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.

SH7730 Group

Example of Writing Back from the Operand Cache

Introduction

This application note describes the operation of writing back from the operand cache of the SH7730. Please refer to this document when the operand cache is enabled in copy-back mode and data in external memory must match data in the operand cache (in cases like DMA transfer for areas where caching is enabled.).

Target Device

SH7730

Contents

1. Preface 2
2. Description of the Sample Application 3
3. Listing of Sample Program 9
4. Documents for Reference 15

1. Preface

1.1 Specifications

The sample program handles the operation of writing back from the operand cache by using the intrinsic function ocbp().

1.2 Modules Used

- Instruction and operand caches

1.3 Applicable Conditions

• Evaluation board	The AP-SH4A-1A board incorporates the SH7730 with SH-4A CPU core is available from AlphaProject Co., Ltd.		
	External memory	(area 0)	4-MB NOR-type flash memory: S29AL032D70TFI04 from Spansion
		(area 3)	32-MB SDR-SDRAM (16 MB × 2): K4S281632F-UC75 from Samsung
• MCU	SH7730 (R8A77301)		
• Operating frequency	Internal clock: 266.66 MHz SuperHyway bus clock: 133.33 MHz Bus clock: 66.66 MHz Peripheral clock: 33.33 MHz		
• Bus width for area 0	16-bit fixed (with the MD3 pin at the low level)		
• Clock operating mode	Mode 2 (with the MD0 pin at the low level and MD1 pin at the high level)		
• Endian	Big endian (with the MD5 pin at the low level)		
• Toolchain	SuperH RISC engine Standard Toolchain Ver.9.1.1.0 from Renesas Technology		
• Compiler options	Default settings of High-performance Embedded Workshop (-cpu=sh4a -debug -optimize=0 –noinline -gbr=auto -macsave=0 -save_cont_reg=0 -chgincpath –errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo)		

1.4 Related Application Note

The operation of the reference program for this document was confirmed with the setting conditions described in the *SH7730 Group Application Note SH7730 Example of Initialization (REJ06B0848)*. Please refer to that document in combination with this one.

2. Description of the Sample Application

2.1 Overview of the SH7730 Cache Memory

Table 1 Overview of the SH7730 Cache Memory

Item	Instruction Cache	Operand Cache
Capacity	32 Kbytes	32 Kbytes
Type	4-way set-associative, virtual address index/physical address tag	4-way set-associative, virtual address index/physical address tag
Line size	32 bytes	32 bytes
Number of entries	256 entries/way	256 entries/way
Write mode	—	Copy-back/write-through selectable
Replacement method	Least-recently-used (LRU) algorithm	Least-recently-used (LRU) algorithm

2.2 Setting up the Caches

2.2.1 Processing for Cache Settings

For the procedure to follow in setting up the cache control register (CCR), see the SH7730 Group Application Note: *Examples of Cache Memory Settings* (REJ06B0851).

2.3 Operation by the Sample Program

2.3.1 Overview of Processing

Figure 1 shows the flow of processing by the sample program.

