

## SH7268/SH7269 Group

R01AN1188EJ0100

Rev.1.00

Jul. 27, 2012

### Transmission/Reception of Renesas SPDIF Interface

---

#### Abstract

This document describes how to transmit/receive digital audio data using Renesas SPDIF interface of the SH7268/SH7269 group.

The features of the Renesas SPDIF interface are described below.

- Supports the IEC60958 standard (stereo and consumer use modes only)
- Supports sampling frequencies (44.1kHz, 48kHz and 32kHz)
- Simultaneous transmit and receive
- Receiver autodetects the IEC61937 compressed mode data

#### Products

SH7268/SH7269 Group

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

## Contents

|                                                            |    |
|------------------------------------------------------------|----|
| 1. Specifications .....                                    | 3  |
| 2. Operation Confirmation Conditions .....                 | 4  |
| 3. Reference Application Note .....                        | 4  |
| 4. Peripheral Functions .....                              | 5  |
| 4.1 Channel Configuration .....                            | 5  |
| 4.2 Transfer Data .....                                    | 5  |
| 4.3 Oversampling Clock and Sampling Frequency .....        | 6  |
| 5. Hardware .....                                          | 7  |
| 5.1 Pins Used .....                                        | 7  |
| 5.2 Hardware Configuration .....                           | 7  |
| 6. Software .....                                          | 8  |
| 6.1 Operation Overview .....                               | 8  |
| 6.2 File Composition .....                                 | 9  |
| 6.3 Constants .....                                        | 9  |
| 6.4 Structure/Union List .....                             | 10 |
| 6.5 Variables .....                                        | 10 |
| 6.6 Functions .....                                        | 10 |
| 6.7 Function Specifications .....                          | 11 |
| 6.8 Flowcharts .....                                       | 15 |
| 6.8.1 Main Processing .....                                | 15 |
| 6.8.2 Initialization of RSPDIF .....                       | 16 |
| 6.8.3 Starting of RSPDIF Transmission .....                | 17 |
| 6.8.4 Starting of RSPDIF Reception .....                   | 18 |
| 6.8.5 Updating Channel Status of RSPDIF Transmission ..... | 18 |
| 6.8.6 Initialization of DMAC Channel 4 .....               | 19 |
| 6.8.7 Initialization of DMAC Channel 5 .....               | 20 |
| 6.8.8 Interrupt Function of RSPDIF .....                   | 21 |
| 6.8.9 DMA Transfer Completion Interrupt .....              | 22 |
| 7. Sample Code .....                                       | 23 |
| 8. Reference Documents .....                               | 23 |

## 1. Specifications

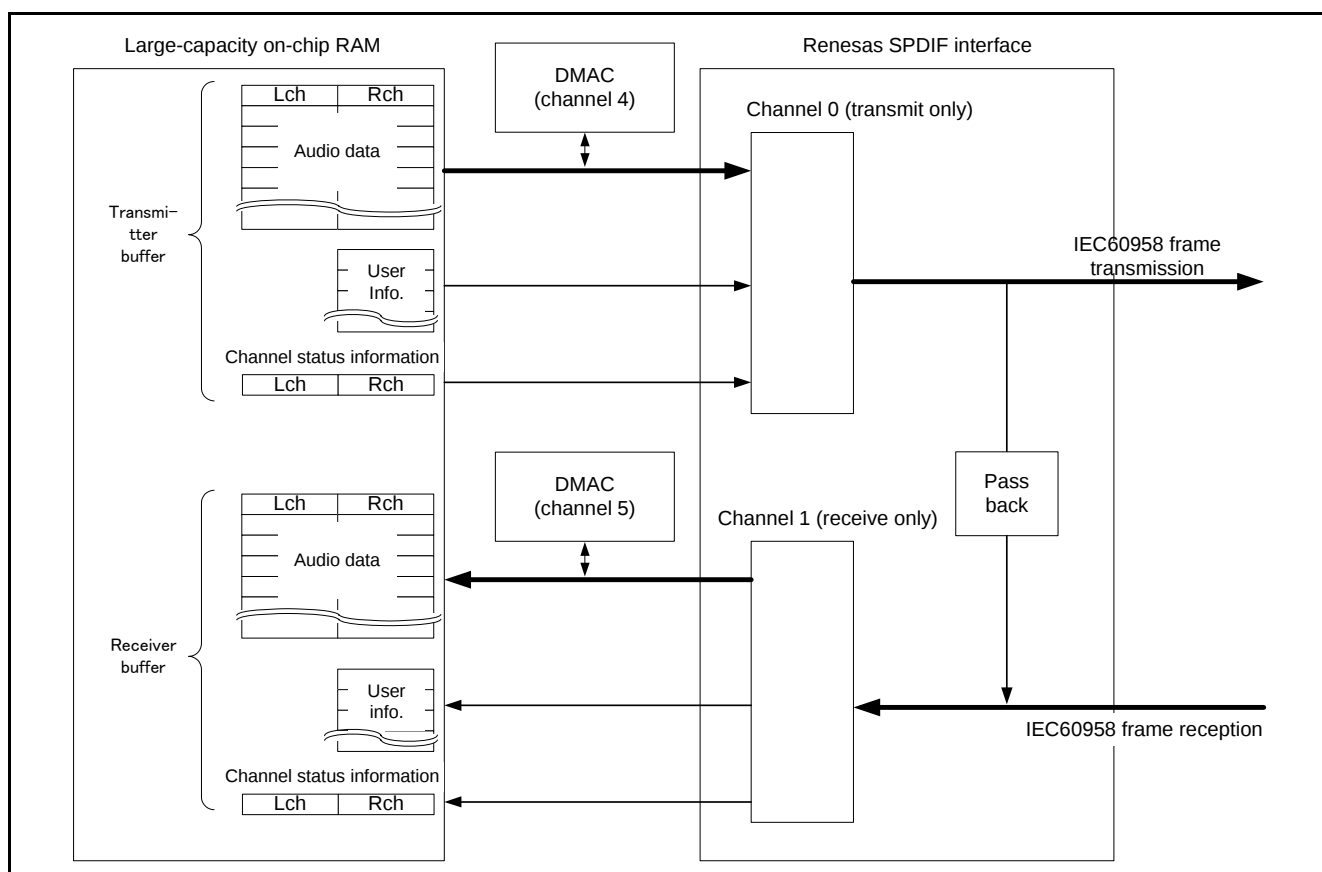
Input the transmitter buffer data stored in the large-capacity on-chip RAM to the Renesas SPDIF interface (hereinafter called RSPDIF) channel 0, and transmits the frame which complies with the IEC60958 standard (hereinafter called IEC60958 frame). Interrupts are used for transferring the user information and channel status information, and Direct Memory Access Controller (hereinafter called DMAC) is used for transferring the audio data.

The IEC60958 frame transmitted from the RSPDIF channel 0 is transferred to the RSPDIF channel 1 by the pass back function. The data received in the RSPDIF channel 1 will be stored in the large-capacity on-chip RAM by the interrupt or the DMAC as well as transmitting the data.

Table 1.1 lists the Peripheral Functions and Their Applications and Figure 1.1 shows the Operation Overview.

**Table 1.1 Peripheral Functions and Their Applications**

| Peripheral Function                    | Application                                               |
|----------------------------------------|-----------------------------------------------------------|
| Renesas SPDIF interface (RSPDIF)       | Transmits and receives the SPDIF                          |
| Direct Memory Access Controller (DMAC) | Transfers audio data                                      |
| Interrupt controller (INTC)            | Transfers user information and channel status information |
| Large-capacity on-chip RAM             | Transmitter buffer and receiver buffer area               |



**Figure 1.1 Operation Overview**

## 2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

**Table 2.1 Operation Confirmation Conditions**

| Item                               | Contents                                                                                                                                                                                                                                                         |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MCU used                           | SH7269                                                                                                                                                                                                                                                           |
| Operating frequency                | CPU internal clock (I $\phi$ ): 266.67MHz<br>Internal clock (B $\phi$ ): 133.33MHz<br>Peripheral clock 1 (P1 $\phi$ ): 66.67MHz<br>Peripheral clock 0 (P0 $\phi$ ): 33.33MHz                                                                                     |
| Operating voltage                  | Source Power (I/O): 3.3V<br>Source Power (internal): 1.25V                                                                                                                                                                                                       |
| Integrated development environment | Renesas Electronics Corporation<br>High-performance Embedded Workshop Ver. 4.07.00                                                                                                                                                                               |
| C compiler                         | Renesas Electronics Corporation<br>SuperH RISC engine Family C/C++ compiler package<br>Ver.9.03 Release 02                                                                                                                                                       |
|                                    | Compile option<br>-cpu=sh2afpu -fpu=single. -include="\$(WORKSPDIR)\inc"<br>-object="\$(CONFIGDIR)\\$(FILELEAF).obj" -debug -gbr=auto<br>-chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0<br>-del_vacant_loop=0 -struct_alloc=1 -nologo |
| Board used                         | R0K572690C000BR                                                                                                                                                                                                                                                  |

## 3. Reference Application Note

For additional information associated with this document, refer to the following application note.

