FM3 CY9AAA0N/1A0N/A30N/130N/130L Series, Flash Programming Specifications

Doc. No. 002-04750 Rev. *D

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
http://www.cypress.com

Copyrights

Preface

- Purpose of this document and intended readers

This document explains the functions, operations and serial programming of the flash memory of this series. This document is intended for engineers engaged in the actual development of products using this series.

- Organization of this document

This document consists of the following 3 chapters.

CHAPTER 1 Flash Memory

This chapter gives an overview of, and explains the structure, operation, and registers of the flash memory.

CHAPTER 2 Flash Security

The flash security feature provides possibilities to protect the content of the flash memory. This chapter section describes the overview and operations of the flash security.

CHAPTER 3 Serial Programming Connection

This chapter explains the basic configuration for serial write to flash memory by using the Cypress Serial Programmer.

- Sample programs and development environment

Cypress offers sample programs free of charge for using the peripheral functions of the FM3 family. Cypress also makes available descriptions of the development environment required for this series. Feel free to use them to verify the operational specifications and usage of this Cypress microcontroller.

- Microcontroller support information:

https://community.cypress.com/community/MCU

*: Note that the sample programs are subject to change without notice. Since they are offered as a way to demonstrate standard operations and usage, evaluate them sufficiently before running them on your system. Cypress assumes no responsibility for any damage that may occur as a result of using a sample program.

How to Use This Document

- Finding a function

The following methods can be used to search for the explanation of a desired function in this document:

- Search from the table of the contents

The table of the contents lists the document contents in the order of description.

- Search from the register

The address where each register is located is not described in the text. To verify the address of a register, see "APPENDIX Register Map" of "FM3 Peripheral Manual".

- Terminology

This document uses the following terminology.

Term	Explanation
Word	Indicates access in units of 32 bits.
Half word	Indicates access in units of 16 bits.
Byte	Indicates access in units of 8 bits.

- Notations

The notations in bit configuration of the register explanation of this document are written as follows.

bit : bit number
Field : bit field name
Attribute : Attributes for read and write of each bit
R : Read only
W : Write only
RW : Readable/Writable
- : Undefined
Initial value : Initial value of the register after reset
0 : Initial value is "0"
1 : Initial value is "1"
X : Initial value is undefined

- The multiple bits are written as follows in this document.
  Example : bit7:0 indicates the bits from bit7 to bit0

- The values such as for addresses are written as follows in this document.
  - ☐ Hexadecimal number: "0x" is attached in the beginning of a value as a prefix (example : 0xFFFF)
  - ☐ Binary number : "0b" is attached in the beginning of a value as a prefix (example: 0b1111)
  - ☐ Decimal number : Written using numbers only (example : 1000)

Table 1. Applicable Products

Type name*	Flash Memory Size	
	128 Kbyte	64 Kbyte
TYPE3	CY9AF132K CY9AF132L MB9AF132K MB9AF132L	CY9AF131K CY9AF131L MB9AF131K MB9AF131L
	CY9AF132KA CY9AF132LA MB9AF132KA MB9AF132LA	CY9AF131KA CY9AF131LA MB9AF131KA MB9AF131LA
	CY9AF132KB CY9AF132LB MB9AF132KB MB9AF132LB	CY9AF131KB CY9AF131LB MB9AF131KB MB9AF131LB
TYPE7	CY9AF132M CY9AF132N MB9AF132M MB9AF132N	CY9AF131M CY9AF131N MB9AF131M MB9AF131N
	CY9AFA32L CY9AFA32M CY9AFA32N MB9AFA32L MB9AFA32M MB9AFA32N	CY9AFA31L CY9AFA31M CY9AFA31N MB9AFA31L MB9AFA31M MB9AFA31N
TYPE7	CY9AF1A2L CY9AF1A2M CY9AF1A2N MB9AF1A2L MB9AF1A2M MB9AF1A2N	CY9AF1A1L CY9AF1A1M CY9AF1A1N MB9AF1A1L MB9AF1A1M MB9AF1A1N
	CY9AFAA2L CY9AFAA2M CY9AFAA2N MB9AFAA2L MB9AFAA2M MB9AFAA2N	CY9AFAA1L CY9AFAA1M CY9AFAA1N MB9AFAA1L MB9AFAA1M MB9AFAA1N

* : These type names are used to group applicable products in FM3 peripheral manual.

Contents

1. Flash Memory

This chapter gives an overview of, and explains the structure, operation, and registers of the flash memory. This series has built-in flash memory with a capacity of 64 KBytes to 128 KBytes that supports erasing by all-sector, erasing by unit sector, and writing by the CPU.

1. Overview
2. Configuration
3. Operating Description
4. Registers

1.1 Overview

This series is equipped with 64 KBytes to 128 KBytes of built-in flash memory.
The built-in flash memory can be erased by all sector and by unit of sector, and written by half words unit (16 bits) by Cortex-M3 CPU.

- Flash Memory Features
- Usable capacity:
  Minimum 64 Kbytes
  Maximum 128 Kbytes
-
- Read access
  At 20MHz : 0Wait
- Operating mode:

1. CPU ROM mode

   This mode only allows reading of flash memory data. Word access is available. However, in this mode, it is not possible to activate the automatic algorithm*1 to perform writing or erasing.

2. CPU programming mode

   This mode allows reading, writing, and erasing of flash memory (automatic algorithm*1). Because word access is not available, programs that are contained in the flash memory cannot be executed while operating in this mode. Half-word access is available.

