To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

H8S Family

Single-Master Mode Communications Using the I^{2}C Bus Interface 2 (IIC2)

Introduction

This application note describes the usage of the IIC2 module in single-master mode.

Target Device

H8S/2378

Contents

1. Specifications

- Figure 1 shows the connections for communications using I²C bus interface 2 in single master mode. The slave addresses and settings for the SAR_0 registers of the individual devices are listed in table 1.
- The single master system in this sample task consists of one master device and one slave device.
- The I²C bus transfer clock rate is 98.2 kbit/s (kHz).

The following describes the procedures for the operation of this sample task.

1. The I²C bus interface single master transfer starts on the input of the low trigger to the $\overline{\text{IRQ0}}$ pin of the master.
2. The master transmits four bytes of data, which have been prepared in the on-chip ROM in advance, to the on-chip RAM on the slave side.
3. The slave device returns the four bytes of data received in step 2 from its on-chip RAM to the on-chip RAM on the master side.
4. The master device performs processing for random reading of the EEPROM. This includes the issuance of a new start condition and reception of the single byte of data read out from the slave.
5. The master compares the received data in its on-chip RAM with the data transmitted from its on-chip ROM, and confirms whether the two match.
6. Based on the results of this comparison, the master outputs levels on the PE1 and PE0 pins that indicate the result of operations.

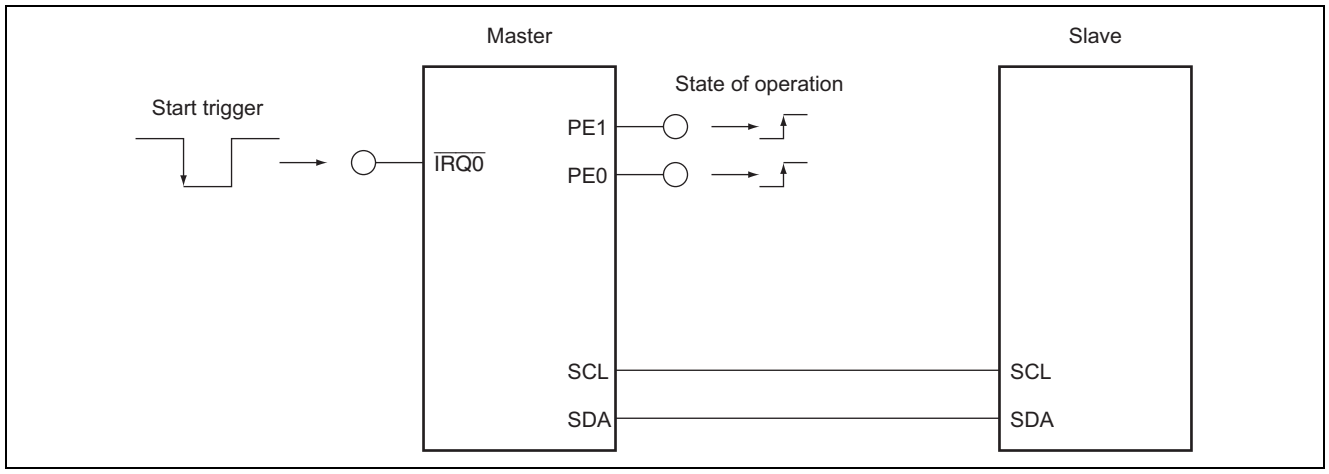


Figure 1 Connections for I²C Bus Interface 2 Single Master Mode Communication

Table 1 Slave Addresses

Device	Slave Address	SAR_0 Setting
Master	1	H'02
Slave	3	H'06

2. Applicable Conditions

Table 2 Applicable Conditions

Items	Description	
Operating frequency	Input clock	8.25 MHz
	System clock (ϕ)	33 MHz
	Peripheral module clock	33 MHz
Mode of operation	Mode 7 (MD2 = 1, MD1 = 1, MD0 = 1)	
Development tools	High-performance Embedded Workshop, ver. 4.02.00	
C/C++ compiler	H8S, H8/300 Series C/C++ Compiler ver. 6.01.02, manufactured by Renesas Technology Corp.	
Compiler options	-cpu = 2000a:24, -code = machinecode, -optimize = 1, -regparam = 3, -speed = (register, shift, struct, expression)	
Evaluation boards	Master:	HSB8S2378ST
	Slave:	HSB8S2378RE

Table 3 Section Settings

Address	Section	Description
H'001000	P	Program area
	C	Data table
H'FF6000	B	Non-initialized data area (RAM area)

3. Description of Functions

3.1 Description of I²C Bus Interface 2 (IIC2)

An I²C bus interface 2 (IIC2) is used in single master operation to demonstrate bidirectional communications in master mode.

3.2 Master Side $\overline{\text{IRQ0}}$ Pin

The trigger to start master transmission and master reception is input to the $\overline{\text{IRQ0}}$ pin on the master side. IRQ0 starts the processing of I²C bus interface communications on the input of a rising edge on the $\overline{\text{IRQ0}}$ pin.

The master judges whether or not the $\overline{\text{IRQ0}}$ pin has received the start trigger by polling the IRQ status flag. The IRQ interrupt is not used.

3.3 Master Side PE1 and PE0 Pins

As indicated in table 4, the pins PE1 and PE0 on the master side indicate the state of I²C bus interface communications (reset state or result of operations).

Table 4 Output Values of Master Side Pins and State of Operations

PE1	PE0	State of Operations
0	0	Reset
0	1	Data match
1	0	Data mismatch

4. Description of Operations

4.1 Timing of Operations in Master Transmit Mode

Figure 2 shows the timing of operations of the I²C bus interface 2 in master transmit mode. Table 5 describes processing by hardware and software at the numbered points in figure 2.

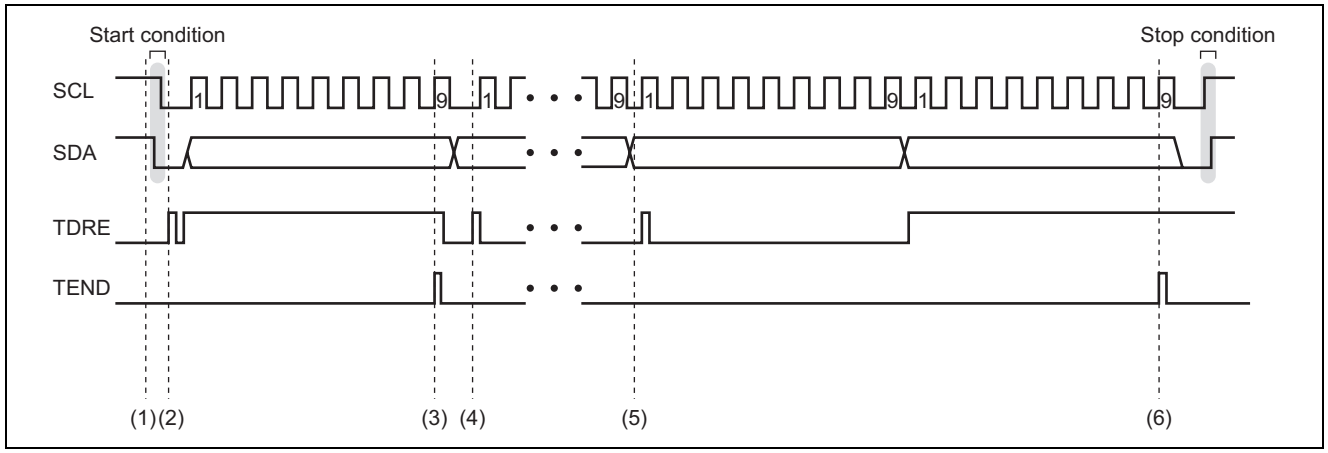


Figure 2 Timing of Operations in Master Transmit Mode

Table 5 Description of Processing

	Hardware Processing	Software Processing
(1)	No processing	a. Set the TIE bit to 1, enabling data empty interrupt. The interrupt will be generated when the TDRE bit is 1. b. Issue the start condition.
(2)	a. Transmit-data empty interrupt generation Start condition is detected and the TDRE bit is set to 1.	a. Write the slave-side address and data-direction bit (R/$\overline{W}$) to ICDRT, then transmit this data. Writing to ICDRT clears the TDRE and TEND flags. b. Clear the IRIC flag. c. Set the TIE bit to 1, disabling the data-empty interrupt.
(3)	a. Transmit-end interrupt generation On the rising edge of the ninth cycle of SCL, the TEND bit is set to 1.	a. Write the data for transmission to ICDRT and transmit the data. Writing to ICDRT clears the TDRE and TEND flags. b. Set the TIE bit to 1, enabling the transmit-end interrupt. c. Set the TEIE bit to 0, disabling the transmit-end interrupt.
(4)	a. Transmit-data empty interrupt generation Because data have been transmitted to ICDRS from ICDRT, and ICDRT has become empty, so the TDRE bit is set to 1.	a. Write the data for transmission to ICDRT and transmit the data. Writing to ICDRT clears the TDRE and TEND flags.

	Hardware Processing	Software Processing
(5)	a. Transmit-data empty interrupt generation Because data have been transmitted to ICDRS from ICDRT, and ICDRT has become empty, the TDRE bit is set to 1.	a. Write the last transmit data to ICDRT and transmit the data. Writing to ICDRT clears the TDRE and TEND flags. b. Set the TIE bit to 0, disabling the data-empty interrupt. c. Set the TEIE bit to 1, enabling the transmit-end interrupt. The interrupt will be generated when the TEND bit is 1.
(6)	a. Transmit-end interrupt generation On the rising edge of the ninth cycle of SCL, the TEND bit is set to 1.	a. Clear the TEND flag. b. Confirm that the SCL signal is at the low level and then issue the stop condition.

4.2 Master Receive Mode Operation Timing

Figures 3 and 4 show the timing of operations of the I²C bus interface 2 in master receive mode. Tables 6 and 7 describe processing by hardware and software at the numbered points in figures 3 and 4.

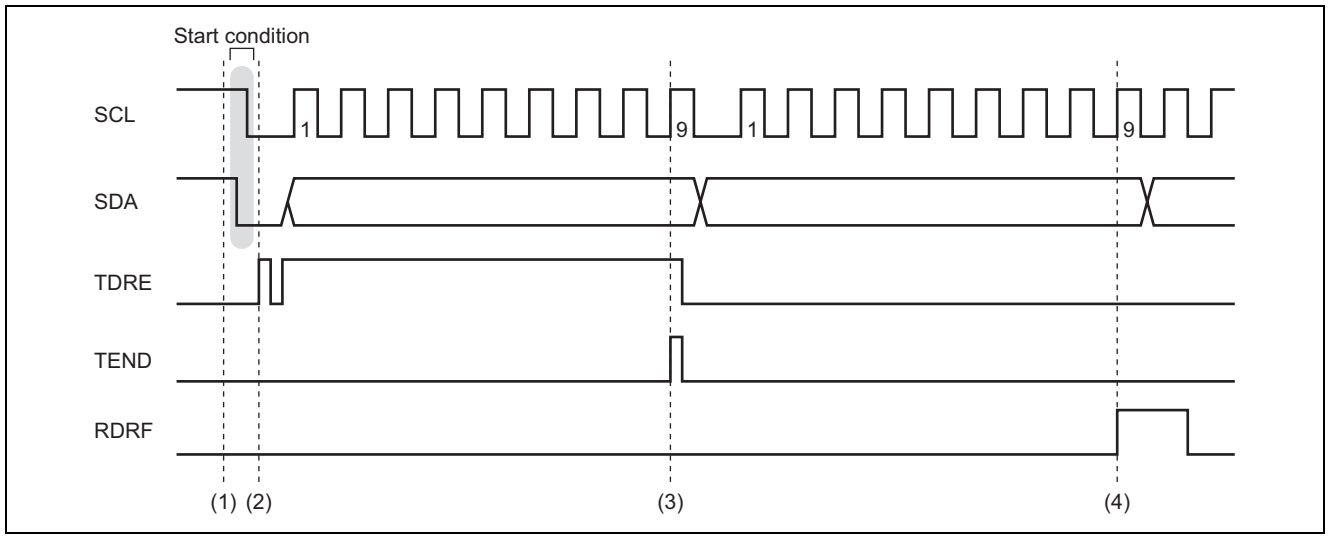


Figure 3 Master Receive Mode Operation Timing 1

Table 6 Description of Processing

	Hardware Processing	Software Processing
(1)	No processing	a. Set the TIE bit to 1, enabling data-empty interrupt. Set the TDRE bit to 1 for interrupt generation. b. Issue the start condition.
(2)	a. Transmit-data empty interrupt generation Start condition is detected and the TDRE bit is set to 1.	a. Write the slave-side address and data-direction bit (R/$\overline{W}$) to ICDRT, then transmit this data. Writing to ICDRT clears the TDRE and TEND flags. b. Set the TIE bit to 0, disabling the transmit-data empty interrupt. c. Set the TEIE bit to 1, enabling the transmit-end interrupt. The interrupt will be generated when the TEND bit is 1.
(3)	a. Transmit-end interrupt generation On the rising edge of the ninth cycle of SCL, the TEND bit is set to 1.	a. Set the TEIE bit to 0, disabling the transmit-end interrupt. b. Set the RIE bit to 1, enabling the receive-data full interrupt. c. Set the RCVD bit to 1, disabling the next round of reception. d. Clear the TEND flag. e. Set the TRS bit to 0: Selects receive mode. f. Clear the TDRE flag. g. Set the ACKBT bit to 0: Outputs 0 at the timing of acknowledge output in receive mode. h. Execute a dummy read of ICDRR; this clears RDRF.
(4)	a. Receive-data interrupt generation After reception of the first frame of data has been completed, the RDRF bit is set to 1 on the rising edge of the ninth cycle of SCL.	a. Read the first byte of receive data from ICDRR and save the data in RAM. Reading ICDRR clears RDRF.

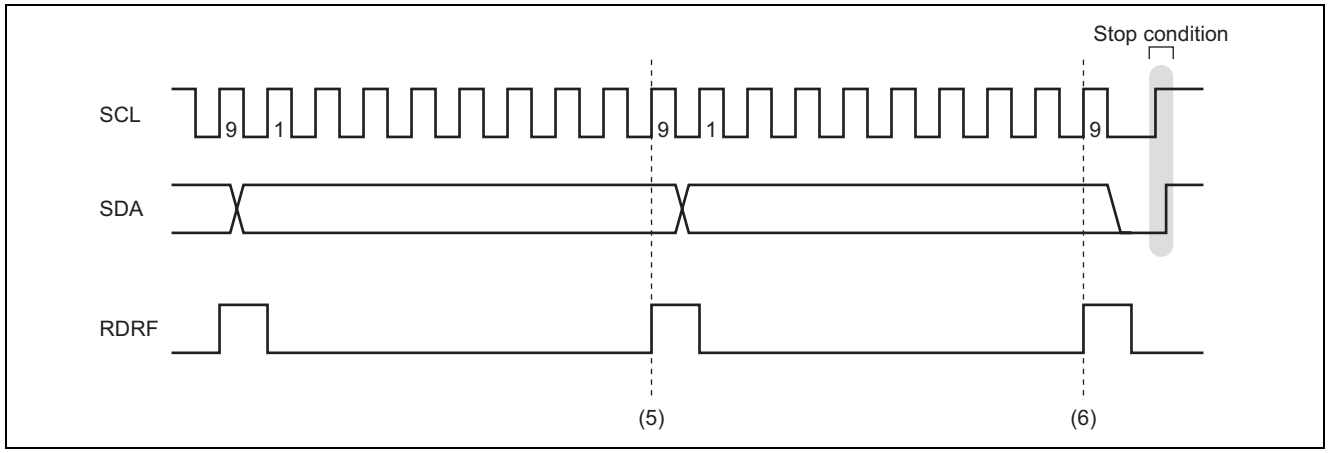


Figure 4 Master Receive Mode Operation Timing 2

Table 7 Description of Processing

	Hardware Processing	Software Processing
(5)	a. Receive-data interrupt generation After the reception of first frame data is completed, and on the rising edge of the ninth cycle of SCL, the RDRF bit is set to 1.	a. Set the ACKBT bit to 1: Outputs 1 at the timing of acknowledge output in receive mode. b. Read the receive data from ICDRR and save the data in RAM. Reading ICDRR clears RDRF.
(6)	a. Receive-data interrupt generation After the reception of first frame data is completed, and on the rising edge of the ninth cycle of SCL, the RDRF bit is set to 1.	a. Read the last receive data from ICDRR and save the data in RAM. Reading ICDRR clears RDRF. b. Set the RIE bit to 0, disabling the receive-data full interrupt. c. Confirm that the SCL signal is at the low level then issue the stop condition.

4.3 Random Read Waveform and Interrupt Timing

Figure 5 shows the timing of operations for the I²C bus interface 2 in master transmit mode. Table 8 describes processing by hardware and software at the numbered points in figure 5.

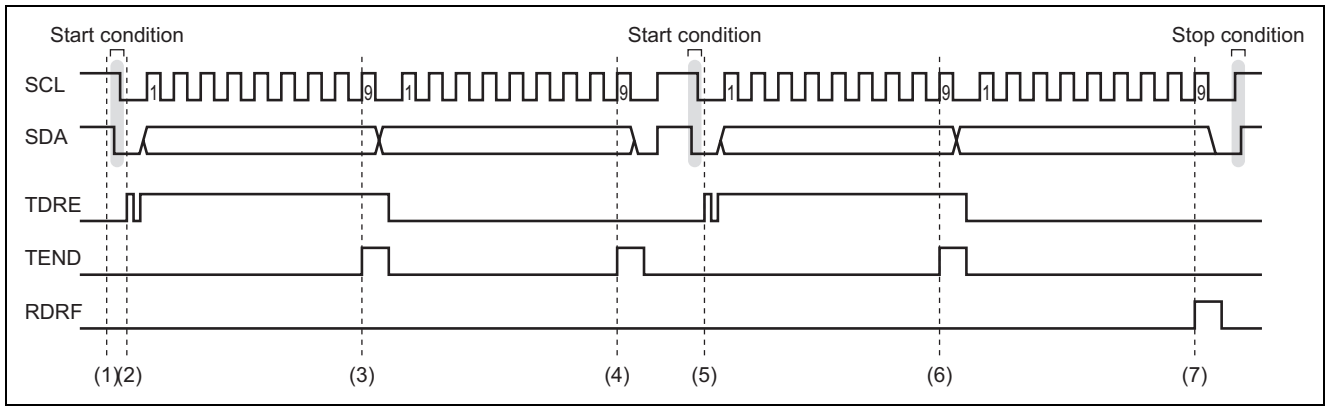


Figure 5 Timing of Operations in Master Transmit Mode

Table 8 Description of Processing

	Hardware Processing	Software Processing
(1)	No processing	a. Set the TIE bit to 1, enabling the data-empty interrupt. Set the TDRE bit to 1 for interrupt generation. b. Issue the start condition.
(2)	a. Transmit-data empty interrupt generation Start condition is detected and the TDRE bit is set to 1.	a. Write the slave-side address and data-direction bit (R/$\overline{W}$) to ICDR, then transmit this data. Writing to ICDRT clears the TDRE and TEND flags. b. Set the TIE bit to 0, disabling the transmit-data empty interrupt. c. Set the TEIE bit to 1, enabling the transmission end interrupt. The interrupt will be generated when the TEND bit is 1.
(3)	a. Transmit-end interrupt generation On the rising edge of the ninth cycle of SCL, the TEND bit is set to 1.	a. Write the memory address to ICDRT and transmit the data. Writing to ICDRT clears the TDRE and TEND flags.
(4)	a. Transmit-end interrupt generation On the rising edge of the ninth cycle of SCL, the TEND bit is set to 1.	a. Set the TIE bit to 1, enabling the data-empty interrupt. Set the TDRE bit to 1 to generate the interrupt. b. Set the TEIE bit to 0, disabling the transmit-end interrupt. c. Clear the TEND flag. d. Confirm that the SCL signal is at the low level and then issue the start condition.
(5)	a. Transmit-data empty interrupt generation Start condition is detected and the TDRE bit is set to 1.	a. Write the slave-side address and data-direction bit (R/$\overline{W}$) to ICDRT, then transmit this data. Writing to ICDRT clears the TDRE and TEND flags. b. Set the TIE bit to 0, disabling the transmit-data empty interrupt. c. Set the TEIE bit to 1, enabling the transmit-end interrupt. The interrupt will be generated when the TEND bit is 1.

