SH7670 Group

Using the DMAC to Transfer Data between Memory Areas

R01AN0305EJ0101
Rev. 1.01
Oct. 15, 2010

Summary

This application note provides an example of transferring data between memory areas with the direct memory access controller (DMAC) of the SH7670.

Target Device

SH7670 MCU

Contents

1. Introduction

1.1 Specifications

- DMAC channel 0 is used to transfer data from the on-chip RAM to external memory. Data are transferred in cycle-stealing mode.
- Auto-request mode (software transfer request) is used for requesting DMA transfer.

1.2 Module Used

- Direct memory access controller (DMAC channel 0)

1.3 Applicable Conditions

MCU	SH7670
Operating Frequency	Internal clock: 200 MHz Bus clock: 66.6 MHz Peripheral clock: 33.3 MHz
Integrated Development Environment	Renesas Electronics High-performance Embedded Workshop Ver.4.03.00
C Compiler	Renesas Electronics SuperH RISC engine Family C/C++ compiler package Ver.9.01 Release 01
Compiler Options	Default setting in the High-performance Embedded Workshop (-cpu=sh2afpu -fpu=single -debug -gbr=auto -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1)

1.4 Related Application Notes

For more information, refer to the following application notes:

- SH7670 Group Example of Initialization
- SH7670 Group DMAC Dual Address mode
- SH7670 Group Using the DMAC to Transfer Data to On-chip Peripheral Modules

2. Description of the Sample Application

This sample application employs the direct memory access controller (DMAC) to transfer data from the on-chip RAM to external memory.

2.1 Operational Overview of Module Used

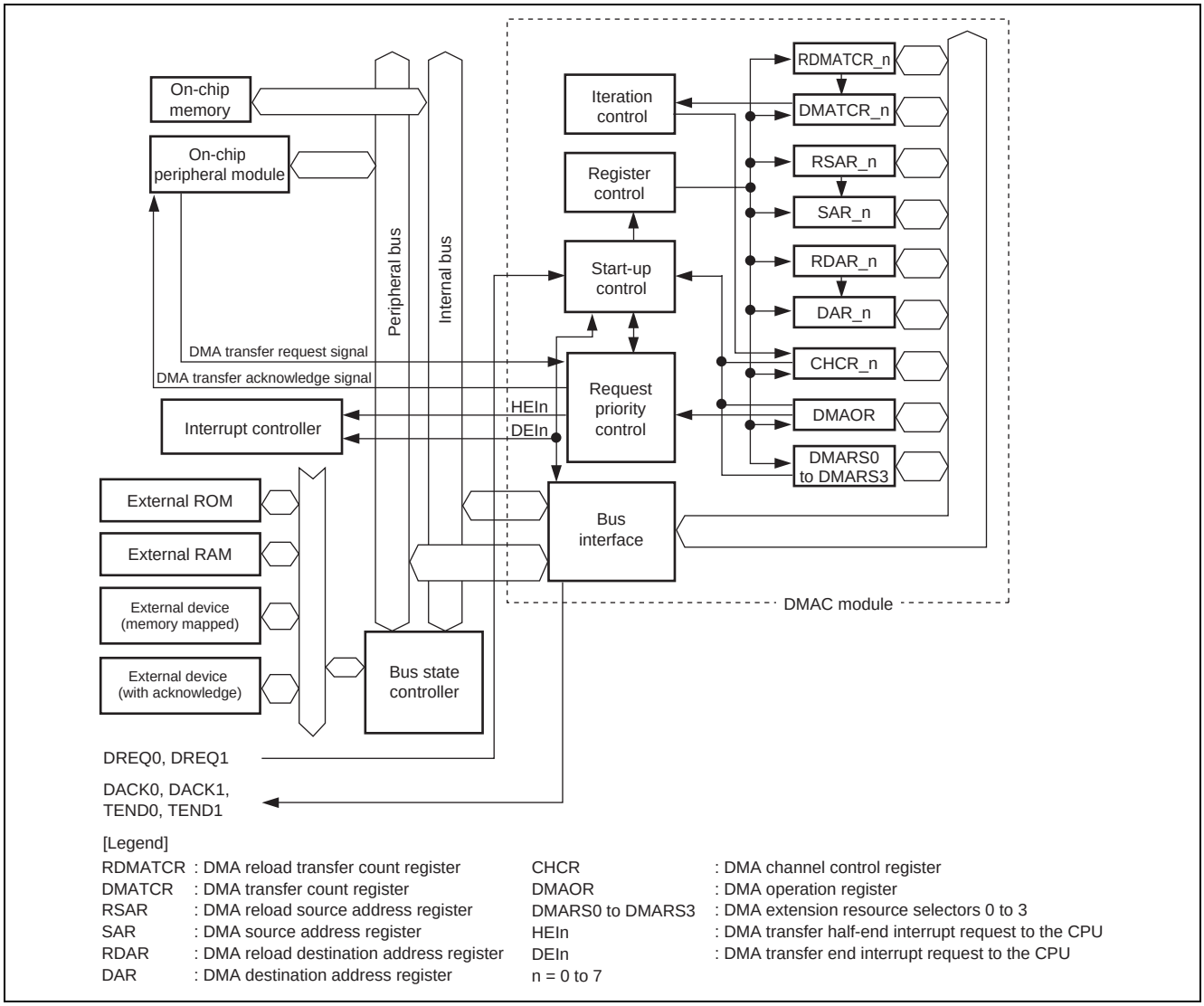
When a DMA transfer request is made, the DMAC starts to transfer data in order of priority of predetermined channels. Then, it continues the transfer operation until transfer end condition is met. It has three transfer request modes: auto request, external request, and on-chip peripheral module request. The bus mode is selectable from burst mode and cycle-stealing mode.

