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

Examples of Cache Memory Settings

Introduction

This application note describes examples of cache settings and operation for the SH7730.

Target Device

SH7730

Contents

1. Preface

1.1 Specifications

In this sample program, the instruction and the operand cache are enabled and placed in the copy-back mode.

As an example for reference, the sample program illustrates a situation where data in the cache memory and in external memory do not match during operation in copy-back mode.

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 and 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

Caching is disabled immediately after a power-on reset or manual reset. This application note describes how to set the caches.

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

Note: For details on cache memory, see the section on cache memory in the *SH7730 Group Hardware Manual* (REJ09B0359).

2.2 Write Modes of the Operand Cache

Two modes of writing, copy-back mode and write-through mode, are available for the operand cache of the SH7730. This section describes operation of the operand cache in cases where this cache is enabled and data are written to a cacheable area. Furthermore, precautions on the software side when using the operand cache are also described. For details on writing, reading, and prefetching, see the sections on write, read, and prefetch operations for the cache in the *SH7730 Group Hardware Manual* (REJ09B0359).

2.2.1 Operation in Write-Through Mode

The tag, V bit, U bit, and LRU bits on each way are read from the cache line indexed by virtual address bits [12:5]. The tags read from each way are compared with bits [28:10] of the physical addresses produced through virtual address translation by the MMU.

- If the tag matches and the V bit is 1, see (1). Cache hit.
- If the tag matches and the V bit is 0, see (2). Cache miss.
- If the tag does not match and the V bit is 0, see (2). Cache miss.
- If the tag does not match, and the V bit is 1 and the U bit is 0, see (2). Cache miss.
- If the tag does not match and both the V bit and U bit are 1, see (2). Cache miss.

(1) Cache hit

The same data are simultaneously written to the cache and external memory.

1. The CPU writes the amount of data corresponding to the access size to a data field on the hit way indicated by virtual address bits [4:0].
2. The data are written to the write-through buffer. When this is completed, the CPU proceeds to the next operation without waiting for completion of the writing of data to external memory.
3. Data are also written to the external memory corresponding to the virtual address.
4. The LRU bits are updated to indicate that the hit way is the most recently used.
5. The U bit is not updated.

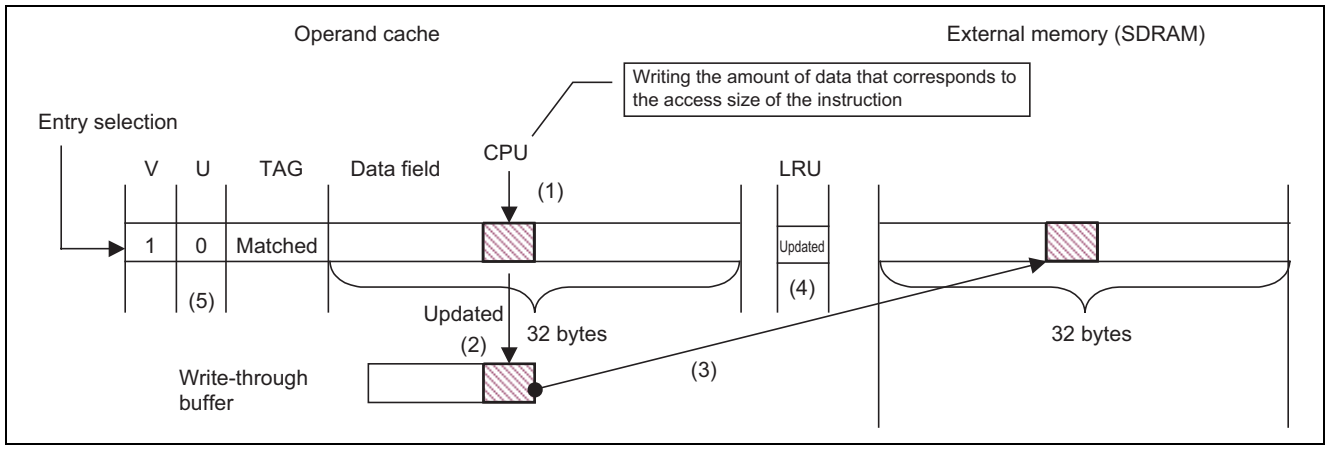

Figure 1 Write-Through Mode (Cache Hit)

(2) Cache miss

Data are not written to the cache.

1. Write data are written to the write-through buffer. When writing of data to the write-through buffer is completed, the CPU proceeds to the next operation without waiting for completion of the writing of data to external memory.
2. Data of the specified access size are written to the external memory corresponding to the virtual address.
3. The tag, V bit, U bit, and LRU bits are not updated.

