Date: Nov. 17, 2014

RENESAS TECHNICAL UPDATE

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan
Renesas Electronics Corporation

Product Category	MPU & MCU		Document No.	TN-RX*-A109A/E	Rev.	1.00
Title	Corrections to Descriptions for the Flash Memory in the RX111 Group User's Manual		Information Category	Technical Notification		
Applicable Product	RX111 Group	Lot No. All	Reference Document	RX111 Group User's Manual: Hardware Rev.1.10 (R01UH0365EJ0110) Specification Changes to Improve the Products in the RX111 Group (TN-RX*-A110A/E)		

This document describes corrections to 35. Flash Memory in RX111 Group User's Manual: Hardware Rev.1.10.

- Page 1095 of 1243

The Software commands column in Table 35.1 Flash Memory Specifications is corrected as follows:

Before correction

Table 35.1 Flash Memory Specifications

Item	Description
Memory space	• User area: Up to 128 Kbytes • Data area: 8 Kbytes
Software commands	• The following commands can be executed in boot mode or during self-programming: blank check, block erase, program, read, set access window • Checksum can be also executed in boot mode. Suspend/resume can be also executed during self-programming.
omitted	

After correction

Table 35.1 Flash Memory Specifications

Item	Description
Memory space	• User area: Up to 128 Kbytes • Data area: 8 Kbytes • Extra area: Stores the start-up area information, access window information, and unique ID
Software commands	• The following commands are implemented: program, blank check, block erase, unique ID read • The following commands are implemented for programming the extra area: start-up area information program, access window information program
omitted	

• Page 1099 of 1243

Description in 35.4.2 Flash P/E Mode Entry Register (FENTRYR) is corrected as follows:

Before correction

To rewrite the ROM, either the FENTRYD or FENTRY0 bit must be set to 1 to place the ROM in P/E mode.

After correction

To rewrite the ROM, either the FENTRYD or FENTRY0 bit must be set to 1 to place the ROM in P/E mode.

• Page 1099 of 1243

Descriptions of clearing conditions for bits FENTRY0 and FENTRYD in 35.4.2 Flash P/E Mode Entry Register (FENTRYR) are corrected as follows:

Before correction

- Data is written by byte access.
- The FEKEY[7:0] bits are set to a value other than AAh and data is written to the FENTRYR register.
- AA00h is written to the FENTRYR register.
- Data is written to the FENTRYR register while the FENTRYR register is a value other than 0000h.

After correction

- AA00h is written to the FENTRYR register.

• Page 1105 of 1243

Descriptions for block erase in 35.4.9 Flash Control Register (FCR) are corrected as follows:

Before correction

- Erase a block of the flash memory.
  Set the start address of the target erasure block in the FSARH and FSARL registers, and set the end address of the target erasure block in the FEARH and FEARL registers. If a setting other than the above is made, erasure may not be executed correctly.

After correction

- Erase a specified block in the flash memory.
  Set the beginning address of the block to be erased in registers FSARH and FSARL, and the last address in registers FEARH and FEARL. If a value other than the above is set, erasure may not be executed correctly.

• Page 1107 of 1243

Descriptions for the access window information program in 35.4.10 Flash Extra Area Control Register (FEXCR) are corrected as follows:

Before correction

This command is used to set the access window used for area protection.
Set the access window in block units.
Set bit 19 to bit 10 of the access window start address in the FWBH register, set bit 19 to bit 10 of the access window end address + 1, and execute this command.
If the start address and end address are set to the same value, all areas can be accessed. Do not set the start address to a value larger than the value of the end address.

After correction

This command is used to set the access window used for area protection.
Set the access window in block units.
Specify the access window start address, which is the beginning address of the access window in the FWBL register, specify the access window end address, which is the next address of the last address of the access window in the FWBH register, and issue this command. Set bit 19 to bit 10 of the address for programming/erasure in each register.
If the same value is set as the start address and end address, all areas can be accessed. Do not set the start address to a value larger than the value of the end address.

• Pages 1107, 1108 of 1243

Descriptions in 35.4.11 Flash Processing Start Address Register H (FSARH) and 35.4.13 Flash Processing End Address Register H (FEARH) are corrected as follows:

Before correction

– omitted –
Set bit 19 to bit 16 of the flash memory address in this register.
– omitted –

After correction

– omitted –
Set bit 19 to bit 16 of the flash memory address for programming/erasure in this register.
– omitted –

• Page 1108 of 1243

Description in 35.4.12 Flash Processing Start Address Register L (FSARL) is corrected as follows:

Before correction

– omitted –
Set bit 15 to bit 0 of the address in this register.
– omitted –

After correction

– omitted –
Set bit 15 to bit 0 of the flash memory address for programming/erasure in this register.
– omitted –

• Page 1109 of 1243

Description in 35.4.14 Flash Processing End Address Register L (FEARL) is corrected as follows:

Before correction

– omitted –

Set bit 15 to bit 0 of the flash memory address in this register.

– omitted –

After correction

– omitted –

Set bit 15 to bit 0 of the flash memory address for programming/erasure in this register.

– omitted –

• Page 1111 of 1243

Descriptions for the reserved bits in 35.4.19 Flash Status Register 0 (FSTATR0) are corrected as follows:

Before correction

bit	symbol	Bit Name	Description	R/W
			omitted	
b2	—	Reserved	This bit is read as 0.	R
			omitted	
b6	—	Reserved	This bit is read as 0.	R
b7	—	Reserved	The read value is undefined.	R

After correction

bit	symbol	Bit Name	Description	R/W
			omitted	
b2	—	Reserved	The read value is undefined.	R
			omitted	
b7, b6	—	Reserved	The read value is undefined.	R

• Page 1111 of 1243

The following description for the PRGERR flag in 35.4.19 Flash Status Register 0 (FSTATR0) is deleted:

The value read from this flag is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during erasure.

• Page 1111, 1112 of 1243

The following description is added to the BCERR flag in 35.4.19 Flash Status Register 0 (FSTATR0):

The value read from this flag is undefined when setting the FCR.STOP bit to 1 (processing is forcibly stopped) during blank check.

• Page 1113 of 1243

Description for the FRDY flag in 35.4.20 Flash Status Register 1 (FSTATR1) is corrected as follows:

Before correction

This flag is used to confirm whether a software command is executed.
This flag is set to 1 when processing of the executed software command is completed, and 0 when the FCR.OPST bit is set to 0.

After correction

This flag is used to confirm whether a software command is executed.
This flag becomes 1 when processing of the executed software command or the forced stop processing is completed, and this flag becomes 0 when setting the FCR.OPST bit to 0.

• Page 1114 of 1243

Descriptions in 35.4.21 Flash Error Address Monitor Register H (FEAMH) are corrected as follows:

Before correction

This register stores the flash memory address where the error has occurred if an error occurs during execution of a software command (program, block erase, or verify). If a blank check error occurs when a blank check command is executed, this register stores the flash memory address where programming has been executed.
– omitted –

After correction

This register is used to check the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 19 to bit 16 of the address where the error has occurred for the program command or blank check command, or it stores bit 19 to bit 16 of the beginning address of the area where the error has occurred for the block erase command.
Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing.
– omitted –

• Page 1114 of 1243

Descriptions in 35.4.22 Flash Error Address Monitor Register L (FEAML) are corrected as follows:

Before correction

This register is used to confirm the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 15 to bit 0 of the flash memory address where the error has occurred or bit 15 to bit 0 of the start address in the flash memory area where the error has occurred.
When the software command terminates normally, this register stores bit 15 to bit 0 of the end address at execution of the command.
To set the ROM area, set bit 1 and bit 0 to 00b.
When a software command for the ROM is executed, lower-order 2 bits are set to 00b.
Refer to Figure 35.1 and Figure 35.2 for details on the addresses of the flash memory.

After correction

This register is used to check the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 15 to bit 0 of the address where the error has occurred for the program command or blank check command, or it stores bit 15 to bit 0 of the beginning address of the area where the error has occurred for the block erase command.
Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing.
When the software command is normally completed, this register stores bit 15 to bit 0 of the last address at execution of the command.
When executing a software command for the ROM or the unique ID read command, low-order 2 bits become 00b.
Refer to Figure 35.1 and Figure 35.2 for details on the addresses of the flash memory.

