SH7670 Group

R01AN0306EJ0101
Rev. 1.01
Oct. 15, 2010

Using the DMAC to Transfer Data to On-chip Peripheral Modules

Summary

This application note provides an example of transferring data to on-chip peripheral modules with the direct memory access controller (DMAC) of the SH7670.

Target Device

SH7670 MCU

Contents

1. Introduction

1.1 Specifications

- DMAC channel 1 is used to transfer data from external memory to the transmit FIFO data register (SCFTDR) in the serial communication interface with FIFO (SCIF channel 0) in order to transmit character string data.
- SCIF transmit-FIFO-data-empty transfer requests (on-chip peripheral module request) are used to request DMA transfer.

1.2 Module Used

- Direct memory access controller (DMAC channel 1)
- Serial communication interface with FIFO (SCIF 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 between Memory Areas

2. Description of the Sample Application

In this sample application, the DMAC and on-chip peripheral module requests are used to transfer data from external memory to the SCIF.

2.1 Operational Overview of Modules 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).

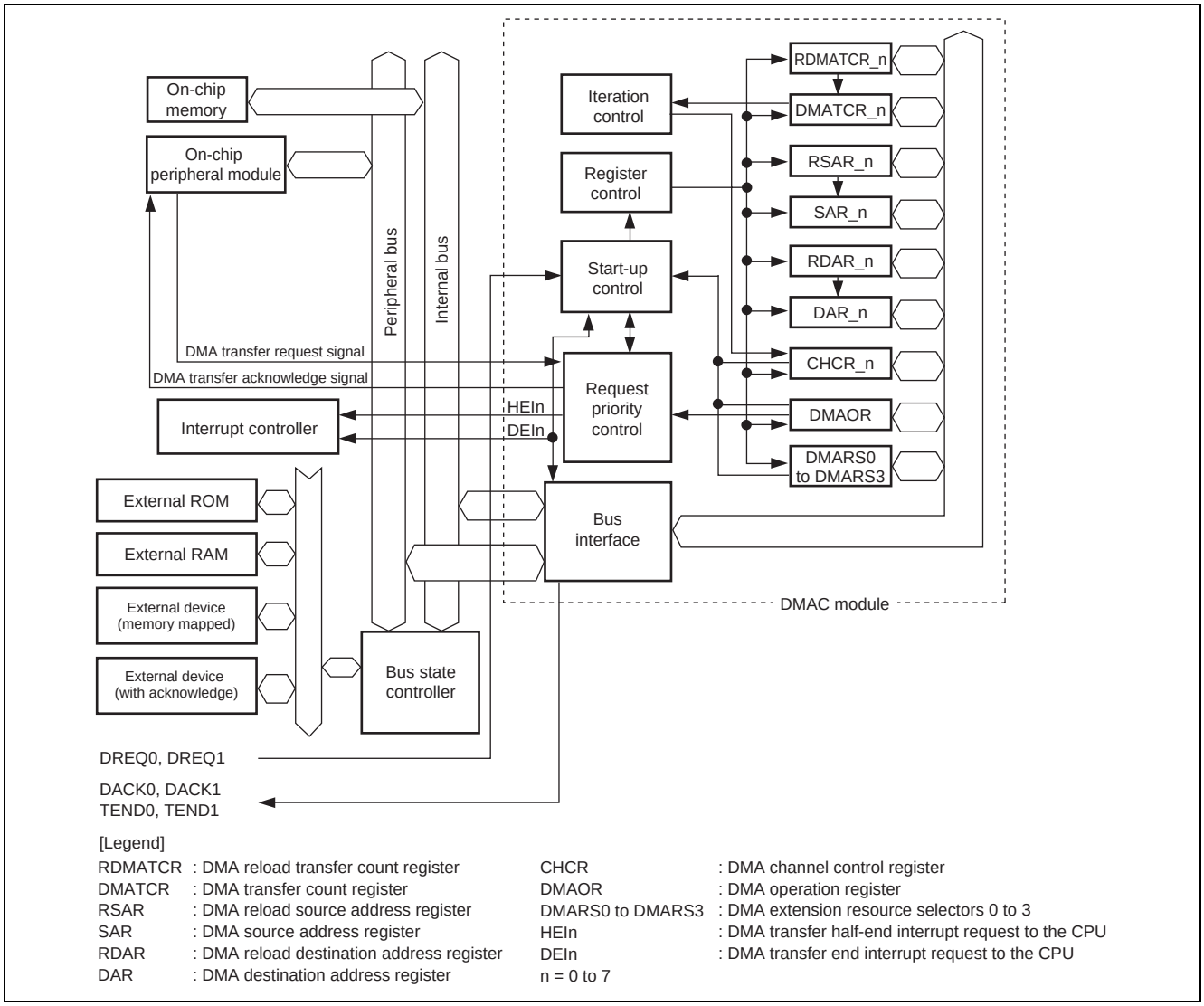

Figure 1 Block Diagram of DMAC

2.2 Procedure for Setting the Modules Used

This section describes the procedure for making initial settings when the DMAC is to be used to transfer data from memory to on-chip peripheral modules. On-chip peripheral module requests are 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).

