renesas Electronics Corporation. and is only
005 ;* intended for use with Renesas products. No other uses are authorized.
006 ;
007 ;* This software is owned by Renesas Electronics Corporation. and is
008 ;* protected under all applicable laws, including copyright laws.
009 ;
010 ;* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
011 ;* REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
012 ;* INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
013 ;* PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
014 ;* DISCLAIMED.
015 ;
016 ;* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
017 ;* ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
018 ;* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
019 ;* FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
020 ;* AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
021 ;
022 ;* Renesas reserves the right, without notice, to make changes to this
023 ;* software and to discontinue the availability of this software.
024 ;* By using this software, you agree to the additional terms and
025 ;* conditions found by accessing the following link:
026 ;* http://www.renesas.com/disclaimer
027 ;***********************************************************************/
028 ;/* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.*/
029 ;/*""FILE COMMENT""************ Technical reference data ****************
030 ;* System Name  : SH7785 Sample Program
031 ;* File Name    : vecttbl.src
032 ;* Abstract     : SH7785 Initial Settings Sample Program
033 ;* Version      : Ver 1.00
034 ;* Device       : SH7785
035 ;* Tool-Chain   : High-performance Embedded Workshop (Version 4.07.00.007)
036 ;*              : C/C++ Compiler Package for SuperH Family (V.9.3.2.0)
037 ;* OS           : None
038 ;* H/W Platform : This is a sample program with examples of initialization for
039 ;* Description  : the SH7785 on the Renesas SH-4A board, model number R0P7785LC0011RL.
040 ;*              :
041 ;*              :
042 ;* Operation    :
043 ;* Limitation   :
044 ;*              :
045 ;***********************************************************************
046 ;* History      : 30.SEP.2010 Ver. 1.00 First Release
047 ;*""FILE COMMENT END""*************************************************/
048 ;----------------------------------------------------------------------
049 ;                                                                      |
050 ;   FILE        :vecttbl.src                                           |
051 ;   DATE        :Thu, May 13, 2010                                     |
052 ;   DESCRIPTION :Initialize of Vector Table                             |
053 ;   CPU TYPE    :Other                                                 |
054 ;                                                                      |
```

```
055 ;   This file is generated by Renesas Project Generator (Ver.4.16).     |
056 ;                                                                       |
057 ;-----------------------------------------------------------------------
058 ;/***********************************************************************
059 ;*
060 ;* Device     : SH-4A/SH7785
061 ;*
062 ;* File Name  : vecttbl.src
063 ;*
064 ;* Abstract   : Initialize of Vector Table.
065 ;*
066 ;* History    : 1.00  (2010-09-30)  [Hardware Manual Revision : 1.00]
067 ;*
068 ;* Copyright(c) 2010 Renesas Electronics Corporation.
069 ;*               And Renesas Solutions Corp.,All Rights Reserved.
070 ;*
071 ;***********************************************************************/
072
073 ;    .include    "config.h"
074
075         .include        "vect.inc"
076
077         .section        VECTTBL,data
078         .export         _RESET_Vectors
079
080 _RESET_Vectors:
081 ;<<VECTOR DATA START (POWER ON RESET)>>
082         ;H'000 Power On Reset (Hitachi-UDI RESET)
083         .data.l  _PowerON_Reset
084 ;<<VECTOR DATA END (POWER ON RESET)>>
085 ;<<VECTOR DATA START (MANUAL RESET)>>
086         ;H'020 Manual Reset
087         .data.l  _Manual_Reset
088 ;<<VECTOR DATA END (MANUAL RESET)>>
089 ; Reserved
090         .datab.l        8,H'00000000
091 ;<<VECTOR DATA START (TLB RESET)>>
092         ;H'140 TLB Reset (DATA TLB Reset)
093         .data.l         _TLB_Reset
094 ;<<VECTOR DATA END (TLB RESET)>>
095
096         .section    INTTBL,data
097         .export     _INT_Vectors
098 _INT_Vectors:
099             ; H'040     Data TLB miss exception(read)
100             .data.l     _INT_TLB_MISS_READ_EXP
101             ; H'060     Data TLB miss exception(write)
102             .data.l     _INT_TLB_MISS_WRITE_EXP
103             ; H'080     Initial page write exception
104             .data.l     _INT_TLB_INIT_PAGE_EXP
105             ; H'0A0     Data TLB protection violation exception (read)
106             .data.l     _INT_TLB_PROTECT_READ_EXP
107             ; H'0C0     Data TLB protection violation exception (write)
108             .data.l     _INT_TLB_PROTECT_WRITE_EXP
109             ; H'0E0     Data address error(read)
110             .data.l     _INT_ADR_ERROR_READ
111             ; H'100     Data address error(write)
```

```
112             .data.l     _INT_ADR_ERROR_WRITE
113             ; H'120     FPU exception
114             .data.l     _INT_FPU_EXP
115             ; H'140     Instruction TLB multiple-hit exception
116             .data.l     _TLB_Reset
117             ; H'160     Unconditional trap(TRAPA)
118             .data.l     _INT_TRAP
119             ; H'180     General illegal instruction exception
120             .data.l     _INT_ILLEGAL_INST_EXP
121             ; H'1A0     Slot illegal instruction exception
122             .data.l     _INT_ILLEGAL_SLOT_EXP
123 ;EXTERNAL INTERRUPT
124             ; H'1C0     NMI
125             .data.l     _INT_NMI
126             ; H'1E0     USER_BREAK
127             .data.l     _INT_USER_BREAK
128
129             ; H'200     IRL30_LEVEL15_IRQ7
130             .data.l     _INT_IRL30_LEVEL15_IRQ7
131             ; H'220     IRL30_LEVEL14
132             .data.l     _INT_IRL30_LEVEL14
133             ; H'240     IRL30_LEVEL13_IRQ0
134             .data.l     _INT_IRL30_LEVEL13_IRQ0
135             ; H'260     IRL30_LEVEL12
136             .data.l     _INT_IRL30_LEVEL12
137             ; H'280     IRL30_LEVEL11_IRQ1
138             .data.l     _INT_IRL30_LEVEL11_IRQ1
139             ; H'2A0     IRL30_LEVEL10
140             .data.l     _INT_IRL30_LEVEL10
141             ; H'2C0     IRL30_LEVEL9_IRQ2
142             .data.l     _INT_IRL30_LEVEL9_IRQ2
143             ; H'2E0     IRL30_LEVEL8
144             .data.l     _INT_IRL30_LEVEL8
145             ; H'300     IRL30_LEVEL7_IRQ3
146             .data.l     _INT_IRL30_LEVEL7_IRQ3
147             ; H'320     IRL30_LEVEL6
148             .data.l     _INT_IRL30_LEVEL6
149             ; H'340     IRL30_LEVEL5_IRQ4
150             .data.l     _INT_IRL30_LEVEL5_IRQ4
151             ; H'360     IRL30_LEVEL4
152             .data.l     _INT_IRL30_LEVEL4
153             ; H'380     IRL30_LEVEL3_IRQ5
154             .data.l     _INT_IRL30_LEVEL3_IRQ5
155             ; H'3A0     IRL30_LEVEL2
156             .data.l     _INT_IRL30_LEVEL2
157             ; H'3C0     IRL30_LEVEL1_IRQ6
158             .data.l     _INT_IRL30_LEVEL1_IRQ6
159             ;H'3E0-540  Reserved
160             .datab.l    12,H'00000000
161 ;WDT
162             ;H'560      WDT_ITI
163             .data.l     _INT_WDT_ITI
164 ;TMU-ch0
165             ;H'580      TMU_TUNI0
166             .data.l     _INT_TMU_TUNI0
167 ;TMU-ch1
168             ;H'5A0      TMU_TUNI1
```

```
169             .data.l     _INT_TMU_TUNI1
170 ;TMU-ch2
171             ;H'5C0      TMU_TUNI2
172             .data.l     _INT_TMU_TUNI2
173             ;H'5E0      TMU_TICPI2
174             .data.l     _INT_TMU_TICPI2
175 ;H-UDI
176             ;H'600      H-UDII
177             .data.l     _INT_H_UDII
178 ;DMAC(0)
179             ;H'620      DMINT0
180             .data.l     _INT_DMAC_DMINT0
181             ;H'640      DMINT1
182             .data.l     _INT_DMAC_DMINT1
183             ;H'660      DMINT2
184             .data.l     _INT_DMAC_DMINT2
185             ;H'680      DMINT3
186             .data.l     _INT_DMAC_DMINT3
187             ;H'6A0      DMINT4
188             .data.l     _INT_DMAC_DMINT4
189             ;H'6C0      DMINT5
190             .data.l     _INT_DMAC_DMINT5
191             ;H'6E0      DMAE0(ch0-5)
192             .data.l     _INT_DMAC_DMAE0
193 ;SCIF(0)
194             ;H'700      SCIF_ERI0
195             .data.l     _INT_SCIF_ERI0
196             ;H'720      SCIF_RXI0
197             .data.l     _INT_SCIF_RXI0
198             ;H'740      SCIF_BRI0
199             .data.l     _INT_SCIF_BRI0
200             ;H'760      SCIF_TXI0
201             .data.l     _INT_SCIF_TXI0
202 ;SCIF(1)
203             ;H'780      SCIF_ERI1
204             .data.l     _INT_SCIF_ERI1
205             ;H'7A0      SCIF_RXI1
206             .data.l     _INT_SCIF_RXI1
207             ;H'7C0      SCIF_BRI1
208             .data.l     _INT_SCIF_BRI1
209             ;H'7E0      SCIF_TXI1
210             .data.l     _INT_SCIF_TXI1
211             ;H'800-860  Reserved
212             .datab.l    4,H'00000000
213 ;DMAC(1)
214             ;H'880      DMINT6
215             .data.l     _INT_DMAC_DMINT6
216             ;H'8A0      DMINT7
217             .data.l     _INT_DMAC_DMINT7
218             ;H'8C0      DMINT8
219             .data.l     _INT_DMAC_DMINT8
220             ;H'8E0      DMINT9
221             .data.l     _INT_DMAC_DMINT9
222             ;H'900      DMINT10
223             .data.l     _INT_DMAC_DMINT10
224             ;H'920      DMINT11
225             .data.l     _INT_DMAC_DMINT11
```

