SH7239 Group

Example of Initialization

R01AN0297EJ0100
Rev. 1.00
Dec. 15, 2010

Summary

This application note gives an example of configuration items to activate the SH7239 Microcomputers (MCUs).

Target Device

SH7239 MCU

Contents

1. Introduction

1.1 Specifications

Configure the clock pulse generator (CPG) after the reset is canceled.

1.2 Modules Used

- Clock pulse generator (CPG)

1.3 Applicable Conditions

MCU	SH7239 (R5F72395ADFP)
Power Supply Voltage	3.3 V
Operating Frequency	Internal clock: 160 MHz Bus clock: 40 MHz Peripheral clock: 40 MHz
Integrated Development Environment	Renesas Electronics High-performance Embedded Workshop Ver.4.07.00
C Compiler	Renesas Electronics SuperH RISC engine Family C/C++ compiler package Ver.9.03 Release 02
Compiler Options	Default setting in the High-performance Embedded Workshop (-cpu=sh2afpu - fpu=single -debug -gbr=auto -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1)

2. Applications

Configuration program for the minimum hardware setup is required to execute the main function created in C code. This application note describes the configuration example for the configuration program.

All of the SH7239 application notes assume to use the sample program described in this application note as the configuration program.

2.1 Sample Program

The configuration program consists of several source files such as the resetprg.c, describing the PowerON_Reset_PC function, and the hwsetup.c, describing the hardware setup function. Main source files are as follows.

- resetprg.c
- hwsetup.c
- cpg.c

"resetprg.c" is a source file created on the file automatically generated by the High-performance Embedded Workshop, and describes the PowerON_ResetPC function. The PowerON_ResetPC function initially executed after the reset is canceled. Its beginning address is set in the reset vector defined by the vecttbl.c.

"hwsetup.c" describes the HardwareSetup function called by the PowerON_Reset_PC function. The HardwareSetup function calls the io_set_cpg function to set the CPG. When using the external bus interface such as interfacing SDRAM, call the io_set_cpg function, and then add processing to set the Bus State Controller (BSC) to the HardwareSetup function as appropriate.

"cpg.c" describes the io_set_cpg function which is called from the HardwareSetup function. The io_set_cpg function sets the Frequency control register (FRQCR) in the program on the on-chip RAM. The sample program execute the _seccpy function to copy the program section to set the FRQCR (section name: PURAM) from on-chip ROM to on-chip RAM at the beginning of the io_set_cpg function, and sets the FRQCR by the io_set_cpg_frqcr function. After setting the FRQCR, set the MTU clock frequency control register (MCLKCR) and the AD clock frequency control register (ACLKCR) to clear the module standby function for internal peripheral modules.

Figure 1 to Figure 4 show flow charts of the configuration program in above source files used in this application.

Supplement: About the stack area

CPU can access pages 0 and 1 in the SH7239 on-chip RAM in one cycle both in reading and writing. This application allocates the stack area at the end of page 1 in the on-chip RAM (address: H'FFF8 4000 to H'FFF8 7FFF) to support the SH7239 high-speed access performance.

The stack area is allocated as section S, which can be set in the High-performance Embedded Workshop. On the [Build] menu, open the [SuperH RISC engine Standard Toolchain] dialog box, and select [Link/Library] tab. Then, select "Section" from the "Category" drop-down list. When it is not required to support the SH7239 high-speed access performance, the stack area can be allocated to pages 4 or 5 in the on-chip RAM. Reallocate the stack area to sections according to the system.