3. ROM writer mode

   This mode allows reading, writing, and erasing of flash memory from a ROM writer (automatic algorithm*1)

- Built-in flash security function
- (Prevents reading of the content of flash memory by a third party)
  See "CHAPTER Flash Security" for details on the flash security function.
-

Note:

- *This document explains flash memory in the case where it is being used in CPU mode.*
  For details on accessing the flash memory from a ROM writer, refer to the instruction manual of the ROM writer that is being used.

*1: Automatic algorithm=Embedded Algorithm

1.2 Configuration

This series consists of a 64 KBytes to 128 KBytes flash memory area, a security code area, and a built-in CR trimming data area.

The following figure illustrates the addresses of the flash memory and of the sector configuration and security/CR trimming data equipped in this series.

See "CHAPTER Flash Security" for details on the security.

For details of the Built-in high-speed CR trimming data, see "1.4.4 CRTRMM (CR Trimming Data Mirror Register)" and refer to the chapter on the high-speed CR trimming function in "FM3 Peripheral Manual".

Figure 1-1. Memory Map of the CY9AF131/CY9AFA31/CY9AF1A1/CY9AFAA1 Flash Memory

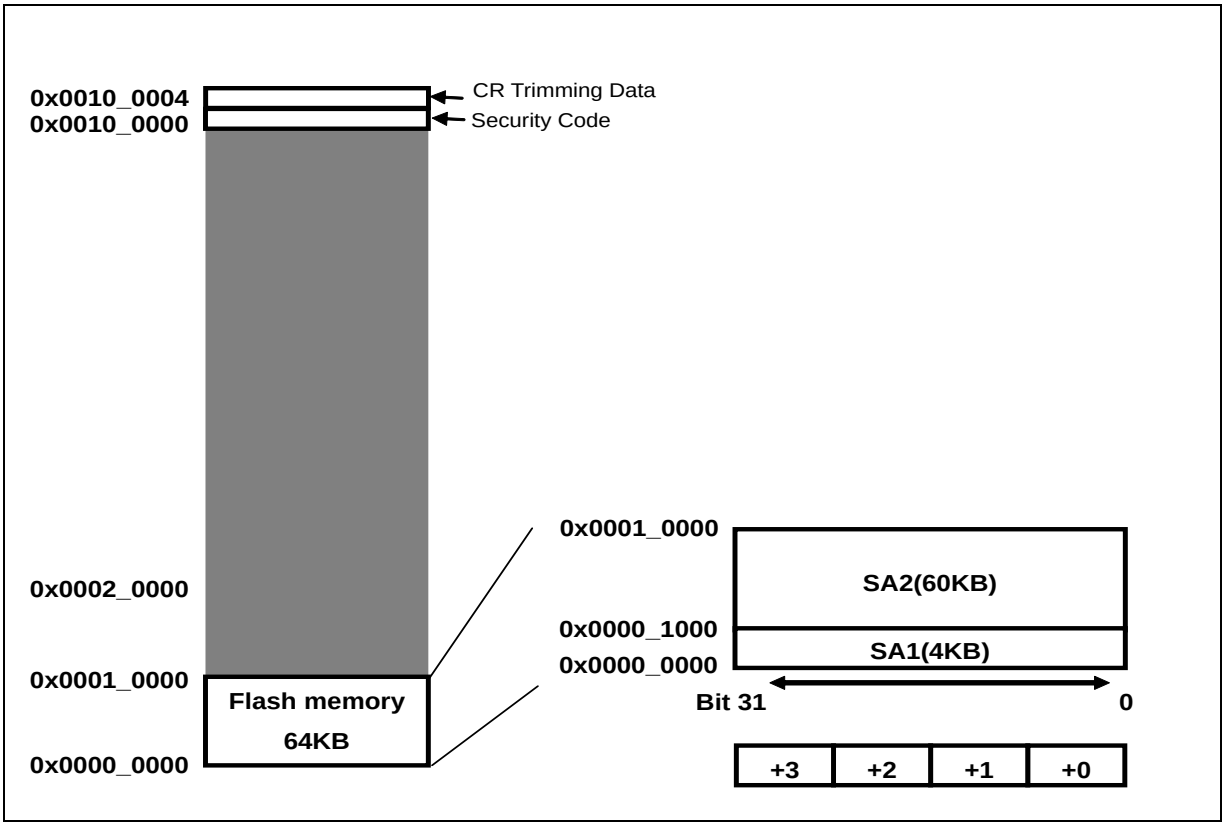


Figure 1-2. Memory Map of the CY9AF132/CY9AFA32/CY9AF1A2/CY9AFAA2 Flash Memory

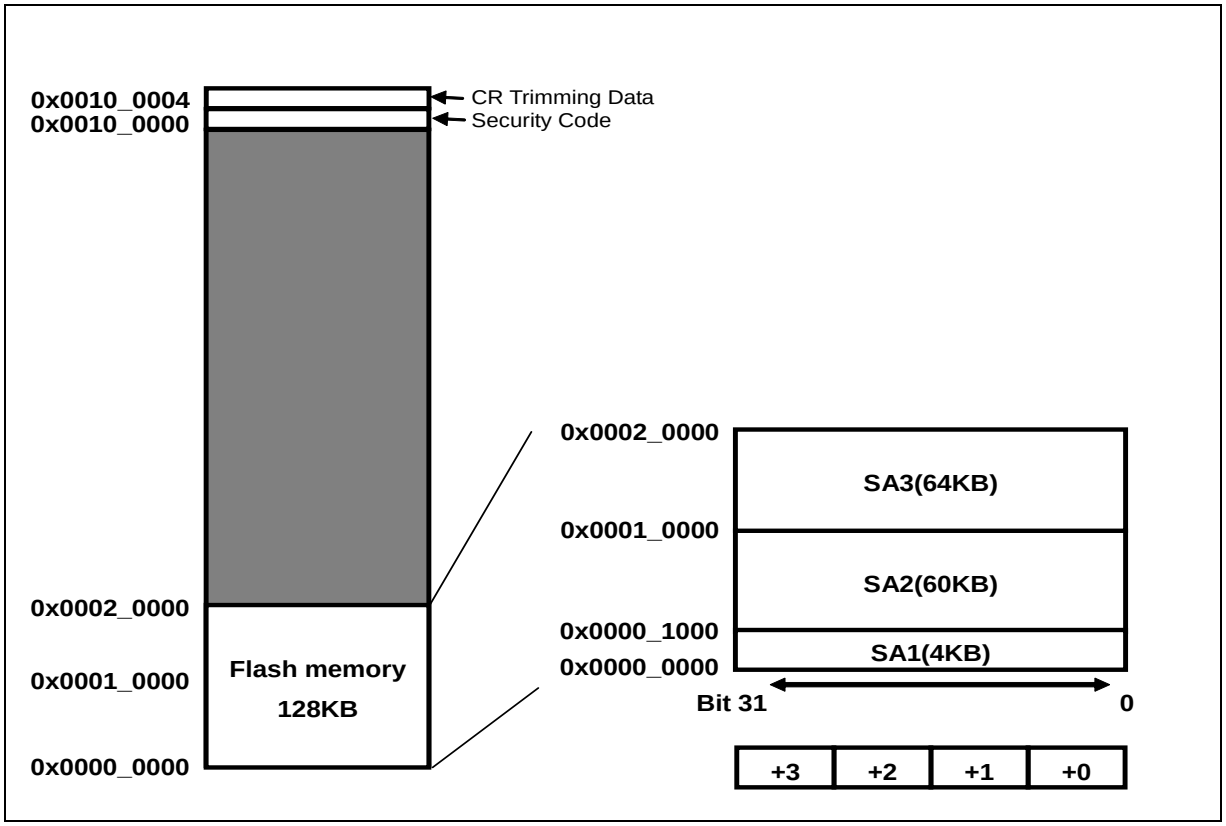
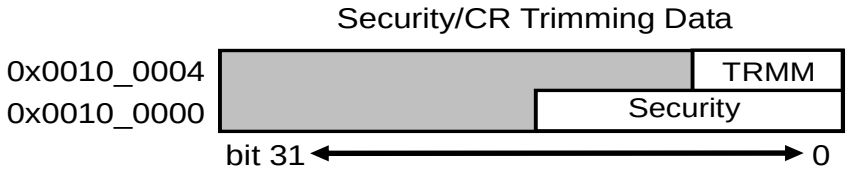


Figure 1-3. Security/CR Trimming Data Address



1.3 Operating Description

This section explains the Flash memory operation.

1.3.1 Flash Memory Access Modes

1.3.2 Automatic Algorithm

1.3.3 Explanation of Flash Memory Operation

1.3.4 Cautions When Using Flash Memory

1.3.1 Flash Memory Access Modes

The following two access modes are available for accessing flash memory from the CPU.

- CPU ROM mode
- CPU programming mode

These modes can be selected by the flash access size bits (FASZR: ASZ).

- CPU ROM Mode

This mode only allows reading of flash memory data.
