To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

SH7206 Group

Example of Application for the Cache Locking Mode

Introduction

This application note describes an example of cache-function settings for the SH7206.

Target Device

SH7206

Contents

1. Overview

1.1 Specifications

The operand cache is set in the cache locking mode, and the data is fetched in the cache memory.

1.2 Function Used

Operand cache

1.3 Applied Conditions

- MCU: SH7206 (R5S72060)
- Operating frequency: Internal clock at 200 MHz
  Bus clock at 66.67 MHz
  Peripheral clock at 33.33 MHz
- C compiler: Manufactured by Renesas Technology Corp.
  C/C++ compiler package Version 9.00 of the SuperH RISC engine Family
- Compile options: Default settings of the High-performance Embedded Workshop (-cpu=sh2a -debug -gbr=auto -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1)

1.4 Related Application Note

Operation of the sample program in this application note has been confirmed with the setting conditions given in the application note on *Example of SH7206 Initial Configuration*. Please refer to that document when setting up this sample task.

2. Description of Application Example

Cache locking function in the operand cache is used in this sample task.

2.1 Operation Overview of Function Used

Cache locking function disables replacing the data fetched in way 2 or way 3 in the operand cache and stores the data in the cache memory. Therefore, by using the cache locking function, cache miss in the frequently-accessed memory area such as the data table and variables used for calculation can be prevented.
Cache locking function is controlled by the cache control register 2 (CCR2).

Table 1 is the overview of cache locking function. Tables 2 to 7 show the setting for each bit of cache control register 2 (CCR2) and relation of way to be replaced.

Table 1 Overview of Cache Lock Function

Item	Overview
Cache locking mode	Mode that the cache locking function can be used (in the state where the LE bit of CCR2 is set to be 1).
Lock enabled function	Ways 2 and 3 in the operand cache
Data read-in	In the cache locking mode, way i load bit is set. After setting the way i lock bit, mishit occurs by executing the prefetching instruction (PREF@Rn) (i is set to be 2 or 3).
Unit of data read-in	1 line (16 bytes)

Note: Please do not set the way 2 and way 3 load bits to 1 at the same time. Please refer to the section 'Cache' in the *SH7206 Group Hardware Manual* for details on the caches.

Table 2 Ways to be Replaced When Cache Miss Occurs by the PREF Instruction

Cache Locking Mode (LE Bit of CRR2)	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Ways to be Replaced
Disable	x	x	x	x	Ways 0, 1, 2, and 3 (table 4)
Enable	x	0	x	0	Ways 0, 1, 2, and 3 (table 4)
Enable	x	0	0	1	Ways 0, 1, and 3 (table 5)
Enable	0	1	x	0	Ways 0, 1, and 2 (table 6)
Enable	0	1	0	1	Ways 0, 1 (table 7)
Enable	0	x	1	1	Way 2
Enable	1	1	0	x	Way 3

Note: x means do not care. W3LOAD and W2LOAD should not be both set to 1.

Table 3 Ways to be Replaced When Cache Miss Occurs by the PREF Instruction

Cache Locking Mode (LE Bit of CRR2)	W3LOAD	W3LOCK	W2LOAD	W2LOCK	Ways to be Replaced
Disable	x	x	x	x	Ways 0, 1, 2, 3 (table 4)
Enable	x	0	x	0	Ways 0, 1, 2, 3 (table 4)
Enable	x	0	x	1	Ways 0, 1, 3 (table 5)
Enable	x	1	x	0	Ways 0, 1, 2 (table 6)
Enable	x	1	x	1	Ways 0, 1 (table 7)

Note: x means do not care. W3LOAD and W2LOAD should not be both set to 1.

Table 4 Ways to be Replaced for the LRU Bit (when th cache lock function is not used)

LRU (Bits 5 to 0)	Way to be Replaced
000000,000100,010100,100000,110000,110100	3
000001,000011,001011,100001,101001,101011	2
000110,000111,001111,010110,011110,011111	1
111000,111001,111011,111100,111110,111111	0

Table 5 Ways to be Replaced for the LRU Bit (when W2LOCK is set to be 1 and W3LOCK to be 0)

LRU (Bits 5 to 0)	Way to be Replaced
000000,000001,000100,010100,100000,100001,110000,110100	3
000011,000110,000111,001011,001111,010110,011110,011111	1
101001,101011,111000,111001,111011,111100,111110,111111	0