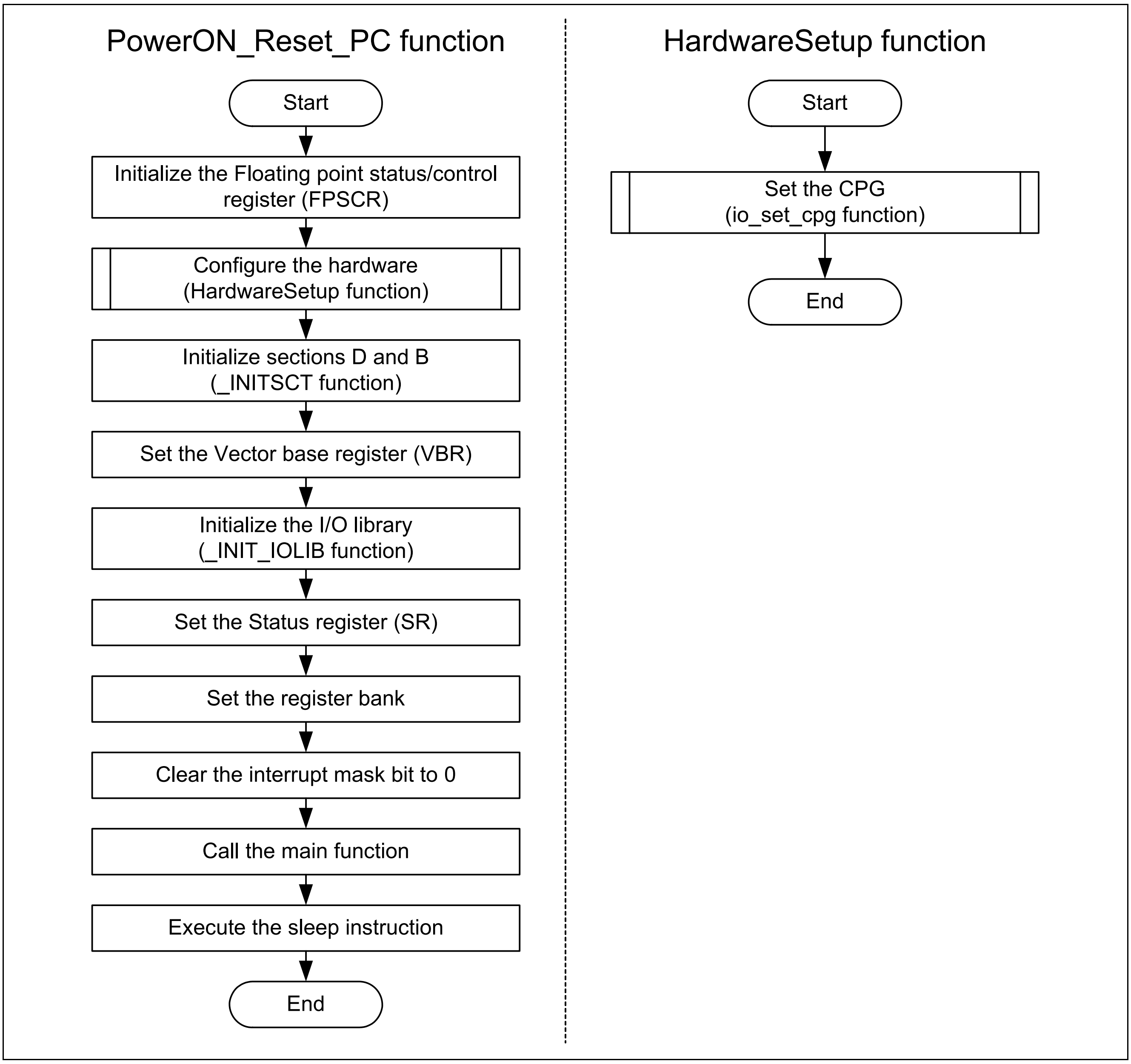

PowerON_Reset_PC function
Start
Initialize the Floating point status/control register (FPSCR)
Configure the hardware (HardwareSetup function)
Initialize sections D and B (_INITSCT function)
Set the Vector base register (VBR)
Initialize the I/O library (_INIT_IOLIB function)
Set the Status register (SR)
Set the register bank
Clear the interrupt mask bit to 0
Call the main function
Execute the sleep instruction
End
HardwareSetup function
Start
Set the CPG (io_set_cpg function)
End

Figure 1 Flow Charts of Functions (PowerON_Reset_PC, HardwareSetup)

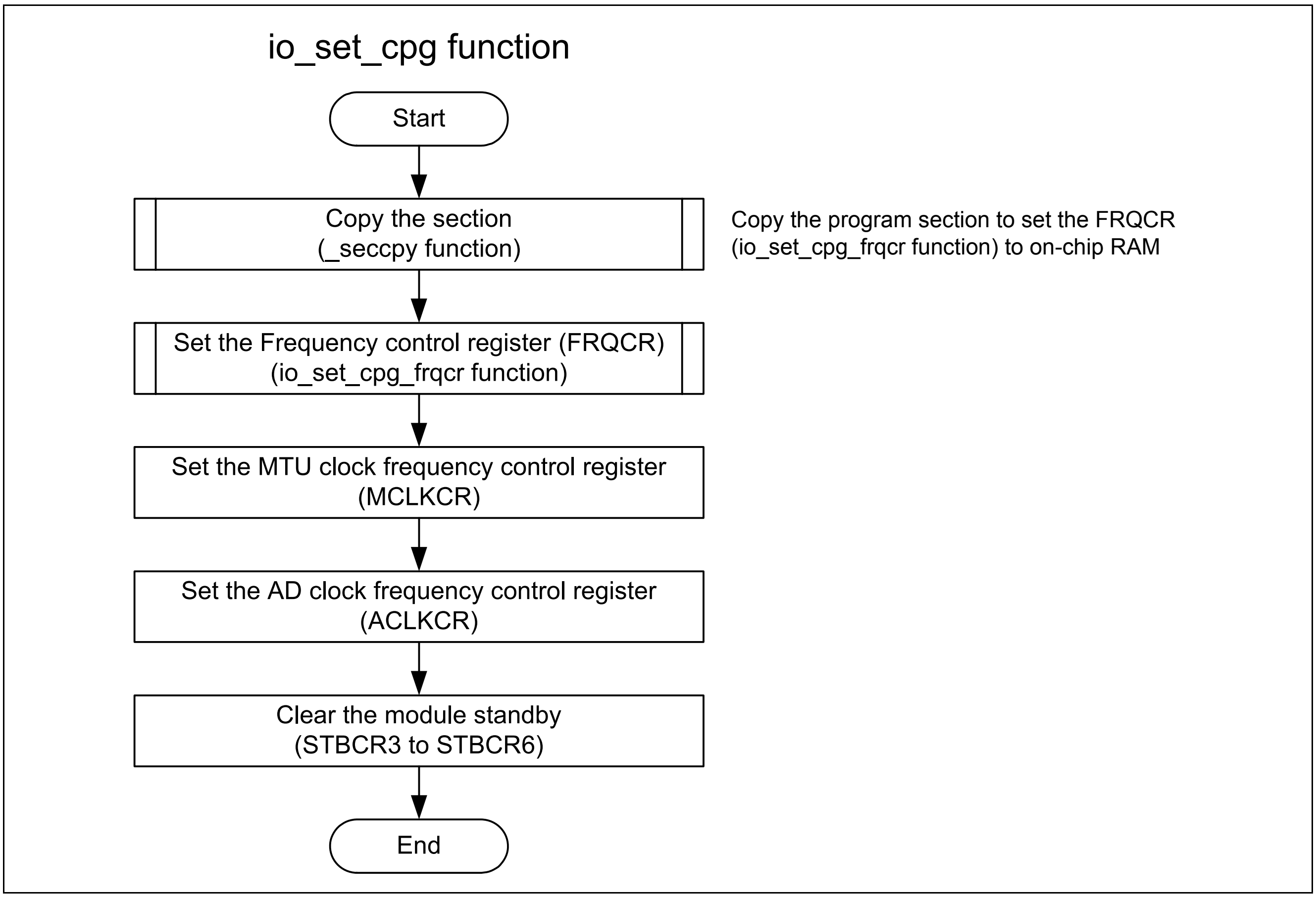
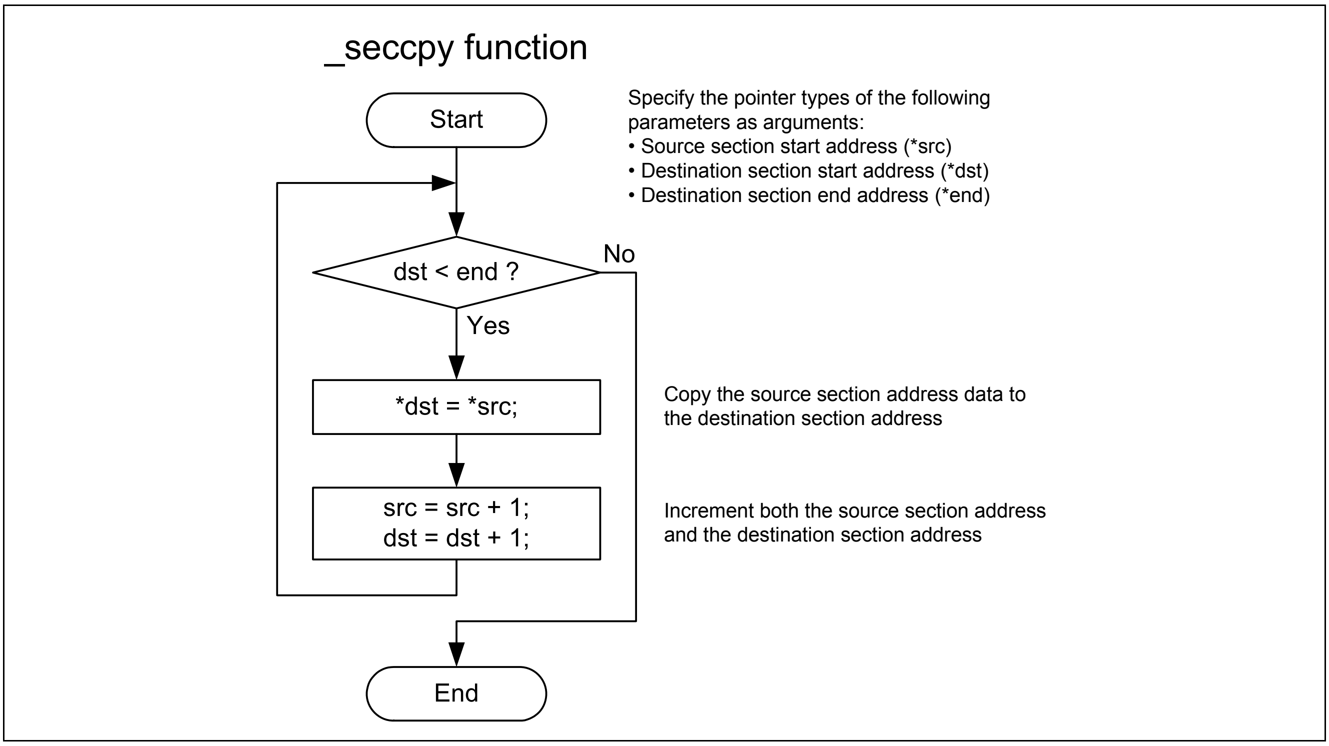

