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.

SH7263/SH7203 Group

Data Transfer between Memory Areas with DMAC

Introduction

This application note provides an example of transferring data between memory areas with the direct memory access controller (DMAC) of the SH7263/SH7203.

Target Device

SH7263/SH7203

Contents

1. Introduction

1.1 Specification

- 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

- Microcontroller: SH7263/SH7203
- Operating Frequency: Internal clock 200 MHz
  Bus clock 66.67 MHz
  Peripheral clock 33.33 MHz
- C Compiler: SuperH RISC engine family C/C++ compiler package Ver.9.01, from Renesas Technology
- Compile Option: -cpu = sh2afpu -fpu = single -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

1.4 Related Application Note

The operation of the reference program for this document was confirmed with the setting conditions described in the application note: *SH7263/SH7203 Initialization Example*. Please refer to the application note in combination with this one.

2. Description of Sample Application

In this sample application, the direct memory access controller (DMAC) is used to transfer data from the on-chip RAM to external memory.

2.1 Operational Overview of Module Used

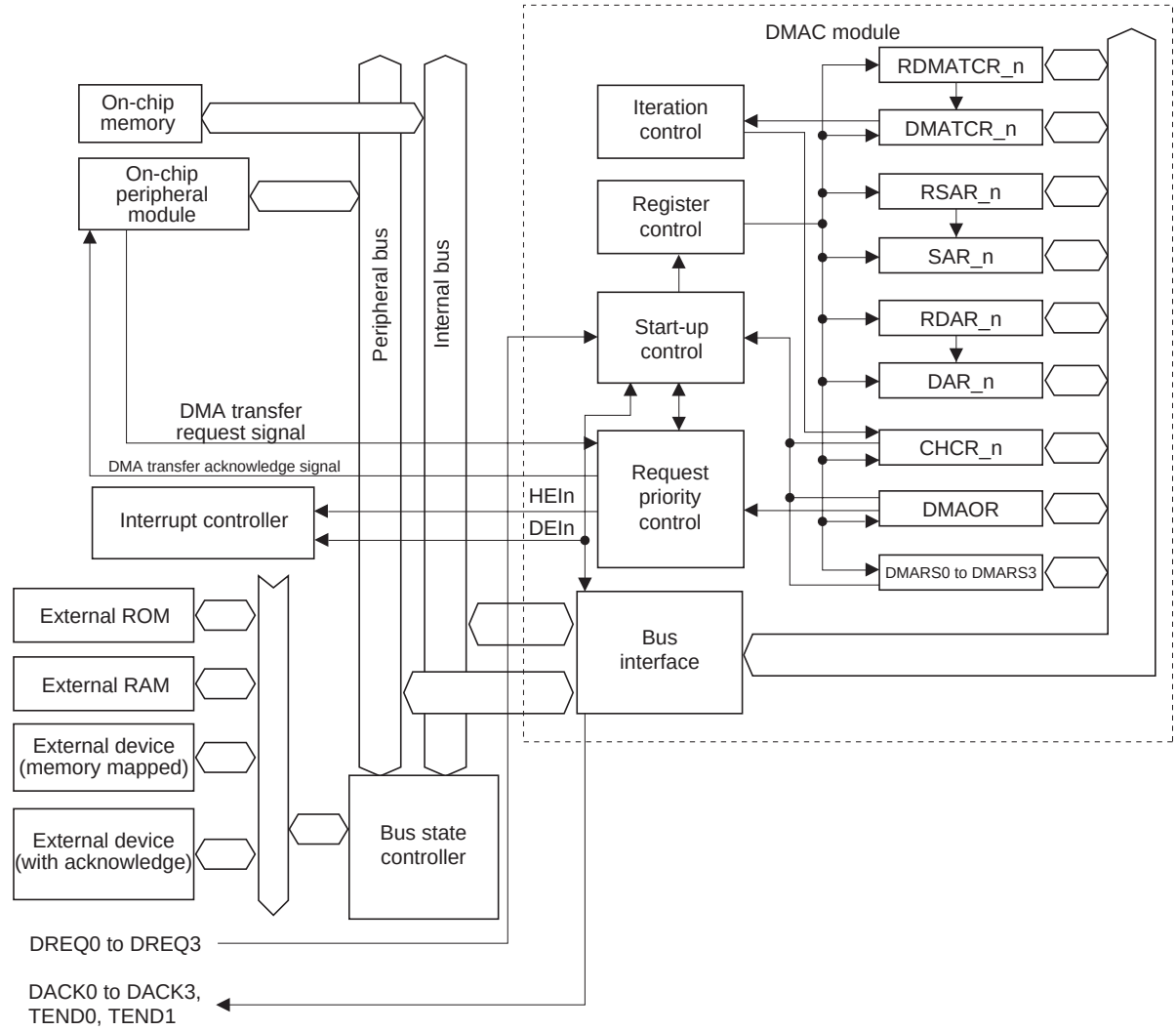
When a DMA transfer request is made, the DMAC starts to transfer data in order of priority of channels. Then, it continues the transfer operation until transfer end condition is met. It has three transfer request modes: auto request, external request, and on-chip peripheral module request. The bus mode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