- SH7262/SH7264 Group Example of Initialization (RJ06B0998)

## 4. Peripheral Functions

This chapter provides supplementary information on RSPDIF, which basic information is described in the hardware manual.

### 4.1 Channel Configuration

The RSPDIF consists of two channels. The directions of transmission/reception for each channel are fixed. The channel 0 is used for transmission, and the channel 1 is used for reception.

### 4.2 Transfer Data

In addition to the audio data which includes sound data, user information and channel status information will also be transmitted and received in the SPDIF. Therefore, it is necessary to write/read these data to/from the RSPDIF register. Figure 4.1 shows the Register Group for Reading/Writing Transfer Data. The length of 32-bit is required.

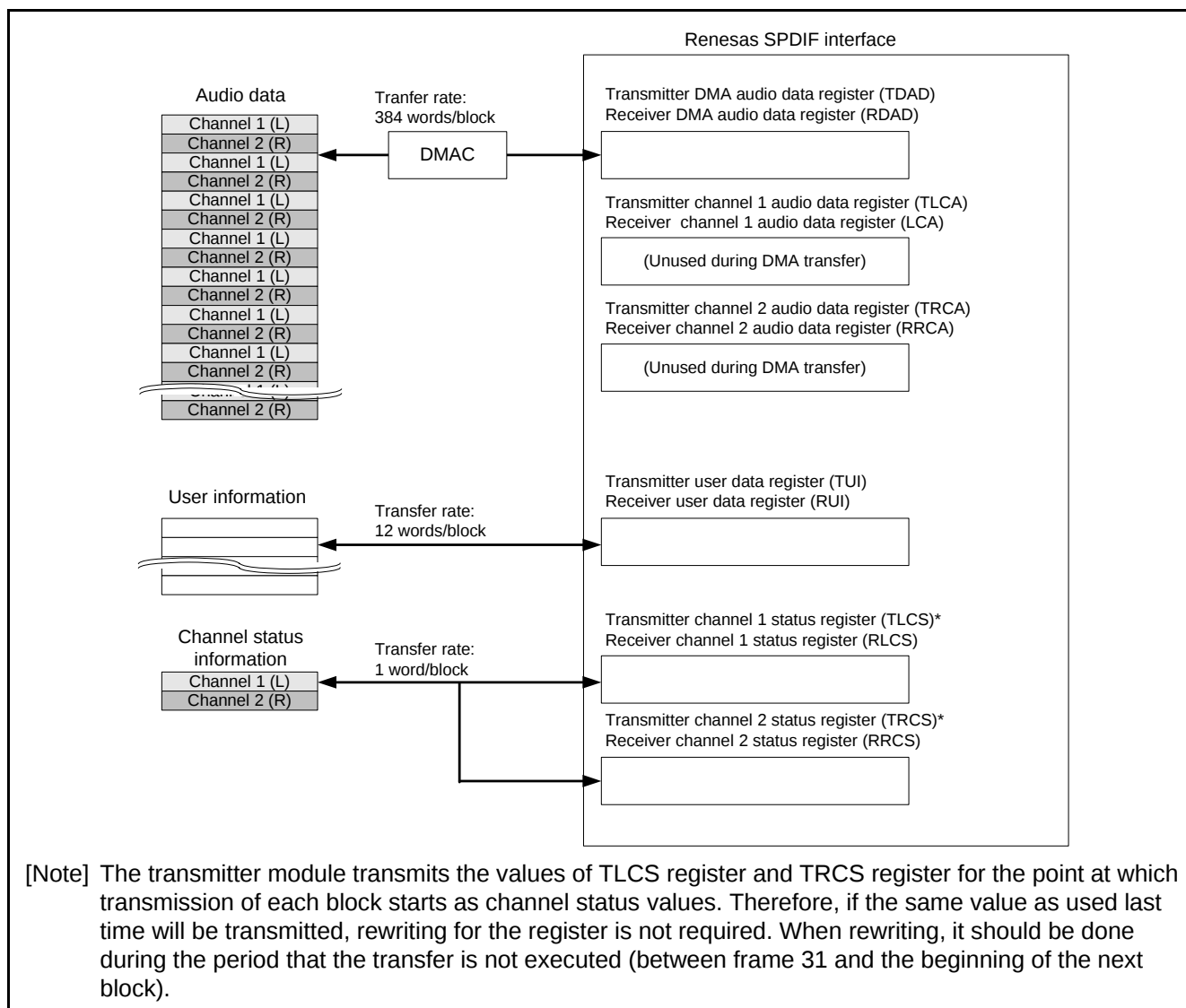


Figure 4.1 Register Group for Reading/Writing Transfer Data

### 4.3 Oversampling Clock and Sampling Frequency

It is necessary to input oversampling clock from the outside to use the RSPDIF. Refer to Chapter 5 "Hardware" in this application note regarding the oversampling clock input terminals.

One sample (= one frame) consists of two channels, each of which contains 32 bits, and the oversample with a frequency of eight times as large as the sampling frequency is required in the RSPDIF. For these reasons, the frequency for oversampling clock supplied from the outside needs to be set to 512 times larger than the sampling frequency. Table 4.1 lists the Frequency of Oversampling Clock over Sampling Frequency.

**Table 4.1 Frequency of Oversampling Clock over Sampling Frequency**

| Sampling frequency of audio data | Oversampling clock frequency |
|----------------------------------|------------------------------|
| 48 kHz                           | 24.5760 MHz                  |
| 44.1 kHz                         | 22.5792 MHz                  |
| 32 kHz                           | 16.3840 MHz                  |

[Note] The oscillator on the used board (R0K572690C000BR) listed in Table 2.1 "Operation Confirmation Conditions" cannot be applied to the oversampling clock shown in Table 4.1. Please take this into account when referring to a configuration.

## 5. Hardware

### 5.1 Pins Used

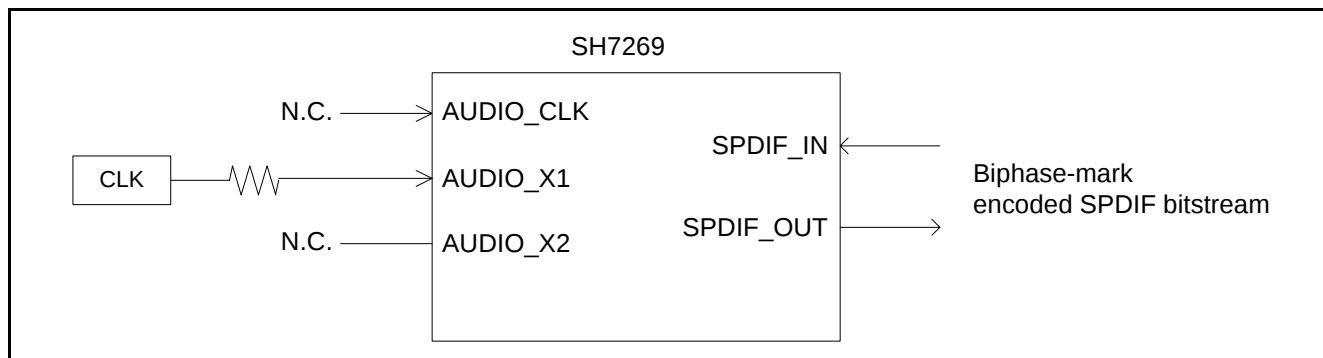
Table 5.1 shows the Pins Used and Their Functions.

**Table 5.1 Pins Used and Their Functions**

| Pin Name  | I/O    | Function                                           |
|-----------|--------|----------------------------------------------------|
| AUDIO_CLK | Input  | Not connected                                      |
| AUDIO_X1  | Input  | Input oversampling clock                           |
| AUDIO_X2  | Output | Not connected (used when connected to oscillation) |
| SPDIF_IN  | Input  | Input SPDIF bitstream                              |
| SPDIF_OUT | Output | Output SPDIF bitstream                             |

### 5.2 Hardware Configuration

Figure 5.1 shows the Connection Example.



**Figure 5.1 Connection Example**

## 6. Software

### 6.1 Operation Overview

Figure 6.1 shows the Sequence Diagram of Sample Code. The CPU initializes the RSPDIF and allows DMA transfer of the audio data, and then the transmission will be started. The reception will also be started at the same time by the pass back function. The audio data is transferred continuously without software by the DMAC continual function. The user information and the channel status information are transferred by the software using interrupts. However, the same value shall be used for the channel status information of the transfer without rewriting by the interrupt.