io_set_cpg function
Start
Copy the section
(_seccpy function)
Set the Frequency control register (FRQCR)
(io_set_cpg_frqcr function)
Set the MTU clock frequency control register
(MCLKCR)
Set the AD clock frequency control register
(ACLKCR)
Clear the module standby
(STBCR3 to STBCR6)
End
Copy the program section to set the FRQCR
(io_set_cpg_frqcr function) to on-chip RAM

Figure 2 Flow Chart for Setting the CPG (io_set_cpg function)



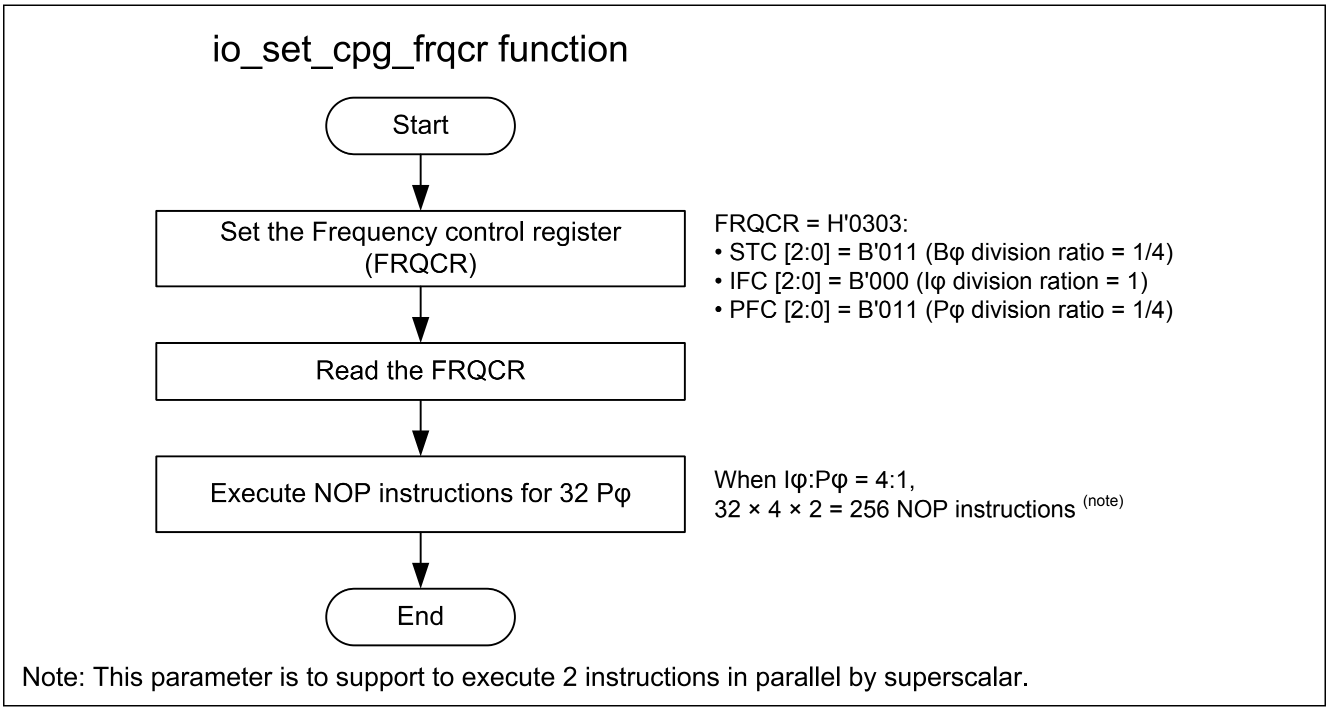
Specify the pointer types of the following parameters as arguments:

- Source section start address (*src)
- Destination section start address (*dst)
- Destination section end address (*end)

Copy the source section address data to the destination section address

Increment both the source section address and the destination section address

Figure 3 Flow Chart for Copying the Section (_seccpy function)


io_set_cpg_frqcr function
Start
Set the Frequency control register (FRQCR)
Read the FRQCR
Execute NOP instructions for 32 Pφ
End

FRQCR = H'0303:

- STC [2:0] = B'011 (Bφ division ratio = 1/4)
- IFC [2:0] = B'000 (Iφ division ration = 1)
- PFC [2:0] = B'011 (Pφ division ratio = 1/4)

When Iφ:Pφ = 4:1,
32 × 4 × 2 = 256 NOP instructions (note)

Note: This parameter is to support to execute 2 instructions in parallel by superscalar.

