SH7670 Group

SH7670 Example of Initialization

REJ06B0799-0101
Rev.1.01
May 07, 2010

Introduction

This application note describes an example of initialization of the CPUs of the SH7670, SH7671, SH7672 and SH7673.

Target Device

SH7670

Contents

1. Preface

1.1 Specifications

- The clock pulse generator (CPG), bus state controller (BSC), pin function controller (PFC), and cache are initialized after release from the reset state.

1.2 Modules Used

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

1.3 Applicable Conditions

- MCU: SH7670/SH7671/SH7672/SH7673
  (R5S76700/R5S76710/R5S76720/R5S76730)
- Operating frequency Internal clock: 200 MHz
  Bus clock: 66.67 MHz
  Peripheral clock: 33.33 MHz
- Integrated development environment:
  High-performance Embedded Workshop Ver.4.03.00
  from Renesas Electronics
- C compiler: SuperH RISC Engine Family C/C++ Compiler Package Ver.9.01 Release01
  from Renesas Electronics
- Compiler options: Default settings of High-performance Embedded Workshop
  (-cpu = sh2afpu -fpu = single -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)

1.4 Related Application Notes

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

- SH7670 Example of Setting the CPG to change the operating frequency (REJ06B0810)
- SH7670 Example of BSC SDRAM Interface Connection (32-Bit Data Bus) (REJ06B0782)
- SH7670 Example of BSC Flash Memory Connection (REJ06B0783)
- SH7670 Example of Cache Memory Setting (REJ06B0779)

2. Description of the Sample Application

Before a C-based main function can be executed, an initialization program must perform the minimum of processing for hardware initialization (memory initialization etc.) after power-on reset. This document describes an example of initial settings for the initialization program.

Use of the program for initial settings described in this application note is a precondition for all of the other application notes for the SH7670.

2.1 Description of the Sample Program

The initialization program consists of multiple files of source code; resetprg.c that includes the PowerON_Reset_PC function, and the called functions such as hwsetup.c and init_section.c, etc. The principal source files are described below;

- resetprg.c

resetprg.c was created on the basis of a file that is automatically generated by the High-performance Embedded Workshop, and the file contains the definition of the PowerON_Reset_PC function. Since PowerON_Reset_PC is the first function to be executed after release from the reset state, the first address of the executable code is placed in the reset vector defined in vecttbl.c. Figure 1 shows the flow of processing by the PowerON_Reset_PC.

- hwsetup.c

The HardwareSetup function, which is called from the PowerON_Reset_PC, is defined in hwsetup.c. The HardwareSetup function calls the individual functions for the clock pulse generator (CPG), bus state controller (BSC), and cache settings, thus making the minimum of hardware settings required by systems. Figure 2 shows the flow of processing by HardwareSetup.

