RX630 Group

I2S Communication Using RSPI, DTCa, and MTU2a

R01AN1338EJ0100
Rev. 1.00
Aug. 1, 2013

Abstract

This application note describes a method for transferring audio data using I2S communication with the serial peripheral interface (RSPI), data transfer controller (DTCa), and multi-function timer pulse unit 2 (MTU2a) in the RX630 Group.

Products

- RX630 Group 177-pin and 176-pin packages with a ROM size between 768 KB and 2 MB
- RX630 Group 145-pin and 144-pin packages with a ROM size between 768 KB and 2 MB
- RX630 Group 100-pin package with a ROM size between 384 KB and 2 MB
- RX630 Group 80-pin package with a ROM size between 384 KB and 512 KB

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

In this application note, the RSPI, DTCa, and MTU2a are used to transmit and receive audio data with I2S communication. The RSPI transfers audio data with serial communication according to the clock signal generated by the MTU2a.

Table 1.1 lists the Peripheral Functions and Their Applications, and Figure 1.1 shows the Block Diagram.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
RSPI channel 0	Input/output audio data for left channel
RSPI channel 1	Input/output audio data for right channel
MTU2a channel 2	Generate a serial transfer clock (SCK)
MTU2a channel 3	Generate a word select signal (WS)
MTU2a channel 4	Generate a slave select signal (SSL) to RSPI
DTCa	Transfer audio data to and from the on-chip RAM

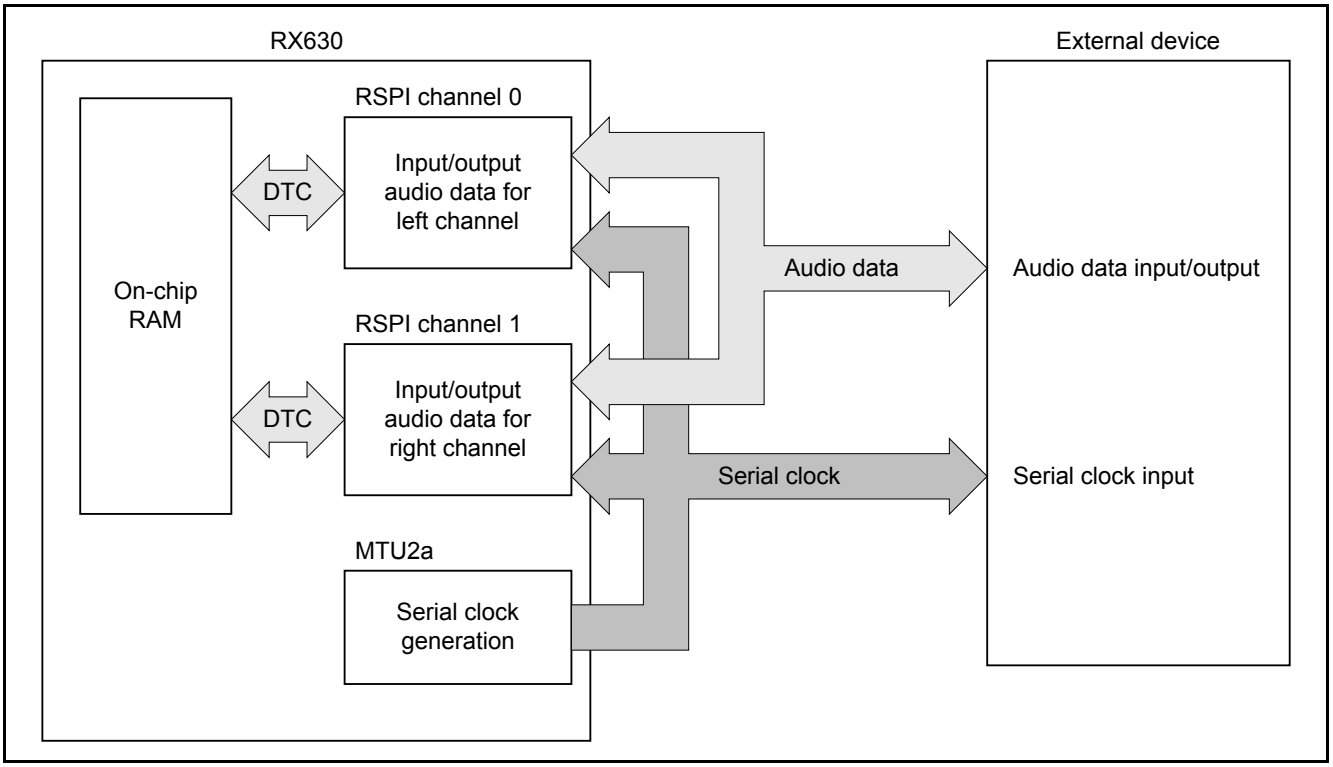

Figure 1.1 Block Diagram

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	R5F5630EDDFP (RX63N Group)
Operating frequencies	- Main clock: 12 MHz - Sub-clock: 32.768 kHz - PLL: 192 MHz (main clock divided by 1 and multiplied by 16) - HOCO: Stopped - System clock (ICLK): 96 MHz (PLL divided by 2) - Peripheral module clock B (PCLKB): 48 MHz (PLL divided by 4) - External bus clock (BCLK): 48 MHz (PLL divided by 4) - FlashIF clock (FCLK): 48 MHz (PLL divided by 4) - IEBUS clock (IECLK): 48 MHz (PLL divided by 4)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics Corporation High-performance Embedded Workshop Version 4.09.01
C compiler	Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.1.02 Release 01
	Compile options -cpu=rx600 -output=obj="$(CONFIGDIR)\$(FILELEAF).obj" -debug -nologo (The default setting is used in the integrated development environment.)
iodefine.h version	Version 1.50
Endian	Little endian
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Sample code version	Version 1.00
Board used	Renesas Starter Kit for RX630 (product part no.: R0K505630S000BE)

3 Reference Application Note

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

- RX630 Group Initial Setting Rev. 1.00 (R01AN1004EJ0100)

The initial setting functions in the reference application note are used in the sample code in this application note. The revision number of the reference application note is the one when this application note was made. However the latest version is always recommended. Visit the Renesas Electronics Corporation website to check and download the latest version.

4 Hardware

4.1 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

Table 4.1 Pins Used and Their Functions

Pin Name	I/O	Function
P1_4/MTCLKA	Input	Input the external clock (12.288 MHz)
PB_5/MTIOC2A	Output	Output the serial transfer clock (SCK)
P1_7/MTIOC3A	Output	Output the word select signal (WS)
PA_0/MTIOC4A	Output	Output the slave select signal (SSL)
PA_4/SSLA0	Input	Input the RSPI0 slave select signal
PA_5/RSPCKA	Input	Input the RSPI0 serial transfer clock
PA_6/MOSIA	Input	Input audio data for left channel
PA_7/MISOA	Output	Output audio data for left channel (1)
PE_4/SSLB0	Input	Input the RSPI1 slave select signal
PE_5/RSPCKB	Input	Input the RSPI1 serial transfer clock
PE_2/MOSIB	Input	Input audio data for right channel
PE_3/MISOB	Output	Output audio data for right channel (1)

Note: When in reception mode, set the pin to high impedance so that data is not output.

4.2 Peripheral Device Connections

Figure 4.1 shows the Connection Diagram. Table 4.2 lists the Clock Signals Generated by MTU2a.
The MTU2a uses the external clock (12.288 MHz) as the count clock and generates the clock signals listed in Table 4.2. RSPI channel 0 (RSPI0) is used to transmit and receive the left channel components (L-ch) of the audio data and RSPI channel 1 (RSPI1) is used for the right channel components (R-ch). The RSPI operates in slave mode and switches the active channel according to the SSL signal.

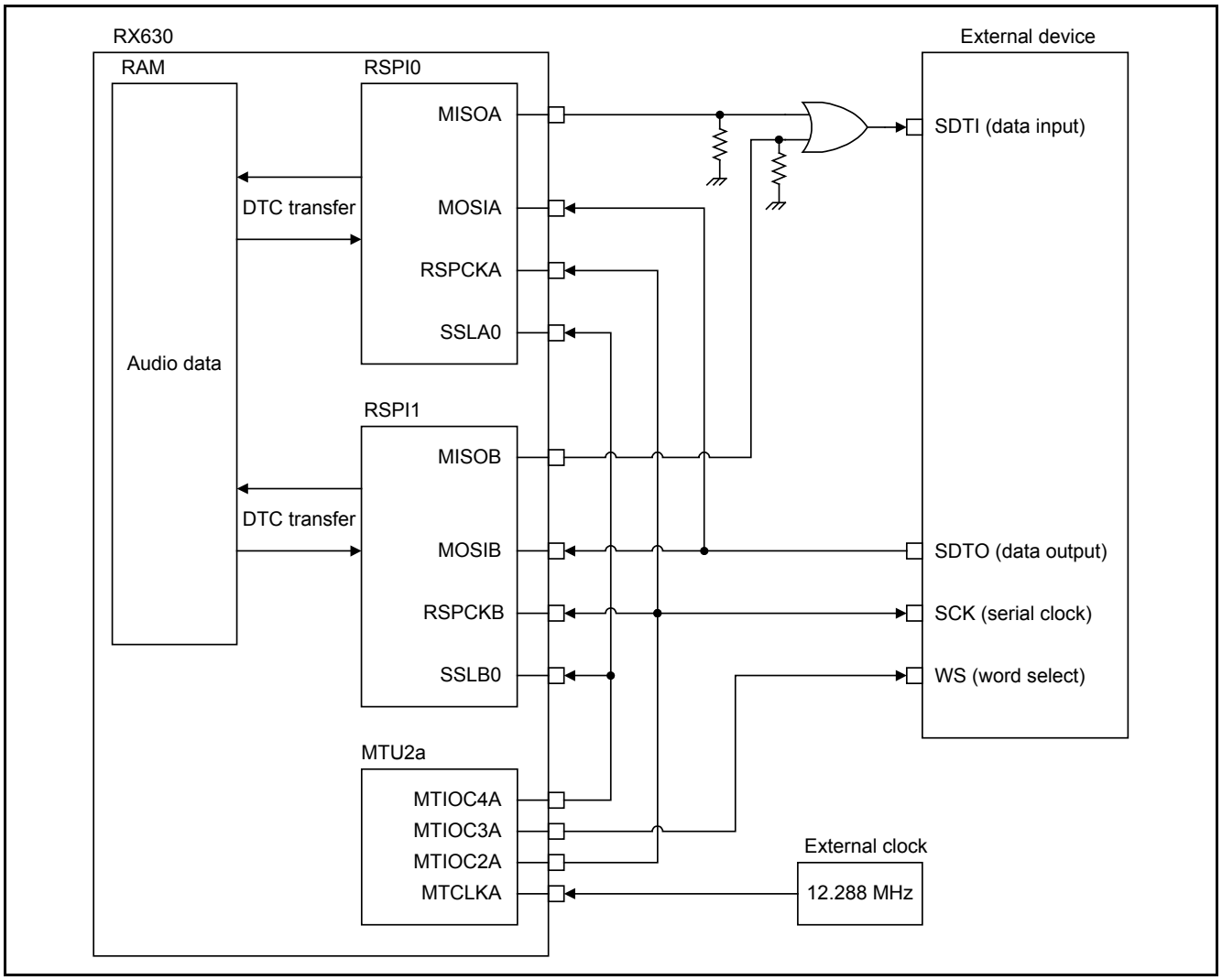

Figure 4.1 Connection Diagram

Table 4.2 Clock Signals Generated by MTU2a

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	48 kHz

5 Software

5.1 Operation Overview

5.1.1 I2S Communication Using RSPI, DTCa, and MTU2a

This section describes the method of I2S communication using RSPI, DTCa, and MTU2a.

5.1.1.1 Audio Data Format

Audio data that consists of 24-bit of data and 8-bit of padding is handled in 32-bit units.

(a) Data Format in the On-Chip RAM

Figure 5.1 shows the Audio Data in the On-Chip RAM.

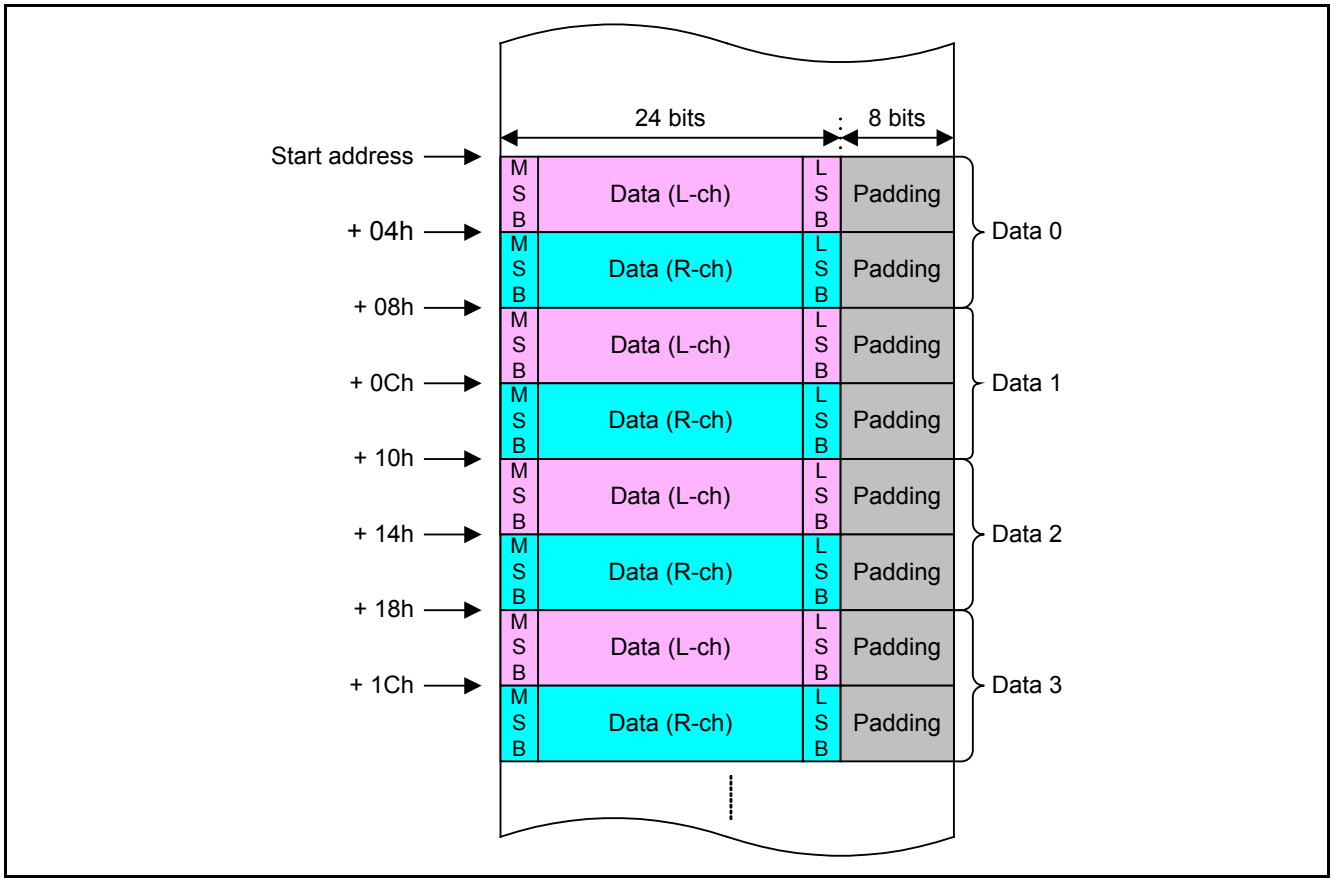

Figure 5.1 Audio Data in the On-Chip RAM

(b) Transmit and Receive Data Format

Serial transfer is performed with 32-bit data length and MSB-first. SCK is used as the serial transfer clock and WS is used as the channel select signal. When WS is low, the L-ch components of the audio data are transferred. When WS is high, the R-ch components are transferred. The position where padding is added to the transmit and receive data is selected according to the audio interface format of the external device.

The data format can be selectable from the following three formats in this application note.

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

Figure 5.2 shows the Transmit and Receive Data Format.

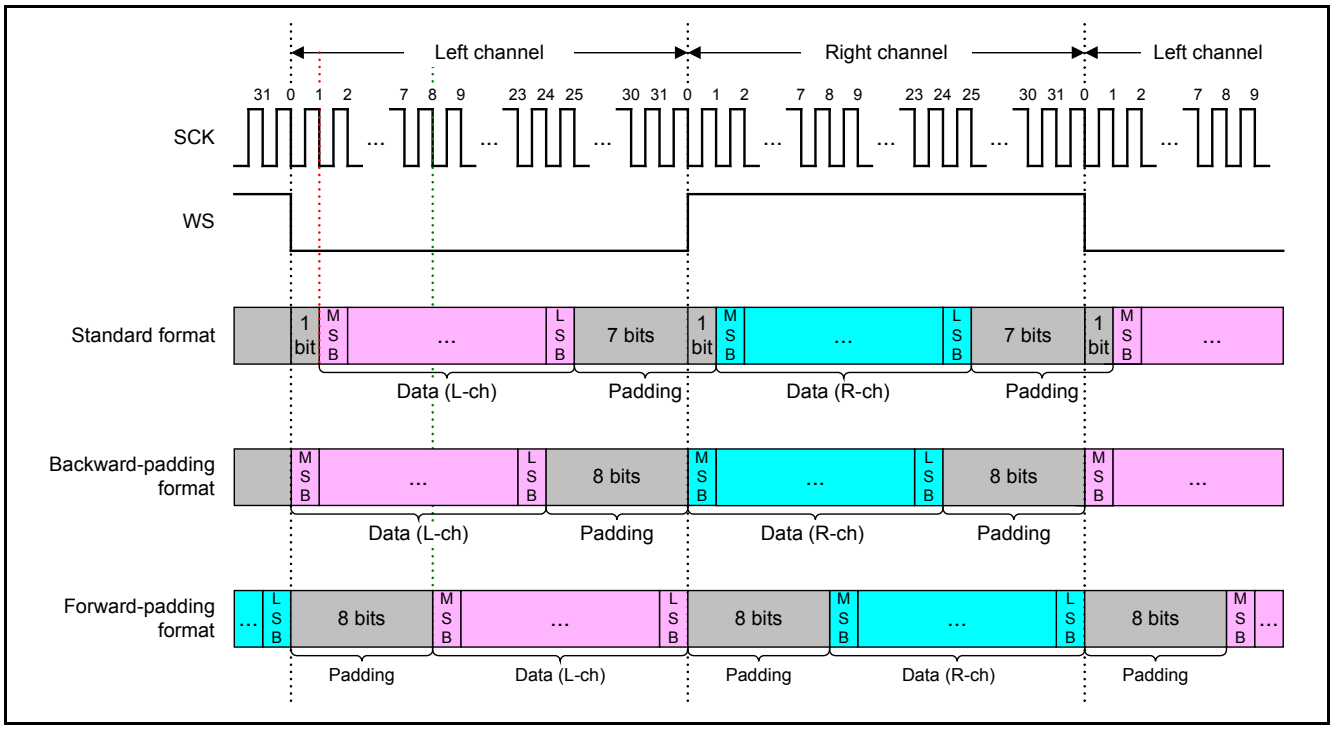

Figure 5.2 Transmit and Receive Data Format

5.1.1.2 Transmitting and Receiving Audio Data

I2S transmission is performed by transferring audio data placed in the on-chip RAM to the RSPI using the DTC, and outputting the audio data from the RSPI. I2S reception is performed by transferring audio data received from the RSPI to the on-chip RAM using the DTC.
The RSPI operates in slave mode (SPI operation) with 32-bit data length and MSB-first.

Figure 5.3 shows the Flow of Audio Data.

The audio data is separated into L-ch and R-ch components. RSPI0 transmits and receives the L-ch components, and RSPI1 transmits and receives the R-ch components. The slave-select polarity of RSPI0 is set to active low and that of RSPI1 is set to active high. RSPI0 or RSPI1 channel is selected according to the SSL output from MTU2a channel 4.

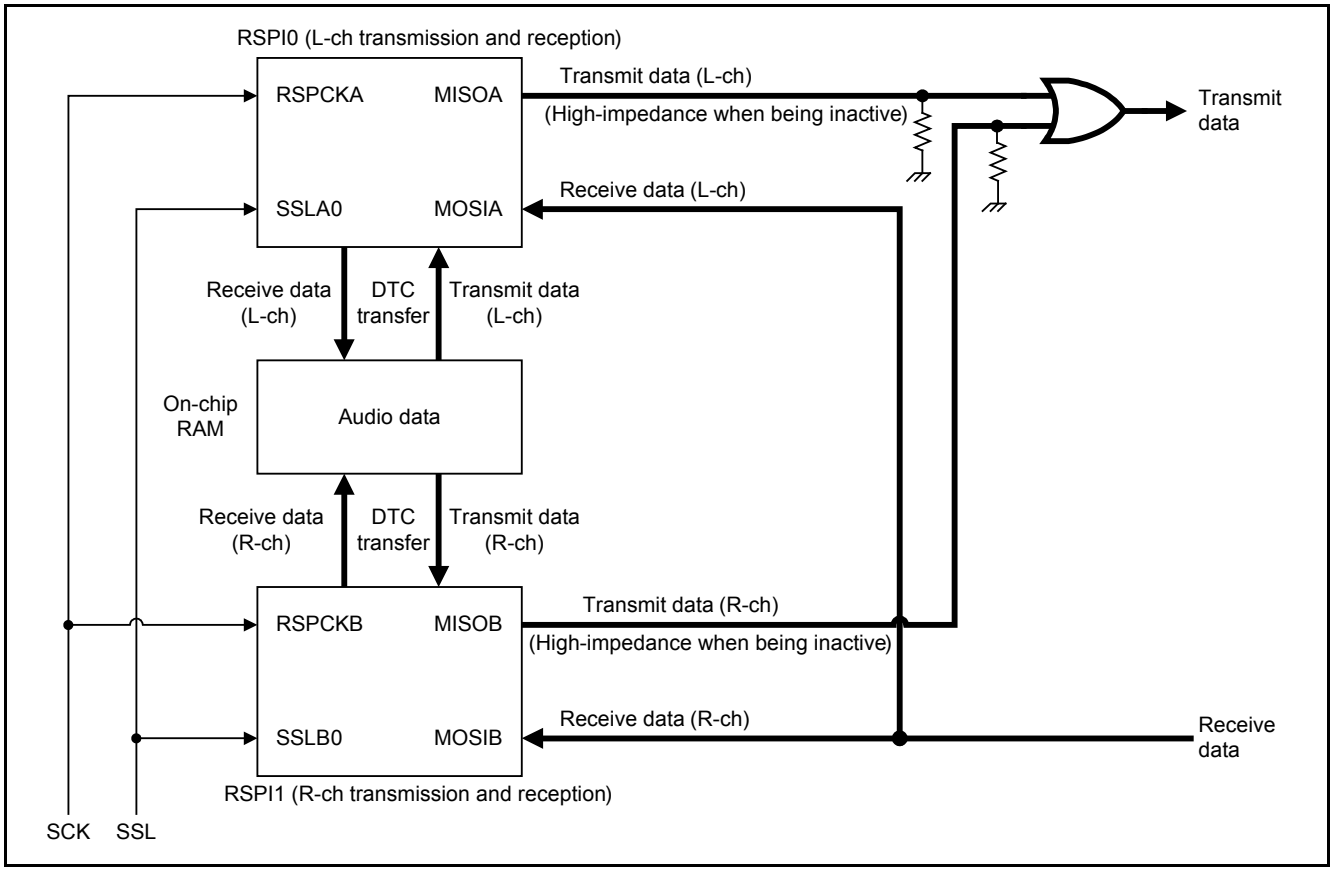

Figure 5.3 Flow of Audio Data

(a) Transmit Operation

Figure 5.4 shows the RSPI Audio Data Transmit Timing.

Audio data for transmitting to the external device is generated by combining the transmit data (L-ch) output from RSPI0 and the transmit data (R-ch) output from RSPI1 with an external OR circuit. The RSPI output channel that is not outputting transmit data is placed in high-impedance, thus those channels need to be pulled down so that they do not affect the data combined with the OR circuit.

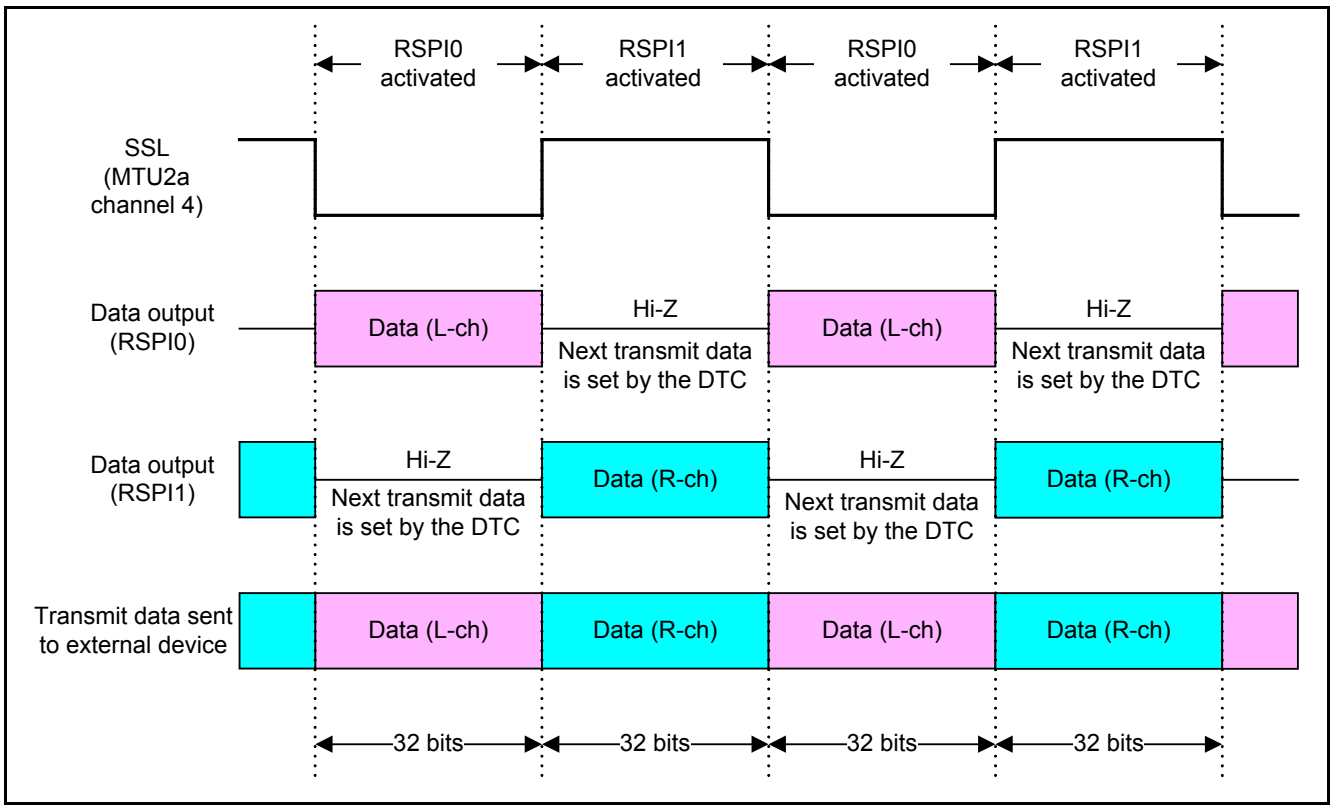

Figure 5.4 RSPI Audio Data Transmit Timing

(b) Receive Operation

Figure 5.5 shows the RSPI Audio Data Receive Timing.

