

---

# RX62N Group, RX621 Group

R01AN0631EJ0100

Rev.1.00

Sep 27, 2011

## I<sup>2</sup>S Communication Using RSPI, DTCa, and MTU2

---

### Introduction

This application note describes a method for transferring audio data by I<sup>2</sup>S communication, using the serial peripheral interface (RSPI), data transfer controller (DTCa), and multi-function timer pulse unit 2 (MTU2) of a RX62N Group or RX621 Group microcontroller (MCU).

### Target Devices

RX62N Group and RX621 Group MCUs

In order to use the sample code with another MCU it must be modified to match the specifications of the target device and its operation tested and evaluated thoroughly.

### Contents

1. Specifications .....	2
2. Operation Confirmation Conditions .....	3
3. Hardware .....	3
4. Software .....	5
5. Sample Code.....	58
6. Reference Documents.....	58

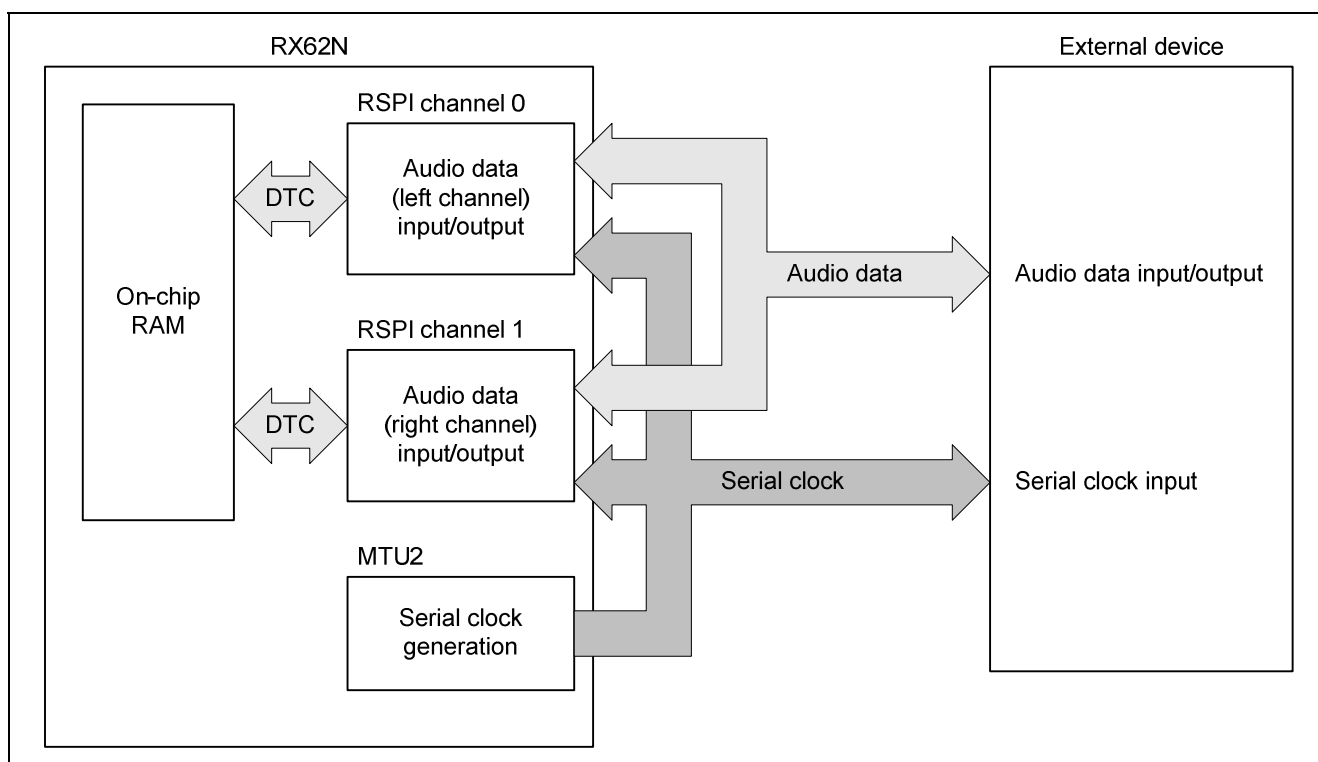
### 1. Specifications

The sample program uses the RSPI, DTCa, and MTU2 to transmit and receive audio data by I<sup>2</sup>S communication. The RSPI performs serial transfer of audio data according to the clock signal generated by the MTU2.

Table 1.1 lists the peripheral functions used by the sample program and their applications, and figure 1.1 shows a block diagram.

**Table 1.1 Peripheral Functions and Their Applications**

Peripheral Function	Application
RSPI channel 0	Audio data (left channel) input/output
RSPI channel 1	Audio data (right channel) input/output
MTU2 channel 2	Generation of serial transfer clock (SCK)
MTU2 channel 3	Generation of word select signal (WS)
MTU2 channel 4	Generation of slave select signal (SSL) to RSPI
DTCa	Transfer of audio data to and from on-chip RAM



**Figure 1.1 Block Diagram**

## 2. Operation Confirmation Conditions

The conditions used for confirming the operation of the sample code included in the application example are listed below.

**Table 2.1 Operation Confirmation Conditions**

Item	Description
MCU	RX62N Group
Device	HD74LV32A (Renesas Electronics)
Operating frequencies	<ul style="list-style-type: none"> <li>Input clock: 12 MHz</li> <li>CPU clock: 96 MHz</li> <li>Peripheral module clock: 48 MHz</li> </ul>
Power supply voltage	3.3 V
Integrated development environment	Renesas Electronics High-performance Embedded Workshop Version 4.08.00.011
C compiler	Renesas Electronics C/C++ compiler package for RX family V.1.00 Release 01 Compile option -cpu=rx600 -lang=c -output=obj="\$(CONFIGDIR)\\$(FILELEAF).obj" debug -nologo (The default settings of the integrated development environment are used.)
Operating mode	Single-chip mode
Sample code version	Version 1.00
Board	Renesas Starter Kit (R0K5562N0S000BE)

## 3. Hardware

### 3.1 List of Pins

Table 3.1 lists the pins used by the sample program.

**Table 3.1 List of Pins**

Pin	Input/Output	Description
P24_MTCLKA-A	Input	External clock (12.288 MHz) input
P26_MTIOC2A	Output	Serial transfer clock (SCK) output
P17_MTIOC3A	Output	Word select signal (WS) output
P82_MTIOC4A-B	Output	Slave select signal (SSL) output
PC4_SSLA0-A	Input	RSPI0 slave select signal input
PC5_RSPCKA-A	Input	RSPI0 serial transfer clock input
PC6_MOSIA-A	Input	Audio data (left channel) input
PC7_MISOA-A	Output	Audio data (left channel) output* <sup>1</sup>
PE4_SSLB0-B	Input	RSPI1 slave select signal input
PE5_RSPCKB-B	Input	RSPI1 serial transfer clock input
PE6_MOSIB-B	Input	Audio data (right channel) input
PE7_MISOB-B	Output	Audio data (right channel) output* <sup>1</sup>

Note: 1. This pin is not used during receive-only operation and is set to high-impedance.

### 3.2 Connections to Peripheral Devices

Figure 3.1 shows a connection diagram. Table 3.2 lists the clock signals generated by the MTU2.

The MTU2 uses the external clock (12.288 MHz) as the count clock and generates the clock signals listed in table 3.2.

RSPI channel 0 (RSPI0) is used to transmit and receive the left channel (L-ch) portion of the audio data and RSPI channel 1 (RSPI1) the right channel (R-ch) portion. The RSPI operates in slave mode and switches to the active channel according to the SSL signal.

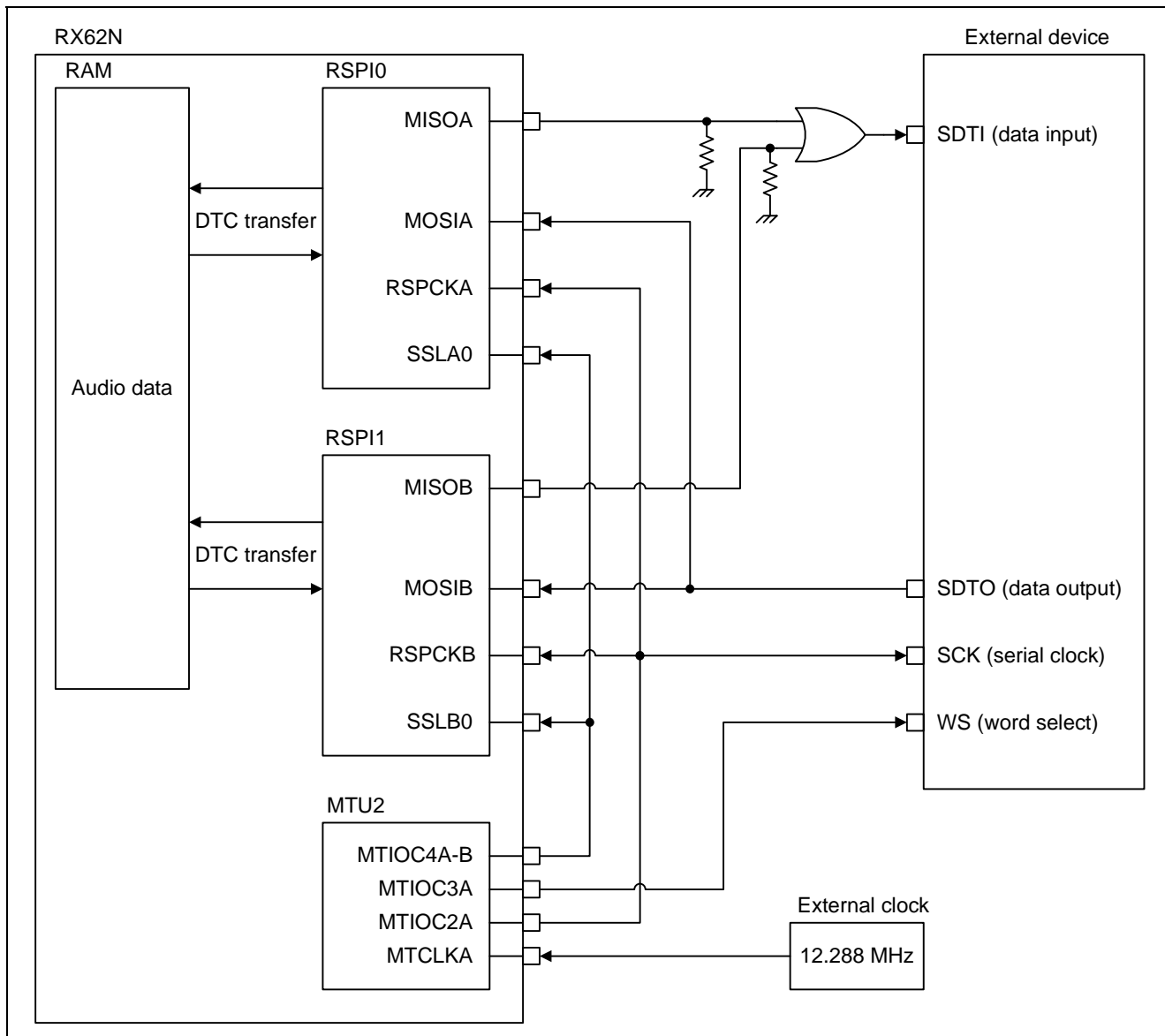


Figure 3.1 Connection Diagram

Table 3.2 Clock Signals Generated by MTU2

Channel	Clock Signal	Symbol	Output Pin	Frequency
Channel 2	Serial transfer clock	SCK	MTIOC2A	3.072 MHz
Channel 3	Word select signal	WS	MTIOC3A	48 kHz
Channel 4	RSPI slave select signal	SSL	MTIOC4A-B	48 kHz



(b) Transmit/Receive Data Format

The sample program performs serial transfers using a data length of 32 bits and MSB-first bit ordering. SCK is used as the serial transfer clock and WS as the channel select signal. The L-ch portion of the audio data is transferred when WS is low-level, and the R-ch portion is transferred when WS is high-level.

The position where padding is added to the transmit/receive data is selected according to the audio interface format of the external device. The sample program allows selection from among the following three formats.

- Standard format
- Backward-padding format
- Forward-padding format

Figure 4.2 shows the transmit/receive data formats.

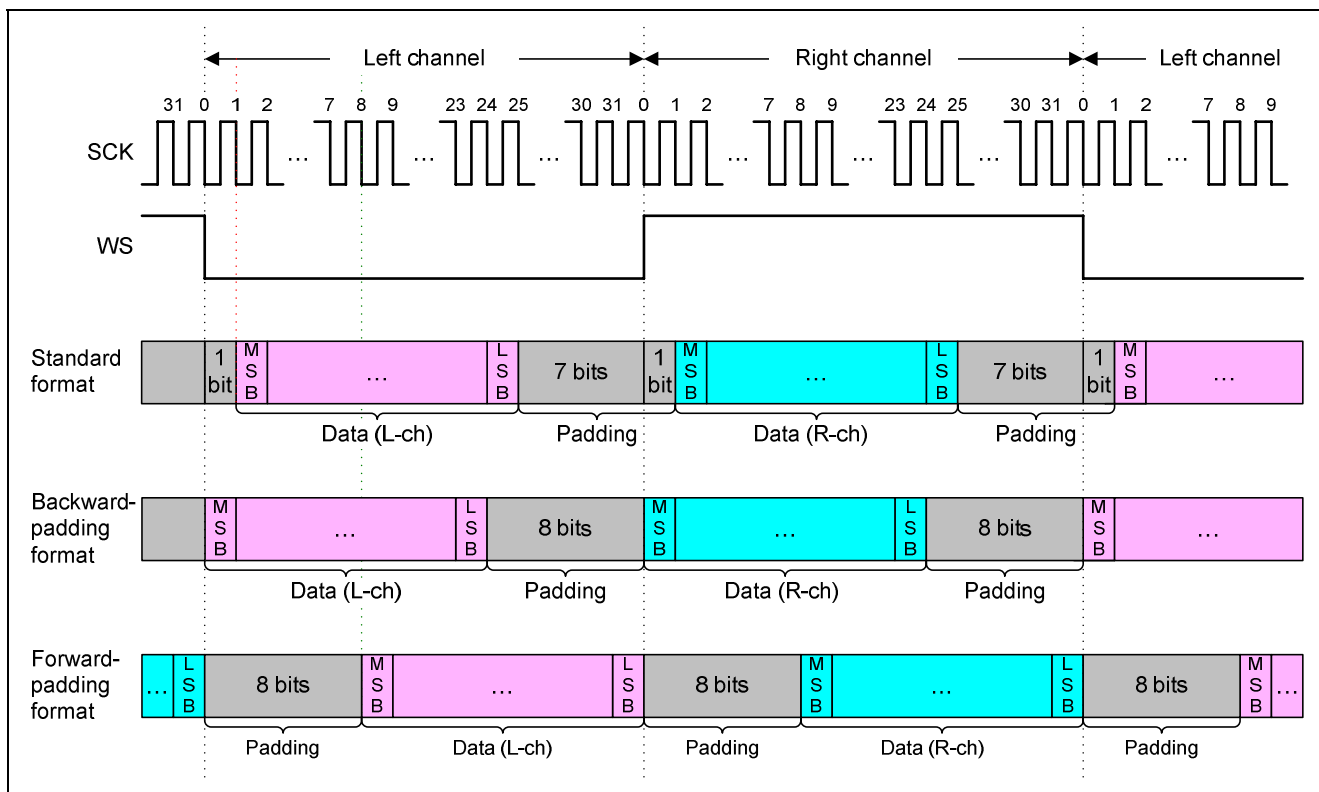


Figure 4.2 Transmit/Receive Data Formats

(2) Transmitting/Receiving Audio Data

In I<sup>2</sup>S transmission, audio data located in the on-chip RAM is transferred to the RSPI by the DTC and then output by the RSPI. In I<sup>2</sup>S reception, audio data received by the RSPI is transferred to the on-chip RAM by the DTC.

The RSPI operates in slave mode (SPI operation) using a data length of 32 bits and MSB-first bit ordering.

Figure 4.3 illustrates the flow of audio data.

The audio data is separated into L-ch and R-ch portions, with the L-ch portion transmitted/received by RSPI0 and the R-ch portion by RSPI1. The slave-select polarity of RSPI0 is set to low-active and that of RSPI1 to high-active. The SSL signal output on MTU2 channel 4 is used for channel selection between RSPI0 and RSPI1.

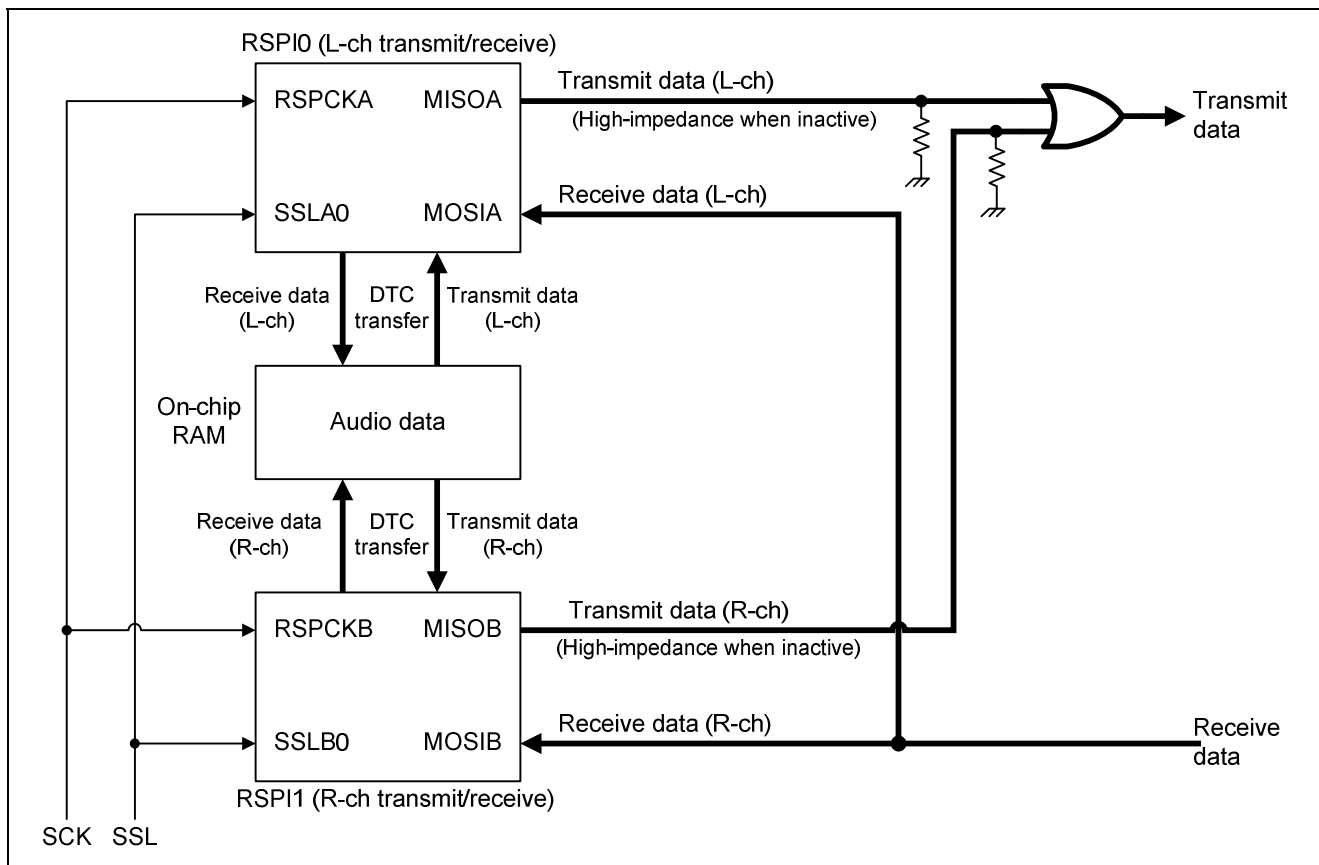


Figure 4.3 Flow of Audio Data

(a) Transmit Operation

Figure 4.4 shows the audio data transmit timing of the RSPI.

For transmission of audio data to the external device, the transmit data (L-ch) output from RSPI0 and the transmit data (R-ch) output from RSPI1 are generated through synthesis by an external OR circuit. At this time, the RSPI output channel that is not outputting transmit data is high-impedance, and it should be pulled down to prevent it from affecting the data synthesized by the OR circuit.

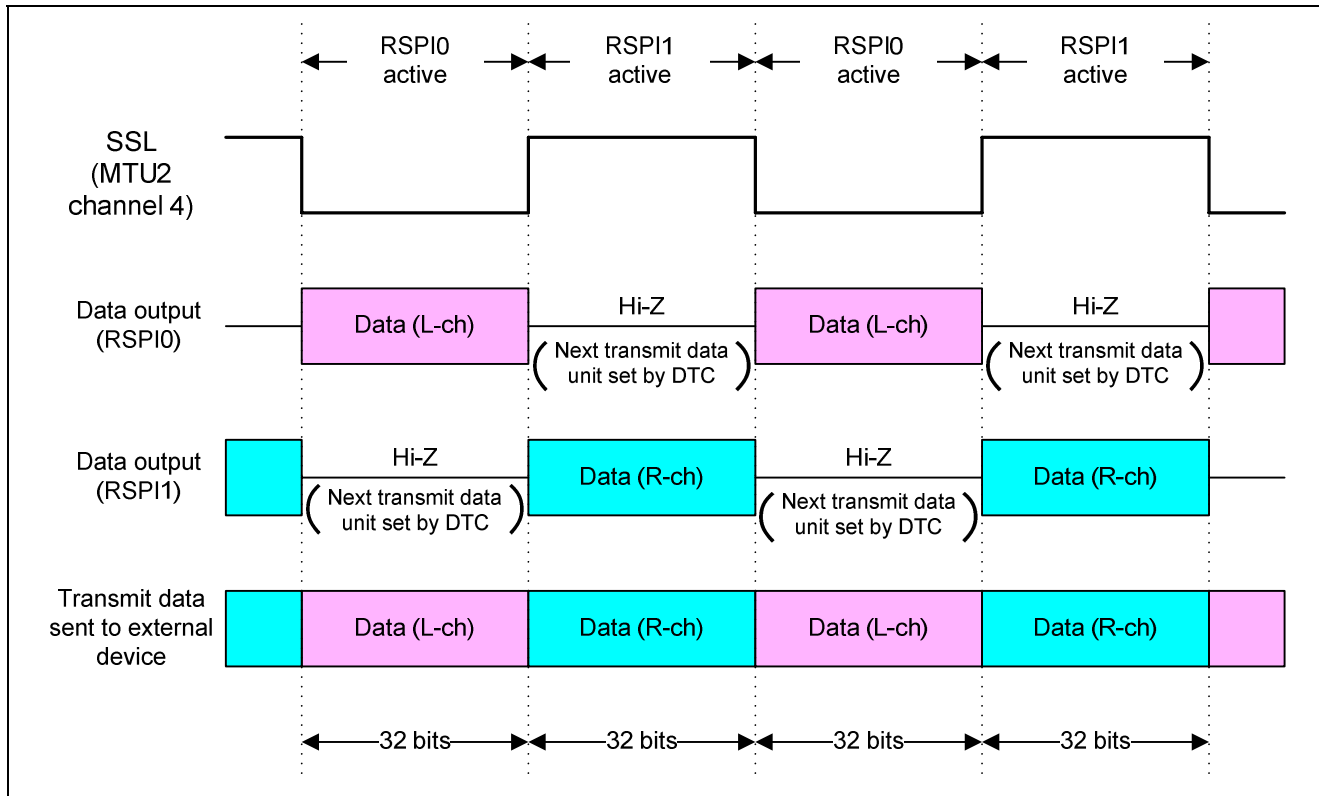


Figure 4.4 RSPI Audio Data Transmit Timing

(b) Receive Operation

Figure 4.5 shows the audio data receive timing of the RSPI.

For reception of audio data from the external device, data is input to both channels, RSPI0 and RSPI1. At this time, the active RSPI channel is selected according to the SSL signal, and the input audio data is divided into L-ch and R-ch portions and received.

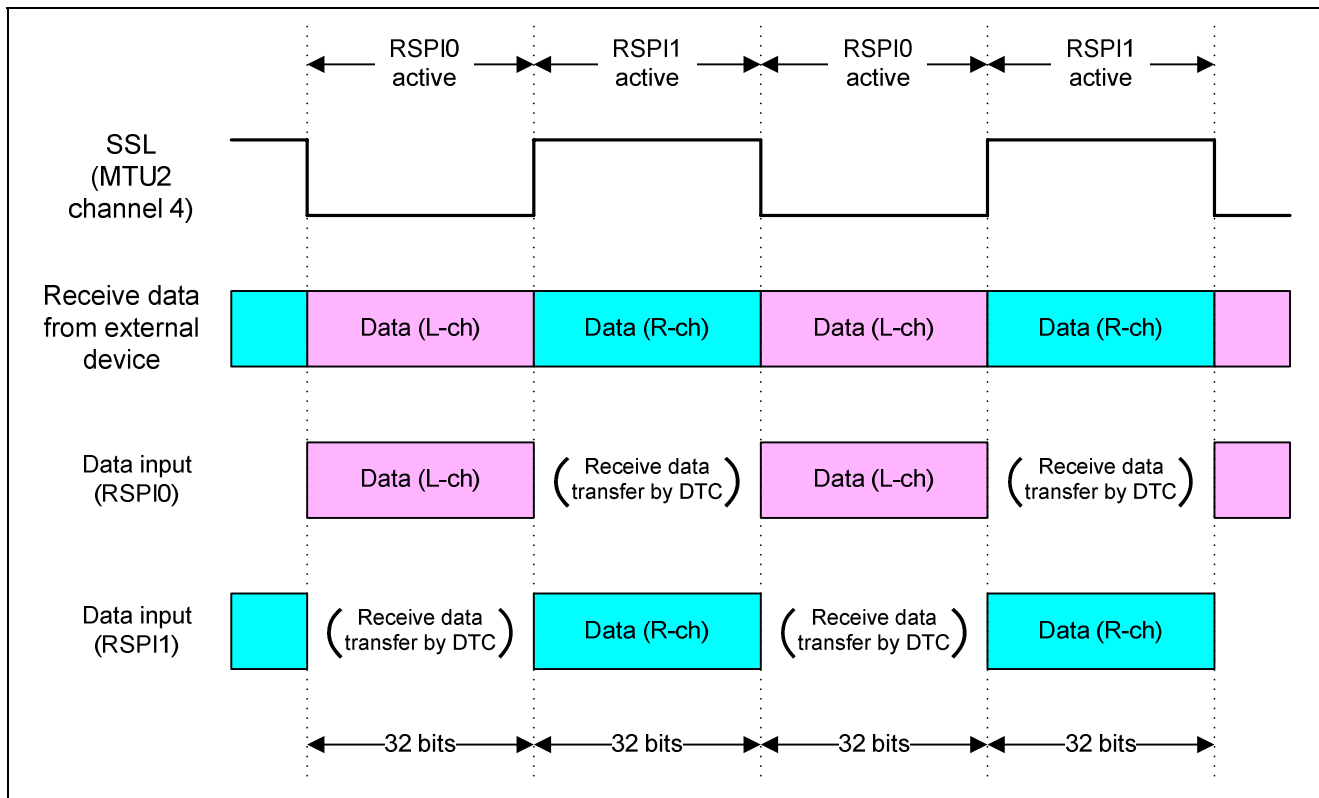


Figure 4.5 RSPI Audio Data Receive Timing

(3) I<sup>2</sup>S Communication Synchronization Recovery and RSPI Initialization

I<sup>2</sup>S communication synchronization recovery is performed separately for L-ch and R-ch by initializing the RSPI. RSPI initialization is performed by the DTC.

