To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

SH7206 Group

Example of SH7206 Initial Configuration

Introduction

This application note presents an example of initial settings to be made when the SH7206 is started up.

Target Device

SH7206

Contents

1. Overview

1.1 Specifications

- Initial settings for the clock pulse generator (CPG), bus state controller (BSC), pin function controller (PFC), and cache are made after the reset state is exited.

1.2 MCU Functions Used

- Clock pulse generator (CPG)
- Bus state controller (BSC)
- Pin function controller (PFC)
- Cache

1.3 Conditions for Application

- MCU: SH7206 (R5S72060)
- Operating frequency: Internal clock: 200 MHz
  Bus clock: 66.67 MHz
  Peripheral clock: 33.33 MHz
- Area 0 bus width: 16 bits (MD0 pin = low, MD2 pin = high)
- Clock operating mode: Mode 2 (MD_CLK0 pin = low, MD_CLK2 pin = low)
- C compiler: SuperH RISC Engine Family C/C++ Compiler Package: version 9.00
  (from Renesas Technology Corp.)
- Compiler options: Default setting of HEW (-cpu = sh2a -debug –gbr = auto -global_volatile = 0
  -opt_range = all -infinite_loop = 0 -del_vacant_loop = 0
  -struct_alloc = 1)

1.4 Related Application Notes

Please refer to the following application notes in combination with this one.

- SH7206 application note, "Example of Setting the CPG to Change the Operating Frequency"
- SH7206 application note, "Example of BSC SDRAM Interface Setting (32-bit bus)"
- SH7206 application note, "Example of BSC Flash Memory Connection "
- SH7206 application note, "Example of Cache Memory Setting"

2. Description of Sample Application

The sample initial configuration program described in this application note is used for the applications of other SH7206 application notes.

2.1 Description of Sample Programs

The initial setting program includes the following two source programs.

- resetprog.c
- hwsetup.c

resetprog.c is created based on the file for the initialization function that is automatically generated by the HEW, and contains description of function PowerON_Reset_PC, which is to be registered with the reset vector. PowerON_Reset_PC is the first function that is executed after the chip exits from the reset state.

hwsetup.c contains description of function HardwareSetup, which is called from function PowerON_Reset_PC. In this HardwareSetup function, the clock pulse generator (CPG), bus state controller (BSC) and cache setting functions are called.

Figure 1 shows the processing flows of functions PowerON_Reset_PC and HardwareSetup.