Audio data for receiving from the external device is input to both channels RSPI0 and RSPI1. The RSPI channel to be activated is selected according to the SSL signal. The input audio data is separated into L-ch and R-ch components for reception.

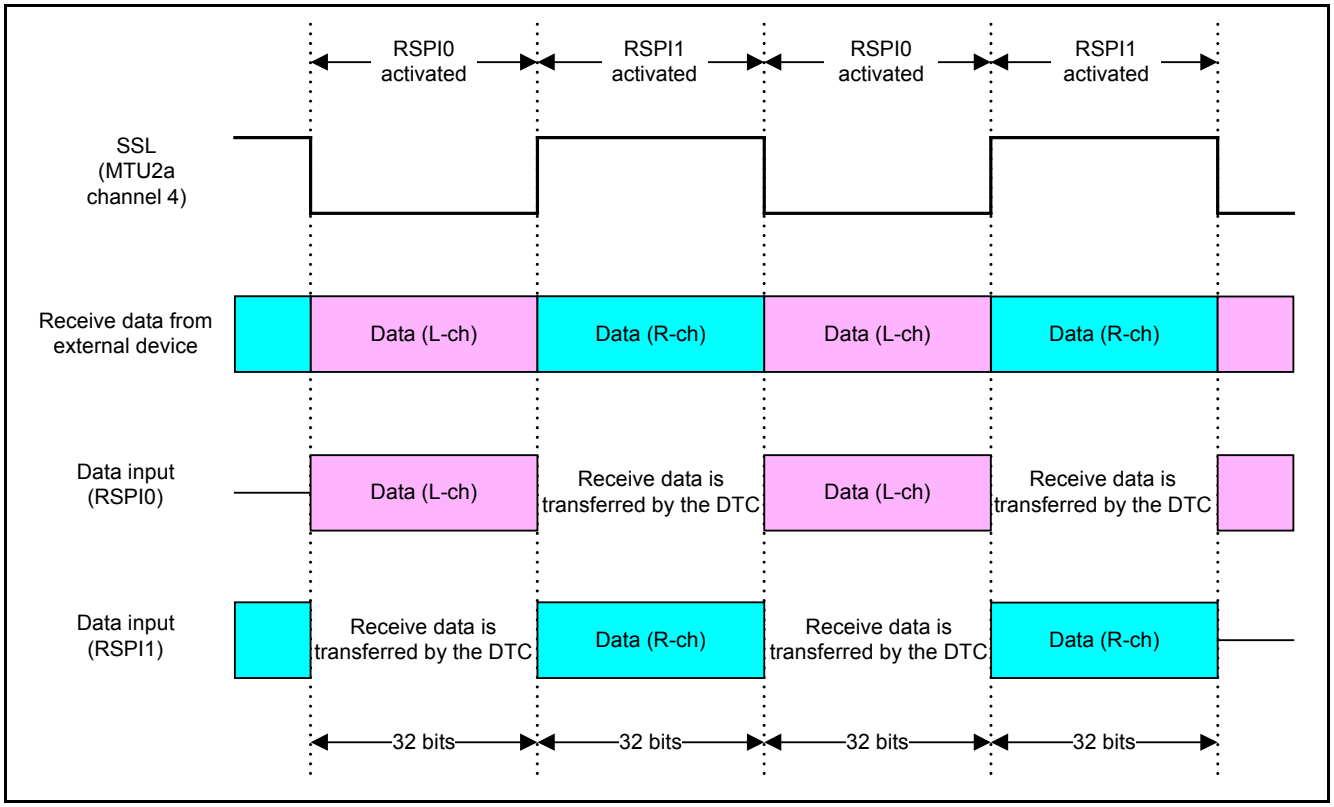

Figure 5.5 RSPI Audio Data Receive Timing

5.1.1.3 I2S Communication Synchronization Recovery and RSPI Initialization

To perform synchronization recovery in the I2S communication when a bit slip occurs due to noise insertion, the RSPI is initialized by the DTC each time L-ch or R-ch data is transferred.

Figure 5.6 shows the RSPI Initialization Timing.

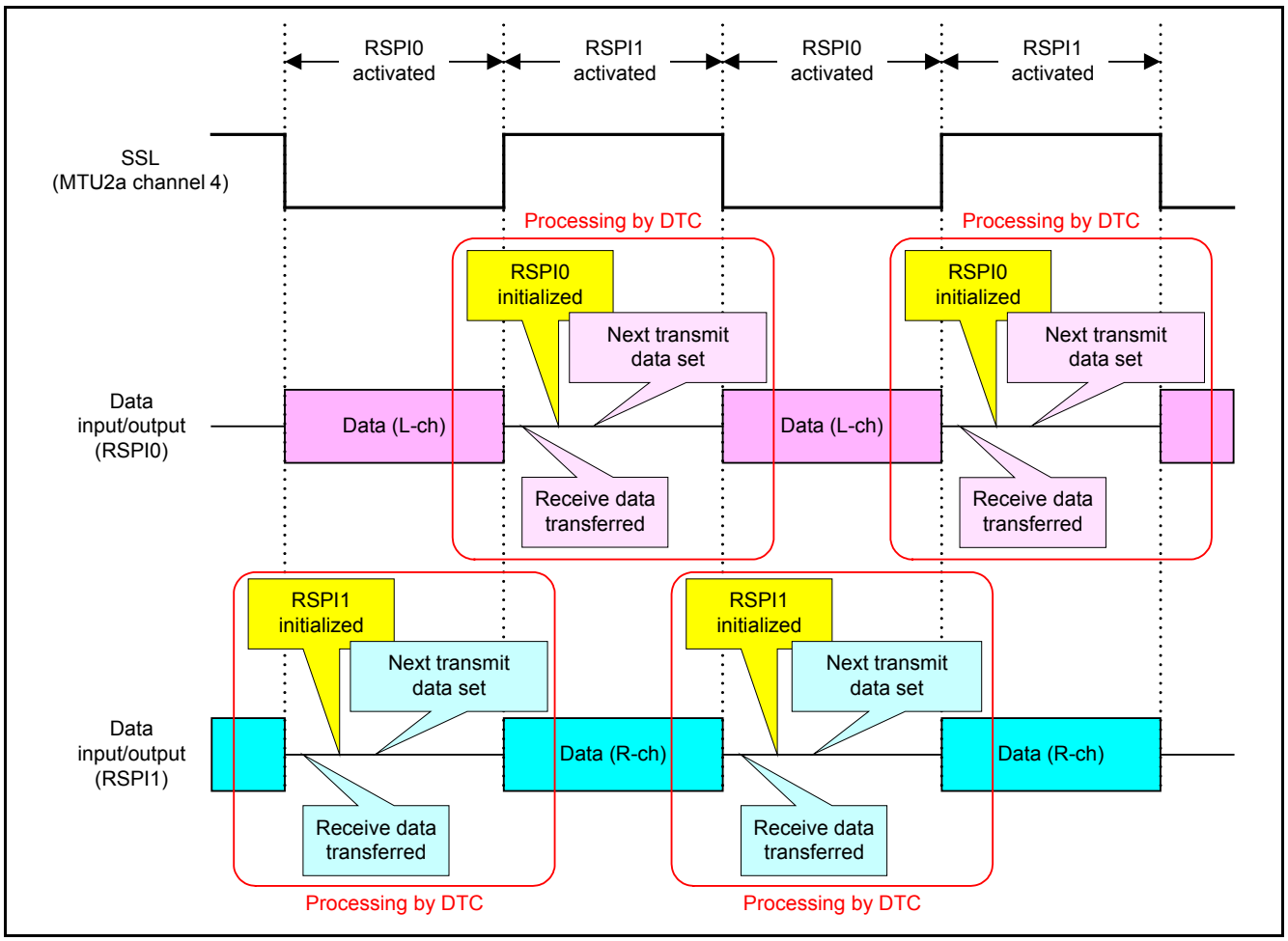

Figure 5.6 RSPI Initialization Timing

5.1.1.4 Generating I2S Communication Clocks and Channel Synchronization Signal

SCK and WS are generated by the MTU2a compare match using an external clock (12.288 MHz) as a count clock, and the RSPI channel select signal (SSL) is also generated by the MTU2a compare match in the same way.

(a) Generating SCK

MTU2a channel 2 is set to PWM mode 1 to output SCK with a frequency of 3.072 MHz (12.288 MHz / 4), duty ratio of 50%, and high initial output.

Figure 5.7 shows the SCK Generation with MTU2a Channel 2.

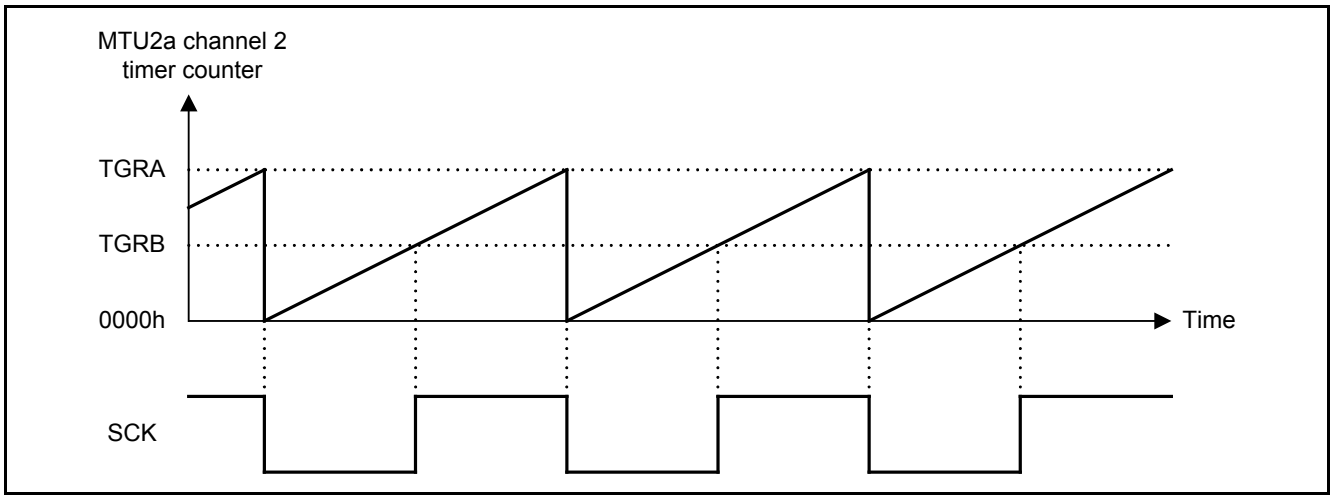

Figure 5.7 SCK Generation with MTU2a Channel 2

(b) Generating WS

MTU2a channel 3 is set to PWM mode 1 to output WS with a frequency of 48 kHz (12.288 MHz / 256), duty ratio of 50%, and high initial output.

Figure 5.8 shows the WS Generation with MTU2a Channel 3.

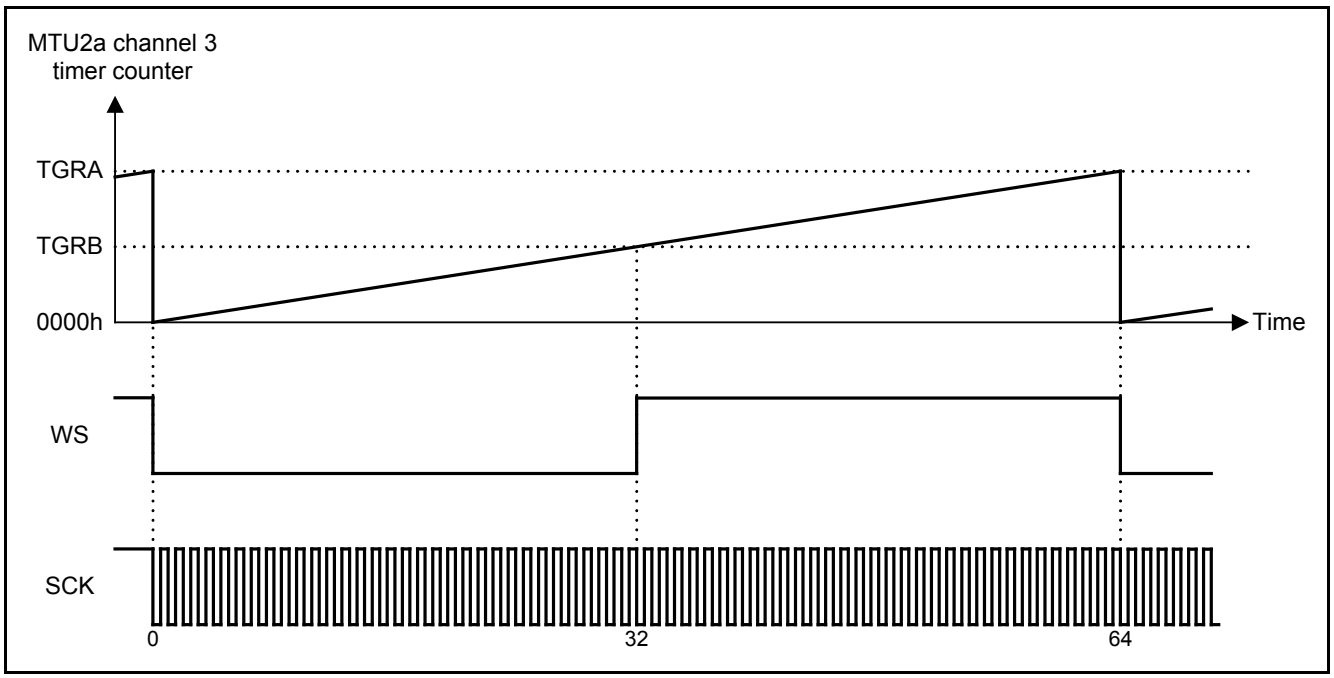

Figure 5.8 WS Generation with MTU2a Channel 3

(c) Generating SSL

MTU2a channel 4 is set to PWM mode 1 to output SSL with a frequency of 48 kHz (12.288 MHz / 256), duty ratio of 50%, and high initial output.
A phase difference is generated between WS and SSL by setting different initial values for the MTU2a channel 4 timer counter and the MTU2a channel 3 timer counter which generates WS. The phase difference between WS and SSL achieves the transmit and receive data formats described in (b) Transmit and Receive Data Format in 5.1.1.1. Refer to 5.1.1.5 Phase Difference between the Word Select Signal and Slave Select Signal for details on the phase difference.

Figure 5.9 shows the SSL Generation with MTU2a Channel 4.

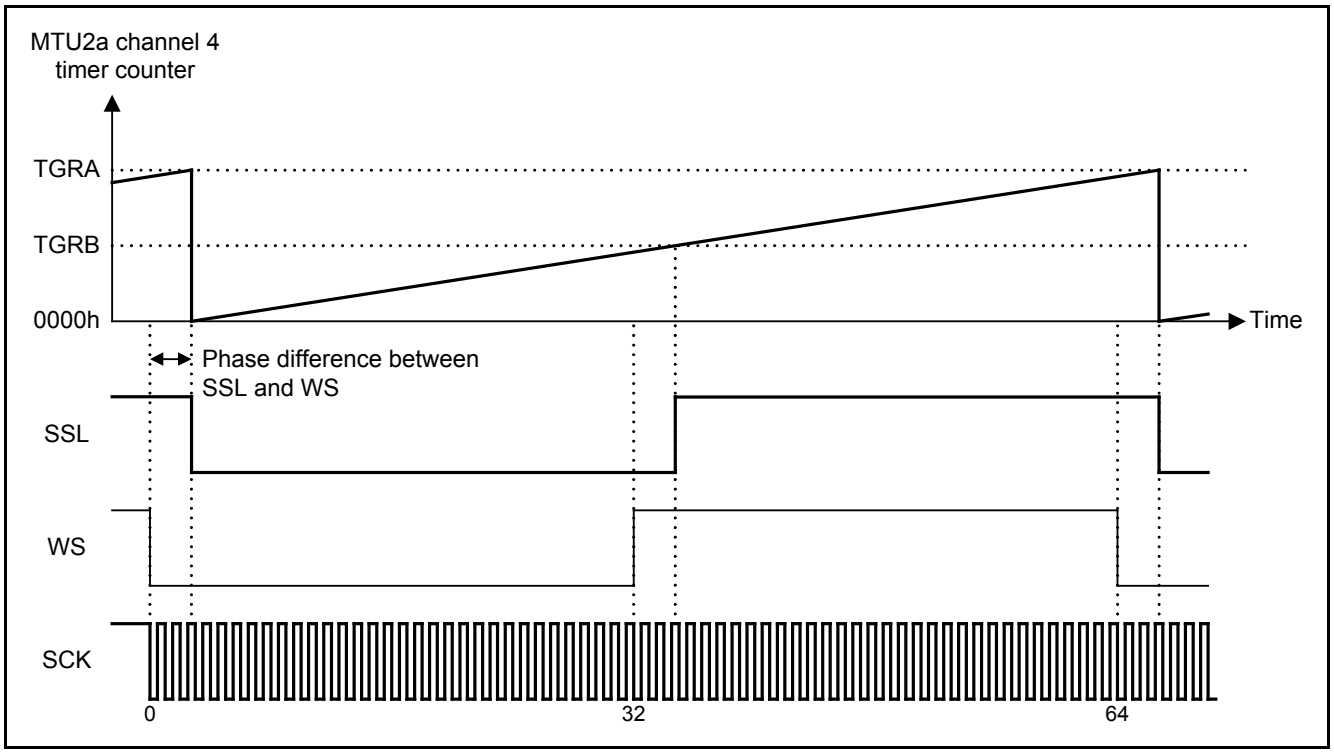

Figure 5.9 SSL Generation with MTU2a Channel 4

5.1.1.5 Phase Difference between the Word Select Signal and Slave Select Signal

Each data format described in (b) Transmit and Receive Data Format in 5.1.1.1 can be achieved by changing the MTU2a channel 4 initial value described in (C) Generating SSL in 5.1.1.4.

Figure 5.10 shows the Phase Difference between WS and SSL in Each Transmit and Receive Data Format.

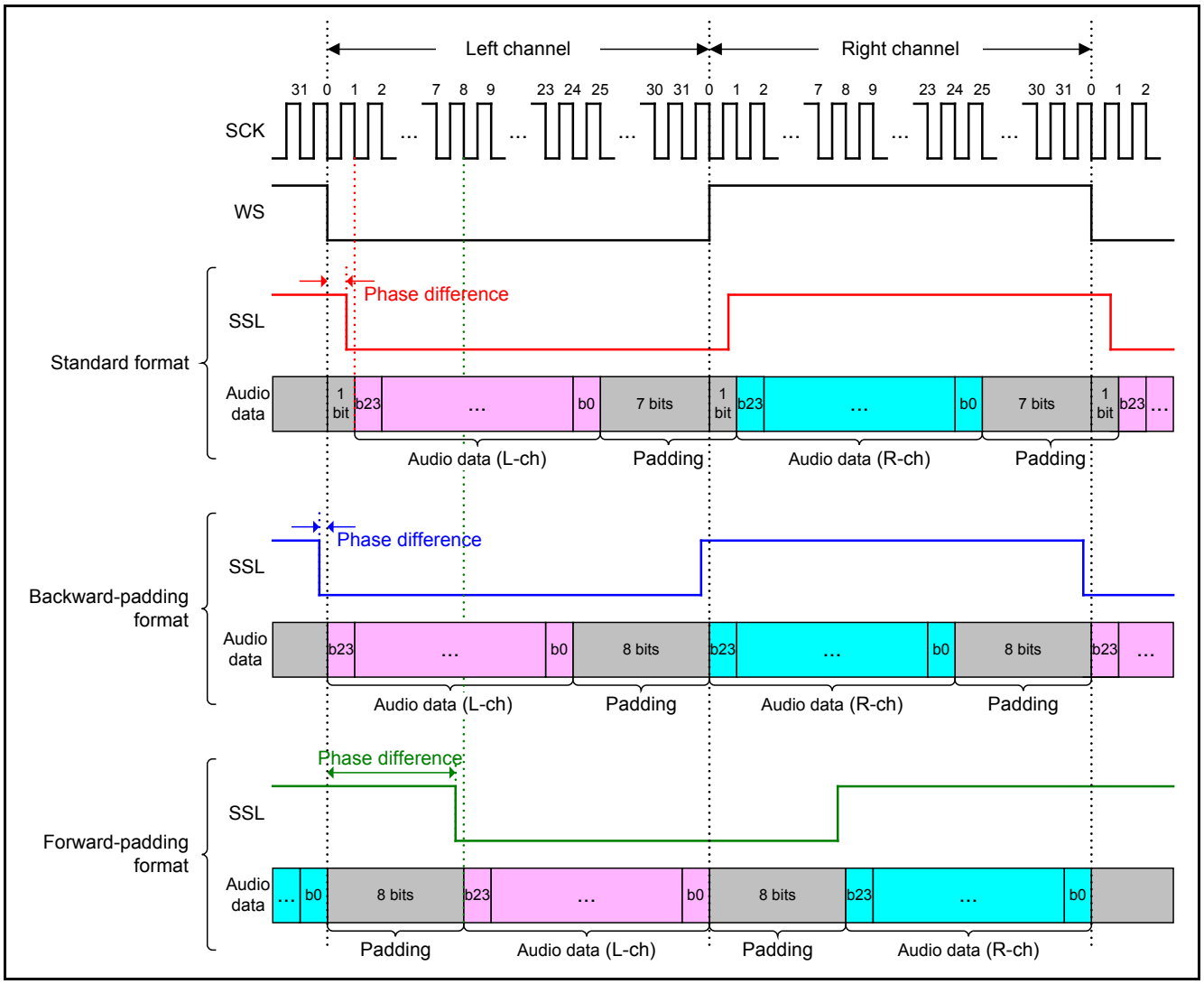

Figure 5.10 Phase Difference between WS and SSL in Each Transmit and Receive Data Format

5.1.2 Transmit Operation

5.1.2.1 Timing of Transmit Operation

Figure 5.11 shows the Timing of Transmit Operation for Standard Format.

The RSPI transmits audio data placed in the on-chip RAM. The RSPI operates in slave mode (SPI operation) with 32-bit data length and MSB-first.

The audio data to be transmitted is separated into L-ch and R-ch components. RSPI0 transmits the L-ch components and RSPI1 transmits the R-ch components. The slave-select polarity of RSPI0 is set to active low and that of RSPI1 is set to active high, and RSPI0 or RSPI1 channel is selected according to SSL. Data output on the active RSPI channel starts at the falling edge of the first SCK signal generated after channel switching, and thereafter data is output synchronizing with SCK.

The MTU2a outputs SSL which has approximately 1-bit phase delay of SCK relative to WS. Refer to 5.1.1.5 Phase Difference between the Word Select Signal and Slave Select Signal for details on the phase difference.

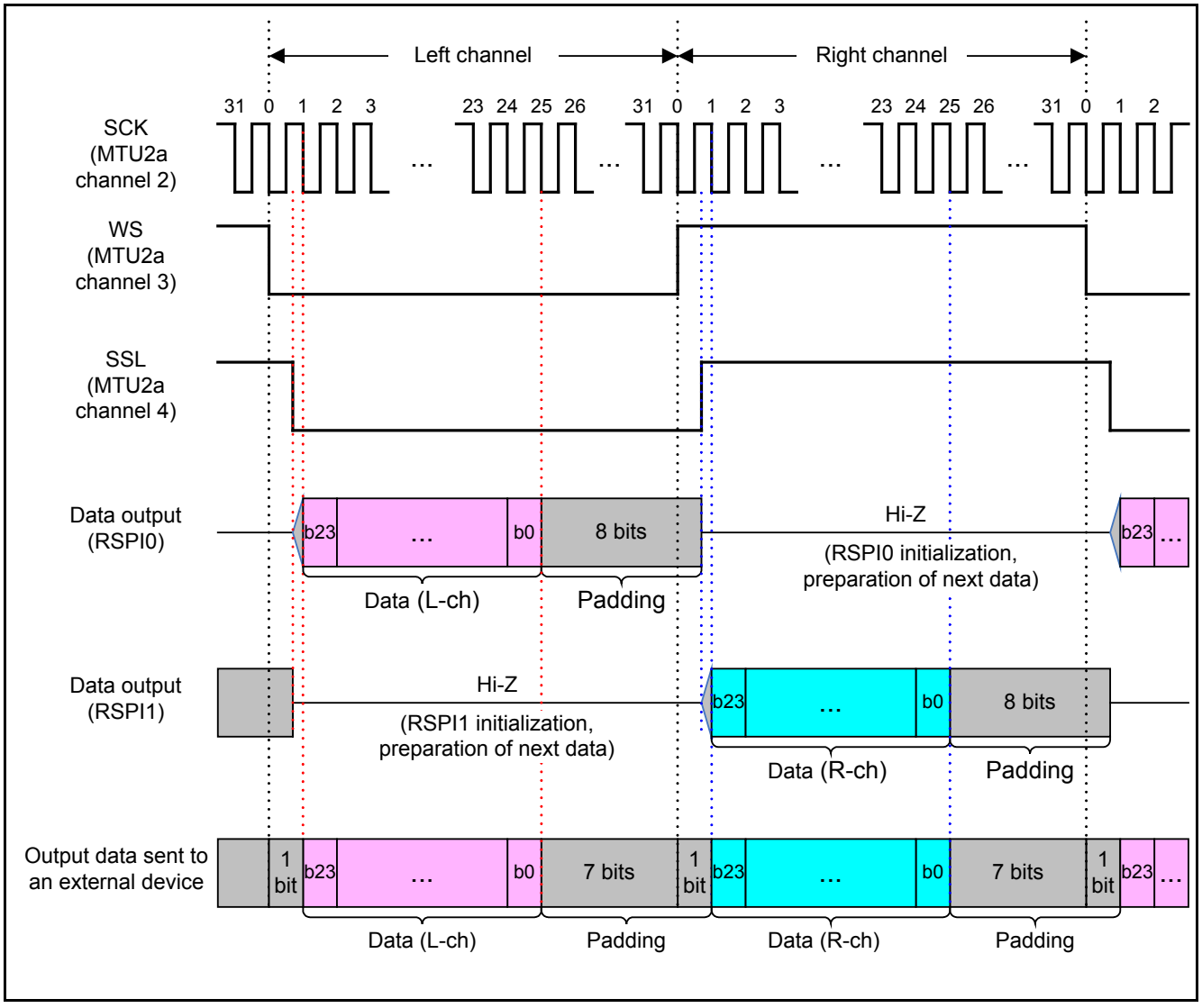

Figure 5.11 Timing of Transmit Operation for Standard Format

5.1.2.2 DTC Operation when Transmitting

Figure 5.12 shows the DTC Operation when Transmitting.