An overview of the DMAC is given in table 1. Also, a block diagram of the DMAC is shown in figure 1.

Table 1 Overview of DMAC

Item	Description
Number of channels	8 (CH0 to CH7) Only 2 channels (CH0 and CH1) can receive external requests.
Address space	4 Gbytes
Length of transfer data	Byte, word (2 bytes), longword (4 bytes), and 16 bytes (longword × 4)
Maximum number of unit transfers	16,777,216 (24 bits)
Address mode	Single address mode and dual address mode
Transfer request	Auto request, external request, and on-chip peripheral module request (SCIF: 6 sources, IIC3: 2 sources, CMT: 2 sources, USB: 2 sources, SSI: 2 sources)
Bus mode	Cycle-stealing mode and burst mode
Priority level	Channel priority fixed mode and round-robin mode
Interrupt request	An interrupt request to the CPU is made when half or all of a transfer process is completed.
External request detection	DREQ input low/high level detection, rising/falling edge detection
Transfer request acknowledge signal/transfer end signal	Active levels for DACK and TEND can be set independently

Note: For details on the DMAC, refer to the section on the direct memory access controller in the *SH7670 Group Hardware Manual* (REJ09B0437).



[Legend]

RDMATCR	: DMA reload transfer count register	CHCR	: DMA channel control register
DMATCR	: DMA transfer count register	DMAOR	: DMA operation register
RSAR	: DMA reload source address register	DMARS0 to DMARS3	: DMA extension resource selectors 0 to 3
SAR	: DMA source address register	HEIn	: DMA transfer half-end interrupt request to the CPU
RDAR	: DMA reload destination address register	DEIn	: DMA transfer end interrupt request to the CPU
DAR	: DMA destination address register	n = 0 to 7	

Figure 1 Block Diagram of DMAC

2.2 Procedure for Setting the Module Used

This section describes the procedure for specifying initial settings for transferring data between memory areas with the DMAC. Auto request mode is used for transfer requests. A flowchart of initializing the DMAC is shown in figure 2. For details on registers, refer to the *SH7670 Group Hardware Manual* (REJ09B0437).