```
226             ;H'940      DMAE1(ch6-11)
227             .data.l     _INT_DMAC_DMAE0
228 ;HSPI
229             ;H'960      HSPI_SPII
230             .data.l     _INT_HSPI_SPII
231 ;SCIF(2)
232             ;H'980      SCIF_SCIFI2
233             .data.l     _INT_SCIF_SCIFI2
234 ;SCIF(3)
235             ;H'9A0      SCIF_SCIFI3
236             .data.l     _INT_SCIF_SCIFI3
237 ;SCIF(4)
238             ;H'9C0      SCIF_SCIFI4
239             .data.l     _INT_SCIF_SCIFI4
240 ;SCIF(5)
241             ;H'9E0      SCIF_SCIFI5
242             .data.l     _INT_SCIF_SCIFI5
243 ;PCIC(0)
244             ;H'A00      PCI_PCISERR
245             .data.l     _INT_PCI_PCISERR
246 ;PCIC(1)
247             ;H'A20      PCI_PCIINTA
248             .data.l     _INT_PCI_PCIINTA
249 ;PCIC(2)
250             ;H'A40      PCI_PCIINTB
251             .data.l     _INT_PCI_PCIINTB
252 ;PCIC(3)
253             ;H'A60      PCI_PCIINTC
254             .data.l     _INT_PCI_PCIINTC
255 ;PCIC(4)
256             ;H'A80      PCI_PCIINTD
257             .data.l     _INT_PCI_PCIINTD
258 ;PCIC(5)
259             ;H'AA0      PCI_PCIERR
260             .data.l     _INT_PCI_PCIERR
261             ;H'AC0      PCI_PCIPWD3_1
262             .data.l     _INT_PCI_PCIPWD3_1
263             ;H'AE0      PCI_PCIPWD0
264             .data.l     _INT_PCI_PCIPWD0
265 ;IRL(7-4)
266             ; H'B00     IRL74_LEVEL15
267             .data.l     _INT_IRL74_LEVEL15
268             ; H'B20     IRL74_LEVEL14
269             .data.l     _INT_IRL74_LEVEL14
270             ; H'B40     IRL74_LEVEL13
271             .data.l     _INT_IRL74_LEVEL13
272             ; H'B60     IRL74_LEVEL12
273             .data.l     _INT_IRL74_LEVEL12
274             ; H'B80     IRL74_LEVEL11
275             .data.l     _INT_IRL74_LEVEL11
276             ; H'BA0     IRL74_LEVEL10
277             .data.l     _INT_IRL74_LEVEL10
278             ; H'BC0     IRL74_LEVEL9
279             .data.l     _INT_IRL74_LEVEL9
280             ; H'BE0     IRL74_LEVEL8
281             .data.l     _INT_IRL74_LEVEL8
282             ; H'C00     IRL74_LEVEL7
```

```
283             .data.l     _INT_IRL74_LEVEL7
284             ; H'C20     IRL74_LEVEL6
285             .data.l     _INT_IRL74_LEVEL6
286             ; H'C40     IRL74_LEVEL5
287             .data.l     _INT_IRL74_LEVEL5
288             ; H'C60     IRL74_LEVEL4
289             .data.l     _INT_IRL74_LEVEL4
290             ; H'C80     IRL74_LEVEL3
291             .data.l     _INT_IRL74_LEVEL3
292             ; H'CA0     IRL74_LEVEL2
293             .data.l     _INT_IRL74_LEVEL2
294             ; H'CC0     IRL74_LEVEL1
295             .data.l     _INT_IRL74_LEVEL1
296 ;SIOF
297             ;H'CE0      SIOF_SIOFI
298             .data.l     _INT_SIOF_SIOFI
299 ;MMCIF
300             ;H'D00      MMCIF_FSTAT
301             .data.l     _INT_MMCIF_FSTAT
302             ;H'D20      MMCIF_TRAN
303             .data.l     _INT_MMCIF_TRAN
304             ;H'D40      MMCIF_ERR
305             .data.l     _INT_MMCIF_ERR
306             ;H'D60      MMCIF_FRDY
307             .data.l     _INT_MMCIF_FRDY
308 ;DU
309             ;H'D80      DU_DUI
310             .data.l     _INT_DU_DUI
311 ;GDTA
312             ;H'DA0      GDTA_GACLI
313             .data.l     _INT_GDTA_GACLI
314             ;H'DC0      GDTA_GAMCI
315             .data.l     _INT_GDTA_GAMCI
316             ;H'DE0      GDTA_GAERI
317             .data.l     _INT_GDTA_GAERI
318 ;TMU-ch3
319             ;H'E00      TMU_TUNI3
320             .data.l     _INT_TMU_TUNI3
321 ;TMU-ch4
322             ;H'E20      TMU_TUNI4
323             .data.l     _INT_TMU_TUNI4
324 ;TMU-ch5
325             ;H'E40      TMU_TUNI5
326             .data.l     _INT_TMU_TUNI5
327             ;H'E60      Reserved
328             .data.l     H'00000000
329 ;SSI-ch0
330             ;H'E80      SSI_SSII0
331             .data.l     _INT_SSI_SSII0
332 ;SSI-ch1
333             ;H'EA0      SSI_SSII1
334             .data.l     _INT_SSI_SSII1
335 ;HAC-ch0
336             ;H'EC0      HAC_HACI0
337             .data.l     _INT_HAC_HACI0
338 ;HAC-ch1
339             ;H'EE0      HAC_HACI1
```

```
340             .data.l     _INT_HAC_HACI1
341 ;FLCTL
342             ;H'F00      FLCTL_FLSTE
343             .data.l     _INT_FLCTL_FLSTE
344             ;H'F20      FLCTL_FLTEND
345             .data.l     _INT_FLCTL_FLTEND
346             ;H'F40      FLCTL_FLTRQ0
347             .data.l     _INT_FLCTL_FLTRQ0
348             ;H'F60      FLCTL_FLTRQ1
349             .data.l     _INT_FLCTL_FLTRQ1
350 ;GPIO
351             ;H'F80      GPIO_GPIOI0
352             .data.l     _INT_GPIO_GPIIOI0
353             ;H'FA0      GPIO_GPIOI1
354             .data.l     _INT_GPIO_GPIIOI1
355             ;H'FC0      GPIO_GPIOI2
356             .data.l     _INT_GPIO_GPIIOI2
357             ;H'FE0      GPIO_GPIOI3
358             .data.l     _INT_GPIO_GPIIOI3
359
360         .export     _INT_MASK
361 _INT_MASK:
362             ; interrupt priority mask level(31 to 0)
363
364             ;H'040      Data TLB miss exception(read)
365             .data.b     H'F0
366             ;H'060      Data TLB miss exception(write)
367             .data.b     H'F0
368             ;H'080      Initial page write exception
369             .data.b     H'F0
370             ;H'0A0      Data TLB protection violation exception (read)
371             .data.b     H'F0
372             ;H'0C0      Data TLB protection violation exception (write)
373             .data.b     H'F0
374             ;H'0E0      Data address error(read)
375             .data.b     H'F0
376             ;H'100      Data address error(write)
377             .data.b     H'F0
378             ;H'120      FPU exception
379             .data.b     H'F0
380             ;H'140      Instruction TLB multiple-hit exception
381             .data.b     H'F0
382             ;H'160      TRAPA
383             .data.b     H'F0
384             ;H'180      ILLEGAL_INST
385             .data.b     H'F0
386             ;H'1A0      ILLEGAL_SLOT
387             .data.b     H'F0
388 ;EXTERNAL INTERRUPT
389             ;H'1c0      NMI
390             .data.b     H'F0
391             ;H'1E0      USER_BREAK
392             .data.b     H'F0
393
394             ; H'200-3C0 IRL_IRQ7-0
395             .datab.b    15,H'F0
396
```

```
397             ;H'3E0-540  Reserved
398             .datab.b    12,H'F0
399 ;WDT
400             ;H'560      WDT_ITI
401             .data.b     H'F0
402 ;TMU-ch0
403             ;H'580      TMU_TUNI0
404             .data.b     H'F0
405 ;TMU-ch1
406             ;H'5A0      TMU_TUNI1
407             .data.b     H'F0
408 ;TMU-ch2
409             ;H'5C0      TMU_TUNI2
410             .data.b     H'F0
411             ;H'5E0      TMU_TICPI2
412             .data.b     H'F0
413 ;H-UDI
414             ;H'600      H-UDII
415             .data.b     H'F0
416 ;DMAC(0)
417             ;H'620      DMINT0
418             .data.b     H'F0
419             ;H'640      DMINT1
420             .data.b     H'F0
421             ;H'660      DMINT2
422             .data.b     H'F0
423             ;H'680      DMINT3
424             .data.b     H'F0
425             ;H'6A0      DMINT4
426             .data.b     H'F0
427             ;H'6C0      DMINT5
428             .data.b     H'F0
429             ;H'6E0      DMAE0(ch0-5)
430             .data.b     H'F0
431 ;SCIF(0)
432             ;H'700      SCIF_ERI0
433             .data.b     H'F0
434             ;H'720      SCIF_RXI0
435             .data.b     H'F0
436             ;H'740      SCIF_BRI0
437             .data.b     H'F0
438             ;H'760      SCIF_TXI0
439             .data.b     H'F0
440 ;SCIF(1)
441             ;H'780      SCIF_ERI1
442             .data.b     H'F0
443             ;H'7A0      SCIF_RXI1
444             .data.b     H'F0
445             ;H'7C0      SCIF_BRI1
446             .data.b     H'F0
447             ;H'7E0      SCIF_TXI1
448             .data.b     H'F0
449             ;H'800-860  Reserved
450             .datab.b    4,H'F0
451 ;           .data.b     H'F0
452 ;DMAC(1)
453             ;H'880      DMINT6
```