The DTC transfers transmit data from the on-chip RAM to the RSPI. The DTC performs two types of transfer operations; L-ch transmission and R-ch transmission. The DTC activation is triggered by a compare match interrupt generated when an edge occurs on SSL. The DTC is activated for L-ch transmission on a rising edge of SSL, and for R-ch transmission on a falling edge of SSL.

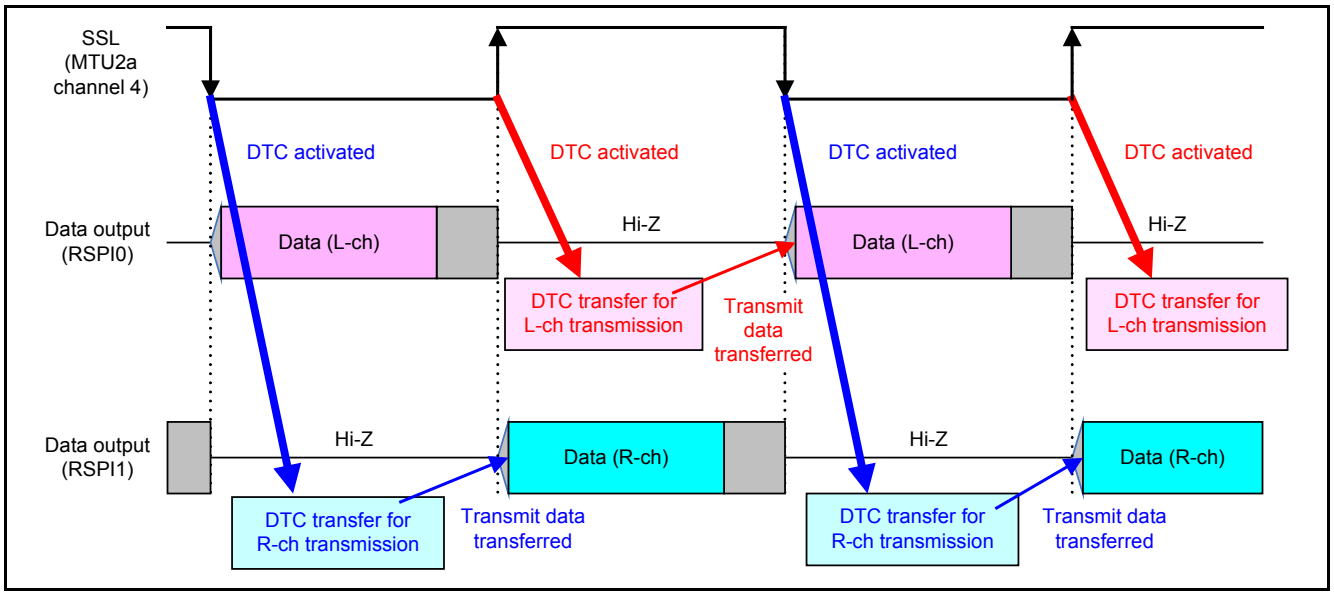

Figure 5.12 DTC Operation when Transmitting

Figure 5.13 shows the DTC Operation when Transmitting L-ch data and Figure 5.14 shows the DTC Operation when Transmitting R-ch data.

The DTC uses chain transfer to initialize the RSPI, transfer the transmit data, and transfer the transmit data address. To initialize the RSPI, disable the RSPI by writing 0 to the RSPI function enable bit (SPE) in the RSPI control register (SPCR), and then enable the RSPI again by writing 1 to the SPE bit.

Note: Refer to the User's Manual: Hardware for details on initialization with the RSPI function enable bit.

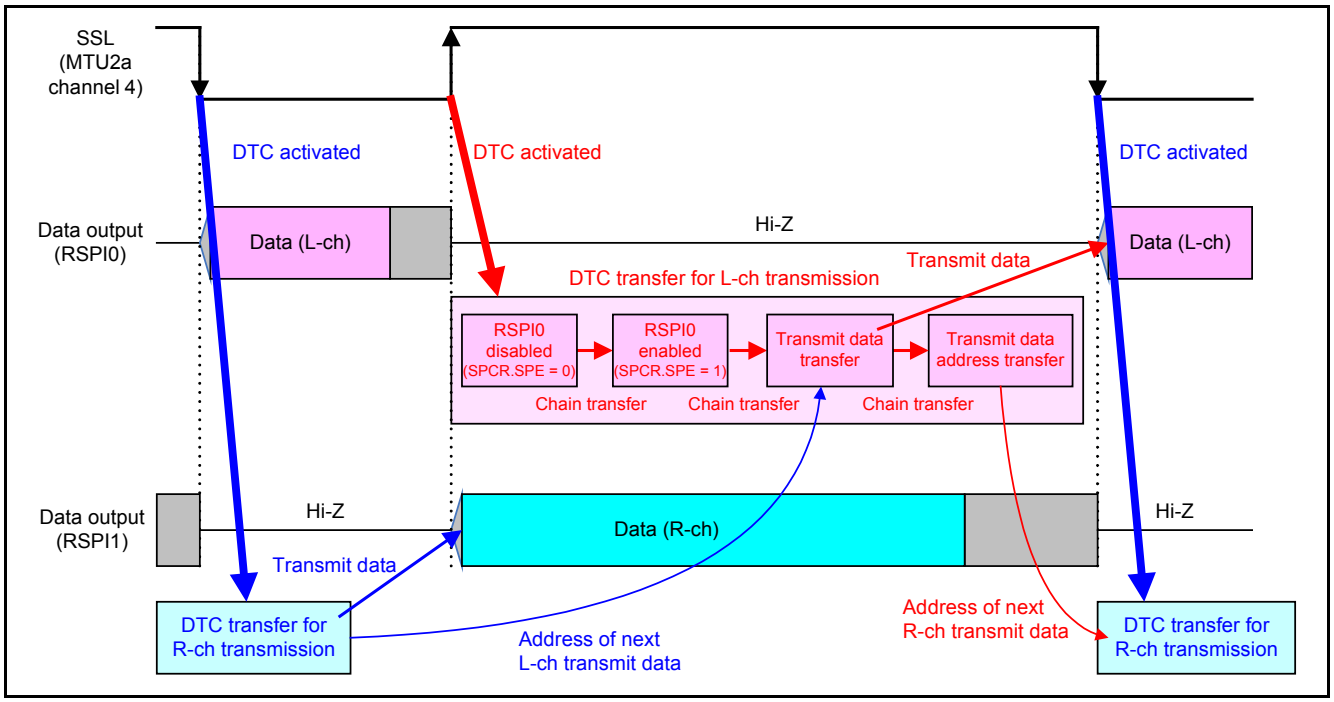

Figure 5.13 DTC Operation when Transmitting L-ch data

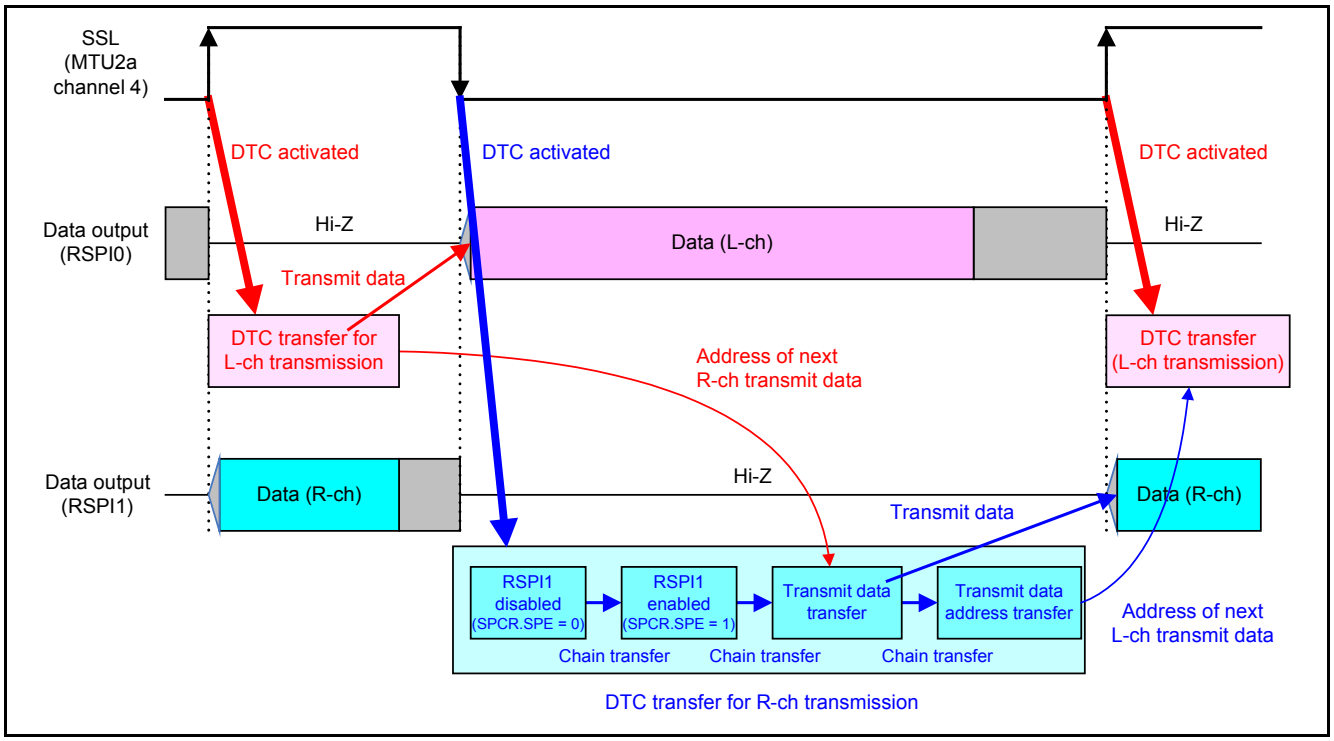

Figure 5.14 DTC Operation when Transmitting R-ch data

5.1.2.3 Transferring the Transmit Data Address by the DTC

Figure 5.15 shows the Transmit Data Address Transfer by the DTC.

L-ch components and R-ch components of the transmit audio data are alternately placed in the on-chip RAM in 4-byte units. Refer to Figure 5.1 Audio Data in the On-Chip RAM. In the sample code, transmit data is transferred by performing L-ch transmission and R-ch transmission alternately using DTC transfer.

As shown in Figure 5.15, the DTC transfers data 0 of the transmit data for L-ch transmission, then the transfer source address after completion of the L-ch transmission is transferred using chain transfer to be used as the transfer source address for data 0 of R-ch transmission. The DTC transfers data 0 of the transmit data for R-ch transmission, then the transfer source address after completion of the R-ch transmission is transferred using chain transfer to be used as the transfer source address for data 1 of L-ch transmission. The DTC transfers data 1 of the transmit data for L-ch transmission, then the transfer source address after completion of the L-ch transmission is transferred using chain transfer to be used as the transfer source address for next R-ch transmission.

In this manner, the DTC transfers the transmit data address as the transfer source address for R-ch transmission when transmitting L-ch data, and as the transfer source address for L-ch transmission when transmitting R-ch data.

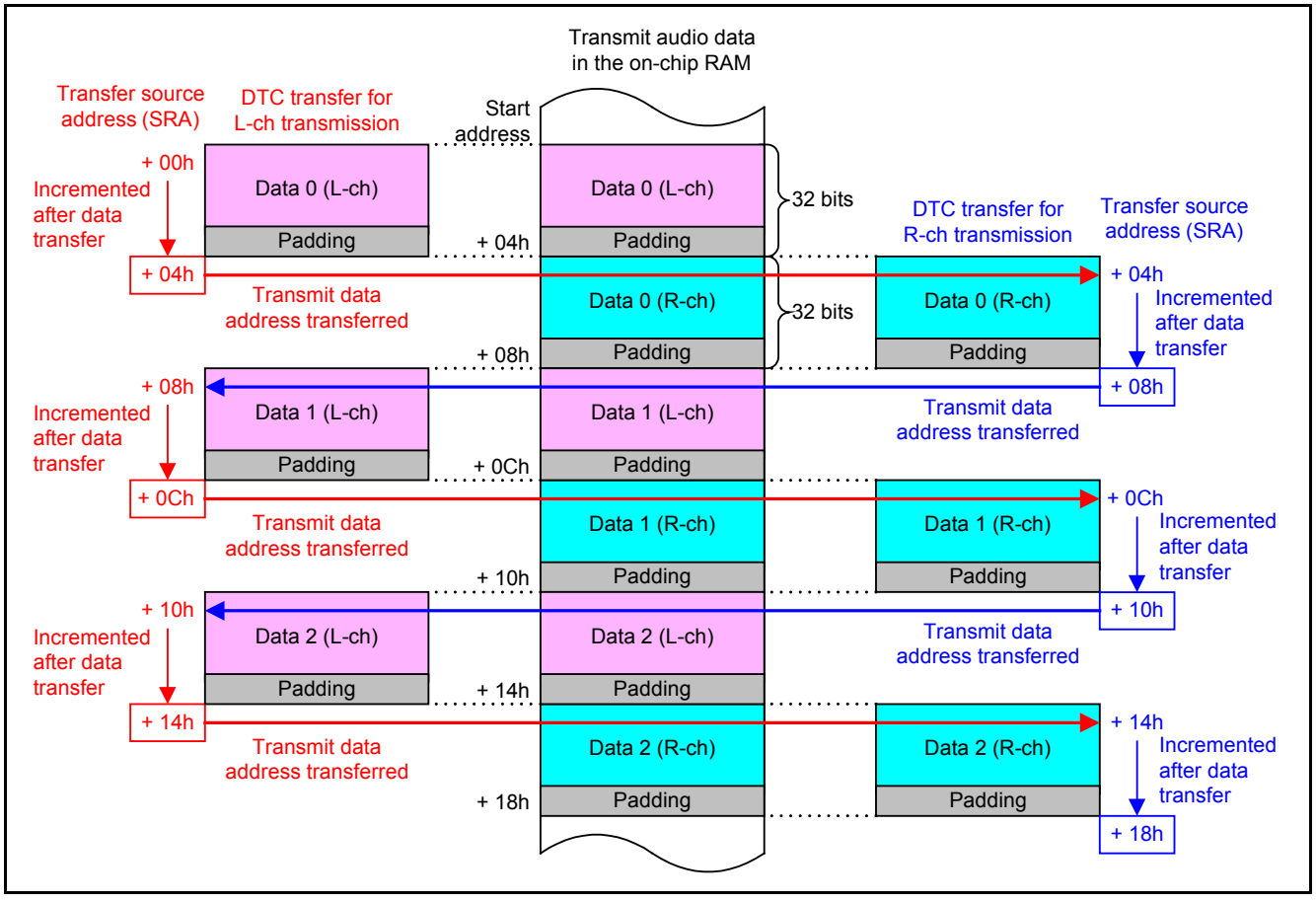

Figure 5.15 Transmit Data Address Transfer by the DTC

5.1.2.4 Completion of Audio Data Transmission

Audio data transmission is completed by disabling the RSPI, thereby halting its operation. Disabling the RSPI is performed by DTC transfer for L-ch and R-ch transmission.

(a) End Processing of L-ch Transmission

Figure 5.16 shows the End Processing of L-ch Transmission.
The DTC performs end processing for L-ch transmission using chain transfer at (n-1)th DTC activation. With end processing, processing at nth DTC activation is only disabling RSPI0, and then the RSPI0 operation stops.

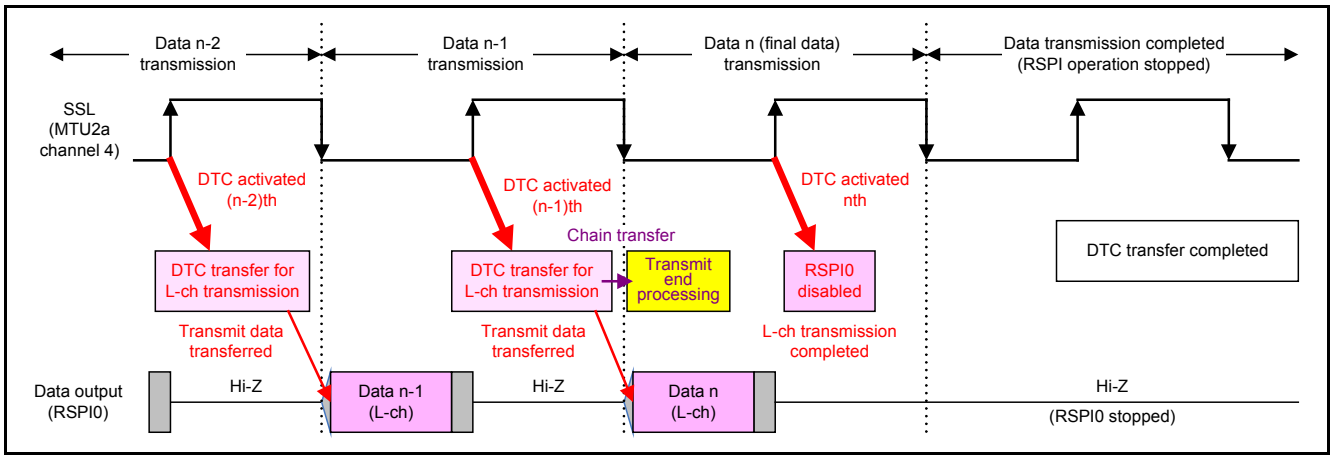

Figure 5.16 End Processing of L-ch Transmission

Figure 5.17 shows the End Processing of L-ch Transmission by DTC Transfer.
As shown in Figure 5.13 DTC Operation when Transmitting L-ch data, with the DTC transfer for L-ch transmission, the chain transfer is used for disabling and enabling RSPI0, transferring the transmit data, and transferring the transmit data address.

By setting the transfer counter, which is included in the transfer information of the transmit data address transfer, to n–1, 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 performed by chain transfer after the transmit data address transfer at (n–1)th DTC activation.

In transmit end processing, by rewriting the DTC chain transfer enable bit in the transfer information for disabling RSPI0, chain transfer is disabled after RSPI0 is disabled at nth DTC activation. Therefore only RSPI0 is disabled at nth DTC activation, then the RSPI0 operation is stopped and the transmit operation is completed.

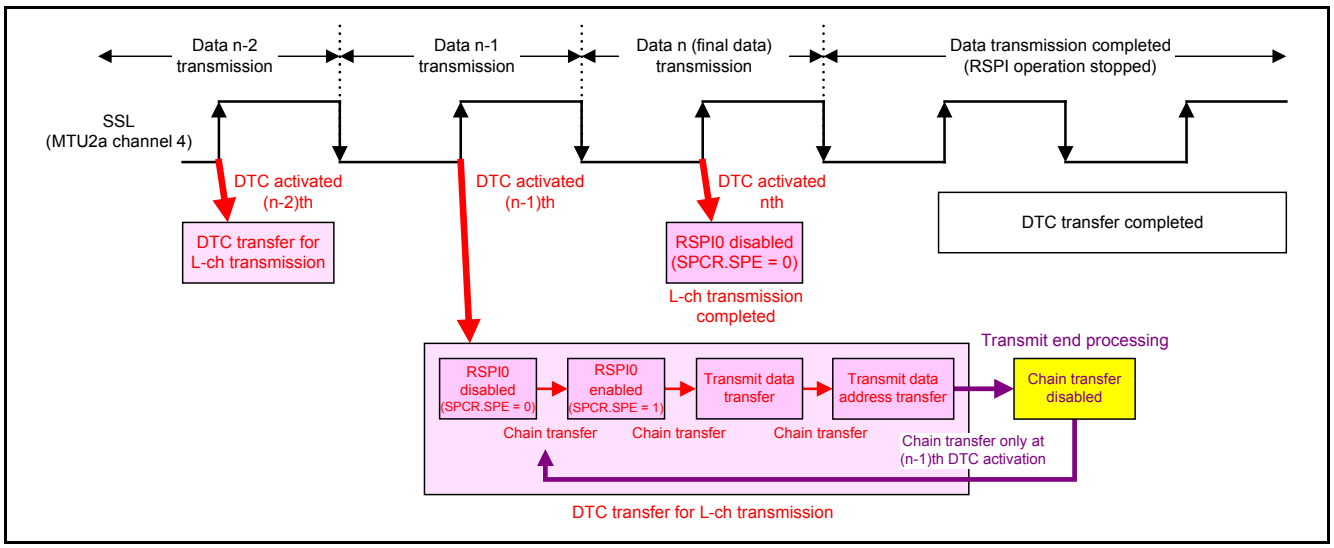

Figure 5.17 End Processing of L-ch Transmission by DTC Transfer

(b) End Processing of R-ch Transmission

Figure 5.18 shows the End Processing of R-ch Transmission.
The DTC performs end processing for R-ch transmission using chain transfer at its nth activation. With end processing, processing at (n+1)th DTC activation is only disabling RSPI1, and then the RSPI1 operation stops.

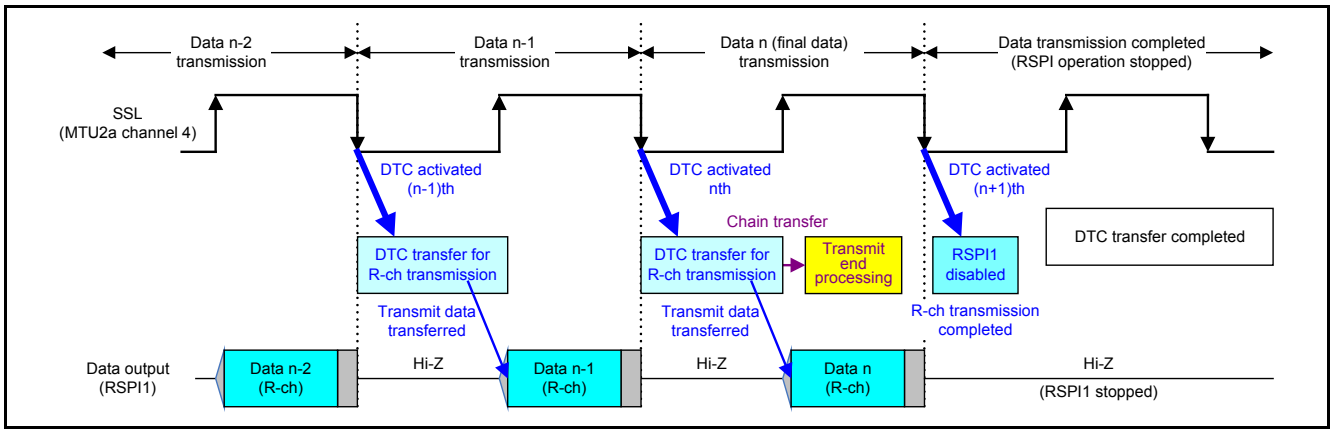

Figure 5.18 End Processing of R-ch Transmission

Figure 5.19 shows the End Processing of R-ch Transmission by DTC Transfer.
As shown in Figure 5.14 DTC Operation when Transmitting R-ch data, with the DTC transfer for R-ch transmission, the chain transfer is used for disabling and enabling RSPI1, transferring the transmit data, and transferring the transmit data address.

By setting the transfer counter, which is included in the transfer information of the transmit data address transfer, to n, 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 performed by chain transfer after the transmit data address transfer at nth DTC activation.

In transmit end processing, by rewriting the DTC chain transfer enable bit in the transfer information for disabling RSPI1, chain transfer is disabled after RSPI1 is disabled at (n+1)th DTC activation. Therefore only RSPI1 is disabled at (n+1)th DTC activation, then the RSPI1 operation is stopped and the transmit operation is completed.

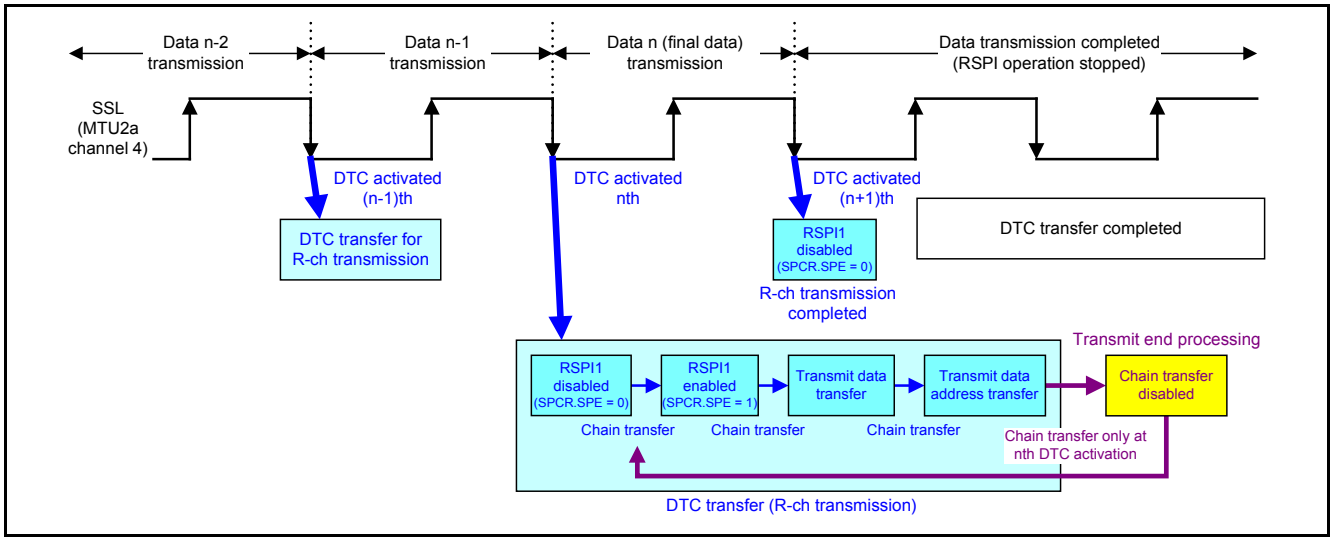

Figure 5.19 End Processing of R-ch Transmission by DTC Transfer

5.1.3 Receive Operation

5.1.3.1 Timing of Receive Operation

Figure 5.20 shows the Timing of Receive Operation for Standard Format.

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

The audio data from an external device is input to both channels RSPI0 and RSPI1, and is separated into L-ch and R-ch components to be received. The slave-select polarity of RSPI0 is set to active low and that of RSPI1 is set to active high. RSPI0 or RSPI1 channel is selected according to SSL. Data is input through the active RSPI channel at the rising edge of the first SCK signal generated after channel switching, and thereafter data is input synchronizing with SCK.