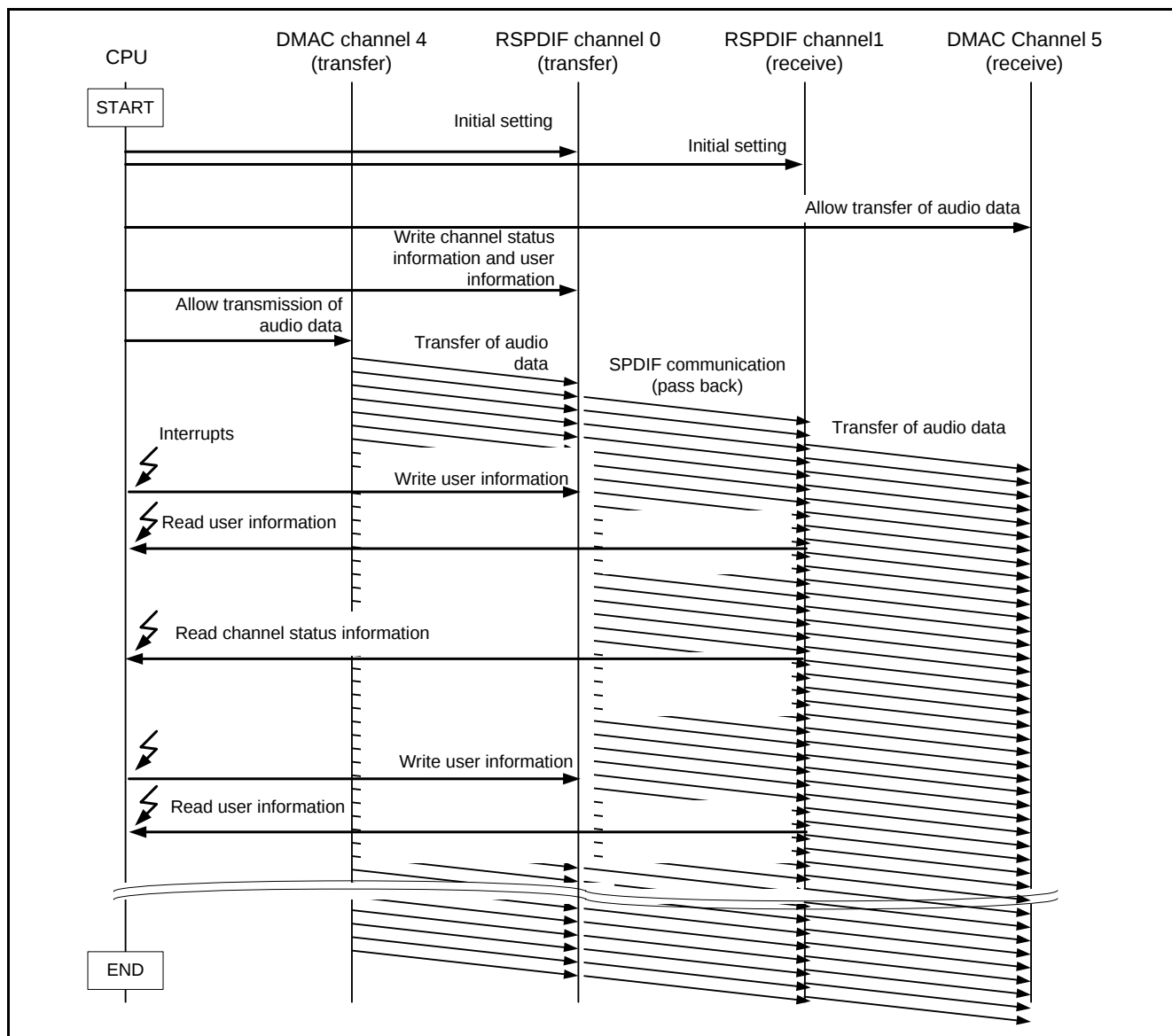


Figure 6.1 Sequence Diagram of Sample Code

## 6.2 File Composition

Table 6.1 lists the Files Used in the Sample Code. Files not generated by the integrated development environment should not be listed in this table.

**Table 6.1 Files Used in the Sample Code**

| File Name   | Outline                                 | Remarks |
|-------------|-----------------------------------------|---------|
| main.c      | Main program                            |         |
| io_rspdif.c | Initialization of the RSPDIF            |         |
| io_rspdif.h | Interface definition of the io_rspdif.c |         |
| io_dmac.c   | Initialization of the DMAC              |         |
| io_dmac.h   | Interface definition of the io_dmac.c   |         |

## 6.3 Constants

Table 6.2 lists the Constants Used in the Sample Code.

**Table 6.2 Constants Used in the Sample Code**

| Constant Name     | Setting Value  | Contents                                                    |
|-------------------|----------------|-------------------------------------------------------------|
| USER_NUMOF_BLOCK  | 1024           | User buffer size (by the block)                             |
| SPDIF_OK          | 0              | Return value when the processing was successful             |
| SPDIF_ERR         | -1             | Return value when the processing was failed                 |
| SPDIF_PASS_BACK   | 1              | Setting value of the RSPDIF pass back function (1: allowed) |
| SPDIF_NUMOF_FRM   | 192            | Number of frames in one block                               |
| SPDIF_NUMOF_CH    | 2              | Number of channels (left channel, right channel)            |
| SPDIF_AUDIO_BUFSZ | 192 * 2        | Buffer size for the audio data (by the longword)            |
| SPDIF_USER_BUFSZ  | (192 * 2) / 32 | Buffer size for the user information (by the longword)      |
| SPDIF_CH1         | 0              | Channel 1 (left channel) index                              |
| SPDIF_CH2         | 1              | Channel 2 (right channel) index                             |
| SPDIF_STATUS_CH1  | H'10100000     | Setting value of channel 1 status to be transmitted         |
| SPDIF_STATUS_CH2  | H'10200000     | Setting value of channel 2 status to be transmitted         |
| SPDIF_UBO_BIT     | H'00002000     | UBO bit of the status register                              |
| SPDIF_UBU_BIT     | H'00001000     | UBU bit of the status register                              |
| SPDIF_CE_BIT      | H'00000800     | CE bit of the status register                               |
| SPDIF_PARE_BIT    | H'00000400     | PARE bit of the status register                             |
| SPDIF_PREE_BIT    | H'00000200     | PREE bit of the status register                             |
| SPDIF_CSE_BIT     | H'00000100     | CSE bit of the status register                              |
| SPDIF_ABO_BIT     | H'00000080     | ABO bit of the status register                              |
| SPDIF_ABU_BIT     | H'00000040     | ABU bit of the status register                              |
| SPDIF_RUIR_BIT    | H'00000020     | RUIR bit of the status register                             |
| SPDIF_TUIR_BIT    | H'00000010     | TUIR bit of the status register                             |
| SPDIF_CSRX_BIT    | H'00000008     | CSRX bit of the status register                             |
| SPDIF_CBRX_BIT    | H'00000004     | CBRX bit of the status register                             |
| SPDIF_CSTX_BIT    | H'00000002     | CSTX bit of the status register                             |
| SPDIF_CBTX_BIT    | H'00000001     | CBTX bit of the status register                             |

## 6.4 Structure/Union List

Figure 6.2 shows the Structure/Union Used in the Sample Code.

```
typedef struct
{
    uint32_t    a_buf [SPDIF_AUDIO_BUFSZ];    /* Audio data buffer */
    uint32_t    u_buf [SPDIF_USER_BUFSZ];     /* User information buffer */
    int32_t     u_idx;                         /* User information index */
    uint32_t    s_buf [SPDIF_NUMOF_CH];       /* Channel status information buffer */

}spdif_t;
```

**Figure 6.2 Structure/Union Used in the Sample Code**

## 6.5 Variables

Table 6.3 lists the Static Variables.

**Table 6.3 Static Variables**

| Type             | Variable Name | Contents                 | Function Used                       |
|------------------|---------------|--------------------------|-------------------------------------|
| volatile spdif_t | g_spdout      | SPDIF transmitter buffer | main, io_spdout_start, io_spdif_isr |
| volatile spdif_t | g_spdin       | SPDIF receiver buffer    | main, io_spdin_start, io_spdif_isr  |

## 6.6 Functions

Table 6.4 lists the Functions.