Figure 4 Flow Chart for Setting the FRQCR (io_set_cpg_frqcr function)

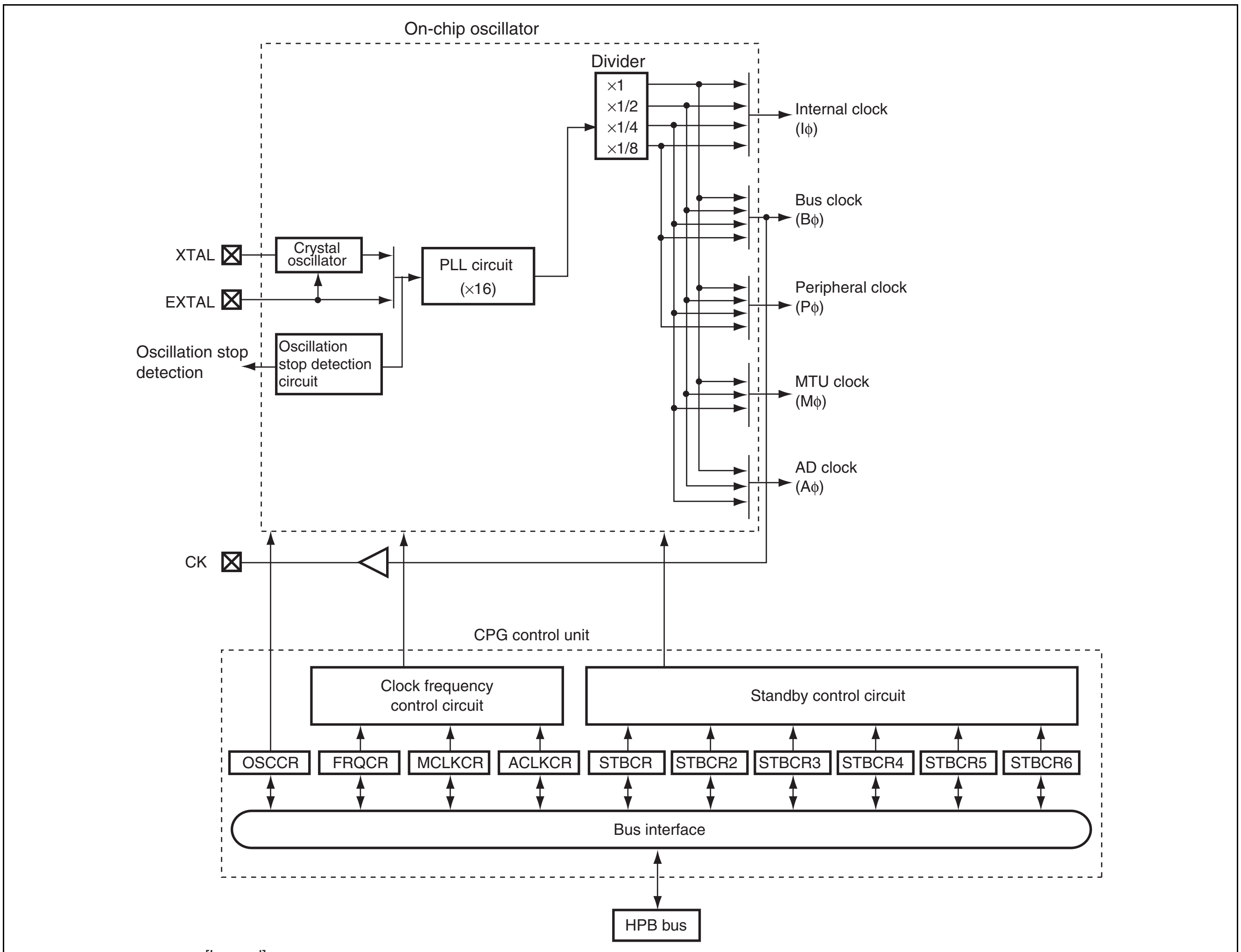
2.2 CPG Operation

CPG generates the internal clock (Iφ), bus clock (Bφ), peripheral clock (Pφ), MTU clock (Mφ), and AD clock (Aφ), as well as controlling power-down mode.

The following table gives an overview of the CPG. Figure 5 shows the CPG block diagram.

Table 1 CPG Overview

Item	Description	
Generate clock	• Internal clock (Iφ): • Bus clock (Bφ): • Peripheral clock (Pφ): • MTU clock (Mφ): • AD clock (Aφ):	Used by the CPU Used by the external bus interface Used by the internal peripheral module Used by the MTU2/MTU2S Used by the ADC module
Change frequency	• Sets frequencies for clocks independently using the PLL (Phase Locked Loop) and divider circuits in the CPG. • Changes frequency by software using the frequency control registers (FRQCR, MCLKCR, and ACLKCR).	
Control power-down mode	Stops clock in sleep mode or software standby mode. Stops the module specified by module standby function.	



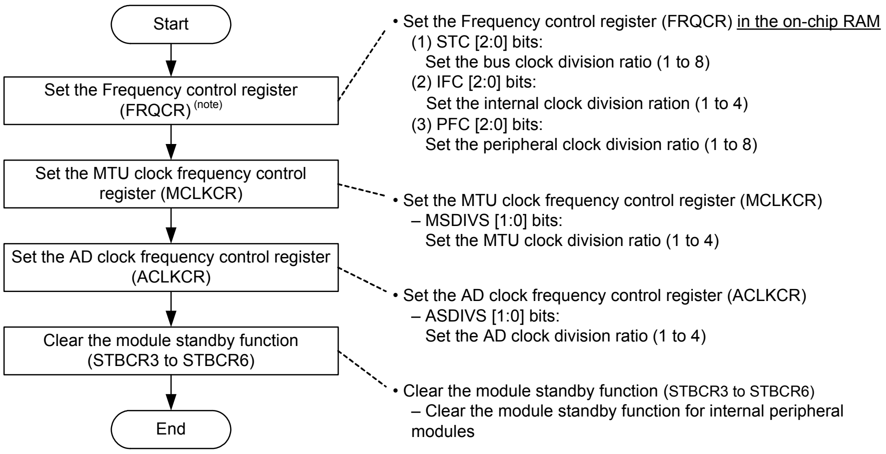
[Legend]

FRQCR:	Frequency control register
MCLKCR:	MTU clock frequency control register
ACLKCR:	AD clock frequency control register
STBCR:	Standby control register
STBCR2:	Standby control register 2
STBCR3:	Standby control register 3
STBCR4:	Standby control register 4
STBCR5:	Standby control register 5
STBCR6:	Standby control register 6
OSCCR:	Oscillation stop detection control register

Figure 5 CPG Block Diagram

2.3 CPG Setting

The figure below shows the flow chart of setting CPG. Internal peripheral modules are in module standby mode after the reset is canceled. The sample program clears the module standby function for internal peripheral module after setting the Frequency control register (FRQCR), MTU clock frequency control register (MCLKCR), and AD clock frequency control register (ACLKCR). For details on these registers, refer to the Clock Pulse Generator (CPG) chapter in the SH7239 Group, SH7237 Group Hardware User's Manual.