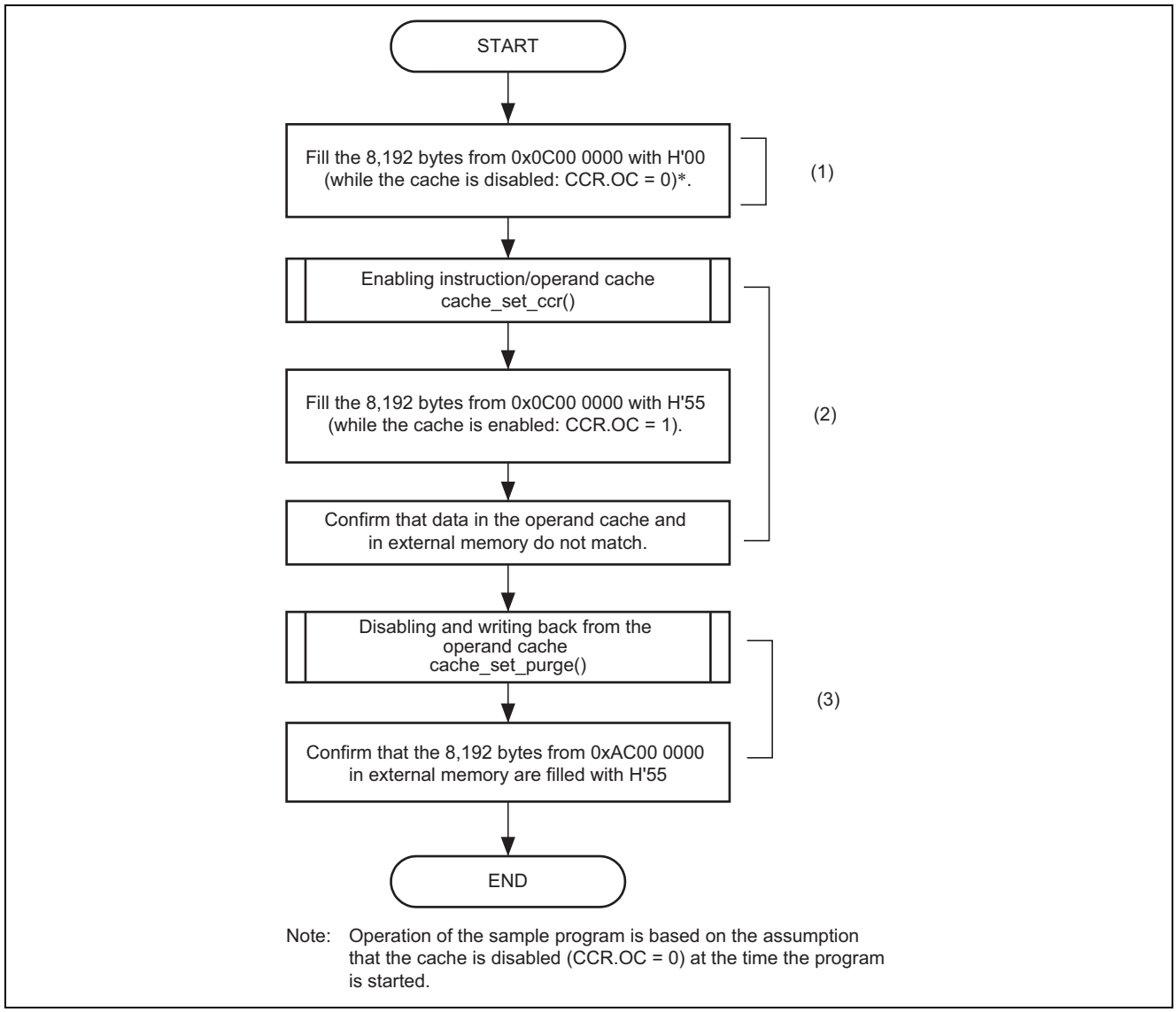


Figure 1 Flow of Processing by the Sample Program

(1) A value (H'00) is written to 8,192 bytes (one way: 32 bytes × 256 entries = 8,192 bytes) of external memory while the operand cache is disabled (CCR.OCE = 0).

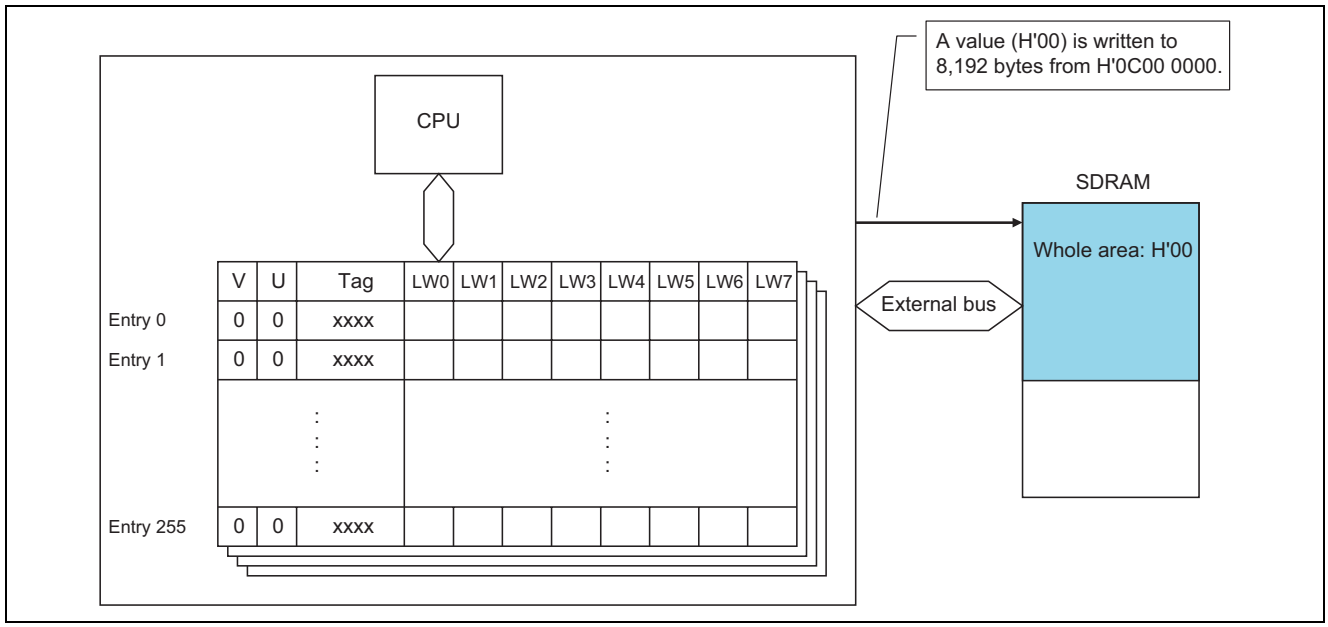


Figure 2 Program Description (1)

(2) The instruction cache or the operand cache is enabled in copy-back mode. A value (H'55) is written to 8,192 bytes of external memory in the cacheable P1 area. Since the cache is enabled in copy-back mode at this point, all data are written to the operand cache instead of external memory. As a result, if data from access to the cacheable P1 area (data in the operand cache) are compared with data from access to the non-cacheable P2 area (data in external memory), they will not match.

