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.

RENESAS

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.

User's Manual

μPD78F0711, 78F0712, 78F0714

8-Bit Single-Chip Microcontroller

Flash Memory Self Programming

μPD78F0711
μPD78F0712
μPD78F0714

Document No. U18886EJ1V0UM00 (1st edition)
Date Published September 2007 NS

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

③ PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ STATUS BEFORE INITIALIZATION

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

⑤ POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

EEPROM is a trademark of NEC Electronics Corporation.

INTRODUCTION

Readers

This User’s Manual is intended for users who wish to understand the self-programming function of the *μ*PD78F0711, 78F0712, and 78F0714 and design application systems using these microcontrollers.

Purpose

This User’s Manual is intended to give users an understanding of the creation of application programs that utilize the *μ*PD78F0711, 78F0712, and 78F0714’s on-chip flash memory.

Organization

This manual can be generally divided into the following sections.

- Description of flash environment
- Description of flash memory control firmware

How to Read This Manual

It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, microcontrollers, assembler, and C language.

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**.
- To check the hardware functions of the *μ*PD78F0711, 78F0712, and 78F0714
  → Refer to the ***μ*PD78F0711, 78F0712 User’s Manual (U17890E)** or ***μ*PD78F0714 User’s Manual (U16928E)**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	$\overline{\text{xxx}}$ (overscore over pin or signal name)
Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numerical representation:	Binary … xxxx or xxxxB
	Decimal … xxxx
	Hexadecimal … xxxxH

Terminology

The following describes the meanings of certain terms used in this manual.

- Self programming

  Self programming operations are flash memory write operations that are performed by user programs.
- Flash memory control firmware

  This firmware provides an interface for flash memory manipulations performed by the μPD78F0711, 78F0712, and 78F0714 using its internal software. In this manual, the term “firmware” refers to this flash memory control firmware.
- Flash environment

  This is the environment that supports flash memory manipulations. It has restrictions that differ from those applied to ordinary program execution.
- Block number

  Block numbers indicate blocks in flash memory. They are used as units during manipulations such as erasures and blank checks.
- Boot cluster (μPD78F0714 only)

  This is the boot area used for boot swaps. Two boot clusters, boot cluster 0 and boot cluster 1, are provided, so that the cluster to be booted can be selected.
- Entry RAM

  This is the area in RAM that is used by flash functions. The user program reserves this area and specifies the start address of the specific area to be used when flash functions are called.
- Internal verification

  After writing to flash memory, signal levels are checked internally to confirm correct reading of data. When an internal verification error occurs, the corresponding device is judged as faulty.

CONTENTS

CHAPTER 1 OVERVIEW OF FLASH MEMORY SELF PROGRAMMING 8

1.1 Functions of Flash Memory Control Firmware 8

1.2 Control of Flash Memory Control Firmware 9

1.3 Self Programming Modes 10

1.4 Hardware Environment 11

1.5 Software Environment 12

CHAPTER 2 SELF PROGRAMMING FUNCTIONS 13

2.1 Registers That Control Self Programming 13

2.1.1 Flash programming mode control register (FLPMC) 13

2.1.2 Flash protect command register (PFCMD) 15

2.1.3 Flash status register (PFS) 16

2.2 FLMD0 Pin Manipulations 17

2.3 Self Programming Mode and FLMD0 Pin Control 18

2.4 Functions of Flash Memory Control Firmware 19

2.5 Flash Memory Control Firmware Use Environment 20

2.6 Parameters for Controlling Flash Memory Control Firmware 21

2.6.1 Register bank 3 22

2.6.2 Entry RAM 23

2.6.3 Data buffer 24

2.6.4 Flash memory block numbers 25

2.6.5 RAM memory map 26

CHAPTER 3 ACCESS TO FLASH MEMORY 27

3.1 Overall Flow 27

3.2 Initial Settings 29

3.3 Initialization 29

3.4 Mode Check 32

3.5 Block Blank Check 34

3.6 Block Erasure 36

3.7 Word Write 38

3.8 Block Verify 41

3.9 Get Information 43

3.10 Set Information 47

3.11 EEPROM Write 49

3.12 EEPROM Erase 52

3.13 Status List 54

3.14 Boot Swap Function (μPD78F0714 only) 55

CHAPTER 4 CC78K0 SELF-WRITE EXPANSION FUNCTION 57

CHAPTER 1 OVERVIEW OF FLASH MEMORY SELF PROGRAMMING

The μPD78F0711, 78F0712, and 78F0714 supports flash memory control firmware that is used to rewrite flash memory. This firmware control enables flash memory to be rewritten from application programs.

Possible uses of self-programming include the following.

- Rewriting of programs by application programs (for program upgrading in the field, etc.)
- Use for EEPROM™ emulation (rewriting of constant data by applications, etc.)

1.1 Functions of Flash Memory Control Firmware

The μPD78F0711, 78F0712, and 78F0714's on-chip flash memory control firmware[Note] provides the functions to erase or write flash memory.

Note Allocated in the memory area where users cannot access.

Table 1-1. Firmware Function (Command) List

Function (Command)	Description	
Initialize	Initialization	Performs self programming setup.
Block erase	Block erasure	Erases specified block (2 KB) of data.
Word write	Word writing	Writes data in RAM to flash memory. Up to 256 bytes (specified in 4-byte units) can be written at one time.
EEPROM write	EEPROM emulation data writing	During EEPROM emulation, this function writes data in RAM to flash memory. Up to 256 bytes (specified in 4-byte units) can be written at one time.
Block verify	Verification	Verifies the specified block (2 KB).
Block blank check	Blank check	Performs a blank check of the specified block (2 KB).
Mode check	FLMD0 voltage check	Checks voltage level of FLMD0 pin.
Get information	Information acquisition	Reads information related to flash memory settings.
Set information	Information setting	Sets for security and boot swap[Note].
EEPROM erase	Data erasure for EEPROM emulation	During EEPROM emulation, the memory in a specified block is deleted only by the duration of given time (10 ms units).

Note The boot swap function is supported only with the μPD78F0714.

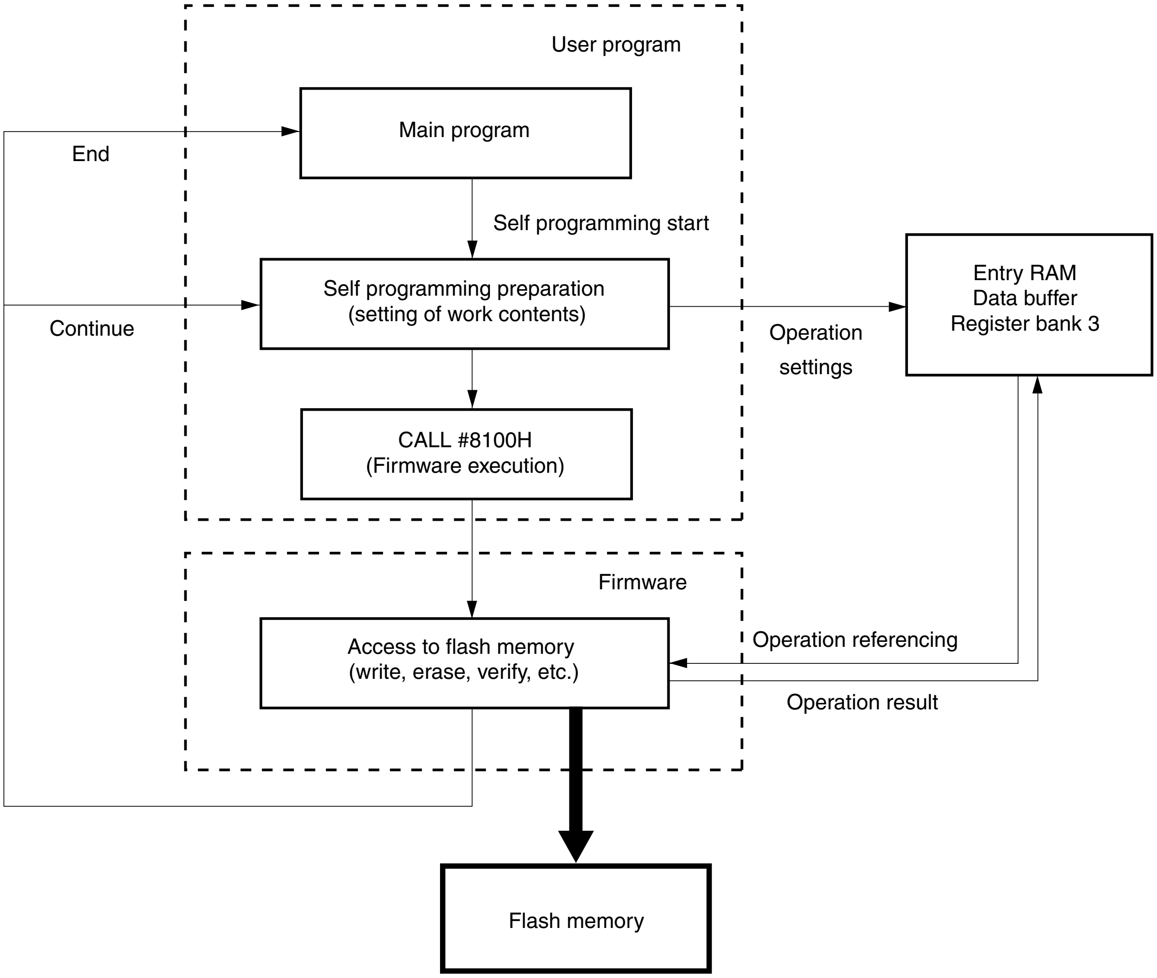
1.2 Control of Flash Memory Control Firmware

The user performs flash memory write or erase via the flash memory control firmware. The self-programming operation contents are indicated to the firmware from the user's application program, and the firmware performs flash memory write or erase. The operation contents are set in the memory area (entry RAM, data buffer, register bank 3).

The application program is in the wait status during firmware operation.

Figure 1-1 shows an outline of self programming.

Figure 1-1. Self Programming Outline



1.3 Self Programming Modes

The self programming mode is determined by the FLSPM values (values of bits 1 and 0 of FLPMC register).

<1> Normal mode
This is the mode for executing user applications.
After reset release, the operation starts in this normal mode.

<2> Self programming mode
This is the mode used to perform self programming preparations and settings.
In this mode, the flash memory control firmware can be executed (CALL !8100H).

Cautions 1. Before firmware execution, be sure to set the self programming mode.
2. When all the self programming work is completed, be sure to set the normal mode.
3. In the self programming mode, addresses 8000H and higher are allocated to firmware. Place the program for controlling flash memory in addresses 0000H to 7FFFH.

Table 1-2. Self Programming Modes

Mode	FLSPMC Register		Firmware Execution (CALL! 8100H)	User Program Execution
	FLSPM1	FLSPM0		
<1> Normal mode	0	0	–	√
<2> Self programming mode	0	1	√	√ Note

Note Only the address range of 0000H to 7FFFH can be accessed (instruction fetch, data read).

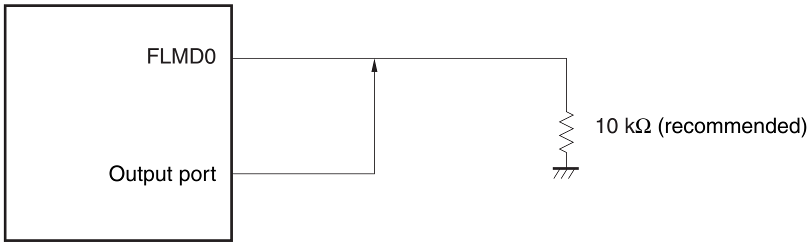
Caution Setting FLSPM1, FLSPM0 = 1, 0 or 1, 1 is prohibited.

Remark √: Enabled, –: Disabled

1.4 Hardware Environment

The voltage of the FLMD0 pin must be set to low level during normal operation and to high level during self programming. Figure 1-2 shows an example of a circuit that switches the voltage of the FLMD0 pin through port manipulation.

Figure 1-2. Example of FLMD0 Voltage Generator



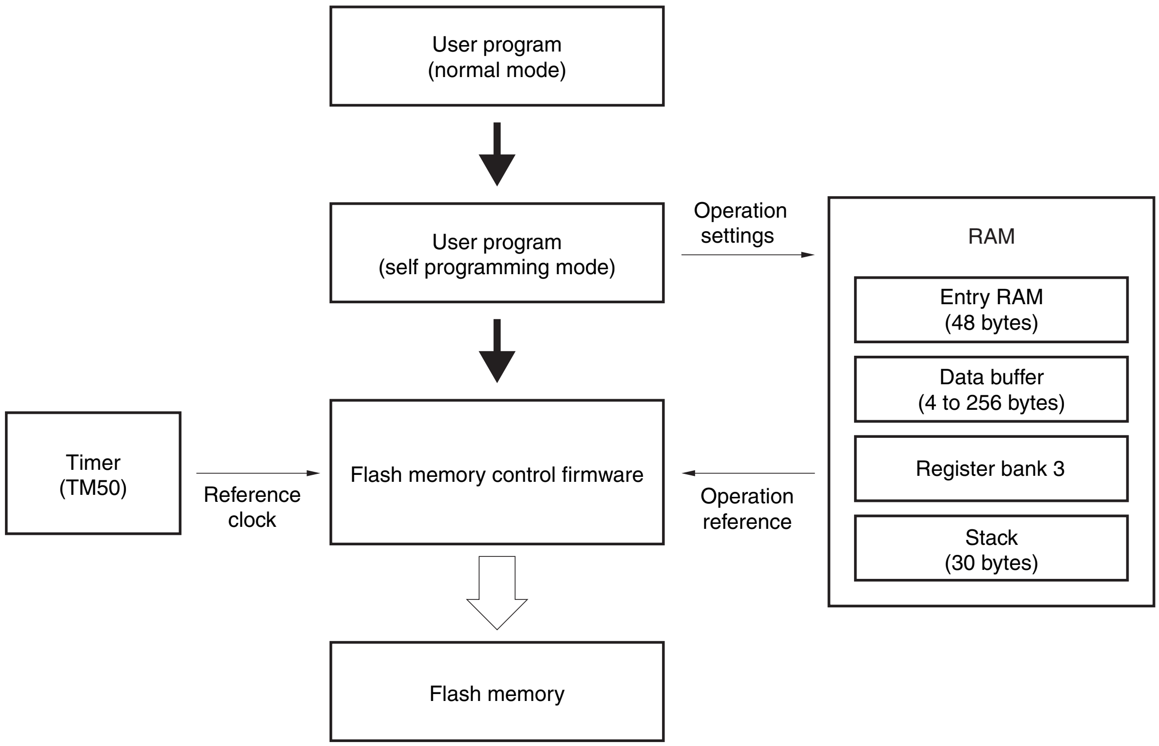
1.5 Software Environment

When performing self programming, the resources shown in Table 1-3 and Figure 1-3 below are required. For details, refer to **2.6 Parameters for Controlling Flash Memory Control Firmware**.

Table 1-3. Software Resources

Item	Description
Register bank	Register bank 3
Timer	8-bit timer (TM50)
Data buffer	4 to 256 bytes (can be set in 4-byte units)
Entry RAM	48 bytes
Stack	Up to 30 bytes (uses same stack area as user program)

Figure 1-3. Software Environment



2.1 Registers That Control Self Programming

2.1.1 Flash programming mode control register (FLPMC)

This register is used to enable/disable flash memory access (write, erase, etc.), and indicate the self programming operation mode.

A particular sequence must be used when writing to this register, in order to prevent inadvertent settings due to noise or manipulation errors. For the specific sequence, refer to **2.1.2 Flash protect command register (PFCMD)**.

After reset: 08H R/W[Note]

Symbol	7	6	5	4	3	2	1	0
FLPMC	0	0	0	0	FWEDIS	FWEPR	FLSPM1	FLSPM0

Note Bit 2 is a read-only bit.