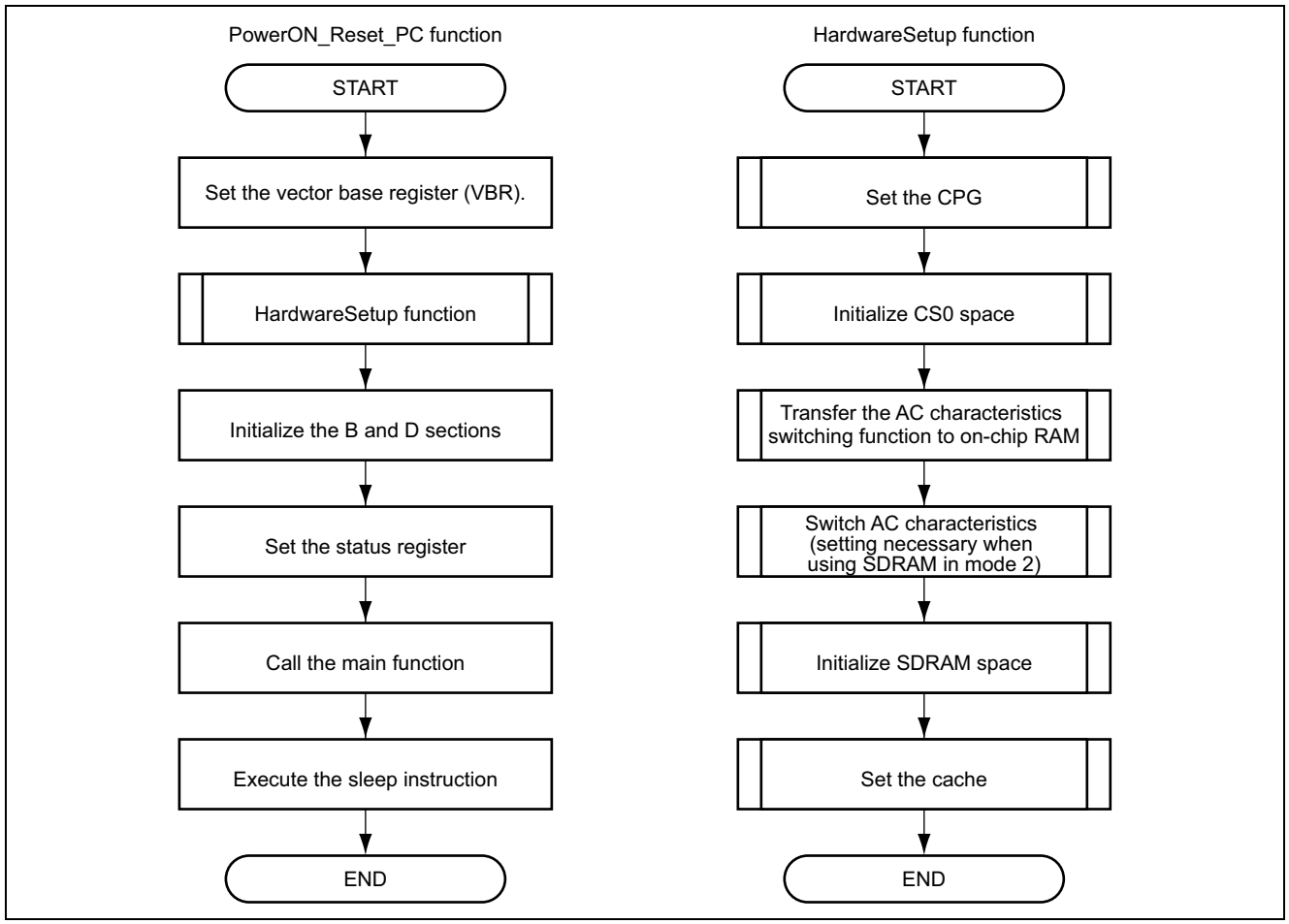

Figure 1 Processing Flows of Functions PowerON_Reset_PC and HardwareSetup

2.2 Settings Made by Sample Program

Table 1 shows the settings in the sample program.

Table 1 Settings in Sample Program

Module	Description
CPG	Internal clock: 200 MHz Bus clock: 66.67 MHz Peripheral clock: 33.33 MHz MTU clock: 100 MHz
BSC	CS0 space: Flash memory Number of access wait cycles: 6 CS3 space: SDRAM Data bus width: 32 bits Row address: 12 bits Column address: 9 bits CAS latency: 2 cycles Setting for AC characteristics switching: Sets delay time extension (necessary when using SDRAM in mode 2)
PFC	Multiplex pins are configured as address bus, data bus, and bus control pins used in CS0 and CS3.
Cache	Instruction/operand cache is enabled

2.3 Notes on Sample Program Usage

In order to allow allocation and initialization of B and D sections in the external memory, the sample program executes function HardwareSetup before initializing B and D sections to initialize the bus state controller. Because of this, do not use variables, such as global variables, allocated in sections that are initialized by function _INITSCT in those functions, including function HardwareSetup, that are executed before section initialization (before execution of function _INITSCT).

In the sample program, the interrupt mask level is set to 15 when the main function is called.

Since the function for setting the AC characteristics switching register needs to be executed in on-chip RAM, the following section setting is necessary:
From the Options pull-down menu of the HEW, select [SuperH RISC engine Standard Toolchain]. Select [linker/library] tab in the dialog box, and at the [Section to be mapped from ROM to RAM] option in the [Output] category, specify PURAM as ROM and RPURAM as RAM.

