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.

H8SX Family

Flash Memory Reprogramming in the User Program Mode for H8SX/1657F

Introduction

This application note describes the reprogramming of flash memory (the user MAT) via a clock synchronous communications interface in the use program mode of H8SX/1657F, and mainly concerns the slave side.
That is, unless otherwise specified, the descriptions are related to the slave side.

Target Device

H8SX/1657F

Contents

1. Specification 2
2. Applicable Conditions 7
3. Description of Functions Used 8
4. Description of Operation 11
5. Description of Software for Normal Program on the Receive Side (Slave) 23
6. Description of Software for Programming/Erasing Procedure Program on the Slave Side 30
7. Description of Software for the Clock Synchronous Serial Communications Program on the Slave Side 50

1. Specification

1.1 Outline

Specifications, which are highly reliable for users to reprogram the on-chip flash memory, are provided for H8SX/1657F. The use starts by downloading the erasing or programming module to the RAM and then calls subroutines to erase/program the flash memory.

In this sample task, data for reprogramming are placed in the flash memory on the master side and sent to the slave side by clock synchronous communications. The following procedure is then applied to reprogram the flash memory on the slave side. A sample configuration for on-board reprogramming is illustrated in figure 1, and the connections for clock synchronous communications between the master and slave are illustrated in figure 2.

- Power-on resets are applied, and the slave and master boot up in the user mode.
- When master side switch 0 (SW0) is turned on, the master side sends the reprogramming command to the slave side, and the slave side reprograms its own flash memory.
- The data in the flash memory is reprogrammed by the user program mode.
- The master side sends the data to be reprogrammed from its own flash memory to the slave side.
- The data for reprogramming are transferred in clock synchronous mode via serial communications interface (SCI) 0. The data is transmitted from the master side and received on the slave side.
- On both the master and slave sides, PD7 is low and PD6 is high while the flash memory is being reprogrammed, and PD7 is high and PD6 is low on completion of reprogramming.

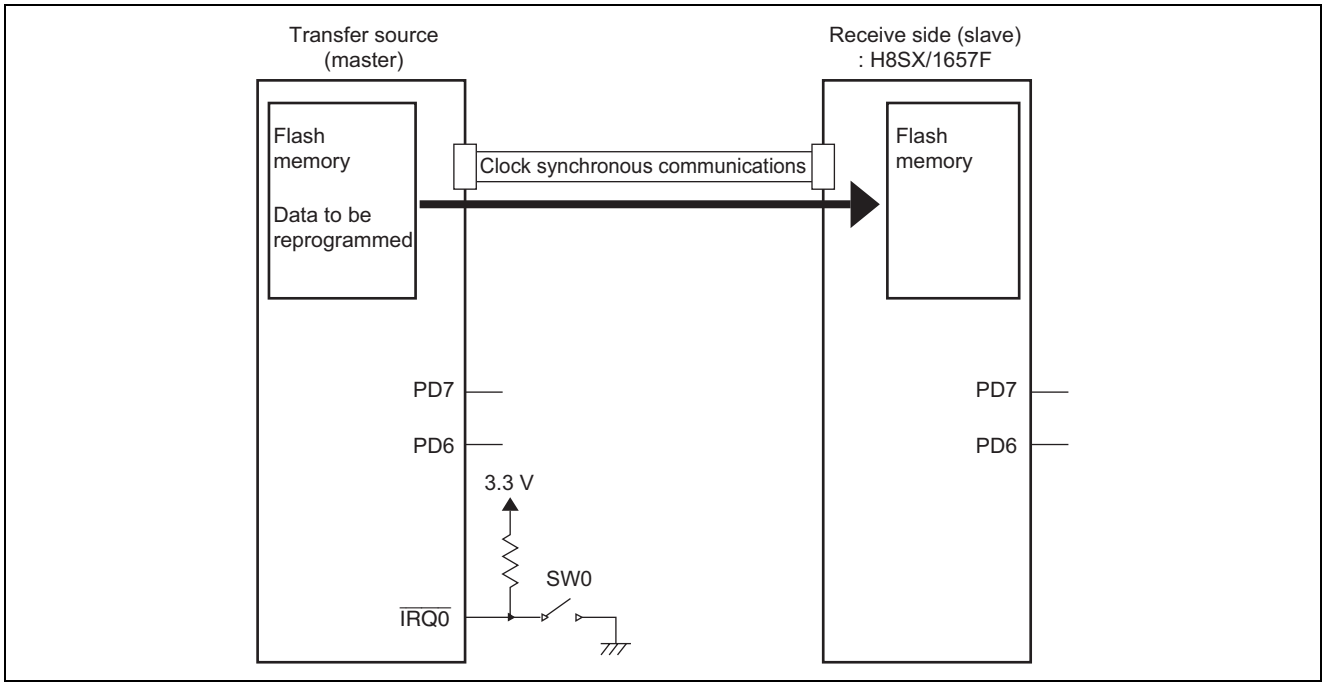


Figure 1 Sample Configuration for On-Board Reprogramming

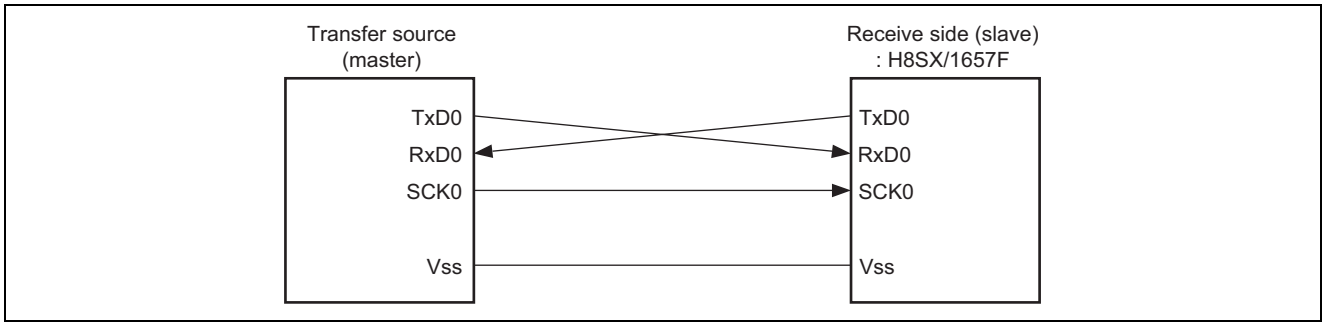


Figure 2 Connections for Clock Synchronous Communications between the Master and Slave

1.2 User Program Mode

The RAM transfer program, clock synchronous communications program, and programming/erasing procedure program must be written to the user MAT in advance.

1.3 Specification for Communications

The specification for communications between the master and slave in this sample task are listed below.

1.3.1 Type of Communications

Table 1 Type of Communications

Item	Setting
Transfer rate	2.5 Mbps
Type	Clock synchronous communications
Data bit	8 bits (1 byte)

1.3.2 Communications Command

Table 2 Communications commands

Command Name	Constant Name	Command Data
Start reprogramming	FSTART	H'10
Erase	ERASE	H'11
Write	WRITE	H'12
Read status	STATUSREAD	H'13
128-byte transmission request	TRS128	H'14

1.3.3 State Indicator

Table 3 State Indicator

Statue Name	Constant Name	Status Data
Normal	OK	H'A5
Erase command error	ER_ECMD	H'C1
Erase download error	ER_EDWNLD	H'C2
Erase initialization error	ER_EINIT	H'C3
Erase error	ER_ERASE	H'C4
Program command error	ER_WCMD	H'A1
Program download error	ER_WDWNLD	H'A2
Program initialization error	ER_WINIT	H'A3
Program error	ER_WRITE	H'A4

1.3.4 Specifying Blocks to be Erased

The erase command “ERASE” is sent from the master to the slave and is immediately followed by an indicator of the block numbers of blocks to be erased. The communications format for block erasure is given in table 4. Block settings are made in a 4-byte (32-bit) unit where bits 19 to 0 correspond to blocks 19 to 0. Since bits 31 to 20 are not used, always set them to 0. An example of the data transmitted to erase block 19 is given in table 5.

Table 4 Correspondence of Blocks to be Erased

Bit	Erased Block	Setting	Description
31 to 20	No used	0 fixed	Not used. Set 0.
19 to 0	EB11 to EB0	0/1	0: The corresponding block is not erased. 1: The corresponding block is erased.

Table 5 Example of Data for Transmission to Erase Block 19

	1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte
Item	Erase Command (ERASE)	Erase block			
Byte	H'11	H'00	H'08	H'00	H'00
Bit	0 0 0 1 0 0 0 1	0 0 0 0 0 0 0 0	0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
	MSB LSB	MSB LSB	MSB LSB	MSB LSB	MSB LSB

Note: The data is transmitted in byte units, with the LSB first.

1.4 Memory Map

The memory map for this sample task is given as figure 3.

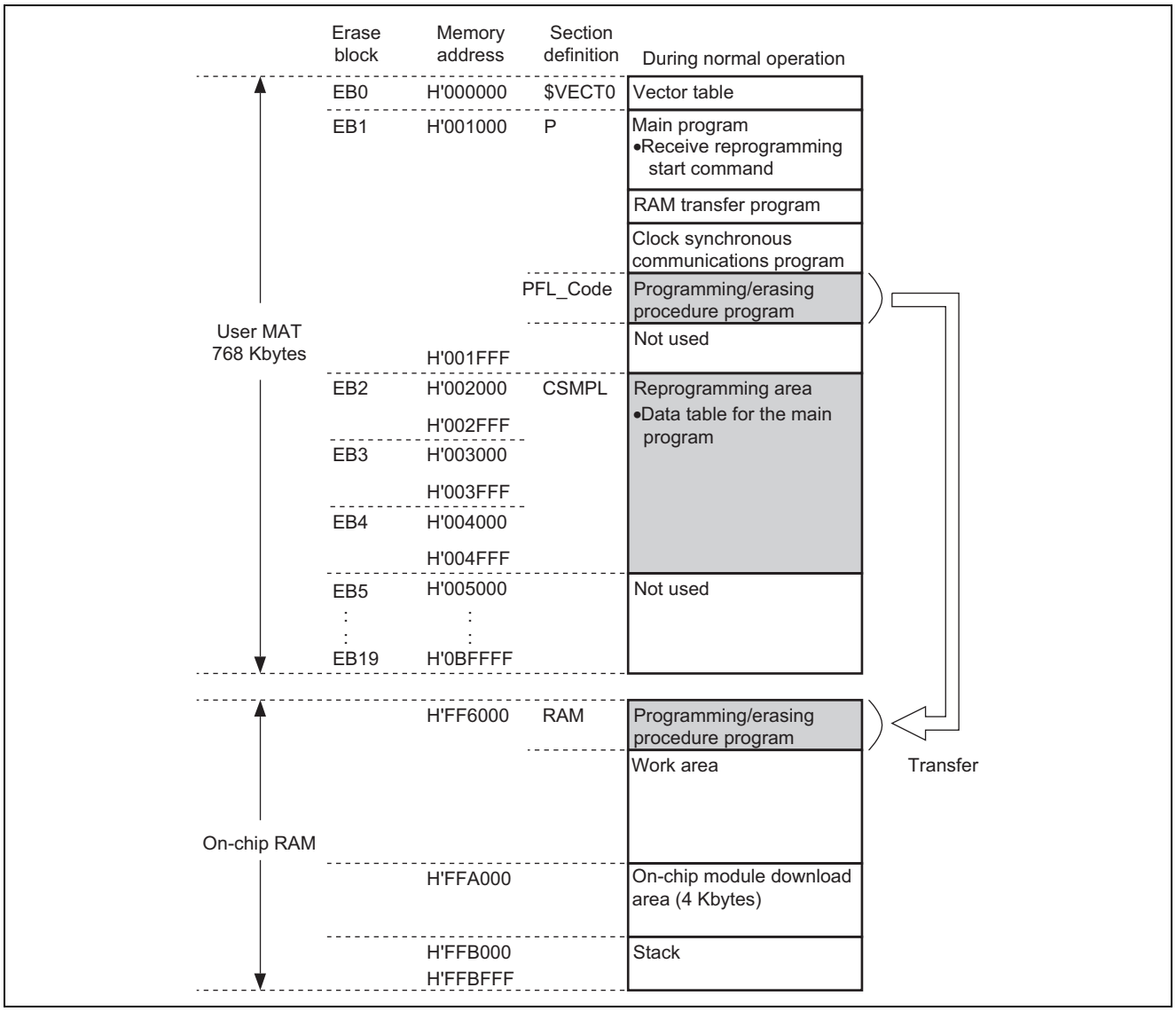


Figure 3 Memory Map

2. Applicable Conditions

Table 6 Applicable Conditions

Item	Description
Operating Frequency	Input clock: 17.5 MHz System clock (Iϕ): 35 MHz Peripheral module clock (Pϕ): 35 MHz External bus clock (Bϕ): 35 MHz
Operating mode	Mode 7 (MD2 = 1, MD1 = 1, MD0 = 1)
On-board programming mode	User program mode
Development tool	High-performance Embedded Workshop Ver.4.01.01
C/C++ compiler	H8S, H8/300 Series C/C++ Compiler Ver.6.01.02 (manufactured by Renesas Technology)
Compiler options	-cpu = h8sxa:24:md, -code = machinecode, -optimize = 1, -regparam = 3 -speed = (register, shift, struct, expression)

Table 7 Section Setting

Address	Section Name	Description
H'001000	P	Program area
	PFL_Code	Areas for storing programming/erasing procedure program
H'002000	CSMPL	Area for reprogramming
H'FF6000	RAM	Area for transferring programming/erasing procedure program

3. Description of Functions Used

3.1 User Program Mode

3.1.1 User MAT and User Boot MAT

The on-chip flash memory consists of the two memory units (memory MATs) listed in table 8. Both are allocated to the same address. The user program mode is an arbitrary boot program that suits the user system, and the user MAT can be programmed/erased.

Table 8 Memory MATs

Memory MAT	Activation	Amount of Memory Used
User MAT	Activates when a power-on reset is performed in the user mode.	768 Kbytes
User Boot MAT	Activates when a power-on reset is performed in the user boot mode.	8 Kbytes

3.1.2 Downloading the On-Chip Program

Programming/erasing of the flash memory on this LSI is done by first downloading the on-chip module for either programming/erasing to the on-chip RAM and then running the individual program.

The destination address for downloading is determined by the setting of the download destination specification register (FTDAR). For the RAM address map after downloading, refer to figure 4. As the figure shows, the on-chip RAM area is the destination for downloading of programming/erasing programs. The corresponding program must be downloaded to the RAM area indicated by FTDAR prior to the required processing.

During programming/erasing, take care to ensure that the download area does not overlap with an area in use by the user.

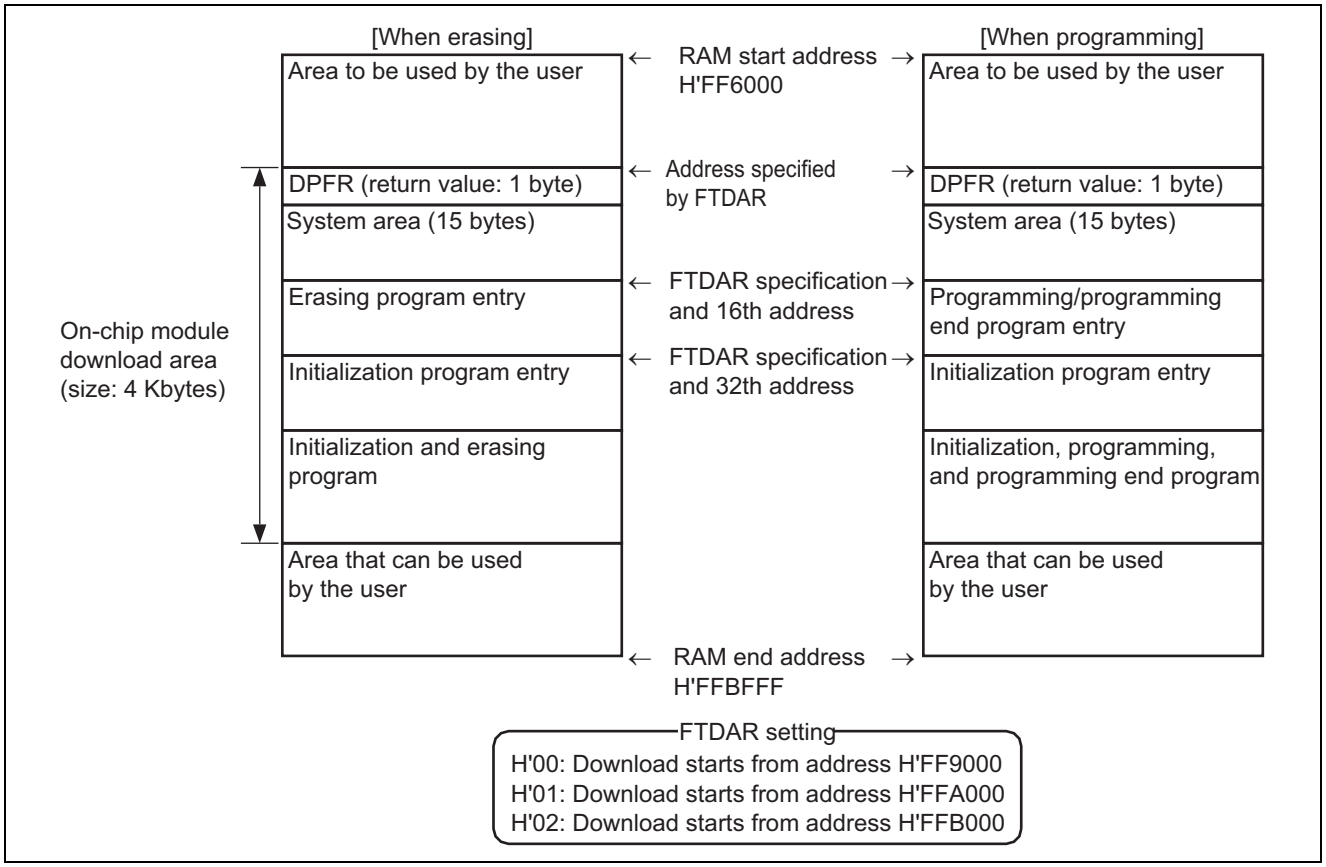


Figure 4 RAM Map for Programming/Erasing

3.2 Block Configuration

Erase blocks of the user MAT are listed in table 9.

Table 9 User MAT Erase Blocks

Block	Unit of Erasure	Addresses
EB0	4 Kbytes	H'000000 to H'000FFF
EB1	4 Kbytes	H'001000 to H'001FFF
EB2	4 Kbytes	H'002000 to H'002FFF
EB3	4 Kbytes	H'003000 to H'003FFF
EB4	4 Kbytes	H'004000 to H'004FFF
EB5	4 Kbytes	H'005000 to H'005FFF
EB6	4 Kbytes	H'006000 to H'006FFF
EB7	4 Kbytes	H'007000 to H'007FFF
EB8	32 Kbytes	H'008000 to H'00FFFF
EB9	64 Kbytes	H'010000 to H'01FFFF
EB10	64 Kbytes	H'020000 to H'02FFFF
EB11	64 Kbytes	H'030000 to H'03FFFF
EB12	64 Kbytes	H'040000 to H'04FFFF
EB13	64 Kbytes	H'050000 to H'05FFFF
EB14	64 Kbytes	H'060000 to H'06FFFF
EB15	64 Kbytes	H'070000 to H'07FFFF
EB16	64 Kbytes	H'080000 to H'08FFFF
EB17	64 Kbytes	H'090000 to H'09FFFF
EB18	64 Kbytes	H'0A0000 to H'0AFFFF
EB19	64 Kbytes	H'0B0000 to H'0BFFFF

3.3 Serial Communications Interface

The SCI operates in the clock synchronous mode. It is used for command-related communications between the master and slave to transfer the data for reprogramming.

4. Description of Operation

4.1 User MAT Reprogramming Procedure

User MAT reprogramming procedure in the user program mode is given in figure 5.