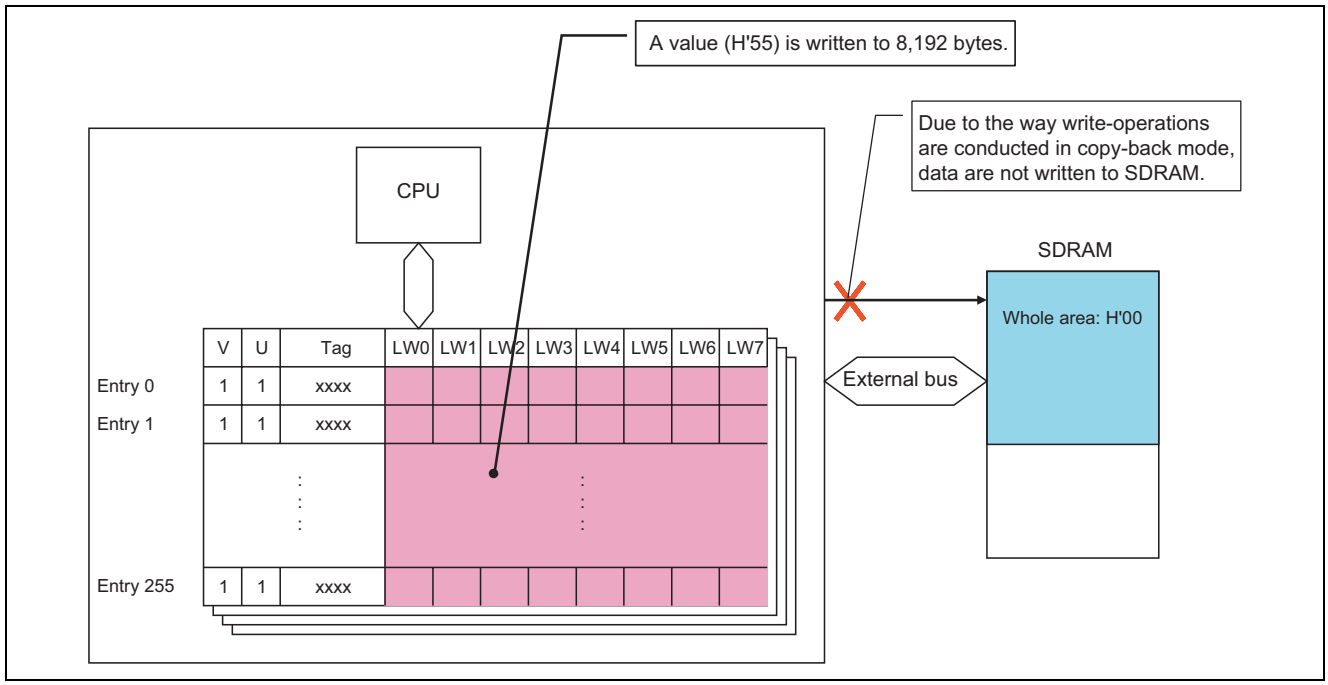


Figure 3 Program Description (2)

(3) Writing back from the operand cache is executed. After this is done, the data from access to the non-cacheable P2 area (data in external memory) are compared with the value H'55. A match indicates that writing back was successful.

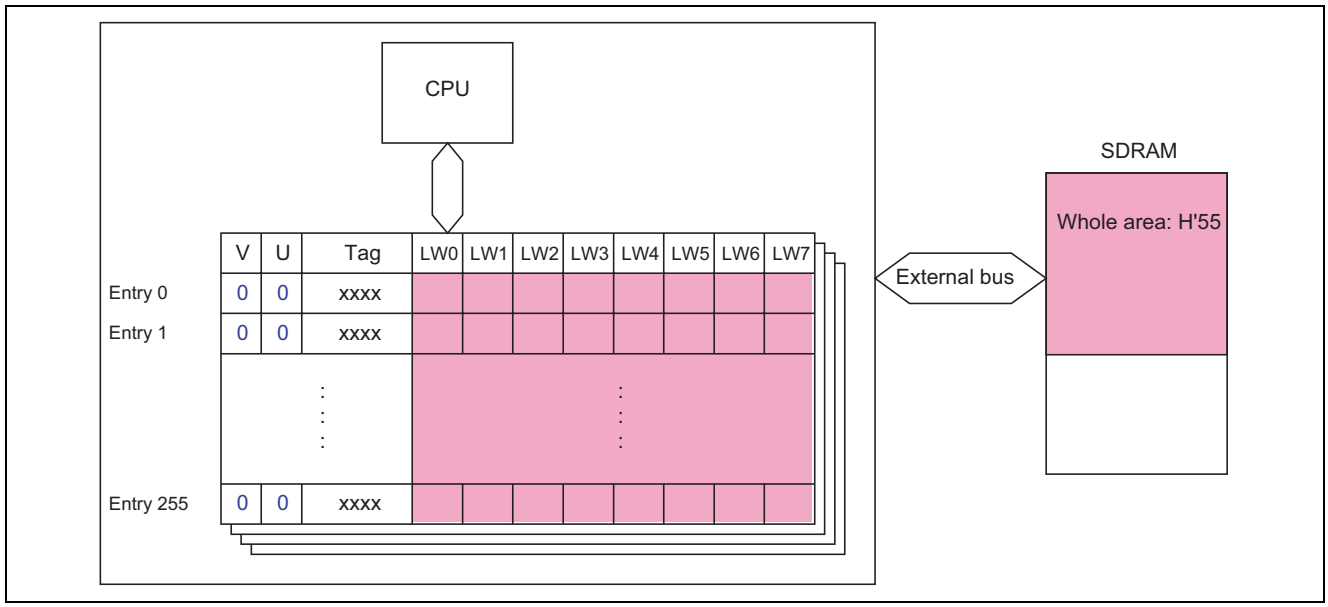


Figure 4 Program Description (3)

2.3.2 Processing to Write Back from the Operand Cache

The procedure for writing back from the operand cache is described below.

