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.

R8C/35C Group

Rewriting the Data Flash (Flash Memory Ready Interrupt)

1. Abstract

This document describes the setting method and an application example for rewriting the data flash using the flash memory ready interrupt (flash ready status interrupt) in the R8C/35C Group.

2. Introduction

The application example described in this document applies to the following MCU:

• MCU	: R8C/35C Group

This program can be used with other R8C/Tiny Series MCUs which have the same special function registers (SFRs) as the R8C/35C Group. Check the manual for any additions and modifications to functions. Careful evaluation is recommended before using this application note.

3. Application Example

When rewriting the flash memory (rewrite or erase) in EW1 mode, the following show the differences depending on flash memory areas:

- Program ROM area: CPU is in hold state (I/O ports retain their states before the command is executed).
- Data flash area: CPU is in operation state due to a background operation (BGO).

When rewriting the data flash area, other processes can be performed during a write or erase operation. Use the flash memory ready interrupt to generate interrupts when: a write operation is completed, an erase operation is completed, an error occurs, etc. In this application note, when executing the block erase, use the flash ready status interrupt to generate interrupts at completion of auto-erasure. During interrupt handling, rewriting status check and data flash block is disabled, and CPU rewrite mode is disabled.

3.1 Program Outline

The switch (SW1) and LEDs (LED0 to LED3) of the Renesas Starter Kit for R8C/35C Group are used to direct write data to a record (1) and display the number of writes. When detecting the switch is pressed, data for one record is written to an empty record (1). The same process is performed every time the pressed switch is detected and the number of writes are counted. However, during a write operation or an erase operation, the presses switch detection is ignored. The lower 4 bits of the number of data writes are displayed on the LEDs (LED0 to LED3). When each bit is 1, the LEDs are turned on. When each bit is 0, the LEDs are turned off. Auto-programming completion is detected by the FST7 bit in the FST register and auto-erasure completion is detected by the flash memory ready interrupt.

Note:
1. Details for the records and empty record search are described below.

Figure 3.1 shows an example of a key and an LED connections. Table 3.1 lists the pins used and their functions.

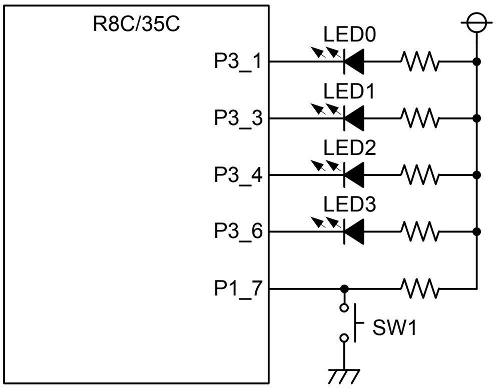


Figure 3.1 Key and LED Connections

Table 3.1 Pins and Their Functions

Pin Name	I/O	Function
P1_7	Input	Switch SW1 input
P3_1	Output	LED0 output (bit 3 value of the number of writes)
P3_3	Output	LED1 output (bit 2 value of the number of writes)
P3_4	Output	LED2 output (bit 1 value of the number of writes)
P3_6	Output	LED3 output (bit 0 value of the number of writes)

3.2 Data Flash Area

In this application note, one record is 64 bytes and blocks are divided by 16. There is a total of 64 records in blocks A to D. Figure 3.2 shows the relationship between the data flash and the records.

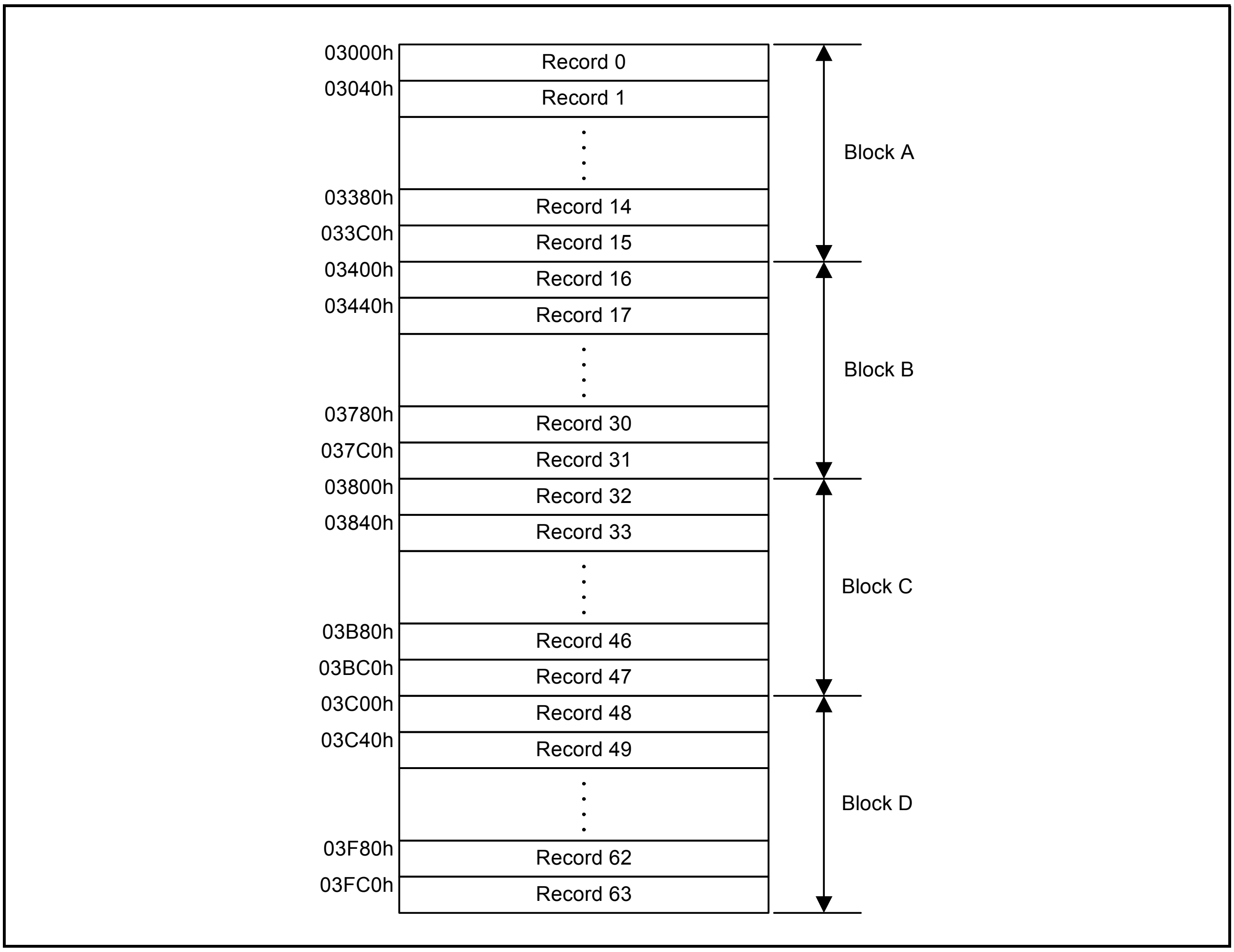


Figure 3.2 Relationship between Data Flash and Records

3.2.1 Empty Record Search (Data FFh Search)

Data written to the data flash is retained even if the power is turned off. Records (empty records) in which all data are FFh, are searched after a reset start. Search methods for empty records are described below.

(1) Set the search pointer to the starting address in record 0.

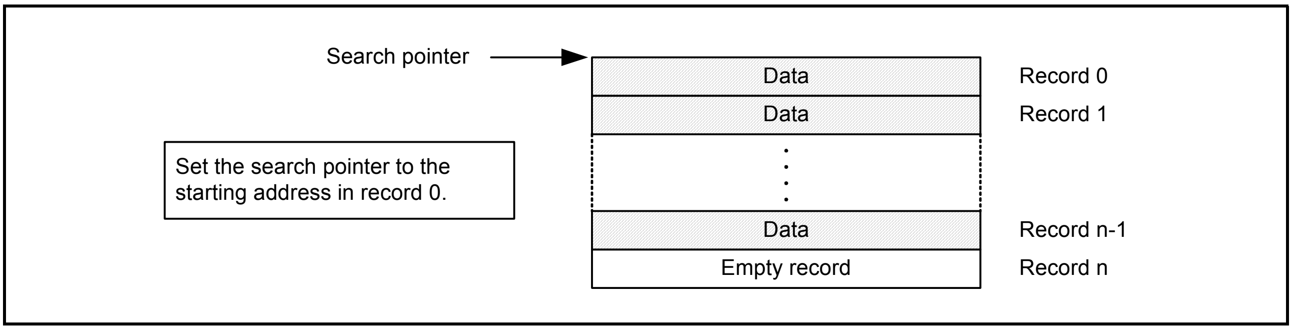


Figure 3.3 Set the Search Pointer

(2) Check to see that the record the search pointer indicates is an empty record.

(3) When the record is not empty, set the search pointer to the starting address in the next record.

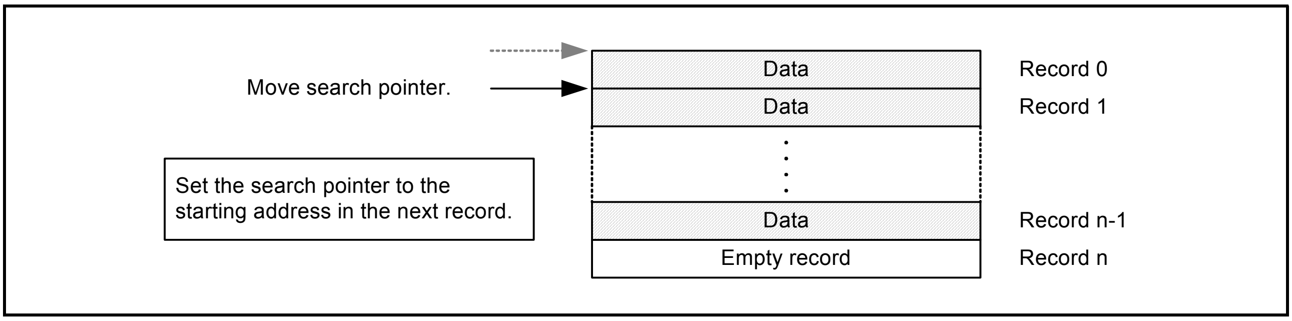


Figure 3.4 Moving the Search Pointer

(4) Repeat steps (2) and (3) until an empty record is found or all records are checked.

(5) When an empty record is found, set the starting address in the empty record to the data write address and set the block in which the empty record is stored as a used block.

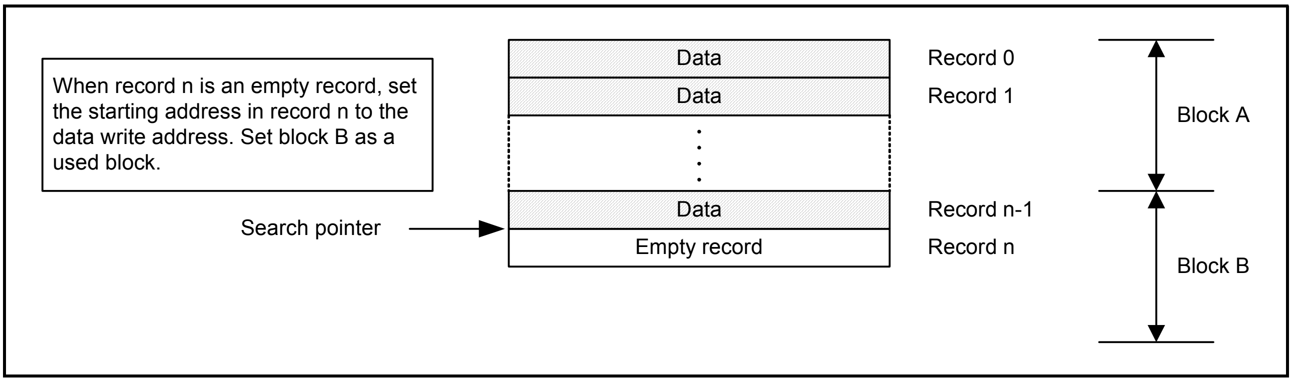

When record n is an empty record, set the starting address in record n to the data write address. Set block B as a used block.
Search pointer
Data
Data
Data
Empty record
Record 0
Record 1
Record n-1
Record n
Block A
Block B

Figure 3.5 When an Empty Record is Found in Block B

(6) When an empty record is not found in any block, erase block A, set the starting address in record 0 to the data write address, and set block A as a used block.