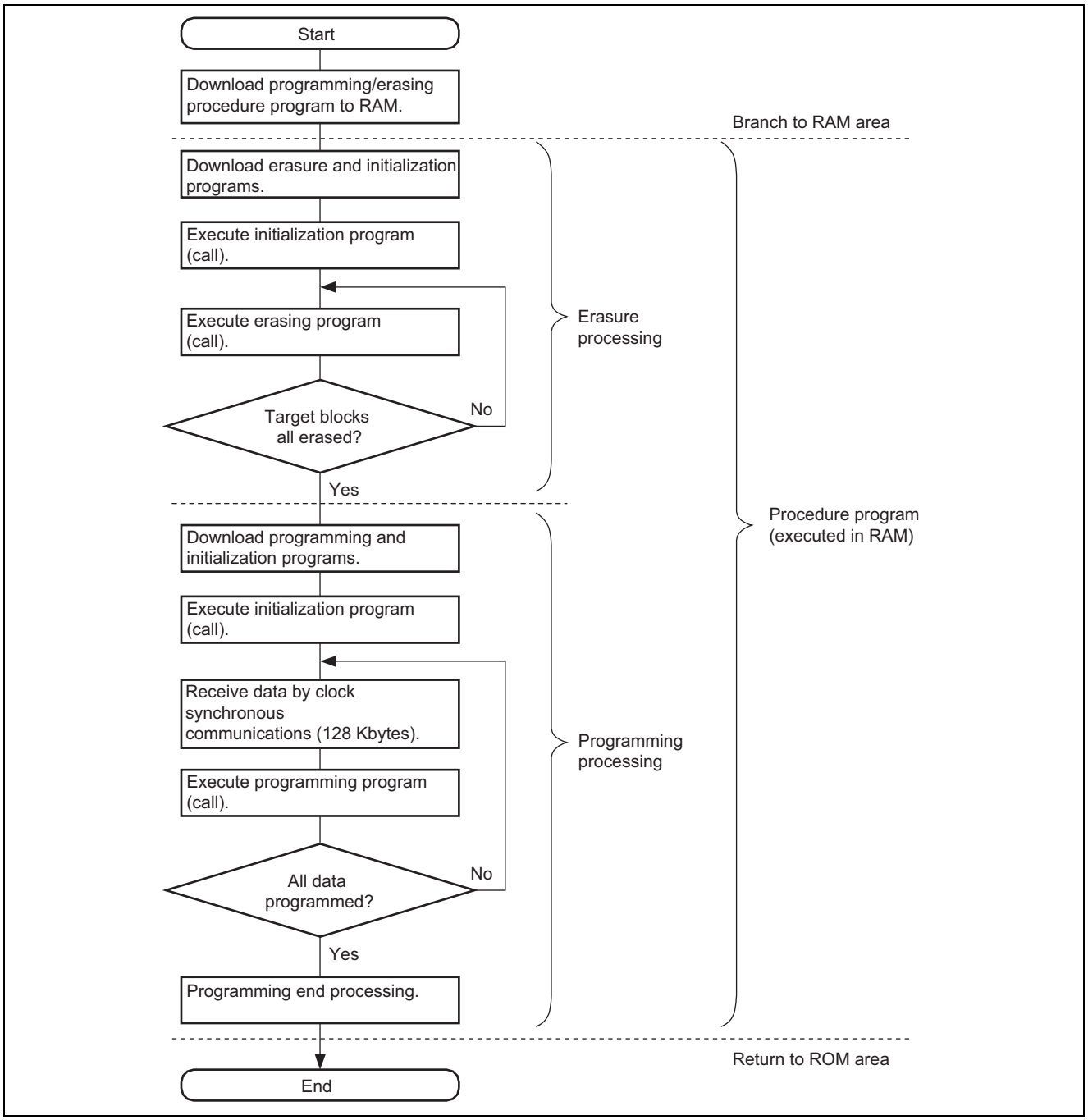


Figure 5 User MAT Reprogramming Procedure

4.2 Operational Overview

4.2.1 Normal Operation

1. The normal application accesses the data table on the user MAT. The data table is received by the master side and is reprogrammed.
2. The RAM transfer program, clock synchronous communications program, and the programming/erasing procedure program are already reprogrammed on the slave side user MAT.
3. Data communications between the master and slave are done in clock synchronous communications.

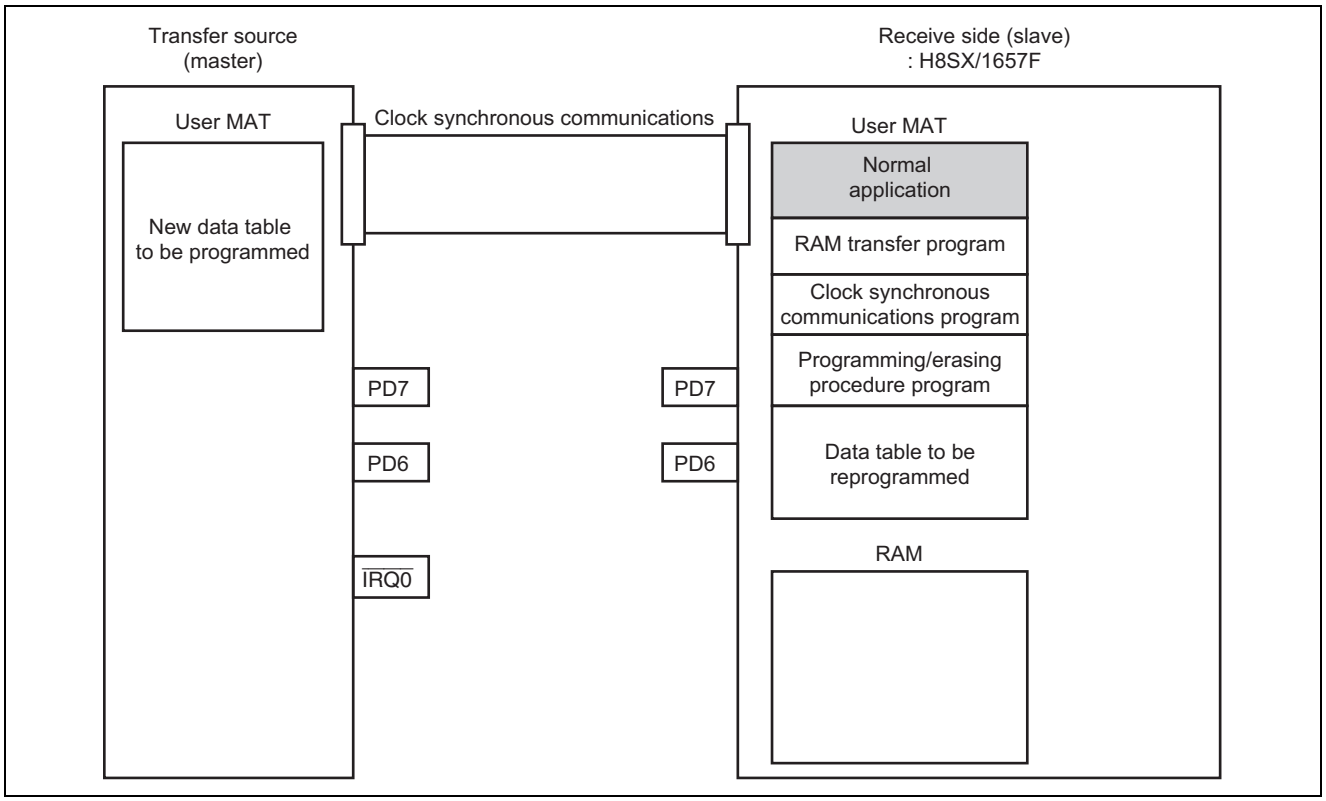


Figure 6 Normal Operation

4.2.2 Preparation of On-Board Reprogramming

(1) When a low level is input to the $\overline{\text{IRQ0}}$ pin on the master side as a trigger, the master sends the reprogramming start command “FSTART”.

(2) Here, PD7 is low and PD6 is high on the master side.

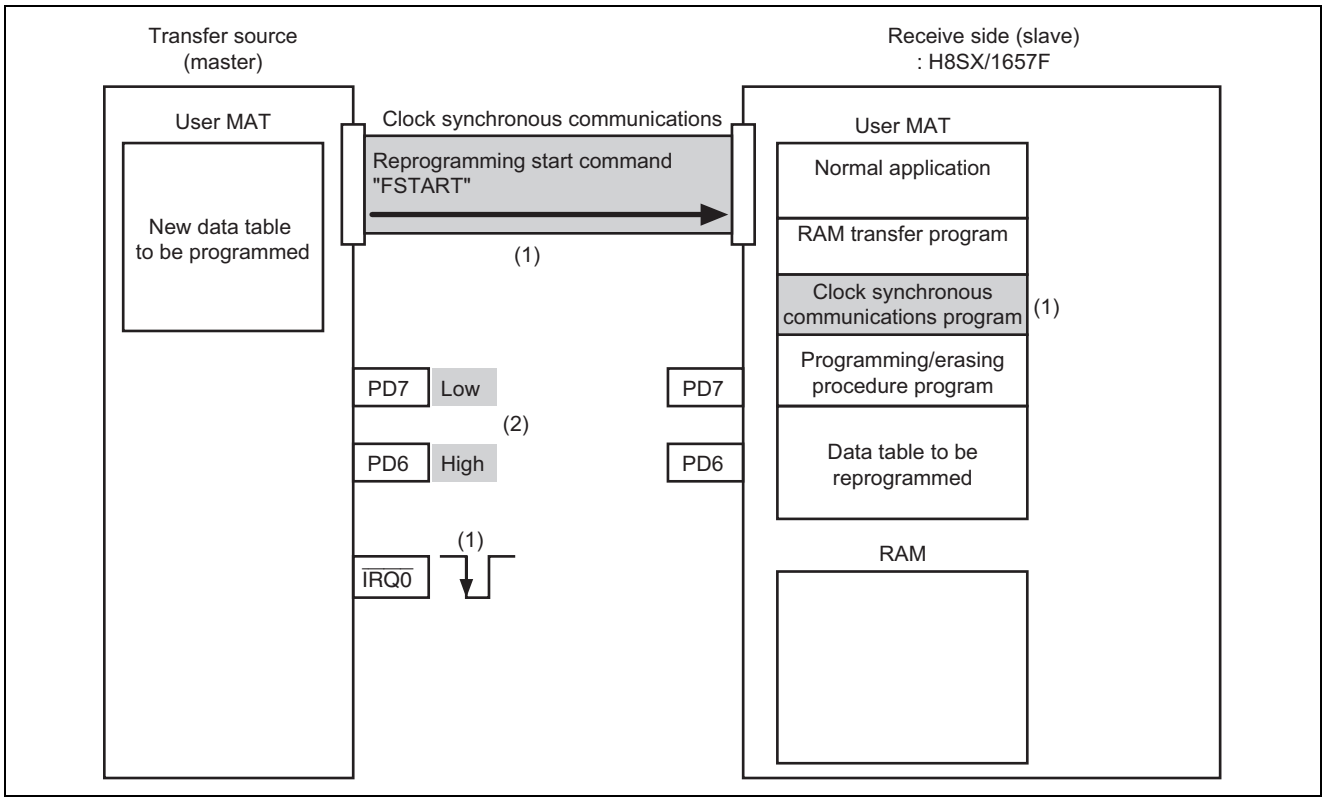


Figure 7 Preparation of On-Board Reprogramming

4.2.3 Start of On-Board Reprogramming

(1) On receiving the reprogramming start command, the slave side initiates the RAM transfer program, and sends the programming/erasing procedure command to the on-chip RAM.

(2) Here, PD7 is low and PD6 is high on the master side.

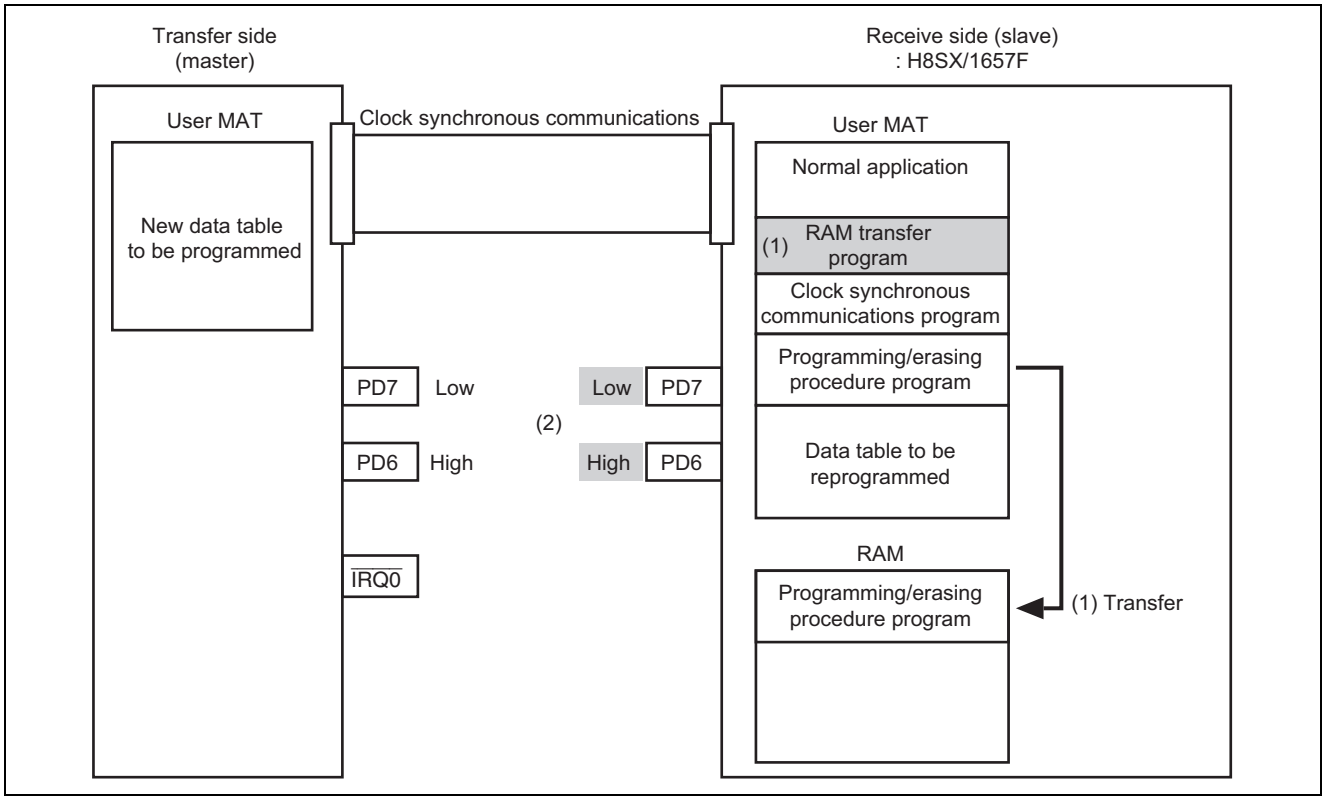


Figure 8 Start of On-Board Reprogramming

4.2.4 Activating the Programming/Erasing Procedure Program

(1) After the transfer programs have been loaded to RAM, the control branches to the programming/erasing procedure program in RAM.

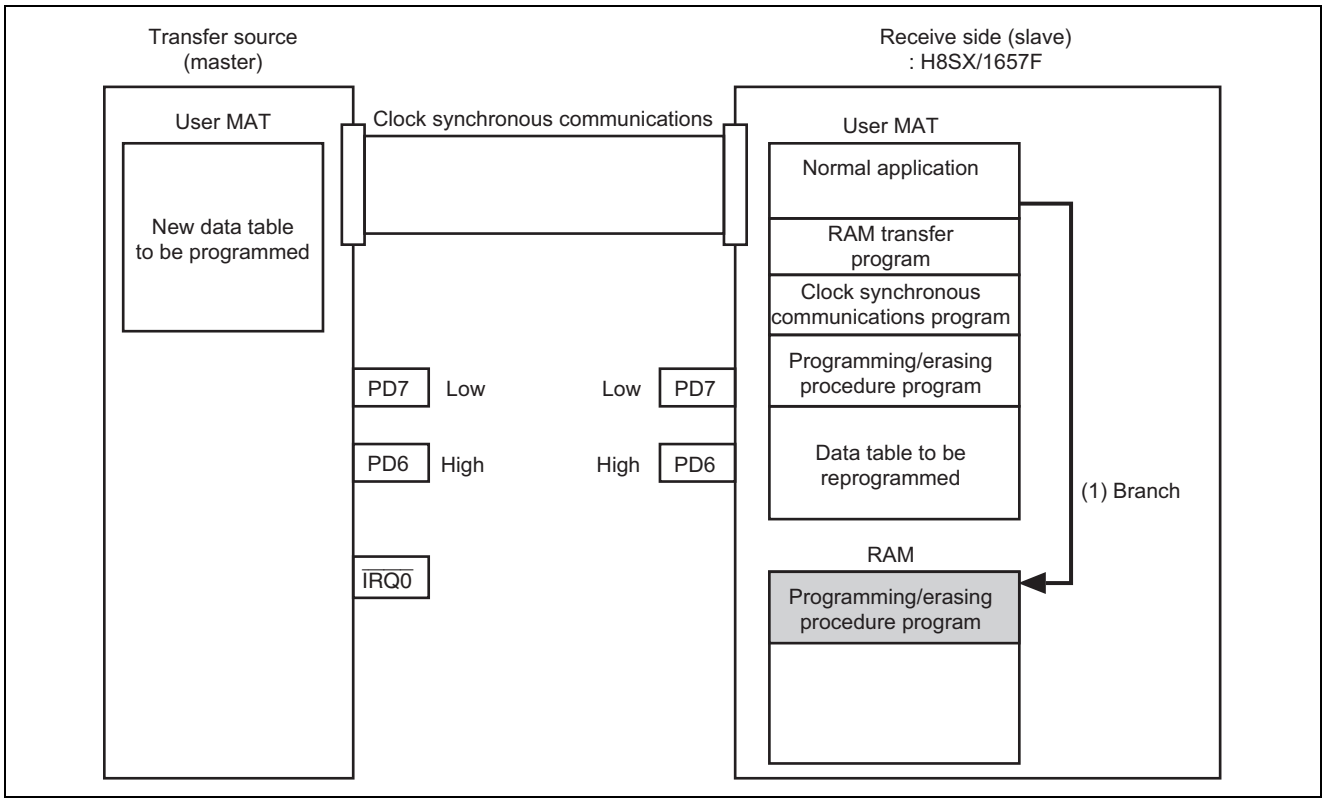


Figure 9 Activation of Programming/Erasing Procedure Program

4.2.5 Erasing User MAT

(1) The master sends the erasing command “ERASE”.
(2) Registers for controlling the flash memory are set (Both the EPVB bit of the FECS register and the SCO bit of the FCCS register to 1), and the initialization program and the erasure program are downloaded.
(3) The initialization program is executed.
(4) The erasing program is executed, and the target block for erasure on the user MAT is erased.

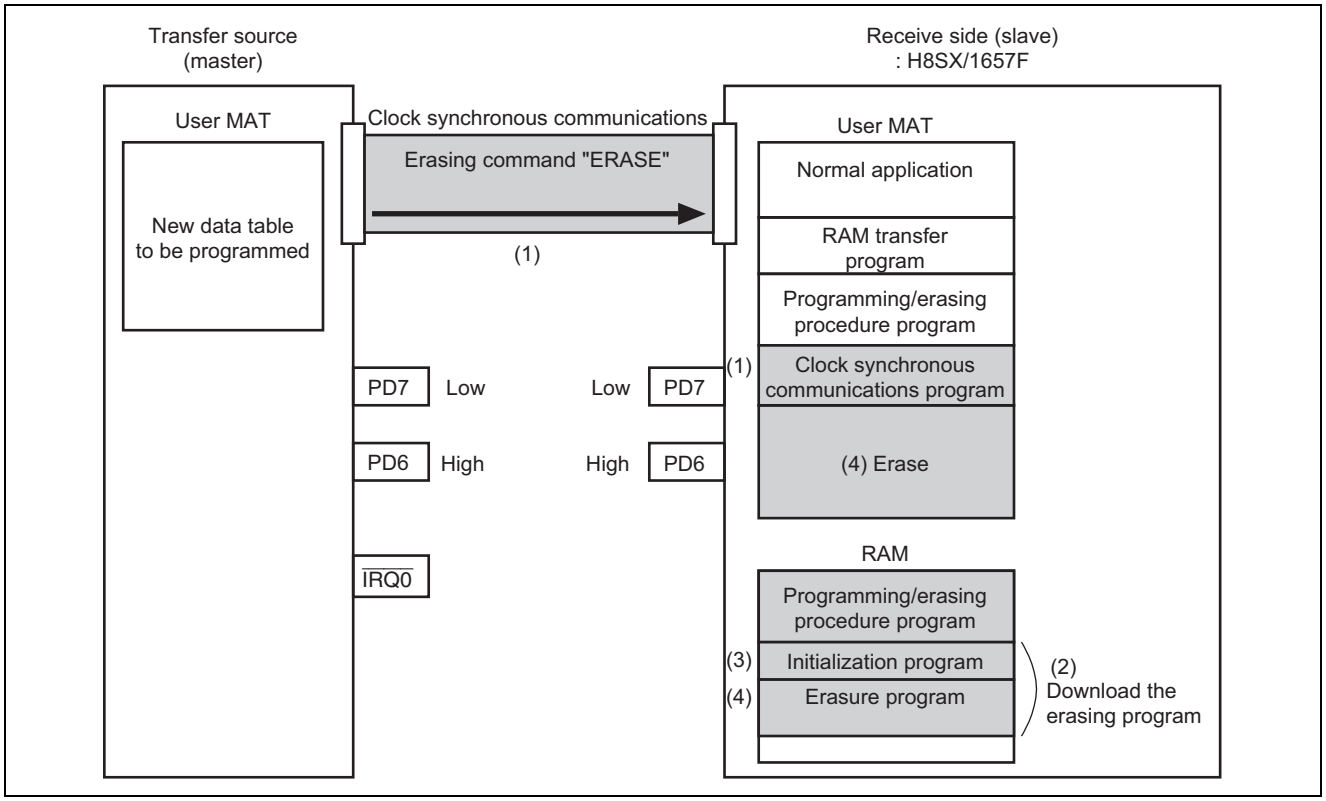


Figure 10 Erasing User MAT

4.2.6 Programming User MAT

(1) The master sends the programming command “WRITE”.
(2) Registers for controlling the flash memory are set (Both the PPVS bit of the FPCS register and the SCO bit of the FCCS register to 1), and initialization program and the programming program are downloaded.
(3) The initialization program is executed.
(4) The following a. and b. are repeated until all data on the master side is programmed on the slave side.
   a. The receive side (slave) receives 128 bytes of new data from the transfer source (master).
   b. The receive side (slave) executes the programming program, and writes 128 bytes of data to the user MAT.
(5) On completion of programming, PD7 is high and PD6 is low on both the master and slave sides.