```
454             .data.b     H'F0
455             ;H'8A0      DMINT7
456             .data.b     H'F0
457             ;H'8C0      DMINT8
458             .data.b     H'F0
459             ;H'8E0      DMINT9
460             .data.b     H'F0
461             ;H'900      DMINT10
462             .data.b     H'F0
463             ;H'920      DMINT11
464             .data.b     H'F0
465             ;H'940      DMAE1(ch6-11)
466             .data.b     H'F0
467 ;HSPI
468             ;H'960      HSPI_SPII
469             .data.b     H'F0
470 ;SCIF(2)
471             ;H'980      SCIF_SCIFI2
472             .data.b     H'F0
473 ;SCIF(3)
474             ;H'9A0      SCIF_SCIFI3
475             .data.b     H'F0
476 ;SCIF(4)
477             ;H'9C0      SCIF_SCIFI4
478             .data.b     H'F0
479 ;SCIF(5)
480             ;H'9E0      SCIF_SCIFI5
481             .data.b     H'F0
482 ;PCIC(0)
483             ;H'A00      PCI_PCISERR
484             .data.b     H'F0
485 ;PCIC(1)
486             ;H'A20      PCI_PCIINTA
487             .data.b     H'F0
488 ;PCIC(2)
489             ;H'A40      PCI_PCIINTB
490             .data.b     H'F0
491 ;PCIC(3)
492             ;H'A60      PCI_PCIINTC
493             .data.b     H'F0
494 ;PCIC(4)
495             ;H'A80      PCI_PCIINTD
496             .data.b     H'F0
497 ;PCIC(5)
498             ;H'AA0      PCI_PCIERR
499             .data.b     H'F0
500             ;H'AC0      PCI_PCIPWD3_1
501             .data.b     H'F0
502             ;H'AE0      PCI_PCIPWD0
503             .data.b     H'F0
504 ;IRL(7-4)
505             ; H'B00-CC0 IRL74
506             .datab.b    15,H'F0
507 ;SIOF
508             ;H'CE0      SIOF_SIOFI
509             .data.b     H'F0
510 ;MMCIF
```

```
511             ;H'D00      MMCIF_FSTAT
512             .data.b     H'F0
513             ;H'D20      MMCIF_TRAN
514             .data.b     H'F0
515             ;H'D40      MMCIF_ERR
516             .data.b     H'F0
517             ;H'D60      MMCIF_FRDY
518             .data.b     H'F0
519 ;DU
520             ;H'D80      MMCIF_FSTAT
521             .data.b     H'F0
522 ;GDTA
523             ;H'DA0      GDTA_GACLI
524             .data.b     H'F0
525             ;H'DC0      GDTA_GAMCI
526             .data.b     H'F0
527             ;H'DE0      GDTA_GAERI
528             .data.b     H'F0
529 ;TMU-ch3
530             ;H'E00      TMU_TUNI3
531             .data.b     H'F0
532 ;TMU-ch4
533             ;H'E20      TMU_TUNI4
534             .data.b     H'F0
535 ;TMU-ch5
536             ;H'E40      TMU_TUNI5
537             .data.b     H'F0
538             ;H'E60      Reserved
539             .data.b     H'F0
540 ;SSI-ch0
541             ;H'E80      SSI_SSII0
542             .data.b     H'F0
543 ;SSI-ch1
544             ;H'EA0      SSI_SSII1
545             .data.b     H'F0
546 ;HAC-ch0
547             ;H'EC0      HAC_HACI0
548             .data.b     H'F0
549 ;HAC-ch1
550             ;H'EE0      HAC_HACI1
551             .data.b     H'F0
552 ;FLCTL
553             ;H'F00      FLCTL_FLSTE
554             .data.b     H'F0
555             ;H'F20      FLCTL_FLTEND
556             .data.b     H'F0
557             ;H'F40      FLCTL_FLTRQ0
558             .data.b     H'F0
559             ;H'F60      FLCTL_FLTRQ1
560             .data.b     H'F0
561 ;GPIO
562             ;H'F80      GPIO_GPIOI0
563             .data.b     H'F0
564             ;H'FA0      GPIO_GPIOI1
565             .data.b     H'F0
566             ;H'FC0      GPIO_GPIOI2
567             .data.b     H'F0
```

```
568             ;H'FE0      GPIO_GPIOI3
569             .data.b     H'F0
570           .end
571
572
```

Sample program listing: vect.inc

This file codes the interrupt tables registered for the interrupt handlers.

```
001 ;/************************************************************************
002 ;*
003 ;* Device     : SH-4A/SH7785
004 ;*
005 ;* File Name  : vect.inc
006 ;*
007 ;* Abstract   : Definition of Vector.
008 ;*
009 ;* History    : 1.00  (2010-09-30)  [Hardware Manual Revision : 1.00]
010 ;*
011 ;* Copyright(c) 2010 Renesas Electronics Corporation.
012 ;*               And Renesas Solutions Corp.,All Rights Reserved.
013 ;*
014 ;************************************************************************/
015 ;------------------------------------------------------------------------
016 ;                                                                       |
017 ;   FILE        :vect.inc                                               |
018 ;   DATE        :Thu, May 13, 2010                                      |
019 ;   DESCRIPTION :Definition of Vector                                    |
020 ;   CPU TYPE    :Other                                                  |
021 ;                                                                       |
022 ;   This file is generated by Renesas Project Generator (Ver.4.16).     |
023 ;                                                                       |
024 ;------------------------------------------------------------------------
025
026 ;<<VECTOR DATA START (POWER ON RESET)>>
027         ;H'000 Power On Reset (Hitachi-UDI RESET)
028         .global  _PowerON_Reset
029 ;<<VECTOR DATA END (POWER ON RESET)>>
030 ;<<VECTOR DATA START (MANUAL RESET)>>
031         ;H'020 Manual Reset
032         .global  _Manual_Reset
033 ;<<VECTOR DATA END (MANUAL RESET)>>
034         ;H'040 Data TLB miss exception(read)
035         .global     _INT_TLB_MISS_READ_EXP
036         ;H'060 Data TLB miss exception(write)
037         .global     _INT_TLB_MISS_WRITE_EXP
038         ;H'080 Initial page write exception
039         .global     _INT_TLB_INIT_PAGE_EXP
040         ;H'0A0 Data TLB protection violation exception (read)
041         .global     _INT_TLB_PROTECT_READ_EXP
042         ;H'0C0 Data TLB protection violation exception (write)
043         .global     _INT_TLB_PROTECT_WRITE_EXP
044         ;H'0E0 Data address error(read)
045         .global     _INT_ADR_ERROR_READ
046         ;H'100 Data address error(write)
047         .global     _INT_ADR_ERROR_WRITE
048         ;H'120 FPU exception
049         .global     _INT_FPU_EXP
050         ;H'140 Instruction TLB multiple-hit exception
051         .global     _TLB_Reset
052         ;H'160 Unconditional trap(TRAPA)
053         .global     _INT_TRAP
054         ;H'180 General illegal instruction exception
```

```
055         .global     _INT_ILLEGAL_INST_EXP
056         ;H'1A0 Slot illegal instruction exception
057         .global     _INT_ILLEGAL_SLOT_EXP
058         ;H'1C0 NMI
059         .global     _INT_NMI
060         ;H'1E0 USER_BREAK
061         .global     _INT_USER_BREAK
062            ; H'200     IRL30_LEVEL15_IRQ7
063            .global     _INT_IRL30_LEVEL15_IRQ7
064            ; H'220     IRL30_LEVEL14
065            .global     _INT_IRL30_LEVEL14
066            ; H'240     IRL30_LEVEL13_IRQ0
067            .global     _INT_IRL30_LEVEL13_IRQ0
068            ; H'260     IRL30_LEVEL12
069            .global     _INT_IRL30_LEVEL12
070            ; H'280     IRL30_LEVEL11_IRQ1
071            .global     _INT_IRL30_LEVEL11_IRQ1
072            ; H'2A0     IRL30_LEVEL10
073            .global     _INT_IRL30_LEVEL10
074            ; H'2C0     IRL30_LEVEL9_IRQ2
075            .global     _INT_IRL30_LEVEL9_IRQ2
076            ; H'2E0     IRL30_LEVEL8
077            .global     _INT_IRL30_LEVEL8
078            ; H'300     IRL30_LEVEL7_IRQ3
079            .global     _INT_IRL30_LEVEL7_IRQ3
080            ; H'320     IRL30_LEVEL6
081            .global     _INT_IRL30_LEVEL6
082            ; H'340     IRL30_LEVEL5_IRQ4
083            .global     _INT_IRL30_LEVEL5_IRQ4
084            ; H'360     IRL30_LEVEL4
085            .global     _INT_IRL30_LEVEL4
086            ; H'380     IRL30_LEVEL3_IRQ5
087            .global     _INT_IRL30_LEVEL3_IRQ5
088            ; H'3A0     IRL30_LEVEL2
089            .global     _INT_IRL30_LEVEL2
090            ; H'3C0     IRL30_LEVEL1_IRQ6
091            .global     _INT_IRL30_LEVEL1_IRQ6
092            ;H'3E0      Reserved
093            .global     _INT_Reserved3E0
094            ;H'400      Reserved
095            .global     _INT_Reserved400
096            ;H'420      Reserved
097            .global     _INT_Reserved420
098            ;H'440      Reserved
099            .global     _INT_Reserved440
100            ;H'460      Reserved
101            .global     _INT_Reserved460
102            ;H'480      Reserved
103            .global     _INT_Reserved480
104            ;H'4A0      Reserved
105            .global     _INT_Reserved4A0
106            ;H'4C0      Reserved
107            .global     _INT_Reserved4C0
108            ;H'4E0      Reserved
109            .global     _INT_Reserved4E0
110            ;H'500      Reserved
111            .global     _INT_Reserved500
```