An overview of the DMAC is provided 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 4 channels (CH0 to CH3) 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 • SH7203/SH7263 (SCIF: 8 sources, I^2C3: 8 sources, ADC: 1 source, MTU2: 5 sources, CMT: 2 sources, USB: 2 sources, FLCTL: 2 sources, RCAN-TL1: 2 sources, SSI: 4 sources, SSU: 4 sources) • SH7263 (SRC: 2 sources, ROM-DEC: 1 source, SDHI: 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 *SH7263/SH7203 Group Hardware Manual*.



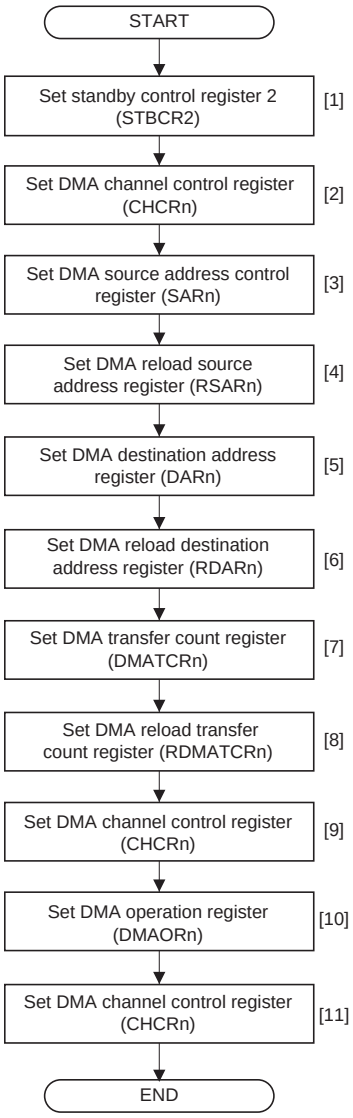
[Legend]

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

Figure 1 Block Diagram of DMAC

2.2 Procedure for Setting 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 *SH7263/SH7203 Group Hardware Manual*.



[1] • Enabling clock supply to the DMAC (STBCR2)
Clear the MSTP8 (module stop 8) bit to 0
[Function] Clock supply to the DMAC

[2] • Disabling DMA transfer (CHCRn)
Clear the DE (DMA enable) bit to 0
[Function] Disable DMA transfer

[3] • Setting DMA transfer source address (SARn)
[Function] Specify DMA transfer source address

[4] • Setting DMA transfer source reload address (RSARn)
[Function] Specify DMA transfer source address to be reloaded

[5] • Setting DMA transfer destination address (DARn)
[Function] Specify DMA transfer destination address

[6] • Setting DMA transfer destination reload address (RDARn)
[Function] Specify DMA transfer destination address to be reloaded

[7] • Setting the DMA transfer count (DMATCRn)
[Function] Set the DMA transfer count

[8] • Setting the DMA transfer reload count (RDMATCRn)
[Function] Set the DMA transfer count to be reloaded