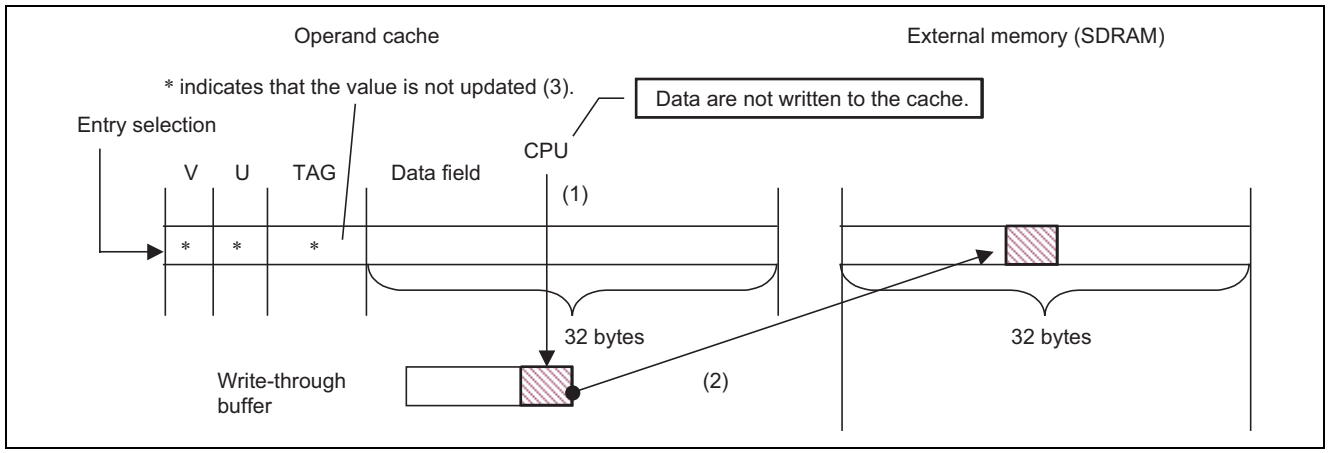

Figure 2 Write-Through Mode (Cache Miss)

2.2.2 Operation in Copy-Back Mode

The tags read from each way are compared with bits [28:10] of the physical address produced through virtual address translation by the MMU.

- If the tag matches and the V bit is 1, see (1). Cache hit.
- If the tag matches and the V bit is 0, see (2). Cache miss (without write-back).
- If the tag does not match and the V bit is 0, see (2). Cache miss (without write-back).
- If the tag does not match, and the V bit is 1 and the U bit is 0, see (2). Cache miss (without write-back).
- If the tag does not match, and both the V bit and U bit are 1, see (3). Cache miss (with write-back).

(1) Cache hit

1. The CPU writes the amount of data corresponding to the access size to a data field on the hit way indicated by virtual address bits [4:0].
2. 1 is written to the U bit.
3. The LRU bits are updated to indicate that the hit way is the most recent one.

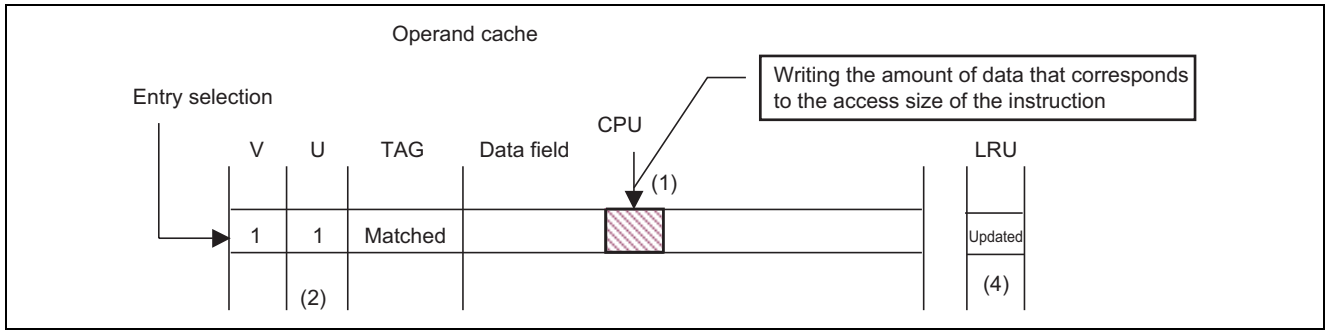

Figure 3 Copy-Back Mode (Cache Hit)

As illustrated in figure 3, data are only written to the cache, i.e. are not written to external memory. The result is that data in the operand cache does not in general match data in the external memory. When an external device (DMAC) refers to an area that is being cached, data should be written back by software to ensure coherence between the cache and external memory. If this is not done, operation may not be as expected.

As for coherence control for the operand cache, see the SH7730 Group Application Note: *Example of Writing Back from the Operand Cache* (REJ06B0853)

(2) Cache miss (without write-back)

1. The CPU writes the amount of data corresponding to the access size to a data field on the target way for writing indicated by virtual address bits [4:0].
2. Data from the physical address space corresponding to the virtual address are written to the cache line on the way which is selected for writing (excluding the data for which the cache miss occurred, since this has already been written). While the remaining data on the cache line are being read, the CPU can execute further processing.
3. At the point where reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and the U bit on the way
4. The LRU bits are updated to indicate that the way is the most recent one.

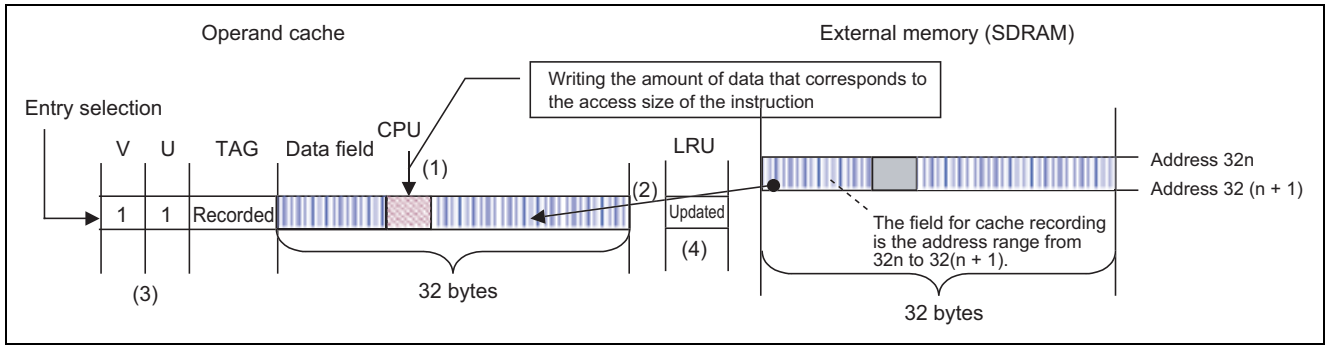

Figure 4 Copy-Back Mode (Cache Miss, without Write-Back)

(3) Cache miss (with write-back)

1. In accord with the LRU algorithm, selection is of the least recently accessed cache line on the way that is the target for replacement. The tag and data field of that cache line on the target way for replacement are then saved in the write-back buffer.
2. The CPU writes the amount of data corresponding to the access size to a data field on the hit way indicated by virtual address bits [4:0].
3. Data from the physical address space corresponding to the virtual address are written to the cache line on the target way for replacement (excluding the data for which the cache miss occurred, since this has already been written). While the remaining data on the cache line are being read, the CPU can execute further processing.
4. At the point where reading of one line of data is completed, the tag corresponding to the physical address is recorded in the cache, 1 is written to the V bit and the U bit on the way.
5. The LRU bits are updated to indicate the replaced way is the most recent one.
6. The data in the write-back buffer is then written back to external memory.