This mode is entered by setting the flash access size bits (FASZR: ASZ) to "10" (32-bit read), and enables word access.

However, in this mode, it is not possible to execute commands, to activate the automatic algorithm or to write or erase data.

The flash memory always enters this mode after reset is released.

- CPU Programming Mode

This mode allows reading, writing, and erasing of data.
This mode is entered by setting the flash access size bits (FASZR: ASZ) to "01" (16-bit read/write), and enables flash programming.

Because word access is not possible in this mode, programs that are contained in the flash memory cannot be executed. The operation while in this mode is as follows.

- During reading

  Flash memory is accessed in half-words, with data read out in blocks of 16 bits.

- During writing commands

  The automatic algorithm can be activated to write or erase data. See Section "1.3.2 Automatic Algorithm" for details on the automatic algorithm.

Table 1-1. Access modes of Flash memory

Access Mode	Access Size	Automatic algorithm	Instruction execution in the Flash Memory
CPU ROM mode	32 bits	disable	Enable
CPU Programming mode	1 6 bits	enable	Prohibited

Note:

- *The flash memory is always set to CPU ROM mode when a reset is released. Therefore, if a reset occurs after entering CPU programming mode, the flash access size bits (FASZR: ASZ) are set to "10" and the flash memory returns to CPU ROM mode.*

1.3.2 Automatic Algorithm

When CPU programming mode is used, writing to and erasing flash memory is performed by activating the automatic algorithm.
This section explains the automatic algorithm.

1.3.2.1 Command Sequences
1.3.2.2 Command Operating Explanations
1.3.2.3 Automatic Algorithm Run States

1.3.2.1 Command Sequences

The automatic algorithm is activated by sequentially writing half-word (16-bit) data to the flash memory one to six times in a row. This is called a command. Table 1-2 shows the command sequences.