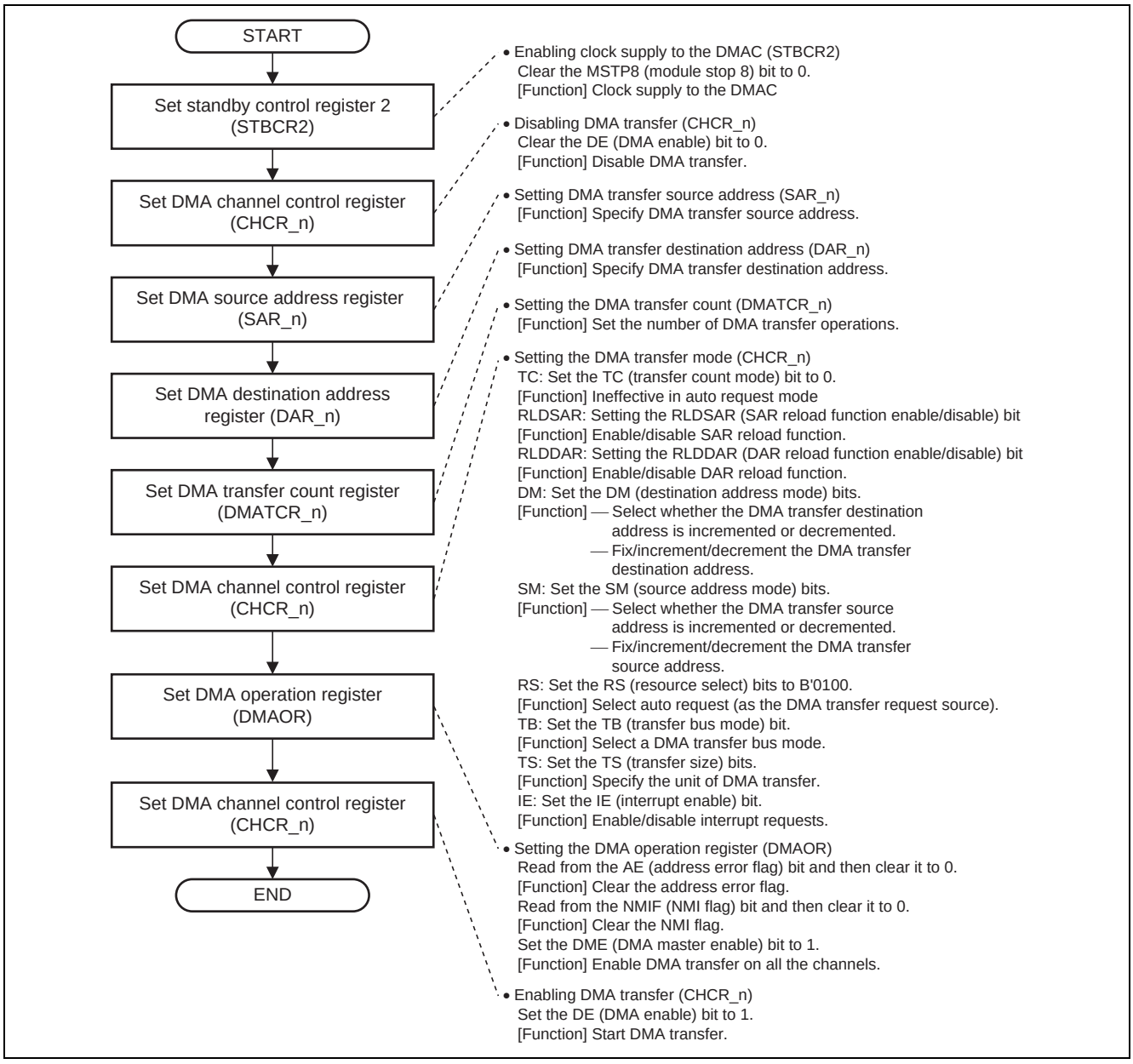


Figure 2 Example of Flow for Initialization of the DMAC

2.3 Operation of the Sample Program

In this sample program, DMAC channel 0 is activated by auto request, and data are transferred from the on-chip RAM to external memory in cycle-stealing mode. In cycle-stealing transfer operation, the DMAC gives the bus mastership to the CPU after each round of transferring a single unit of data. An operation timing of the sample application is shown in figure 3.