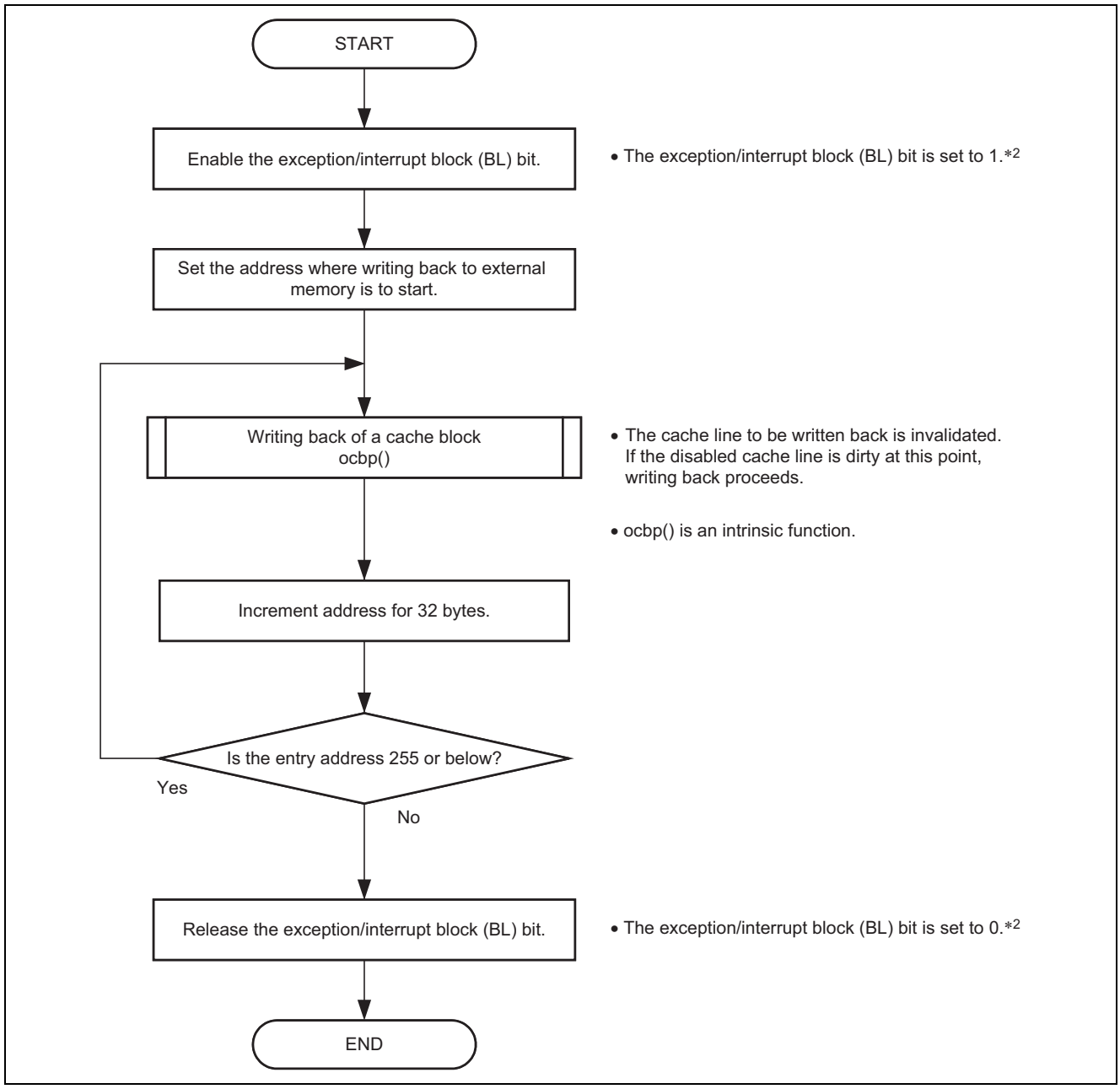


Figure 5 Procedure for Writing Back from the Operand Cache

Notes: 1. This sample program should be executed in privileged mode.

2. In this sample program, the exception/interrupt block (BL) bit is set to 1 during the operation of writing back from the operand cache. This prevents interrupt handling that might lead to access to external memory for which data would have to be written back. In general, the prevention of unanticipated access due to the handling of interrupts to memory areas for which writing back is in progress must be taken into account on the system side. In this sample program, however, interrupts are prohibited so that unanticipated access occurs.

2.4 Allocation of Sections in the Sample Program

The compiler extended function: #pragma section is used to change the section name of the function that handles the cache control register. In this sample program, the program area for the cache_set_ccr() function is changed to the PnonCache section. This ensures that only the PnonCache section will be placed in a cache-disabled space while other program sections are placed in cache-enabled spaces. Allocation of sections (address specification) is specified by options of the linker.

Table 2 gives information on sections in the sample program.

Table 2 Section Information of the Sample Program

Section Name	Application of Section	Allocation Address (Virtual Address)	
P	Program area (in case of non specified)	0x0000 3000	Area P0 (caching is enabled, MMU address can be translated)
C	Constant area		
C$BSEC	Address structure for non-initialized data area		
C$DSEC	Address structure for initialized data area		
D	Initialized data (initial value)		
B	Non-initialized data area	0x0C00 0000	
R	Initialized data area		
S	Stack area	0x0FFF F9F0	
INTHandler	Exception/interrupt handler	0x8000 0800	Area P1 (caching is enabled, MMU address cannot be translated)
VECTTBL	Reset vector table Interrupt vector table		
INTTBL	Interrupt mask table		
PIntPRG	Interrupt function		
SP_S	Stack area for handler of TLB miss	0x8FFF FDF0	
RSTHandler	Reset handler	0xA000 0000	Area P2 (caching is disabled, MMU address cannot be translated)
PResetPRG	Reset program		
PnonCache	Program area (cache disabled access)		

3. Listing of Sample Program

The listing for a sample program that implements the flow of processing shown in figure 1 is given below.