[FWEDIS]

This flag is used to control flash memory access (write, erase, etc.) enable/disable through software. The initial value of this flag is 1, and flash memory access is enabled by writing 0 to this flag.

FWEDIS	Function
0	Enable write/erase
1	Disable write/erase

[FWEPR]

This flag is used to control flash memory access (write, erase, etc.) enable/disable through hardware. It directly reflects the voltage of the FLMD0 pin.

FLMD0 Pin Voltage	FWEPR[Note]	Function
Low level (V_{SS})	0	Disable write/erase
High level (V_{DD})	1	Enable write/erase

Note The FWEPR bit is a read-only bit. Its value cannot be changed by software. However, when using an in-circuit emulator, the value can be changed even by overwriting.

Flash memory access can be enabled through the combination of FWEDIS and FWEPR.

FWEDIS	FWEPR	Flash Memory Write/Erase Enable
0	1	Enable write/erase
Other than above		Disable write/erase

Cautions 1. When executing flash memory control firmware, such as flash memory write/erase, be sure to set FWEDIS to 0.

2. In the normal mode, be sure to set FWEDIS to 1.

[FLSPM0 and FLSPM1]

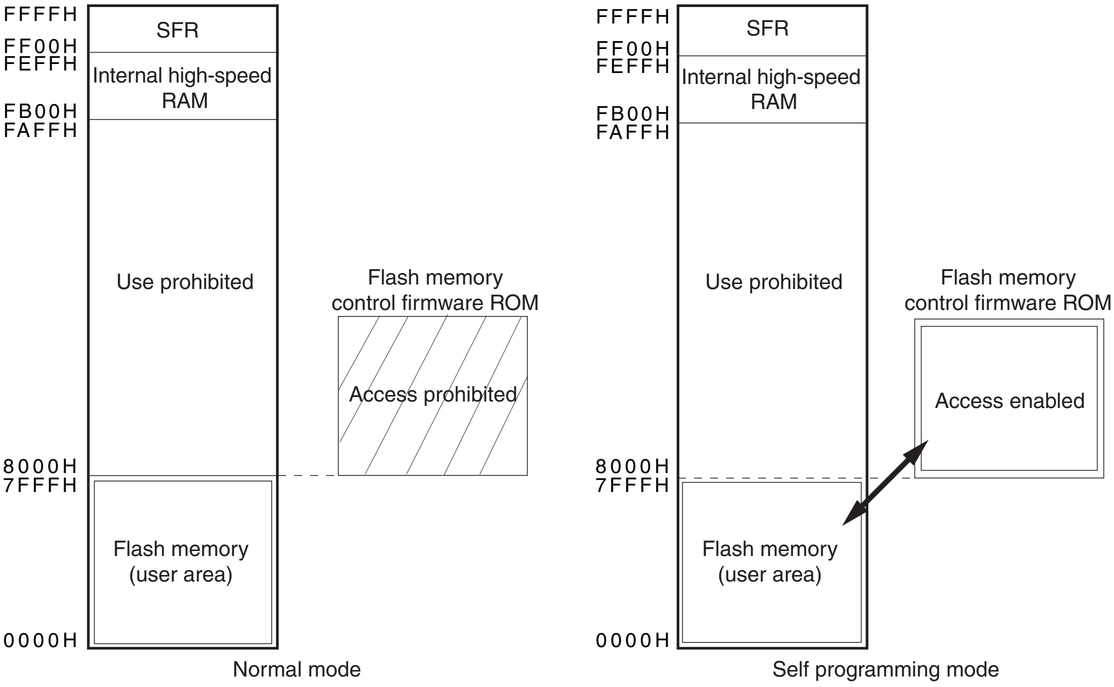
These control flags are used to select the self programming operation mode.

FLSPM1	FLSPM0	Mode Selection
0	0	Normal mode • Access (instruction fetch, data read) to the entire address range of flash memory is possible.
0	1	Self programming mode • Firmware execution "CALL #8100H" is possible. • Access (instruction fetch, data read) to flash memory is possible.

Caution Setting FLSPM1, FLSPM0 = 1, 0 or 1, 1 is prohibited.

Figure 2-1 shows the self programming operation mode and memory map.

Figure 2-1. Self Programming Operation Mode and Memory Map (μPD78F0714)



Caution Place the program that controls the flash memory control firmware in the address range of 0000H to 7FFFH.

2.1.2 Flash protect command register (PFCMD)

To prevent erroneous flash memory write or erase caused by an inadvertent program loop, etc., protection is implemented by this register for flash programming mode control register (FLPMC) write.

The FLPMC register is a special register that is valid for write operations only when the write operations are performed via following special sequence.

<1> Write a specified value (= A5H) to the PFCMD register.
<2> Write the value to be set to the FLPMC register (writing is invalid at this step).
<3> Write the inverted value of the value to be set to the FLPMC register (writing is invalid at this step).
<4> Write the value to be set to the FLPMC register (writing is valid at this step).

Caution The above sequence must be executed every time the value of the FLPMC register is changed.

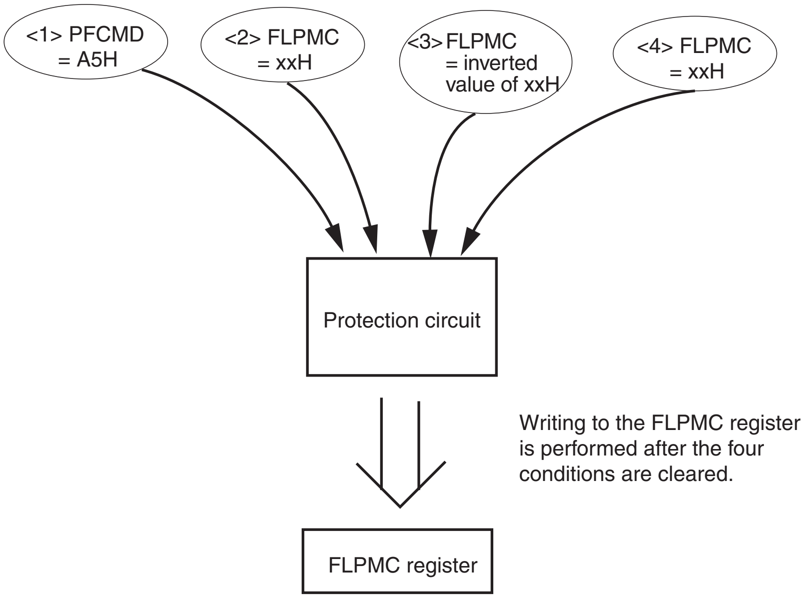
After reset: Undefined W

Symbol	7	6	5	4	3	2	1	0
PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

<Coding example of special sequence>
When writing 05H to FLPMC register:

```
MOV PFCMD, #0A5H    ; Writes A5H to PFCMD
MOV FLPMC, #05H     ; Writes 05H to FLPMC
MOV FLPMC, #0FAH    ; Writes 0FAH (inverted value of 05H) to FLPMC
MOV FLPMC, #05H     ; Writes 05H to FLPMC
```

Figure 2-2. Write Protection


<1> PFCMD = A5H
<2> FLPMC = xxH
<3> FLPMC = inverted value of xxH
<4> FLPMC = xxH
Protection circuit
Writing to the FLPMC register is performed after the four conditions are cleared.
FLPMC register

2.1.3 Flash status register (PFS)

If the flash programming mode control register (FLPMC) is not written in the correct sequence, the FLPMC register is not set and a protection error occurs. At this time, bit 0 (FPRERR) of the PFS register is set to 1.

This flag is a cumulative flag.

After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFS	0	0	0	0	0	0	0	FPRERR

The FPRERR flag's operation conditions are as follows.

<Setting conditions>

- When the PFCMD register is written to at a time when the store instruction's operation for the latest peripheral register was not a write operation to the PFCMD register using a specified value (A5H)
- When the first store instruction operation after <1> above is for a peripheral register other than the FLPMC register
- When the first store instruction operation after <2> above is for a peripheral register other than the FLPMC register
- When the first store instruction operation after <2> above writes a value other than the inverted value of the value to be set to the FLPMC register
- When the first store instruction operation after <3> above is for a peripheral register other than the FLPMC register
- When the first store instruction operation after <3> above writes a value other than the value (write value in <2>) to be set to the FLPMC register.

Remark The numbers shown in angle brackets above correspond to the numbers shown in angle brackets in section 2.1.2 above.

<Reset conditions>

- When 0 is written to bit 0 (FPRERR) in the PFS register.
- When a system reset is performed.

2.2 FLMD0 Pin Manipulations

For self programming, the level of the FLMD0 pin must be changed.
The firmware execution requires that FWEDIS = 0, in addition to high level input to the FLMD0 pin.

Table 2-1. Levels of FLMD0 Pin Based on Flash Programming Mode

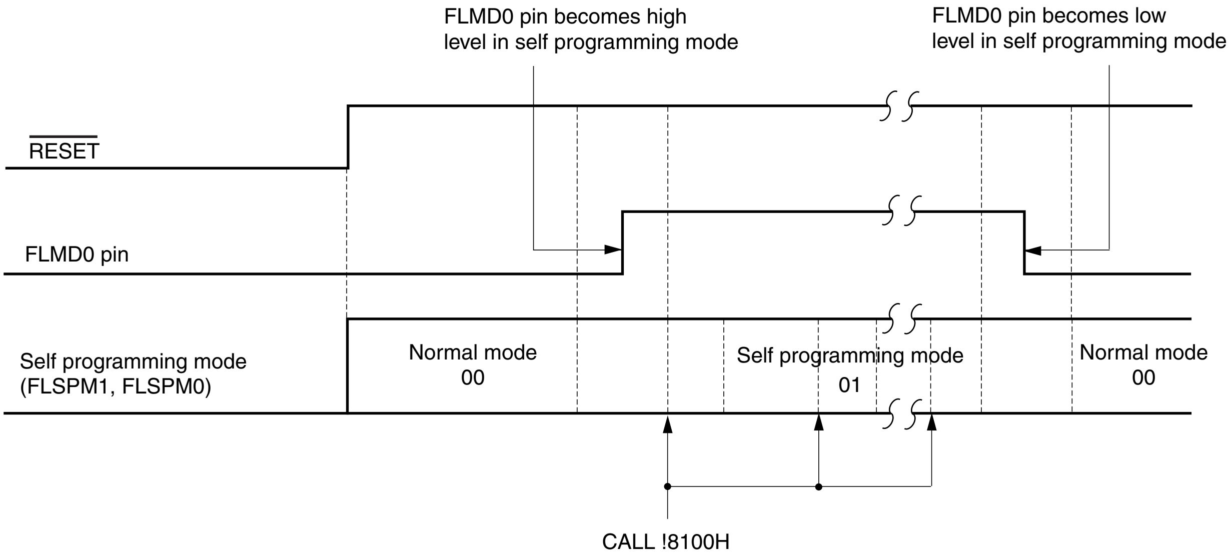
Pin Name	Normal Mode (FLSPM1, FLSPM0 = 0, 0)	Self-Programming Mode (FLSPM1, FLSPM0 = 0, 1)
FLMD0	Low level	High level

Caution Since FLMD0 pin is also used as alternate-function test pin, make the FLMD0 pin voltage V_{SS} after a reset is released.

2.3 Self Programming Mode and FLMD0 Pin Control

Figure 2-3 shows the actual voltage change timing for each register and the FLMD0 pin.

Figure 2-3. Self Programming Mode and FLMD0 Signal Timing


FLMD0 pin becomes high level in self programming mode
FLMD0 pin becomes low level in self programming mode
RESET
FLMD0 pin
Self programming mode (FLSPM1, FLSPM0)
Normal mode 00
Self programming mode 01
Normal mode 00
CALL !8100H

Caution Release a reset while low level is input to the FLMD0 pin.
After setting the self programming mode, input a high level to the FLMD0 pin prior to firmware execution (CALL !8100H) and keep this input level fixed until the firmware operation ends. Also, before returning to the normal mode, input a low level to the FLMD0 pin.

2.4 Functions of Flash Memory Control Firmware

Table 2-2 lists the flash memory control firmware functions (commands).
To use these functions, specify the function numbers in self programming mode.
For details about the functions, refer to **CHAPTER 3 ACCESS TO FLASH MEMORY**.

Table 2-2. Firmware Function (Command) List

Function No.	Function (Command)	Description	
00H	Initialize	Initialization	Performs initial settings for self programming.
01H	Reserved		
02H	Reserved		
03H	Block erase	Block erasure	Erases the data of the specified block (2 KB).
04H	Word write	Word writing	Writes the data in RAM to the flash memory. Up to 256 bytes (in 4-byte units) can be written at one time.
05H	Reserved		
06H	Block verify	Verification	Performs verify of the specified block (2 KB).
07H	Reserved		
08H	Block blank check	Blank check	Performs blank check of the specified block (2 KB).
09H	Get information	Information acquisition	Reads the set information in the flash memory.
0AH	Set information	Information setting	Performs security, boot swap[Note], and other settings.
0BH	Reserved		
0CH	Reserved		
0DH	Reserved		
0EH	Mode check	FLMD0 voltage check	Checks the voltage level of the FLMD0 pin.
0FH to 16H	Reserved		
17H	EEPROM write	EEPROM emulation data writing	During EEPROM emulation, writes the data in RAM to the flash memory. Up to 256 bytes (in 4-byte units) can be written at one time.
1CH	EEPROM erase	Data erasure for EEPROM emulation	During EEPROM emulation, the memory in a specified block is deleted only by the duration of given time (10 ms units).

Note The boot swap function is supported only with the μPD78F0714.

2.5 Flash Memory Control Firmware Use Environment

Table 2-3 lists the conditions required for operation of the flash memory control firmware.

Table 2-3. Conditions for Firmware Operation

Item	Description
Secure entry RAM area	During firmware operation, 48 bytes are required as the entry RAM area. Any addresses in the internal high-speed RAM can be specified as this entry RAM area.
Secure stack area	During firmware operation, the stacks used by the user program are taken over and used. An additional 30-byte stack area must be secured from the stack addresses at firmware execution start.
Secure data buffer	During flash memory write, an area for temporarily holding the write data must be secured. This area is known as a data buffer, and any address in the internal high-speed RAM and any size from 4 to 256 bytes that is a multiple of 4 can be specified for this buffer.
Save general-purpose register	Because the value of register bank 3 is overwritten during firmware operation. Save and restore the value of register bank 3 as needed.
Timer (TM50)	Since internal timer (TM50) is used by the firmware, TM50 cannot be used by user programs during firmware operation. Be sure to mask the interrupt of TM50. Moreover, since TM50 is initialized at the end of firmware operation, if it needs to be used by a user program, it must be set again.
WDT operation	The firmware refreshes the watchdog timer enable register of WDT periodically. During a firmware execution, a reset by WDT, etc. does not occur.
Interrupt masking	Before executing firmware, disable all interrupts. Disable interrupts using the interrupt mask flag register or the DI instruction. However, be sure to use the interrupt mask flag register for TM50.
Program allocation of self programming	In the self programming mode, instructions at addresses 0 to 7FFFH can be executed.
Manipulation of FLMD0 voltage	Before executing the firmware, stabilize the voltage input to the FLMD0 pin with the V_{DD} voltage. Before changing the mode to the normal mode, set the voltage input to the FLMD0 pin to V_{SS}.
Reset	Do not reset this microcontroller during firmware operation. The data of the flash memory accessed upon reset becomes undefined[Note].
Power supply cutoff/instantaneous power supply interruption	Supply a stable voltage to the microcontroller during firmware operation. The data of flash memory accessed during power supply cutoff/instantaneous power supply interruption becomes undefined[Note].

Note For countermeasures in regards to instantaneous power supply interruptions, refer to **3.14 Boot Swap Function (μPD78F0714 only)**.

2.6 Parameters for Controlling Flash Memory Control Firmware

As was mentioned above, access to flash memory (for erasing or writing) is performed by the firmware. The operation instructions are sent from the user application to the firmware via parameters in RAM. There are the following three types of control parameters.