```
112             ;H'520      Reserved
113             .global     _INT_Reserved520
114             ;H'540      Reserved
115             .global     _INT_Reserved540
116 ;WDT
117             ;H'560      WDT_ITI
118             .global     _INT_WDT_ITI
119 ;TMU-ch0
120             ;H'580      TMU_TUNI0
121             .global     _INT_TMU_TUNI0
122 ;TMU-ch1
123             ;H'5A0      TMU_TUNI1
124             .global     _INT_TMU_TUNI1
125 ;TMU-ch2
126             ;H'5C0      TMU_TUNI2
127             .global     _INT_TMU_TUNI2
128             ;H'5E0      TMU_TICPI2
129             .global     _INT_TMU_TICPI2
130 ;H-UDI
131             ;H'600      H-UDII
132             .global     _INT_H_UDII
133 ;DMAC(0)
134             ;H'620      DMINT0
135             .global     _INT_DMAC_DMINT0
136             ;H'640      DMINT1
137             .global     _INT_DMAC_DMINT1
138             ;H'660      DMINT2
139             .global     _INT_DMAC_DMINT2
140             ;H'680      DMINT3
141             .global     _INT_DMAC_DMINT3
142             ;H'6A0      DMINT4
143             .global     _INT_DMAC_DMINT4
144             ;H'6C0      DMINT5
145             .global     _INT_DMAC_DMINT5
146             ;H'6E0      DMAE0(ch0-5)
147             .global     _INT_DMAC_DMAE0
148 ;SCIF(0)
149             ;H'700      SCIF_ERI0
150             .global     _INT_SCIF_ERI0
151             ;H'720      SCIF_RXI0
152             .global     _INT_SCIF_RXI0
153             ;H'740      SCIF_BRI0
154             .global     _INT_SCIF_BRI0
155             ;H'760      SCIF_TXI0
156             .global     _INT_SCIF_TXI0
157 ;SCIF(1)
158             ;H'780      SCIF_ERI1
159             .global     _INT_SCIF_ERI1
160             ;H'7A0      SCIF_RXI1
161             .global     _INT_SCIF_RXI1
162             ;H'7C0      SCIF_BRI1
163             .global     _INT_SCIF_BRI1
164             ;H'7E0      SCIF_TXI1
165             .global     _INT_SCIF_TXI1
166             ;H'800      Reserved
167             .global     _INT_Reserved800
168             ;H'820      Reserved
```

```
169             .global     _INT_Reserved820
170             ;H'840      Reserved
171             .global     _INT_Reserved840
172             ;H'860      Reserved
173             .global     _INT_Reserved860
174 ;DMAC(1)
175             ;H'880      DMINT6
176             .global     _INT_DMAC_DMINT6
177             ;H'8A0      DMINT7
178             .global     _INT_DMAC_DMINT7
179             ;H'8C0      DMINT8
180             .global     _INT_DMAC_DMINT8
181             ;H'8E0      DMINT9
182             .global     _INT_DMAC_DMINT9
183             ;H'900      DMINT10
184             .global     _INT_DMAC_DMINT10
185             ;H'920      DMINT11
186             .global     _INT_DMAC_DMINT11
187             ;H'940      DMAE1(ch6-11)
188             .global     _INT_DMAC_DMAE1
189 ;HSPI
190             ;H'960      HSPI_SPII
191             .global     _INT_HSPI_SPII
192 ;SCIF(2)
193             ;H'980      SCIF_SCIFI2
194             .global     _INT_SCIF_SCIFI2
195 ;SCIF(3)
196             ;H'9A0      SCIF_SCIFI3
197             .global     _INT_SCIF_SCIFI3
198 ;SCIF(4)
199             ;H'9C0      SCIF_SCIFI4
200             .global     _INT_SCIF_SCIFI4
201 ;SCIF(5)
202             ;H'9E0      SCIF_SCIFI5
203             .global     _INT_SCIF_SCIFI5
204 ;PCIC(0)
205             ;H'A00      PCI_PCISERR
206             .global     _INT_PCI_PCISERR
207 ;PCIC(1)
208             ;H'A20      PCI_PCIINTA
209             .global     _INT_PCI_PCIINTA
210 ;PCIC(2)
211             ;H'A40      PCI_PCIINTB
212             .global     _INT_PCI_PCIINTB
213 ;PCIC(3)
214             ;H'A60      PCI_PCIINTC
215             .global     _INT_PCI_PCIINTC
216 ;PCIC(4)
217             ;H'A80      PCI_PCIINTD
218             .global     _INT_PCI_PCIINTD
219 ;PCIC(5)
220             ;H'AA0      PCI_PCIERR
221             .global     _INT_PCI_PCIERR
222             ;H'AC0      PCI_PCIPWD3_1
223             .global     _INT_PCI_PCIPWD3_1
224             ;H'AE0      PCI_PCIPWD0
225             .global     _INT_PCI_PCIPWD0
```

```
226 ;IRL(7-4)
227             ; H'B00     IRL74_LEVEL15
228             .global     _INT_IRL74_LEVEL15
229             ; H'B20     IRL74_LEVEL14
230             .global     _INT_IRL74_LEVEL14
231             ; H'B40     IRL74_LEVEL13
232             .global     _INT_IRL74_LEVEL13
233             ; H'B60     IRL74_LEVEL12
234             .global     _INT_IRL74_LEVEL12
235             ; H'B80     IRL74_LEVEL11
236             .global     _INT_IRL74_LEVEL11
237             ; H'BA0     IRL74_LEVEL10
238             .global     _INT_IRL74_LEVEL10
239             ; H'BC0     IRL74_LEVEL9
240             .global     _INT_IRL74_LEVEL9
241             ; H'BE0     IRL74_LEVEL8
242             .global     _INT_IRL74_LEVEL8
243             ; H'C00     IRL74_LEVEL7
244             .global     _INT_IRL74_LEVEL7
245             ; H'C20     IRL74_LEVEL6
246             .global     _INT_IRL74_LEVEL6
247             ; H'C40     IRL74_LEVEL5
248             .global     _INT_IRL74_LEVEL5
249             ; H'C60     IRL74_LEVEL4
250             .global     _INT_IRL74_LEVEL4
251             ; H'C80     IRL74_LEVEL3
252             .global     _INT_IRL74_LEVEL3
253             ; H'CA0     IRL74_LEVEL2
254             .global     _INT_IRL74_LEVEL2
255             ; H'CC0     IRL74_LEVEL1
256             .global     _INT_IRL74_LEVEL1
257 ;SIOF
258             ;H'CE0      SIOF_SIOFI
259             .global     _INT_SIOF_SIOFI
260 ;MMCIF
261             ;H'D00      MMCIF_FSTAT
262             .global     _INT_MMCIF_FSTAT
263             ;H'D20      MMCIF_TRAN
264             .global     _INT_MMCIF_TRAN
265             ;H'D40      MMCIF_ERR
266             .global     _INT_MMCIF_ERR
267             ;H'D60      MMCIF_FRDY
268             .global     _INT_MMCIF_FRDY
269 ;DU
270             ;H'D80      DU_DUI
271             .global     _INT_DU_DUI
272 ;GDTA
273             ;H'DA0      GDTA_GACLI
274             .global     _INT_GDTA_GACLI
275             ;H'DC0      GDTA_GAMCI
276             .global     _INT_GDTA_GAMCI
277             ;H'DE0      GDTA_GAERI
278             .global     _INT_GDTA_GAERI
279 ;TMU-ch3
280             ;H'E00      TMU_TUNI3
281             .global     _INT_TMU_TUNI3
282 ;TMU-ch4
```

```
283             ;H'E20      TMU_TUNI4
284             .global     _INT_TMU_TUNI4
285 ;TMU-ch5
286             ;H'E40      TMU_TUNI5
287             .global     _INT_TMU_TUNI5
288             ;H'E60      Reserved
289             .global     _INT_ReservedE60
290 ;SSI-ch0
291             ;H'E80      SSI_SSII0
292             .global     _INT_SSI_SSII0
293 ;SSI-ch1
294             ;H'EA0      SSI_SSII1
295             .global     _INT_SSI_SSII1
296 ;HAC-ch0
297             ;H'EC0      HAC_HACI0
298             .global     _INT_HAC_HACI0
299 ;HAC-ch1
300             ;H'EE0      HAC_HACI1
301             .global     _INT_HAC_HACI1
302 ;FLCTL
303             ;H'F00      FLCTL_FLSTE
304             .global     _INT_FLCTL_FLSTE
305             ;H'F20      FLCTL_FLTEND
306             .global     _INT_FLCTL_FLTEND
307             ;H'F40      FLCTL_FLTRQ0
308             .global     _INT_FLCTL_FLTRQ0
309             ;H'F60      FLCTL_FLTRQ1
310             .global     _INT_FLCTL_FLTRQ1
311 ;GPIO
312             ;H'F80      GPIO_GPIOI0
313             .global     _INT_GPIO_GPIIOI0
314             ;H'FA0      GPIO_GPIOI1
315             .global     _INT_GPIO_GPIIOI1
316             ;H'FC0      GPIO_GPIOI2
317             .global     _INT_GPIO_GPIIOI2
318             ;H'FE0      GPIO_GPIOI3
319             .global     _INT_GPIO_GPIIOI3
320
321
322
323 ;<<VECTOR DATA START (TBL RESET)>>
324         ;H'140 TBL Reset (DATA TBL Reset)
325         .global         _TBL_Reset
326 ;<<VECTOR DATA END (TBL RESET)>>
327
```

Sample program listing: intprg.c

The code in this file registers the interrupt processing functions in the interrupt handlers.

```
001 /***********************************************************************
002 * DISCLAIMER
003
004 * This software is supplied by Renesas Electronics Corporation. and is only
005 * intended for use with Renesas products. No other uses are authorized.
006
007 * This software is owned by Renesas Electronics Corporation. and is
008 * protected under all applicable laws, including copyright laws.
009
010 * only
011 * REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
012 * INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
013 * PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
014 * DISCLAIMED.
015
016 * TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
017 * ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
018 * FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
019 * FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
020 * AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
021
022 * Renesas reserves the right, without notice, to make changes to this
023 * software and to discontinue the availability of this software.
024 * By using this software, you agree to the additional terms and
025 * conditions found by accessing the following link:
026 * http://www.renesas.com/disclaimer
027 ***********************************************************************/
028 /* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.*/
029 /*""FILE COMMENT""************ Technical reference data ****************
030 * System Name  : SH7785 Sample Program
031 * File Name    : intprg.c
032 * Abstract     : SH7785 Initial Settings Sample Program
033 * Version      : Ver 1.00
034 * Device       : SH7785
035 * Tool-Chain   : High-performance Embedded Workshop (Version 4.07.00.007)
036 *              : C/C++ Compiler Package for SuperH Family (V.9.3.2.0)
037 * OS           : None
038 * H/W Platform : This is a sample program with examples of initialization for
039 * Description  : the SH7785 on the Renesas SH-4A board, model number R0P7785LC0011RL.
040 *              : This file is the intprg.src file converted to the C language format.
041 *              :
042 * Operation    :
043 * Limitation   :
044 *              :
045 ******************************************************************
046 * History      : 30.SEP.2010 Ver. 1.00 First Release
047 *""FILE COMMENT END""*********************************************/
048 #include <machine.h>
049 #include "iodefine.h"
050
051 /* --- RAM allocation variable declaration --- */
052
053 #pragma section IntPRG
054 /* H'040 Data TLB miss exception(read) */
```