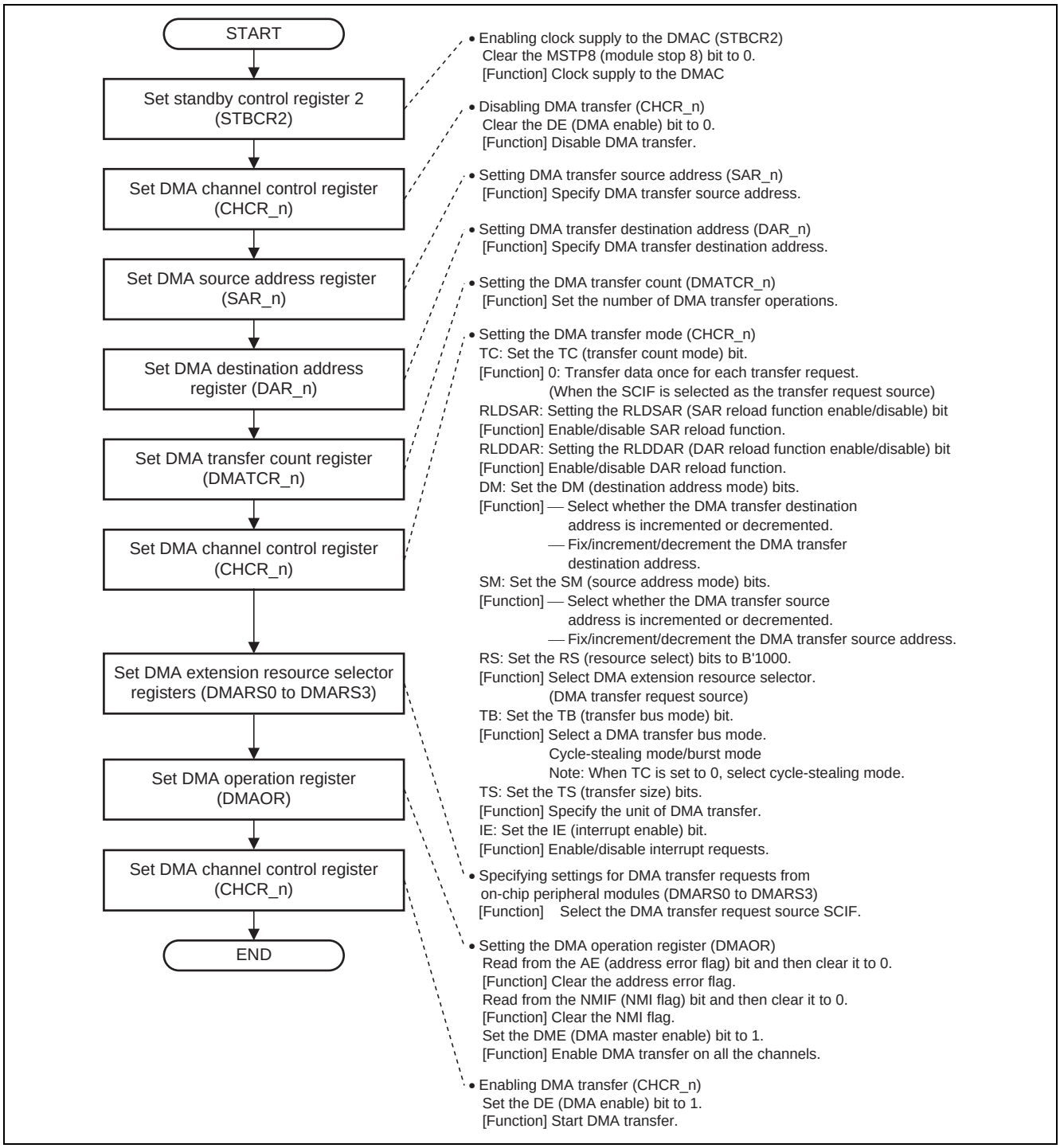

Figure 2 Example of Flow for Initialization of the DMAC

2.3 Operation of the Sample Program

In this sample program, SCIF transmit-FIFO-data-empty transfer requests are made to activate DMAC channel 1, and to transfer data from external memory to the transmit FIFO data register (SCFTDR) on SCIF channel 0. The data written to SCFTDR on SCIF channel 0 are transmitted in UART mode. An operation timing of the sample program is shown in figure 3.

