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.

SH7280 Group

Data Transfer to On-chip Peripheral Modules with DMAC

Introduction

This application note provides an example of transferring data to on-chip peripheral modules with the direct memory access controller (DMAC) of the SH7285.

Target Device

SH7285

Contents

1. Introduction

1.1 Specification

- DMAC channel 1 is used to transfer data from external memory to the transmit FIFO data register (SCFTDR) in the serial communications interface with FIFO (SCIF channel 3) 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 Modules Used

- Direct memory access controller (DMAC channel 1)
- Serial communications interface with FIFO (SCIF channel 3)

1.3 Applicable Conditions

• MCU:	SH7285/SH7286/SH7243	
• Operating frequency:	Internal clock	100 MHz
	Bus clock	50 MHz
	Peripheral clock	50 MHz
• C compiler:	SuperH RISC Engine Family C/C++ Compiler Package Ver.9.01 Release01 from Renesas Technology	
• Compiler options:	-cpu = sh2a -include = "$(WORKSPDIR)\inc" -object = "$(CONFIGDIR)\$(FILELEAF).obj" -debug -gbr= auto -chgincpath -errorpath -global_volatile = 0 -opt_range = all -infinite_loop = 0-del_vacant_loop = 0 -struct_alloc = 1 -nologo	

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.