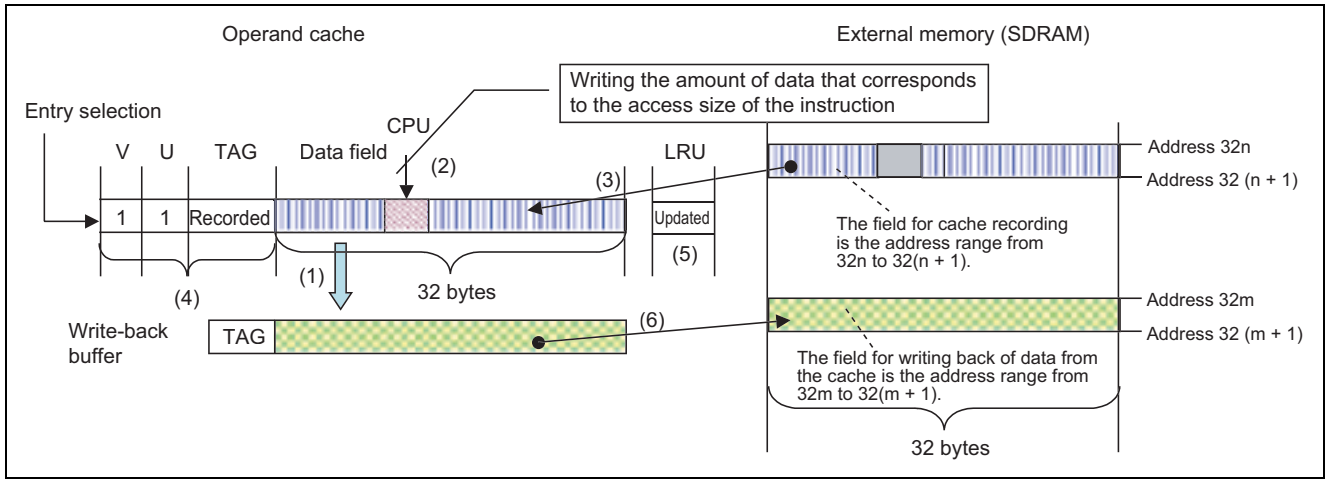

Figure 5 Copy-Back Mode (Cache Miss, with Write-Back)

2.3 Advantages and Disadvantages of Write-Through and Copy-Back Operation

Table 2 Advantages and Disadvantages of Write-Through and Copy-Back Operation

	Advantages	Disadvantages
Write-through	For access from the CPU, data in the operand cache and in external memory always match.	The more often data are written, the more frequently the external bus is used. The speed of operations is thus lower than in copy-back mode.
Copy-back	Since data are only written to the cache in the case of a cache hit, high-speed operation is enabled by making effective use of the external bus.	Even in the case of access from the CPU, data in the operand cache and in external memory do not necessarily match. Software must thus ensure coherence.

2.4 Setting up the Caches

2.4.1 Procedure for Cache Settings

This section describes how the caches are set up. The cache control register (CCR) is used to make settings for the cache mode. After the CCR setting, the ICBI instruction is issued in this sample program as processing before access to cacheable areas. Any code that manipulates the CCR should be placed in the non-cacheable P2 area or in IL memory. In the sample program, such code is placed in the non-cacheable P2 area.

In the sample program, the exception/interrupt block (BL) bit in the status register (SR) is set to 1 so that interrupts are not accepted during access to a cache-enabled space while the cache mode is being updated.

Figure 6 shows an example of the procedure for settings to enable the instruction cache and operand cache.