• Page 1115 of 1243

Note 1 in 35.4.24 Flash Access Window Start Address Monitor Register (FAWSMR) is corrected as follows:

Before correction

Note 1. The value of the blank product is 1. It is set to the same value set in bit 9 to bit 0 in the FWBH register after the access window information program command is executed.

After correction

Note 1. The value of the blank product is 1. It is set to the same value set in bit 9 to bit 0 in the FWBL register after the access window information program command is executed.

• Page 1116 of 1243

Note 1 in 35.4.25 Flash Access Window End Address Monitor Register (FAWEMR) is corrected as follows:

Before correction

Note 1. The value of the blank product is 1. It is set to the same value set in bit 9 to bit 0 in the FWBL register after the access window information program command is executed.

After correction

Note 1. The value of the blank product is 1. It is set to the same value set in bit 9 to bit 0 in the FWBH register after the access window information program command is executed.

• Page 1117 of 1243

(2) of Figure 35.3 in 35.5 Start-Up Program Protection is corrected as follows:

Before correction

(2) After the alternate area is successfully rewritten, the default area and the alternate area are switched using the self-programming library. After that, the program in the alternate area starts after a reset.

After correction

(2) After the alternate area is successfully rewritten, the default area and the alternate area are switched using the start-up area information program command. After that, the program in the alternate area starts after a reset.

• Page 1118 of 1243

Description in 35.6 Area Protection and Figure 35.4 are changed as follows:

Before correction

Select the start block and end block to set the access window. While the access window can be set in boot mode or by self-programming, area protection is enabled only during self-programming in single-chip mode.
Figure 35.4 shows the Area Protection Overview.

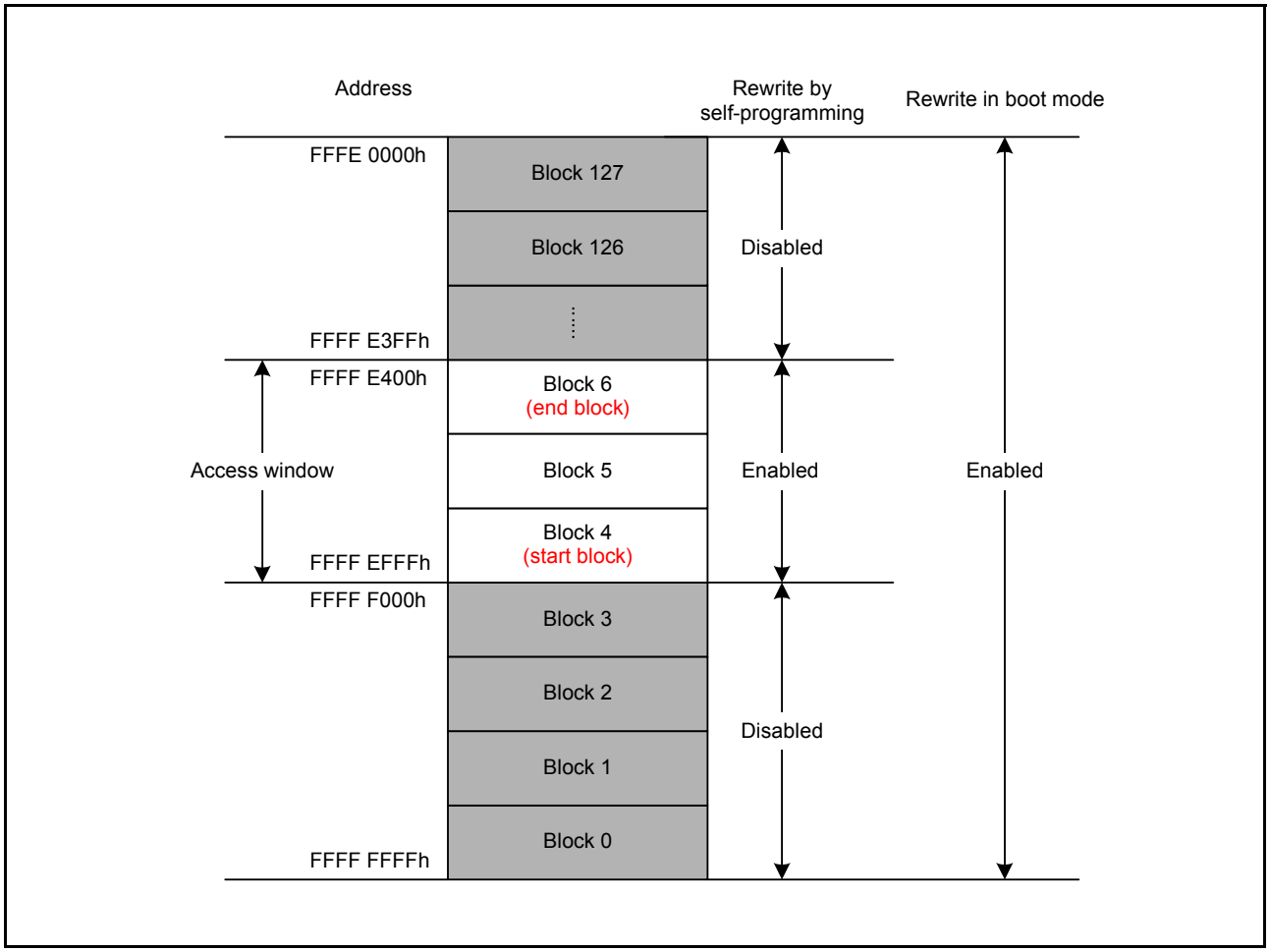


Figure 35.4 Area Protection Overview

After correction

Specify the start address and end address to set the access window. While the access window can be set in boot mode or by self-programming, area protection is enabled only during self-programming in single-chip mode.
Figure 35.4 shows the Area Protection Overview.