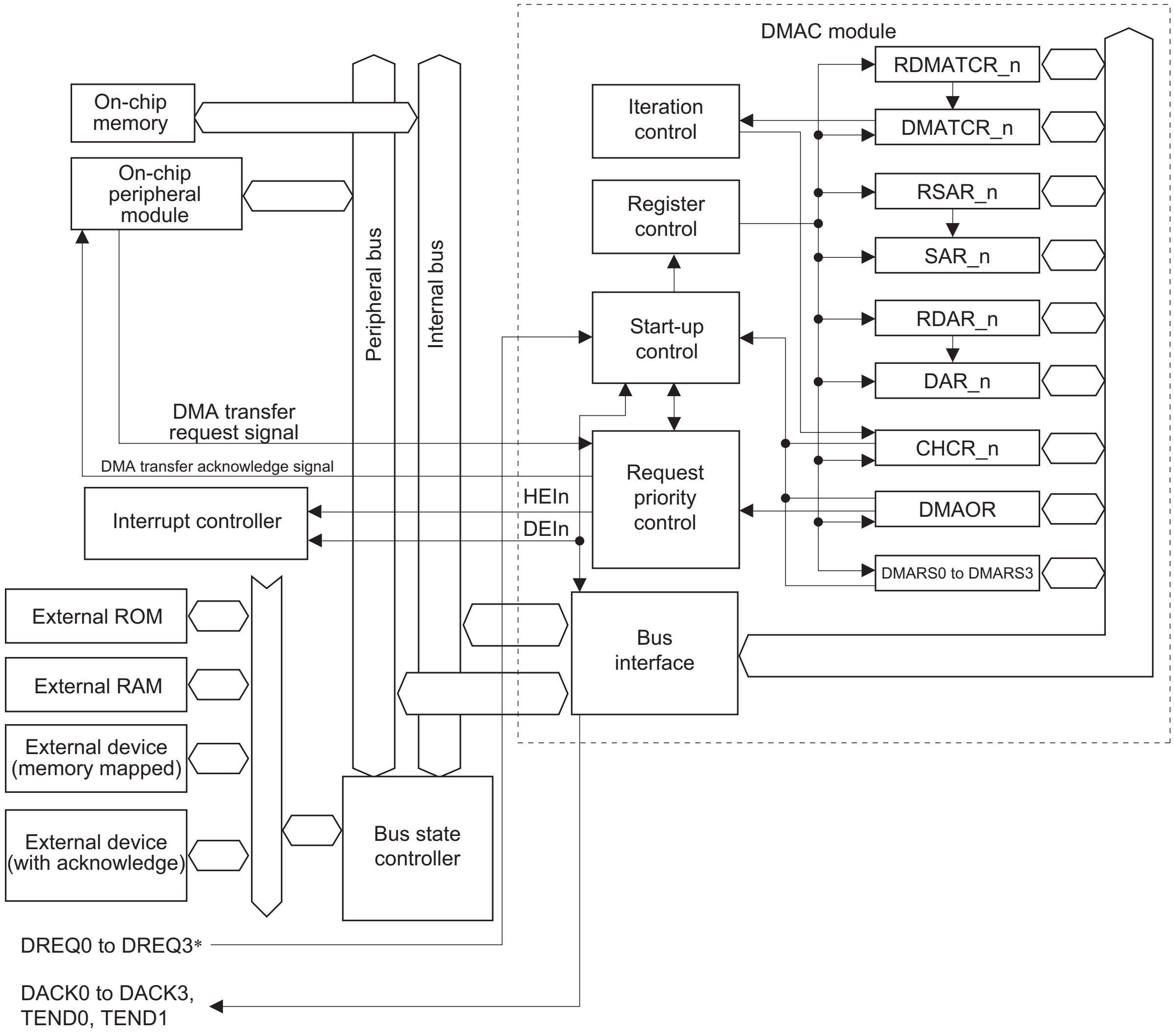
2.1 Summary of MCU Module Used

When a DMA transfer request is made, the DMAC starts to transfer data in order of priority of predetermined channels, and continues the transfer operation until the transfer end condition is met. It has three transfer request modes: auto requests, external requests, and on-chip peripheral module requests. The bus mode is selectable from burst mode or 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 channels (CH0 to CH7) Only 3 channels (CH0 to CH2) 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 transfer count	16,777,216 (24 bits) transfers
Address mode	Single address mode and dual address mode
Transfer request	Auto request, external request, and on-chip peripheral module request (SCIF: 2 sources, IIC3: 2 sources, A/D converter: 1 source, MTU2: 5 sources, CMT: 2 sources, USB: 2 sources, SSU: 2 sources, RCAN: 1 source)
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 SH7280 Group Hardware Manual.



[Legend]

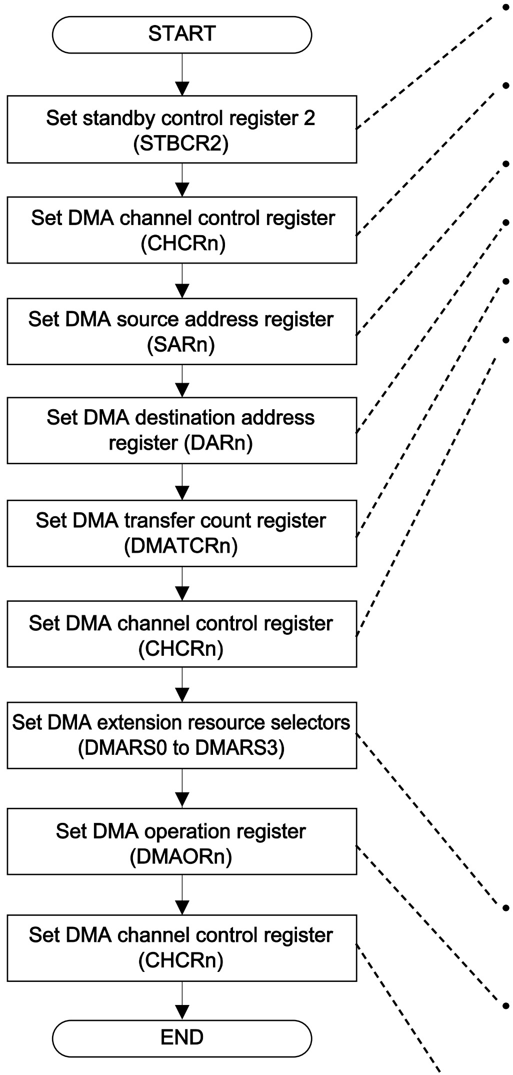
RDMATCR:	DMA reload transfer count register	CHCR:	DMA channel control register
DMATCR:	DMA transfer count register	DMAOR:	DMA operation register
RSAR:	DMA reload source address register	DMARS0 to DMARS3:	DMA extension resource selectors 0 to 3
SAR:	DMA source address register	HEIn:	DMA transfer half-end 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, 1, 2, 3, 4, 5, 6, 7

Notes: * DREQ0 and DREQ1 (SH7285 and SH7243)
DREQ0 to DREQ3 (SH7286)

Figure 1 Block Diagram of DMAC

2.2 Procedure for Setting the Module 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 DMAC initialization is shown in figure 2. For details on registers, refer to the SH7280 Group Hardware Manual.