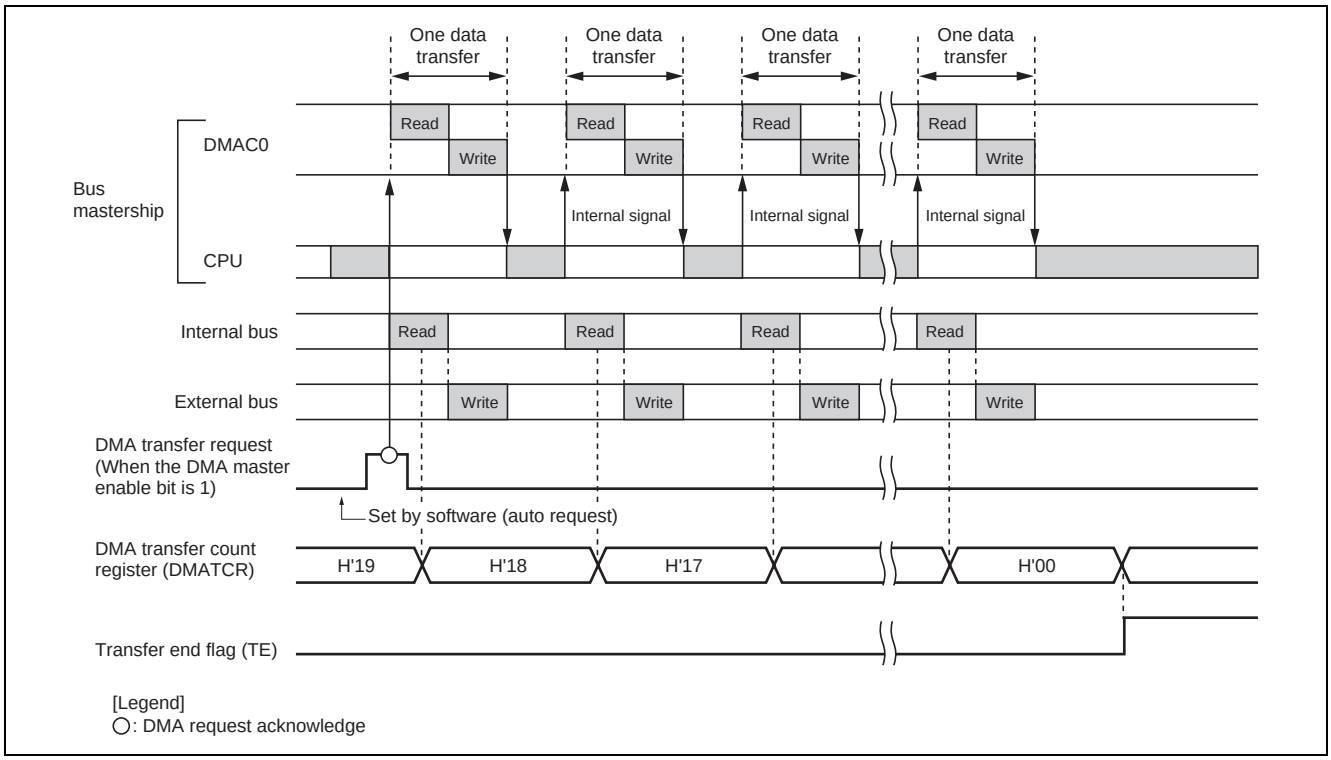


Figure 3 Operation Timing of Sample Application

2.4 Notes on Using the Sample Program

In the reference program, the addresses where the source and destination areas of the transfer start are assigned as absolute addresses for clarity. Ensure that sections used by the user program do not overlap with the source and destination regions that start from the absolute addresses.

2.5 Procedure for Processing by the Sample Program

In this sample program 100-byte data stored in the on-chip RAM are transferred to external memory by DMA transfer. The transfer end flag (TE bit) is used to check whether DMA transfer is completed.

The register settings for the sample program are listed in table 2. The macro definitions used in this sample program are also listed in table 3. A flowchart of the sample program is illustrated in figure 4.