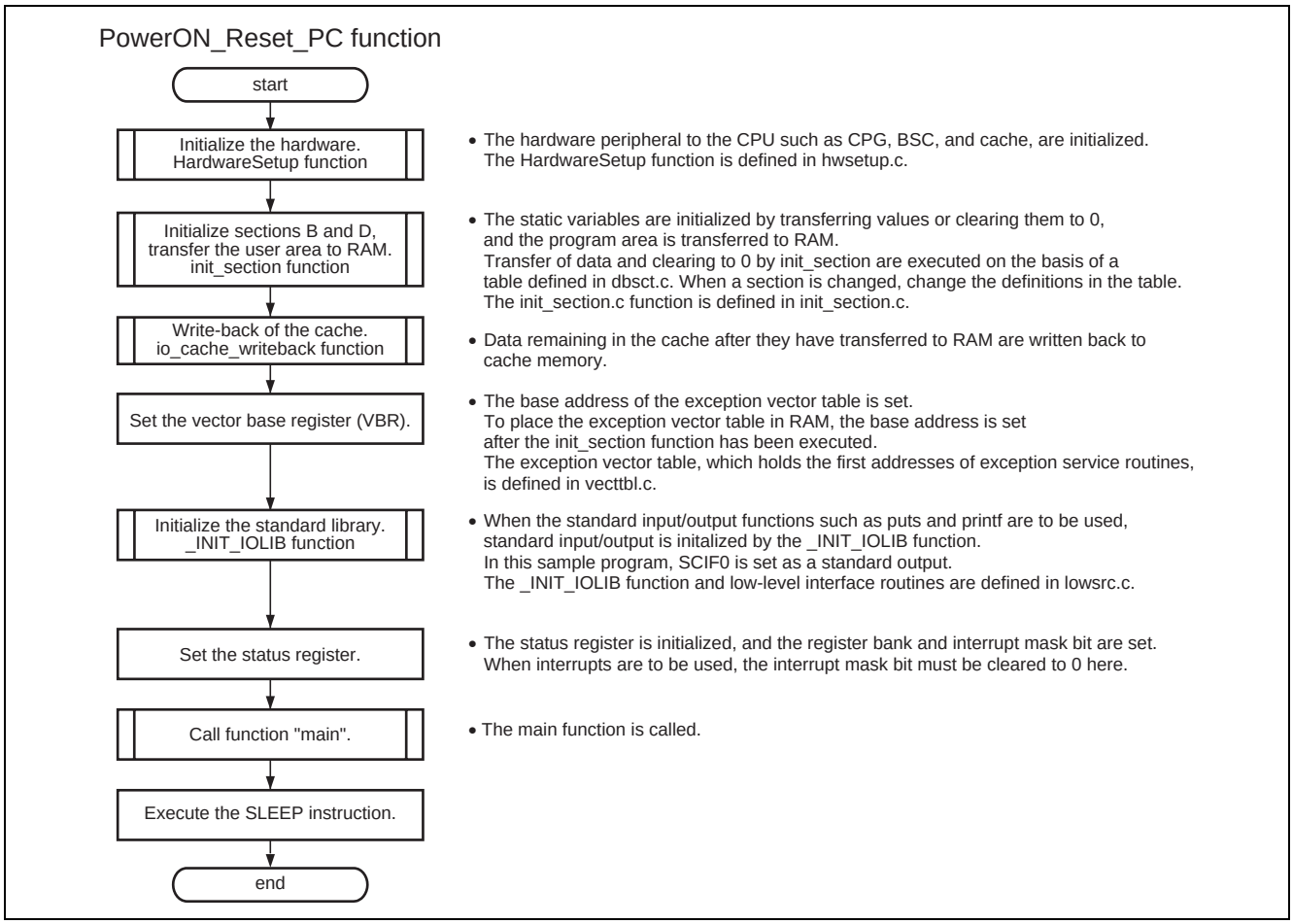

Figure 1 Flow of Processing by the Reset Program

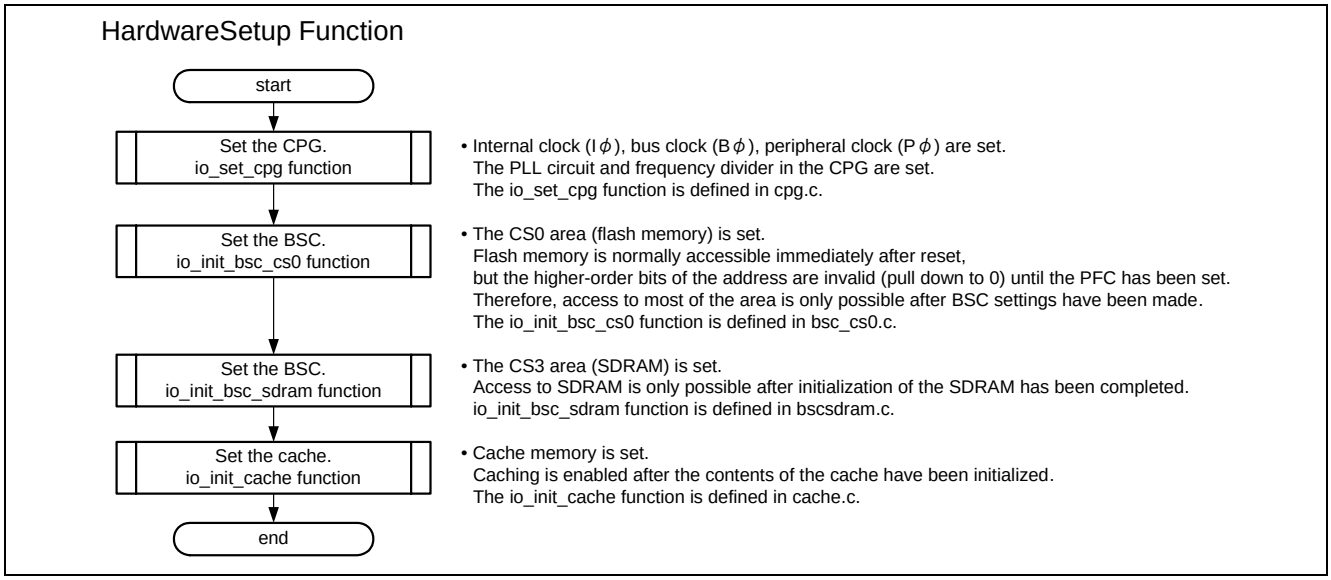

Figure 2 Flow of Processing by the Hardware Initialization Function

2.2 Description of Settings in the Sample Program

Table 1 is a list of the settings in the sample program.