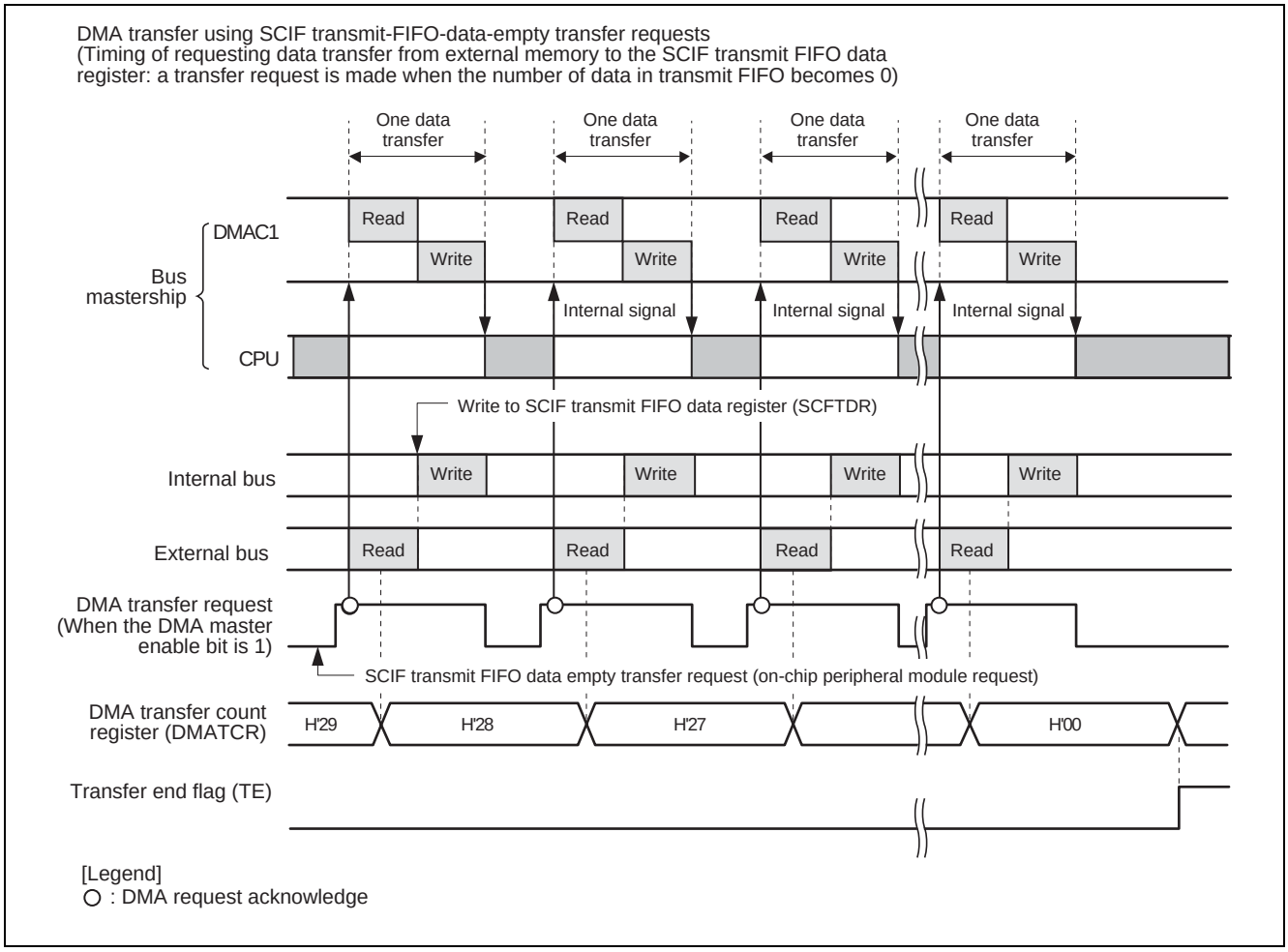

Figure 3 Operation Timing of Sample Application

2.4 Procedure for Processing by the Sample Program

In this sample program, character string data stored in external memory are transferred by DMA to the transmit FIFO data register (SCFTDR) on SCIF channel 0, and then are transmitted in UART mode.

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_1 (CHCR_1)	H'FFFE 101C	H'0000 0000	DE = 0: Disables DMA transfer
		H'0000 1800	TC = 0: Transfers data once for each DMA transfer request RLDSAR = 0: Disables SAR reload function RLDDAR = 0: Disables DAR reload function DM = B'00: Fixes destination address SM = B'01: Increments source address RS = B'1000: Extension resource selector TB = 0: Cycle-stealing mode TS = B'00: Byte transfer IE = 0: Disables interrupt request
		H'0000 1801	DE = 1: Enables DMA transfer
DMA source address register_1 (SAR_1)	H'FFFE 1010	Address where character string data are stored	Start address of transfer source: Start address of character string stored in external memory
DMA destination address register_1 (DAR_1)	H'FFFE 1014	H'FFFE 800C	Start address of transfer destination: Address of the SCIF transmit FIFO data register_0 (SCFTDR_0)
DMA transfer count register_1 (DMATCR_1)	H'FFFE 1018	Number of character string data	Number of unit transfers: the number of character string data
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'8100	MID = B'100000 RID = B'01 Set to SCIF_0 transmit-FIFO-data-empty transfer request

Table 3 Macro Definitions Used in Sample Program

Macro Definition	Setting	Description
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