3.2.2 Write Record and Erase Block

Write records sequentially based on the data write address and used blocks received by the empty record search.
When writing data up to record 15, erase (block erase) all data in the next block (block B). When writing data to the record in the next step, start writing from record 16. When writing data up to the last record in each block, erase (block erase) all data in the next block. When writing data up to record 63, return to block A,
erase (block erase) all data in block A, and write data from record 0 again.
When auto-erasing (erase) blocks, enable the flash ready status interrupt and generate it after auto-erasure is completed.
The following are performed in the flash memory ready interrupt handling:
(1) The flash memory ready interrupt is disabled.
(2) The flash ready status interrupt is disabled.
(3) The flash ready status interrupt request flag is cleared.
(4) The full status check is performed.
(5) The rewrite disable bit for a used block is set.
(6) CPU rewrite mode is disabled.

3.3 Memory

Table 3.2 Memory

Memory	Size	Remarks
ROM	1081 bytes	In the rej05b1199_src.c module
RAM	72 bytes	In the rej05b1199_src.c module
Maximum user stack	28 bytes	
Maximum interrupt stack	25 bytes	

Memory size varies depending on the C compiler version and compile options. The above applies to the following conditions:

C compiler: M16C/60, 30, 20, 10, and Tiny and R8C/Tiny Series Compiler V.5.45 Release 00
Compile option: -c -finfo (see Note below) -dir “$(CONFIGDIR)” -R8C

4. Software

This section shows the initial setting procedures and values to set the example described in section **3. Application Example**. Refer to the latest **R8C/35C Group Hardware Manual** for details on individual registers.

The × in the register's Setting Value represents bits not used in this application, blank spaces represent bits that do not change, and the dash represents reserved bits or bits that have nothing assigned.

4.1 Function Tables

Declaration	void main(void)		
Outline	Main handling		
Argument	Argument name		Meaning
	None		—
Variable (global)	Variable name		Contents
	unsigned char status		Full status check result
Returned value	Type	Value	Meaning
	None	—	—
Function	When the switch is pressed and CPU rewrite mode is disabled, control writing the data.		

Declaration	void mcu_init(void)		
Outline	System clock setting		
Argument	Argument name		Meaning
	None		—
Variable (global)	Variable name		Contents
	None		—
Returned value	Type	Value	Meaning
	None	—	—
Function	Set the system clock (high-speed on-chip oscillator).		

Declaration	void write_address_init(void)		
Outline	Record write address initial setting		
Argument	Argument name		Meaning
	None		—
Variable (global)	Variable name		Contents
	unsigned char *write_addr		Write address
	unsigned char block_select		Block selected
Returned value	Type	Value	Meaning
	None	—	—
Function	Search for an empty record and set a block which has an empty record as a used block (block_select). Set the starting address in the empty record to the write address (write_addr).		

Declaration	unsigned char write_control(void)		
Outline	Data write control		
Argument	Argument name		Meaning
	None		—
Variable (global)	Variable name		Contents
	unsigned char record_data[]		Record data
	unsigned char write_cnt		Number of writes
	unsigned char *write_addr		Write address
	unsigned char block_select		Block selected
Returned value	Type	Value	Meaning
	unsigned char	NORMAL	Completed normally
		CMD_SEQ_ERROR	Command sequence error
		ERS_BLK_CHK_ERROR	Erase/blank check error
		PROGRAM_ERROR	Program error
Function	Set the write data in the write data making and write the record data. After writing the record data, set the starting address in the next record to the write address (write_addr). When writing data to the last record of each block, perform the block erase to the block to be written next and set its block as a used block (block_select). Set the starting address of the block to the write address (write_addr). If the write is successful, increment the number of writes (write_cnt).		

Declaration	void set_data(unsigned char *data)		
Outline	Write data made		
Argument	Argument name		Meaning
	unsigned char *data		Write data starting address
Variable (global)	Variable name		Contents
	None		—
Returned value	Type	Value	Meaning
	None	—	—
Function	Make the record data written to the data flash. No handling is performed in this application note. Add handlings based on the user system.		

Declaration	void block_erase(unsigned char block_no)		
Outline	Block erase		
Argument	Argument name		Meaning
	unsigned char block_no		Erase block number
Variable (global)	Variable name		Contents
	unsigned char *ers_addr		Erase address
Returned value	Type	Value	Meaning
	None	—	—
Function	Erase the specified block in CPU rewrite mode (EW1 mode). Start auto-erasure and do not wait until auto-erasure is completed in this function.		

Declaration	unsigned char data_write(unsigned char *data)		
Outline	Data written		
Argument	Argument name		Meaning
	unsigned char *data		Write data starting address
Variable (global)	Variable name		Contents
	unsigned char block_select		Block selected
	unsigned char *write_addr		Write address
Returned value	Type	Value	Meaning
	unsigned char	NORMAL	Completed normally
		CMD_SEQ_ERROR	Command sequence error
		ERS_BLK_CHK_ERROR	Erase/blank check error
		PROGRAM_ERROR	Program error
		FLASH_BUSY	Flash memory busy state
Function	Write data for one record from the write address (write_addr) in CPU rewrite mode (EW1 mode).		

Declaration	unsigned char full_sts_chk(unsigned char *chk_adr)		
Outline	Full status check		
Argument	Argument name		Meaning
	unsigned char *chk_adr		Address where erase command or program command data is written
Variable (global)	Variable name		Contents
	None		—
Returned value	Type	Value	Meaning
	unsigned char	NORMAL	Completed normally
		CMD_SEQ_ERROR	Command sequence error
		ERS_BLK_CHK_ERROR	Erase/blank check error
		PROGRAM_ERROR	Program error
Function	Perform full status check.		

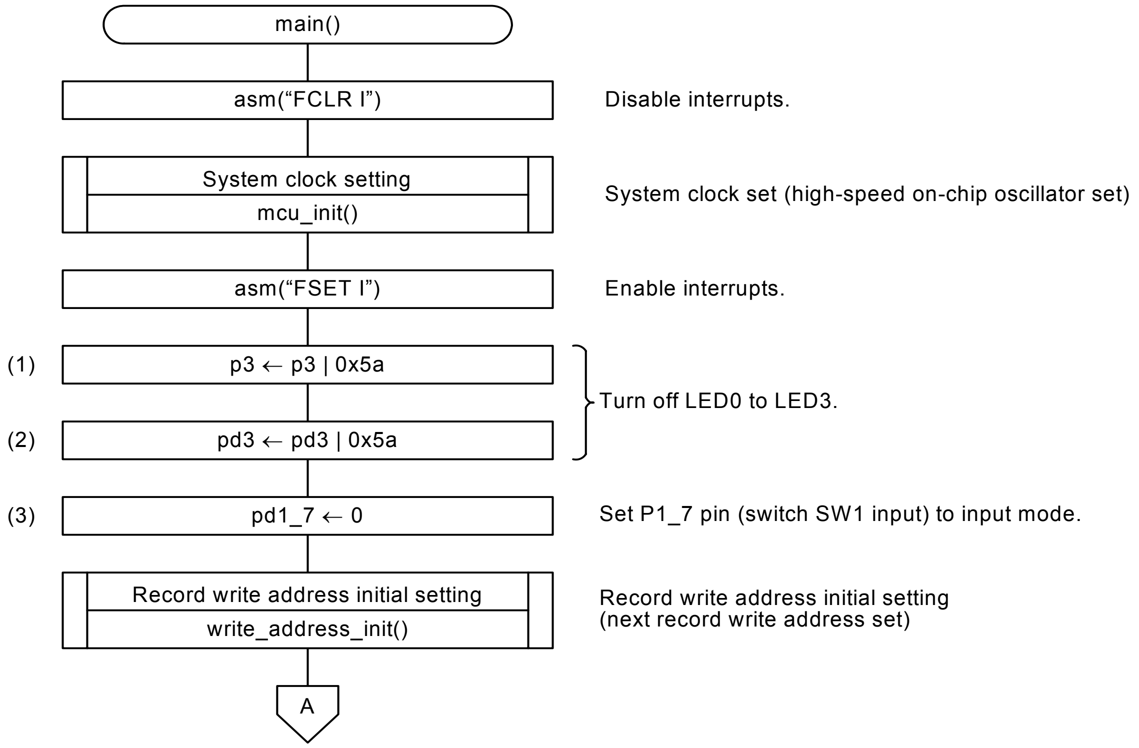
Declaration	void _flash_memory_ready(void)		
Outline	Flash memory ready interrupt		
Argument	Argument name		Meaning
	None		—
Variable (global)	Variable name		Contents
	unsigned char *ers_addr		Erase address
	unsigned char status		Full status check result
	unsigned char block_select		Block selected
Returned value	Type	Value	Meaning
	None	—	—
Function	An interrupt is generated when auto-erasure is completed. Disable rewriting the data flash block and disable CPU rewrite.		

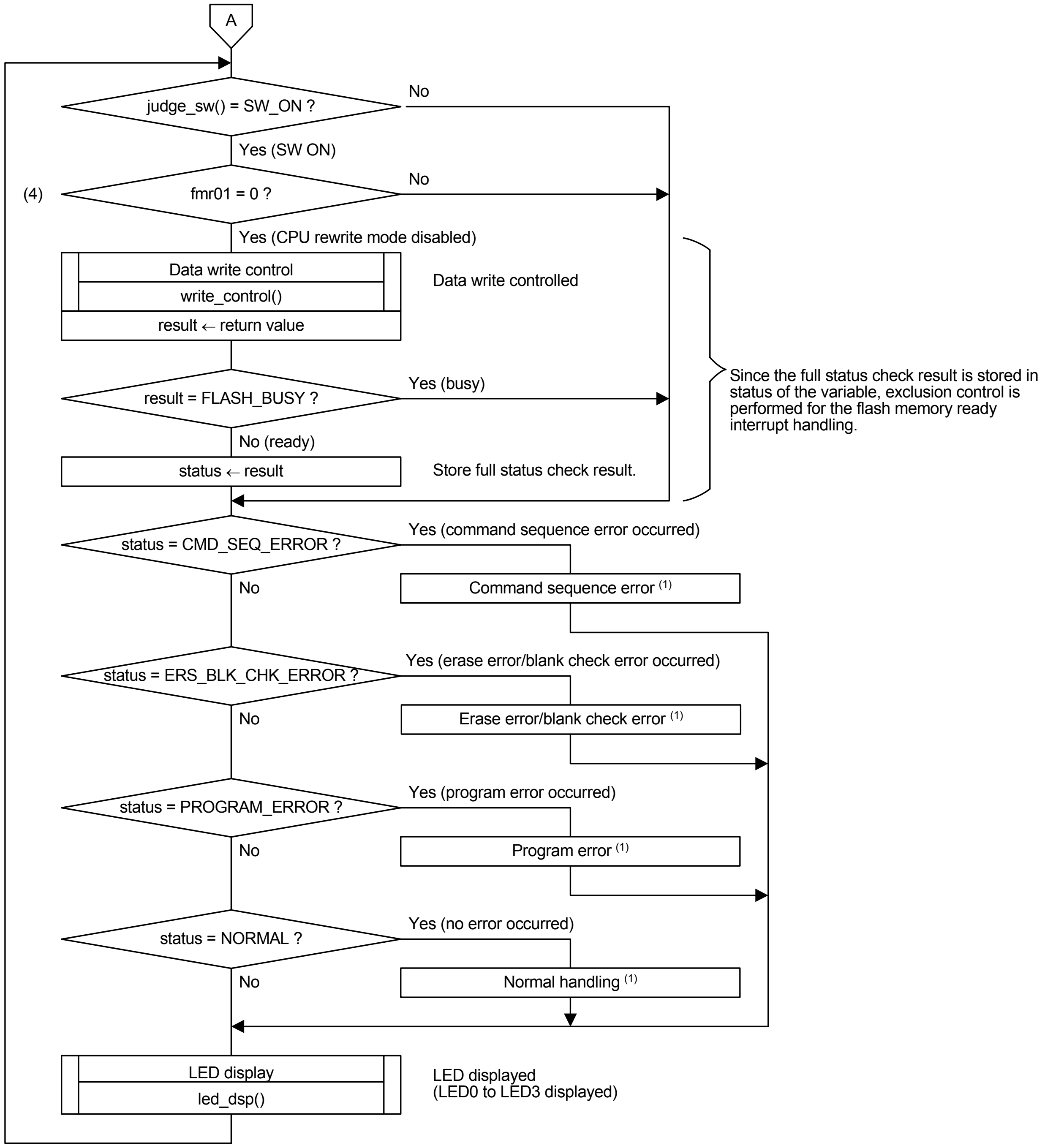
Declaration	unsigned char judge_sw(void)		
Outline	SW input judgment handling		
Argument	Argument name		Meaning
	None		—
Variable (global)	Variable name		Contents
	None		—
Returned value	Type	Value	Meaning
	unsigned char	SW_ON	SW input
		SW_OFF	No SW input
Function	Judge switch input and return the result.		