Table 1 Settings in the Sample Program

Module	Description
CPG	Internal clock: 200 MHz Bus clock: 66.67 MHz Peripheral clock: 33.33 MHz
BSC	CS0 space: flash memory Number of cycles to wait for access: 5 CS3 space: SDRAM Data bus width: 32 bits Row address bits: 12 bits Column address bits: 9 bits CAS latency: 2 cycles
PFC	The address bus, data bus, and bus control pin functions for use in the CS0 and CS3 spaces are set.
Cache	Enabled
SCIF	Set as a standard output. • SCIF0 is used. • Asynchronous/8-bit data length/ no parity bit/1 stop bit • 115,200 bps

2.3 Notes on Using the Sample Program

- SDRAM only becomes accessible after initialization has been performed.

In this sample program, the memory space in the SDRAM is only used after the bus state controller has been initialized by the HardwareSetup function. Please note that attempting to use of memory space in the SDRAM before initialization has been performed leads to abnormal operation.

- The stack area (S section) must not be placed in the SDRAM.

The value set for the reset vector (last address of the S section + 1) is set as the initial value of the stack pointer (R15). In this sample program, the S section is placed in on-chip memory. If the S section were to be placed in the SDRAM, access to non-initialized SDRAM would proceed when the functions of the initialization program are called.

- Access to the static variable area must only proceed after the init_section function has been executed.

Execution of the init_section function initializes the static variable area for the C language. Values are undefined if the area is accessed before initialization.

3. Settings for Transfer of the User Program Area to RAM

This sample program is executed from user RAM. A program area and constants area are set up in the user area where the main function etc. will be of RAM. This section describes the process and settings of the program.

3.1 Section Allocation of the Sample Program

Table 2 shows the section allocation of the sample program. Table 3 shows the memory areas of the SuperH RISC engine C/C ++ compiler and the corresponding section names.

For higher-speed processing, this sample program is executed from RAM after the contents of the P section in ROM have been transferred to the RP section in RAM. The P section holds the user program, which contains the main function as well as the standard library. The contents of the C section and DINTTBL are also transferred to RAM.

Transfer of some programs (such as the initialization program) to RAM is impossible. In such cases, the #pragma section function is used to allocate them to the PResetPRG section or the PIntPRG section. These sections are outside the scope for transfer to RAM.

For details on the compiler, see section 4, Optimizing Linkage Editor Options, and section 9.1, Program Structure, in the *SuperH™ RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package V.9.01 User's Manual* (REJ10J1571).

Table 2 Section Allocation of the Sample Program

Address	Device	Section	Description
0x00000000	Flash memory	DVECTTBL	Reset vector
		DINTTBL	Exception vector table
0x00001000		PResetPRG	Program area of the initialization program
		PIntPRG	Interrupt program area (i.e. NMI interrupt)
		C$BSEC	Table defined in dbsct.c for clearing to 0
		C$DSEC	Table defined in dbsct.c for data transfer
		P	Program area for the user program and standard library
		C	Constants area for the user program and standard library
		D	Initialized data area (with initial values) for the user program and standard library
		PURAM	Program area to be allocated to on-chip RAM
0x0C000000	SDRAM	RP	Destination in RAM for transfer of the P section
		RC	Destination in RAM for transfer of the C section
0x20000800	Flash memory (Cache disabled)	PCACHE	Program allocated to a non-cache area
0xFFF80000	On-chip RAM	RINTTBL	Destination in RAM for transfer of the exception vector table
		R	Initialized data area (variable) for the user program and standard library
		B	Non-initialized data area for the user program and standard library
		RPURAM	Destination in RAM for transfer of the PURAM section
0xFFF87C00		S	Stack area

Table 3 Memory Areas and Sections Controlled by Complier

Memory	Section*[1]	Function
Program area	P	Holds machine language
Constants area	C	Holds const-type data
Initialized data area	D*[2]	Holds data with an initial value
Non-initialized data area	B	Holds data without an initial value

Notes: 1. When #pragma section is used, other section name can be specified. Please note that the actual symbol names will be the section names specified by using #pragma section with the relevant section name given in table 3 appended as a prefix.
e.g. #pragma section ResetPRG → PResetPRG, CResetPRG, DResetPRG, and BResetPRG
2. Allocation of the variable area allocated to RAM is defined by the sections option of the optimizing linkage editor. This will normally be the R section. When R is set up in RAM and the ROM support function of the sections option is used to also set up a section D in ROM, problems related to the relocation of symbols in RAM are resolved.

3.2 Settings Related to Transfer to RAM

This section describes the procedure to transfer a section from an area in ROM to an area in RAM.

3.2.1 Setting up the Tables of Data for Transfer (dbsct.c)

The tables DTBL[] of initial value for transfer and BTBL[] for clearing to 0 are defined in dbsct.c. DTBL[] is placed in the section to be transferred.

3.2.2 init_section Function

The init_section function transfers and clears the respective sections set up by dbsct.c as desbribed in section 3.2.1. The function is executed from the initialization program after SDRAM has been initialized.