[9] • Setting the DMA transfer mode (CHCRn)
Set the TC (transfer count mode) bit to 1
[Function] Transfer data for the count specified in DMATCRn for each transfer request
Set the RLDSAR (SAR reload function enable/disable) bit
[Function] Enables/disables reload function to SAR and DMATCR
Set the RLDDAR (DAR reload function enable/disable) bit
[Function] Enables/disables reload function to DAR and DMATCR
Set the DM (destination address mode) bits
[Function] Select whether the DMA transfer destination address is incremented or decremented Fix/increment/decrement the DMA transfer destination address
Set the SM (source address mode) bits
[Function] Select whether the DMA transfer source address is incremented or decremented Fix/increment/decrement the DMA transfer source address
Set the RS (resource select) bits to B'0100.
[Function] Select auto request (as the DMA transfer request source)
Set the TB (transfer bus mode) bit
[Function] Select a DMA transfer bus mode.
Set the TS (transfer size) bits
[Function] Specify the DMA transfer size
Set the IE (interrupt enable) bit
[Function] Enable/disable interrupt requests

[10] • Setting the DMA operation register (DMAOR)
Read from the AE (address error flag) bit and then clear it to 0
[Function] Clear the address error flag
Read from the NMIF (NMI flag) bit and then clear it to 0
[Function] Clear the NMI flag
Set the DME (DMA master enable) bit to 1
[Function] Enable DMA transfer on all the channels

[11] • Enabling DMA transfer (CHCRn)
Set the DE (DMA enable) bit to 1
[Function] Start DMA transfer