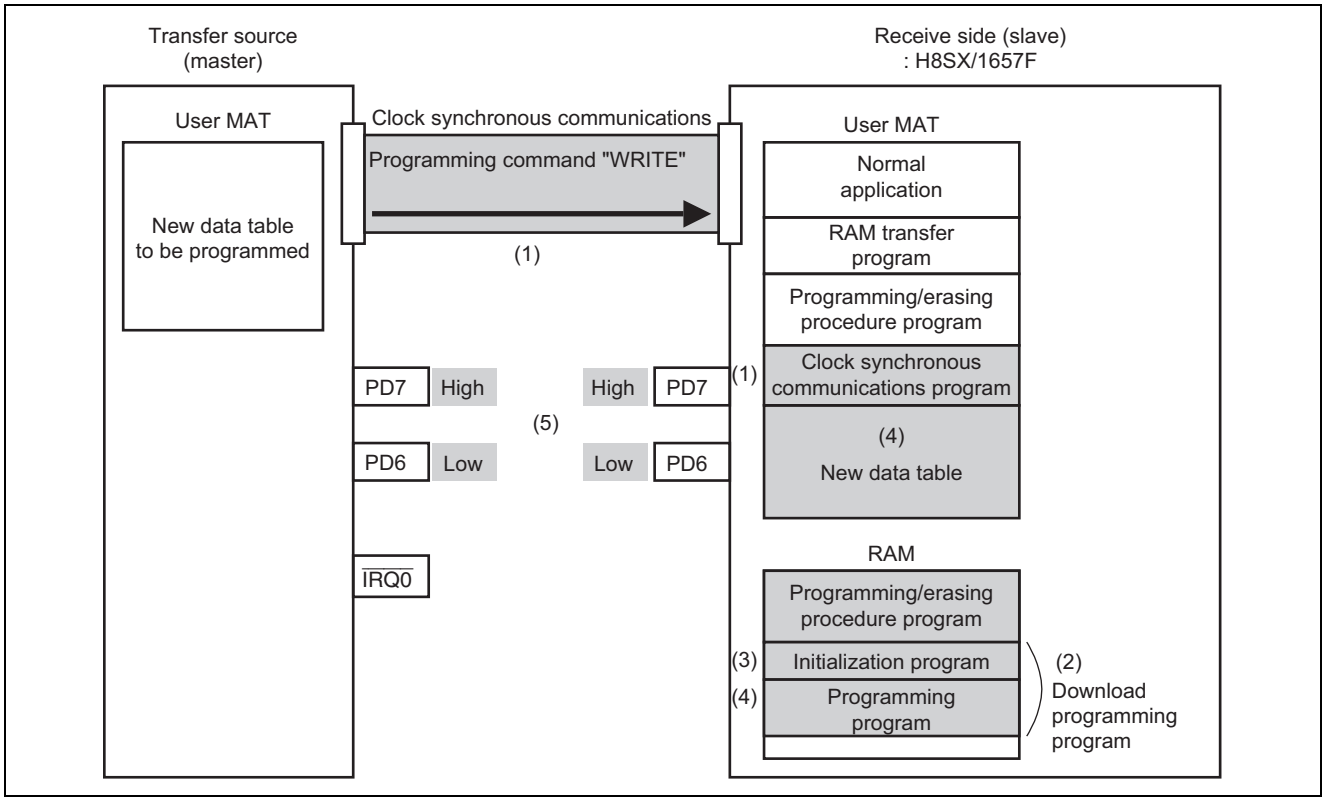


Figure 11 Programming User MAT

4.2.7 Activating the Program

(1) After a reset, the normal application that accesses the new data table is activated.