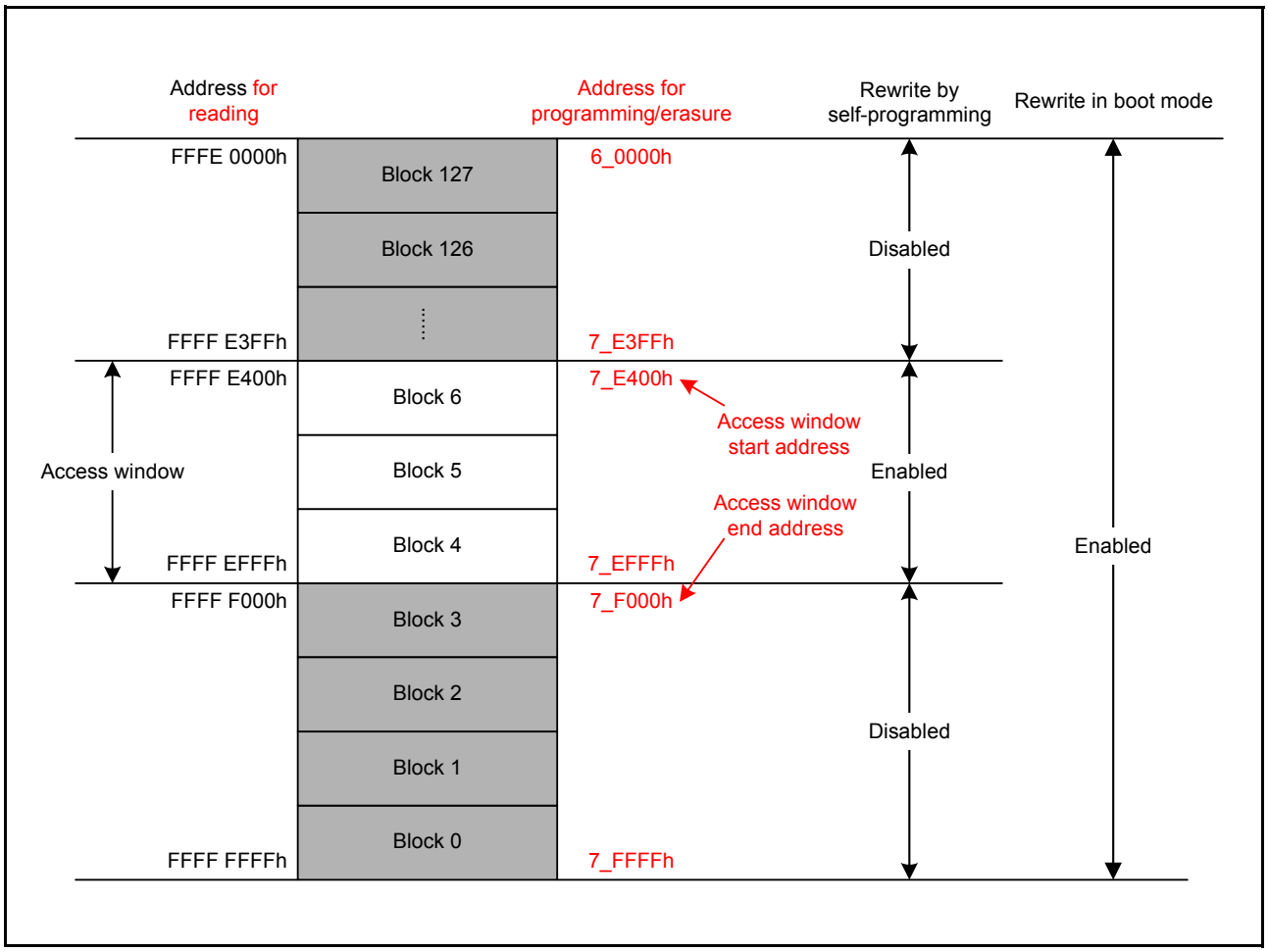


Figure 35.4 Area Protection Overview (Set Blocks 4 to 6 as the Access Window)

• Page 1119, 1120 of 1243

Description in 35.7.1 Sequencer Modes and Figure 35.5 are changed as follows:

Before correction

The sequence has three modes. Transitions between modes are caused by writing to the FENTRYR register or issuing commands. Figure 35.5 is a diagram of mode transitions of the flash memory.

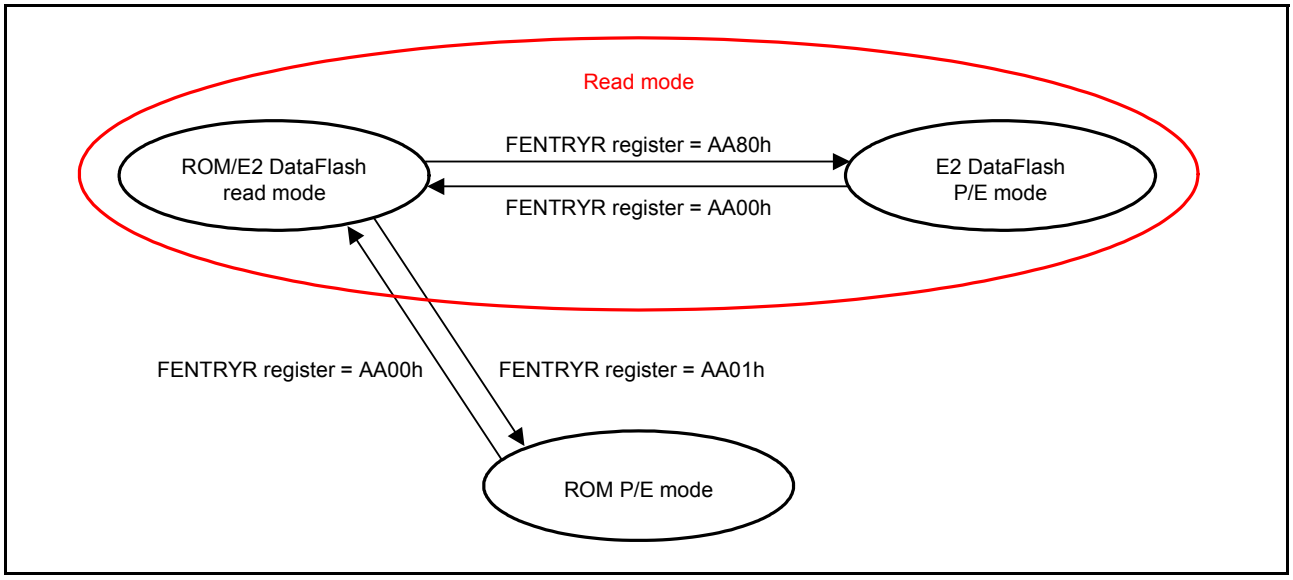


Figure 35.5 Mode Transitions of the Flash Memory

35.7.1.1 Read Mode

Read mode is for high-speed reading of the ROM/E2 DataFlash. Reading from a ROM address for reading can be accomplished in one ICLK clock.
There is one read mode: ROM/E2 DataFlash read mode.

(1) ROM/E2 DataFlash Read Mode
This mode is for reading the ROM and E2 DataFlash. The sequencer enters this mode when the FENTRYR.FENTRY0 bit is set to 0 with the FENTRYR.FENTRYD bit set to 0.
When accessing the E2 DataFlash, also set the DFLCTL.DFLEN bit to 1.

35.7.1.2 ROM P/E Modes

There are two P/E modes: ROM P/E mode and E2 DataFlash P/E mode.

(1) ROM P/E Mode
The ROM P/E mode is for programming and erasure of the ROM. The sequencer enters this mode when the FENTRYR.FENTRYD bit is set to 0 with the FENTRYR.FENTRY0 bit set to 1.

(2) E2 DataFlash P/E Mode
The E2 DataFlash P/E Mode is for programming and erasure of the E2 DataFlash. High-speed reading from the ROM is possible. The sequencer enters this mode when the FENTRYR.FENTRY0 bit is set to 0 with the FENTRYR.FENTRYD bit set to 1.
When accessing the E2 DataFlash, also set the DFLCTL.DFLEN bit to 1.

After correction

The sequencer has 4 modes. Transitions between modes are caused by writing to the DFLCTL and FENTRYR registers and setting the FPMCR register. Figure 35.5 is a diagram of mode transitions of the flash memory.