Figure 2 Flowchart of Initializing DMAC

2.3 Operation of 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.

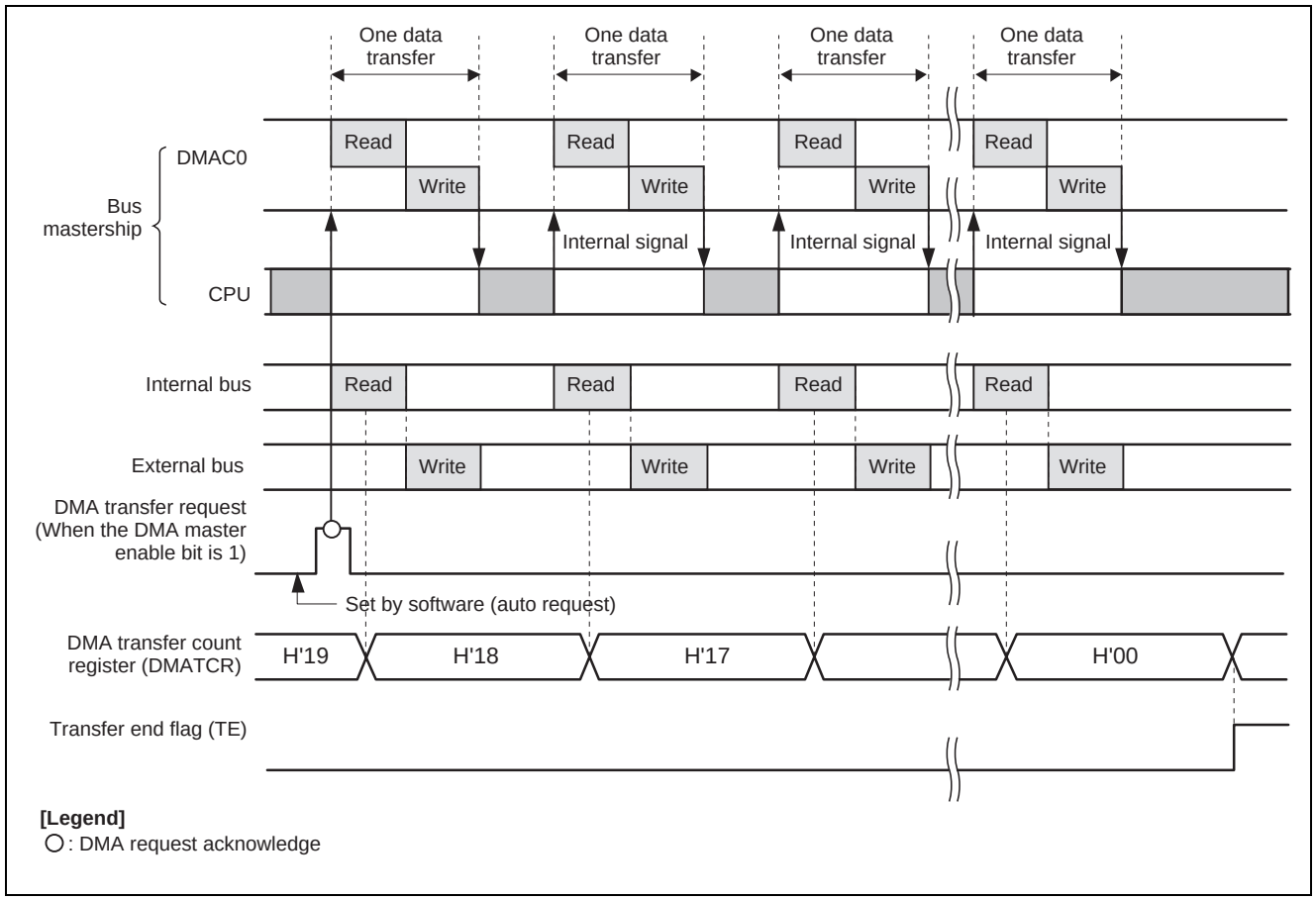


Figure 3 Operation Timing of Sample Application

2.4 Usage Notes on 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.
- In DMA transfer with operand cache enabled, coherency must be kept by disabling or writing back the cache. In the sample program, coherency is kept because a cache-disabled space is accessed from the CPU.

2.5 Processing Procedure of 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 Value	Description
Standby control register 2 (STBCR2)	H'FFFE 0018	H'00	MSTP8 = 0: DMAC operates
DMA channel control register 0 (CHCR0)	H'FFFE 100C	H'0000 0000	DE = 0: Disables DMA transfer
		H'8000 5410	TC = 1 Transfers data for the count specified in DMATCR0 for each DMA transfer request RLDSAR = 0: Disables SAR reload function RLDDAR = 0: Disables DAR reload function DM = B'01: Increments destination address SM = B'01: Increments source address RS = B'0100: Auto request TB = 0: Cycle-stealing mode TS = B'10: Longword transfer IE = 0: Disables interrupt request
		H'8000 5411	DE = 1: Enables DMA transfer
DMA source address register 0 (SAR0)	H'FFFE 1000	H'FFF8 8000	Set start address of transfer source in an on-chip RAM area
DMA destination address register 0 (DAR0)	H'FFFE 1004	H'2C00 0000	Set start address of transfer destination in an external memory area*
DMA transfer count register 0 (DMATCR0)	H'FFFE 1008	H'64	Transfer count: 100 transfers (H'64)
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'0000 0000	Not used for auto request

Note: * The address of external memory varies depending on the target board to be used.

Table 3 Macro Definitions Used in Sample Program

Macro Definition	Setting Value	Description
SDRAM_DST_ADR	H'2C00 0000	• Start address of SDRAM
SRAM_SRC_ADR	H'FFF8 8000	• Start address of on-chip RAM
SIZE	H'64	• Transfer count
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 disabled
DMA_INT_ENABLE	H'0010	• DMA transfer end interrupt enabled