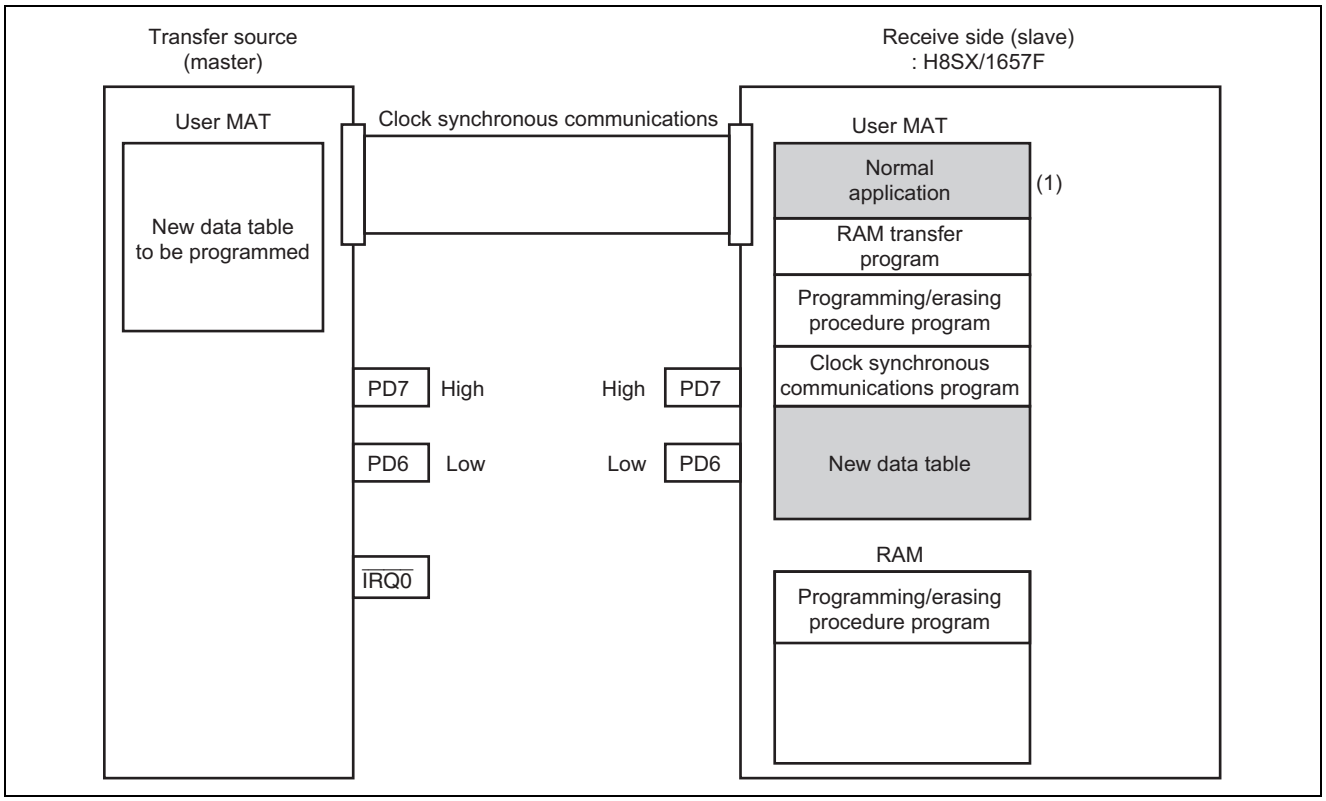


Figure 12 Activating the Program

4.3 Sequence Diagram

4.3.1 Normal Operation

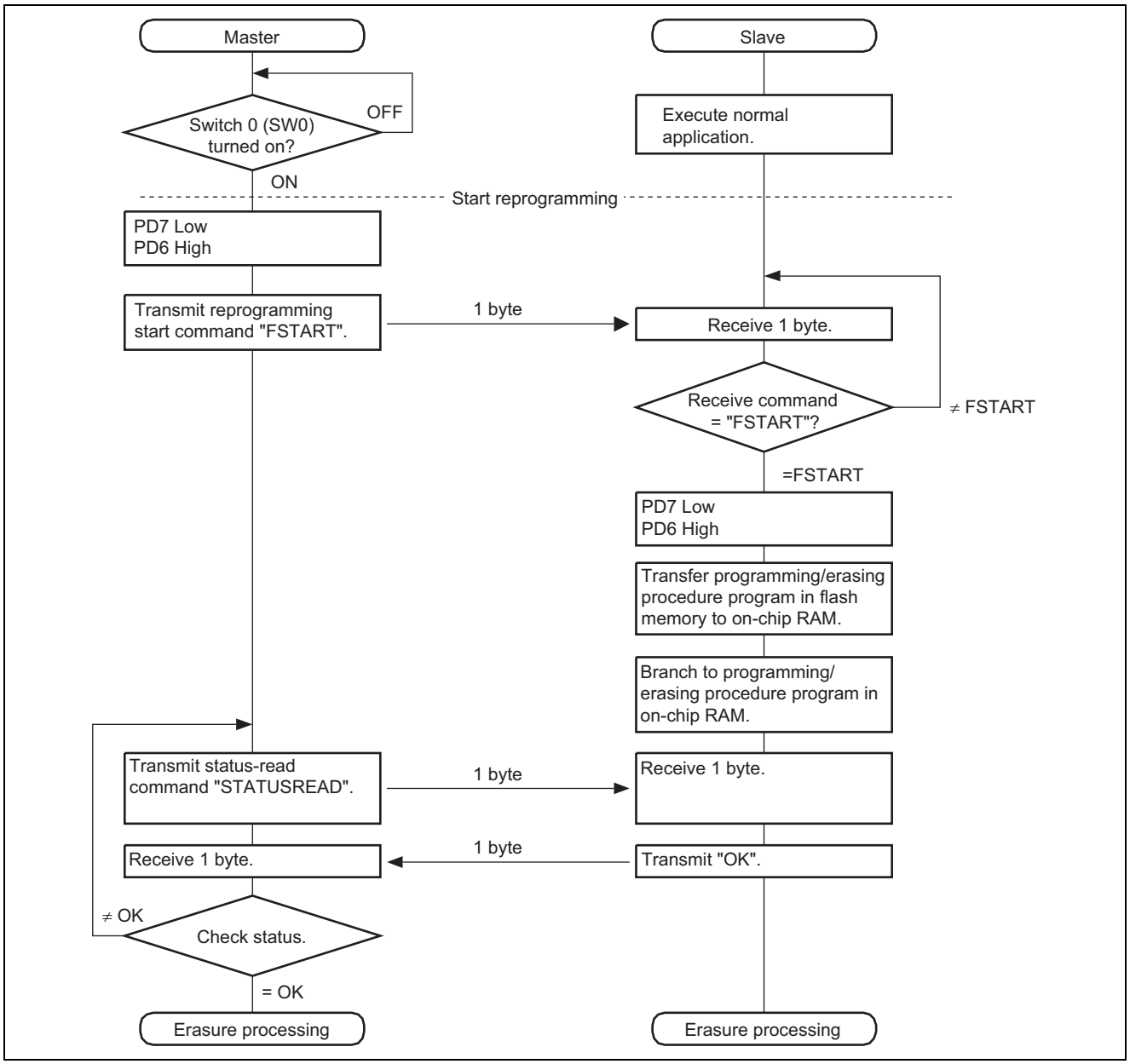


Figure 13 Normal Operation

4.3.2 Erasure Processing

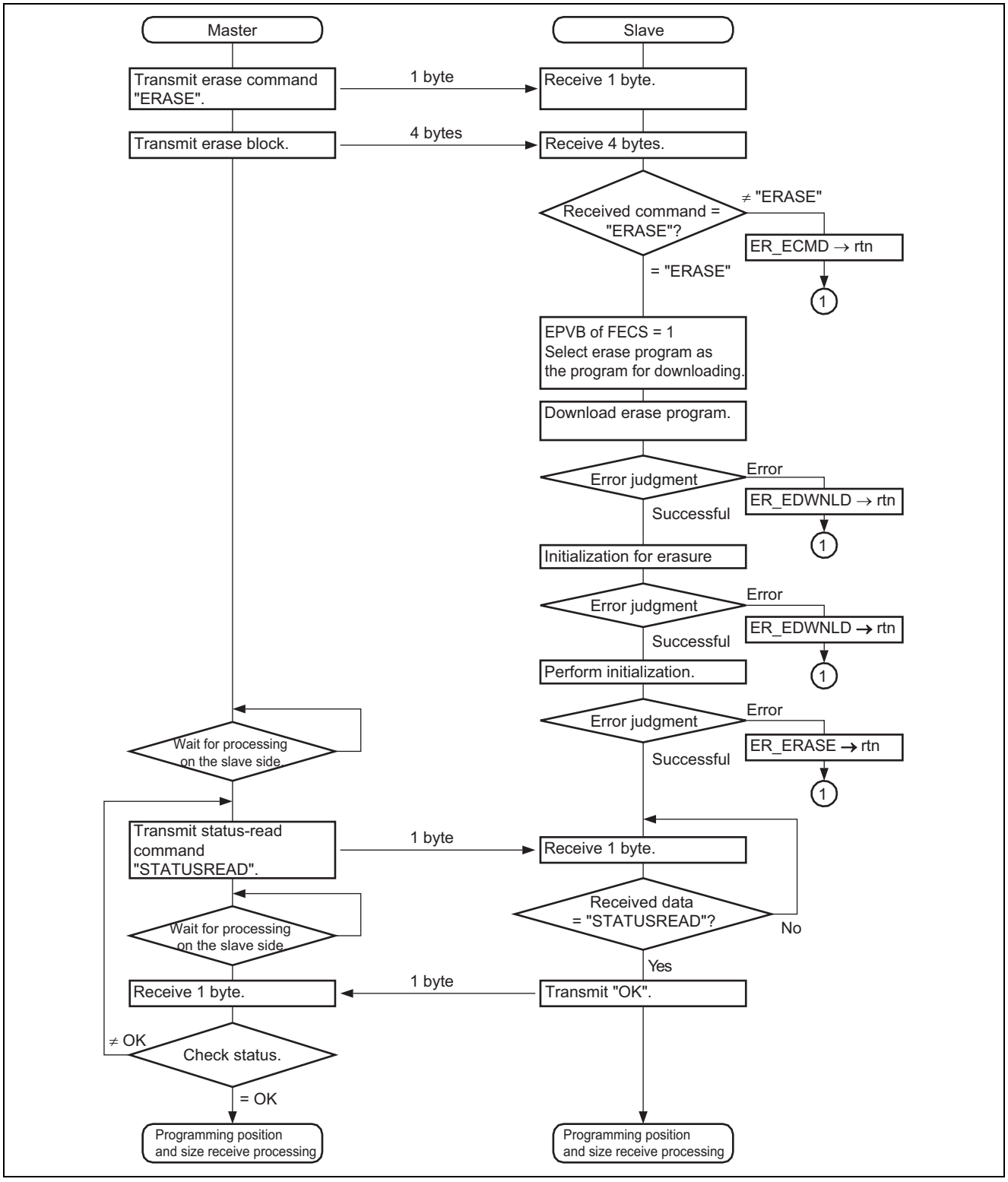


Figure 14 Erasure processing

4.3.3 Processing to Receive the Position and Range for Programming

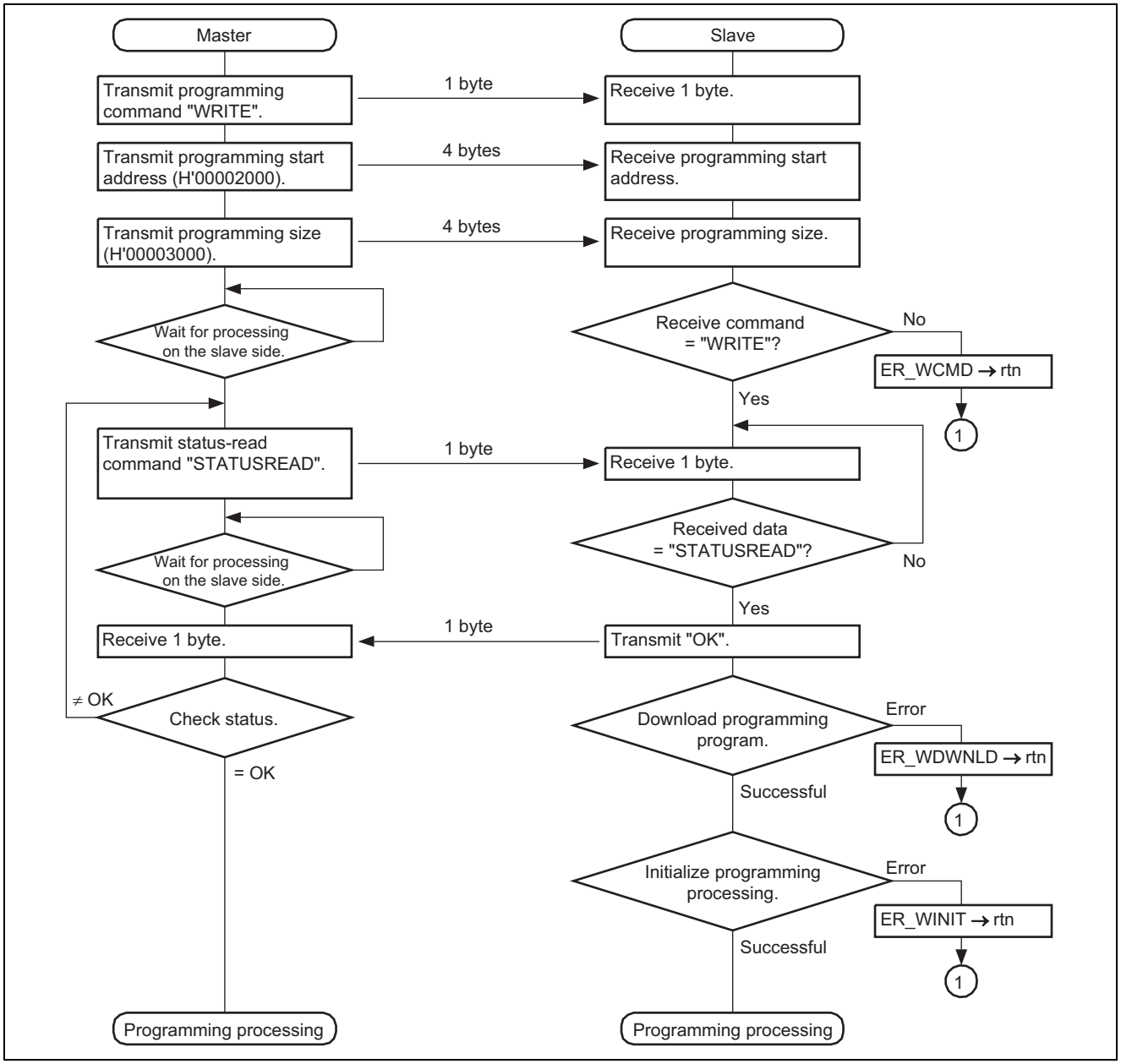


Figure 15 Processing to Receive the Position and Range for Programming

4.3.4 Programming Processing

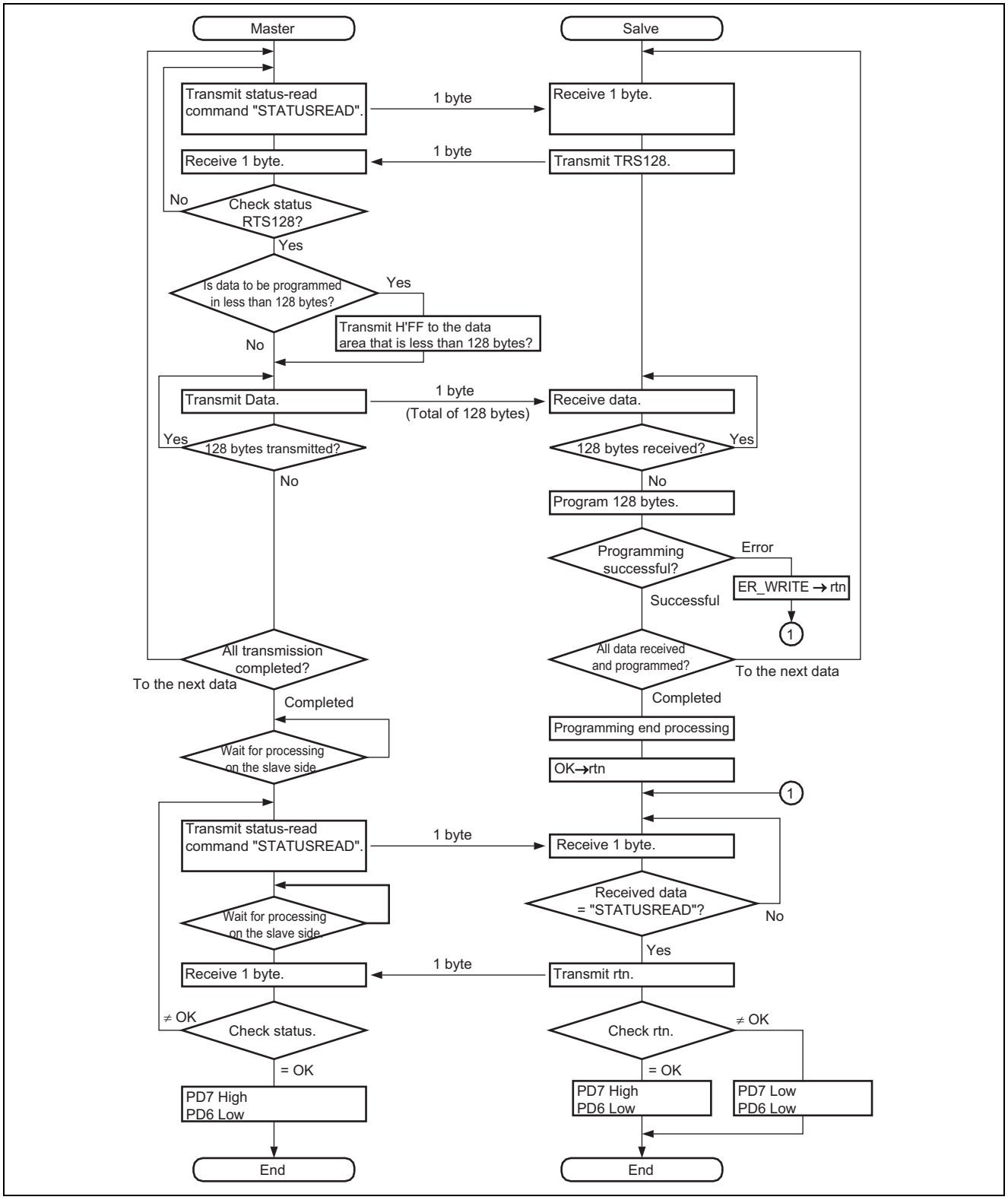


Figure 16 Programming Processing

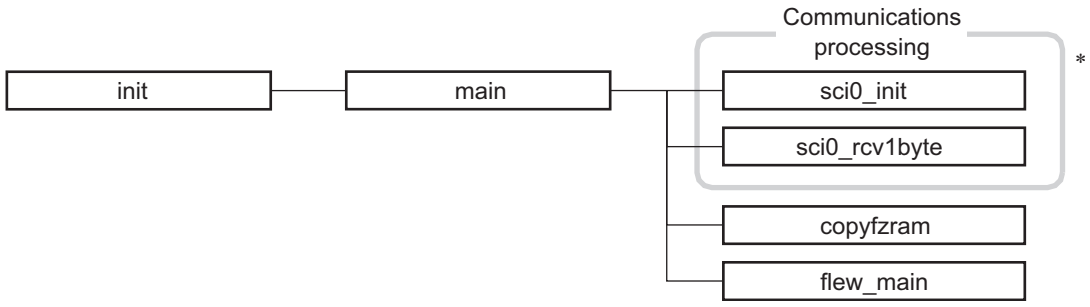
5. Description of Software for Normal Program on the Receive Side (Slave)

5.1 List of Functions

The normal program (main.c) on the receive side (slave) is used to execute the user application program (normal application), receive the reprogramming start command, and transfer the programming/erasing procedure program to the on-chip RAM. A list of functions used in the normal program on the receive side (slave) is given in table 10, and the hierarchical structure of calls is given in figure 17.

Table 10 Functions in the Normal Program on the Receive Side (Slave)

Function Name	Description
init	Initialization routine Releases from the module stop mode, makes clock settings, and calls the main function.
main	Main routine Executes the normal application program, receives the reprogramming start command, and transfers the programming/erasing procedure to the on-chip RAM.
copyfzram	Transfers the programming/erasing procedure program from the user MAT to the on-chip RAM.
flew_main	Programming/erasing procedure program in the user MAT.



* Called from the clock synchronous serial communications program

Figure 17 Normal Program on the Slave Side

5.2 Description of Functions

5.2.1 init Function

1. Functional overview

Initialization routine, releases the module stop mode, sets the clock, and calls the main function.

2. Arguments

None

3. Return value

None

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Mode control register (MDCR) Number of bits: 16 Address: H'FFFDC0

Bit	Bit Name	Setting	R/W	Description
11	MDS3	Undefined*	R	Mode Select 3 to 0
10	MDS2	Undefined*	R	These bits indicate the operating mode selected by the mode pins (MD1 and MD0; see table 11). When MDCR is read, the signal levels input on pins MD2 to MD0 are latched into these bits. The latch is released by a reset.
9	MDS1	Undefined*	R	
8	MDS0	Undefined*	R	

Note: * Determined by the settings on pins MD1 and MD0.

Table 11 Values of bits MDS3 to MDS0

MCU Operating Mode	Mode Pins			MDCR			
	MD2	MD1	MD0	MDS3	MDS2	MDS1	MDS0
1	0	0	1	1	1	0	1
2	0	1	0	1	1	0	0
4	1	0	0	0	0	1	0
5	1	0	1	0	0	0	1
6	1	1	0	0	1	0	1
7	1	1	1	0	1	0	0

- System clock control register (SCKCR) Number of bits: 16 Address: H'FFFDC4

Bit	Bit Name	Setting	R/W	Description
10	ICK2	0	R/W	System Clock (Iϕ) Select
9	ICK1	0		These bits select the frequency of the system clock provided to the CPU, DMAC, and DTC.
8	ICK0	1		000: Input clock × 2
6	PCK2	0	R/W	Peripheral Module Clock (Pϕ) Select
5	PCK1	0		These bits select the frequency of the peripheral module clock.
4	PCK0	1		001: Input clock × 2
2	BCK2	0	R/W	External Bus Clock (Bϕ) Select
1	BCK1	0		These bits select the frequency of the external bus clock.
0	BCK0	1		001: Input clock × 2

- MSTPCRA, MSTPCRB, and MSTPCRC control the module stop mode. Setting a bit to 1 makes the corresponding module enter the module stop mode, while clearing the bit to 0 releases the module from module stop state.
- Module stop control register A (MSTPCRA) Number of bits: 16 Address: H'FFFDC8

Bit	Bit Name	Setting	R/W	Description
15	ACSE	0	R/W	All-Module-Clock-Stop Mode Enable Enables/disables all-module-clock-stop mode for reducing current drawn by stopping the operation of bus controller and I/O ports when the CPU executes the SLEEP instruction after the module stop mode has been set for all of the on-chip peripheral modules controlled by MSTPCR. 0: Disables all-module-clock-stop mode 1: Enables all-module-clock-stop mode
12	MSTPA12	1	R/W	Data transfer controller (DTC)
9	MSTPA9	1	R/W	8-bit timer unit (TMR_3, TMR_2)
8	MSTPA8	1	R/W	8-bit timer unit (TMR_1 and TMR_0)
5	MSTPA5	1	R/W	D/A converter (channels 1 and 0)
3	MSTPA3	1	R/W	A/D converter (unit 0)
0	MSTPA0	1	R/W	16-bit timer pulse unit (TPU channels 5 to 0)

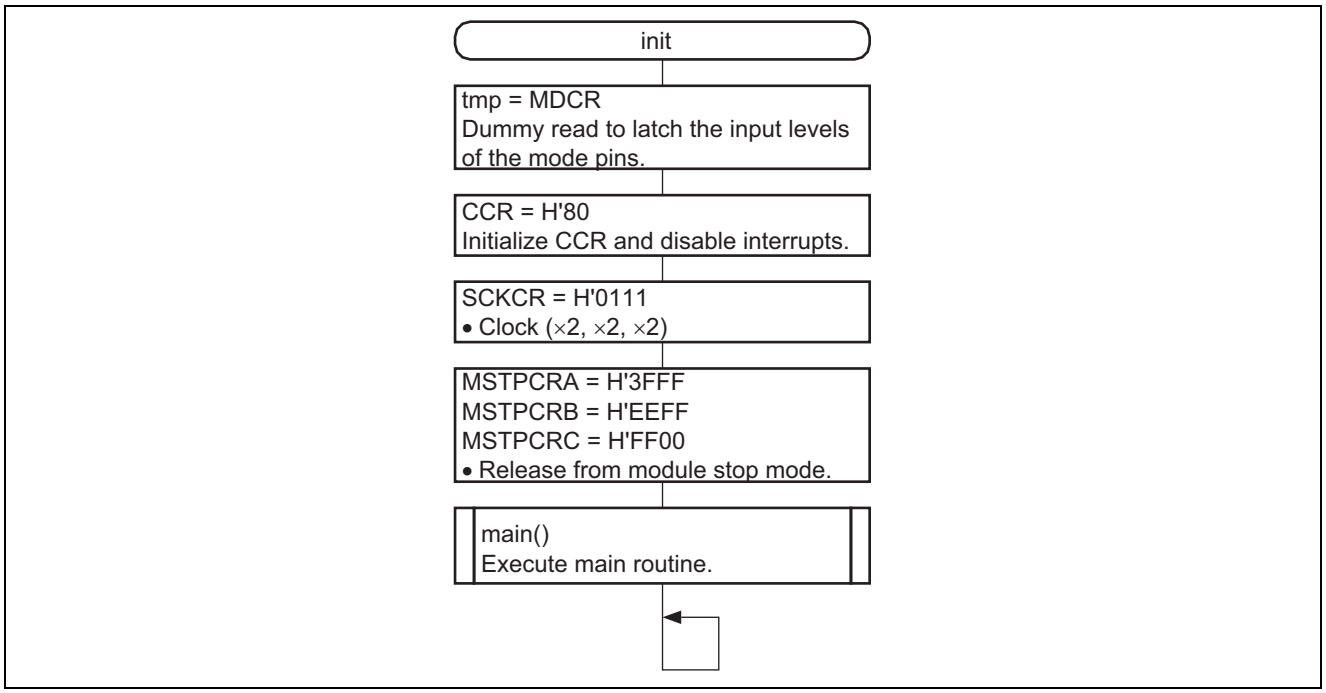
- Module stop control register B (MSTPCRB) Number of bits: 16 Address: H'FFFDCA

Bit	Bit Name	Setting	R/W	Description
15	MSTPB15	1	R/W	Programmable pulse generator (PPG)
12	MSTPB12	0	R/W	Serial communications interface_4 (SCI_4)
10	MSTPB10	1	R/W	Serial communications interface_2 (SCI_2)
9	MSTPB9	1	R/W	Serial communications interface 1 (SCI_1)
8	MSTPB8	0	R/W	Serial communications interface_0 (SCI_0)
7	MSTPB7	1	R/W	I^2C bus interface_1 (IIC_1)
6	MSTPB6	1	R/W	I^2C bus interface_0 (IIC_0)

- Module stop control register C (MSTPCRC) Number of bits: 16 Address: H'FFFDCC

Bit	Bit Name	Setting	R/W	Description
2	MSTPC2	0	R/W	On-chip RAM_2 (H'FF6000 to H'FF7FFF)
1	MSTPC1	0	R/W	On-chip RAM_1 (H'FF8000 to H'FF9FFF)
0	MSTPC0	0	R/W	On-chip RAM_0 (H'FFA000 to H'FFBFFF)

5. Flowchart


init
tmp = MDCR
Dummy read to latch the input levels of the mode pins.
CCR = H'80
Initialize CCR and disable interrupts.
SCKCR = H'0111
• Clock (×2, ×2, ×2)
MSTPCRA = H'3FFF
MSTPCRB = H'EEFF
MSTPCRC = H'FF00
• Release from module stop mode.
main()
Execute main routine.

5.2.2 main Function

1. Functional overview

- Executes the user application program (normal application)
- Receives the reprogramming start command
- Branches to programming/erasing procedure program

2. Arguments

None

3. Return value

None

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Port D data direction register (PDDDR) Number of bits: 8 Address: H'FFFB8C

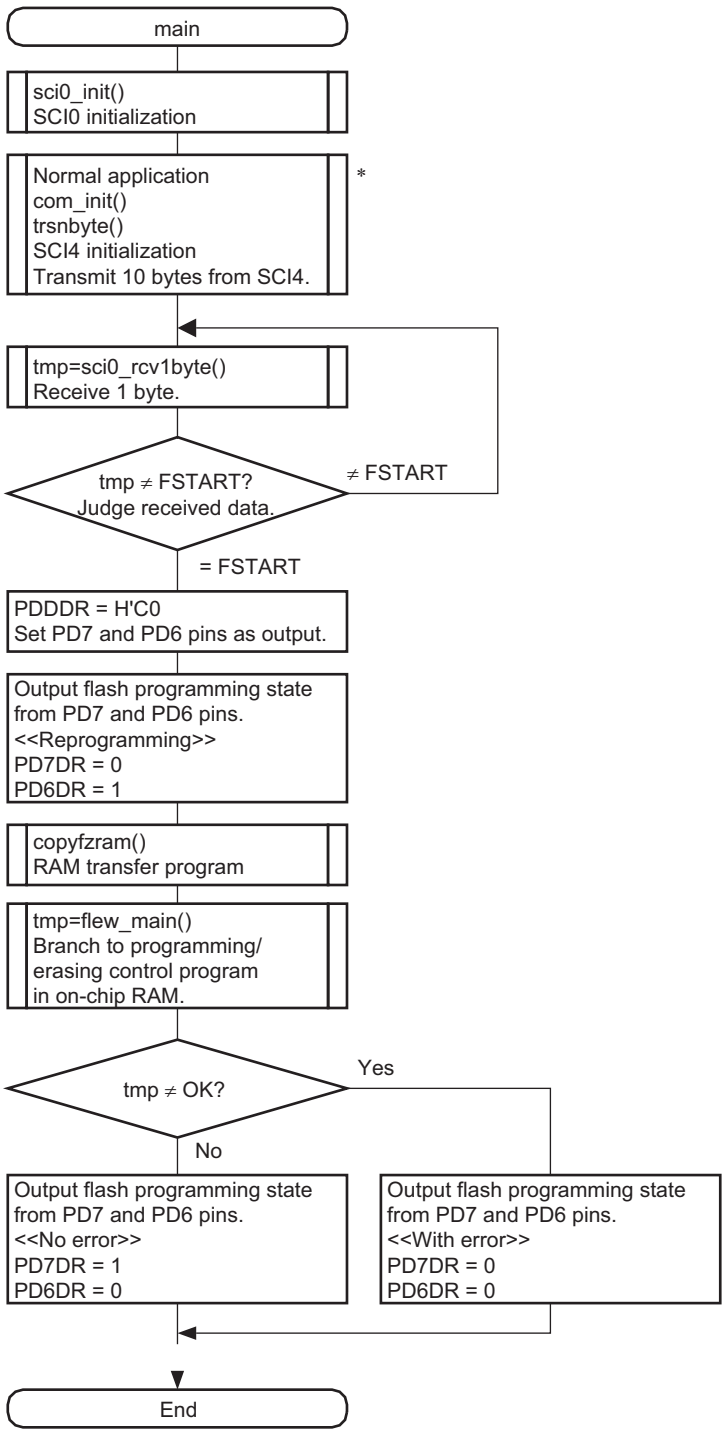
Bit	Bit Name	Setting	R/W	Description
7	PD7DDR	1	R/W	0: Sets PD7 as an input pin. 1: Sets PD7 as an output pin.
6	PD6DDR	1	R/W	0: Sets PD6 as an input pin. 1: Sets PD6 as an output pin.

- Port D data register (PDDR) Number of bits: 8 Address: H'FFFF5C

In this sample task, the PD7 and PD6 pins are used for output pins of flash reprogramming.

Bit	Bit Name	Setting	R/W	Description
7	PD7DR	Undefined	R/W	0: PD7 pin is set to a low level. 1: PD7 pin is set to a high level.
6	PD6DR	Undefined	R/W	0: PD6 pin is set to a low level. 1: PD6 pin is set to a high level.

5. Flowchart



* Normal application is a sample program that has nothing to do with the user MAT reprogramming. In this sample task, it is set to one point above.

5.2.3 copyfzram Function

1. Functional overview

Transfers the programming/erasing procedure program to the on-chip RAM.

2. Arguments

None

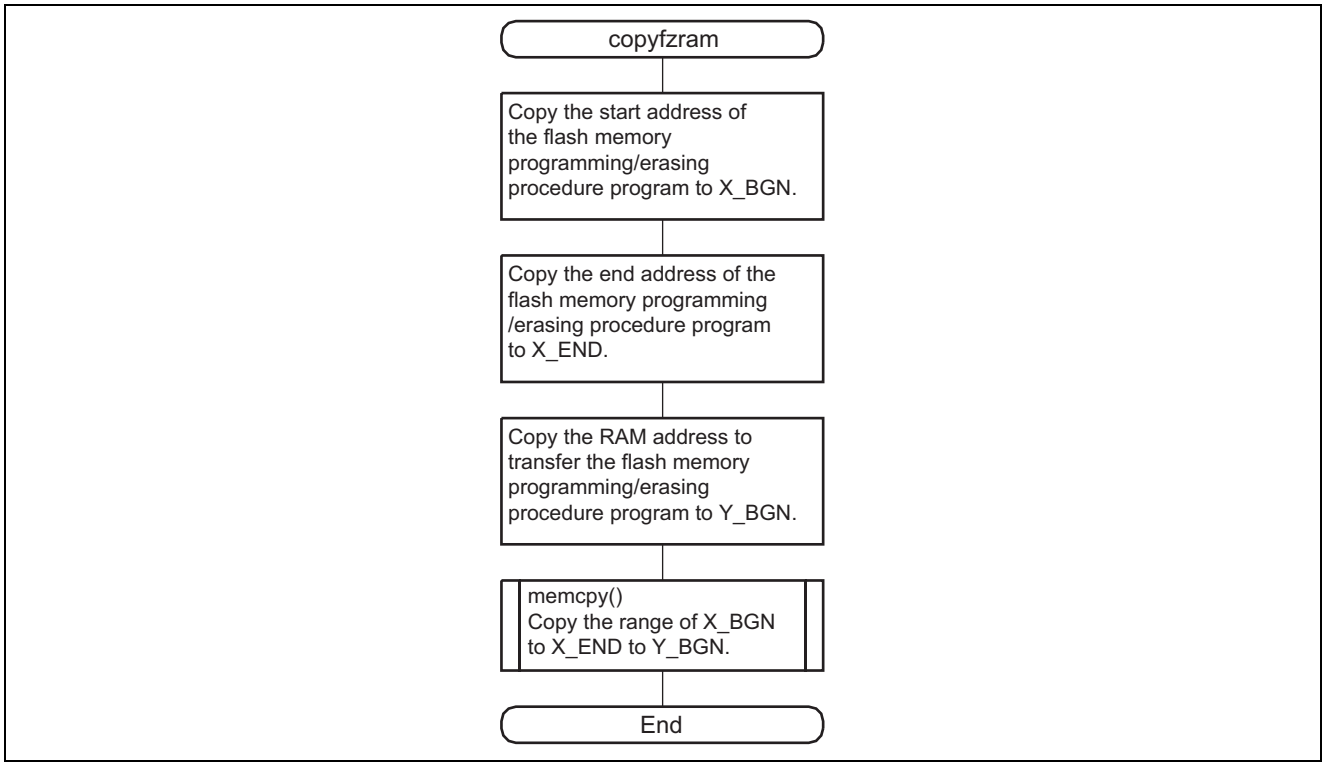
3. Return value

None

4. Internal registers used

None

5. Flowchart



5.2.4 flew_main Function

1. Functional overview

Calls the main routine of the programming/erasing procedure program.