1. Sample Program Listing: "sh7730.c"

```
/*""FILE COMMENT""*********** Technical reference data ***************
* System Name  : SH7730 Sample Program
* File Name    : sh7730.c
* Abstract     : Sample Program of Write-Back Operation for the SH7730 Cache
* Version      : Ver 1.00
* Device       : SH7730
* Tool-Chain   : SuperH RISC engine Standard Toolchain Ver.9.1.1.0
* OS           : None
* H/W Platform : The AP-SH4A-1A board incorporates the SH7730 with SH4A-CPU core
*                is available from AlphaProject Co., Ltd.
* Description  : Sample program of write-back operation for the SH7730 cache
*              :
* Operation    :
* Disclaimer   :
*              :
* Copyright (C) 2008. Renesas Technology Corp., All Rights Reserved.
*
*********************************************************************
* History      : 27.May.2008 Ver. 1.00 First Release
*""FILE COMMENT END""**************************************************/
#include <machine.h>
#include "cache.h"        /* Add cache function */

//#include "typedefine.h"
#ifdef __cplusplus
//#include <ios>                        // Remove the comment when you use ios
//_SINT ios_base::Init::init_cnt;       // Remove the comment when you use ios
#endif

void main(void);
#ifdef __cplusplus
extern "C" {
void abort(void);
}
#endif

/* ==== Macro definition ==== */
/* ---- Addresses of areas in SDRAM ---- */
#define D_SDRAM_ADDR1  (unsigned char *)(0x0c000000)     /* Cacheable area */
#define D_SDRAM_ADDR2  (unsigned char *)(0xac000000)     /* Non-cacheable area */

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

/*""FUNC COMMENT""*************************************************
* ID                          :
* Outline                     : Sample program main (example of cache usage)
* Include                     :
* Declaration                 : void main(void)
* Description                 : This is a sample program for write-back operation
*                             : in the operand cache by using the intrinsic
*                             : function: ocbp().
*                             :
*                             : Procedure
*                             : 1.
*                             : While the operand cache is being disabled,
*                             : fill data of one way (8,192 bytes) from
```

```
 58    *                               : 0x0c000000 with 0x00.
 59    *                               : → Fill the external memory with 0x00.
 60    *                               : 2.
 61    *                               : Enable the operand cache to fill 8,192-byte data from
 62    *                               : 0x0c000000 with 0x55.
 63    *                               : → Fill the operand cache with 0x55.
 64    *                               :
 65    *                               : 3.
 66    *                               : Compare the 8,192 bytes from 0x0c000000 (operand cache)
 67    *                               : with the 8,192 bytes from 0xAC000000 (external memory) to
 68    *                               : illustrate how a mismatch between the data in the external
 69    *                               : memory and the data in the operand cache is generated.
 70    *                               :
 71    *                               :
 72    *                               : 4.
 73    *                               : ocbp() is used to perform write-back operation
 74    *                               : in the operand cache.
 75    *                               : 5.
 76    *                               : After the write-back operation, the 8,192 bytes from
 77    *                               : 0xAC000000 (external memory)are compared with Hx55.
 78    *                               : Matching means that writing back succeeded.
 79    *                               :
 80    *                               :
 81    *                               :
 82    * Disclaimer                    : Program operation is based on the assumption that function
 83    *                               : main is called while the instruction/operand caches are
 84    *                               : disabled. In the example of initial settings, which should
 85    *                               : be made before calling the main() function described below,
 86    *                               : the instruction and operand caches are enabled.
 87    *                               : In response to this, delete the processing to enable
 88    *                               : the instruction/operand cache from the sample code for
 89    *                               : initialization, and then call this sample program.
 90    *                               :
 91    *                               :
 92    *                               :
 93    * Argument                      : none
 94    * Return Value                  : none
 95    * Calling Functions             :
 96    *""FUNC COMMENT END""*********************************************/
 97    void main(void)
 98    {
 99        int i;
100        unsigned char *ptr1,*ptr2;
101
102        /* ==== Caution ==== */
103        /* ==== It is assumed that the processing below is executed while   ==== */
104        /* ==== the IC/OC is disabled.                                        ==== */
105
106        /* ==== Data are written from the cacheable P0 area while the OC is disabled. ==== */
107        /* ==== External memory is filled with 0x00.                                ==== */
108        ptr1 = D_SDRAM_ADDR1;        /* Cacheable area (P0) */
109        for(i=0; i<8192; i++){
110                *ptr1++ = 0;
111        }
112
113        /* ==== The IC/OC is enabled. ==== */
114        cache_set_ccr(D_CACHE_I_ON | D_CACHE_O_ON );
115
116        /* ==== Data are written from the cacheable P0 area while the OC is enabled.    ==== */
117        /* ==== The operand cache is filled with 0x55.                                  ==== */
118        /* An amount of data that will not lead to writing back (8,192 bytes) is written to cache   */
119        ptr1 = D_SDRAM_ADDR1;          /* Cacheable area (P0) */
120        for(i=0; i<8192; i++){
121                *ptr1++ = 0x55;        /* ==== Set1 ==== */
```

```
122         }
123
124         /* ==== Comparison between cacheable area and non-cacheable area ==== */
125         ptr1 = D_SDRAM_ADDR1;          /* Cacheable area (P0)     */
126         ptr2 = D_SDRAM_ADDR2;          /* Non-cacheable area (P2) */
127         /* The program code under Check1 below is to ensure that writing back from     * /
128         /* the operand cache does not proceed before cache_set_purge() is executed.    * /
129         /* The Check1 program is executed under the following conditions:              * /
130         /* 1. By using the calculation below (calculation of comparison with maximum cache /
131         /* capacity), the amount of data written into the cache is small enough, relative /
132         /* to the maximum cache capacity, that writing back will not be generated.     * /
133         /* 2. Cache entries for stack, literal, and data by program operation other than /
134         /* the above setting of 0x55 (Set1) and cache entries produced by the above     */
135         /* setting of 0x55 (Set1)do not lead to more cache entries than the number      */
136         /* corresponding to the setting for number of ways in the  RAMCR.               */
137         /* As a result, the Check1 program will not go into an infinite loop due to    */
138         /* the generation of write-back operation.                                     */
139         /* If the Check1 program does go into an infinite loop, focus on the amount of  */
140         /* data being written into the cache as described above, and the writing of    */
141         /* entries to the cache for the stack area, literals, and data produced by     */
142         /* program operation other than the above setting of 0x55 to identify the      */
143         /* factor that is leading to write-back operations.                            */
144         /*                                                                             */
145         /* Calculation of comparison with maximum cache capacity                       */
146         /* Amount to be written into the cache (8,192 bytes)                           */
147         /*   < Maximum cache capacity: 32,768 bytes                                    */
148         /*     = line size (32 bytes) x number of entries (4) x number of ways (4)     */
149         /* ==== Check1 ==== */
150         for(i=0; i<8192; i++){
151                 if(*ptr1++ == *ptr2++){
152                         while(1){
153                                 /* Mistake in operand cache setting */
154                         }
155                 }
156         }
157         /* ==== Check1 ==== */
158
159         /* ==== Write-back in the operand cache ==== */
160         cache_set_purge();
161
162         /* ==== Comparison between cacheable area and non-cacheable area ==== */
163         ptr2 = D_SDRAM_ADDR2;        /* Non-cacheable area (P2) */
164
165         /* The Check2 program below confirms whether data have been written to external    */
166         /* memory by the write-back operation for the operand cache (cache_set_purge()).   */
167
168         /* ==== Check2 ==== */
169         for(i=0; i<8192; i++){
170                 if(*ptr2++ != 0x55){
171                         while(1){
172                                 /* Failure of writing back from the operand cache */
173                         }
174                 }
175         }
176         /* ==== Check2 ==== */
177
178         while(1){
179                 /* program end */
180         }
181
182     }
183
184
185
```

