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

SSI Master Transmitter

Introduction

This application note presents an example of data transfer by the serial sound interface (SSI).

Target Device

SH7263/SH7203

Contents

1. Preface

1.1 Specifications

The serial sound interface (SSI) is set to master transmitter mode for PCM data transmission.

The direct memory access controller (DMAC) is used for data transfer to the SSI.

1.2 Module Used

- Serial sound interface (SSI)
- Direct memory access controller (DMAC)

1.3 Applicable Conditions

• MCU:	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 Release01 from Renesas Technology	
• Compiler options:	-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 sample program in this application note was confirmed with the configuration specified in the application note " Example of Initialization " for the SH7263/SH7203 Group (REJ06B0740). Please refer to that document when setting up this sample task.

2. Description of the Sample Application

In this sample application, the SSI operates as a master transmitter with the sampling rate set to 44.1 kHz.

2.1 Operational Overview of Module Used

The following are the features of the serial sound interface (SSI):

- Number of channels: Four channels
- Operating mode: Non-compressed mode
  The non-compressed mode supports serial audio streams divided by channels.
- Serves as both a transmitter and a receiver
- Capable of using serial bus format
- Handles asynchronous transfer between the data buffer and the shift register.
- It is possible to select a dividing ratio for the clock used by the serial bus interface.
- It is possible to control data transmission and reception with DMAC or interrupt requests.
- Selects the oversampling clock input from among the following pins:
  EXTAL, XTAL (Clock operation modes 0)
  CKIO (Clock operation mode 2)
  AUDIO_CLK
  AUDIO_X1, AUDIO_X2

To change the oversampling clock, change the value in the SSI oversampling clock selection register (SCSR) of the pin function controller (PFC).

Table 1 shows the oversampling clock source selection made by setting the SSInCKS bits in the SCSR. Figure 1 shows the block diagram of the SSI.

Table 1 Oversampling Clock Sources Selected by SSInCKS Bits

Settings of SSInCKS[2:0] *[1]	Clock Operation Mode		
	0 or 1	2	3
000	AUDIO_X1 input		
001	AUDIO_X1 input / 4		
010	AUDIO_CLK input *[2]		
011	AUDIO_CLK input *[2] / 4		
100	EXTAL input	CKIO input	Setting prohibited
101	EXTAL input / 4	CKIO input / 4	Setting prohibited
110	EXTAL input / 2	CKIO input / 2	Setting prohibited
111	EXTAL input / 8	CKIO input / 8	Setting prohibited

Notes: 1. n = 0 to 3
2. When using the AUDIO_CLK input clock, set the PF30MD0 bit to 1 in the port F control register H4 (PFCRH4).