- Register bank 3
- Entry RAM
- Data buffer

Each type of parameter is described below.

2.6.1 Register bank 3

When executing the flash memory control firmware, set the firmware's function numbers to C register of register bank 3, and the start address of the entry RAM to HL register.

The firmware execution results can be checked with B register.

Table 2-4. List of Register Bank 3 Parameters

Register Function Name	C Register Function No.	B Register's Execution Result (Return Value)	HL Register	AX Register	DE Register
Initialization	00H	00H: Normal end 05H: Parameter error	Start address of entry RAM[Note]	Not used (used by firmware)	
Block erase	03H	00H: Normal end 05H: Parameter error 1AH: Erasure error			
Word write	04H	00H: Normal end 05H: Parameter error 18H: FLMD0 error 1CH: Write error			
Block verify	06H	00H: Normal end 05H: Parameter error 1BH: Internal verification error			
Block blank check	08H	00H: Normal end 05H: Parameter error 1BH: Blank check error			
Get information	09H	00H: Normal end 05H: Parameter error			
Set information	0AH	00H: Normal end 05H: Parameter error 18H: FLMD0 error 1BH: Internal verification error 1CH: Write error			
Mode check	0EH	00H: Normal end 01H: Error			
EEPROM write	17H	00H: Normal end 05H: Parameter error 18H: FLMD0 error 1CH: Write error 1DH: Internal verification error 1EH: Blank error			
EEPROM erase	1CH	00H: Normal end 05H: Parameter error 1AH: Erasure error			

Note Any address in the internal high-speed RAM can be set. However, secure a 48-byte area.

2.6.2 Entry RAM

The entry RAM is a 48-byte area used to set the firmware functions. The setting contents vary according to the firmware function.

The entry RAM can be placed to any addresses in the internal high-speed RAM, and the start address of the entry RAM is set with HL register of register bank 3. As shown in Table 2-5, each parameter is placed to “start address of entry RAM + offset value”.

The entry RAM can also be used as the work area of the firmware. Therefore, do not change data other than parameters during self programming.

Table 2-5. List of Entry RAM Parameters

Offset Value Function (C Register)	Function No.	+00H	+01H	+02H	+03H	+04H to +06H	+07H	+08H, +09H	+0AH	+0BH	+0CH to +2FH
Initialization	00H	00H: Normal end 05H: Parameter error						Start address of data buffer			
Block erase	03H	00H: Normal end 05H: Parameter error 1AH: Erasure error					Block no.				
Word write	04H	00H: Normal end 05H: Parameter error 18H: FLMD0 error 1CH: Write error	Start address of flash memory Lower	Higher	Highest[Note]		Word count	Start address of data buffer			
Block verify	06H	00H: Normal end 05H: Parameter error 1BH: Internal verification error					Block no.				
Block blank check	08H	00H: Normal end 05H: Parameter error 1BH: Blank error					Block no.				
Get information	09H	00H: Normal end 05H: Parameter error	Block no.				Option value	Start address of data buffer			
Set information	0AH	00H: Normal end 05H: Parameter error 18H: FLMD0 error 1BH: Internal verification error 1CH: Write error						Start address of data buffer			
Mode check	0EH	00H: Normal end 01H: Error									
EEPROM write	17H	00H: Normal end 05H: Parameter error 18H: FLMD0 error 1CH: Write error 1DH: Internal verification error 1EH: Blank error	Start address of flash memory Lower	Higher	Highest[Note]		Word count	Start address of data buffer			
EEPROM erase	1CH	00H: Normal end 05H: Parameter error 1AH: Erasure error					Block no.			Retry count	

Note Set the highest address to 00H.

2.6.3 Data buffer

The data buffer is an area used to pass and receive data to be written to the flash memory and set information, and its contents vary according to the firmware function.

The data buffer can be placed to any address in the internal high-speed RAM, and its start address is specified in the entry RAM. The data buffer size differs according to the function, and an area between 4 and 256 bytes is required[Note].

Note The data buffer size is set in the entry RAM only when performing word write. The size of the data written at one time can be set in the range of 4 to 256 bytes (in 4-byte units).

Table 2-6. List of Data Buffer Parameters

Function Name		Function No.	Data Buffer Size (Bytes)		Data Buffer Contents				
	Option Value Entry RAM				+00H	+01H	+02H	+03H	+04H to +FFH
Initialization		00H	4	Frequency data	Data 1	Data 2	Data 3	Data 4	Not used
Block erase		03H	–	–	Not used				
Word write		04H	4 to 256	Write data	Write data				
Block verify		06H	–	–	Not used				
Block blank check		08H	–	–	Not used				
Get information	03H	09H	1	Security flag	Bits 1, 0: Block erase disable (other than 1, 1)/ enable (1, 1) Bits 3, 2: Chip erase disable (other than 1, 1)/ enable (1, 1) Bits 5, 4: Write disable (other than 1, 1)/enable (1, 1) **Note 1**	Not used			
	04H		1	Boot flag	00H: Boot area is swapped 01H: Boot area is not swapped	Not used			
	05H		3	End address of specified block	Lower address	Higher address	Highest address[Note 2]	Not used	
Set information		0AH	1	Information flag	Bit 0: Boot swap enable (0)/ disable (1)[Note 3] Bit 1: Chip erase disable (0)/ enable (1) Bit 2: Block erase disable (0)/ enable (1) Bit 3: Write disable (0)/ enable (1)	Not used			
Mode check		0EH		–	Not used				
EEPROM write		17H	4 to 256	Write data	Write data				
EEPROM erase		1CH	–	–	Not used				

Notes 1. Set bits 6 and 7 to 1.

2. Set the highest address to 00H.

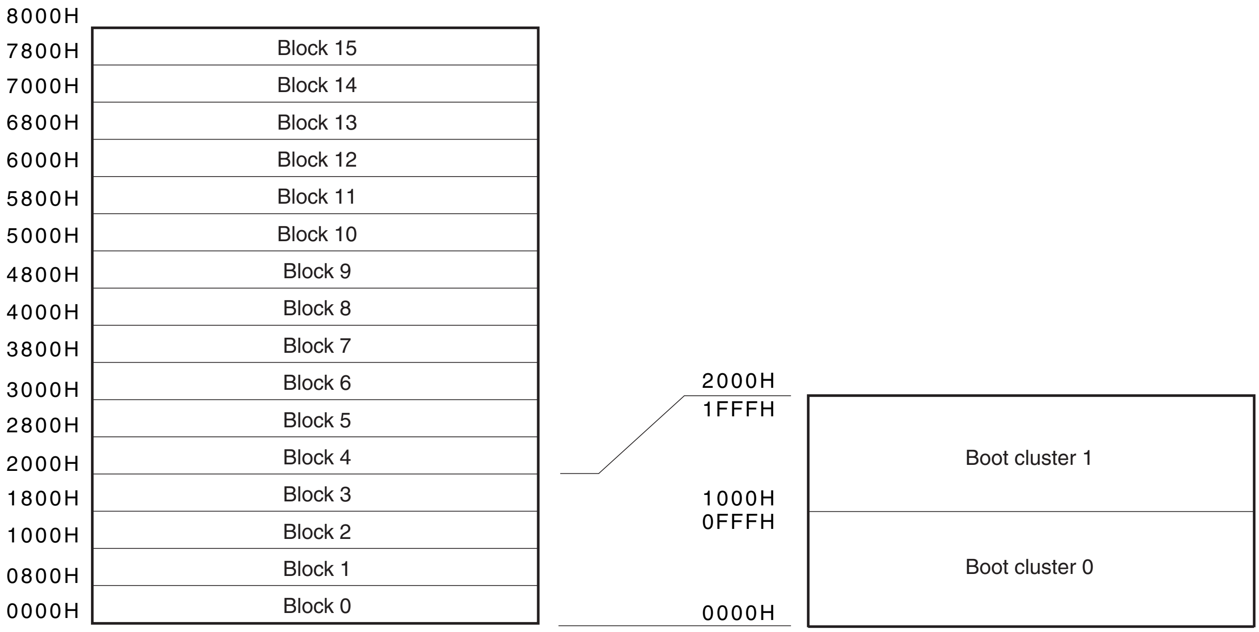
3. Setting bit 0 is supported only with the μPD78F0714.

2.6.4 Flash memory block numbers

The areas used to perform blank check, erase, and verify are specified in block (2 KB) units.
The boot swap (μPD78F0714 only) is performed in cluster (4 KB) units.

Caution Areas (banks) other than the on-chip flash memory of the product cannot be accessed.

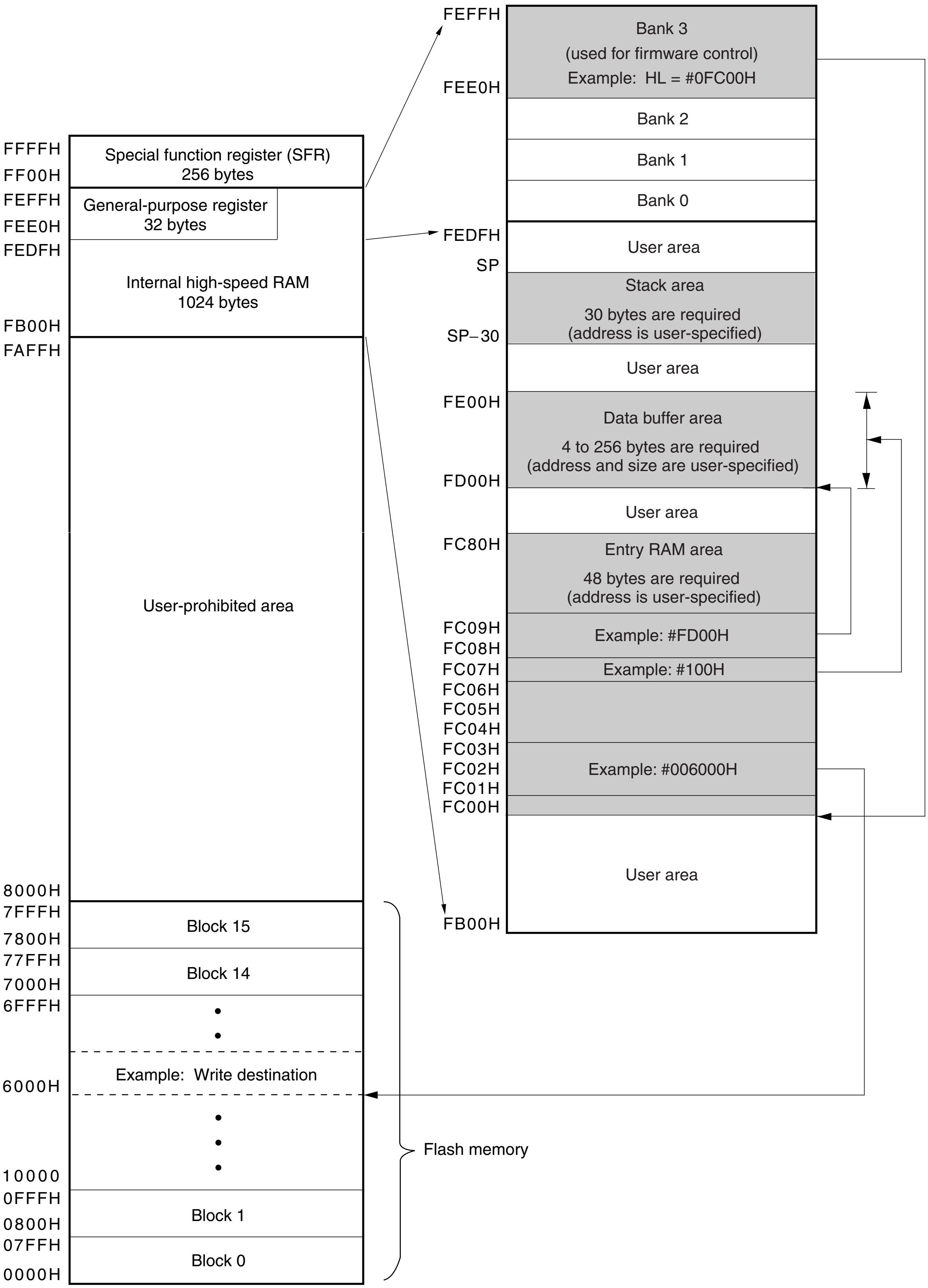
Figure 2-4. Allocation of Block Numbers (μPD78F0714)



2.6.5 RAM memory map

Figure 2-5 shows the memory map during flash programming.

Figure 2-5. Memory Map During Flash Programming (Word Write) (Example in μPD78F0714)



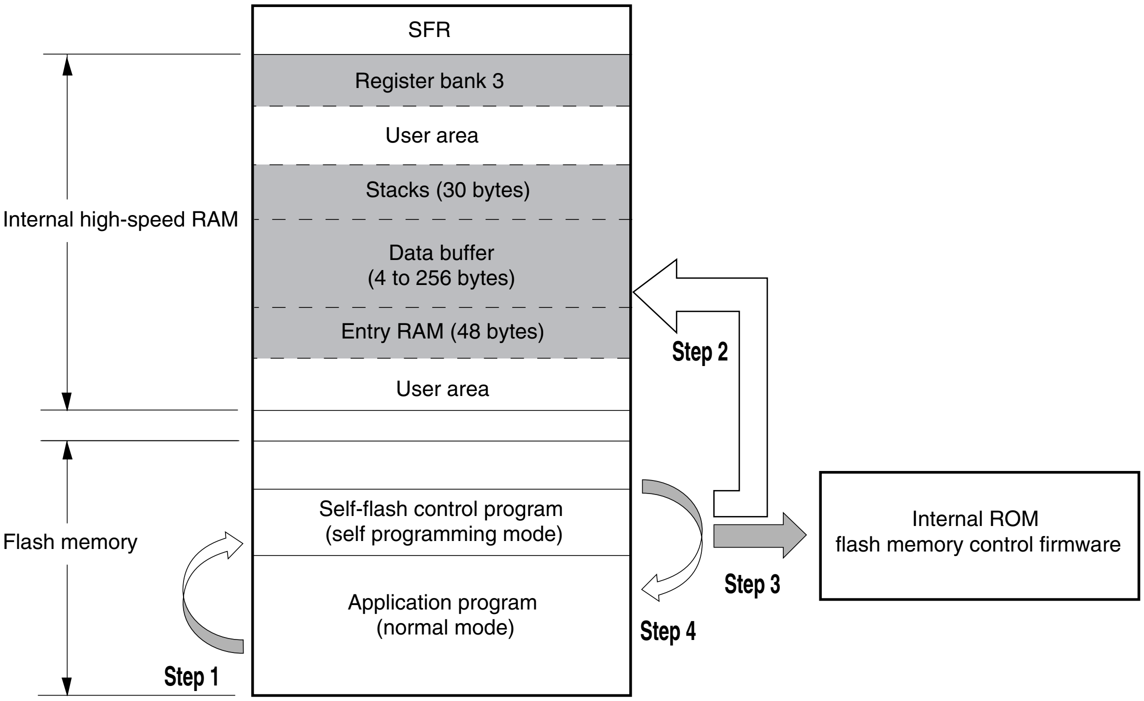
CHAPTER 3 ACCESS TO FLASH MEMORY

This chapter describes flash memory access methods.

3.1 Overall Flow

The following describes the overall flow of flash memory manipulations by the entry program. At each stage in this flow of manipulations, the flash memory control firmware is called only when all of the hardware and software use conditions have been met and all of the required parameters have been set.