Figure 4.6 shows the RSPI initialization timing.

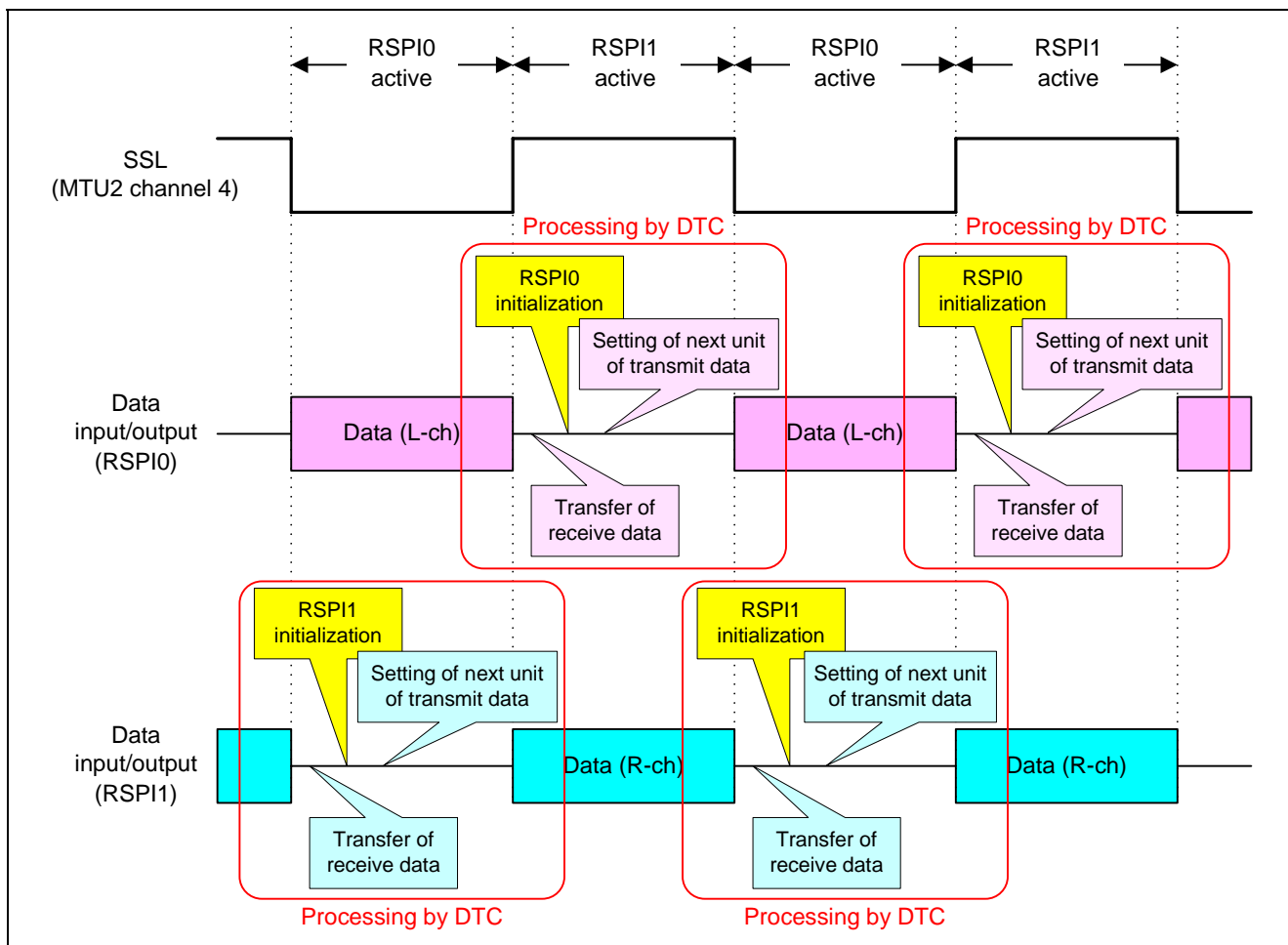


Figure 4.6 RSPI Initialization Timing

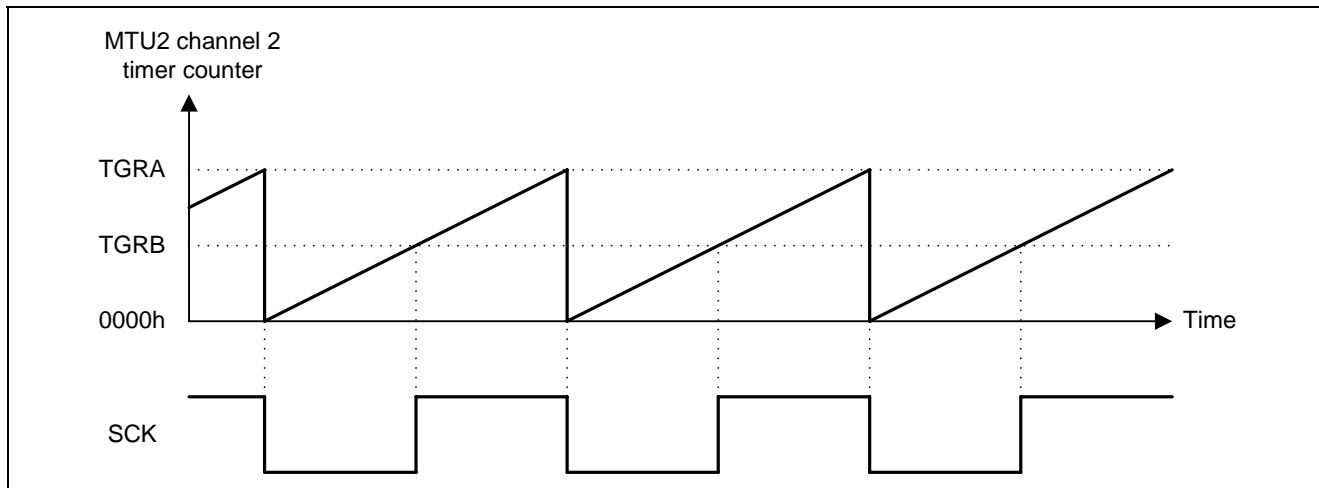
**(4) Generation of I<sup>2</sup>S Communication Clock and Channel Synchronization Signals**

The external clock (12.288 MHz) is used as the count clock, and SCK and WS are generated by the MTU2 by using compare-match. In like manner, the RSPI channel select signal (SSL) is generated by the MTU2 by using compare-match.

**(a) Generation of SCK**

MTU2 channel 2 is set to PWM mode 1 and SCK is output with a frequency of 3.072 MHz (12.288 MHz / 4), a duty ratio of 50%, and high-level initial output.

Figure 4.7 illustrates the generation of SCK on MTU2 channel 2.

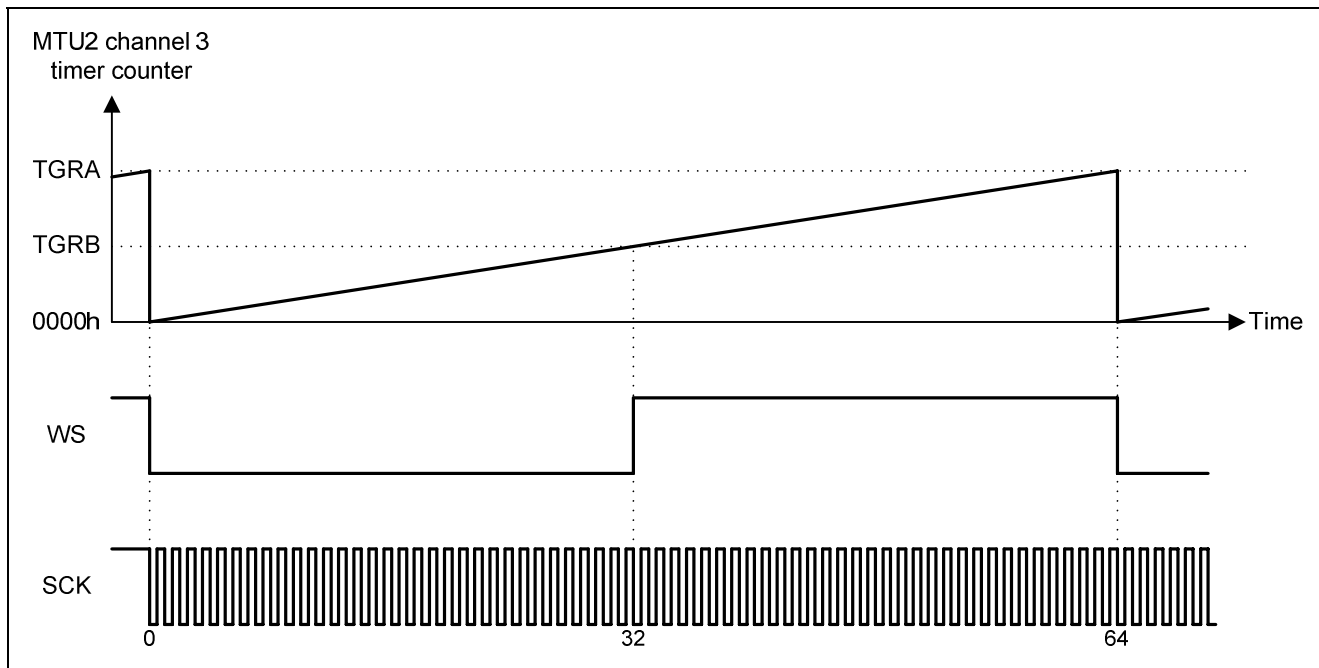


**Figure 4.7 Generation of SCK on MTU2 Channel 2**

**(b) Generation of WS**

MTU2 channel 3 is set to PWM mode 1 and WS is output with a frequency of 48 kHz (12.288 MHz / 256), a duty ratio of 50%, and high-level initial output.

Figure 4.8 illustrates the generation of WS on MTU2 channel 3.



**Figure 4.8 Generation of WS on MTU2 Channel 3**

(c) Generation of SSL

MTU2 channel 4 is set to PWM mode 1 and SSL is output with a frequency of 48 kHz (12.288 MHz / 256), a duty ratio of 50%, and high-level initial output.

A phase difference is generated between WS and SSL by setting a timer counter initial value for MTU2 channel 4 that is different from the timer counter initial value of MTU2 channel 3, which generates WS. Different phase differences between WS and SSL correspond to the transmit/receive data formats shown in 4.1.1 (1) (b), Transmit/receive data format. For details of the phase difference between WS and SSL, see 4.1.1 (5), Phase Difference between Word Select Signal and Slave Select Signal.

Figure 4.9 illustrates the generation of SSL on MTU2 channel 4.

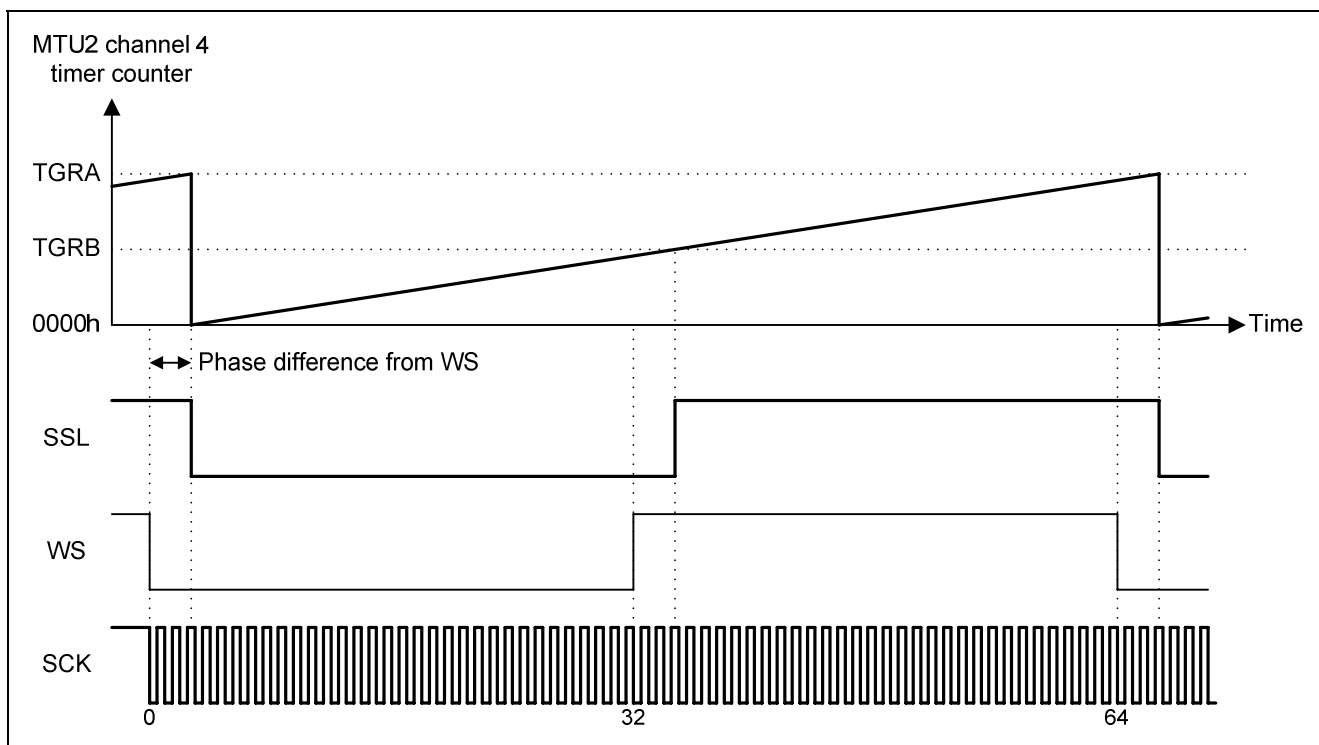


Figure 4.9 Generation of SSL on MTU2 Channel 4

(5) Phase Difference between Word Select Signal and Slave Select Signal

The sample program supports the transmit/receive data formats shown in 4.1.1 (1) (b), Transmit/receive data format, by changing the initial value of MTU2 channel 4 as described in 4.1.1 (4) (c), Generation of SSL.

Figure 4.10 shows the phase difference between WS and SSL in each transmit/receive data format.

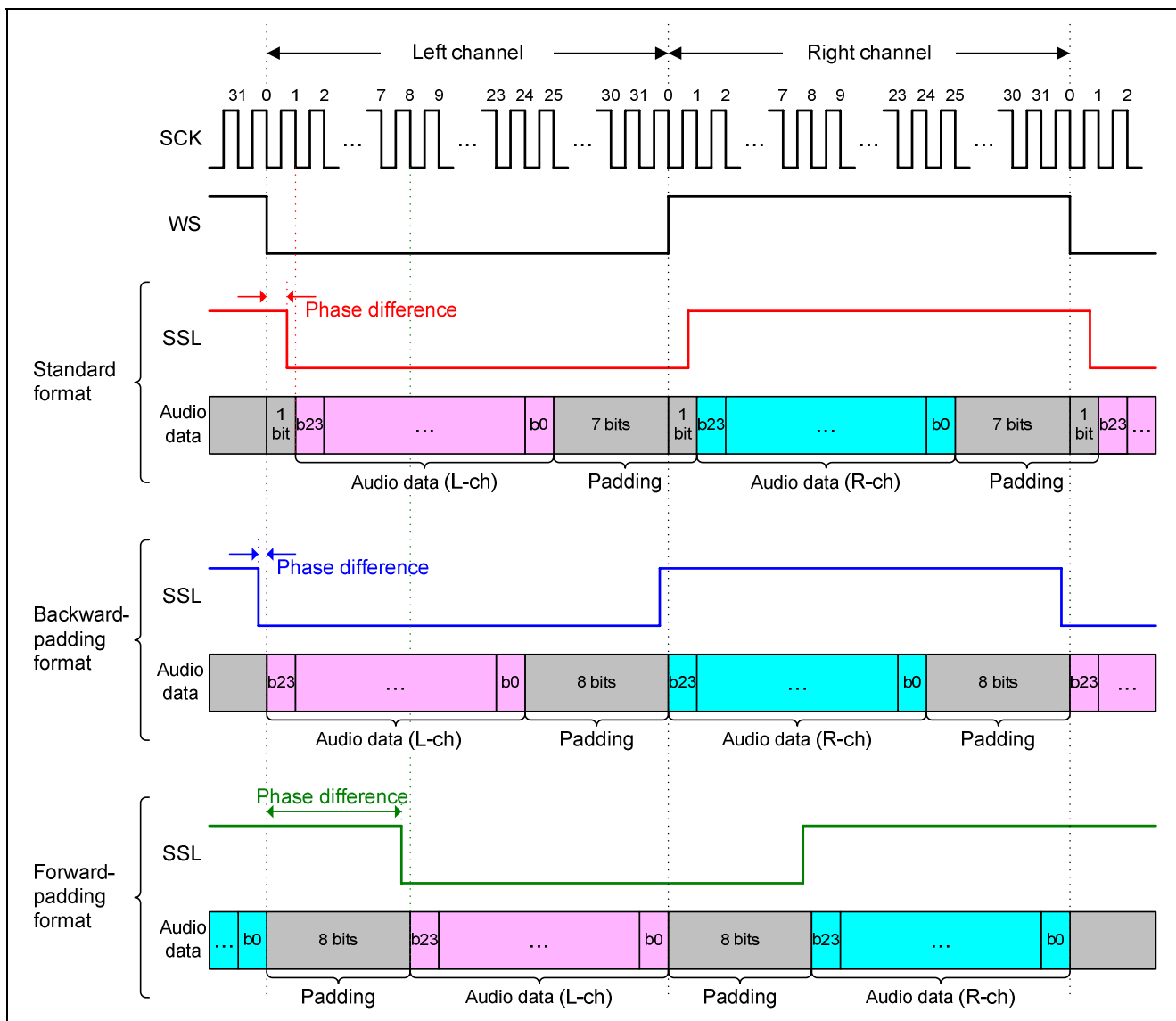


Figure 4.10 Phase Difference between WS and SSL in Each Transmit/Receive Data Format

4.1.2 Transmit Operation

(1) Transmit Operation Timing

Figure 4.11 shows the transmit operation timing (standard format).

The RSPI transmits audio data located in the on-chip RAM. The RSPI operates in slave mode (SPI operation) using a data length of 32 bits and MSB-first bit ordering.

The audio data to be transmitted is separated into L-ch and R-ch portions, with the L-ch portion transmitted by RSPI0 and the R-ch portion by RSPI1. The slave-select polarity of RSPI0 is set to low-active and that of RSPI1 to high-active. Channel selection between RSPI0 and RSPI1 takes place according to the SSL signal.

Data output on the active RSPI channel starts at the falling edge of the first SCK signal generated after channel switching, and thereafter data output is synchronized with SCK.

The MTU2 outputs the SSL signal with a phase delay relative to WS of approximately one bit according to SCK. For details of the phase difference between WS and SSL, see 4.1.1 (5), Phase Difference between Word Select Signal and Slave Select Signal.

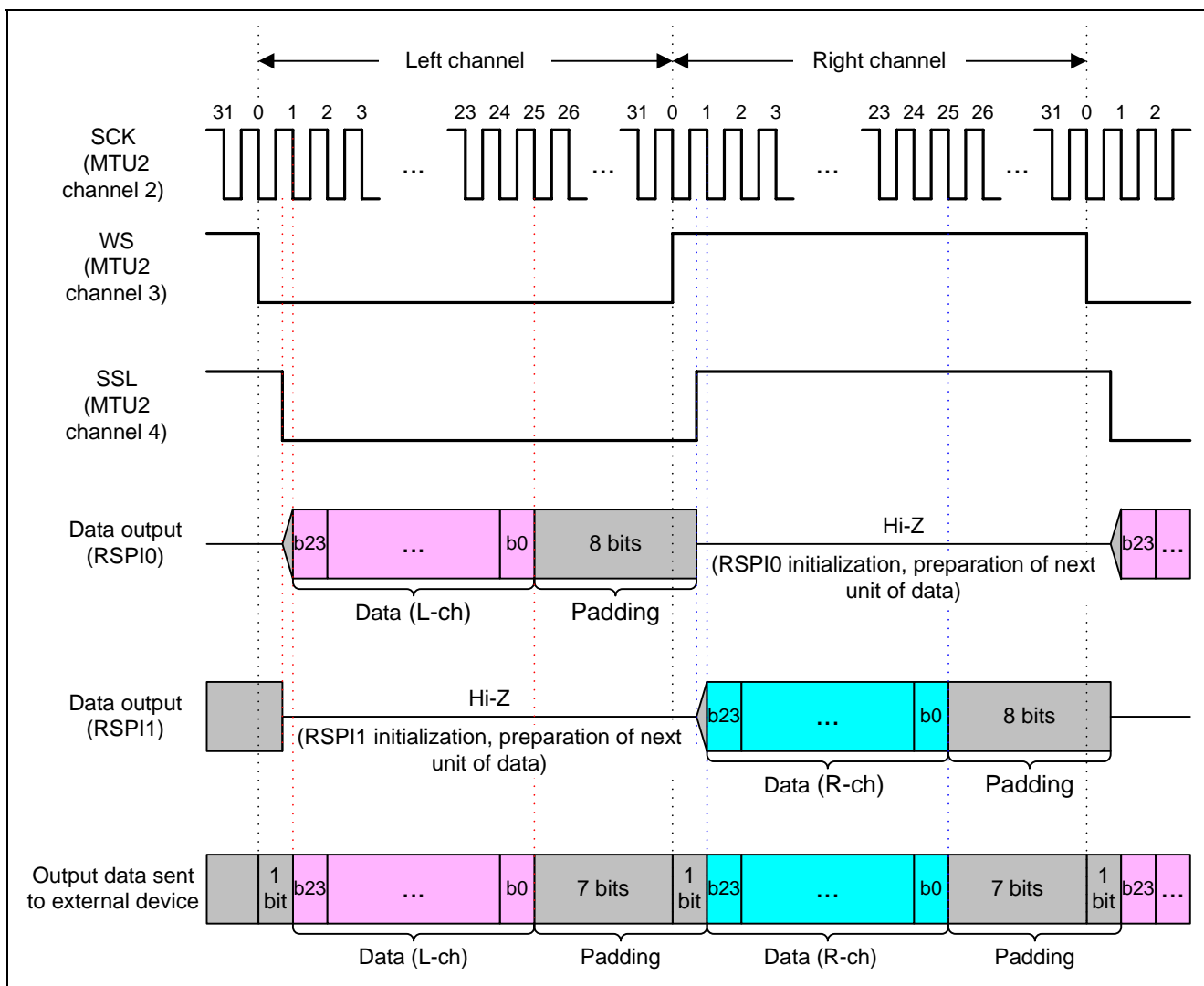


Figure 4.11 Transmit Operation Timing (Standard Format)

(2) DTC Operation during Transmission

Figure 4.12 illustrates the operation of the DTC during transmission.

The DTC transfers transmit data from the on-chip RAM to the RSPI. DTC transfers take place in pairs, one for L-ch transmit and one for R-ch transmit. The DTC generates a compare-match interrupt at each SSL edge. The DTC transfer for L-ch transmit starts at the SSL rising edge, and the DTC transfer for R-ch transmit starts at the SSL falling edge.

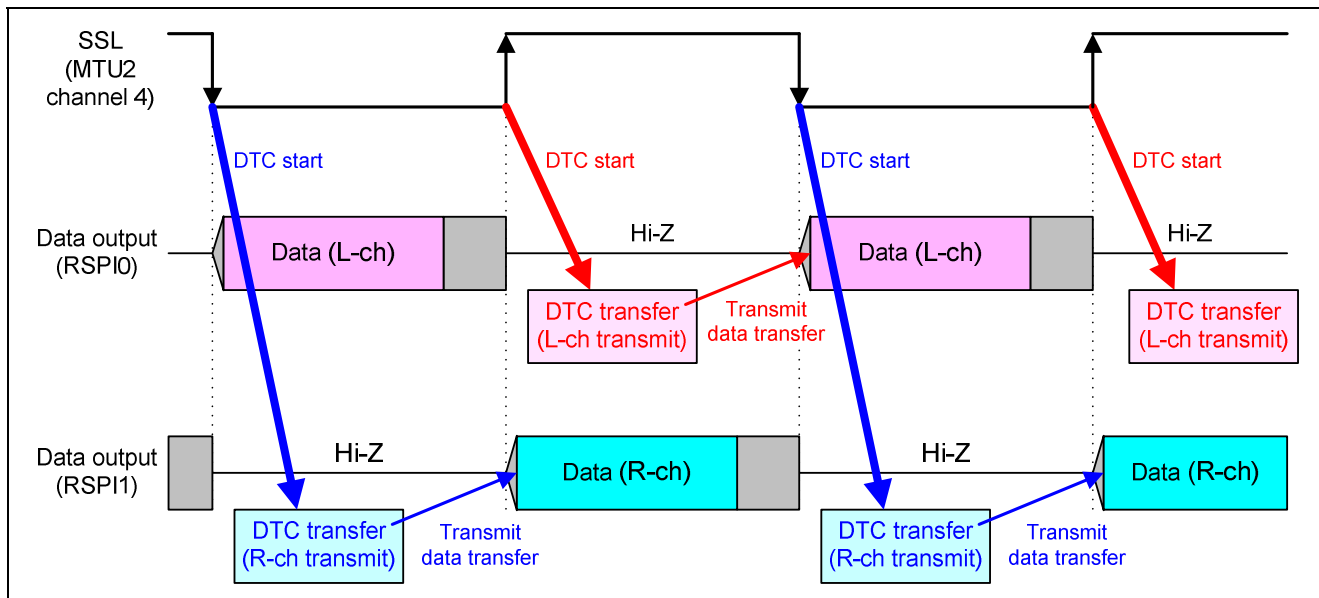


Figure 4.12 DTC Operation during Transmission

Figure 4.13 illustrates DTC operation during L-ch transmit, and figure 4.14 illustrates DTC operation during R-ch transmit.

The DTC uses chained transfer to initialize the RSPI, transfer transmit data, and transfer the transmit data address. Note that the RSPI is initialized by writing 0 to the RSPI function enable bit (SPE) in the RSPI control register (SPCR) to disable the RSPI and then writing 1 to the same bit to re-enable the RSPI.