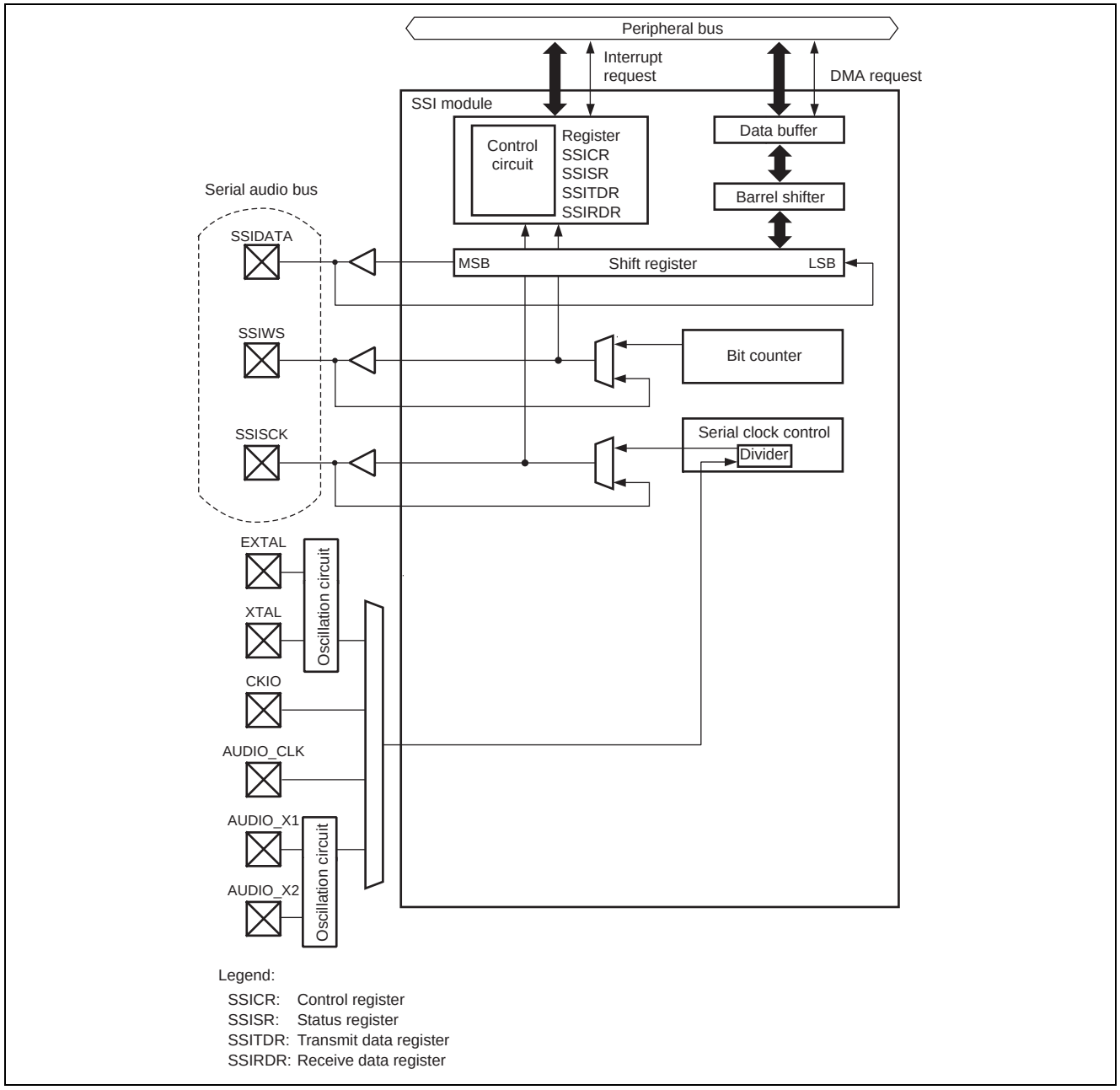

Figure 1 Block Diagram of SSI

2.2 Procedure for Setting the Module Used

Figures 2 and 3 show the examples of the SSI and DMAC setting procedures, respectively.

For details on the settings of individual registers, see the SH7263/SH7203 Group Hardware Manual (REJ09B0290/REJ09B0313).

