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

Cache: Example of Write-Back Operation in the Operand Cache

Introduction

This application note describes a write-back method in the operand cache for the SH7206.

Target Device

SH7206

Contents

1. Overview

1.1 Specifications

- The instruction cache and operand cache are enabled and placed in the write-back mode.
- Address array in the operand cache is operated, and the operand cache is written back.

1.2 Function Used

Instruction cache and 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. Version 9.00 C/C++ compiler package for the SuperH RISC engine Family
• Compile option:	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 Examples

Address array in the operand cache is used in this sample task.

2.1 Operation Overview of Function Used

When the CPU and bus master except for CPU, such as the internal DMA controller, uses the same external memory allocated in the cache enabled space, cache is disabled or written back by software, and coherency of cache and external memory needs to be maintained. The cache for the SH7206 can be read from and written to the content by the MOV instruction. To operate the write-back mode in the operand cache, address arrays in the operand cache are operated. Address arrays in the operand cache are allocated to H'F080 0000 to H'F0FF FFFF and data arrays to H'F180 0000 to H'F1FF FFFF.

When the U bit (read) and the V bit (entry data enabled) in the operand cache address line are set to 1 and written to the cache line, that the cache line is written back. In this case, writing method varies as follows depending on the state of the associative bit (A bit):

- No associative is available.

  When there is no associative (A bit = 0), the entry corresponding to the specified entry address and way are written.
- Associative is available.

  When there is associative (A bit = 1), the entry of way, that the tag address in cache corresponding to the specified entry address and the specified tag address became coherent, is written. Coherency judgment is operated to all 4 ways. The U and V bits are written to the coherent entry. However, the tag address and the LRU bit are not reflected on the entry. When the operand cache is not hit to any way, writing will not be operated.

Table 1 describes the overview of cache. Figure 1 shows the scheme of cache-search method, and figure 2 shows the cache access specifying to allocate the operand cache memory.

Table 1 Overview of Cache

Item	Overview
Capacity	Instruction cache: 8 Kbytes Operand cache: 8 Kbytes
Structure	Instruction and data are separated, each cache is 4-way set associative
Cache lock function	Ways 2 and 3 can be locked (only in the operand cache)
Line size	16 bytes
Number of entries/ways	128
Write system	Write-back and write-through methods are selectable.
Replacement method	Least-recently-used (LRU) algorithm

Note: Please refer to the section 'Cache' in the *SH7206 Group Hardware Manual* for details on the caches.