2. Sample Program Listing: "cache.c"

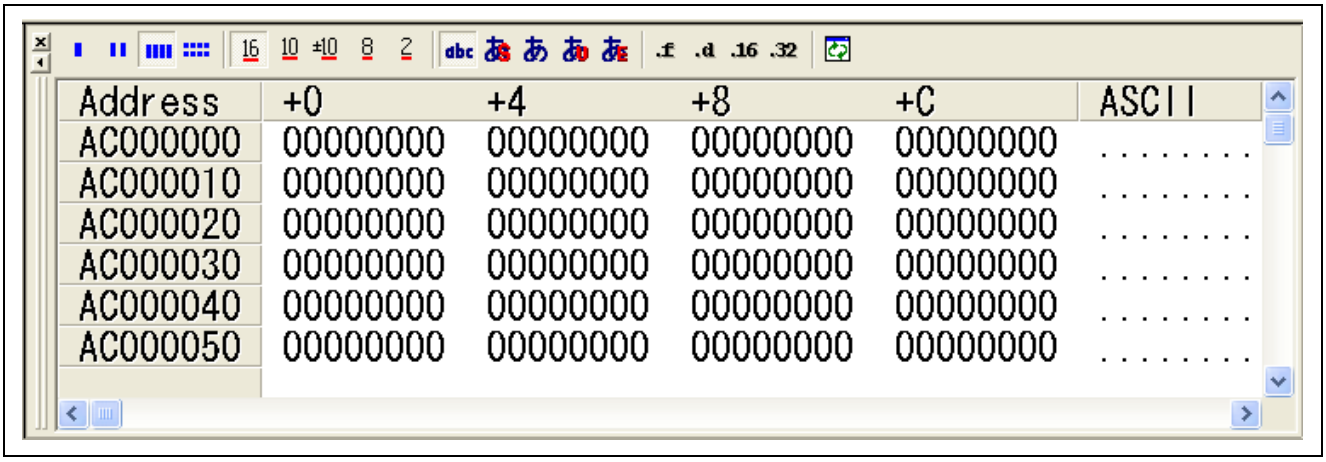
```
/*""FILE COMMENT""********** Technical reference data ***************
* System Name  : SH7730 Sample Program
* File Name    : cache.c
* Abstract     : Sample Program for Setting the SH7730 Cache
* Version      : Ver 1.00
* Device       : SH7730
* Tool-Chain   : SuperH RISC engine Standard Toolchain Ver.9.1.1.0
* OS           : None
* H/W Platform : The AP-SH4A-1A board incorporates the SH7730 with SH4A-CPU core
*                is available from AlphaProject Co., Ltd.
* Description  : Sample program for setting the SH7730 cache
*              :
* Operation    :
* Disclaimer   :
*              :
* Copyright (C) 2008. Renesas Technology Corp., All Rights Reserved.
*
*************************************************************************
* History      : 27.May.2008 Ver. 1.00 First Release
*""FILE COMMENT END""*************************************************/
#include <machine.h>
#include "iodefine.h"
#include "cache.h"

      …snip…

      /*""FUNC COMMENT""*****************************************************
      * ID                          :
      * Outline                     : Writing back from the cache
      * Include                     :
      * Declaration                 : void cache_set_purge(void)
      * Description                 : 8,192 bytes of data are written back to
      *                             : the addresses from the start of the data area.
      * Argument                    : none
      * Return Value                : none
      * Calling Functions           :
      *""FUNC COMMENT END""*************************************************/
      void cache_set_purge(void)
      {
          unsigned long addr = 0x0C000000;
          unsigned int i;

          /* ==== Setting the exception/interrupt block (BL) bit ==== */
          set_cr(get_cr() | 0x10000000);

          /* ====  Writing back all entries ==== */
          for(i=0; i<256; i++) {            /* 256 entries x 32 bytes = 8,192 bytes          */
                 ocbp((void *)addr);        /* Invalidation and writing back of a cache block
                                               (intrinsic function)                          */
                 addr +=0x20 ;
          }

          /* ==== Releasing the exception/interrupt block (BL) bit ==== */
          set_cr(get_cr() & ~(0x10000000));

      }
```