Figure 3-1. Overall Flow (Outline)



Step 1 Self flash control program execution

Step 2 Make settings

Step 3 Firmware execution (CALL #8100H)

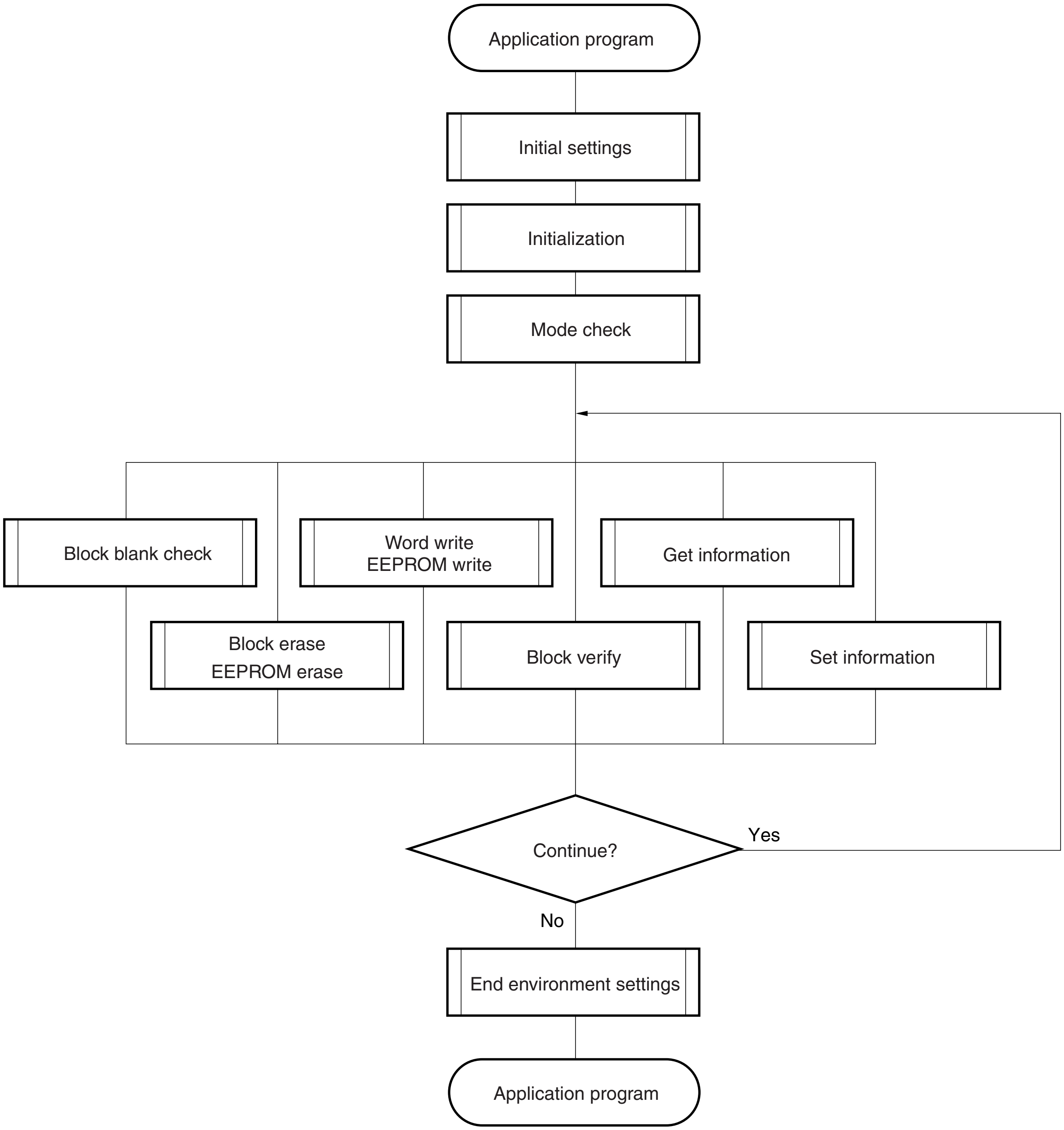
Step 4 Restore to normal operation

Caution Place the self-flash control program in the range of 0000H to 7FFFH.

Figure 3-2 outlines the programming part for self-flash.

Before performing flash memory write/erase, perform [Initial settings] → [Initialization] → [Mode check], in this order. Then execute the required functions the required number of times.

Figure 3-2. Self Flash Programming (Outline)



3.2 Initial Settings

Check the usage conditions and enable self programming.

<1> All interrupts are masked (interrupts are masked by the interrupt mask flag or executed by the DI instruction).
<2> TM50 is not being used (TM50 will be used by the firmware).
<3> Entry RAM area is available (reserve a 48-byte area at any address in RAM).
<4> Stack area is available (a 30-byte stack area is required for use by the firmware. Save the required data).
<5> Data buffer area is available (Secure an area of 4 to 256 bytes covering any addresses in RAM).
<6> Select register bank 3 (register bank 3 will be used to control the firmware, so be sure to save any required data from that area).
<7> Change the FLPMC register[Note] and go to self programming mode & the write/erase enable mode.
<8> Check whether FLPMC register rewrite was done in the correct sequence (if the PFS register is set to 1, an error results).
<9> Input V_{DD} to the FLMD0 pin (stabilize V_{DD} until all accesses to flash memory are completed).
<10> Check that the FWEDIS flag and FWEPR flag have been set to 0 and 1, respectively.

Note A special sequence must be used when accessing the FLPMC register. For details, refer to **2.1.2 Flash protect command register (PFCMD)**.

3.3 Initialization

(1) Functions

- Perform initial settings of flash firmware
- The parameters used by the firmware are calculated based on the frequency data set to the data buffer and stored in the firmware use area in the entry RAM.

(2) Arguments

Item	Description
Function number	Set 00H to C register.
Start address of entry RAM	Set any address in internal high-speed RAM to HL register.
Start address of data buffer	Set any address in internal high-speed RAM to entry RAM (+08H, +09H).
Frequency data	Set frequency data to the data buffers (+00H, +01H, +02H, +03H).

(3) Return value

Return Value[Note]	Description	
00H	Normal end	
05H	Parameter error	Occurs when the frequency data is outside the allowable range.

Note Return value = B register, or in entry RAM (+00H)

(4) Register memory state following firmware execution

- Start address (HL register) of entry RAM held

(5) Stack size

- 26 bytes

(6) Other

- Frequency data: The frequency data (Hz) is set to the data buffer as a 4-digit hexadecimal value.

Table 3-1. Format of Oscillation Frequency Data

Offset	Description
+0	First digit
+1	Second digit
+2	Third digit
+3	Fourth digit

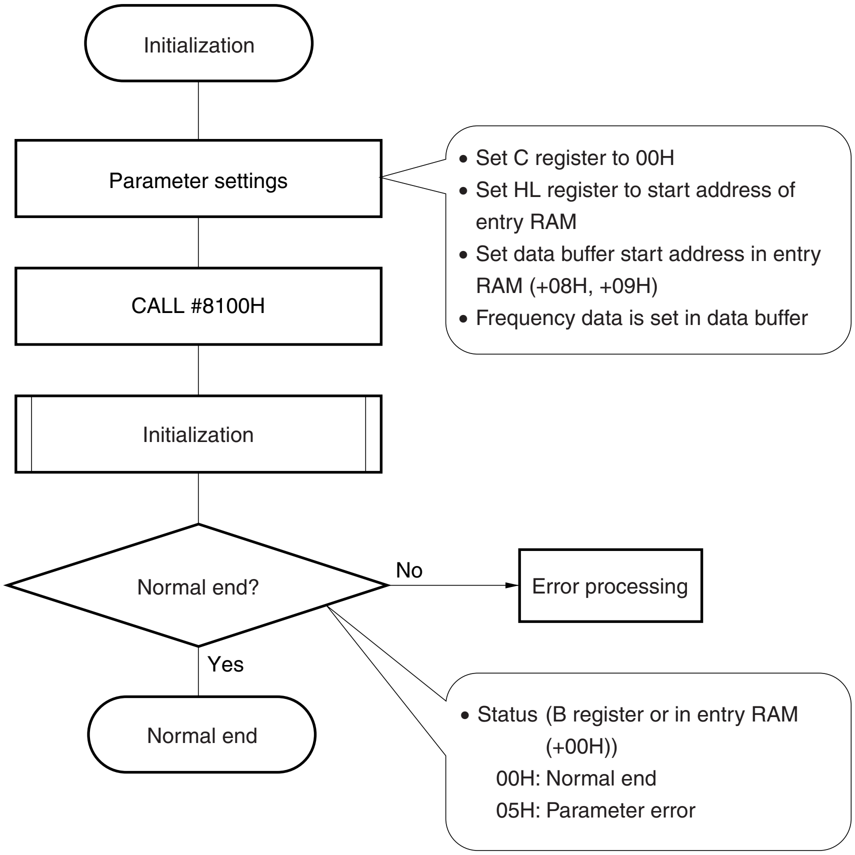
Oscillation frequency (Hz) = Fourth digit value × 1000000H + third digit value × 10000H + second digit value × 100H + first digit value

Example: 5 MHz = 5000000 Hz = 004C4B40H

Offset	Description
+0	40H: First digit
+1	4BH: Second digit
+2	4CH: Third digit
+3	00H: Fourth digit

(7) Flowchart example

Figure 3-3. Initialization Flowchart



- Set C register to 00H
- Set HL register to start address of entry RAM
- Set data buffer start address in entry RAM (+08H, +09H)
- Frequency data is set in data buffer

- Status (B register or in entry RAM (+00H))
  00H: Normal end
  05H: Parameter error

(8) Call example

```
MOV C,#00H          ; 00H → C register  Selects initialize function
MOVW HL,#0FC00H     ; 0FC00H → HL register  Sets start address of entry RAM to 0FC00H
MOVW AX,#0FD00H     ; 0FD00H → AX register
MOVW !0FC08H,AX     ; 0FD00H → entry RAM (+08H, +09H)  Sets start address of data buffer to
                    ; 0FD00H
PUSH HL             ; Saves HL register
MOVW HL,AX          ; AX register → HL register
MOV A,#60H          ; Sets frequency data 8.38 MHz = 007FDE60H
MOV [ HL+0] ,A      ; 00H → 0FB50H (+00H)
MOV A,#DEH          ; DEH → A register
MOV [ HL+1] ,A      ; DEH → 0FB51H (+01H)
MOV A,#7FH          ; 7FH → A register
MOV [ HL+2] ,A      ; 7FH → 0FB52H (+02H)
MOV A,#00H          ; 00H → A register
MOV [ HL+3] ,A      ; 00H → 0FB53H (+03H)
POP HL              ; Restores HL register
CALL !8100H         ; Executes flash firmware
```

3.4 Mode Check

(1) Function

- Check level (high or low) of FLMD0 pin

Caution If FLMD0 pin is at low level, flash memory cannot be erased or programmed. Therefore, when writing to flash memory with self programming, execute this subroutine after initialization.

(2) Arguments

Item	Description
Function number	Set C register to 0EH.
Start address of entry RAM	Set any address in internal high-speed RAM to HL register.

(3) Return value

Return Value[Note]	Description	
00H	Normal end	
01H	Error	If FWEPR (FLPMC.2) = 0, an error is set.

Note Return value = B register, or in entry RAM (+00H)

(4) Register memory state following firmware execution

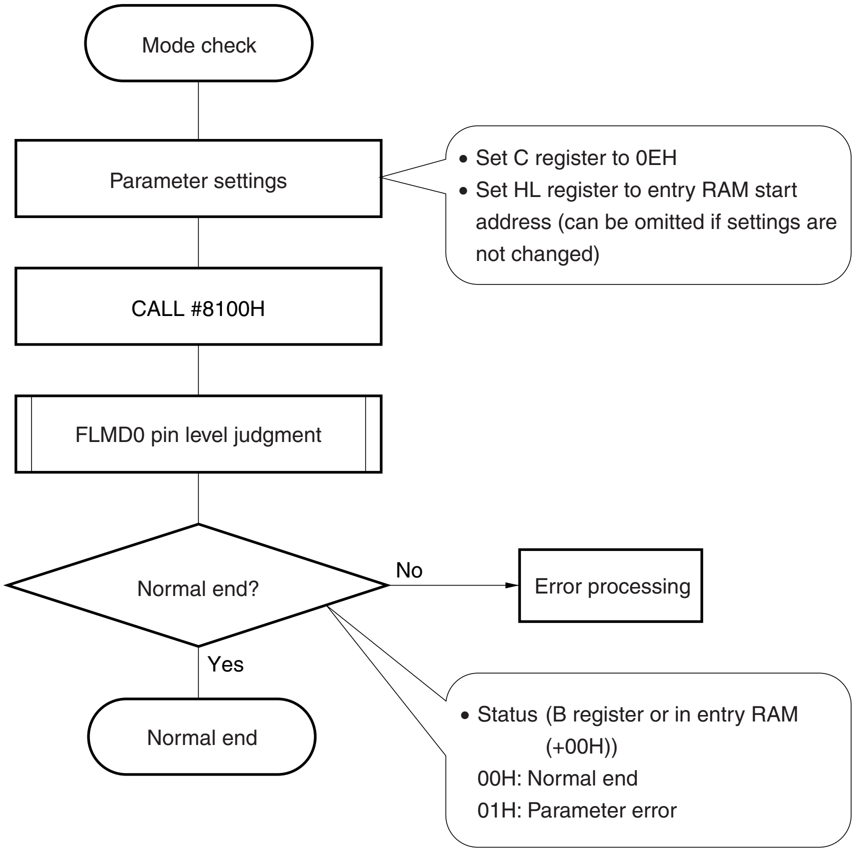
- Start address (HL register) of entry RAM held

(5) Stack size

- 12 bytes

(6) Flowchart example

Figure 3-4. Mode Check Flowchart



(7) Call example

```
MOV C,#0EH            ; 0EH → C register  Selects mode check function
MOVW HL,#0FC00H       ; 0FC00H → HL register  Sets start address of entry RAM to 0FC00H
CALL !8100H           ; Executes flash firmware
```

3.5 Block Blank Check

(1) Function

- Performs a blank check of a specified block.

(2) Arguments

Item	Description
Function no.	Set C register to 08H.
Start address of entry RAM	Set any address in internal high-speed RAM to HL register.
Blank check block	Set block number to be blank-checked to entry RAM (+07H).

(3) Return value

Return Value[Note]	Description	
00H	Normal end	
05H	Parameter error	Occurs when the specified block number is outside the specifiable range (exceeds total block count).
1BH	Blank check error	Occurs when the state was other than the blank state in blank check processing.