Although the init_section function has the same functionality as the _INITSCT function of the standard library, the _INITSCT function does not treat the P section as a source of data for transfer. Since the _INITSCT function is placed in the P section by default, the function itself would become fall within the scope of data for transfer. The P section is treated as a source of data transfer in this sample program, so the init_section function is used instead. Since the init_section function is allocated to PResetPRG section, the P section that includes the user program and standard library can be the target for transfer.

3.2.3 ROM Support Function

When a program has been transferred with data from ROM to RAM, simply copying the corresponding contents of memory is not sufficient to enable execution from RAM. Execution also requires settings so that symbols that were defined in the ROM section have been relocated to addresses within the RAM at the time of linkage.

Go to the Optimizing linkage editor, and select the [Output category] > [ROM Support Function] menu item (open the [SuperH RISC Engine Standard Toolchain] window from the [Build] menu of the High-performance Embedded Workshop, click on the[Link/Library] tab, then select “Output” from the [category] pull-down menu and “ROM to RAM mapped sections” from the [Show entries for]), and set the ROM and RAM sections to be transferred. Specifying these options ensures the proper relocation of symbols to the addresses in RAM.

4. Listing of the Sample Program

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

```
/***********************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products. No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
***********************************************************************
* (C) 2007(2009,2010) Renesas Electronics Corporation. All rights reserved.
*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7671 Sample Program
*   File Name   : resetprg.c
*   Abstract    : SH7671 Initial Settings
*   Version     : 1.00.03
*   Device      : SH7671
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                               (Ver.9.01 Release01).
*   OS          : None
*   H/W Platform: M3A-HS71(CPU board)
*   Description :
***********************************************************************
*   History     : Jul.03,2007 ver.1.00.00
*               : Dec.06,2007 ver.1.00.01 PowerON_Reset_PC() header is modified
*               : Dec.18,2009 ver.1.00.02 Updated header comments
*               : Apr.07,2010 ver.1.00.03 Changed the company name and device name
*""FILE COMMENT END""*************************************************************/
#include <machine.h>
#include <_h_c_lib.h>
#include "stacksct.h"
#include "iodefine.h"

```

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

```
52    #define FPSCR_Init    0x00040001
53
54    #define SR_Init    0x000000F0
55    #define INT_OFFSET 0x10
56
57    extern unsigned int INT_Vectors;
58    void PowerON_Reset_PC(void);
59    void Manual_Reset_PC(void);
60
61    extern void main(void);
62    extern void HardwareSetup(void);
63    extern int io_cache_writeback(void);
64
65
66
67    //extern void srand(unsigned int);    // Remove the comment when you use rand()
68    //extern char *_s1ptr;        // Remove the comment when you use strtok()
69
70    /*==== Switch section name to ResetPRG ====*/
71    #pragma section ResetPRG
72
73    /*==== Specifying the entry function ====*/
74    #pragma entry PowerON_Reset_PC
75
76    /*""FUNC COMMENT""*****************************************************************
77     * ID            :
78     * Outline       : CPU initialization function
79     *-----------------------------------------------------------------------------
80     * Include       : "iodefine.h"
81     *-----------------------------------------------------------------------------
82     * Declaration   : void PowerON_Reset_PC(void) ;
83     *-----------------------------------------------------------------------------
84     * Description   : It is the CPU initialization process to register the power on
85     *               : reset exception vector table.
86     *               : This function is firstly executed after power on reset.
87     *-----------------------------------------------------------------------------
88     * Argument      : void
89     *-----------------------------------------------------------------------------
90     * Return Value  : void
91     *-----------------------------------------------------------------------------
92     * Note          : Enable the processes that are commented depending on its needs.
93     *""FUNC COMMENT END""*************************************************************/
94    void PowerON_Reset_PC(void)
95    {
96      set_fpscr(FPSCR_Init);
97
98      /*==== HardwareSetup function====*/
99      HardwareSetup(); // Use Hardware Setup
100
101     /*==== B and D sections initialization ====*/
102   //_INITSCT();
103     init_section();  /* INITSCT is not used since the P section is also transferred to RAM */
104     io_cache_writeback();
        /* Note that operand cache code does not remain on program transfer   */
```