	Hardware Processing	Software Processing
(6)	a. Transmit-end interrupt generation On the rising edge of the ninth cycle of SCL, the TEND bit is set to 1.	a. Set the TEIE bit to 0, disabling the transmit-end interrupt. b. Set the RIE bit to 1, enabling the receive-data full interrupt. c. Set the RCVD bit to 1, disabling next receive operation. d. Clear the TEND flag. e. Set the TRS bit to 0: Selects receive mode. f. Clear the TDRE flag. g. Set the ACKBT bit to 1: Outputs 1 at the timing of acknowledge output in receive mode. h. Execute a dummy read of ICDRR; this clears RDRF.
(7)	a. Receive-data interrupt generation After the reception of first frame data is completed, and on the rising edge of the ninth cycle of SCL, the RDRF bit is set to 1.	a. Read the final byte of received data from ICDRR and store the data in RAM. Reading ICDRR clears RDRF. b. Set the RIE bit to 0, disabling the receive-data full interrupt. c. Clear the TEND flag. d. Confirm that the SCL signal is at the low level and then issue the stop condition.

4.4 State Transition Diagram

Figure 6 is a state-transition diagram for this sample task. In this sample task, the idle mode is selected as the default.

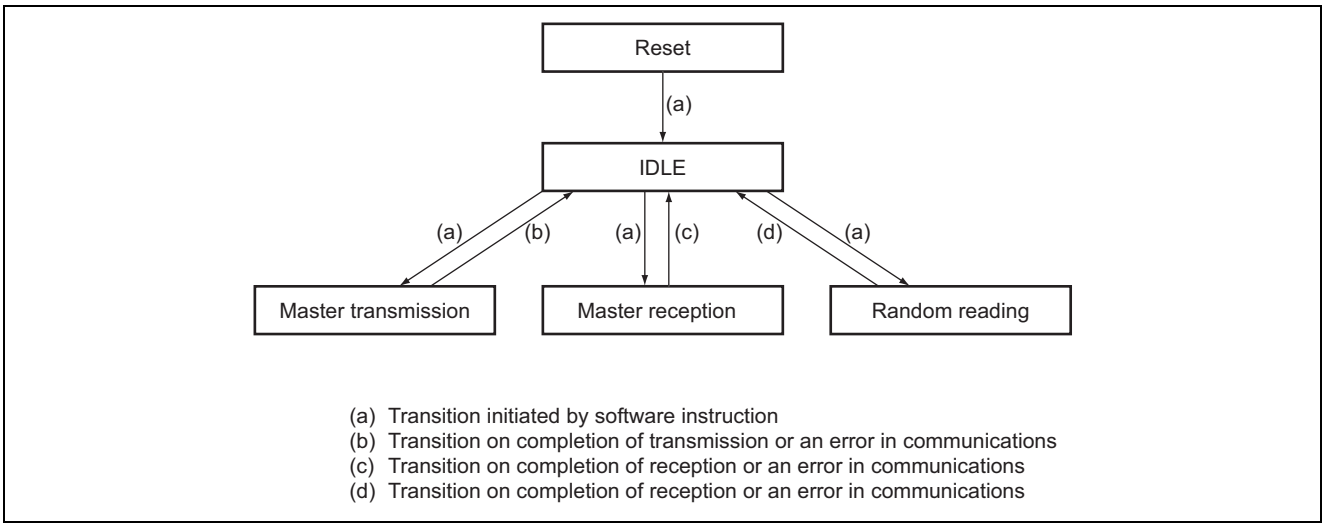


Figure 6 State Transition Diagram

5. Description of Software

5.1 List of Functions

Table 9 List of Functions: main.c

Function	Description
init	Initialization routine Sets the CCR and clock, releases IIC2 module 0 from module stop mode, and calls function "main".
main	Main routine Selects master mode operation, judges the state of the $\overline{\text{IRQ0}}$ pin, and handles master-transmission/reception processing.

Table 10 List of Functions: iic.c

Function	Description
iic_init	I²C bus interface initialization routine
mtrs_start	Sets I²C bus interface master transmission. Issues the start condition.
mrcv_start	Sets I²C bus interface master reception. Issues the start condition.
mrandrd_start	Sets random reading. Issues the start condition.
iici0_int	Handler for I²C bus interface interrupts. According to the state of operations, the functions for receiving the stop condition, master transmission, master reception, and random reading are called from this function.
receive_stop_condition	Detects the stop condition.
master_transfer	When the state of operation of this sample task is master transmission, this function for master transmission processing is called from the I²C bus interface interrupt handler. One byte of data is transferred per call of this function.
master_receive	When the state of operation of this sample task is master reception, this function for master reception processing is called from the I²C bus interface interrupt handler. One byte of data is received per call of this function.
master_randomread	When the state of operation of this sample task is random-reading, this function for random reading is called from the I²C bus interface interrupt. The function transmits the specified address and receives the byte of data from that address.

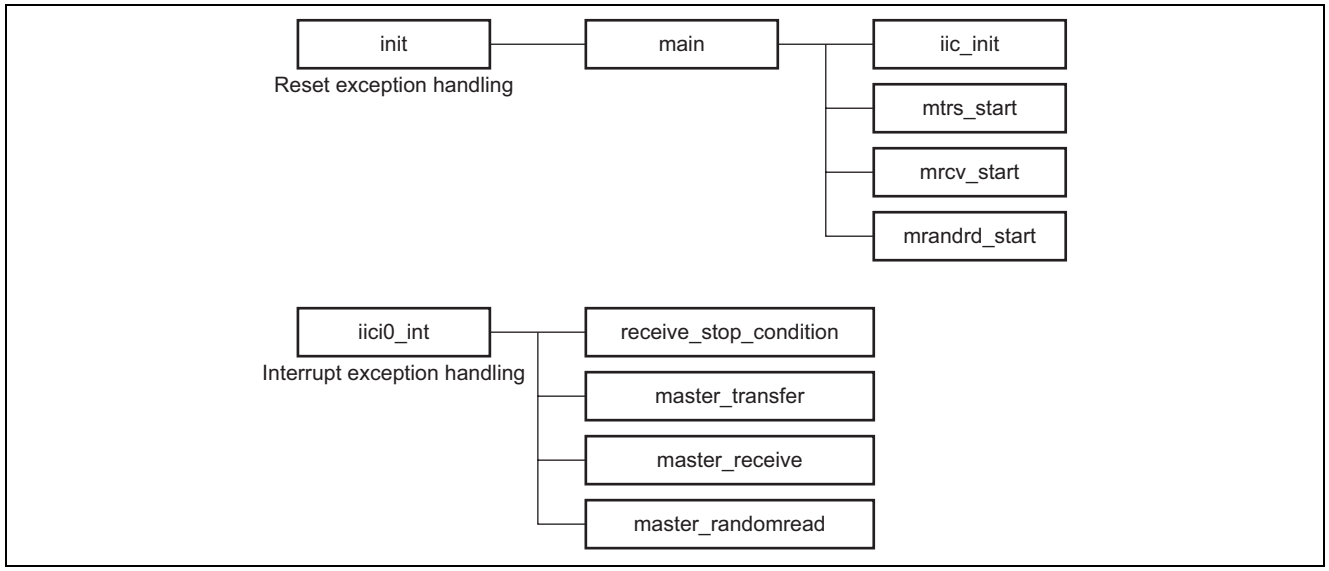


Figure 7 Hierarchy of Calls in the User Program

5.2 Vector Table

Table 11 Exception-Handling Vector Table

Origin of Exception	Vector Number	Vector Table Address	Target Function of the Vector
Task "Reset"	0	H'000000	main
IICI0 interrupt	116	H'0001D0	iici0_int

5.3 RAM Usage

Table 12 Description of RAM Usage

Type	Name of Variable	Description	Usage in Functions
unsigned char	iic_mode	Sets state of processing by this sample task.	main iic_init mtrs_start mrcv_start mrandrd_start iici0_int receive_stop_condition
unsigned short	mt_cnt	Counter used for master transmission	iic_init mtcv_start master_transfer
unsigned short	mr_cnt	Counter used for master reception	iic_init mrcv_start receive_stop_condition master_receive
unsigned char	randrd_cnt	Counter used for random reading	iic_init mrandrd_start master_randomread
unsigned short	mt_num	Number of bytes for master transmission	mtrs_start master_transfer
unsigned short	mr_num	Number of bytes for master reception	mrcv_start receive_stop_condition master_receive
unsigned char	*mt_data	Pointer to data for transmission	mtrs_start mrandrd_start master_transfer master_randomread
unsigned char	*mr_data	Pointer to received data	mrcv_start mrandrd_start receive_stop_condition master_receive
unsigned char	MemAddress	Memory area used to set the address of the data for random reading from the EEPROM area	mrandrd_start
unsigned char	MRcv_dt[4]	Master-side reception area	main

5.4 Constants

Table 13 Constants

Type	Name of Variable	Setting	Description	Usage in Function
unsigned char	MTrs_dt[4]	H'81, H'01, H'02, H'03	Data for master transmission	main

5.5 Macro Constants

Table 14 Macro Constants

Name of Variable	Setting	Description	Function Used
DTNUM	4	Number of data for transmission/reception	main
SLAVE_ADDR	H'02	Slave address	iic_init
MT_ID	H'06	Slave address + R/$\overline{W}$ bit for master transmission Slave-side slave address + 0 (transmission to the slave)	master_transfer master_randomread
MR_ID	H'07	Slave address + R/$\overline{W}$ bit for master reception Slave-side slave address + 1 (reception from the slave)	master_receive master_randomread
MODE_MR_RAND	5	State of processing of this sample task: Random-read mode	mrandrd_start iici0_int
MODE_MT	4	State of processing of this sample task: Master transmission	mtrs_start iici0_int
MODE_MR	3	State of processing of this sample task: Master reception	mrcv_start iici0_int receive_stop_condition
MODE_IDLE	0	State of processing of this sample task: Idle	main iic_init mtrs_start mrcv_start randrd_start receive_stop_condition

5.6 Functions of File main.c

5.6.1 Function init

1. Overview
   This initialization routine releases I²C bus interface 2, 0 from module-stop mode, sets the clock, and calls function "main".
2. Arguments
   None
3. Return value
   None
4. Internal registers used
   The following describes internal registers used in this sample task. The settings are those used in this sample task rather than the initial settings.

- System clock control register (SCKCR) Address: H'FFFF3B

Bit	Bit Name	Setting	R/W	Function
3	STCS	1	R/W	Frequency Multiplication Factor Switching Mode Select Selects the operation when the PLL circuit frequency multiplication factor is changed. 0: Specified multiplication factor is valid after transition to software standby mode. 1: Specified multiplication factor is valid immediately after STC1 and STC0 bits are rewritten.
2	SCK2	0	R/W	System Clock Select 2 to 0
1	SCK1	0	R/W	Select the frequency division ratio for the clock.
0	SCK0	0	R/W	000: 1/1

- Mode control register (MDCR) Address: H'FFFF3E

Bit	Bit Name	Setting	R/W	Function
2	MDS2	—*	R	Mode Select 2 to 0
1	MDS1	—*	R	Indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS to MDS0 correspond to pins MD2 to MD0. MDS2 to MDS0 are read-only bits, and cannot be modified. The levels being input on the mode pins (MD2 to MD0) are latched into these bits when MDCR is read. The latching is released by a power-on reset.
0	MDS0	—*	R	

Note: * Determined by the levels on pins MD2 to MD0.

MSTPCRH and MSTPCRL control the module stop mode. Setting a bit in these registers places the corresponding module in the module stop mode. Clearing a bit takes the module out of module stop mode.

- Module stop control register H (MSTPCRH) Address: H'FFFF40

Bit	Bit Name	Setting	R/W	Function
15	ACSE	0	R/W	All Module Clocks Stop Mode Enable Enables or disables the all module clocks stop mode, in which, when the CPU executes a SLEEP instruction after module stop mode has been set for all the on-chip peripheral functions controlled by MSTPCR or the on-chip peripheral functions except the TMR. 0: Disables all-module-clocks-stop mode. 1: Enables all-module-clocks-stop mode.
14	MSTP14	1	R/W	EXDMA controller (EXDMAC)
13	MSTP13	1	R/W	DMA controller (DMAC)
12	MSTP12	1	R/W	Data transfer controller (DTC)
11	MSTP11	1	R/W	16-bit timer-pulse unit (TPU)
10	MSTP10	1	R/W	Programmable pulse generator (PPG)
9	MSTP9	1	R/W	D/A converter (channels 0 and 1)
8	MSTP8	1	R/W	D/A converter (channels 2 and 3)

- Module stop control register L (MSTPCRL) Address: H'FFFF41

Bit	Bit Name	Setting	R/W	Function
7	MSTP7	1	R/W	D/A converter (channels 4 and 5)
6	MSTP6	1	R/W	A/D converter
5	MSTP5	1	R/W	Serial communication interface_4 (SCI_4)
4	MSTP4	1	R/W	Serial communication interface_3 (SCI_3)
3	MSTP3	1	R/W	Serial communication interface_2 (SCI_2)
2	MSTP2	1	R/W	Serial communication interface_1 (SCI_1)
1	MSTP1	1	R/W	Serial communication interface_0 (SCI_0)
0	MSTP0	1	R/W	8-bit timer (TMR)

EXMSTPCR performs the all module clocks stop mode control with MSTPCR. When entering the all module clocks stop mode, set EXMSTPCR to H'FFFF. Otherwise, set EXMSTPCR to H'FFFD.

- Extension module stop control register H (EXMSTPCRH) Address: H'FFFF42

Bit	Bit Name	Setting	R/W	Function
15 to 12	—	0	R/W	Reserved Read/Write is enabled. 1 should be written in writing.

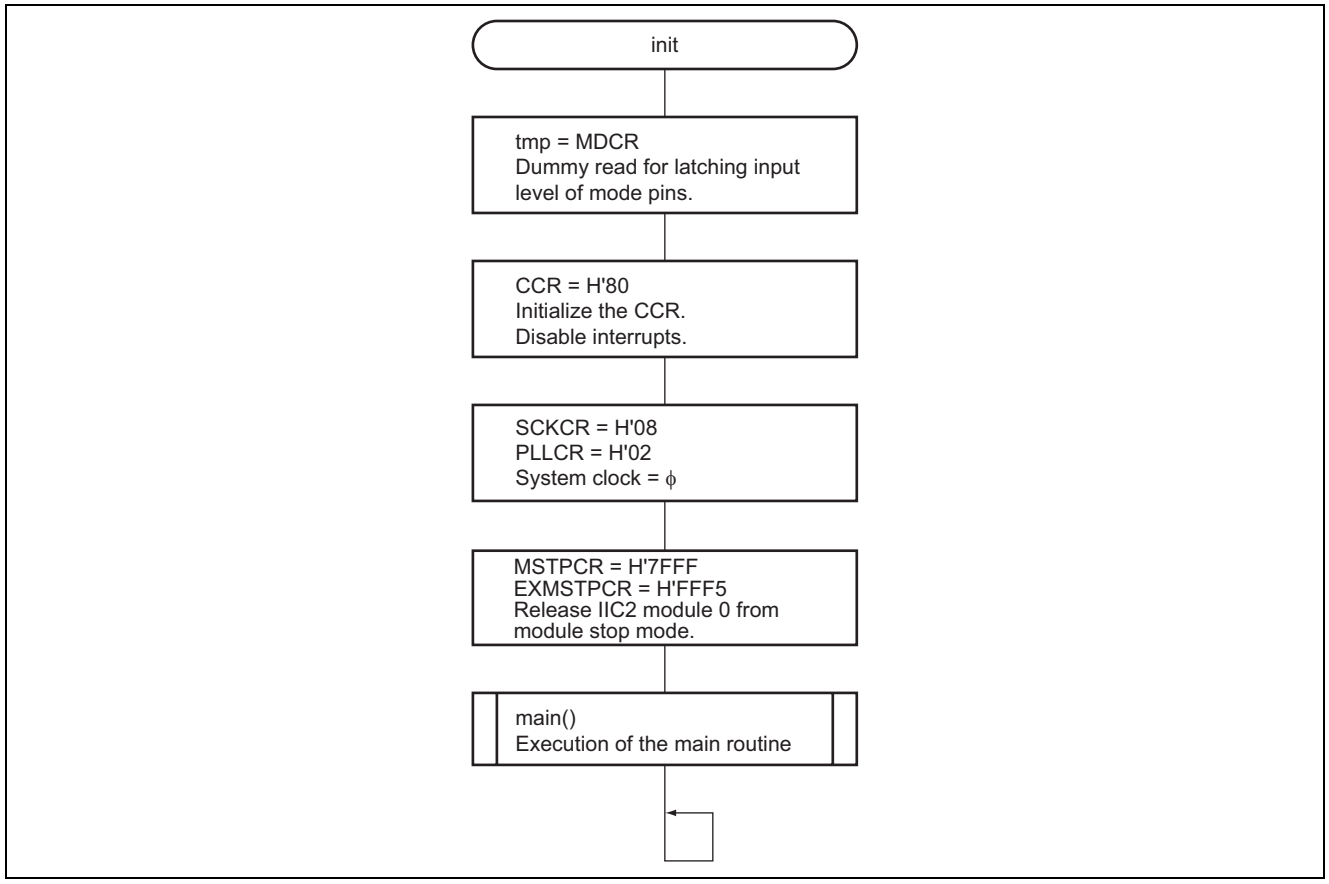
- Extension module stop control register L (EXMSTPCRL) Address: H'FFFF43

Bit	Bit Name	Setting	R/W	Function
4	MSTP20	1	R/W	I²C bus interface 2_1 (IIC2_1)
3	MSTP19	0	R/W	I²C bus interface 2_0 (IIC2_0)
1	MSTP17	0	R/W	—

- PLL control register (PLLCR) Address: H'FFFF45

Bit	Bit Name	Setting	R/W	Function
1	STC1	1	R/W	Frequency Multiplication Factor
0	STC0	0	R/W	Specify the frequency multiplication factor used by the PLL circuit. 10: × 4

5. Flowchart



5.6.2 Function main

1. Overview
   - On falling edges of the $\overline{\text{IRQ0}}$ signal, this function performs 4-byte master transmission, 4-byte master reception, and 1-byte random reading.
   - Compares the master transmit data with the master receive data, and outputs an indicator of the results of comparison to pins PE1 and PE0.

2. Arguments
   None

3. Return value
   None

4. Internal registers used
   The following describes internal registers used in this sample task. The settings are those used for this sample task, and are not initial settings.