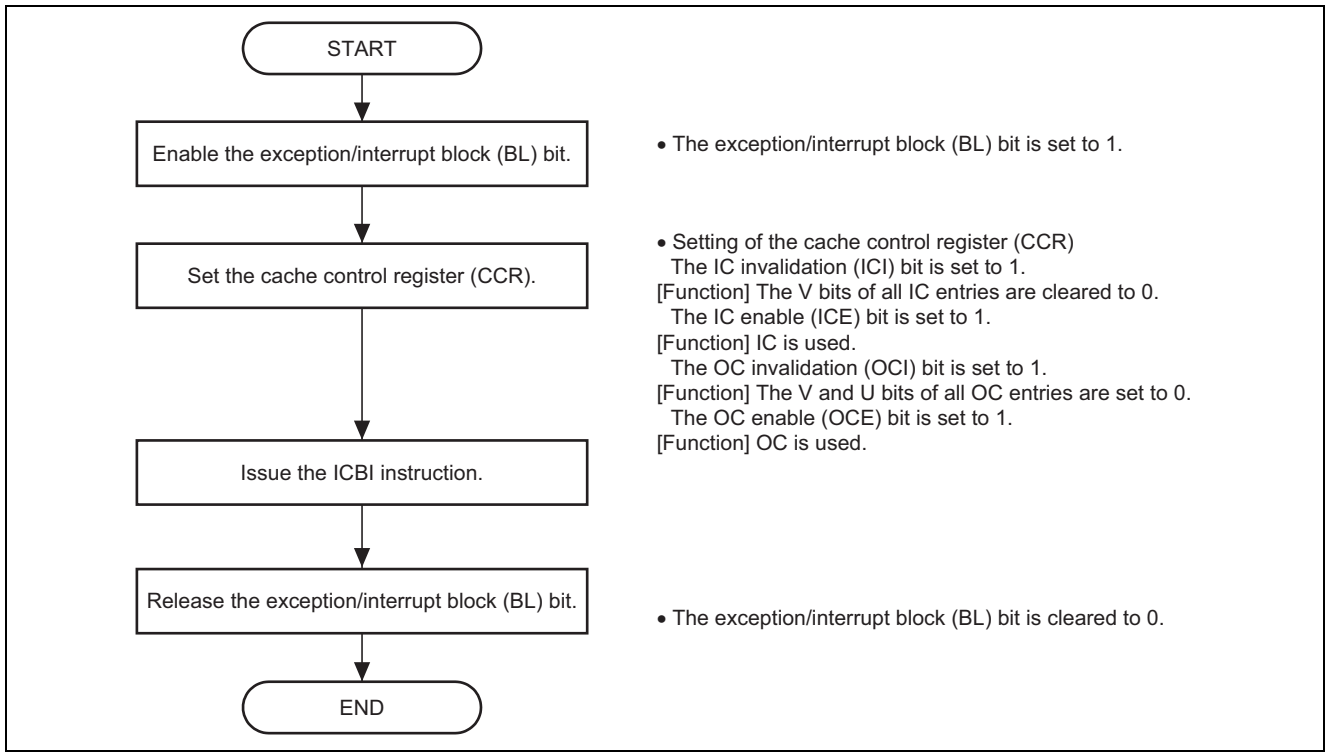

Figure 6 Procedure for Cache Settings

Notes: 1. Execute the sample program in privileged mode.
2. In this sample program, the MMU is left at its initial value (the MMU is disabled).

2.4.2 Settings for the Cache Control Register (CCR)

Table 3 Configuration of Bits in the CCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ICI	—	—	ICE	—	—	—	—	OCI	CB	WT	OCE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 4 Description of Bits in the CCR

Bit	Bit Name	Initial value	R/W	Description
11	ICI	0	R/W	IC Invalidation Bit When 1 is written to this bit, the V bits of all IC entries are cleared to 0. This bit is always read as 0.
8	ICE	0	R/W	IC Enable Bit Selects whether the IC is used. Note that however when address translation is performed, the IC cannot be used unless the C bit in the page management information is also 1. 0: IC is not used. 1: IC is used.
3	OCI	0	R/W	OC Invalidation Bit When 1 is written to this bit, the V and U bits of all OC entries are cleared to 0. This bit is always read as 0.
2	CB	0	R/W	Copy-Back Bit Indicates the write mode of caching for the P1 area. 0: Write-through mode 1: Copy-back mode
1	WT	0	R/W	Write-Through Mode Indicates write mode of cache in the P0, U0, and P3 areas. When address translation is performed, the value of the WT bit in the page management information has priority. 0: Copy-back mode 1: Write-through mode
0	OCE	0	R/W	OC Enable Bit Selects whether the OC is used. Note that however when address translation is performed, the OC cannot be used unless the C bit in the page management information is also 1. 0: OC is not used. 1: OC is used.

Table 5 gives a list of the cache settings made by the sample program.

Table 5 Cache Settings

Name of Register	Address	Setting	Function
Cache Control Register (CCR)	H'FF00 001C	H'0000 0909	ICI = 1: IC invalidation bit ICE = 1: IC enable bit OCI = 1: OC invalidation bit CB = 0: Copy-back bit WT = 0: Write-through mode OCE = 1: OC enable bit Note: The ICI and OCI bits are read as 0.

2.5 Operation by the Sample Program

The sample program illustrates a situation where data in the cache memory do not match data in the corresponding area of external memory during operation in copy-back mode.

Section 2.5.1 describes the procedure for processing by the sample program and section 2.5.2 describes the cache-related operations produced by running the sample program.

2.5.1 Procedure for Processing by the Sample Program

The procedure for processing by the sample program is described below. Please refer to this section in conjunction with figure 7.

(1) While the instruction cache and operand cache are disabled, memory corresponding to a single cache line of data (32 bytes) from H'0C00 0000 is filled with H'00. Since the operand cache is disabled, the data are written to external memory instead of the operand cache.
(2) The instruction cache and operand cache are enabled in copy-back mode.
(3) The 32 bytes from H'0C00 0000 are filled with H'55. Since copy-back mode has been selected by this point, data are written to the operand cache rather than to external memory.
(4) The 32 bytes from H'AC00 0000 (data in the external memory) and the 32 bytes from H'0C00 0000 (data in the operand cache, since there is a cache hit for this address) are compared. Since the value of each byte in external memory is H'00 and the value of each byte in the operand cache is H'55, the data do not match.