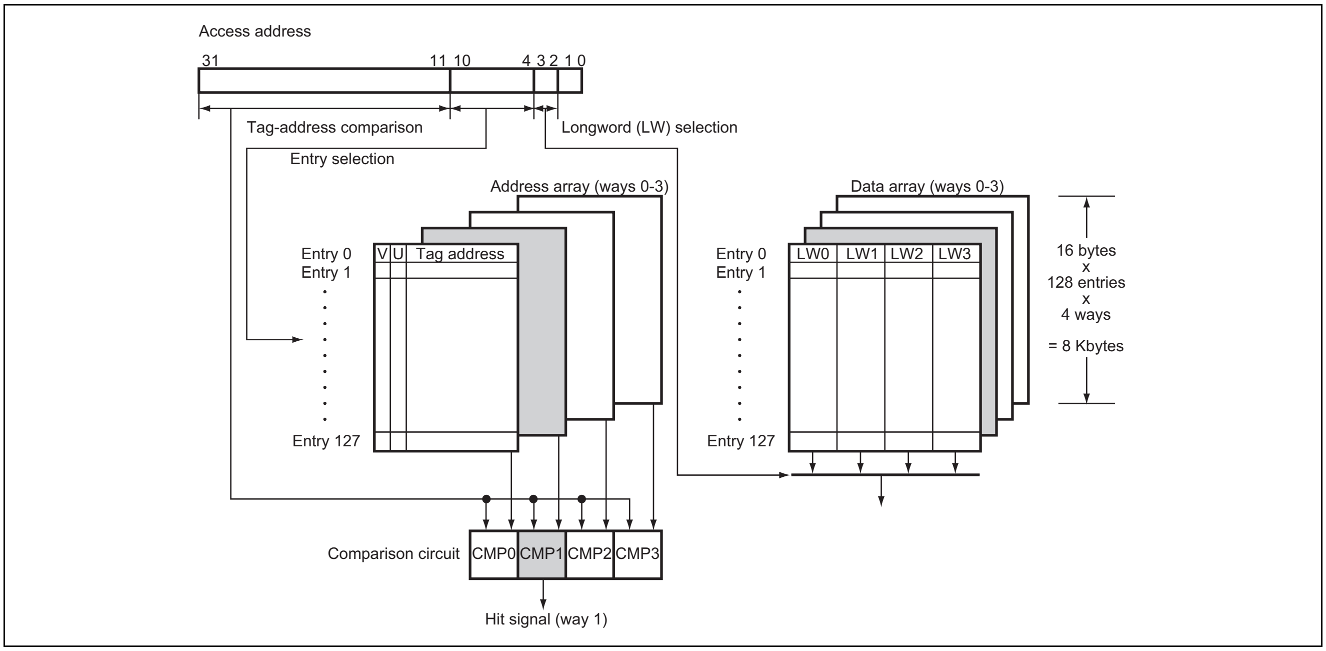


Figure 1 Overview of the Cache-Search Scheme

Reading address array:

Address

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12 11	10 … 4	3	2	1	0
1	1	1	1	0	0	0	0	1	-	-	-	-	-	-	-	-	-	-	W	Entry address	0	-	0	0

Data

31	30	29	28 … 11	10	… 4	3	2	1	0
0	0	0	Tag address	E	LRU	0	0	U	V

Writing address array:

Address

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12 11	10 … 4	3	2	1	0
1	1	1	1	0	0	0	0	1	-	-	-	-	-	-	-	-	-	-	W	Entry address	A	-	0	0

Data

31	30	29	28 … 11	10	… 4	3	2	1	0
0	0	0	Tag address	-	LRU	0	0	U	V

Data array access (shared at reading and writing):

Address

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12 11	10 … 4	3 2	1	0
1	1	1	1	0	0	0	0	1	1	-	-	-	-	-	-	-	-	-	W	Entry address	L	0	0

Data

31 … 0
Longword data

A : Associative bit (1: associative, 0: not associative)
W : Way (B'00: way 0, B'01: way 1, B'10: way 2, B'11: way 3)
LRU : Least-Recently-Used (LRU) bit
U : U bit (1: written, 0: not written)
V : V bit (1: entry enable, 0: entry disable)
L : Longword position (B'00: longword 0, B'01: longword 1, B'10: longword 2, B'11: longword 3)

Figure 2 Cache Access Specifying Method for Memory Allocation (Operand Cache)

2.2 Setting Procedure of Functions Used

Procedure for writing back in the operand cache is described below.

In this sample program, all the entries in the operand cache will be disabled (the V bit will be cleared as 0). The cache memory, then, will be written back to the entry where the U bit in the specified address array is set to 1.

Figure 3 shows an example of flow chart to process when disabling the operand cache.