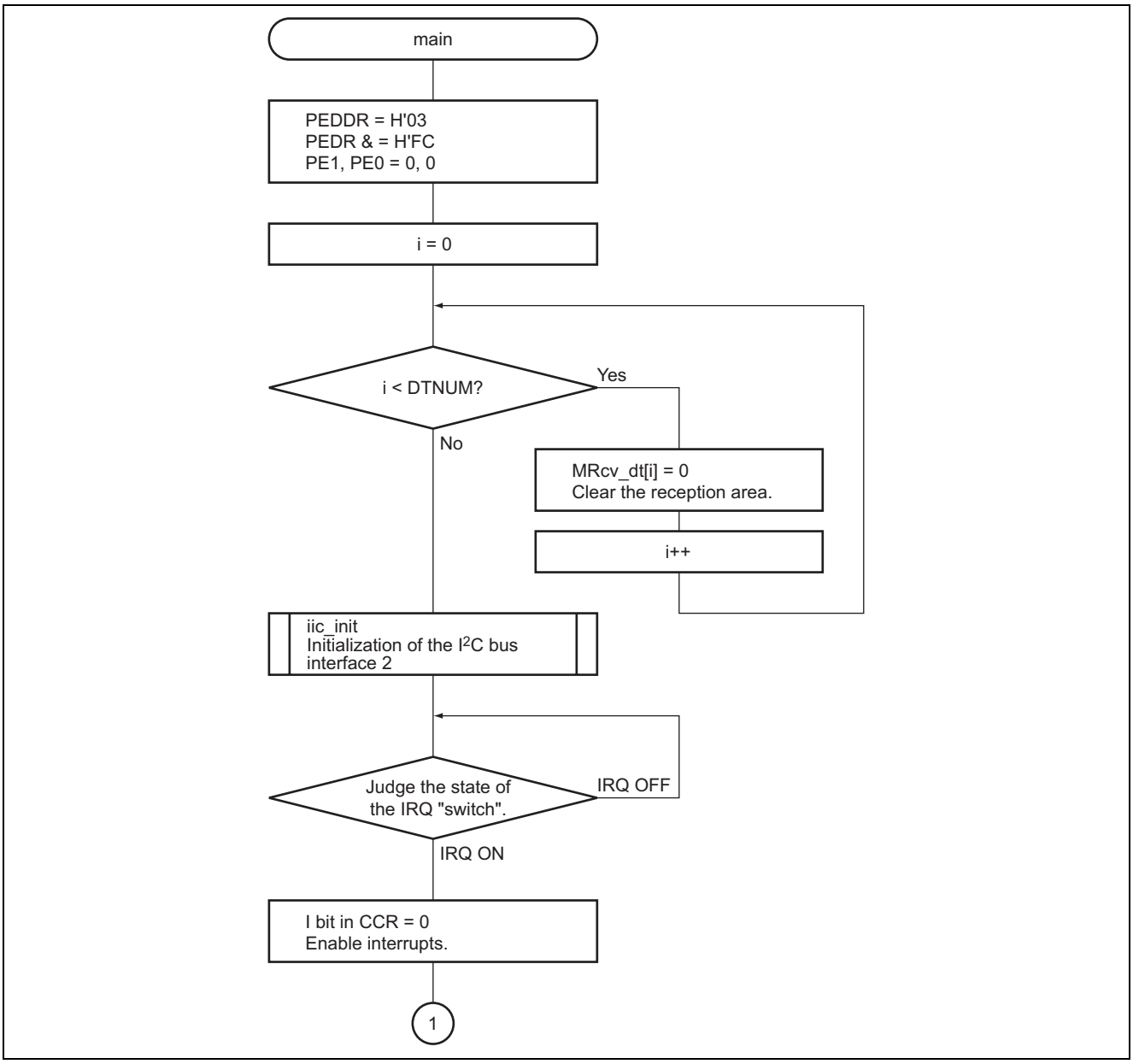
- Port E data direction register (PEDDR) Address: H'FFFE2D

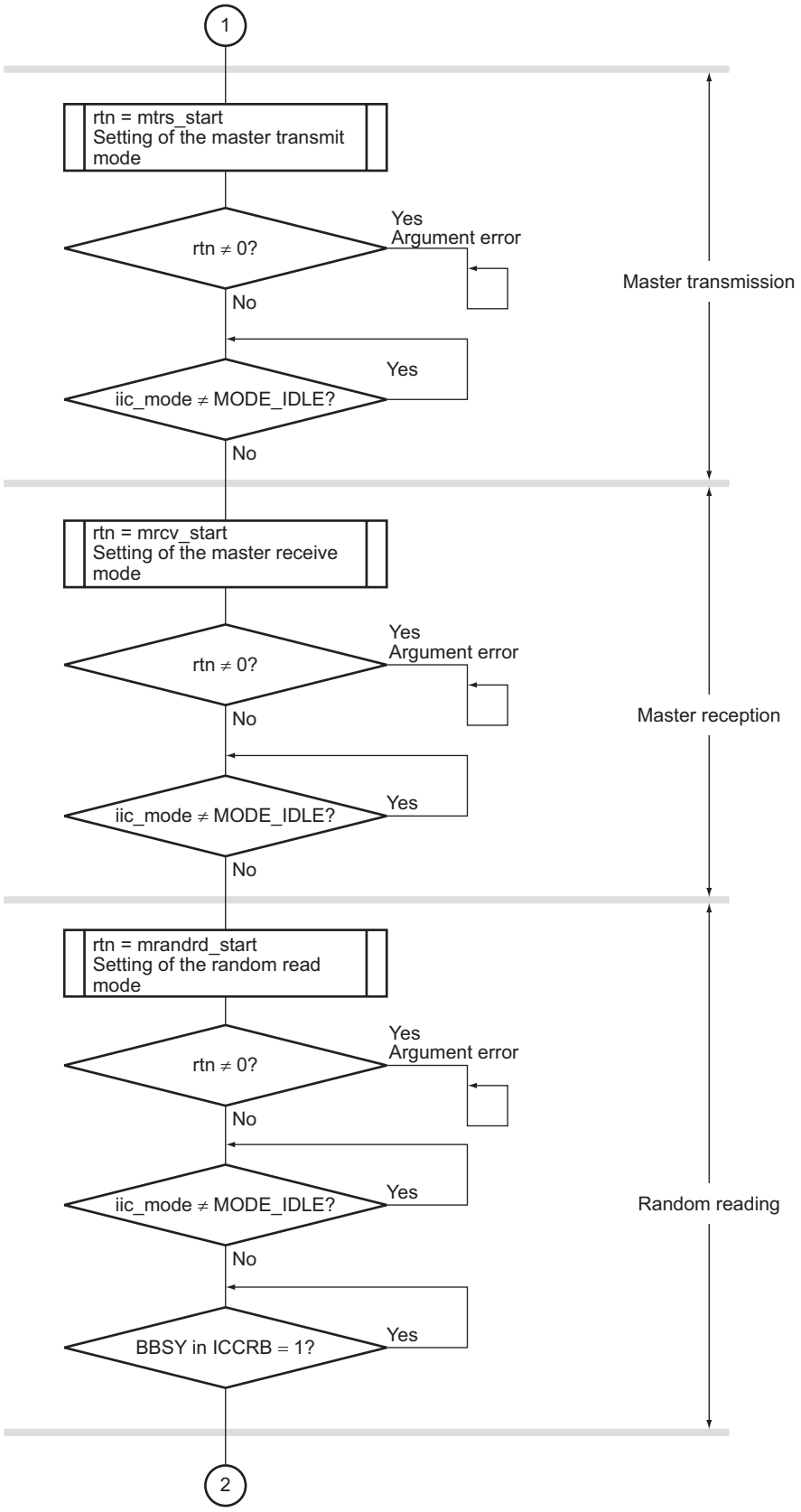
Bit	Bit Name	Setting	R/W	Function
1	PE1DDR	1	R/W	0: Sets the PE1 pin as an input pin. 1: Sets the PE1 pin as an output pin.
0	PE0DDR	1	R/W	0: Sets the PE0 pin as an input pin. 1: Sets the PE0 pin as an output pin.

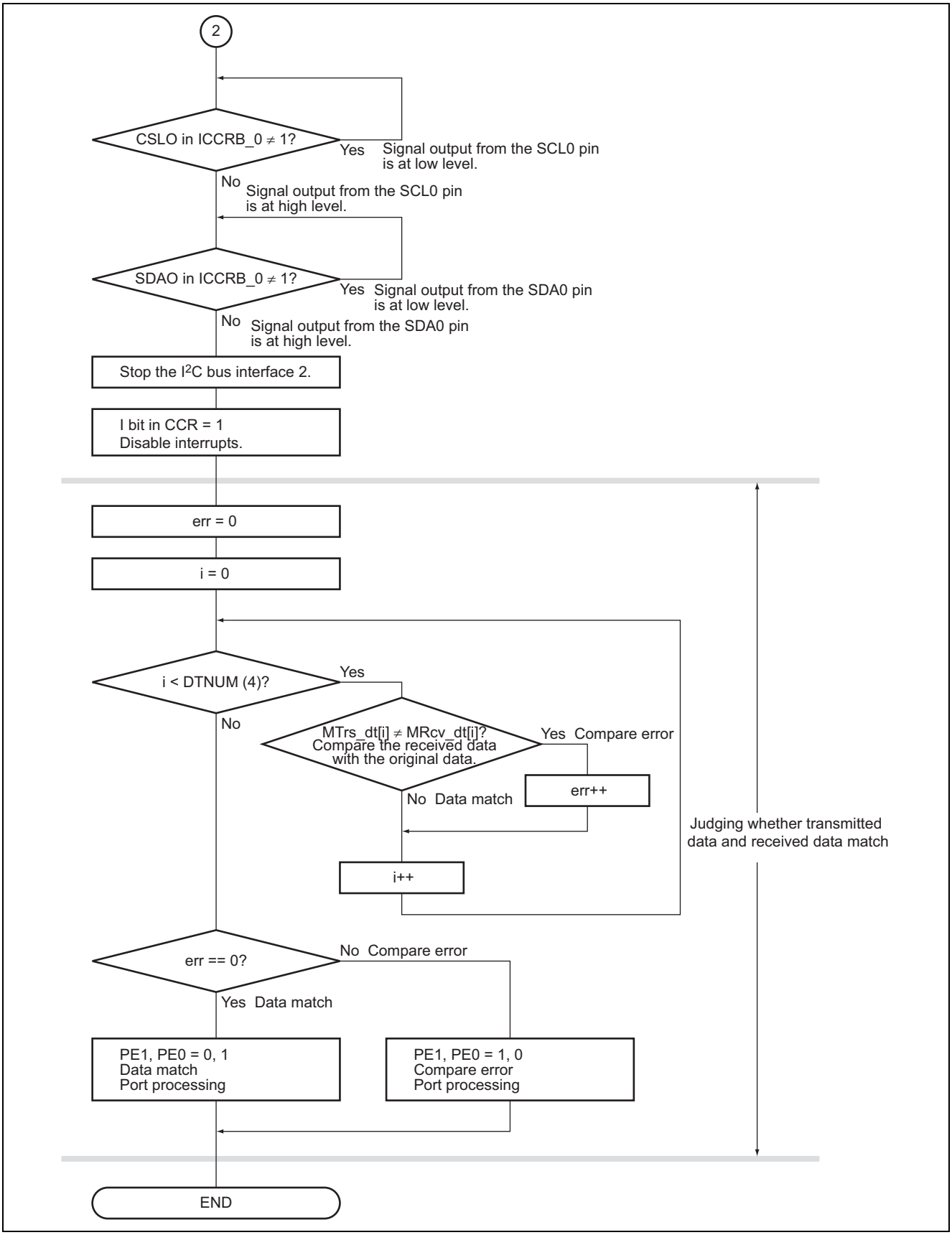
- Port E data register (PEDR) Address: H'FFFF6D

Bit	Bit Name	Setting	R/W	Function
1	PE1DR	0/1	R/W	0: Sets the PE1 pin to the low level. 1: Sets the PE1 pin to the high level.
0	PE0DR	0/1	R/W	0: Sets the PE0 pin to the low level. 1: Sets the PE0 pin to the high level.

5. Flowchart


main
PEDDR = H'03
PEDR & = H'FC
PE1, PE0 = 0, 0
i = 0
i < DTNUM?
Yes
No
MRcv_dt[i] = 0
Clear the reception area.
i++
iic_init
Initialization of the I²C bus interface 2
Judge the state of the IRQ "switch".
IRQ OFF
IRQ ON
I bit in CCR = 0
Enable interrupts.
1


1
rtn = mtrs_start
Setting of the master transmit mode
rtn ≠ 0?
Yes
Argument error
No
iic_mode ≠ MODE_IDLE?
Yes
No
Master transmission
rtn = mrcv_start
Setting of the master receive mode
rtn ≠ 0?
Yes
Argument error
No
iic_mode ≠ MODE_IDLE?
Yes
No
Master reception
rtn = mrandrd_start
Setting of the random read mode
rtn ≠ 0?
Yes
Argument error
No
iic_mode ≠ MODE_IDLE?
Yes
No
BBSY in ICCRB = 1?
Yes
Random reading
2


2
CSLO in ICCRB_0 ≠ 1?
Yes
Signal output from the SCL0 pin is at low level.
No
Signal output from the SCL0 pin is at high level.
SDAO in ICCRB_0 ≠ 1?
Yes
Signal output from the SDA0 pin is at low level.
No
Signal output from the SDA0 pin is at high level.
Stop the I²C bus interface 2.
I bit in CCR = 1
Disable interrupts.
err = 0
i = 0
i < DTNUM (4)?
Yes
No
MTrs_dt[i] ≠ MRcv_dt[i]?
Compare the received data with the original data.
Yes Compare error
err++
No Data match
i++
Judging whether transmitted data and received data match
err == 0?
No Compare error
Yes Data match
PE1, PE0 = 0, 1
Data match
Port processing
PE1, PE0 = 1, 0
Compare error
Port processing
END

5.7 Functions of File iic.c

5.7.1 Function iic_init

1. Overview
   I²C bus interface initialization routine
2. Arguments
   None
3. Return value
   None
4. Internal registers used
   The following describes internal registers used in this sample task. The settings are those used in this sample task rather than the initial settings.

- I²C bus control register A_0 (ICCRA_0) Address: H'FFFD58

Bit	Bit Name	Setting	R/W	Function
7	ICE	1	R/W	I²C Bus Interface Enable 0: Disables the IIC2 module. 1: Enables transfer via the IIC2 module (pins SCL/SDA are driving the bus).
6	RCVD	0	R/W	Reception Disable Enables or disables the next operation when ICDRR is read while the TRS bit is 0. 0: Enables the next operation. 1: Disables the next operation.
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	00: Slave receive mode
3	CKS3	1	R/W	Transfer Clock Select 3 to 0
2	CKS2	1	R/W	1100: Transfer rate is 98.2 kbps with ϕ = 33 MHz.
1	CKS1	0	R/W	
0	CKS0	0	R/W	

- I²C mode register_0 (ICMR_0) Address: H'FFFD5A

Bit	Bit Name	Setting	R/W	Function
6	WAIT	0	R/W	Wait Insertion Bit Selects whether to insert a wait after data transfer except for the acknowledge bit. When WAIT is set to 1, after the fall of the clock for the final data bit, low period is extended for two transfer clocks. When WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted. Note: When WAIT in the I²C mode register is set to 1, and SCL is driven low for two or more transfer clocks by the slave device at the eighth and ninth clock may be shortened. In this case, set WAIT to 0. This does not occur in other cases.

- I²C bus interrupt enable register_0 (ICIER_0) Address: H'FFFD5B

Bit	Bit Name	Setting	R/W	Function
7	TIE	1	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, enables or disables the transmit-data empty interrupt (TXI). 0: Disables transmit-data empty interrupt request (TXI). 1: Enables transmit-data empty interrupt request (TXI).
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by cleaning the TEND bit or the TEIE bit to 0. 0: Disables transmit-end interrupt request (TEI). 1: Enables transmit-end interrupt request (TEI).
5	RIE	1	R/W	Receive Interrupt Enable Enables or disables receive-data full interrupt request (RXI) when a received data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Disables receive-data full interrupt request (RXI). 1: Enables receive-data full interrupt request (RXI).
4	NAKIE	1	R/W	NACK Receive Interrupt Enable Enables or disables the NACK receive interrupt request (NAKI) when the NACKF and AL bits in ICSR are set to 1. NAKI can be canceled by clearing the NACKF, AL, or NAKIE bits to 0. 0: Disables NACK receive interrupt request (NAKI). 1: Enables NACK receive interrupt request (NAKI).
3	STIE	1	R/W	Stop Condition Interrupt Enable 0: Disables stop condition interrupt (STPI). 1: Enables stop condition interrupt (STPI).
2	ACKE	1	R/W	Acknowledgment Bit Judgment Select 0: Ignores the value of the acknowledge bit and transfers data continuously. 1: Interrupts continuous transfer when the acknowledge bit is 1.
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, sets the value of the bit to be sent at acknowledge timing here. 0: Outputs 0 at acknowledge timing. 1: Outputs 1 at acknowledge timing.