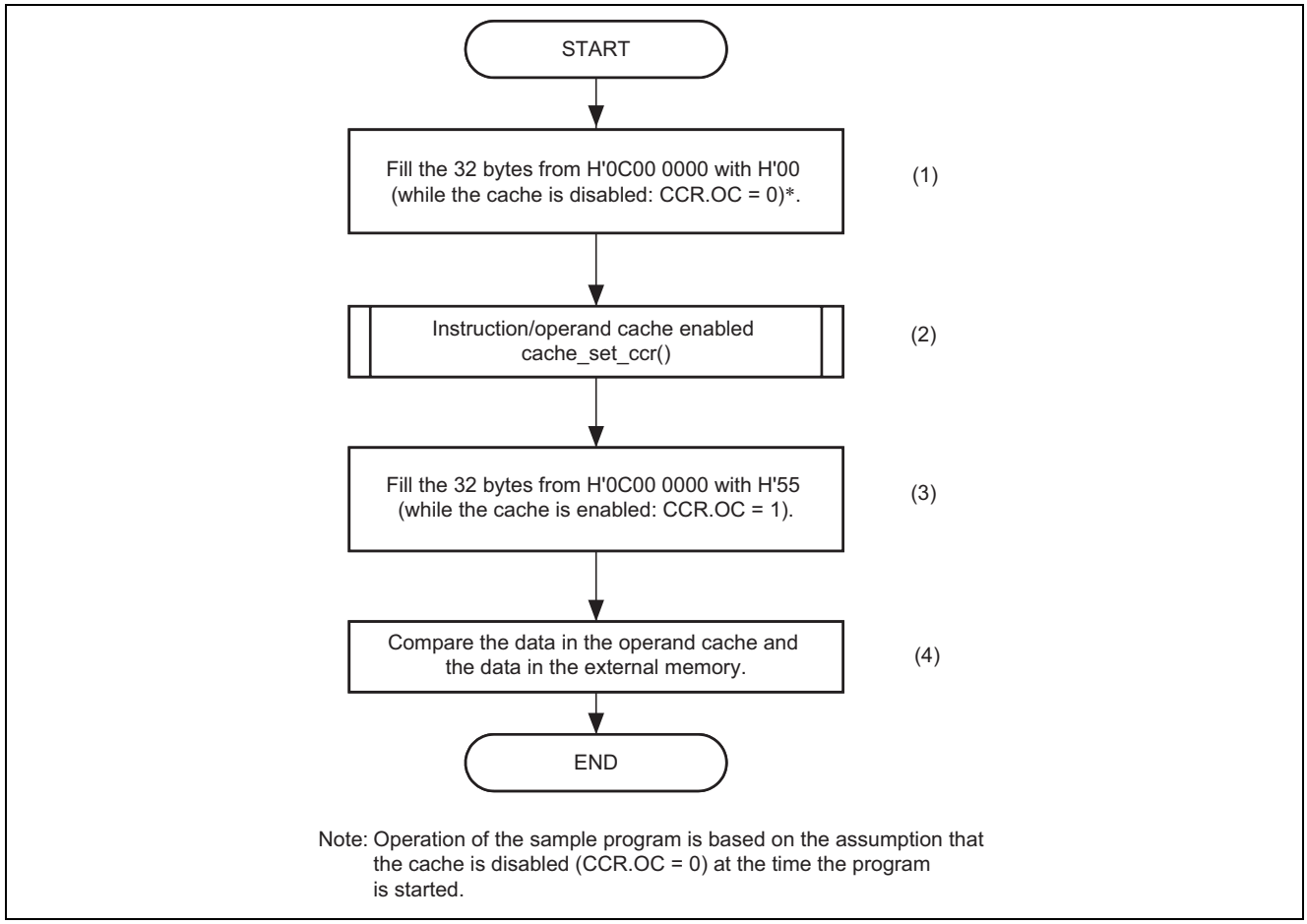

Figure 7 Flow of Processing by the Sample Program

2.5.2 Cache Operation during Operation by the Sample Program

This section describes operations after the state of the operand cache has been changed from being disabled to enabled, specifically those related to the cache miss (without write-back) the first time the value H'55 is written (to the address H'0C00 0000) and to the cache hit the second time the value H'55 is written (to the address H'0C00 0001). Accordingly, it describes the cache operations during execution of the sample program.

1. Writing the first time
   The following describes operations after the operand cache has been enabled, specifically those related to the cache miss (without write-back) the first time the value H'55 is written (to the address H'0C00 0000). Refer to the descriptions in conjunction with figure 8.

   (1) Since data are to be written to the address H'0C00 0000, the entry is set to 0 from the [12:5] bits. The tag, V bit, U bit, and LRU bits on each way are read from the cache line indicated by entry 0. Immediately before the data are written, the V and U bits of all entries in the operand cache will have been cleared to 0 by the OC invalidation (OCI) bit in the cache control register (CCR), so the V and U bits which are read out will be 0. Since the tag that was read is not initialized in a power-on reset or manual reset, its value is undefined.
   (2) The [28:10] bits of the H'0C00 0000 address are compared with the tags which have been read from each way, and values of the V and U bits are consulted. Since the tag values are undefined and the value of the V and U bits is 0 as stated in (1), a cache miss (without write-back) occurs.
   (3) The value H'55 is written to the destination within the operand cache, i.e. to entry 0 on the target way for replacement, in the data field which is indicated by the [4:0] bits of the address (H'0C00 0000).
   (4) Data other than that for which the cache miss occurred are read from an external memory with the corresponding addresses H'0C00 0000 to H'0C00 001F and written to the data field of the entry 0 cache line on the target way for replacement.
   (5) At the point where data for one line have been read into the data field of the entry 0 cache line (4), the [28:10] bits of the address (H'0C00 0000) are recorded in the entry 0 tag on the target way for replacement, then 1 is written to the V and U bits.
   (6) The LRU bits are updated to indicate that the replaced way is the most recently used.