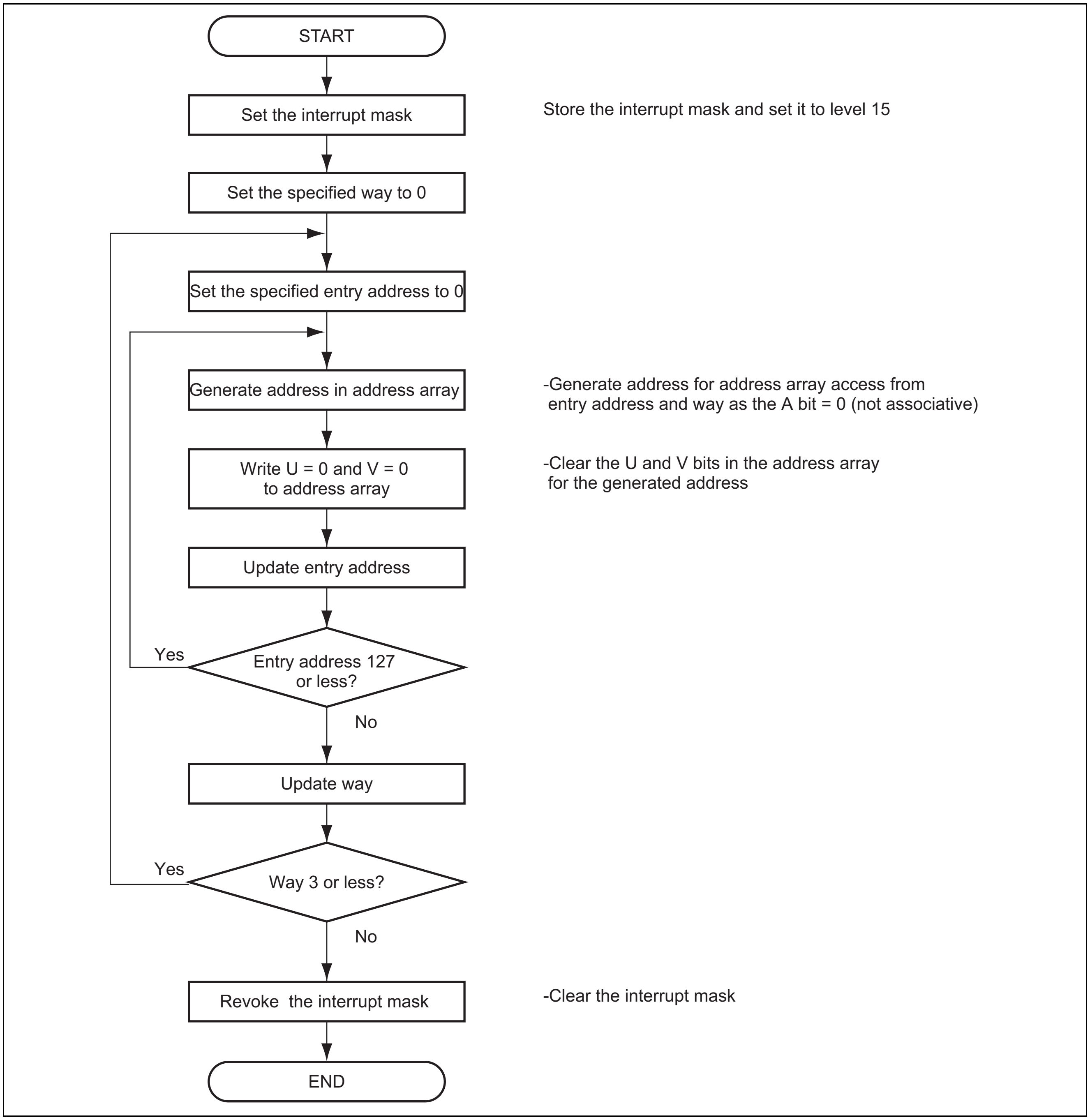

START
Set the interrupt mask
Store the interrupt mask and set it to level 15
Set the specified way to 0
Set the specified entry address to 0
Generate address in address array
-Generate address for address array access from
entry address and way as the A bit = 0 (not associative)
Write U = 0 and V = 0
to address array
-Clear the U and V bits in the address array
for the generated address
Update entry address
Yes
Entry address 127
or less?
No
Update way
Yes
Way 3 or less?
No
Revoke the interrupt mask
-Clear the interrupt mask
END

Figure 3 Example of Processing Flowchart for the Operand Cache Write-Back

2.3 Operation of the Sample Program

In the sample program, the operand cache is enabled with operation in the write-back mode. At this time, a single line of cache memory is filled. Since the operand cache is enabled (the write-back mode) for the target region of memory, the data is actually written to the cache memory. That is, the data is not reflected in the external memory (SDRAM). After that, the operand cache is written back and confirmed that the data is reflected in the external memory (SDRAM).

For the cache operation and operand cache write-back functions, the file names are changed to allocate the functions in the cache disabled space.