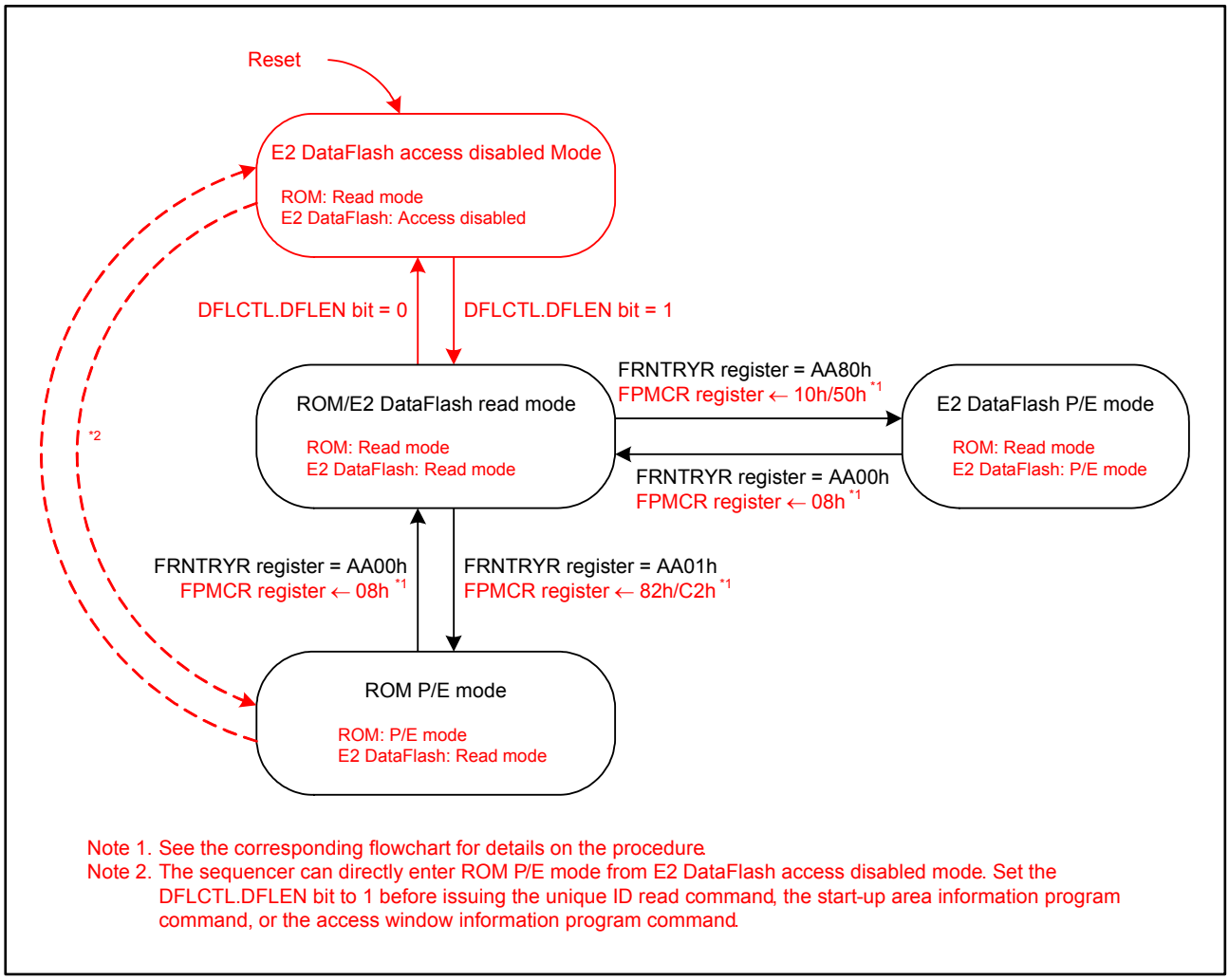


Figure 35.5 Mode Transitions of the Flash Memory

35.7.1.1 E2 DataFlash Access Disabled Mode

In E2 DataFlash access dsiabled mode, access to the E2 DataFlash is disabled. After a reset, the sequencer enters this mode.

When setting the DFLCTL.DFLEN bit to 1, the E2 DataFlash is placed in read mode.

35.7.1.2 Read Mode

Read mode is for high-speed reading of the ROM/E2 DataFlash. Reading from a ROM address for reading can be accomplished in one ICLK clock.

(1) ROM/E2 DataFlash Read Mode

In this mode, both the ROM and E2 DataFlash are in read mode. The sequencer enters this mode from P/E mode when setting the FPMCR register to 08h, setting the FENTRYR.FENTRYD bit to 0, and setting the FENTRYR.FENTRY0 bit to 0.

35.7.1.3 P/E Modes

The P/E mode is for programming and erasure of the ROM/E2 DataFlash.

(1) ROM P/E Mode

In this mode, the ROM is in P/E mode, and the E2 DataFlash is in read mode. The sequencer enters this mode when the setting the FENTRYR.FENTRYD to 0, setting the FENTRYR.FENTRY0 bit to 1, and setting the FPMCR register 82h or C2h.

(2) E2 DataFlash P/E Mode

In this mode, the ROM is in read mode, and the E2 DataFlash is in P/E mode. The sequencer enters this mode when the setting the FENTRYR.FENTRYD to 1, setting the FENTRYR.FENTRY0 bit to 0, and setting the FPMCR register 10h or 50h.

• Page 1121 of 1243

Figure 35.6 in (1) Switching to ROM P/E Mode of 35.7.3 Software Command Usage is corrected as follows:

Before correction

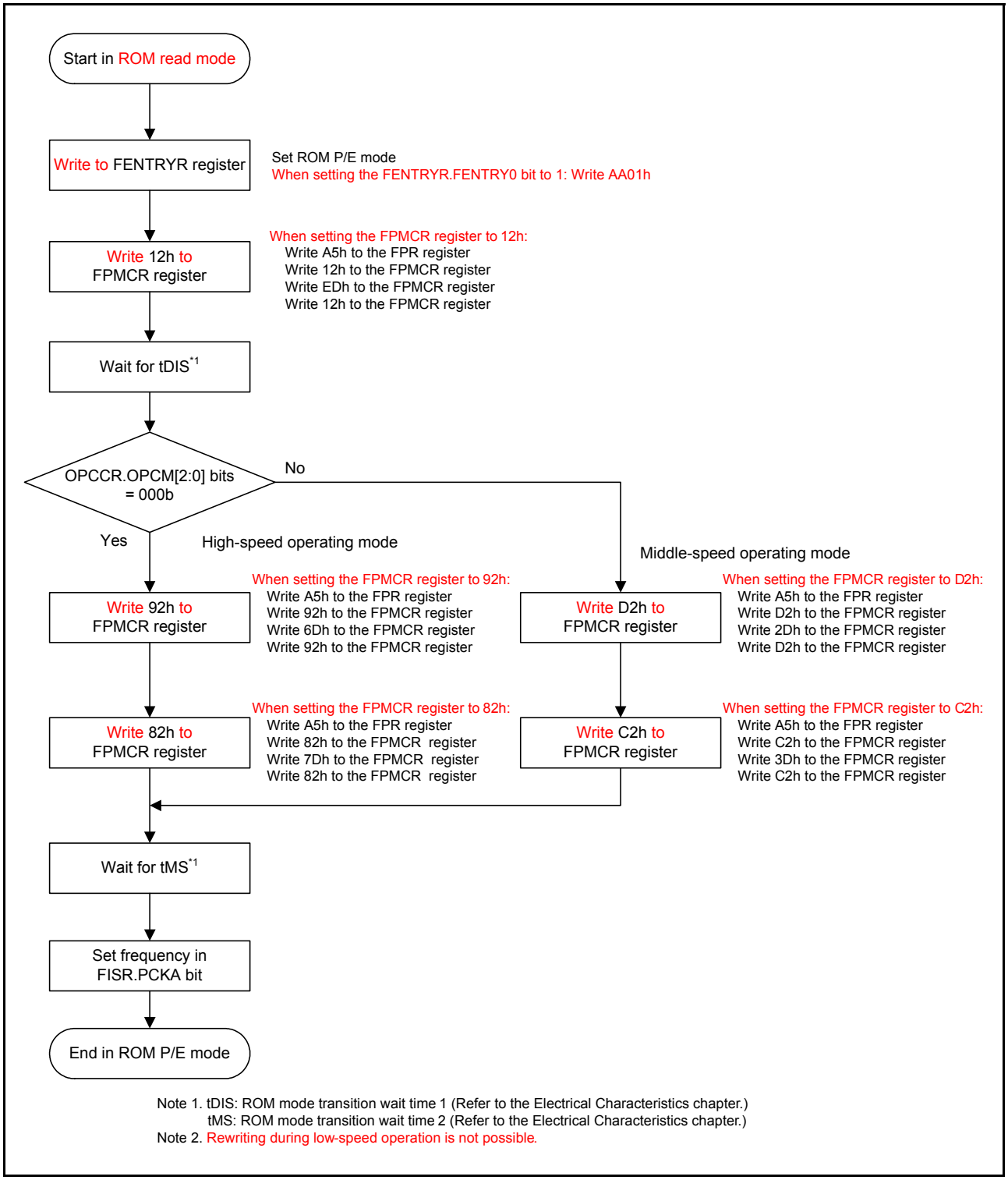
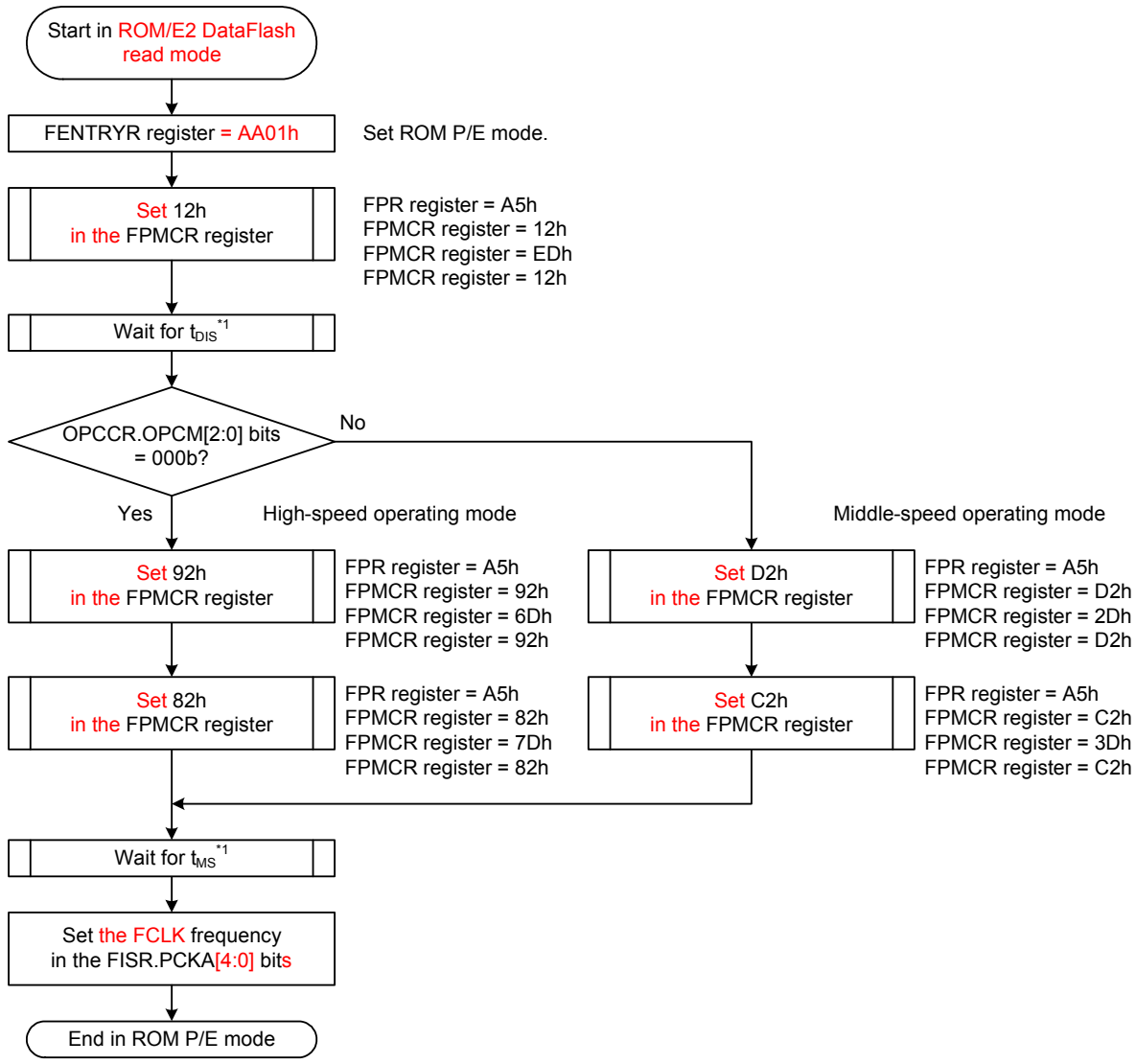