Table 6 Ways to be Replaced for the LRU Bit (when W2LOCK is set to be 0 and W3LOCK to be 1)

LRU (Bits 5 to 0)	Way to be Replaced
000000,000001,000011,001011,100000,100001,101001,101011	2
000100,000110,000111,001111,010100,010110,011110,011111	1
110000,110100,111000,111001,111011,111100,111110,111111	0

Table 7 Ways to be Replaced for the LRU Bit (when W2LOCK is set to be 1 and W3LOCK to be 1)

LRU (Bits 5 to 0)	Way to be Replaced
000000,000001,000011,000100,000110,000111,001011,001111, 010100,010110,011110,011111	1
100000,100001,101001,101011,110000,110100,111000,111001, 111011,111100,111110,111111	0

2.2 Setting Procedure of Functions Used

The procedure for setting up the lock function in the operand cache is described below

The cache lock function is used by setting the cache control register 2 (CCR2).

An example of procedure for setting the data read to way 3 in the operand cache and for setting the locking function is shown as follows:

1. Interrupt is disabled.
2. Way 3 is written back and disabled.
3. Way load bit and way lock bit, which correspond lock-enabled bit of CCR2 and way to fetch data, are set to 1.
4. CCR2 register is dummy-read.
5. Prefetch instruction (PREF@Rn) is executed.
   When a cache miss occurs, the line of data pointed to by Rn is read into the way corresponding to the setting 2. On the other hand, when a cache hit occurs, new data is not fetched and the entry, which is already enabled, is held.
6. After reading all data in, the way load bit is cleared as 0, and replacement of cache data is disabled.
7. CCR2 register is dummy-read.
8. Interrupt is enabled.

After reading the CCR2 register, cache-enabled space needs to be accessed not to access cache enabled space during the cache mode update after the setting of the CCR2 register. Also, the program that operates the cache control register 2 needs to be allocated in the cache-disabled space.

In this sample task, the interrupt mask is changed not to accept interrupt processing that accesses cache enabled space during the cache mode update.

Figure 1 shows an example of flow chart for setting the cache locking function and fetching the data.