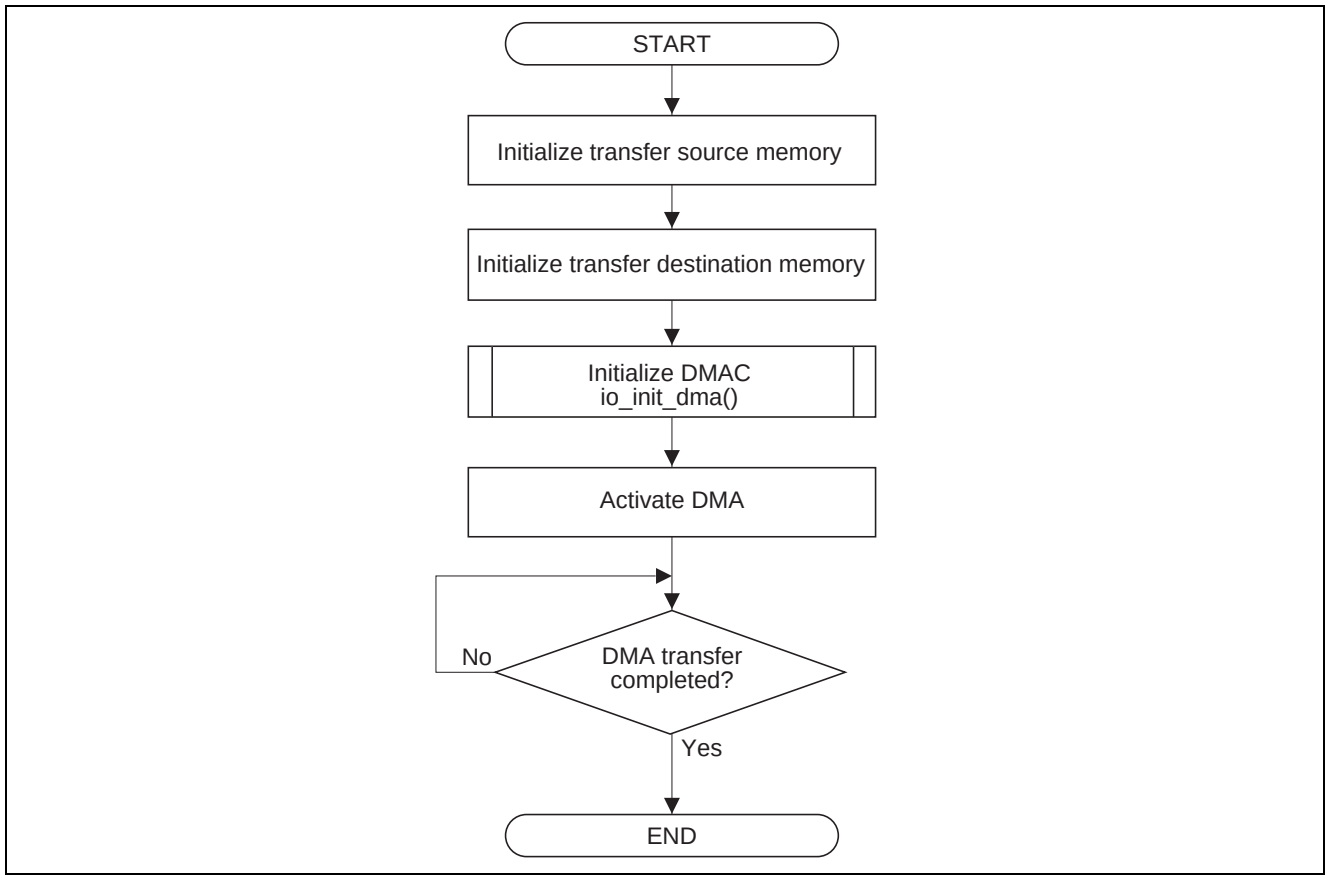


Figure 4 Flowchart of Sample Program

3. Sample Program

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

```
/*""FILE COMMENT""*******************************************************************************
*
*        System Name : SH7203 Sample Program
*        File Name   : main.c
*        Contents    : Data transfer between memory areas with DMAC
*        Version     : 1.00.00
*        Model       : M3A-HS30
*        CPU         : SH7203
*        Compiler    : SHC9.1.1.0
*        note        : A sample program for transferring data with the DMAC0.
*                      Using software triggers transfers 100-byte data from the on-chip RAM to
*                      external memory.
*
*                    <Caution>
*                    This sample program is for reference
*                    and its operation is not guaranteed.
*                    Customers should use this sample program for technical reference
*                    in software development.
*
*        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) 2007 Renesas Technology Corp. All Rights Reserved
*        AND Renesas Solutions Corp. All Rights Reserved
*
*        history     : 2007.12.27 ver.1.00.00
*""FILE COMMENT END""*****************************************************/
#include <stdio.h>
#include "iodefine.h"              /* iodefine.h is automatically created by HEW */

/* ==== Macro declaration ==== */
#define SDRAM_DST_ADR   ((void *)0x2c000000)   /* External SDRAM start address       */
#define SRAM_SRC_ADR ((void *)0xfff88000)      /* Internal SRAM start address        */
#define SIZE         100                       /* 100 bytes of data are transferred  */


#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_dma0(void *src, void *dst, size_t size, unsigned int mode);
void io_dma0_enable(void);
void io_dma0_stop(void);