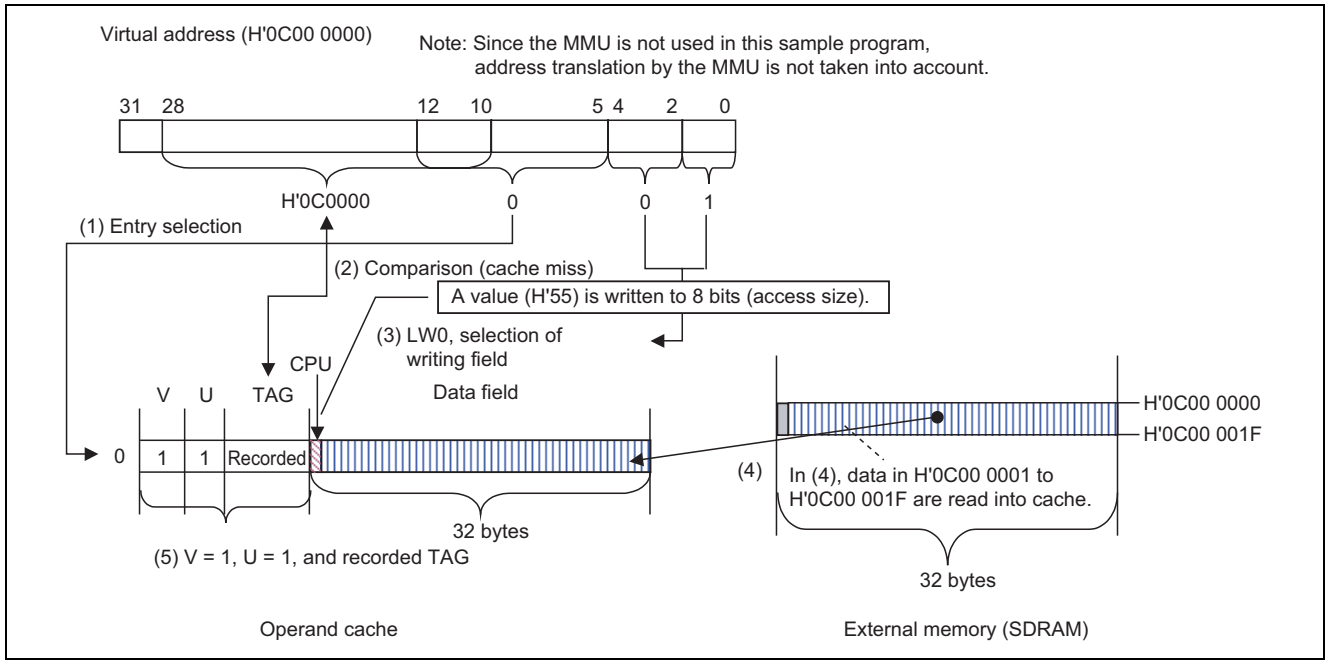

Figure 8 Cache Operation by the Sample Program (Writing the First Byte)

2. Writing the second time

The following describes operations related to the cache hit that occurs in response to writing of the value H'55 for the second time (to the address H'0C00 0001), i.e. after the value H'55 data was written for the first time (to address H'0C00 0000). Refer to the descriptions in conjunction with figure 9

(1) Since data are to be written to the address H'0C00 0001, the entry is set to 0 from the [12:5] bits. The tag, V bit, U bit, and LRU bits on each way are read from the cache line indicated by entry 0. From writing of the first byte, one way will have V and U bits that are 1 and hold H'0C0000 as its tag.

(2) The [28:10] bits of the H'0C00 0001 address are compared with the tag which have been read from each way, and values of the V and U bits are consulted. Since there is a way including an entry whose tag matches at H'0C0000 and the V bit is 1 from (1), a cache hit occurs.

(3) The value H'55 is written to the destination within the operand cache, i.e. to entry 0 on the hit way, in the data field which is indicated by the [4:0] bits of the address (H'0C00 0001).

(4) 1 is written to the U bit.

(5) The LRU bits are updated to indicate the hit way is the most recently used.

Subsequent writing of data up to H'0C00 001F will lead to cache hits, so the data will only be set in the cache. Furthermore, writing back from the cache will not occur. The result is that data in the cache will differ from data in external memory.

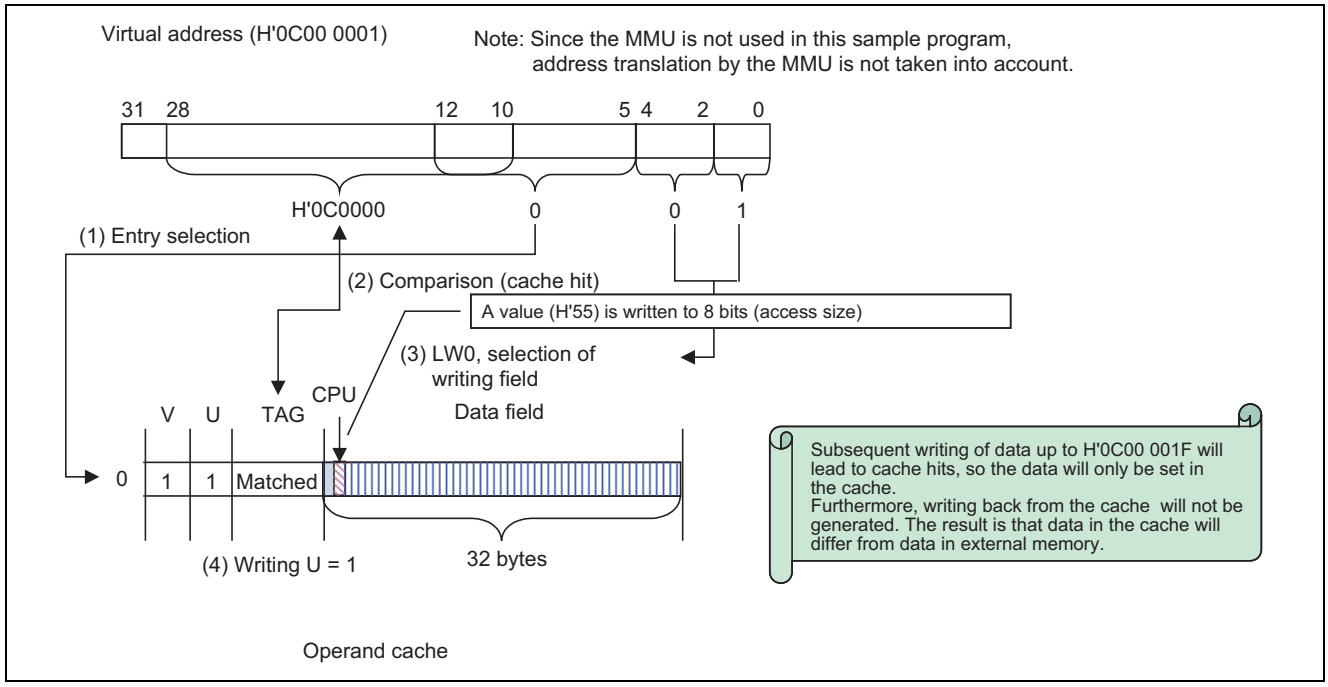

Figure 9 Cache Operation by the Sample Program (Writing the Second Byte)