Start
Set the Frequency control register (FRQCR) (note)
Set the MTU clock frequency control register (MCLKCR)
Set the AD clock frequency control register (ACLKCR)
Clear the module standby function (STBCR3 to STBCR6)
End

- Set the Frequency control register (FRQCR) in the on-chip RAM
  (1) STC [2:0] bits:
  Set the bus clock division ratio (1 to 8)
  (2) IFC [2:0] bits:
  Set the internal clock division ration (1 to 4)
  (3) PFC [2:0] bits:
  Set the peripheral clock division ratio (1 to 8)
- Set the MTU clock frequency control register (MCLKCR)
  – MSDIVS [1:0] bits:
  Set the MTU clock division ratio (1 to 4)
- Set the AD clock frequency control register (ACLKCR)
  – ASDIVS [1:0] bits:
  Set the AD clock division ratio (1 to 4)
- Clear the module standby function (STBCR3 to STBCR6)
  – Clear the module standby function for internal peripheral modules

Note: Make sure to set the FRQCR in the program located in the on-chip RAM. FRQCR can only be accessed in words. Read the FRQCR and verify if its value is the same as the set value, and then execute NOP instructions for 32 Pφ. This application reads the FRQCR immediately after setting the FRQCR, and executes NOP instructions for 32 Pφ (Refer to Figure 4 FRQCR flow chart).

Figure 6 Flow Chart of CPG Setting

2.4 Setting in the Sample Program

Table 2 lists the setting in the sample program. Table 3 and Table 4 list register settings for each module.

Table 2 Module Setting in the Sample Program

Module	Setting
Floating point status/control unit (FPU)	• Precision mode Executes floating-point instructions in single-precision • Round mode Round to zero
Clock pulse generator (CPG)	• Clock frequency (input clock is 10 MHz) — Internal clock: 160 MHz — Bus clock: 40 MHz — Peripheral clock: 40 MHz — MTU clock: 80 MHz — AD clock: 40 MHz • Modules cleared the module standby function MTU2S, MTU2, ADC0, ADC1, ADC2, CMT, SCI0, SCI1, SCI2, SCIF3, RSPI, RCAN-ET

Table 3 CPG Register Settings (1/2)

Register Name	Address	Setting	Description
Frequency control register (FRQCR)	H'FFFE 0010	H'0303	• STC [2:0] = "B'011": Bus clock (Bφ) division ratio: 4 • IFC [2:0] = "B'000": Internal clock (Iφ) division ratio = 1 • PFC [2:0] = "B'011": Peripheral clock (Pφ) division ratio = 4
MTU clock frequency control register (MCLKCR)	H'FFFE 0410	H'41	• MSDIVS [1:0] = "B'01": MTU clock (Mφ) division ratio = 2
AD clock frequency control register (ACLKCR)	H'FFFE 0414	H'43	• ASDIVS [1:0] = "B'11": AD clock (Aφ) division ratio = 4

Table 4 CPG Register Settings (2/2)

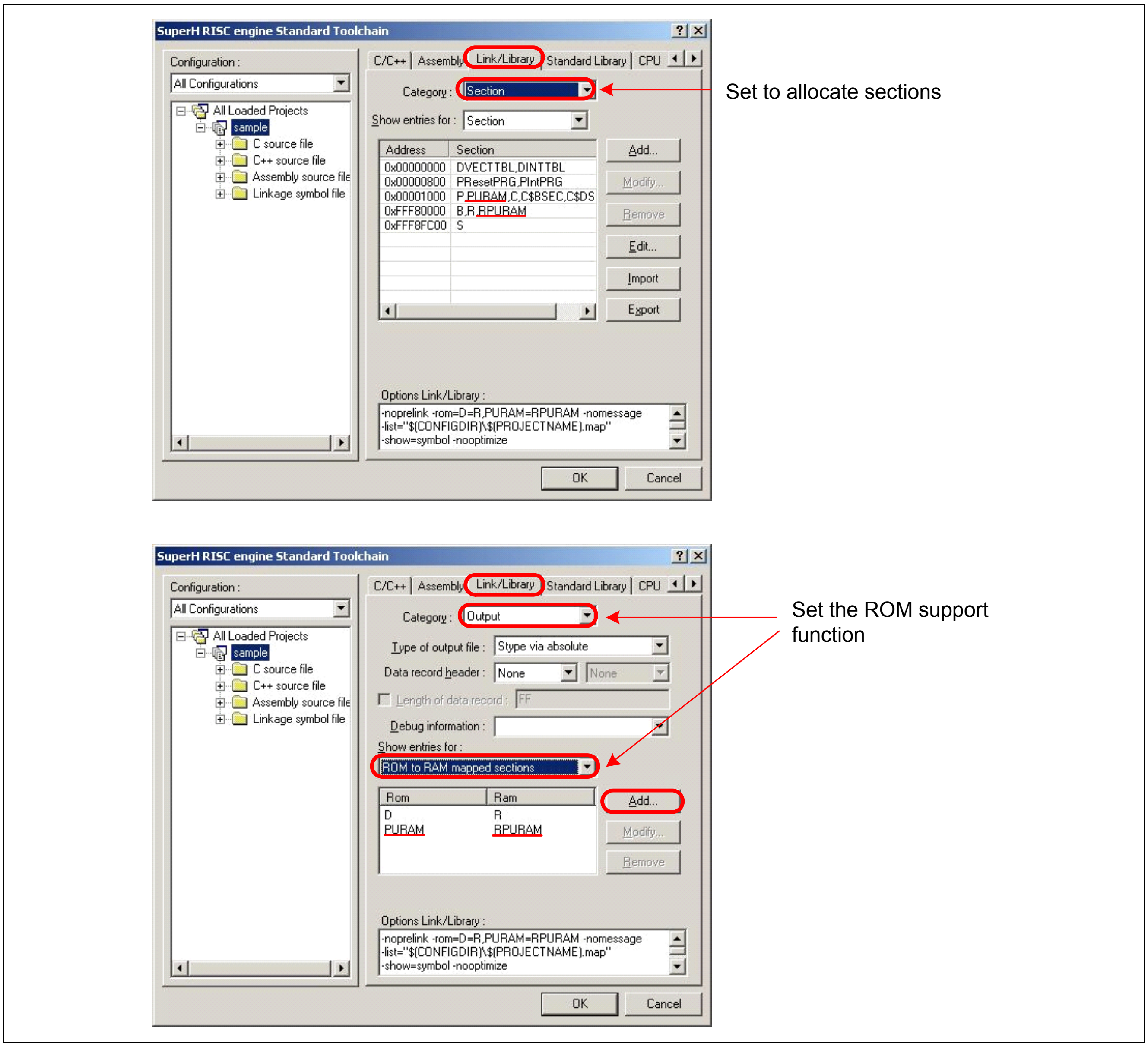
Register Name	Address	Setting	Description
Standby control register 3 (STBCR3)	H'FFFE 0408	H'1A	• HIZ = "0": The pin state is held in software standby mode • MSTP36 = "0": MTU2S is operating • MSTP35 = "0": MTU2 is operating • MSTP32 = "0": ADC0 is operating • MSTP30 = "0": Flash memory is operating
Standby control register 4 (STBCR4)	H'FFFE 040C	H'E3	• MSTP44 = "0": SCIF3 is operating • MSTP42 = "0": CMT is operating
Standby control register 5 (STBCR5)	H'FFFE 0418	H'18	• MSTP57 = "0": SCI0 is operating • MSTP56 = "0": SCI1 is operating • MSTP55 = "0": SCI2 is operating • MSTP52 = "0": ADC1 is operating • MSTP51 = "0": ADC2 is operating • MSTP50 = "0": RSPI is operating
Standby control register 6 (STBCR6)	H'FFFE 041C	H'CF	• MSTP64 = "0": RCAN-ET is operating