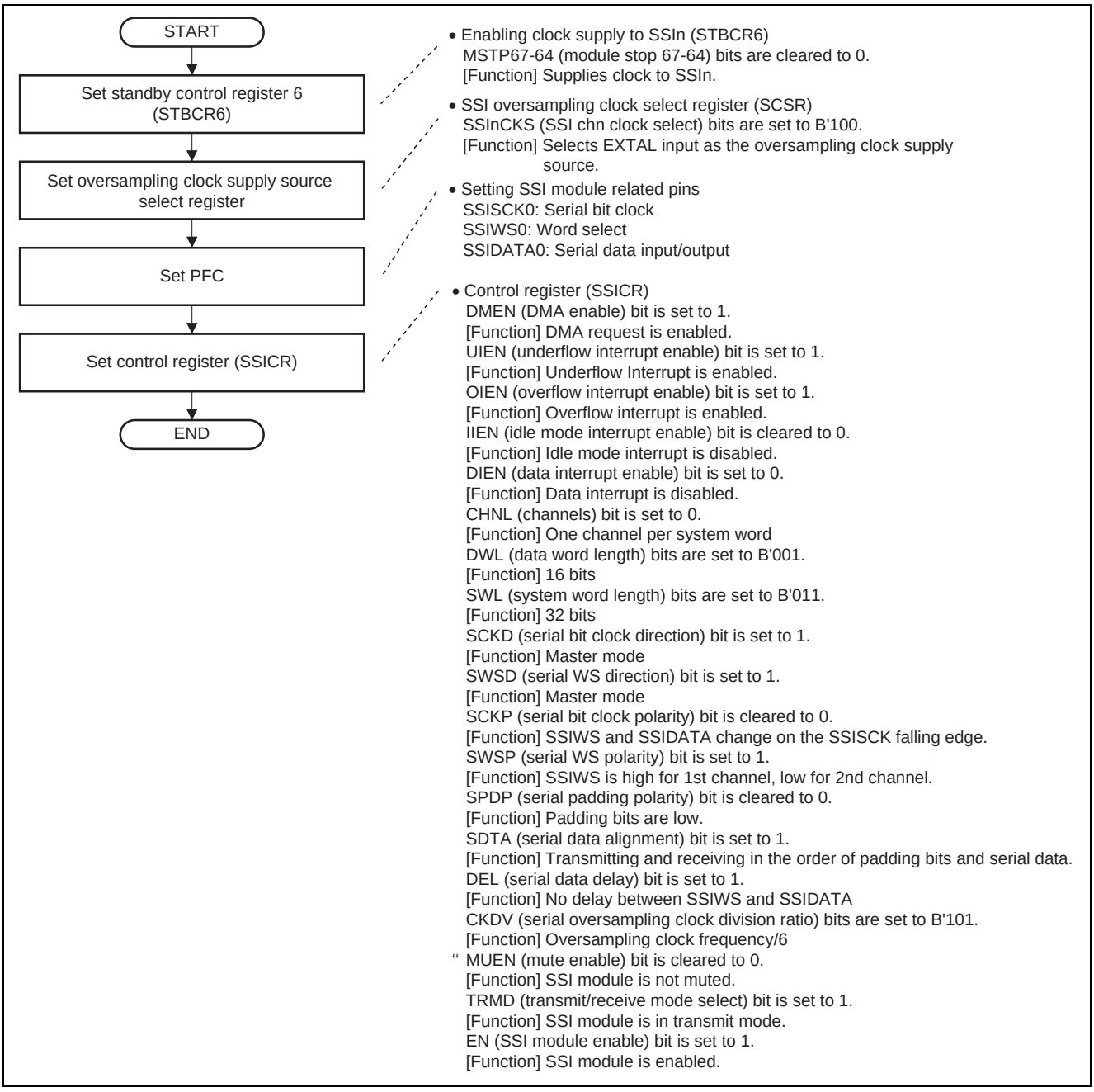

Figure 2 Example of SSI Setting Procedure

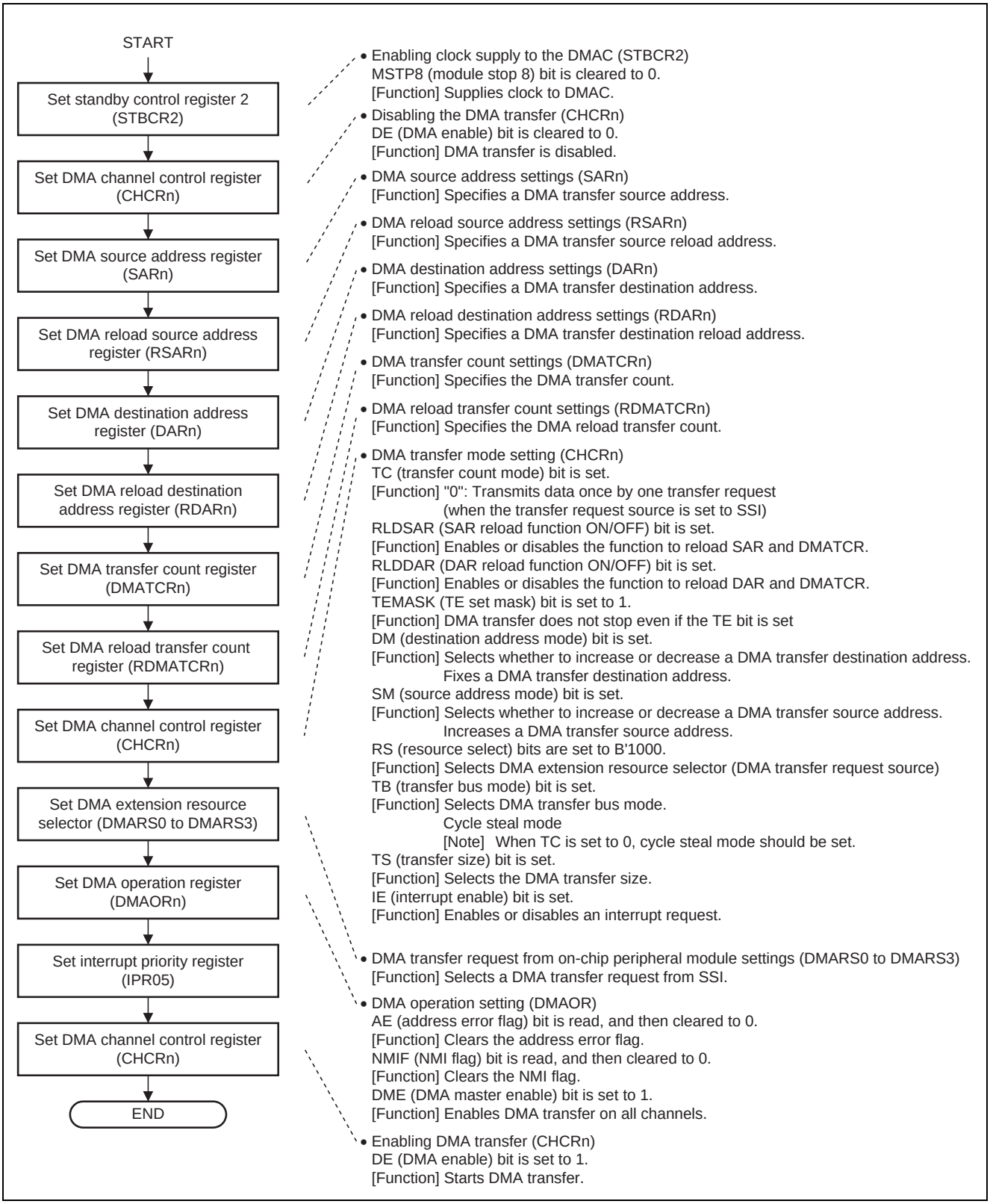

Figure 3 Example of DMAC Setting Procedure

2.3 Operation of the Sample Program

In the sample program, the DMAC channel 1 is activated by the DMA transfer request from the SSI, and data is transferred from the external memory to the transmit data register (SSITDR) in the SSI channel 0. The data written to SSITDR is transferred to the shift register upon transmission request, and is then output from the SSIDATA pin.

In the sample program, 10 samples (40 bytes) of PCM data are transferred four times. When the transfer has been completed, the SSI output is muted.

The SSI settings for the sample program are as follows:

- Channel used: channel 0
- Operation mode: master transmitter mode
- Data transmission control method: DMAC
- Oversampling clock: AUDIO_X1 input (16.9344 MHz)
- Serial oversampling clock frequency: Oversampling clock frequency/6 (2.8224 MHz)
- Data word length: 16 bits
- System word length: 32 bits
- Padding bit: "L" level
- No delay between SSIWS and SSIDATA
- SSIWS and SSIDATA change on the falling edge of SSISCK.
- Sampling frequency: 44.1 kHz [354 ns (2.8224 MHz) × 32 bits × 2]
- "H'FFFF" and "H'0000" are set in data word 1 (L channel) of the 1st channel and data word 2 (R channel) of the 2nd channel, respectively.

Figure 4 shows the output waveform of the sample program and figure 5 shows the block diagram of the configuration used for the sample program.