START
Set standby control register 2 (STBCR2)
Set DMA channel control register (CHCRn)
Set DMA source address register (SARn)
Set DMA destination address register (DARn)
Set DMA transfer count register (DMATCRn)
Set DMA channel control register (CHCRn)
Set DMA extension resource selectors (DMARS0 to DMARS3)
Set DMA operation register (DMAORn)
Set DMA channel control register (CHCRn)
END

- Enabling clock supply to the DMAC (STBCR2)
  Clear the MSTP8 (module stop 8) to 0
  [Function] Clock supply to the DMAC
- Disabling DMA transfer (CHCRn)
  Clear the DE (DMA enable) to 0
  [Function] Disable DMA transfer
- Setting DMA transfer source address (SARn)
  [Function] Specify DMA transfer source address
- Setting DMA transfer destination address (DARn)
  [Function] Specify DMA transfer destination address
- Setting the DMA transfer count (DMATCRn)
  [Function] Set the DMA transfer count
- Setting the DMA transfer mode (CHCRn)
  TC: Set the TC (transfer count mode)
  [Function]
  "0": Transfer data once for each transfer request
  (When the SCIF_3 or IIC3 is selected as the transfer request source)
  "1":Transfer data for the count specified in DMATCRn for each transfer request
  RLD: Setting the RLD (reload function enable/disable)
  [Function] Enable/disable reload function
  DM: Set the DM[1:0] (destination address mode)
  [Function] Select whether the DMA transfer destination address is incremented or decremented
  Fix/increment/decrement the DMA transfer destination address
  SM: Set the SM[1:0] (source address mode)
  [Function] Select whether the DMA transfer source address is incremented or decremented
  Fix/increment/decrement the DMA transfer source address
  RS: Set the RS[3:0] (resource select) to B'1000.
  [Function] Select DMA extension resource selector (DMA transfer request source)
  TB: Set the TB (transfer bus mode)
  [Function] Select a DMA transfer bus mode from cycle-stealing mode or burst mode
  Note: When TC is set to 0, select cycle-stealing mode.
  TS: Set the TS[1:0] (transfer size)
  [Function] Specify the unit of DMA transfer
  IE: Set the IE (interrupt enable)
  [Function] Enable/disable interrupt requests
- Settings DMA transfer requests from on-chip peripheral modules (DMARS0 to DMARS3)
  [Function] Select the DMA transfer request source from SCIF_3, IIC3, A/D converter, MTU2, CMT, USB, SSU or RCAN
- Setting the DMA operation register (DMAOR)
  AE: Read from the AE (address error flag) and then clear it to 0
  [Function] Clear the address error flag
  NMIF: Read from the NMIF (NMI flag) and then clear it to 0
  [Function] Clear the NMI flag
  DME: Set the DME (DMA master enable) to 1
  [Function] Enable DMA transfer on all the channels
- Enabling DMA transfer (CHCRn)
  DE: Set the DE (DMA enable) to 1
  [Function] Start DMA transfer