```
055 void INT_TLB_MISS_READ_EXP(void)
056 {
057 }
058
059 /* H'060 Data TLB miss exception(write) */
060 void INT_TLB_MISS_WRITE_EXP(void)
061 {
062 }
063
064 /* H'080 Initial page write exception */
065 void INT_TLB_INIT_PAGE_EXP(void)
066 {
067 }
068
069 /* H'0A0 Data TLB protection violation exception (read) */
070 void INT_TLB_PROTECT_READ_EXP(void)
071 {
072 }
073
074 /* H'0C0 Data TLB protection violation exception (write) */
075 void INT_TLB_PROTECT_WRITE_EXP(void)
076 {
077 }
078
079 /* H'0E0 Data address error(read) */
080 void INT_ADR_ERROR_READ(void)
081 {
082 }
083
084 /* H'100 Data address error(write) */
085 void INT_ADR_ERROR_WRITE(void)
086 {
087 }
088
089 /* H'120 FPU exception */
090 void INT_FPU_EXP(void)
091 {
092 }
093
094 /* H'140 Instruction TLB multiple-hit exception */
095 void TLB_Reset(void)
096 {
097 }
098
099 /* H'160 Unconditional trap(TRAPA) */
100 void INT_TRAP(void)
101 {
102 }
103
104 /* H'180 General illegal instruction exception */
105 void INT_ILLEGAL_INST_EXP(void)
106 {
107 }
108
109 /* H'1A0 Slot illegal instruction exception */
110 void INT_ILLEGAL_SLOT_EXP(void)
111 {
```

```
112 }
113
114 /* H'1C0 NMI */
115 void INT_NMI(void)
116 {
117 }
118
119 /* H'1E0 USER_BREAK */
120 void INT_USER_BREAK(void)
121 {
122 }
123
124 /* H'200 IRL30_LEVEL15_IRQ7 */
125 void INT_IRL30_LEVEL15_IRQ7(void)
126 {
127 }
128
129 /* H'220 IRL30_LEVEL14 */
130 void INT_IRL30_LEVEL14(void)
131 {
132 }
133
134 /* H'240 IRL30_LEVEL13_IRQ0 */
135 void INT_IRL30_LEVEL13_IRQ0(void)
136 {
137 }
138
139 /* H'260 IRL30_LEVEL12 */
140 void INT_IRL30_LEVEL12(void)
141 {
142 }
143
144 /* H'280 IRL30_LEVEL11_IRQ1 */
145 void INT_IRL30_LEVEL11_IRQ1(void)
146 {
147 }
148
149 /* H'2A0 IRL30_LEVEL10 */
150 void INT_IRL30_LEVEL10(void)
151 {
152 }
153
154 /* H'2C0 IRL30_LEVEL9_IRQ2 */
155 void INT_IRL30_LEVEL9_IRQ2(void)
156 {
157 }
158
159 /* H'2E0 IRL30_LEVEL8 */
160 void INT_IRL30_LEVEL8(void)
161 {
162 }
163
164 /* H'300 IRL30_LEVEL7_IRQ3 */
165 void INT_IRL30_LEVEL7_IRQ3(void)
166 {
167 }
168
```

```
169 /* H'320 IRL30_LEVEL6 */
170 void INT_IRL30_LEVEL6(void)
171 {
172 }
173
174 /* H'340 IRL30_LEVEL5_IRQ4 */
175 void INT_IRL30_LEVEL5_IRQ4(void)
176 {
177 }
178
179 /* H'360 IRL30_LEVEL4 */
180 void INT_IRL30_LEVEL4(void)
181 {
182 }
183
184 /* H'380 IRL30_LEVEL3_IRQ5 */
185 void INT_IRL30_LEVEL3_IRQ5(void)
186 {
187 }
188
189 /* H'3A0 IRL30_LEVEL2 */
190 void INT_IRL30_LEVEL2(void)
191 {
192 }
193
194 /* H'3C0 IRL30_LEVEL1_IRQ6 */
195 void INT_IRL30_LEVEL1_IRQ6(void)
196 {
197 }
198
199 /* H'560 WDT overflow interrupt */
200 void INT_WDT_ITI(void)
201 {
202 }
203
204 /* H'580 TMU ch-0 underflow interrupt */
205 void INT_TMU_TUNI0(void)
206 {
207 }
208
209 /* H'5A0 TMU ch-1 underflow interrupt */
210 void INT_TMU_TUNI1(void)
211 {
212 }
213
214 /* H'5C0 TMU ch-2 underflow interrupt */
215 void INT_TMU_TUNI2(void)
216 {
217 }
218
219 /* H'5E0 TMU ch-2 input capture interrupt */
220 void INT_TMU_TICPI2(void)
221 {
222 }
223
224 /* H'600 H-UDI Update interrupt */
225 void INT_H_UDII(void)
```

```
226 {
227 }
228
229 /* H'620 DMAC0 interrupt */
230 void INT_DMAC_DMINT0(void)
231 {
232   irq_disable( _DMAC0 );
233   dmac0_irq();
234 }
235
236 /* H'640 DMAC1 interrupt */
237 void INT_DMAC_DMINT1(void)
238 {
239 }
240
241 /* H'660 DMAC2 interrupt */
242 void INT_DMAC_DMINT2(void)
243 {
244 }
245
246 /* H'680 DMAC3 interrupt */
247 void INT_DMAC_DMINT3(void)
248 {
249 }
250
251 /* H'6A0 DMAC4 interrupt */
252 void INT_DMAC_DMINT4(void)
253 {
254 }
255
256 /* H'6C0 DMAC5 interrupt */
257 void INT_DMAC_DMINT5(void)
258 {
259 }
260
261 /* H'6E0 DMAC 0-5 Address error interrupt */
262 void INT_DMAC_DMAE0(void)
263 {
264 }
265
266 /* H'700 SCIF0 error interrupt */
267 void INT_SCIF_ERI0(void)
268 {
269 }
270
271 /* H'720 SCIF0 Recive interrupt */
272 void INT_SCIF_RXI0(void)
273 {
274 }
275
276 /* H'740 SCIF0 Brake interrupt */
277 void INT_SCIF_BRI0(void)
278 {
279 }
280
281 /* H'760 SCIF0 Trans interrupt */
282 void INT_SCIF_TXI0(void)
```

```
283 {
284 }
285
286 /* H'780 SCIF1 error interrupt */
287 void INT_SCIF_ERI1(void)
288 {
289 }
290
291 /* H'7A0 SCIF1 Recive interrupt */
292 void INT_SCIF_RXI1(void)
293 {
294 }
295
296 /* H'7C0 SCIF1 Brake interrupt */
297 void INT_SCIF_BRI1(void)
298 {
299 }
300
301 /* H'7E0 SCIF1 Trans interrupt */
302 void INT_SCIF_TXI1(void)
303 {
304 }
305
306 /* H'880 DMAC6 interrupt */
307 void INT_DMAC_DMINT6(void)
308 {
309 }
310
311 /* H'8A0 DMAC7 interrupt */
312 void INT_DMAC_DMINT7(void)
313 {
314 }
315
316 /* H'8C0 DMAC8 interrupt */
317 void INT_DMAC_DMINT8(void)
318 {
319 }
320
321 /* H'8E0 DMAC9 interrupt */
322 void INT_DMAC_DMINT9(void)
323 {
324 }
325
326 /* H'900 DMAC11 interrupt */
327 void INT_DMAC_DMINT10(void)
328 {
329 }
330
331 /* H'920 DMAC11 interrupt */
332 void INT_DMAC_DMINT11(void)
333 {
334 }
335
336 /* H'940 DMAC 6-11 Address error interrupt */
337 void INT_DMAC_DMACE1(void)
338 {
339 }
```

```
340
341 /* H'960 HSPI interrupt */
342 void INT_HSPI_SPII(void)
343 {
344 }
345
346 /* H'980 SCIF2 interrupt */
347 void INT_SCIF_SCIFI2(void)
348 {
349 }
350
351 /* H'9A0 SCIF3 interrupt */
352 void INT_SCIF_SCIFI3(void)
353 {
354 }
355
356 /* H'9C0 SCIF4 interrupt */
357 void INT_SCIF_SCIFI4(void)
358 {
359 }
360
361 /* H'9E0 SCIF5 interrupt */
362 void INT_SCIF_SCIFI5(void)
363 {
364 }
365
366 /* H'A00 PCI SERR interrupt */
367 void INT_PCI_PCISERR(void)
368 {
369 }
370
371 /* H'A20 PCI INTA interrupt */
372 void INT_PCI_PCIINTA(void)
373 {
374 }
375
376 /* H'A40 PCI INTB interrupt */
377 void INT_PCI_PCIINTB(void)
378 {
379 }
380
381 /* H'A60 PCI INTC interrupt */
382 void INT_PCI_PCIINTC(void)
383 {
384 }
385
386 /* H'A80 PCI INTD interrupt */
387 void INT_PCI_PCIINTD(void)
388 {
389 }
390
391 /* H'AA0 PCI ERR interrupt */
392 void INT_PCI_PCIERR(void)
393 {
394 }
395
396 /* H'AC0 PCI PWD3-1 interrupt */
```