Table 2 Register Settings for Sample Program

Register Name	Address	Setting	Description
Standby control register 2 (STBCR2)	H'FFFE 0018	H'00	MSTP8 = 0: DMAC operates
DMA channel control register_0 (CHCR_0)	H'FFFE 100C	H'0000 0000	DE = 0: Disables DMA transfer
		H'0000 5410	TC = 0: Ineffective in auto-request mode RLDSAR = 0: Disables the SAR reload function RLDDAR = 0: Disables the DAR reload function DM = B'01: Increments the destination address SM = B'01: Increments the source address RS = B'0100: Auto request TB = 0: Cycle-stealing mode TS = B'10: Longword transfer IE = 0: Disables interrupt request
		H'0000 5411	DE = 1: Enables DMA transfer
DMA source address register_0 (SAR_0)	H'FFFE 1000	H'FFF8 4000	Sets start address of transfer source in an on-chip RAM area
DMA destination address register_0 (DAR_0)	H'FFFE 1004	H'2C00 1000	Sets start address of transfer destination in an external memory area*
DMA transfer count register_0 (DMATCR_0)	H'FFFE 1008	H'64	Number of unit transfers: 100 (H'64)
DMA operation register (DMAOR)	H'FFFE 1200	H'0001	DME = 1: Enables DMA transfer on all the channels
DMA extension resource selector0 (DMARS0)	H'FFFE 1300	H'0000	Not used for auto request

Note: * Addresses in external memory area differ with the target board.

Table 3 Macro Definitions Used in Sample Program

Macro Definition	Setting	Description
SDRAM_DST_ADR	H'2C00 1000	Start address of SDRAM
SRAM_SRC_ADR	H'FFF8 4000	Start address of on-chip RAM
SIZE	H'64	Number of unit transfers
DMA_SIZE_BYTE	H'0000	Byte transfer
DMA_SIZE_WORD	H'0001	Word transfer
DMA_SIZE_LONG	H'0002	Longword transfer
DMA_SIZE_LONGx4	H'0003	16-byte transfer
DMA_INT_DISABLE	H'0000	DMA transfer end interrupt not in use
DMA_INT_ENABLE	H'0010	DMA transfer end interrupt in use