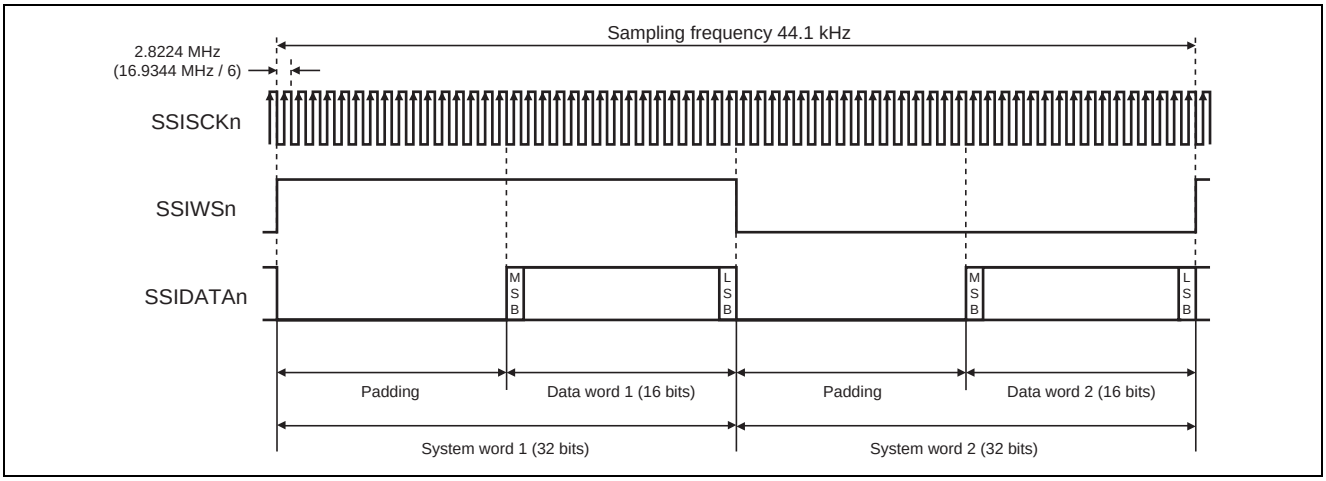

Figure 4 Output Waveform of Sample Program

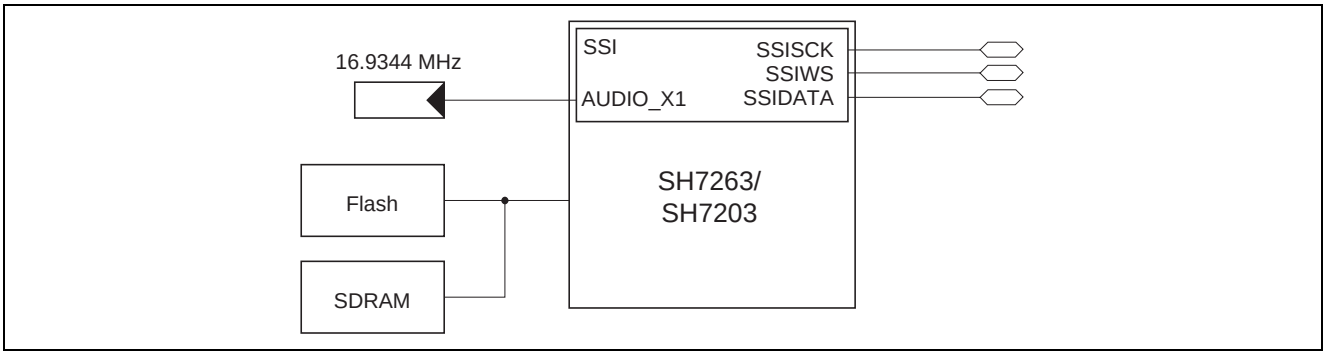

Figure 5 Block Diagram of Configuration Used for Sample Program

2.4 Sequence of Processing by the Sample Program

Tables 2 and 3 show the settings of the SSI and DMAC registers used in the sample program, respectively.

Figure 6 shows the processing flow of the sample program.

Table 2 SSI Register Settings Used in Sample Program

Register Name	Address	Setting Value	Description
Control register (SSICR_0)	H'FFFF C000	H'1C0B D553	DMEN = "1": DMA request is enabled. UIEN = "1": Underflow Interrupt is enabled. OIEN = "1": Overflow interrupt is enabled. IIEN = "0": Idle mode interrupt is disabled. CHNL = "B'00": One channel per system word DWL = "B'001": Data Word Length 16 bits SWL = "B'011": System Word Length 32 bits SCKD = "1": Serial bit clock is output. Master mode. SWSD = "1": Serial word is set as input. Master mode. SCKP = "0": Serial Bit Clock Polarity. SSIWS and SSIDATA change on the SSISCK falling edge. SWSP = "1": Serial WS Polarity, high for 1st channel, low for 2nd channel. SPDP = "0": Padding bits are low. SDTA = "1": Transmitting and receiving in the order of padding bits and serial data PDTA = "0": When a data word length is 16 bits, the PDTA setting is ignored. The first data word is held by bits 15 to 0 and the second data word is held by bits 31 to 16. DEL = "1": No delay between SSIWS and SSIDATA CKDV = "B'101": Oversampling clock frequency / 6 MUEN = "0": Module is not muted. TRMD = "1": SSI module is in transmit mode. EN = "1": SSI module is enabled.

Table 3 DMAC Register Settings Used in Sample Program

Register Name	Address	Setting Value	Description
Standby control register 2 (STBCR2)	H'FFFE 0018	H'00	MSTP8 = "0": DMAC is operational.
DMA channel control register_1 (CHCR1)	H'FFFE 101C	H'0000 0000	DE = "0": DMA transfer disabled
		H'2010 1814	TC = "0": Transmits data once. RLDSAR = "1": Enables the function to reload SAR. RLDDAR = "0": Disables the function to reload DAR. TEMASK = "1": *1 DMA transfer does not stop even if the TE bit is set. DM = "B'00": Fixed destination address SM = "B'01": Source address is incremented. RS = "B'1000": DMA extension resource selector TB = "0": Cycle steal mode TS = "B'10": Longword transfer IE = "1": Enables an interrupt request.
		H'2010 1815	DE = "1": DMA transfer enabled
DMA source address register_1 (SAR1)	H'FFFE 1010	On-chip RAM	Transfer source start address: Sets an area in the on-chip RAM.
DMA reload source address register_1 (RSAR1)	H'FFFE 1110	On-chip RAM	Transfer source start address: Sets an area in the on-chip RAM.
DMA destination address register_1 (DAR1)	H'FFFE 1014	H'FFFF C008	Transfer destination start address: SSI transmit data register (SSITDR_0)
DMA transfer count register_1 (DMATCR1)	H'FFFE 1018	H'0000 000A	Transfer count: 10 (H'0A)
DMA operation register (DMAOR)	H'FFFE 1200	H'0001	DME = "1": DMA transfer is enabled on all channels.
DMA extension resource selector 0 (DMARS0)	H'FFFE 1300	H'0023	SSI channel 0

Note: 1. PCM data must be output from the SSI at a constant timing.
When TEMASK is set to “0”, the DMA is disabled upon the completion of DMA transfer. Thus, the SSI might have an underflow if interrupt processing on completion of DMA transfer is delayed due to the period over which interrupts are disabled in the main routine and so on. To prevent this, we recommend the setting TEMASK = 1 so that DMA transfer can continue immediately after a previous round of DMA transfer is completed.