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

```
105
106       /*==== Vector base register (VBR) setting ====*/
107       set_vbr((void *)((char *)&INT_Vectors - INT_OFFSET));
108
109       _INIT_IOLIB();              // Use stdio I/O
110
111     //errno=0;                // Remove the comment when you use errno
112     //srand(1);               // Remove the comment when you use rand()
113     //_s1ptr=NULL;            // Remove the comment when you use strtok()
114
115       /*==== Status register setting ====*/
116       set_cr(SR_Init);
117       nop();
118
119       /* ==== Bank number register setting ==== */
120       INTC.IBNR.BIT.BE = 0x01;/* Use the register bank in all interrupts */
121
122       /* ==== Interrupt mask level change ==== */
123       set_imask(0);
124
125       /*==== Function call of main function ====*/
126       main();
127
128       /*==== sleep instruction execution ====*/
129       sleep();
130     }
131
132
133     //#pragma entry Manual_Reset_PC       // Remove the comment when you use Manual Reset
134     /*""FUNC COMMENT""*****************************************************************
135      * ID            :
136      * Outline       : Manual reset process
137      *-----------------------------------------------------------------------------
138      * Include       :
139      *-----------------------------------------------------------------------------
140      * Declaration   : void Manual_Reset_PC(void);
141      *-----------------------------------------------------------------------------
142      * Description   : It is the function to register the manual reset exception vector table.
143      *               : The process is not defined in the reference program.
144      *               : Add the processes depending on its needs
145      *-----------------------------------------------------------------------------
146      * Argument      : void
147      *-----------------------------------------------------------------------------
148      * Return Value  : void
149      *-----------------------------------------------------------------------------
150      * Note          :
151      *""FUNC COMMENT END""*************************************************************/
152     void Manual_Reset_PC(void)
153     {
154       /* NOP */
155     }
156     /* END of File */
157
```

4.4 Sample Program Listing: "hwsetup.c" (1)

```
/***********************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products. No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
***********************************************************************
* (C) 2007(2010) Renesas Electronics Corporation. All rights reserved.
*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7671 Sample Program
*   File Name   : hwsetup.c
*   Abstract    : SH7671 Initial Settings
*   Version     : 1.01.00
*   Device      : SH7671
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                               (Ver.9.01 Release01).
*   OS          : None
*   H/W Platform: M3A-HS71(CPU board)
*   Description :
***********************************************************************
*   History     : Jul.04,2007 ver.1.00.00
*               : Oct.26 2007 ver.1.00.01 AC characteristics switch function added
*               : Dec.18,2009 ver.1.00.02 Updated header comments
*               : Apr.07,2010 ver.1.00.03 Changed the company name and device name
*               : Apr.12,2010 ver.1.01.00 Deleted AC characteristics register
*""FILE COMMENT END""*************************************************************/
#include "iodefine.h"

```

4.5 Sample Program Listing: "hwsetup.c" (2)

```
50    /* ==== Prototype declaration ==== */
51    void  HardwareSetup(void);
52
53    /* ==== referenced external Prototype declaration ==== */
54    extern void io_set_cpg(void);
55    extern void io_init_bsc_cs0(void);
56    extern void io_init_sdram(void);
57    extern void io_init_cache(void);
58
59    #pragma section ResetPRG
60    /*""FUNC COMMENT""*****************************************************************
61     * ID            :
62     * Outline       : Hardware initialization function
63     *-----------------------------------------------------------------------------
64     * Include       : "iodefine.h"
65     *-----------------------------------------------------------------------------
66     * Declaration   : void  HardwareSetup(void);
67     *-----------------------------------------------------------------------------
68     * Description   : The initial settings of CPG, PFC, and BSC（Flash memory
69     *               : access control and SDRAM initialization) are processed.
70     *-----------------------------------------------------------------------------
71     * Argument      : void
72     *-----------------------------------------------------------------------------
73     * Return Value  : void
74     *-----------------------------------------------------------------------------
75     * Note          :
76     *""FUNC COMMENT END""*************************************************************/
77    void  HardwareSetup(void)
78    {
79      /*====CPG setting====*/
80      io_set_cpg();
81
82      /*====CS0 initialization====*/
83      io_init_bsc_cs0();
84
85      /*====SDRAM area initialization====*/
86      io_init_sdram();
87
88      /*====Cache setting====*/
89      io_init_cache();
90
91    }
92
93    /* End of File */
94
```

4.6 Sample Program Listing: "cpg.c" (1)

```
/***********************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products. No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
***********************************************************************
* (C) 2007(2010) Renesas Electronics Corporation. All rights reserved.
*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7671 Sample Program
*   File Name   : cpg.c
*   Abstract    : CPG setting process
*   Version     : 1.01.02
*   Device      : SH7671
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                               (Ver.9.01 Release01).
*   OS          : None
*   H/W Platform: M3A-HS71(CPU board)
*   Description :
***********************************************************************
*   History     : Jul.04,2007 ver.1.00.00
*               : Aug.07,2007 ver.1.01.00 Secure frequency stablity time according
*               :                         to multiplication change
*               : Dec.18,2009 ver.1.01.01 Updated header comments
*               : Apr.07,2010 ver.1.01.02 Changed the company name and device name
*""FILE COMMENT END""*************************************************************/
#include "iodefine.h"