Note: See the Hardware Manual for information on using the RSPI function enable bit to initialize the RSPI.

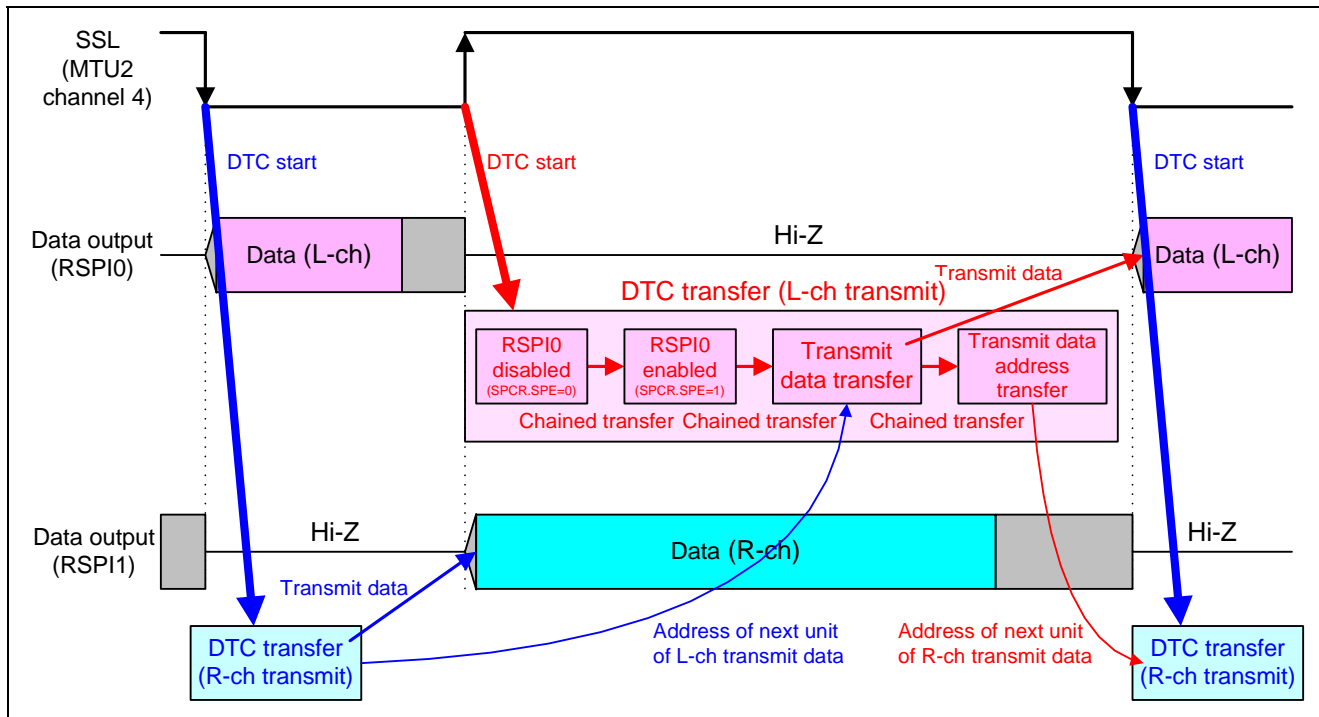


Figure 4.13 DTC Operation during L-ch Transmit

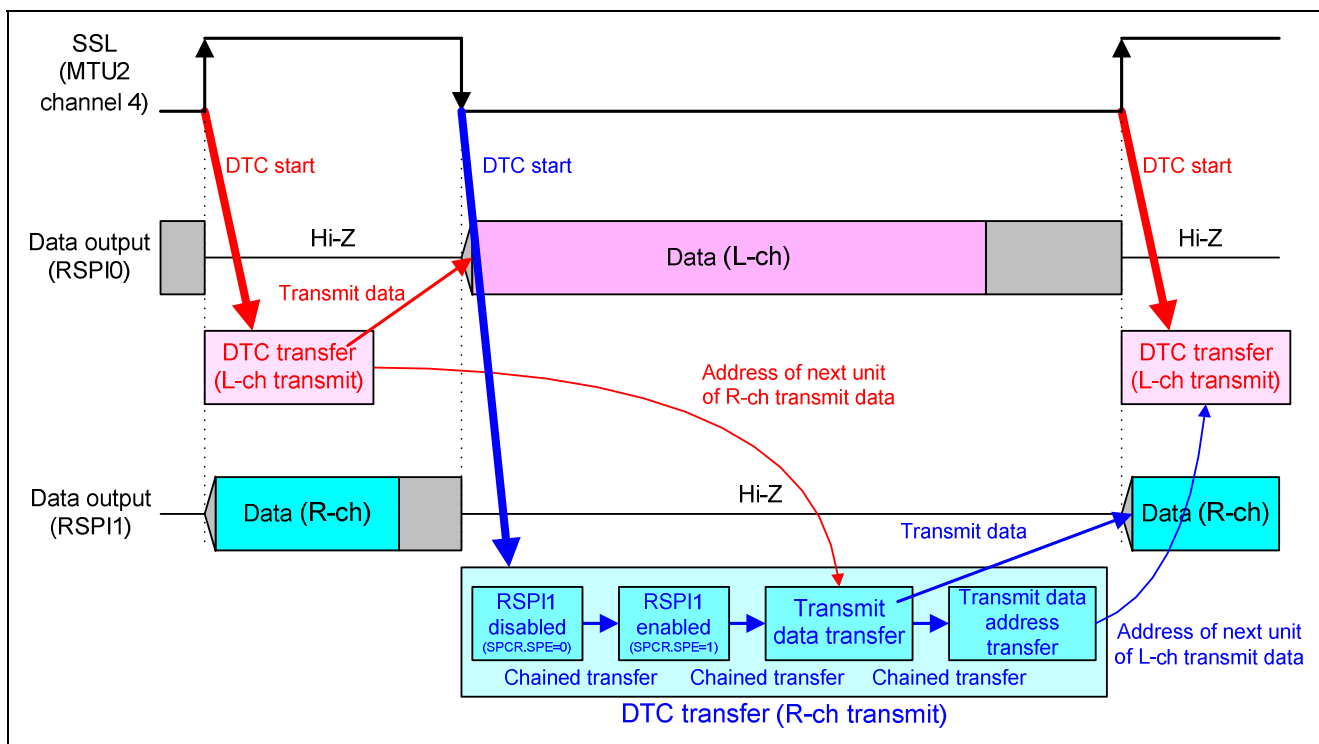


Figure 4.14 DTC Operation during R-ch Transmit

(3) Transmit Data Address Transfer by DTC

Figure 4.15 illustrates transfer of transmit data addresses by the DTC.

The sample program arranges the audio data to be transmitted in alternating L-ch and R-ch portions, each 4 bytes in length, in the on-chip RAM as shown in figure 4.1, Audio Data Format in On-Chip RAM. The sample program transfers the transmit data in pairs of DTC transfers, each of which consists of an L-ch transmit DTC transfer and an R-ch transmit DTC transfer.

As shown in figure 4.15, in the L-ch transmit DTC transfer, data 0 (L-ch) of the transmit data is transferred, after which chained transfer is used to transfer the transfer source address after the L-ch transmit DTC transfer to the transfer source address of the R-ch transmit DTC transfer. In the R-ch transmit DTC transfer, the transfer source address after the L-ch transmit DTC transfer is used as the start address for transferring data 0 (R-ch), then chained transfer is used to transfer the transfer source address after the R-ch transmit DTC transfer to the transfer source address of the L-ch transmit DTC transfer.

In the L-ch transmit DTC transfer, the transfer source address after the R-ch transmit DTC transfer, transferred by the R-ch transmit chained transfer, is used as the start address for transferring data 1 (L-ch), then chained transfer is used to transfer the transfer source address after the L-ch transmit DTC transfer to the transfer source address of the R-ch transmit DTC transfer, which is required as the start address for the next R-ch transmit DTC transfer.

In this manner, each L-ch transmit DTC transfer transfers the transmit data address required by the following R-ch transmit DTC transfer, and each R-ch transmit DTC transfer transfers the transmit data address required by the following L-ch transmit DTC transfer.

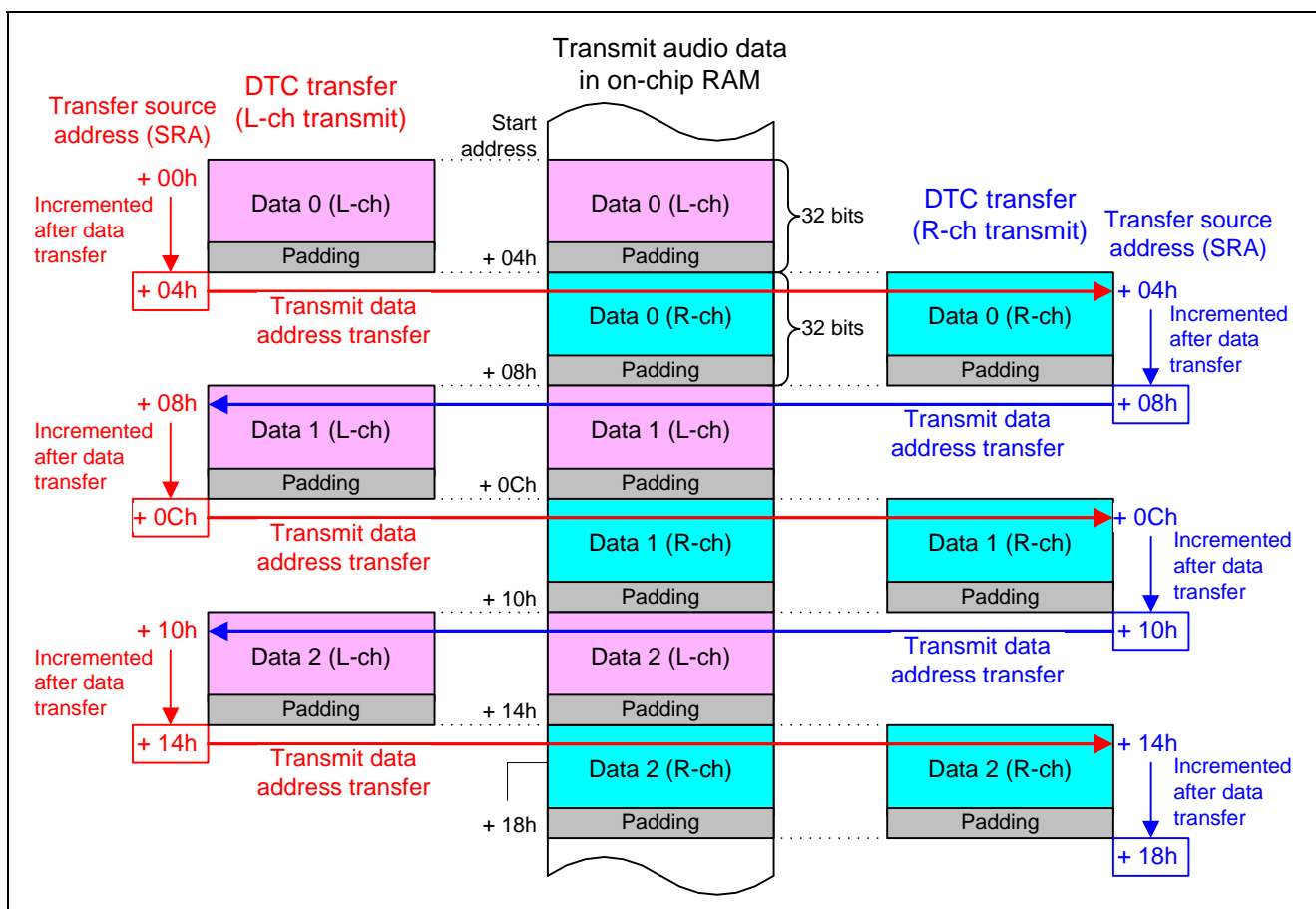


Figure 4.15 Transmit Data Address Transfer by DTC

(4) Audio Data Transmit End

The sample program ends transmission of audio data by disabling the RSPI, thereby halting its operation. Disabling the RSPI is performed for L-ch transmit and R-ch transmit DTC transfers.

(a) Transmit end processing for L-ch transmit

Figure 4.16 shows an outline of transmit end operation for L-ch transmit.

When the n-1'th L-ch transmit DTC transfer starts, chained transfer is used to perform transmit end processing. By means of transmit end processing, channel RSPI0 only is disabled when the n'th transfer starts, thereby ending RSPI0 operation.

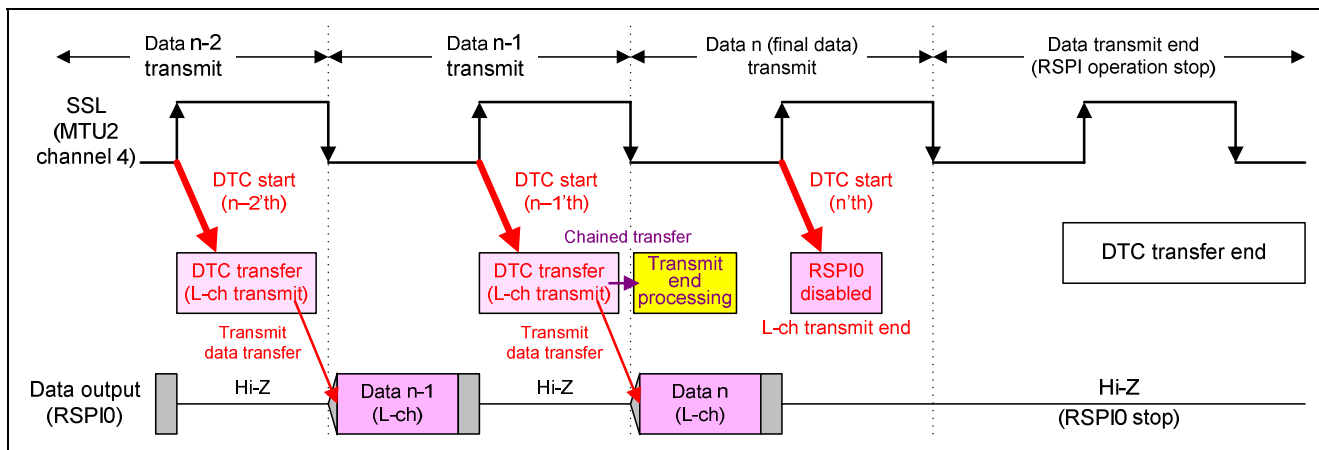


Figure 4.16 Transmit End Operation for L-ch Transmit

Figure 4.17 illustrates in detail the use of chained transfer to perform transmit end processing for L-ch transmit.

As shown in figure 4.13, DTC Operation during L-ch Transmit, for L-ch transmit DTC transfer, chained transfer is used to disable RSPI0, to enable RSPI0, to transfer the transmit data, and to transfer the transmit data address.

By setting to n-1 the transfer counter, which is part of the transfer information of the transmit data address transfer, and setting the DTC chain transfer select bit to 1 (chain transfer is performed only when the transfer counter is 0), transmit end processing is triggered by chained transfer after the transmit data address transfer at the n-1'th start.

Transmit end processing disables chained transfer after the disabling of RSPI0 at the n'th start by overwriting the value of the DTC chain transfer enable bit in the RSPI0 disable transfer information. In this way, RSPI0 only is disabled at the n'th start, stopping RSPI0 operation and ending transmit operation.

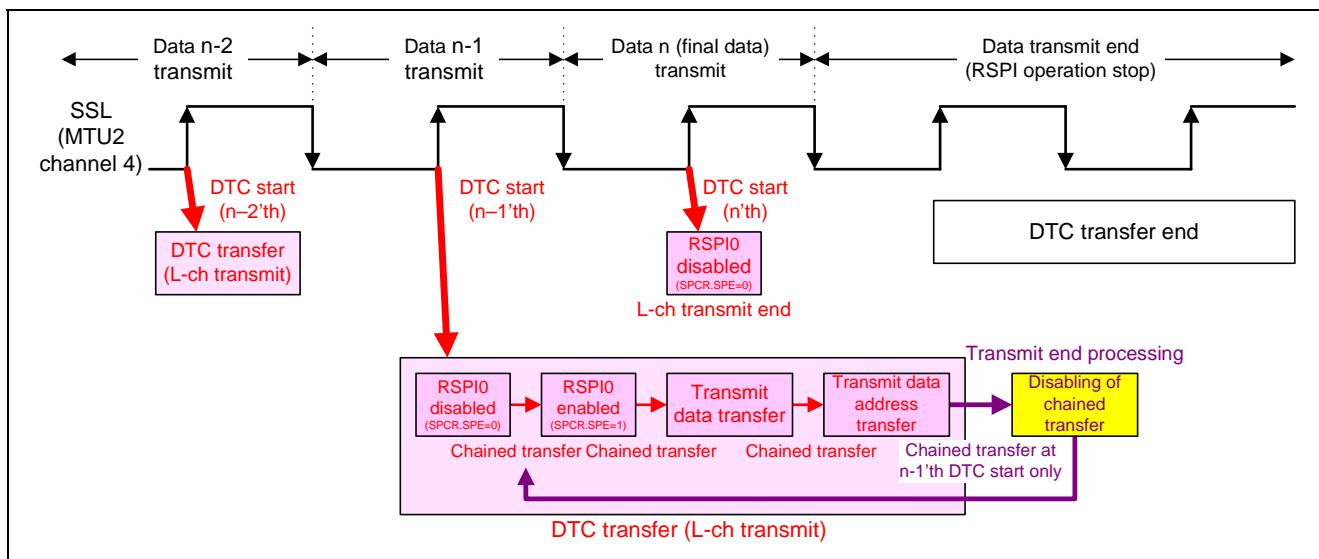


Figure 4.17 Transmit End Processing by L-ch Transmit DTC Transfer

(b) Transmit end processing for R-ch transmit

Figure 4.18 shows an outline of transmit end operation for R-ch transmit.

When the n<sup>th</sup> R-ch transmit DTC transfer starts, chained transfer is used to perform transmit end processing. By means of transmit end processing, channel RSPI1 only is disabled when the n+1<sup>th</sup> transfer starts, thereby ending RSPI1 operation.

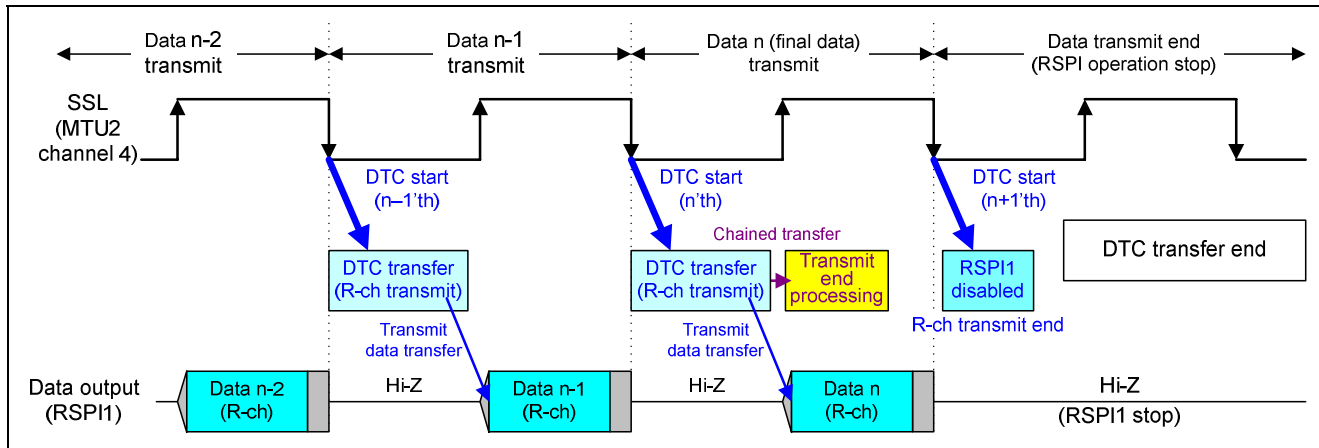


Figure 4.18 Transmit End Operation for R-ch Transmit

Figure 4.19 illustrates in detail the use of chained transfer to perform transmit end processing for R-ch transmit.

As shown in figure 4.14, DTC Operation during R-ch Transmit, for R-ch transmit DTC transfer, chained transfer is used to disable RSPI1, to enable RSPI1, to transfer the transmit data, and to transfer the transmit data address.

By setting to n the transfer counter, which is part of the transfer information of the transmit data address transfer, and setting the DTC chain transfer select bit to 1 (chain transfer is performed only when the transfer counter is 0), transmit end processing is triggered by chained transfer after the transmit data address transfer at the n<sup>th</sup> start.

Transmit end processing disables chained transfer after the disabling of RSPI0 at the n+1<sup>th</sup> start by overwriting the value of the DTC chain transfer enable bit in the RSPI1 disable transfer information. In this way, RSPI1 only is disabled at the n+1<sup>th</sup> start, stopping RSPI1 operation and ending transmit operation.

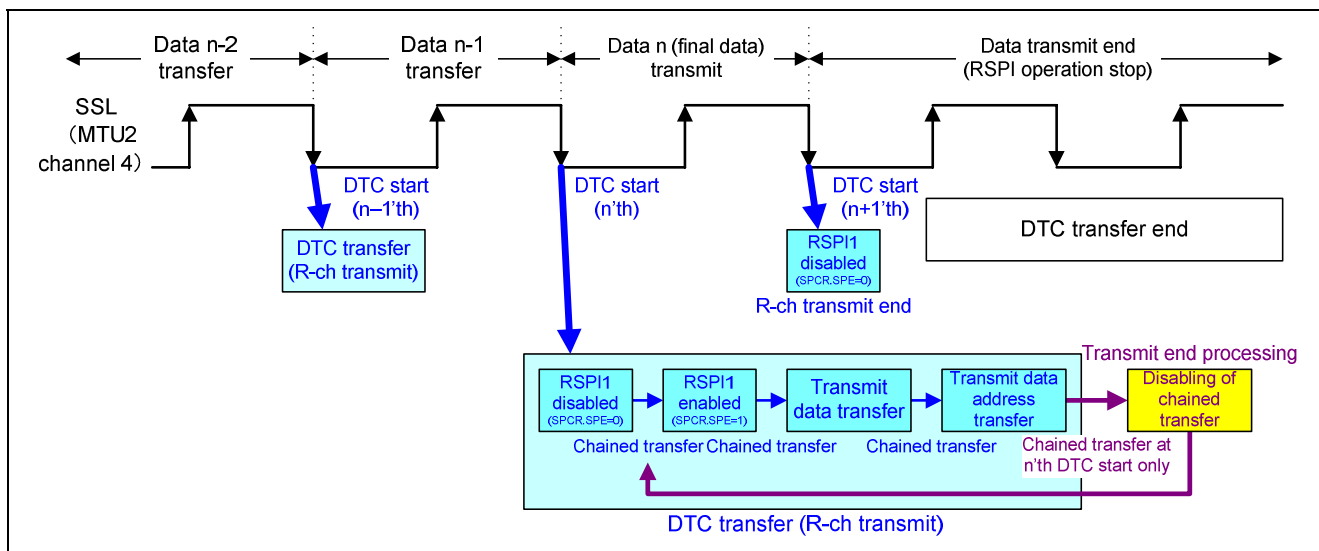


Figure 4.19 Transmit End Processing by R-ch Transmit DTC Transfer

4.1.3 Receive Operation

(1) Receive Operation Timing

Figure 4.20 shows the timing of receive operation (standard format).

The RSPI receives audio data from the external device. The RSPI operates in slave mode (SPI operation) using a data length of 32 bits and MSB-first bit ordering.

The audio data from the external device is input to both channels, RSPI0 and RSPI1, and separated into L-ch and R-ch portions for reception. The slave-select polarity of RSPI0 is set to low-active and that of RSPI1 to high-active. Channel selection between RSPI0 and RSPI1 takes place according to the SSL signal. Data is imported on the active RSPI channel at the rising edge of the first SCK signal generated after channel switching, and thereafter importing of input data is synchronized with SCK.

The input data imported by the RSPI is transferred to the on-chip RAM by the DTC and stored as shown in figure 4.1.

The MTU2 outputs the SSL signal with a phase delay relative to WS of approximately one bit according to SCK. For details of the phase difference between WS and SSL, see 4.1.1 (5), Phase Difference between Word Select Signal and Slave Select Signal.

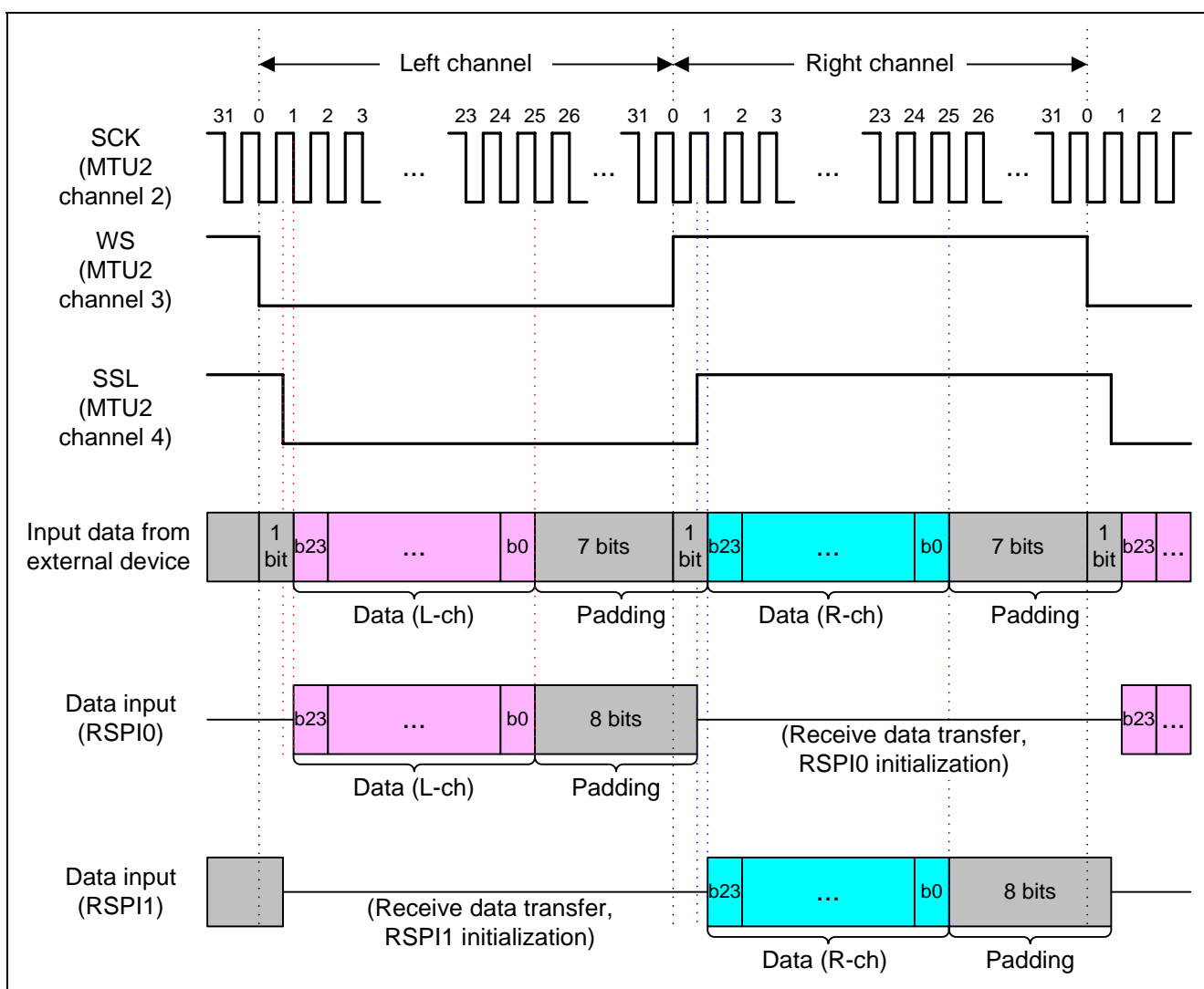


Figure 4.20 Receive Operation Timing (Standard Format)

(2) DTC Operation during Reception

Figure 4.21 illustrates the DTC operation during reception.