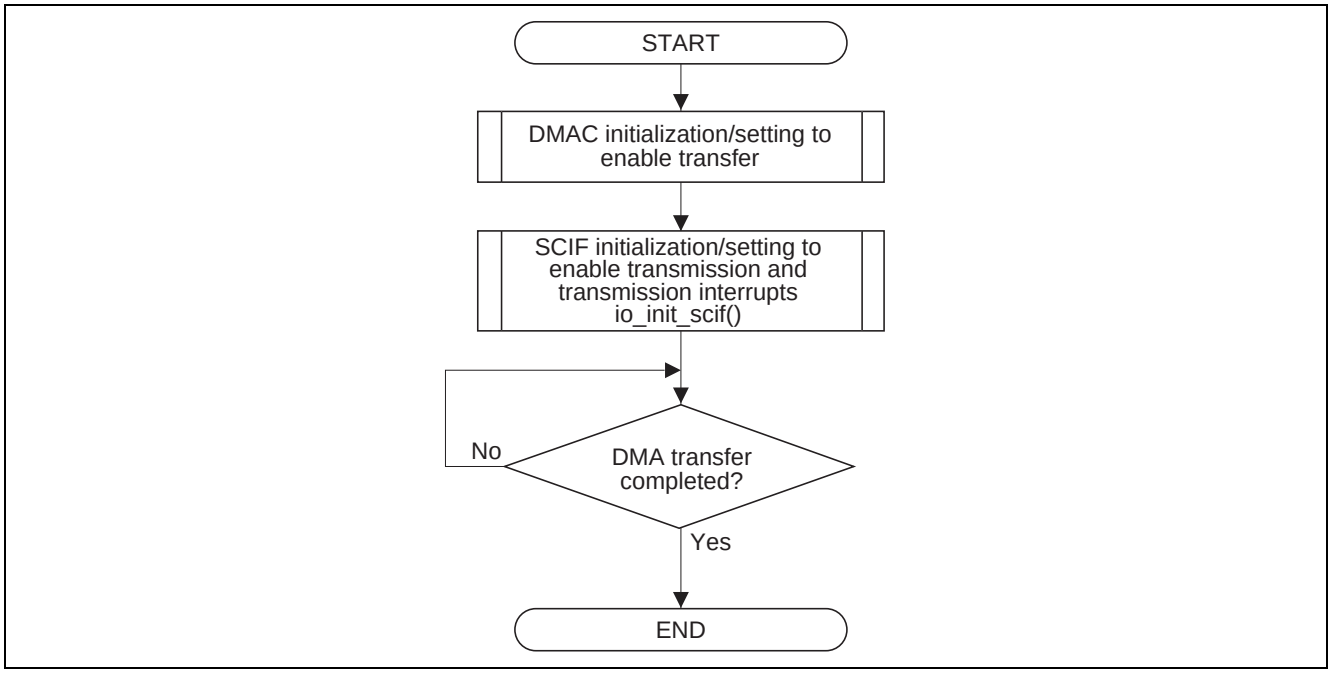

Figure 4 Flow of Processing by the Sample Program

3. Sample Program Listing

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

```
1   /***********************************************************************
2   *   DISCLAIMER
3   *
4   *   This software is supplied by Renesas Electronics Corporation and is only
5   *   intended for use with Renesas products. No other uses are authorized.
6   *
7   *   This software is owned by Renesas Electronics Corporation and is protected under
8   *   all applicable laws, including copyright laws.
9   *
10  *   THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
11  *   REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
12  *   INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
13  *   PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
14  *   DISCLAIMED.
15  *
16  *   TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
17  *   ELECTRONICS CORPORATION NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
18  *   FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
19  *   FOR ANY REASON RELATED TO THIS SOFTWARE, EVEN IF RENESAS OR ITS
20  *   AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
21  *
22  *   Renesas reserves the right, without notice, to make changes to this
23  *   software and to discontinue the availability of this software.
24  *   By using this software, you agree to the additional terms and
25  *   conditions found by accessing the following link:
26  *   http://www.renesas.com/disclaimer
27  ***********************************************************************
28  * Copyright (C) 2008(2010) Renesas Electronics Corporation. All rights reserved.
29  *""FILE COMMENT""*********** Technical reference data *************************
30  *   System Name : SH7671 Sample Program
31  *   File Name   : main.c
32  *   Abstract    : Data transfer to on-chip modules with DMAC
33  *   Version     : 1.00.01
34  *   Device      : SH7671
35  *   Tool-Chain  : High-performance Embedded Workshop (Ver.4.03.00).
36  *               : C/C++ compiler package for the SuperH RISC engine family
37  *               :                               (Ver.9.01 Release01).
38  *   OS          : None
39  *   H/W Platform: M3A-HS71(CPU board)
40  *   Description :
41  ***********************************************************************
42  *   History     : Apr.24,2008 ver.1.00.00
43  *               : Oct.08,2010 ver.1.00.01 Changed the company name and device name
44  *""FILE COMMENT END""*****************************************************/
45  #include <string.h>
46  #include "iodefine.h"       /* SH7670 iodefine  */
47
```