Figure 2 Flowchart of DMAC Initialization

2.3 Operation of Sample Program

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

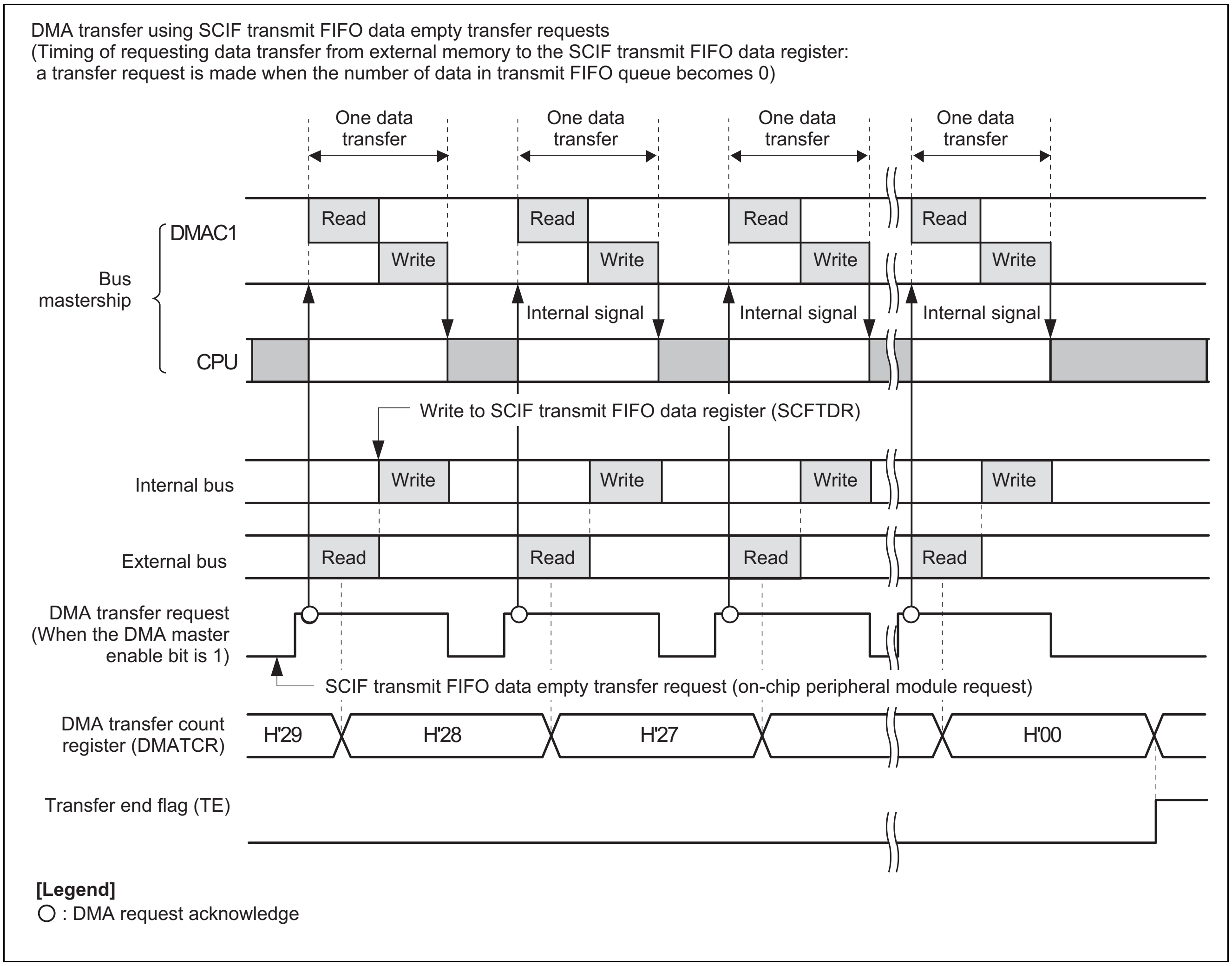

DMA transfer using SCIF transmit FIFO data empty transfer requests
(Timing of requesting data transfer from external memory to the SCIF transmit FIFO data register:
a transfer request is made when the number of data in transmit FIFO queue becomes 0)
One data transfer
One data transfer
One data transfer
One data transfer
Bus mastership
DMAC1
Read
Write
Read
Write
Read
Write
Read
Write
Internal signal
Internal signal
Internal signal
CPU
Write to SCIF transmit FIFO data register (SCFTDR)
Internal bus
Write
Write
Write
Write
External bus
Read
Read
Read
Read
DMA transfer request (When the DMA master enable bit is 1)
SCIF transmit FIFO data empty transfer request (on-chip peripheral module request)
DMA transfer count register (DMATCR)
H'29
H'28
H'27
H'00
Transfer end flag (TE)
[Legend]
○ : DMA request acknowledge