Supplement: About the ROM support function

This application copies the program section to set the FRQCR (io_set_cpg_frqcr function) from the on-chip ROM to on-chip RAM. The ROM support function must be set by the C compiler optimizing linkage editor to add such copy processing.

On the [Build] menu of the High-performance Embedded Workshop, open the [SuperH RISC engine Standard Toolchain] dialog box, and select [Link/Library] tab. Select "Output" from the "Category" drop-down list, and specify the "Show entries for" as "ROM to RAM mapped sections". Click "Add", specify the source section as the ROM section, and the destination section as the RAM section. Before setting the ROM support function, set where to allocate sections both in the source and destination in the "Category" drop-down list on the [Link/Library] tab. This application sets "PURAM" as the program section to set the FRQCR, and "RPURAM" as the destination RAM section.

Figure 7 shows an example of setting the ROM support function. For more information, refer to the SuperH C/C++ Compiler Package V.9.04 User's Manual.



Set to allocate sections

Category : Section

Show entries for : Section

Address	Section
0x00000000	DVECTTBL,DINTTBL
0x00000800	PResetPRG,PIntPRG
0x00001000	P,PURAM,C,C$BSEC,C$DS
0xFFF80000	B,R,RPURAM
0xFFF8FC00	S

Options Link/Library :

```
-noprelink -rom=D=R,PURAM=RPURAM -nomessage
-list="$(CONFIGDIR)\$(PROJECTNAME).map"
-show=symbol -nooptimize
```

Set the ROM support function

Category : Output

Type of output file : Stype via absolute

Data record header : None None

□ Length of data record : FF

Debug information :

Show entries for : ROM to RAM mapped sections

Rom	Ram
D	R
PURAM	RPURAM

Options Link/Library :

```
-noprelink -rom=D=R,PURAM=RPURAM -nomessage
-list="$(CONFIGDIR)\$(PROJECTNAME).map"
-show=symbol -nooptimize
```

Figure 7 Example to Set the ROM Support Function

3. Sample Program Listing

3.1 Sample Program Listing "resetprg.c" (1/4)

```
/*******************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corporation and is only
*   intended for use with Renesas products.  No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corporation 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 CORPORATION 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
********************************************************************************
*   Copyright (C) 2010 Renesas Electronics Corporation. All rights reserved.
*******************************************************************************/
/*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7239 Sample Program
*   File Name    : resetprg.c
*   Abstract     : SH7239 Initial Setting
*   Version      : 1.01.00
*   Device       : SH7239
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 :                                   (Ver.9.03 Release02).
*   OS            : None
*   H/W Platform: R0K572390 (CPU board)
*   Description :
********************************************************************************
*   History      : Aug.20,2010 Ver.1.00.00
*                 : Oct.20,2010 Ver.1.01.00  Add the IO library initialization
*""FILE COMMENT END""**********************************************************/
```

3.2 Sample Program Listing "resetprg.c" (2/4)

```
#include <machine.h>
#include <_h_c_lib.h>
#include "stacksct.h"
#include "iodefine.h"

/* ==== Macro definition ==== */
#define FPSCR_Init 0x00040001
#define SR_Init    0x000000f0
#define INT_OFFSET 0x10

/* ==== Prototype declaration ==== */
void PowerON_Reset_PC(void);
void Manual_Reset_PC(void);

/* ==== External reference declaration ==== */
/* ---- Function prototype ---- */
extern void main(void);
extern void HardwareSetup(void);
/* ---- Global variable ---- */
extern unsigned int INT_Vectors;

/* ==== Section name changed to ResetPRG ==== */
#pragma section ResetPRG

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

/*""FUNC COMMENT""**************************************************************
 * ID             :
 * Outline      : CPU initialization
 *------------------------------------------------------------------------------
 * Include      : <machine.h>, <_h_c_lib.h>, and "iodefine.h"
 *------------------------------------------------------------------------------
 * Declaration  : void PowerON_Reset_PC(void);
 *------------------------------------------------------------------------------
 * Description  : Executes the CPU initialization processing to register
 *                : the power-on reset vector to the exception vector table.
 *------------------------------------------------------------------------------
 * Argument       : void
 *------------------------------------------------------------------------------
 * Return Value : void
 *------------------------------------------------------------------------------
 * Note           : This function is executed first after power-on reset.
 *""FUNC COMMENT END""**********************************************************/
```

3.3 Sample Program Listing "resetprg.c" (3/4)

```
void PowerON_Reset_PC(void)
{
  /* ==== Floating Point Status/Control Register setting ==== */
  set_fpscr(FPSCR_Init);

  /* ==== Hardware initialization ==== */
  HardwareSetup();            /* HardwareSetup function */

  /* ==== Sections initialization ==== */
  _INITSCT();

  /* ==== Vector Base Register setting ==== */
  set_vbr((void *)((char *)&INT_Vectors - INT_OFFSET));

  /* ==== IO library initialization ==== */
  _INIT_IOLIB();

  /* ==== Status Register setting ==== */
  set_cr(SR_Init);
  nop();

  /* ==== Bunk Number Register setting ==== */
  INTC.IBNR.BIT.BE = 1;           /* Use of register banks enabled for all */
                                  /*  interrupts except NMI and user break */

  /* ==== Interrupt mask bits clear ==== */
  set_imask(0);

  /* ==== Main function call ==== */
  main();

  /* ==== Sleep instruction execution ==== */
  sleep();
}