The data input by the RSPI is transferred to the on-chip RAM by the DTC and stored as shown in Figure 5.1. The MTU2a outputs SSL which has approximately 1-bit phase delay of SCK relative to WS. Refer to 5.1.1.5 Phase Difference between the Word Select Signal and Slave Select Signal for details on the phase difference.

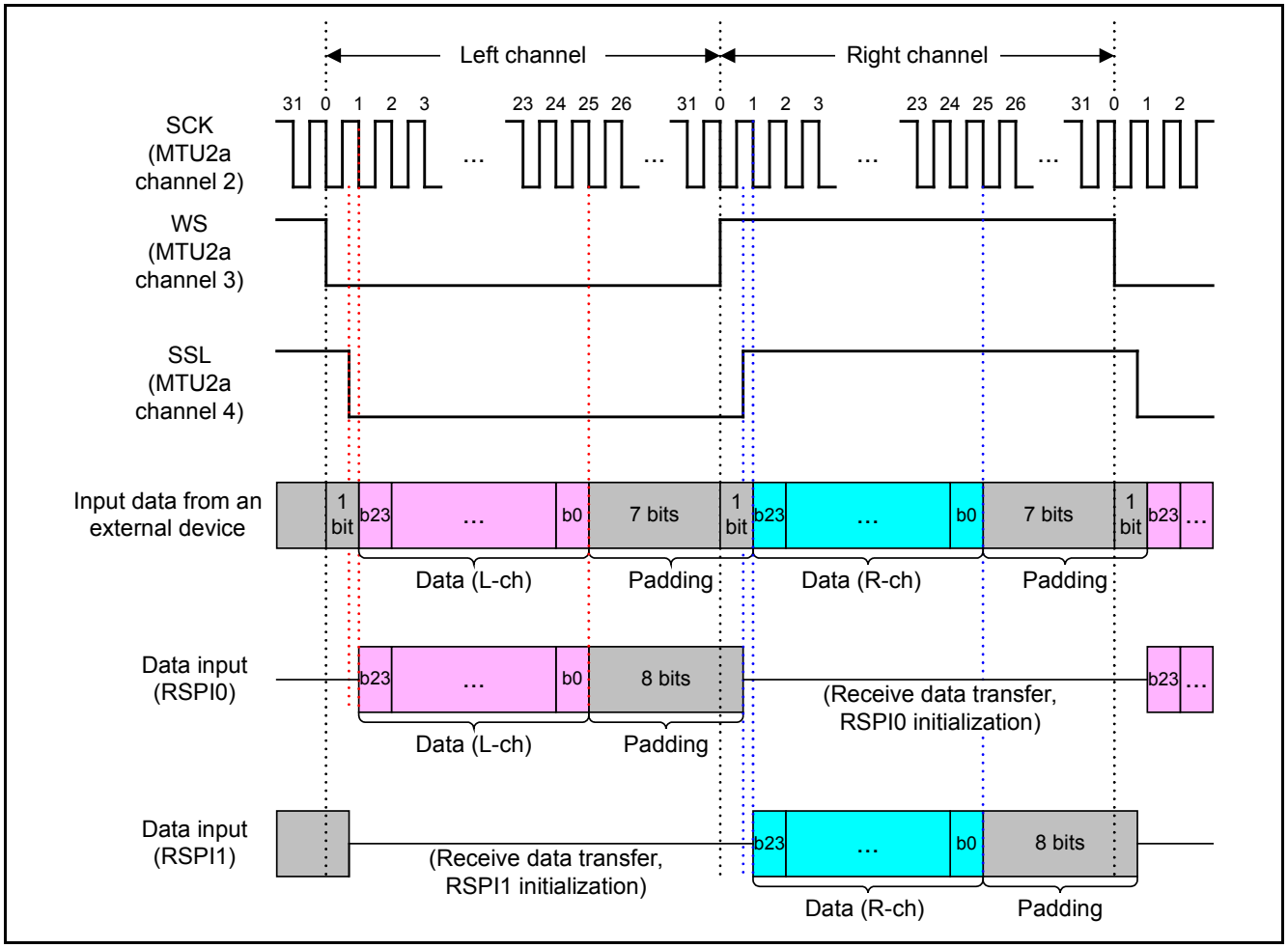

Figure 5.20 Timing of Receive Operation for Standard Format

5.1.3.2 DTC Operation when Receiving

The DTC transfers receive data from the RSPI to the on-chip RAM. L-ch reception, R-ch reception, RSPI0 initialization, and RSPI1 initialization are performed using DTC transfer. The DTC is activated for data reception by the RSPI receive buffer full interrupt. The DTC is activated for L-ch reception by the RSPI0 receive buffer full interrupt, and for R-ch reception by the RSPI1 receive buffer full interrupt. The DTC is activated for RSPI initialization by a compare match interrupt when an edge occurs on SSL. The DTC is activated for RSPI0 initialization on the rising edge of SSL and for RSPI1 initialization on the falling edge of SSL.

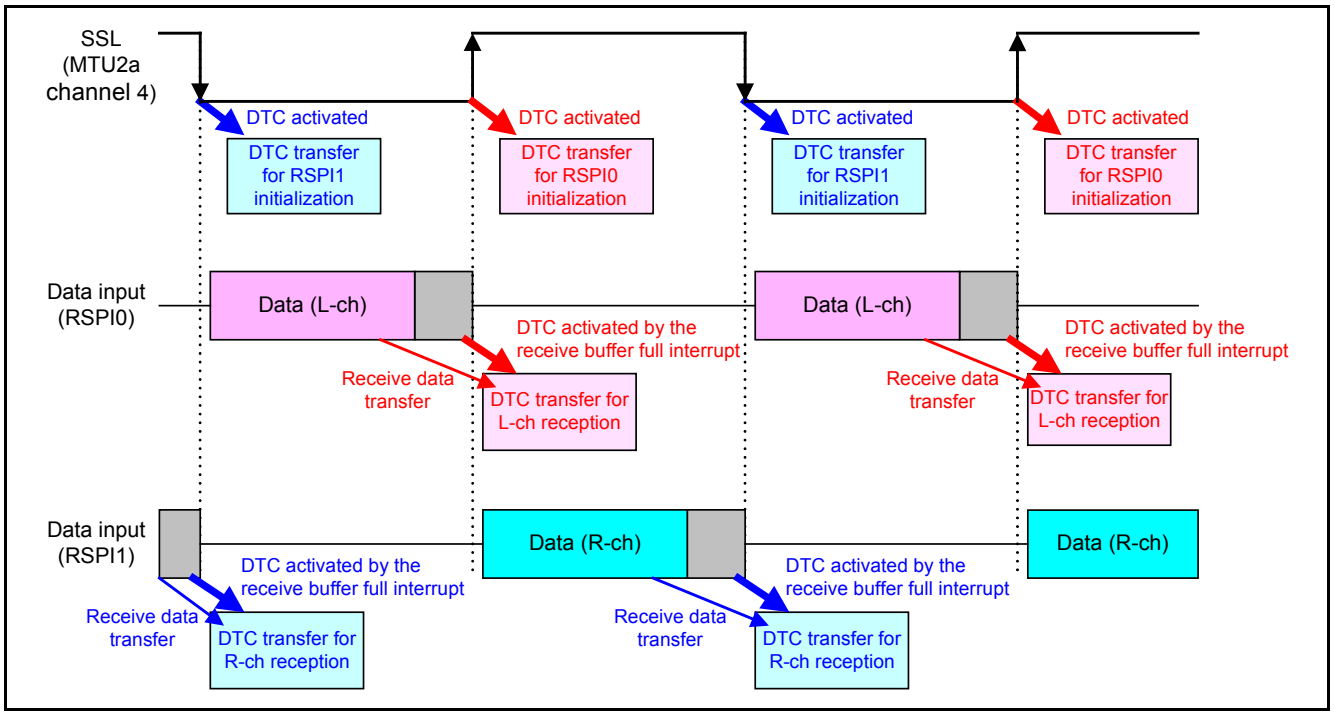

Figure 5.21 DTC Operation when Receiving

Figure 5.22 shows DTC Operation when Receiving L-ch data and Figure 5.23 shows DTC Operation when Receiving R-ch data

The DTC uses chain transfer to transfer the receive data and receive data address, and initialize the RSPI. To initialize the RSPI, disable the RSPI by writing 0 to the RSPI function enable bit (SPE) in the RSPI control register (SPCR), and then enable the RSPI again by writing 1 to the SPE bit.

Note: Refer to the User's Manual: Hardware for details on initialization with the RSPI function enable bit.

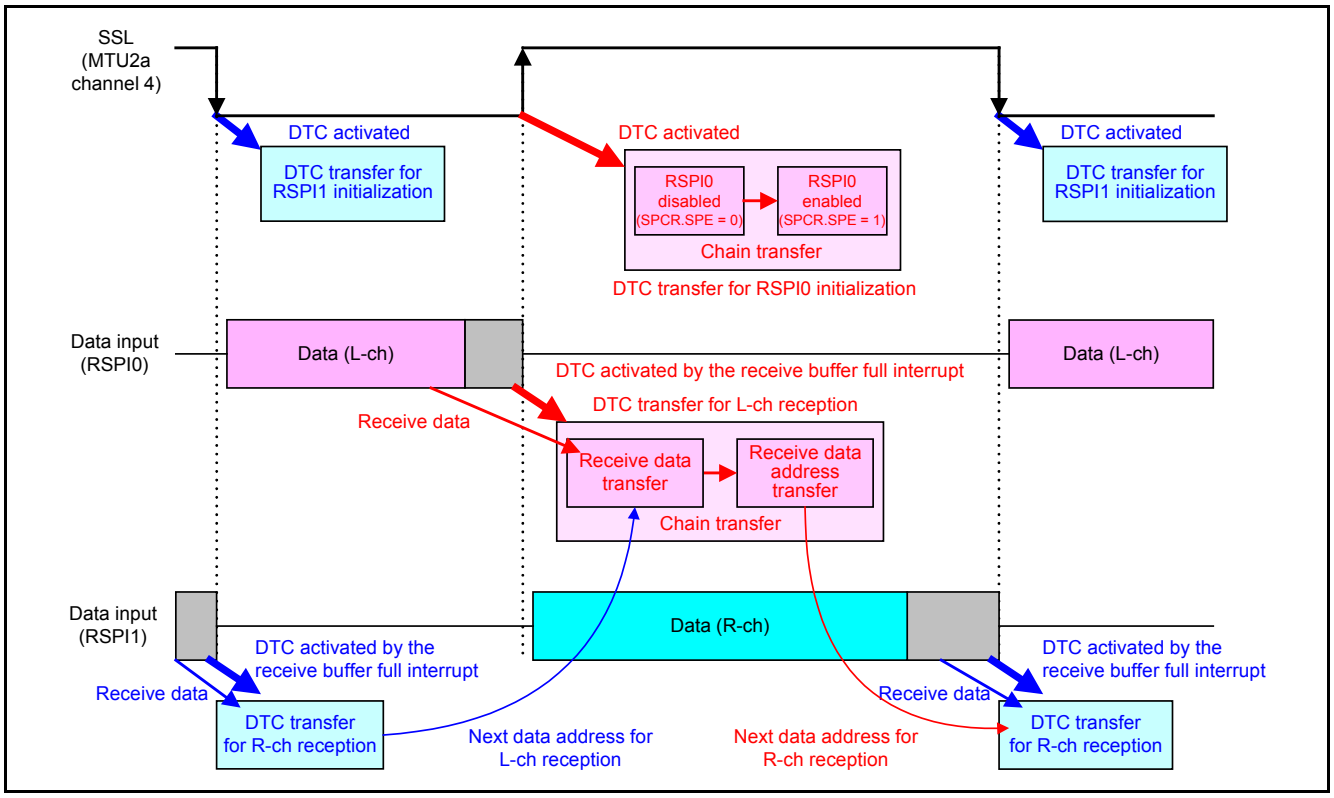

Figure 5.22 DTC Operation when Receiving L-ch data

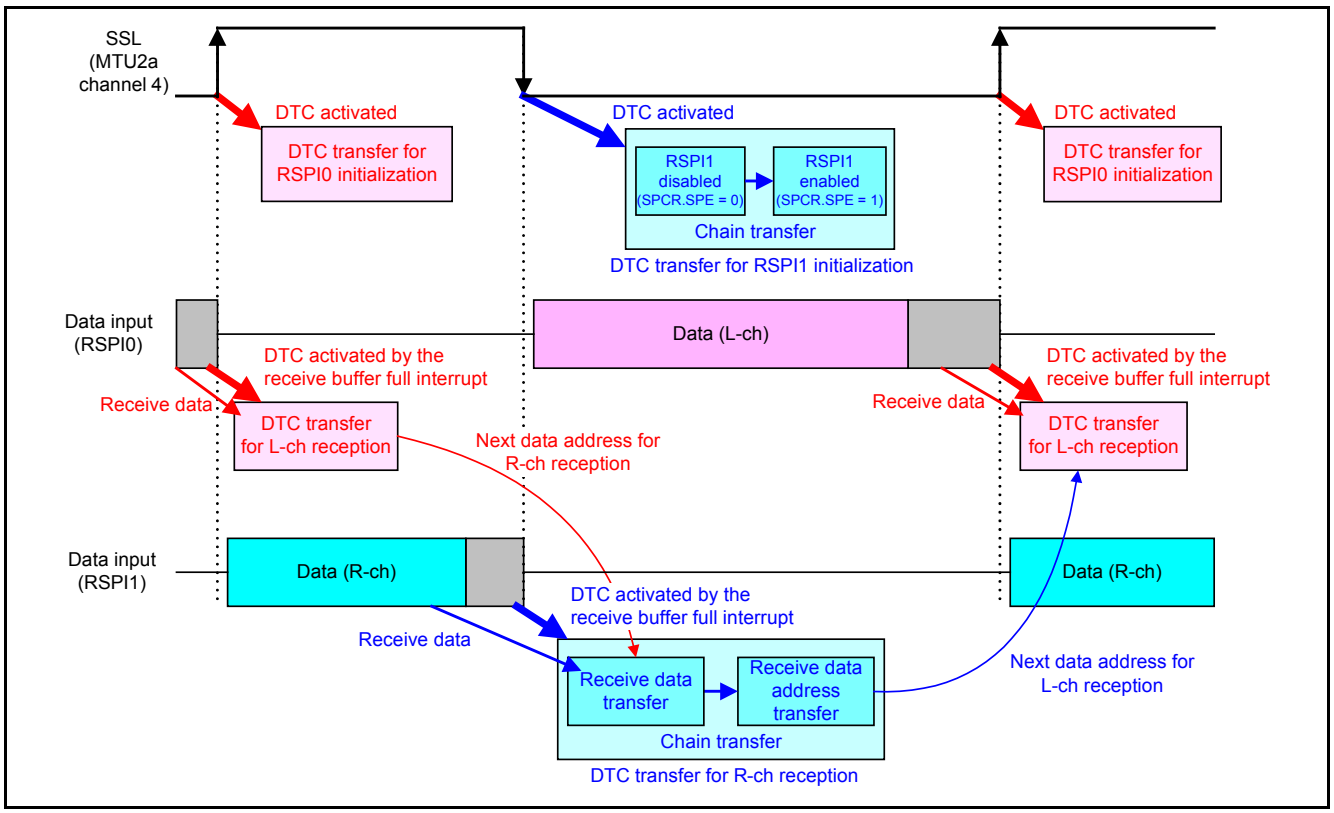

Figure 5.23 DTC Operation when Receiving R-ch data

5.1.3.3 Transferring Receive Data Address by the DTC

Figure 5.24 shows Receive Data Address Transfer by the DTC.

L-ch components and R-ch components of the received audio data are alternately placed in the on-chip RAM in 4-byte units. Refer to Figure 5.1 Audio Data in the On-Chip RAM. In the sample code, the receive data is transferred by performing L-ch reception and R-ch reception alternately using DTC transfer.

As shown in Figure 5.24, the DTC transfers data 0 of the receive data for L-ch reception, then the transfer destination address after completion of the L-ch reception is transferred using chain transfer to be used as the transfer destination address for data 0 of R-ch reception. The DTC transfers data 0 of the receive data for R-ch reception, then the transfer destination address after completion of the R-ch reception is transferred using chain transfer to be used as the transfer destination address for data 1 of L-ch reception. The DTC transfers data 1 of the receive data for L-ch reception, then the transfer destination address after completion of the L-ch reception is transferred using chain transfer to be used as the transfer destination address for next R-ch reception.

In this manner, the DTC transfers the receive data address as the transfer destination address for R-ch reception when transferring L-ch data, and as the transfer destination address for L-ch reception when transferring R-ch data.

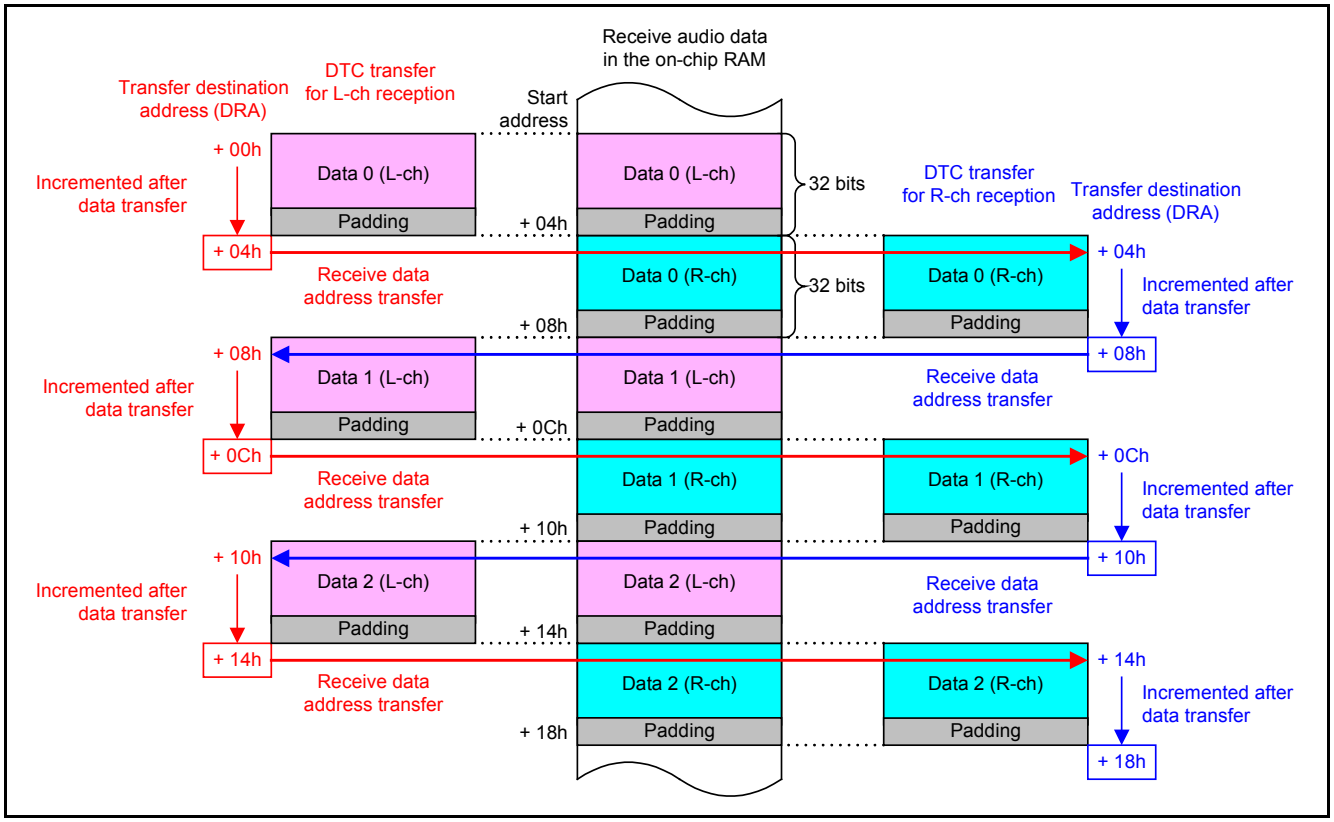

Figure 5.24 Receive Data Address Transfer by the DTC

5.1.3.4 Completion of Audio Data Reception

Audio data reception is completed by disabling the RSPI, thereby halting its operation. The RSPI is disabled by initializing RSPI0 and RSPI1 using DTC transfer.

(a) End Processing of L-ch Reception

Figure 5.25 shows End Processing of L-ch Reception.
End processing is performed by DTC transfer for RSPI0 initialization using the chain transfer at (n-1)th DTC activation. With end processing, processing at nth DTC activation is only disabling RSPI0, and then the RSPI0 operation stops.

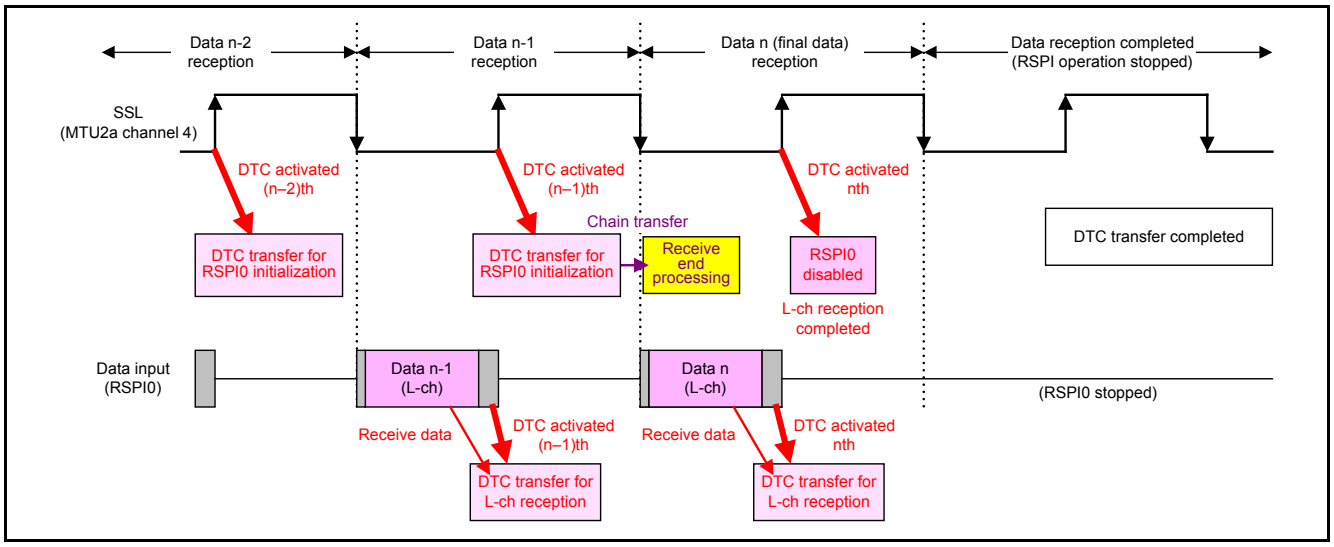

Figure 5.25 End Processing of L-ch Reception

Figure 5.26 shows End Processing of L-ch Reception by DTC Transfer for RSPI0 Initialization.
As shown in Figure 5.22 DTC Operation when Receiving L-ch data, with DTC transfer for RSPI0 initialization, the chain transfer is used for disabling and enabling RSPI0.
By setting the transfer counter, which is included in the transfer information for enabling RSPI0, to n–1, 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 performed by chain transfer after RSPI0 is enabled at (n–1)th DTC activation.
In receive end processing, by rewriting the DTC chain transfer enable bit in the transfer information for disabling RSPI0, chain transfer is disabled after RSPI0 is disabled at nth DTC activation. Therefore only RSPI0 is disabled at nth DTC activation, then the RSPI0 operation is stopped and the receive operation is completed.

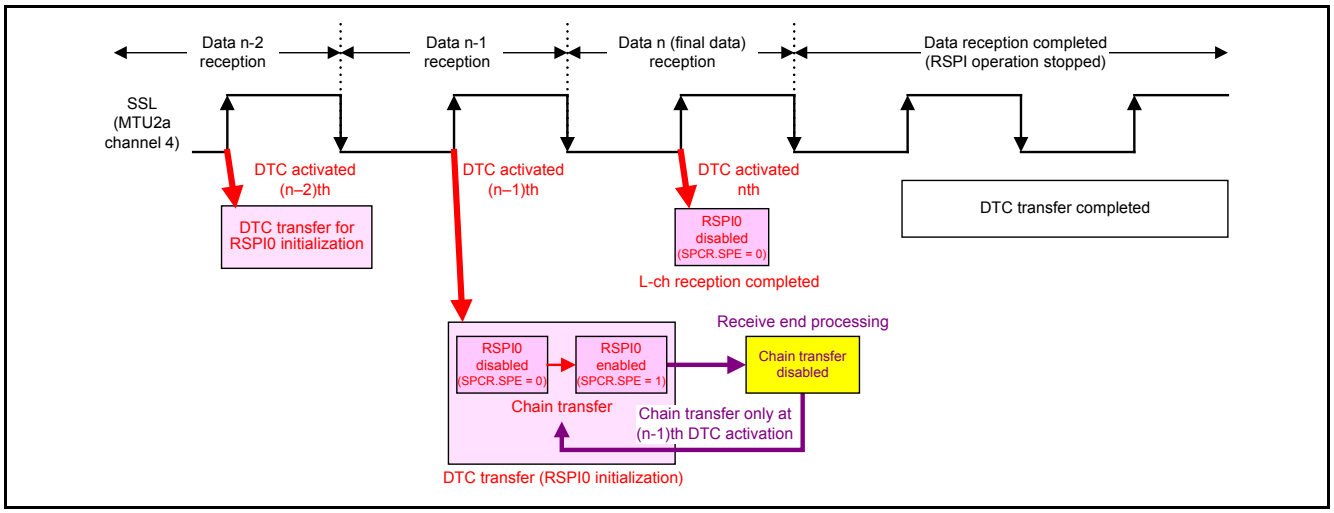

Figure 5.26 End Processing of L-ch Reception by DTC Transfer for RSPI0 Initialization

(b) End Processing of R-ch Reception

Figure 5.27 shows End Processing of R-ch Reception.

End processing for R-ch reception is performed by DTC transfer for initializing RSPI1 using chain transfer at nth DTC activation. With end processing, processing at (n+1)th DTC activation is only disabling RSPI1, and then the RSPI1 operation stops.