**Table 6.4 Functions**

| Function Name           | Outline                                                       |
|-------------------------|---------------------------------------------------------------|
| main                    | Main processing                                               |
| io_init_rspdif          | Initialization of the RSPDIF                                  |
| io_spdout_start         | Starting of the RSPDIF transmission                           |
| io_spdin_start          | Starting of the RSPDIF reception                              |
| io_spdout_change_status | Updating channel status of the RSPDIF transmission            |
| io_rspdif_isr           | Interrupt function of the RSPDIF                              |
| io_init_dmac4           | Initialization of the DMAC channel 4                          |
| io_init_dmac5           | Initialization of the DMAC channel 5                          |
| io_reload_dmac4         | Reload register setting for the DMAC channel 4                |
| io_reload_dmac5         | Reload register setting for the DMAC channel 5                |
| io_dmac4_dei4_isr       | Transfer completion interrupt function for the DMAC channel 4 |
| io_dmac5_dei5_isr       | Transfer completion interrupt function for the DMAC channel 5 |

## 6.7 Function Specifications

The following tables list the sample code function specifications.

|                     |                                                                                                                               |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------|
| main                |                                                                                                                               |
| <b>Outline</b>      | Main processing                                                                                                               |
| <b>Header</b>       | io_rspdif.h                                                                                                                   |
| <b>Declaration</b>  | void main(void)                                                                                                               |
| <b>Description</b>  | Initializes RSPDIF transmitter module and receiver module to start operation.<br>Continues RSPDIF transmission and reception. |
| <b>Arguments</b>    | None                                                                                                                          |
| <b>Return Value</b> | None                                                                                                                          |

|                     |                                                     |
|---------------------|-----------------------------------------------------|
| io_init_rspdif      |                                                     |
| <b>Outline</b>      | Initialization of the RSPDIF                        |
| <b>Header</b>       | io_rspdif.h                                         |
| <b>Declaration</b>  | void io_init_rspdif (void)                          |
| <b>Description</b>  | Initializes transmitter module and receiver module. |
| <b>Arguments</b>    | None                                                |
| <b>Return Value</b> | None                                                |

|                     |                                                                                                |
|---------------------|------------------------------------------------------------------------------------------------|
| io_spdout_start     |                                                                                                |
| <b>Outline</b>      | Starting of the RSPDIF transmission                                                            |
| <b>Header</b>       | io_rspdif.h                                                                                    |
| <b>Declaration</b>  | void io_spdout_start(void)                                                                     |
| <b>Description</b>  | Writes data to transmitter module and allows the DMA transfer to start the SPDIF transmission. |
| <b>Arguments</b>    | None                                                                                           |
| <b>Return Value</b> | None                                                                                           |

|                     |                                                                   |
|---------------------|-------------------------------------------------------------------|
| io_spdin_start      |                                                                   |
| <b>Outline</b>      | Starting of the RSPDIF reception                                  |
| <b>Header</b>       | io_rspdif.h                                                       |
| <b>Declaration</b>  | void io_spdin_start(void)                                         |
| <b>Description</b>  | Allows receiver module DMA transfer to start the SPDIF reception. |
| <b>Arguments</b>    | None                                                              |
| <b>Return Value</b> | None                                                              |

---

|                         |                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| io_spdout_change_status |                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Outline</b>          | Updating channel status of the RSPDIF transmission                                                                                                                                                                                                                                                                                                                                               |
| <b>Header</b>           | io_rspdif.h                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>Declaration</b>      | int32_t io_spdout_change_status (void)                                                                                                                                                                                                                                                                                                                                                           |
| <b>Description</b>      | Executes this function when changing the transmitter channel status information. This function shall be executed after setting the value to the g_spdout.s_buf. It is necessary to update the channel status information register between frame 31 and frame 192. Therefore, this function provides only interrupt allowance, and register updating shall be executed by the interrupt function. |
| <b>Arguments</b>        | None                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Return Value</b>     | SPDIF_OK (0): succeeded<br>SPDIF_ERR (-1): failed (due to being in the process of updating)                                                                                                                                                                                                                                                                                                      |

---

|                     |                                                                                                                                                                                                               |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| io_rspdif_isr       |                                                                                                                                                                                                               |
| <b>Outline</b>      | Interrupt function of the RSPDIF                                                                                                                                                                              |
| <b>Header</b>       | None                                                                                                                                                                                                          |
| <b>Declaration</b>  | void io_rspdif_isr (void)                                                                                                                                                                                     |
| <b>Description</b>  | This function will be executed when the RSPDIF interrupt is received. Verifies interrupt source by this function, and executes error processing, transfer of user information and channel status information. |
| <b>Arguments</b>    | None                                                                                                                                                                                                          |
| <b>Return Value</b> | None                                                                                                                                                                                                          |

---

|                        |                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>io_init_dmac4</b>   |                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Outline</b>         | Initialization of DMAC channel 4                                                                                                                                                                                                                                                                                                                                  |
| <b>Header</b>          | io_dmac.h                                                                                                                                                                                                                                                                                                                                                         |
| <b>Declaration</b>     | void io_init_dmac4(void * src, void * dest, int32_t count)                                                                                                                                                                                                                                                                                                        |
| <b>Description</b>     | Sets the DMAC channel 4, and allows the DMA transfer from source address to destination address specified by the argument. Allows register reload, and continues to transfer after the transfer has been completed. Transfer completion interrupt is allowed. The RSPDIF transmitter DMA audio data register shall be specified for transfer destination address. |
| <b>Arguments</b>       | void *src : Transfer source address<br>void *dest : Transfer destination address<br>int32_t count : Number of transfers                                                                                                                                                                                                                                           |
| <b>Return Value</b>    | None                                                                                                                                                                                                                                                                                                                                                              |
| <b>io_init_dmac5</b>   |                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Outline</b>         | Initialization of DMAC channel 5                                                                                                                                                                                                                                                                                                                                  |
| <b>Header</b>          | io_dmac.h                                                                                                                                                                                                                                                                                                                                                         |
| <b>Declaration</b>     | void io_init_dmac5(void * src, void * dest, int32_t count)                                                                                                                                                                                                                                                                                                        |
| <b>Description</b>     | Set the DMAC channel 5, and allows the DMA transfer from source address to destination address specified by the argument. Allows register reload, and continues to transfer after the transfer has been completed. Transfer completion interrupt is allowed. The RSPDIF receiver DMA audio data register shall be specified for transfer source address.          |
| <b>Arguments</b>       | void *src : Transfer source address<br>void *dest : Transfer destination address<br>int32_t count : Number of transfers                                                                                                                                                                                                                                           |
| <b>Return Value</b>    | None                                                                                                                                                                                                                                                                                                                                                              |
| <b>io_reload_dmac4</b> |                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Outline</b>         | Reload register setting for the DMAC channel 4                                                                                                                                                                                                                                                                                                                    |
| <b>Header</b>          | io_dmac.h                                                                                                                                                                                                                                                                                                                                                         |
| <b>Declaration</b>     | void io_reload_dmac4(void * rsrc, void * rdest, int32_t rcount)                                                                                                                                                                                                                                                                                                   |
| <b>Description</b>     | Sets the reload register for the DMAC channel 4.                                                                                                                                                                                                                                                                                                                  |
| <b>Arguments</b>       | void *rsrc : Transfer source address to be reloaded<br>void *rdest : Transfer destination address to be reloaded<br>int32_t rcount : Number of transfers to be reloaded                                                                                                                                                                                           |
| <b>Return Value</b>    | None                                                                                                                                                                                                                                                                                                                                                              |
| <b>io_reload_dmac5</b> |                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Outline</b>         | Reload register setting for the DMAC channel 5                                                                                                                                                                                                                                                                                                                    |
| <b>Header</b>          | io_dmac.h                                                                                                                                                                                                                                                                                                                                                         |
| <b>Declaration</b>     | void io_reload_dmac5(void * rsrc, void * rdest, int32_t rcount)                                                                                                                                                                                                                                                                                                   |
| <b>Description</b>     | Sets the reload register for the DMAC channel 5.                                                                                                                                                                                                                                                                                                                  |
| <b>Arguments</b>       | void *rsrc : Transfer source address to be reloaded<br>void *rdest : Transfer destination address to be reloaded<br>int32_t rcount : Number of transfers to be reloaded                                                                                                                                                                                           |
| <b>Return Value</b>    | None                                                                                                                                                                                                                                                                                                                                                              |

---

**io\_dmac4\_dei4\_isr**

---

|                     |                                                                                                                      |
|---------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>Outline</b>      | Transfer completion interrupt function for the DMAC channel 4                                                        |
| <b>Header</b>       | io_dmac.h                                                                                                            |
| <b>Declaration</b>  | void io_dmac4_dei4_isr(void)                                                                                         |
| <b>Description</b>  | Executes this function and clears the transfer completion flag after the DMAC channel 4 transfer has been completed. |
| <b>Arguments</b>    | None                                                                                                                 |
| <b>Return Value</b> | None                                                                                                                 |

---

**io\_dmac5\_dei5\_isr**

---

|                     |                                                                                                                      |
|---------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>Outline</b>      | Transfer completion interrupt function for the DMAC channel 5                                                        |
| <b>Header</b>       | io_dmac.h                                                                                                            |
| <b>Declaration</b>  | void io_dmac5_dei5_isr(void)                                                                                         |
| <b>Description</b>  | Executes this function and clears the transfer completion flag after the DMAC channel 5 transfer has been completed. |
| <b>Arguments</b>    | None                                                                                                                 |
| <b>Return Value</b> | None                                                                                                                 |

## 6.8 Flowcharts

### 6.8.1 Main Processing

Figure 6.3 shows the Main Processing.