Table 1-2. Command Sequence Chart

Command	Number of writes	1st write		2nd write		3rd write		4th write		5th write		6th write	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Read/ Reset	1	0xXXX	0xF0	--	--	--	--	--	--	--	--	--	--
Write	4	0xAAA8	0xAA	0x554	0x55	0xAAA8	0xA0	PA	PD	--	--	--	--
Chip erase	6	0xAAA8	0xAA	0x554	0x55	0xAAA8	0x80	0xAAA8	0xAA	0x554	0x55	0xAAA8	0x10
Sector erase	6	0xAAA8	0xAA	0x554	0x55	0xAAA8	0x80	0xAAA8	0xAA	0x554	0x55	SA	0x30
Sector erase suspend	1	0xXXX	0xB0	--	--	--	--	--	--	--	--	--	--
Sector erase resume	1	0xXXX	0x30	--	--	--	--	--	--	--	--	--	--

X : Any value

PA : Write address

SA : Sector address(Specify any address within the address range of the sector to erase)

PD : Write data

Notes:

- *The data notation in the table only shows the lower 8 bits. The upper 8 bits can be set to any value.*
- *Write commands as half-words at any time.*
- *The addresses notation in the above table only show the lower 16-bit. Set the upper 16-bit to any address within the address range of the target flash memory. When any address outside of the flash address range is set, the flash memory cannot recognize the command. Therefore, the command sequence will not be executed successfully.*
- *For the Flash security code, write to 0x0010_0000 as PA.*
- *For the CR trimming data, write to 0x0010_0004 as PA.*

- *For the address to erase the Flash security code or CR trimming data, specify "0x0010_0000 or 0x0010_0004". Both Flash security code and CR trimming data would be erased even if either address is specified. Furthermore, either-or could not be erased.*

1.3.2.2 Command Operating Explanations

This section explains the command operating.

- Read/Reset Command

The flash memory can be read and reset by sending the read/reset command to the target sector in sequence.

When a read/reset command is issued, the flash memory maintains the read state until another command is issued.

When the execution of the automatic algorithm exceeds the time limit, the flash memory is returned to the read/reset state by issuing the read/reset command.

See Section "1.3.3.1 Read/Reset Operation" for details on the actual operation.

- Program (Write) Command

To write data to the flash memory, activate the automatic algorithm by issuing write command to the target sector in four consecutive writes. Data writes can be performed in any order of addresses, and may also cross sector boundaries.
In CPU programming mode, data is written in half-words.
Once the fourth command issuance has finished, the automatic algorithm is activated and the automatic write to the flash memory starts. After executing the automatic write algorithm command sequence, there is no need to control the flash memory externally.
See Section "1.3.3.2 Write Operation" for details on the actual operation.

<Notes>

- The write is not performed properly if the fourth write command (write data cycle) is issued to an odd address. Always write to an even address.
- Only a single half-word of data can be written for each write command sequence.
  To write multiple pieces of data, issue one write command sequence for each piece of data.

- Chip Erase Command

All of the sectors in flash memory can be batch-erased by sending the chip erase command to the target sector in six consecutive writes. Once the sixth sequential write has finished, the automatic algorithm is activated and the chip erase operation starts.

- Sector Erase Command

A single sector of flash memory can be erased by sending the sector erase command to the target sector in six consecutive writes. Once the sixth sequential write has finished and 35µs has elapsed (timeout interval), the automatic algorithm is activated and the sector erase operation begins.
To erase multiple sectors, issue the erase code (0x30) which is the sixth write code of the sector erase command to the address of the sector to erase within 35µs (timeout interval). If the sector erase code is not issued within the timeout interval, the sector erase code that is added after the timeout interval is elapsed may become inactive.

- Sector Erase Suspend Command

Issuing the sector erase suspend command while the sector erase is in process or while the command time-out is executed can suspend the sector erase. In the state of a sector erase suspend, the memory cell of the sector which is not the erase target can be read.
For the actual operation, see "1.3.3.5 Sector Erase Suspend Operation".

Note:

- *This command is enabled only while the sector erase is in process. This command is ignored even when it is issued during the chip erase or writing.*

■ Sector Erase Resume Command

To resume a suspended erase from the sector erase suspend state, issue the sector erase resume command. When the sector erase resume command is issued, the state is returned to the sector erase state. Then, the erase operation is resumed.
For the actual operation, see "1.3.3.6 Sector Erase Resume Operation".

Note:

- *This command is enabled only while the sector erase suspend is in process. This command is ignored even when it is issued during the sector erase.*

1.3.2.3 Automatic Algorithm Run States

Because writing and erasing of flash memory is performed by the automatic algorithm, whether or not the automatic algorithm is currently executing can be checked using the flash ready bit (FSTR:RDY) and the operating status can be checked using the hardware sequence flags.

■ Hardware Sequence Flags

These flags indicate the status of the automatic algorithm. When the flash ready bit (FSTR:RDY) is "0", the operating status can be checked by reading any address in flash memory.

Figure 1.3-1 shows the bit structure of the hardware sequence flags.