Figure 35.6 Procedure for Transition to ROM P/E Mode

After correction



Note 1. t_{DIS}: ROM mode transition wait time 1 (Refer to the Electrical Characteristics chapter.)
t_{MS}: ROM mode transition wait time 2 (Refer to the Electrical Characteristics chapter.)
Note 2. The ROM cannot be rewritten in low-speed operating mode.

Figure 35.6 Procedure for Transition from ROM/E2 DataFlash Read Mode to ROM P/E Mode

• Page 1122 of 1243

Figure 35.7 in (1) Switching to ROM P/E Mode of 35.7.3 Software Command Usage is corrected as follows:

Before correction

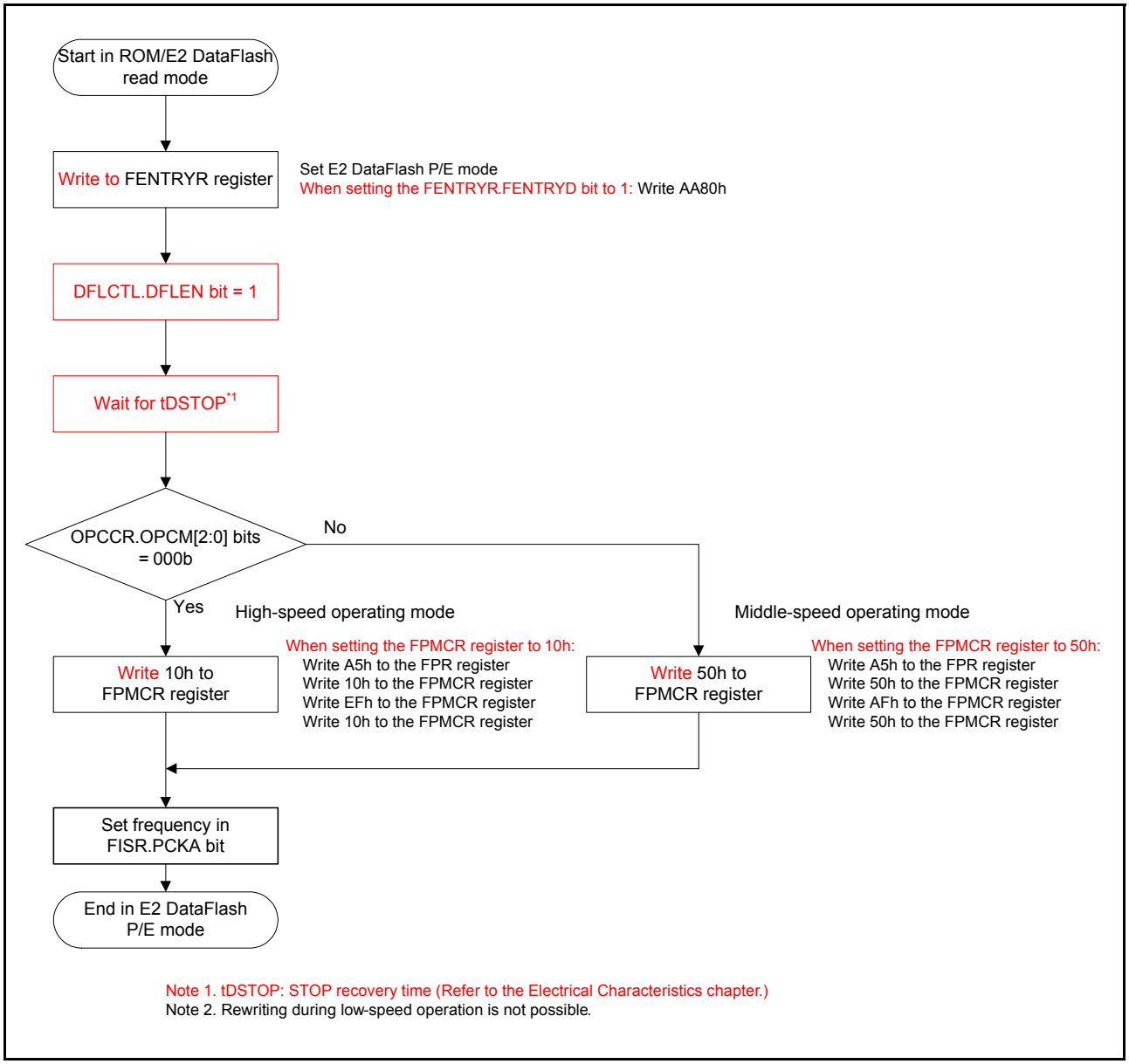
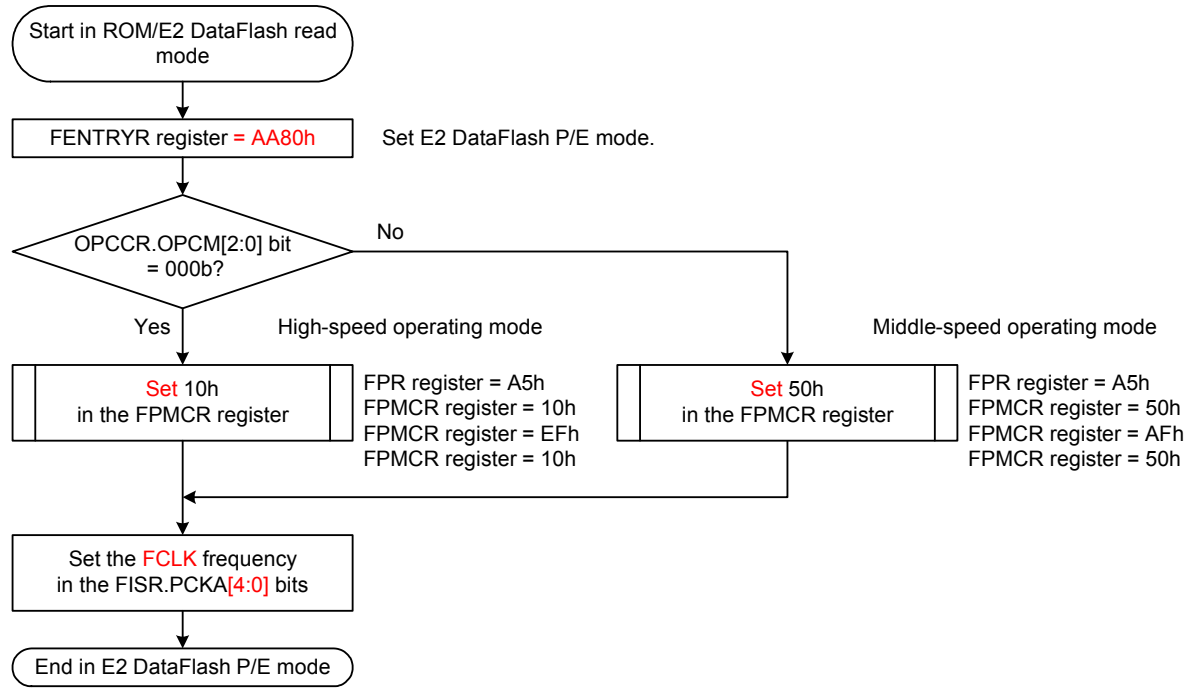