The DTC transfers receive data from the RSPI to the on-chip RAM. DTC transfers take place in sets of four: L-ch receive and R-ch receive, and RSPI0 initialization and RSPI1 initialization. For data reception, the DTC is started by a receive buffer full interrupt. The DTC activation for L-ch receive occurs at the RSPI0 receive buffer full interrupt, and the DTC activation for R-ch receive occurs at the RSPI1 receive buffer full interrupt. For RSPI initialization, the DTC is activated by a compare-match interrupt at each SSL edge. The DTC activation for RSPI0 initialization occurs at the SSL rising edge, and the DTC activation for RSPI1 initialization occurs at the SSL falling edge.

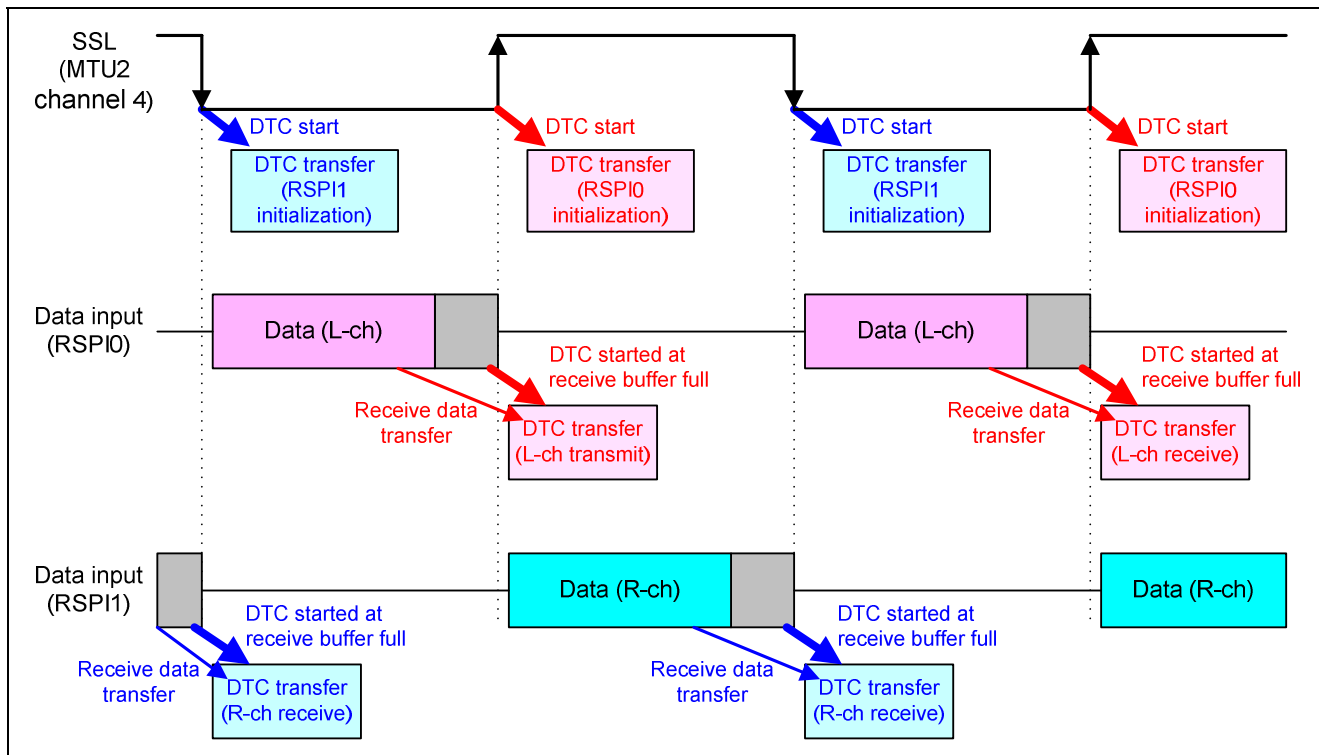


Figure 4.21 DTC Operation during Reception

Figure 4.22 illustrates DTC operation during L-ch receive, and figure 4.23 illustrates DTC operation during R-ch receive.

For data reception the DTC uses chained transfer to transfer receive data and transfer the receive data address. In addition, the DTC initializes the RSPI by using chained transfer. Note that the RSPI is initialized by writing 0 to the RSPI function enable bit (SPE) in the RSPI control register (SPCR) to disable the RSPI and then writing 1 to the same bit to re-enable the RSPI.

Note: See the Hardware Manual for information on using the RSPI function enable bit to initialize the RSPI.

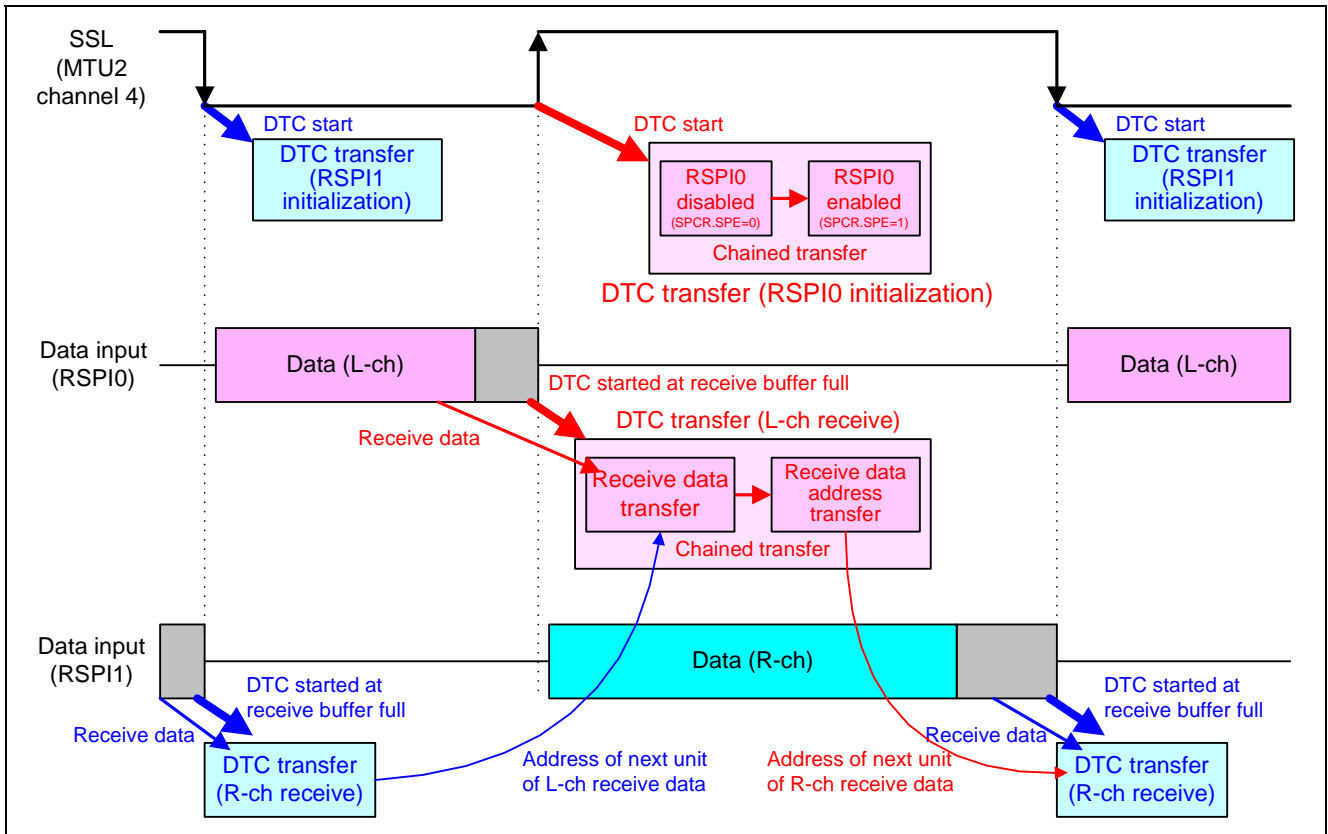


Figure 4.22 DTC Operation during L-ch Receive

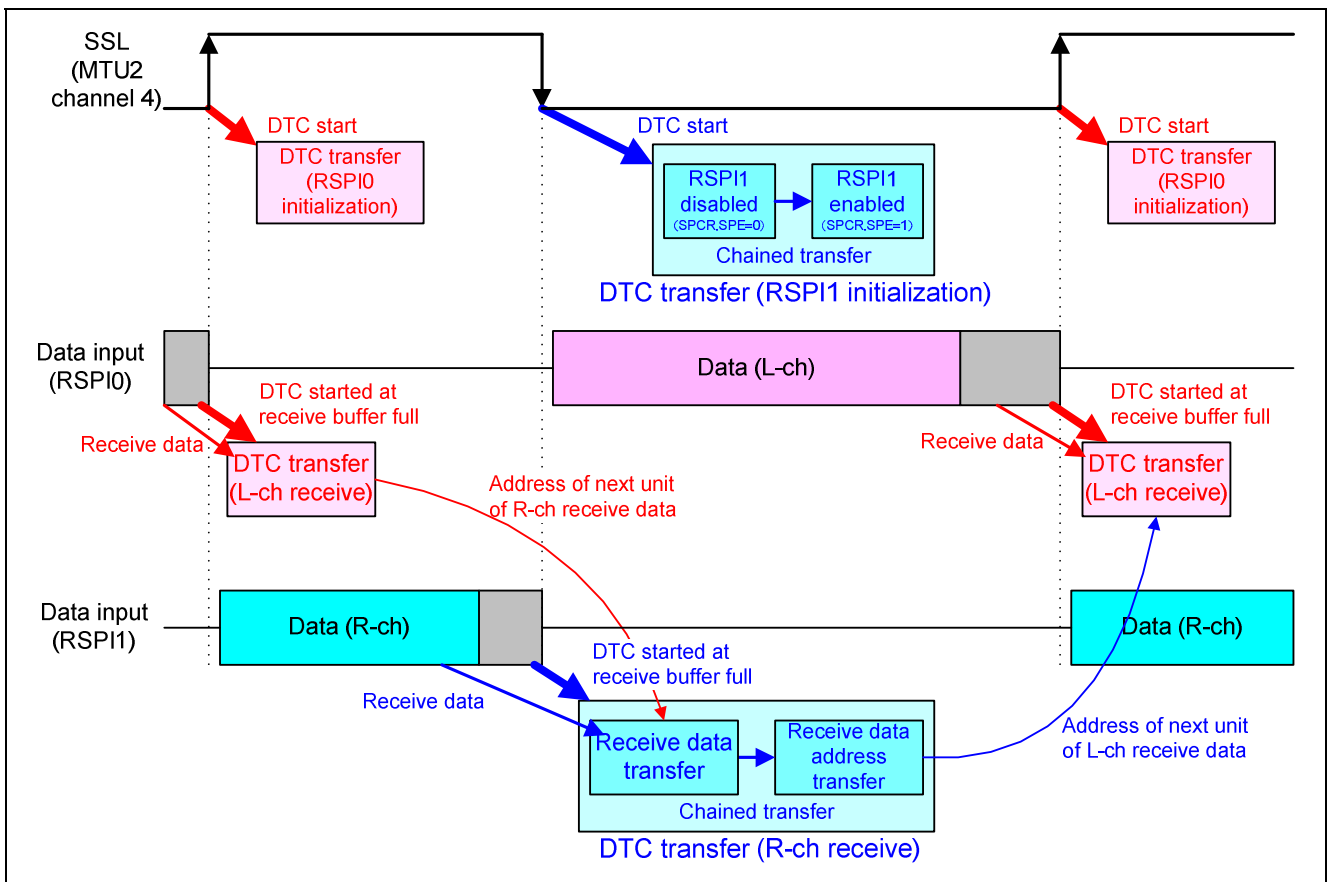


Figure 4.23 DTC Operation during R-ch Receive

(3) Receive Data Address Transfer by DTC

Figure 4.24 illustrates transfer of receive data addresses by the DTC.

The sample program arranges the received audio data in alternating L-ch and R-ch portions, each 4 bytes in length, in the on-chip RAM as shown in figure 4.1, Audio Data Format in On-Chip RAM. The sample program transfers the receive data in pairs of DTC transfers, each of which consists of an L-ch receive DTC transfer and an R-ch receive DTC transfer.

As shown in figure 4.24, in the L-ch receive DTC transfer, data 0 (L-ch) of the receive data is transferred, after which chained transfer is used to transfer the transfer destination address after the L-ch receive DTC transfer to the transfer destination address of the R-ch receive DTC transfer. In the R-ch receive DTC transfer, the transfer destination address after the L-ch receive DTC transfer is used as the start address for transferring data 0 (R-ch), then chained transfer is used to transfer the transfer destination address after the R-ch receive DTC transfer to the transfer destination address of the L-ch receive DTC transfer.

In the L-ch receive DTC transfer, the transfer destination address after the R-ch receive DTC transfer, transferred by the R-ch receive chained transfer, is used as the start address for transferring data 1 (L-ch), then chained transfer is used to transfer the transfer destination address after the L-ch receive DTC transfer to the transfer destination address of the R-ch receive DTC transfer, which is required as the start address for the next R-ch receive DTC transfer.

In this manner, each L-ch receive DTC transfer transfers the receive data address required by the following R-ch receive DTC transfer, and each R-ch receive DTC transfer transfers the receive data address required by the following L-ch receive DTC transfer.

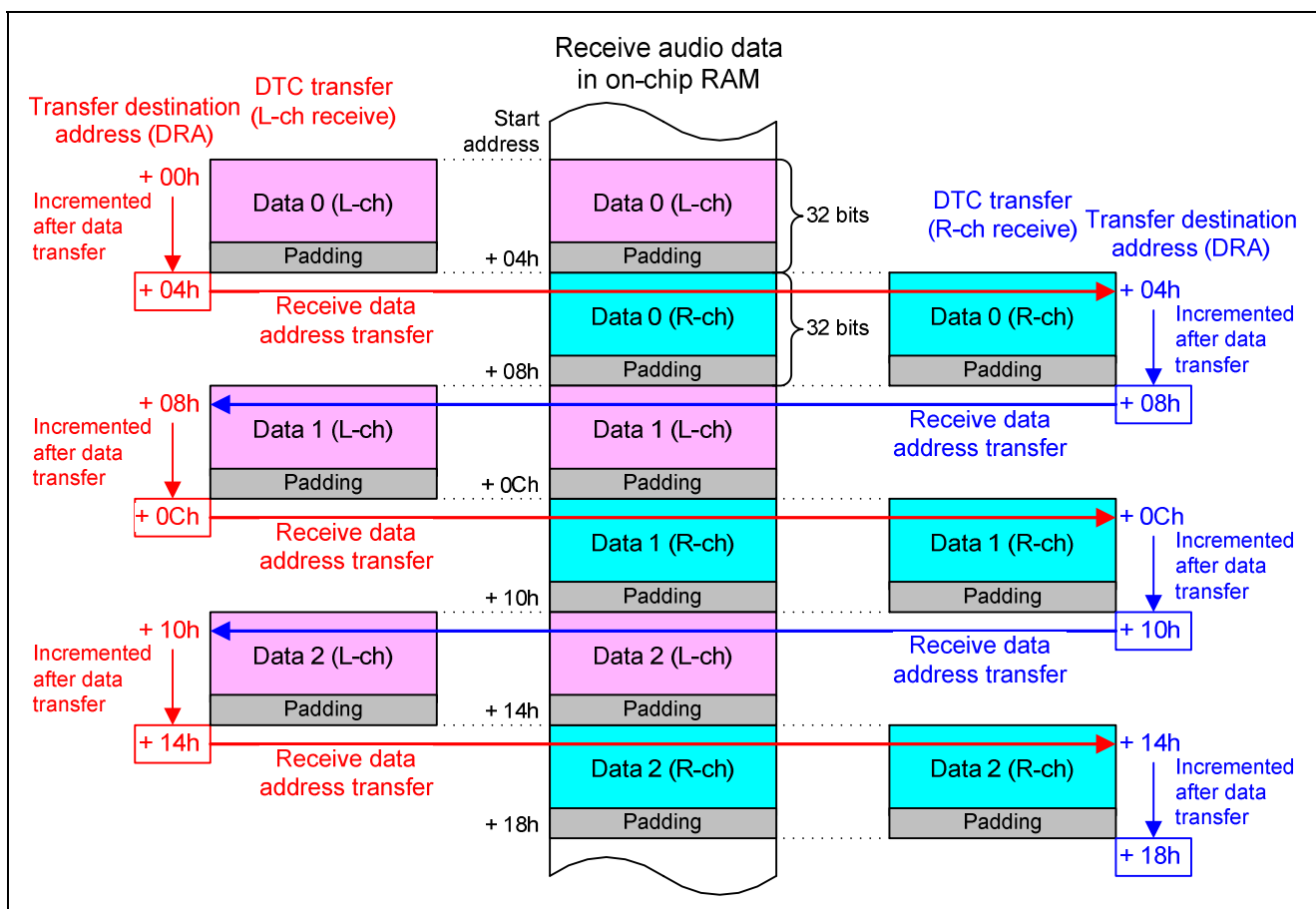


Figure 4.24 Receive Data Address Transfer by DTC

(4) Audio Data Receive End

The sample program ends reception of audio data by disabling the RSPI, thereby halting its operation. Disabling the RSPI is performed for RSPI0 initialization and RSPI1 initialization DTC transfers.

(a) Receive end processing for L-ch receive

Figure 4.25 shows an outline of receive end operation for L-ch receive.

When the n-1'th RSPI0 initialization DTC transfer starts, chained transfer is used to perform receive end processing. By means of receive end processing, channel RSPI0 is disabled when the n'th transfer starts, thereby ending RSPI0 operation.

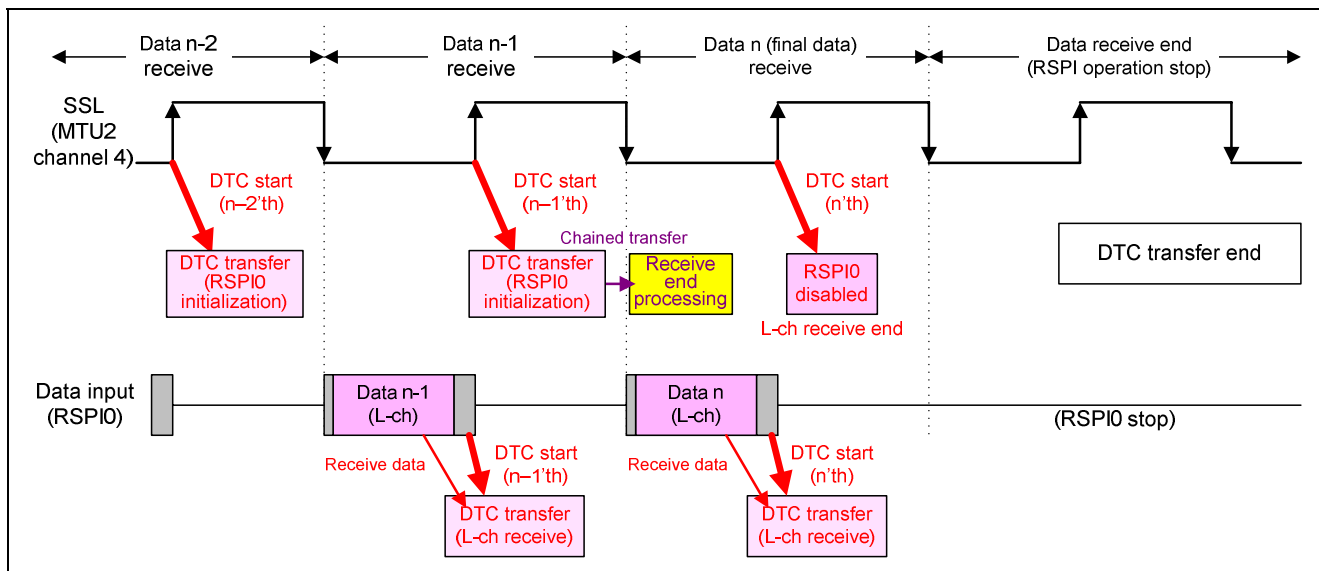


Figure 4.25 Receive End Operation for L-ch Receive

Figure 4.26 illustrates in detail the use of chained transfer to perform receive end processing for L-ch receive.

As shown in figure 4.22, DTC Operation during L-ch Receive, for RSPI0 initialization DTC transfer, chained transfer is used to disable RSPI0 and to enable RSPI0.

By setting to n-1 the transfer counter, which is part of the transfer information of the RSPI0 enable transfer, and setting the DTC chain transfer select bit to 1 (chain transfer is performed only when the transfer counter is 0), receive end processing is triggered by chained transfer after the RSPI0 enable transfer at the n-1'th start.

Receive end processing disables chained transfer after the disabling of RSPI0 at the n'th start by overwriting the value of the DTC chain transfer enable bit in the RSPI0 disable transfer information. In this way, RSPI0 is disabled at the n'th start, stopping RSPI0 operation and ending receive operation.

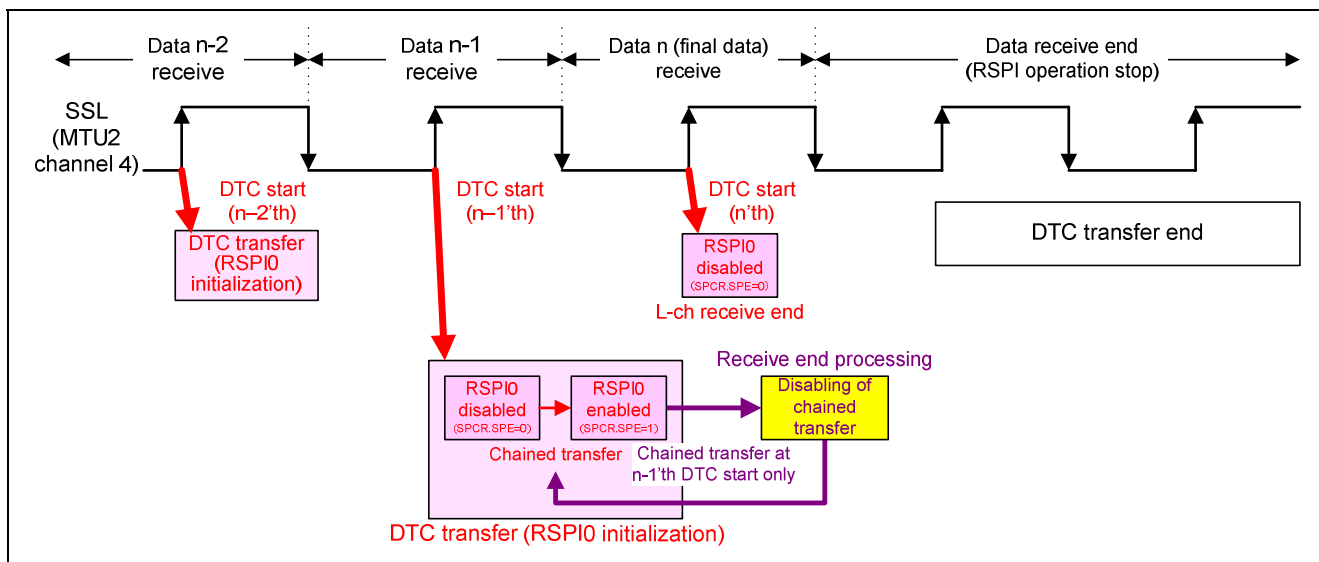


Figure 4.26 Receive End Processing by RSPI0 Initialization DTC Transfer

(b) Receive end processing for R-ch receive

Figure 4.27 shows an outline of receive end operation for R-ch receive.

When the n'th RSPI1 initialization DTC transfer starts, chained transfer is used to perform receive end processing. By means of receive end processing, channel RSPI1 is disabled when the n+1'th transfer starts, thereby ending RSPI1 operation.

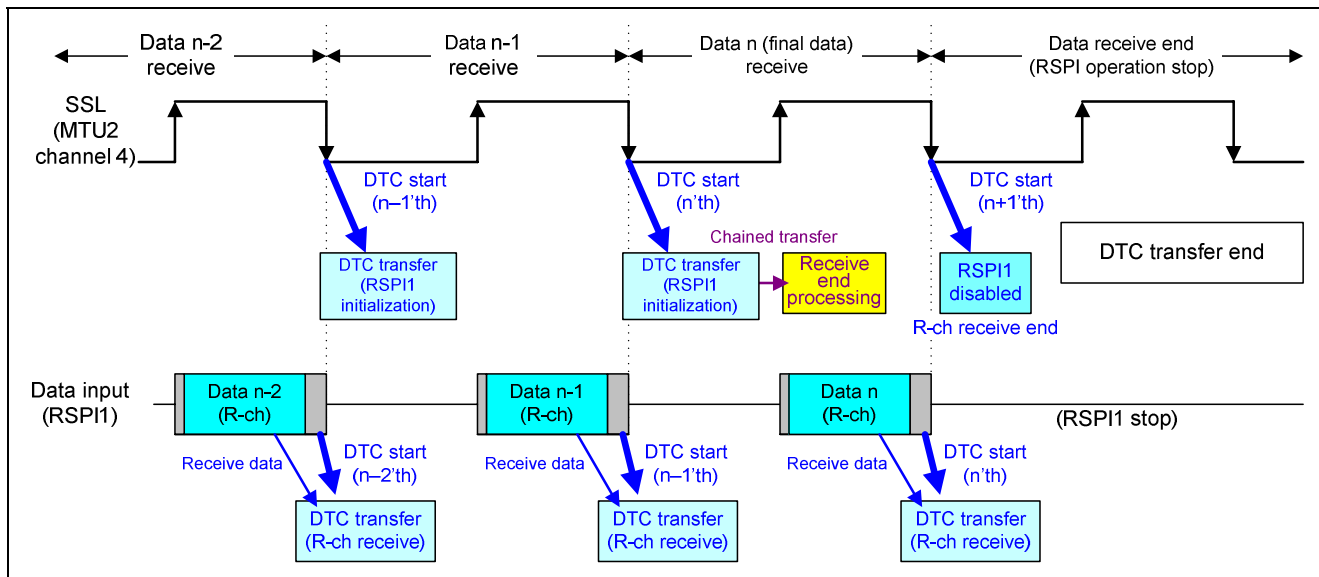


Figure 4.27 Receive End Operation for R-ch Receive

Figure 4.28 illustrates in detail the use of chained transfer to perform receive end processing for R-ch receive.