Note Return value = B register, or in entry RAM (+00H)

(4) Register memory state following firmware execution

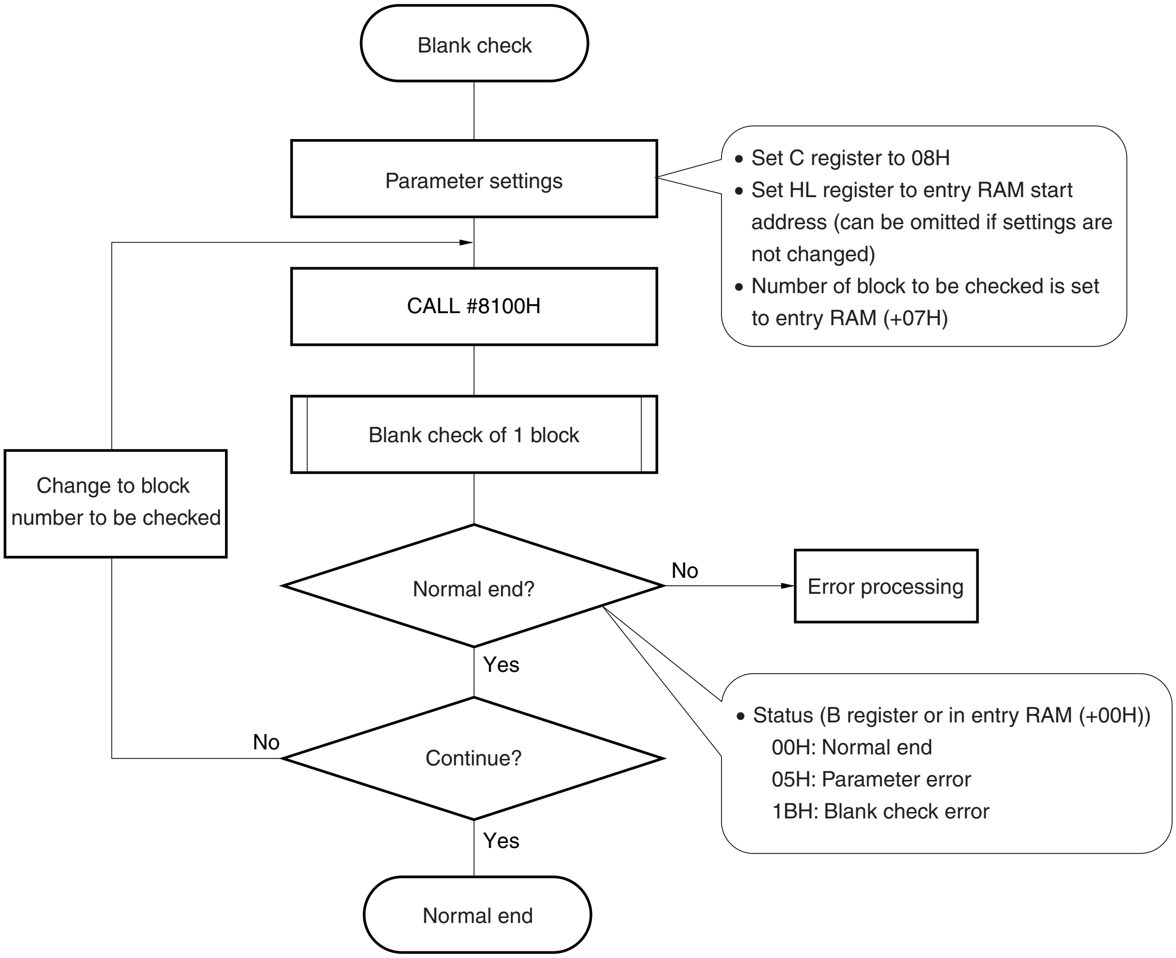
- Start address (HL register) of entry RAM held

(5) Stack size

- 14 bytes

(6) Flowchart

Figure 3-5. Block Blank Check Flowchart


Blank check
Parameter settings
• Set C register to 08H
• Set HL register to entry RAM start address (can be omitted if settings are not changed)
• Number of block to be checked is set to entry RAM (+07H)
CALL #8100H
Blank check of 1 block
Change to block number to be checked
Normal end?
No
Error processing
Yes
• Status (B register or in entry RAM (+00H))
00H: Normal end
05H: Parameter error
1BH: Blank check error
No
Continue?
Yes
Normal end

(7) Call example

```
MOV C,#08H          ; 08H → C register  Selects block blank check function
MOVW HL,#0FC00H     ; 0FC00H → HL register  Sets start address of entry RAM to 0FC00H
MOV A,#0            ; 00H → A register
MOV [ HL+7] ,A      ; 00H → entry RAM (+07H)  Specifies area for blank check to block 0
CALL !8100H         ; Executes flash firmware
```

3.6 Block Erasure

(1) Functions

- Erases a specified block.

(2) Arguments

Item	Description
Function no.	Set C register = 03H.
Start address of entry RAM	Set any address in internal high-speed RAM to HL register.
Erase block	Set block number to be erased to entry RAM (+07H).

(3) Return value

Return Value[Note]	Description	
00H	Normal end	
05H	Parameter error	Occurs when the specified block number is outside the specifiable range (exceeds total block count).
1AH	Erasure error	Occurs when the block could not be erased by erase processing.

Note Return value = B register, or in entry RAM (+00H)

(4) Register memory state following firmware execution

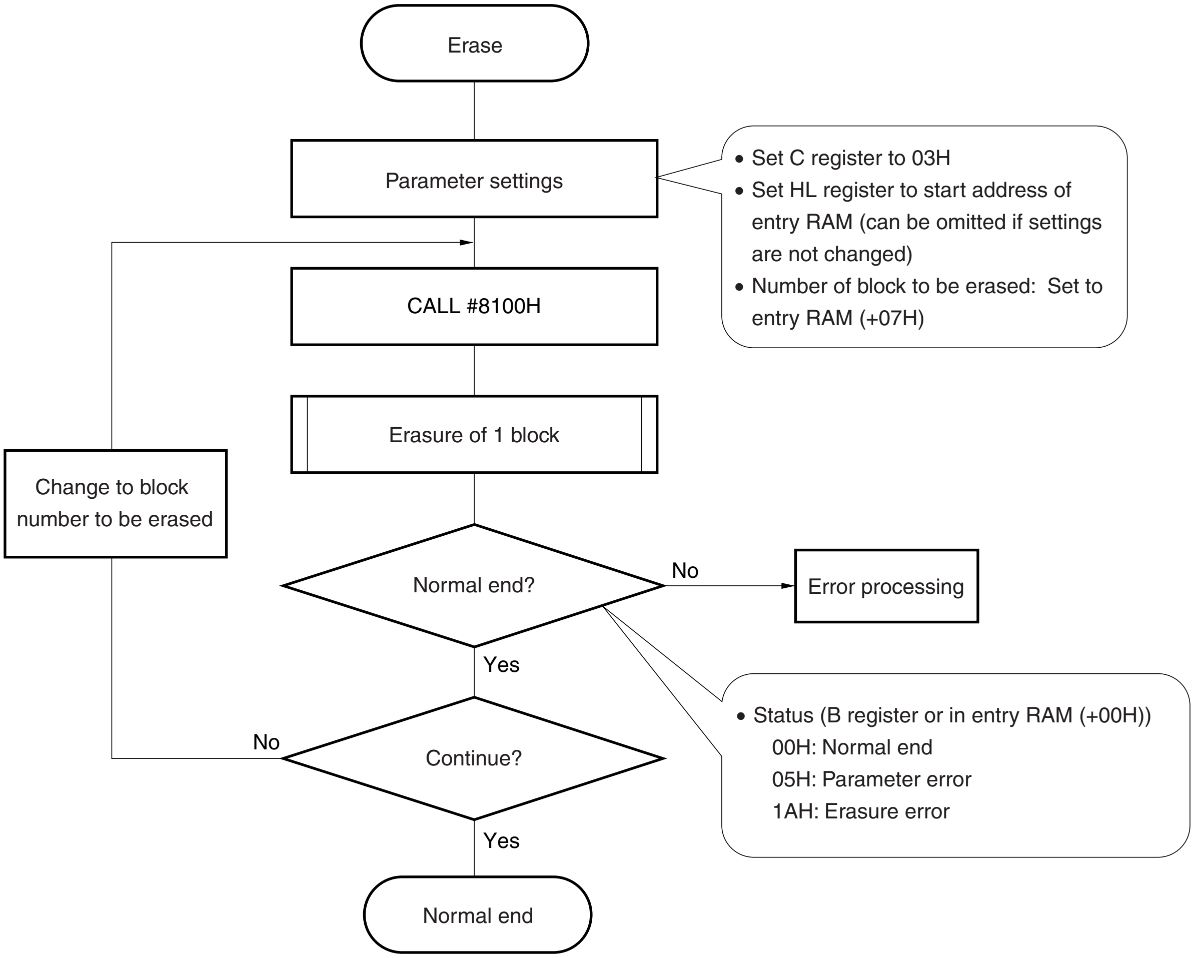
- Start address (HL register) of entry RAM held

(5) Stack size

- 30 bytes

(6) Flowchart

Figure 3-6. Block Erasure Flowchart


Erase
Parameter settings
• Set C register to 03H
• Set HL register to start address of entry RAM (can be omitted if settings are not changed)
• Number of block to be erased: Set to entry RAM (+07H)
CALL #8100H
Erasure of 1 block
Change to block number to be erased
Normal end?
No
Error processing
Yes
• Status (B register or in entry RAM (+00H))
00H: Normal end
05H: Parameter error
1AH: Erasure error
No
Continue?
Yes
Normal end

(7) Call example

```
MOV C,#03H            ; 03H → C register  Selects block erasure function
MOVW HL,#0FC00H       ; 0FC00H → HL register  Sets start address of entry RAM to 0FC00H
MOV A,#0              ; 00H → A register
MOV [ HL+7] ,A        ; 00H → entry RAM (+07H)  Specifies area for block erasure to block 0
CALL !8100H           ; Executes flash firmware
```

3.7 Word Write

(1) Function

- Writes data to the flash memory.
- Used to write programs.

(2) Arguments

Item	Description
Function no.	Set C register to 04H.
Start address of entry RAM	Set any address in internal high-speed RAM to HL register.
Start address of data buffer	Set any address in internal high-speed RAM to entry RAM (+08H, +09H).
Start address of write destination (start address of flash memory)	Set[Note 2] any address in internal high-speed RAM to entry RAM (+01H, +02H, +03H)[Note 1].
Write data size	Set any word count to entry RAM (+07H)[Note 3].
Write data	Place in data buffer

Notes **1.** +01H: Lower; +02H: Higher; +03H: Highest

2. Set address that is a multiple of 4.

3. Set within range of 1 to 64 words (1 word = 4 bytes).

(3) Return value

Return Value[Note]	Description	
00H	Normal end	
05H	Parameter error	• Occurs when start address is other than a multiple of 1 word (4 bytes). • Occurs when word count is 0. • Occurs when word count exceeds 64. • Occurs when start address and end address calculated from word count exceed flash memory area.
18H	FLMD0 error	Occurs when input voltage to FLMD0 pin is abnormal.
1CH	Write error	Occurs when anomaly occurs in verification for READ level.

Note Return value = B register, or in entry RAM (+00H)

(4) Register memory state following firmware execution

- Start address (HL register) of entry RAM held

(5) Stack size

- 18 bytes

(6) Flowchart

Figure 3-7. Word Write Flowchart

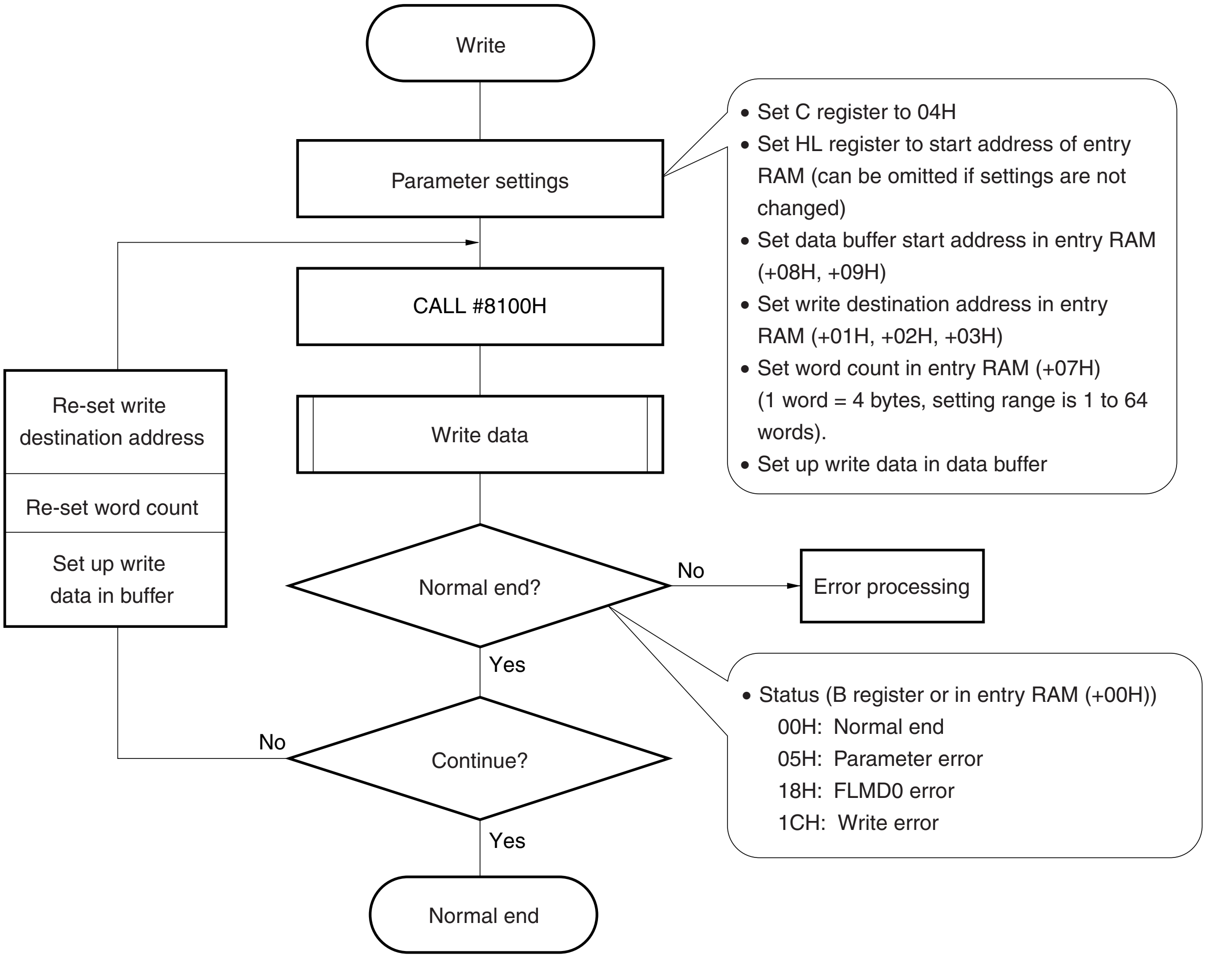
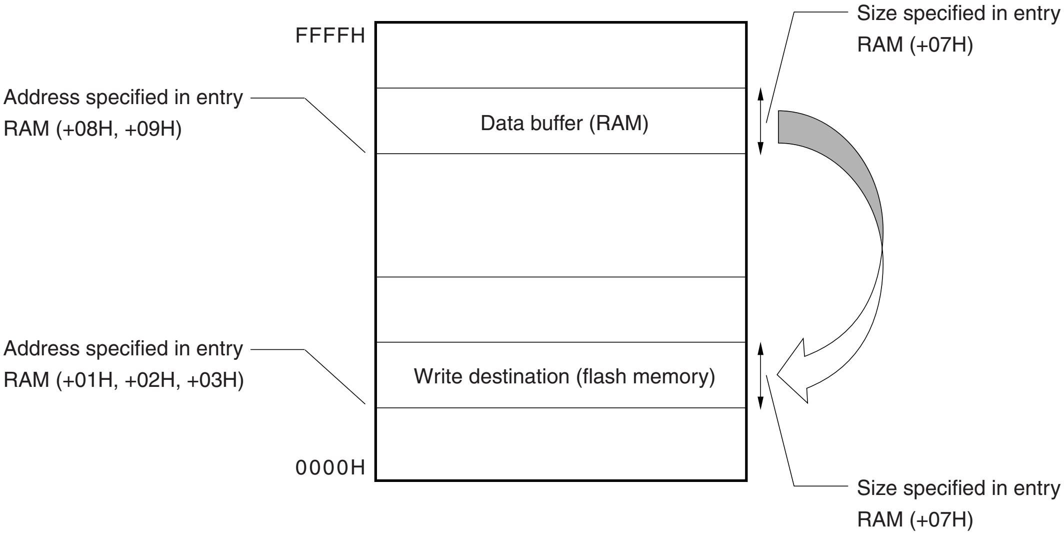

Write
Parameter settings
• Set C register to 04H
• Set HL register to start address of entry RAM (can be omitted if settings are not changed)
• Set data buffer start address in entry RAM (+08H, +09H)
• Set write destination address in entry RAM (+01H, +02H, +03H)
• Set word count in entry RAM (+07H) (1 word = 4 bytes, setting range is 1 to 64 words).
• Set up write data in data buffer
CALL #8100H
Write data
Re-set write destination address
Re-set word count
Set up write data in buffer
Normal end?
No
Error processing
Yes
• Status (B register or in entry RAM (+00H))
00H: Normal end
05H: Parameter error
18H: FLMD0 error
1CH: Write error
No
Continue?
Yes
Normal end

Figure 3-8. Memory Map


FFFFH
0000H
Address specified in entry RAM (+08H, +09H)
Data buffer (RAM)
Size specified in entry RAM (+07H)
Address specified in entry RAM (+01H, +02H, +03H)
Write destination (flash memory)
Size specified in entry RAM (+07H)

(7) Call example

```
MOV C,#04H           ;  04H → C register  Selects word write function
MOVW HL,#0FC00H      ; 0FC00H → HL register  Sets start address of entry RAM to 0FC00H
MOVW AX,#0FD00H      ; 0FD00H → AX register
MOVW !0FC08H,AX      ; 0FD00H → entry RAM (+08H, 09H)  Sets start address of data buffer to
                     ; 0FD00H
MOV A,#00H           ; 00H → A register
MOV [ HL+1] ,A       ; 00H → entry RAM (+01H)  Sets start address (lower) of write destination
                     ; to 00H
MOV A,#60H           ; 60H → A register
MOV [ HL+2] ,A       ; A register → entry RAM (+02H)  Sets start address (higher) of write destination
                     ; to 60H
MOV A,#00H           ; 00H → A register
MOV [ HL+3] ,A       ; A register → entry RAM (+03H)  Sets start address (highest) of write
                     ; destination to 00H
MOV A,#08H           ; 08H → A register
MOV [ HL+7] ,A       ; 08H → entry RAM (+07H)  Sets write size to 8 words (8 × 4 bytes)
CALL !8100H          ; Executes flash firmware
```

3.8 Block Verify

(1) Function

- Performs internal verification of the specified block[Note].
- After a write operation, be sure to perform internal verification of blocks that include the write range.