Figure 35.7 Procedure for Transition from ROM/E2 DataFlash Read Mode to E2 DataFlash P/E Mode

After correction



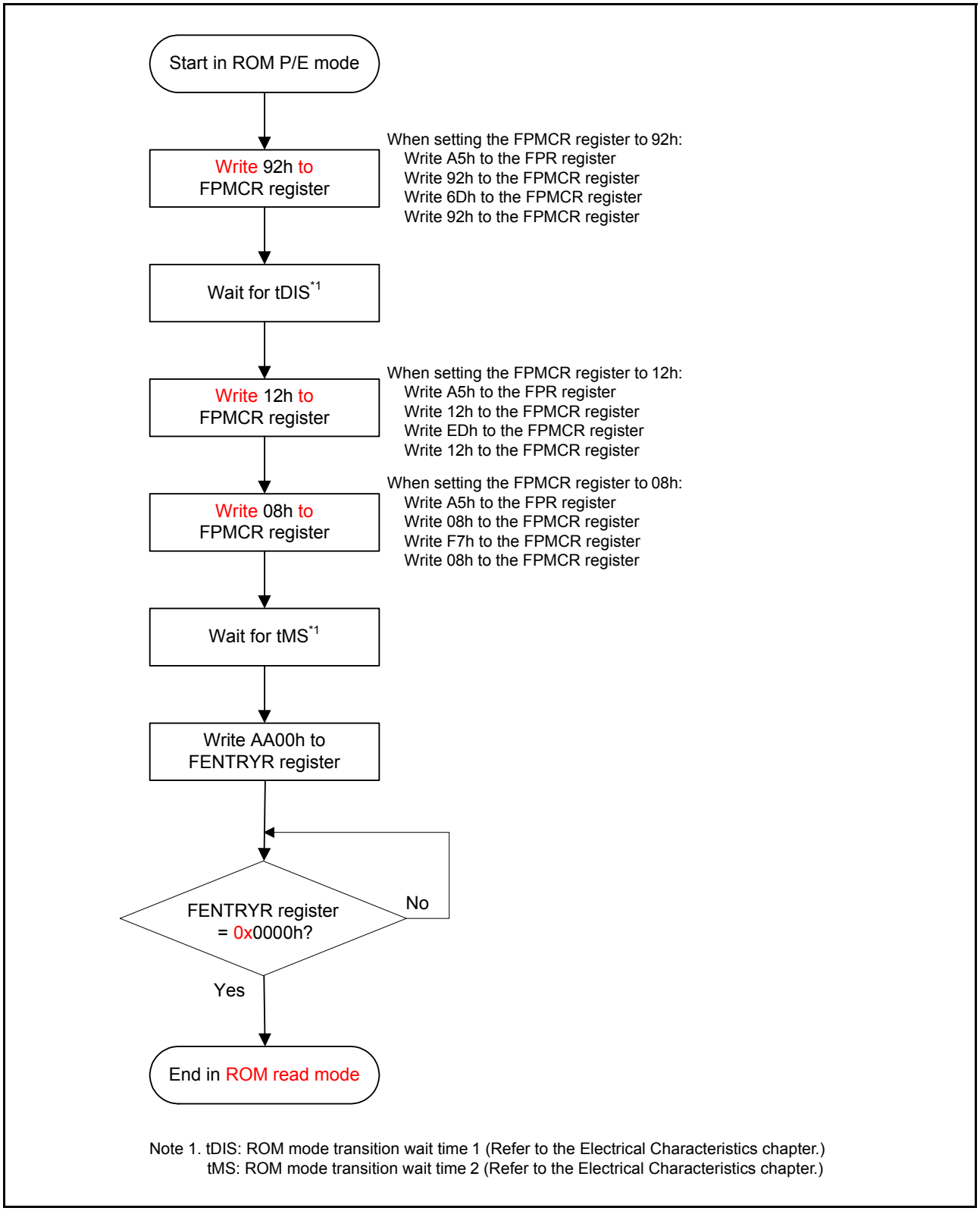
Note 1. The E2 DataFlash cannot be rewritten in low-speed operating mode.

Figure 35.7 Procedure for Transition from ROM/E2 DataFlash Read Mode to E2 DataFlash P/E Mode

• Page 1123 of 1243

Figure 35.8 in (2) Switching to ROM Read Mode of 35.7.3 Software Command Usage is corrected as follows:

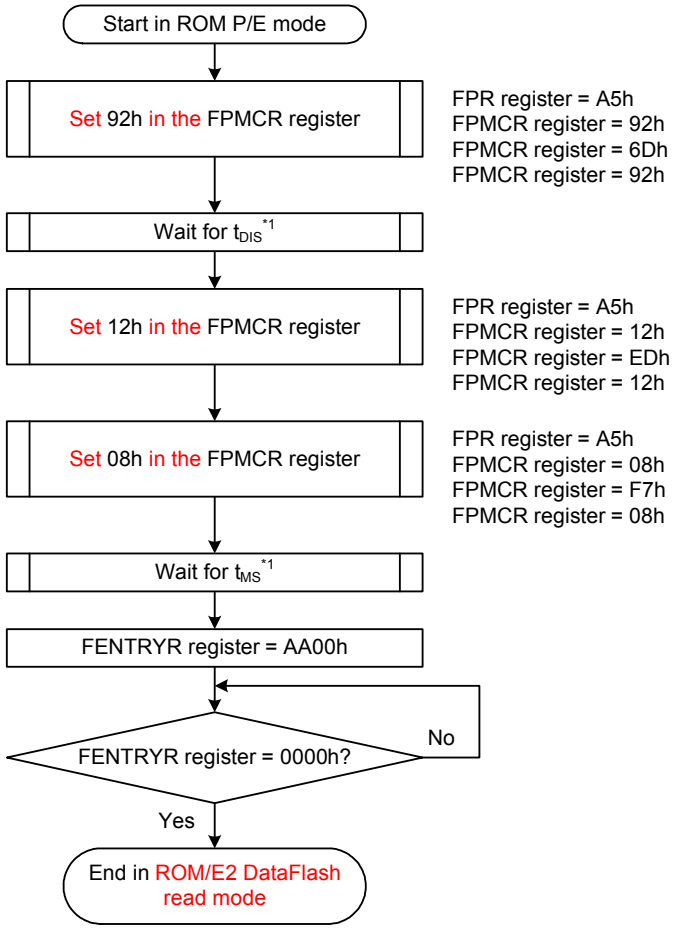
Before correction



Note 1. tDIS: ROM mode transition wait time 1 (Refer to the Electrical Characteristics chapter.)
tMS: ROM mode transition wait time 2 (Refer to the Electrical Characteristics chapter.)