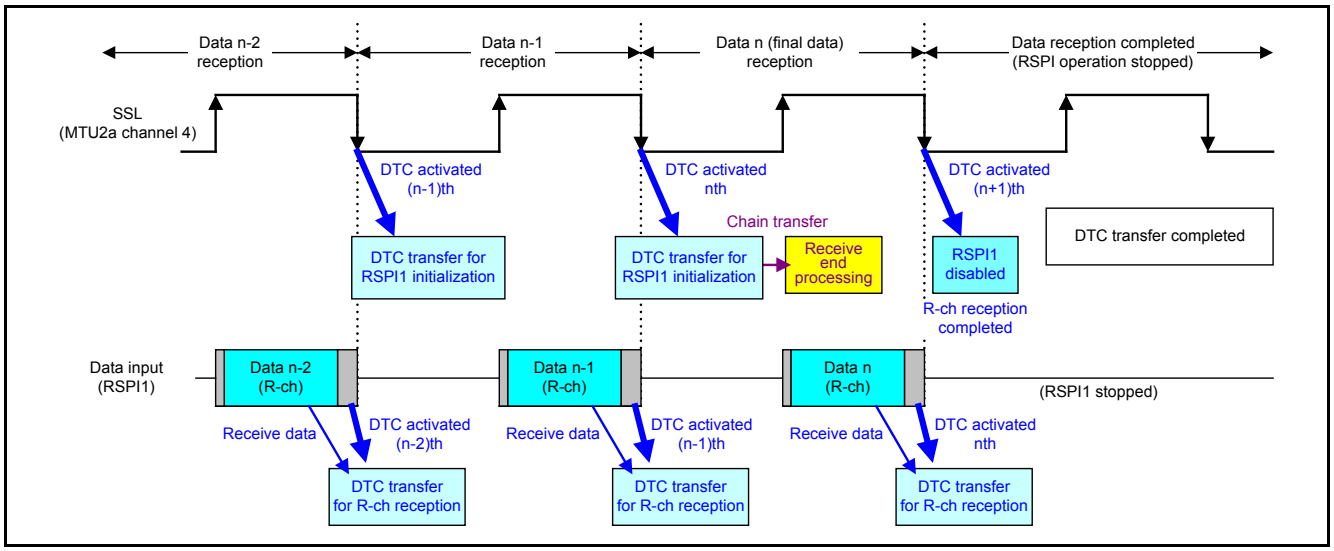

Figure 5.27 End Processing of R-ch Reception

Figure 5.28 shows End Processing of R-ch Reception by DTC Transfer for RSPI1 Initialization.

As shown in Figure 5.23 DTC Operation when Receiving R-ch data, with the DTC transfer for RSPI1 initialization, the chain transfer is used for disabling and enabling RSPI1.

By setting the transfer counter, which is included in the transfer information for enabling RSPI1, to n, 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 performed by chain transfer after RSPI1 is enabled at nth DTC activation.

In receive end processing, by rewriting the DTC chain transfer enable bit in the transfer information for disabling RSPI1, chain transfer is disabled after RSPI1 is disabled at (n+1)th DTC activation. Therefore only RSPI1 is disabled at (n+1)th DTC activation, then the RSPI1 operation is stopped and the receive operation is completed.

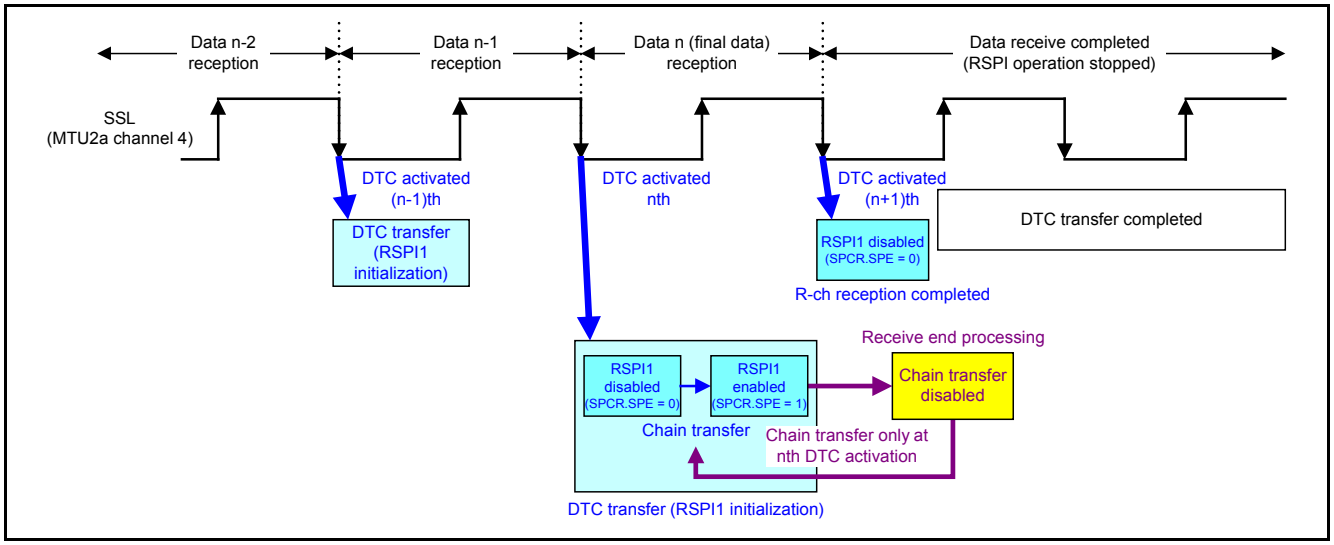

Figure 5.28 End Processing of R-ch Reception by DTC Transfer for RSPI1 Initialization

5.2 File Composition

Table 5.1 lists the Files Used in the Sample Code. Files generated by the integrated development environment are not included in this table.

Table 5.1 Files Used in the Sample Code

File Name	Outline	Remarks
main.c	Main Processing	
r_init_stop_module.c	Stop processing for active peripheral functions after a reset	
r_init_stop_module.h	Header file for r_init_stop_module.c	
r_init_non_existent_port.c	Nonexistent port initialization	
r_init_non_existent_port.h	Header file for r_init_non_existent_port.c	
r_init_clock.c	Clock initialization	
r_init_clock.h	Header file for r_init_clock.c	
i2s_dtc.c	Settings for the DTCa operation	
i2s_mtu2.c	Settings for the MTU2a operation	
i2s_rspi.c	Settings for the RSPI operation	Settings for transmit and receive data format
i2s.h	Macro definitions for I2S communication	

5.3 Option-Setting Memory

Table 5.2 lists the Option-Setting Memory Configured in the Sample Code. When necessary, set a value suited to the user system.

Table 5.2 Option-Setting Memory Configured in the Sample Code

Symbol	Address	Setting Value	Contents
OFS0	FFFF FF8Fh to FFFF FF8Ch	FFFF FFFFh	The IWDT is stopped after a reset. The WDT is stopped after a reset.
OFS1	FFFF FF8Bh to FFFF FF88h	FFFF FFFFh	The voltage monitor 0 reset is disabled after a reset. HOCO oscillation is disabled after a reset.
MDES	FFFF FF83h to FFFF FF80h	FFFF FFFFh	Little endian

5.4 Constants

Table 5.3 lists the Constants Used in the Sample Code.

Table 5.3 Constants Used in the Sample Code

Constant Name	Setting Value	Contents
L_CH	0	L-ch index No. of the audio data array in the on-chip RAM
R_CH	1	R-ch index No. of the audio data array in the on-chip RAM
NULL_DATA	0x00000000	Initialization data for the receive audio data area
TRANSMIT_FORMAT	STANDARD_PADDING or BACKWARD_PADDING or FORWARD_PADDING	Transmit and receive data format selection (select from 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 mode
RECEIVE_MODE	1	Receive mode
TRANSCEIVE_MODE	2	Transmit and receive mode
AUD_SIZE	0x100	Audio data total byte count
AUD_NUM	(AUD_SIZE / 4) / 2	The number of L-ch/R-ch audio data
DTC_TX_CHAIN_SIZE	5	The number of transfer information for DTCa chain transfer when transmitting
DTC_RX_CHAIN_SIZE	2	The number of transfer information for DTCa chain transfer when receiving
DTC_TX_L_COUNT	AUD_NUM - 1	The number of DTC transfers for L-ch transmission or RSPI0 initialization
DTC_TX_R_COUNT	AUD_NUM	The number of DTC transfers for R-ch transmission or RSPI1 initialization
DTC_RX_L_COUNT	AUD_NUM	The number of DTC transfers for L-ch reception
DTC_RX_R_COUNT	AUD_NUM	The number of DTC transfers for R-ch reception
SCK_CYCLE_VALUE	0x0003	SCK cycle
WS_SSL_CYCLE_VALUE	0x00FF	WS and SSL cycle
SSL_DELAY_VALUE_S	0x0003	Phase value of WS and SSL for standard format
SSL_DELAY_VALUE_B	0x0001	Phase value of WS and SSL for backward-padding format
SSL_DELAY_VALUE_F	0x001F	Phase value of WS and SSL for forward-padding format