3.1 Results of Execution

3.1.1 State before Write-Back Operation

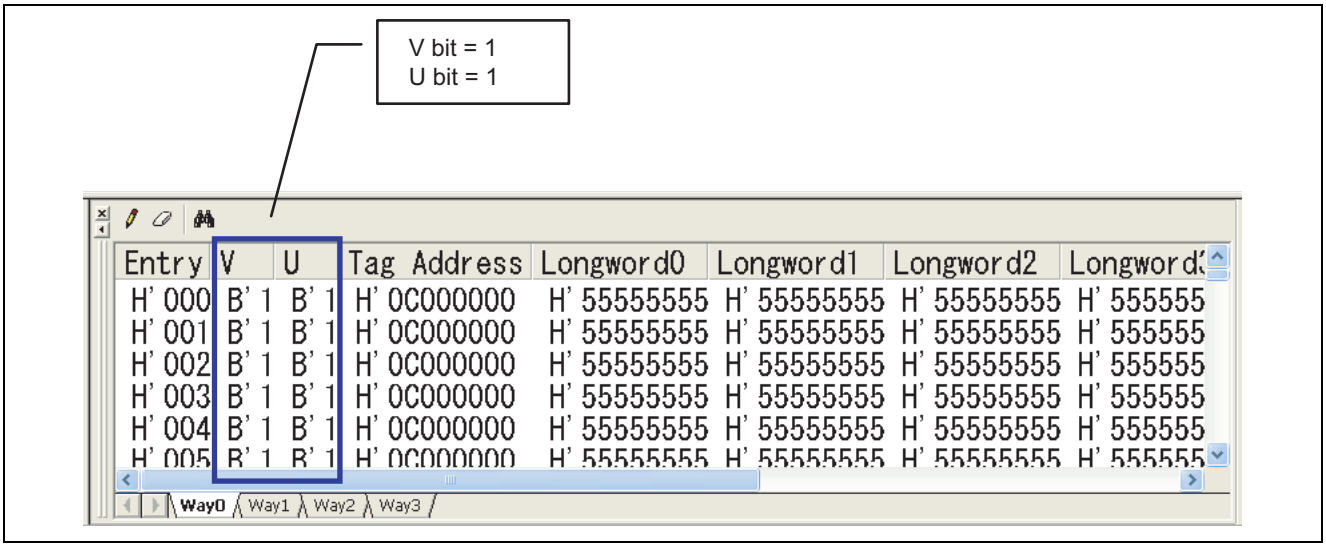
Figures 6 and 7 show data dumped from external memory and from the operand cache before the write-back operation. Figures 6 and 7 illustrate how data in the cache line and in external memory do not match. We can also see this in the value of the U bits (1) in figure 7.

Figures 6 and 7 below were obtained from the memory and operand-cache windows of the High-performance Embedded Workshop.



Address	+0	+4	+8	+C	ASCII
AC000000	00000000	00000000	00000000	00000000	....
AC000010	00000000	00000000	00000000	00000000	....
AC000020	00000000	00000000	00000000	00000000	....
AC000030	00000000	00000000	00000000	00000000	....
AC000040	00000000	00000000	00000000	00000000	....
AC000050	00000000	00000000	00000000	00000000	....

Figure 6 External Memory Area before Write-Back Operation



Entry	V	U	Tag Address	Longword0	Longword1	Longword2	Longword3
H'000	B'1	B'1	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555
H'001	B'1	B'1	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555
H'002	B'1	B'1	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555
H'003	B'1	B'1	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555
H'004	B'1	B'1	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555
H'005	B'1	B'1	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555