As shown in figure 4.23, DTC Operation during R-ch Receive, for RSPI1 initialization DTC transfer, chained transfer is used to disable RSPI1 and to enable RSPI1.

By setting to n the transfer counter, which is part of the transfer information of the RSPI1 enable transfer, and setting the DTC chain transfer select bit to 1 (chain transfer is performed only when the transfer counter is 0), receive end processing is triggered by chained transfer after the RSPI1 enable transfer at the n'th start.

Receive end processing disables chained transfer after the disabling of RSPI1 at the n+1'th start by overwriting the value of the DTC chain transfer enable bit in the RSPI1 disable transfer information. In this way, RSPI1 is disabled at the n+1'th start, stopping RSPI1 operation and ending receive operation.

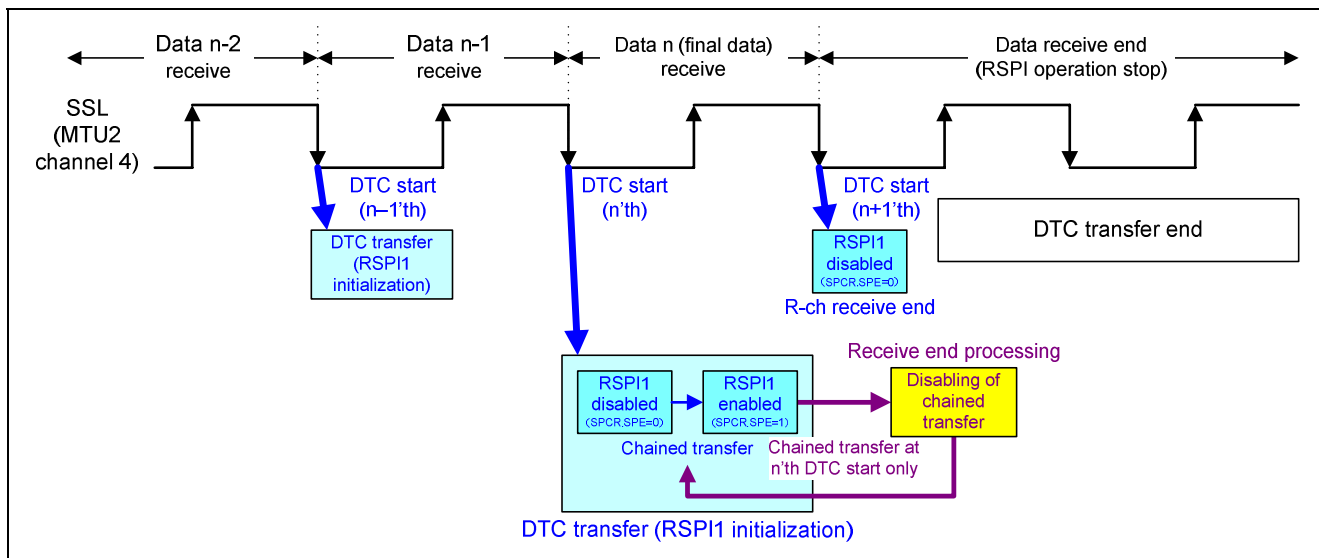


Figure 4.28 Receive End Processing by RSPI1 Initialization DTC Transfer

## 4.2 File Structure

Table 4.1 lists the files used by the sample program. Note that files that are created automatically by the integrated development environment are omitted.

**Table 4.1 File Structure**

<b>File Name</b>	<b>Description</b>	<b>Remarks</b>
i2s_main.c	I <sup>2</sup> S communication main process	
i2s_dtc.c	DTCa operation settings	
i2s_mtu2.c	MTU2 operation settings	
i2s_rspl.c	RSPI operation settings	
i2s.h	I <sup>2</sup> S communication macro definitions	Transmit/receive data format settings
iodefine_RX62N.h	RX62N Series I/O header file	

### 4.3 List of Constants

Table 4.2 lists the constants used by the sample program.

**Table 4.2 Constants Used by Sample Program**

Constant	Setting Value	Description
L_CH	0	L-ch index No. of audio data array in on-chip RAM
R_CH	1	R-ch index No. of audio data array in on-chip RAM
NULL_DATA	0x00000000	Initialization data for receive audio data area
TRANSMIT_FORMAT	STANDARD_PADDING or BACKWARD_PADDING or FORWARD_PADDING	Transmit/receive data format selection (selection among standard, backward-padding, and forward-padding)
STANDARD_PADDING	0	Standard format
BACKWARD_PADDING	1	Backward-padding format
FORWARD_PADDING	2	Forward-padding format
TRANSMIT_MODE	0	Transmit operation mode
RECEIVE_MODE	1	Receive operation mode
TRANSCEIVE_MODE	2	Transceive operation mode
AUD_SIZE	0x100	Audio data total byte count
AUD_NUM	(AUD_SIZE / 4) / 2	Audio data (L-ch/R-ch) data count
DTC_TX_CHAIN_SIZE	5	Transfer information count for DTCa chained transfers during transmit operation
DTC_RX_CHAIN_SIZE	2	Transfer information count for DTCa chained transfers during receive operation
DTC_TX_L_COUNT	AUD_NUM - 1	DTC transfer (L-ch transmit, RSPI0 initialization) count
DTC_TX_R_COUNT	AUD_NUM	DTC transfer (R-ch transmit, RSPI1 initialization) count
DTC_RX_L_COUNT	AUD_NUM	DTC transfer (L-ch transmit) count
DTC_RX_R_COUNT	AUD_NUM	DTC transfer (R-ch transmit) count
SCK_CYCLE_VALUE	0x0003	SCK cycle
WS_SSL_CYCLE_VALUE	0x00FF	WS and SSL cycle
SSL_DELAY_VALUE_S	0x0003	WS and SSL phase value (standard format)
SSL_DELAY_VALUE_B	0x0001	WS and SSL phase value (Backward-padding format)
SSL_DELAY_VALUE_F	0x001F	WS and SSL phase value (Forward-padding format)

#### 4.4 List of Structures and Unions

Figure 4.29 lists the structures and unions used by the sample program.

```

#ifdef __LIT
struct st_dtc_data{
    unsigned char    wk[2];    /* Reserved area */
    unsigned char    MRB;     /* DTC mode register B */
    unsigned char    MRA;     /* DTC mode register A */
    unsigned long    SAR;     /* DTC transfer source address register */
    unsigned long    DAR;     /* DTC transfer destination address register */
    unsigned short   CRB;     /* DTC transfer count register B */
    unsigned short   CRA;     /* DTC transfer count register A */
};
#endif
#ifdef __BIG
struct st_dtc_data{
    unsigned char    MRA;     /* DTC mode register A */
    unsigned char    MRB;     /* DTC mode register B */
    unsigned char    wk[2];   /* Reserved area */
    unsigned long    SAR;     /* DTC transfer source address register */
    unsigned long    DAR;     /* DTC transfer destination address register */
    unsigned short   CRA;     /* DTC transfer count register A */
    unsigned short   CRB;     /* DTC transfer count register B */
};
#endif

struct st_dtc_data DTC_TX_L[DTC_TX_CHAIN_SIZE];    /* DTC transfer information (L-ch transmit) */
struct st_dtc_data DTC_TX_R[DTC_TX_CHAIN_SIZE];    /* DTC transfer information (R-ch transmit) */
struct st_dtc_data DTC_RX_L[DTC_RX_CHAIN_SIZE];    /* DTC transfer information (L-ch receive) */
struct st_dtc_data DTC_RX_R[DTC_RX_CHAIN_SIZE];    /* DTC transfer information (R-ch receive) */

```

**Figure 4.29 Structures and Unions Used by Sample Program**

## 4.5 List of Variables

Table 4.3 lists the global variables and table 4.4 the **const** variables.

**Table 4.3 Global Variables**

Type	Variable Name	Description	Functions Used By
unsigned long	tx_au_data[AUD_NUM][2]	Transmit audio data	i2s_au_data_init i2s_start i2s_dtc_tx_l_init i2s_dtc_tx_r_init
unsigned long	rx_au_data[AUD_NUM][2]	Receive audio data	i2s_au_data_init i2s_dtc_rx_l_init i2s_dtc_rx_r_init
unsigned long	DTC_VECT_TABLE[256]	DTC vector table	i2s_dtc_init

**Table 4.4 const Variables**

Type	Variable Name	Description	Functions Used By
const unsigned char	RSPI_TX_DISABLE	Register setting values for RSPI initialization (transmit operation)	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const unsigned char	RSPI_TX_ENABLE	Register setting values for RSPI initialization cancellation (transmit operation)	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const unsigned char	RSPI_RX_DISABLE	Register setting values for RSPI initialization (receive operation)	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const unsigned char	RSPI_RX_ENABLE	Register setting values for RSPI initialization cancellation (receive operation)	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const unsigned char	RSPI_TRX_DISABLE	Register setting values for RSPI initialization (transceive operation)	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const unsigned char	RSPI_TRX_ENABLE	Register setting values for RSPI initialization cancellation (transceive operation)	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const unsigned char	DTC_CHAIN_DISABLE	Register setting values for disabling chained transfers	i2s_dtc_tx_l_init i2s_dtc_tx_r_init

## 4.6 List of Functions

Table 4.5 lists the functions.

**Table 4.5 Functions**

Function	Description
main	I <sup>2</sup> S communication main function
i2s_mcu_init	CPU initial settings
i2s_au_data_init	Audio data creation in on-chip RAM
i2s_start	I <sup>2</sup> S communication start process
i2s_dtc_init	DTC setting function
i2s_dtc_tx_l_init	DTC transfer information creation (L-ch transmit, RSPI0 initialization)
i2s_dtc_tx_r_init	DTC transfer information creation (R-ch transmit, RSPI1 initialization)
i2s_dtc_rx_l_init	DTC transfer information creation (L-ch receive)
i2s_dtc_rx_r_init	DTC transfer information creation (R-ch receive)
i2s_mtu2_init	MTU2 setting function
i2s_mtu2_ch2_init	MTU2 channel 2 setting function (SCK generation)
i2s_mtu2_ch3_init	MTU2 channel 3 setting function (WS generation)
i2s_mtu2_ch4_init	MTU2 channel 4 setting function (SSL generation)
i2s_rspi_init	RSPI setting function
i2s_rspi0_init	RSPI0 setting function (L-ch transceive)
i2s_rspi1_init	RSPI1 setting function (R-ch transceive)

## 4.7 Function Specifications

The specifications of the functions used by the sample program are listed below.

<b>main</b>	
Overview	This is the main function for I <sup>2</sup> S communication operations.
Header	i2s.h
Declaration	void main(void)
Description	
Arguments	None
Return value	None
Notes	

<b>i2s_mcu_init</b>	
Overview	This function makes initial settings to the CPU.
Header	iodefine_RX62N.h
Declaration	void i2s_mcu_init(void)
Description	<ul style="list-style-type: none"> <li>• Sets the operating frequencies.</li> <li>• Cancels module stop for the DTC, MTU2, and RSPI.</li> </ul>
Arguments	None
Return value	None
Notes	

<b>i2s_aud_data_init</b>	
Overview	This function creates audio data in the on-chip RAM.
Header	i2s.h
Declaration	void i2s_data_init(void)
Description	<ul style="list-style-type: none"> <li>• Creates transmit audio data (tx_aud_data).</li> <li>• Clears to 0 the receive audio data area (rx_aud_data).</li> </ul>
Arguments	None
Return value	None
Notes	

<b>i2s_start</b>	
Overview	This function starts I <sup>2</sup> S communication.
Header	i2s.h, iodefine_RX62N.h
Declaration	void i2s_start(char i2s_mode)
Description	<ul style="list-style-type: none"> <li>• Enables the RSPI function.</li> <li>• Starts the DTC module.</li> <li>• Starts MTU2 count operation</li> </ul>
Arguments	1st argument: i2s_mode: Operation mode selection
Return value	None
Notes	

<b>i2s_dtc_init</b>	
Overview	This function makes initial settings to the DTC.
Header	iodefine_RX62N.h
Declaration	void i2s_dtc_init(char i2s_mode)
Description	<ul style="list-style-type: none"> <li>Creates the DTC vector table.</li> <li>Enables DTC activation by interrupt.</li> </ul>
Arguments	1st argument: i2s_mode: Operation mode selection
Return value	None
Notes	

<b>i2s_dtc_tx_l_init</b>	
Overview	This function creates DTC transfer information (L-ch transmit, RSPI0 initialization).
Header	i2s.h, iodefine_RX62N.h
Declaration	void i2s_dtc_tx_l_init(char i2s_mode)
Description	Transfers transmit data from on-chip RAM to RSPI0.
Arguments	1st argument: i2s_mode: Operation mode selection
Return value	None
Notes	

<b>i2s_dtc_tx_r_init</b>	
Overview	This function creates DTC transfer information (R-ch transmit, RSPI1 initialization)
Header	i2s.h, iodefine_RX62N.h
Declaration	void i2s_dtc_tx_r_init(char i2s_mode)
Description	Transfers transmit data from on-chip RAM to RSPI1.
Arguments	1st argument: i2s_mode: Operation mode selection
Return value	None
Notes	

<b>i2s_dtc_rx_l_init</b>	
Overview	This function creates DTC transfer information (L-ch receive).
Header	i2s.h, iodefine_RX62N.h
Declaration	void i2s_dtc_rx_l_init(void)
Description	Transfers receive data from RSPI0 to on-chip RAM.
Arguments	None
Return value	None
Notes	

<b>i2s_dtc_rx_r_init</b>	
Overview	This function creates DTC transfer information (R-ch receive).
Header	i2s.h, iodefine_RX62N.h
Declaration	void i2s_dtc_rx_r_init(void)
Description	Transfers receive data from RSPI1 to on-chip RAM.
Arguments	None
Return value	None
Notes	

<b>i2s_mtu2_init</b>	
Overview	This function makes initial settings to the MTU2.
Header	iodefine_RX62N.h
Declaration	void i2s_mtu2_init(void)
Description	Makes MTU2 port settings.
Arguments	None
Return value	None
Notes	

<b>i2s_mtu2_ch2_init</b>	
Overview	This function makes initial settings to MTU2 channel 2 (SCK generation).
Header	i2s.h, iodefine_RX62N.h
Declaration	void i2s_mtu2_ch2_init(void)
Description	Generates SCK from the external clock and outputs it.
Arguments	None
Return value	None
Notes	

<b>i2s_mtu2_ch3_init</b>	
Overview	This function makes initial settings to MTU2 channel 3 (WS generation).
Header	i2s.h, iodefine_RX62N.h
Declaration	void i2s_mtu2_ch3_init(void)
Description	Generates WS from the external clock and outputs it.
Arguments	None
Return value	None
Notes	

<b>i2s_mtu2_ch4_init</b>	
Overview	This function makes initial settings to MTU2 channel 4 (SSL generation).
Header	i2s.h, iodefine_RX62N.h
Declaration	void i2s_mtu2_ch4_init(void)
Description	<ul style="list-style-type: none"> <li>Generates SSL from the external clock and outputs it.</li> <li>Enables compare-match interrupts.</li> </ul>
Arguments	None
Return value	None
Notes	

<b>i2s_rspi_init</b>	
Overview	This function makes initial settings to the RSPI.
Header	None
Declaration	void i2s_rspi_init(char i2s_mode)
Description	
Arguments	1st argument: i2s_mode: Operation mode selection
Return value	None
Notes	

<b>i2s_rspi0_init</b>	
Overview	This function makes initial settings to RSPI0 (L-ch transmit/receive).
Header	iodefine_RX62N.h
Declaration	void i2s_rspi0_init(char i2s_mode)
Description	Makes settings to RSPI0.
Arguments	1st argument: i2s_mode: Operation mode selection
Return value	None
Notes	

<b>i2s_rspi1_init</b>	
Overview	This function makes initial settings to RSPI1 (R-ch transmit/receive).
Header	iodefine_RX62N.h
Declaration	void i2s_rspi1_init(char i2s_mode)
Description	Makes settings to RSPI1.
Arguments	1st argument: i2s_mode: Operation mode selection
Return value	None
Notes	

## 4.8 Flowcharts

### 4.8.1 main Function

Figure 4.30 is a flowchart of the main function.

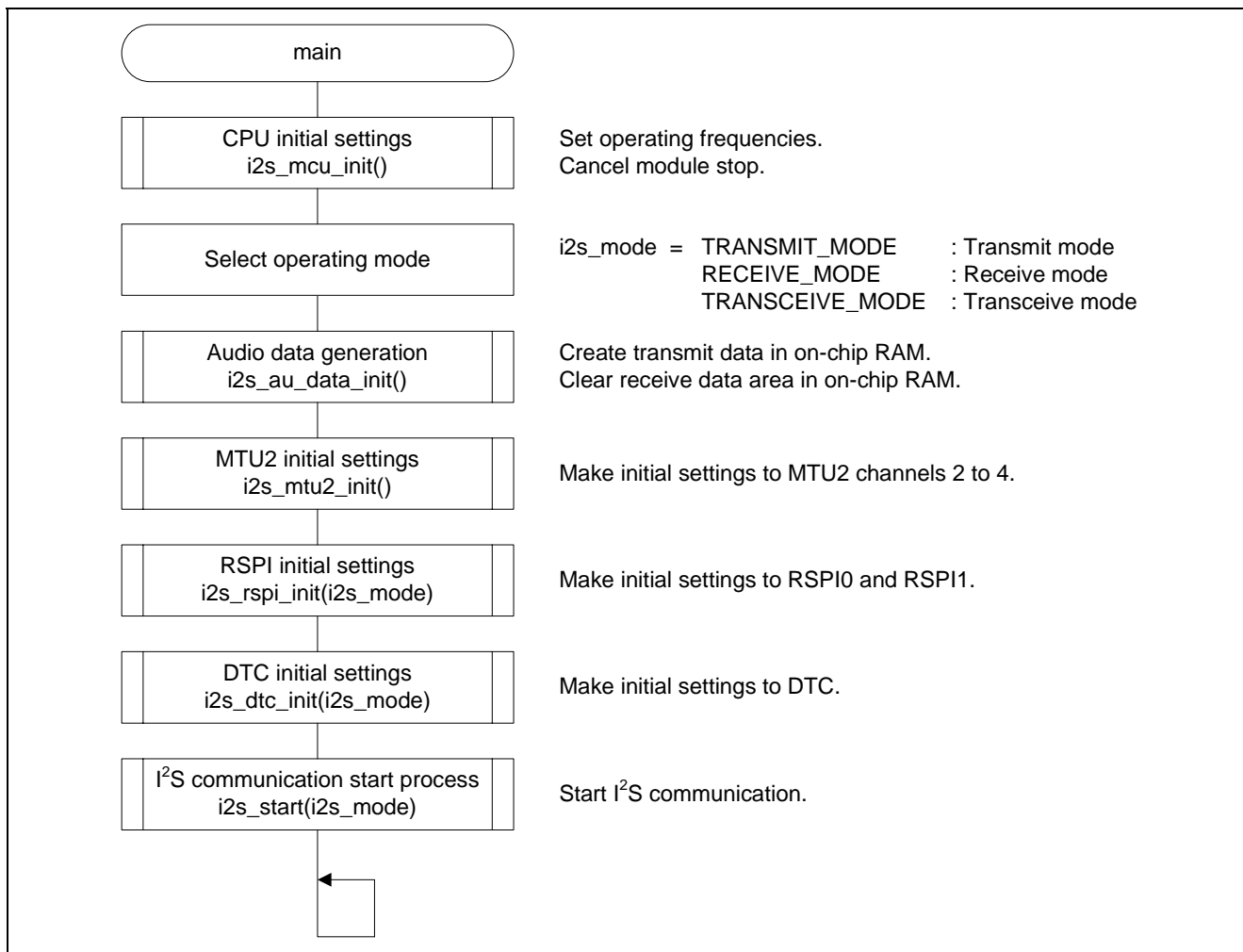
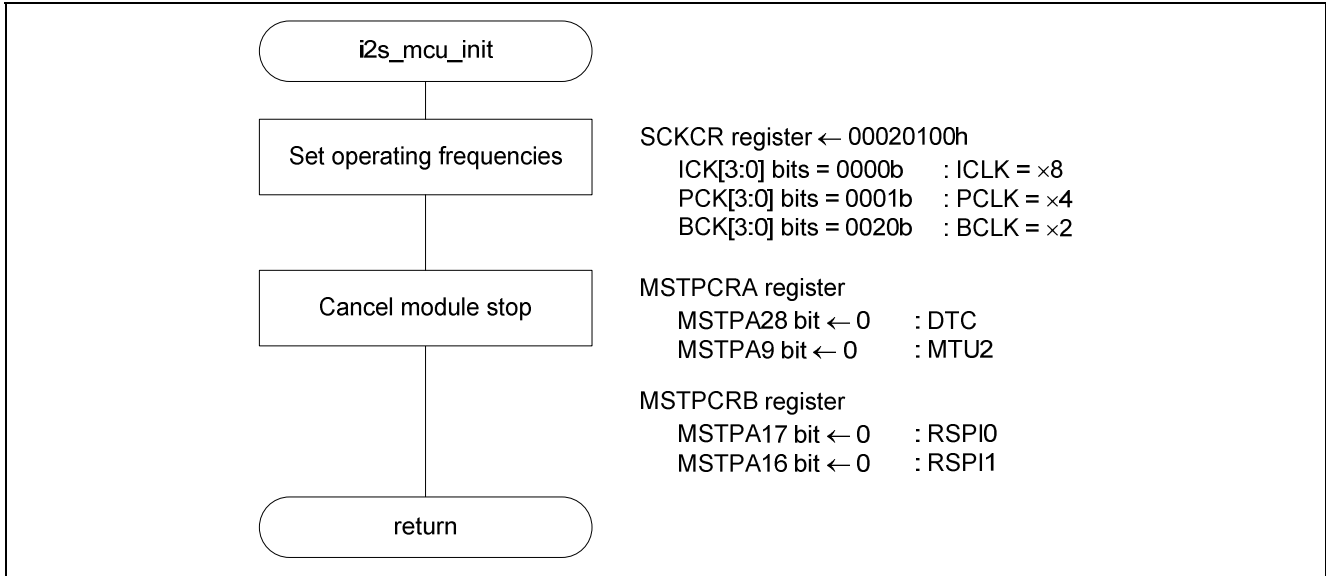


Figure 4.30 main Function

**4.8.2 i2s\_mcu\_init Function**

Figure 4.31 is a flowchart of the i2s\_mcu\_init function.

The i2s\_mcu\_init function makes initial settings to the CPU.

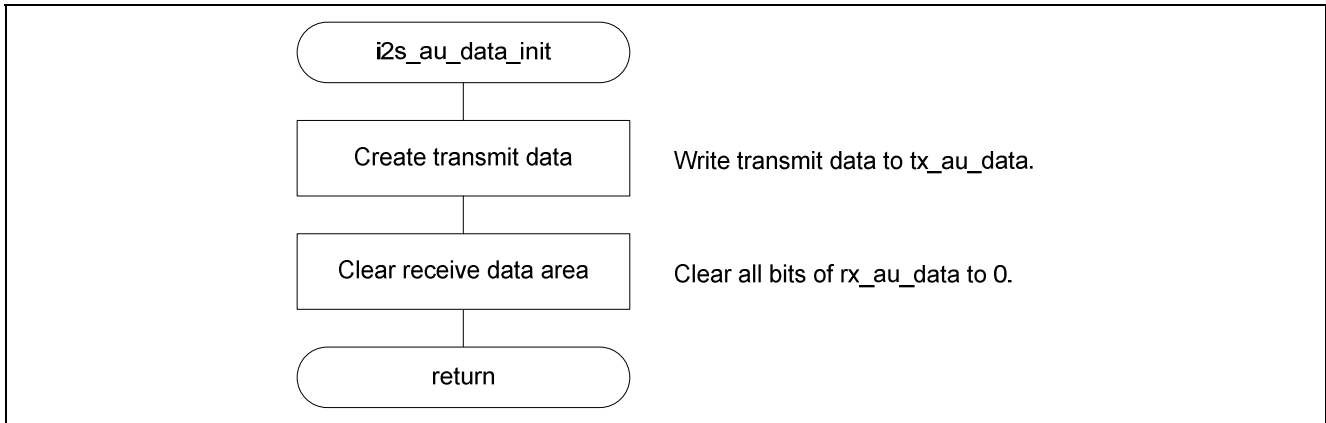


**Figure 4.31 i2s\_mcu\_init Function**

**4.8.3 i2s\_au\_data\_init Function**

Figure 4.32 is a flowchart of the i2s\_au\_data\_init function.

The i2s\_au\_data\_init function creates audio data in the on-chip RAM.



**Figure 4.32 i2s\_au\_data\_init Function**

4.8.4 i2s\_start Function

Figure 4.33 is a flowchart of the i2s\_start function.

The i2s\_start function enables peripheral functions and starts I<sup>2</sup>S communication.