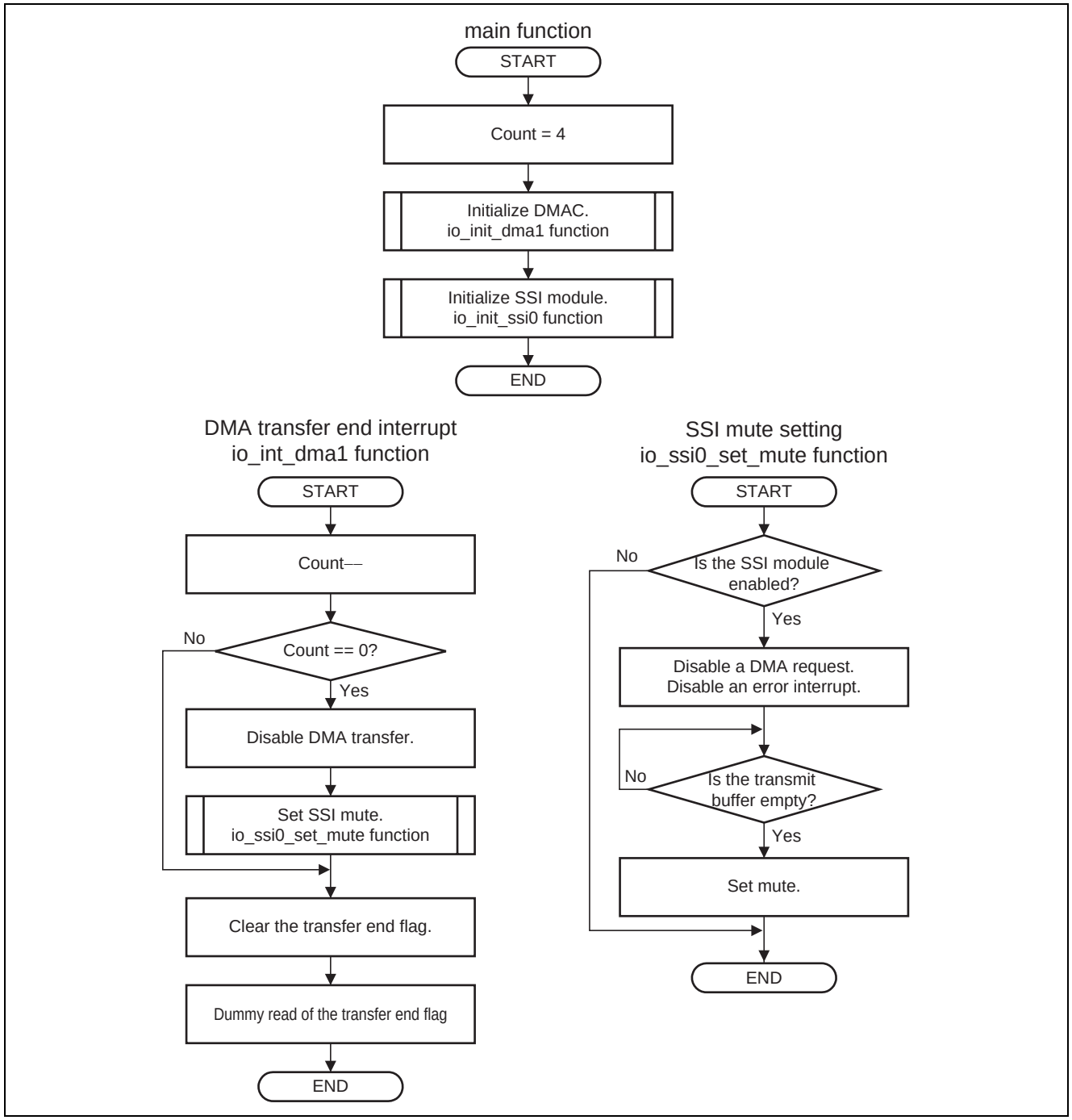

Figure 6 Flow of Processing by the Sample Program

2.5 Listing of the Sample Program

```
/*""FILE COMMENT""*****************************************************************
*
* System Name  : SH7203 Sample Program
* File Name    : main.c
* Contents     : SSI data transfer
* Version      : 1.00.00
* Model        : M3A-HS30
* CPU          : SH7203
* Compiler     : SHC9.1.1.0
* note         : A data transfer sample program using SSI0
*
*
*                 Note
*                 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) 2008 Renesas Technology Corp. All Rights Reserved
* AND Renesas Solutions Corp. All Rights Reserved
*
* history : 2008.04.25 ver.1.00.00
*""FILE COMMENT END""*************************************************************/
#include <string.h>
#include "iodefine.h"              /* iodefine.h is a file automatically created by HEW*/

/* ==== Macro declaration ==== */
/* ==== Set DMAC ==== */
#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_ssi0(void);
void io_ssi0_set_mute(void);
void io_init_dma1(void *src, void *dst, size_t size, unsigned int mode);

unsigned long Data[10] = {
            0x0000ffffu,0x0000ffffu,
            0x0000ffffu,0x0000ffffu,
            0x0000ffffu,0x0000ffffu,
            0x0000ffffu,0x0000ffffu,
            0x0000ffffu,0x0000ffffu};
unsigned int Count;