```

4.7 Sample Program Listing: "cpg.c" (2)

```
50
51  /* ==== Prototype Declaration ==== */
52  void io_set_cpg(void);
53
54  #pragma section ResetPRG
55  /*""FUNC COMMENT""*****************************************************************
56   * ID            :
57   * Outline       : CPG settings
58   *-----------------------------------------------------------------------------
59   * Include       : "iodefine.h"
60   *-----------------------------------------------------------------------------
61   * Declaration   : void io_set_cpg(void);
62   *-----------------------------------------------------------------------------
63   * Description   : Clock pulse generator (CPG) is set to set to the internal clock
64   *               : (I Clock), peripheral clock (P Clock), bus clock (B Clock), and
65   *               : I Clock = 200MHz, B Clock = 66.67MHz,  P Clock = 33.3MHz
66   *-----------------------------------------------------------------------------
67   * Argument      : void
68   *-----------------------------------------------------------------------------
69   * Return Value  : void
70   *-----------------------------------------------------------------------------
71   * Note          : This setting example is the case that the function's input clock
72   *               : is 16.67MHz and clock mode is 1.
73   *""FUNC COMMENT END""*************************************************************/
74  void io_set_cpg(void)
75  {
76      /* ==== CPG Setting ==== */
77      WDT.WTCSR.WORD = 0xa51e;        /* WDT Clock select */
78                                      /* 1/4096xP-phy (33.3MHz) */
79      WDT.WTCNT.WORD = 0x5aad;        /* Initial value of Counter: D'173 10mS */
80      CPG.FRQCR.WORD = 0x1104;        /* PLL1(x12),I:B:P=12:4:2
81                                      * CKIO:Output at time usually,Output when bus right is
                                                    opened,output at standby"L"
82                                      * Clockin = 16.67MHz, CKIO = 66.6MHz
83                                      * I Clock = 200MHz, B Clock = 66.6MHz,
84                                      * P Clock = 33.3MHz
85                                      */
86      /* ---- Enables clocks for all modules ---- */
87      CPG.STBCR3.BYTE = 0x00;
88      CPG.STBCR4.BYTE = 0x00;
89  }
90
91  /* End of File */
92
```

4.8 Sample Program Listing: "bsc_cs0.c" (1)

```
/***********************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products. No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
***********************************************************************
* (C) 2007(2010) Renesas Electronics Corporation. All rights reserved.
*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7671 Sample Program
*   File Name   : bsc_cs0.c
*   Abstract    : SH7671 Initial Settings
*   Version     : 1.00.02
*   Device      : SH7671
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                             (Ver.9.01 Release01).
*   OS          : None
*   H/W Platform: M3A-HS71(CPU board)
*   Description :
***********************************************************************
*   History     : Jul.03,2007 ver.1.00.00
*               : Dec.18,2009 ver.1.00.01 Updated header comments
*               : Apr.07,2010 ver.1.00.02 Changed the company name and device name
*""FILE COMMENT END""*****************************************************/
#include "iodefine.h"

```

4.9 Sample Program Listing: "bsc_cs0.c" (2)

```
48  /* ==== Prototype Declaration ==== */
49  void io_init_bsc_cs0(void);
50
51  #pragma section ResetPRG
52  /*""FUNC COMMENT""*****************************************************************
53   * ID            :
54   * Outline       : CS0 setting
55   *-----------------------------------------------------------------------------
56   * Include       : "iodefine.h"
57   *-----------------------------------------------------------------------------
58   * Declaration   : void io_init_bsc_cs0(void);
59   *-----------------------------------------------------------------------------
60   * Description   : Pin function controller (PFC) and bus state controller (BSC)
61   *               : are set, and the access timing to the FlashMemory of CS0 area
62   *               : is set.
63   *-----------------------------------------------------------------------------
64   * Argument      : void
65   *-----------------------------------------------------------------------------
66   * Return Value  : void
67   *-----------------------------------------------------------------------------
68   * Note          :
69   *""FUNC COMMENT END""*************************************************************/
70  void io_init_bsc_cs0(void)
71  {
72
73    /* ==== PFC settings ==== */
74    PORT.PACRH1.WORD = 0x1554;      /* Set A17-A22 */
75
76    /* ==== CS0BCR settings ==== */
77    BSC.CS0BCR.LONG = 0x10000400UL;
78                                /* Idle Cycles between Write-read Cycles */
79                                /*  and Write-write Cycles :1idle cycles */
80                                /* Data Bus Size:16-bit size */
81
82    /* ==== CS0WCR settings ====  */
83    BSC.CS0WCR.LONG = 0x00000ac1UL;
84                                /* Number of Delay Cycles from Adress, */
85                                /*  CS0# Assertion to RD#,WEn Assertion */
86                                /*  :1.5cycles */
87                                /* Number of Access Wait Cycles:5cycles */
88                                /* Delay Cycles from RD,WEn# negation to */
89                                /*  Address,CSn# negation:1.5cycles */
90  }
91
92  /* End of File */
93
```