```

3.4 Sample Program Listing "resetprg.c" (4/4)

```
/*""FUNC COMMENT""***************************************************************
 * ID            :
 * Outline      : Manual reset processing
 *-------------------------------------------------------------------------------
 * Include      :
 *-------------------------------------------------------------------------------
 * Declaration  : void Manual_Reset_PC(void);
 *-------------------------------------------------------------------------------
 * Description  : Registers the manual reset vector to the exception vector
 *                : table.
 *-------------------------------------------------------------------------------
 * Argument      : void
 *-------------------------------------------------------------------------------
 * Return Value : void
 *-------------------------------------------------------------------------------
 * Note           : This sample does not describe the processing content at all.
 *                 : Add the program in this function as needed.
 *""FUNC COMMENT END""**********************************************************/
void Manual_Reset_PC(void)
{
  /* NOP */
}

/* END of File */
```

3.5 Sample Program Listing "hwsetup.c" (1/2)

```
/*******************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corporation and is only
*   intended for use with Renesas products.  No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corporation 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 CORPORATION 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
********************************************************************************
*   Copyright (C) 2010 Renesas Electronics Corporation. All rights reserved.
*******************************************************************************/
/*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7239 Sample Program
*   File Name    : hwsetup.c
*   Abstract     : Hardware Function Initial Setting
*   Version      : 1.00.00
*   Device       : SH7239
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 :                                    (Ver.9.03 Release02).
*   OS            : None
*   H/W Platform: R0K572390 (CPU board)
*   Description :
********************************************************************************
*   History      : Aug.20,2010 Ver.1.00.00
*""FILE COMMENT END""**********************************************************/
#include "iodefine.h"

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

```

3.6 Sample Program Listing "hwsetup.c" (2/2)

```
/* ==== External reference ==== */
/* ---- Function prototype ---- */
extern void io_set_cpg(void);

/*""FUNC COMMENT""***************************************************************
 * ID            :
 * Outline      : Hardware initialization
 *-------------------------------------------------------------------------------
 * Include      :
 *-------------------------------------------------------------------------------
 * Declaration  : void HardwareSetup(void);
 *-------------------------------------------------------------------------------
 * Description  : Initializes the hardware function.
 *-------------------------------------------------------------------------------
 * Argument      : void
 *-------------------------------------------------------------------------------
 * Return Value : void
 *-------------------------------------------------------------------------------
 * Note           : None
 *""FUNC COMMENT END""************************************************************/
void HardwareSetup(void)
{
  /* ==== CPG setting ==== */
  io_set_cpg();
}

/* End of File */
```

3.7 Sample Program Listing "cpg.c" (1/5)

```
/*******************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corporation and is only
*   intended for use with Renesas products.  No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corporation 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 CORPORATION 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
********************************************************************************
*   Copyright (C) 2010 Renesas Electronics Corporation. All rights reserved.
*******************************************************************************/
/*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7239 Sample Program
*   File Name    : cpg.c
*   Abstract     : CPG Setting Processing
*   Version      : 1.01.00
*   Device       : SH7239
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 :                                  (Ver.9.03 Release02).
*   OS            : None
*   H/W Platform: R0K572390 (CPU board)
*   Description :
********************************************************************************
*   History      : Aug.20,2010 Ver.1.00.00
*                 : Oct.20,2010 Ver.1.01.00  - Divide the FRQCR setting into
*                 :                                subroutine "io_set_cpg_frqcr"
*                 :                                which is allocated to on-chip RAM
*                 :                             - Add the processing of section copy
*                 :                                function "_seccpy"
*                 :                             - Modify comment
*""FILE COMMENT END""**********************************************************/
```

3.8 Sample Program Listing "cpg.c" (2/5)

```
#include <machine.h>
#include "iodefine.h"
#include "cpumodel.h"

/* ==== Prototype declaration ==== */
void io_set_cpg(void);
void io_set_cpg_frqcr(void);
static void _seccpy(unsigned long *src, unsigned long *dst, unsigned long *end);

/*""FUNC COMMENT""**************************************************************
 * ID            :
 * Outline      : CPG setting
 *------------------------------------------------------------------------------
 * Include      : "iodefine.h"
 *------------------------------------------------------------------------------
 * Declaration  : void io_set_cpg(void);
 *------------------------------------------------------------------------------
 * Description  : Initializes the clock pulse generator (CPG) as follows:
 *                :  SH7239A: I-clock = 160MHz, B-clock = 40MHz, P-clock = 40MHz,
 *                :             M-clock =  80MHz, and A-clock = 40MHz.
 *                :  SH7239B: I-clock = 100MHz, B-clock = 50MHz, P-clock = 50MHz,
 *                :             M-clock = 100MHz, and A-clock = 50MHz.
 *                : And then supplies clock to all peripheral modules.
 *------------------------------------------------------------------------------
 * Argument      : void
 *------------------------------------------------------------------------------
 * Return value : void
 *------------------------------------------------------------------------------
 * Note           : This function is an example of CPG setting at the input clock
 *                 : of 10MHz/12.5MHz.
 *""FUNC COMMENT END""**********************************************************/
void io_set_cpg(void)
{
  /* ==== CPG setting ==== */
  /* ---- Program section initialization for FRQCR setting ---- */
  _seccpy((unsigned long *)__sectop("PURAM"), (unsigned long *)__sectop("RPURAM"),
           (unsigned long *)__secend("RPURAM"));
           /* Program section copying from "PURAM" to on-chip RAM */
  /* ---- FRQCR setting (Running on-chip RAM) ---- */
  io_set_cpg_frqcr();         /* Clock-in =  10MHz/ 12.5MHz:  */
                                  /*  I-clock = 160MHz/100MHz,    */
                                  /*  B-clock =  40MHz/ 50MHz,    */
                                  /*  P-clock =  40MHz/ 50MHz     */
  /* ---- MCLKCR setting ---- */
  CPG.MCLKCR.BYTE = 0x41;    /*  M-clock =  80MHz/100MHz     */
  /* ---- ACLKCR setting ---- */
  CPG.ACLKCR.BYTE = 0x43;    /*  A-clock =  40MHz/ 50MHz     */