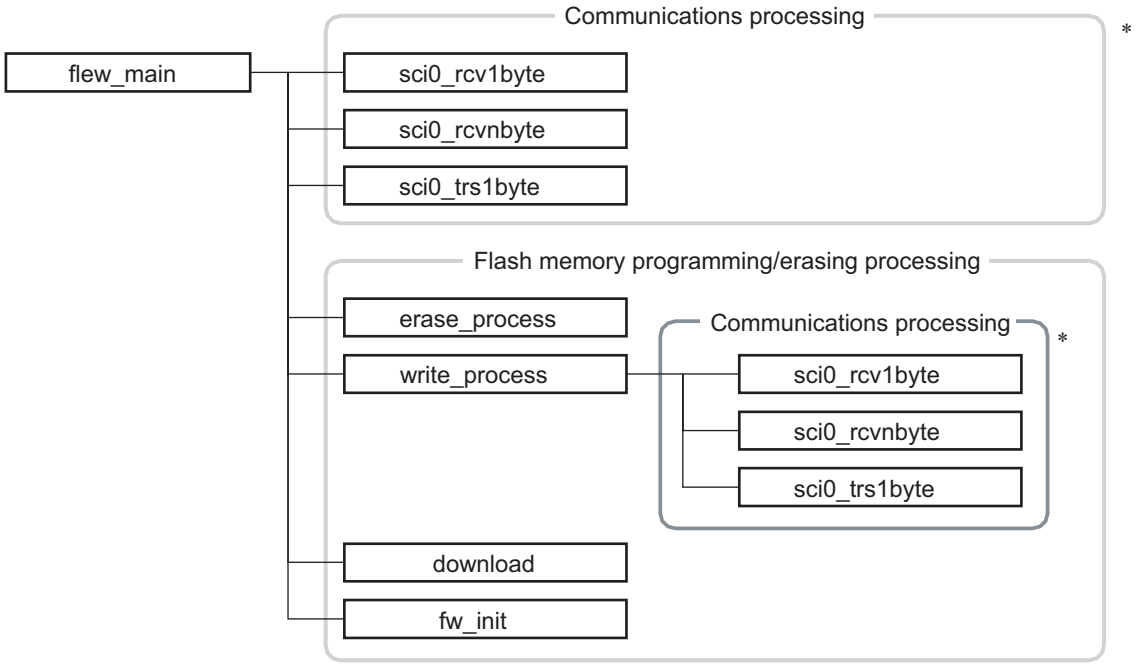
6. Description of Software for Programming/Erasing Procedure Program on the Slave Side

6.1 List of Functions

Programming/erasing procedure program (fwrite.c) performs erasing block units, receives flash memory programming data, and performs programming to flash memory. A list of functions for the routines used in the programming/erasing procedure program is given in table 12. The hierarchical structure is shown in figure 18.

Table 12 List of Functions for Programming/Erasing Procedure Program

Function Name	Description
flew_main	Main processing of flash memory erasing/programming
erase_process	Erases flash memory
write_process	Programs flash memory
download	Downloads on-chip modules
fw_init	Initialization before flash memory erasing and programming



* Called from the clock synchronous serial communications program.

Figure 18 Programming/Erasing Procedure Program

6.2 Description of Functions

6.2.1 flew_main Function

1. Functional overview

Main processing of flash memory erasing/programming

2. Arguments

None

3. Return value

Type	Description
unsigned char	Error status

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

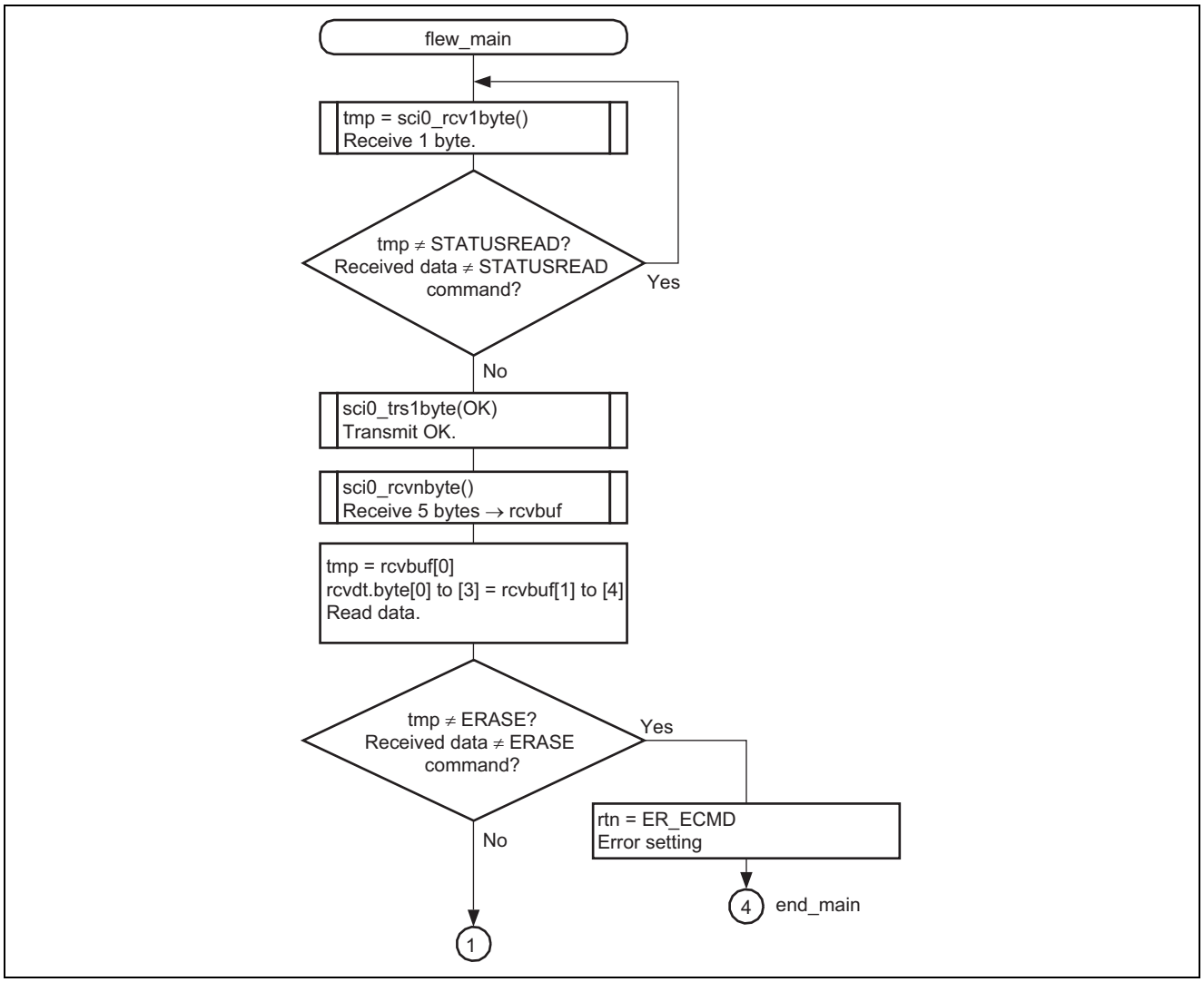
- Flash program code select register (FPCS) Number of bits: 8 Address: H'FFFDE9

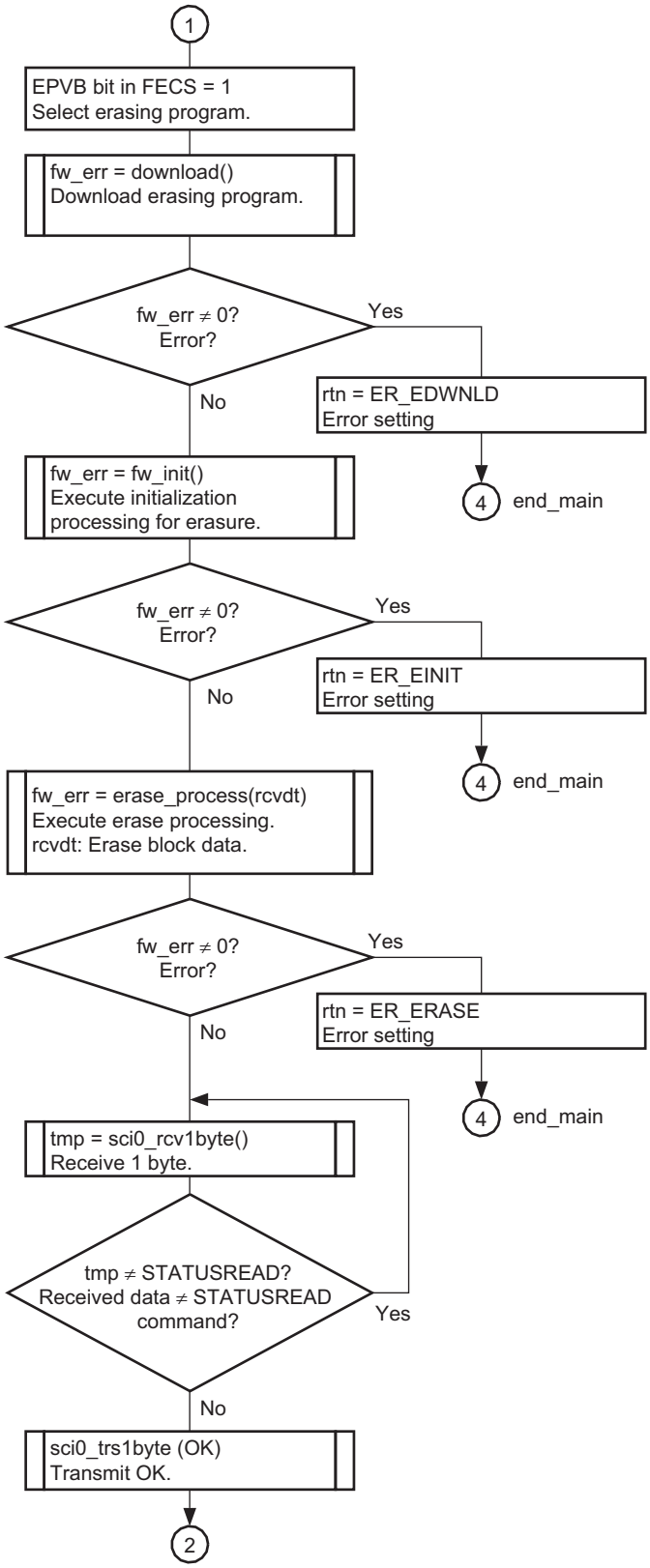
Bit	Bit Name	Setting	R/W	Description
0	PPVS	1	R/W	Program Pulse Verify Selects the programming program to be downloaded. 0: Does not select programming program. [Clearing condition] • Completion of transfer 1: Selects programming program.

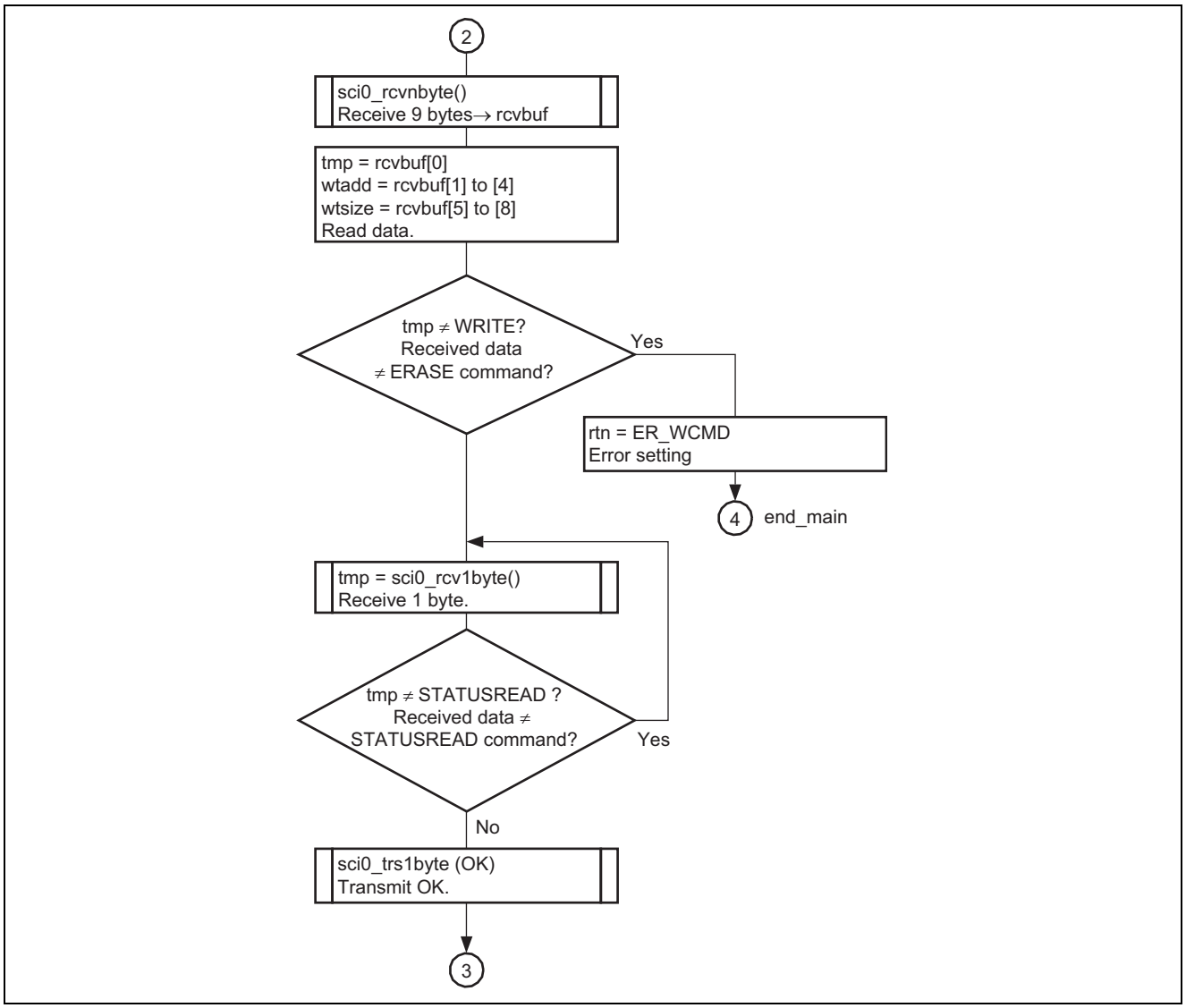
- Flash erase code select register (FECS) Number of bits: 8 Address: H'FFFDEA

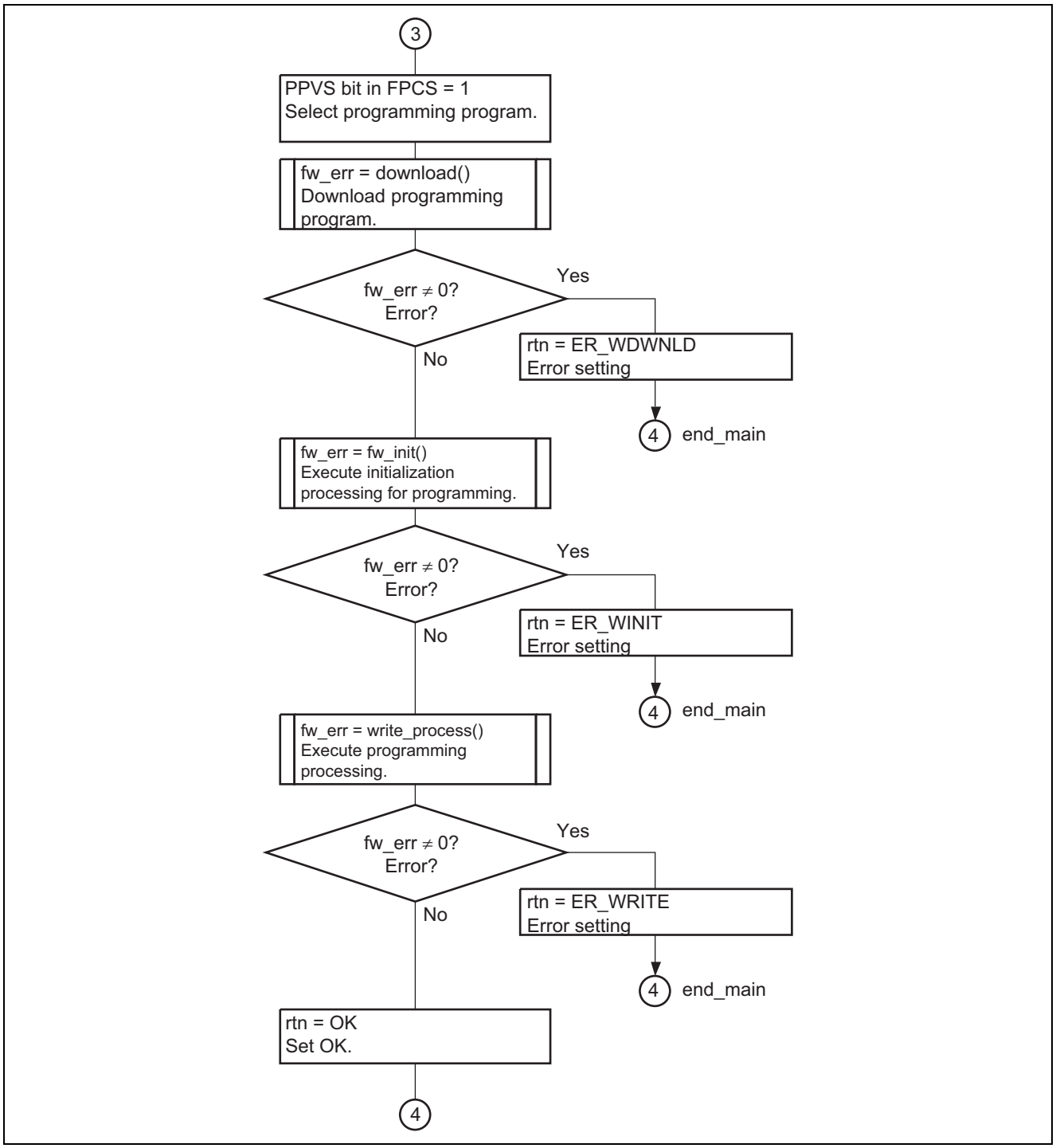
Bit	Bit Name	Setting	R/W	Description
0	EPVB	1	R/W	Erase Pulse Verify Block Selects the erasing program to be downloaded. 0: Does not select erasing program. [Clearing condition] • Completion of transfer 1: Selects erasing program.