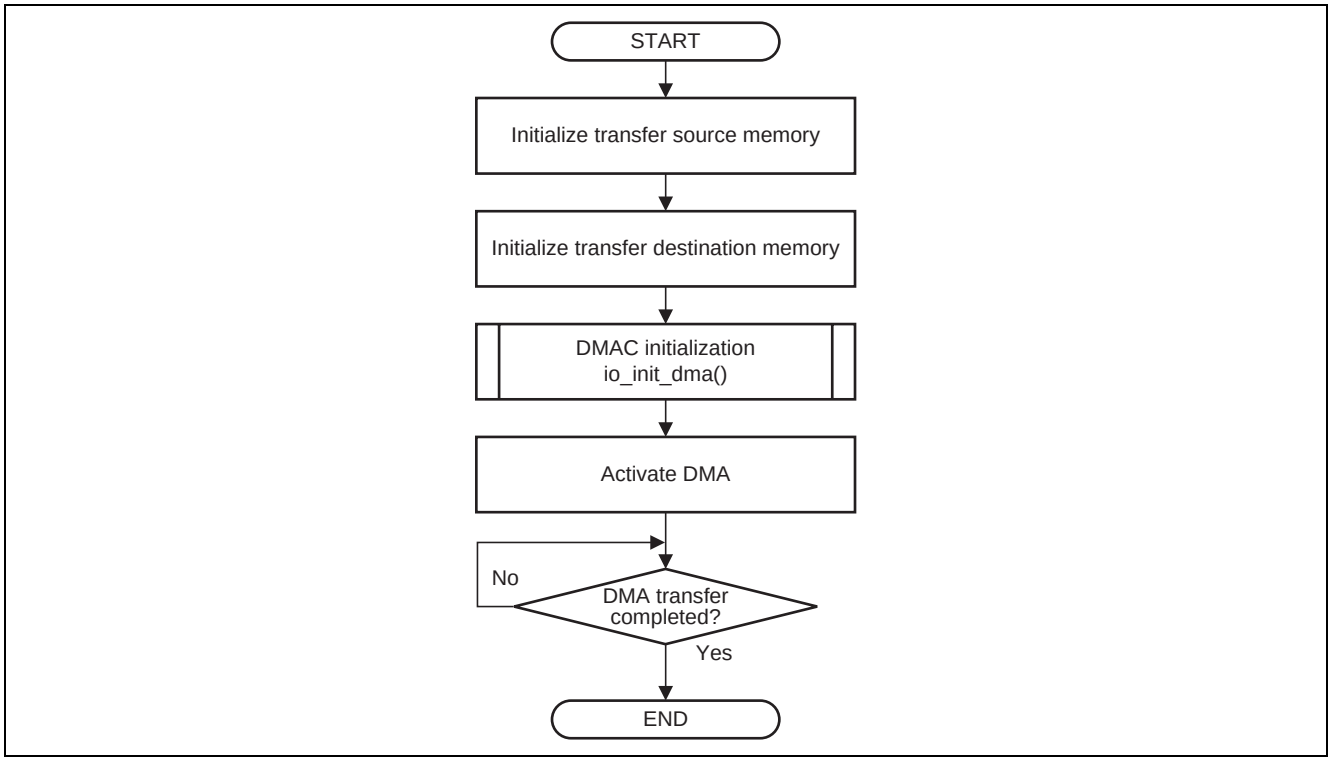


Figure 4 Flow of Processing by the Sample Program

3. Sample Program Listing

3.1 Sample program list "main.c" (1)

```
/***********************************************************************
*   DISCLAIMER
*
*   This software is supplied by Renesas Electronics Corporation and is only
*   intended for use with Renesas products. No other uses are authorized.
*
*   This software is owned by Renesas Electronics Corporation and is protected under
*   all applicable laws, including copyright laws.
*
*   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
*   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
*   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
*   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
*   DISCLAIMED.
*
*   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
*   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
*   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
*   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
*   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
*   Renesas reserves the right, without notice, to make changes to this
*   software and to discontinue the availability of this software.
*   By using this software, you agree to the additional terms and
*   conditions found by accessing the following link:
*   http://www.renesas.com/disclaimer
***********************************************************************
* Copyright (C) 2008(2010) Renesas Electronics Corporation. All rights reserved.
*""FILE COMMENT""*********** Technical reference data *************************
*   System Name : SH7671 Sample Program
*   File Name   : main.c
*   Abstract    : Sample program of DMAC
*   Version     : 1.00.01
*   Device      : SH7671
*   Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
*               : C/C++ compiler package for the SuperH RISC engine family
*               :                               (Ver.9.01 Release01).
*   OS          : None
*   H/W Platform: M3A-HS71(CPU board)
*   Description :
***********************************************************************
*   History     : Apr.24,2008 ver.1.00.00
*               : Oct.08,2010 ver.1.00.01 Changed the company name and device name
*""FILE COMMENT END""*****************************************************/
```

3.2 Sample program list "main.c" (2)

```
45    #include <machine.h>
46    #include <stdio.h>
47    #include "iodefine.h"                          /* SH7670 iodefine  */
48
49    /* ==== symbol definition ==== */
50    #define SDRAM_DST_ADR ((void *)0x2c001000)      /* External SDRAM address  */
51    #define SRAM_SRC_ADR  ((void *)0xfff84000)      /* Internal SRAM address   */
52    #define SIZE 100                                /* Transmission bytes  */
53    #define DMA_SIZE_BYTE   0x0000u
54    #define DMA_SIZE_WORD   0x0001u
55    #define DMA_SIZE_LONG   0x0002u
56    #define DMA_SIZE_LONGx4 0x0003u
57    #define DMA_INT_DISABLE 0x0000u
58    #define DMA_INT_ENABLE  0x0010u
59    #define DMA_INT (DMA_INT_ENABLE >> 4u)
60
61    /* ==== prototype declaration ==== */
62    void main(void);
63    void io_init_dma(void *src, void *dst, size_t size, unsigned int mode);
64    void io_dma_enable(void);
65    void io_dma_stop(void);
66
```