3. Sample Program Listing

1. Sample Program Listing: resetprog.c (1)

```
/*""FILE COMMENT""*********************************************************************
*
*   System Name : SH7206 Sample Program
*   File Name   : resetprg.c
*   Version     : 1.00.00
*   Contents    : SH7206 initial configuration
*   Model       : M3A-HS60
*   CPU         : SH7206
*   Compiler    : SHC9.0.00
*   OS          : none
*
*   Note        : This file was originally created by the Renesas Project Generator
*                 (Ver.3.1) and modified for the application note.
*
*                 <Caution>
*                 This sample program is for reference
*                 and its operation is not guaranteed.
*                 Customers should use this sample program for technical reference
*                 in software development.
*
*   This file is generated by Renesas Project Generator (Ver.3.1).
*
*   Copyright (C) 2004 Renesas Technology Corp. All Rights Reserved
*   AND Renesas Solutions Corp. All Rights Reserved
*
*       history   : 2004.10.14 ver.1.00.00
*""FILE COMMENT END""*****************************************************************/
#include    <machine.h>
#include    <_h_c_lib.h>
#include    "stacksct.h"

#define SR_Init    0x000000F0
#define INT_OFFSET 0x10

extern unsigned int INT_Vectors;
void PowerON_Reset_PC(void);
void Manual_Reset_PC(void);
extern void main(void);
extern void HardwareSetup(void);

//extern void srand(unsigned int);    // Remove the comment characters when you use rand()
//extern char *_s1ptr;                // Remove the comment characters when you use strtok()

/*==== Changing section name to ResetPRG ====*/
#pragma section ResetPRG

/*==== Entry function specification ====*/
#pragma entry PowerON_Reset_PC

```

2. Sample Program Listing: resetprog.c (2)

```
/*""FUNC COMMENT""*************************************************************************
* ID            :
* Module summary: CPU initialization function
*-------------------------------------------------------------------------------------------
* Include       : #include "iodefine.h"
*-------------------------------------------------------------------------------------------
* Declaration   : void io_init_cache(void)
*-------------------------------------------------------------------------------------------
* Functional description:
*               : CPU initialization processing that is registered with the power-on
*               : reset exception vector table.
*               : This function is executed first after power-on reset.
*-------------------------------------------------------------------------------------------
* Argument      : None
*-------------------------------------------------------------------------------------------
* Return value  : None
*-------------------------------------------------------------------------------------------
* Note          : Commented-out processing should be enabled as necessary.
*
*""FUNC COMMENT END""*********************************************************************/
void PowerON_Reset_PC(void)
{
    /*==== Setting vector base register (VBR) ====*/
    set_vbr((void *)((char *)&INT_Vectors - INT_OFFSET));

    /*==== HardwareSetup function ====*/
    HardwareSetup(); // Use Hardware Setup

    /*==== Initializing B and D sections ====*/
    _INITSCT();

//    errno=0;                      // Remove the comment characters when you use errno
//    srand(1);                     // Remove the comment characters when you use rand()
//    _s1ptr=NULL;                  // Remove the comment characters when you use strtok()

    /*==== Setting status register ====*/
    set_cr(SR_Init);
    nop();

    /*==== Function call to main function ====*/
    main();

    /*==== Executing sleep instruction ====*/
    sleep();
}

```

3. Sample Program Listing: resetprog.c (3)

```
 96  //#pragma entry Manual_Reset_PC  // Remove the comment characters when you use Manual Reset
 97  /*""FUNC COMMENT""*****************************************************************************
 98  * ID            :
 99  * Module summary: Manual reset processing
100  *-----------------------------------------------------------------------------------------------
101  * Include       :
102  *-----------------------------------------------------------------------------------------------
103  * Declaration   : void Manual_Reset_PC(void)
104  *-----------------------------------------------------------------------------------------------
105  * Functional description: Function registered with the manual reset exception vector table.
106  *               : No processing is defined in the sample program.
107  *               : Add processing as necessary.
108  *-----------------------------------------------------------------------------------------------
109  * Argument      : None
110  *-----------------------------------------------------------------------------------------------
111  * Return value  : None
112  *-----------------------------------------------------------------------------------------------
113  * None   : None
114  *""FUNC COMMENT END""*************************************************************************/
115  void Manual_Reset_PC(void)
116  {
117      /* NOP */
118  }
119  /* END of File */
```