- I²C bus status register_0 (ICSR_0) Address: H'FFFD5C

Bit	Bit Name	Setting	R/W	Function
7	TDRE	0	R/W	Transmit Data Empty [Setting condition] • ICDRT becoming empty after the transfer of data from ICDRT to ICDRS • Setting of TRS • Issuance of the start condition (including re-transmission) • Transition from reception to transmission in the slave mode [Clearing condition] • Writing of 0 to this bit after reading it as 1 • Writing of data to ICDRT
6	TEND	0	R/W	Transmit End [Setting condition] • Rising edge of the ninth cycle of SCL while the TDRE flag is 1 [Clearing condition] • Writing of 0 to this bit after reading it as 1 • Writing of data to ICDRT
5	RDRF	0	R/W	Receive Data Register Full [Setting condition] • Transfer of received data from ICDRS to ICDRR [Clearing condition] • Writing of 0 to this bit after reading it as 1 • Reading of ICDRR
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • In transmission with the ACKE bit in ICIER set to 1, detection of the not-acknowledge bit from the receiver device [Clearing condition] • Writing of 0 to this bit after reading it as 1
3	STOP	0	R/W	Stop Condition Detection Flag [Setting condition] • Detection of a stop condition in master mode after frame transfer • Detection of a stop condition in slave mode after the general call address or the first byte slave address, next to detection of start condition, matches the address set in SAR. [Clearing condition] • Writing of 0 to this bit after reading it as 1

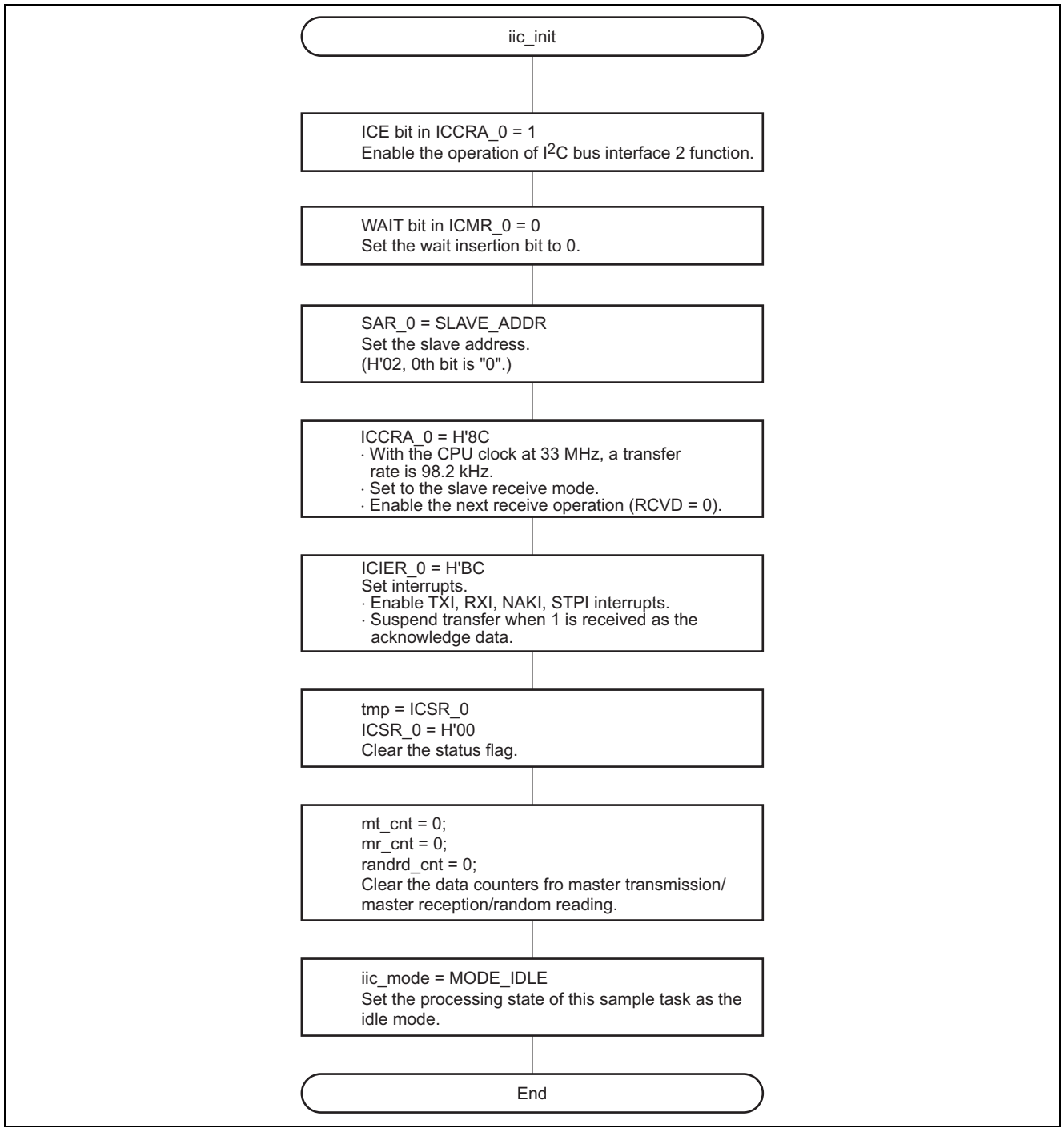
Bit	Bit Name	Setting	R/W	Function
2	AL	0	R/W	Arbitration Lost Flag This flag indicates that arbitration was lost in master mode. When two or more master devices attempt to seize the bus at nearly the same time, when the I²C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master. [Setting condition] • Different values for the internal SDA signal and SDA pin on a rising edge of SCL in master transmit mode • The signal output from the SDA pin being at high level in master mode at the time of start-condition detection. [Clearing condition] • Writing of 0 to this bit after reading it as 1
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 when the first frame following a start condition matches bits SVA6 to SVA0 in SAR. [Setting condition] • Detection of the slave address in slave receive mode • Detection of the general call address in slave receive mode. [Clearing condition] • Writing of 0 to this bit after reading it as 1

- Slave address register_0 (SAR_0) Address: H'FFFD5D

The slave address is set in the SAR bits. An interface in slave mode responds as the slave device when the 7 higher-order bits of SAR match the 7 higher-order bits of the first frame received after a start condition.

Bit	Bit Name	Setting	R/W	Function
7 to 1	SVA6 to SVA0	SLAVE_ADDR	R/W	Slave Address 6 to 0 Unique address setting (address differing from the addresses of other slave devices connected to the I²C bus for the device
0	—		R/W	Reserved Read/Write is enabled. 0 should be written in writing.

5. Flowchart


iic_init
ICE bit in ICCRA_0 = 1
Enable the operation of I2C bus interface 2 function.
WAIT bit in ICMR_0 = 0
Set the wait insertion bit to 0.
SAR_0 = SLAVE_ADDR
Set the slave address.
(H'02, 0th bit is "0".)
ICCRA_0 = H'8C
· With the CPU clock at 33 MHz, a transfer rate is 98.2 kHz.
· Set to the slave receive mode.
· Enable the next receive operation (RCVD = 0).
ICIER_0 = H'BC
Set interrupts.
· Enable TXI, RXI, NAKI, STPI interrupts.
· Suspend transfer when 1 is received as the acknowledge data.
tmp = ICSR_0
ICSR_0 = H'00
Clear the status flag.
mt_cnt = 0;
mr_cnt = 0;
randrd_cnt = 0;
Clear the data counters fro master transmission/ master reception/random reading.
iic_mode = MODE_IDLE
Set the processing state of this sample task as the idle mode.
End

5.7.2 Function mtrs_start

1. Overview

   This function sets up the task for I²C bus interface master transmission and issues the start condition.

2. Arguments

Type	Name of Variable	Description
const unsigned char	*dtadd	First address of data for transmission
unsigned short	dtnum	Number of data to be transmitted

3. Return value

Type	Description
unsigned char	0: Starts normal transmission. 1: Transmission in progress 2: Bus busy 3: Argument error

4. Internal registers used

   The following describes internal registers used in this sample task. The settings are those used in this sample task rather than the initial settings.

- I²C bus control register A_0 (ICCRA_0) Address: H'FFFD58

Bit	Bit Name	Setting	R/W	Function
5	MST	1	R/W	Master/Slave Select
4	TRS	1	R/W	11: Master transmit mode

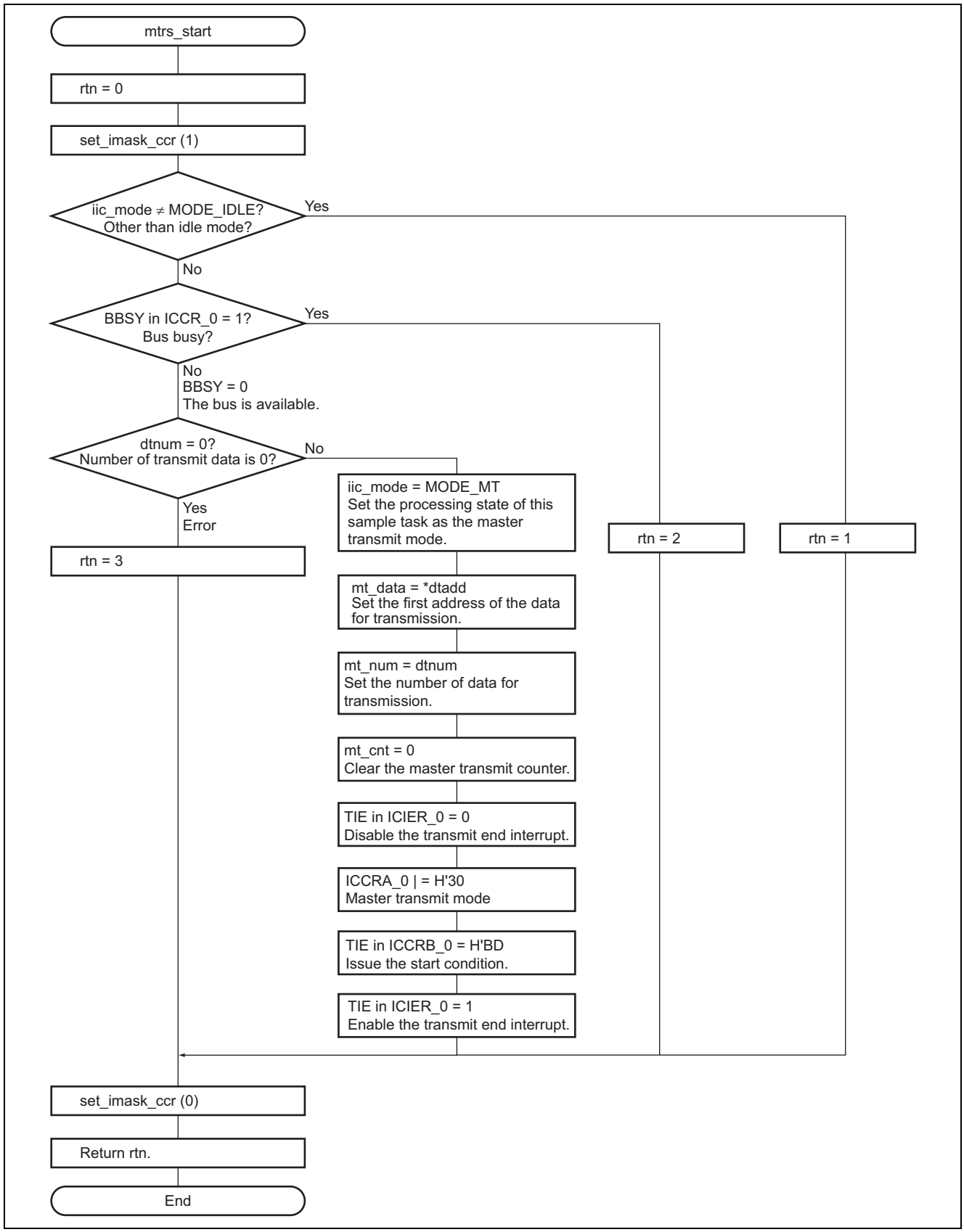
- I²C bus control register B_0 (ICCRB_0) Address: H'FFFD59

Bit	Bit Name	Setting	R/W	Function
7	BBSY	1	R/W	Bus Busy Enables to confirm whether the I²C bus is occupied or released and to issue start and stop conditions in master mode. This bit is set to 1 when the signal output from SDA changes from high to low level under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the signal level output from SDA changes from low to high level under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Write to 0 and 0 to SCP to issue a stop condition. To issue a start/stop condition, use the MOV instruction.
6	SCP	0	R/W	Start Condition/Stop Condition Prohibit Controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 to BBSY and 0 to SCP. A retransmit start condition is issued in the same way. Write 0 to BBSY and 0 to SCP to issue a stop condition. This bit is always read as 1. When 1 is written, the data is not stored

- I²C bus interrupt enable register_0 (ICIER_0) Address: H'FFFD5B

Bit	Bit Name	Setting	R/W	Function
7	TIE	1	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, enables or disables the transmit-data empty interrupt (TXI). 0: Disables transmit-data empty interrupt request (TXI). 1: Enables transmit-data empty interrupt request (TXI).

5. Flowchart


mtrs_start
rtn = 0
set_imask_ccr (1)
iic_mode ≠ MODE_IDLE?
Other than idle mode?
Yes
No
BBSY in ICCR_0 = 1?
Bus busy?
Yes
No
BBSY = 0
The bus is available.
dtnum = 0?
Number of transmit data is 0?
No
Yes
Error
rtn = 3
iic_mode = MODE_MT
Set the processing state of this sample task as the master transmit mode.
mt_data = *dtadd
Set the first address of the data for transmission.
mt_num = dtnum
Set the number of data for transmission.
mt_cnt = 0
Clear the master transmit counter.
TIE in ICIER_0 = 0
Disable the transmit end interrupt.
ICCRA_0 | = H'30
Master transmit mode
TIE in ICCRB_0 = H'BD
Issue the start condition.
TIE in ICIER_0 = 1
Enable the transmit end interrupt.
rtn = 2
rtn = 1
set_imask_ccr (0)
Return rtn.
End

5.7.3 Function mrcv_start

1. Overview

This function sets up the task for I²C bus interface master reception and issues the start condition.

2. Arguments

Type	Name of Variable	Description
const unsigned char	*dtadd	First address of received data
unsigned short	dtnum	Number of data (bytes) received

3. Return value

Type	Description
unsigned char	0: Starts normal transmission. 1: Transmission in progress 2: Bus busy 3: Argument error

4. Internal registers used

The following describes internal registers used in this sample task. The settings are those used in this sample task rather than the initial settings.

- I²C bus control register A_0 (ICCRA_0) Address: H'FFFD58

Bit	Bit Name	Setting	R/W	Function
5	MST	1	R/W	Master/Slave Select
4	TRS	1	R/W	11: Master transmit mode

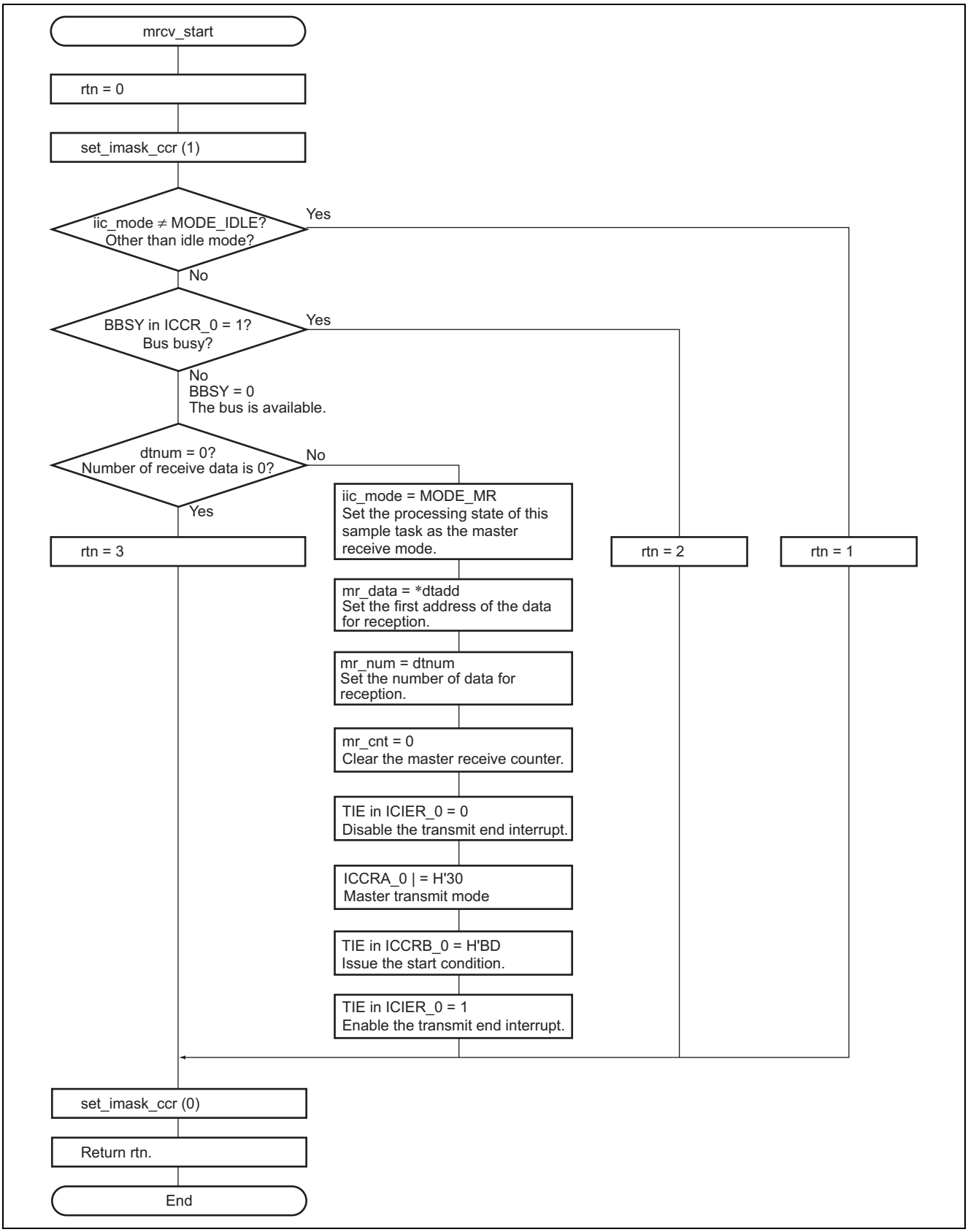
- I²C bus control register B_0 (ICCRB_0) Address: H'FFFD59

Bit	Bit Name	Setting	R/W	Function
7	BBSY	1	R/W	Bus Busy Enables confirmation of whether the I²C bus is occupied or released and the issuance of start and stop conditions in master mode. When the start condition has been issued, this bit is set to 1 when the signal output from SDA changes from high to low level while SCL is high. When the stop condition has been issued, this bit is cleared to 0 when the signal level output from SDA changes from low to high level while SCL is high. Write 1 to BBSY and 0 to SCP to issue a start condition. Write 0 to BBSY and 0 to SCP to issue a stop condition. Use the MOV instruction to make the settings for the issuance of start and stop conditions.
6	SCP	0	R/W	Start Condition/Stop Condition Prohibit Controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 to BBSY and 0 to SCP. A retransmit start condition is issued in the same way. Write 0 to BBSY and 0 to SCP to issue a stop condition. This bit is always read as 1. When 1 is written, the data is not stored

- I²C bus interrupt enable register_0 (ICIER_0) Address: H'FFFD5B

Bit	Bit Name	Setting	R/W	Function
7	TIE	1	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, enables or disables the transmit-data empty interrupt (TXI) 0: Disables transmit-data empty interrupt request (TXI). 1: Enables transmit-data empty interrupt request (TXI).

5. Flowchart


mrcv_start
rtn = 0
set_imask_ccr (1)
iic_mode ≠ MODE_IDLE?
Other than idle mode?
Yes
No
BBSY in ICCR_0 = 1?
Bus busy?
Yes
No
BBSY = 0
The bus is available.
dtnum = 0?
Number of receive data is 0?
No
Yes
rtn = 3
iic_mode = MODE_MR
Set the processing state of this sample task as the master receive mode.
mr_data = *dtadd
Set the first address of the data for reception.
mr_num = dtnum
Set the number of data for reception.
mr_cnt = 0
Clear the master receive counter.
TIE in ICIER_0 = 0
Disable the transmit end interrupt.
ICCRA_0 | = H'30
Master transmit mode
TIE in ICCRB_0 = H'BD
Issue the start condition.
TIE in ICIER_0 = 1
Enable the transmit end interrupt.
rtn = 2
rtn = 1
set_imask_ccr (0)
Return rtn.
End

5.7.4 Function mrandrd_start

1. Overview
   This function sets up the task for random reading, including issuance of a start condition to initiate reading from the EEPROM.

2. Arguments

Type	Name of Variable	Description
const unsigned char	*mrbuf	Location where the random-read data is to be stored
unsigned short	mtbuf	EEPROM memory address

3. Return value
   None

4. Internal registers used
   The following describes internal registers used in this sample task. The settings are those used in this sample task rather than the initial settings.