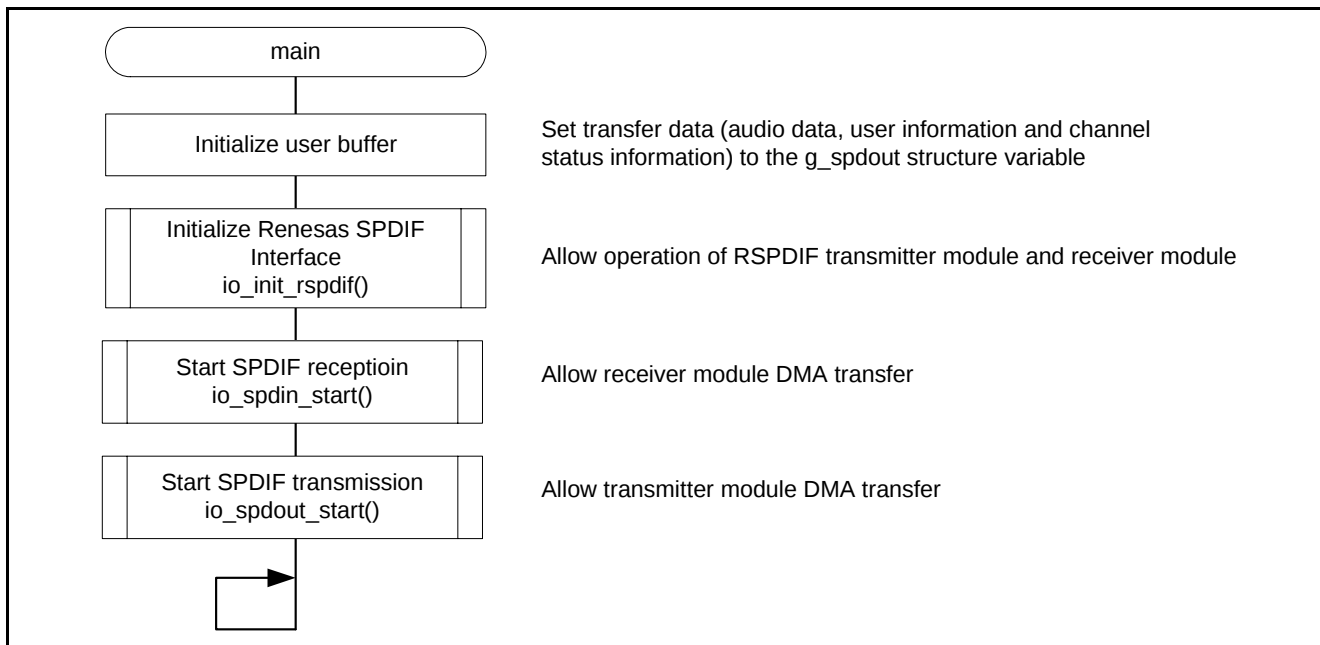
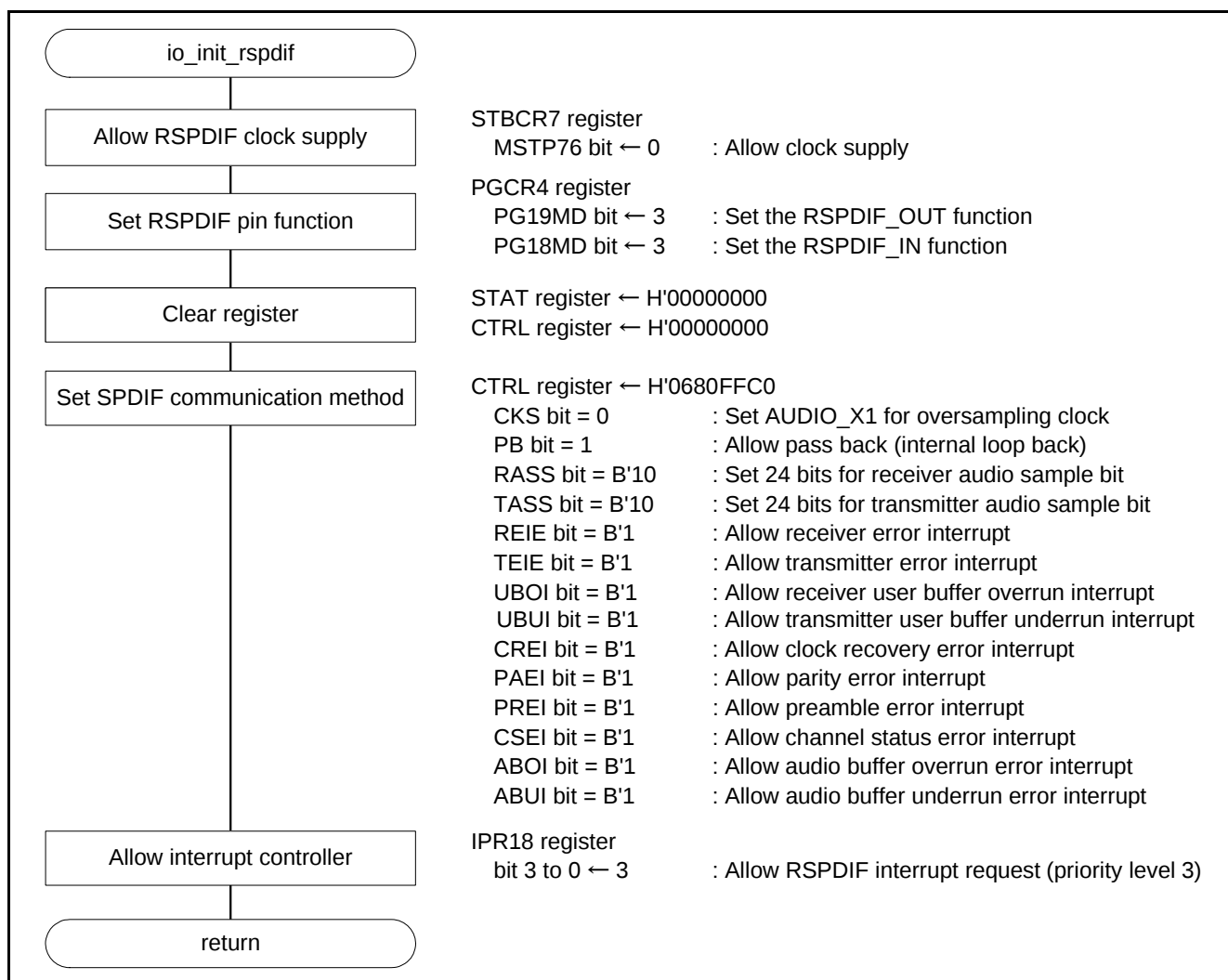


Figure 6.3 Main Processing

### 6.8.2 Initialization of RSPDIF

Figure 6.4 shows the Initialization of RSPDIF.



**Figure 6.4 Initialization of RSPDIF**

### 6.8.3 Starting of RSPDIF Transmission

Figure 6.5 shows the Starting of RSPDIF Transmission.