4. Sample Program Listing: hwsetup.c (1)

```
/*""FILE COMMENT""*********************************************************************
*
*   System Name : SH7206 Sample Program
*   File Name   : hwsetup.c
*   Version     : 1.00.00
*   Contents    : Hardware initialization function
*   Model       : M3A-HS60
*   CPU         : SH7206
*   Compiler    : SHC9.0.00
*   OS          : none
*
*   note        :
*                <Caution>
*                This sample program is for reference
*                and its operation is not guaranteed.
*                Customers should use this sample program for technical reference
*                in software development.
*
*   Copyright (C) 2005 Renesas Technology Corp. All Rights Reserved
*   AND Renesas Solutions Corp. All Rights Reserved
*
*   history     : 2004.10.14 ver.1.00.00
*   history     : 2005.02.21 ver.1.00.01  AC characteristics switching setting added
*""FILE COMMENT END""*****************************************************************/
#include "iodefine.h"

/*  Remove this #define directive when you use clock mode7 */
#define USE_CLKMODE2

/* ==== Prototype declarations ==== */
void  HardwareSetup(void);
static void init_puram_section(void);
void set_acswr(void);


/* ==== Externally referenced prototype declarations ==== */
extern void io_set_cpg(void);
extern void io_init_bsc_cs0(void);
extern void io_init_sdram(void);
extern void io_init_cache(void);

```

5. Sample Program Listing: hwsetup.c (2)

```
43  /*""FUNC COMMENT""*****************************************************************************
44  * ID            :
45  * Module summary: Hardware initialization function
46  *-----------------------------------------------------------------------------------------------
47  * Include       : #include "iodefine.h"
48  *-----------------------------------------------------------------------------------------------
49  * Declaration   : void  HardwareSetup(void)
50  *-----------------------------------------------------------------------------------------------
51  * Functional description: Initializes the CPG, PFC, and BSC
52  *               : (flash memory access control and SDRAM initialization)
53  *-----------------------------------------------------------------------------------------------
54  * Argument      : None
55  *-----------------------------------------------------------------------------------------------
56  * Return value  : None
57  *-----------------------------------------------------------------------------------------------
58  * Functions used:
59  *-----------------------------------------------------------------------------------------------
60  * Note          :
61  *""FUNC COMMENT END""*************************************************************************/
62  void  HardwareSetup(void)
63  {
64      /*==== Configuring the CPG ====*/
65      io_set_cpg();
66
67      /*==== CS0 initialization ====*/
68      io_init_bsc_cs0();
69
70  #ifdef USE_CLKMODE2
71      /*==== Setting AC characteristics switching ====*/
72      init_puram_section();
73      set_acswr();
74  #endif
75      /*==== SDRAM space initialization ====*/
76      io_init_sdram();
77
78      /*==== Setting the cache ====*/
79      io_init_cache();
80  }
81
82
```