Figure 3 Operation Timing of Sample Application

2.4 Processing Procedure 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 3, 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 Value	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
			RLD = "0": Disable 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 980C	Start address of transfer destination: Address of the SCIF transmit FIFO data register_3 (SCFTDR_3)
DMA transfer count register_1 (DMATCR_1)	H'FFFE 1018	Number of character string data	Transfer count: 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 selector 0 (DMARS0)	H'FFFE 1300	H'8D00	MID = "B'100011"
			RID = "B'01"
			Set as SCIF_3 transmit FIFO data empty transfer request

Table 3 Macro Definitions Used in Sample Program

Macro Definition	Setting Value	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

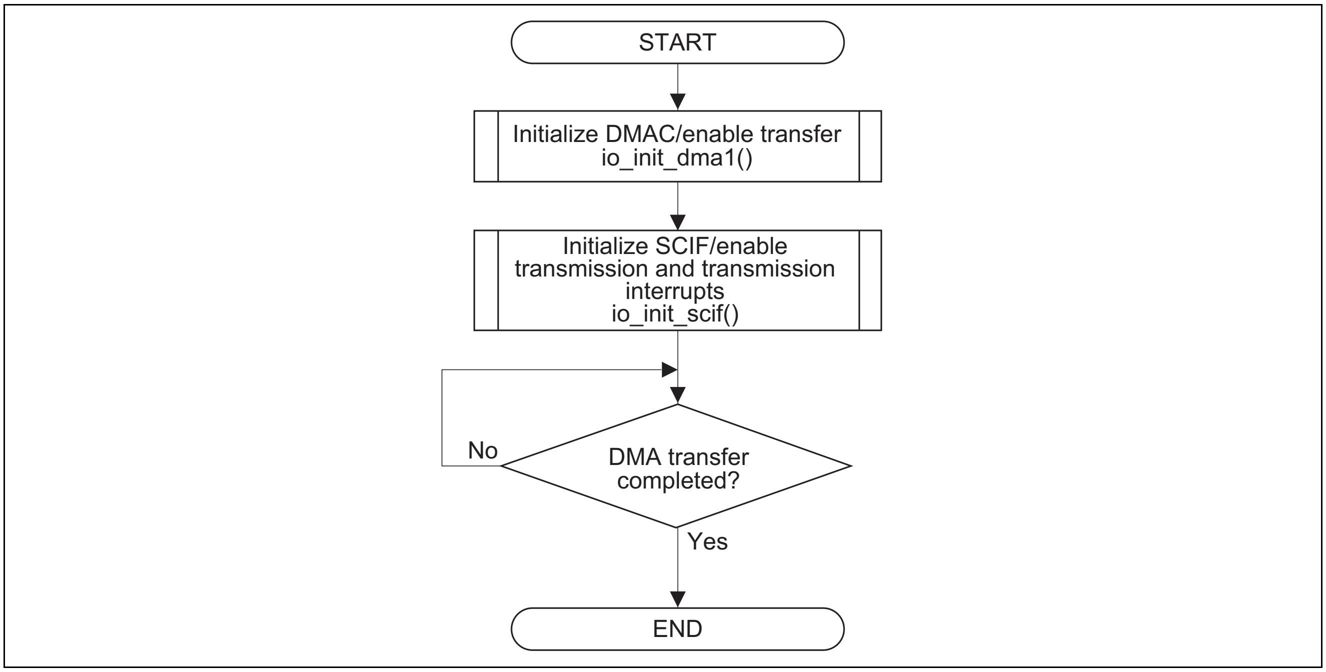

START
Initialize DMAC/enable transfer
io_init_dma1()
Initialize SCIF/enable transmission and transmission interrupts
io_init_scif()
No
DMA transfer completed?
Yes
END