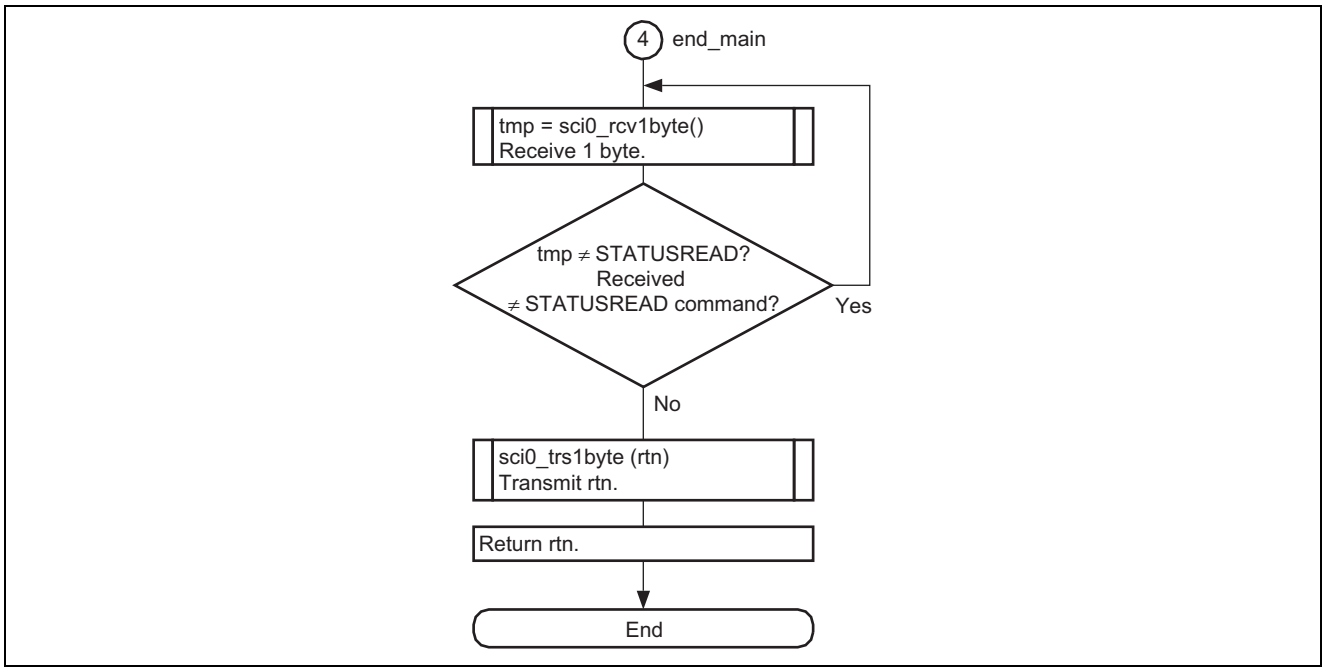
5. Flowchart


flew_main
tmp = sci0_rcv1byte()
Receive 1 byte.
tmp ≠ STATUSREAD?
Received data ≠ STATUSREAD command?
Yes
No
sci0_trs1byte(OK)
Transmit OK.
sci0_rcvnbyte()
Receive 5 bytes → rcvbuf
tmp = rcvbuf[0]
rcvdt.byte[0] to [3] = rcvbuf[1] to [4]
Read data.
tmp ≠ ERASE?
Received data ≠ ERASE command?
Yes
No
rtn = ER_ECMD
Error setting
4
end_main
1


1
EPVB bit in FECS = 1
Select erasing program.
fw_err = download()
Download erasing program.
fw_err ≠ 0?
Error?
Yes
No
rtn = ER_EDWNLD
Error setting
4
end_main
fw_err = fw_init()
Execute initialization
processing for erasure.
fw_err ≠ 0?
Error?
Yes
No
rtn = ER_EINIT
Error setting
4
end_main
fw_err = erase_process(rcvdt)
Execute erase processing.
rcvdt: Erase block data.
fw_err ≠ 0?
Error?
Yes
No
rtn = ER_ERASE
Error setting
4
end_main
tmp = sci0_rcv1byte()
Receive 1 byte.
tmp ≠ STATUSREAD?
Received data ≠ STATUSREAD
command?
Yes
No
sci0_trs1byte (OK)
Transmit OK.
2


2
sci0_rcvnbyte()
Receive 9 bytes→ rcvbuf
tmp = rcvbuf[0]
wtadd = rcvbuf[1] to [4]
wtsize = rcvbuf[5] to [8]
Read data.
tmp ≠ WRITE?
Received data
≠ ERASE command?
Yes
rtn = ER_WCMD
Error setting
4
end_main
tmp = sci0_rcv1byte()
Receive 1 byte.
tmp ≠ STATUSREAD ?
Received data ≠
STATUSREAD command?
Yes
No
sci0_trs1byte (OK)
Transmit OK.
3


3
PPVS bit in FPCS = 1
Select programming program.
fw_err = download()
Download programming program.
fw_err ≠ 0?
Error?
Yes
No
rtn = ER_WDWNLD
Error setting
4
end_main
fw_err = fw_init()
Execute initialization processing for programming.
fw_err ≠ 0?
Error?
Yes
No
rtn = ER_WINIT
Error setting
4
end_main
fw_err = write_process()
Execute programming processing.
fw_err ≠ 0?
Error?
Yes
No
rtn = ER_WRITE
Error setting
4
end_main
rtn = OK
Set OK.
4


4
end_main
tmp = sci0_rcv1byte()
Receive 1 byte.
tmp ≠ STATUSREAD?
Received
≠ STATUSREAD command?
Yes
No
sci0_trs1byte (rtn)
Transmit rtn.
Return rtn.
End

6.2.2 erase_process Function

1. Functional overview

Erase flash memory.

2. Arguments

Type	Variable Name	Description
unsigned long	ERASEBLK	Erases block.

3. Return value

Type	Description
unsigned char	Flash pass and fail parameter (FPFR) Return value of the erase result.

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Flash key code register (FKEY) Number of bits: 8 Address: H'FFFDEC

Bit	Bit Name	Setting	R/W	Description
7	K7	0	R/W	Key Code When H'A5 is written to FKEY, writing to the SCO bit in FCCS is enabled. When a value other than H'A5 is written, the SCO bit cannot be set to 1. Therefore, the on-chip program cannot be downloaded to the on-chip RAM. Only When H'5A is written, programming/erasing of the flash memory can be executed. When a value other than H'5A is written, even if the programming/erasing program is executed, programming/erasing cannot be performed. H'A5: Writing to the SCO bit is enabled. (The SCO bit cannot be set to 1 when FKEY is a value other than H'A5.) H'5A: Programming/erasing of the flash memory is enabled. (When FKEY is a value other than H'5A, the software protection state is entered.) H'00: Initial value
6	K6	1	R/W	
5	K5	0	R/W	
4	K4	1	R/W	
3	K3	1	R/W	
2	K2	0	R/W	
1	K1	1	R/W	
0	K0	0	R/W	

- Flash pass and fail parameter (FPFR)
  (CPU general register R0L)

Return value of the erase results

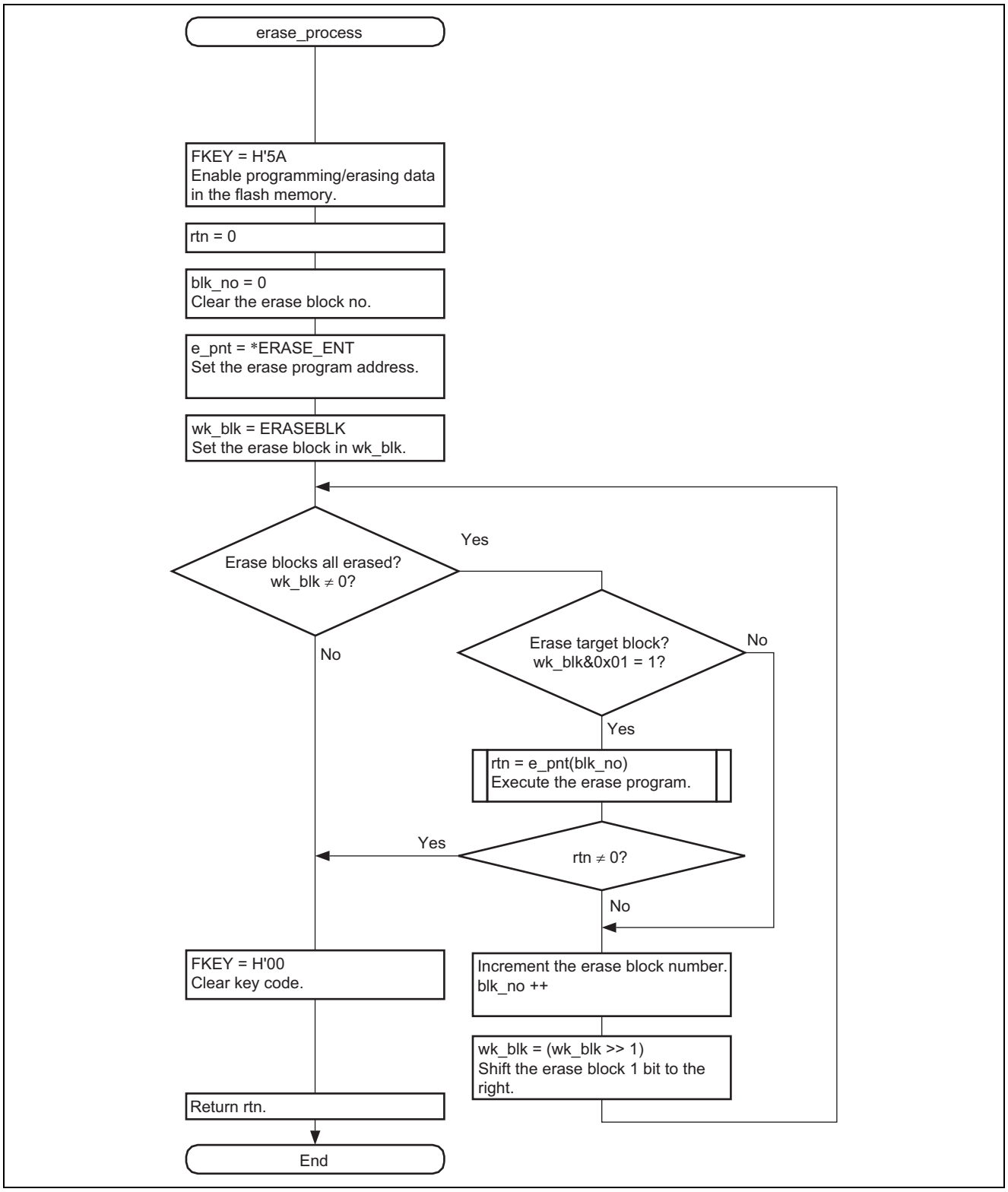
Bit	Bit Name	Setting	R/W	Description
6	MD	Undefined	R/W	Erasing Mode Related Setting Error Detect Detects the error protection state and returns the result. When the error protection state is entered, this bit is set to 1. Whether the error protection state is entered or not can be performed with the FLER bit of the FCCS register. 0: Normal operation (FLER = 0) 1: Error protection state, and programming cannot be performed (FLER = 1)
5	EE	Undefined	R/W	Erasing Execution Error Detect Writes 1 to this bit when the specified data could not be written because the user MAT was not erased. If this is set to 1, there is a high possibility that the user MAT has been written partially. In this case, after removing the error factor, erase the user MAT. 0: Erasure has ended normally. 1: Erasure has ended abnormally.
4	FK	Undefined	R/W	Flash Key Register Error Detect Checks the FKEY value (H'A5) before programming starts, and returns the results. 0: FKEY setting is normal (H'5A). 1: FKEY setting is abnormal (a value other than H'5A).
3	EB	Undefined	R/W	Erase Block Selection Error Detect Checks whether the specified erase block number is in the block range of user MAT and returns the result. 0: Setting of erase block number is normal. 1: Setting of erase block number is abnormal.
0	SF	Undefined	R/W	Success/Fail Indicates the erasure results. 0: Erasure ended normally (no error). 1: Erasure ended abnormally (error occurred).

- Flash erase block select parameter (FEBS)
  (CPU general register ER0)

Sets the erase block number in the range from 0 to 19. Number 0 corresponds to block EB0 and number 19 corresponds to block EB19. An error occurs when a number other than 0 to 19 is set.

Setting: blk_no

5. Flowchart


erase_process
FKEY = H'5A
Enable programming/erasing data in the flash memory.
rtn = 0
blk_no = 0
Clear the erase block no.
e_pnt = *ERASE_ENT
Set the erase program address.
wk_blk = ERASEBLK
Set the erase block in wk_blk.
Erase blocks all erased?
wk_blk ≠ 0?
Yes
No
Erase target block?
wk_blk&0x01 = 1?
No
Yes
rtn = e_pnt(blk_no)
Execute the erase program.
rtn ≠ 0?
Yes
No
FKEY = H'00
Clear key code.
Increment the erase block number.
blk_no ++
wk_blk = (wk_blk >> 1)
Shift the erase block 1 bit to the right.
Return rtn.
End

6.2.3 write_process Function

1. Functional overview

Programs flash memory.

2. Arguments

Type	Variable Name	Description
unsigned long	fladr	Reprogramming start address
unsigned long	flsize	Reprogramming size

3. Return value

Type	Description
unsigned char	Flash pass and fail parameter (FPFR) Return value of the programming result.

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Flash key code register (FKEY) Number of bits: 8 Address: H'FFFDEC

Bit	Bit Name	Setting	R/W	Description
7	K7	0	R/W	Key code
6	K6	1	R/W	When H'A5 is written to FKEY, writing to the SCO bit in FCCS is enabled. When a value other than H'A5 is written, the SCO bit cannot be set to 1. Therefore, the on-chip program cannot be downloaded to the on-chip RAM. Only When H'5A is written, programming/erasing of the flash memory can be executed. When a value other than H'5A is written, even if the programming/erasing program is executed, programming/erasing cannot be performed. H'A5: Writing to the SCO bit is enabled. (The SCO bit cannot be set to 1 when FKEY is a value other than H'A5.) H'5A: Programming/erasing of the flash memory is enabled. (When FKEY is a value other than H'5A, the software protection state is entered.) H'00: Initial value
5	K5	0	R/W	
4	K4	1	R/W	
3	K3	1	R/W	
2	K2	0	R/W	
1	K1	1	R/W	
0	K0	0	R/W	

- Flash pass and fail parameter (FPFR)
  (CPU general register R0L)

Return value of the erase results

Bit	Bit Name	Setting	R/W	Description
6	MD	Undefined	R/W	Programming Mode Related Setting Error Detect Detects the error protection state and returns the result. When the error protection state is entered, this bit is set to 1. Whether the error protection state is entered or not can be performed with the FLER bit of the FCCS register. 0: Normal operation (FLER = 0) 1: Error protection state, and programming cannot be performed (FLER = 1)
5	EE	Undefined	R/W	Programming Execution Error Detect Writes 1 to this bit when the specified data could not be written because the user MAT was not erased. If this is set to 1, there is a high possibility that the user MAT has been written partially. In this case, after removing the error factor, erase the user MAT. 0: Erasure has ended normally. 1: Erasure has ended abnormally.
4	FK	Undefined	R/W	Flash Key Register Error Detect Checks the FKEY value (H'A5) before programming starts, and returns the results. 0: FKEY setting is normal (H'5A). 1: FKEY setting is abnormal (a value other than H'5A).
2	WD	Undefined	R/W	Write Data Address Detect When an address not in the flash memory area is specified as the start address of the storage destination for the program data, an error occurs. 0: Setting of the start address of the storage destination for the program data is normal. 1: Setting of the start address of the storage destination for the program data is abnormal.
1	WA	Undefined	R/W	Write Address Error Detect When the following items are specified as the start address of the programming destination, an error occurs. • An area other than flash memory • The specified address is not aligned with the 128-byte boundary. (lower eight bits of the address are other than H'00 and H'80) 0: Setting of the start address of the programming destination is normal. 1: Setting of the start address of the programming destination is abnormal.
0	SF	Undefined	R/W	Success/Fail Indicates the program results. 0: Programming ended normally (no error). 1: Programming ended abnormally (error occurred).

- Flash multipurpose address area parameter (FMPAR)
  (CPU general register ER1)

FMPAR sets the start address of the programming destination on the user MAT.

When an address in an area other than the flash memory is set, or the start address of the programming destination is not aligned with the 128-byte boundary, an error occurs. The error occurrence is indicated by the WA bit of the FPFR register.

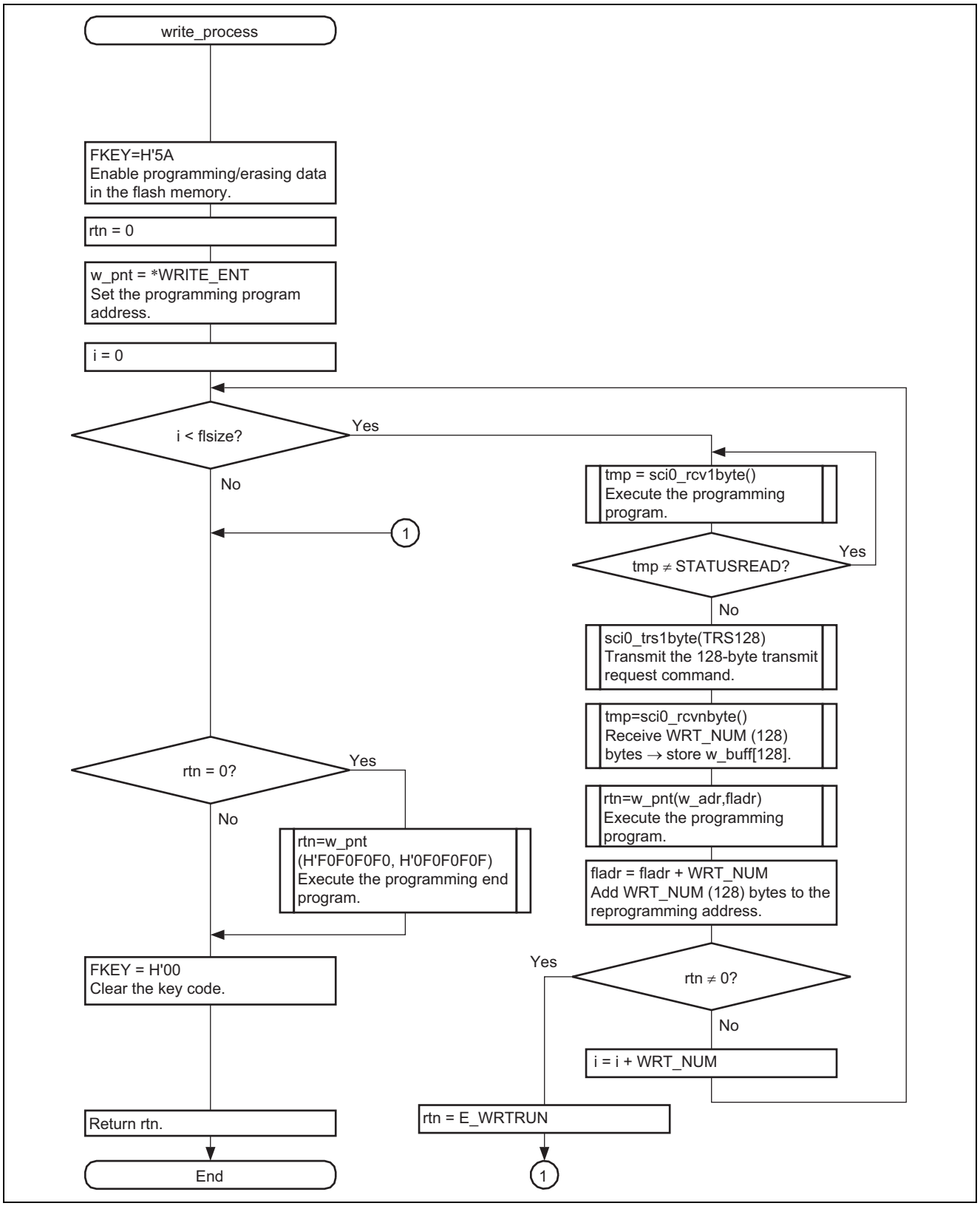
Bit	Bit Name	Setting	R/W	Description
31 to 0	MOA31 to MOA0	fladr (local variable)	R/W	Sets the start address of the programming destination on the user MAT. Consecutive 128-byte programming is executed starting from the specified start address of the user MAT. Therefore, the specified start address of the programming destination becomes a 128-byte boundary, and MOA6 to MOA0 are always cleared to 0.

- Flash multipurpose data destination parameter (FMPDR)
  (CPU general register ER0)

FMPDR sets the start address in the area which stores the data to be programmed in the user MAT. When the storage destination for the program data is in flash memory, an error occurs. The error occurrence is indicated by the WD bit in FPFR.

Bit	Bit Name	Setting	R/W	Description
31 to 0	MOD31 to MOD0	fladr (local variable)	R/W	Sets the start address of the area which stores the program data for the user MAT. Consecutive 128-byte data is programmed to the user MAT starting from the specified start address.

5. Flowchart


write_process
FKEY=H'5A
Enable programming/erasing data in the flash memory.
rtn = 0
w_pnt = *WRITE_ENT
Set the programming program address.
i = 0
i < flsize?
Yes
No
tmp = sci0_rcv1byte()
Execute the programming program.
tmp ≠ STATUSREAD?
Yes
No
sci0_trs1byte(TRS128)
Transmit the 128-byte transmit request command.
tmp=sci0_rcvnbyte()
Receive WRT_NUM (128) bytes → store w_buff[128].
rtn=w_pnt(w_adr,fladr)
Execute the programming program.
fladr = fladr + WRT_NUM
Add WRT_NUM (128) bytes to the reprogramming address.
rtn ≠ 0?
Yes
No
i = i + WRT_NUM
rtn = 0?
Yes
No
rtn=w_pnt
(H'F0F0F0F0, H'0F0F0F0F)
Execute the programming end program.
FKEY = H'00
Clear the key code.
Return rtn.
End
rtn = E_WRTRUN
1
1

6.2.4 download Function

1. Functional overview

Downloads the on-chip modules.

2. Arguments

None

3. Return value

Type	Description
unsigned char	Download pass and fail result parameter (DPFR) Return value of the download result.