2.4 Procedure for Processing by the Sample Program

Table 2 describes how the cache is set up by the sample program, and table 3 describes macro definitions used in the sample program. Figure 4 is a flowchart of processing by the sample program.

Table 2 Cache Setting

Name of Register	Address	Setting Value	Function
Cache control register 1 (CCR1)	H’FFFC 1000	H’0000 0909	-ICF = 1 : Instruction cache flush -ICE = 1 : Instruction cache enable -OCF = 1: Operand cache flush -WT = 0 : Write-back mode -OCE = 1: Operand cache enable Note: ICF and OCF are always read as 0.

Table 3 Cache-Related Macro Definitions in the Sample Program

Macro Definition	Setting Value	Function
CACHE_OFF	H’0000	Turns the cache off
CACHE_I_FLUSH	H’0800	Instruction cache flush
CACHE_I_ON	H’0100	Instruction cache enable
CACHE_O_FLUSH	H’0001	Operand cache flush
CACHE_O_ON	H’0008	Operand cache enable
CACHE_O_WT	H’0002	Operand cache write-through mode

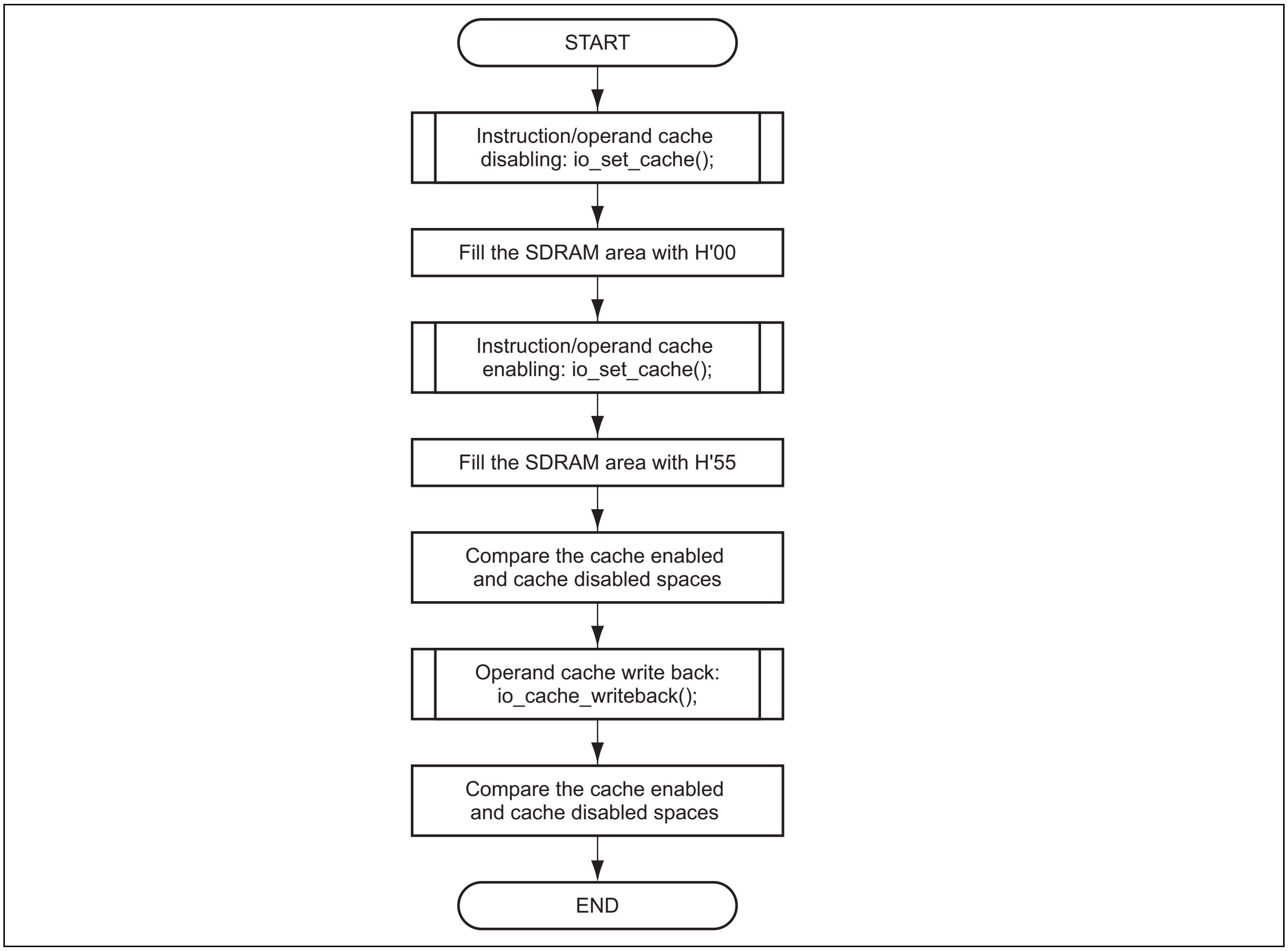

START
Instruction/operand cache disabling: io_set_cache();
Fill the SDRAM area with H'00
Instruction/operand cache enabling: io_set_cache();
Fill the SDRAM area with H'55
Compare the cache enabled and cache disabled spaces
Operand cache write back: io_cache_writeback();
Compare the cache enabled and cache disabled spaces
END

Figure 4 Flow of Processing by the Sample Program

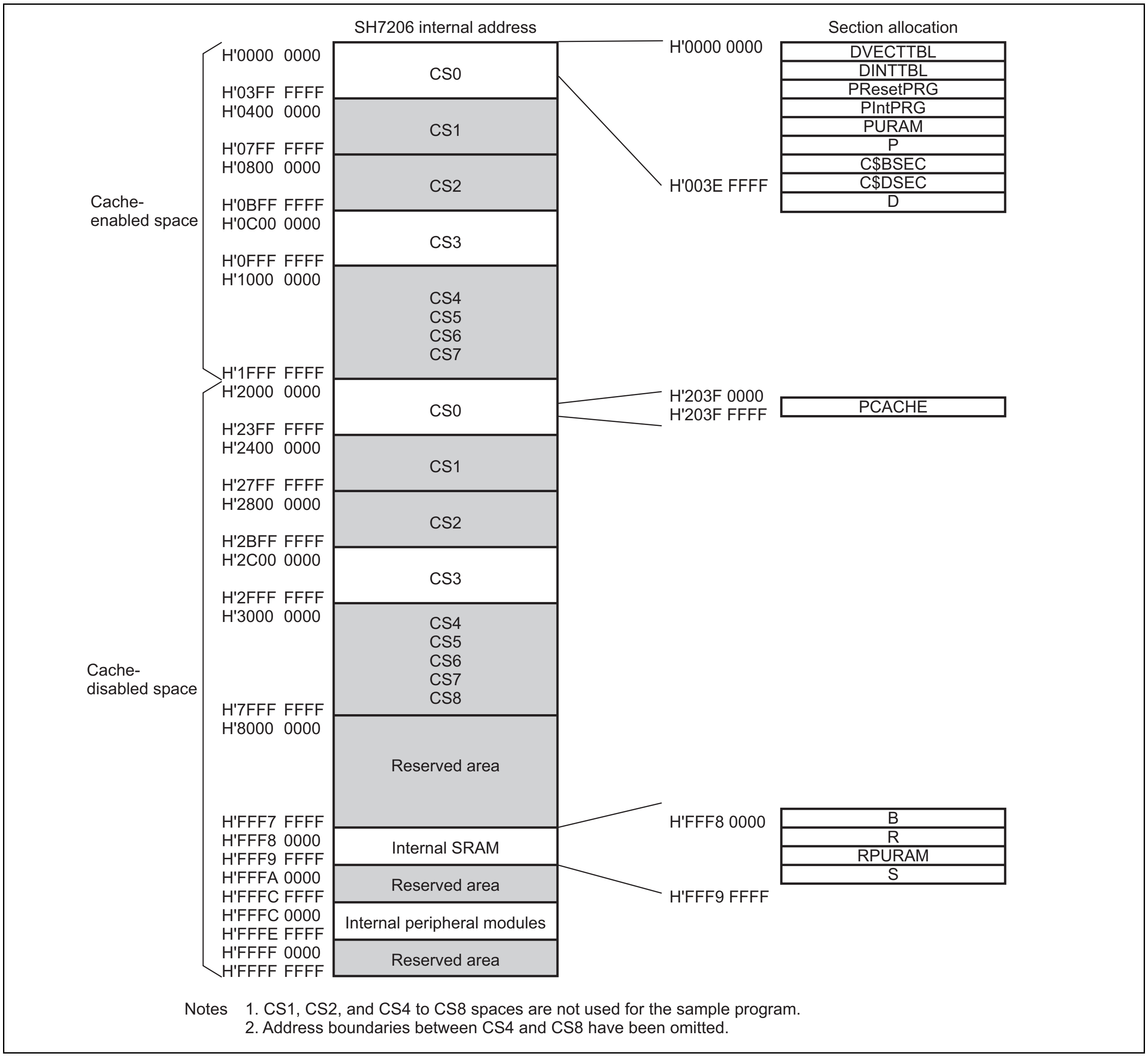
2.5 Allocation of Sections in the Sample Program