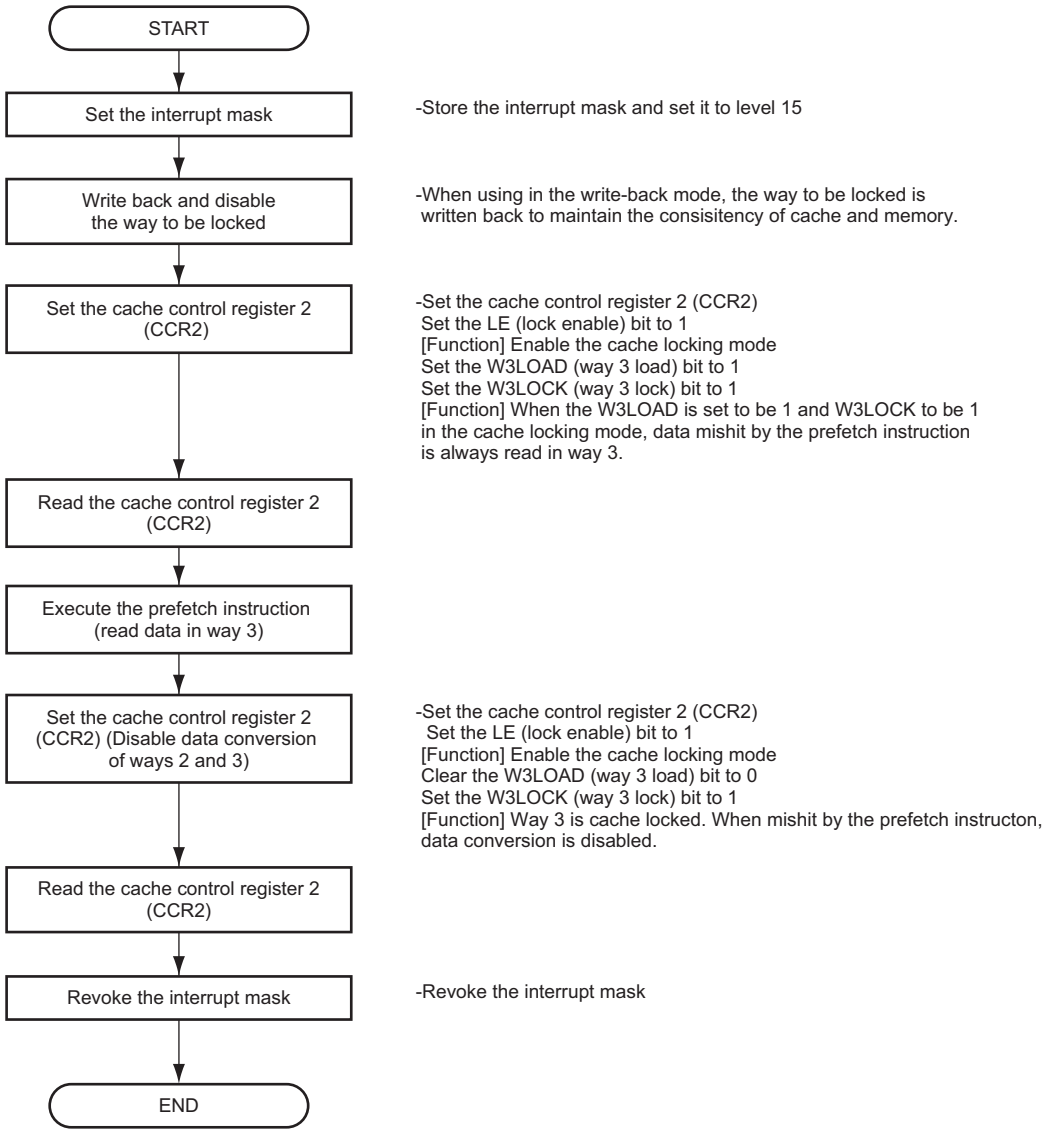


Figure 1 Example Flow for Processing the Sample Program

2.3 Operation of the Sample Program

In the sample program, the cache locking mode is set, and the bit inverse table is read in way 3. (In the example, the table data is multiples of 16 bytes.) After the replacement of way 3 is disabled, dummy access is operated to each entry. After that operation, reversion is output in the variable area using the table. For the cache operation function, the file names are changed to allocate the function in the cache-disabled space.

2.4 Processing Procedure of Sample Program

Table 8 describes how to set the cache memory in the sample program, and figure 2 is a flowchart processing by the sample program.

Table 8 Cache Setting

Name of Register	Address	Setting Value	Function
Cache control register 2 (CCR2)	H'FFFC 1004	H'0001 0301	-LE = 1 : Cache locking mode -W3LOAD = 1 : Replacing way 3 by the PREF instruction at miss hit. -W3LOCK = 1 : Locking way 3
		H'0001 0103	W3LOAD = 0: Replacing way 3 is disabled.