2.6 Allocation of Sections for 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. Then the PnonCache section is placed in the non-cacheable P2 area of the SH7730. Allocation of sections (address specification) is specified by an option of a linker.

Table 6 gives information on sections in the sample program.

Table 6 Information on Sections for 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 misses	0x8FFF FDF0	
RSTHandler	Reset handler	0xA000 0000	Area P2 (caching is disabled, MMU address cannot be translated)
PResetPRG	Reset program		
PnonCache	Program area (non-cacheable access)		

3. Listing of Sample Program

1. Sample Program Listing: "sh7730.c"

```
/*""FILE COMMENT""*********** Technical reference data ***************
* System Name  : SH7730 Sample Program
* File Name    : sh7730.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 and 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 "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

/* ---- Addresses of areas in SDRAM ---- */
#define D_SDRAM_ADDR1  (unsigned char *)(0x0c000000)     /* Cacheable P0 area */
#define D_SDRAM_ADDR2  (unsigned char *)(0xac000000)     /* Non-cacheable P2 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 which indicates a mismatch
*                             : between the data in the operand cache and the data in
*                             : external memory in copy-back operation.
*                             :
*                             :   Procedure
*                             : 1.
*                             : While the operand cache is being disabled,
*                             : fill data of one cache line (32 bytes) from
*                             : 0x0c000000 with 0x00.
*                             : → Fill the external memory with 0x00.
*                             : 2.
*                             : Enable the operand cache to fill 32-byte data from
*                             : 0x0c000000 with 0x55.
```

```
 62    *                               : → Fill the operand cache with 0x55.
 63    *                               : 3.
 64    *                               : Compare the 32 bytes from 0x0c000000 (operand cache) with
 65    *                               : the 32 bytes from 0xAC000000 (external memory) to
 66    *                               : illustrate how a mismatch between the data in the external
 67    *                               : memory and the data in the operand cache is generated.
 68    *                               :
 69    *                               :
 70    * Disclaimer                    : Program operation is based on the assumption that function
 71    *                               : main is called while the instruction/operand caches are
 72    *                               : disabled. In the example of initial settings, which should
 73    *                               : be made before calling the main() function described below,
 74    *                               : the instruction and operand caches are enabled.
 75    *                               : In response to this, delete the processing to enable
 76    *                               : the instruction/operand caches from the sample code for
 77    *                               : initialization, and then call this sample program.
 78    *                               :
 79    *                               :
 80    *                               :
 81    * Argument                      : none
 82    * Return Value                  : none
 83    * Calling Functions             :
 84    *""FUNC COMMENT END""*********************************************/
 85    void main(void)
 86    {
 87      int i;
 88      unsigned char *ptr1,*ptr2;
 89
 90      /* ==== Caution ==== */
 91      /* ==== It is assumed that the processing below is executed while     ==== */
 92      /* ==== the IC/OC is disabled.                                        ==== */
 93
 94
 95      /* ==== Data are written from the cacheable P0 area while the OC is disabled. ==== */
 96      /* ==== External memory is filled with 0x00.                                   ==== */
 97      ptr1 = D_SDRAM_ADDR1;
 98      for(i=0; i<32 ; i++){
 99        *ptr1++ = 0x00;
100      }
101
102      /* ==== The IC/OC is enabled. ==== */
103      cache_set_ccr(D_CACHE_I_ON | D_CACHE_O_ON | D_CACHE_I_INVALID | D_CACHE_O_INVALID);
104
105      /* ==== Data are written from the cacheable P0 area while the OC is enabled. ==== */
106      /* ==== The operand cache is filled with 0x55.                               ==== */
107      ptr1 = D_SDRAM_ADDR1;
108      for(i=0; i<32 ; i++){
109        *ptr1++ = 0x55;
110      }
111
112      /* ==== Data in external memory and data in OC are compared. ==== */
113      ptr1 = D_SDRAM_ADDR1;       /* Cacheable P0 area */
114      ptr2 = D_SDRAM_ADDR2;       /* Non-cacheable P2 area */
115
116      for(i=0; i<32; i++){
117        if(*ptr1++ == *ptr2++){
118            while(1){
119                /* Data in external memory match data in the OC */
120                /* (unintended case). */
121
122            }
123        }
124      }
125
```

```
126       /* ==== Non-matching of the data is indicated by execution not going into the above
127       infinite loop. ==== */
128       while(1){
129         /* Program end */
130       }
131
        }
```

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 and 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"

#pragma section nonCACHE  /* Allocation to the CS0 cache disabled space */
/*""FUNC COMMENT""*****************************************************
* ID                          :
* Outline                     : Cache Setting
* Include                     :
* Declaration                 : void cache_set_ccr(unsigned int i_mode)
* Description                 : Cache registers are set.
*                             :
* Argument                    : unsigned int i_mode
*                             : Following modes are set by logical OR.
*                             : D_CACHE_I_INVALID       : IC invalidation
*                             : D_CACHE_I_ON            : IC enabled
*                             : D_CACHE_O_INVALID       : OC invalidation
*                             : D_CACHE_O_ON            : OC enabled
*                             : D_CACHE_IO_ON           : IC/OC enabled
*                             : D_CACHE_O_WT            : Write-through mode
*                             : D_CACHE_OFF             : IC/OC disabled
* Return Value                : none
* Calling Functions           :
*""FUNC COMMENT END""************************************************/
void cache_set_ccr(unsigned int i_mode)
{
  /* ==== Setting the exception/interrupt block bit (BL) ==== */
  set_cr(get_cr() | 0x10000000);