- I²C bus control register A_0 (ICCRA_0) Address: H'FFFD58

Bit	Bit Name	Setting	R/W	Function
5	MST	1	R/W	Master/Slave Select
4	TRS	1	R/W	11: Master transmit mode

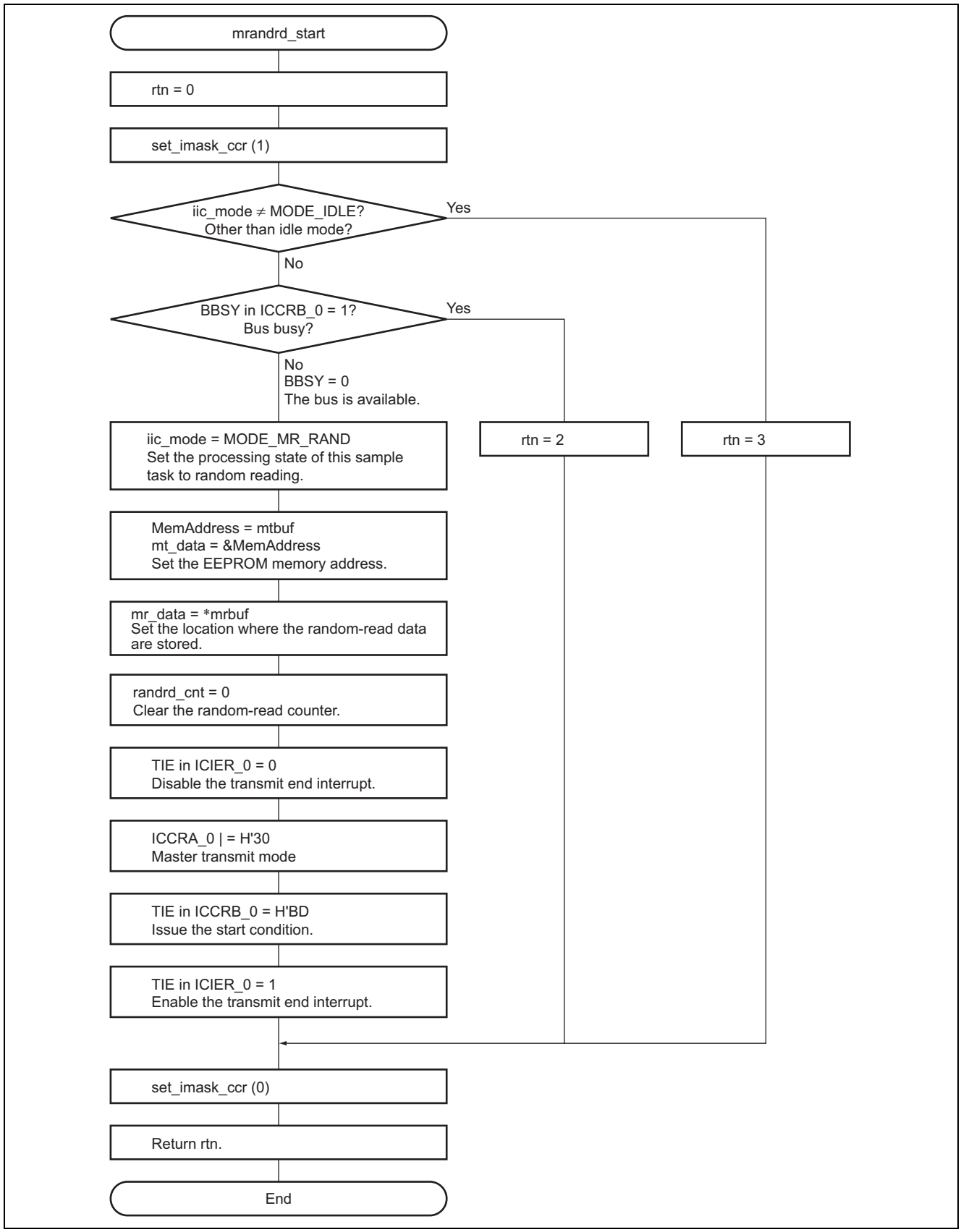
- I²C bus control register B_0 (ICCRB_0) Address: H'FFFD59

Bit	Bit Name	Setting	R/W	Function
7	BBSY	1	R/W	Bus Busy Enables to confirm whether the I²C bus is occupied or released and to issue start and stop conditions in master mode. This bit is set to 1 when the signal output from SDA changes from high to low level under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the signal level output from SDA changes from low to high level under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Write to 0 and 0 to SCP to issue a stop condition. To issue a start/stop condition, use the MOV instruction.
6	SCP	0	R/W	Start Condition/Stop Condition Prohibit Controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 to BBSY and 0 to SCP. A retransmit start condition is issued in the same way. Write 0 to BBSY and 0 to SCP to issue a stop condition. This bit is always read as 1. When 1 is written, the data is not stored

- I²C bus interrupt enable register_0 (ICIER_0) Address: H'FFFD5B

Bit	Bit Name	Setting	R/W	Function
7	TIE	1	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, enables or disables the transmit-data empty interrupt (TXI) 0: Disables transmit-data empty interrupt request (TXI). 1: Enables transmit-data empty interrupt request (TXI).

5. Flowchart


mrandrd_start
rtn = 0
set_imask_ccr (1)
iic_mode ≠ MODE_IDLE?
Other than idle mode?
Yes
No
BBSY in ICCRB_0 = 1?
Bus busy?
Yes
No
BBSY = 0
The bus is available.
iic_mode = MODE_MR_RAND
Set the processing state of this sample task to random reading.
rtn = 2
rtn = 3
MemAddress = mtbuf
mt_data = &MemAddress
Set the EEPROM memory address.
mr_data = *mrbuf
Set the location where the random-read data are stored.
randrd_cnt = 0
Clear the random-read counter.
TIE in ICIER_0 = 0
Disable the transmit end interrupt.
ICCRA_0 | = H'30
Master transmit mode
TIE in ICCRB_0 = H'BD
Issue the start condition.
TIE in ICIER_0 = 1
Enable the transmit end interrupt.
set_imask_ccr (0)
Return rtn.
End

5.7.5 Function iici0_int

1. Overview
   Handler for I²C bus interface interrupts. According to the state of operations, this function calls the functions for receiving the stop condition, master transmission, master reception, and random reading.
2. Arguments
   None
3. Return value
   None
4. Internal registers used
   The following describes internal registers used in this sample task. The settings are those used in this sample task rather than the initial settings.

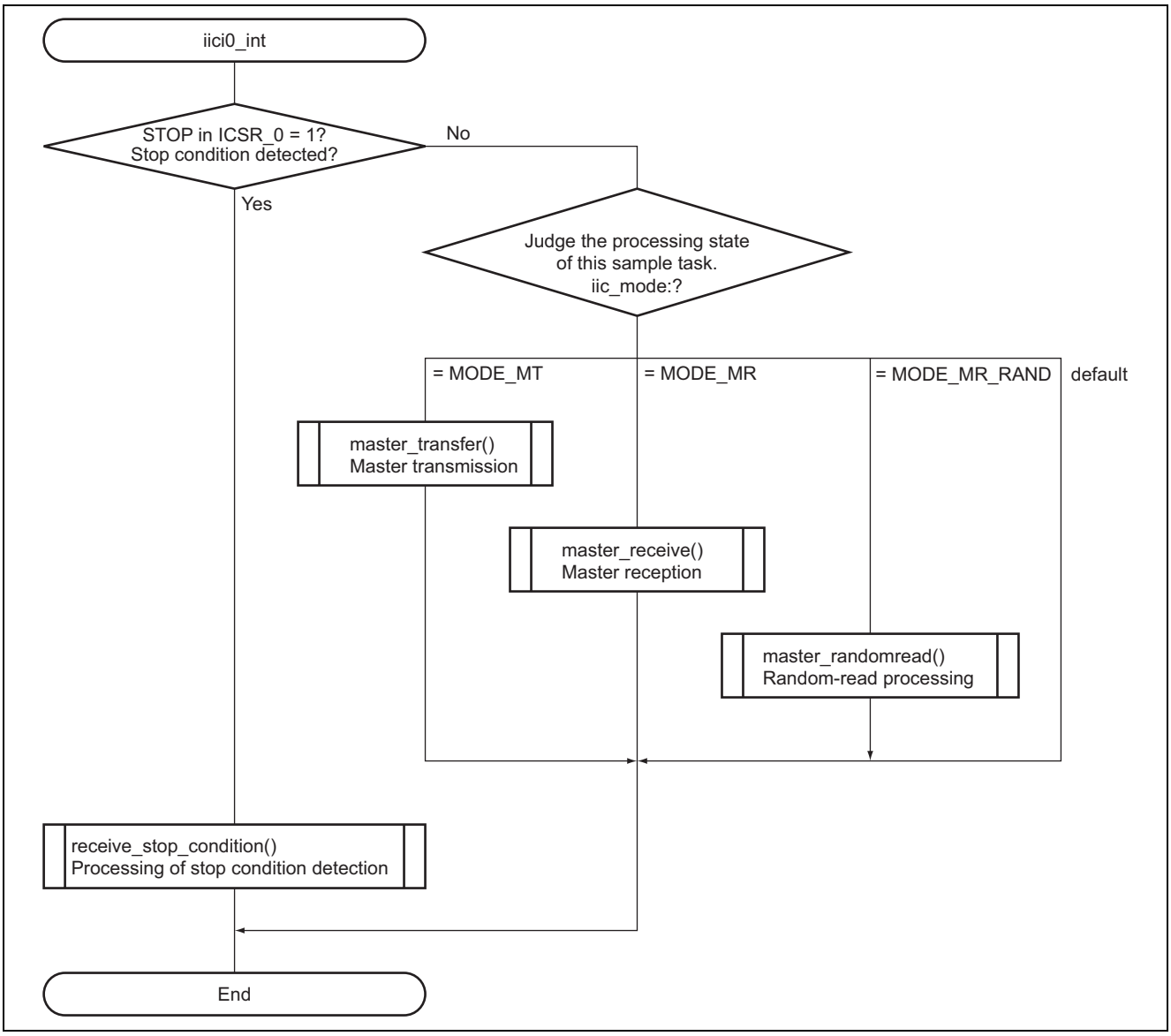
- I²C bus control register A_0 (ICCRA_0) Address: H'FFFD58

Bit	Bit Name	Setting	R/W	Function
4	TRS	Undefined	R/W	Master/Slave Select 0: Receive mode 1: Transmit mode

- I²C bus status register_0 (ICSR_0) Address: H'FFFD5C

Bit	Bit Name	Setting	R/W	Function
3	STOP	Undefined	R/W	Stop Condition Detection Flag [Setting condition] • Detection of a stop condition in master mode after frame transfer • Detection of a stop condition in slave mode after the general call address or the first byte slave address, next to detection of start condition, matches the address set in SAR. [Clearing condition] • Writing of 0 to this bit after reading it as 1

5. Flowchart


iici0_int
STOP in ICSR_0 = 1?
Stop condition detected?
No
Yes
Judge the processing state
of this sample task.
iic_mode:?
= MODE_MT
= MODE_MR
= MODE_MR_RAND
default
master_transfer()
Master transmission
master_receive()
Master reception
master_randomread()
Random-read processing
receive_stop_condition()
Processing of stop condition detection
End

5.7.6 Function receive_stop_condition

1. Overview
   This function handles processing on detection of the stop condition.
2. Arguments
   None
3. Return value
   None
4. Internal registers used
   The following describes internal registers used in this sample task. The settings are those used in this sample task rather than the initial settings.

- I²C bus control register A_0 (ICCRA_0) Address: H'FFFD58

Bit	Bit Name	Setting	R/W	Function
6	RCVD	0	R/W	Reception Disable Enables or disables the next operation when ICDRR is read while the TRS bit is 0. 0: Enables next operation 1: Disables next operation
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	00: Slave receive mode

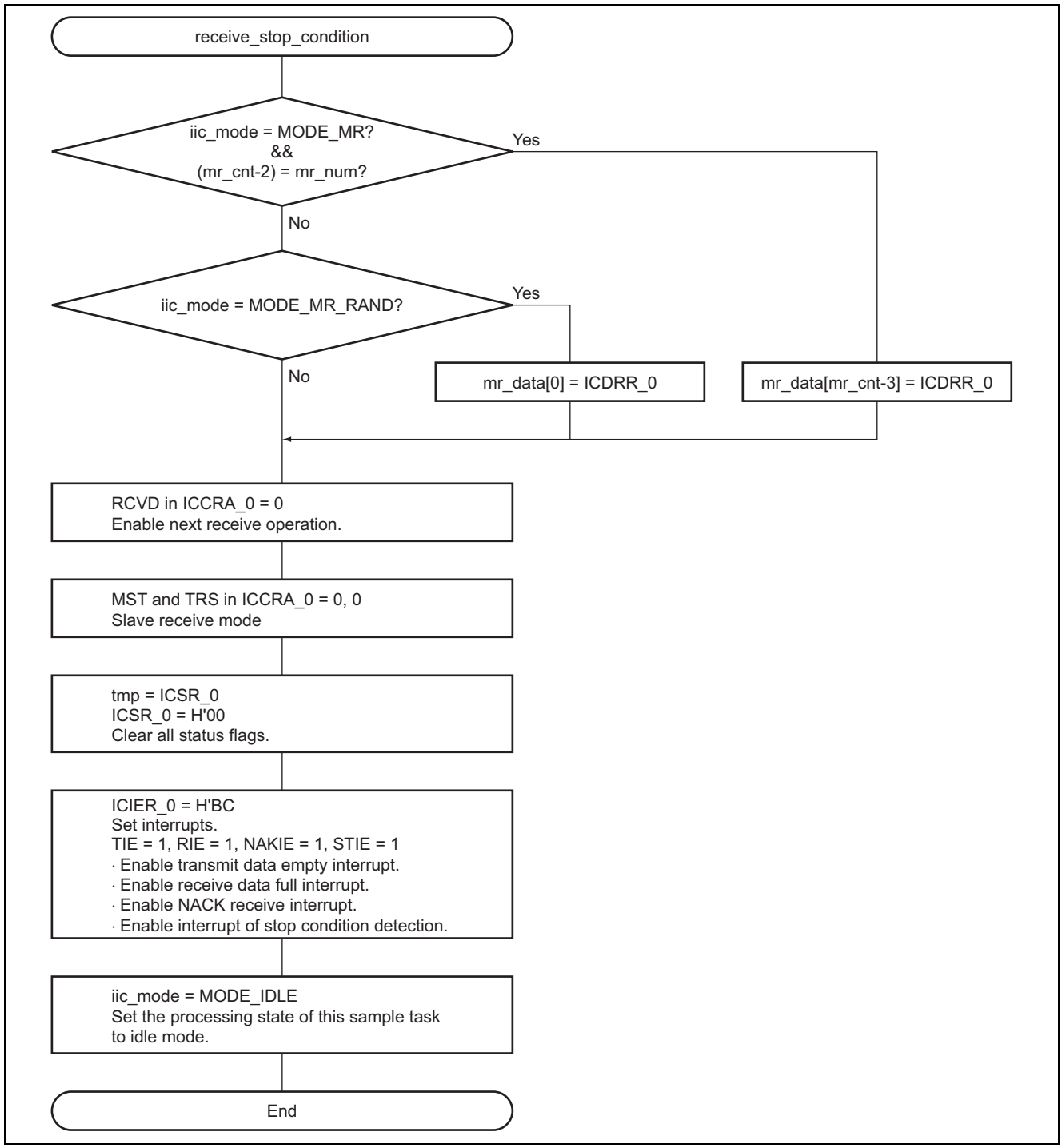
- I²C bus interrupt enable register_0 (ICIER_0) Address: H'FFFD5B

Bit	Bit Name	Setting	R/W	Function
7	TIE	1	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, enables or disables the transmit-data empty interrupt (TXI) 0: Disables transmit-data empty interrupt request (TXI). 1: Enables transmit-data empty interrupt request (TXI).
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by cleaning the TEND bit or the TEIE bit to 0. 0: Disables transmit end interrupt request (TEI). 1: Enables transmit end interrupt request (TEI).
5	RIE	1	R/W	Receive Interrupt Enable Enables or disables receive data full interrupt request (RXI) when a received data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Disables receive data full interrupt request (RXI). 1: Enables receive data full interrupt request (RXI).
4	NAKIE	1	R/W	NACK Receive Interrupt Enable Enables or disables the NACK-receive interrupt request (NAKI) when the NACKF and AL bits in ICSR are set to 1. NAKI can be canceled by clearing the NACKF, AL, or NAKIE bits to 0. 0: Disables NACK receive interrupt request (NAKI). 1: Enables NACK receive interrupt request (NAKI).
3	STIE	1	R/W	Stop Condition Interrupt Enable 0: Disables stop condition interrupt (STPI). 1: Enables stop condition interrupt (STPI).
2	ACKE	1	R/W	Acknowledgment Bit Judgment Select 0: Ignores the value of the acknowledge bit and transfers data continuously. 1: Interrupts continuous transfer when the acknowledge bit is 1.
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, sets the value of the bit to be sent at acknowledge timing here. 0: Outputs 0 at acknowledge timing. 1: Outputs 1 at acknowledge timing.

- I²C bus status register_0 (ICSR_0) Address: H'FFFD5C

Bit	Bit Name	Setting	R/W	Function
7	TDRE	0	R/W	Transmit Data Empty [Setting condition] • Transferring data from ICDRT to ICDRS and making ICDRT empty • Setting the TRS • Issuance of the start condition (including re-transmission) • Making a transition from the receive mode to the transmit mode in the slave mode [Clearing condition] • Writing of 0 to this bit after reading it as 1 • Writing data to ICDRT
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • No detection of acknowledge from the receiver device when transmitting [Clearing condition] • Writing of 0 to this bit after reading it as 1
1	AAS	0	R/W	Slave Address Recognition Flag In slave receive mode, this flag is set to 1 when the first frame following a start condition matches bits SVA6 to SVA0 in SAR. [Setting condition] • Detection of the slave address in slave receive mode • Detection of the general call address in slave receive mode. [Clearing condition] • Writing of 0 to this bit after reading it as 1

5. Flowchart


receive_stop_condition
iic_mode = MODE_MR?
&&
(mr_cnt-2) = mr_num?
Yes
No
iic_mode = MODE_MR_RAND?
Yes
No
mr_data[0] = ICDRR_0
mr_data[mr_cnt-3] = ICDRR_0
RCVD in ICCRA_0 = 0
Enable next receive operation.
MST and TRS in ICCRA_0 = 0, 0
Slave receive mode
tmp = ICSR_0
ICSR_0 = H'00
Clear all status flags.
ICIER_0 = H'BC
Set interrupts.
TIE = 1, RIE = 1, NAKIE = 1, STIE = 1
· Enable transmit data empty interrupt.
· Enable receive data full interrupt.
· Enable NACK receive interrupt.
· Enable interrupt of stop condition detection.
iic_mode = MODE_IDLE
Set the processing state of this sample task
to idle mode.
End

5.7.7 Function master_transfer

1. Overview
   Master transmission process which is called from the I²C bus interface interrupt handler. In this case, the interrupt source will be the transmit data empty interrupt for each byte of transmitted data. When arbitration is lost, it places the interface in slave receive mode.
2. Arguments
   None
3. Return value
   None
4. Internal registers used
   The following describes internal registers used in this sample task. The settings are those used in this sample task rather than the initial settings.

- I²C bus control register B_0 (ICCRB_0) Address: H'FFFD59

Bit	Bit Name	Setting	R/W	Function
3	SCLO	1	R/W	Monitors the signal level output from the SCL pin. When SCLO is read as 1, the signal output from SCL is at the high level. When SCLO is read as 0, the signal output from SCL is at the low level.

- I²C bus interrupt enable register_0 (ICIER_0) Address: H'FFFD5B

Bit	Bit Name	Setting	R/W	Function
7	TIE	1	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, enables or disables the transmit-data empty interrupt (TXI) 0: Disables transmit-data empty interrupt request (TXI). 1: Enables transmit-data empty interrupt request (TXI).
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by cleaning the TEND bit or the TEIE bit to 0. 0: Disables transmit end interrupt request (TEI). 1: Enables transmit end interrupt request (TEI).