Declaration	void led_dsp(void)		
Outline	LED display		
Argument	Argument name		Meaning
	None		—
Variable (global)	Variable name		Contents
	unsigned char write_cnt		Number of writes
Returned value	Type	Value	Meaning
	None	—	—
Function	The lower 4 bits of the number of data writes (write_cnt) are displayed on LED0 to LED3. When bit 0 is 0, turn off LED3 (high level). When bit 0 is 1, turn on LED3 (low level). When bit 1 is 0, turn off LED2 (high level). When bit 1 is 1, turn on LED2 (low level). When bit 2 is 0, turn off LED1 (high level). When bit 2 is 1, turn on LED1 (low level). When bit 3 is 0, turn off LED0 (high level). When bit 3 is 1, turn on LED0 (low level).		

4.2 Main Function

- Flowchart


main()
asm(“FCLR I”)
Disable interrupts.
System clock setting
mcu_init()
System clock set (high-speed on-chip oscillator set)
asm(“FSET I”)
Enable interrupts.
(1)
p3 ← p3 | 0x5a
(2)
pd3 ← pd3 | 0x5a
Turn off LED0 to LED3.
(3)
pd1_7 ← 0
Set P1_7 pin (switch SW1 input) to input mode.
Record write address initial setting
write_address_init()
Record write address initial setting
(next record write address set)
A


A
judge_sw() = SW_ON ?
No
Yes (SW ON)
(4)
fmr01 = 0 ?
No
Yes (CPU rewrite mode disabled)
Data write control
write_control()
result ← return value
Data write controlled
result = FLASH_BUSY ?
Yes (busy)
No (ready)
status ← result
Store full status check result.
Since the full status check result is stored in status of the variable, exclusion control is performed for the flash memory ready interrupt handling.
status = CMD_SEQ_ERROR ?
Yes (command sequence error occurred)
No
Command sequence error (1)
status = ERS_BLK_CHK_ERROR ?
Yes (erase error/blank check error occurred)
No
Erase error/blank check error (1)
status = PROGRAM_ERROR ?
Yes (program error occurred)
No
Program error (1)
status = NORMAL ?
Yes (no error occurred)
No
Normal handling (1)
LED display
led_dsp()
LED displayed
(LED0 to LED3 displayed)

Note:

1. These handlings are not performed in this application note. Perform these handlings based on the user system.

- Register settings

(1) Set P3_1 (LED0 output), P3_3 (LED1 output), P3_4 (LED2 output), and P3_6 (LED3 output) to high.

Port P3 Register (P3)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	x	1	x	1	1	x	1	x

Bit	Symbol	Bit Name	Function	R/W
b1	P3_1	Port P3_1 bit	1: High level	R/W
b3	P3_3	Port P3_3 bit	1: High level	R/W
b4	P3_4	Port P3_4 bit	1: High level	R/W
b6	P3_6	Port P3_6 bit	1: High level	R/W

(2) Set P3_1 (LED0 output), P3_3 (LED1 output), P3_4 (LED2 output), and P3_6 (LED3 output) as output ports.

Port P3 Direction Register (PD3)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	x	1	x	1	1	x	1	x

Bit	Symbol	Bit Name	Function	R/W
b1	PD3_1	Port P3_1 direction bit	1: Output mode (functions as output port)	R/W
b3	PD3_3	Port P3_3 direction bit	1: Output mode (functions as output port)	R/W
b4	PD3_4	Port P3_4 direction bit	1: Output mode (functions as output port)	R/W
b6	PD3_6	Port P3_6 direction bit	1: Output mode (functions as output port)	R/W

(3) Set P1_7 (switch SW1 input) as an input port.

Port P1 Direction Register (PD1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	0	x	x	x	x	x	x	x

Bit	Symbol	Bit Name	Function	R/W
b7	PD1_7	Port P1_7 direction bit	0: Input mode (functions as input port)	R/W

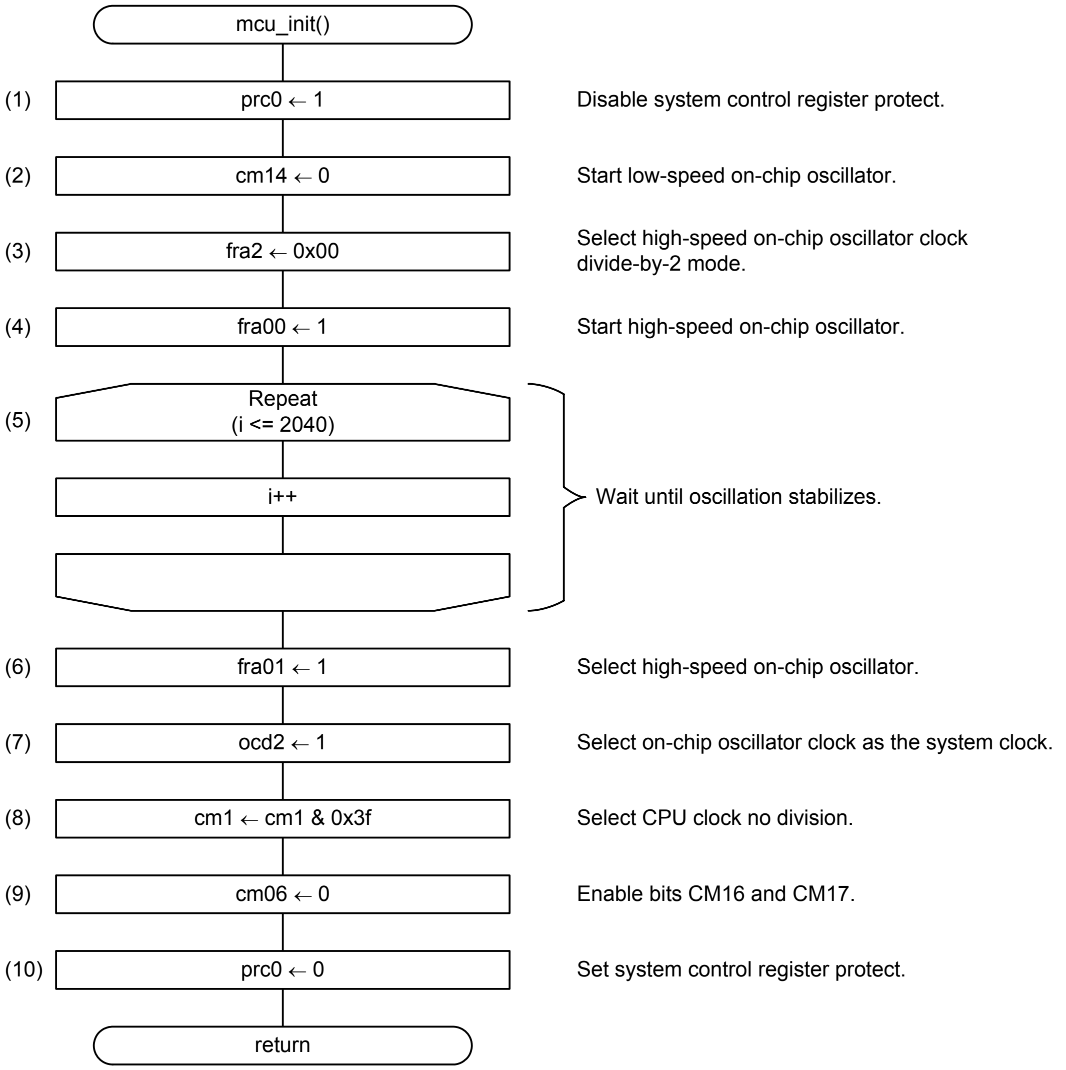
(4) Verify that CPU rewrite mode is disabled (record write or block erase is completed).

Flash Memory Control Register 0 (FMR0)

Bit	Symbol	Bit Name	Function	R/W
b1	FMR01	CPU rewrite mode select bit	0: CPU rewrite mode disabled 1: CPU rewrite mode enabled	R/W

4.3 System Clock Setting

- Flowchart


mcu_init()
(1)
prc0 ← 1
Disable system control register protect.
(2)
cm14 ← 0
Start low-speed on-chip oscillator.
(3)
fra2 ← 0x00
Select high-speed on-chip oscillator clock divide-by-2 mode.
(4)
fra00 ← 1
Start high-speed on-chip oscillator.
(5)
Repeat
(i <= 2040)
i++
Wait until oscillation stabilizes.
(6)
fra01 ← 1
Select high-speed on-chip oscillator.
(7)
ocd2 ← 1
Select on-chip oscillator clock as the system clock.
(8)
cm1 ← cm1 & 0x3f
Select CPU clock no division.
(9)
cm06 ← 0
Enable bits CM16 and CM17.
(10)
prc0 ← 0
Set system control register protect.
return

- Register settings

(1) Enable writing to registers CM0, CM1, CM3, OCD, FRA0, FRA1, FRA2, and FRA3.

Protect Register (PRCR)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	—	—	—	—	x	x	x	1

Bit	Symbol	Bit Name	Function	R/W
b0	PRC0	Protect bit 0	Enables writing to registers CM0, CM1, CM3, OCD, FRA0, FRA1, FRA2, and FRA3. 1: Write enabled	R/W

(2) Start the low-speed on-chip oscillator.

System Clock Control Register 1 (CM1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value			—	0	x	x	x	x

Bit	Symbol	Bit Name	Function	R/W
b4	CM14	Low-speed on-chip oscillator stop bit	0: Low-speed on-chip oscillator on	R/W

(3) Set the division ratio for the high-speed on-chip oscillator.

High-Speed On-Chip Oscillator Control Register 2 (FRA2)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	—	—	—	—	—	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	FRA20	High-speed on-chip oscillator frequency switching bit	Division selection These bits select the division ratio for the high-speed on-chip oscillator clock. b2 b1 b0 0 0 0: Divide-by-2 mode	R/W
b1	FRA21			R/W
b2	FRA22			R/W

(4) Start the high-speed on-chip oscillator.

High-Speed On-Chip Oscillator Control Register 0 (FRA0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	—	—	—	—	x	—		1

Bit	Symbol	Bit Name	Function	R/W
b0	FRA00	High-speed on-chip oscillator enable bit	1: High-speed on-chip oscillator on	R/W

(5) Wait until oscillation stabilizes.

(6) Select the high-speed on-chip oscillator.

High-Speed On-Chip Oscillator Control Register 0 (FRA0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	—	—	—	—	x	—	1	

Bit	Symbol	Bit Name	Function	R/W
b1	FRA01	High-speed on-chip oscillator select bit	1: High-speed on-chip oscillator selected	R/W

(7) Select the on-chip oscillator clock as the system clock.

Oscillation Stop Detection Register (OCD)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	—	—	—	—	x	1	x	x

Bit	Symbol	Bit Name	Function	R/W
b2	OCD2	System clock select bit	1: On-chip oscillator clock selected	R/W

(8) Set the CPU clock division select bit 1.

System Clock Control Register 1 (CM1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	0	0	—		x	x	x	x

Bit	Symbol	Bit Name	Function	R/W
b6	CM16	CPU clock division select bit 1	b7 b6 0 0: No division mode	R/W
b7	CM17			R/W

(9) Set the CPU clock division select bit 0.

System Clock Control Register 0 (CM0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	x	0	x	x	x	x	—	—

Bit	Symbol	Bit Name	Function	R/W
b6	CM06	CPU clock division select bit 0	0: Bits CM16 and CM17 in CM1 register enabled	R/W

(10) Disable writing to registers CM0, CM1, CM3, OCD, FRA0, FRA1, FRA2, and FRA3.