4.10 Sample Program Listing: "bscsdram.c" (1)

```
/***********************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products. No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
***********************************************************************
* (C) 2007(2010) Renesas Electronics Corporation. All rights reserved.
*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7671 Sample Program
*   File Name   : bscsdram.c
*   Abstract    : SH7671 Initial Settings
*   Version     : 1.02.02
*   Device      : SH7671
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                               (Ver.9.01 Release01).
*   OS          : None
*   H/W Platform: M3A-HS71(CPU board)
*   Description :
***********************************************************************
*   History     : Jul.04,2007 ver.1.00.00
*              : Oct.29,2007 ver.1.00.01 Modification due to change of iodefine.h(v1.00.00)
*               : Jan.17,2008 ver.1.00.02 Wait change
*               : Feb.07,2008 ver.1.01.00 Changed to be after refresh start mode setting
*               : Mar.03,2008 ver.1.02.00 Setting procedure is unified
*               : Dec.18,2009 ver.1.02.01 Updated header comments
*               : Apr.07,2010 ver.1.02.02 Changed the company name and device name
*""FILE COMMENT END""*************************************************/
#include "iodefine.h"
#include "defs.h"

```

4.11 Sample Program Listing: "bscsdram.c" (2)

```
53    /* ==== Macro name definition ==== */
54
55    /* The address when writing in a SDRAM mode register */
56    #define SDRAM_MODE      (*(volatile unsigned short *)(0xfffc5080))
57
58    /* ==== Prototype Declaration ==== */
59    void io_init_sdram(void);
60
61    #pragma section ResetPRG
62    /*""FUNC COMMENT""*****************************************************************
63     * ID            :
64     * Outline       : SDRAM 16 bit bus width connection settings
65     *-----------------------------------------------------------------------------
66     * Include       : "iodefine.h", "defs.h"
67     *-----------------------------------------------------------------------------
68     * Declaration   : void io_init_sdram(void);
69     *-----------------------------------------------------------------------------
70     * Description   : A connection setup to SDRAM of CS3 space.
71     *-----------------------------------------------------------------------------
72     * Argument      : void
73     *-----------------------------------------------------------------------------
74     * Return Value  : void
75     *-----------------------------------------------------------------------------
76     * Note          :
77     *""FUNC COMMENT END""*************************************************************/
78    void io_init_sdram(void)
79    {
80
81        volatile int j = LOOP_100us*2;          /* 200usec wait */
82
83        /* ==== 200us interval elapsed ? ==== */
84        while(j-- > 0){
85            /* wait */
86        }
87
88        /* ==== CS3BCR settings ==== */
89        BSC.CS3BCR.LONG = 0x00004600ul; /* Idle Cycles between Write-read Cycles
90                                           and Write-write Cycles :2idle cycles */
91                                        /* Memory type :SDRAM */
92                                        /* Data Bus Size :32-bit size */
93
94
```

4.12 Sample Program Listing: "bscsdram.c" (3)

```
 95        /* ==== CS3WCR settings ==== */
 96        BSC.CS3WCR_SDRAM.LONG = 0x00002892ul;
 97                                        /* Precharge completion wait cycles
 98                                           :1cycles */
 99                                        /* Wait cycles between ACTV command
100                                           and READ(A)/WRITE(A) command :2cycles */
101                                        /* CAS latency for Area 3 :2cycles */
102                                        /* Auto-precharge startup wait cycles
103                                           :2cycles */
104                                        /* Idle cycles from REF command/self-refresh
105                                           Release to ACTV/REF/MRS command
106                                           :5cycles */
107
108        /* ==== SDCR settings ==== */
109        BSC.SDCR.LONG = 0x00000811ul;   /*
110                                            Refresh Control :Refresh start
111                                            RMODE :Auto-refresh is performed
112                                            BACTV :Auto-precharge mode
113                                            Row address for Area3 :13-bits
114                                            Column Address for Area3 :9-bits
115                                        */
116
117        /* ==== RTCOR settings ==== */
118        BSC.RTCOR.LONG = 0xa55a0020ul;  /*
119                                            7.8usec /240nsec
120                                            >= 32(0x20)cycles per refresh
121                                        */
122
123        /* ==== RTCSR settings ==== */
124        BSC.RTCSR.LONG = 0xa55a0010ul;
125                                        /*
126                                            Initialization sequence start
127                                            Clock select B-phy/16 = 240nsec
128                                            Refresh count :Once
129                                        */
130        /* ==== Written in SDRAM Mode Register ==== */
131        SDRAM_MODE = 0;                 /*
132                                          SDRAM mode register setting(CS3 area)
133                                          dummy write
134                                          burst read / burst write (burst length 1)
135                                       */
136    }
137    /* End of File */
```