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 and io_cache_writeback functions are 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 (P section).

Section allocation is specified by linkage editor options.

Figure 5 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 5 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   : main.c
 *       Version     : 1.00.00
 *       Contents    : Example of writing back the cache memory
 *       Model       : M3A-HS60
 *       CPU         : SH7206
 *       Compiler    : SHC9.0.00
 *       OS          : None
 *
 *       Note        : Sample program to write back the operand cache
 *
 *                    <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   : 2004.10.14 ver.1.00.00
 *""FILE COMMENT END""**********************************************************/
#include <machine.h>
#include "iodefine.h"

/* ==== Macro definitions ==== */
/* ---- Cache settings ---- */
#define CACHE_OFF     0x0000u
#define CACHE_I_FLUSH    0x0800u
#define CACHE_I_ON    0x0100u
#define CACHE_O_FLUSH    0x0008u
#define CACHE_O_ON    0x0001u
#define CACHE_IO_ON      (CACHE_I_ON | CACHE_O_ON)
#define CACHE_O_WT    0x0002u

/* ---- SDRAM area addresses ---- */
#define SDRAM_ADDR1   (unsigned char *)(0x0c000000)   /* Cache-enabled space */
#define SDRAM_ADDR2   (unsigned char *)(0x2c000000)   /* cache-disabled space */


/* ==== Prototype declaration ==== */
int io_set_cache(unsigned int mode);
int io_cache_writeback(void);
void main(void);

```

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

```
/*""FUNC COMMENT""************************************************************
* ID                 :
* Overview of module : Sample program main (example of using cache memory)
*-----------------------------------------------------------------------------
* Include            : #include "iodefine.h"
*-----------------------------------------------------------------------------
* Declaration        : void main(void)
*-----------------------------------------------------------------------------
* Functions      : Sample of enabling/disabling cache memory.
*                : After the SDRAM area has been initialized with the operand OFF,
*                : a fill operation is performed with the operand cache ON and the
*                : cached area is compared with its shadow in the cache-disabled
*                : space. The operand cache is written back next and confirmed
*                : that the data is reflected in the external memory.
*-----------------------------------------------------------------------------
* Argument       : None
*-----------------------------------------------------------------------------
* Return value   : None
*-----------------------------------------------------------------------------
* Caution        : In this sample program, the cache is flushed. Therefore,
*                : when the cache is enabled by a program for initialization,
*                : content of cache will be invalidated.
*""FUNC COMMENT END""********************************************************/
void main(void)
{
     int i;
     unsigned char *ptr1,*ptr2;

     /* ==== Disabling instruction and operand caches ==== */
     io_set_cache(CACHE_OFF | CACHE_I_FLUSH | CACHE_O_FLUSH);

     /* ==== Filling SDRAM area with 0x00 ==== */
     ptr1 = SDRAM_ADDR1;   /* Cache-enabled space */
     for(i=0; i < 8192; i++){
        *ptr1++ = 0;
     }