6. Sample Program Listing: hwsetup.c (3)

```
 83  /*""FUNC COMMENT""*****************************************************************************
 84  * ID            :
 85  * Module summary: Program section transfer from ROM to on-chip RAM
 86  *-----------------------------------------------------------------------------------------------
 87  * Include       : #include "iodefine.h"
 88  *-----------------------------------------------------------------------------------------------
 89  * Declaration   : static void init_puram_section(void)
 90  *-----------------------------------------------------------------------------------------------
 91  * Function      : Transfers URAM section from ROM to on-chip RAM
 92  *-----------------------------------------------------------------------------------------------
 93  * Argument      : None
 94  *-----------------------------------------------------------------------------------------------
 95  * Return value  : None
 96  *-----------------------------------------------------------------------------------------------
 97  * Functions used:
 98  *-----------------------------------------------------------------------------------------------
 99  * Note          : A separate function is used to transfer PURAM section because set_acswr
100  *               : must be transferred prior to section initialization by _INITSCT().
101  *""FUNC COMMENT END""*************************************************************************/
102  static void init_puram_section(void)
103  {
104      unsigned long *src,*end,*dst;
105
106      src = (unsigned long *)__sectop("PURAM");
107      end = (unsigned long *)__secend("PURAM");
108      dst = (unsigned long *)__sectop("RPURAM");
109
110      while(src < end){
111          *dst++ = *src++;
112      }
113  }
114
115  #pragma section URAM
116  /*""FUNC COMMENT""*****************************************************************************
117  * ID            :
118  * Module summary: AC characteristics switching function
119  *-----------------------------------------------------------------------------------------------
120  * Include       : #include "iodefine.h"
121  *-----------------------------------------------------------------------------------------------
122  * Declaration   : void  set_acswr(void)
123  *-----------------------------------------------------------------------------------------------
124  * Function      : Changes delay time extension in AC characteristics.
125  *-----------------------------------------------------------------------------------------------
126  * Argument      : None
127  *-----------------------------------------------------------------------------------------------
128  * Return value  : None
129  *-----------------------------------------------------------------------------------------------
130  * Functions used:
131  *-----------------------------------------------------------------------------------------------
132  * Note          : This setting is necessary when SDRAM is used in clock mode 2.
133  *               : This function must be executed on URAM.
134  *               : This setting must not be made in clock mode 7.
135  *""FUNC COMMENT END""*************************************************************************/
136  void set_acswr(void)
137  {
138      volatile unsigned long reg;
139
140      do{
141      /* ==== Setting AC characteristics switching ==== */
142          AC.ACKEYR.BYTE = 0;            /* Write byte data twice to the AC               */
143          AC.ACKEYR.BYTE = 0;            /* characteristics switching key register (ACKEYR)*/
144          AC.ACSWR.LONG = 0x9;           /* Set delay time extension                      */
145      }while(AC.ACSWR.LONG != 0x9);
146  }
147  /* End of File */
148
```

7. Sample Program Listing: cpg.c (1)

```
/*""FILE COMMENT""*******************************************************************************
*
*   System Name: SH7206 Sample Program
*   File Name  : cpg.c
*   Version    : 1.00.00
*   Contents   : CPG setting processing
*   Model      : M3A-HS60
*   CPU        : SH7206
*   Compiler   : SHC9.0.00
*   OS         : none
*
*   note       :
*              <Caution>
*              This sample program is for reference
*              and its operation is not guaranteed.
*              Customers should use this sample program for technical reference
*              in software development.
*
*   Copyright (C) 2004 Renesas Technology Corp. All Rights Reserved
*   AND Renesas Solutions Corp. All Rights Reserved
*
*   history   : 2004.10.14 ver.1.00.00
*""FILE COMMENT END""*******************************************************************************/
#include "iodefine.h"

/* ==== Prototype declaration ==== */
void io_set_cpg(void);

/*""FUNC COMMENT""*******************************************************************************
* ID            :
* Module summary: CPG setting
*-------------------------------------------------------------------------------------------------
* Include       : #include "iodefine.h"
*-------------------------------------------------------------------------------------------------
* Declaration   : void io_set_cpg(void)
*-------------------------------------------------------------------------------------------------
* Functional description:
*               : Configures the clock pulse generator (CPG) as follows:
*               : Internal clock (I Clock) = 200 MHz, bus clock (B Clock) = 66.6 MHz,
*               : peripheral clock (P Clock) = 33.3 MHz, and MTU clock (M Clock) = 100 MHz
*-------------------------------------------------------------------------------------------------
* Argument      : None
*-------------------------------------------------------------------------------------------------
* Return value  : None
*-------------------------------------------------------------------------------------------------
* Note          : The settings made by this function are example when the input clock is
*               : 16.6 MHz in clock mode 2.
*""FUNC COMMENT END""*******************************************************************************/
void io_set_cpg(void)
{
    /* ==== CPG Setting ==== */
    WDT.WTCSR.WORD = 0xa51e;            /* Stop WDT; set WDT counter clock division ratio*/
                                        /* 1/4096 x P-phi (33.3MHz)                     */
    WDT.WTCNT.WORD = 0x5aad;            /* Initial counter value: D'173 (= 10 ms )      */
    CPG.FRQCR.WORD = 0x120c;            /* PLL1(x3),PLL2(x4),I:B:P=12:4:2
                                         * Clockin = 16.67 MHz, CKIO = 66.6 MHz
                                         * I Clock = 200 MHz, B Clock = 66.6 MHz,
                                         * P Clock = 33.3 MHz
                                         */
    CPG.MCLKCR.BYTE  = 0x41;            /* MTU = 100 MHz                                */
}

/* End of File */