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

```
48    /* ==== symbol definition ==== */
49    /* ==== DMAC setting ==== */
50    #define DMA_SIZE_BYTE   0x0000u
51    #define DMA_SIZE_WORD   0x0001u
52    #define DMA_SIZE_LONG   0x0002u
53    #define DMA_SIZE_LONGx4 0x0003u
54    #define DMA_INT_DISABLE 0x0000u
55    #define DMA_INT_ENABLE  0x0010u
56    #define DMA_INT (DMA_INT_ENABLE >> 4u)
57
58    /* ==== prototype declaration ==== */
59    void main(void);
60    void io_init_dma1(void *src, void *dst, size_t size, unsigned int mode);
61    void io_dma1_stop(void);
62    void io_init_scif(int bps);
63
64    /* ==== RAM allocation variable declaration ==== */
65    typedef struct {
66        unsigned char scbrr;
67        unsigned short scsmr;
68    } SH7670_BAUD_SET;
69
70    /* ---- baud rate ---- */
71    enum{
72        CBR_1200,
73        CBR_2400,
74        CBR_4800,
75        CBR_9600,
76        CBR_19200,
77        CBR_31250,
78        CBR_38400,
79        CBR_57600,
80        CBR_115200
81    };
82
83    static SH7670_BAUD_SET scif_baud[] = {
84        {216u, 1u},             /*   1200 bps (-0.003%) */
85        {107u, 1u},             /*   2400 bps ( 0.459%) */
86        {216u, 0u},             /*   4800 bps (-0.003%) */
87        {107u, 0u},             /*   9600 bps ( 0.459%) */
88        { 53u, 0u},             /*  19200 bps ( 0.459%) */
89        { 32u, 0u},             /*  31250 bps ( 1.00%)  */
90        { 26u, 0u},             /*  38400 bps ( 0.459%) */
91        { 17u, 0u},             /*  57600 bps ( 0.459%) */
92        {  8u, 0u}              /* 115200 bps ( 0.459%) */
93    };
94
95    /* Transmission character string */
96    signed char data[] = "SCIF request DMAC Sample Software.\r\n";
97
```

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

```
98     /*""FUNC COMMENT""*****************************************************************
99      * Outline      : Sample program main
100     *-------------------------------------------------------------------------------
101     * Include      : #include "iodefine.h"
102     *-------------------------------------------------------------------------------
103     * Declaration  : void main(void);
104     *-------------------------------------------------------------------------------
105     * Function     : Sample program main
106     *-------------------------------------------------------------------------------
107     * Argument     : void
108     *-------------------------------------------------------------------------------
109     * Return Value : none
110     *-------------------------------------------------------------------------------
111     * Notice       :
112     *""FUNC COMMENT END""*************************************************************/
113    void main(void)
114    {
115        /* ==== Setting of DMAC ==== */
116        io_init_dma1(data, (void *)&SCIF0.SCFTDR, sizeof(data),
117                        DMA_SIZE_BYTE | DMA_INT_DISABLE);
118                                            /* Transfer requests : SCIF0 transmitter */
119                                            /* RAM -> SCIF transmitter               */
120        /* ==== Setting of SCIF ==== */
121        io_init_scif(CBR_115200);
122                                            /* UART mode             */
123                                            /* bit rate : 115200 bps */
124        /* ==== DMA start ==== */
125        DMAC.CHCR1.BIT.DE = 1ul;            /* DMA enable */
126
127        /* ==== DMA stop ==== */
128        io_dma1_stop();
129
130        while(1){
131            /* Program end */
132        }
133    }
134
```