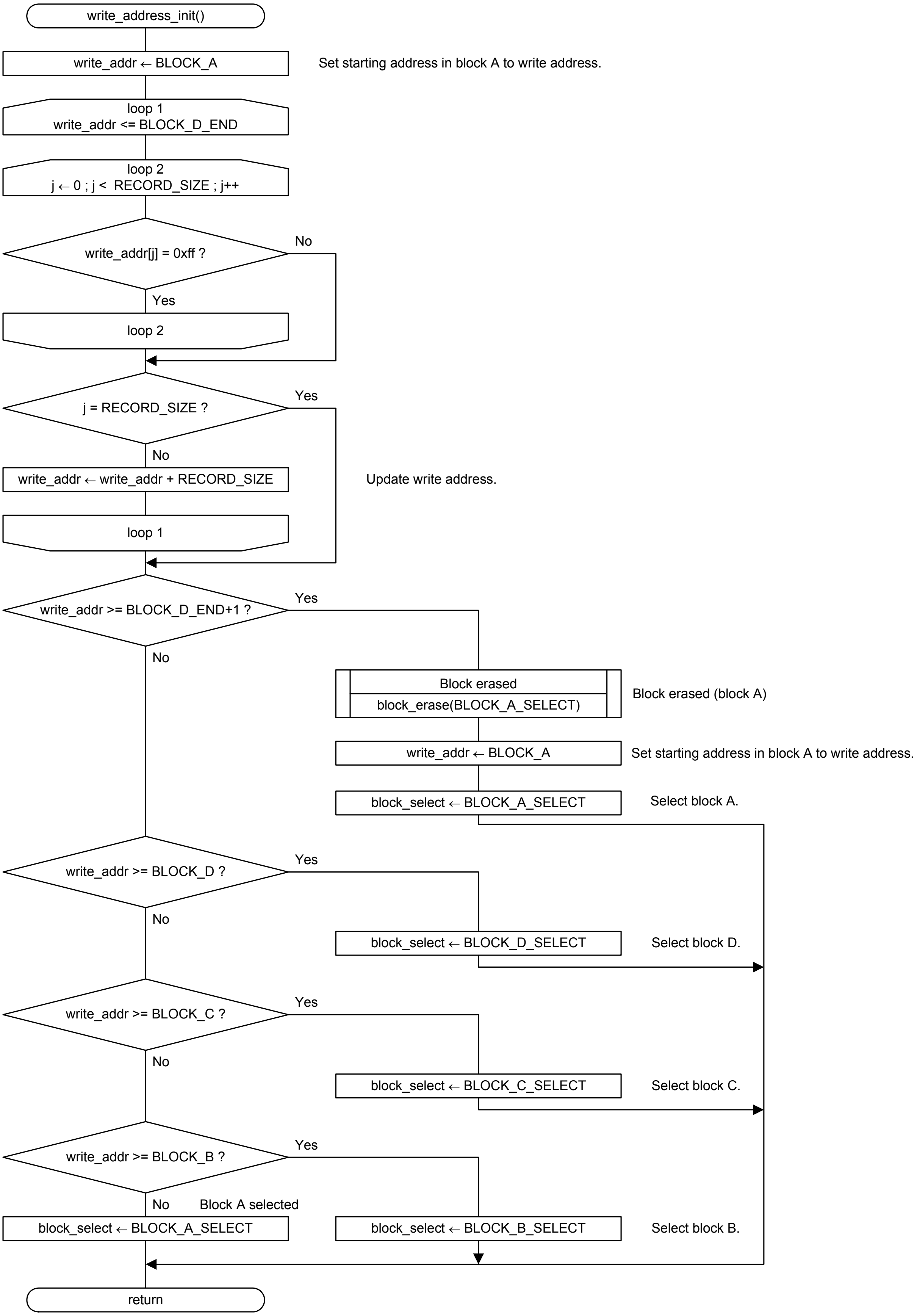
Protect Register (PRCR)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	—	—	—	—	x	x	x	0

Bit	Symbol	Bit Name	Function	R/W
b0	PRC0	Protect bit 0	Enables writing to registers CM0, CM1, CM3, OCD, FRA0, FRA1, FRA2, and FRA3. 0: Write disabled	R/W

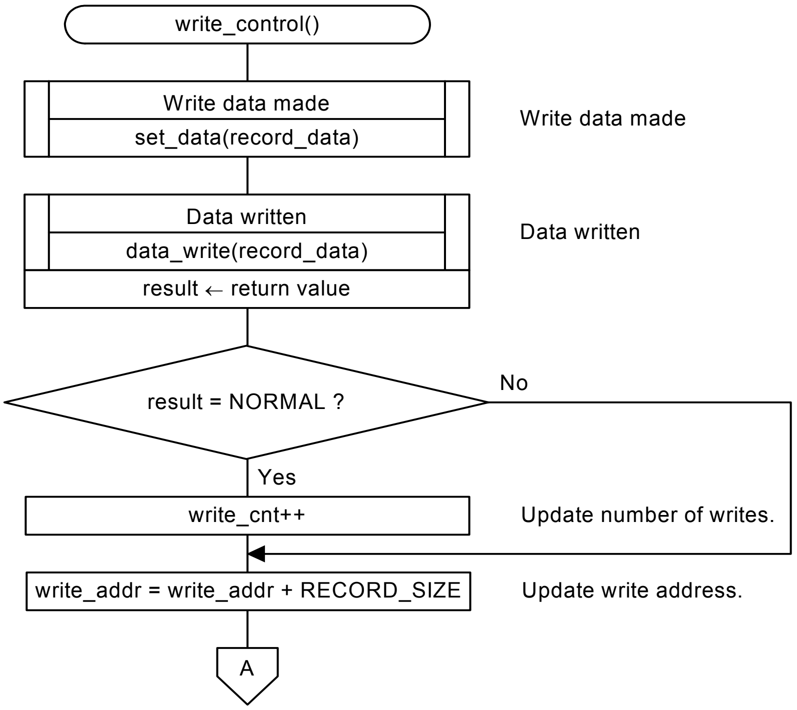
4.4 Record Write Address Initial Setting

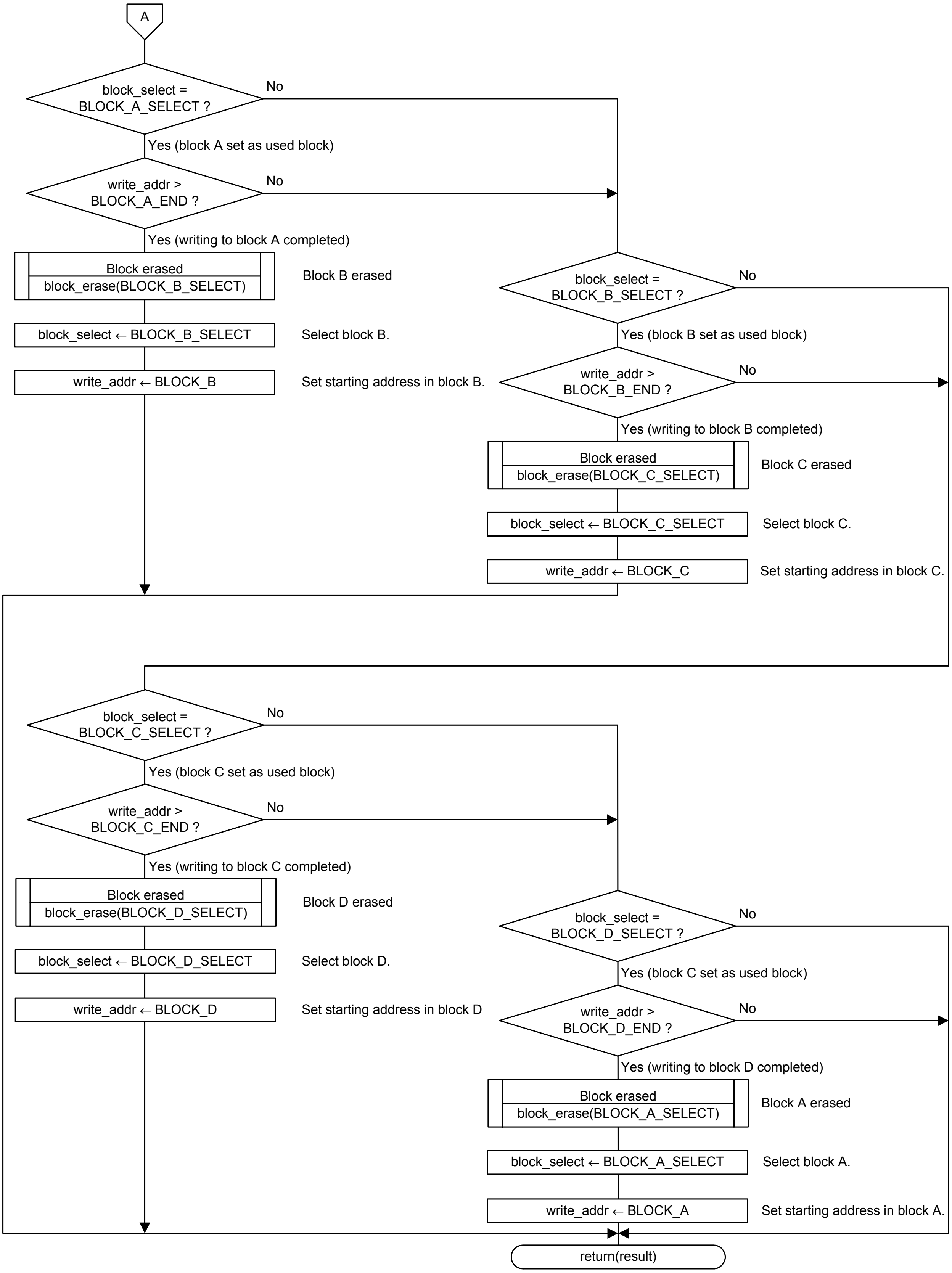
- Flowchart


write_address_init()
write_addr ← BLOCK_A
Set starting address in block A to write address.
loop 1
write_addr <= BLOCK_D_END
loop 2
j ← 0 ; j < RECORD_SIZE ; j++
write_addr[j] = 0xff ?
No
Yes
loop 2
j = RECORD_SIZE ?
Yes
No
write_addr ← write_addr + RECORD_SIZE
Update write address.
loop 1
write_addr >= BLOCK_D_END+1 ?
Yes
No
Block erased
block_erase(BLOCK_A_SELECT)
Block erased (block A)
write_addr ← BLOCK_A
Set starting address in block A to write address.
block_select ← BLOCK_A_SELECT
Select block A.
write_addr >= BLOCK_D ?
Yes
No
block_select ← BLOCK_D_SELECT
Select block D.
write_addr >= BLOCK_C ?
Yes
No
block_select ← BLOCK_C_SELECT
Select block C.
write_addr >= BLOCK_B ?
Yes
No
Block A selected
block_select ← BLOCK_A_SELECT
block_select ← BLOCK_B_SELECT
Select block B.
return