```

8. Sample Program Listing: bsc_cs0.c (1)

```
/*""FILE COMMENT""*********************************************************************
*
*   System Name: SH7206 Sample Program
*   File Name  : bsc_cs0.c
*   Version    : 1.00.00
*   Contents   : SH7206 initial configuration
*   Model      : M3A-HS60
*   CPU        : SH7206
*   Compiler   : SHC9.0.00
*   OS         : none
*
*   note       :
*              <Caution>
*              This sample program is for reference
*              and its operation is not guaranteed.
*              Customers should use this sample program for technical reference
*              in software development.
*
*   Copyright (C) 2004 Renesas Technology Corp. All Rights Reserved
*   AND Renesas Solutions Corp. All Rights Reserved
*
*   history     : 2004.10.01 ver.1.00.00
*               : 2005.03.17 ver.1.00.01 Wait Invalidity
*""FILE COMMENT END""*****************************************************************/
#include "iodefine.h"

/* ==== Prototype declaration ==== */
void io_init_bsc_cs0(void);
```

9. Sample Program Listing: bsc_cs0.c (2)

```
/*""FUNC COMMENT""*****************************************************************************
* ID             :
* Module summary : CS0 setting
*-----------------------------------------------------------------------------------------------
* Include        : #include "iodefine.h"
*-----------------------------------------------------------------------------------------------
* Declaration    : void io_init_bsc_cs0(void)
*-----------------------------------------------------------------------------------------------
* Functional description: Configures the pin function controller (PFC) and the bus state
*                : controller (BSC) to set timing of access to flash memory in the CS0 space.
*-----------------------------------------------------------------------------------------------
* Argument       : None
*-----------------------------------------------------------------------------------------------
* Return value   : None
*-----------------------------------------------------------------------------------------------
* Note           : The PFC is set by bit manipulation so as not to change the PFC setting
*                : values set by other processing.
*""FUNC COMMENT END""*************************************************************************/
void io_init_bsc_cs0(void)
{
    /* ==== Setting PFC ==== */
    PORT.PACRL4.BIT.PA12MD = 0x1;      /* Specify as WE0 output                          */
    PORT.PBCR3.BIT.PB9MD   = 0x2;      /* A21 output                                     */

    /* ==== Setting CS0 space bus control register (CS0BCR) ==== */
    BSC.CS0BCR.LONG = 0x20000400ul;
                                       /* Idle cycle insertion between write and read    */
                                       /* cycles or between write and write cycles:      */
                                       /* 2 idle cycles.                                 */
                                       /* Data bus width: 16 bits                        */

    /* ==== Setting CS0 space wait control register (CS0WCR) ====  */
    BSC.UN0_BSC.NORMAL.REG_CS0WCR.LONG = 0x00000b41ul;
                                       /* Addr,CS assert -> RD,WE assert delay cycles    */
                                       /* : 1.5 cycles                                   */
                                       /* Access wait cycles: 6 cycles                   */
                                       /* Ignore external wait input                     */
                                       /* CS,RD,WE negate -> Addr,CS negate delay cycles */
                                       /* : 1.5 cycles                                   */
}

/* End of File */
```

10. Sample Program Listing: bscsdram.c (1)

```
/*""FILE COMMENT""*********************************************************************
*
*   System Name: SH7206 Sample Program
*   File Name  : bscsdram.c
*   Version    : 1.00.00
*   Contents   : SH7206 initial configuration
*   Model      : M3A-HS60
*   CPU        : SH7206
*   Compiler   : SHC9.0.00
*   OS         : none
*
*   note       :
*              <Caution>
*              This sample program is for reference
*              and its operation is not guaranteed.
*              Customers should use this sample program for technical reference
*              in software development.
*
*   Copyright (C) 2004 Renesas Technology Corp. All Rights Reserved
*   AND Renesas Solutions Corp. All Rights Reserved
*
*       history   : 2004.10.14 ver.1.00.00
*""FILE COMMENT END""*****************************************************************/
#include "iodefine.h"

/* ==== Macro definition ==== */

/* Access address for writing to the SDRAM mode register */
#define    SDRAM_MODE        (*(volatile unsigned short *)(0xfffc5880))

/* ==== Prototype declaration ==== */
void io_init_sdram(void);