Figure 1.3-1 Bit structure of the hardware sequence flags

In the event of half-word access

bit	15	14	13	12	11	10	9	8
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

bit	7	6	5	4	3	2	1	0
	DPOL	TOGG	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

In the event of byte access

bit	7	6	5	4	3	2	1	0
	DPOL	TOGG	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

Notes:

- *These flags cannot be read using word access. When in CPU programming mode, always read using half-word or byte access.*
- *In CPU ROM mode, the hardware sequence flags cannot be read no matter which address is read.*
- *Because the correct value might not be read out immediately after issuing a command, ignore the first value of the hardware sequence flags that is read after issuing a command.*

- Status of each bit and flash memory

Table 1-3 shows the correspondence between each bit of the hardware sequence flags and the status of the flash memory.

Table 1-3. List of Hardware Sequence Flag States

State				DPOL	TOGG	TLOV	SETI	TOGG 2
Running	Automatic write operation			Inverted data (*1)	Toggle	0	0	0
	Automatic erase operation	Chip erase		0	Toggle	0	1	Toggle
		Sector erase	Timeout interval	0	Toggle	0	0	Toggle
			Erase	0	Toggle	0	1	Toggle
		Sector erase suspend	Read (Sector to erase)	0	0	0	1	Toggle
			Read (Sector not to erase)	Data (*1)	Data (*1)	Data (*1)	Data (*1)	Data (*1)
Time Limit exceeded	Automatic write operation			Inverted data (*1)	Toggle	1	0	0
	Automatic erase operation			0	Toggle	1	1	Toggle

*1: See "•Bit Descriptions" for the values that can be read.

- Bit Descriptions

[bit15:8] Undefined bits

[bit7] DPOL : Data polling flag bit

When the hardware sequence flags are read, by specifying an arbitrary address, this bit uses a data polling function to indicate whether or not the automatic algorithm is currently running.
The value that is read out varies depending on the operating state.

- During writing
  - ☐ While write is in progress:

    Reads out the opposite value (inverse data) of bit 7 of data written at the last command sequence (PD). This does not access the address that was specified for reading the hardware sequence flags.
  - ☐ After write finishes:

    Reads out the value of bit 7 of the address specified for reading the hardware sequence flags.
- During sector erase
  - ☐ While sector is executing:

    Reads out "0" from all regions of flash memory.
  - ☐ After sector erase finishes:

    Always reads out "1".
- During chip erase
  - ☐ While chip erase is executing: Always reads out "0".
  - ☐ After chip erase: Always reads out "1".

- At the sector erase suspend
  - When the address of the sector set for the sector erase is specified and read:

    Reads out "0".
- When the address whose sector is not set for the sector erase is specified and read:

  Reads out the value of bit7 of the specified address.

<Note>

The data for a specified address cannot be read while the automatic algorithm is running.
Confirm that the automatic algorithm has finished running by using this bit before reading data.

[bit6] TOGG : Toggle Flag Bit

When the hardware sequence flags are read by specifying an arbitrary address, this bit indicates whether or not the automatic algorithm is currently running.
The value that is read out varies depending on the operating state.

- During write, sector erase, or chip erase
  - While write, sector erase, or chip erase is in progress:
    When this bit is read out continuously, it alternatively returns "1" and "0" (toggles). The address that was specified for reading the hardware sequence flags is not accessed.
  - After write, sector erase, or chip erase has finished:

    Reads out the value of bit 6 of the address specified for reading the hardware sequence flags.
- At the sector erase suspend
  - When the address of the sector set for the sector erase is specified and read:

    Reads out "0".
  - When the address whose sector is not set for the sector erase is specified and read:

    Reads out the value of bit6 of the specified address.

[bit5] TLOV : Timing Limit Exceeded Flag Bit

When the hardware sequence flags are read by specifying an arbitrary address, this bit indicates whether or not the execution time of the automatic algorithm has exceeded the rated time defined internally within the flash memory (number of internal pulses).
The value that is read out varies depending on the operating state.

- During write, sector erase, or chip erase

  The following values are read out.

  0 : Within the rated time
  1 : Rated time exceeded

When this bit is "1", if the DPOL bit and TOGG bit indicate that the automatic algorithm is currently executing, that means a failure occurred during the write or erase.

For example, because data that has been written to "0" cannot be overwritten to "1" in flash memory, if "1" is written to an address that has been written to "0", the flash memory is locked and the automatic algorithm does not finish. In this case, the value of the DPOL bit remains invalid, and "1" and "0" are continuously read out alternatingly from the TOGG bit.
Once the rated time is exceeded while still in this state, this bit changes to "1". If this bit changes to "1", issue the reset command.

- At the sector erase suspend
  - ☐ When the address of the sector set for the sector erase is specified and read:

    Reads out "0".
  - ☐ When the address whose sector is not set for the sector erase is specified and read:

    Reads out the value of bit5 of the specified address.

Note:

- *If this bit is "1", it indicates that the flash memory was not used correctly. This is not a malfunction of the flash memory.*
  Perform the appropriate processing after issuing the reset command.

[bit4] Undefined bit

[bit3] SETI : Sector Erase Timer Flag Bit

When a sector is erased, a timeout interval of 35 µs is required from when the sector erase command is issued until the sector erase actually begins.
When the hardware sequence flags are read by specifying an arbitrary address, this bit indicates whether or not the flash memory is currently in the sector erase command timeout interval.
The value that is read out varies depending on the operating state.

- During sector erase:

  When sectors are being erasing, it can be checked whether or not the following sector erase code can be accepted by checking this bit before inputting the following sector erase code.
  The following values are read out without accessing the address specified in order to read the hardware sequence flags.