     /* ==== Enabling instruction/operand cache ==== */
     io_set_cache(CACHE_I_ON | CACHE_O_ON );

     /* ==== Filling SDRAM area with 0x55 ==== */
     ptr1 = SDRAM_ADDR1;   /* Cache-enabled space */
     for(i=0; i < 8192; i++){
        *ptr1++ = 0x55;
     }

```

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

```
    /* ==== Comparing cache-enabled and cache-disabled spaces ==== */
    ptr1 = SDRAM_ADDR1;   /* Cache-enabled space */
    ptr2 = SDRAM_ADDR2;   /* Cache-disabled space */

    for(i=0; i < 8192; i++){
        if(*ptr1++ == *ptr2++){
            while(1){
                /* Error in operand-cache setting */
            }
        }
    }

    /* ==== Writing back the operand cache ==== */
    io_cache_writeback();

    /* ==== Comparing cache-enabled and cache-disabled spaces ==== */
    ptr2 = SDRAM_ADDR2;   /* Cache-enabled space */

    for(i=0; i < 8192; i++){
        if(*ptr2++ != 0x55){
            while(1){
                /* Error in operand-cache write back */
            }
        }
    }

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

}
```

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

```
#pragma section CACHE    /* Allocated in the CS0 cache-disabled space */
/*""FUNC COMMENT""************************************************************
* ID                    :
* Overview of modules : Cache setting
*-----------------------------------------------------------------------------
* Include               : #include "iodefine.h"
*-----------------------------------------------------------------------------
* Declaration           : int io_set_cache(unsigned int mode)
*-----------------------------------------------------------------------------
* Function              : Cache memory is set in the mode specified by mode.
*                       :
*-----------------------------------------------------------------------------
* Argument : unsigned int mode : The following modes are set with the logical OR.
*          :           : CACHE_I_FLUSH  : Instruction cache flush
*          :           : CACHE_I_ON     : Instruction cache enable
*          :           : CACHE_O_FLUSH  : Operand cache flush
*          :           : CACHE_O_ON     : Operand cache enable
*          :           : CACHE_IO_ON    : Instruction/operand cache on
*          :           : CACHE_O_WT     : Write-through mode
*          :           : CHAHE_OFF      : Instruction/operand cache disable
*-----------------------------------------------------------------------------
* Return value          : 0 : Normal end
*-----------------------------------------------------------------------------
* Caution               :
*""FUNC COMMENT END""********************************************************/
int io_set_cache(unsigned int mode)
{
     volatile unsigned long reg;
     int mask;

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

     /* ==== Setting cache register ==== */
     CCNT.CCR1.LONG = mode;

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

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

     return 0;
}

```

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

```
/*""FUNC COMMENT""*************************************************************
* ID                   :
* Overview of module : Operand cache write back
*------------------------------------------------------------------------------
* Include              : #include "iodefine.h"
*------------------------------------------------------------------------------
* Declaration          : int io_cache_writeback(void)
*------------------------------------------------------------------------------
* Function       : All lines in the operand cache is disabled, and contents in
*                : the cache memory are written back in the external memory.
*------------------------------------------------------------------------------
* Argument             : None
*------------------------------------------------------------------------------
* Return value         : 0 : Normal end
*------------------------------------------------------------------------------
* Caution        : Write-back operation is not executed in the write-through mode.
*""FUNC COMMENT END""**********************************************************/
int io_cache_wrtieback(void)
{
     volatile unsigned long *arry;
     unsigned int i,j;
     int mask;

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

     /* ==== Disabling all entries ==== */
     for(i=0u; i <4u; i++){
        for(j=0u; j < 128u; j++){
            /* ---- Creating address array address ---- */
            arry = (volatile unsigned long *)(0xf0800000 | (i<<11) | (j<<4));
            /* ---- Writing U = 0 and V = 0 in the address array ---- */
            *arry &= 0xfffffffcul;             /* V = 0, U = 0 */
        }
     }

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

     return 0;
}

/* 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.