Figure 35.8 Procedure for Transition to ROM Read Mode

After correction



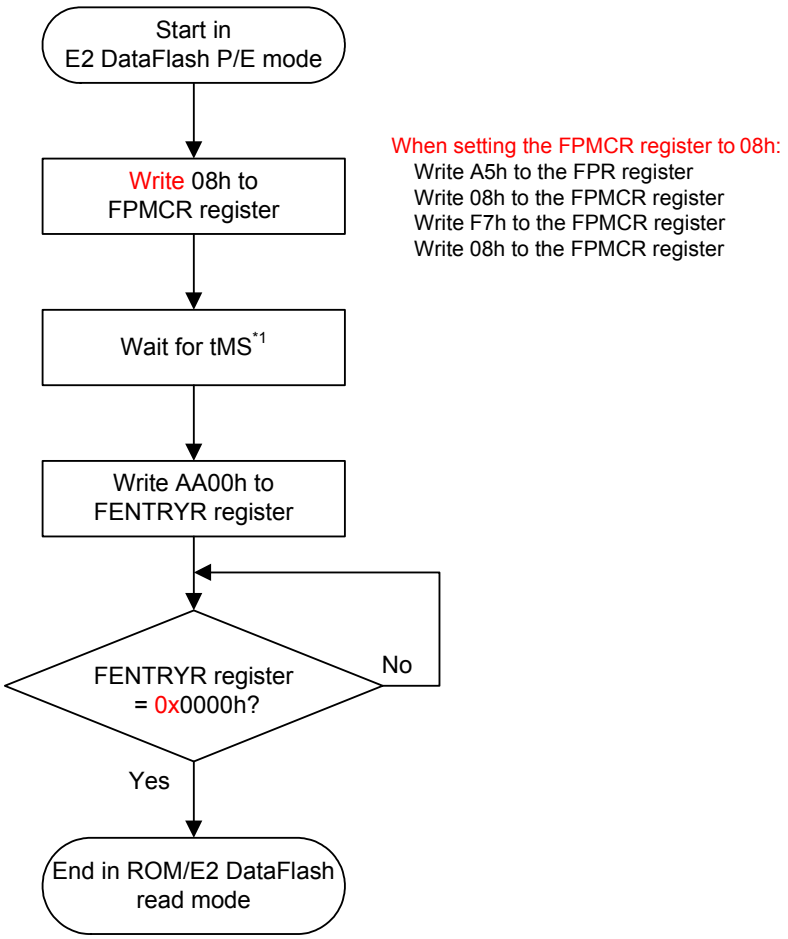
Note 1. t_{DIS}: ROM mode transition wait time 1 (Refer to the Electrical Characteristics chapter.)
t_{MS}: ROM mode transition wait time 2 (Refer to the Electrical Characteristics chapter.)

Figure 35.8 Procedure for Transition from ROM P/E Mode to ROM/E2 DataFlash Read Mode

• Page 1124 of 1243

Figure 35.9 in (2) Switching to ROM Read Mode of 35.7.3 Software Command Usage is corrected as follows:

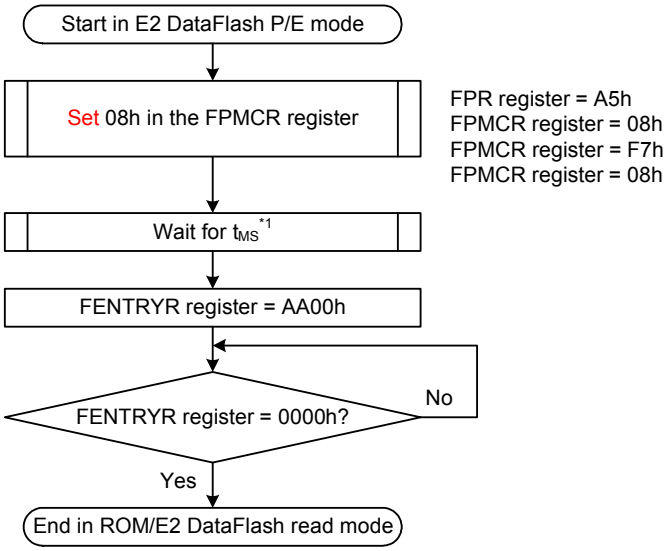
Before correction



Note 1. tMS: ROM mode transition wait time 2 (Refer to the Electrical Characteristics chapter.)

Figure 35.9 Procedure for Transition from E2 DataFlash P/E Mode to ROM/E2 DataFlash Read Mode

After correction



Note 1. t_{MS}: ROM mode transition wait time 2 (Refer to the Electrical Characteristics chapter.)

Figure 35.9 Procedure for Transition from E2 DataFlash P/E Mode to ROM/E2 DataFlash Read Mode

• Page 1119 to 1132 of 1243

The structure of sections in 35.7 Programming and Erasure are changed as follows:

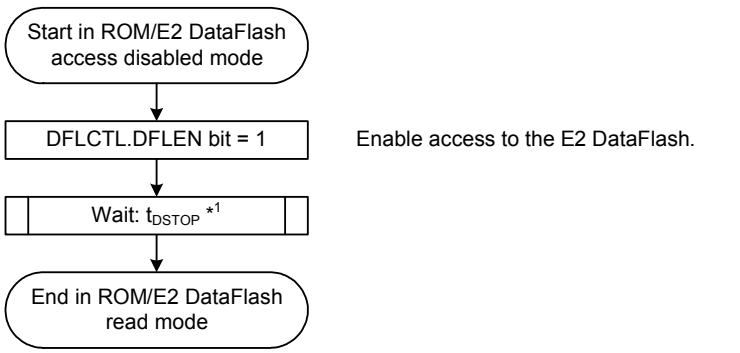
Before correction

35.7 Programming and Erasure
35.7.1 Sequencer Modes
35.7.1.1 Read Mode
35.7.1.2 ROM P/E Modes
35.7.2 Software Commands
35.7.3 Software Command Usage
(1) Switching to ROM P/E Mode
(2) Switching to ROM Read Mode
(3) Programming and Erasure Procedures
(4) Start-Up Area Information Program/Access Window Information Program
(5) Consecutive Read

After correction

35.7 Programming and Erasure
35.7.1 Sequencer Modes
35.7.1.1 E2 DataFlash Access Disabled Mode
35.7.1.2 Read Mode
35.7.1.3 P/E Modes
35.7.2 Mode Transitions
35.7.2.1 Transition from E2 DataFlash Access Disable Mode to Read Mode
35.7.2.2 Transition from Read Mode to P/E Mode
35.7.2.3 Transition from P/E Mode to Read Mode
35.7.3 Software Commands
35.7.4 Software Command Usage
35.7.4.1 Program
35.7.4.2 Block Erase
35.7.4.3 Blank Check
35.7.4.4 Start-Up Area Information Program/Access Window Information Program
35.7.4.5 Unique ID Read
35.7.4.6 Forced Stop of Software Commands

The following flowchart is added to 35.7.2.1 Transition from E2 DataFlash Access Disable Mode to Read Mode.



Note 1. t_{DSTOP}: STOP recovery time (refer to the 36. Electrical Characteristics)

Figure 35.x Transition from E2 DataFlash Access Disabled Mode to ROM/E2 DataFlash Read Mode

The following descriptions and flowchart are added to 35.7.4.6 Forced Stop of Software Commands.

35.7.4.6 Forced Stop of Software Commands

Perform the procedure shown in Figure 35.xx to forcibly stop the blank check command or block erase command. When the command processing is forcibly stopped, registers FEAMH and FEAML store the address at the time of the forced stop. For blank check, the stopped processing can be continued by copying the FEAMH and FEAML register values to registers FSARH and FSARL.