4.5 Data Write Control

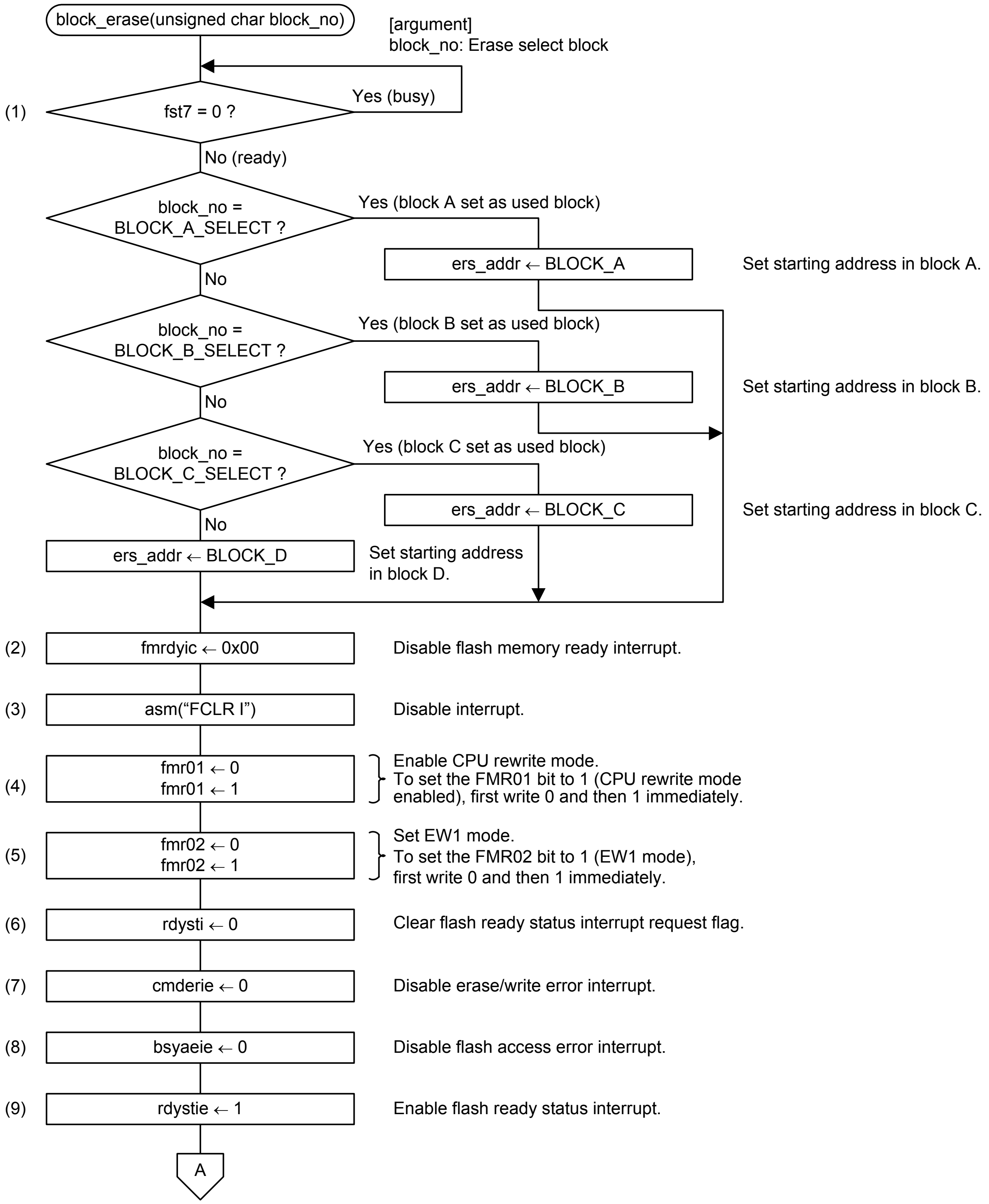
- Flowchart

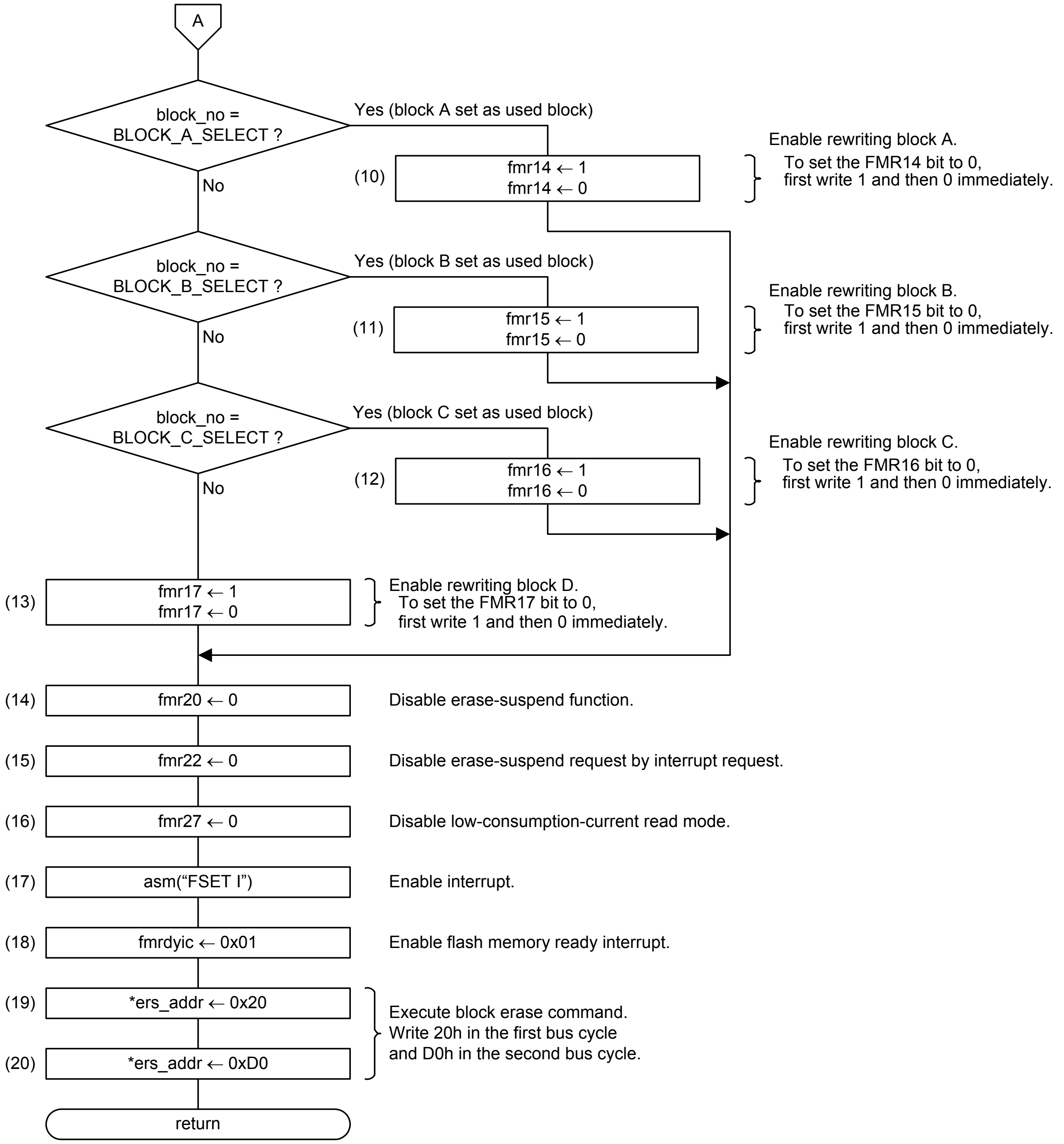



A
block_select = BLOCK_A_SELECT ?
No
Yes (block A set as used block)
write_addr > BLOCK_A_END ?
No
Yes (writing to block A completed)
Block erased
block_erase(BLOCK_B_SELECT)
Block B erased
block_select ← BLOCK_B_SELECT
Select block B.
write_addr ← BLOCK_B
Set starting address in block B.
block_select = BLOCK_B_SELECT ?
No
Yes (block B set as used block)
write_addr > BLOCK_B_END ?
No
Yes (writing to block B completed)
Block erased
block_erase(BLOCK_C_SELECT)
Block C erased
block_select ← BLOCK_C_SELECT
Select block C.
write_addr ← BLOCK_C
Set starting address in block C.
block_select = BLOCK_C_SELECT ?
No
Yes (block C set as used block)
write_addr > BLOCK_C_END ?
No
Yes (writing to block C completed)
Block erased
block_erase(BLOCK_D_SELECT)
Block D erased
block_select ← BLOCK_D_SELECT
Select block D.
write_addr ← BLOCK_D
Set starting address in block D
block_select = BLOCK_D_SELECT ?
No
Yes (block C set as used block)
write_addr > BLOCK_D_END ?
No
Yes (writing to block D completed)
Block erased
block_erase(BLOCK_A_SELECT)
Block A erased
block_select ← BLOCK_A_SELECT
Select block A.
write_addr ← BLOCK_A
Set starting address in block A.
return(result)

4.6 Block Erase

- Flowchart


block_erase(unsigned char block_no)
[argument]
block_no: Erase select block
(1)
fst7 = 0 ?
Yes (busy)
No (ready)
block_no = BLOCK_A_SELECT ?
Yes (block A set as used block)
ers_addr ← BLOCK_A
Set starting address in block A.
No
block_no = BLOCK_B_SELECT ?
Yes (block B set as used block)
ers_addr ← BLOCK_B
Set starting address in block B.
No
block_no = BLOCK_C_SELECT ?
Yes (block C set as used block)
ers_addr ← BLOCK_C
Set starting address in block C.
No
ers_addr ← BLOCK_D
Set starting address in block D.
(2)
fmrdyic ← 0x00
Disable flash memory ready interrupt.
(3)
asm(“FCLR I”)
Disable interrupt.
(4)
fmr01 ← 0
fmr01 ← 1
Enable CPU rewrite mode.
To set the FMR01 bit to 1 (CPU rewrite mode enabled), first write 0 and then 1 immediately.
(5)
fmr02 ← 0
fmr02 ← 1
Set EW1 mode.
To set the FMR02 bit to 1 (EW1 mode), first write 0 and then 1 immediately.
(6)
rdysti ← 0
Clear flash ready status interrupt request flag.
(7)
cmderie ← 0
Disable erase/write error interrupt.
(8)
bsyaeie ← 0
Disable flash access error interrupt.
(9)
rdystie ← 1
Enable flash ready status interrupt.
A


A
block_no = BLOCK_A_SELECT ?
Yes (block A set as used block)
No
(10)
fmr14 ← 1
fmr14 ← 0
Enable rewriting block A.
To set the FMR14 bit to 0, first write 1 and then 0 immediately.
block_no = BLOCK_B_SELECT ?
Yes (block B set as used block)
No
(11)
fmr15 ← 1
fmr15 ← 0
Enable rewriting block B.
To set the FMR15 bit to 0, first write 1 and then 0 immediately.
block_no = BLOCK_C_SELECT ?
Yes (block C set as used block)
No
(12)
fmr16 ← 1
fmr16 ← 0
Enable rewriting block C.
To set the FMR16 bit to 0, first write 1 and then 0 immediately.
(13)
fmr17 ← 1
fmr17 ← 0
Enable rewriting block D.
To set the FMR17 bit to 0, first write 1 and then 0 immediately.
(14)
fmr20 ← 0
Disable erase-suspend function.
(15)
fmr22 ← 0
Disable erase-suspend request by interrupt request.
(16)
fmr27 ← 0
Disable low-consumption-current read mode.
(17)
asm(“FSET I”)
Enable interrupt.
(18)
fmrdyic ← 0x01
Enable flash memory ready interrupt.
(19)
*ers_addr ← 0x20
(20)
*ers_addr ← 0xD0
Execute block erase command.
Write 20h in the first bus cycle and D0h in the second bus cycle.
return

- Register settings

(1) Wait until auto-programming or auto-erasure is completed.

Flash Memory Status Register (FST)

Bit	Symbol	Bit Name	Function	R/W
b7	FST7	Ready/busy status flag	0: Busy 1: Ready	R

(2) Disable the flash memory ready interrupt.

Interrupt Control Register (FMRDYIC)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	—	—	—	—	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0: Level 0 (interrupt disabled)	R/W
b1	ILVL1			R/W
b2	ILVL2			R/W
b3	IR	Interrupt request bit	0: No interrupt requested 1: Interrupt requested	R

(3) Clear the I flag to disable an interrupt.

(4) Enable CPU rewrite mode. When setting the FMR01 bit to 1, first write 0 and then write 1 immediately.

Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value				x	x		1	—

Bit	Symbol	Bit Name	Function	R/W
b1	FMR01	CPU rewrite mode select bit	1: CPU rewrite mode enabled	R/W

(5) Set EW1 mode. When setting the FMR02 bit to 1, first write 0 and then write 1 immediately.

Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value				x	x	1		—

Bit	Symbol	Bit Name	Function	R/W
b2	FMR02	EW1 mode select bit	1: EW1 mode	R/W

(6) Set no flash ready status interrupt request.

Flash Memory Status Register (FST)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		x			—	x	x	0

Bit	Symbol	Bit Name	Function	R/W
b0	RDYSTI	Flash ready status interrupt request flag	0: No flash ready status interrupt request	R/W

(7) Disable the erase/write error interrupt.

Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value			0	x	x			—

Bit	Symbol	Bit Name	Function	R/W
b5	CMDERIE	Erase/write error interrupt enable bit	0: Erase/write error interrupt disabled	R/W

(8) Disable the flash access error interrupt.

Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		0		x	x			—

Bit	Symbol	Bit Name	Function	R/W
b6	BSYAEIE	Flash access error interrupt enable bit	0: Flash access error interrupt disabled	R/W

(9) Enable the flash ready status interrupt.

Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	1			x	x			—

Bit	Symbol	Bit Name	Function	R/W
b7	RDYSTIE	Flash ready status interrupt enable bit	1: Flash ready status interrupt enabled	R/W

(10) Enable rewriting of data flash block A when erasing block A. When setting the FMR14 bit, first write 1 and then write 0 immediately after.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value				0	x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b4	FMR14	Data flash block A rewrite disable bit	0: Rewrite enabled (software command acceptable)	R/W

(11) Enable rewriting of data flash block B when erasing block B. When setting the FMR15 bit, first write 1 and then write 0 immediately after.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value			0		x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b5	FMR15	Data flash block B rewrite disable bit	0: Rewrite enabled (software command acceptable)	R/W

(12) Enable rewriting of data flash block C when erasing block C. When setting the FMR16 bit, first write 1 and then write 0 immediately after.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		0			x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b6	FMR16	Data flash block C rewrite disable bit	0: Rewrite enabled (software command acceptable)	R/W

(13) Enable rewriting of data flash block D when erasing block D. When setting the FMR17 bit, first write 1 and then write 0 immediately after.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	0				x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b7	FMR17	Data flash block D rewrite disable bit	0: Rewrite enabled (software command acceptable)	R/W

(14) Disable the erase-suspend function.

Flash Memory Control Register 2 (FMR2)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		—	—	—	—		x	0

Bit	Symbol	Bit Name	Function	R/W
b0	FMR20	Erase-suspend enable bit	0: Erase-suspend disabled	R/W

(15) Disable the erase-suspend request by an interrupt request.

Flash Memory Control Register 2 (FMR2)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		—	—	—	—	0	x	

Bit	Symbol	Bit Name	Function	R/W
b2	FMR22	Interrupt request suspend request enable bit	0: Erase-suspend request disabled by interrupt request	R/W

(16) Disable low-consumption-current read mode.

Flash Memory Control Register 2 (FMR2)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	0	—	—	—	—		x	

Bit	Symbol	Bit Name	Function	R/W
b7	FMR27	Low-consumption-current read mode enable bit	0: Low-consumption-current read mode disabled	R/W

(17) Set the I flag to enable an interrupt.

(18) Enable the flash memory ready interrupt.