```

Figure 7 Sample Program Listing: main.c (1)

```
56    /*""FUNC COMMENT""*****************************************************
57    * Outline      : Sample program main
58    *-------------------------------------------------------------------------
59    * Include      : #include "iodefine.h"
60    *-------------------------------------------------------------------------
61    * Declaration  : void main(void);
62    *-------------------------------------------------------------------------
63    * Function     : Initializes the SSI module, and then transfers data.
64    *-------------------------------------------------------------------------
65    * Argument     : void
66    *-------------------------------------------------------------------------
67    * Return Value : void
68    *-------------------------------------------------------------------------
69    * Notice       :
70    *""FUNC COMMENT END""************************************************/
71    void main(void)
72    {
73        Count = 4;                                          /* DMA transfer count */
74
75        /* ==== Initialize DMAC and enable transfer ==== */
76      io_init_dma1( Data,                                   /* Source address */
77                      (void *)&SSI0.SSITDR,                 /* Destination address */
78                      sizeof(Data),                         /* Number of bytes */
79                      DMA_SIZE_LONG | DMA_INT_ENABLE);      /* 32 bits; interrupts enabled */
80
81        /* ==== Initialize SSI0 ==== */
82        io_init_ssi0();
83
84        while(1){
85               /* Program end */
86        }
87    }
```

Figure 8 Sample Program Listing: main.c (2)

```
 88    /*""FUNC COMMENT""*****************************************************
 89    * Outline      : SSI module initialization
 90    *-----------------------------------------------------------------------
 91    * Include      : #include "iodefine.h"
 92    *-----------------------------------------------------------------------
 93    * Declaration  : void io_init_ssi0(void);
 94    *-----------------------------------------------------------------------
 95    * Function     : Transfers data in master transmitter mode.
 96    *              : Sampling frequency is 44.1 kHz
 97    *-----------------------------------------------------------------------
 98    * Argument     : void
 99    *-----------------------------------------------------------------------
100    * Return Value : void
101    *-----------------------------------------------------------------------
102    * Notice       :
103    *""FUNC COMMENT END""*************************************************/
104    void io_init_ssi0(void)
105    {
106        /* ==== Supply clock to SSI module ==== */
107        CPG.STBCR6.BIT.MSTP67 = 0u;           /* SSI0 */
108
109        /* ==== Select an oversampling clock supply source ==== */
110        PORT.SCSR.BIT.SSI0CKS = 0u;           /* AUDIO_X1 input 16.9344 MHz */
111
112        /* ----SSI module pin enabled ---- */
113        PORT.PFCRH1.BIT.PF18MD = 1u;          /* SSISCK0 */
114        PORT.PFCRH1.BIT.PF19MD = 1u;          /* SSIWS0 */
115        PORT.PFCRH2.BIT.PF20MD = 1u;          /* SSIDATA0 */
116
117        /* ==== Control register (SSICR) ==== */
118        SSI0.SSICR.LONG = 0x1c0bd553ul;
119               /*
120                   bit31-29 : reserve 0
121                   bit28 : DMEN : 1----------------   DMA request is enabled
122                   bit27 : UIEN : 1----------------   Underflow interrupt is enabled
123                   bit26 : OIEN : 1----------------   Overflow interrupt is enabled
124                   bit25 : IIEN : 0----------------   Idle mode interrupt is disabled
125                   bit24 : DIEN : 0----------------   Data interrupt is disabled
126                   bit23-22 : CHNL : 0-------------   Having one channel per system word
127                   bit21-19 : DWL : B'001----------   Data word length 16 bits
128                   bit18-16 : SWL : B'011----------   System word length 32 bits
129                   bit15 : SCKD : 1----------------   Serial bit clock is output; master mode
130                   bit14 : SWSD : 1----------------   Serial word select is output; master mode
131                   bit13 : SCKP : 0----------------   SSIWS and SSIDATA change at the SSISCK rising edge
132                   bit12 : SWSP : 1----------------   SSIWS is high for first channel, and low for second channel
133                   bit11 : SPDP : 0----------------   Padding bits are low
134                   bit10 : SDTA : 1----------------   Transmitting and receiving padding bits and serial data in this order
135                   bit9 : PDTA : 0-----------------   Not used
136                   bit8 : DEL : 1------------------   No delay between SSIWS and SSIDATA
137                   bit7 : reserve 0
138                   bit6-4 : CKDV : B'101-----------   Oversampling clock frequency (16.9344 MHz) / 6 [44.1 kHz]
139                   bit3 : MUEN : 0-----------------   SSI module is not muted
140                   bit2 : reserve 0
141                   bit1 : TRMD : 1-----------------   SSI module is in transmit mode
142                   bit0 : EN : 1 ------------------   SSI module is enabled
143               */
144    }
145
```