  0 : Within sector erase timeout interval
  The following sector erase code (0x30) can be accepted.

  1 : Sector erase timeout interval exceeded
  In this case, if the DPOL bit and TOGG bit indicate that the automatic algorithm is currently executing, the erase operation has started internally within the flash memory. In this case, commands other than sector erase suspend (0xB0) are ignored until the internal flash memory erase operation has finished.
- At the sector erase suspend
  - ☐ When the address of the sector set for the sector erase is specified and read:

    Reads out "1".
  - ☐ When the address whose sector is not set for the sector erase is specified and read:

    Reads out the value of bit3 of the specified address.

[bit2] TOGG2: Toggle Flag Bit

In the sector erase suspend state, a sector which is not the erase target can be read. However, the erase target sector cannot be read. This toggle bit flag can detect whether the corresponding sector is the erase target sector during the sector erase suspend by checking the toggle operation of the read data.

- At writing

  Reads out "0".

- At the sector erase/chip erase

  When this bit is read consecutively, "1" and "0" are alternately read (toggle operation).

- At the sector erase suspend
  - When the address of the sector set for the sector erase is specified and read:

    When this bit is read consecutively, "1" and "0" are alternately read (toggle operation).

- When the address whose sector is not set for the sector erase is specified and read:

  Reads out the value of bit2 of the specified address.

[bit1:0] Undefined bits

1.3.3 Explanation of Flash Memory Operation

The operation of the flash memory is explained for each command.

1.3.3.1 Read/Reset Operation

This section explains the read/reset operation.

The flash memory can be put into the read/reset state by sending the read/reset command to the target sector sequentially.
Because the read/reset state is the default state of the flash memory, the flash memory always returns to this state when the power is turned on or when a command finishes successfully. When the power is turned on, there is no need to issue a data read command. Furthermore, because data can be read by normal read access and programs can be accessed by the CPU while in the read/reset state, there is no need to issue read/reset commands.

1.3.3.2 Write Operation

This section explains the write operation.

Writes are performed according to the following procedure.

1. The write command is issued to the target sector sequentially

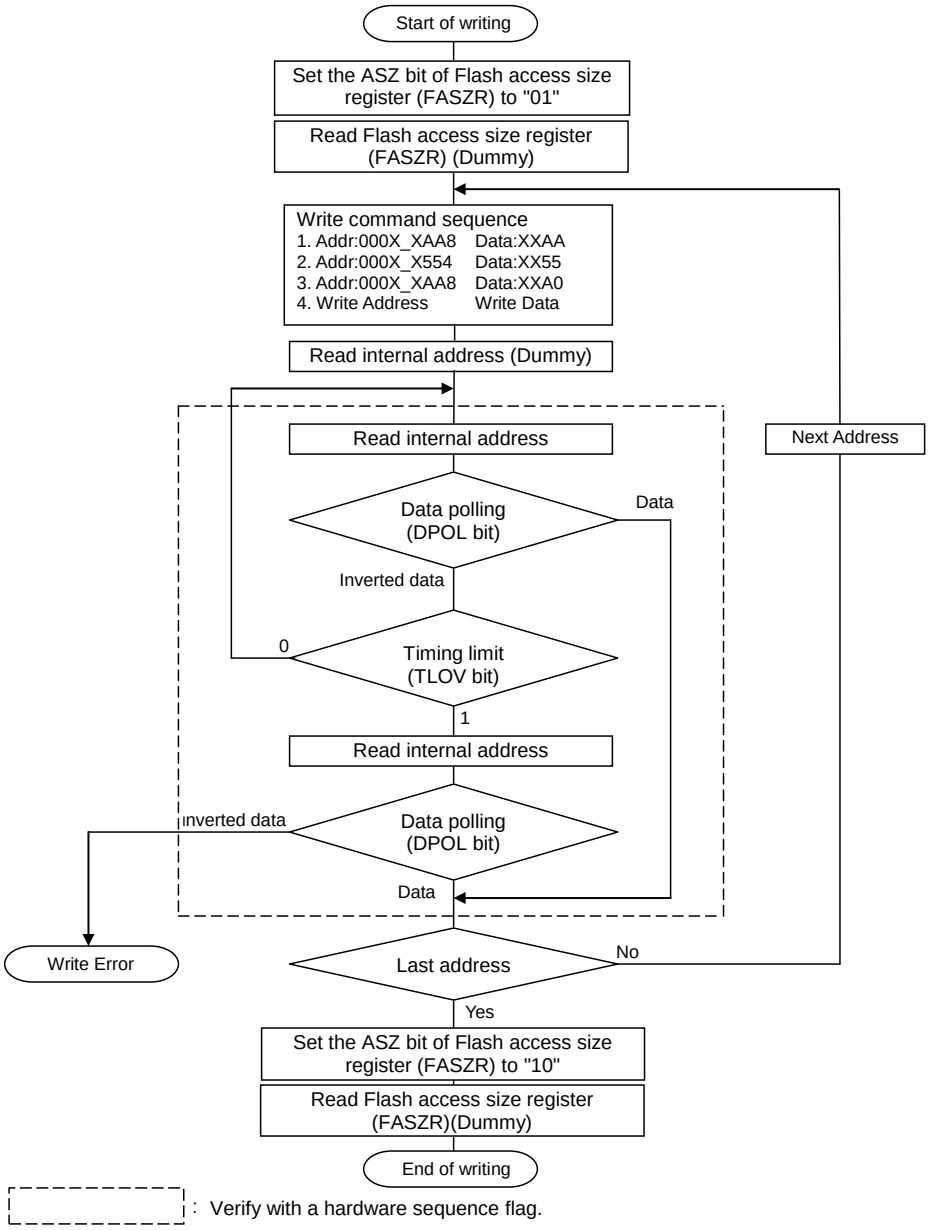
   The automatic algorithm activates and the data is written to the flash memory.
   After the write command is issued, there is no need to control the flash memory externally.