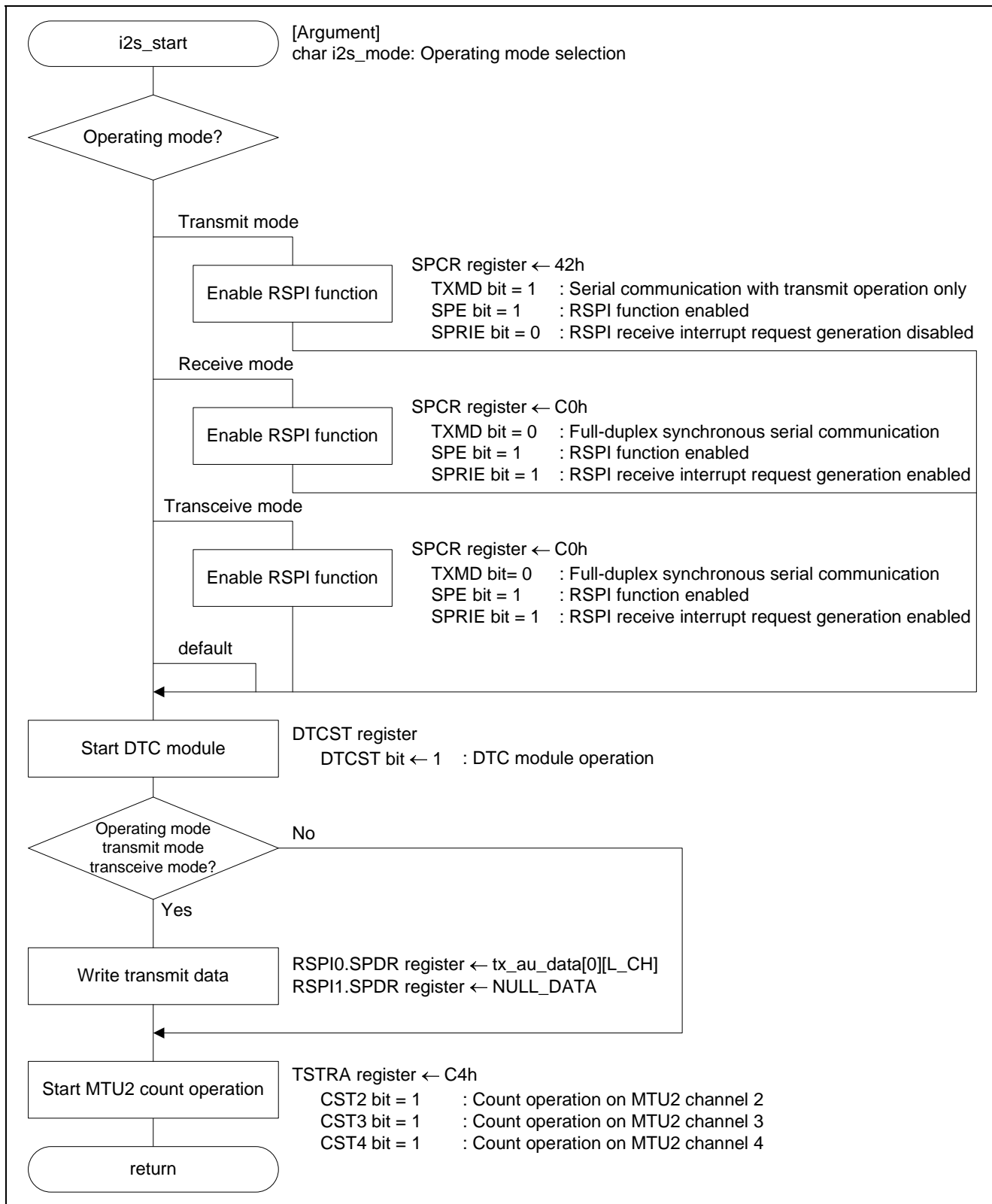


Figure 4.33 i2s\_start Function

4.8.5 i2s\_dtc\_init Function

Figure 4.34 is a flowchart of the i2s\_dtc\_init function.

The i2s\_dtc\_init function makes initial settings to the DTCa.

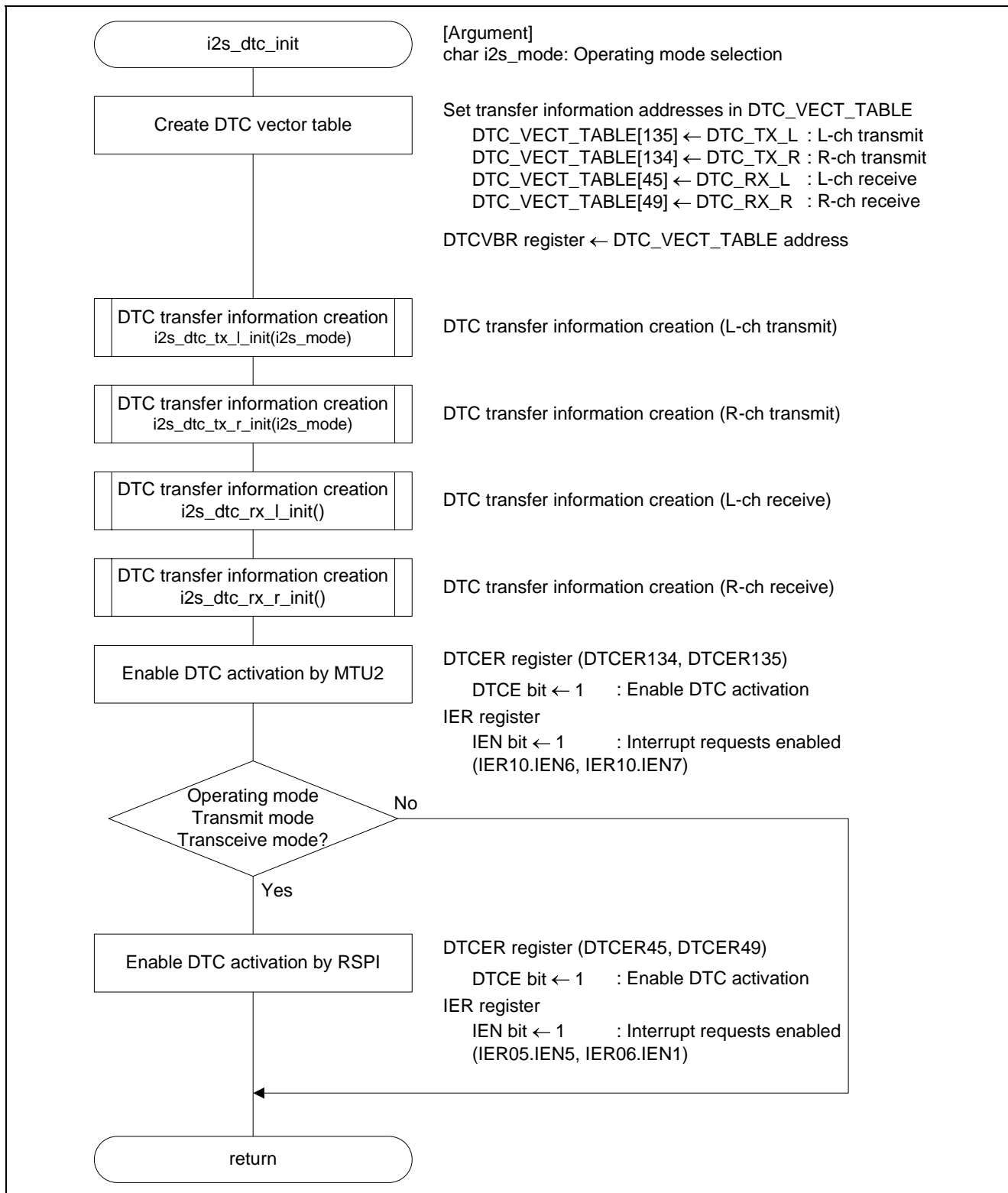


Figure 4.34 i2s\_dtc\_init Function

4.8.6 i2s\_dtc\_tx\_l\_init Function

Figure 4.35 is a flowchart of the i2s\_dtc\_tx\_l\_init function.

The i2s\_dtc\_tx\_l\_init function creates DTC transfer information for L-ch transmit (transmit mode or transceive mode) or RSPI0 initialization (receive mode).

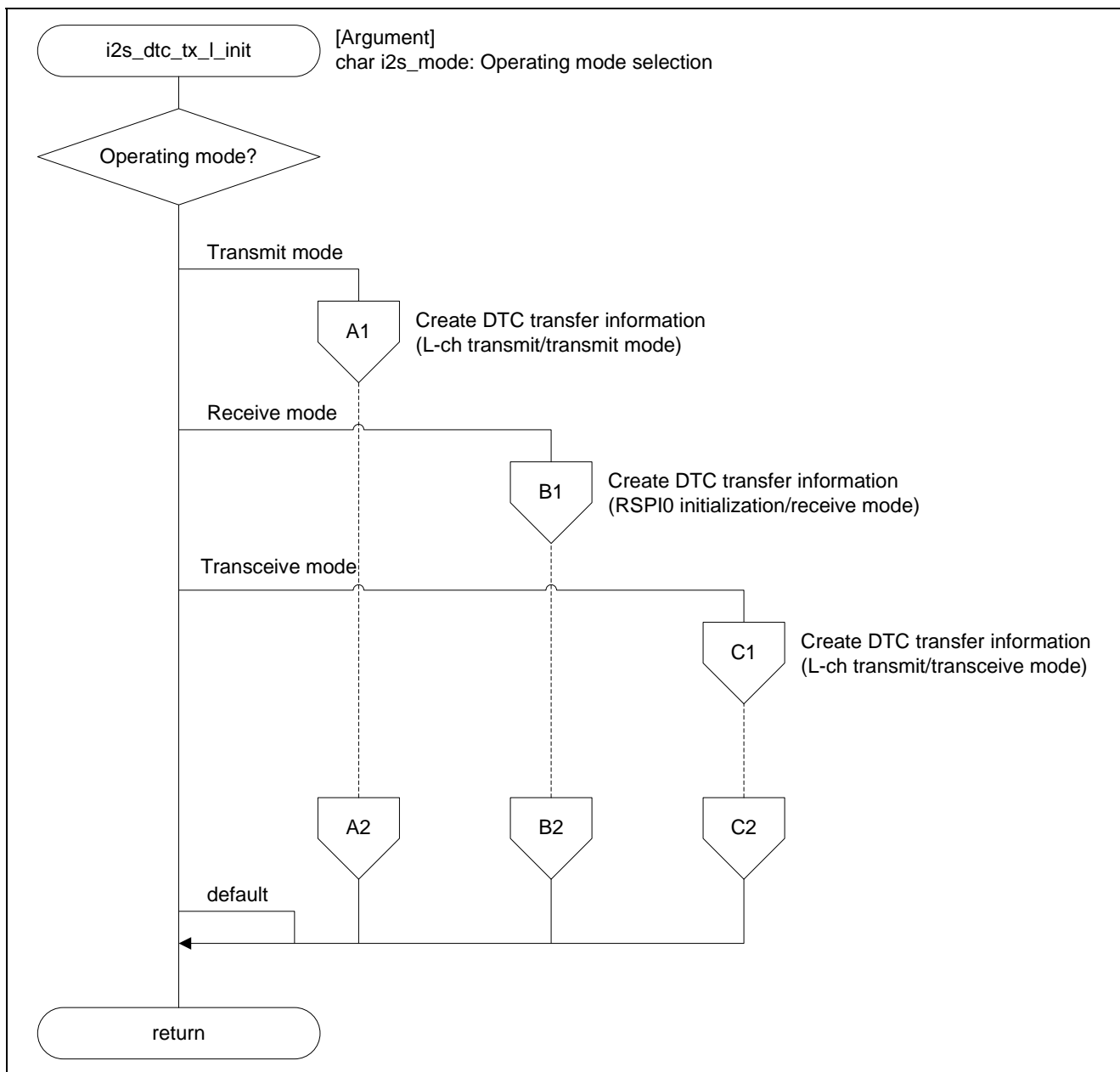


Figure 4.35 i2s\_dtc\_tx\_l\_init Function (1/4)

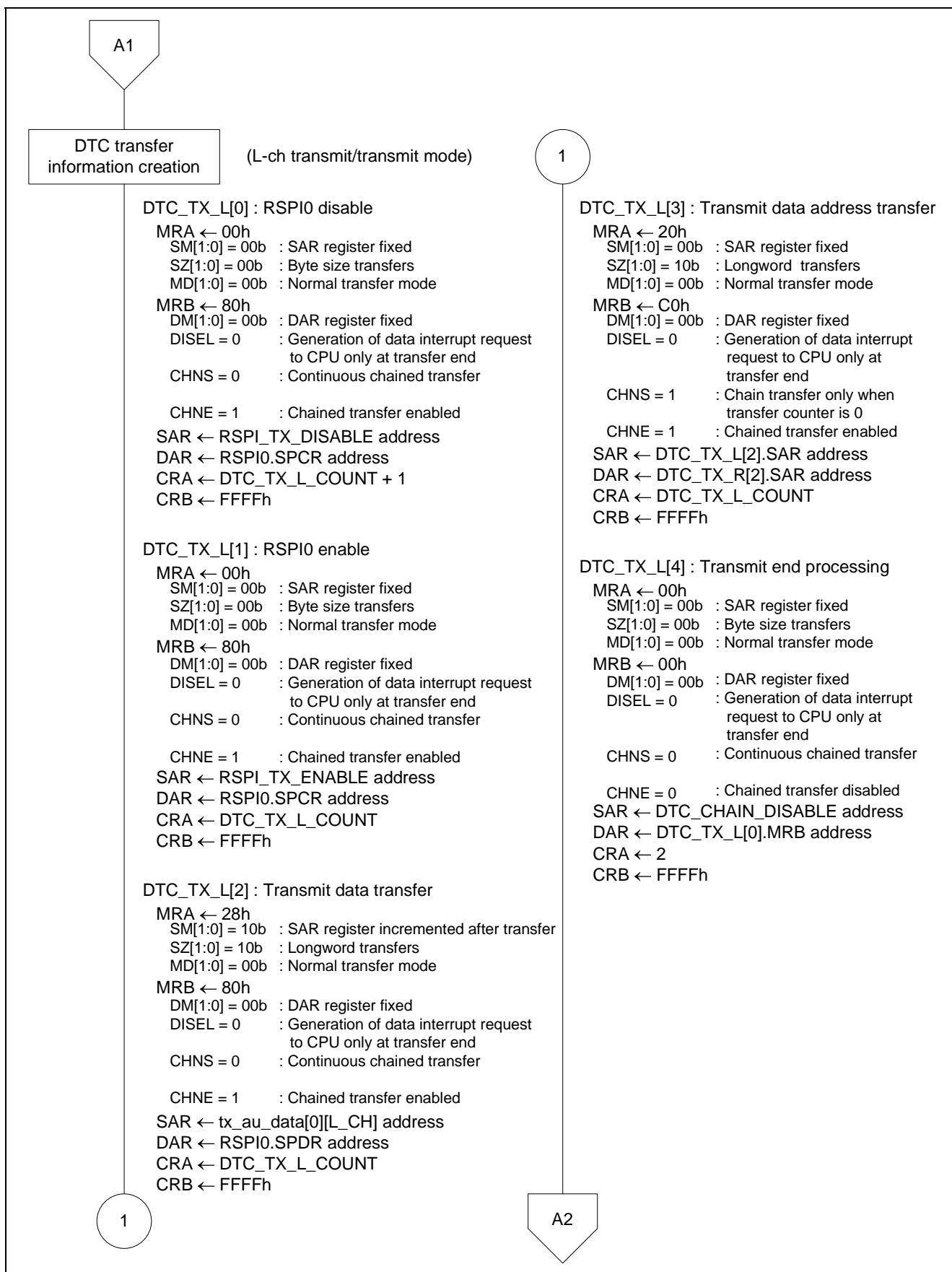


Figure 4.35 i2s\_dtc\_tx\_l\_init Function (2/4)

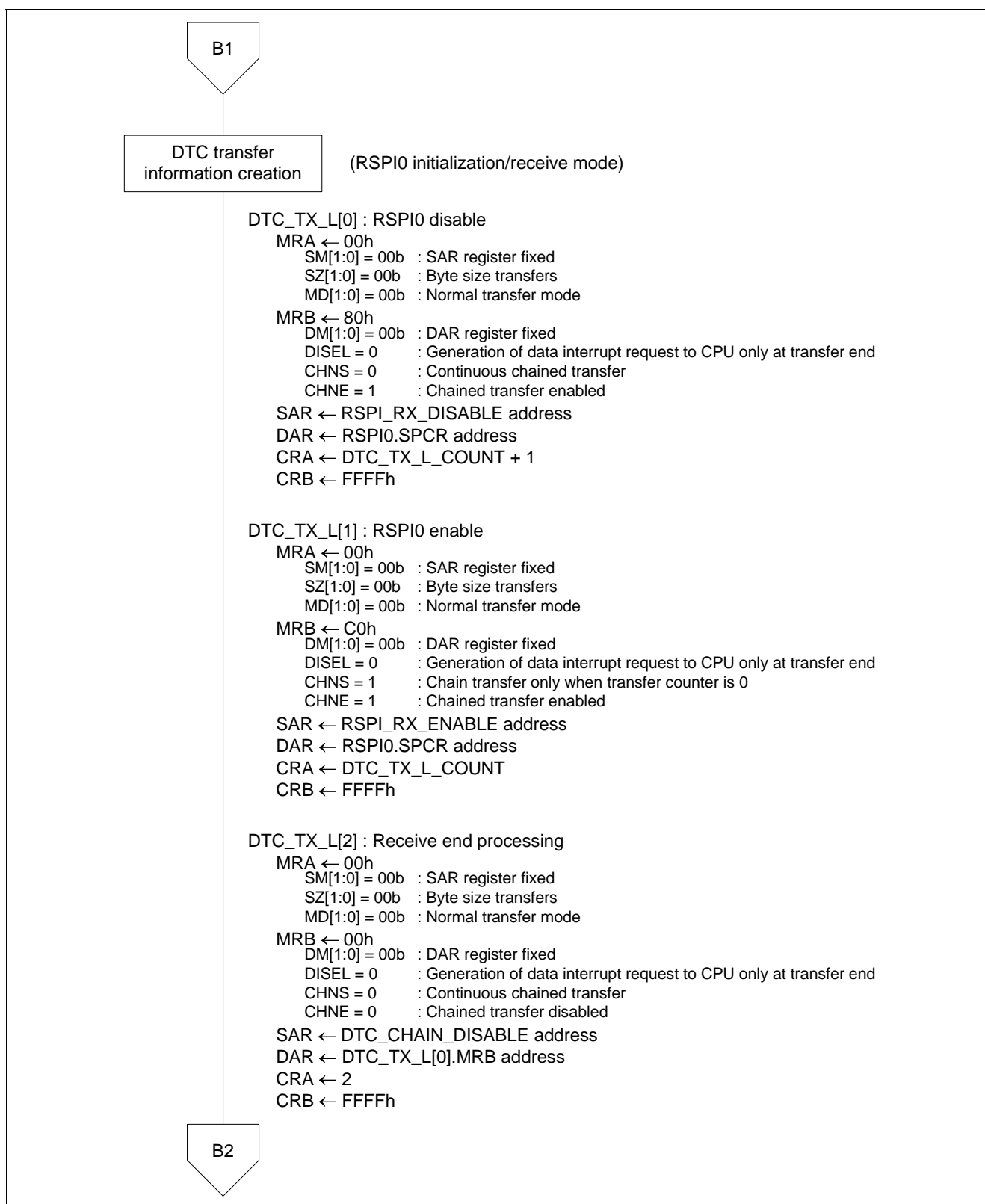


Figure 4.35 i2s\_dtc\_tx\_l\_init Function (3/4)

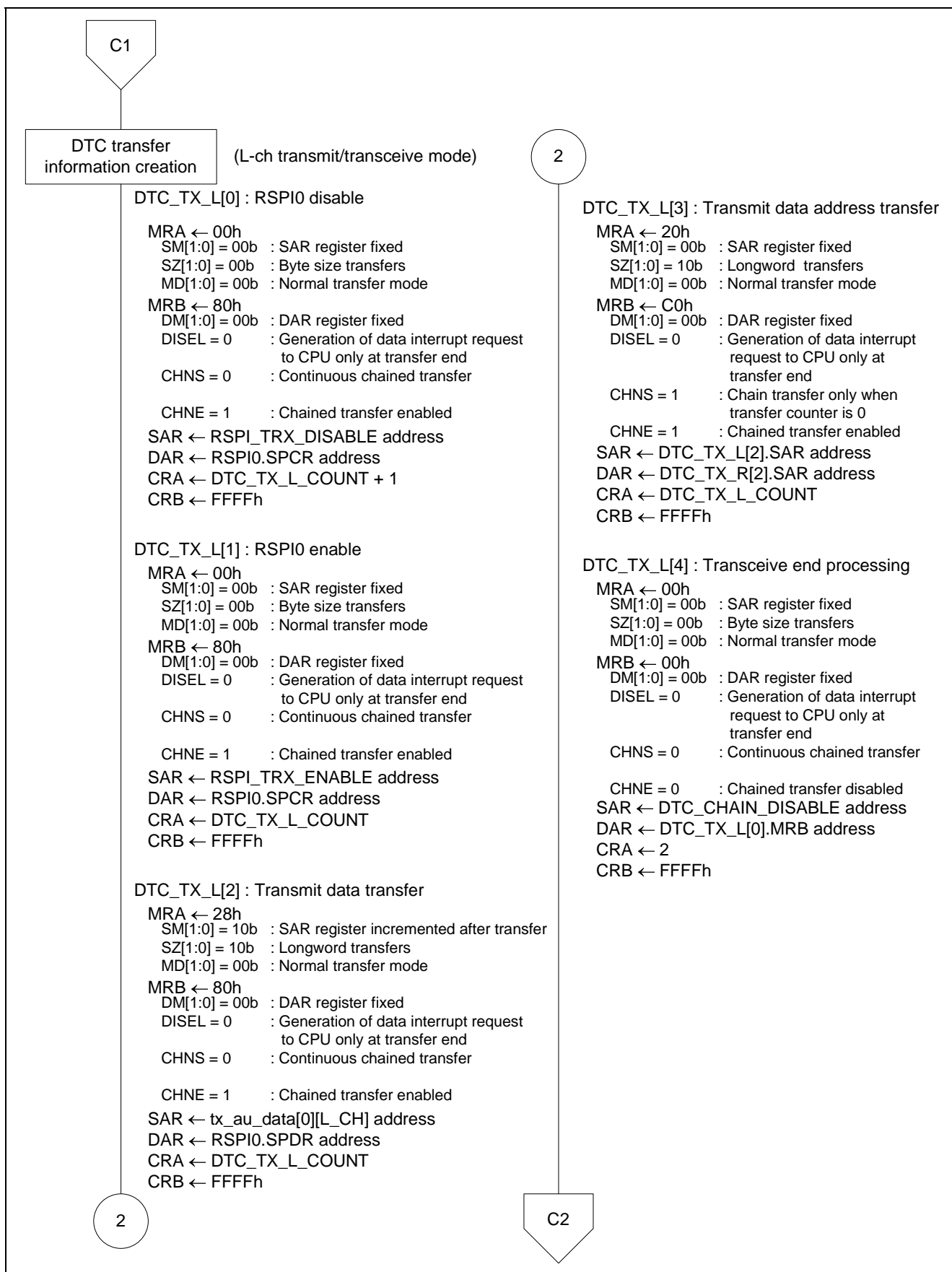


Figure 4.35 i2s\_dtc\_tx\_l\_init Function (4/4)

4.8.7 i2s\_dtc\_tx\_r\_init Function

Figure 4.36 is a flowchart of the i2s\_dtc\_tx\_r\_init function.

The i2s\_dtc\_tx\_r\_init function creates DTC transfer information for R-ch transmit (transmit mode or transceive mode) or RSPI1 initialization (receive mode).

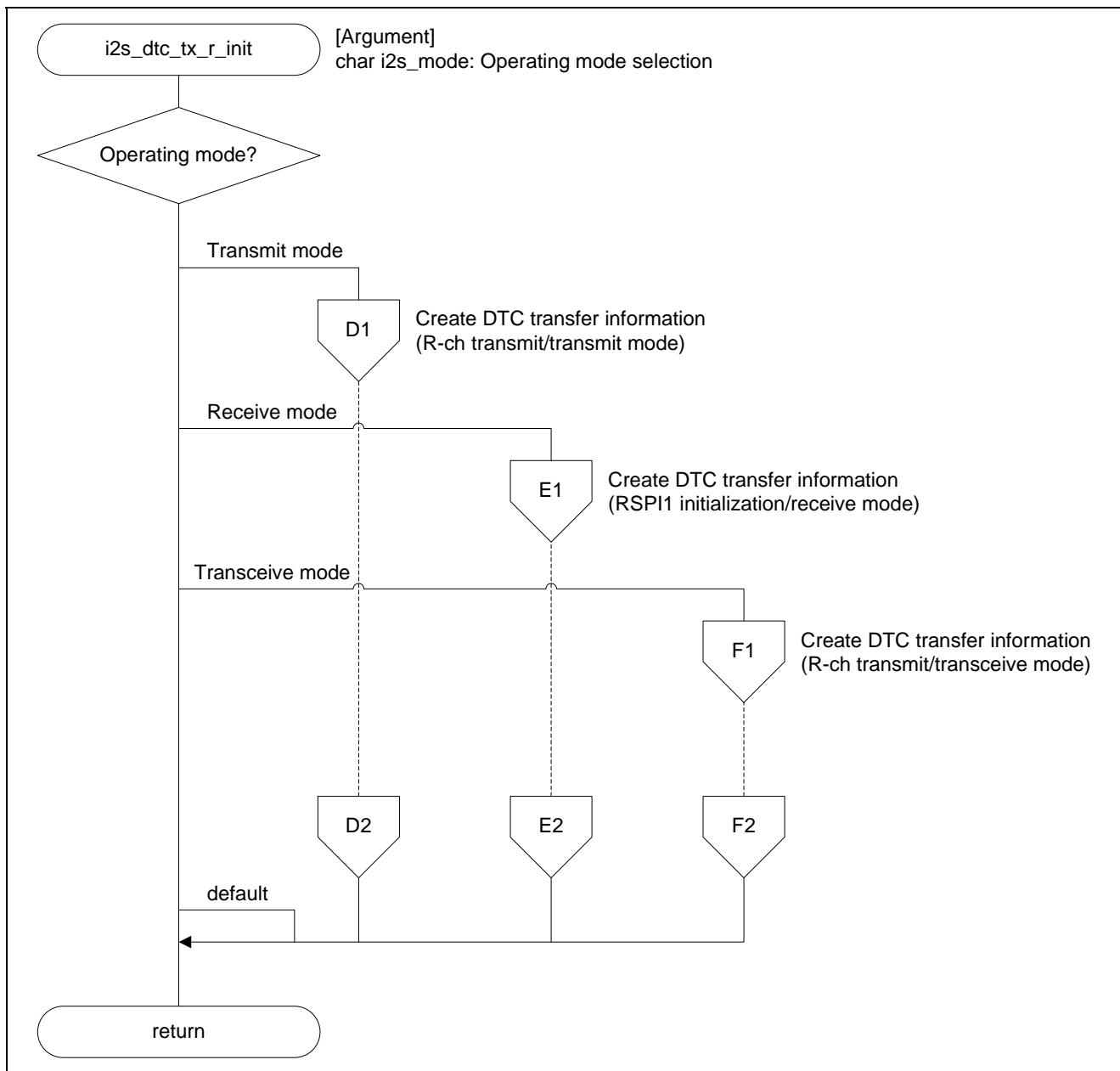


Figure 4.36 i2s\_dtc\_tx\_r\_init Function (1/4)