Figure 9 Sample Program Listing: main.c (3)

```
146    /*""FUNC COMMENT""*****************************************************
147    * Outline      : SSI mute setting
148    *-----------------------------------------------------------------------
149    * Include      : #include "iodefine.h"
150    *-----------------------------------------------------------------------
151    * Declaration  : void io_ssi0_set_mute(void);
152    *-----------------------------------------------------------------------
153    * Function     : Shifts SSI to the mute state.
154    *-----------------------------------------------------------------------
155    * Argument     : void
156    *-----------------------------------------------------------------------
157    * Return Value : void
158    *-----------------------------------------------------------------------
159    * Notice       :
160    *""FUNC COMMENT END""*************************************************/
161    void io_ssi0_set_mute(void)
162    {
163        if(SSI0.SSICR.BIT.EN == 1ul){
164                /* ---- disable SSI interrupt ---- */
165                SSI0.SSICR.BIT.UIEN = 0ul;
166
167                /* ---- disable dreq ---- */
168                SSI0.SSICR.BIT.DMEN =0ul;
169
170                while(SSI0.SSISR.BIT.DIRQ == 0ul){
171                        /* ---- wait data req ---- */
172                }
173                SSI0.SSICR.BIT.MUEN = 1ul; /* mute start */
174        }
175    }
176    /*""FUNC COMMENT""*****************************************************
177    * Outline      : SSI interrupts
178    *-----------------------------------------------------------------------
179    * Include      : #include "iodefine.h"
180    *-----------------------------------------------------------------------
181    * Declaration  : void io_int_ssi0(void);
182    *-----------------------------------------------------------------------
183    * Function     : Processes SSI interrupts
184    *-----------------------------------------------------------------------
185    * Argument     : void
186    *-----------------------------------------------------------------------
187    * Return Value : void
188    *-----------------------------------------------------------------------
189    * Notice       :
190    *""FUNC COMMENT END""*************************************************/
191    void io_int_ssi0(void)
192    {
193        /* Underflow error */
194        if(SSI0.SSISR.BIT.UIRQ == 1ul){
195                SSI0.SSISR.BIT.UIRQ = 0ul;
196                while(1){
197                        /* dead loop */
198                }
199        }
200        /* Overflow error */
201        if(SSI0.SSISR.BIT.OIRQ == 1ul){
202                SSI0.SSISR.BIT.OIRQ = 0ul;
203                while(1){
204                        /* dead loop */
205                }
206        }
207        /* Idle mode */
208        if(SSI0.SSISR.BIT.IIRQ == 1ul){
209                SSI0.SSISR.BIT.IIRQ = 0ul;
210        }}
```

Figure 10 Sample Program Listing: main.c (4)

```
211     /*""FUNC COMMENT""*****************************************************
212     * Outline       : DMA transfer initial setting
213     *-----------------------------------------------------------------------
214     * Include       : #include "iodefine.h"
215     *-----------------------------------------------------------------------
216     * Declaration   : io_init_dma1(void *src, void *dst, size_t size, unsigned int mode);
217     *-----------------------------------------------------------------------
218     * Function      : Transfers data for the number of bytes specified by “size” from source
219     *               : address src to destination address dst using the DMAC.
220     *               :
221     *               : Specifies the transfer size and whether to use interrupts in “mode”.
222     *-----------------------------------------------------------------------
223     * Argument      : void *src : Source address
224     *               : void *dst : Destination address
225     *               : size_t size : Transfer size (byte)
226     *               : unsigned int mode : Transfer mode: The following modes are specified with logical OR.
227     *               :                     DMA_SIZE_BYTE(0x0000) Byte transfer
228     *               :                     DMA_SIZE_WORD(0x0001) Word transfer
229     *               :                     DMA_SIZE_LONG(0x0002) Longword transfer
230     *               :                     DMA_SIZE_LONGx4(0x0003) 16-byte transfer
231     *               :                     DMA_INT_DISABLE(0x0000) DMA transfer end interrupt is not used
232     *               :                     DMA_INT_ENABLE(0x0010) DMA transfer end interrupt is used
233     *-----------------------------------------------------------------------
234     * Return Value  : void
235     *-----------------------------------------------------------------------
236     * Notice        : If the transfer size and source/destination address alignment
237     *               : do not match, correct operation is not guaranteed.
238     *               : To use an interrupt, the interrupt routine should be registered.
239     *""FUNC COMMENT END""*************************************************/
240     void io_init_dma1(void *src, void *dst, size_t size, unsigned int mode)
241     {
242         unsigned int ts;
243         unsigned long ie;
244
245         ts = mode & 3u;
246         ie = (mode & 0x00f0u ) >> 4u;
247         /* ==== Set standby control register 2 (STBCR2) ==== */
248         CPG.STBCR2.BIT.MSTP8 = 0u;                      /* Cancel DMAC module stop */
249         /* ---- Set DMA channel control registers ---- */
250         DMAC.CHCR1.BIT.DE = 0ul;                        /* Disable DMA transfer */
251         /* ---- Set DMA source address register ---- */
252         DMAC.SAR1.LONG = (unsigned long)src;
253         /* ---- Set DMA reload address register ---- */
254         DMAC.RSAR1.LONG = (unsigned long)src;
255         /* ---- Set DMA destination address register ---- */
256         DMAC.DAR1.LONG = (unsigned long)dst;
257         /* ---- Set DMA reload destination address register ---- */
258         DMAC.RDAR1.LONG = (unsigned long)dst;
259         /* ---- Set DMA transfer count register ---- */
260         /* ---- Set DMA reload transfer count register ---- */
261         switch(ts){
262         case DMA_SIZE_BYTE:
263                 DMAC.DMATCR1.LONG = size;               /* Set transfer count (1/1) */
264                 DMAC.RDMATCR1.LONG = size;
265                 break;
266         case DMA_SIZE_WORD:
267                 DMAC.DMATCR1.LONG = size >> 1u;         /* Set transfer count (1/2) */
268                 DMAC.RDMATCR1.LONG = size >> 1u;
269                 break;
270         case DMA_SIZE_LONG:
271                 DMAC.DMATCR1.LONG = size >> 2u;         /* Set transfer count (1/4) */
272                 DMAC.RDMATCR1.LONG = size >> 2u;
273                 break;
274         case DMA_SIZE_LONGx4:
275                 DMAC.DMATCR1.LONG = size >> 4u;         /* Set transfer count (1/16) */
276                 DMAC.RDMATCR1.LONG = size >> 4u;
277                 break;
278         default:
279                 break;
280         }
```