  /* ==== Setting cache registers ==== */
  CACHE.CCR.LONG = i_mode;

  /* ==== Issuing the ICBI instruction to enable cache ==== */
  icbi(0);

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

}
```

3. Sample Program Listing: "cache.h"

```
/*""FILE COMMENT""************ Technical reference data ***************
* System Name  : SH7730 Sample Program
* File Name    : cache.h
* 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 and 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""*************************************************/

#ifndef  __CACHE_DEF_H__
#define __CACHE_DEF_H__

/* ==== Macro definition ==== */
/* ---- Cache setting ---- */
#define D_CACHE_OFF               0x0000u
#define D_CACHE_I_INVALID         0x0800u
#define D_CACHE_I_ON              0x0100u
#define D_CACHE_O_INVALID         0x0008u
#define D_CACHE_O_ON              0x0001u
#define D_CACHE_IO_ON             (CACHE_I_ON | CACHE_O_ON)
#define D_CACHE_O_WT              0x0002u

/* ---- Area to be purged ---- */
#define D_CACHE_PURGE_SDRAM_START     *( volatile unsigned long  *)(0x0C000000)
#define D_CACHE_PURGE_SDRAM_END       *( volatile unsigned long  *)(0x0FFFFFFF)

/* ==== Function declaration ==== */
void cache_set_ccr(unsigned int i_mode);
void cache_Purge_OCBP(unsigned long *i_start, unsigned long *i_end);


#endif /* __CACHE_DEF_H__ */
```

3.1 Result of Execution

The data in external memory and in the cache after execution of the sample program are as shown below.

When data are read from the non-cacheable P2 area, the data are actually read from external memory (see figure 10).

When data that have been cached are read from the cacheable P0 area, the data are read from cache memory instead of external memory (see figure 11).

In figures 10 and 11, we can see how the contents of cache memory may differ from the values in external memory when copy-back mode is selecting as the mode of writing for the cache memory.

In other words, coherence between cache memory and external memory is not necessarily maintained when the cache is in copy-back mode. When external devices such as the DMAC refer to an area for which caching has been enabled, software must ensure that a write-back operation proceeds beforehand. For details on write-back operations, see the SH7730 Group Application Note: *Example of Writing Back from the Operand Cache* (REJ06B0853).

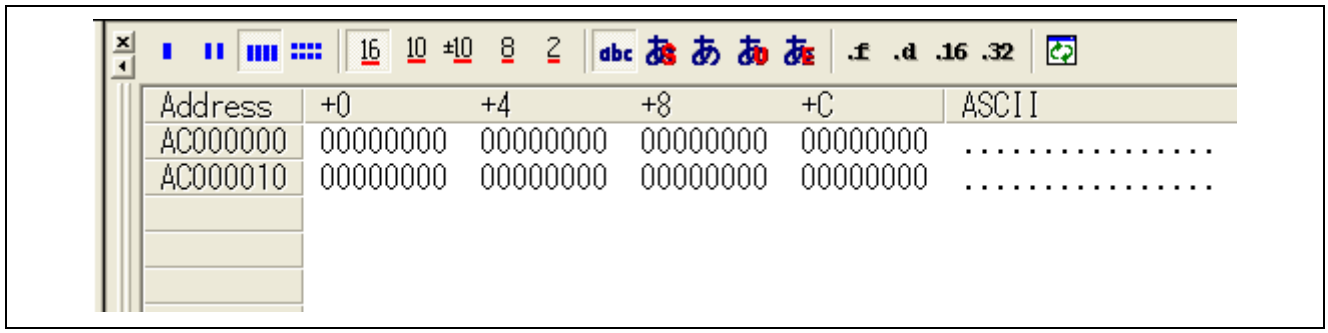

Figure 10 Memory Dump from Access to the Non-Cacheable P2 Area

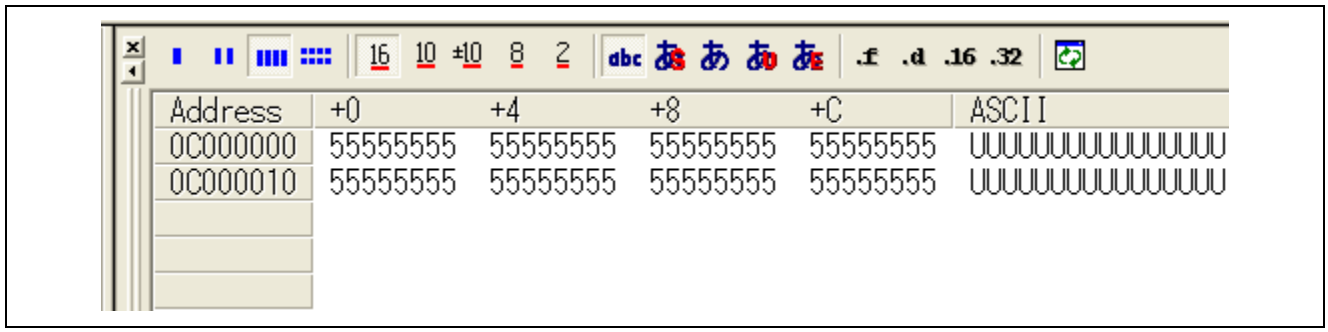

Figure 11 Memory Dump from Access to the Cacheable P0 Area

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.