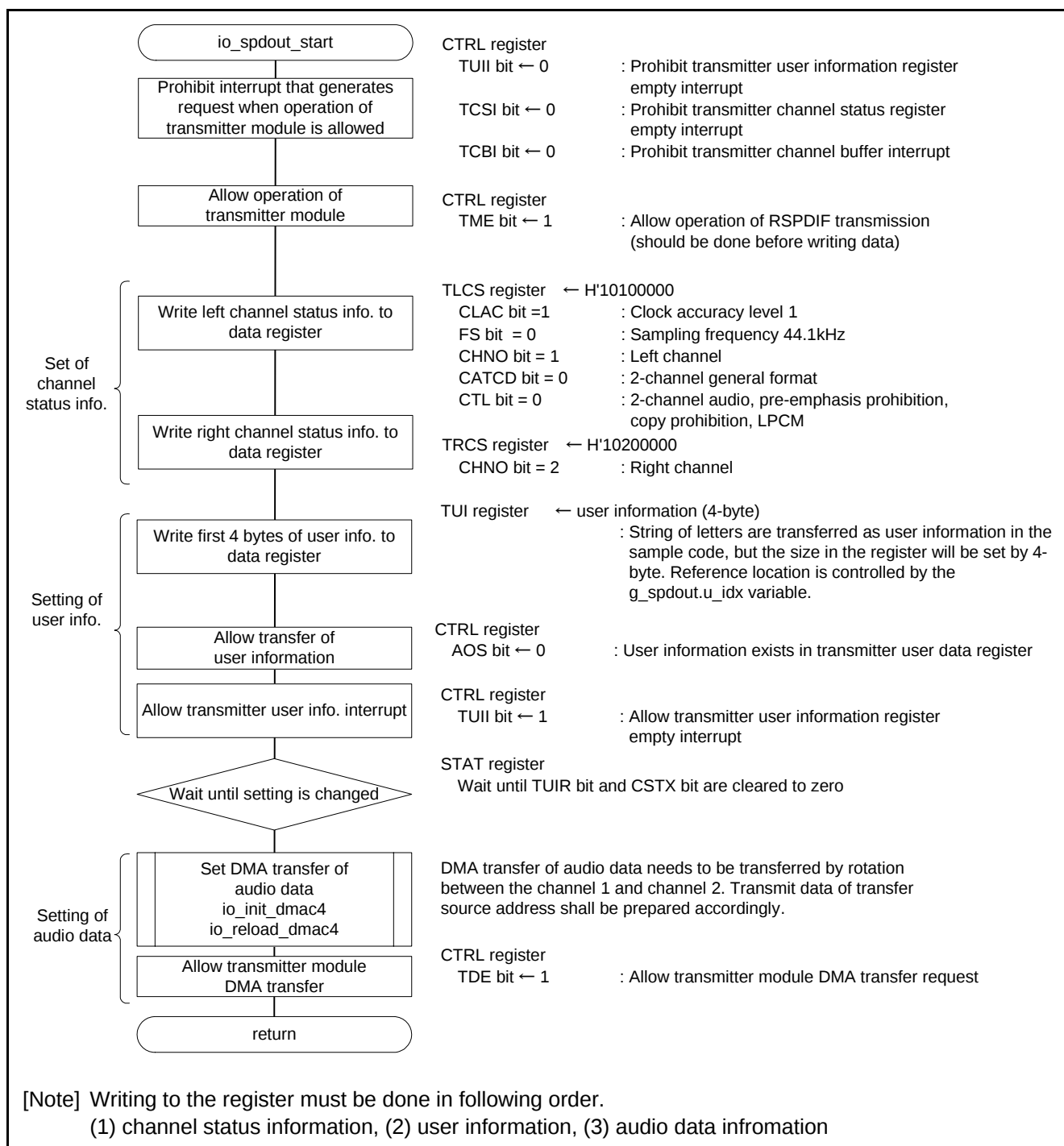


Figure 6.5 Starting of RSPDIF Transmission

#### 6.8.4 Starting of RSPDIF Reception

Figure 6.6 shows the Starting of RSPDIF Reception.

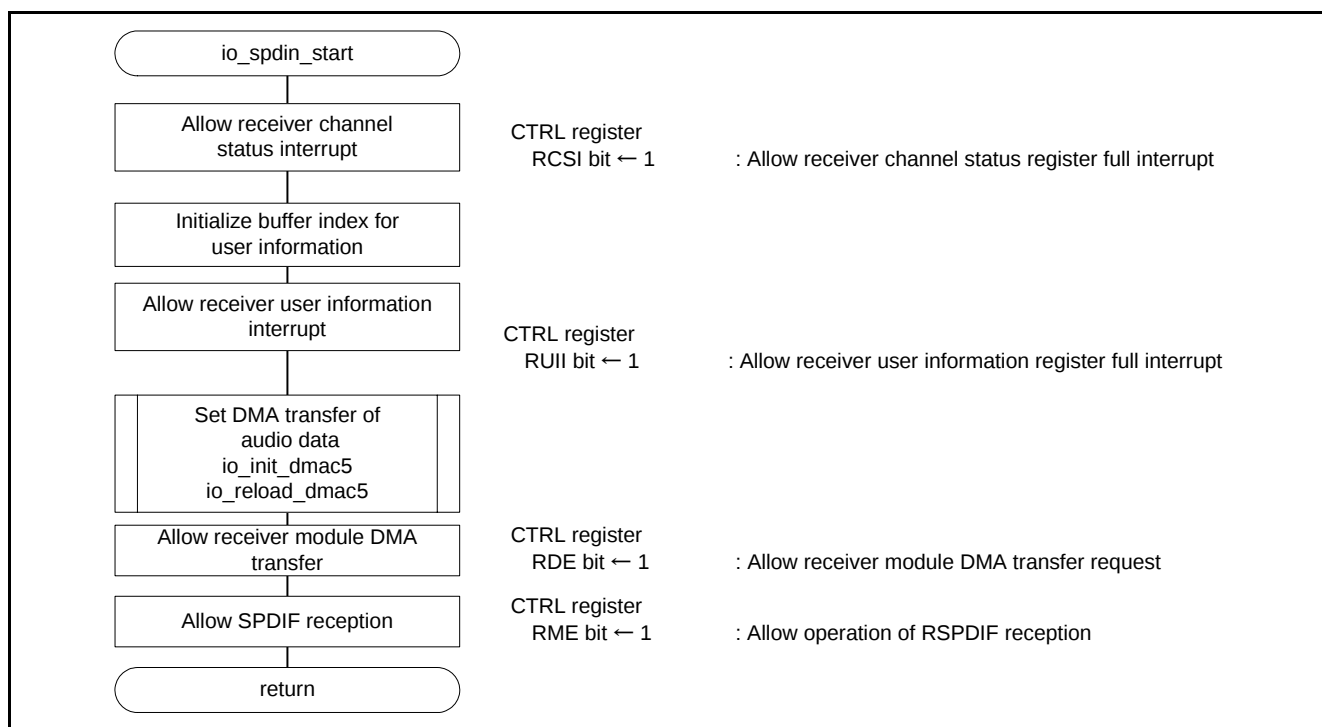


Figure 6.6 Starting of RSPDIF Reception

#### 6.8.5 Updating Channel Status of RSPDIF Transmission

Figure 6.7 shows the Updating Channel Status of RSPDIF Transmission.

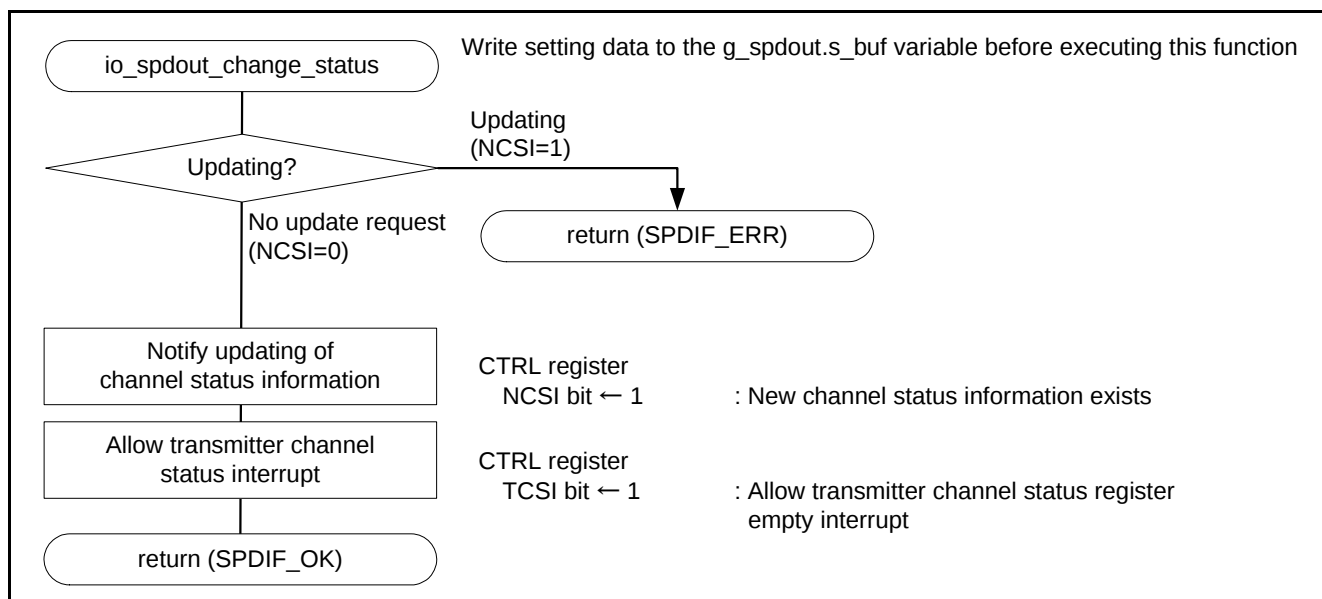


Figure 6.7 Updating Channel Status of RSPDIF Transmission

### 6.8.6 Initialization of DMAC Channel 4

Figure 6.8 shows the Initialization of DMAC Channel 4.