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Flash key code register (FKEY) Number of bits: 8 Address: H'FFFDEC

Bit	Bit Name	Setting	R/W	Description
7	K7	1	R/W	Key Code When H'A5 is written to FKEY, writing to the SCO bit in FCCS is enabled. When a value other than H'A5 is written, the SCO bit cannot be set to 1. Therefore, the on-chip program cannot be downloaded to the on-chip RAM. Only When H'5A is written, programming/erasing of the flash memory can be executed. When a value other than H'5A is written, even if the programming/erasing program is executed, programming/erasing cannot be performed. H'A5: Writing to the SCO bit is enabled. (The SCO bit cannot be set to 1 when FKEY is a value other than H'A5.) H'5A: Programming/erasing of the flash memory is enabled. (When FKEY is a value other than H'5A, the software protection state is entered.) H'00: Initial value
6	K6	0	R/W	
5	K5	1	R/W	
4	K4	0	R/W	
3	K3	0	R/W	
2	K2	1	R/W	
1	K1	0	R/W	
0	K0	1	R/W	

- Flash transfer destination address register (FTDAR) Number of bits: 8 Address: H'FFFDEE

Bit	Bit Name	Setting	R/W	Description
7	TDER	0	R/W	Transfer Destination Address Setting Error This bit is set to 1 when an error has occurred in setting the start address specified by bits TDA6 to TDA0. A start address error is determined by whether the value set in bits TDA6 to TDA0 is within the range of H'00 to H'02 when download is executed by setting the SCO bit in FCCS to 1. Make sure that this bit is cleared to 0 before setting the SCO bit to 1 and the value specified by bits TDA6 to TDA0 should be within the range of H'00 to H'02. 0: The value specified by bits TDA6 to TDA0 is written the range. 1: The value specified by TDER and bits TDA6 to TDA0 is between H'03 and H'FF and download has stopped.
6	TDA6	0	R/W	Transfer Destination Address Specifies the on-chip RAM start address of the download destination. A value between H'00 and H'02, and up to 4 Kbytes can be specified as the start address of the on-chip RAM. H'00: H'FF9000 is specified as the start address. H'01: H'FFA000 is specified as the start address. H'02: H'FFB000 is specified as the start address. H'03 to H'7F: Setting prohibited. (Specifying a value from H'03 to H7F sets the TDER bit to 1 and stops download of the on-chip program.)
5	TDA5	0	R/W	
4	TDA4	0	R/W	
3	TDA3	0	R/W	
2	TDA2	0	R/W	
1	TDA1	1	R/W	
0	TDA0	0	R/W	

- Flash code control status register (FCCS) Number of bits: 8 Address: H'FFFDE8

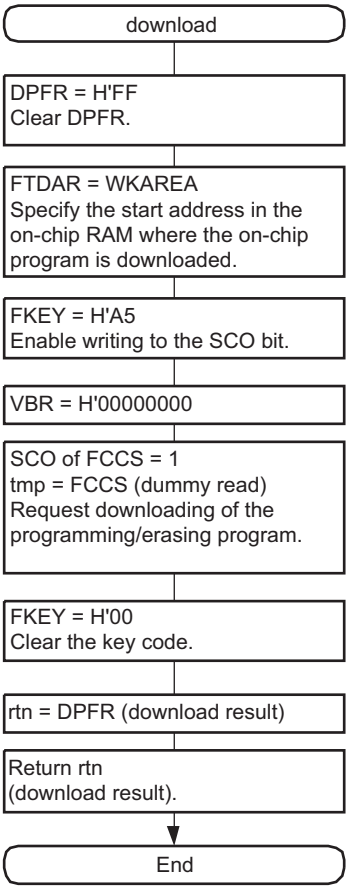
Bit	Bit Name	Setting	R/W	Description
4	FLER	0	R	Flash Memory Error Indicates that an error has occurred during programming/erasing the flash memory. When this bit is set to 1, the flash memory enters the error protection state. When this bit is set to 1, high voltage is applied to the internal flash memory. To reduce the damage to the flash memory, the reset must be released after the reset input period (period of $\overline{RES}$ = 0) of at least 100 µs. 0: Flash memory operates normally (Error protection is invalid). [Clearing condition] • At a power-on reset 1: An error occurs during programming/erasing flash memory (error protection is valid). [Setting conditions] • When an interrupt, such as NMI, occurs during programming/erasing. • When the flash memory is read during programming/erasing (including a vector read and an instruction fetch). • When the SLEEP instruction is executed during programming/erasing (including software standby mode). • When a bus master other than the CPU, such as the DTC and DMAC, obtains bus mastership during programming/erasing.
0	SCO	0	(R)/W*	Source Program Copy Operation Requests the on-chip programming/erasing program to be downloaded to the on-chip RAM. When this bit is set to 1, the on-chip program which is selected by FPCS or FECS is automatically downloaded in the on-chip RAM area specified by FTDAR. In order to set this bit to 1, the RAM emulation mode must be canceled, H'A5 must be written to FKEY, and the setting of SCO bit must be executed in the on-chip RAM. Dummy read of FCCS must be executed twice immediately after setting this bit to 1. All interrupts must be disabled during download. This bit is cleared to 0 when download is completed. During program download initiated with this bit, particular processing which accompanies bank switching of the program storage area is executed. Before a download request, initialize the VBR contents to H'00000000. After download is completed, the VBR contents can be changed. 0: Download of the programming/erasing program is not requested. [Clearing condition] • When download is completed. 1: Download of the programming/erasing program is not requested. [Setting conditions] (When all of the following conditions are satisfied.) • Not in RAM emulation mode (The RAMS bit of RAMER is cleared to 0). • H'A5 is written to FKEY. • Setting of SCO bit in FCCS is executed in the on-chip RAM.

Note: * SCO is a write-only bit. This bit is always read as 0.

- Download pass and fail result parameter (DPFR)
  (Single byte of start address in on-chip RAM specified by FTDAR.)
  DPFR indicates the return value of the download result. The DPFR value is used to determine the download result.

Bit	Bit Name	Setting	R/W	Description
2	SS	1	R/W	Source Select Error Detect Only one type can be specified for the on-chip program which can be downloaded. When the program to be downloaded is not selected, more than two types of programs are selected, or a program which is not mapped is selected, an error occurs. 0: Download program selection is normal. 1: Download program selection is abnormal.
1	FK	1	R/W	Flash Key Register Error Detect Checks the FKEY value (H'A5) and returns the result. 0: FKEY setting is normal (H'A5). 1: FKEY setting is abnormal (value other than H'A5)
0	SF	1	R/W	Success/Fail Returns the download result. Reads back the program downloaded to the on-chip RAM and determines whether it has been transferred to the on-chip RAM. 0: Download of the program has ended normally. 1: Download of the program has ended abnormally (error occurred).

5. Flowchart



6.2.5 fw_init Function

1. Functional overview

Initializes flash memory before programming.

2. Arguments

None

3. Return value

Type	Description
unsigned char	Flash pass and fail parameter (FPFR) Return value of the initialization result.

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Flash program/erase frequency parameter (FPEFEQ)
  (CPU general register ER0)

FPEFEQ sets the operating frequency of the CPU. The CPU operating frequency available in this LSI ranges from 8 to 50 MHz.

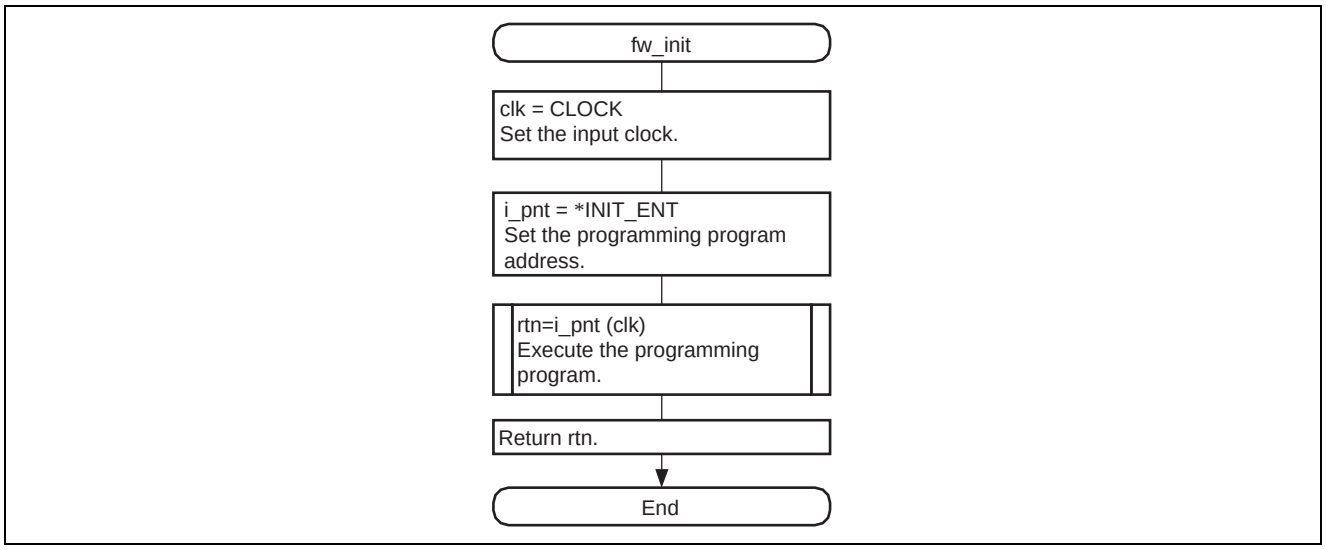
Bit	Bit Name	Setting	R/W	Description
15 to 0	F15 to F0	CLOCK	R/W	Frequency Set Sets the operating frequency of the CPU. When the PLL multiplication function is used, set the multiplied frequency. The setting value must be calculated as follows: • The operating frequency shown in MHz units must be rounded in a number of three decimal places and be shown in a number of two decimal places. • The value multiplied by 100 is converted to the binary digit and is written to FPEFEQ (general register ER0). For example, when the operating frequency of the CPU is 35.000MHz, the value is as follows: 1. The number of three decimal places of 35.000 is rounded. 2. The formula of 35.00 × 100 = 3500 is converted to the binary digit and B'0000 1101 1010 1100 (H'0CE4) is set to ER0.

- Flash pass and fail parameter (FPFR)
  (CPU general register R0L)

Return value of the initialization results

Bit	Bit Name	Setting	R/W	Description
1	FQ	Undefined	R/W	Frequency Error Detect Compares the specified CPU operating frequency with the operating frequencies supported by this LSI, and returns the result. 0: Setting of operating frequency is normal. 1: Setting of operating frequency is abnormal.
0	SF	Undefined	R/W	Success/Fail Indicates the erasure results. 0: Erasure ended normally (no error). 1: Erasure ended abnormally (error occurred).

5. Flowchart


fw_init
clk = CLOCK
Set the input clock.
i_pnt = *INIT_ENT
Set the programming program address.
rtn=i_pnt (clk)
Execute the programming program.
Return rtn.
End

7. Description of Software for the Clock Synchronous Serial Communications Program on the Slave Side

7.1 List of Functions

The clock synchronous serial communications program (sci0.c) performs communications processing to the master side. Table 13 is a list of functions in the clock synchronous serial communications program, and figure 19 shows the hierarchy structure.

Table 13 Functions in the Clock Synchronous Serial Communications Program

Function Name	Description
sci0_init	Initializes clock synchronous serial communications
sci0_rcv1byte	Receives one byte of data
sci0_rcvnbyte	Receives n bytes of data
sci0_trs1byte	Transmits one byte of data
sci0_trsnbyte	Transmits n bytes of data

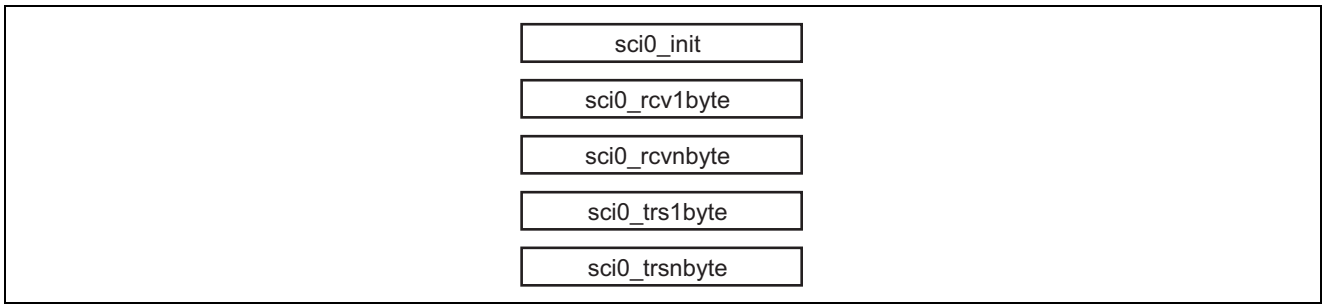


Figure 19 Clock Synchronous Serial Communications Program

7.2 Description of Functions

7.2.1 sci0_init Function

1. Functional overview

Initializes clock synchronous serial communications

2. Arguments

None

3. Return value

None

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Serial mode register_0 (SMR_0) Number of bits: 8 Address: H'FFFF80

Bit	Bit Name	Setting	R/W	Description
7	C/$\overline{A}$	1	R/W	Communications Mode 0: Asynchronous mode 1: Clock synchronous mode
6	CHR	0	R/W	Character Length 0: Selects 8 bits as the data length 1: Selects 7 bits as the data length
1	CKS1	0	R/W	Clock Selection 1, 0
0	CKS0	0	R/W	B'00: Clock source of the on-chip baud rate generator is set to Pϕ clock.

- Serial control register_0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
5	TE	0	R/W	Transmit Enable 0: Disables transmission. 1: Enables transmission.
4	RE	0	R/W	Receive Enable 0: Disables reception. 1: Enables reception.
1	CKE1	1	R/W	Clock Enable 1, 0
0	CKE0	0	R/W	In the clock synchronous mode: B'0X: Internal clock (SCK pin functions as the clock output pin.) B'1X: External clock (SCK pin functions as the clock input pin.)

Legend

X: Don't care.

- Serial status register_0 (SSR_0) Number of bits: 8 Address: H'FFFF84

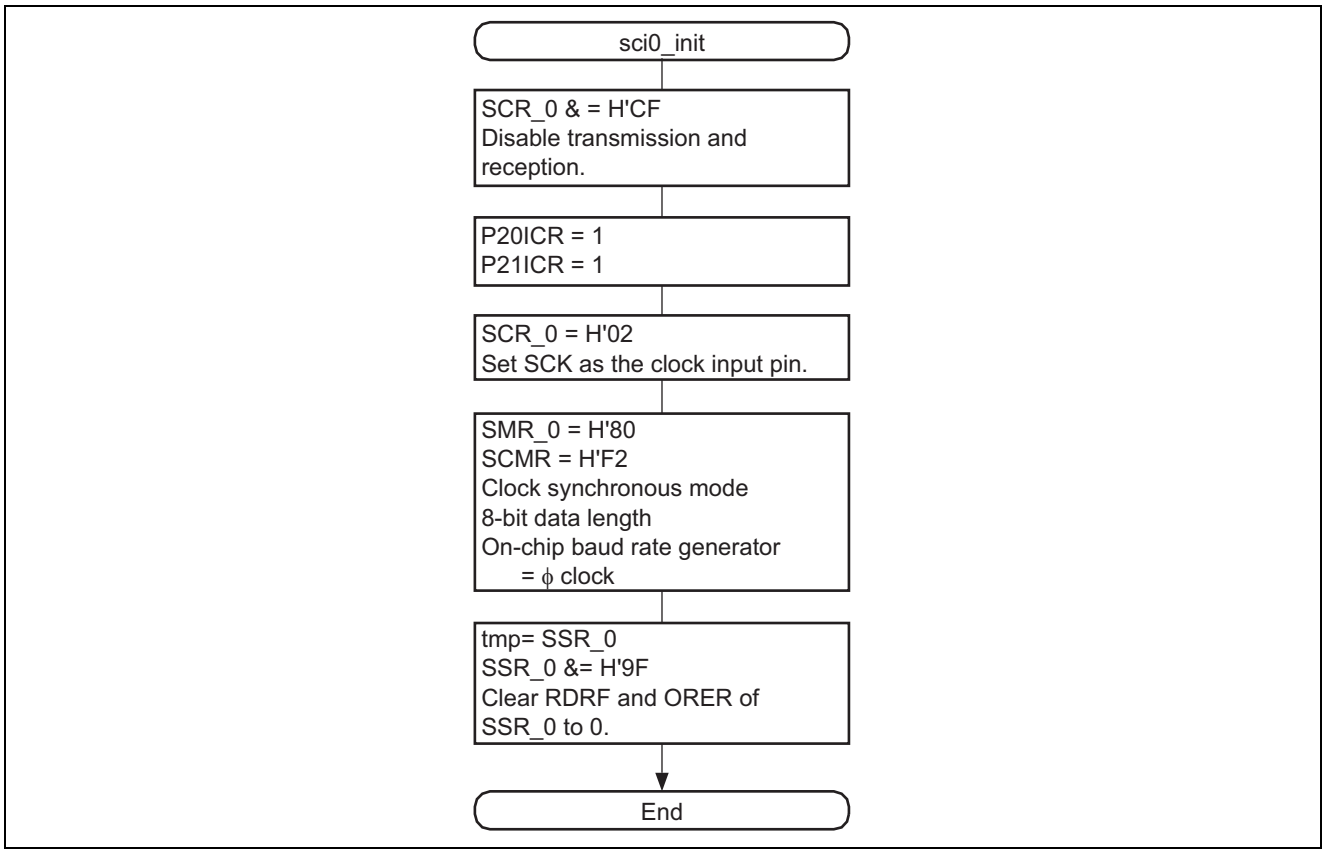
Bit	Bit Name	Setting	R/W	Description
6	RDRF	0	R/(W)*	Receive Data Register Full Indicates whether receive data is stored in RDR. [Setting condition] • Normal ending of serial reception and transferring of receive data from RSR to RDR [Clearing conditions] • Writing of 0 to RDRF after having read RDRF as 1. • Generation of an RXI interrupt request allowing DMAC or DTC to read data from RDR. The RDRF flag is not affected and retains its previous value when the RE bit in SCR is cleared to 0. Note that when the next serial reception is completed while the RDRF flag is being set to 1, an overrun error occurs and the received data is lost.
5	ORER	0	R/(W)*	Overrun Error [Setting condition] • Occurrence of overrun error during reception [Clearing condition] • Writing of 0 after having read ORER as 1

Note: * Only 0 can be written here, to clear the flags.

- Port 2 input buffer control register (P2ICR) Number of bits: 8 Address: H'FFFB91

Bit	Bit Name	Setting	R/W	Description
1	P21ICR	1	R/W	0: Input buffer for P21 (SCK0) pin is disabled. 1: Input buffer for P21 (SCK0) pin is enabled.
0	P20ICR	1	R/W	0: Input buffer for P20 (RxD0) pin is disabled. 1: Input buffer for P20 (RxD0) pin is enabled.

5. Flowchart


sci0_init
SCR_0 & = H'CF
Disable transmission and
reception.
P20ICR = 1
P21ICR = 1
SCR_0 = H'02
Set SCK as the clock input pin.
SMR_0 = H'80
SCMR = H'F2
Clock synchronous mode
8-bit data length
On-chip baud rate generator
= ϕ clock
tmp= SSR_0
SSR_0 &= H'9F
Clear RDRF and ORER of
SSR_0 to 0.
End

7.2.2 sci0_rcv1byte Function

1. Functional overview

Receives one byte of clock synchronous serial data.

2. Arguments

None

3. Return value

Type	Description
unsigned char	Receives one byte of data.

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Serial control register_0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
4	RE	0	R/(W)*	Receive Enable 0: Disables reception. 1: Enables reception.