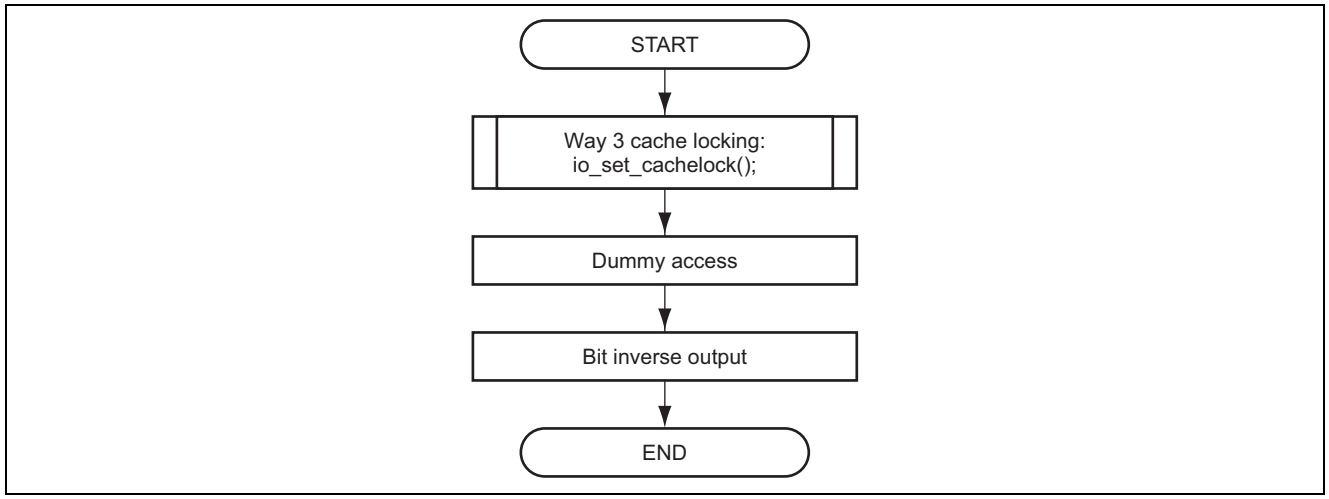


Figure 2 Example Flow for Processing the Sample program

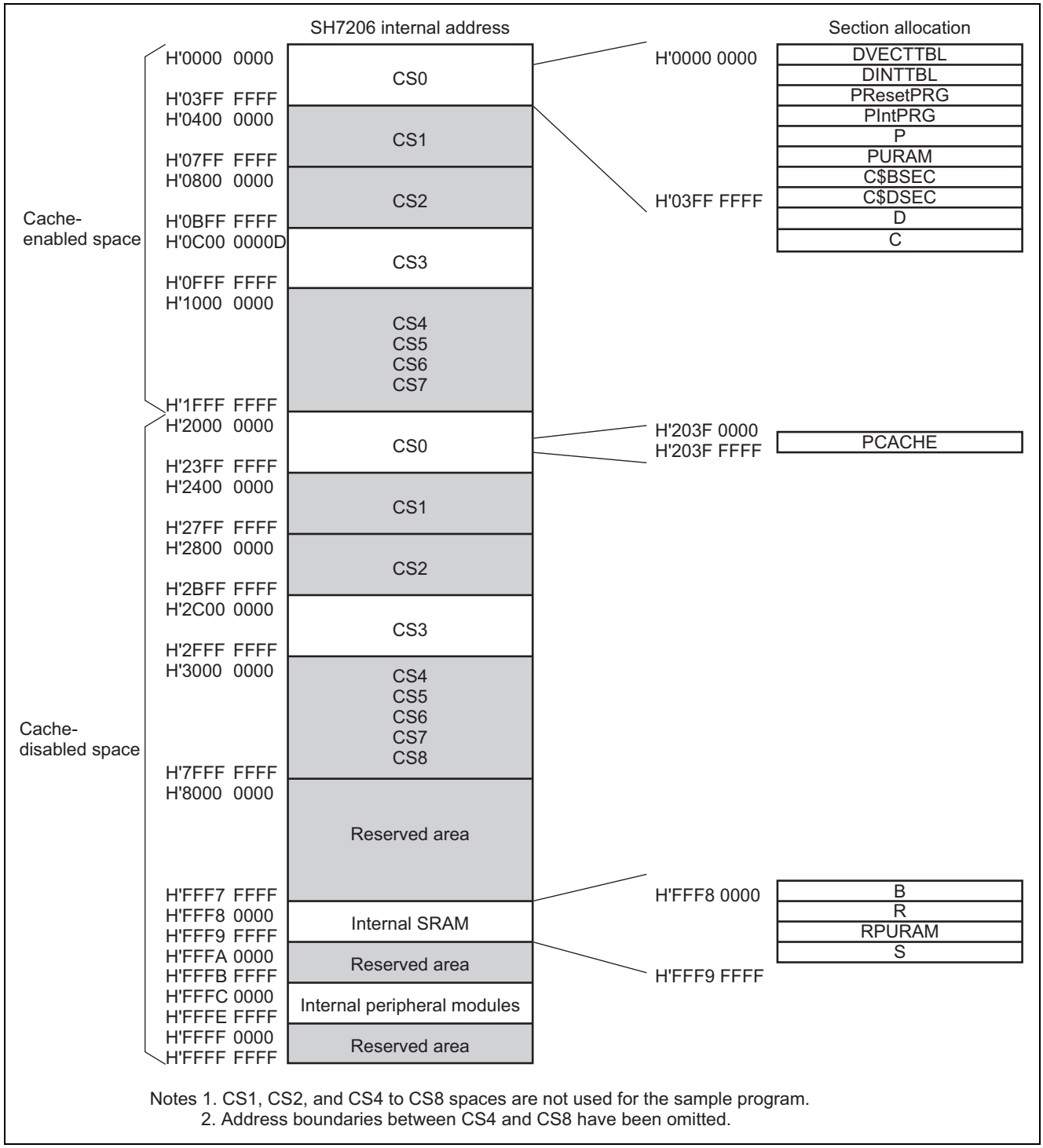
2.5 Allocation of Sections in the Allocation in the Sample Program

The #pragma section directive is used with the corresponding extended compiler function to set a section name for the function that actually manipulates the cache control registers.