Interrupt Control Register (FMRDYIC)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	—	—	—	—	0	0	0	1

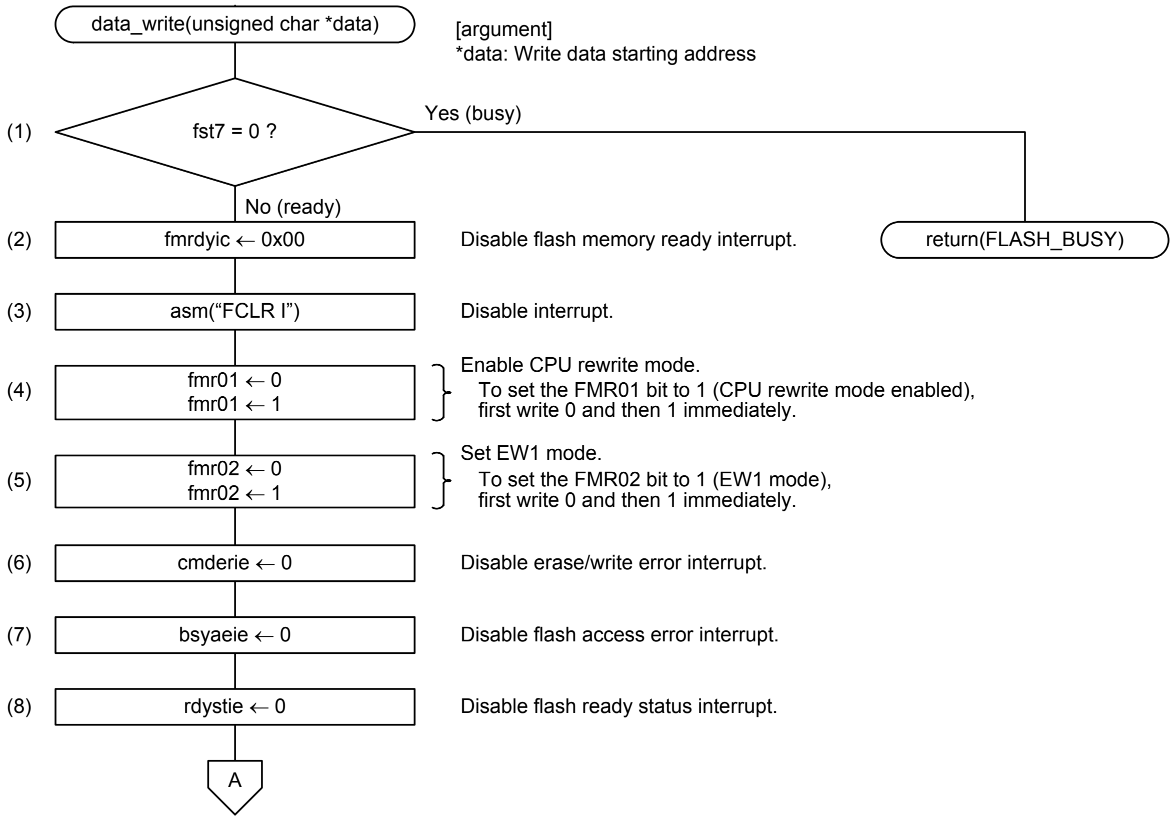
Bit	Symbol	Bit Name	Function	R/W
b0	ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 1: Level 1	R/W
b1	ILVL1			R/W
b2	ILVL2			R/W
b3	IR	Interrupt request bit	0: No interrupt requested 1: Interrupt requested	R

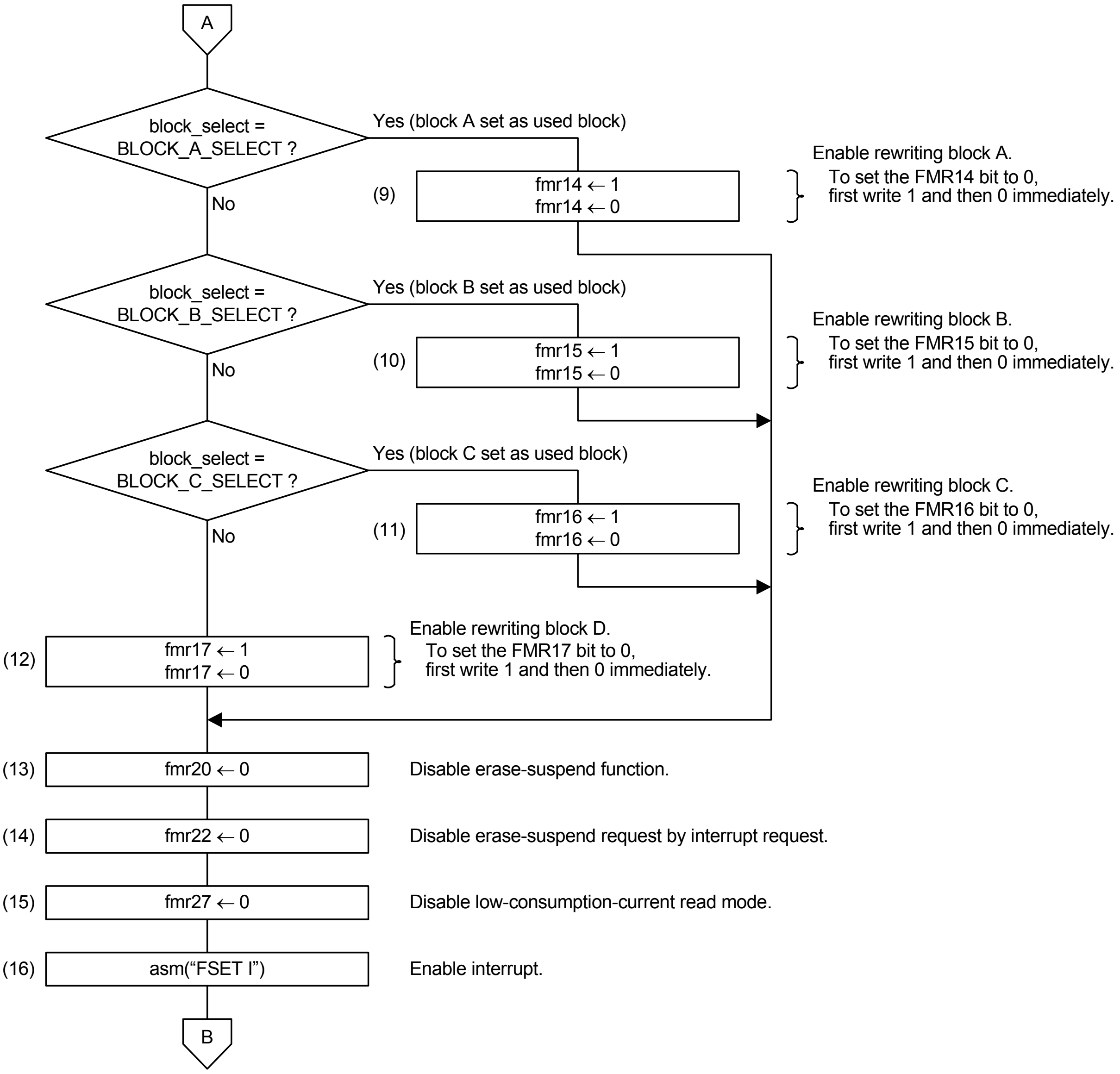
(19) Write block erase command 20h to a given address in the block to be erased in the first bus cycle.

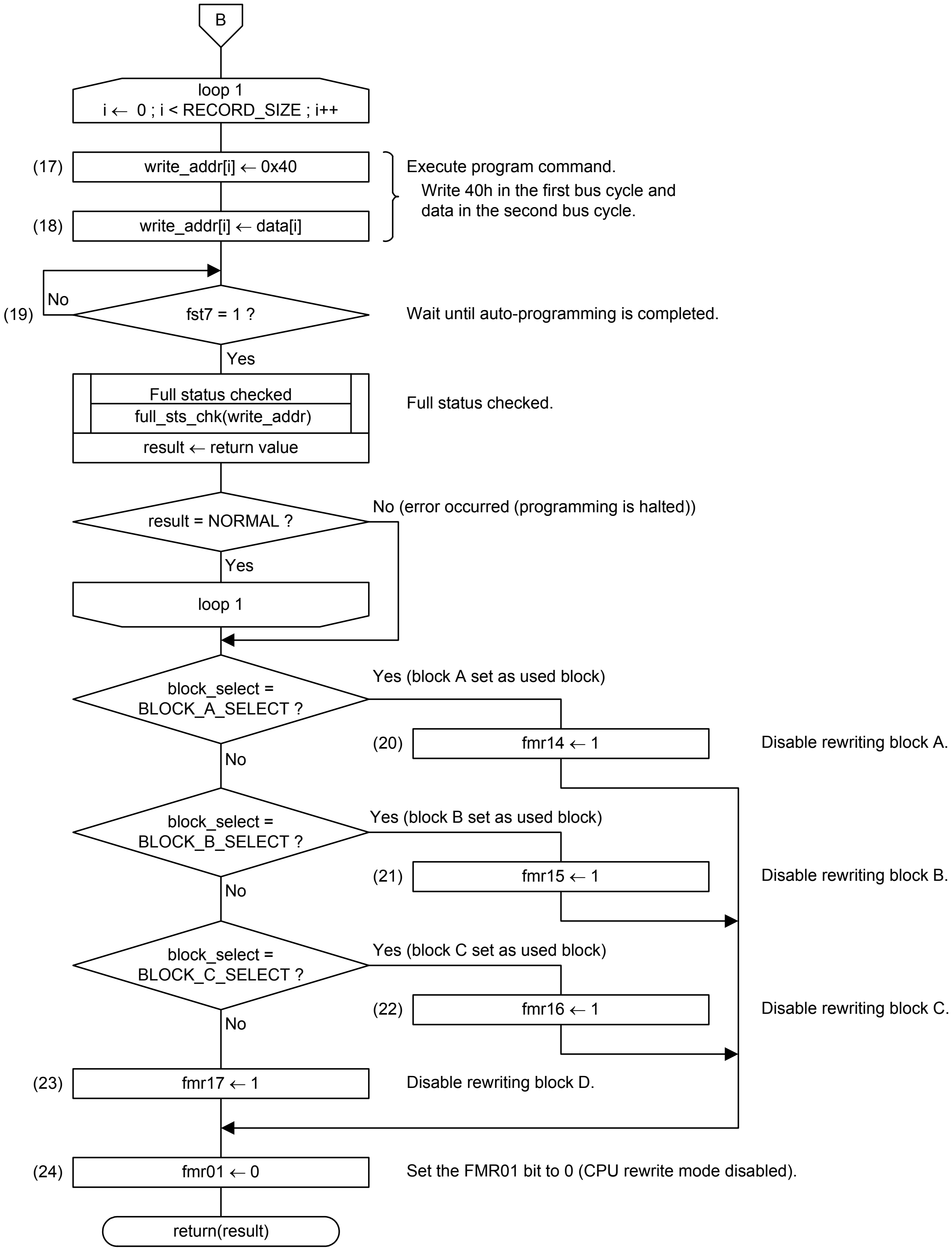
(20) Auto-erasure (erase and erase verify) starts by writing confirmation command D0h in the second bus cycle.

4.7 Data Written

- Flowchart


data_write(unsigned char *data)
[argument]
*data: Write data starting address
(1)
fst7 = 0 ?
Yes (busy)
No (ready)
return(FLASH_BUSY)
(2)
fmrdyic ← 0x00
Disable flash memory ready interrupt.
(3)
asm("FCLR I")
Disable interrupt.
(4)
fmr01 ← 0
fmr01 ← 1
Enable CPU rewrite mode.
To set the FMR01 bit to 1 (CPU rewrite mode enabled),
first write 0 and then 1 immediately.
(5)
fmr02 ← 0
fmr02 ← 1
Set EW1 mode.
To set the FMR02 bit to 1 (EW1 mode),
first write 0 and then 1 immediately.
(6)
cmderie ← 0
Disable erase/write error interrupt.
(7)
bsyaeie ← 0
Disable flash access error interrupt.
(8)
rdystie ← 0
Disable flash ready status interrupt.
A


A
block_select = BLOCK_A_SELECT ?
Yes (block A set as used block)
No
(9)
fmr14 ← 1
fmr14 ← 0
Enable rewriting block A.
To set the FMR14 bit to 0, first write 1 and then 0 immediately.
block_select = BLOCK_B_SELECT ?
Yes (block B set as used block)
No
(10)
fmr15 ← 1
fmr15 ← 0
Enable rewriting block B.
To set the FMR15 bit to 0, first write 1 and then 0 immediately.
block_select = BLOCK_C_SELECT ?
Yes (block C set as used block)
No
(11)
fmr16 ← 1
fmr16 ← 0
Enable rewriting block C.
To set the FMR16 bit to 0, first write 1 and then 0 immediately.
(12)
fmr17 ← 1
fmr17 ← 0
Enable rewriting block D.
To set the FMR17 bit to 0, first write 1 and then 0 immediately.
(13)
fmr20 ← 0
Disable erase-suspend function.
(14)
fmr22 ← 0
Disable erase-suspend request by interrupt request.
(15)
fmr27 ← 0
Disable low-consumption-current read mode.
(16)
asm(“FSET I”)
Enable interrupt.
B



Execute program command.
Write 40h in the first bus cycle and data in the second bus cycle.