Figure 4 Flowchart of Sample Program

3. Listing of the Sample Program

1. Sample Program Listing: main.c (1)

```
/*""FILE COMMENT""**************************************************************
*
*       System Name : SH7285 Sample Program
*       File Name   : main.c
*       Contents    : Data transfer to on-chip peripheral modules with DMAC
*       Version     : 1.00.00
*       Model       : M3A-HS85
*       CPU         : SH7285
*       Compiler    : SHC9.1.1.0
*       note        : Data transfer to the SCIF is performed using the DMAC.
*
*       The information described here may contain technical inaccuracies or
*       typographical errors. Renesas Technology Corporation and Renesas Solutions
*       assume no responsibility for any damage, liability, or other loss rising
*       from these inaccuracies or errors.
*
*       Copyright (C) 2008 Renesas Technology Corp. All Rights Reserved
*       AND Renesas Solutions Corp. All Rights Reserved
*
*       history   : 2008.02.26 ver.1.00.00
*""FILE COMMENT END""*********************************************************/
#include <string.h>
#include "iodefine.h"       /* SH7285 iodefine  */

/* ==== symbol definition ==== */
/* ==== DMAC Settings ==== */
#define DMA_SIZE_BYTE 0x0000u
#define DMA_SIZE_WORD 0x0001u
#define DMA_SIZE_LONG 0x0002u
#define DMA_SIZE_LONGx4 0x0003u
#define DMA_INT_DISABLE 0x0000u
#define DMA_INT_ENABLE 0x0010u
#define DMA_INT (DMA_INT_ENABLE >> 4u)

/* ==== prototype declaration ==== */
void main(void);
void io_init_dma1(void *src, void *dst, size_t size, unsigned int mode);
void io_dma1_stop(void);
void io_init_scif(int);

/* ==== RAM allocation variable declaration ==== */
typedef struct {
    unsigned char scbrr;
    unsigned short scsmr;
} SH_BAUD_SET;

/* ---- baud rate ---- */
enum{
    CBR_1200,
    CBR_2400,
    CBR_4800,
    CBR_9600,
    CBR_19200,
    CBR_31250,
    CBR_38400,
    CBR_57600,
    CBR_115200
};

```

2. Sample Program Listing: main.c (2)

```
static SH_BAUD_SET sci_baud[] = {
    { 80u, 2u}, /* 1200bps    error  0.46% */
    {162u, 1u}, /* 2400bps    error·-0.14% */
    { 80u, 1u}, /* 4800bps    error  0.46% */
    {162u, 0u}, /* 9600bps    error -0.14% */
    { 80u, 0u}, /* 19200bps   error  0.46% */
    { 49u, 0u}, /* 31250bps   error· 0.00% */
    { 40u, 0u}, /* 38400bps   error·-0.75% */
    { 26u, 0u}, /* 57600bps   error· 0.46% */
    { 13u, 0u}  /*115200bps   error·-3.11% */
};

/* Transmission character string */
signed char data[] = "SCIF request DMAC Sample Software.\r\n";

/*""FUNC COMMENT""*******************************************************
 * Outline     : Sample Program Main
 *-----------------------------------------------------------------------
 * Include     : #include "iodefine.h"
 *             : #include <machine.h>
 *-----------------------------------------------------------------------
 * Declaration : void main(void);
 *-----------------------------------------------------------------------
 * Function    : Sample Program Main
 *-----------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------
 * Return Value: void
 *-----------------------------------------------------------------------
 * Notice      :
 *""FUNC COMMENT END""***************************************************/
void main(void)
{
    /* ==== Setting of DMAC ==== */
    io_init_dma1(data, (void *)&SCIF3.SCFTDR ,sizeof(data),
                  DMA_SIZE_BYTE | DMA_INT_DISABLE);
                                /* Transfer requests : SCIF3 transmitter */
                                /* RAM -> SCIF transmitter               */

    /* ==== Setting of SCIF ==== */
    io_init_scif(CBR_57600);
                                /* UART mode            */
                                /* bit rate : 57600bps */

    /* ==== DMA start ==== */
    DMAC1.CHCR.BIT.DE = 1ul;    /* DMA enable */

    /* ==== DMA stop ==== */
    io_dma1_stop();

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