In the sample program, the area for program code of the io_set_cache function is set to the PCACHE section. Only this part of the program is allocated to a cache-disabled space of the SH7206. That is, the rest of the program is allocated to a space where caching is performed if it is enabled (the P section).

Section allocation is specified by linkage editor options.

Figure 3 is a memory map for the sample program.



Notes 1. CS1, CS2, and CS4 to CS8 spaces are not used for the sample program.
2. Address boundaries between CS4 and CS8 have been omitted.

Figure 3 Memory Map for the Sample Program

3. Sample Program

- Sample Program: Listing of “main.c” (1)

```
/*""FILE COMMENT""*****************************************************************
*
*       System Name : SH7206 Sample Program
*       File Name   : cachelock.c
*       Version     : 1.00.00
*       Contents    : sample of cache lock
*       Model       : M3A-HS60
*       CPU         : SH7206
*       Compiler    : SHC9.0.01
*       OS          : None
*
*       Note        : Sample program to confirm the cache operation
*
*                     <Caution>
*                     This entire sample program is for reference only and
*                     its operation is not guaranteed.
*                     Please use this sample as a technical reference
*                     in software development.

*       Copyright (C) 2004 Renesas Technology Corp. All Rights Reserved
*       AND Renesas Solutions Corp. All Rights Reserved
  *
  *       History   : 2004.10.28 ver.1.00.00
  *""FILE COMMENT END""*************************************************************/
  #include <machine.h>
#include "iodefine.h"    /* /* iodefine.h is automatically generated by the High-
                            performance Embedded Workshop.                    */

/* ---- Dummy Access Area Address ---- */
#define DAMY_ADD  (unsigned int *) (0x0C000000) /* Cache-enabled area */

```

- Sample Program: Listing of “main.c” (2)

```
/* ---- Bit inverse table ---- */
#pragma section ROM      /* The following sections are CROM sections  */
unsigned char table[]={       255,254,253,252,251,250,
                    249,248,247,246,245,244,243,242,241,240,
                    239,238,237,236,235,234,233,232,231,230,
                    229,228,227,226,225,224,223,222,221,220,
                    219,218,217,216,215,214,213,212,211,210,
                    209,208,207,206,205,204,203,202,201,200,
                    199,198,197,196,195,194,193,192,191,190,
                    189,188,187,186,185,184,183,182,181,180,
                    179,178,177,176,175,174,173,172,171,170,
                    169,168,167,166,165,164,163,162,161,160,
                    159,158,157,156,155,154,153,152,151,150,
                    149,148,147,146,145,144,143,142,141,140,
                    139,138,137,136,135,134,133,132,131,130,
                    129,128,127,126,125,124,123,122,121,120,
                    119,118,117,116,115,114,113,112,111,110,
                    109,108,107,106,105,104,103,102,101,100,
                    99,98,97,96,95,94,93,92,91,90,
                    89,88,87,86,85,84,83,82,81,80,
                    79,78,77,76,75,74,73,72,71,70,
                    69,68,67,66,65,64,63,62,61,60,
                    59,58,57,56,55,54,53,52,51,50,
                    49,48,47,46,45,44,43,42,41,40,
                    39,38,37,36,35,34,33,32,31,30,
                    29,28,27,26,25,24,23,22,21,20,
                    19,18,17,16,15,14,13,12,11,10,
                    9,8,7,6,5,4,3,2,1,0};

/* ---- Destination to store data after the bit inverse ---- */
#pragma section CACHE    /* Allocate in the CS0 cache-disabled space */
unsigned char buff[256];
```