2. Perform read access on the address that was written

   The data that is read is the hardware sequence flags. Therefore, once bit 7 (the DPOL bit) of the read data matches the value that was written, the write to the flash memory has finished. If the write has not finished, the reverse value (inverted data) of bit 7 of data written at the last command sequence (PD) is read out.

Figure 1-4 shows an example of a write operation to the flash memory.

Figure 1-4. Example of Write Operation



Notes:

- *See Section "1.3.2 Automatic Algorithm" for details on the write command.*
- *The address notations in command sequences only show the lower 12 bits. The upper 20bits should be set to any address within the address range of the target flash memory. When the address outside the flash address range is specified, the command sequence would not operate correctly since the flash memory cannot recognize the command.*
- *Because the value of the DPOL bit of the hardware sequence flags changes at the same time as the TLOV bit, the value needs to be checked again even if the TLOV bit is "1".*
- *The toggle operation stops at the same time as the TOGG bit and TLOV bit of the hardware sequence flags change to "1". Therefore, even if the TLOV bit is "1", the TOGG bit needs to be checked again.*

- *Although the flash memory can be written in any sequence of addresses regardless of crossing sector boundaries, only a single half-word of data can be written with each write command sequence. To write multiple pieces of data, issue one write command sequence for each piece of data.*
- *All commands issued to the flash memory during the write operation are ignored.*
- *If the device is reset while the write is in progress, the data that is written is not guaranteed.*

1.3.3.3 Chip Erase Operation

This section explains the chip erase operation.

This operation can erase all the sectors in the flash memory simultaneously. This is called Chip Erase.

Sending the chip erase command consecutively to the target sector starts the automatic algorithm and erases all the sectors simultaneously.
For the chip erase command, see "1.3.2 Automatic Algorithm".

1. Issue the chip erase command sequentially to the target sector

   The automatic algorithm is activated and the chip erase operation of the flash memory is started.

2. Perform read access to an arbitrary address

   The data that is read is the hardware sequence flag. Therefore, if the value of bit 7 (the DPOL bit) of the data that was read is "1", that means the chip erase has finished.

The time required to erase the chip is "sector erase time × total number of sectors + chip write time (preprogramming)".
Once the chip erase operation has finished, the flash memory returns to read/reset mode.

1.3.3.4 Sector Erase Operation

This section explains the sector erase operation.

Sectors in the flash memory can be selected and the data of only the selected sectors can be erased. Multiple sectors can be specified at the same time.

Sectors are erased according to the following sequence.

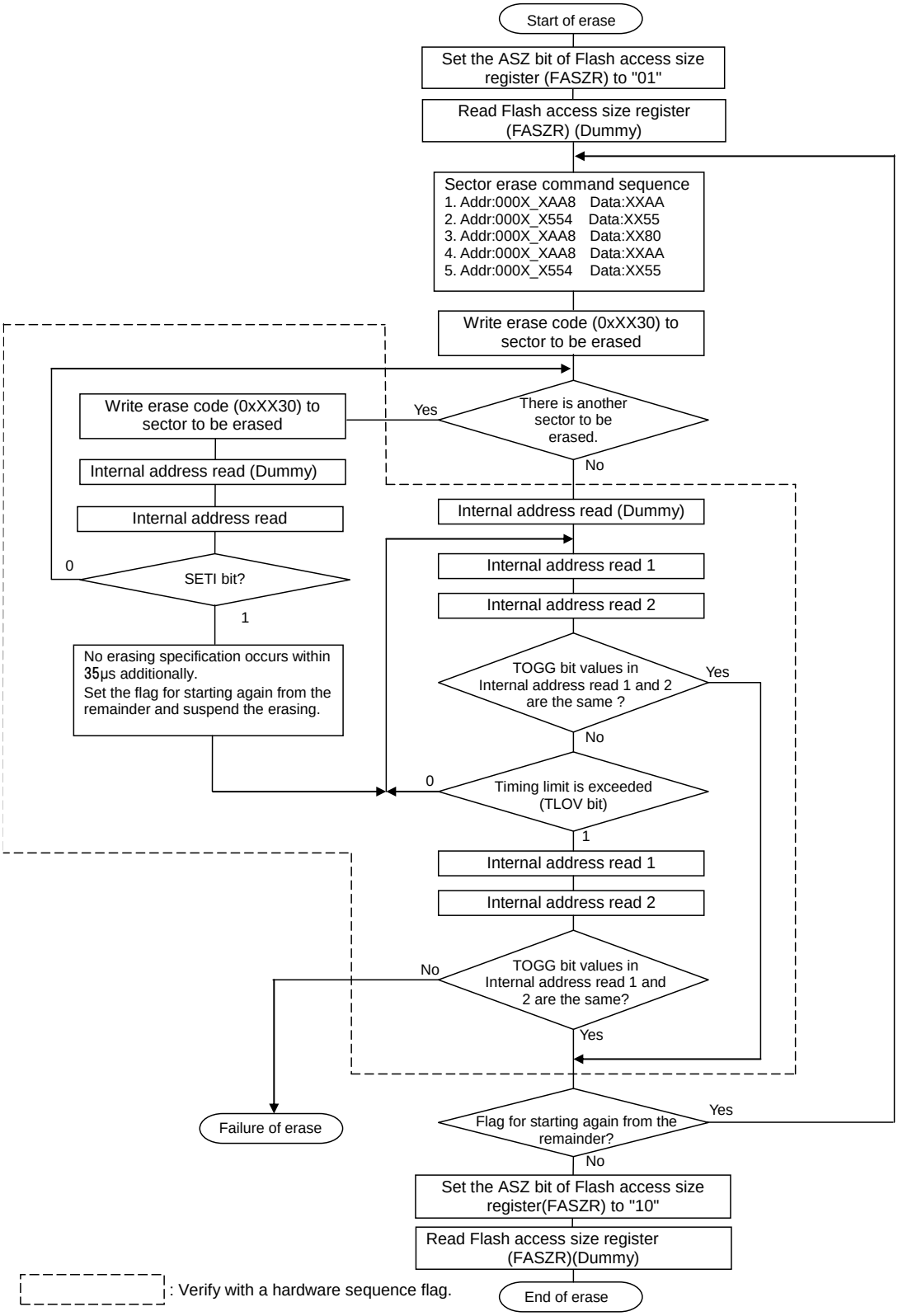
1. Issue the sector erase command sequentially to the target sector