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

```
135    /*""FUNC COMMENT""*****************************************************************
136     * Outline      : Initialization for data transfer between memory areas with DMAC
137     *-------------------------------------------------------------------------------
138     * Include      : #include "iodefine.h"
139     *-------------------------------------------------------------------------------
140     * Declaration  : void io_init_dma1(void *src, void *dst, size_t size, unsigned int mode);
141     *-------------------------------------------------------------------------------
142     * Function     : The DMAC transfers the amount of data specified by "size"
143     *              : from the source address "src" to the destination address "dst".
144     *              : Transfer is performed using requests from the SCIF0.
145     *              : Transfer size and use or non-use of interrupts are specified for "mode".
146     *-------------------------------------------------------------------------------
147     * Argument     : void *src         ; Source address
148     *              : void *dst         ; Destination address
149     *              : size_t size       ; Tnrasfer size (byte)
150     *              : unsigned int mode ; Combos of the transfer and the following modes
151     *              :                     are obtained by logical OR.
152     *              :     DMA_SIZE_BYTE  (0x0000)  Byte transfer
153     *              :     DMA_SIZE_WORD  (0x0001)  Word transfer
154     *              :     DMA_SIZE_LONG  (0x0002)  Longword transfer
155     *              :     DMA_SIZE_LONGx4(0x0003)  16-byte transfer
156     *              :     DMA_INT_DISABLE(0x0000)  DMA transfer end interrupt not in use
157     *              :     DMA_INT_ENABLE (0x0010)  DMA transfer end interrupt in use
158     *-------------------------------------------------------------------------------
159     * Return Value : none
160     *-------------------------------------------------------------------------------
161     * Notice       : Operation is not guaranteed when the source/destination address is not
162     *              : on a boundary corresponding to the transfer size.
163     *              : If interrupts are to be used, the interrupt routines must be registered.
164     *""FUNC COMMENT END""*************************************************************/
165    void io_init_dma1(void *src, void *dst, size_t size, unsigned int mode)
166    {
167        unsigned int ts;
168        unsigned long ie;
169
170        ts = mode & 0x3u;
171        ie = (mode & 0x00f0u) >> 4u;
172
173        /* ==== Setting of power down mode ==== */
174        CPG.STBCR2.BIT.MSTP8 = 0x0u;        /* Clear the DMAC module standby mode */
175
176        /* ---- DMA Channel Control Registers (CHCR) ---- */
177        DMAC.CHCR1.BIT.DE = 0ul;            /* DMA disable */
178
179        /* ---- DMA Source Address Registers (SAR) ---- */
180        DMAC.SAR1 = (unsigned long)src;
181
182        /* ---- DMA Destination Address Registers (DAR) ---- */
183        DMAC.DAR1 = (unsigned long)dst;
184
```

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

```
185         /* ---- DMA Transfer Count Registers (DMATCR) ---- */
186         switch(ts){
187         case DMA_SIZE_BYTE:
188             DMAC.DMATCR1 = size;            /* Specify the number of unit transfers (1/1)  */
189             DMAC.RDMATCR1 = size;
190             break;
191         case DMA_SIZE_WORD:
192             DMAC.DMATCR1 = size >> 1u;      /* Specify the number of unit transfers (1/2)  */
193             DMAC.RDMATCR1 = size >> 1u;
194             break;
195         case DMA_SIZE_LONG:
196             DMAC.DMATCR1 = size >> 2u;      /* Specify the number of unit transfers (1/4)  */
197             DMAC.RDMATCR1 = size >> 2u;
198             break;
199         case DMA_SIZE_LONGx4:
200             DMAC.DMATCR1 = size >> 4u;      /* Specify the number of unit transfers (1/16) */
201             DMAC.RDMATCR1 = size >> 4u;
202             break;
203         default:
204             break;
205         }
206         /* ---- DMA Channel Control Registers (CHCR) ---- */
207         DMAC.CHCR1.LONG = 0x00001800ul | (ts << 3u) | (ie << 2u);
208                                             /* Fixed destination address       */
209                                             /* Source address is incremented   */
210                                             /* DMA extension resource selector */
211                                             /* Cycle steal mode                */
212                                             /* Transfer Size : Byte unit       */
213
214         /* ---- DMA Extension Resource Selectors 0 (DMARS0) ---- */
215         DMAC.DMARS0.BIT.CH1MID = 0x20u;     /* Transfer requests : SCIF0 transmitter */
216         DMAC.DMARS0.BIT.CH1RID = 0x1u;
217
218         /* ---- DMA Operation Register (DMAOR) ---- */
219         DMAC.DMAOR.WORD &= 0xfff9u;         /* AE,NMIF clear */
220
221         if(DMAC.DMAOR.BIT.DME == 0ul){      /* DMA Master Enable */
222             DMAC.DMAOR.BIT.DME = 1ul;
223         }
224
225     }
226
```