```

3.9 Sample Program Listing "cpg.c" (3/5)

```
  /* ==== Module standby clear ==== */
  /* ---- STBCR3 setting ---- */
  STB.CR3.BYTE = 0x1a; /* HIZ,MTU2S,MTU2,Reserve(1),                    */
                           /* Reserve(1),ADC0,Reserve(1),FLASH             */
  /* ---- STBCR4 setting ---- */
  STB.CR4.BYTE = 0xe3; /* Reserve(1),Reserve(1),Reserve(1),SCIF3,     */
                           /* Reserve(0),CMT,Reserve(1),Reserve(1)         */
  /* ---- STBCR5 setting ---- */
  STB.CR5.BYTE = 0x18; /* SCI0,SCI1,SCI2,Reserve(1),                     */
                           /* Reserve(1),ADC1,ADC2,RSPI                      */
  /* ---- STBCR6 setting ---- */
  STB.CR6.BYTE = 0xcf; /* Reserve(1),Reserve(1),Reserve(0),RCAN-ET,   */
                           /* Reserve(1),Reserve(1),Reserve(1),Reserve(1) */
}

/* ==== Section name changed to URAM ==== */
#pragma section URAM
/*""FUNC COMMENT""****************************************************************
 * ID            :
 * Outline      : FRQCR register setting
 *-------------------------------------------------------------------------------
 * Include      : <machine.h> and "iodefine.h"
 *-------------------------------------------------------------------------------
 * Declaration  : void io_set_cpg_frqcr(void);
 *-------------------------------------------------------------------------------
 * Description  : Initializes the clock pulse generator (CPG) as follows:
 *                :  SH7239A: I-clock = 160MHz, B-clock = 40MHz, P-clock = 40MHz.
 *                :  SH7239B: I-clock = 100MHz, B-clock = 50MHz, P-clock = 50MHz.
 *-------------------------------------------------------------------------------
 * Argument      : void
 *-------------------------------------------------------------------------------
 * Return Value : void
 *-------------------------------------------------------------------------------
 * Note           : - This function needs to be run on internal RAM.
 *                 : - This function is also an example of CPG setting at the
 *                 :    input clock of 10MHz/12.5MHz.
 *""FUNC COMMENT END""************************************************************/
void io_set_cpg_frqcr(void)
{
#ifndef CPU_MODEL_SH7239B     /* When using SH7239A  */
  CPG.FRQCR.WORD = 0x0303;     /* Clock-in =  10MHz:  */
                                   /*  I-clock = 160MHz,  */
                                   /*  B-clock =  40MHz,  */
                                   /*  P-clock =  40MHz   */
  CPG.FRQCR.WORD;              /* FRQCR readout */

  /* ---- 256 NOPs for 32 x P-clock (I:P = 4:1) ---- */
  nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
  nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
  nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
  nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
```

3.10 Sample Program Listing "cpg.c" (4/5)

```
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();

  #else                                 /* When using SH7239B    */
    CPG.FRQCR.WORD = 0x0313;    /* Clock-in =  12.5MHz:  */
                                        /*  I-clock = 100MHz,    */
                                        /*  B-clock =  50MHz,    */
                                        /*  P-clock =  50MHz     */
    CPG.FRQCR.WORD;             /* FRQCR readout */

    /* ---- 128 NOPs for 32 x P-clock (I:P = 2:1) ---- */
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
    nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
```

3.11 Sample Program Listing "cpg.c" (5/5)

```
  nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
  nop(); nop(); nop(); nop(); nop(); nop(); nop(); nop();
#endif
}
#pragma section    /* End of "URAM" section */

/*""FUNC COMMENT""**************************************************************
 * ID              :
 * Outline        : Section copy function
 *------------------------------------------------------------------------------
 * Include        :
 *------------------------------------------------------------------------------
 * Declaration    : static void _seccpy(unsigned long *src, unsigned long *dst,
 *                 :                         unsigned long *end);
 *------------------------------------------------------------------------------
 * Description    : Copies a source section to specified target.
 *------------------------------------------------------------------------------
 * Argument       : unsigned long *src ; Initial address of source section
 *                 : unsigned long *dst ; Initial address of target section
 *                 : unsigned long *end ; Final address of target section
 *------------------------------------------------------------------------------
 * Return Value : void
 *------------------------------------------------------------------------------
 * Note            : None
 *""FUNC COMMENT END""**********************************************************/
static void _seccpy(unsigned long *src, unsigned long *dst, unsigned long *end)
{
  while(dst < end){
    *dst++ = *src++;
  }
}

/* End of File */
```

3.12 Sample Program Listing "vecttbl.c" (1/2)

```
/*******************************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corporation and is only
*   intended for use with Renesas products.  No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corporation 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 CORPORATION 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
********************************************************************************
*   Copyright (C) 2010 Renesas Electronics Corporation. All rights reserved.
*******************************************************************************/
/*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7239 Sample Program
*   File Name    : vecttbl.c
*   Abstract     : Initialization for Vector Table
*   Version      : 1.00.00
*   Device       : SH7239
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.07.00).
*                 : C/C++ compiler package for the SuperH RISC engine family
*                 :                                   (Ver.9.03 Release02).
*   OS            : None
*   H/W Platform: R0K572390 (CPU board)
*   Description :
********************************************************************************
*   History      : Aug.20,2010 Ver.1.00.00
*""FILE COMMENT END""**********************************************************/
#include "vect.h"

#pragma section VECTTBL
void *RESET_Vectors[] = {
// <<VECTOR DATA START (POWER ON RESET)>>
// 0 Power On Reset PC
    (void *)PowerON_Reset_PC,
```

3.13 Sample Program Listing "vecttbl.c" (2/2)

```
// <<VECTOR DATA END (POWER ON RESET)>>
// 1 Power On Reset SP
    __secend("S"),
// <<VECTOR DATA START (MANUAL RESET)>>
// 2 Manual Reset PC
    (void *)Manual_Reset_PC,
// <<VECTOR DATA END (MANUAL RESET)>>
// 3 Manual Reset SP
    __secend("S")
};

#pragma section INTTBL
void *INT_Vectors[] = {
// 4 Illegal code
    (void *)INT_Illegal_code,

…

// 255 SCIF SCIF3 TXI3
    (void *)INT_SCIF_SCIF3_TXI3,
// xx Reserved
    (void *)Dummy
};

/* End of File */
```

4. References

- Software Manual
  SH-2A, SH2A-FPU Software Manual Rev. 3.00
  The latest version of the software manual can be downloaded from the Renesas Electronics website.

- Hardware Manual
  SH7239 Group, SH7237 Group Hardware User's Manual Rev. 1.00
  The latest version of the hardware manual can be downloaded from the Renesas Electronics website.

- Development Tool Manual
  SuperH C/C++ Compiler Package V.9.04 User's Manual Rev.1.00
  The latest version can be downloaded from 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	Dec.15.10	–	First edition issued

General Precautions in the Handling of MPU/MCU Products

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

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

Notice

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

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

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

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

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

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

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