Note Internal verification is a function that checks to confirm that the data written to flash memory was written at an adequate level. This type of verification differs from data comparison verification.

(2) Arguments

Item	Description
Function no.	Set C register to 06H.
Start address of entry RAM	Set any address in internal high-speed RAM to HL register.
Verification block	Set block number to be verified to entry RAM (+07H).

(3) Return value

Return Value[Note]	Description	
00H	Normal end	
05H	Parameter error	Occurs when block number is outside settable range (exceeds total block count).
1BH	Internal verification error	Occurs when an error occurs during internal verify processing.

Note Return value = B register, or in entry RAM (+00H)

(4) Register memory state following firmware execution

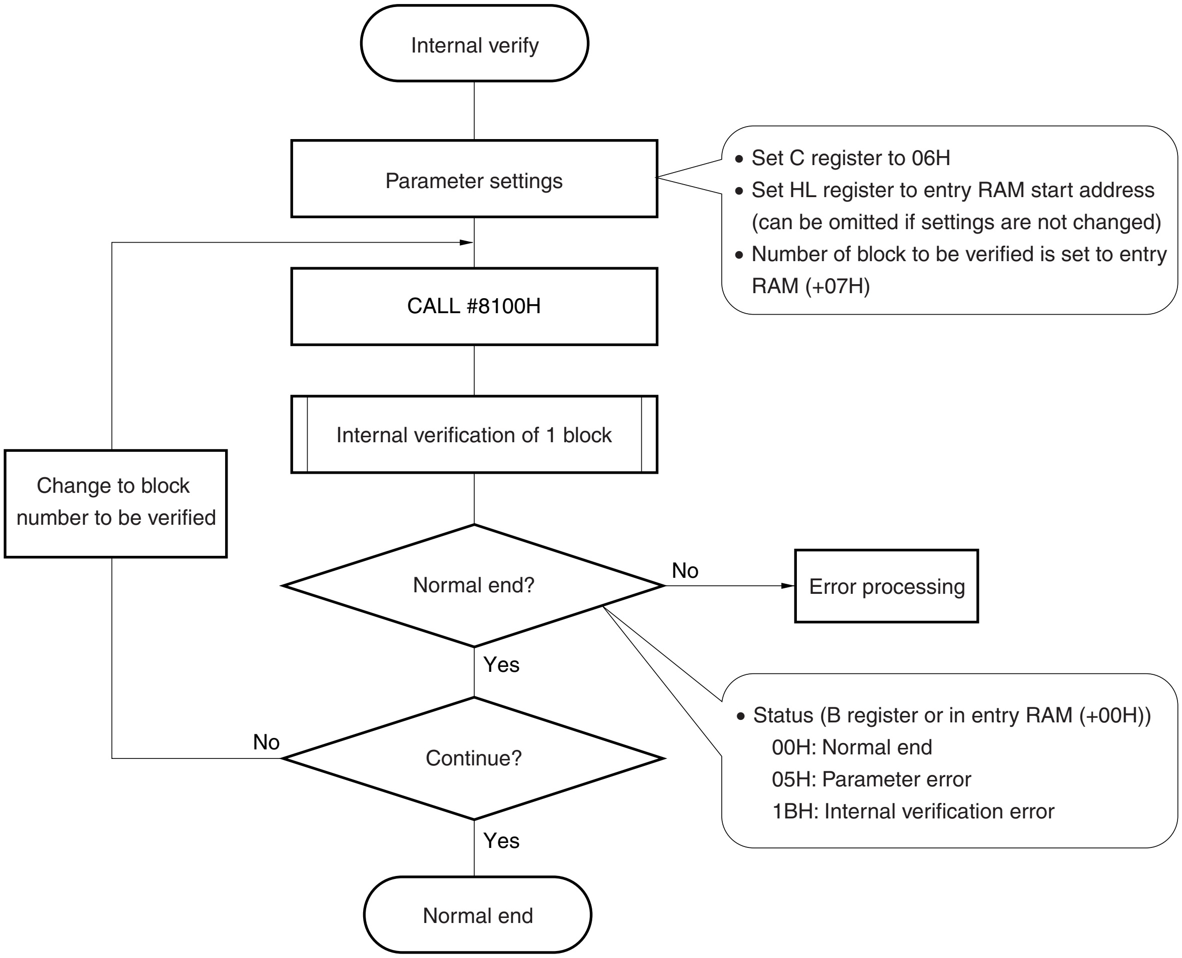
- Start address (HL register) of entry RAM held

(5) Stack size

- 14 bytes

(6) Flowchart

Figure 3-9. Block Verification Flowchart


Internal verify
Parameter settings
• Set C register to 06H
• Set HL register to entry RAM start address (can be omitted if settings are not changed)
• Number of block to be verified is set to entry RAM (+07H)
CALL #8100H
Internal verification of 1 block
Change to block number to be verified
Normal end?
No
Error processing
Yes
• Status (B register or in entry RAM (+00H))
00H: Normal end
05H: Parameter error
1BH: Internal verification error
No
Continue?
Yes
Normal end

(7) Call example

```
MOV C,#06H          ;  06H → C register  Selects block verify function
MOVW HL,#0FC00H     ;  0FC00H → HL register  Sets start address of entry RAM to 0FC00H
MOV A,#0            ;  00H → A register
MOV [ HL+7] ,A      ;  00H → entry RAM (+07H)  Specify area to be verified to block 0
CALL !8100H         ;  Executes flash firmware
```

3.9 Get Information

The firmware's flash information read function is used to read product information. The read information is then checked to see if it poses any problems for subsequent flash memory manipulations.

(1) Function

- Obtains various setting-related information

(2) Arguments

Item	Description
Function no.	Set C register to 09H.
Start address of entry RAM	Set any address in internal high-speed RAM to HL register.
Start address of data buffer	Set any address in internal high-speed RAM to entry RAM (+08H, +09H).
Option value	Set type of information to be obtained to entry RAM (+07H). 03H: Security flag information 04H: Boot flag information 05H: End address information of specified block
Information get block[Note]	Set specified block number to entry RAM (+01H)

Note Specification of the block number is valid only when the option value is 05H.

(3) Return value

Return Value[Note]	Description	
00H	Normal end	
05H	Parameter error	Occurs when option value exceeds range.

Note Return value = B register, or in entry RAM (+00H)

(4) Register memory state following firmware execution

- Start address (HL register) of entry RAM held
- Acquired information held in data buffer

(5) Stack size

- 22 bytes

(6) Get information for option value

The information selected with an option value is stored in the data buffer.

<1> Security flag information (Option value: 03H):

Obtain the onboard[Note] write/erase disable/enable status via the 1 byte of data from the beginning of the data buffer.

Note Onboard refers to write/erase using an external tool such as a flash programmer.

Table 3-2. Security Flag Data Format

Offset	Description
+0	Security flag information

Bit No.	Description
Security flag information – Bits 1, 0	Block erase enable flag Other than 1, 1: Onboard block erase disable 1, 1: Onboard block erase enable
Security flag information – Bits 3, 2	Chip erase enable flag Other than 1, 1: Onboard chip erase disable 1, 1: Onboard chip erase enable
Security flag information – Bits 5, 4	Write enable flag Other than 1, 1: Onboard write disable 1, 1: Onboard write enable
Security flag information – Bit 6	1
Security flag information – Bit 7	1

<2> Boot flag information (option value: 04H)

Get boot flag area state with 1 byte of data from the beginning of the data buffer.

Table 3-3. Boot Flag Data Format

Offset	Description
+0	Boot flag information 00H: Boot cluster 0 (0000H to 0FFFH, blocks 0 and 1) is selected. 01H: Boot cluster 1 (1000H to 1FFFH, blocks 2 and 3) is selected.

<3> End address of specified block (option value: 05H)

Get end address of specified block with 3 bytes of data from start of data buffer.

Table 3-4. End Address Data Format

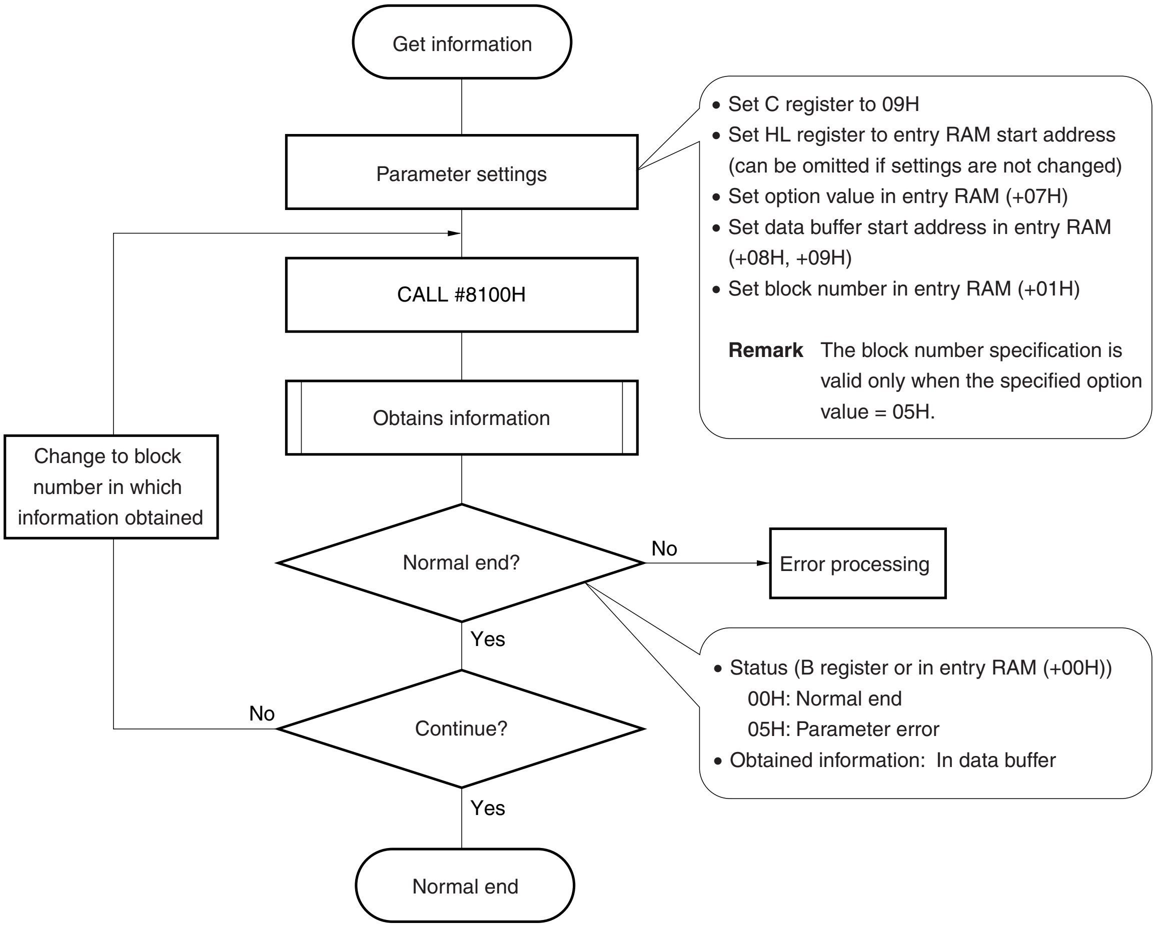
Offset	Description
+0	Block end address, lower address
+1	Block end address, higher address
+2	Block end address, highest address

Example: Block 0 (when end address is 0007FFH)

Offset	Description
+0	FFH
+1	07H
+2	00H

(7) Flowchart

Figure 3-10. Get Information Flowchart



- Set C register to 09H
- Set HL register to entry RAM start address (can be omitted if settings are not changed)
- Set option value in entry RAM (+07H)
- Set data buffer start address in entry RAM (+08H, +09H)
- Set block number in entry RAM (+01H)

Remark The block number specification is valid only when the specified option value = 05H.

- Status (B register or in entry RAM (+00H))
  00H: Normal end
  05H: Parameter error
- Obtained information: In data buffer

(8) Call example

```
MOV C,#09H           ; 09H → C register  Selects get information function
MOVW HL,#0FC00H      ; 0FC00H → HL register  Sets start address of entry RAM to 0FC00H
MOVW AX,#0FD00H      ; 0FD00H → AX register
MOVW !0FC08H,AX      ; 0FD00H → entry RAM (+08H, +09H)  Sets start address of data buffer to
                     ; 0FD00H
MOV A,#03H           ; 03H → A register
MOV [ HL+7] ,A       ; 03H → entry RAM (+07H)  Sets security flag information as information to get
CALL !8100H          ; Executes flash firmware
```

3.10 Set Information

(1) Function

- Sets the boot flag. As a result, boot area swap is possible (boot swap function[Note]).

Note The boot swap function is supported only with the μPD78F0714.

(2) Arguments

Item	Description
Function no.	Set C register to A0H.
Start address of entry RAM	Set any address in internal high-speed RAM to HL register.
Start address of data buffer	Set any address in internal high-speed RAM to entry RAM (+08H, +09H).
Option value	Set option value to 1st byte (+00H) in data buffer. Bit 0 → 0: Don't perform boot swap 1: Perform boot swap[Note 1] Bit 1 → 0: Onboard chip erase disable 1: Onboard chip erase enable[Notes 2, 3] Bit 2 → 0: Onboard block erase disable 1: Onboard block erase enable[Notes 2, 3] Bit 3 → 0: Onboard write disable 1: Onboard write enable[Notes 2, 3]

Notes 1. Because of the reset after the "Perform boot swap" was set, the boot cluster 0 (0000H to 0FFFH) and boot cluster 1 (1000H to 1FFFH) are swapped. For details, refer to **3.14 Boot Swap Function (μPD78F0714 only)**.

2. If rewriting the boot swap bit, overwrite the same value. If a different value is attempted to be input, a parameter error occurs (value is not rewritten) (μPD78F0714 only).

3. Restrictions imposed by this security flag do not apply to write/erase through self programming.

(3) Return value

Return Value[Note]	Description	
00H	Normal end	
05H	Parameter error	Occurs when the information flag security values (bit 1, bit 2, bit 3) are not the same as the option values of arguments.
18H	FLMD0 error	Occurs when input voltage of FLMD0 pin is abnormal.
1BH	Internal verification error	Occurs when an error occurs during internal verify processing.
1CH	Write error	Occurs when anomaly occurs in verify for READ level.