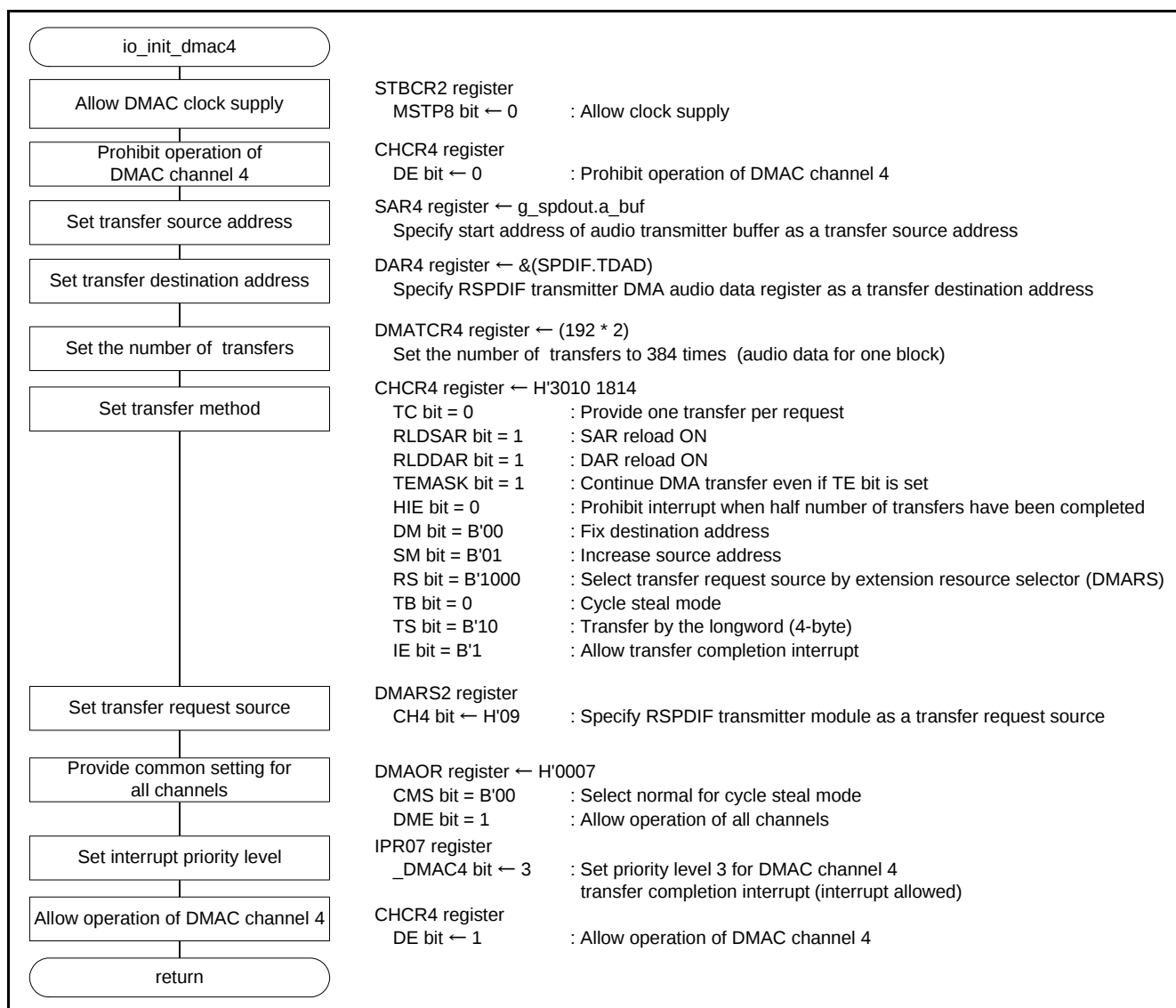


Figure 6.8 Initialization of DMAC Channel 4

### 6.8.7 Initialization of DMAC Channel 5

Figure 6.9 shows the Initialization of DMAC Channel 5.

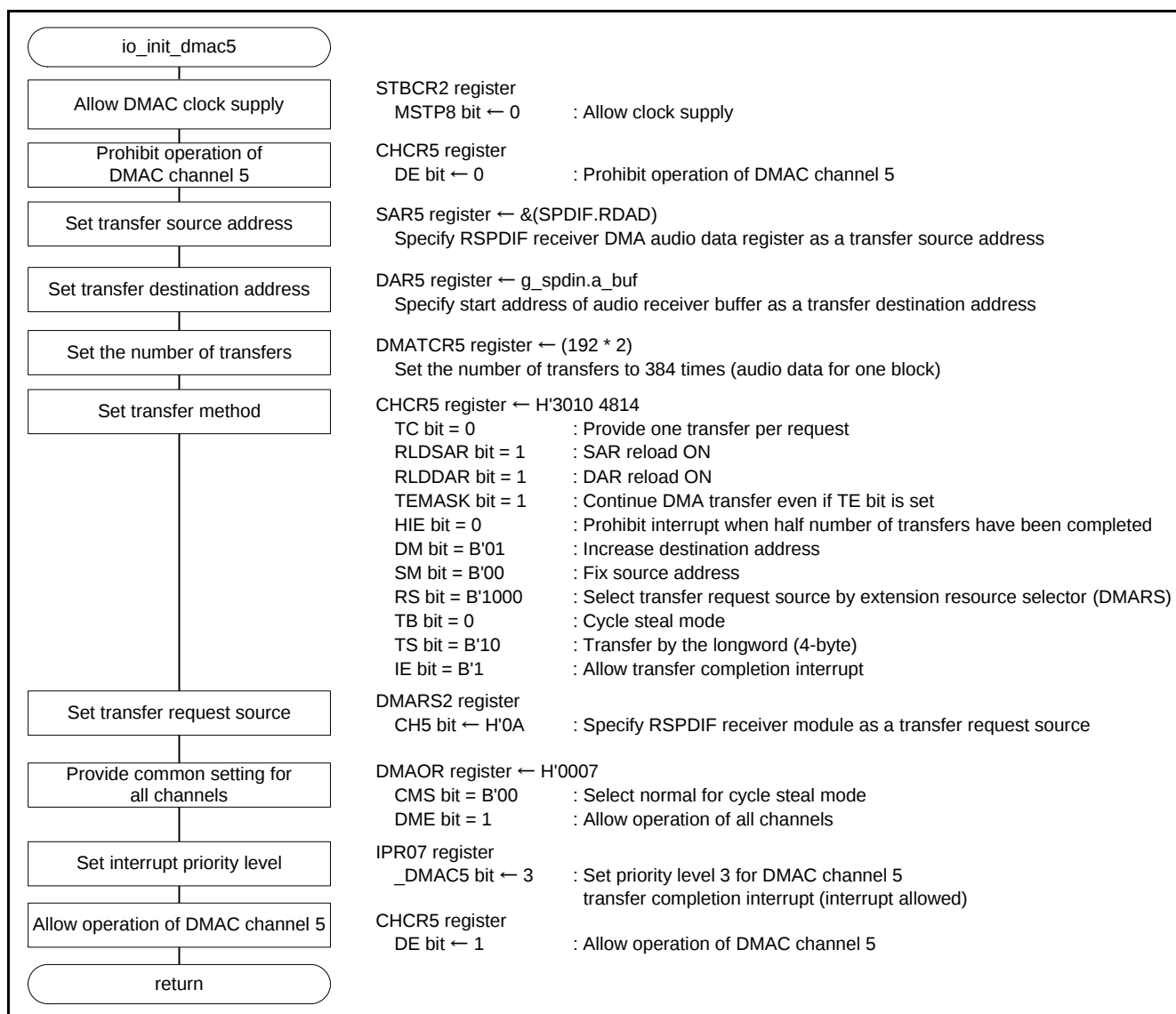


Figure 6.9 Initialization of DMAC Channel 5

### 6.8.8 Interrupt Function of RSPDIF

Figure 6.10 shows the Interrupt Function of RSPDIF.