Figure 11 Sample Program Listing: main.c (5)

```
281         /* ---- Set DMA channel control registers ---- */
282         DMAC.CHCR1.LONG = 0x20101800ul | (ts << 3u) | (ie << 2u) ;
283             /*
284                 bit31 : TC DMATCR transfer 0-------  1   Transfers data once
285                 bit30 : reserve 0
286                 bit29 : RLDSAR OFF : 1-------------      Enables the function to reload SAR
287                 bit28 : RLDDAR OFF : 0-------------      Disables the function to reload DAR
288                 bit27-24 : reserve 0
289                 bit23 : DO over run0 : 0-----------      Not used
290                 bit22 : TL TEND low active : 0-----      Not used
291                 bit21 : reserve 0
292                 bit20 : TEMASK :TE set mask : 1----      DMA transfer does not stop even if the TE bit is set
293                 bit19 : HE :0----------------------      Not used
294                 bit18 : HIE :0---------------------      Not used
295                 bit17 : AM :0----------------------      Not used
296                 bit16 : AL :0----------------------      Not used
297                 bit15-14 : DM1:0 DM0:0----------------   Fixed destination address
298                 bit13-12 : SM1:0 SM0:1----------------   Source address is incremented
299                 bit11-8 : RS : auto request : B'1000-    DMA extension resource selector
300                 bit7 : DL : DREQ level : 0 -------       Not used
301                 bit6 : DS : DREQ select :0 Low level     Not used
302                 bit5 : TB : cycle :0---------------      Cycle steal mode
303                 bit4-3 : TS : transfer size: B'10---     Longword unit
304                 bit2 : IE : interrupt enable: 1---       Enables an interrupt request
305                 bit1 : TE : transfer end----------
306                 bit0 : DE : DMA enable bit: 0-----      DMA transfer is disabled
307             */
308         /* ---- Set DMA extension resource selector 0  ---- */
309         DMAC.DMARS0.BIT.CH1MID = 0x08u;             /* MID = SSI0 */
310         DMAC.DMARS0.BIT.CH1RID = 0x03u;             /* RID */
311         /* ---- Set DMA operation register ---- */
312         DMAC.DMAOR.WORD &= 0xfff9u;                 /* Clear the AE and NMIF bits */
313
314         if(DMAC.DMAOR.BIT.DME == 0ul){              /* Enable DMA transfer on all channels */
315                 DMAC.DMAOR.BIT.DME = 1ul;
316         }
317         if(ie == 1ul){
318                 INTC.IPR06.BIT._DMAC1 = 1u;         /* Set the interrupt priority */
319         }
320         /* ---- Perform DMA transfer ---- */
321         DMAC.CHCR1.BIT.DE = 1ul;                    /* Enable DMA transfer */
322     }
323     /*""FUNC COMMENT""************************************************************
324     * Outline       : DMA transfer end interrupt
325     *-----------------------------------------------------------------------
326     * Include       : #include "iodefine.h"
327     *-----------------------------------------------------------------------
328     * Declaration   : void io_int_dma1(void);
329     *-----------------------------------------------------------------------
330     * Function      : Shifts the SSI to the mute state when the DMA transfers are executed for the specified count.
331     *-----------------------------------------------------------------------
332     * Argument      : void
333     *-----------------------------------------------------------------------
334     * Return Value  : void
335     *-----------------------------------------------------------------------
336     * Notice        :
337     *""FUNC COMMENT END""*************************************************/
338     void io_int_dma1(void)
339     {
340         volatile unsigned long dummy;
341
342         Count--;
343         if(Count == 0){                     /* Transfers are executed for the specified count */
344                 /* ---- Stop DMA transfer---- */
345                 DMAC.CHCR1.BIT.DE = 0ul;        /* Disable DMA1 transfer */
346                 io_ssi0_set_mute();             /* Set mute */
347         }
348
349         DMAC.CHCR1.BIT.TE = 0ul;                /* Clear the transfer end flag */
350         dummy = DMAC.CHCR1.BIT.TE;
351     }
352     /* End of File */
```

Figure 12 Sample Program Listing: main.c (6)

```
   1    /***********************************************************************/
   2    /*                                                                     */
   3    /* FILE :intprg.c                                                      */
   4    /* DATE :Tue, Nov 13, 2007                                             */
   5    /* DESCRIPTION :Interrupt Program                                      */
   6    /* CPU TYPE :SH7203                                                    */
   7    /*                                                                     */
   8    /* This file is generated by Renesas Project Generator (Ver.4.5).      */
   9    /*                                                                     */
  10    /***********************************************************************/
  11
  12
  13
  14    #include <machine.h>
  15    #include "vect.h"
  16    #pragma section IntPRG
  17
        (Snip)
 426
 427    // 112 DMAC1 TEI1
 428    void INT_DMAC1_TEI1(void)
 429    {
 430        extern void io_int_dma1(void);
 431        io_int_dma1();
 432    }
 433
        (Snip)
 983
 984    // 214 SSI0
 985    void INT_SSI0(void)
 986    {
 987        extern void io_int_ssi0(void);
 988        io_int_ssi0();
 989    }
 990
        (Snip)
1139
1140    /* End of File */
```

Figure 13 Sample Program Listing: intprg.c (1)

3. Documents for Reference

- Software Manual
  SH-2A, SH2A-FPU Software Manual (REJ09B0051)
  The most up-to-date version of this document is available on the Renesas Technology Website.

- Hardware Manual
  SH7203 Group Hardware Manual (REJ09B0313)
  The most up-to-date version of this document is available on the Renesas Technology Website.
  SH7263 Group Hardware Manual (REJ09B0290)
  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	Jan.19.09	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.