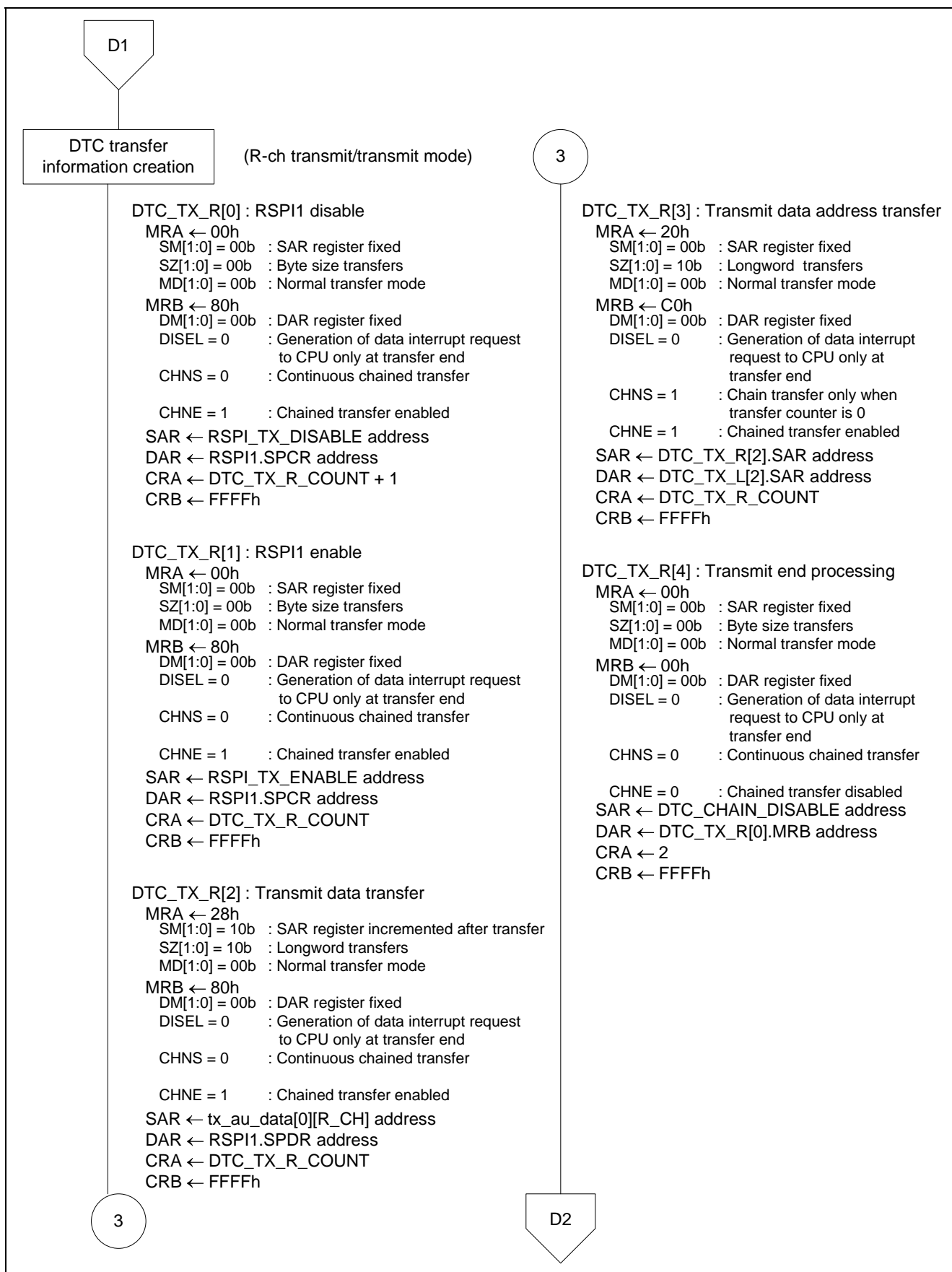


Figure 4.36 i2s\_dtc\_tx\_r\_init Function (2/4)

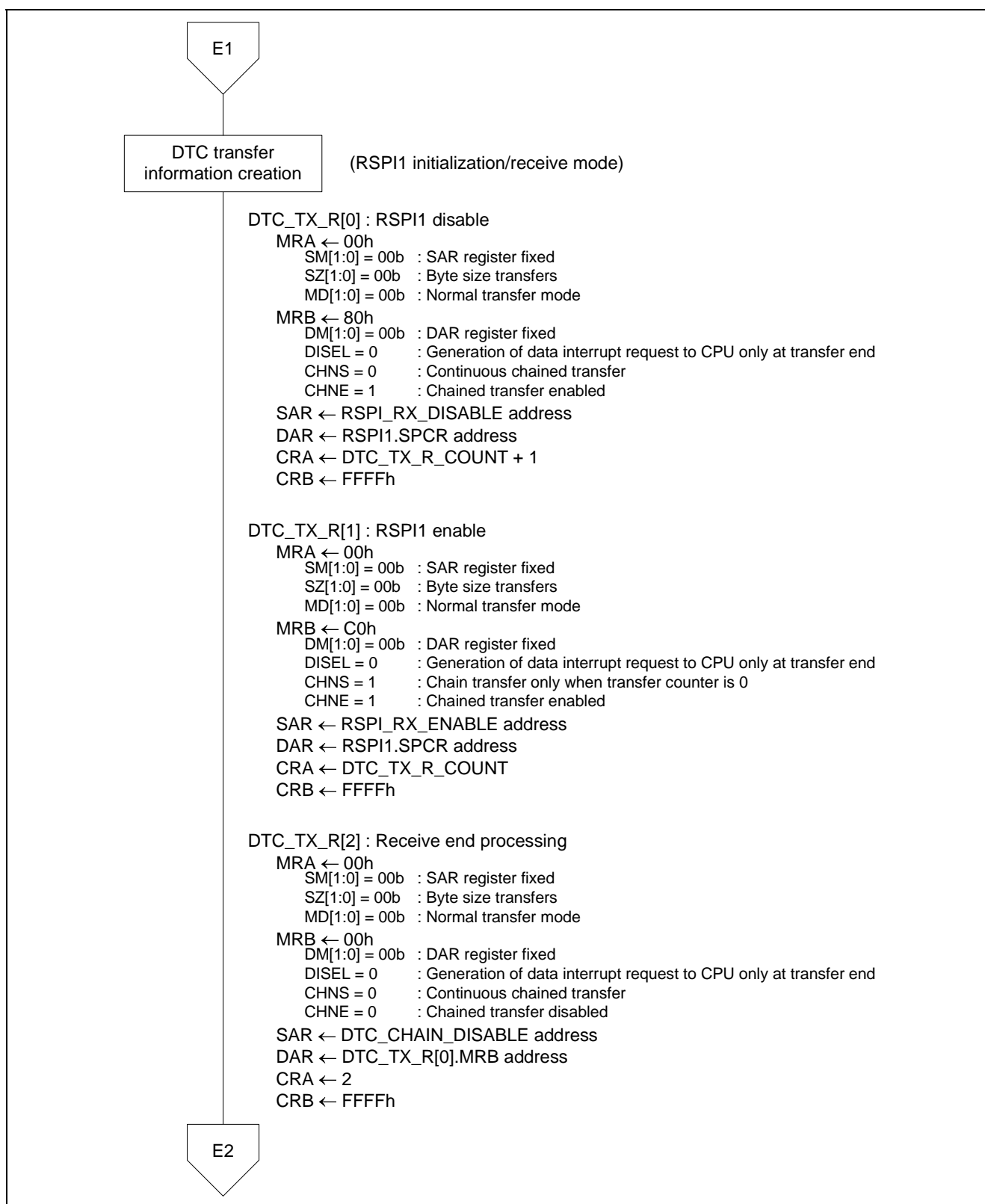


Figure 4.36 i2s\_dtc\_tx\_r\_init Function (3/4)

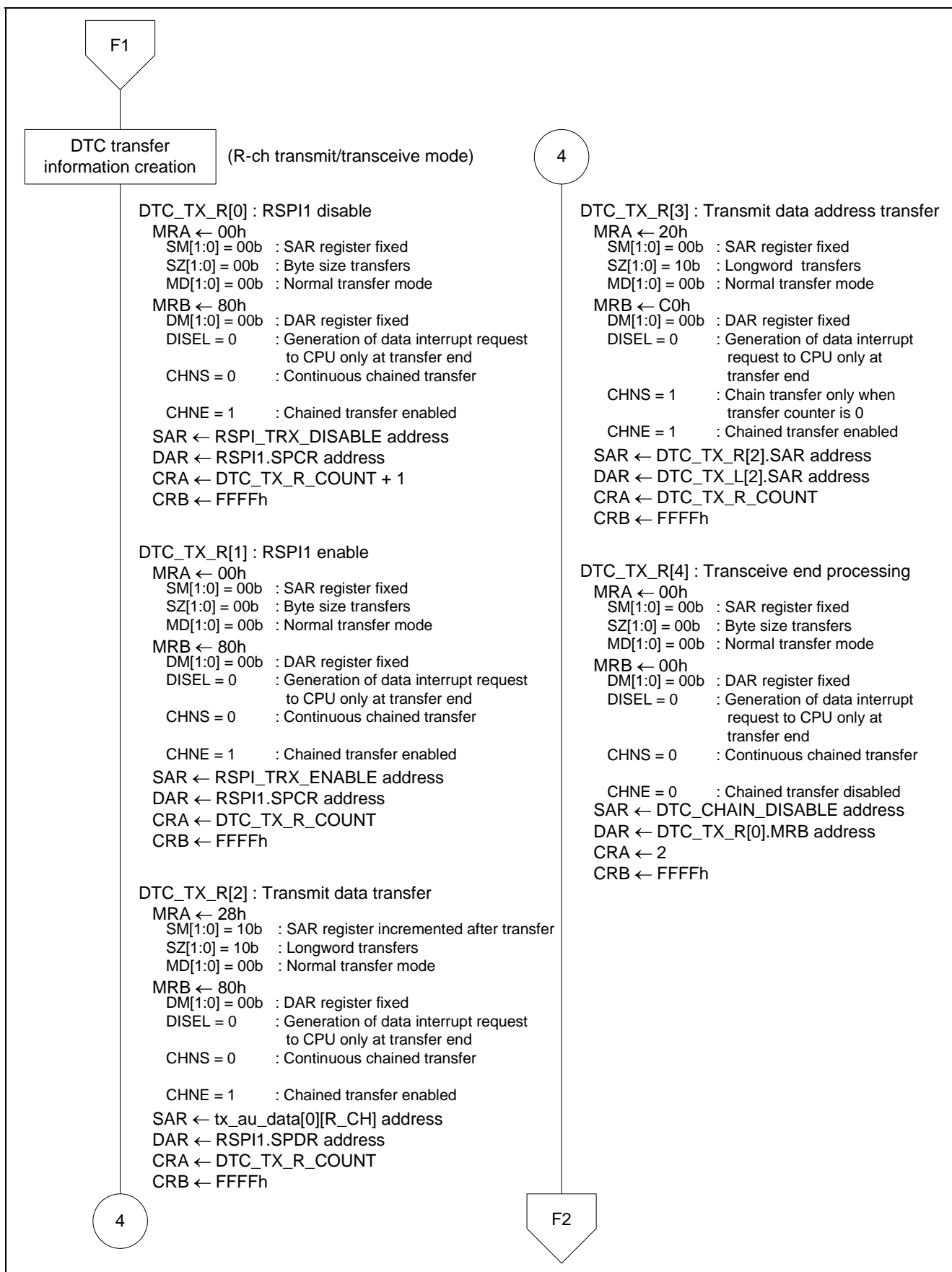


Figure 4.36 i2s\_dtc\_tx\_r\_init Function (4/4)

#### 4.8.8 i2s\_dtc\_rx\_l\_init Function

Figure 4.37 is a flowchart of the i2s\_dtc\_rx\_l\_init function.

The i2s\_dtc\_rx\_l\_init function creates DTC transfer information for L-ch receive.

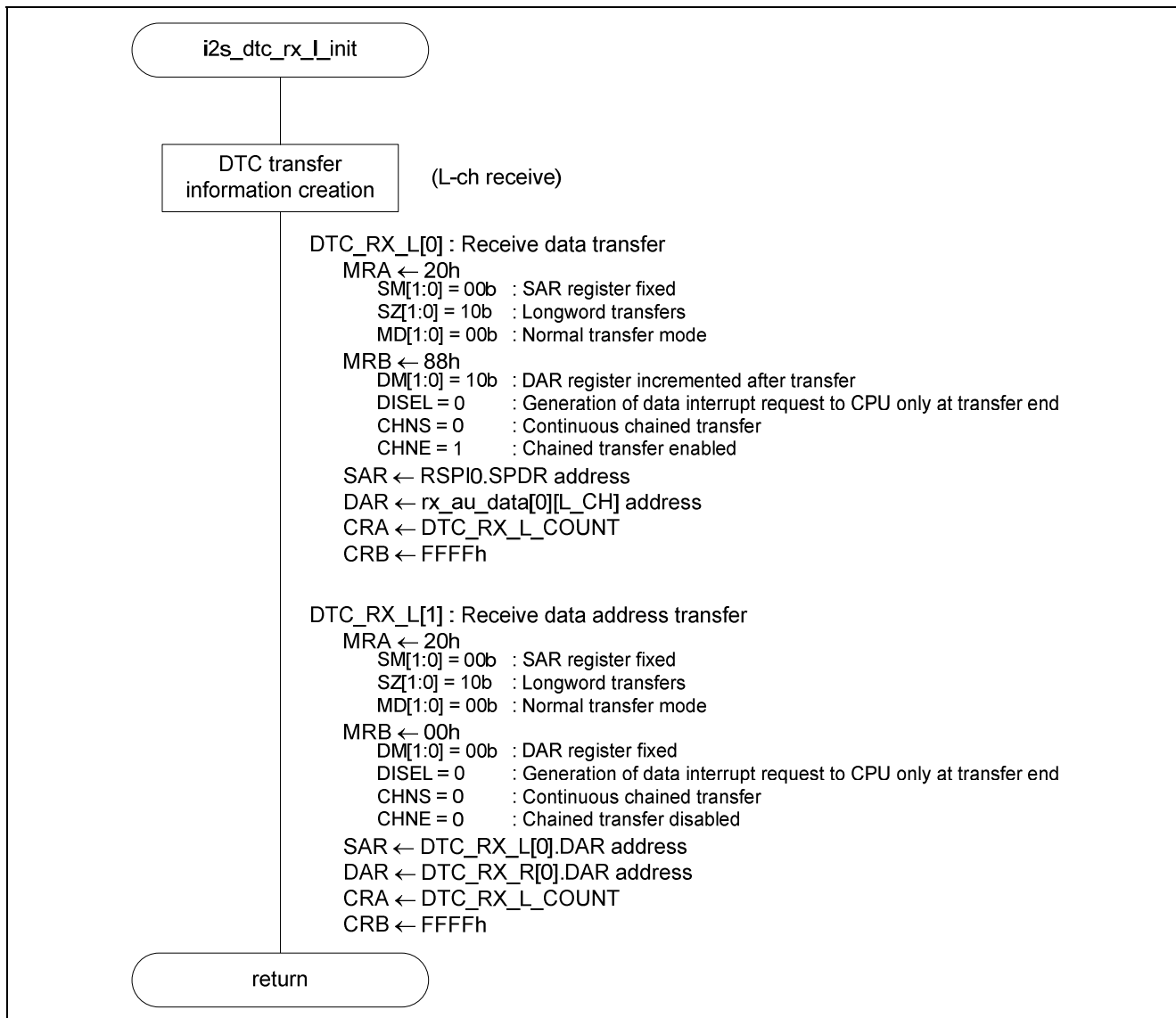


Figure 4.37 i2s\_dtc\_rx\_l\_init Function

#### 4.8.9 i2s\_dtc\_rx\_r\_init Function

Figure 4.38 is a flowchart of the i2s\_dtc\_rx\_r\_init function.

The i2s\_dtc\_rx\_r\_init function creates DTC transfer information for R-ch receive.

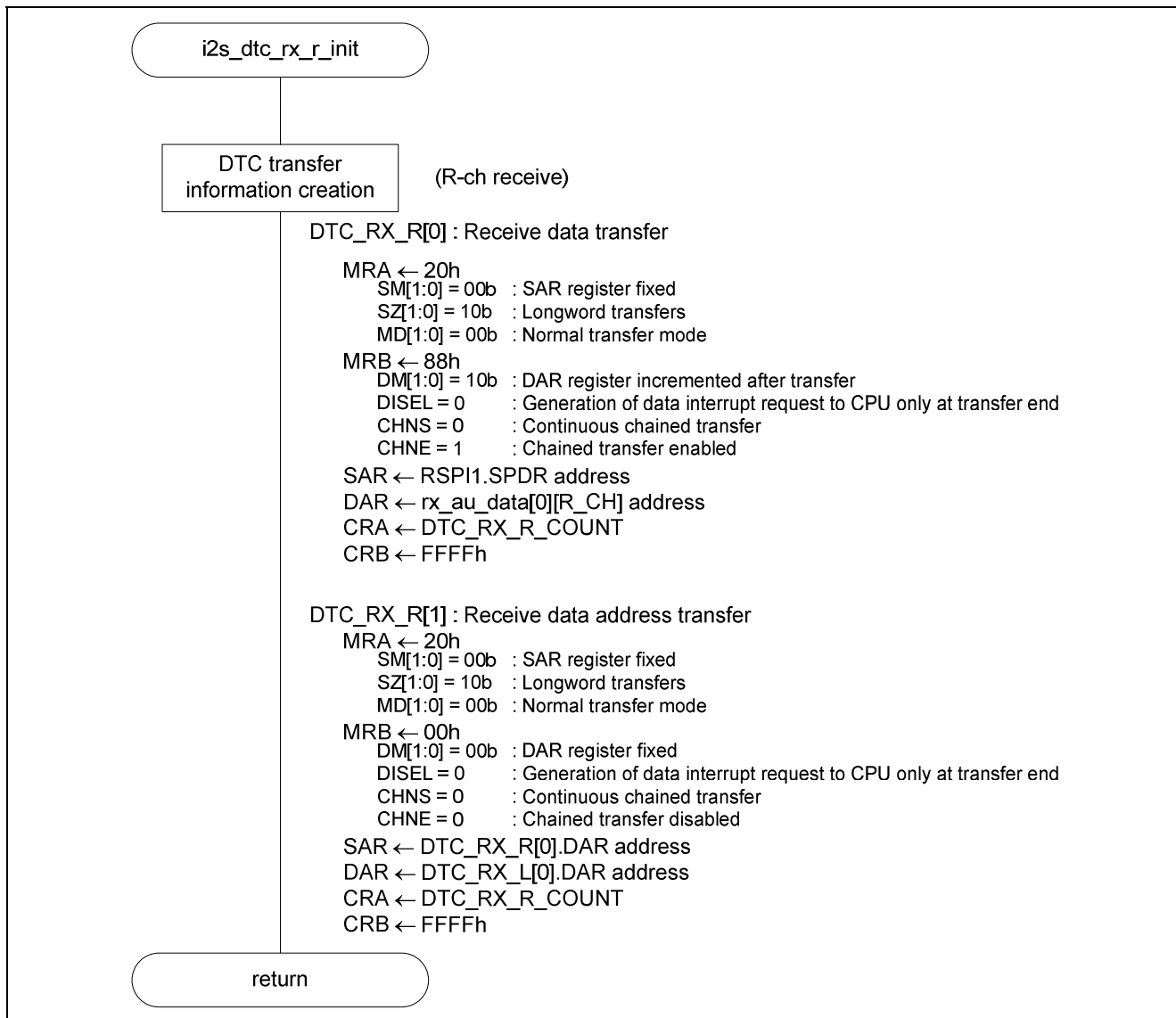


Figure 4.38 i2s\_dtc\_rx\_r\_init Function

4.8.10 i2s\_mtu2\_init Function

Figure 4.39 is a flowchart of the i2s\_mtu2\_init function.

The i2s\_mtu2\_init function makes initial settings to the MTU2.

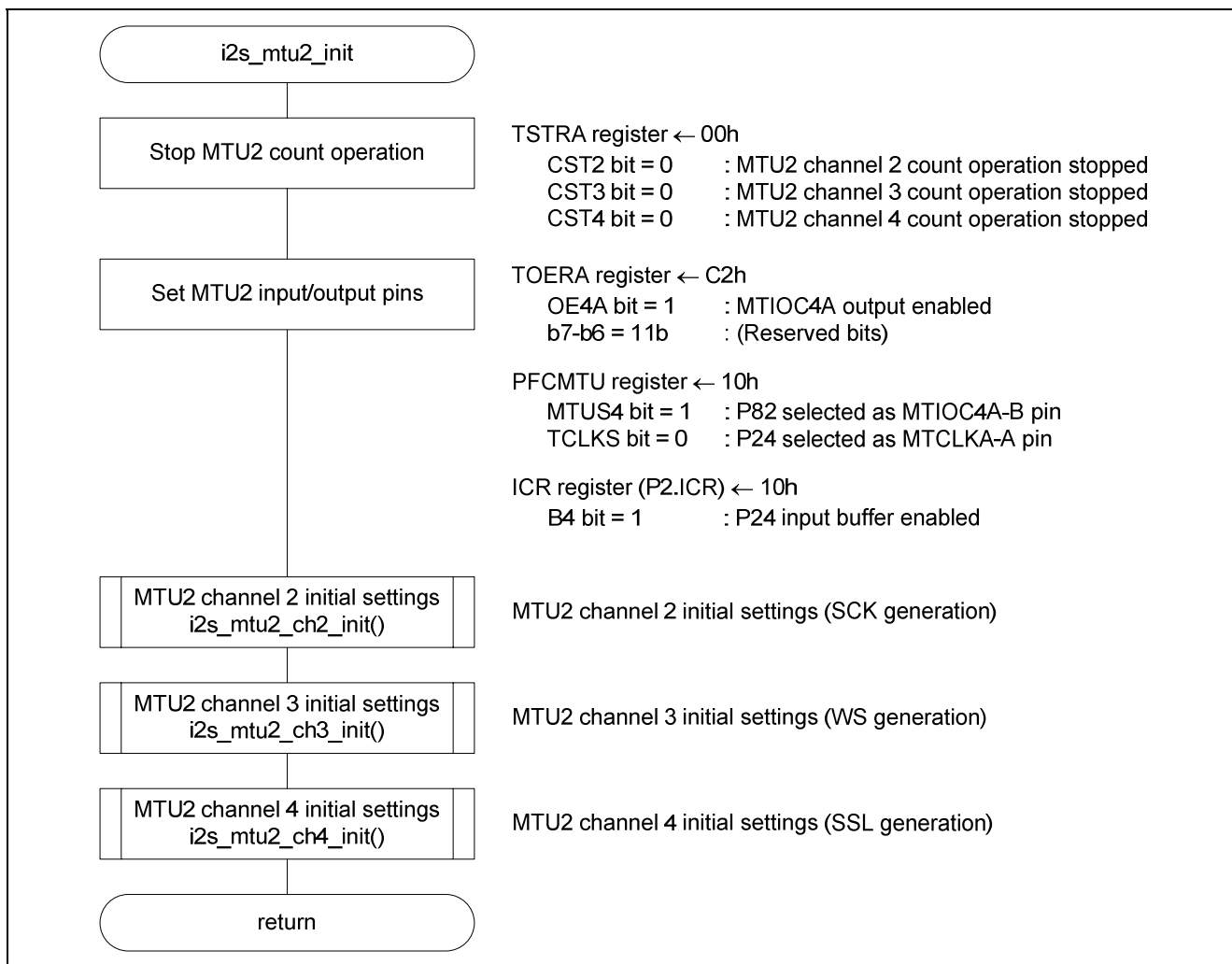


Figure 4.39 i2s\_mtu2\_init Function

4.8.11 i2s\_mtu2\_ch2\_init Function

Figure 4.40 is a flowchart of the i2s\_mtu2\_ch2\_init function.

The i2s\_mtu2\_ch2\_init function makes settings to enable generation of SCK by using MTU2 channel 2.

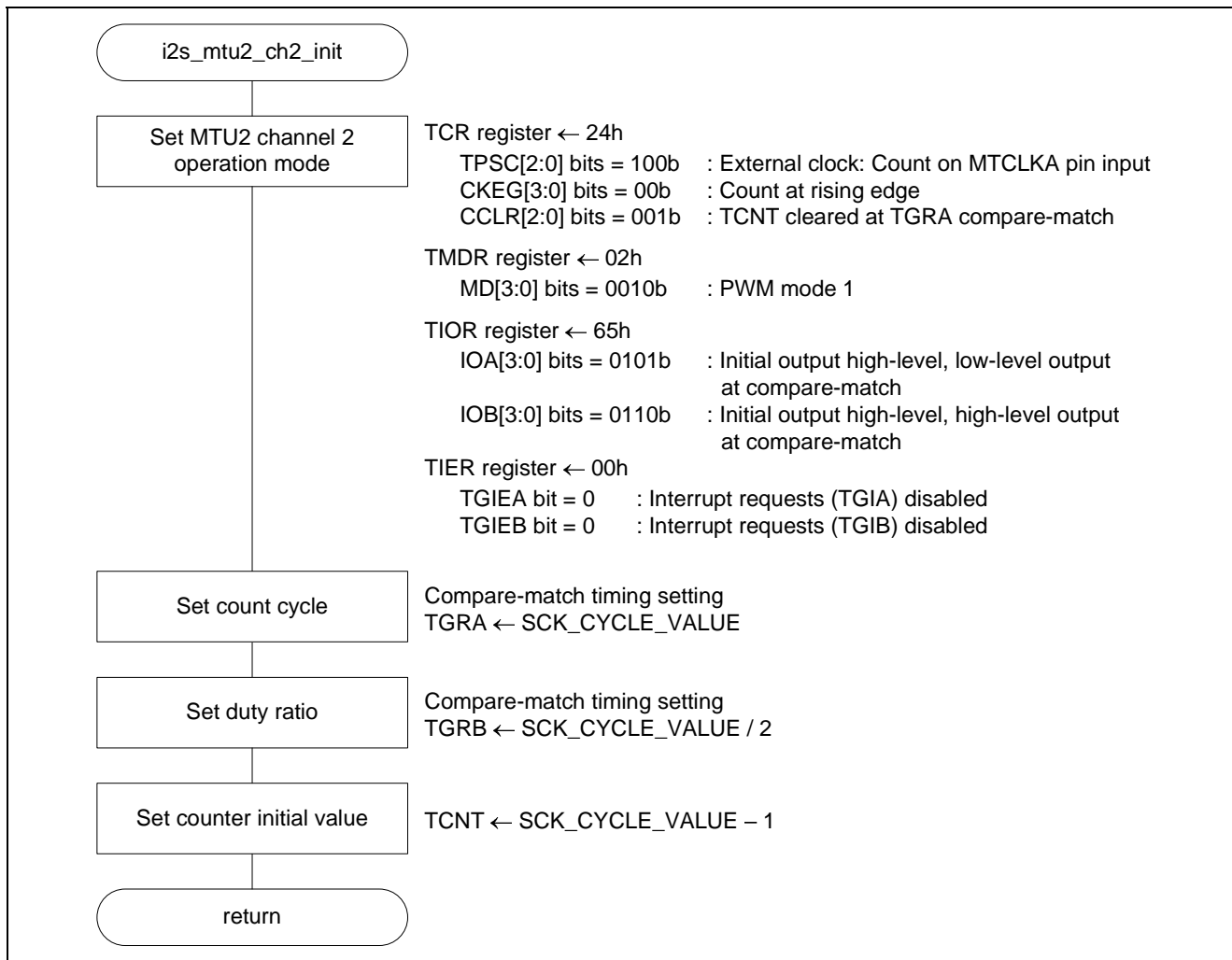


Figure 4.40 i2s\_mtu2\_ch2\_init Function

4.8.12 i2s\_mtu2\_ch3\_init Function

Figure 4.41 is a flowchart of the i2s\_mtu2\_ch3\_init function.