```
397 void INT_PCI_PCIPWD3_1(void)
398 {
399 }
400
401 /* H'AE0 PCI PWD0 interrupt */
402 void INT_PCI_PCIPWD0(void)
403 {
404 }
405
406 /* H'B00 IRL74_LEVEL15 interrupt */
407 void INT_IRL74_LEVEL15(void)
408 {
409 }
410
411 /* H'B20 IRL74_LEVEL14 interrupt */
412 void INT_IRL74_LEVEL14(void)
413 {
414 }
415
416 /* H'B40 IRL74_LEVEL13 interrupt */
417 void INT_IRL74_LEVEL13(void)
418 {
419 }
420
421 /* H'B60 IRL74_LEVEL12 interrupt */
422 void INT_IRL74_LEVEL12(void)
423 {
424 }
425
426 /* H'B80 IRL74_LEVEL11 interrupt */
427 void INT_IRL74_LEVEL11(void)
428 {
429 }
430
431 /* H'BA0 IRL74_LEVEL10 interrupt */
432 void INT_IRL74_LEVEL10(void)
433 {
434 }
435
436 /* H'BC0 IRL74_LEVEL9 interrupt */
437 void INT_IRL74_LEVEL9(void)
438 {
439 }
440
441 /* H'BE0 IRL74_LEVEL8 interrupt */
442 void INT_IRL74_LEVEL8(void)
443 {
444 }
445
446 /* H'C00 IRL74_LEVEL7 interrupt */
447 void INT_IRL74_LEVEL7(void)
448 {
449 }
450
451 /* H'C20 IRL74_LEVEL6 interrupt */
452 void INT_IRL74_LEVEL6(void)
453 {
```

```
454 }
455
456 /* H'C40 IRL74_LEVEL5 interrupt */
457 void INT_IRL74_LEVEL5(void)
458 {
459 }
460
461 /* H'C60 IRL74_LEVEL4 interrupt */
462 void INT_IRL74_LEVEL4(void)
463 {
464 }
465
466 /* H'C80 IRL74_LEVEL3 interrupt */
467 void INT_IRL74_LEVEL3(void)
468 {
469 }
470
471 /* H'CA0 IRL74_LEVEL2 interrupt */
472 void INT_IRL74_LEVEL2(void)
473 {
474 }
475
476 /* H'CC0 IRL74_LEVEL1 interrupt */
477 void INT_IRL74_LEVEL1(void)
478 {
479 }
480
481 /* H'CE0 SIOF interrupt */
482 void INT_SIOF_SIOFI(void)
483 {
484 }
485
486 /* H'D00 MMCIF FIFO Empty/Full interrupt */
487 void INT_MMCIF_FSTAT(void)
488 {
489 }
490
491 /* H'D20 MMCIF Transfer end interrupt */
492 void INT_MMCIF_TRAN(void)
493 {
494 }
495
496 /* H'D40 MMCIF Error interrupt */
497 void INT_MMCIF_ERR(void)
498 {
499 }
500
501 /* H'D60 MMCIF FIFO Ready interrupt */
502 void INT_MMCIF_FRDY(void)
503 {
504 }
505
506 /* H'D80 DU interrupt */
507 void INT_DU_DUI(void)
508 {
509 }
510
```

```
511 /* H'DA0 GDTA CL end interrupt */
512 void INT_GDTA_GACLI(void)
513 {
514 }
515
516 /* H'DC0 GDTA MC end interrupt */
517 void INT_GDTA_GAMCI(void)
518 {
519 }
520
521 /* H'DE0 GDTA Error interrupt */
522 void INT_GDTA_GAERI(void)
523 {
524 }
525
526 /* H'E00 TMU ch-3 underflow interrupt */
527 void INT_TMU_TUNI3(void)
528 {
529 }
530
531 /* H'E20 TMU ch-4 underflow interrupt */
532 void INT_TMU_TUNI4(void)
533 {
534 }
535
536 /* H'E40 TMU ch-5 underflow interrupt */
537 void INT_TMU_TUNI5(void)
538 {
539 }
540
541 /* H'E80 SSI ch-0 interrupt */
542 void INT_SSI_SSII0(void)
543 {
544 }
545
546 /* H'EA0 SSI ch-1 interrupt */
547 void INT_SSI_SSII1(void)
548 {
549 }
550
551 /* H'EC0 HAC ch-0 interrupt */
552 void INT_HAC_HACI0(void)
553 {
554 }
555
556 /* H'EE0 HAC ch-1 interrupt */
557 void INT_HAC_HACI1(void)
558 {
559 }
560
561 /* H'F00 FLCTL error interrupt */
562 void INT_FLCTL_FLSTE(void)
563 {
564 }
565
566 /* H'F20 FLCTL Transfer end interrupt */
567 void INT_FLCTL_FLTEND(void)
```

```
568 {
569 }
570
571 /* H'F40 FLCTL FIFO0 request interrupt */
572 void INT_FLCTL_FLTRQ0(void)
573 {
574 }
575
576 /* H'F60 FLCTL FIFO1 request interrupt */
577 void INT_FLCTL_FLTRQ1(void)
578 {
579 }
580
581 /* H'F80 GPIO0 interrupt */
582 void INT_GPIO_GPIIOI0(void)
583 {
584 }
585
586 /* H'FA0 GPIO1 interrupt */
587 void INT_GPIO_GPIIOI1(void)
588 {
589 }
590
591 /* H'FC0 GPIO2 interrupt */
592 void INT_GPIO_GPIIOI2(void)
593 {
594 }
595
596 /* H'FE0 GPIO3 interrupt */
597 void INT_GPIO_GPIIOI3(void)
598 {
599 }
600
601 void INT_Reserved(void)
602 {
603 }
```

Sample program listing: lowlevelinit src

The code in this file initializes the LBSC and DBSC2.

```
001 ;/*********************************************************************
002 ;* DISCLAIMER
003 ;
004 ;* This software is supplied by Renesas Electronics Corporation. and is only
005 ;* intended for use with Renesas products. No other uses are authorized.
006 ;
007 ;* This software is owned by Renesas Electronics Corporation. and is
008 ;* protected under all applicable laws, including copyright laws.
009 ;
010 ;* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
011 ;* REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
012 ;* INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
013 ;* PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
014 ;* DISCLAIMED.
015 ;
016 ;* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
017 ;* ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
018 ;* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
019 ;* FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
020 ;* AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
021 ;
022 ;* Renesas reserves the right, without notice, to make changes to this
023 ;* software and to discontinue the availability of this software.
024 ;* By using this software, you agree to the additional terms and
025 ;* conditions found by accessing the following link:
026 ;* http://www.renesas.com/disclaimer
027 ;*********************************************************************/
028 ;/* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.*/
029 ;/*""FILE COMMENT""************ Technical reference data ****************
030 ;* System Name  : SH7785 Sample Program
031 ;* File Name    : lowlevelinit.src
032 ;* Abstract     : SH7785 Initial Settings Sample Program
033 ;* Version      : Ver 1.00
034 ;* Device       : SH7785
035 ;* Tool-Chain   : High-performance Embedded Workshop (Version 4.04.01.001)
036 ;*              : C/C++ Compiler Package for SuperH Family
(V.9.02release00)
037 ;* OS           : None
038 ;* H/W Platform : This is a sample program with examples of initialization for
039 ;* Description  : the SH7785 on the Renesas SH-4A board, model number R0P7785LC0011RL.
040 ;*              :
041 ;*              :
042 ;* Operation    :
043 ;* Limitation   :
044 ;*              :
045 ;*********************************************************************
046 ;* History      : 30.SEP.2010 Ver. 1.00 First Release
047 ;*""FILE COMMENT END""*************************************************/
048 ;---------------------------------------------------------------------
049 ;                                                                     |
050 ;   FILE        :lowlevelinit.src                                     |
051 ;   DATE        :Thu, May 13, 2010                                    |
052 ;   DESCRIPTION :DBSC2/LBSC Initial                                    |
053 ;   CPU TYPE    :Other                                                |
```

```
;                                                                              |
;   This file is generated by Renesas Project Generator (Ver.4.16).            |
;                                                                              |
;-------------------------------------------------------------------------------

 .macro  write32  addr, data
  mov.l \addr, r1
  mov.l \data, r0
  mov.l r0, @r1
 .endm

 .macro  write16  addr, data
  mov.l \addr,r1
  mov.l \data,r0
  mov.w r0, @r1
 .endm

 .macro  write8  addr, data
  mov.l \addr ,r1
  mov.l \data ,r0
  mov.b r0, @r1
 .endm

 .macro wait_timer  time
  mov.l \time ,r3
loop\@:
  nop
  tst r3, r3
  bf/s  loop\@
  dt  r3
 .endm

 .include  "lowlevelinit.inc"

  .export _lowlevel_init

_lowlevel_init:

;  /*------- LBSC -------*/
  write32 #MMSELR_A, #MMSELR_D

;  /*------- DBSC2 -------*/
  write32 #DBSC2_DBCONF_A, #DBSC2_DBCONF_D
  write32 #DBSC2_DBTR0_A,  #DBSC2_DBTR0_D
  write32 #DBSC2_DBTR1_A,  #DBSC2_DBTR1_D
  write32 #DBSC2_DBTR2_A,  #DBSC2_DBTR2_D
  write32 #DBSC2_DBFREQ_A, #DBSC2_DBFREQ_D1
  write32 #DBSC2_DBFREQ_A, #DBSC2_DBFREQ_D2
  wait_timer  #WAIT_200US

  write32 #DBSC2_DBDICODTOCD_A,  #DBSC2_DBDICODTOCD_D
  write32 #DBSC2_DBCMDCNT_A, #DBSC2_DBCMDCNT_D_CKE_H
  wait_timer  #3333
  write32 #DBSC2_DBCMDCNT_A, #DBSC2_DBCMDCNT_D_PALL
  write32 #DBSC2_DBMRCNT_A,  #DBSC2_DBMRCNT_D_EMRS2
  write32 #DBSC2_DBMRCNT_A,  #DBSC2_DBMRCNT_D_EMRS3
  write32 #DBSC2_DBMRCNT_A,  #DBSC2_DBMRCNT_D_EMRS1_1
```