Wait until auto-programming is completed.

Full status checked.

Disable rewriting block A.

Disable rewriting block B.

Disable rewriting block C.

Disable rewriting block D.

Set the FMR01 bit to 0 (CPU rewrite mode disabled).

- Register settings

(1) Verify that auto-programming or auto-erasure is completed.

Flash Memory Status Register (FST)

Bit	Symbol	Bit Name	Function	R/W
b7	FST7	Ready/busy status flag	0: Busy 1: Ready	R

(2) Disable the flash memory ready interrupt.

Interrupt Control Register (FMRDYIC)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	—	—	—	—	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0: Level 0 (interrupt disabled)	R/W
b1	ILVL1			R/W
b2	ILVL2			R/W
b3	IR	Interrupt request bit	0: No interrupt requested 1: Interrupt requested	R

(3) Clear the I flag to disable an interrupt.

(4) Enable CPU rewrite mode. When setting the FMR01 bit to 1, first write 0 and then write 1 immediately.

Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value				x	x		1	—

Bit	Symbol	Bit Name	Function	R/W
b1	FMR01	CPU rewrite mode select bit	1: CPU rewrite mode enabled	R/W

(5) Set EW1 mode. When setting the FMR02 bit to 1, first write 0 and then write 1 immediately.

Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value				x	x	1		—

Bit	Symbol	Bit Name	Function	R/W
b2	FMR02	EW1 mode select bit	1: EW1 mode	R/W

(6) Disable the erase/write error interrupt.

Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value			0	x	x			—

Bit	Symbol	Bit Name	Function	R/W
b5	CMDERIE	Erase/write error interrupt enable bit	0: Erase/write error interrupt disabled	R/W

(7) Disable the flash access error interrupt.

Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		0		x	x			—

Bit	Symbol	Bit Name	Function	R/W
b6	BSYAEIE	Flash access error interrupt enable bit	0: Flash access error interrupt disabled	R/W

(8) Disable the flash ready status interrupt.

Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	0			x	x			—

Bit	Symbol	Bit Name	Function	R/W
b7	RDYSTIE	Flash ready status interrupt enable bit	0: Flash ready status interrupt disabled	R/W

(9) Enable rewriting of data flash block A when rewriting block A. When setting the FMR14 bit, first write 1 and then write 0 immediately after.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value				0	x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b4	FMR14	Data flash block A rewrite disable bit	0: Rewrite enabled (software command acceptable)	R/W

(10) Enable rewriting of data flash block B when rewriting block B. When setting the FMR15 bit, first write 1 and then write 0 immediately after.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value			0		x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b5	FMR15	Data flash block B rewrite disable bit	0: Rewrite enabled (software command acceptable)	R/W

(11) Enable rewriting of data flash block C when rewriting block C. When setting the FMR16 bit, first write 1 and then write 0 immediately after.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		0			x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b6	FMR16	Data flash block C rewrite disable bit	0: Rewrite enabled (software command acceptable)	R/W

(12) Enable rewriting of data flash block D when rewriting block D. When setting the FMR17 bit, first write 1 and then write 0 immediately after.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	0				x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b7	FMR17	Data flash block D rewrite disable bit	0: Rewrite enabled (software command acceptable)	R/W

(13) Disable the erase-suspend function.

Flash Memory Control Register 2 (FMR2)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		—	—	—	—		x	0

Bit	Symbol	Bit Name	Function	R/W
b0	FMR20	Erase-suspend enable bit	0: Erase-suspend disabled	R/W

(14) Disable the erase-suspend request by an interrupt request.

Flash Memory Control Register 2 (FMR2)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		—	—	—	—	0	x	

Bit	Symbol	Bit Name	Function	R/W
b2	FMR22	Interrupt request suspend request enable bit	0: Erase-suspend request disabled by interrupt request	R/W

(15) Disable the low-consumption-current read mode.

Flash Memory Control Register 2 (FMR2)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	0	—	—	—	—		x	

Bit	Symbol	Bit Name	Function	R/W
b7	FMR27	Low-consumption-current read mode enable bit	0: Low-consumption-current read mode disabled	R/W

(16) Set the I flag to enable an interrupt.

(17) Write program command 40h in the first bus cycle to the write address.

(18) Auto-programming (data programmed and verified) starts by writing data in the second bus cycle. Set the same address value in the second bus cycle as the address value specified in the first bus cycle.

(19) Wait until auto-programming is completed.

Flash Memory Status Register (FST)

Bit	Symbol	Bit Name	Function	R/W
b7	FST7	Ready/busy status flag	0: Busy 1: Ready	R

(20) Disable rewriting of data flash block A when rewriting block A is completed.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value				1	x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b4	FMR14	Data flash block A rewrite disable bit	1: Rewrite disabled (software command not acceptable, no error occurred)	R/W

(21) Disable rewriting of data flash block B when rewriting block B is completed.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value			1		x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b5	FMR15	Data flash block B rewrite disable bit	1: Rewrite disabled (software command not acceptable, no error occurred)	R/W

(22) Disable rewriting of data flash block C when rewriting block C is completed.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		1			x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b6	FMR16	Data flash block C rewrite disable bit	1: Rewrite disabled (software command not acceptable, no error occurred)	R/W

(23) Disable rewriting of data flash block D when rewriting block D is completed.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	1				x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b7	FMR17	Data flash block D rewrite disable bit	1: Rewrite disabled (software command not acceptable, no error occurred)	R/W

(24) Disable CPU rewrite mode.

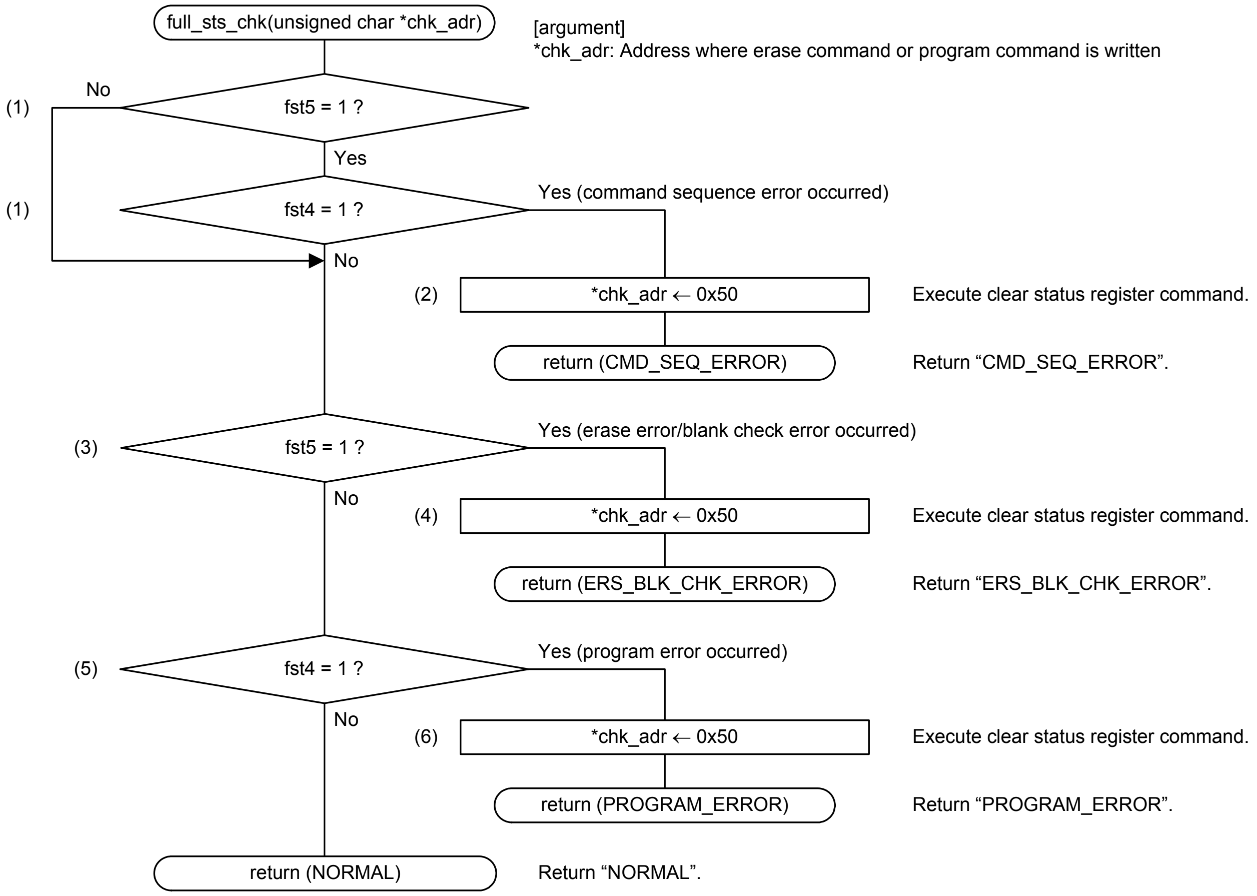
Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value				x	x		0	—

Bit	Symbol	Bit Name	Function	R/W
b1	FMR01	CPU rewrite mode select bit	0: CPU rewrite mode disabled	R/W

4.8 Full Status Check

- Flowchart


full_sts_chk(unsigned char *chk_adr)
[argument]
*chk_adr: Address where erase command or program command is written
(1)
No
fst5 = 1 ?
Yes
(1)
fst4 = 1 ?
Yes (command sequence error occurred)
No
(2)
*chk_adr ← 0x50
Execute clear status register command.
return (CMD_SEQ_ERROR)
Return "CMD_SEQ_ERROR".
(3)
fst5 = 1 ?
Yes (erase error/blank check error occurred)
No
(4)
*chk_adr ← 0x50
Execute clear status register command.
return (ERS_BLK_CHK_ERROR)
Return "ERS_BLK_CHK_ERROR".
(5)
fst4 = 1 ?
Yes (program error occurred)
No
(6)
*chk_adr ← 0x50
Execute clear status register command.
return (PROGRAM_ERROR)
Return "PROGRAM_ERROR".
return (NORMAL)
Return "NORMAL".

- Register settings

(1) Verify that a command sequence error occurs by reading bits FST4 and FST5 in the FST register.

Flash Memory Status Register (FST)

Bit	Symbol	Bit Name	Function	R/W
b4	FST4	Program error flag	0: No program error 1: Program error	R
b5	FST5	Erase error/blank check error flag	0: No erase error/blank check error 1: Erase error/blank check error	R

(2) Write clear status register command 50h to the address where erase command 20h or program command 40h was written when a program error (FST4 = 1) and an erase error (FST5 = 1) occur.

(3) Confirm if an erase error/blank check error occurs by reading the FST5 bit.

Flash Memory Status Register (FST)

Bit	Symbol	Bit Name	Function	R/W
b5	FST5	Erase error/blank check error flag	0: No erase error/blank check error 1: Erase error/blank check error	R

(4) Write clear status register command 50h to the address where erase command 20h was written when an erase error (FST5 = 1) occurs.

(5) Verify that a program error occurs by reading the FST4 bit.