Note Return value = B register, or in entry RAM (+00H)

(4) Register memory state following firmware execution

- Start address (HL register) of entry RAM held

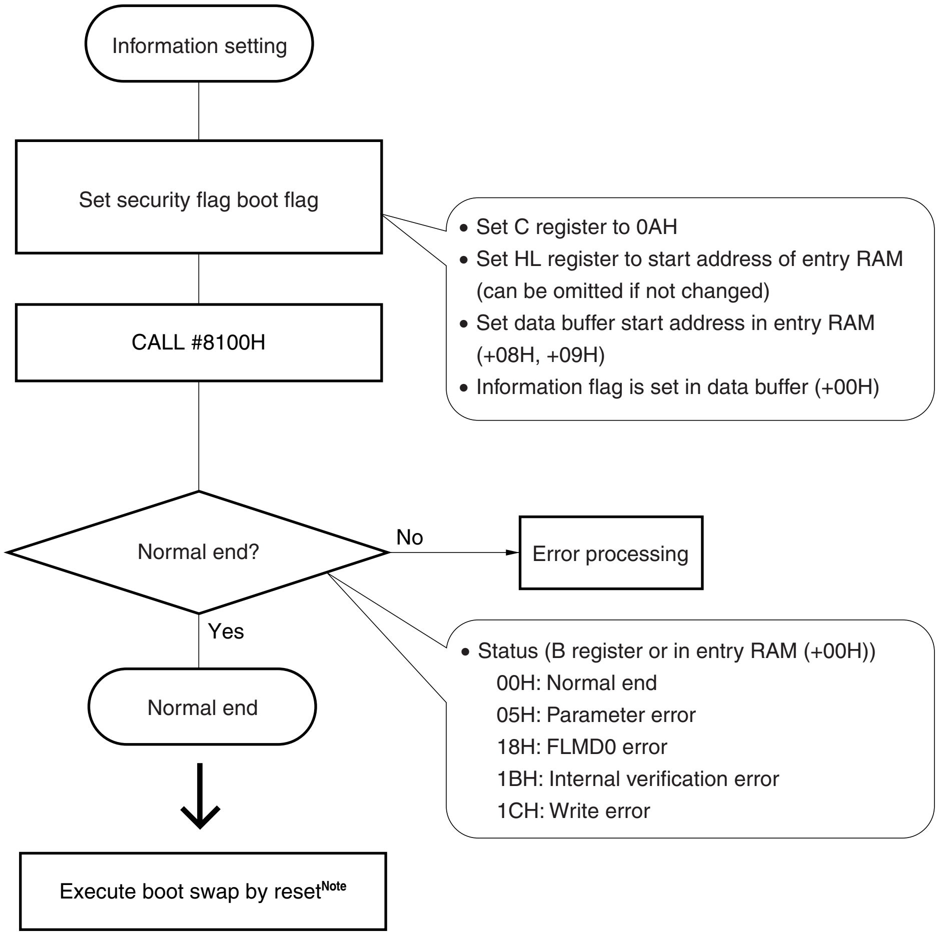
(5) Stack size

- 12 bytes

Caution The maximum number of times the security flag can be set is 100 times.

(6) Flowchart

Figure 3-11. Set Information Flowchart


Information setting
Set security flag boot flag
CALL #8100H
Normal end?
No
Error processing
Yes
Normal end
Execute boot swap by reset[Note]
• Set C register to 0AH
• Set HL register to start address of entry RAM (can be omitted if not changed)
• Set data buffer start address in entry RAM (+08H, +09H)
• Information flag is set in data buffer (+00H)
• Status (B register or in entry RAM (+00H))
00H: Normal end
05H: Parameter error
18H: FLMD0 error
1BH: Internal verification error
1CH: Write error

Note The boot swap function is supported only with the μPD78F0714.

(7) Call example

```
MOV C,#0AH              ; 0AH → C register  Selects set information function
MOVW HL,#0FC00H         ; 0FC00H → HL register  Sets start address of entry RAM to 0FC00H
MOVW AX,#0FD00H         ; 0FD00H → AX register
MOVW !0FC08H,AX         ; 0FD00H → entry RAM (+08H, +09H)  Sets start address of data buffer to
                        ; 0FD00H
MOV A,#00H              ; 00H → A register
MOV [HL] ,A             ; 00H → entry RAM (+00H)  Sets option value to 00H
CALL !8100H             ; Executes flash firmware
```

3.11 EEPROM Write

(1) Function

- Writes data to the flash memory during EEPROM emulation.
- Used to write data.

Remark For the data hold period and rewrite count, refer to the **μPD78F0711, 78F0712 User's Manual (U17890E)** or **μPD78F0714 User's Manual (U16928E)**.

(2) Arguments

Item	Description
Function no.	Set C register to 17H.
Start address of entry RAM	Set any address in internal high-speed RAM to HL register.
Start address of data buffer	Set any address in internal high-speed RAM to entry RAM (+08H, +09H).
Start address of write destination (start address of flash memory)	Set[Note 2] any address in internal high-speed RAM to entry RAM (+01H, +02H, +03H)[Note 1].
Write data size	Set any word count to entry RAM (+07H)[Note 3].
Write data	Place in data buffer

Notes **1.** +01H: Lower; +02H: Higher; +03H: Highest

2. Set address that is a multiple of 4.

3. Set within range of 1 to 64 words (1 word = 4 bytes).

(3) Return value

Return Value[Note]	Description	
00H	Normal end	
05H	Parameter error	• Occurs when start address is other than a multiple of 1 word (4 bytes). • Occurs when word count is 0. • Occurs when word count exceeds 64. • Occurs when start address and end address calculated from word count exceed flash memory area.
18H	FLMD0 error	Occurs when input voltage to FLMD0 pin is abnormal.
1CH	Write error	Occurs when anomaly occurs in verification for READ level.
1DH	Internal verification error	Occurs when an error occurs during internal verify processing.
1EH	Blank check error	Occurs when a free area corresponding to the write word count is not available.

Note Return value = B register, or in entry RAM (+00H)

(4) Register memory state following firmware execution

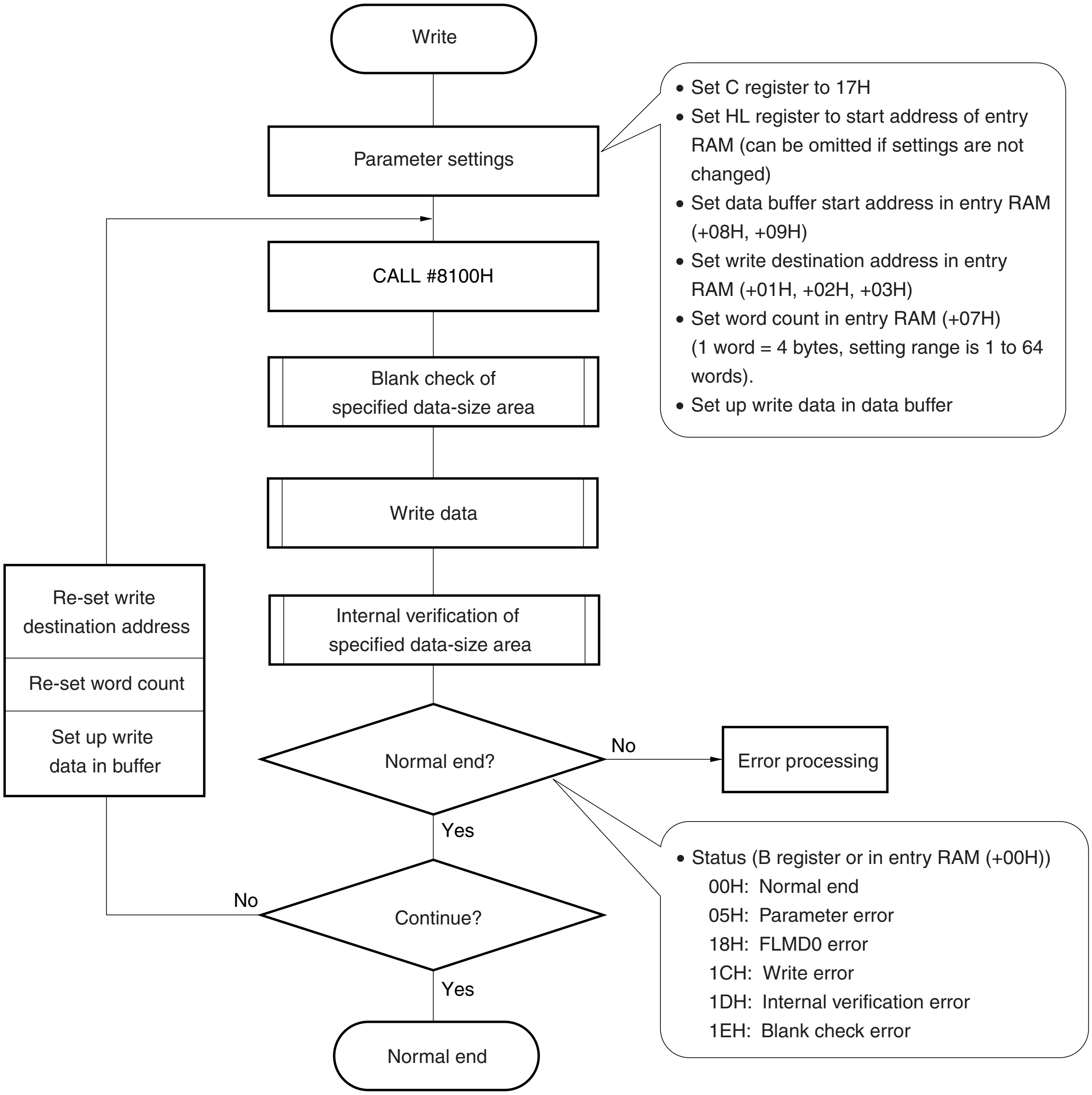
- Start address (HL register) of entry RAM held

(5) Stack size

- 15 bytes

(6) Flowchart

Figure 3-12. EEPROM Write Flowchart

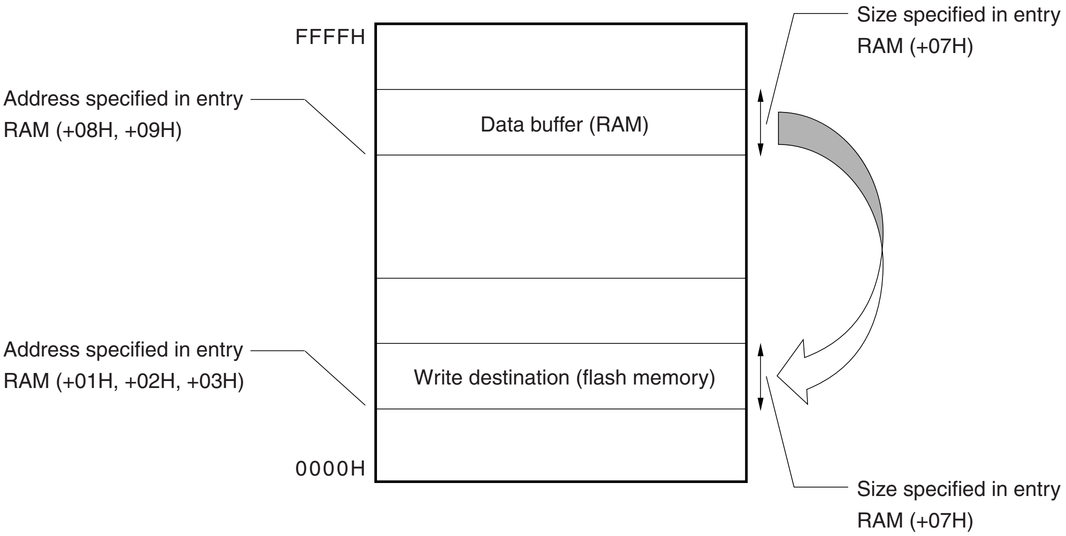

Write
Parameter settings
CALL #8100H
Blank check of specified data-size area
Write data
Internal verification of specified data-size area
Normal end?
No
Error processing
Yes
Continue?
No
Yes
Normal end
Re-set write destination address
Re-set word count
Set up write data in buffer

- Set C register to 17H
- Set HL register to start address of entry RAM (can be omitted if settings are not changed)
- Set data buffer start address in entry RAM (+08H, +09H)
- Set write destination address in entry RAM (+01H, +02H, +03H)
- Set word count in entry RAM (+07H) (1 word = 4 bytes, setting range is 1 to 64 words).
- Set up write data in data buffer

- Status (B register or in entry RAM (+00H))

00H:	Normal end
05H:	Parameter error
18H:	FLMD0 error
1CH:	Write error
1DH:	Internal verification error
1EH:	Blank check error

Figure 3-13. Memory Map


FFFFH
0000H
Address specified in entry RAM (+08H, +09H)
Data buffer (RAM)
Address specified in entry RAM (+01H, +02H, +03H)
Write destination (flash memory)
Size specified in entry RAM (+07H)
Size specified in entry RAM (+07H)

(7) Call example

```
MOV C,#17H          ;  17H → C register  Selects EEPROM write function
MOVW HL,#0FC00H     ; 0FC00H → HL register  Sets start address of entry RAM to 0FC00H
MOVW AX,#0FD00H     ; 0FD00H → AX register
MOVW !0FC08H,AX     ; 0FD00H → entry RAM (+08H, 09H)  Sets start address of data buffer to
                    ; 0FD00H
MOV A,#00H          ; 00H → A register
MOV [ HL+1] ,A      ; 00H → entry RAM (+01H)  Sets start address (lower) of write destination
                    ; to 00H.
MOV A,#60H          ; 60H → A register
MOV [ HL+2] ,A      ; A register → entry RAM (+02H)  Sets start address (higher) of write destination
                    ; to 60H.
MOV A,#00H          ; 00H → A register
MOV [ HL+3] ,A      ; A register → entry RAM (+03H)  Sets start address (highest) of write
                    ; destination to 00H.
MOV A,#08H          ; 08H → A register
MOV [ HL+7] ,A      ; 08H → entry RAM (+07H)  Sets write size to 8 words (8 × 4 bytes).
CALL !8100H         ; Executes flash firmware
```

3.12 EEPROM Erase

(1) Functions

- During EEPROM emulation, the memory in a specified block is deleted only by the duration of given time (10 ms units). With this function, the deletion of one block can be divided into a number of short erasure processings.
- Used to erase data.

(2) Arguments

Item	Description
Function no.	Sets C register to 1CH.
Start address of entry RAM	Sets given addresses in the internal high-speed RAM to HL register.
Erase block	Sets the block no. of the block to be erased to the entry RAM (+07H).
Retry count	Sets the retry count to the entry RAM (+0BH) to determine the erasure time. Erasure time = retry count × 10 ms

Remarks 1. The user program is suspended (interrupt disabled) during an erasure. Therefore, set the erasure time so that no problems occur if the user program is suspended for the set time.

2. It is recommended that making the accumulated erasure time by an EEPROM erasure reach the erasure time of 1 block before writing to the specified block (for the erasure time of 1 block, refer to the electrical specifications in the user's manual of each microcontroller).

3. After the erasure of the specified block completes, any other erasure processing for that block is canceled. Therefore, no over erasures or the like occur.

(3) Return value

Return Value[Note]	Description	
00H	Normal end	
05H	Parameter error	Occurs when specified block no. is out of the settable range (total block number or more).
1AH	Erase error	Occurs if data cannot be erased by erasure processing.