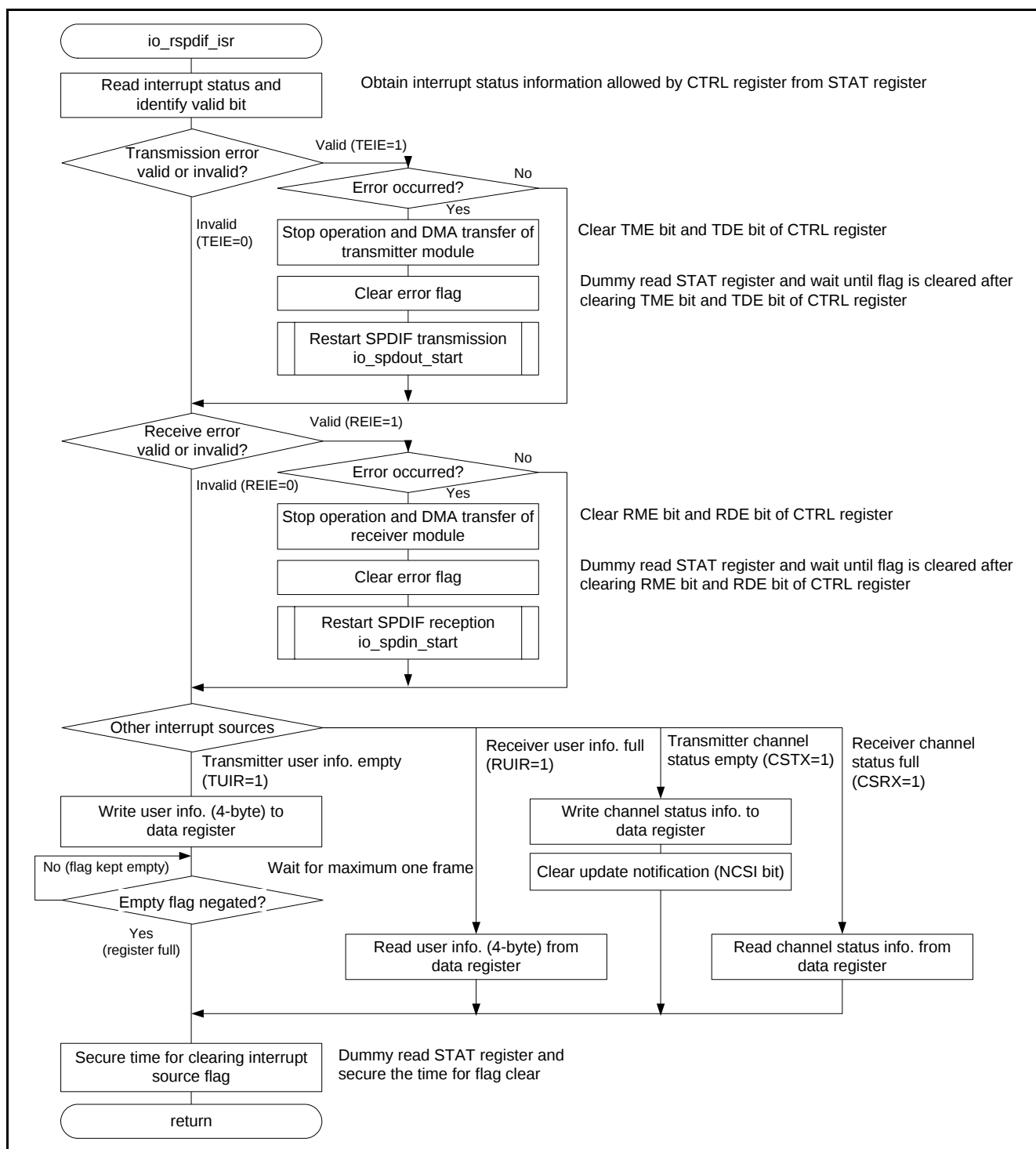


Figure 6.10 Interrupt Function of RSPDIF

6.8.9 DMA Transfer Completion Interrupt

Figure 6.11 shows the DMA Transmission Completion Interrupt.

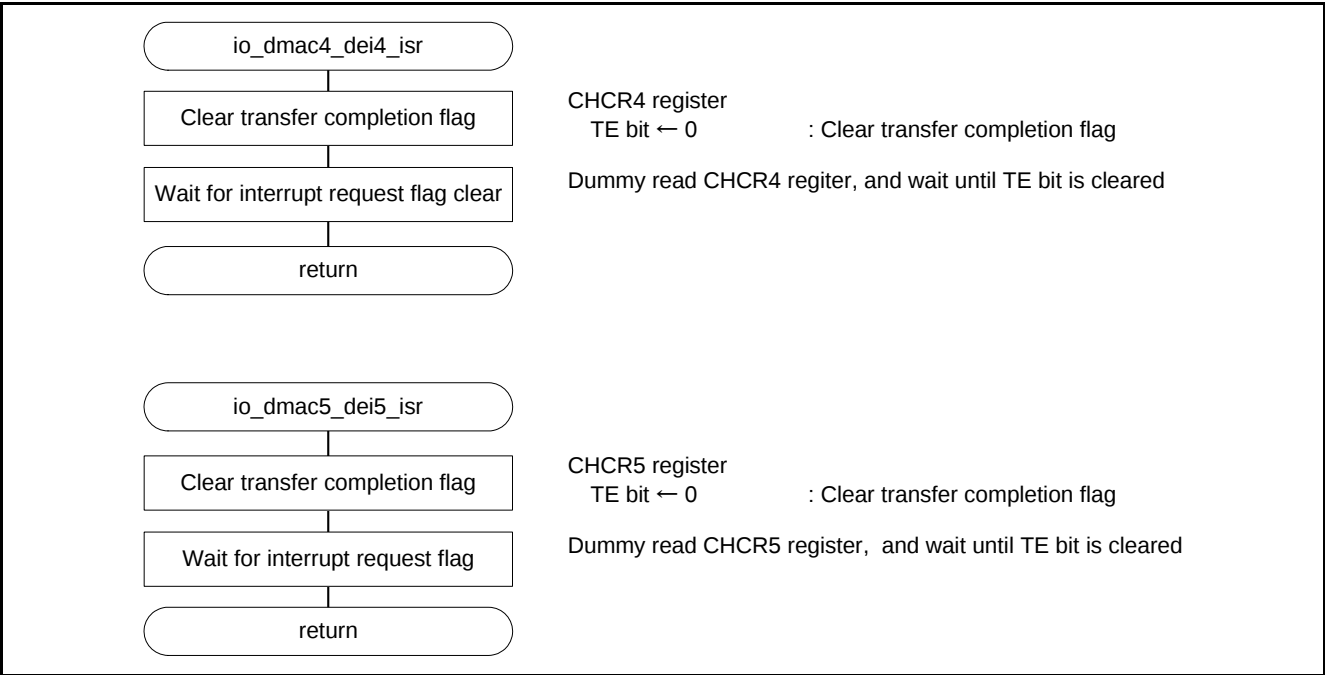


Figure 6.11 DMA Transmission Completion Interrupt

## 7. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

## 8. Reference Documents

User's Manual: Hardware

SH7268/SH7269 Group User's Manual: Hardware Rev.1.00

The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

C Compiler Manual

SuperH RISC engine C/C++ Compiler Package V.9.04

C Compiler User's Manual Rev.1.01. ([R number-not including the revision classification])

The latest version can be downloaded from the Renesas Electronics website.

## Website and Support

Renesas Electronics website

<http://www.renesas.com>

Inquiries

<http://www.renesas.com/contact/>

|                         |                                                                                   |
|-------------------------|-----------------------------------------------------------------------------------|
| <b>REVISION HISTORY</b> | SH7268/SH7269 Group Application Note<br>Transmission/Reception of SPDIF Interface |
|-------------------------|-----------------------------------------------------------------------------------|

| Rev.        | Date                 | Description |                      |
|-------------|----------------------|-------------|----------------------|
|             |                      | Page        | Summary              |
| <b>1.00</b> | <b>Jul. 27, 2012</b> | —           | First edition issued |
|             |                      |             |                      |

|                                                                                       |
|---------------------------------------------------------------------------------------|
| All trademarks and registered trademarks are the property of their respective owners. |
|---------------------------------------------------------------------------------------|

## General Precautions in the Handling of MPU/MCU Products

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

### 1. Handling of Unused Pins

Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

### 2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

### 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

### 4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

### 5. Differences between Products

Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.

- The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

## Notice

1. 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.
2. 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.
3. 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.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.  
"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 etc.  
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.  
Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). 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 for which it is not intended. 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 which the product is not intended by Renesas Electronics.
6. 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.
7. 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 systems manufactured by you.
8. 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.
9. 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. You should not use Renesas Electronics products or 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. When exporting the Renesas Electronics 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.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
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.



### SALES OFFICES

### Renesas Electronics Corporation

<http://www.renesas.com>

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

#### **Renesas Electronics America Inc.**

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

#### **Renesas Electronics Canada Limited**

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

#### **Renesas Electronics Europe Limited**

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

#### **Renesas Electronics Europe GmbH**

Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-65030, Fax: +49-211-6503-1327

#### **Renesas Electronics (China) Co., Ltd.**

7th Floor, Quantum Plaza, No.27 ZhichunLu Haidian District, Beijing 100083, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

#### **Renesas Electronics (Shanghai) Co., Ltd.**

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

#### **Renesas Electronics Hong Kong Limited**

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

#### **Renesas Electronics Taiwan Co., Ltd.**

13F, No. 363, Fu Shing North Road, Taipei, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

#### **Renesas Electronics Singapore Pte. Ltd.**

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

#### **Renesas Electronics Malaysia Sdn.Bhd.**

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

#### **Renesas Electronics Korea Co., Ltd.**

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