```

11. Sample Program Listing: bscsdram.c (2)

```
/*""FUNC COMMENT""*****************************************************************************
* ID            :
* Module summary: Setting for 32-bit bus width connection with SDRAM
*-----------------------------------------------------------------------------------------------
* Include       : #include "iodefine.h"
*-----------------------------------------------------------------------------------------------
* Declaration   : void io_init_sdram(void)
*-----------------------------------------------------------------------------------------------
* Functional description:
                : Configure the pin function controller (PFC) and the bus state
*               : controller (BSC) to enable SDRAM in the CS3 space.
*-----------------------------------------------------------------------------------------------
* Argument      : None
*-----------------------------------------------------------------------------------------------
* Return value  : None
*-----------------------------------------------------------------------------------------------
* Note          : The PFC is set by bit manipulation so as not to change the PFC setting
*               : values set by other processing.
*""FUNC COMMENT END""*************************************************************************/
void io_init_sdram(void)
{
    volatile int j = 40000;             /* 200 usec wait count @200 MHz                  */

    /* ==== PFC setting ==== */
    PORT.PACRH2.BIT.PA23MD = 0x1;       /* DQMUU output                                  */
    PORT.PACRH2.BIT.PA22MD = 0x1;       /* DQMUL output                                  */
    PORT.PACRL4.BIT.PA13MD = 0x1;       /* DQMLU output                                  */
    PORT.PACRL4.BIT.PA12MD = 0x1;       /* DQMLL output                                  */
    PORT.PACRL3.BIT.PA9MD = 0x5;        /* CKE output                                    */
    PORT.PACRL3.BIT.PA8MD = 0x5;        /* RD/WR# output                                 */
    PORT.PACRL2.BIT.PA7MD = 0x2;        /* CS3 output                                    */
    PORT.PBCR2.BIT.PB5MD = 0x4;         /* CASL output                                   */
    PORT.PBCR2.BIT.PB4MD = 0x4;         /* RASL output                                   */

    PORT.PDCRH4.WORD = 0x1111;          /* D31-D28 input/output                          */
    PORT.PDCRH3.WORD = 0x1111;          /* D27-D24 input/output                          */
    PORT.PDCRH2.WORD = 0x1111;          /* D23-D20 input/output                          */
    PORT.PDCRH1.WORD = 0x1111;          /* D19-D16 input/output                          */

    /* ==== Setting CS3 space bus control register (CS3BCR) ==== */
    BSC.CS3BCR.LONG = 0x10004600ul;     /*
                                        * Idle cycle insertion between write and read cycles
                                        * or between write and write cycles: 1 idle cycle
                                        * Memory type    : SDRAM
                                        * Data bus width : 32 bits
                                        */
    /* ==== Setting CS3 space wait control register (CS3WCR) ==== */
    BSC.UN2_BSC.SDRAM.REG_CS3WCR.LONG = 0x00004892ul;
                                        /*
                                         * Precharge cycles: 2 cycles
                                         * Wait cycles between ACT -> READ commands
                                         *                          : 2 cycles
                                         * Area 3 CAS latency       : 2 cycles
                                         * Precharge activation cycles: 2 cycles
                                         * Idle cycles between REF -> ACT/REF/MRS commands
                                         *                          : 5 cycles
                                        */
```