5.5 Structure/Union List

Figure 5.29 shows the Structure/Union Used in the Sample Code.

```
#ifdef __LIT
struct st_dtc_data{                         /* Little-endian */
        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{                         /* Big-endian */
        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 5.29 Structure/Union Used in the Sample Code

5.6 Variables

Table 5.4 lists the Global Variables and Table 5.5 lists the const Variables.

Table 5.4 Global Variables

Type	Variable Name	Contents	Function Used
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]	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 5.5 const Variables

Type	Variable Name	Contents	Function Used
const uint8_t	RSPI_TX_DISABLE	Register setting value for RSPI initialization when transmitting	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const uint8_t	RSPI_TX_ENABLE	Register setting value for canceling RSPI initialization when transmitting	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const uint8_t	RSPI_RX_DISABLE	Register setting value for RSPI initialization when receiving	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const uint8_t	RSPI_RX_ENABLE	Register setting value for canceling RSPI initialization when receiving	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const uint8_t	RSPI_TRX_DISABLE	Register setting value for RSPI initialization when transmitting and receiving	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const uint8_t	RSPI_TRX_ENABLE	Register setting value for canceling RSPI initialization when transmitting and receiving	i2s_dtc_tx_l_init i2s_dtc_tx_r_init
const uint8_t	DTC_CHAIN_DISABLE	Register setting value for disabling chain transfer	i2s_dtc_tx_l_init i2s_dtc_tx_r_init

5.7 Functions

Table 5.6 lists the Functions Used in the Sample Code.

Table 5.6 Functions Used in the Sample Code

Function Name	Outline
main	Main processing
R_INIT_StopModule	Stop processing for active peripheral functions after a reset
R_INIT_NonExistentPort	Nonexistent port initialization
R_INIT_Clock	Clock initialization
i2s_au_data_init	Creating audio data in the on-chip RAM
i2s_start	I2S communication start processing
i2s_dtc_init	DTC initialization
i2s_dtc_tx_l_init	Creating DTC transfer information for L-ch transmission or RSPI0 initialization
i2s_dtc_tx_r_init	Creating DTC transfer information for R-ch transmission or RSPI1 initialization
i2s_dtc_rx_l_init	Creating DTC transfer information for L-ch reception
i2s_dtc_rx_r_init	Creating DTC transfer information for R-ch reception
i2s_mtu2_init	MTU2a initialization
i2s_mtu2_ch2_init	MTU2a channel 2 initialization for SCK generation
i2s_mtu2_ch3_init	MTU2a channel 3 initialization for WS generation
i2s_mtu2_ch4_init	MTU2a channel 4 initialization for SSL generation
i2s_rspi_init	RSPI initialization
i2s_rspi0_init	RSPI0 initialization for L-ch transmission and reception
i2s_rspi1_init	RSPI1 initialization for R-ch transmission and reception

5.8 Function Specifications

The following tables list the sample code function specifications.

main	
Outline	Main processing
Header	i2s.h
Declaration	void main(void)
Description	Call following functions; port initialization, clock initialization, peripheral function initialization (MTU2a, RSPI, and DTC)
Arguments	None
Return Value	None

R_INIT_StopModule	
Outline	Stop processing for active peripheral functions after a reset
Header	r_init_stop_module.h
Declaration	void R_INIT_StopModule(void)
Description	Configure the setting to enter the module-stop state.
Arguments	None
Return Value	None
Remarks	Transition to the module-stop state is not performed in the sample code. Refer to the RX630 Group Initial Setting Rev. 1.00 application note for details on this function.

R_INIT_NonExistentPort	
Outline	Nonexistent port initialization
Header	r_init_non_existent_port.h
Declaration	void R_INIT_NonExistentPort(void)
Description	Initialize port direction registers for ports that do not exist in products with less than 176 pins.
Arguments	None
Return Value	None
Remarks	The number of pins in the sample code is set for the 100-pin package (PIN_SIZE=100). After this function is called, when writing in byte units to the PDR registers or PODR registers which have nonexistent ports, set the corresponding bits for nonexistent ports as follows: set the I/O select bits in the PDR registers to 1 and set the output data store bits in the PODR registers to 0. Refer to the RX630 Group Initial Setting Rev. 1.00 application note for details on this function.

R_INIT_Clock	
Outline	Clock initialization
Header	r_init_clock.h
Declaration	void R_INIT_Clock (void)
Description	Initialize clocks.
Arguments	None
Return Value	None
Remarks	The sample code selects processing which uses PLL as the system clock without using the sub-clock. Refer to the RX630 Group Initial Setting Rev. 1.00 application note for details on this function.

i2s_au_data_init	
Outline	Creating audio data in the on-chip RAM
Header	i2s.h
Declaration	void i2s_au_data_init (void)