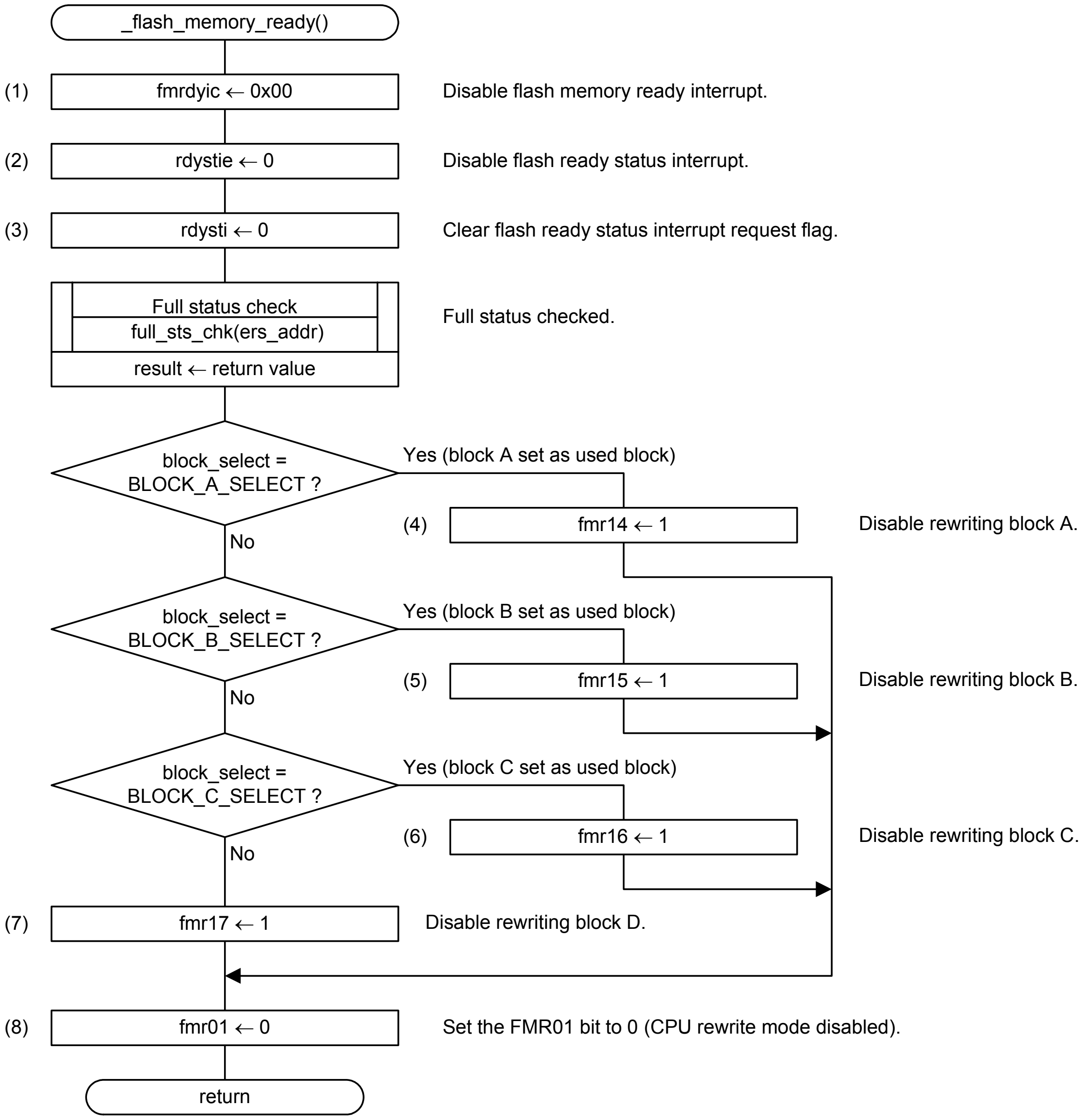
Flash Memory Status Register (FST)

Bit	Symbol	Bit Name	Function	R/W
b4	FST4	Program error flag	0: No program error 1: Program error	R

(6) Write clear status register command 50h to the address where program command 40h was written when a program error (FST4 = 1) occurs.

4.9 Flash Memory Ready Interrupt

- Flowchart


_flash_memory_ready()
(1)
fmrdyic ← 0x00
Disable flash memory ready interrupt.
(2)
rdystie ← 0
Disable flash ready status interrupt.
(3)
rdysti ← 0
Clear flash ready status interrupt request flag.
Full status check
full_sts_chk(ers_addr)
result ← return value
Full status checked.
block_select = BLOCK_A_SELECT ?
Yes (block A set as used block)
(4)
fmr14 ← 1
Disable rewriting block A.
No
block_select = BLOCK_B_SELECT ?
Yes (block B set as used block)
(5)
fmr15 ← 1
Disable rewriting block B.
No
block_select = BLOCK_C_SELECT ?
Yes (block C set as used block)
(6)
fmr16 ← 1
Disable rewriting block C.
No
(7)
fmr17 ← 1
Disable rewriting block D.
(8)
fmr01 ← 0
Set the FMR01 bit to 0 (CPU rewrite mode disabled).
return

- Register settings

(1) Disable the flash memory ready interrupt.

Interrupt Control Register (FMRDYIC)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	—	—	—	—	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0: Level 0 (interrupt disabled)	R/W
b1	ILVL1			R/W
b2	ILVL2			R/W
b3	IR	Interrupt request bit	0: No interrupt requested 1: Interrupt requested	R

(2) Disable the flash ready status interrupt.

Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	0			x	x			—

Bit	Symbol	Bit Name	Function	R/W
b7	RDYSTIE	Flash ready status interrupt enable bit	0: Flash ready status interrupt disabled	R/W

(3) Set no flash ready status interrupt request.

Flash Memory Status Register (FST)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		x			—	x	x	0

Bit	Symbol	Bit Name	Function	R/W
b0	RDYSTI	Flash ready status interrupt request flag	0: No flash ready status interrupt request	R/W

(4) Disable rewriting of data flash block A when erasing block A is completed.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value				1	x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b4	FMR14	Data flash block A rewrite disable bit	1: Rewrite disabled (software command not acceptable, no error occurred)	R/W

(5) Disable rewriting of data flash block B when erasing block B is completed.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value			1		x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b5	FMR15	Data flash block B rewrite disable bit	1: Rewrite disabled (software command not acceptable, no error occurred)	R/W

(6) Disable rewriting of data flash block C when erasing block C is completed.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value		1			x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b6	FMR16	Data flash block C rewrite disable bit	1: Rewrite disabled (software command not acceptable, no error occurred)	R/W

(7) Disable rewriting of data flash block D when erasing block D is completed.

Flash Memory Control Register 1 (FMR1)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	1				x	—	—	—

Bit	Symbol	Bit Name	Function	R/W
b7	FMR17	Data flash block D rewrite disable bit	1: Rewrite disabled (software command not acceptable, no error occurred)	R/W

(8) Disable CPU rewrite mode.

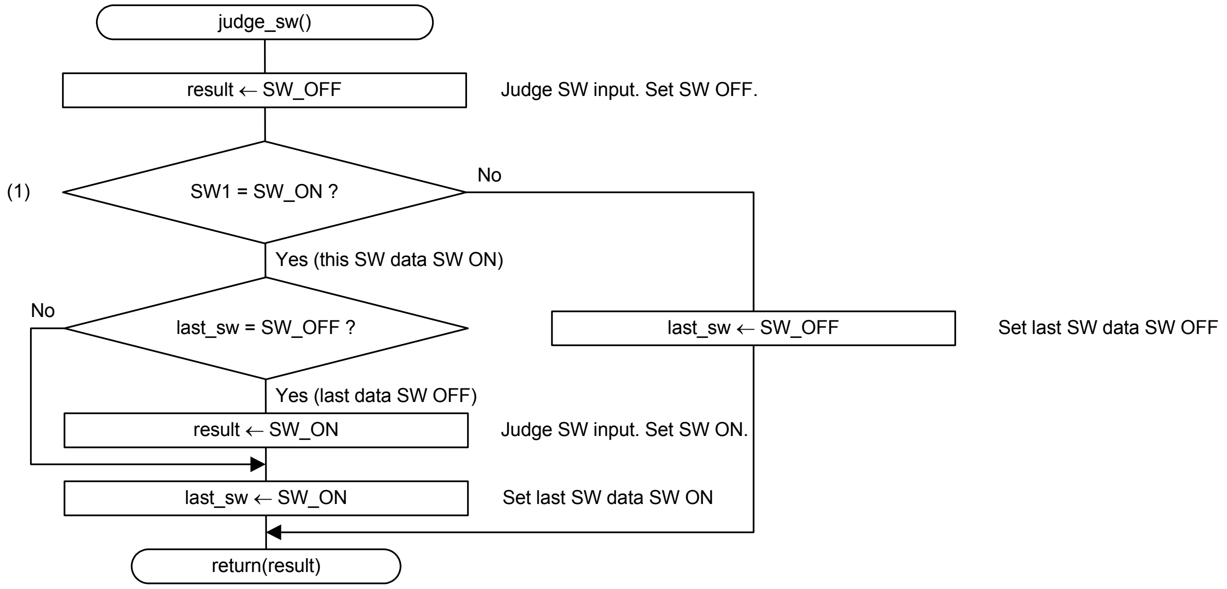
Flash Memory Control Register 0 (FMR0)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value				x	x		0	—

Bit	Symbol	Bit Name	Function	R/W
b1	FMR01	CPU rewrite mode select bit	0: CPU rewrite mode disabled	R/W

4.10 SW Input Judgment

- Flowchart



- Register settings

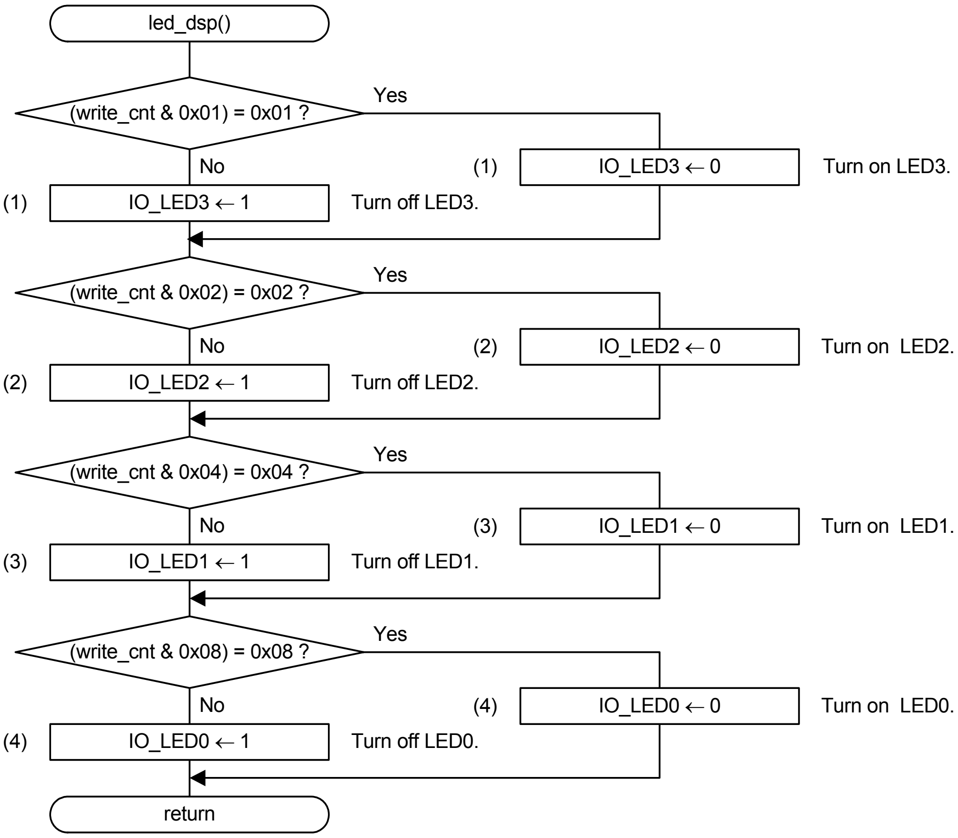
(1) Verify that P1_7 is low.

Port P1 Register (P1)

Bit	Symbol	Bit Name	Function	R/W
b7	P1_7	Port P1_7 bit	0: “L” level 1: “H” level	R/W

4.11 LED Display

- Flowchart



- Register settings

(1) Set P3_6 (LED3 output) to high or low depending on the bit 0 value of the number of writes (write_cnt).

Port P3 Register (P3)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	x	0/1	x			x		x

Bit	Symbol	Bit Name	Function	R/W
b6	P3_6	Port P3_6 bit	0: “L” level 1: “H” level	R/W

(2) Set P3_4 (LED2 output) to high or low depending on the bit 1 value of the number of writes (write_cnt).

Port P3 Register (P3)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	x		x	0/1		x		x

Bit	Symbol	Bit Name	Function	R/W
b4	P3_4	Port P3_4 bit	0: “L” level 1: “H” level	R/W

(3) Set P3_3 (LED1 output) to high or low depending on the bit 2 value of the number of writes (write_cnt).

Port P3 Register (P3)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	x		x		0/1	x		x

Bit	Symbol	Bit Name	Function	R/W
b3	P3_3	Port P3_3 bit	0: “L” level 1: “H” level	R/W

(4) Set P3_1 (LED0 output) to high or low depending on the bit 3 value of the number of writes (write_cnt).

Port P3 Register (P3)

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Setting Value	x		x			x	0/1	x

Bit	Symbol	Bit Name	Function	R/W
b1	P3_1	Port P3_1 bit	0: “L” level 1: “H” level	R/W

5. Sample Programming Code

A sample program can be downloaded from the Renesas Technology website.
To download, click "Application Notes" in the left-hand side menu of the R8C/Tiny Series page.

6. Reference Documents

Hardware Manual
R8C/35C Group Hardware Manual Rev. 0.10
The latest version can be downloaded from the Renesas Technology website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Technology website.

Website and Support

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

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

REVISION HISTORY	R8C/35C Group Rewriting the Data Flash (Flash Memory Ready Interrupt)

Rev.	Date	Description	
		Page	Summary
1.00	Dec 29, 2009	–	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.