```

3. Sample Program Listing: main.c (3)

```
/*""FUNC COMMENT""*******************************************************
 * Outline     : Initialization of DMAC transfer
 *-----------------------------------------------------------------------
 * Include     : #include "iodefine.h"
 *-----------------------------------------------------------------------
 * Declaration : void io_init_dma1(void *src, void *dst, size_t size, unsigned int mode);
 *-----------------------------------------------------------------------
 * Function    : The DMAC transfers the amount of data specified by “size"
 *             : from the source address “src" to the destination address “dst".
 *             : Transfer is performed using requests from the SCIF3.
 *             : Transfer size and use or non-use of interrupts are specified for the
 *             : “mode”.
 *-----------------------------------------------------------------------
 * Argument    : void *src   : Source address
 *             : void *dst   : Destination address
 *             : size_t size : Transfer size (byte)
 *             : unsigned int mode : Transfer mode, specifies the following with logical OR.
 *             :     DMA_SIZE_BYTE  (0x0000) Byte transfer
 *             :     DMA_SIZE_WORD  (0x0001) Word transfer
 *             :     DMA_SIZE_LONG  (0x0002) Longword transfer
 *             :     DMA_SIZE_LONGx4(0x0003) 16-byte transfer
 *             :     DMA_INT_DISABLE(0x0000) DMA transfer end interrupt not in use
 *             :     DMA_INT_ENABLE (0x0010) DMA transfer end interrupt in use
 *-----------------------------------------------------------------------
 * Return Value: void
 *-----------------------------------------------------------------------
 * Notice      : Operation is not guaranteed when the source/destination address is not
 *             : on a boundary corresponding to the transfer size.
 *             : If interrupts are to be used, the interrupt routines must be registered.
 *""FUNC COMMENT END""***************************************************/
void io_init_dma1(void *src, void *dst, size_t size, unsigned int mode)
{
    unsigned int ts;
    unsigned long ie;

    ts = mode & 0x3u;
    ie = (mode & 0x00f0u ) >> 4u;

    /* ==== Setting of power down mode ==== */
    STB.CR2.BIT._DMAC = 0x0;        /* Clear the DMAC module standby mode */

    /* ---- DMA Channel Control Registers(CHCR) ---- */
    DMAC1.CHCR.BIT.DE = 0ul;        /* DMA disable */

    /* ---- DMA Source Address Registers(SAR) ---- */
    DMAC1.SAR = (void *)src;

    /* ---- DMA Destination Address Registers(DAR) ---- */
    DMAC1.DAR = (void *)dst;
```