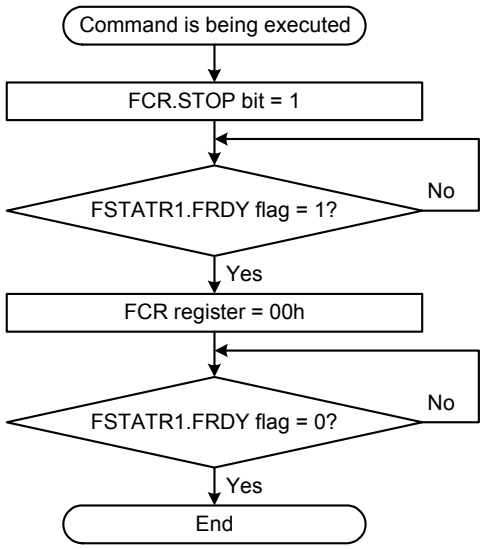


Figure 35.xx Procedure for Forced Stop of Software Commands

• Page 1152 of 1243

The section number of 35.10.6 Operating Frequency Select is corrected to 35.10.5.2. Accordingly, the subsequent section numbers are corrected as follows:

Before correction

35.10.6 Operating Frequency Select
35.10.6.1 Program/Erase Status Transition
35.10.7 ID Code Authentication Command
⋮

After correction

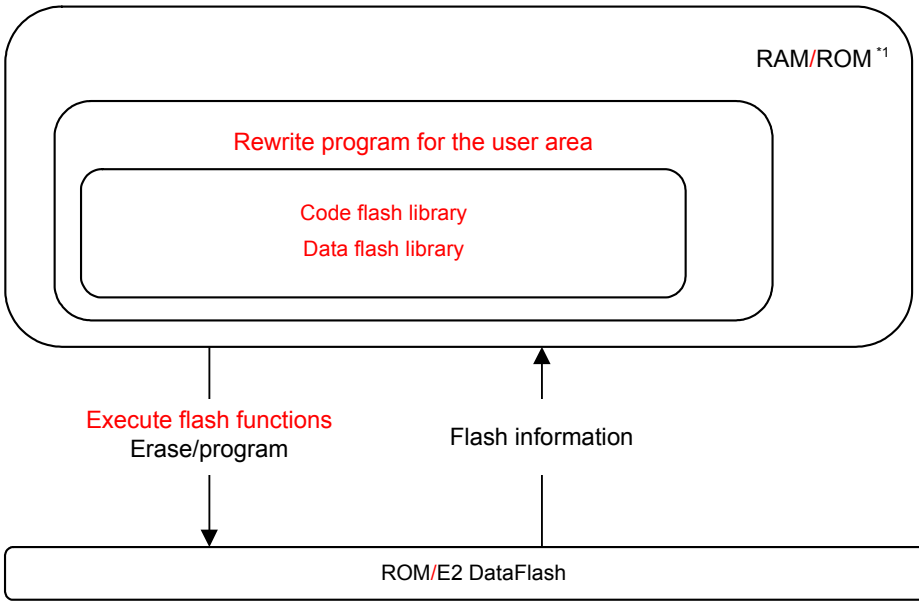
35.10.5.2 Operating Frequency Select
35.10.5.3 Program/Erase State Transition
35.10.6 ID Code Authentication Command
⋮

• Page 1175 of 1243

Descriptions in 35.11.1 Overview of 35.11 Rewriting by Self-Programming and Figure 35.38 are corrected as follows:

Before correction

The MCU supports rewriting the flash memory by the user program. Using the self-programming library provided from Renesas Electronics, the ROM and E2 DataFlash can be rewritten.
When rewriting the E2 DataFlash, the BGO can be used to execute the rewrite program on the ROM. The E2 DataFlash can also be rewritten by executing the rewrite program that is transferred on the RAM in advance.



Note 1: The ROM cannot be rewritten by executing the rewrite program on the ROM.

Figure 35.38 Self-Programming Overview

Refer to the user's manual of "Code Flash Libraries" and "Data Flash Libraries" for comprehensive information about flash self-programming.

After correction

The MCU supports rewriting the flash memory by the user program. The ROM and E2 DataFlash can be rewritten by preparing a routine to rewrite the flash memory (flash rewrite routine) in the user program.
When rewriting the E2 DataFlash, the BGO can be used to execute the flash rewrite routine on the ROM. The E2 DataFlash can also be rewritten by executing the flash rewrite routine that is transferred on the RAM in advance.

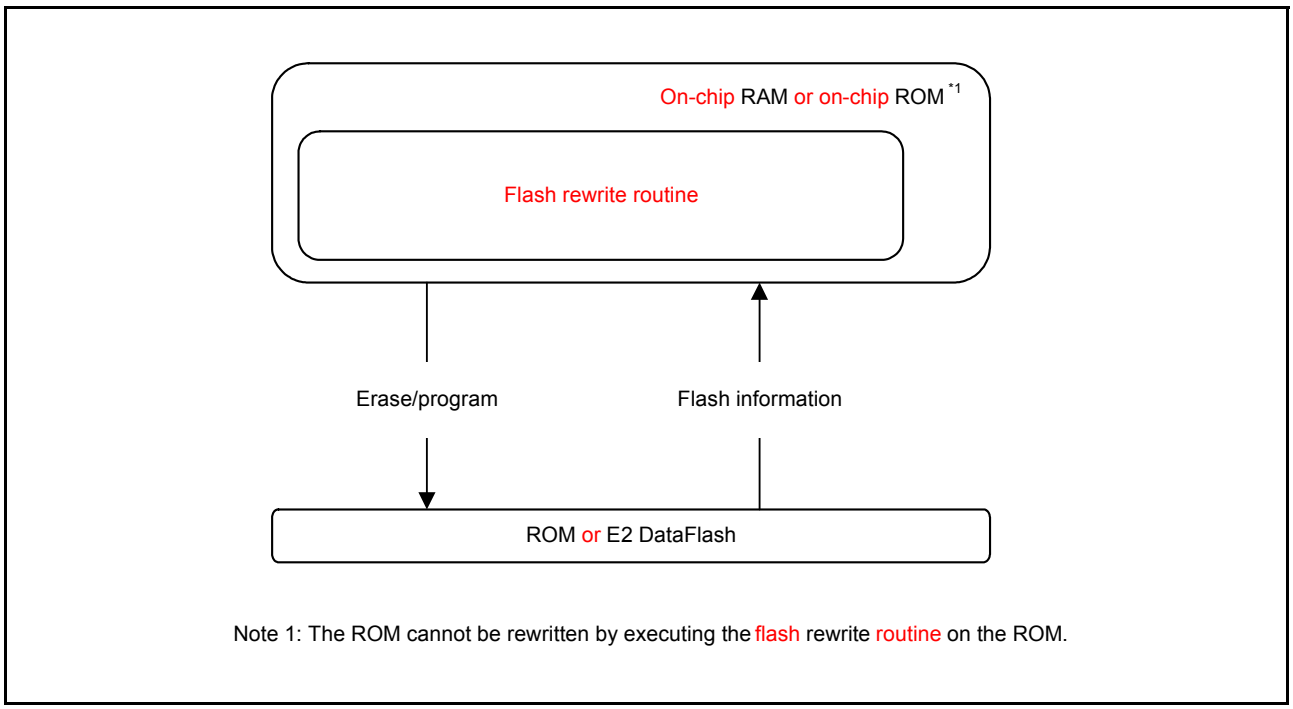


Figure 35.38 Self-Programming Overview

- Page 1176 of 1243

Descriptions in (1) and (2) of 35.12 Usage Notes are corrected as follows:

Before correction

(1) Erase Suspended Area
Data in areas where an erase operation is suspended is undefined. To avoid malfunctions caused by reading undefined data, do not execute commands and read data in the area where an erase operation is suspended.
(2) Suspension by Erase Suspend Commands
When suspending an erase operation by the erase suspend command, complete the operation by a resume command.

After correction

(1) Access the Block Where Erase Operation Is Forcibly Stopped
When forcibly stopping an erase operation, data in the block where the erase operation is aborted is undefined. To avoid malfunctions caused by reading undefined data, do not execute instructions or read data in the block where an erase operation is forcibly stopped.
(2) Processing After Forced Stop of Erase Operation
After forcibly stopping an erase operation, issue the block erase command to the same block again.