3.3 Sample program list "main.c" (3)

```
67    /*""FUNC COMMENT""*****************************************************************
68     * Outline      : Sample program main
69     *-------------------------------------------------------------------------------
70     * Include      : #include "iodefine.h"
71     *-------------------------------------------------------------------------------
72     * Declaration  : void main(void);
73     *-------------------------------------------------------------------------------
74     * Function     : Sample program main
75     *-------------------------------------------------------------------------------
76     * Argument     : void
77     *-------------------------------------------------------------------------------
78     * Return Value : none
79     *-------------------------------------------------------------------------------
80     * Notice       :
81     *""FUNC COMMENT END""*************************************************************/
82    void main(void)
83    {
84        int i;
85        volatile unsigned char *ptr;
86
87        /* ==== Initialize source memory ==== */
88        ptr = SRAM_SRC_ADR;
89        for(i=0; i < SIZE; i++){
90            *ptr++ = 0x55;
91        }
92
93        /* ==== Initialize destination memory ==== */
94        ptr = SDRAM_DST_ADR;
95        for(i=0; i < SIZE; i++){
96            *ptr++ = 0;
97        }
98
99        /* ==== Setting of DMAC ==== */
100       io_init_dma(SRAM_SRC_ADR, SDRAM_DST_ADR, SIZE, DMA_SIZE_LONG | DMA_INT_DISABLE);
101
102       /* ---- DMA start ---- */
103       io_dma_enable();
104
105       /* ---- DMA stop ---- */
106       io_dma_stop();
107
108       while(1){
109           /* Program end */
110       }
111   }
112
```

3.4 Sample program list "main.c" (4)

```
113    /*""FUNC COMMENT""*****************************************************************
114     * Outline      : Initialization for data transfer between memory devices by DMAC
115     *------------------------------------------------------------------------------
116     * Include      : #include "iodefine.h"
117     *------------------------------------------------------------------------------
118     * Declaration  : void io_init_dma(void *src, void *dst, size_t size, unsigned int mode);
119     *------------------------------------------------------------------------------
120     * Function     : The DMAC transfers the amount of data specified by "size"
121     *              : from the source address "src" to the destination address "dst".
122     *              : Auto request mode is used to transfer data.
123     *              : Transfer size and use or non-use of interrupts are specified
124     *              : for the "mode".
125     *------------------------------------------------------------------------------
126     * Argument     : void *src         ; Source address
127     *              : void *dst         ; Destination address
128     *              : size_t size       ; Size of data for transfer (byte)
129     *              : unsigned int mode ; Combos of the transfer and the following modes
130     *              :                     are obtained by logical OR.
131     *              :     DMA_SIZE_BYTE  (0x0000)  Transfer in untis of byte
132     *              :     DMA_SIZE_WORD  (0x0001)  Transfer in units of word
133     *              :     DMA_SIZE_LONG  (0x0002)  Transfer in units of longword
134     *              :     DMA_SIZE_LONGx4(0x0003)  16-byte transfer
135     *              :     DMA_INT_DISABLE(0x0000)  DMA transfer end interrupt is not in use.
136     *              :     DMA_INT_ENABLE (0x0010)  DMA transfer end interrupt is in use.
137     *------------------------------------------------------------------------------
138     * Return Value : none
139     *------------------------------------------------------------------------------
140     * Notice       : Operation is not guaranteed when the source/destination address is not
141     *              : on a boundary corresponding to the transfer size.
142     *              : If interrupts are to be used, the interrupt routines must be registered.
143     *""FUNC COMMENT END""*************************************************************/
144    void io_init_dma(void *src, void *dst, size_t size, unsigned int mode)
145    {
146        unsigned int ts;
147        unsigned long ie;
148
149        ts = mode & 0x3u;
150        ie = (mode & 0x00f0u) >> 4u;
151
152        /* ==== Setting of DMAC ==== */
153        /* ==== Setting of power down mode ==== */
154        CPG.STBCR2.BIT.MSTP8 = 0x0u;        /* Clear the DMAC module standby mode */
155
156        /* ---- DMA Channel Control Registers (CHCR) ---- */
157        DMAC.CHCR0.BIT.DE = 0ul;            /* DMA disable */
158
159        /* ---- DMA Source Address Registers (SAR) ---- */
160        DMAC.SAR0 = (unsigned long)src;
161
162        /* ---- DMA Destination Address Registers (DAR) ---- */
163        DMAC.DAR0 = (unsigned long)dst;
164
```