4. Sample Program Listing: main.c (4)

```
    /* ---- DMA Transfer Count Registers(DMATCR) ---- */
    switch(ts){
    case DMA_SIZE_BYTE:
        DMAC1.DMATCR = size;              /* Specify transfer count (1/1) */
        DMAC1.RDMATCR = size;
        break;
    case DMA_SIZE_WORD:
        DMAC1.DMATCR = size >> 1u;        /* Specify transfer count (1/2) */
        DMAC1.RDMATCR = size >> 1u;
        break;
    case DMA_SIZE_LONG:
        DMAC1.DMATCR = size >> 2u;        /* Specify transfer count (1/4) */
        DMAC1.RDMATCR = size >> 2u;
        break;
    case DMA_SIZE_LONGx4:
        DMAC1.DMATCR = size >> 4u;        /* Specify transfer count (1/16) */
        DMAC1.RDMATCR = size >> 4u;
        break;
    default:
        break;
    }
    /* ---- DMA Channel Control Registers (CHCR) ---- */
    DMAC1.CHCR.LONG = 0x00001800ul | (ts << 3u) | (ie << 2u) ;
                                    /* Fixed destination address      */
                                    /* Source address is incremented  */
                                    /* DMA extension resource selector */
                                    /* Cycle steal mode               */
                                    /* Transfer Size : Byte unit      */
    /* ---- DMA Extension Resource Selectors 0(DMARS0) ---- */
    DMAC.DMARS0.BIT.C1MID = 0x23u;   /* Transfer requests : SCIF3 transmitter */
    DMAC.DMARS0.BIT.C1RID = 0x1;

    /* ---- DMA Operation Register(DMAOR) ---- */
    DMAC.DMAOR.WORD &= 0xfff9u;     /* AE,NMIF clear */

    if(DMAC.DMAOR.BIT.DME == 0ul){  /* DMA Master Enable */
        DMAC.DMAOR.BIT.DME = 1ul;
    }
}
/*""FUNC COMMENT""*******************************************************
 * Outline     : DMAC stop
 *-----------------------------------------------------------------------
 * Include     : #include "iodefine.h"
 *-----------------------------------------------------------------------
 * Declaration : void io_dma1_stop(void);
 *-----------------------------------------------------------------------
 * Function    : Stops DMA transfer
 *-----------------------------------------------------------------------
 * Argument    : void
 *-----------------------------------------------------------------------
 * Return Value: void
 *-----------------------------------------------------------------------
 * Notice      :
 *""FUNC COMMENT END""**************************************************/
void io_dma1_stop(void)
{
    /* Transmission end detection */;
    while(DMAC1.CHCR.BIT.TE == 0ul){
        /* wait TE bit set */
    }
    /* ---- DMA end ---- */
    DMAC1.CHCR.BIT.DE = 0ul;    /* DMA disable */
}

```

5. Sample Program Listing: main.c (5)

```
/*""FUNC COMMENT""********************************************************
 * Outline     : SCIF Settings
 *------------------------------------------------------------------------
 * Include     : #include "iodefine.h"
 *------------------------------------------------------------------------
 * Declaration : void io_init_scif(void);
 *------------------------------------------------------------------------
 * Function    : Specifies settings for the serial communications interface
 *             : with FIFO (SCIF).
 *------------------------------------------------------------------------
 * Argument    : int bps : Specified baud rate
 *------------------------------------------------------------------------
 * Return Value: void
 *------------------------------------------------------------------------
 * Notice      :
 *""FUNC COMMENT END""****************************************************/
void io_init_scif(int bps)
{
    /* ==== Setting of power down mode ==== */
    STB.CR4.BIT._SCIF3 = 0u;    /* Clear the SCIF3 module standby mode */

    /* ==== Setting of SCIF ==== */
    /* ---- Serial Control Register(SCSCR) ---- */
    SCIF3.SCSCR.WORD &= 0x00u;  /* Transmitter/Receiver disabled */
    SCIF3.SCSCR.BIT.CKE = 0x0u; /* Internal clock */

    /* ---- Serial Mode Register(SCSMR) ---- */
    SCIF3.SCSMR.WORD = sci_baud[bps].scsmr;
                                /* Asynchronous mode               */
                                /* 8-bit data                      */
                                /* Parity bit not added or checked */
                                /* One stop bit                    */

    /* ---- Bit Rate Register(SCBRR) ---- */
    SCIF3.SCBRR = sci_baud[bps].scbrr;

    /* ==== Setting of PFC ==== */
    /* ---- port E control register L2 ---- */
    PFC.PECRL2.BIT.PE5MD = 5u;  /* Set TxD3 */

    /* ---- Serial Control Register(SCSCR) ---- */
    SCIF3.SCSCR.BIT.TIE = 1u;   /* Transmit interrupt enabled */
    SCIF3.SCSCR.BIT.TE = 1u;    /* Transmitter enabled */}

/* End of File */
```

4. Documents for Reference

- Software Manual
  SH-2A, SH2A-FPU Software Manual
  The most up-to-date version of this document is available on the Renesas Technology Website.
- Hardware Manual
  SH7280 Group Hardware Manual
  The most up-to-date version of this document is 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	Oct.31.08	—	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.