Description	- Create transmit audio data (tx_au_data). - Clear the receive audio data area to 0 (rx_au_data).
Arguments	None
Return Value	None

i2s_start		
Outline	I2S communication start processing	
Header	i2s.h, iodefine.h	
Declaration	void i2s_start (char i2s_mode)	
Description	- Enable the RSPI function. - Activate the DTC module. - Start MTU2a count operation.	
Arguments	i2s_mode	: Operating mode selection
Return Value	None	

i2s_dtc_init		
Outline	DTC initialization	
Header	iodefine.h	
Declaration	void i2s_dtc_init (char i2s_mode)	
Description	- Create the DTC vector table. - Enable DTC activation by interrupts.	
Arguments	i2s_mode	: Operating mode selection
Return Value	None	

i2s_dtc_tx_l_init		
Outline	Creating DTC transfer information for L-ch transmission or RSPI0 initialization	
Header	i2s.h, iodefine.h	
Declaration	void i2s_ dtc_tx_l_init (char i2s_mode)	
Description	Transfer transmit data from the on-chip RAM to RSPI0	
Arguments	i2s_mode	: Operating mode selection
Return Value	None	

i2s_dtc_tx_r_init		
Outline	Creating DTC transfer information for R-ch transmission or RSPI1 initialization	
Header	i2s.h, iodefine.h	
Declaration	void i2s_ dtc_tx_r_init (char i2s_mode)	
Description	Transfer transmit data from the on-chip RAM to RSPI1	
Arguments	i2s_mode	: Operating mode selection
Return Value	None	

i2s_dtc_rx_l_init	
Outline	Creating DTC transfer information for L-ch reception
Header	i2s.h, iodefine.h
Declaration	void i2s_dtc_rx_l_init (void)
Description	Transfer receive data from RSPI0 to the on-chip RAM.
Arguments	None
Return Value	None

i2s_dtc_rx_r_init	
Outline	Creating DTC transfer information for R-ch reception
Header	i2s.h, iodefine.h
Declaration	void i2s_dtc_rx_r_init (void)
Description	Transfer receive data from RSPI1 to the on-chip RAM.
Arguments	None
Return Value	None

i2s_mtu2_init	
Outline	MTU2a initialization
Header	iodefine.h
Declaration	void i2s_mtu2_init (void)
Description	Configure MTU2a ports.
Arguments	None
Return Value	None

i2s_mtu2_ch2_init	
Outline	MTU2a channel 2 initialization for SCK generation
Header	i2s.h, iodefine.h
Declaration	void i2s_mtu2_ch2_init (void)
Description	Generate SCK from the external clock and output it.
Arguments	None
Return Value	None

i2s_mtu2_ch3_init	
Outline	MTU2a channel 3 initialization for WS generation
Header	i2s.h, iodefine.h
Declaration	void i2s_mtu2_ch3_init(void)
Description	Generate WS from the external clock and output it.
Arguments	None
Return Value	None

i2s_mtu2_ch4_init	
Outline	MTU2a channel 4 initialization for SSL generation
Header	i2s.h, iodefine.h
Declaration	void i2s_mtu2_ch4_init(void)
Description	- Generate SSL from the external clock and output it. - Enable the compare match interrupt
Arguments	None
Return Value	None

i2s_rspi_init		
Outline	RSPI initialization	
Header	None	
Declaration	void i2s_rspi_init (char i2s_mode)	
Description	Configure the RSPI.	
Arguments	i2s_mode	: Operating mode selection
Return Value	None	

i2s_rspi0_init		
Outline	RSPI0 initialization for L-ch transmission and reception	
Header	iodefine.h	
Declaration	void i2s_rspi0_init (char i2s_mode)	
Description	Configure RSPI0.	
Arguments	i2s_mode	: Operating mode selection
Return Value	None	

i2s_rspi1_init		
Outline	RSPI1 initialization for R-ch transmission and reception	
Header	iodefine.h	
Declaration	void i2s_rspi1_init (char i2s_mode)	
Description	Configure RSPI1.	
Arguments	i2s_mode	: Operating mode selection
Return Value	None	

6 Flowcharts

6.1 Main Processing

Figure 6.1 shows the Main Processing.

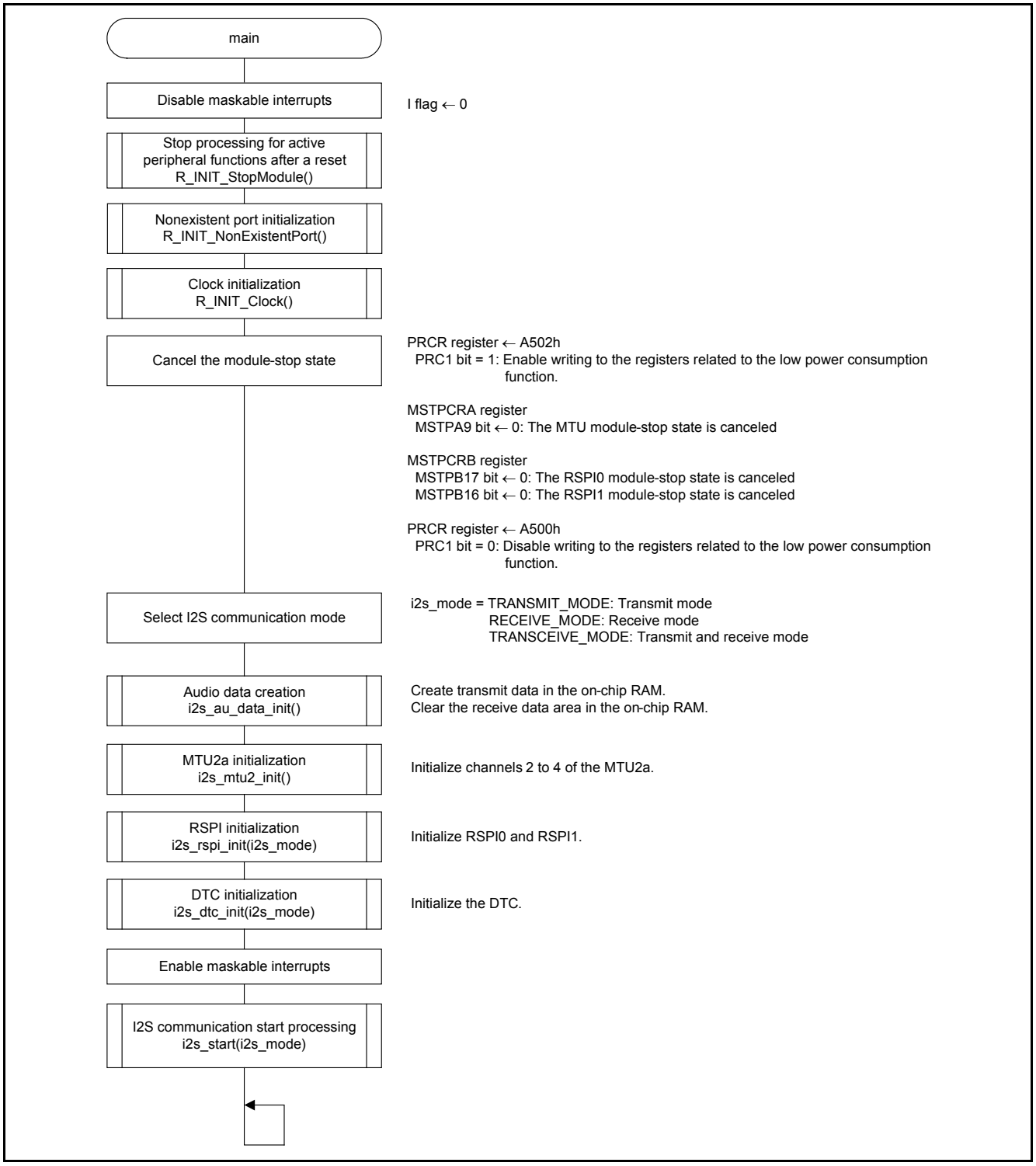

Figure 6.1 Main Processing

6.2 Functions

6.2.1 i2s_au_data_init

Figure 6.2 shows the Audio Data Creation.

The i2s_au_data_init function creates audio data in the on-chip RAM.

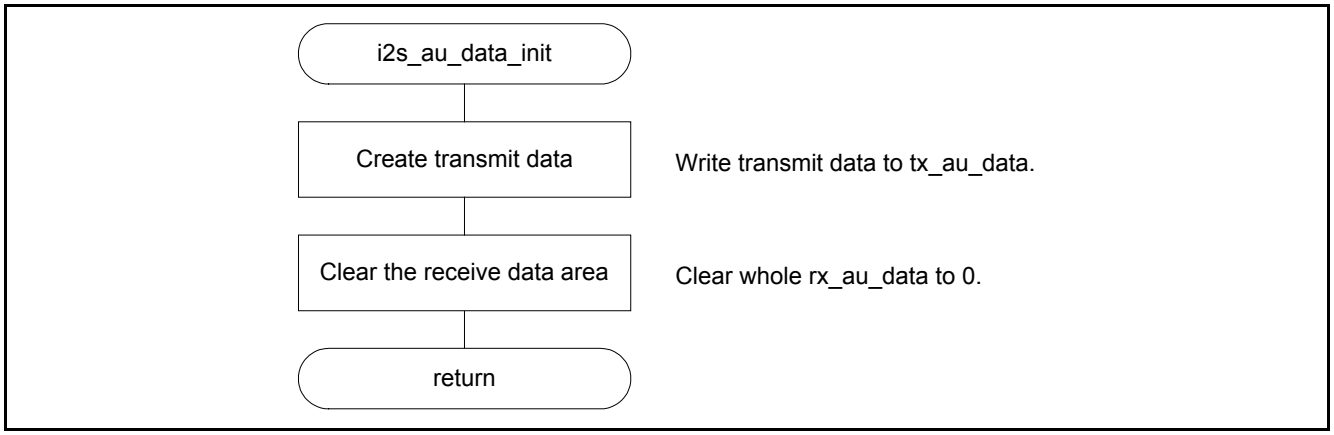

Figure 6.2 Audio Data Creation

6.2.2 i2s_start

Figure 6.3 shows the I2S Communication Start Processing.

The i2s_start function enables peripheral functions and starts the I2S communication.

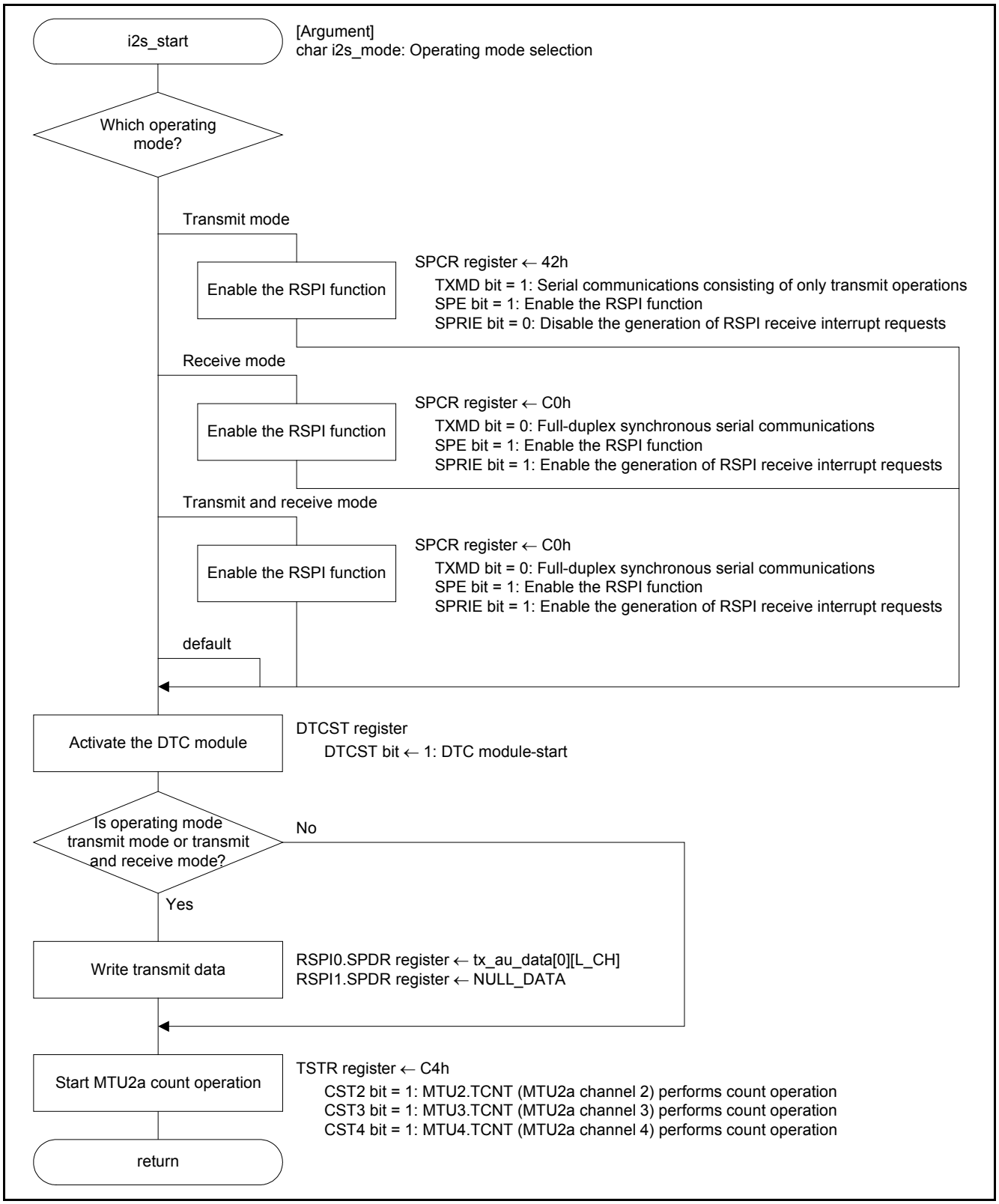

Figure 6.3 I2S Communication Start Processing

6.2.3 i2s_dtc_init

Figure 6.4 shows the DTC Initialization.

The i2s_dtc_init function initializes the DTCa.

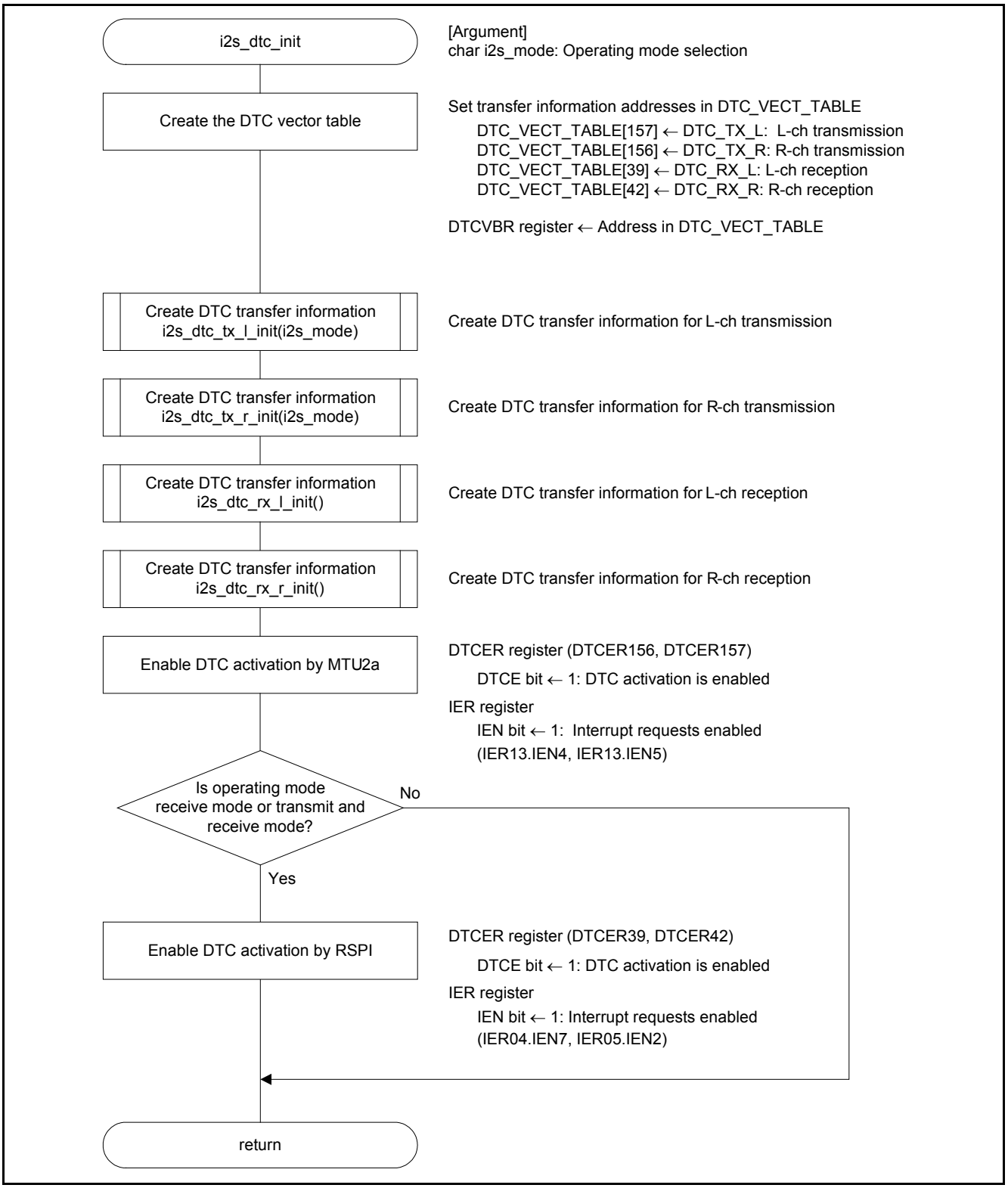

Figure 6.4 DTC Initialization

6.2.4 i2s_dtc_tx_l_init

Figure 6.5 to Figure 6.8 show the Creating DTC Transfer Information for L-ch Transmission or RSPI0 Initialization.

The i2s_dtc_tx_l_init function creates DTC transfer information for L-ch transmission for transmit mode, or transmit and receive mode, or for RSPI0 initialization for receive mode.

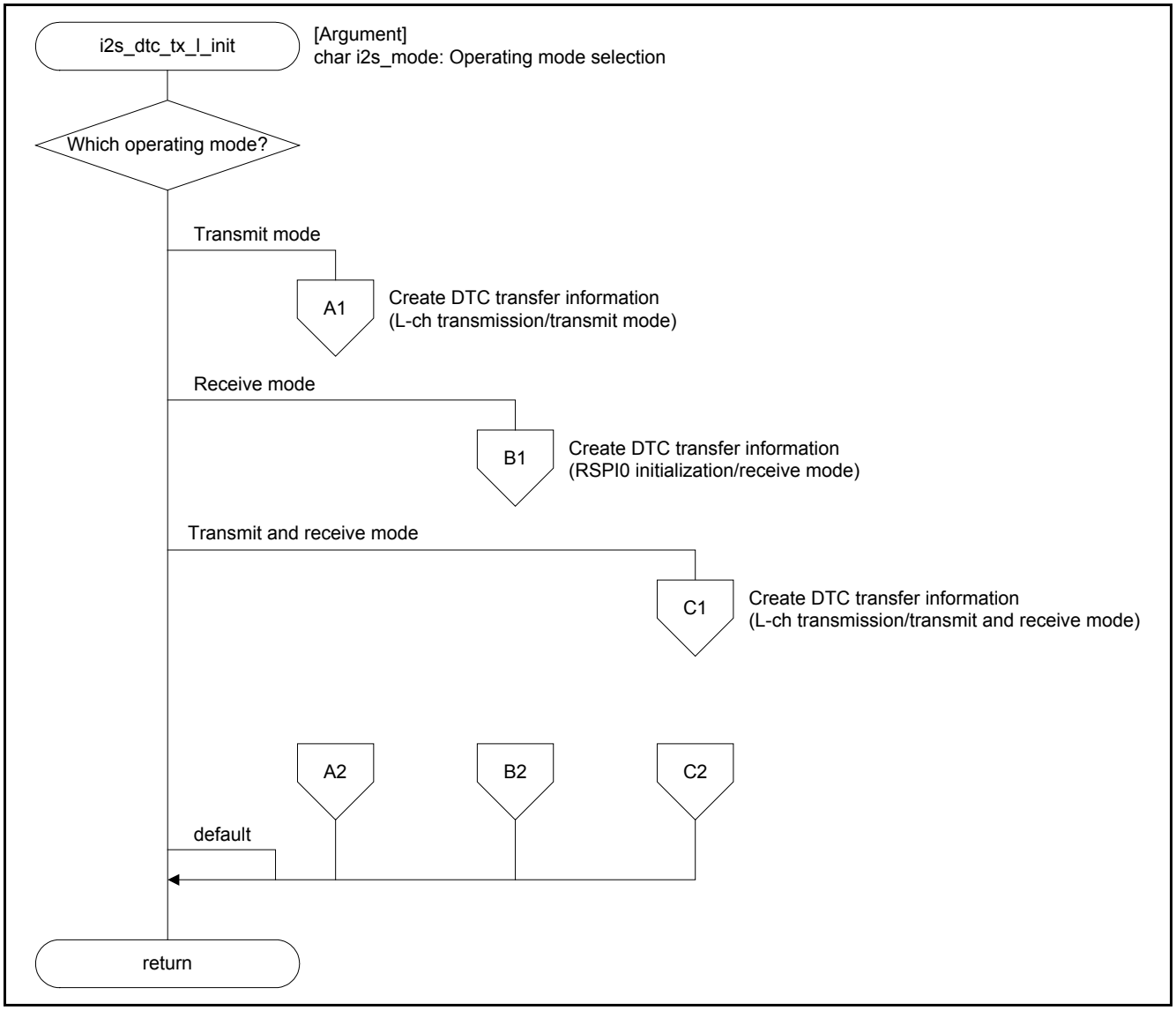

Figure 6.5 Creating DTC Transfer Information for L-ch Transmission or RSPI0 Initialization (1/4)

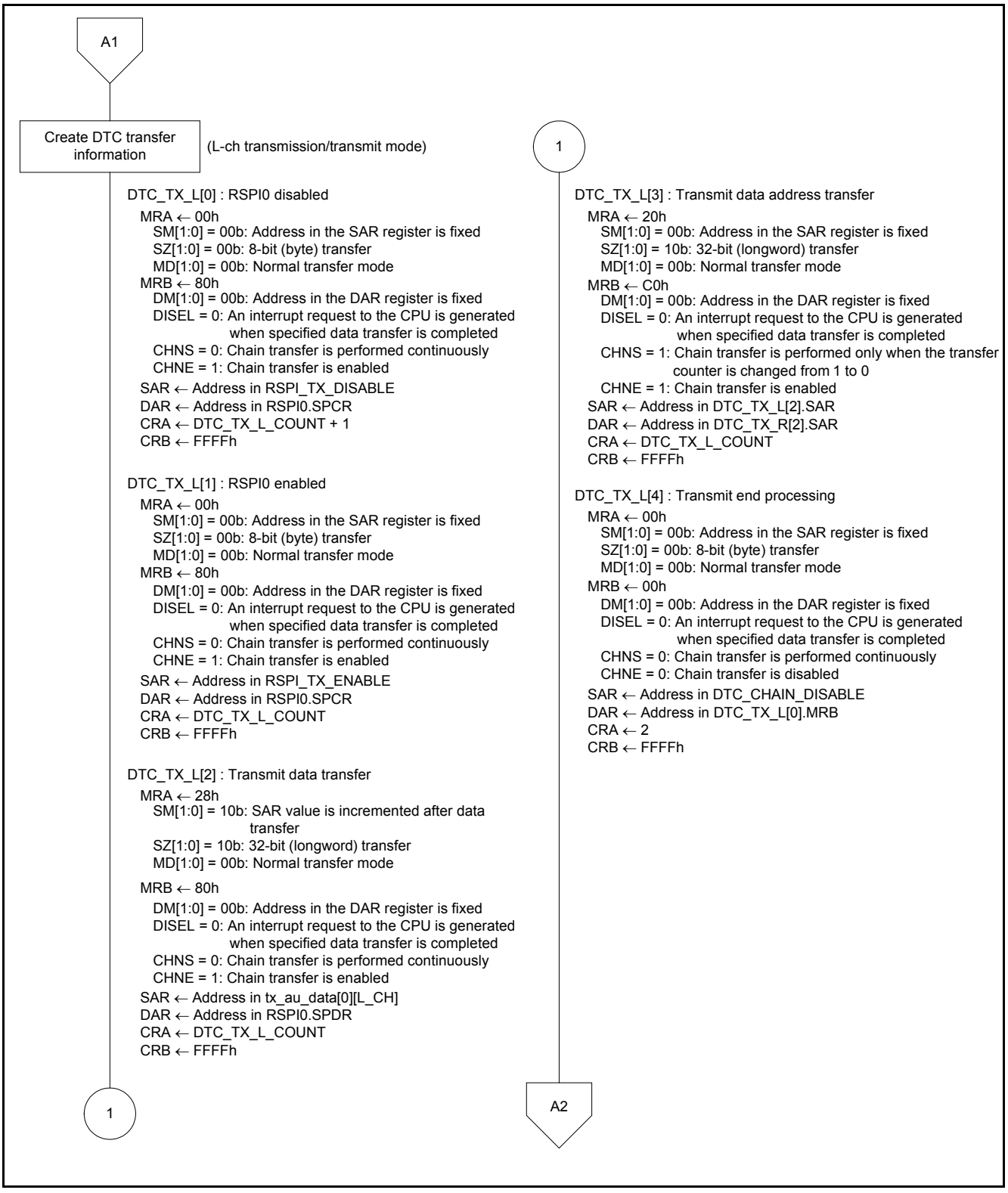

Figure 6.6 Creating DTC Transfer Information for L-ch Transmission or RSPI0 Initialization (2/4)

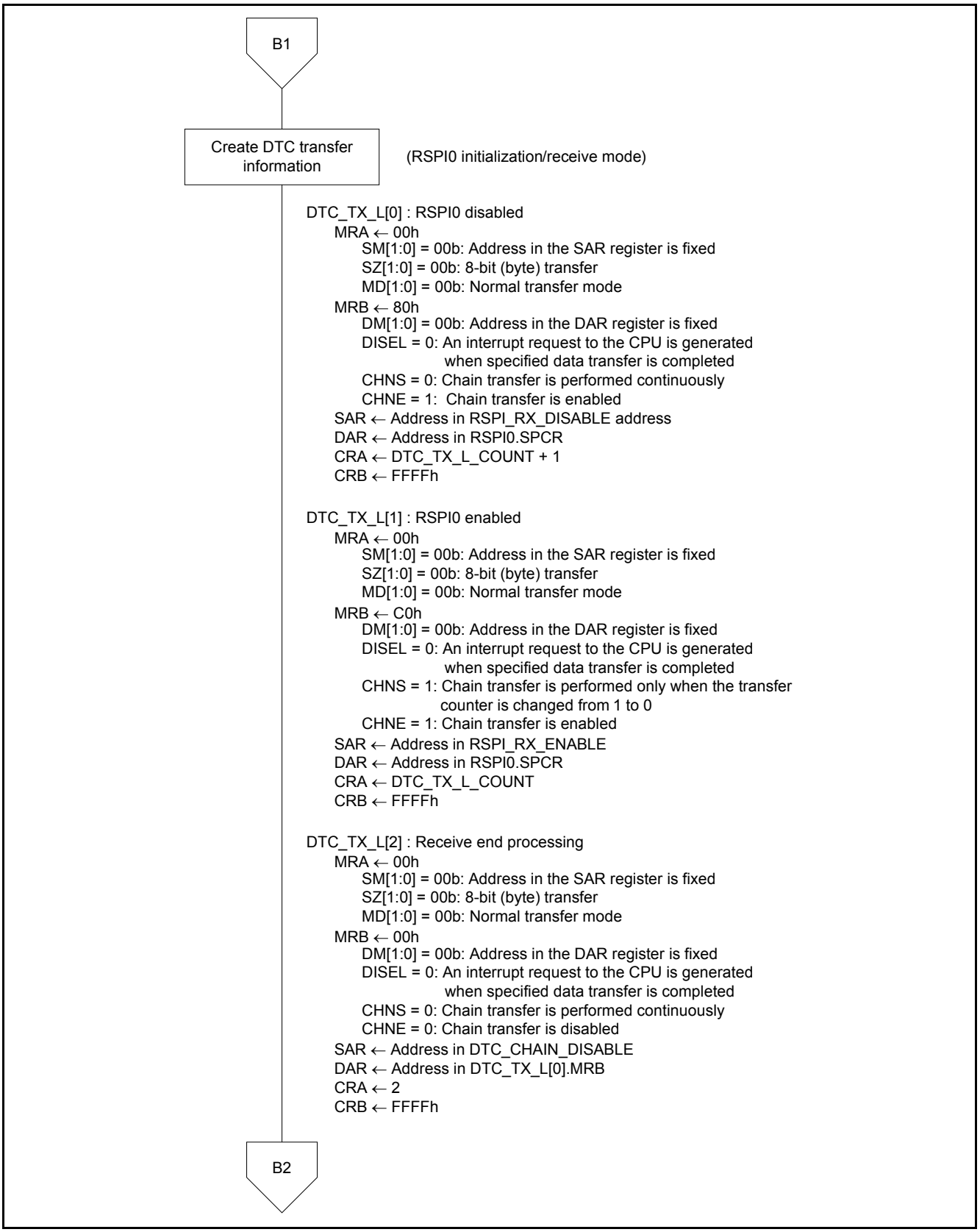

Figure 6.7 Creating DTC Transfer Information for L-ch Transmission or RSPI0 Initialization (3/4)

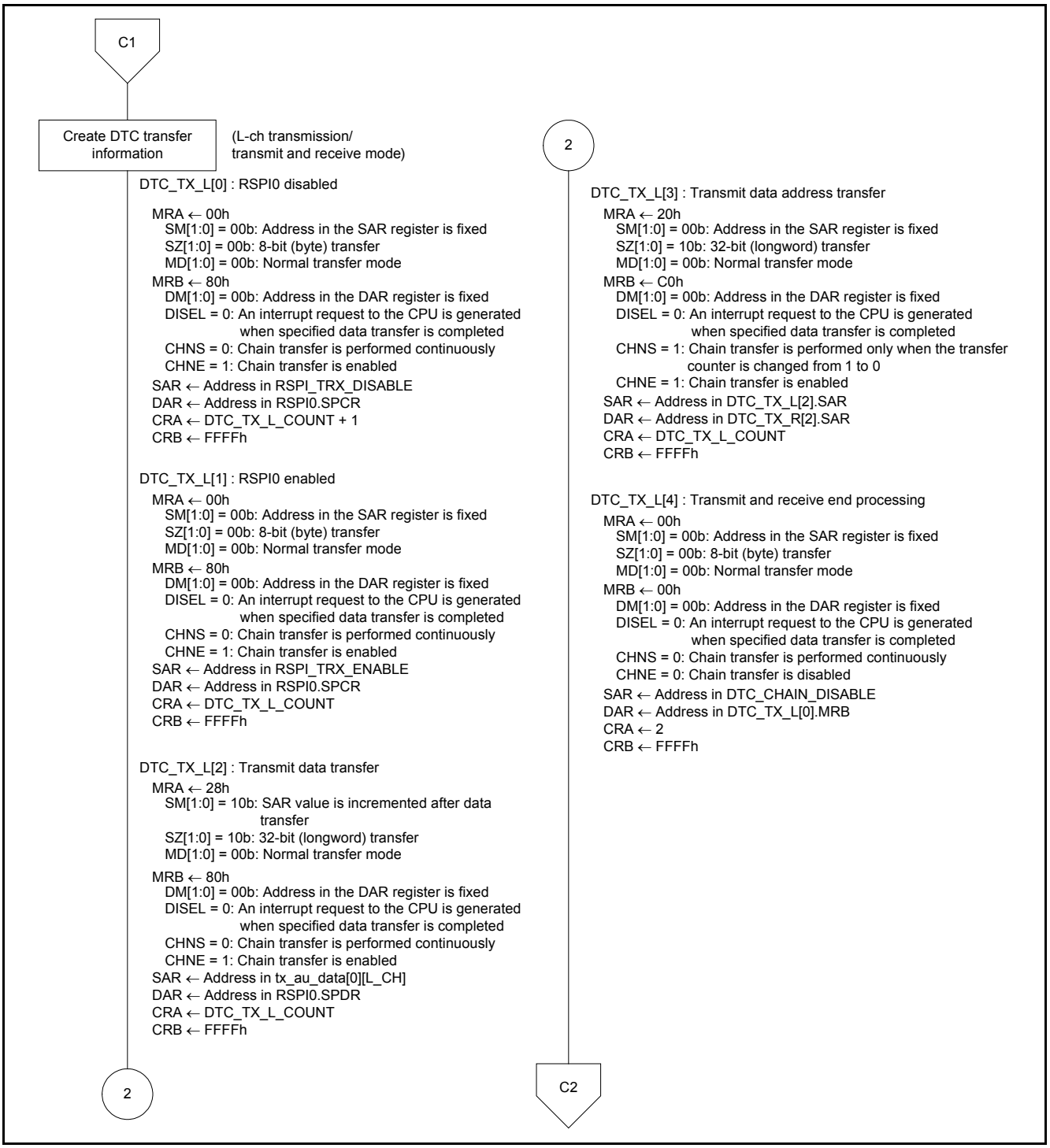