- Sample Program: Listing of “main.c” (3)

```
#pragma section
/* ==== Prototype declaration ==== */
void main(void);
int io_set_cachelock(unsigned char *ptr,unsigned int size);

/*""FUNC COMMENT""*************************************************************
* ID              :
* Overvie of module: Sample program main (example of using cache lock)
*------------------------------------------------------------------------------
* Include         :
*------------------------------------------------------------------------------
* Declaration     : void main(void)
*------------------------------------------------------------------------------
* Functions       : Sample of cache lock function
*                  : Cache locking mode is set, and the bit inverse table is
*                  : read to way 3 in the operand cache. After dummy access,
*                  : the inverse output is executed in the variable area
*                  : using the inverse table.
*------------------------------------------------------------------------------
* Argument        : None
*------------------------------------------------------------------------------
* Return value    : None
*------------------------------------------------------------------------------
* Caution         :
*""FUNC COMMENT END""*********************************************************/

void main(void)
{
     unsigned int i,*ptr;


     /* ==== Locking way 3 cache ==== */
     io_set_cachelock(table,sizeof(table));

     /* ==== Dummy access ==== */
     ptr=DAMY_ADD;
     for(i=0; i<16384; i++){
          *ptr++=0x5555;
     }

     /* ==== Bit inverse output ==== */
     for(i=0; i<0x100; i++){
          buff[i]=table[i];
     }

     while(1){
        /* Program end */
     }

}
```

- Sample Program: Listing of “main.c” (4)

```
112  #pragma section CACHE    /* Allocate in the CS0 cache-disabled space             */
113  /*""FUNC COMMENT""*****************************************************************
114  * ID                 :
115  * Overview of module : Cache lock setting
116  *---------------------------------------------------------------------------------
117  * Include            : #include "iodefine.h"
118  *---------------------------------------------------------------------------------
119  * Declaration        : int io_cacherok (unsigned int mode)
120  *---------------------------------------------------------------------------------
121  * Function           : The operand cache is set in the cache locking mode
122  *                    : and locked after reading the data.
123  *                    : It is the multiples of 16 bytes.
124  *---------------------------------------------------------------------------------
125  * Argument: unsigned int mode: The following modes are set with the logical OR
126  *  :      : CACHE_LOCK_OFF   : Cache locking mode is off
127  *  :      : CACHE_LOCK_ON    : Cache locking mode is on
128  *  :      : CACHE_3_LOCK_ON  : Lock way 3
129  *  :      : CACHE_3_LOAD_ON  : Enable reading with way 3 Prefetch instruction
130  *---------------------------------------------------------------------------------
131  * Return value       :
132  *---------------------------------------------------------------------------------
133  * Caution            :
134  *""FUNC COMMENT END""*************************************************************/
135
136  int io_set_cachelock(unsigned char *ptr,unsigned int size)
137  {
138       volatile unsigned long *arry;
139       unsigned int i;
140       int mask,reg;
141
142       /* ==== Setting interrupt mask ==== */
143       mask = get_imask();
144       set_imask(15);                    /* Set to level 15 */
145
146       /* ==== Writing back entry 3 ==== */
147       for(i=0u; i < 128u; i++){
148          /* ---- Creating address array address ---- */
149          arry = (volatile unsigned long *)(0xf0801800 | (i<<4));
150          /* ---- Writing U = 0 and V = 0 in the address array ---- */
151          *arry &= 0xfffffffcul;         /* V=0, U=0 */
152       }
153
```

- Sample Program: Listing of “main.c” (5)

```
154       /* ==== Setting the cache register 2 ==== */
155       CCNT.CCR2.LONG = 0x00010300;
156
157       /* ==== Reading the cache register ==== */
158       reg = CCNT.CCR2.LONG;
159
160       /* ==== Reading data in way 3 ==== */
161       for(i=size/16; i>0; i--){
162          prefetch (ptr);
163          ptr +=16;
164       }
165
166       /* ==== Setting the cache register 2 ==== */
167       CCNT.CCR2.LONG = 0x00010100;       /* Lock way 3 */
168
169       /* ==== Reading the cache register ==== */
170       reg = CCNT.CCR2.LONG ;
171
172       /* ==== Revoking the interrupt mask ==== */
173       set_imask(mask);                   /* Set to the original level */
174
175  }
176  /* End of file */
```

4. Documents for Reference

- Software manual
  SH-2A SH2A-FPU Software Manual Rev.3.00
  If you don't already have it, please download the latest version from the homepage of Renesas Technology Corp.

- Hardware manual
  SH7206 Group Hardware Manual Rev.1.00
  If you don't already have it, please download the latest version from the homepage of Renesas Technology Corp.

5. Website and Support Window

Website of Renesas Technology, Corp.

http://www.renesas.com/

Revision Record

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

Keep safety first in your circuit designs!

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

Notes regarding these materials

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