- I²C bus status register_0 (ICSR_0) Address: H'FFFD5C

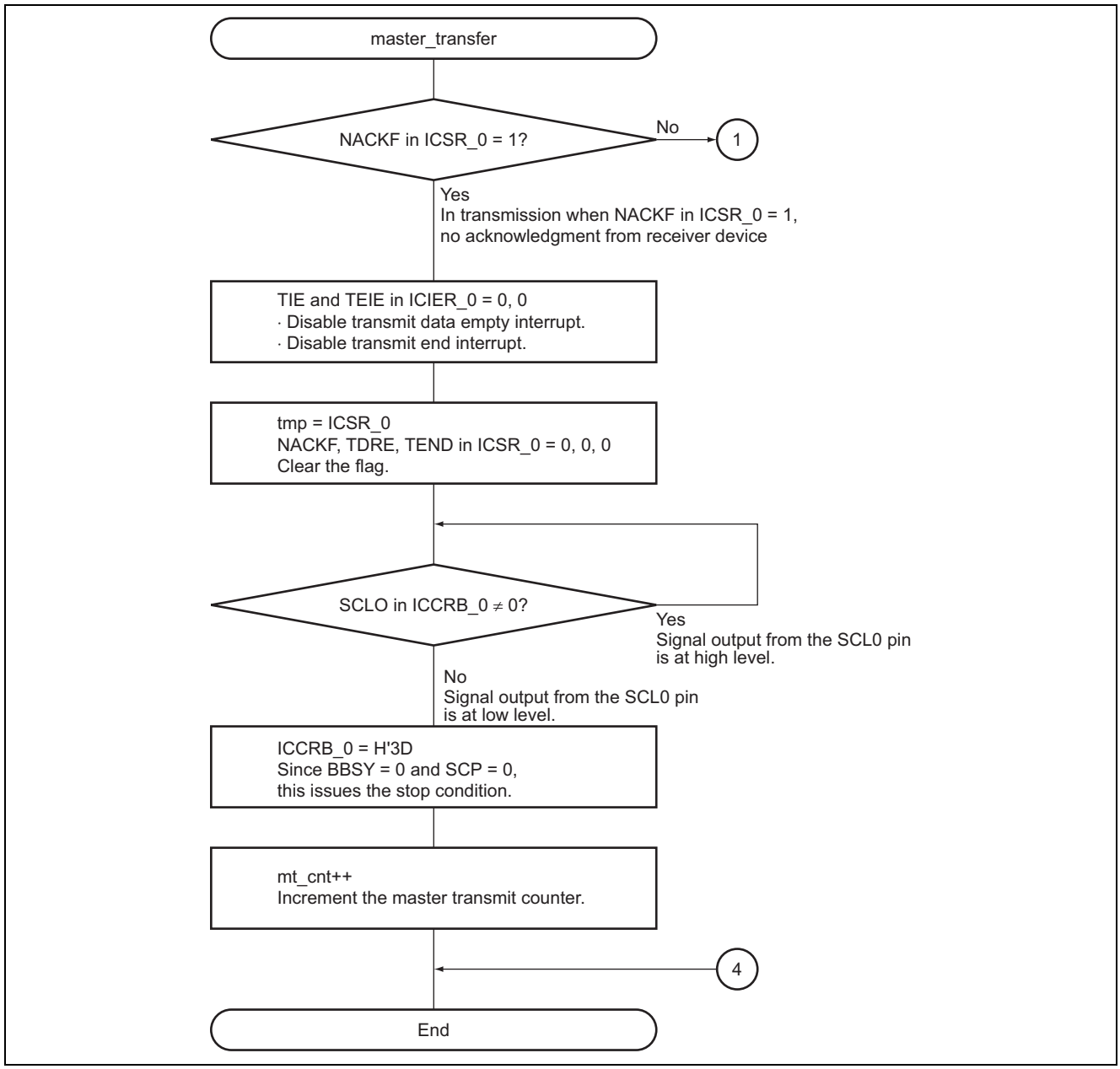
Bit	Bit Name	Setting	R/W	Function
7	TDRE	0	R/W	Transmit Data Empty [Setting condition] • Transferring data from ICDRT to ICDRS and making ICDRT empty • Setting the TRS • Issuance of the start condition (including re-transmission) • Making a transition from the receive mode to the transmit mode in the slave mode [Clearing condition] • Writing of 0 to this bit after reading it as 1 • Writing data to ICDRT
6	TEND	0	R/W	Transmit End [Setting condition] • Rising edge of the ninth cycle of SCL while the TDRE flag is 1 [Clearing condition] • Writing of 0 to this bit after reading it as 1 • Writing data to ICDRT

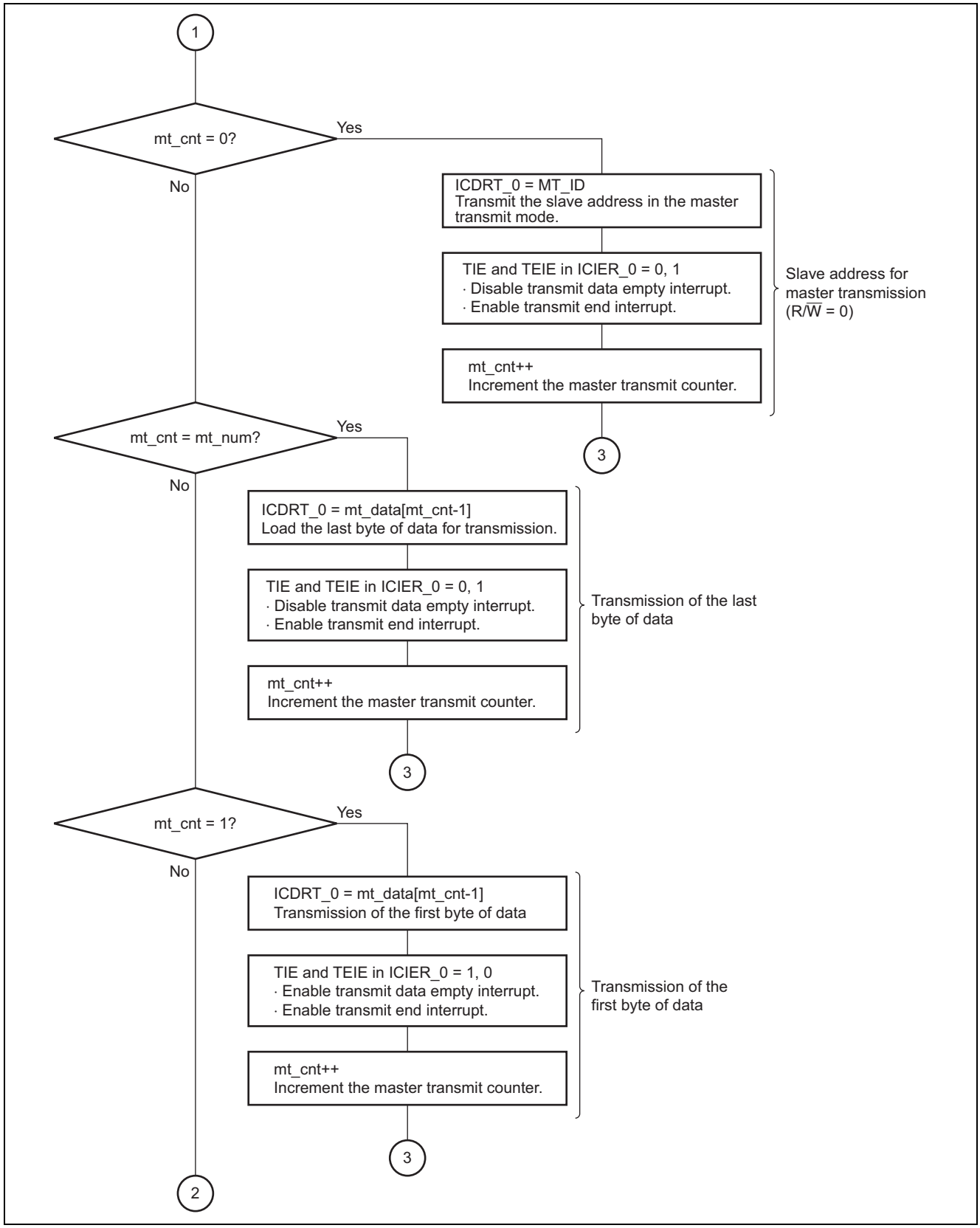
- I²C bus transmit data register_0 (ICDRT_0) Address: H'FFFD5E

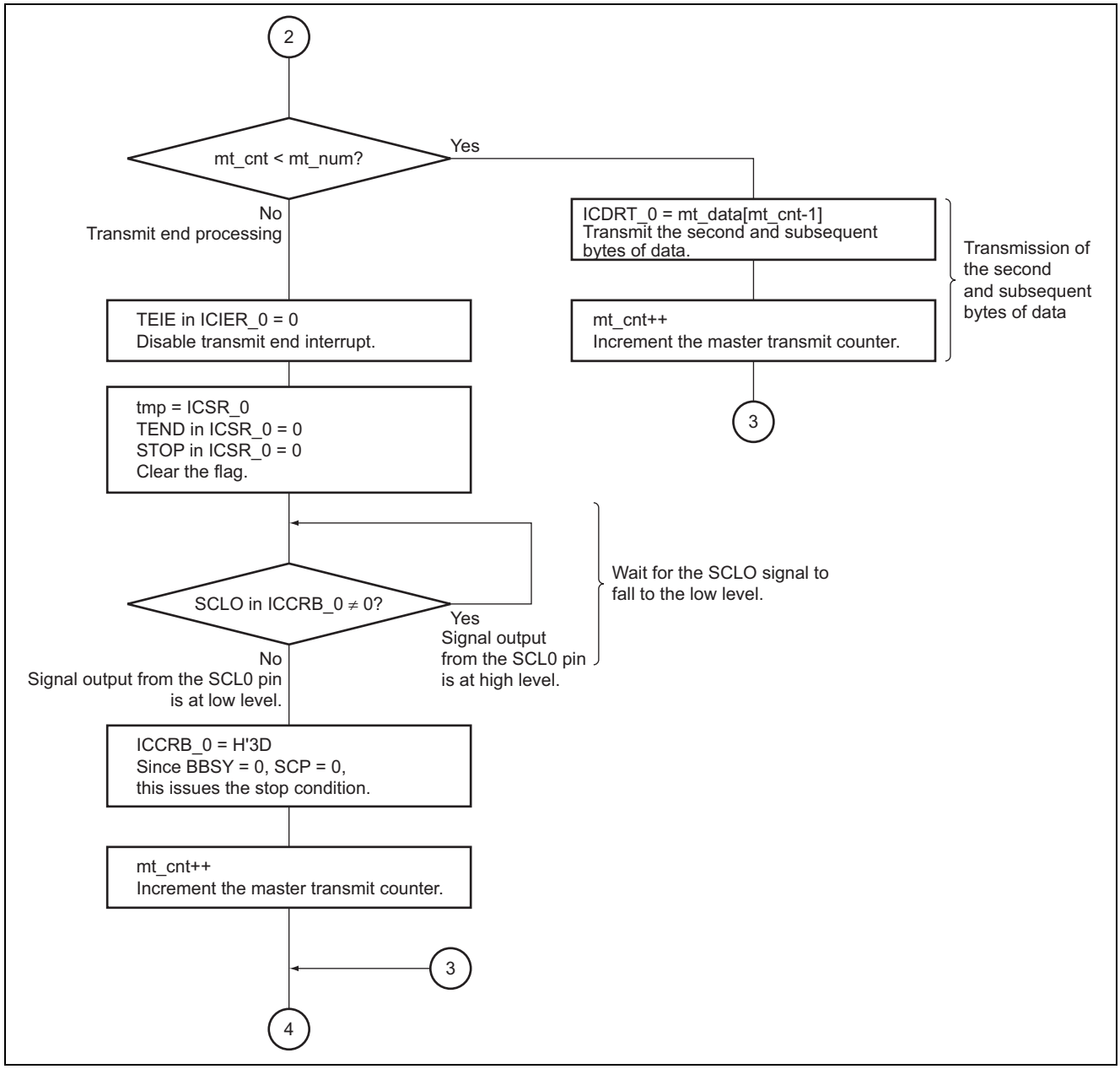
  Function: ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the I²C bus shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. When the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. The initial value of ICDRT is H'FF.

  Setting: S_Sent[st_cnt]

5. Flowchart


master_transfer
NACKF in ICSR_0 = 1?
No
1
Yes
In transmission when NACKF in ICSR_0 = 1,
no acknowledgment from receiver device
TIE and TEIE in ICIER_0 = 0, 0
· Disable transmit data empty interrupt.
· Disable transmit end interrupt.
tmp = ICSR_0
NACKF, TDRE, TEND in ICSR_0 = 0, 0, 0
Clear the flag.
SCLO in ICCRB_0 ≠ 0?
Yes
Signal output from the SCL0 pin
is at high level.
No
Signal output from the SCL0 pin
is at low level.
ICCRB_0 = H'3D
Since BBSY = 0 and SCP = 0,
this issues the stop condition.
mt_cnt++
Increment the master transmit counter.
4
End


1
mt_cnt = 0?
Yes
No
ICDRT_0 = MT_ID
Transmit the slave address in the master transmit mode.
TIE and TEIE in ICIER_0 = 0, 1
· Disable transmit data empty interrupt.
· Enable transmit end interrupt.
mt_cnt++
Increment the master transmit counter.
Slave address for master transmission (R/W = 0)
3
mt_cnt = mt_num?
Yes
No
ICDRT_0 = mt_data[mt_cnt-1]
Load the last byte of data for transmission.
TIE and TEIE in ICIER_0 = 0, 1
· Disable transmit data empty interrupt.
· Enable transmit end interrupt.
mt_cnt++
Increment the master transmit counter.
Transmission of the last byte of data
3
mt_cnt = 1?
Yes
No
ICDRT_0 = mt_data[mt_cnt-1]
Transmission of the first byte of data
TIE and TEIE in ICIER_0 = 1, 0
· Enable transmit data empty interrupt.
· Enable transmit end interrupt.
mt_cnt++
Increment the master transmit counter.
Transmission of the first byte of data
3
2


2
mt_cnt < mt_num?
Yes
No
Transmit end processing
ICDRT_0 = mt_data[mt_cnt-1]
Transmit the second and subsequent bytes of data.
Transmission of the second and subsequent bytes of data
TEIE in ICIER_0 = 0
Disable transmit end interrupt.
mt_cnt++
Increment the master transmit counter.
3
tmp = ICSR_0
TEND in ICSR_0 = 0
STOP in ICSR_0 = 0
Clear the flag.
SCLO in ICCRB_0 ≠ 0?
Yes
Signal output from the SCL0 pin is at high level.
Wait for the SCLO signal to fall to the low level.
No
Signal output from the SCL0 pin is at low level.
ICCRB_0 = H'3D
Since BBSY = 0, SCP = 0,
this issues the stop condition.
mt_cnt++
Increment the master transmit counter.
3
4

5.7.8 Function master_receive

1. Overview
   Master-reception process which is called by the I²C bus interface interrupt handler. In this case, the interrupt source will be the receive data full interrupt for each byte of received data.
2. Arguments
   None
3. Return value
   None
4. Internal registers used
   The following describes internal registers used in this sample task. The settings are those used in this sample task rather than the initial settings.

- I²C bus control register A_0 (ICCRA_0) Address: H'FFFD58

Bit	Bit Name	Setting	R/W	Function
6	RCVD	0	R/W	Reception Disable Enables or disables the next operation when ICDRR is read while the TRS bit is 0. 0: Enables next operation 1: Disables next operation
4	TRS	Undefined	R/W	Master/Slave Select 0: Receive mode 1: Transmit mode

- I²C bus control register B_0 (ICCRB_0) Address: H'FFFD59

Bit	Bit Name	Setting	R/W	Function
7	BBSY	1	R/W	Bus Busy Enables to confirm whether the I²C bus is occupied or released and to issue start and stop conditions in master mode. This bit is set to 1 when the signal output from SDA changes from high to low level under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the signal level output from SDA changes from low to high level under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Write to 0 and 0 to SCP to issue a stop condition. To issue a start/stop condition, use the MOV instruction.
6	SCP	0	R/W	Start Condition/Stop Condition Prohibit Controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 to BBSY and 0 to SCP. A retransmit start condition is issued in the same way. Write 0 to BBSY and 0 to SCP to issue a stop condition. This bit is always read as 1. When 1 is written, the data is not stored
3	SCLO	1	R/W	Monitors signal level output from the SCL pin. When SCLO is read as 1, signal output from SCL is at high level. When SCLO is read as 0, signal output from SCL is at low level.

- I²C bus interrupt enable register_0 (ICIER_0) Address: H'FFFD5B

Bit	Bit Name	Setting	R/W	Function
7	TIE	1	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, enables or disables the transmit-data empty interrupt (TXI) 0: Disables transmit-data empty interrupt request (TXI). 1: Enables transmit-data empty interrupt request (TXI).
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by cleaning the TEND bit or the TEIE bit to 0. 0: Disables transmit end interrupt request (TEI). 1: Enables transmit end interrupt request (TEI).
5	RIE	1	R/W	Receive Interrupt Enable Enables or disables receive-data full interrupt request (RXI) when a received data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Disables receive-data full interrupt request (RXI). 1: Enables receive-data full interrupt request (RXI).
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, sets the value of the bit to be sent at acknowledge timing here. 0: Outputs 0 at acknowledge timing. 1: Outputs 1 at acknowledge timing.