12. Sample Program Listing: bscsdram.c (3)

```
 91      /* ==== Setting SDRAM control register (SDCR) ==== */
 92      BSC.SDCR.LONG = 0x00000809ul;       /*
 93                                          * Refresh control 1     : Perform refresh
 94                                          * Refresh control 2     : Auto refresh
 95                                          * Bank active mode
 96                                          *                       : Auto precharge mode
 97                                          * Area 3 row address    : 12 bits
 98                                          * Area 3 column address : 9 bits
 99                                          */
100      /* ==== Setting refresh timer constant register (RTCOR) ==== */
101      BSC.RTCOR.LONG = 0xa55a0041ul;      /*
102                                          * Refresh interval:
103                                          * 15.625 usec/240 nsec = 64(0x41) cycles
104                                         */
105      /* ==== Setting refresh timer control/status register (RTCSR) setting ==== */
106      BSC.RTCSR.LONG = 0xa55a0010ul;      /*
107                                          *  Start initialization sequence
108                                          *  Clock select: B phi 16: 1 cycle = 240 nsec
109                                          *  Refresh cycle count: once
110                                          */
111      /* ==== Has 200 usec duration elapsed? ==== */
112      while(j-- > 0){
113          /* wait */
114      }
115
116      /* ==== Writing to SDRAM mode register ==== */
117      SDRAM_MODE = 0;                     /*
118                                          * Arbitrary data may be written
119                                          * SDRAM mode register setting for CS3 space
120                                          * Burst read (burst length: 1)/single write
121                                          */
122  }
123  /* End of File */
```

13. Sample Program Listing: cache.c (1)

```
/*""FILE COMMENT""*********************************************************************
*
*   System Name: SH7206 Sample Program
*   File Name  : cache.c
*   Version    : 1.00.00
*   Contents   : sample of cache register
*   Model      : M3A-HS60
*   CPU        : SH7206
*   Compiler   : SHC9.0.00
*   OS         : none
*
*   note       : Sample program for setting the cache
*
*              <Caution>
*              This sample program is for reference
*              and its operation is not guaranteed.
*              Customers should use this sample program for technical reference
*              in software development.
*
*   Copyright (C) 2004 Renesas Technology Corp. All Rights Reserved
*   AND Renesas Solutions Corp. All Rights Reserved
*
*   history    : 2004.08.24 ver.0.01.00
*""FILE COMMENT END""*****************************************************************/
#include <machine.h>
#include "iodefine.h"

/* ==== Prototype declaration ==== */
void io_init_cache(void);

```

14. Sample Program Listing: cache.c (2)

```
#pragma section CACHE       /* This section is to be allocated in non-cacheable CS0 space */
/*""FUNC COMMENT""*****************************************************************************
* ID            :
* Module summary: Cache initialization
*-----------------------------------------------------------------------------------------------
* Include       : #include "iodefine.h"
*-----------------------------------------------------------------------------------------------
* Declaration   : void io_init_cache(void)
*-----------------------------------------------------------------------------------------------
* Function      : Flushes and enables the instruction/operand cache.
*               :
*-----------------------------------------------------------------------------------------------
* Argument      : None
*-----------------------------------------------------------------------------------------------
* Return value  : None
*-----------------------------------------------------------------------------------------------
* Note          : The section name for this function has been changed because the program
*               : should be allocated in non-cacheable space.
*               : If this function is only used with the interrupt mask level 15,
*               : interrupt mask setting/clearing processing is not necessary.
*""FUNC COMMENT END""*************************************************************************/
void io_init_cache(void)
{
    volatile unsigned long reg;
    int mask;

    /* ==== Setting the interrupt mask ==== */
    mask = get_imask();
    set_imask(15);                      /* Set to level 15                              */

    /* ==== Setting the cache register ==== */
    CCNT.CCR1.LONG = 0x0909ul;          /*
                                         ICF=1: Flushes instruction cache
                                         ICE=1: Enables instruction cache
                                         OCF=1: Flushes operand cache
                                         WT =0: Cache write back mode
                                         OCE=1: Enables operand cache
                                        */

    /* ==== Reading the cache register ==== */
    reg = CCNT.CCR1.LONG ;

    /* ==== Clearing the interrupt mask ==== */
    set_imask(mask);                    /* Set to the original level                    */

}

/* End of File */
```

4. Reference Documents

- SH-2A SH2A-FPU Software Manual (Rev.3.00)
  (Download the latest edition from the website of Renesas Technology Corp.)

- SH7206 Group Hardware Manual (Rev. 1.00)
  (Download the latest edition from the website of Renesas Technology Corp.)

5. Website

- Website of Renesas Technology Corp.
  http://www.renesas.com/

Revision Record

		Description	
Rev.	Date	Page	Summary
1.00	Sep.05.05	—	First edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.