3.5 Sample program list "main.c" (5)

```
165         /* ---- DMA Transfer Count Registers (DMATCR) ---- */
166         switch(ts){
167         case DMA_SIZE_BYTE:
168             DMAC.DMATCR0 = size;            /* Specify number of unit transfers (1/1)  */
169             DMAC.RDMATCR0 = size;
170             break;
171         case DMA_SIZE_WORD:
172             DMAC.DMATCR0 = size >> 1u;      /* Specify number of unit transfers (1/2)  */
173             DMAC.RDMATCR0 = size >> 1u;
174             break;
175         case DMA_SIZE_LONG:
176             DMAC.DMATCR0 = size >> 2u;      /* Specify number of unit transfers (1/4)  */
177             DMAC.RDMATCR0 = size >> 2u;
178             break;
179         case DMA_SIZE_LONGx4:
180             DMAC.DMATCR0 = size >> 4u;      /* Specify number of unit transfers (1/16) */
181             DMAC.RDMATCR0 = size >> 4u;
182             break;
183         default:
184             break;
185         }
186
187         /* ---- DMA Channel Control Registers (CHCR) ---- */
188         DMAC.CHCR0.LONG = 0x00005400ul | (ts << 3u) | (ie << 2u);
189                                             /* Destination address is incremented  */
190                                             /* Source address is incremented       */
191                                             /* Auto request                        */
192                                             /* Cycle-stealing mode                 */
193                                             /* Transfer size : Longword unit       */
194
195         /* ---- DMA Operation Register (DMAOR) ---- */
196         DMAC.DMAOR.WORD &= 0xfff9u;         /* AE,NMIF clear */
197
198         if(DMAC.DMAOR.BIT.DME == 0ul){      /* DMA Master Enable */
199             DMAC.DMAOR.BIT.DME = 1ul;
200         }
201     }
202
```

3.6 Sample program list "main.c" (6)

```
203    /*""FUNC COMMENT""*****************************************************************
204     * Outline      : Activation of DMAC
205     *-------------------------------------------------------------------------------
206     * Include      : #include "iodefine.h"
207     *-------------------------------------------------------------------------------
208     * Declaration  : void io_dma_enable(void);
209     *-------------------------------------------------------------------------------
210     * Function     : Performing DMA transfer
211     *-------------------------------------------------------------------------------
212     * Argument     : void
213     *-------------------------------------------------------------------------------
214     * Return Value : none
215     *-------------------------------------------------------------------------------
216     * Notice       :
217     *""FUNC COMMENT END""*************************************************************/
218    void io_dma_enable(void)
219    {
220        /* ---- DMA start ---- */
221        DMAC.CHCR0.BIT.DE = 1ul;            /* DMA enable */
222    }
223
224    /*""FUNC COMMENT""*****************************************************************
225     * Outline      : Halt of DMAC
226     *-------------------------------------------------------------------------------
227     * Include      : #include "iodefine.h"
228     *-------------------------------------------------------------------------------
229     * Declaration  : void io_dma_stop(void);
230     *-------------------------------------------------------------------------------
231     * Function     : Checking whether the transfer is completed and stopping the DMA transfer
232     *-------------------------------------------------------------------------------
233     * Argument     : void
234     *-------------------------------------------------------------------------------
235     * Return Value : none
236     *-------------------------------------------------------------------------------
237     * Notice       :
238     *""FUNC COMMENT END""*************************************************************/
239    void io_dma_stop(void)
240    {
241        /* Transmission end detection */
242        while(DMAC.CHCR0.BIT.TE == 0ul){
243            /* wait TE bit set */
244        }
245        /* ---- DMA stop ---- */
246        DMAC.CHCR0.BIT.DE = 0ul;            /* DMA disable */
247    }
248
249    /* End of File */
```

4. References

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

- Hardware Manual
  SH7670 Group Hardware Manual Rev. 2.00
  The latest version of the hardware user's manual can be downloaded from the Renesas Electronics website.

Website and Support

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Nov.19.08	—	First edition issued
1.01	Oct.15.10	—	Changed the sample program (AC Switching Characteristics are removed)

General Precautions in the Handling of MPU/MCU Products

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

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.0