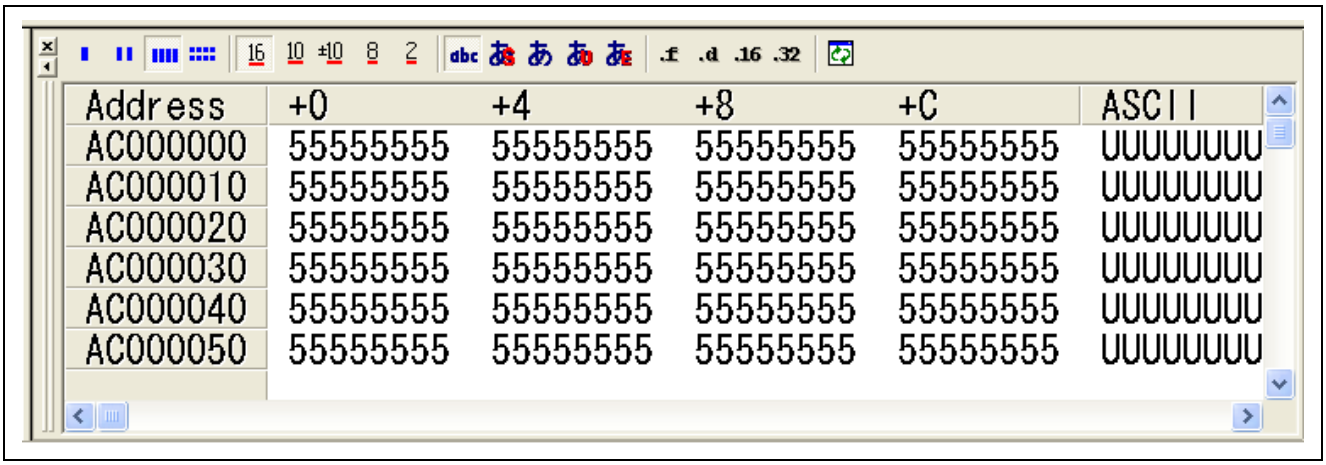
Way0 Way1 Way2 Way3

Figure 7 Operand Cache Area before Write-Back Operation

3.1.2 State after Write-Back Operation

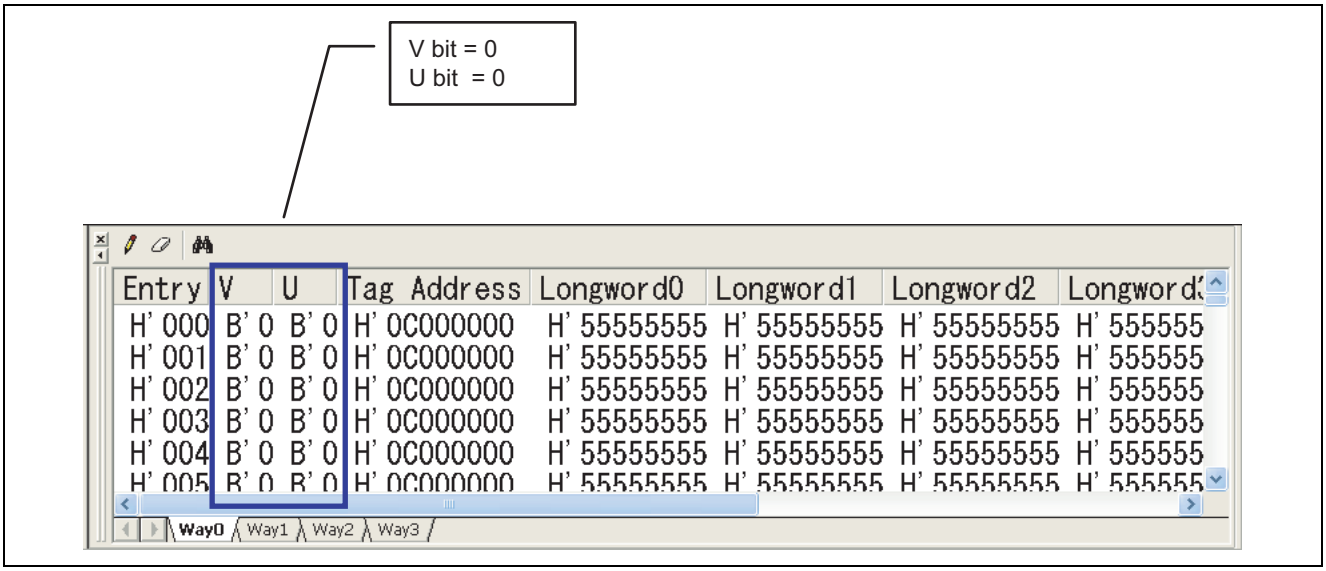
If the ocbp() intrinsic function is called to invalidate the cache and execute writing back from the cache while the V bit is 1 (indicating that the cache line contains valid data) and the U bit is 1 (indicating that data in the cache line and in external memory do not match), data dumped from external memory and from the operand cache are as shown in figures 8 and 9. That is to say, both the U and V bits are 0 for the cache entries and the data from the cache line have been written back to external memory.

Figures 7 and 8 below were obtained from the memory and operand-cache windows of the High-performance Embedded Workshop.



Address	+0	+4	+8	+C	ASCII
AC000000	55555555	55555555	55555555	55555555	UUUUUUUU
AC000010	55555555	55555555	55555555	55555555	UUUUUUUU
AC000020	55555555	55555555	55555555	55555555	UUUUUUUU
AC000030	55555555	55555555	55555555	55555555	UUUUUUUU
AC000040	55555555	55555555	55555555	55555555	UUUUUUUU
AC000050	55555555	55555555	55555555	55555555	UUUUUUUU

Figure 8 External Memory Area after Write-Back Operation



Entry	V	U	Tag Address	Longword0	Longword1	Longword2	Longword3
H'000	B'0	B'0	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555
H'001	B'0	B'0	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555
H'002	B'0	B'0	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555
H'003	B'0	B'0	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555
H'004	B'0	B'0	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555
H'005	B'0	B'0	H'0C000000	H'55555555	H'55555555	H'55555555	H'555555

Way0 Way1 Way2 Way3

Figure 9 Operand Cache Area after Write-Back Operation

4. Documents for Reference

- Software Manual
  SH-4A Software Manual (REJ09B0003)
  The most up-to-date versions of the documents are available on the Renesas Technology Website.
- Hardware Manual
  SH7730 Group Hardware Manual (REJ09B0359)
  The most up-to-date versions of the documents are available on the Renesas Technology Website.

Website and Support

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

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

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Mar.27.09	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.