Figure 6.8 Creating DTC Transfer Information for L-ch Transmission or RSPI0 Initialization (4/4)

6.2.5 i2s_dtc_tx_r_init

Figure 6.9 to Figure 6.12 show the Creating DTC Transfer Information for R-ch Transmission or RSPI1 Initialization.

The i2s_dtc_tx_r_init function creates DTC transfer information for R-ch transmission for transmit mode, or transmit and receive mode, or for RSPI1 initialization for receive mode.

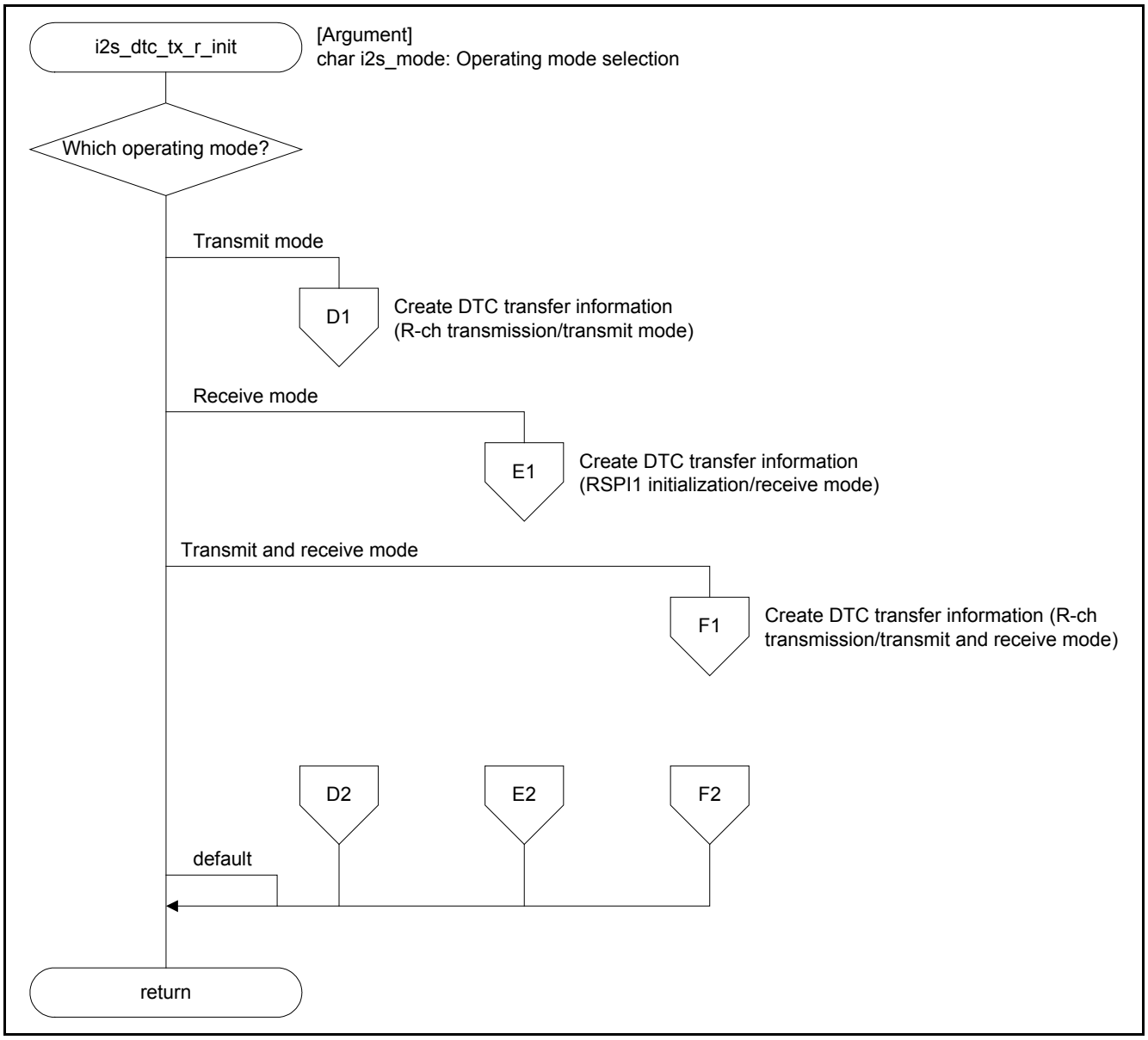

Figure 6.9 Creating DTC Transfer Information for R-ch Transmission or RSPI1 Initialization (1/4)

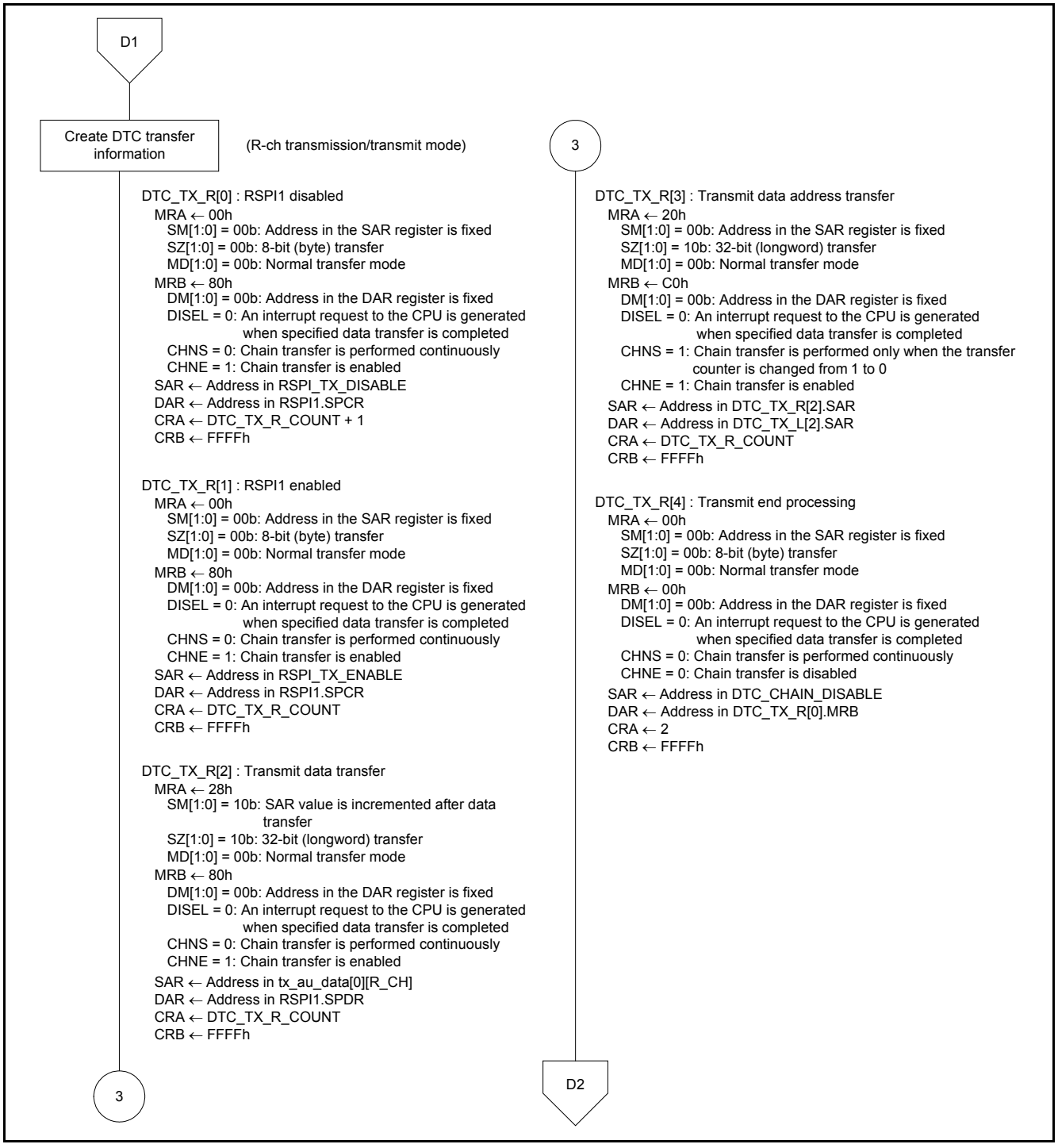

Figure 6.10 Creating DTC Transfer Information for R-ch Transmission or RSPI1 Initialization (2/4)

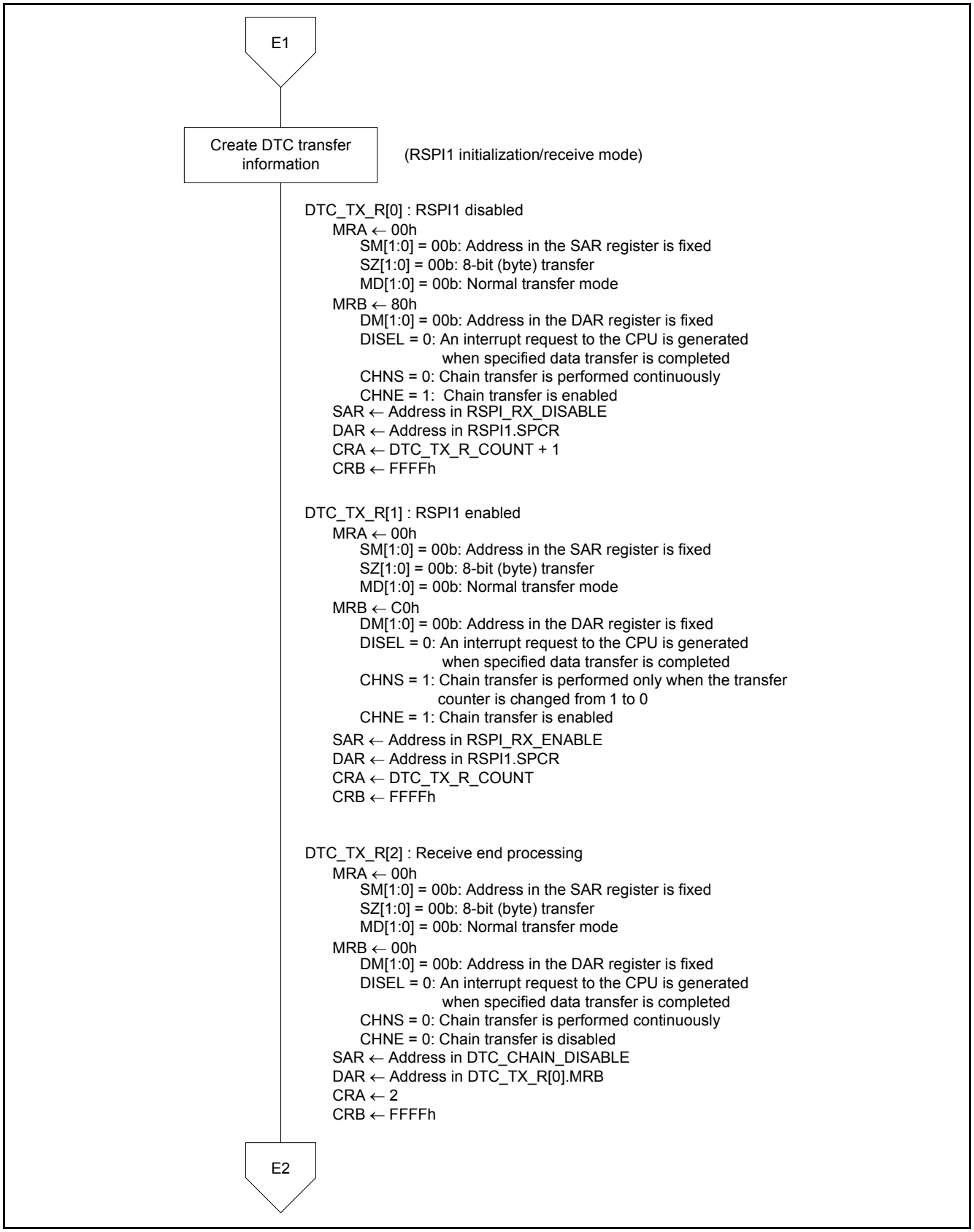

Figure 6.11 Creating DTC Transfer Information for R-ch Transmission or RSPI1 Initialization (3/4)

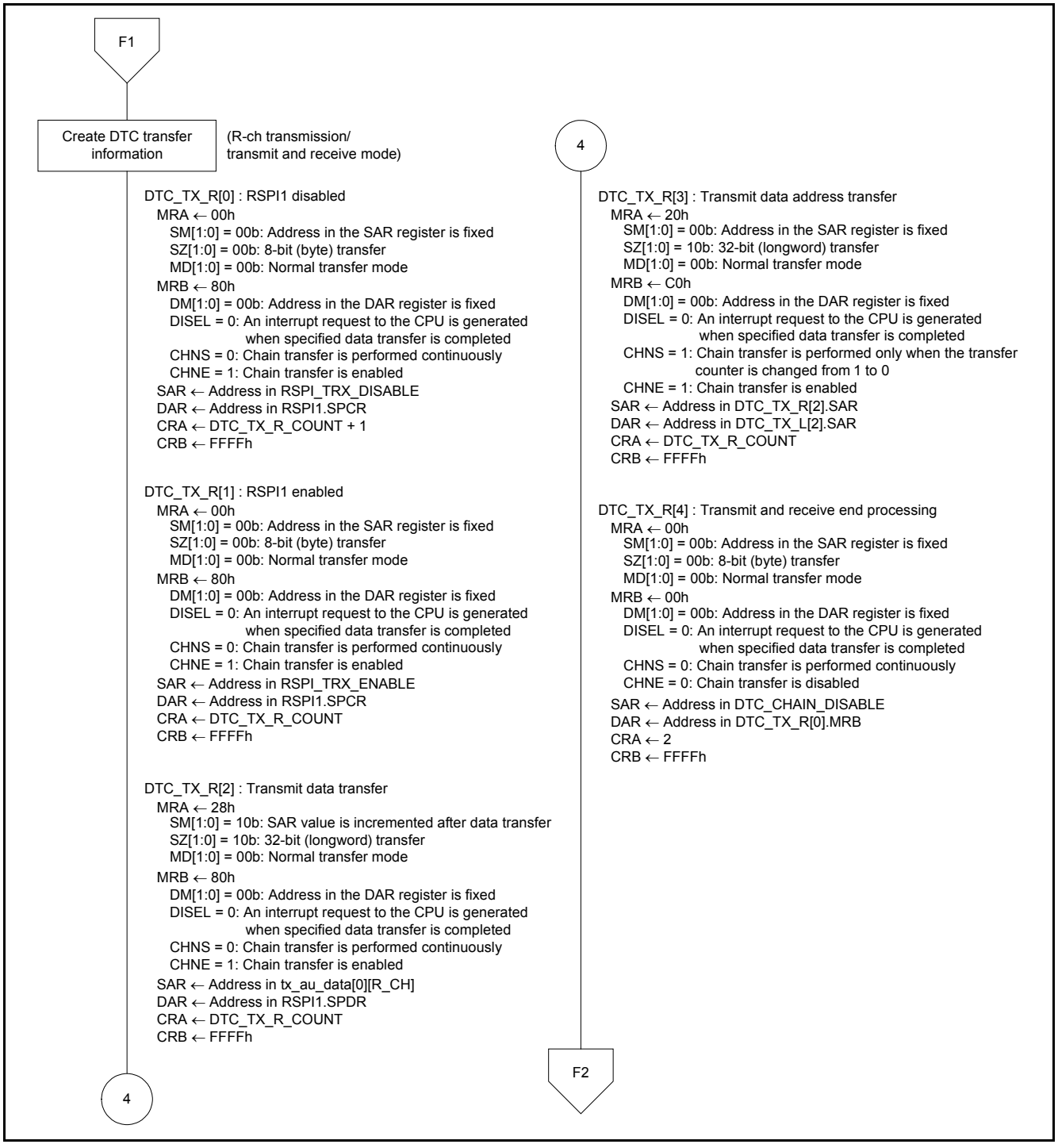

Figure 6.12 Creating DTC Transfer Information for R-ch Transmission or RSPI1 Initialization (4/4)

6.2.6 i2s_dtc_rx_l_init

Figure 6.13 shows the Creating DTC Transfer Information for L-ch Reception.

The i2s_dtc_rx_l_init function creates DTC transfer information for L-ch reception.

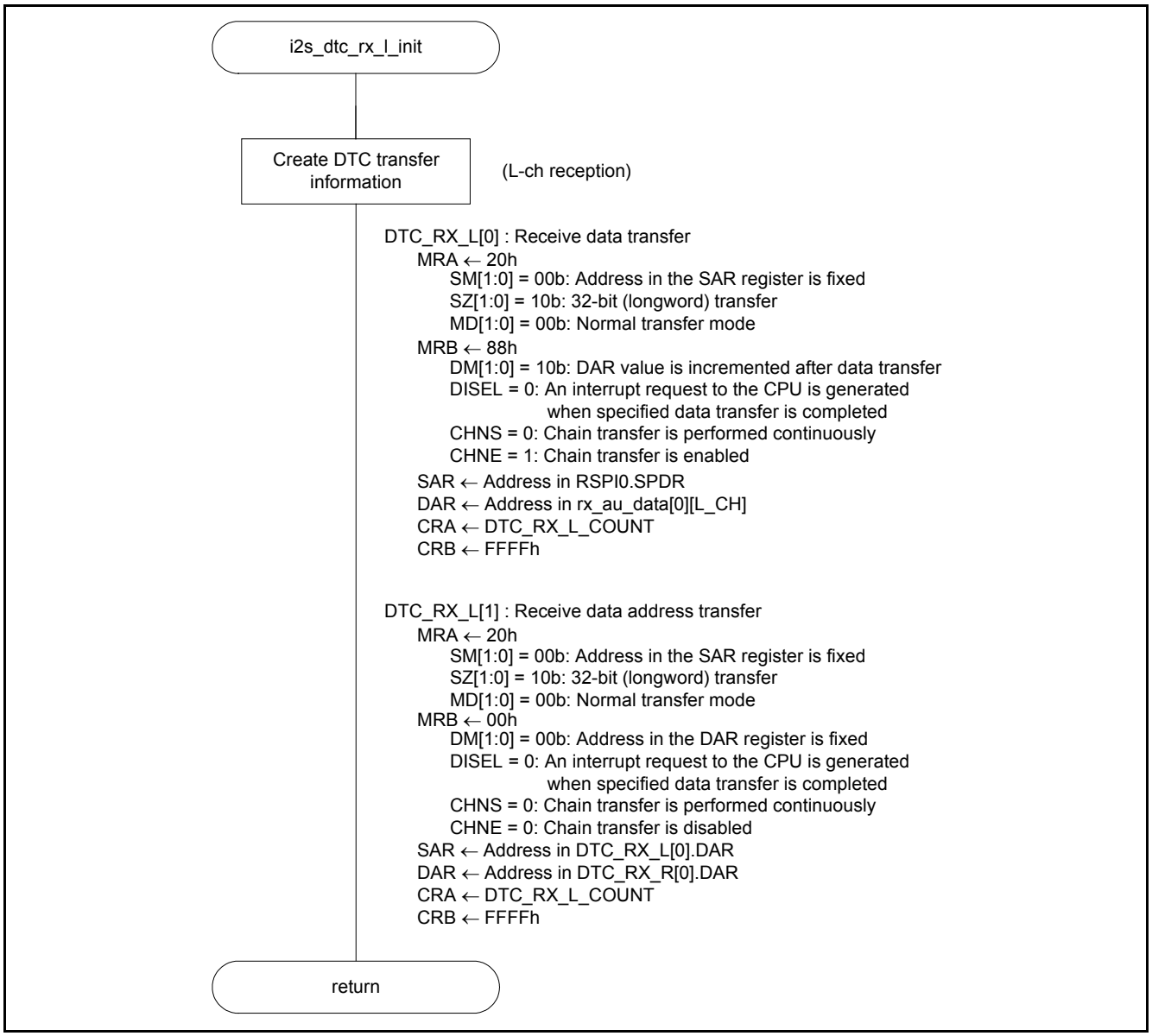

Figure 6.13 Creating DTC Transfer Information for L-ch Reception

6.2.7 i2s_dtc_rx_r_init

Figure 6.14 shows the Creating DTC Transfer Information for R-ch Reception.

The i2s_dtc_rx_r_init function creates DTC transfer information for R-ch reception.

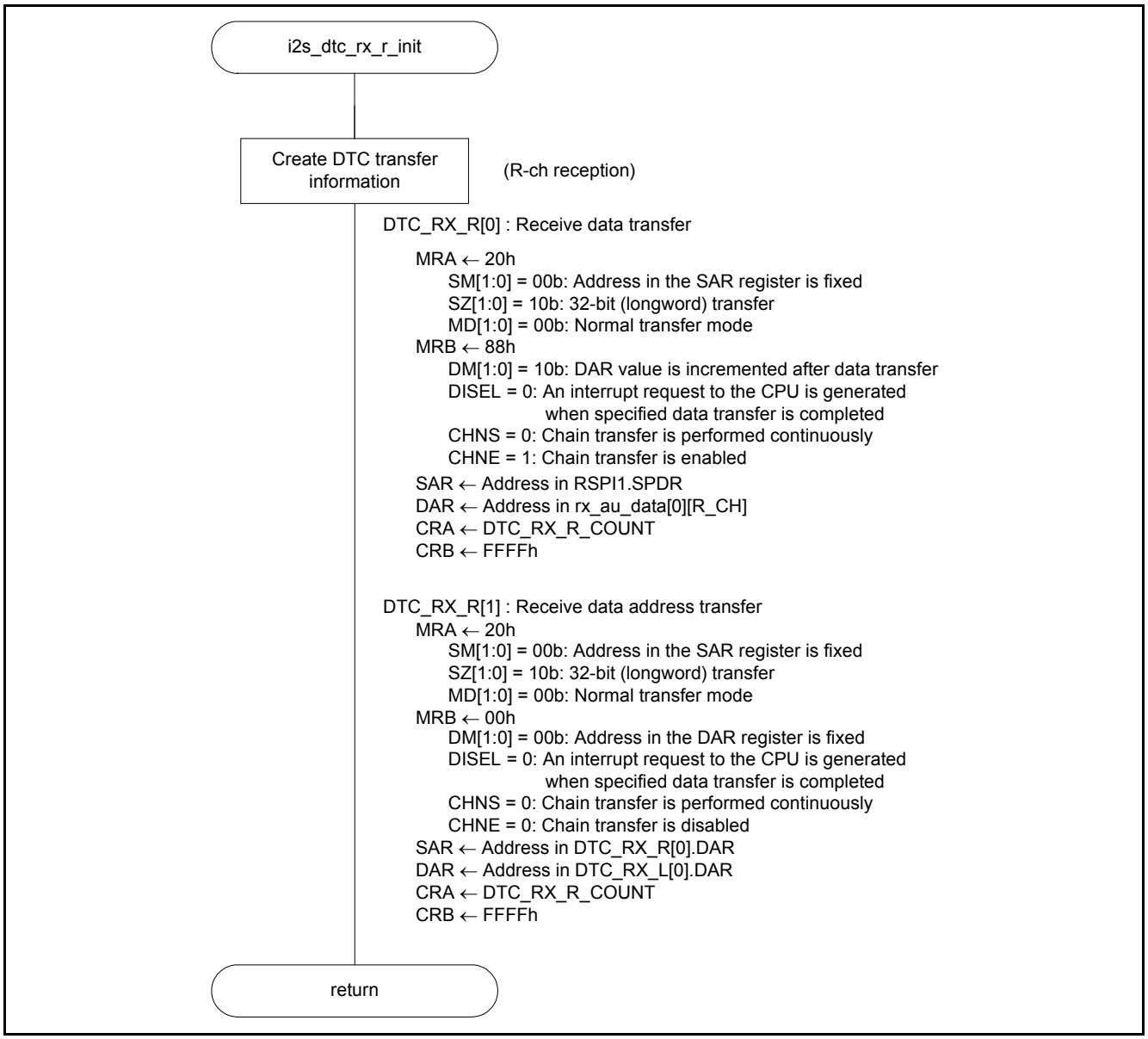

Figure 6.14 Creating DTC Transfer Information for R-ch Reception

6.2.8 i2s_mtu2_init

Figure 6.15 shows the MTU2a Initialization.

The i2s_mtu2_init function initializes the MTU2a.

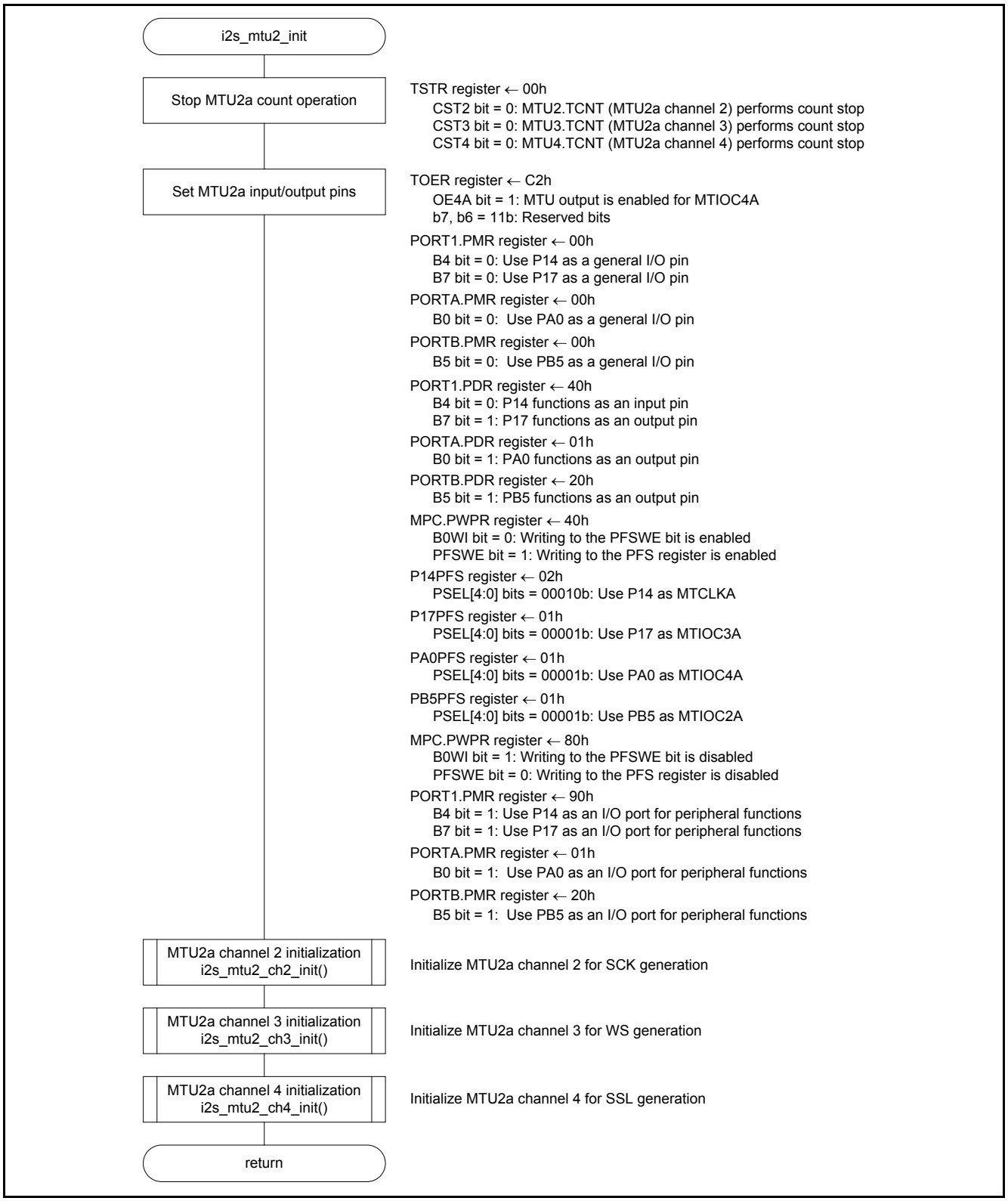

Figure 6.15 MTU2a Initialization

6.2.9 i2s_mtu2_ch2_init

Figure 6.16 shows the MTU2a Channel 2 Initialization for SCK Generation.

The i2s_mtu2_ch2_init function configures the setting to generate SCK using MTU2a channel 2.

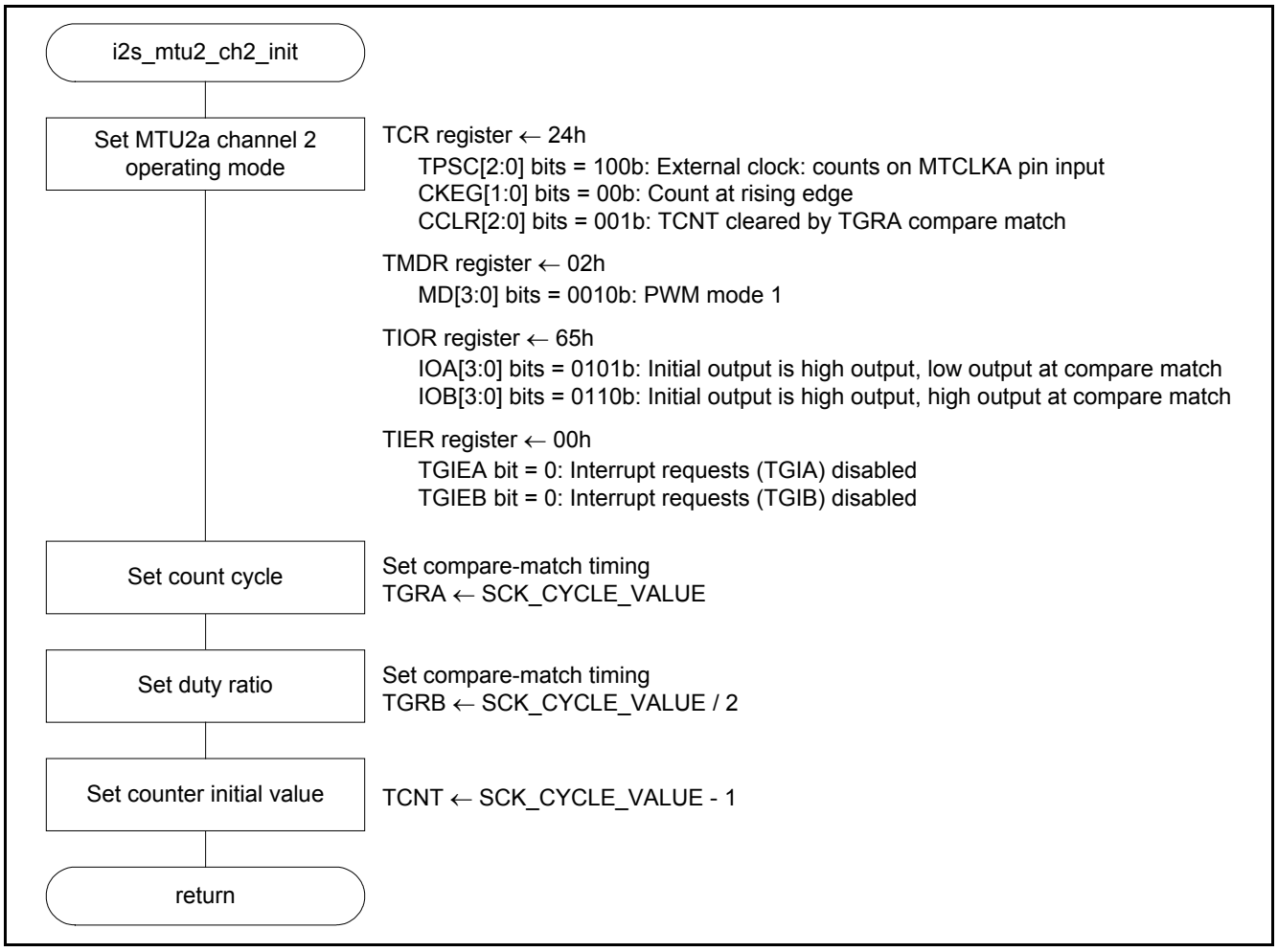

Figure 6.16 MTU2a Channel 2 Initialization for SCK Generation

6.2.10 i2s_mtu2_ch3_init

Figure 6.17 shows the MTU2a Channel 3 Initialization for WS Generation.

The i2s_mtu2_ch3_init function configures the setting to generate WS using MTU2a channel 3.

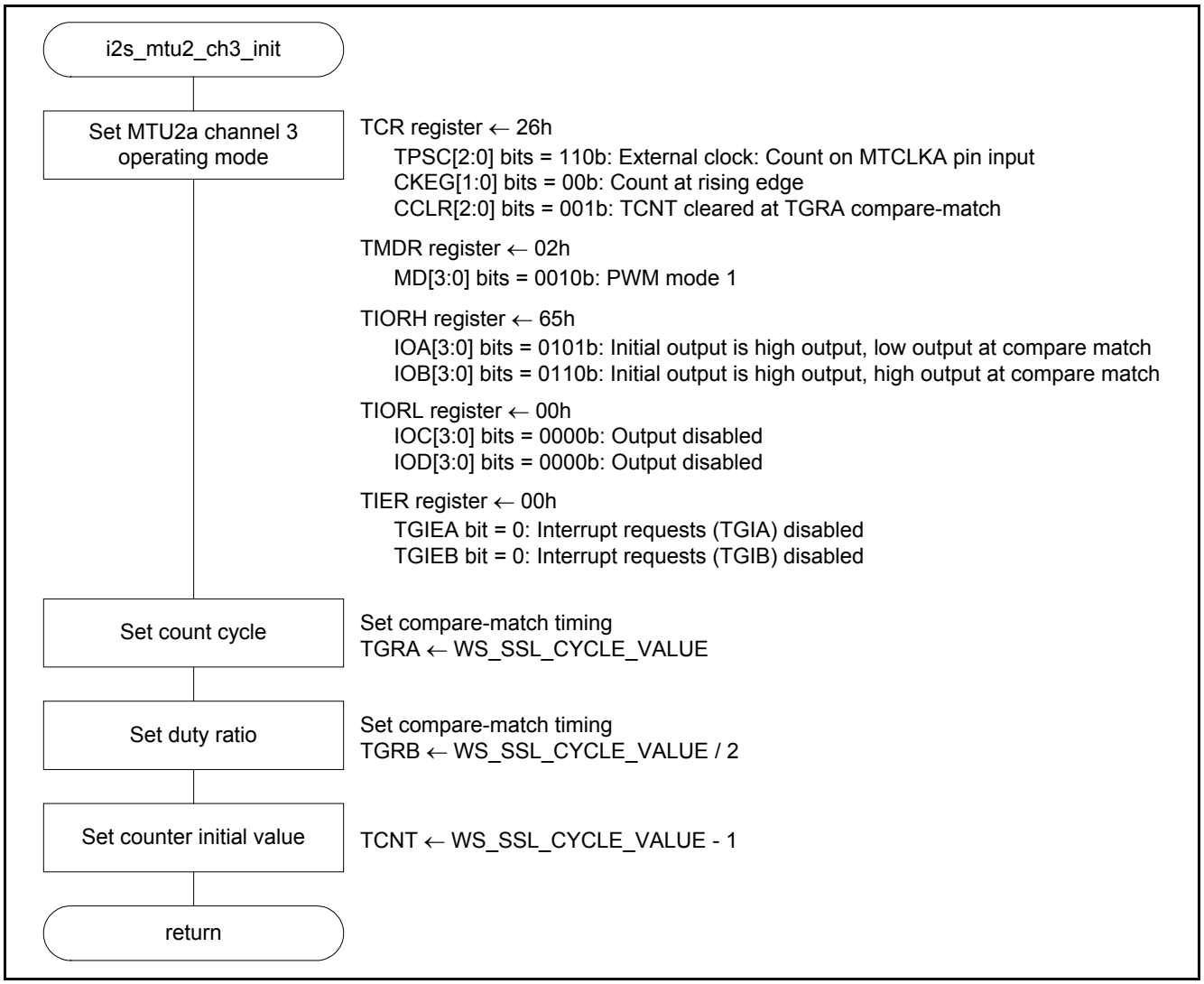

Figure 6.17 MTU2a Channel 3 Initialization for WS Generation

6.2.11 i2s_mtu2_ch4_init

Figure 6.18 shows the MTU2a Channel 4 Initialization for SSL Generation.

The i2s_mtu2_ch4_init function configures the setting to generate SSL using MTU2a channel 4.

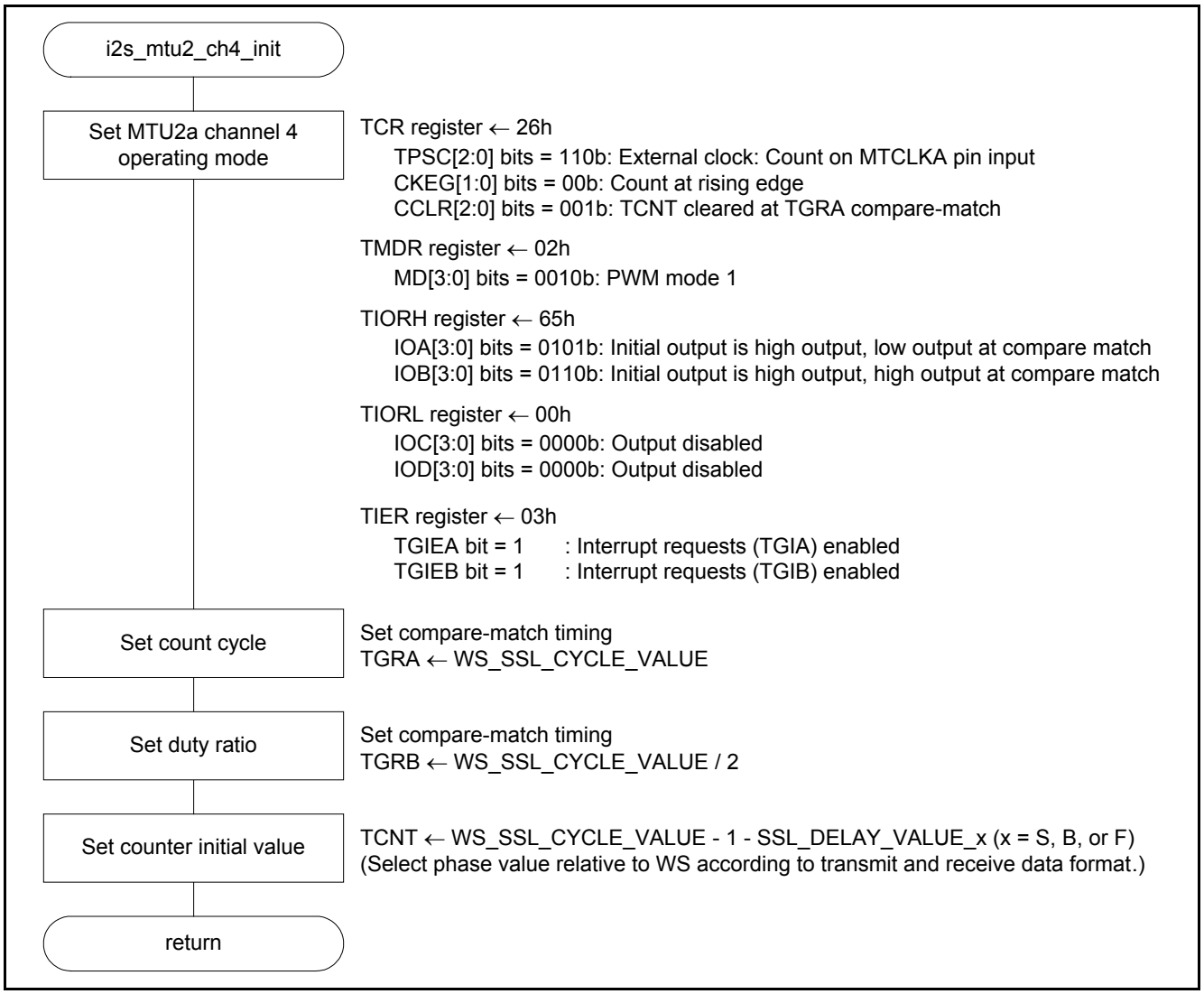

Figure 6.18 MTU2a Channel 4 Initialization for SSL Generation

6.2.12 i2s_rspi_init

Figure 6.19 shows the RSPI Initialization.

The i2s_rspi_init function initializes the RSPI.

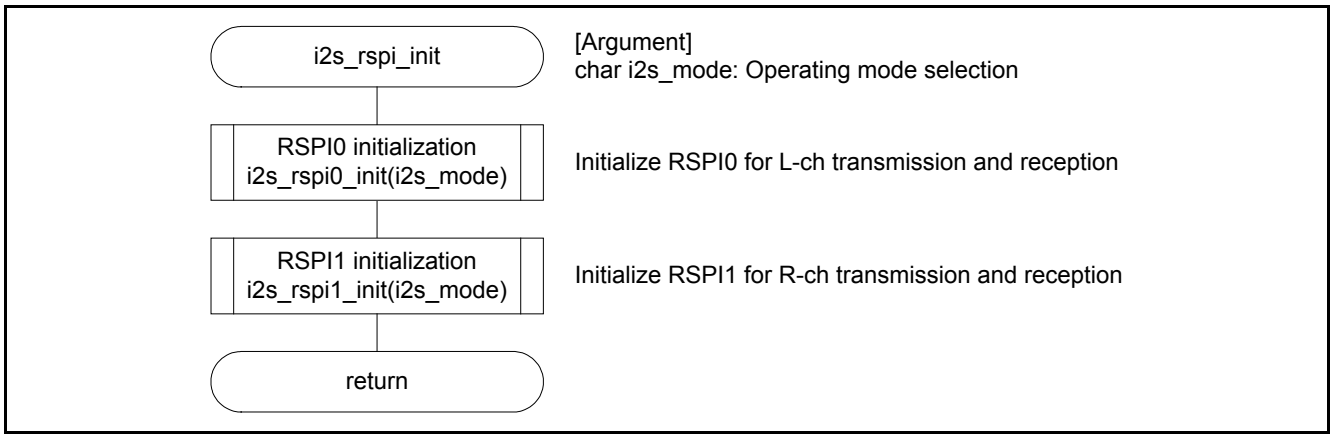

Figure 6.19 RSPI Initialization

6.2.13 i2s_rspi0_init

Figure 6.20 and Figure 6.21 show the RSPI0 Initialization for L-ch Transmission and Reception.

The i2s_rspi0_init function initializes the RSPI0.

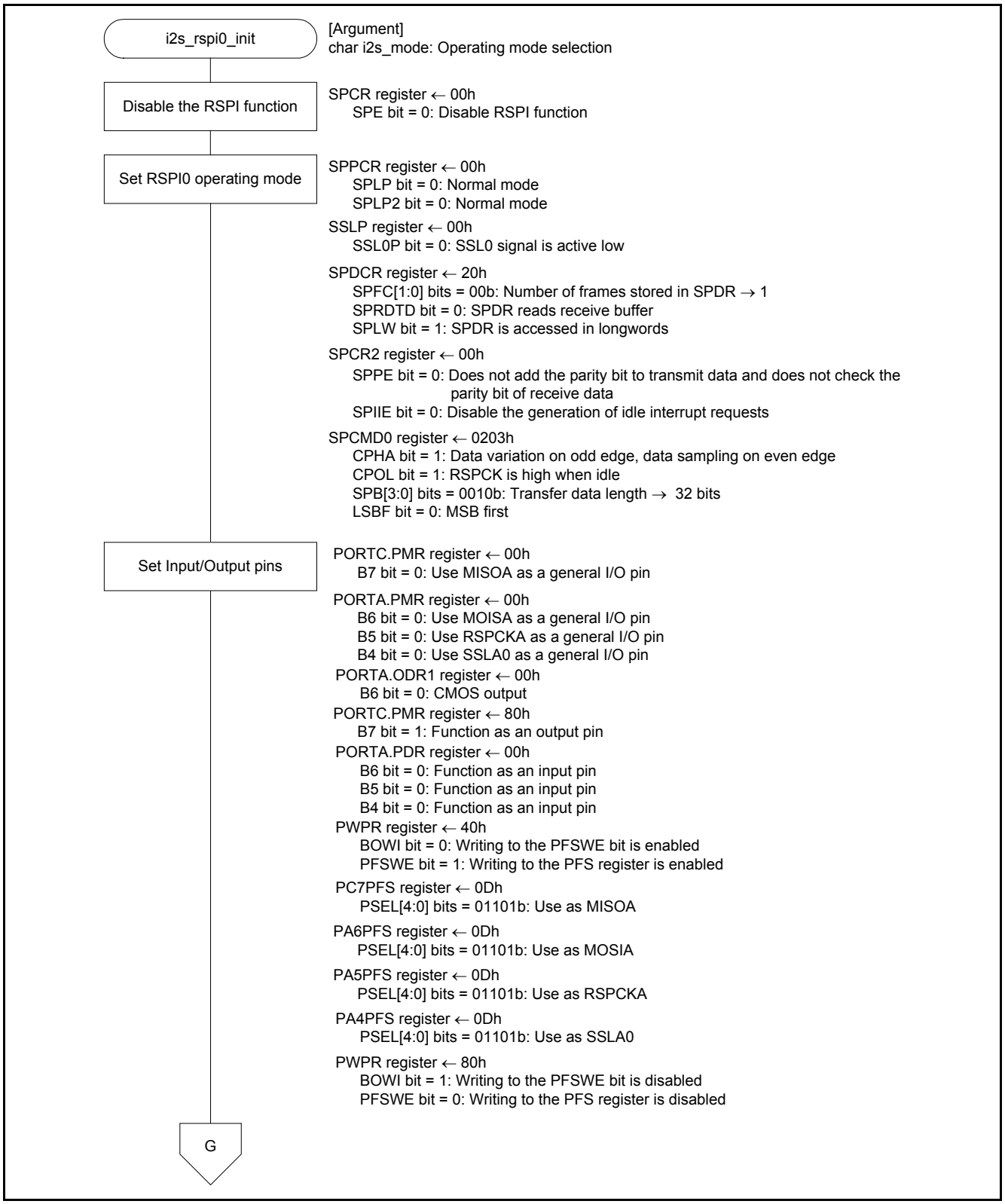

Figure 6.20 RSPI0 Initialization for L-ch Transmission and Reception (1/2)

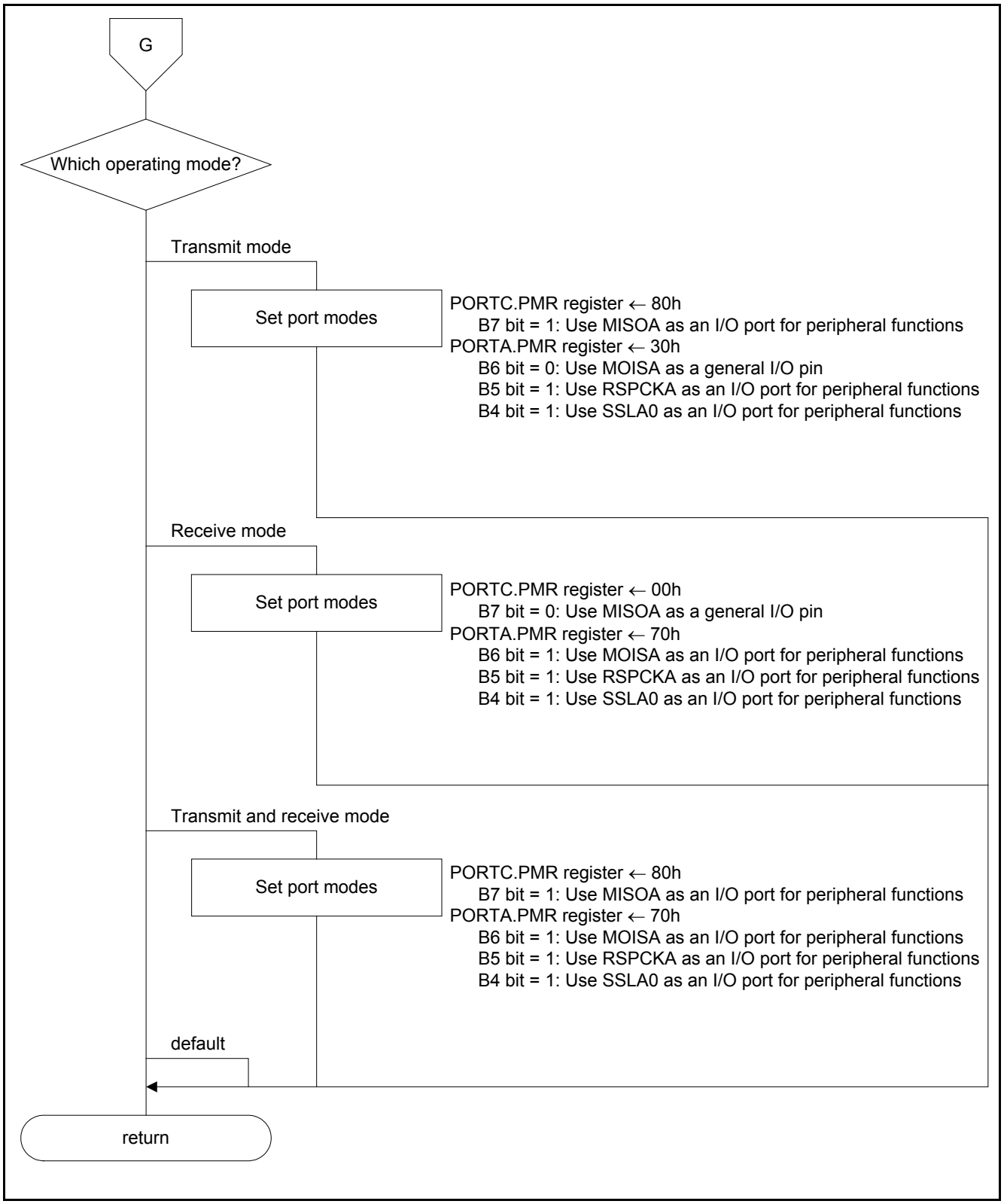

Figure 6.21 RSPI0 Initialization for L-ch Transmission and Reception (2/2)

6.2.14 i2s_rspi1_init

Figure 6.22 to Figure 6.23 show the RSPI1 Initialization for R-ch Transmission and Reception.

The i2s_rspi1_init function initializes the RSPI1.

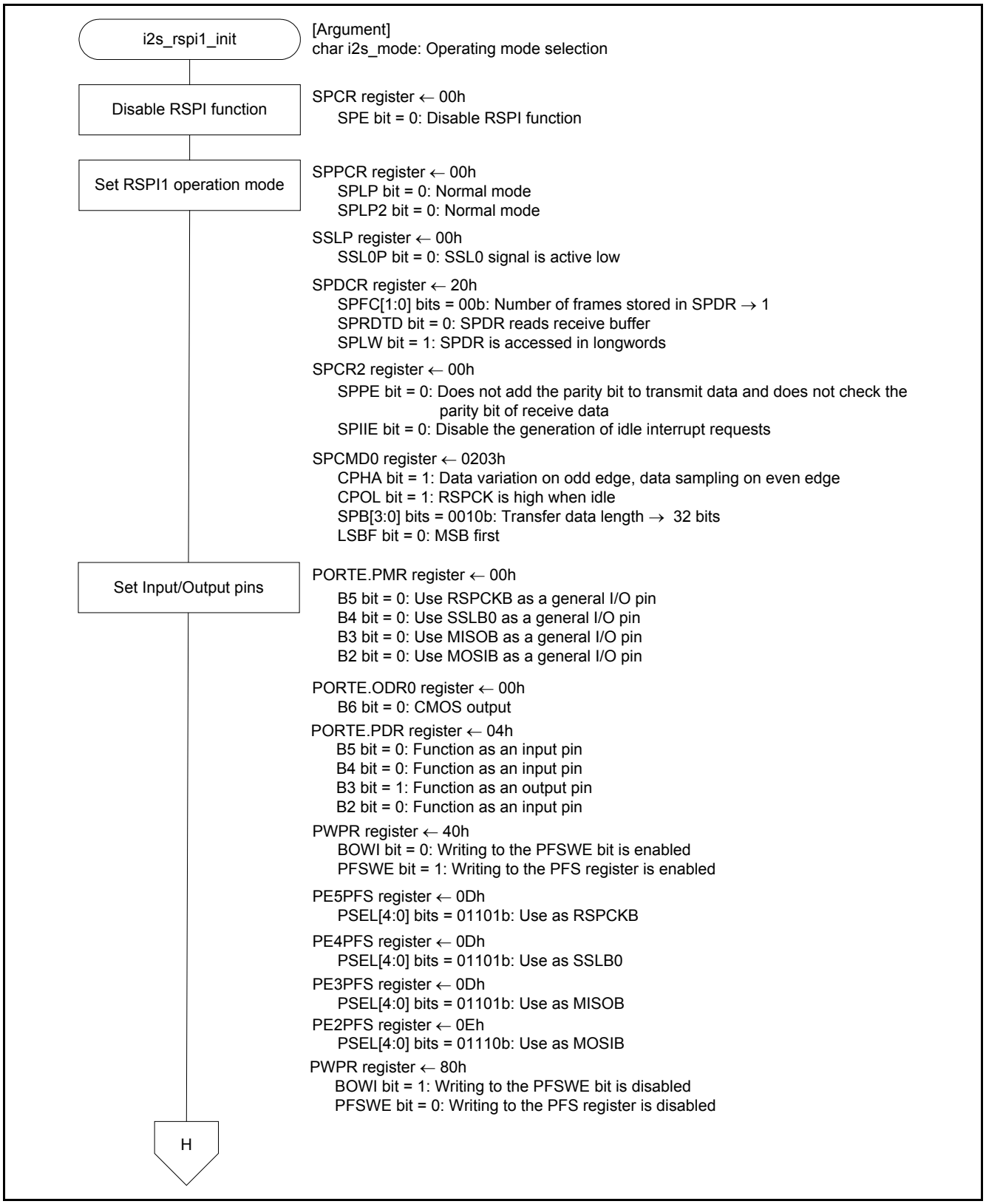

Figure 6.22 RSPI1 Initialization for R-ch Transmission and Reception (1/2)

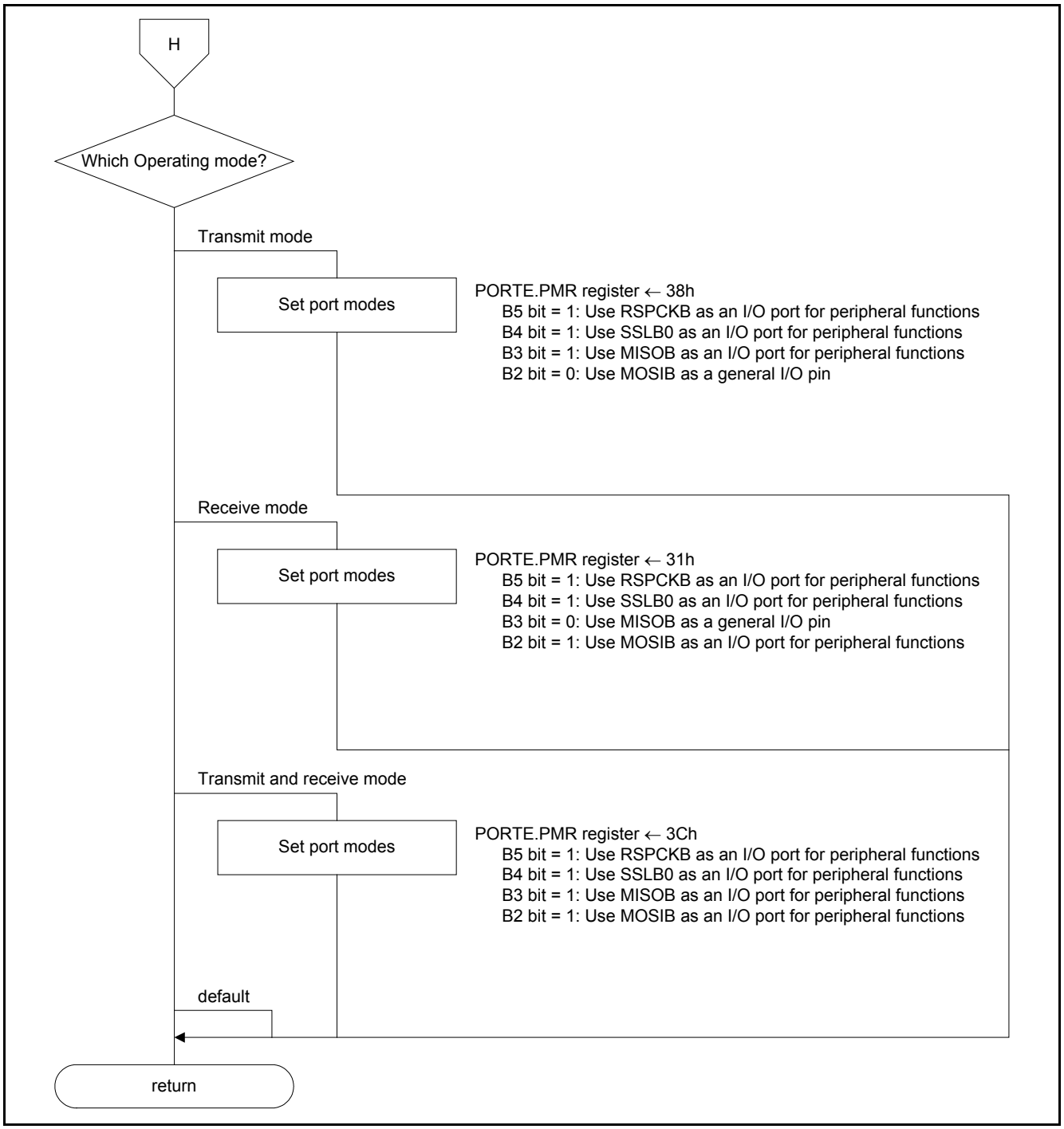

Figure 6.23 RSPI1 Initialization for R-ch Transmission and Reception (2/2)

7 Appendix

When the Renesas Starter Kit for RX630 is used with the sample code, change the following place on the board.

Figure 7.1 shows the change on the board.

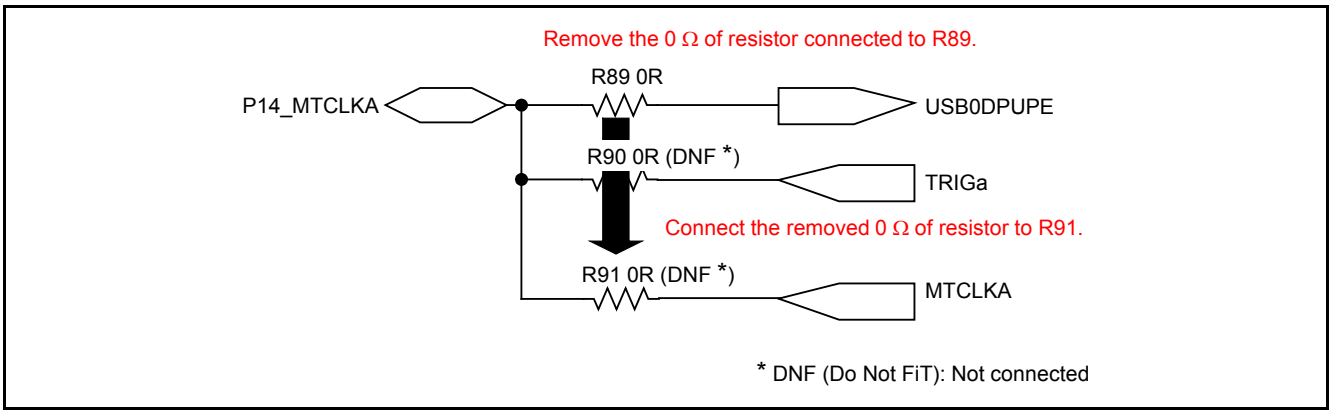

Figure 7.1 Change on the Board when Using P14 with MTCLKA

8 Sample Code

Sample code can be downloaded from the Renesas Electronics website.

9 Reference Documents

User's Manual: Hardware

RX630 Group User's Manual: Hardware Rev.1.50 (R01UH0040EJ)
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.

User's Manual: Development Tools

RX Family C/C++ Compiler Package V.1.01 User's Manual Rev.1.00 (R20UT0570EJ)
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/

REVISION HISTORY	RX630 Group Application Note I2S Communication Using RSPI, DTCa, and MTU2a

Rev.	Date	Description	
		Page	Summary
1.00	Aug. 1, 2013	—	First edition issued

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 document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

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 a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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.

RENESAS

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: +44-1628-651-700, Fax: +44-1628-651-804

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.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

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

Colophon 2.2