The i2s\_mtu2\_ch3\_init function makes settings to enable generation of WS by using MTU2 channel 3.

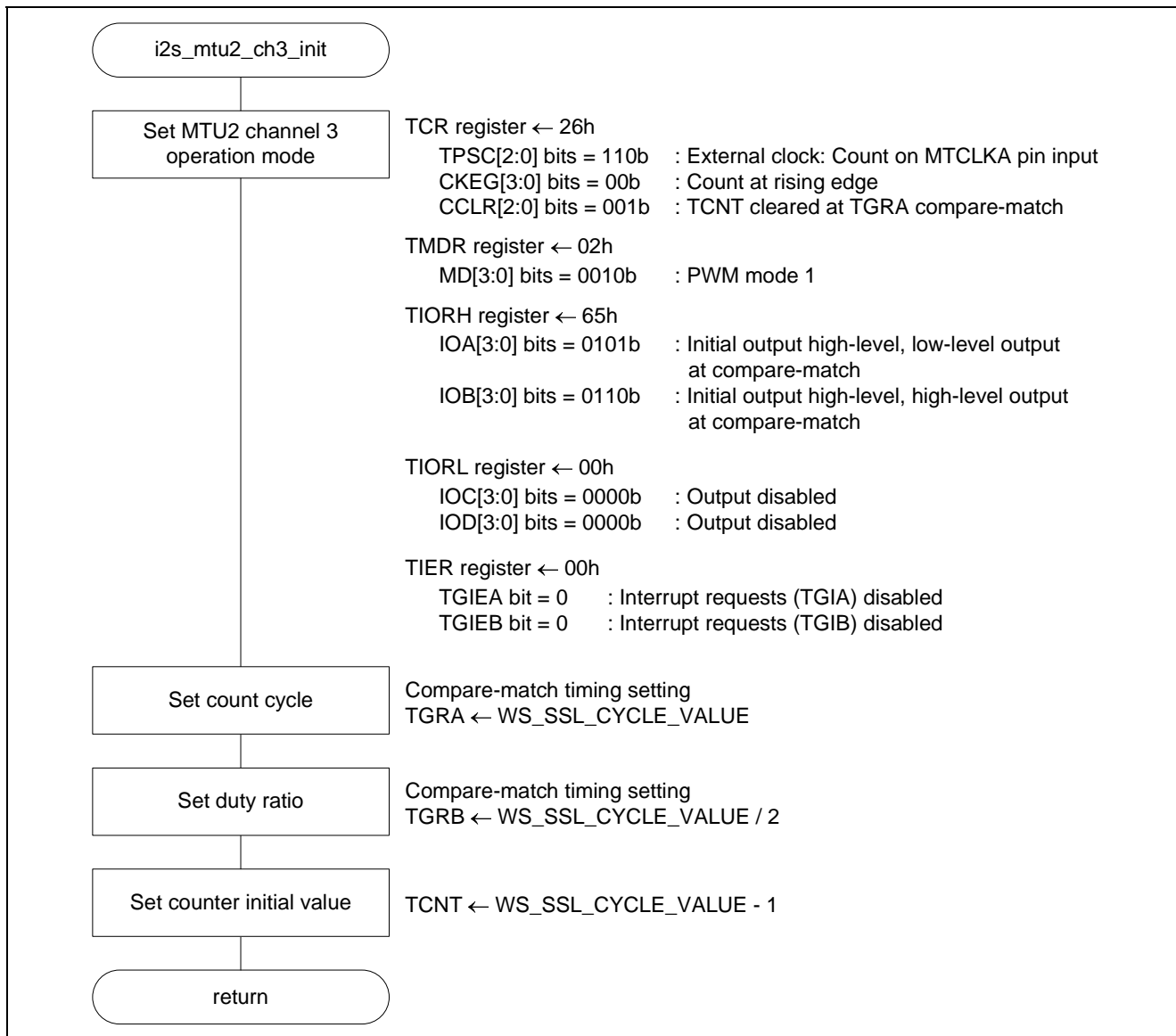


Figure 4.41 i2s\_mtu2\_ch3\_init Function

4.8.13 i2s\_mtu2\_ch4\_init Function

Figure 4.42 is a flowchart of the i2s\_mtu2\_ch4\_init function.

The i2s\_mtu2\_ch4\_init function makes settings to enable generation of SSL by using MTU2 channel 4.

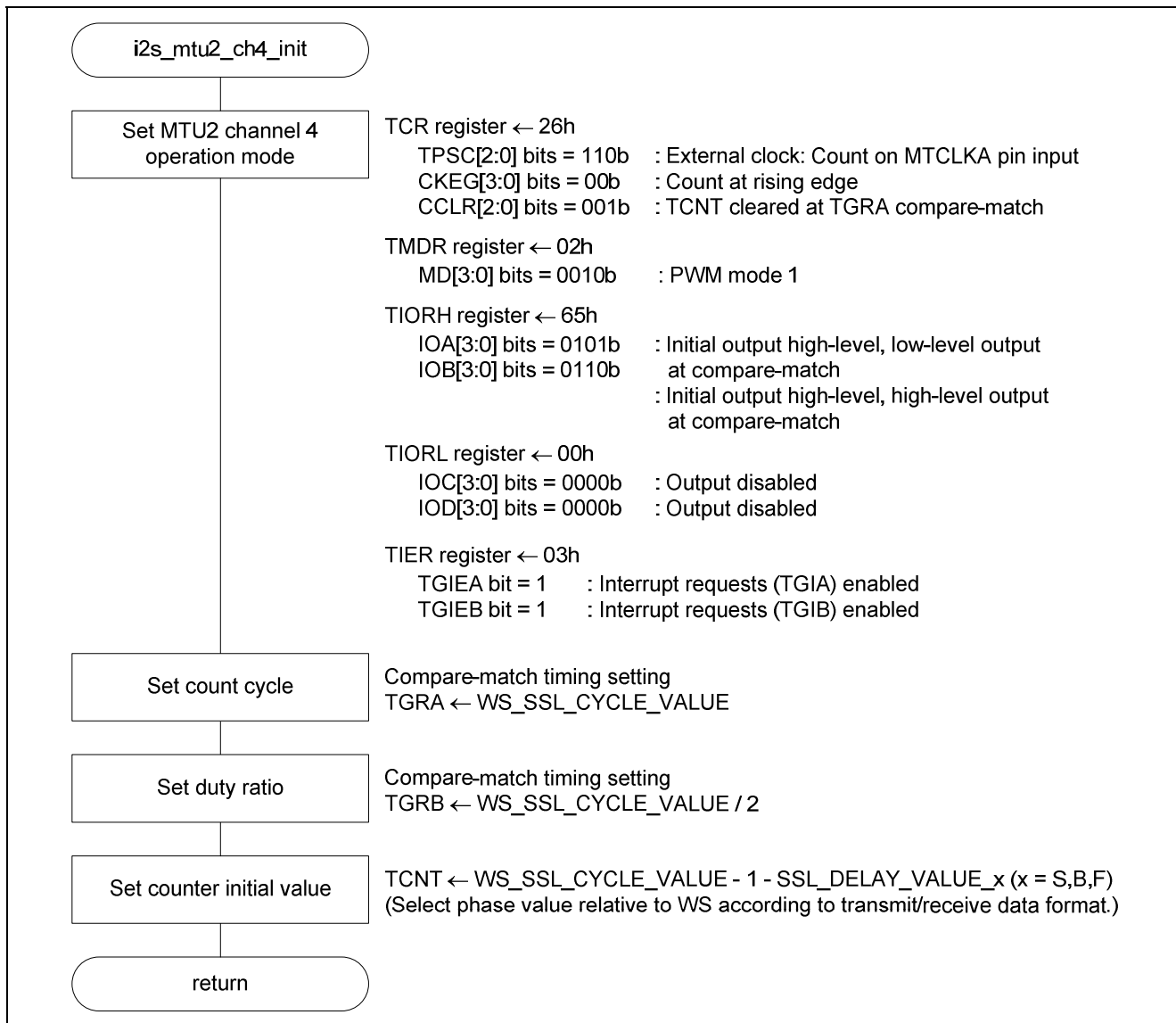


Figure 4.42 i2s\_mtu2\_ch4\_init Function

### 4.8.14 i2s\_rspi\_init Function

Figure 4.43 is a flowchart of the i2s\_rspi\_init function.

The i2s\_rspi\_init function makes the initial settings to the RSPI.

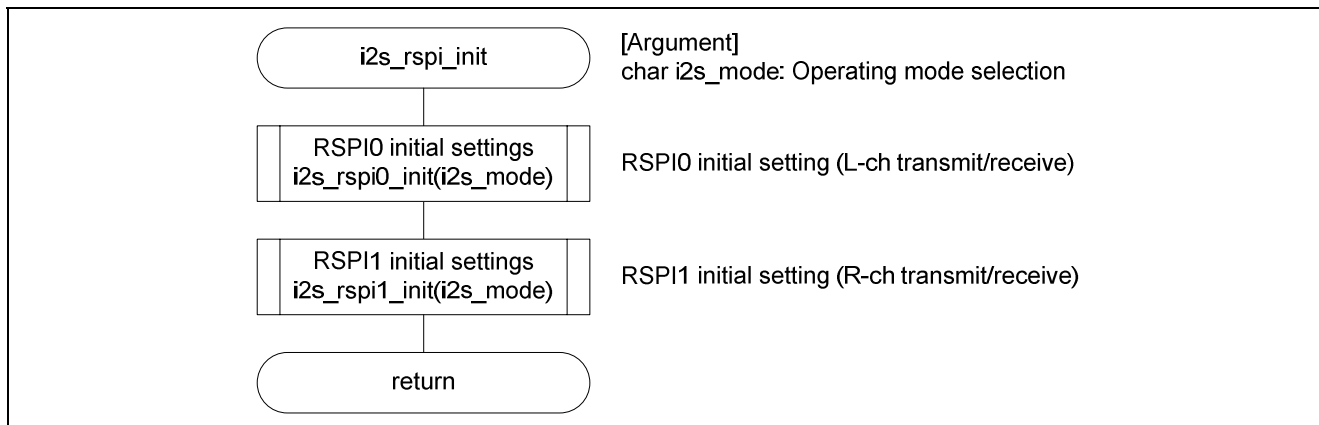


Figure 4.43 i2s\_rspi\_init Function

### 4.8.15 i2s\_rspi0\_init Function

Figure 4.44 is a flowchart of the i2s\_rspi0\_init function.

The i2s\_rspi0\_init function makes the initial settings to the RSPI0.

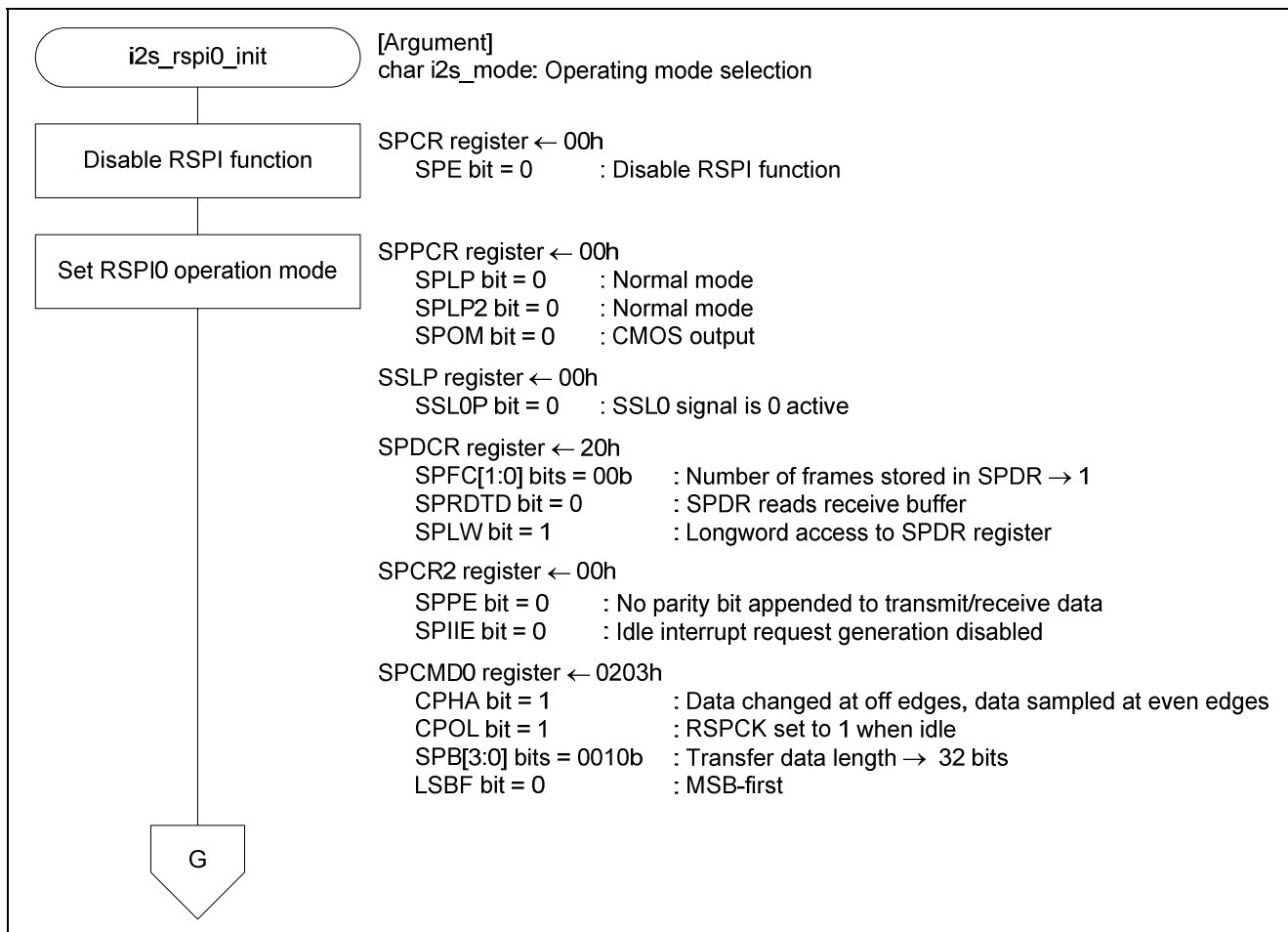


Figure 4.44 i2s\_rspi0\_init Function (1/2)

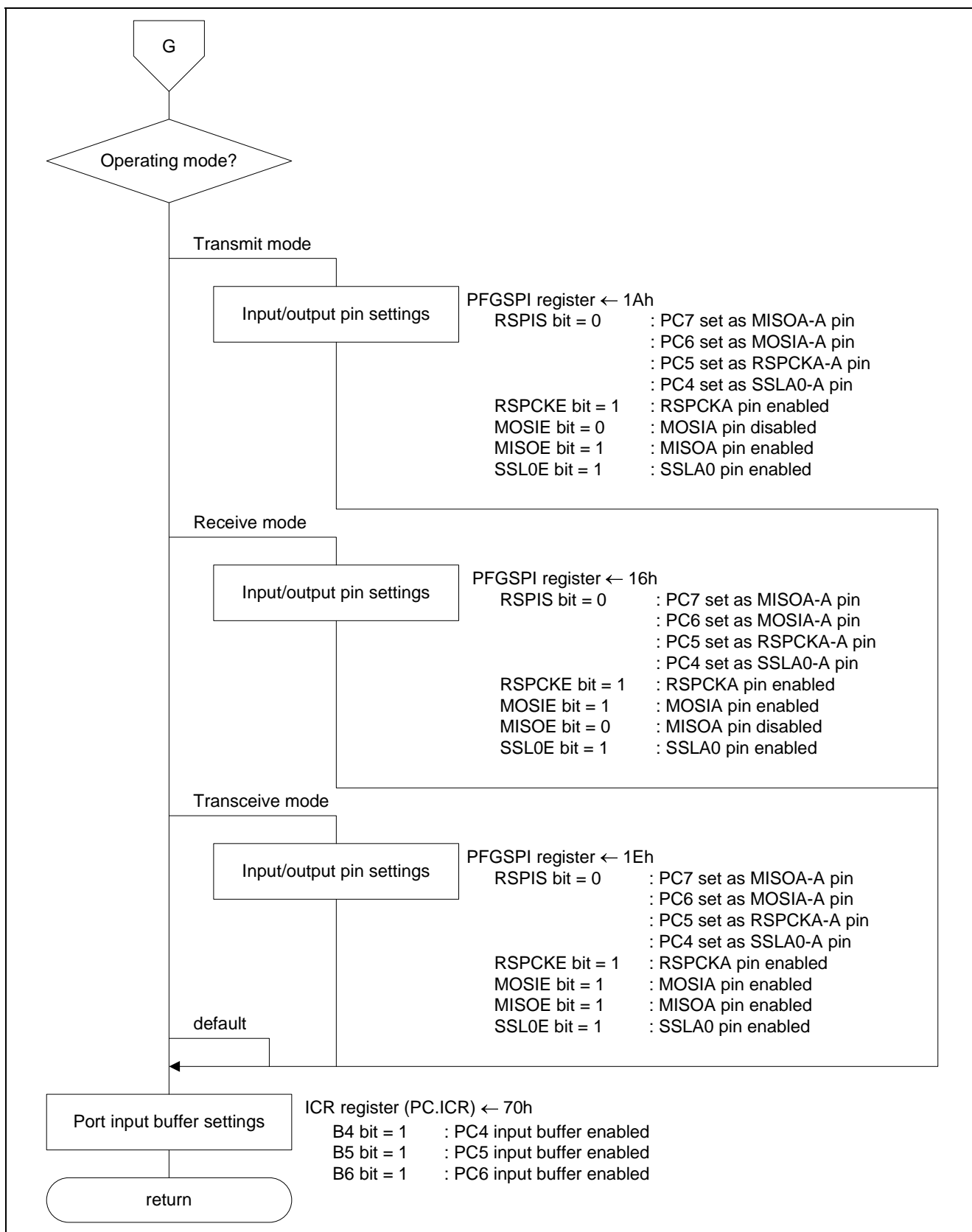


Figure 4.44 i2s\_rspi0\_init Function (2/2)

4.8.16 i2s\_rspl1\_init Function

Figure 4.45 is a flowchart of the i2s\_rspl1\_init function.

The i2s\_rspl1\_init function makes the initial settings to the RSPI1.

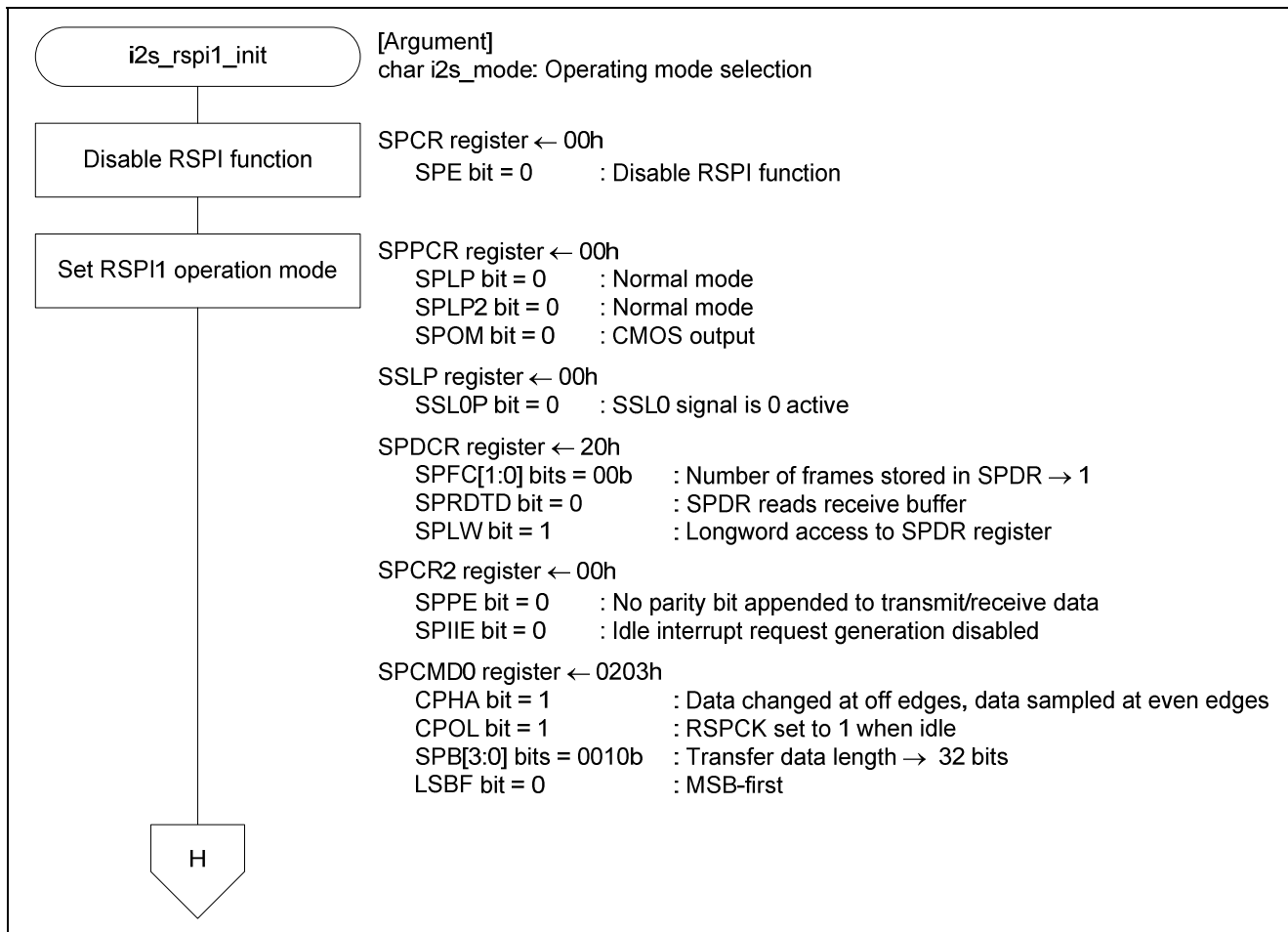


Figure 4.45 i2s\_rspl1\_init Function (1/2)

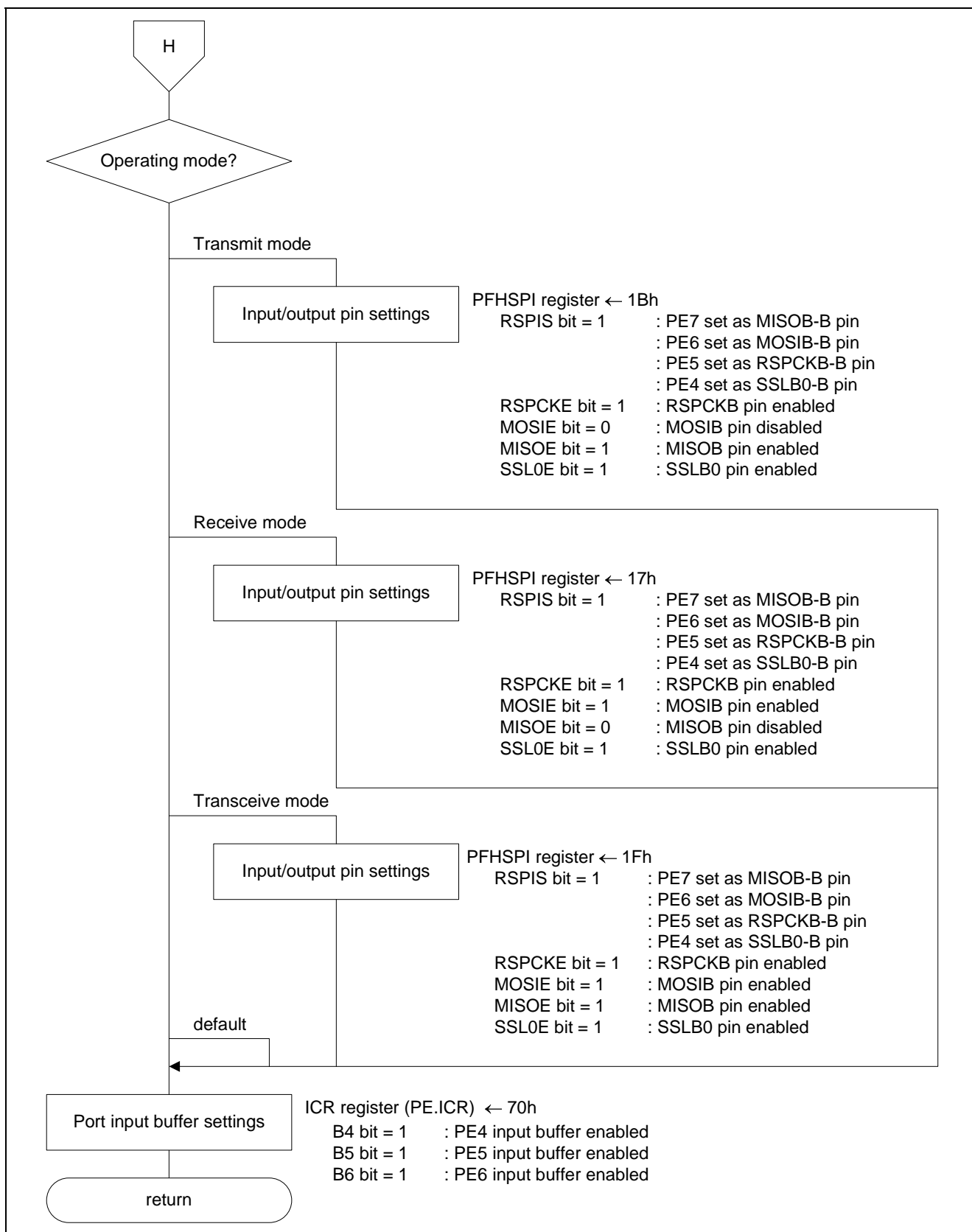


Figure 4.45 i2s\_rspi1\_init Function (2/2)

## 5. Sample Code

The sample code can be downloaded from the Renesas Electronics Corporation web site.

## 6. Reference Documents

- RX62N Group, RX621 Group User's Manual: Hardware, Revision 1.11  
(The latest version can be downloaded from the Renesas Electronics Web site.)
- Technical Updates and Technical Manuals  
(The latest information can be accessed at the Renesas Electronics Web site.)
- RX Family C Compiler Package, Version.1.0.1.0  
(The latest version can be downloaded from the Renesas Electronics Web site.)
- C Compiler User's Manual, Revision 1.00  
(The latest version can be downloaded from the Renesas Electronics Web site.)

## Website and Support

Renesas Electronics Website

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

Inquiries

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

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. 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.



### 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