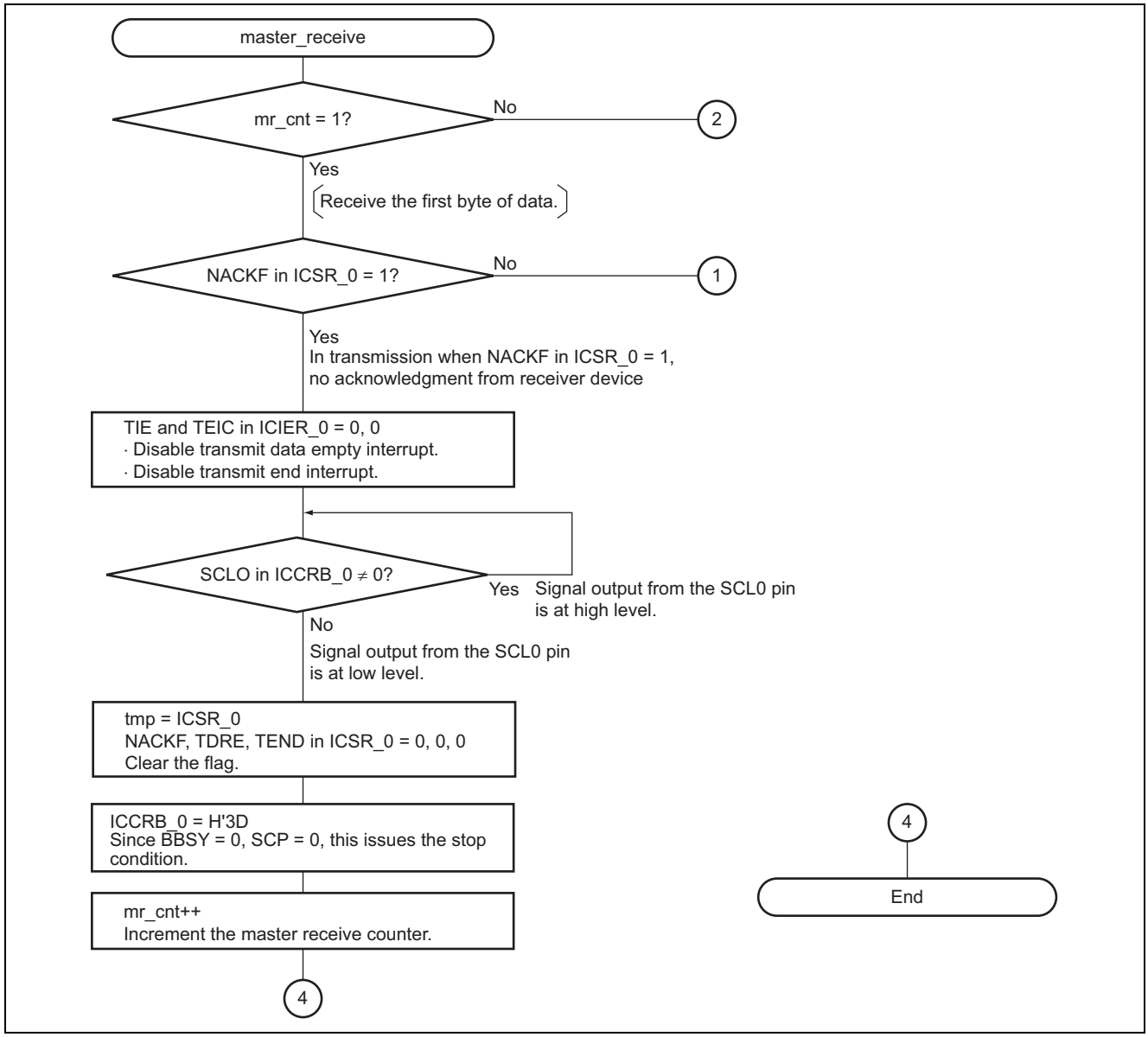
- I²C bus status register_0 (ICSR_0) Address: H'FFFD5C

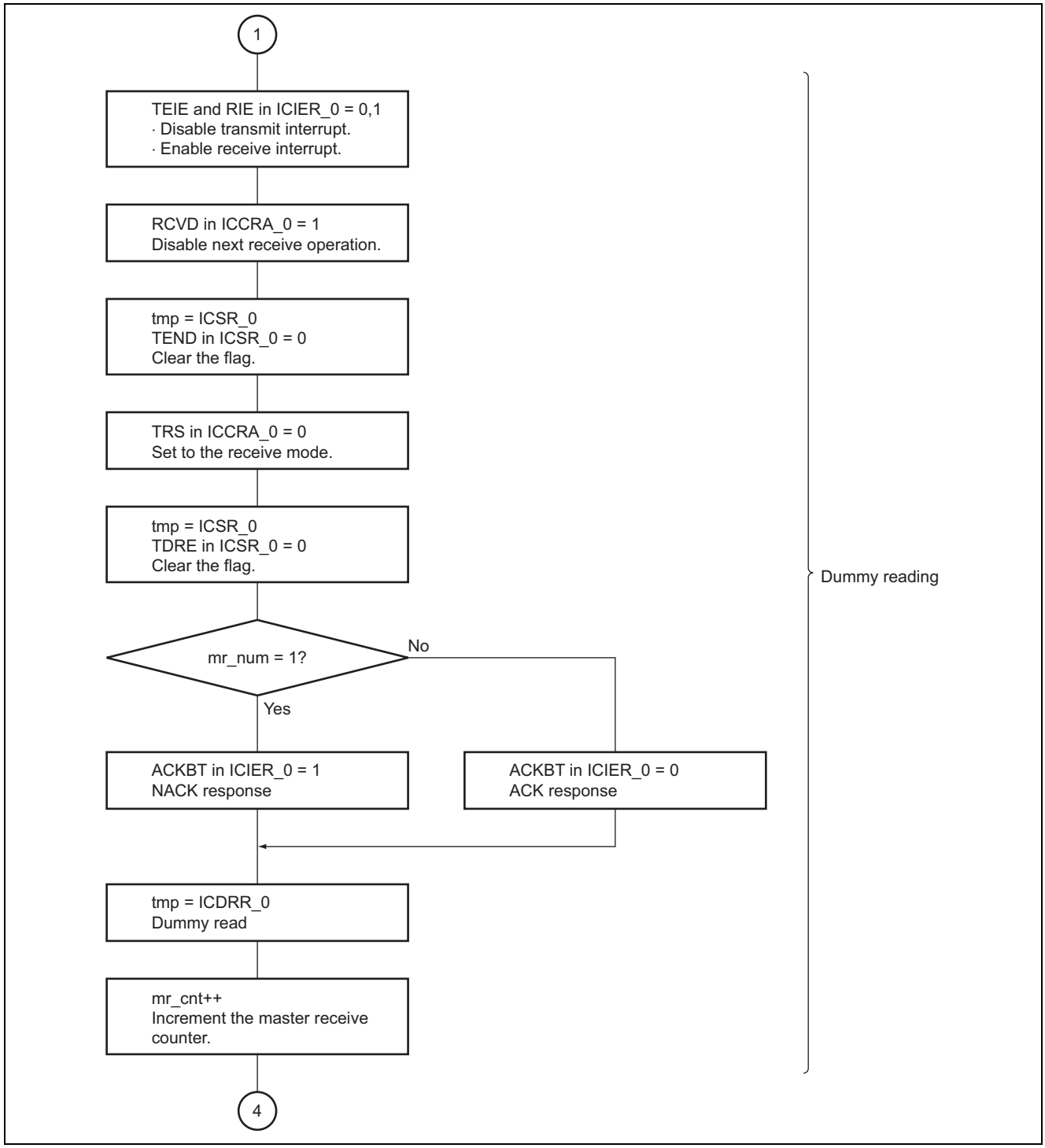
Bit	Bit Name	Setting	R/W	Function
6	TEND	0	R/W	Transmit End [Setting condition] • Rising edge of the ninth cycle of SCL while the TDRE flag is 1 [Clearing condition] • Writing of 0 to this bit after reading it as 1 • Writing data to ICDRT
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • No detection of acknowledge from the receiver device when transmitting while the ACKE bit in ICIER is 1. [Clearing condition] • Writing of 0 to this bit after reading it as 1

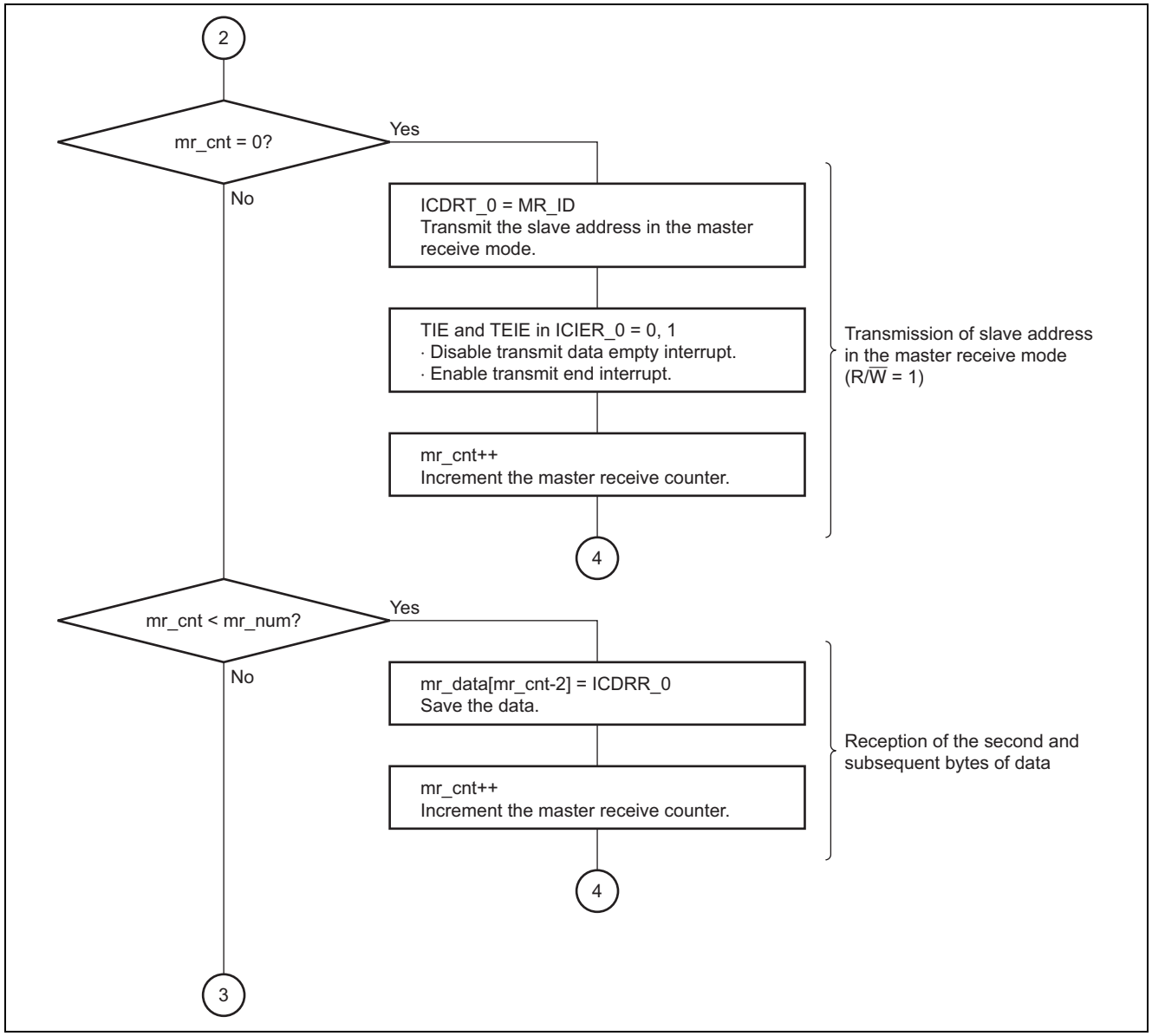
- I²C bus transmit data register_0 (ICDRT_0) Address: H'FFFD5E
  Function: ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the I²C bus shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. When the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. The initial value of ICDRT is H'FF.
  Setting: S_Sent[st_cnt]

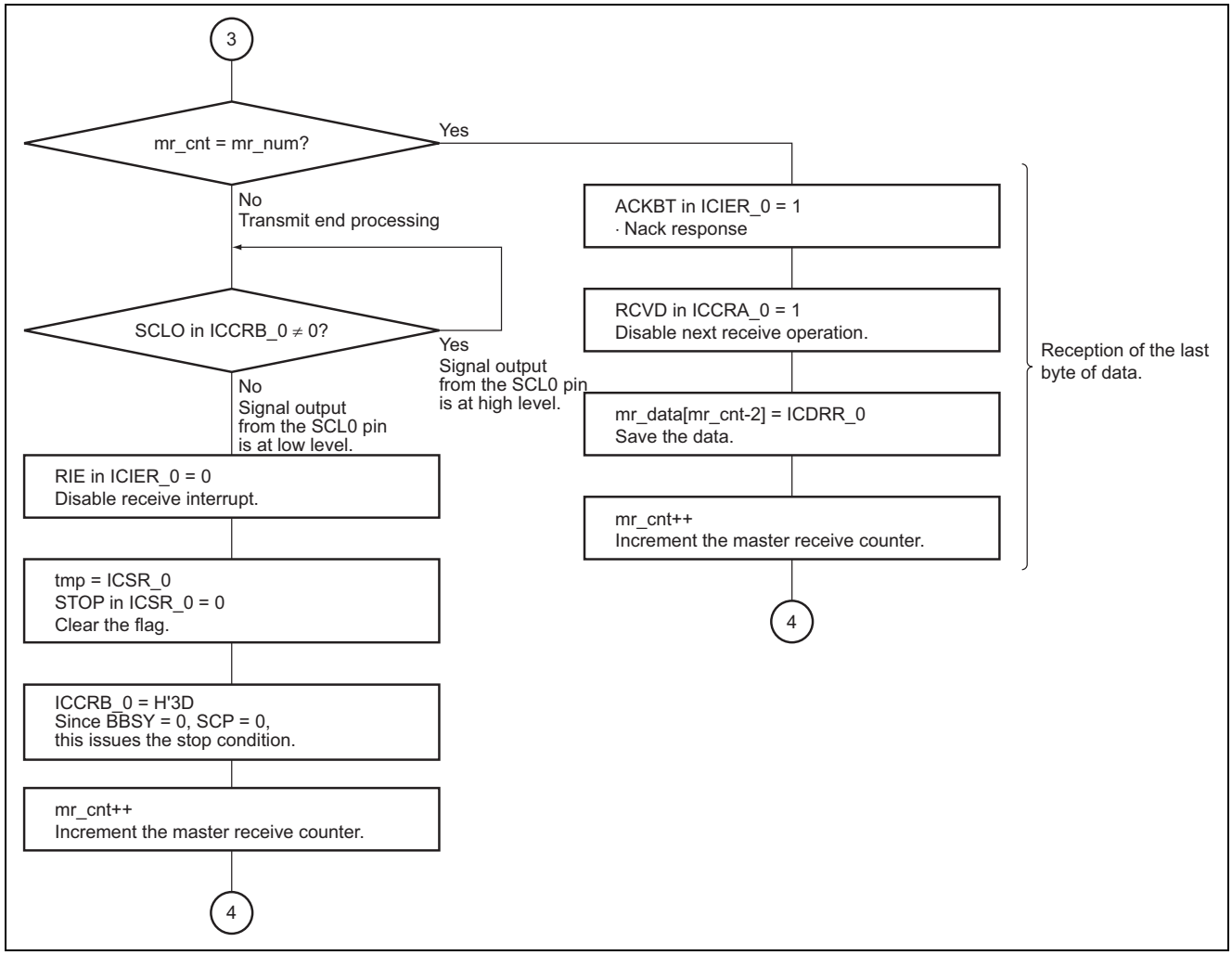
- Bus receive data register_0 (ICDRR_0) Address: H'FFFD5F
  Function: ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the received data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot be written to this register. The initial value of ICDRR is H'FF.
  Setting: Undefined

5. Flowchart




1
TEIE and RIE in ICIER_0 = 0,1
· Disable transmit interrupt.
· Enable receive interrupt.
RCVD in ICCRA_0 = 1
Disable next receive operation.
tmp = ICSR_0
TEND in ICSR_0 = 0
Clear the flag.
TRS in ICCRA_0 = 0
Set to the receive mode.
tmp = ICSR_0
TDRE in ICSR_0 = 0
Clear the flag.
mr_num = 1?
No
Yes
ACKBT in ICIER_0 = 1
NACK response
ACKBT in ICIER_0 = 0
ACK response
tmp = ICDRR_0
Dummy read
mr_cnt++
Increment the master receive counter.
4
Dummy reading


2
mr_cnt = 0?
Yes
No
ICDRT_0 = MR_ID
Transmit the slave address in the master receive mode.
TIE and TEIE in ICIER_0 = 0, 1
· Disable transmit data empty interrupt.
· Enable transmit end interrupt.
mr_cnt++
Increment the master receive counter.
Transmission of slave address in the master receive mode (R/W = 1)
4
mr_cnt < mr_num?
Yes
No
mr_data[mr_cnt-2] = ICDRR_0
Save the data.
mr_cnt++
Increment the master receive counter.
Reception of the second and subsequent bytes of data
4
3


3
mr_cnt = mr_num?
Yes
No
Transmit end processing
SCLO in ICCRB_0 ≠ 0?
Yes
Signal output
from the SCL0 pin
is at high level.
No
Signal output
from the SCL0 pin
is at low level.
RIE in ICIER_0 = 0
Disable receive interrupt.
tmp = ICSR_0
STOP in ICSR_0 = 0
Clear the flag.
ICCRB_0 = H'3D
Since BBSY = 0, SCP = 0,
this issues the stop condition.
mr_cnt++
Increment the master receive counter.
4
ACKBT in ICIER_0 = 1
· Nack response
RCVD in ICCRA_0 = 1
Disable next receive operation.
mr_data[mr_cnt-2] = ICDRR_0
Save the data.
mr_cnt++
Increment the master receive counter.
4
Reception of the last
byte of data.

5.7.9 Function master_randomread

1. Overview
   Master-reception process which is called by the I²C bus interface interrupt handler. In this case, the interrupt source will be the receive data full interrupt for each byte of received data.
2. Arguments
   None
3. Return value
   None
4. Internal registers used
   The following describes internal registers used in this sample task. The settings are those used in this sample task rather than the initial settings.

- I²C bus control register A_0 (ICCRA_0) Address: H'FFFD58

Bit	Bit Name	Setting	R/W	Function
6	RCVD	1	R/W	Reception Disable Enables or disables the next operation when ICDRR is read while the TRS bit is 0. 0: Enables next operation 1: Disables next operation
5	MST	1	R/W	Master/Slave Select
4	TRS	1		10: Master receive mode 11: Master transmit mode

- I²C bus control register B_0 (ICCRB_0) Address: H'FFFD59

Bit	Bit Name	Setting	R/W	Function
7	BBSY	0/1	R/W	Bus Busy Enables to confirm whether the I²C bus is occupied or released and to issue start and stop conditions in master mode. This bit is set to 1 when the signal output from SDA changes from high to low level under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the signal level output from SDA changes from low to high level under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Write to 0 and 0 to SCP to issue a stop condition. To issue a start/stop condition, use the MOV instruction.
6	SCP	0	R/W	Start Condition/Stop Condition Prohibit Controls the issue of start/stop conditions in master mode. To issue a start condition, write 1 to BBSY and 0 to SCP. A retransmit start condition is issued in the same way. Write 0 to BBSY and 0 to SCP to issue a stop condition. This bit is always read as 1. When 1 is written, the data is not stored
3	SCLO	Undefined	R/W	Monitors signal level output from the SCL pin. When SCLO is read as 1, signal output from SCL is at high level. When SCLO is read as 0, signal output from SCL is at low level.

- I²C bus interrupt enable register_0 (ICIER_0) Address: H'FFFD5B

Bit	Bit Name	Setting	R/W	Function
7	TIE	1	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, enables or disables the transmit-data empty interrupt (TXI) 0: Disables transmit-data empty interrupt request (TXI). 1: Enables transmit-data empty interrupt request (TXI).
6	TEIE	0	R/W	Transmit End Interrupt Enable Enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by cleaning the TEND bit or the TEIE bit to 0. 0: Disables transmit end interrupt request (TEI). 1: Enables transmit end interrupt request (TEI).
5	RIE	1	R/W	Receive Interrupt Enable Enables or disables receive-data full interrupt request (RXI) when a received data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0. 0: Disables receive-data full interrupt request (RXI). 1: Enables receive-data full interrupt request (RXI).
0	ACKBT	0	R/W	Transmit Acknowledge In receive mode, sets the value of the bit to be sent at acknowledge timing here. 0: Outputs 0 at acknowledge timing. 1: Outputs 1 at acknowledge timing.

- I²C bus status register_0 (ICSR_0) Address: H'FFFD5C

Bit	Bit Name	Setting	R/W	Function
7	TDRE	0	R/W	Transmit Data Empty [Setting condition] • Transferring data from ICDRT to ICDRS and making ICDRT empty • Setting the TRS • Issuance of the start condition (including re-transmission) • Making a transition from the receive mode to the transmit mode in the slave mode [Clearing condition] • Writing of 0 to this bit after reading it as 1 • Writing data to ICDRT
6	TEND	0	R/W	Transmit End [Setting condition] • Rising edge of the ninth cycle of SCL while the TDRE flag is 1 [Clearing condition] • Writing of 0 to this bit after reading it as 1 • Writing data to ICDRT
4	NACKF	0	R/W	No Acknowledge Detection Flag [Setting condition] • No detection of acknowledge from the receiver device when transmitting while the ACKE bit in ICIER is 1. [Clearing condition] • Writing of 0 to this bit after reading it as 1

- I²C bus transmit data register_0 (ICDRT_0) Address: H'FFFD5E

  Function: ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the I²C bus shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. When the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. The initial value of ICDRT is H'FF.