```
111   write32 #DBSC2_DBMRCNT_A,  #DBSC2_DBMRCNT_D_MRS_1
112   write32 #DBSC2_DBCMDCNT_A, #DBSC2_DBCMDCNT_D_PALL
113   write32 #DBSC2_DBCMDCNT_A, #DBSC2_DBCMDCNT_D_REF
114   write32 #DBSC2_DBCMDCNT_A, #DBSC2_DBCMDCNT_D_REF
115   write32 #DBSC2_DBMRCNT_A,  #DBSC2_DBMRCNT_D_MRS_2
116   wait_timer  #WAIT_200US
117
118   write32 #DBSC2_DBMRCNT_A,  #DBSC2_DBMRCNT_D_EMRS1_2
119   write32 #DBSC2_DBMRCNT_A,  #DBSC2_DBMRCNT_D_EMRS1_1
120
121   write32 #DBSC2_DBEN_A,   #DBSC2_DBEN_D
122   write32 #DBSC2_DBRFCNT1_A, #DBSC2_DBRFCNT1_D
123   write32 #DBSC2_DBRFCNT2_A, #DBSC2_DBRFCNT2_D
124   write32 #DBSC2_DBRFCNT0_A, #DBSC2_DBRFCNT0_D
125
126 ;  /*------- LBSC -------*/
127   write32 #BCR_A,    #BCR_D
128
129 ;  mov.l #PASCR_A, r0
130 ;  mov.l @r0, r2
131 ;  mov.l #PASCR_32BIT_MODE, r1
132 ;  tst r1, r2
133 ;  bt  lbsc_29bit
134
135  .aifdefCONFIG_CS0
136   write32 #CS0_BCR_A, #CS0_BCR_D
137   write32 #CS0_WCR_A, #CS0_WCR_D
138  .aendi
139
140  .aifdefCONFIG_CS1
141   write32 #CS1_BCR_A, #CS1_BCR_D
142   write32 #CS1_WCR_A, #CS1_WCR_D
143  .aendi
144
145  .aifdefCONFIG_CS2
146   write32 #CS2_BCR_A, #CS2_BCR_D
147   write32 #CS2_WCR_A, #CS2_WCR_D
148  .aendi
149
150  .aifdefCONFIG_CS3
151   write32 #CS3_BCR_A, #CS3_BCR_D
152   write32 #CS3_WCR_A, #CS3_WCR_D
153  .aendi
154
155  .aifdefCONFIG_CS4
156   write32 #CS4_BCR_A, #CS4_BCR_D
157   write32 #CS4_WCR_A, #CS4_WCR_D
158  .aendi
159
160  .aifdefCONFIG_CS5
161   write32 #CS5_BCR_A, #CS5_BCR_D
162   write32 #CS5_WCR_A, #CS5_WCR_D
163  .aendi
164
165  .aifdefCONFIG_CS6
166   write32 #CS6_BCR_A, #CS6_BCR_D
167   write32 #CS6_WCR_A, #CS6_WCR_D
```

```
168  .aendi
169
170   bra lbsc_end
171    nop
172
173 lbsc_end:
174  .aifdefCONFIG_SH_32BIT
175 ; /*------- set PMB -------*/
176   write32 #PASCR_A,  #PASCR_29BIT_D
177   write32 #MMUCR_A,  #MMUCR_D
178
179 ;  /*************************************************************
180 ;    * ent  virt    phys    v sz  c wt
181 ;    * 0  0xa0000000  0x00000000  1 64M 0 0
182 ;    * 1  0xa4000000  0x04000000  1 16M 0 0
183 ;    * 2  0xa6000000  0x08000000  1 16M 0 0
184 ;    * 9  0x88000000  0x48000000  1 128M  1 1
185 ;    * 10 0x90000000  0x50000000  1 128M  1 1
186 ;    * 11 0x98000000  0x58000000  1 128M  1 1
187 ;    * 13 0xa8000000  0x48000000  1 128M  0 0
188 ;    * 14 0xb0000000  0x50000000  1 128M  0 0
189 ;    * 15 0xb8000000  0x58000000  1 128M  0 0
190
191   write32 #PMB_ADDR_FLASH_A, #PMB_ADDR_FLASH_D
192   write32 #PMB_DATA_FLASH_A, #PMB_DATA_FLASH_D
193   write32 #PMB_ADDR_CPLD_A,      #PMB_ADDR_CPLD_D
194   write32 #PMB_DATA_CPLD_A,      #PMB_DATA_CPLD_D
195   write32 #PMB_ADDR_USB_A,    #PMB_ADDR_USB_D
196   write32 #PMB_DATA_USB_A,    #PMB_DATA_USB_D
197   write32 #PMB_ADDR_DDR_C1_A,#PMB_ADDR_DDR_C1_D
198   write32 #PMB_DATA_DDR_C1_A,#PMB_DATA_DDR_C1_D
199   write32 #PMB_ADDR_DDR_C2_A,#PMB_ADDR_DDR_C2_D
200   write32 #PMB_DATA_DDR_C2_A,#PMB_DATA_DDR_C2_D
201   write32 #PMB_ADDR_DDR_C3_A,#PMB_ADDR_DDR_C3_D
202   write32 #PMB_DATA_DDR_C3_A,#PMB_DATA_DDR_C3_D
203   write32 #PMB_ADDR_DDR_N1_A,#PMB_ADDR_DDR_N1_D
204   write32 #PMB_DATA_DDR_N1_A,#PMB_DATA_DDR_N1_D
205   write32 #PMB_ADDR_DDR_N2_A,#PMB_ADDR_DDR_N2_D
206   write32 #PMB_DATA_DDR_N2_A,#PMB_DATA_DDR_N2_D
207   write32 #PMB_ADDR_DDR_N3_A,#PMB_ADDR_DDR_N3_D
208   write32 #PMB_DATA_DDR_N3_A,#PMB_DATA_DDR_N3_D
209
210   write32 #PASCR_A,  #PASCR_INIT
211   mov.l #DUMMY_ADDR, r0
212   icbi  @r0
213  .aendi
214
215   write32 #CCR_A,  #CCR_D
216   rts
217   nop
218
219
220
221  .end
```

Sample program listing: lowlevelinit.inc

The code in this file provides an example of LBSC and DBSC2 initialization.

```
001
002 ; The choice of the area to initialize.
003 ;-----------------------------------------------
004 ;CONFIG_SH_32BIT: .define "" ;Address 32Bit Mode
005
006
007 CONFIG_CS0:.define "" ;Initial CS0
008 CONFIG_CS1:.define "" ;Initial CS1
009 ;
010  .aifdef CONFIG_SH_32BIT
011 CONFIG_CS2:.define "" ;Initial CS2
012 CONFIG_CS3:.define "" ;Initial CS3
013  .aendi
014
015 CONFIG_CS4:.define "" ;Initial CS4
016 CONFIG_CS5:.define "" ;Initial CS5
017 CONFIG_CS6:.define "" ;Initial CS6
018 ;CONFIG_CS5_PCMCIA: .define ""
019 ;CONFIG_CS6_PCMCIA: .define ""
020
021 ;-----------------------------------------------
022
023 ;/*------- DBSC2 -------*/
024 DBSC2_BASE:             .equ  H'fe800000
025 DBSC2_DBSTATE_A:        .equ  DBSC2_BASE + H'0c
026 DBSC2_DBEN_A:          .equ  DBSC2_BASE + H'10
027 DBSC2_DBCMDCNT_A:      .equ  DBSC2_BASE + H'14
028 DBSC2_DBCONF_A:           .equ  DBSC2_BASE + H'20
029 DBSC2_DBTR0_A:          .equ  DBSC2_BASE + H'30
030 DBSC2_DBTR1_A:          .equ  DBSC2_BASE + H'34
031 DBSC2_DBTR2_A:          .equ  DBSC2_BASE + H'38
032 DBSC2_DBRFCNT0_A:       .equ  DBSC2_BASE + H'40
033 DBSC2_DBRFCNT1_A:       .equ  DBSC2_BASE + H'44
034 DBSC2_DBRFCNT2_A:       .equ  DBSC2_BASE + H'48
035 DBSC2_DBRFSTS_A:        .equ  DBSC2_BASE + H'4c
036 DBSC2_DBFREQ_A:            .equ  DBSC2_BASE + H'50
037 DBSC2_DBDICODTOCD_A:    .equ  DBSC2_BASE + H'54
038 DBSC2_DBMRCNT_A:        .equ  DBSC2_BASE + H'60
039 DDR_DUMMY_ACCESS_A:        .equ  H'40000000
040
041 DBSC2_DBCONF_D:            .equ  H'00630002
042 DBSC2_DBTR0_D:             .equ  H'050b1f04
043 DBSC2_DBTR1_D:             .equ  H'00040204
044 DBSC2_DBTR2_D:             .equ  H'02100308
045 DBSC2_DBFREQ_D1:        .equ  H'00000000
046 DBSC2_DBFREQ_D2:        .equ  H'00000100
047 DBSC2_DBDICODTOCD_D:    .equ  H'000f0907
048
049 DBSC2_DBMRCNT_D_EMRS2:     .equ  H'00020000
050 DBSC2_DBMRCNT_D_EMRS3:     .equ  H'00030000
051 DBSC2_DBMRCNT_D_EMRS1_1:   .equ  H'00010006
052 DBSC2_DBMRCNT_D_EMRS1_2:   .equ  H'00010386
053 DBSC2_DBMRCNT_D_MRS_1:     .equ  H'00000952
054 DBSC2_DBMRCNT_D_MRS_2:     .equ  H'00000852
```

```
055
056 DBSC2_DBPDCNT0_D3:         .equ  H'00000080
057 DBSC2_DBRFCNT1_D:       .equ  H'00000926
058 DBSC2_DBRFCNT2_D:       .equ  H'00fe00fe
059 DBSC2_DBRFCNT0_D:       .equ  H'00010000
060
061 DBSC2_DBCMDCNT_D_CKE_H:    .equ  H'00000003
062 DBSC2_DBCMDCNT_D_PALL:     .equ  H'00000002
063 DBSC2_DBCMDCNT_D_REF:      .equ  H'00000004
064
065 DBSC2_DBEN_D:           .equ  H'00000001
066
067 WAIT_200US:             .equ  33333
068
069 ;/*------- LBSC -------*/
070 PASCR_A:             .equ  H'ff000070
071 PASCR_32BIT_MODE:       .equ  H'80000000  ;/* check booting mode */
072
073 MMSELR_A:               .equ  H'fc400020
074 BCR_A:                  .equ  H'FF801000
075 CS0_BCR_A:              .equ  H'FF802000
076 CS1_BCR_A:              .equ  H'FF802010
077 CS2_BCR_A:              .equ  H'FF802020
078 CS3_BCR_A:              .equ  H'FF802030
079 CS4_BCR_A:              .equ  H'FF802040
080 CS5_BCR_A:              .equ  H'FF802050
081 CS6_BCR_A:              .equ  H'FF802060
082 CS0_WCR_A:              .equ  H'FF802008
083 CS1_WCR_A:              .equ  H'FF802018
084 CS2_WCR_A:              .equ  H'FF802028
085 CS3_WCR_A:              .equ  H'FF802038
086 CS4_WCR_A:              .equ  H'FF802048
087 CS5_WCR_A:              .equ  H'FF802058
088 CS6_WCR_A:              .equ  H'FF802068
089 CS5_PCMCIA_A:           .equ  H'FF802070
090 CS6_PCMCIA_A:           .equ  H'FF802080
091
092  .aifdefCONFIG_SH_32BIT
093 MMSELR_D:               .equ  H'a5a50005
094  .aelse
095 MMSELR_D:               .equ  H'a5a50002
096  .aendi
097
098 BCR_D:                  .equ  H'80000003
099 CS0_BCR_D:              .equ  H'22222340
100 CS0_WCR_D:              .equ  H'00111118
101 CS1_BCR_D:              .equ  H'11111100
102 CS1_WCR_D:              .equ  H'33333303
103 CS4_BCR_D:              .equ  H'11111300
104 CS4_WCR_D:              .equ  H'00101012
105
106 ;/* USB setting : 32bit mode = CS2, 29bit mode = CS5 */
107  .aifdef CONFIG_SH_32BIT
108 CS2_BCR_D:              .equ  H'11111200
109 CS2_WCR_D:              .equ  H'00020004
110  .aelse
111 CS5_BCR_D:              .equ  H'11111200
```