- Serial status register_0 (SSR_0) Number of bits: 8 Address: H'FFFF84

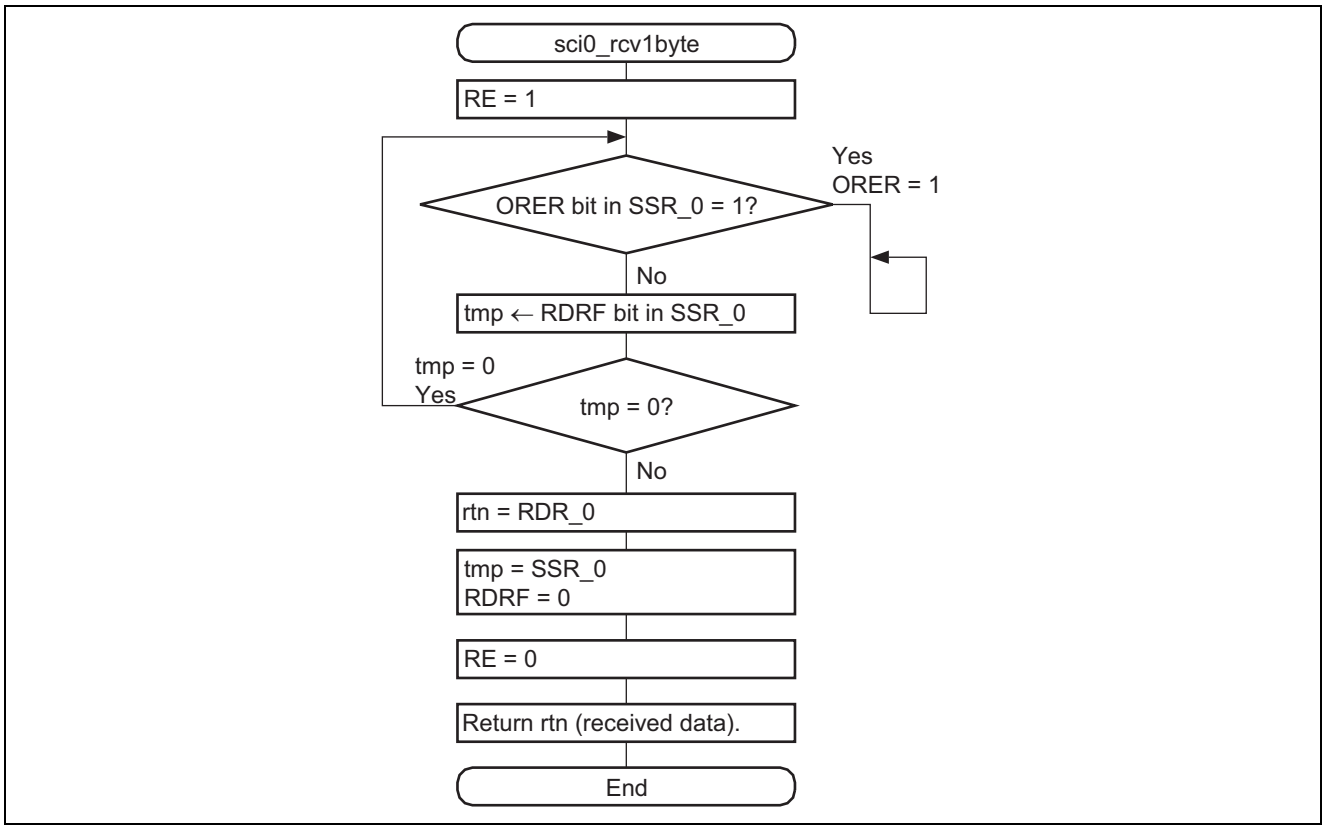
Bit	Bit Name	Setting	R/W	Description
6	RDRF	Undefined	R/(W)*	Receive Data Register Full Indicates whether receive data is stored in RDR. [Setting condition] • Normal ending of serial reception and transferring of receive data from RSR to RDR [Clearing conditions]] • Writing of 0 to RDRF after having read RDRF as 1. • Issuing of an RXI interrupt request and allowing DMAC or DTC to read data from RDR. The RDRF flag is not affected and retains its previous value when the RE bit in SCR is cleared to 0. Note that when the next serial reception is completed while the RDRF flag is being set to 1, an overrun error occurs and the received data is lost.
5	ORER	Undefined	R/(W)*	Overrun Error [Setting condition] • Occurrence of overrun error during reception [Clearing condition] • Writing of 0 after having read ORER as 1

Note: * Only 0 can be written here, to clear the flags

- Receive data register_0 (RDR_0) Number of bits: 8 Address: H'FFFF85

Function: An 8-bit register for storing receive data.
Setting: Undefined

5. Flowchart


sci0_rcv1byte
RE = 1
ORER bit in SSR_0 = 1?
Yes
ORER = 1
No
tmp ← RDRF bit in SSR_0
tmp = 0
Yes
tmp = 0?
No
rtn = RDR_0
tmp = SSR_0
RDRF = 0
RE = 0
Return rtn (received data).
End

7.2.3 sci0_rcvbyte Function

1. Functional overview

Receives n byte of clock synchronous serial data.

2. Arguments

Type	Variable Name	Description
unsigned char	dtno	No. of receive bytes
unsigned char	*ram	The start address of RAM in which the receive data is stored.

3. Return value

None

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Serial control register_0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
4	RE	0	R/W	Receive Enable 0: Disables reception. 1: Enabled reception.

- Serial status register_0 (SSR_0) Number of bits: 8 Address: H'FFFF84

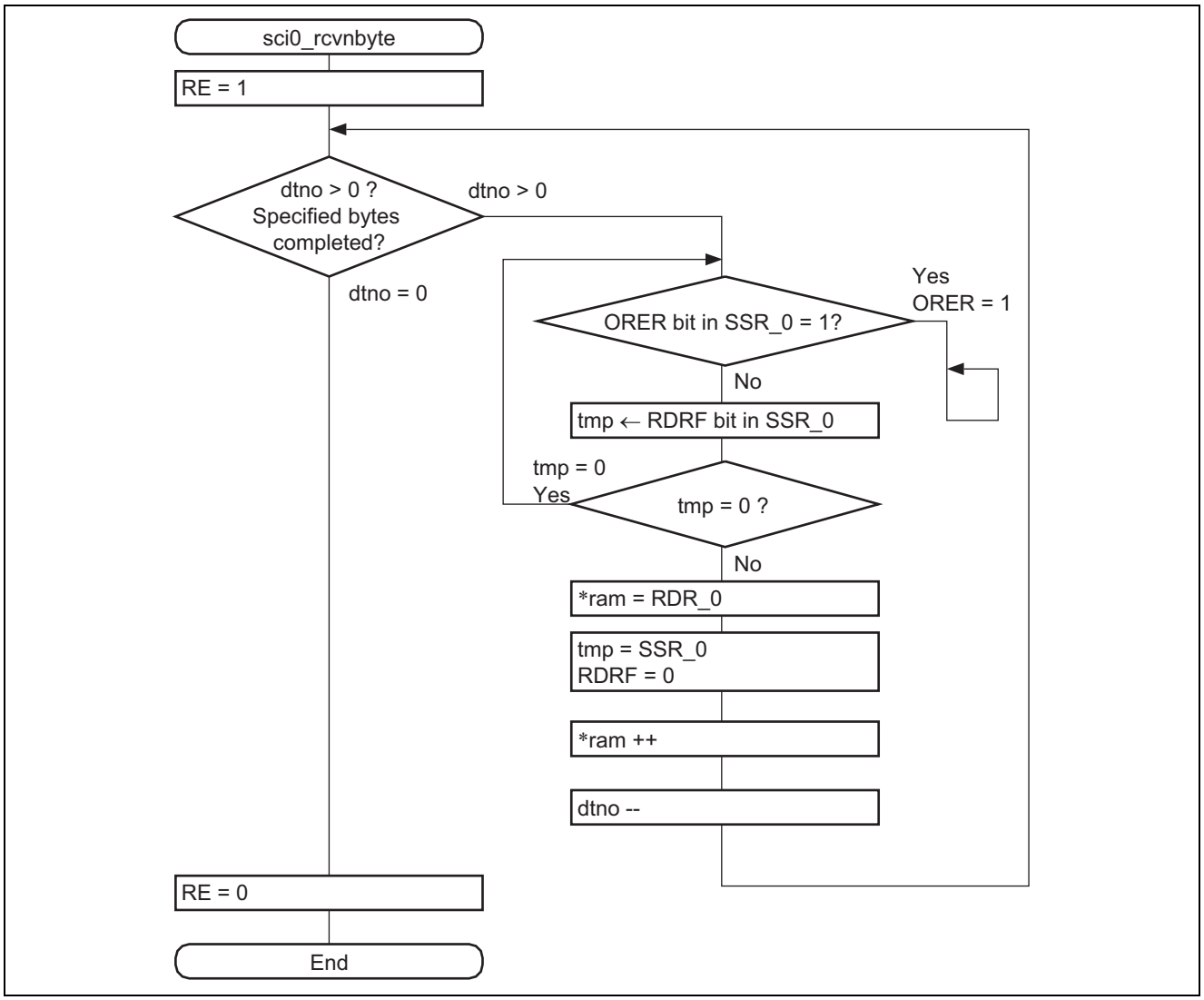
Bit	Bit Name	Setting	R/W	Description
6	RDRF	Undefined	R/(W)*	Receive Data Register Full Indicates whether receive data is stored in RDR. [Setting condition] • Normal ending of serial reception and transferring of receive data from RSR to RDR [Clearing conditions] • Writing of 0 to RDRF after having read RDRF as 1. • Issuing of an RXI interrupt request and allowing DMAC or DTC to read data from RDR. The RDRF flag is not affected and retains its previous value when the RE bit in SCR is cleared to 0. Note that when the next serial reception is completed while the RDRF flag is being set to 1, an overrun error occurs and the received data is lost.
5	ORER	Undefined	R/(W)*	Overrun Error [Setting condition] • Occurrence of overrun error during reception [Clearing condition] • Writing of 0 after having read ORER as 1

Note: * Only 0 can be written here, to clear the flags.

- Receive data register_0 (RDR_0) Number of bits: 8 Address: H'FFFF85

Function: An 8-bit register for storing receive data.
Setting: Undefined

5. Flowchart


sci0_rcvnbyte
RE = 1
dtno > 0 ?
Specified bytes
completed?
dtno > 0
dtno = 0
ORER bit in SSR_0 = 1?
Yes
ORER = 1
No
tmp ← RDRF bit in SSR_0
tmp = 0
Yes
tmp = 0 ?
No
*ram = RDR_0
tmp = SSR_0
RDRF = 0
*ram ++
dtno --
RE = 0
End

7.2.4 sci0_trs1byte Function

1. Functional overview

Receives one byte of clock synchronous serial data.

2. Arguments

Type	Variable Name	Description
unsigned char	tdt	Transmits one byte of data

3. Return value

None

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Serial control register_0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
5	TE	0	R/W	Transmit Enable 0: Disables transmission. 1: Enables transmission.

- Transmit data register_0 (TDR_0) Number of bits: 8 Address: H'FFFF83

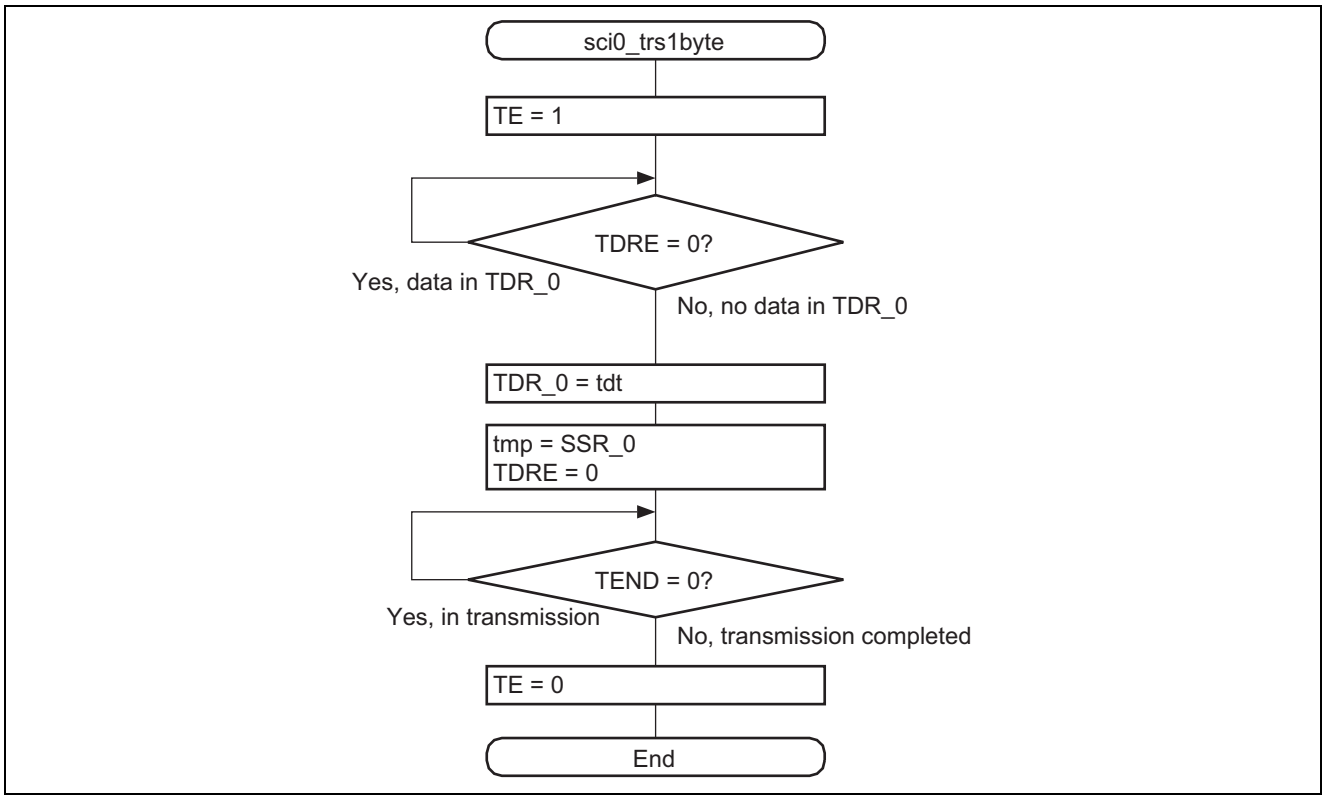
Function: An 8-bit register of storing transmit data.
Setting: Undefined.

- Serial status register_0 (SSR_0) Number of bits: 8 Address: H'FFFF84

Bit	Bit Name	Setting	R/W	Description
7	TDRE	Undefined	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] • Clearing of TE bit in SCR to 0. • Transferring of data from TDR to TSR. [Clearing conditions] • Writing of 0 to TDRE after having read TDRE as 1. • Generation of a TXI interrupt request allowing DMAC to transfer transmitted data to TDR.
2	TEND	Undefined	R	Transmit End [Setting conditions] • Clearing of TE bit in SCR to 0. • TDRE = 1 after a specified time passed after completion of 1-byte data transfer. [Clearing conditions] • Writing of 0 to TDRE after having read TDRE as 1 • Generation of a TXI interrupt request allowing DMAC to write transmitted data to TDR.

Note: * Only 0 can be written here, to clear the flag for TDRE.

5. Flowchart


sci0_trs1byte
TE = 1
TDRE = 0?
Yes, data in TDR_0
No, no data in TDR_0
TDR_0 = tdt
tmp = SSR_0
TDRE = 0
TEND = 0?
Yes, in transmission
No, transmission completed
TE = 0
End

7.2.5 sci0_trsnbyte Function

1. Functional overview

Transmits n bytes of clock synchronous serial data.

2. Arguments

Type	Variable Name	Description
unsigned short	dtno	Transmit size
unsigned char	*tdt	Start address of the transmit data

3. Return value

None

4. Internal registers used

The internal registers used in this sample task are described below. Note that the settings shown below are not the initial values but the values used in this sample task.

- Serial control register_0 (SCR_0) Number of bits: 8 Address: H'FFFF82

Bit	Bit Name	Setting	R/W	Description
5	TE	0	R/W	Transmit Enable 0: Disables transmission. 1: Enables transmission.

- Transmit data register_0 (TDR_0) Number of bits: 8 Address: H'FFFF83

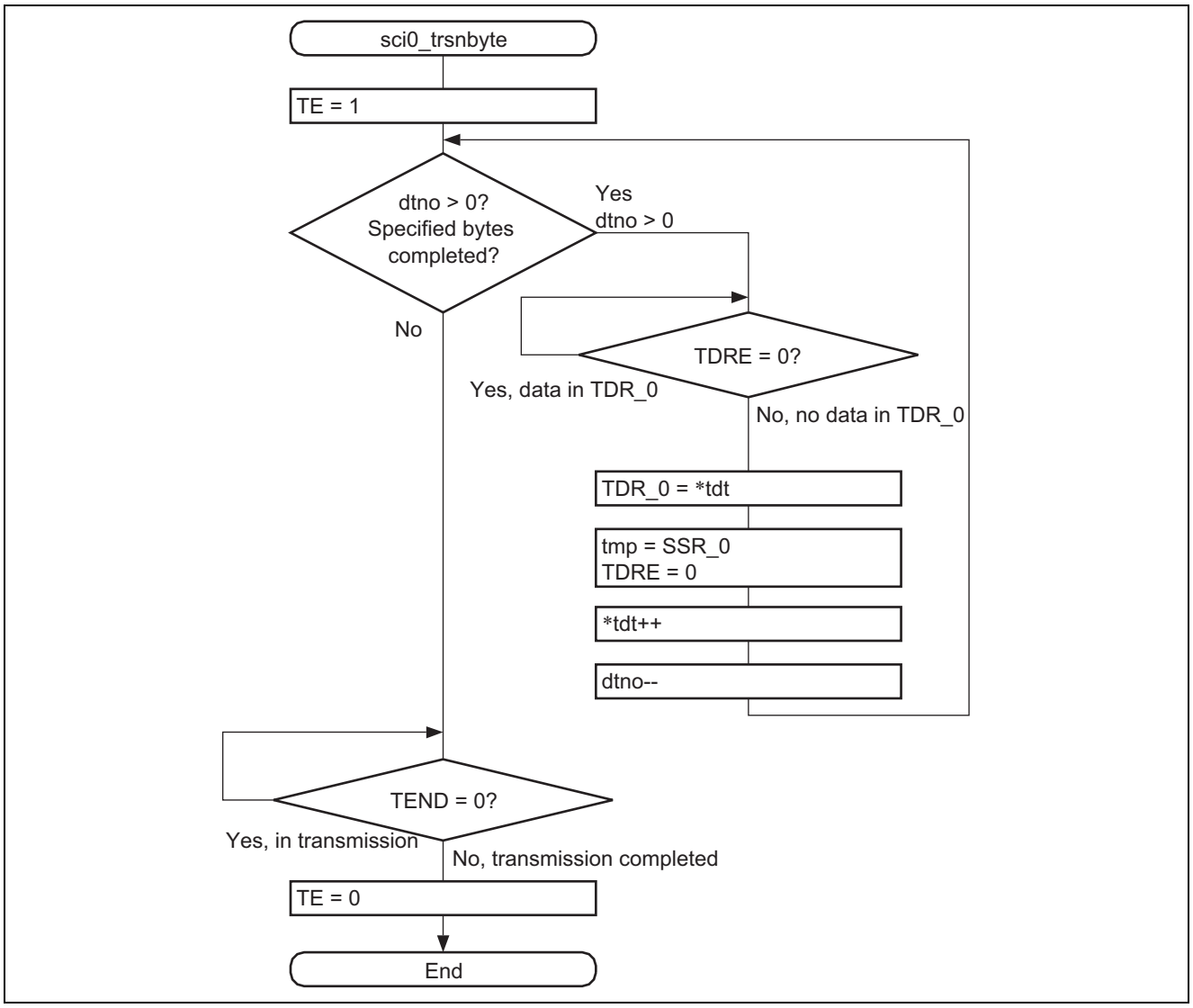
Function: An 8-bit register of storing transmit data.
Setting: Undefined.

- Serial status register_0 (SSR_0) Number of bits: 8 Address: H'FFFF84

Bit	Bit Name	Setting	R/W	Description
7	TDRE	Undefined	R/(W)*	Transmit Data Register Empty Indicates whether TDR contains transmit data. [Setting conditions] • Clearing of TE bit in SCR to 0. • Transferring of data from TDR to TSR. [Clearing conditions] • Writing of 0 to TDRE after having read TDRE as 1. • Generation of a TXI interrupt request allowing DMAC to transfer transmitted data to TDR.
2	TEND	Undefined	R	Transmit End [Setting conditions] • Clearing of TE bit in SCR to 0. • TDRE = 1 after a specified time passed after completion of 1-byte data transfer. [Clearing conditions] • Writing of 0 to TDRE after having read TDRE as 1 • Generation of a TXI interrupt request allowing DMAC to write transmitted data to TDR.

Note: * Only 0 can be written here, to clear the flag for TDRE.

5. Flowchart


sci0_trsnbyte
TE = 1
dtno > 0?
Specified bytes
completed?
Yes
dtno > 0
No
TDRE = 0?
Yes, data in TDR_0
No, no data in TDR_0
TDR_0 = *tdt
tmp = SSR_0
TDRE = 0
*tdt++
dtno--
TEND = 0?
Yes, in transmission
No, transmission completed
TE = 0
End

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	Jul.20.07	—	First edition issued

Notes regarding these materials

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

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