```

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

```
/*""FUNC COMMENT""*****************************************************
 * Outline     : Sample Program Main
 *---------------------------------------------------------------------
 * Include     :
 *---------------------------------------------------------------------
 * Declaration : void main(void);
*---------------------------------------------------------------------
* Function     : Sample program for transferring 100-byte data from on-chip RAM to external
*              : SDRAM.
*              : Completion of DMA transfer is detected through the DMA transfer-end flag.
*              : When DMA transfer ends, the processing enters infinite loop.
*---------------------------------------------------------------------
* Argument     : void
*---------------------------------------------------------------------
* Return Value : void
*---------------------------------------------------------------------
* Notice       :·In the sample program, absolute addresses are used to clarify
*              : the start addresses of the data transfer source and destination.
*              : When allocating memory areas by absolute addresses, be careful so that
*              : they do not overlap with the sections used by user programs.
*              :·In DMA transfer with the operand cache enabled,
*              : coherency must be kept by disabling or writing back the cache.
*              : In the sample program, coherency is kept because cache-disabled space is
*              : accessed from the CPU.
*""FUNC COMMENT END""*************************************************/
void main(void)
{
            int i;
            volatile unsigned char *ptr;

            /* ====  Transfer source memory initialization ==== */
            ptr = SRAM_SRC_ADR;
            for(i=0; i < SIZE; i++){
                    *ptr++ = 0x55;            /* Fill the transfer source memory with 0x55 */
            }

            /* ==== Transfer destination memory initialization ==== */
            ptr = SDRAM_DST_ADR;
            for(i=0; i < SIZE; i++){
                    *ptr++ = 0;               /* Clear transfer destination memory to all 0 */
            }

            /* ==== DMAC initialization ==== */
            io_init_dma0(SRAM_SRC_ADR, SDRAM_DST_ADR, SIZE , DMA_SIZE_LONG | DMA_INT_DISABLE);

            /* ---- Start DMA transfer ----   */
            io_dma0_enable();

            /* ---- Stop DMA transfer ----    */
            io_dma0_stop();

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

```

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

```
/*""FUNC COMMENT""*****************************************************************************
 * Outline        : Initialization for DATA transfer between memory areas with DMAC
 *-------------------------------------------------------------------------------------------
 * Include        : #include "iodefine.h"
 *-------------------------------------------------------------------------------------------
 * Declaration    : io_init_dma0(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."
 *                : Auto request mode is used to transfer data.
 *                : "mode" is specified for transfer size and interrupt used/not used
 *-------------------------------------------------------------------------------------------
 * 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 disabled
 *                :              DMA_INT_ENABLE(0x0010)  DMA transfer end interrupt enabled
 *                :
 *-------------------------------------------------------------------------------------------
 * Return Value   : void
 *-------------------------------------------------------------------------------------------
 * Notice         : Operation is not guaranteed when the alignment of the source/destination
 *                : address is inconsistent.
 *                : When interrupts are used, interrupt routines must be registered.
 *""FUNC COMMENT END""*************************************************************************/
void io_init_dma0(void *src, void *dst,  size_t size, unsigned int mode)
{
      unsigned int ts;
      unsigned long ie;

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

      /* ==== Set standby control register 2 (STBCR2) ==== */
      CPG.STBCR2.BIT.MSTP8 = 0x0;   /* Cancel module stop mode of the DMAC    */

      /* ---- Set DMA channel control register ---- */
      DMAC.CHCR0.BIT.DE = 0ul;      /* Disable DMA transfer       */

      /* ---- Set DMA source address register ---- */
      DMAC.SAR0.LONG = (unsigned long)src;

      /* ---- Set DMA reload source address register ---- */
      DMAC.RSAR0.LONG = (unsigned long)src;

      /* ---- Set DMA destination address register ---- */
      DMAC.DAR0.LONG = (unsigned long)dst;

      /* ---- Set DMA reload destination address register ---- */
      DMAC.RDAR0.LONG = (unsigned long)dst;

      /* ---- Set DMA transfer count register ---- */
```