```
112 CS5_WCR_D:              .equ  H'00020004
113  .aendi
114
115 ;/* SD setting : 32bit mode = CS3, 29bit mode = CS6 */
116  .aifdef CONFIG_SH_32BIT
117 CS3_BCR_D:              .equ  H'00000300
118 CS3_WCR_D:              .equ  H'00030108
119  .aelse
120 CS6_BCR_D:              .equ  H'00000300
121 CS6_WCR_D:              .equ  H'00030108
122  .aendi
123
124 ;/* I2C setting : 32bit mode = CS5, 29bit mode = CS1(already setting) */
125  .aifdef CONFIG_SH_32BIT
126 CS5_BCR_D:              .equ  H'11111100
127 CS5_WCR_D:              .equ  H'00000003
128  .aendi
129
130 ;/* Chache */
131 CCR_A:                  .equ  H'ff00001c
132 CCR_D:                  .equ  H'0000090b
133
134 ;PMB
135  .aifdef CONFIG_SH_32BIT
136 PMB
137 PMB_ADDR_ARRAY:            .equ  H'f6100000
138 PMB_ADDR_ENTRY:            .equ  8
139 PMB_VPN:             .equ  24
140
141 PMB_DATA_ARRAY:            .equ  H'f7100000
142 PMB_DATA_ENTRY:            .equ  8
143 PMB_PPN:             .equ  24
144 PMB_UB:                .equ  9  ;/* Buffered write */
145 PMB_V:                 .equ  8  ;/* Valid */
146 PMB_SZ1:             .equ  7  ;/* Page size (upper bit) */
147 PMB_SZ0:             .equ  4  ;/* Page size (lower bit) */
148 PMB_C:                 .equ  3  ;/* Cacheability */
149 PMB_WT:                .equ  0  ;/* Write-through */
150
151 ;#define PMB_ADDR_BASE(entry)  (PMB_ADDR_ARRAY + (entry << PMB_ADDR_ENTRY))
152 ;#define PMB_DATA_BASE(entry)  (PMB_DATA_ARRAY + (entry << PMB_DATA_ENTRY))
153 ;#define mk_pmb_addr_val(vpn)  ((vpn << PMB_VPN))
154 ;#define mk_pmb_data_val(ppn, ub, v, sz1, sz0, c, wt)  \
155 ;        ((ppn << PMB_PPN) | (ub << PMB_UB) | \
156 ;         (v << PMB_V) | (sz1 << PMB_SZ1) | \
157 ;         (sz0 << PMB_SZ0) | (c << PMB_C) | \
158 ;         (wt << PMB_WT))
159
160 PMB_ADDR_FLASH_A:          .equ PMB_ADDR_ARRAY + 0
161 PMB_ADDR_CPLD_A:        .equ PMB_ADDR_ARRAY + 1 <<  PMB_ADDR_ENTRY
162 PMB_ADDR_USB_A:         .equ PMB_ADDR_ARRAY + 2 <<  PMB_ADDR_ENTRY
163 PMB_ADDR_DDR_C1_A:         .equ PMB_ADDR_ARRAY + 9 <<  PMB_ADDR_ENTRY
164 PMB_ADDR_DDR_C2_A:         .equ PMB_ADDR_ARRAY + 10 <<  PMB_ADDR_ENTRY
165 PMB_ADDR_DDR_C3_A:         .equ PMB_ADDR_ARRAY + 11 <<  PMB_ADDR_ENTRY
166 PMB_ADDR_DDR_N1_A:         .equ PMB_ADDR_ARRAY + 13 <<  PMB_ADDR_ENTRY
167 PMB_ADDR_DDR_N2_A:         .equ PMB_ADDR_ARRAY + 14 <<  PMB_ADDR_ENTRY
168 PMB_ADDR_DDR_N3_A:         .equ PMB_ADDR_ARRAY + 15 <<  PMB_ADDR_ENTRY
```

```
169
170 PMB_ADDR_FLASH_D:       .equ H'a0 << PMB_VPN
171 PMB_ADDR_CPLD_D:        .equ H'a4 << PMB_VPN
172 PMB_ADDR_USB_D:            .equ H'a6 << PMB_VPN
173 PMB_ADDR_DDR_C1_D:         .equ H'88 << PMB_VPN
174 PMB_ADDR_DDR_C2_D:         .equ H'90 << PMB_VPN
175 PMB_ADDR_DDR_C3_D:         .equ H'98 << PMB_VPN
176 PMB_ADDR_DDR_N1_D:         .equ H'a8 << PMB_VPN
177 PMB_ADDR_DDR_N2_D:         .equ H'b0 << PMB_VPN
178 PMB_ADDR_DDR_N3_D:         .equ H'b8 << PMB_VPN
179
180 PMB_DATA_FLASH_A:       .equ PMB_DATA_ARRAY + 0
181 PMB_DATA_CPLD_A:        .equ PMB_DATA_ARRAY + 1 << PMB_DATA_ENTRY
182 PMB_DATA_USB_A:            .equ PMB_DATA_ARRAY + 2 << PMB_DATA_ENTRY
183 PMB_DATA_DDR_C1_A:         .equ PMB_DATA_ARRAY + 9 << PMB_DATA_ENTRY
184 PMB_DATA_DDR_C2_A:         .equ PMB_DATA_ARRAY + 10 << PMB_DATA_ENTRY
185 PMB_DATA_DDR_C3_A:         .equ PMB_DATA_ARRAY + 11 << PMB_DATA_ENTRY
186 PMB_DATA_DDR_N1_A:         .equ PMB_DATA_ARRAY + 13 << PMB_DATA_ENTRY
187 PMB_DATA_DDR_N2_A:         .equ PMB_DATA_ARRAY + 14 << PMB_DATA_ENTRY
188 PMB_DATA_DDR_N3_A:         .equ PMB_DATA_ARRAY + 15 << PMB_DATA_ENTRY
189
190 ;/*             ppn   ub v s1 s0  c  wt */
191 PMB_DATA_FLASH_D:       .equ H'00 << PMB_PPN | 1 << PMB_UB | 1 << PMB_V | 0
<< PMB_SZ1 | 1 << PMB_SZ0 | 0 << PMB_C | 1 << PMB_WT
192 PMB_DATA_CPLD_D:        .equ H'04 << PMB_PPN | 1 << PMB_UB | 1 << PMB_V | 0
<< PMB_SZ1 | 0 << PMB_SZ0 | 0 << PMB_C | 1 << PMB_WT
193 PMB_DATA_USB_D:            .equ H'08 << PMB_PPN | 1 << PMB_UB | 1 << PMB_V
| 0 << PMB_SZ1 | 0 << PMB_SZ0 | 0 << PMB_C | 1 << PMB_WT
194 PMB_DATA_DDR_C1_D:         .equ H'48 << PMB_PPN | 0 << PMB_UB | 1 << PMB_V
| 1 << PMB_SZ1 | 0 << PMB_SZ0 | 1 << PMB_C | 1 << PMB_WT
195 PMB_DATA_DDR_C2_D:         .equ H'50 << PMB_PPN | 0 << PMB_UB | 1 << PMB_V
| 1 << PMB_SZ1 | 0 << PMB_SZ0 | 1 << PMB_C | 1 << PMB_WT
196 PMB_DATA_DDR_C3_D:         .equ H'58 << PMB_PPN | 0 << PMB_UB | 1 << PMB_V
| 1 << PMB_SZ1 | 0 << PMB_SZ0 | 1 << PMB_C | 1 << PMB_WT
197 PMB_DATA_DDR_N1_D:         .equ H'48 << PMB_PPN | 1 << PMB_UB | 1 << PMB_V
| 1 << PMB_SZ1 | 0 << PMB_SZ0 | 0 << PMB_C | 1 << PMB_WT
198 PMB_DATA_DDR_N2_D:         .equ H'50 << PMB_PPN | 1 << PMB_UB | 1 << PMB_V
| 1 << PMB_SZ1 | 0 << PMB_SZ0 | 0 << PMB_C | 1 << PMB_WT
199 PMB_DATA_DDR_N3_D:         .equ H'58 << PMB_PPN | 1 << PMB_UB | 1 << PMB_V
| 1 << PMB_SZ1 | 0 << PMB_SZ0 | 0 << PMB_C | 1 << PMB_WT
200
201 DUMMY_ADDR:             .equ H'a0000000
202 PASCR_29BIT_D:          .equ H'00000000
203 PASCR_INIT:             .equ H'80000080  ;/* check booting mode */
204 MMUCR_A:            .equ H'ff000010
205 MMUCR_D:            .equ H'00000004  ;/* clear ITLB */
206
207  .aendi
```

6. Reference Documents

- Software Manual
  SH4-A Software Manual (REJ09B0003)
  (The latest version can be downloaded from the Renesas Electronics Web site.)
- Hardware Manual
  SH7785 Group Hardware Manual (REJ09B0261)
  (The latest version can be downloaded from the Renesas Electronics Web site.)

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.01	Mar.17.11	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

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.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.0