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

```
/***********************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corp. and is only
*   intended for use with Renesas products. No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corp. and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORP. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
***********************************************************************
* (C) 2007(2010) Renesas Electronics Corporation. All rights reserved.
*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7671 Sample Program
*   File Name   : cache.c
*   Abstract    : sample of cache register
*   Version     : 1.01.02
*   Device      : SH7671
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                               (Ver.9.01 Release01).
*   OS          : None
*   H/W Platform: M3A-HS71(CPU board)
*   Description :
***********************************************************************
*   History     : Jul.06,2007 ver.1.00.00
*               : Jul.09,2007 ver.1.01.00 Changed of function names
*                                         Function deleted (cache_writeback)
*                                         The A bit is used
*                                         Changed in section allocation
*               : Dec.18,2009 ver.1.01.01 Updated header comments
*               : Apr.07,2010 ver.1.01.02 Changed the company name and device name
*""FILE COMMENT END""*************************************************/
#include <machine.h>
#include "defs.h"
#include "iodefine.h"

```

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

```
54    /* ==== Prototype Declaration ==== */
55    void io_init_cache(void);
56    int io_cache_writeback(void);
57
58    #pragma section CACHE     /* It is placed in the CS0 cache-disabled space */
59    /*""FUNC COMMENT""*****************************************************************
60     * ID           :
61     * Outline      : Cache initialization
62     *-----------------------------------------------------------------------------
63     * Include      : iodefine.h
64     *-----------------------------------------------------------------------------
65     * Declaration  : void io_init_cache(void);
66     *-----------------------------------------------------------------------------
67     * Description  : Instruction/operand cache are flushed and enabled.
68     *              : The section name of this function is changed to be placed in
69     *              : the cache-disabled.
70     *              : When this function is used only in the state of interrupt level 15,
71     *              : the setting and clearing of interrupt mask need not be processed.
72     *-----------------------------------------------------------------------------
73     * Argument     : void
74     *-----------------------------------------------------------------------------
75     * Return Value : void
76     *-----------------------------------------------------------------------------
77     * Note         : None
78     *""FUNC COMMENT END""*************************************************************/
79    void io_init_cache(void)
80    {
81      volatile unsigned long reg;
82      int mask;
83
84      /* ==== Interrupt mask setting ==== */
85      mask = get_imask();
86      set_imask(15);                 /* Set to the level 15 */
87
88      /* ==== Cache register setting ==== */
89      CCNT.CCR1.LONG = 0x0909ul;  /* Write back ON */
90
91                                  /*
92                                      ICF=1:Instruction cache flushed
93                                      ICE=1:Instruction cache enabled
94                                      OCF=1:Operand cache flushed
95                                      OCE=1:Operand cache enabled
96                                  */
97      /* ==== Reading cache register ==== */
98      reg = CCNT.CCR1.LONG ;
99
100     /* ==== Clearing interrupt mask ==== */
101     set_imask(mask);               /* Set to the original level */
102
103   }
104
```

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

```
105    /*""FUNC COMMENT""*****************************************************************
106     * ID           :
107     * Outline      : Write-back of cache
108     *-------------------------------------------------------------------------------
109     * Include      : iodefine.h
110     *-------------------------------------------------------------------------------
111     * Declaration  : int io_cache_writeback(void);
112     *-------------------------------------------------------------------------------
113     * Description  : All lines of operand cache are disabled, and the contents of
114     *              : cache memory are written back to the external memory.
115     *              : It has nothing to do with the write-through mode.
116     *-------------------------------------------------------------------------------
117     * Argument     : void
118     *-------------------------------------------------------------------------------
119     * Return Value : 0 : Normal completion
120     *-------------------------------------------------------------------------------
121     * Note         : None
122     *""FUNC COMMENT END""*************************************************************/
123    int io_cache_writeback(void)
124    {
125      volatile unsigned long *arry;
126      unsigned int i,j;
127      int mask;
128
129      /* ==== Interrupt mask setting ==== */
130      mask = get_imask();
131      set_imask(15);                 /* Set to the level 15 */
132
133      /* ====  All entries disabled ==== */
134      for(i=0u; i <4u; i++){
135        for(j=0u; j < 128u; j++){
136            /* ---- Creating an address array address ---- */
137            arry = (volatile unsigned long *)(0xf0800000 | (i<<11ul) | (j<<4ul) | 0x8 );
138            /* ---- Write U=0 and V=0 in the address array ---- */
139            *arry &= 0xffffffcul;     /* V=0,U=0 */
140        }
141      }
142
143      /* ==== Interrupt mask recovery ==== */
144      set_imask(mask);            /* Set to the original level */
145
146      return 0;
147    }
148
149    /* End of File */
150
```

5. Documents for Reference

- Software Manual
  SH-2A/SH2A-FPU Software Manual (REJ09B0051)
  The most up-to-date version of this document is available on the Renesas Electronics Website.
- Hardware Manuals
  SH7670 Group Hardware Manual (REJ09B0437)
  The most up-to-date version of this document is available on the Renesas Electronics Website.

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Nov.28.08	—	First edition issued
1.01	May 07.10	4	AC Switching Characteristics are removedd
		—	Format is changed

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: +49-211-6503-0, 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, R.O.C.
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