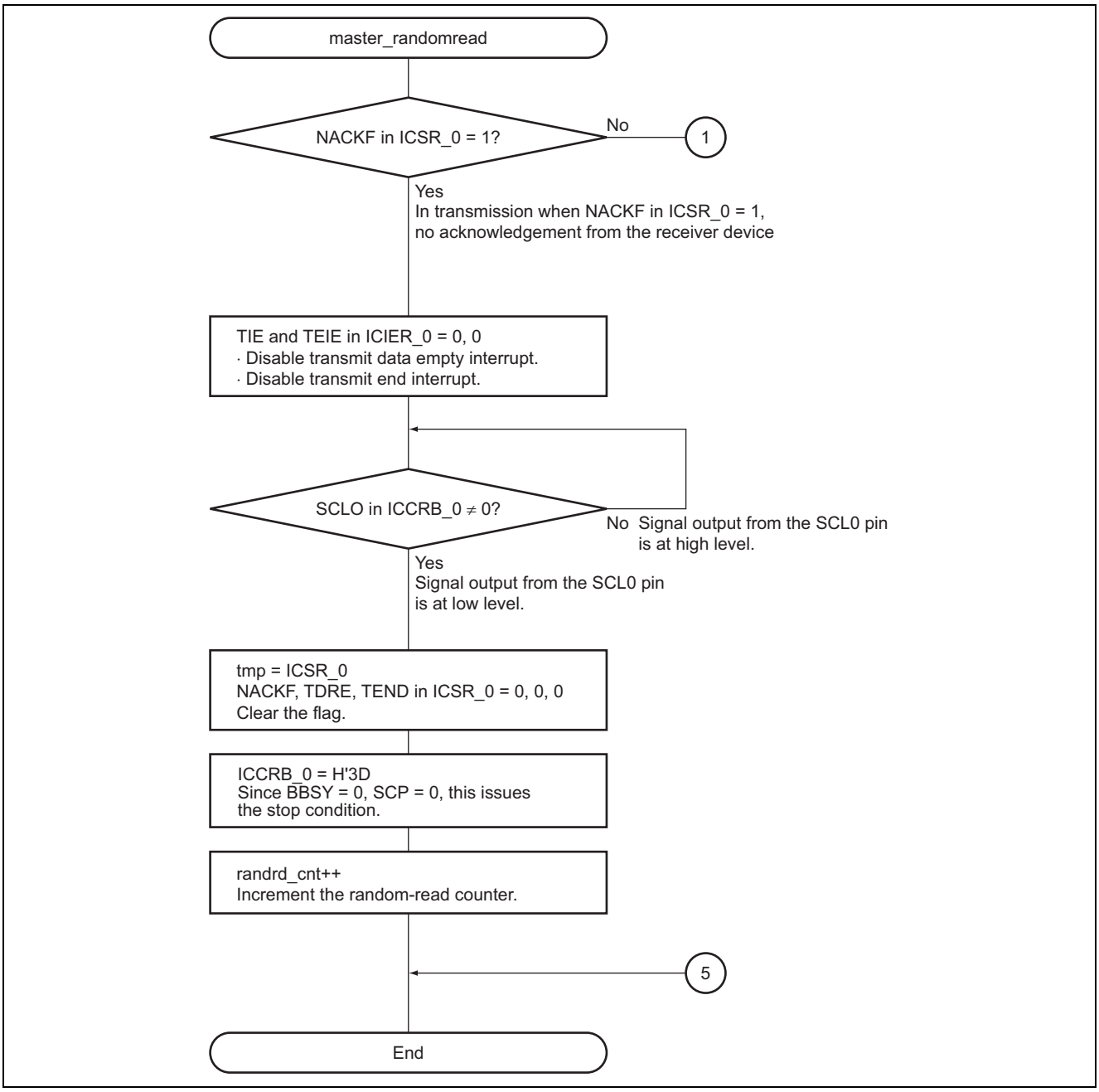
  Setting: MT_ID, MR_ID, mt_data[0]

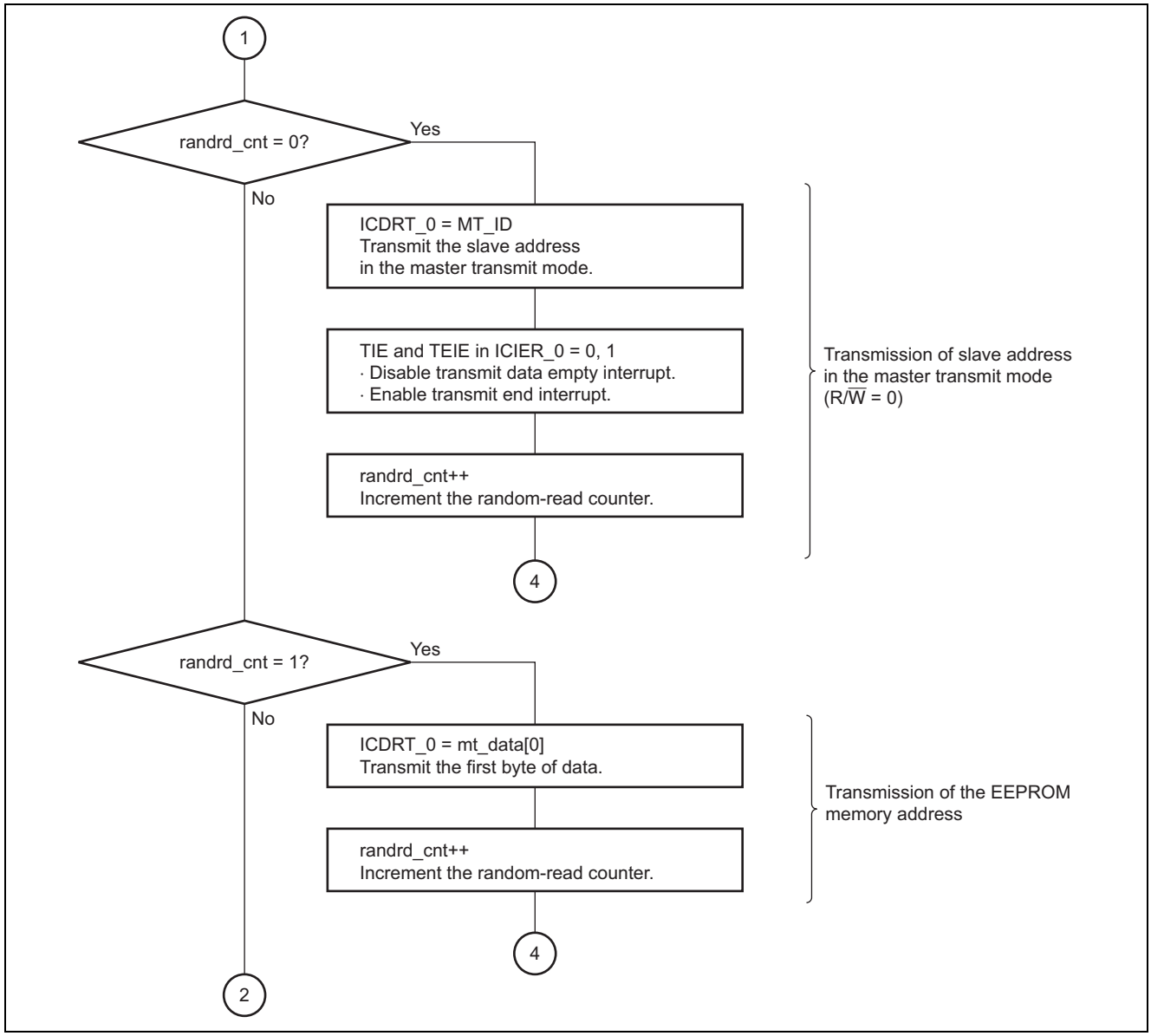
- I²C bus receive data register_0 (ICDRR_0) Address: H'FFFD5F

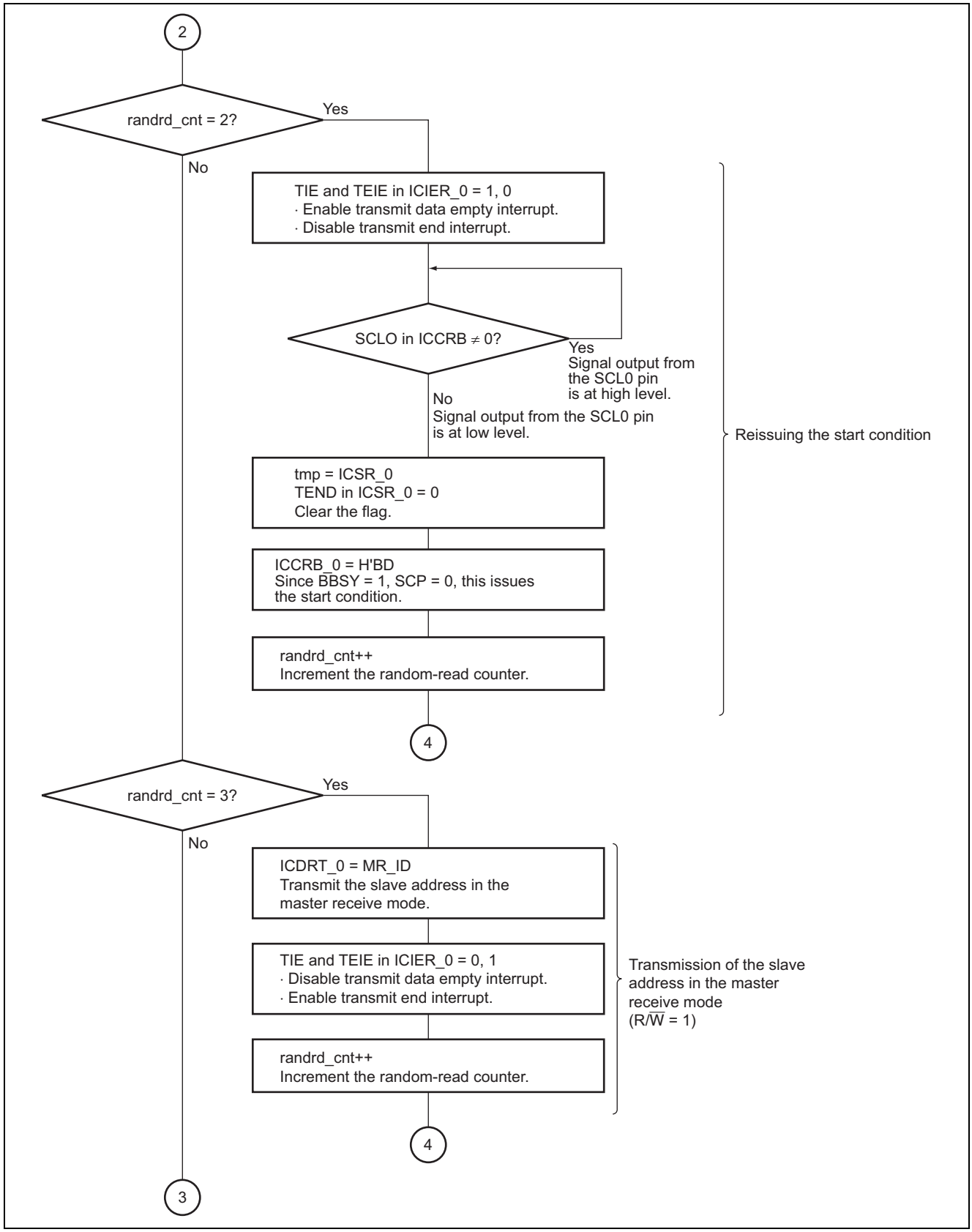
  Function: ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the received data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot be written to this register. The initial value of ICDRR is H'FF.

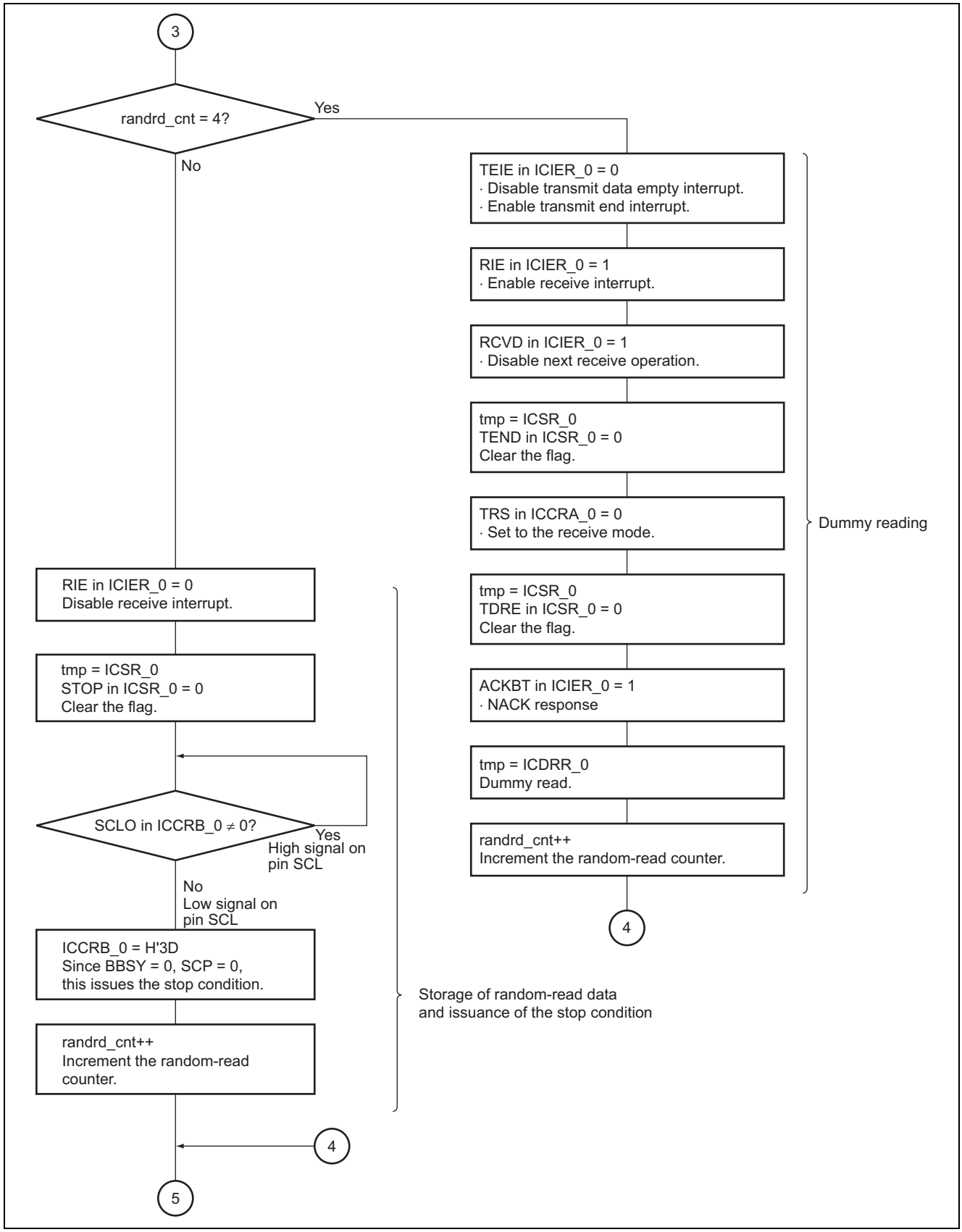
  Setting: Undefined

5. Flowchart


master_randomread
NACKF in ICSR_0 = 1?
No
1
Yes
In transmission when NACKF in ICSR_0 = 1,
no acknowledgement from the receiver device
TIE and TEIE in ICIER_0 = 0, 0
· Disable transmit data empty interrupt.
· Disable transmit end interrupt.
SCLO in ICCRB_0 ≠ 0?
No Signal output from the SCL0 pin
is at high level.
Yes
Signal output from the SCL0 pin
is at low level.
tmp = ICSR_0
NACKF, TDRE, TEND in ICSR_0 = 0, 0, 0
Clear the flag.
ICCRB_0 = H'3D
Since BBSY = 0, SCP = 0, this issues
the stop condition.
randrd_cnt++
Increment the random-read counter.
5
End


1
randrd_cnt = 0?
Yes
No
ICDRT_0 = MT_ID
Transmit the slave address
in the master transmit mode.
TIE and TEIE in ICIER_0 = 0, 1
· Disable transmit data empty interrupt.
· Enable transmit end interrupt.
randrd_cnt++
Increment the random-read counter.
4
Transmission of slave address
in the master transmit mode
(R/W = 0)
randrd_cnt = 1?
Yes
No
ICDRT_0 = mt_data[0]
Transmit the first byte of data.
randrd_cnt++
Increment the random-read counter.
4
Transmission of the EEPROM
memory address
2


2
randrd_cnt = 2?
Yes
No
TIE and TEIE in ICIER_0 = 1, 0
· Enable transmit data empty interrupt.
· Disable transmit end interrupt.
SCLO in ICCRB ≠ 0?
Yes
Signal output from the SCL0 pin is at high level.
No
Signal output from the SCL0 pin is at low level.
tmp = ICSR_0
TEND in ICSR_0 = 0
Clear the flag.
ICCRB_0 = H'BD
Since BBSY = 1, SCP = 0, this issues the start condition.
randrd_cnt++
Increment the random-read counter.
Reissuing the start condition
4
randrd_cnt = 3?
Yes
No
ICDRT_0 = MR_ID
Transmit the slave address in the master receive mode.
TIE and TEIE in ICIER_0 = 0, 1
· Disable transmit data empty interrupt.
· Enable transmit end interrupt.
randrd_cnt++
Increment the random-read counter.
Transmission of the slave address in the master receive mode (R/W = 1)
4
3


3
randrd_cnt = 4?
Yes
No
TEIE in ICIER_0 = 0
· Disable transmit data empty interrupt.
· Enable transmit end interrupt.
RIE in ICIER_0 = 1
· Enable receive interrupt.
RCVD in ICIER_0 = 1
· Disable next receive operation.
tmp = ICSR_0
TEND in ICSR_0 = 0
Clear the flag.
TRS in ICCRA_0 = 0
· Set to the receive mode.
tmp = ICSR_0
TDRE in ICSR_0 = 0
Clear the flag.
ACKBT in ICIER_0 = 1
· NACK response
tmp = ICDRR_0
Dummy read.
randrd_cnt++
Increment the random-read counter.
4
Dummy reading
RIE in ICIER_0 = 0
Disable receive interrupt.
tmp = ICSR_0
STOP in ICSR_0 = 0
Clear the flag.
SCLO in ICCRB_0 ≠ 0?
Yes
High signal on pin SCL
No
Low signal on pin SCL
ICCRB_0 = H'3D
Since BBSY = 0, SCP = 0,
this issues the stop condition.
randrd_cnt++
Increment the random-read counter.
4
5
Storage of random-read data
and issuance of the stop condition

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Jan.31.07	—	First edition issued
2.00	Sep.25.07	2, 3, 10, 12 to 14, 16, 18, 19, 21, 25, 26, 28 to 30, 33, 34, 39, 44, 47, 50, 50, 54, 56, and 57	Corrections on content and program source due to an additional item, idle mode

Notes regarding these materials

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

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