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

```
167         /* ---- Set DMA reload transfer count register ---- */
168
169         switch(ts){
170         case DMA_SIZE_BYTE:
171             DMAC.DMATCR0.LONG = size;             /* Specify transfer count (1/1) */
172             DMAC.RDMATCR0.LONG = size;
173             break;
174         case DMA_SIZE_WORD:
175             DMAC.DMATCR0.LONG = size >> 1u;       /* Specify transfer count (1/2) */
176             DMAC.RDMATCR0.LONG = size >> 1u;
177             break;
178         case DMA_SIZE_LONG:
179             DMAC.DMATCR0.LONG = size >> 2u;       /* Specify transfer count (1/4) */
180             DMAC.RDMATCR0.LONG = size >> 2u;
181             break;
182         case DMA_SIZE_LONGx4:
183             DMAC.DMATCR0.LONG = size >> 4u;       /* Specify transfer count (1/16) */
184             DMAC.RDMATCR0.LONG = size >> 4u;
185             break;
186         default:
187             break;
188         }
189
190         /* ---- Set DMA channel control register ---- */
191         DMAC.CHCR0.LONG = 0x80005400ul |  (ts << 3u) | (ie << 2u) ;
192             /*
193              bit31      : TC DMATCR transfer:1--------    DMA transfer count specified in DMATC
194              bit30      : reserve 0
195              bit29      : RLDSAR OFF : 0-------------     Disable SAR reload function
196              bit28      : RLDDAR OFF : 0-------------     Disable DAR reload function
197              bit27-24   : reserve 0
198              bit23      : DO over run0 : 0----------      Unused
199              bit22      : TL TEND low active : 0----      Unused
200              bit21      : reserve 0
201              bit20      : TEMASK : TE set mask : 0--      Disable DMA transfer when TE bit is
202                                                          set
203              bit19      : HE :0---------------------      Unused
204              bit18      : HIE :0--------------------      Unused
205              bit17      : AM :0---------------------      Unused
206              bit16      : AL :0---------------------      Unused
207              bit15-14   : DM1:0 DM0:1---------------      Increment destination address
208              bit13-12   : SM1:0 SM0:1---------------      Increment source address
209              bit11-8    : RS : auto request : B'0100-     Auto request
210              bit7       : DL : DREQ level : 0 -------     Unused
211              bit6       : DS : DREQ select :0 Low level   Unused
212              bit5       : TB : cycle :0---------------    Cycle-stealing mode
213              bit4-3     : TS : transfer size:B'10---      Longword transfer
214              bit2       : IE : interrupt enable:0---      Disable interrupts
215              bit1       : TE : transfer end----------     Clear TE flag
216              bit0       : DE : DMA enable bit:0-----      Disable DMA transfer
217             */
218
219         /* ----Set DMA operation register---- */
220         DMAC.DMAOR.WORD &= 0xfff9u;                  /* AE,NMIF */
```

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

```
221         if(DMAC.DMAOR.BIT.DME == 0){                /* Enable DMA transfer on all channels */
222                 DMAC.DMAOR.BIT.DME = 1;
223         }
224
225 }
226 /*""FUNC COMMENT""*************************************************
227  * Outline         : DMAC Actibation
228  *-----------------------------------------------------------------------
229  * Include         : #include "iodefine.h"
230  *-----------------------------------------------------------------------
231  * Declaration     : void io_dma0_enable(void);
232  *-----------------------------------------------------------------------
233  * Function    : Performs DMA transfer
234  *-----------------------------------------------------------------------
235  * Argument    : void
236  *-----------------------------------------------------------------------
237  * Return Value: void
238  *-----------------------------------------------------------------------
239  * Notice          :
240  *""FUNC COMMENT END""************************************************/
241 void io_dma0_enable(void)
242 {
243         /* ---- Execute DMA transfer ---- */
244         DMAC.CHCR0.BIT.DE = 1ul;              /* DMA */
245
246 }
247 /*""FUNC COMMENT""*************************************************
248  * Outline         : DMAC Stop
249  *-----------------------------------------------------------------------
250  * Include         : #include "iodefine.h"
251  *-----------------------------------------------------------------------
252  * Declaration     : void io_dma0_stop(void);
253  *-----------------------------------------------------------------------
254  * Function    : Checks whether the transfer is completed and stops the DMA transfer.
255  *-----------------------------------------------------------------------
256  * Argument    : void
257  *-----------------------------------------------------------------------
258  * Return Value: void
259  *-----------------------------------------------------------------------
260  * Notice          :
261  *""FUNC COMMENT END""************************************************/
262 void io_dma0_stop(void)
263 {
264         /* Detect the end of transfer */;
265         while(DMAC.CHCR0.BIT.TE == 0ul){
266                 /* Wait until the TE bit is set */
267         }
268         /* ---- Stop DMA transfer ---- */
269         DMAC.CHCR0.BIT.DE = 0ul;              /* Disable transfer by DMA0 */
270 }
271
272 /* 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
  SH7203 Group Hardware Manual
  SH7263 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	Apr.17.08	—	First edition issued
1.01	Dec.17.08	—	Source file is updated

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.