Note Return value = B register, or in the entry RAM (+00H)

(4) Register memory status after firmware execution

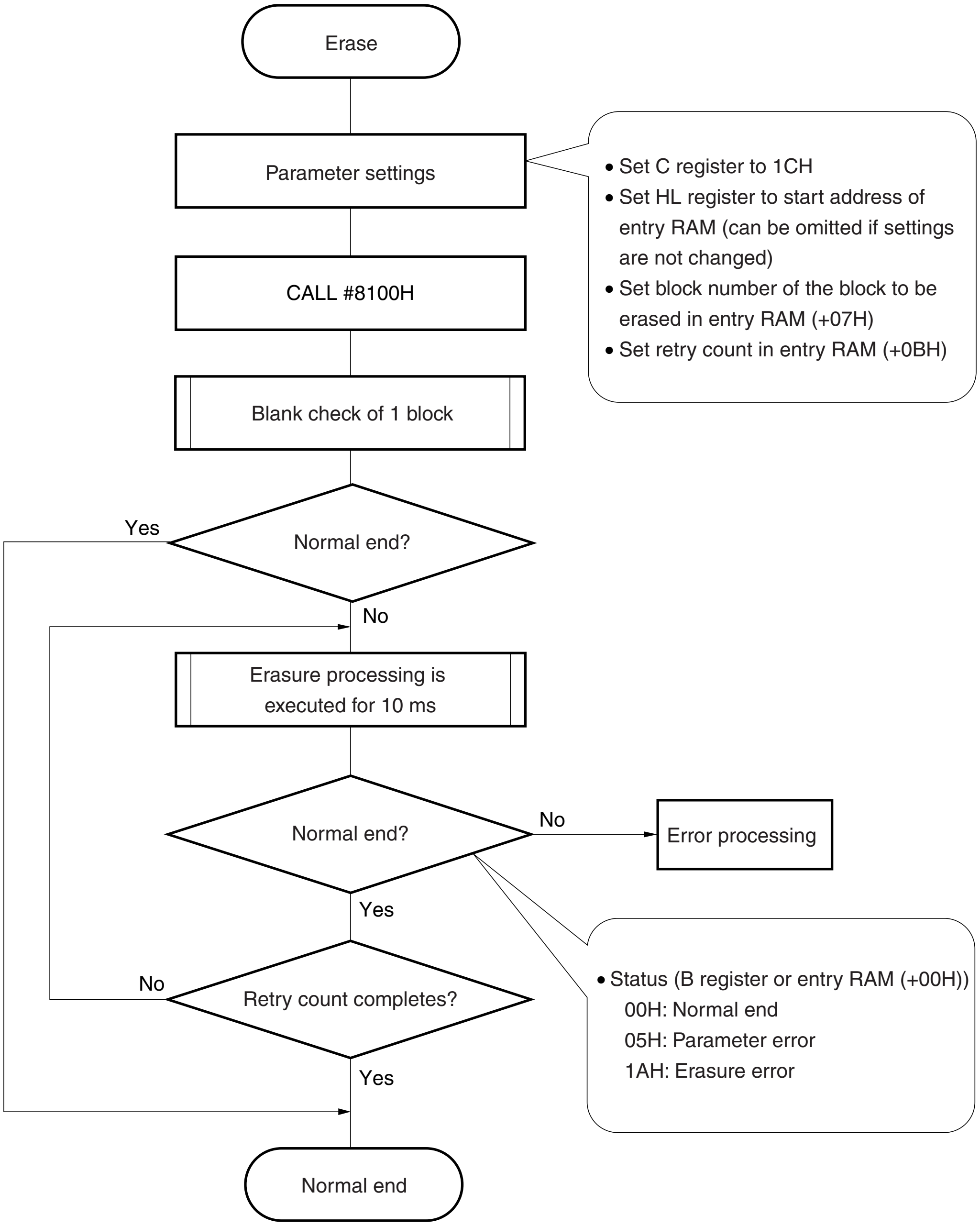
- Retains the start address (HL register) of the entry RAM.

(5) Stack size

- 17 bytes

(6) Flowchart

Figure 3-14. EEPROM Erase Flow


Erase
Parameter settings
• Set C register to 1CH
• Set HL register to start address of entry RAM (can be omitted if settings are not changed)
• Set block number of the block to be erased in entry RAM (+07H)
• Set retry count in entry RAM (+0BH)
CALL #8100H
Blank check of 1 block
Yes
Normal end?
No
Erasure processing is executed for 10 ms
Normal end?
No
Error processing
Yes
• Status (B register or entry RAM (+00H))
00H: Normal end
05H: Parameter error
1AH: Erasure error
No
Retry count completes?
Yes
Normal end

(7) Call example

```
MOV C,#1CH;         1CH → C register  Selects EEPROM erasure function
MOVW HL,#0FC00H;    0FC00H → HL register  Sets start address of entry RAM to 0FC00H
MOV A,#0;           00H → A register
MOV [HL+7],A;       00H → entry RAM (+07H)  Specify block 0 as the area for which EEPROM erasure is
                    performed.
MOV A,#02H;         02H → A register
MOV [HL+0BH],A;     02H → entry RAM (+0BH)  Sets retry count to twice (erasure time: 20 ms)
CALL !8100H;        Executes flash firmware
```

3.13 Status List

Table 3-5 shows the list of the firmware statuses (return values).

Table 3-5. Status List

Status	Description
00H	Normal end
05H	Parameter error (parameter setting error)
18H	FLMD0 error (write error due to abnormal FLMD0 levels)
1AH	MRG10 error (erasure error)
1BH	MRG11 error (internal verification error, blank check error)
1CH	Write error (write error due to verification abnormality at READ level)
1DH	MRG11 error (internal verification error) [on EEPROM write]
1EH	MRG11 error (blank check error) [on EEPROM write]

3.14 Boot Swap Function (μPD78F0714 only)

If, during boot area rewrite, rewrite fails due to an instantaneous power supply interruption, etc., the data in the boot area is lost and the program cannot be restarted through reset.

The boot swap function is provided to avoid this problem.

Before boot cluster 0[Note], which is a boot program area, is rewritten during self programming, a new boot program is written to boot cluster 1. Once write to boot cluster 1 ends normally, boot cluster 1 and boot cluster 0 are swapped with the firmware's set information function, and boot cluster 1 becomes the boot area.

As a result, even if an instantaneous power supply interruption occurs during boot programming area rewrite, the next reset start performs booting from swap target boot cluster 1, so that normal program operation is achieved. Then erase or write is performed, if necessary, for boot cluster 0, which is the original boot program area.

Note A boot cluster is a 4 KB area, and boot cluster 0 and boot cluster 1 are swapped through boot swap.
Boot cluster 0 (0000H to 0FFFH): Original boot program area
Boot cluster 1 (1000H to 1FFFH): Boot swap target area

Figures 3-15 and 3-16 show boot swap execution examples.

Figure 3-15. Boot Swap Execution Example 1

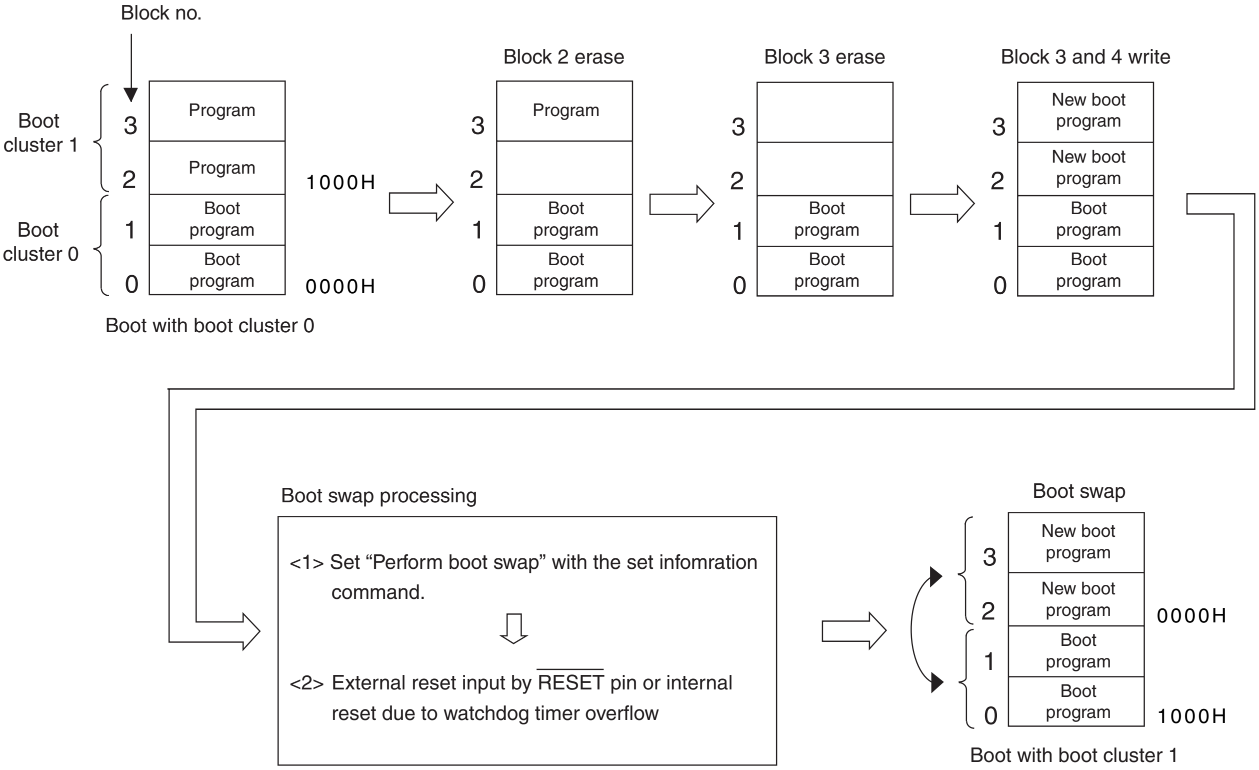
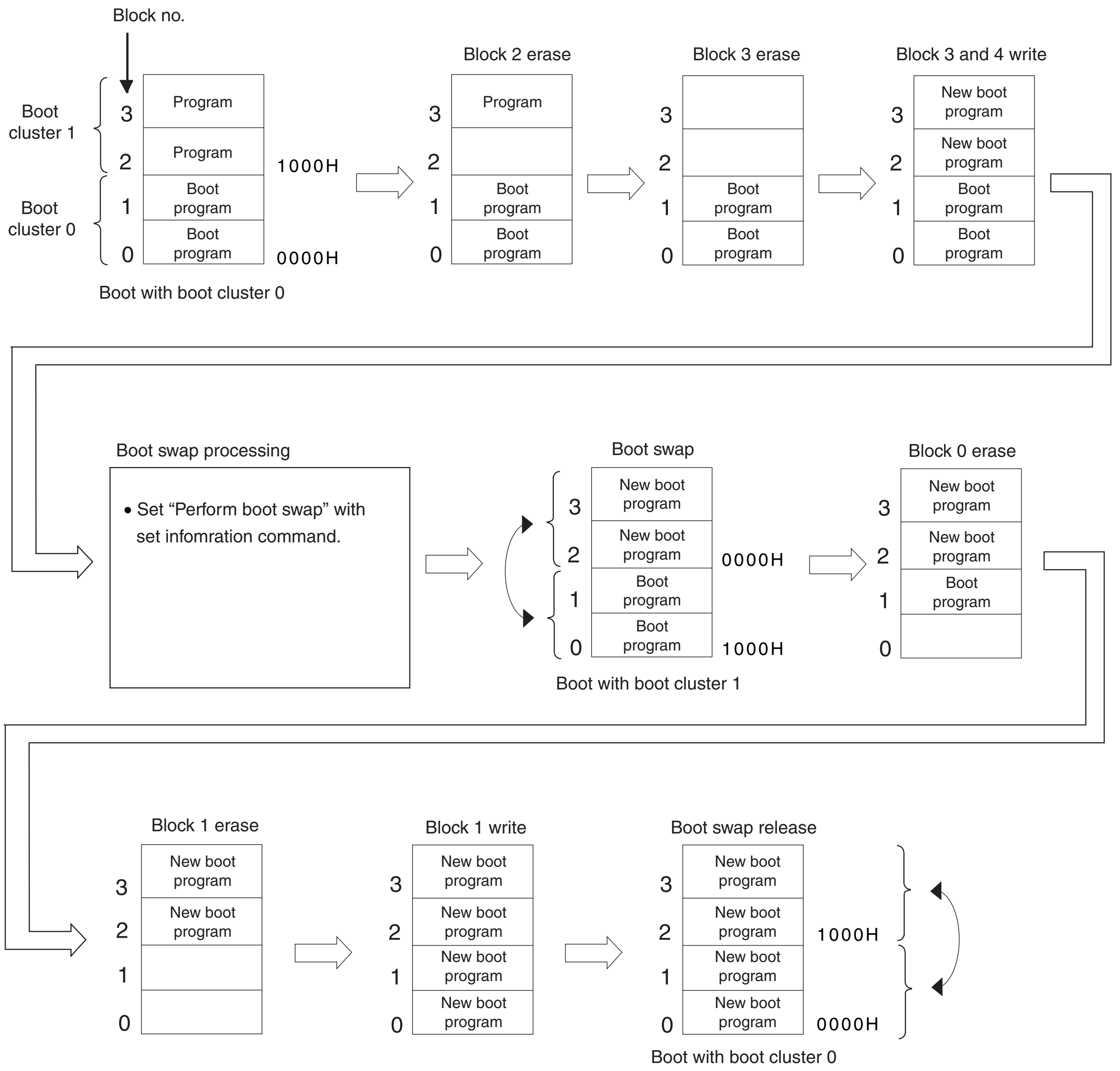


Figure 3-16. Boot Swap Execution Example 2



Remark In the boot swap of the execution example 2, the boot area can be changed without a reset.

CHAPTER 4 CC78K0 SELF-WRITE EXPANSION FUNCTION

The CC78K0 has self-write subroutine direct call functions in the firmware.

Flash memory control firmware can be called by using the __hromcall function.

The __hromcall function temporarily switches the register bank to bank 3, sets function numbers to C register and entry RAM addresses to HL register and calls the specified address. The value of B register is the return value.

unsigned char__hromcall (unsigned int entryaddr, unsigned char funcno, void *entrydata);

[#pragma hromcall] must be described.

Register bank 3 is temporarily switched to, entrydata is set to HL register, funcno to C register and the entryaddr address is called.

The value of B register is the return value.

Only constants can be specified for the first argument and second argument.

Example Executing firmware with function number 0x3 (CALL #8100H)

[C Source]

```
#pragma hromcall
unsigned char entrydata[48];
unsigned char ret;
_ _hromcall(0x8100,0x03,entrydata);
```

[Output Assembler]

```
push    psw
sel     rb3
movw    hl,#_entrydata
mov     c,#03H
call    !08100H
pop     psw
mov     a,0FEE3H
mov     !_ret,a
```

When creating a function corresponding to a function number, describing as follows using #define is possible.

Note that in the case of the CC78K0 C compiler, functions with the same names as in the following example are supported, but if used as is, the function numbers may differ.

Example

```
#define _ _FlashEnv(entrydata_addr)               _ _hromcall(0x8100,0x00,entrydata_addr)
#define _ _FlashBlockErase(entrydata_addr)        _ _hromcall(0x8100,0x03,entrydata_addr)
#define _ _FlashWordWrite(entrydata_addr)         _ _hromcall(0x8100,0x04,entrydata_addr)
#define _ _FlashBlockIVerify(entrydata_addr)      _ _hromcall(0x8100,0x06,entrydata_addr)
#define _ _FlashBlockBlankCheck(entrydata_addr)   _ _hromcall(0x8100,0x08,entrydata_addr)
#define _ _FlashGetInfo(entrydata_addr)           _ _hromcall(0x8100,0x09,entrydata_addr)
#define _ _FlashSetInfo(entrydata_addr)           _ _hromcall(0x8100,0x0a,entrydata_addr)
#define _ _FlashCheckFLMD(entrydata_addr)         _ _hromcall(0x8100,0x0e,entrydata_addr)
```

For further information, please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

Shanghai Branch
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai, P.R.China P.C:200120
Tel:021-5888-5400
http://www.cn.necel.com/

Shenzhen Branch
Unit 01, 39/F, Excellence Times Square Building,
No. 4068 Yi Tian Road, Futian District, Shenzhen,
P.R.China P.C:518048
Tel:0755-8282-9800
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/