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

```
227    /*""FUNC COMMENT""*****************************************************************
228     * Outline      : DMAC stop
229     *-------------------------------------------------------------------------------
230     * Include      : #include "iodefine.h"
231     *-------------------------------------------------------------------------------
232     * Declaration  : void io_dma1_stop(void);
233     *-------------------------------------------------------------------------------
234     * Function     : Stopping the DMA transfer
235     *-------------------------------------------------------------------------------
236     * Argument     : void
237     *-------------------------------------------------------------------------------
238     * Return Value : none
239     *-------------------------------------------------------------------------------
240     * Notice       :
241     *""FUNC COMMENT END""*************************************************************/
242    void io_dma1_stop(void)
243    {
244        /* Transmission end detection */
245        while(DMAC.CHCR1.BIT.TE == 0ul){
246            /* wait TE bit set */
247        }
248        /* ---- DMA end ---- */
249        DMAC.CHCR1.BIT.DE = 0ul;            /* DMA disable */
250    }
251
```

3.7 Sample program list "main.c" (7)

```
252    /*""FUNC COMMENT""*****************************************************************
253     * Outline      : SCIF setting
254     *-------------------------------------------------------------------------------
255     * Include      : #include "iodefine.h"
256     *-------------------------------------------------------------------------------
257     * Declaration  : void io_init_scif(int bps);
258     *-------------------------------------------------------------------------------
259     * Function     : Setting the serial communication interface with FIFO (SCIF)
260     *-------------------------------------------------------------------------------
261     * Argument     : int bps ; value for baud rate specification
262     *-------------------------------------------------------------------------------
263     * Return Value : none
264     *-------------------------------------------------------------------------------
265     * Notice       :
266     *""FUNC COMMENT END""*************************************************************/
267    void io_init_scif(int bps)
268    {
269        /* ==== Setting of power down mode ==== */
270        CPG.STBCR3.BIT.MSTP30 = 0u;         /* Clear the SCIF0 module standby mode */
271
272        /* ==== Setting of SCIF ==== */
273        /* ---- Serial Control Register (SCSCR) ---- */
274        SCIF0.SCSCR.WORD &= 0x00u;          /* Transmitter/Receiver disabled */
275        SCIF0.SCSCR.BIT.CKE = 0x0u;         /* Internal clock */
276
277        /* ---- Serial Mode Register (SCSMR) ---- */
278        SCIF0.SCSMR.WORD = scif_baud[bps].scsmr;
279                                            /* Asynchronous mode               */
280                                            /* 8-bit data                      */
281                                            /* Parity bit not added or checked */
282                                            /* One stop bit                    */
283
284        /* ---- Bit Rate Register (SCBRR) ---- */
285        SCIF0.SCBRR.BYTE = scif_baud[bps].scbrr;
286
287        /* ==== Setting of PFC ==== */
288        /* ---- port F control register L1 ---- */
289        PORT.PFCRL1.BIT.PF1MD = 2u;         /* Set TxD0 */
290
291        /* ---- Serial Control Register (SCSCR) ---- */
292        SCIF0.SCSCR.BIT.TIE = 1u;           /* Transmit interrupt enabled */
293        SCIF0.SCSCR.BIT.TE = 1u;            /* Transmitter enabled */
294    }
295
296    /* End of File */
297
```

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