   Once 35 µs has elapsed (the timeout interval), the automatic algorithm activates and the sector erase operation begins.
   To erase multiple sectors, issue the erase code (0x30) to an address in the sector to erase within 35 µs (timeout interval). If the code is issued after the timeout interval has elapsed, the sector erase code may be invalid.

2. Perform read access to an arbitrary address

   The data that is read is the hardware sequence flags. Therefore, if the value of bit 7 (the DPOL bit) of the data that was read is "1", that means the sector erase has finished.
   Furthermore, it can be checked whether or not the sector erase has finished by using the TOGG bit.
   Figure 1-5 shows an example of the sector erase procedure for the case of using the TOGG bit for confirmation.

Figure 1-5. Example of Sector Erase Procedure



The time required to erase a sector is "(sector erase time + sector write time (preprogramming)) × number of sectors". Once the sector erase operation has finished, the flash memory returns to read/reset mode.

Notes:

- *See Section "1.3.2 Automatic Algorithm" for details on the sector erase command.*
- *The address notations in command sequences only show the lower 12 bits. The upper 20bits should be set to any address within the address range of the target flash memory. When the address outside the flash address range is specified, the command sequence would not operate correctly since the flash memory cannot recognize the command.*
- *Because the value of the DPOL bit of the hardware sequence flags changes at the same time as the TLOV bit, the value needs to be checked again even if the TLOV bit is "1".*
- *The toggle operation stops at the same time as the TOGG bit and TLOV bit of the hardware sequence flags change to "1". Therefore, even if the TLOV bit is "1", the TOGG bit needs to be checked again.*
- *When a command other than the sector erase command/erase suspend command is issued during the sector erase including the time-out period, it is ignored.*

1.3.3.5 *Sector Erase Suspend Operation*

This section explains the sector erase suspend operation.

When the sector erase suspend command is sent during the sector erase or in the command time-out state, the device is transferred to the sector erase suspend state. Then, the erase operation is temporarily stopped. When the erase resume command is sent, the state is returned to the sector erase state to resume the interrupted erase operation. However, although the state transition is made from the command time-out to the sector erase suspend state, a transition is not made to the command time-out in the case of successful erase resume command writing. In that case, the transition is made to the sector erase state, and the sector erase is immediately resumed.

■ Sector Erase Suspend Operation

The sector erase is suspended in the following sequence:

1. In the period from the command time-out to the sector erase in process, write the sector erase suspend command to any address within the flash memory address range.
2. When the sector erase suspend command is issued during the command time-out, immediately terminate the time-out and suspend the erase operation. When the sector erase suspend command is issued during the sector erase, it takes a maximum of 35 µs to stop the actual erase.

Notes:

- *For details of the sector erase suspend command, see "1.3.2 Automatic Algorithm".*
- *The sector erase can be suspended only in the period from the sector erase command time-out to the sector erase in process. The chip erase cannot be suspended. In addition, although the sector erase suspend command is issued again during the sector erase suspend, it is ignored.*

■ State after Sector Erase Suspend

When the erase target sector is read after the sector erase suspend, the hardware sequence flag is read out. Furthermore, when the sector which is not the erase target is read, the memory cell data is read out.

Note:

- *In the state of the sector erase suspend, a new write or a new erase command is ignored.*

1.3.3.6 *Sector Erase Resume Operation*

This section explains the sector erase resume operation while the sector erase suspend is in process.

The sector erase can be resumed when the sector erase resume command is issued to any address while the sector erase is suspended.
When the sector erase resume command is issued, the sector erase operation which is in the state of sector erase suspend is resumed.
For details of the sector erase resume command, see "1.3.2 Automatic Algorithm".

Notes:

- *The sector erase resume command is enabled only when the sector erase suspend is in process. Although the sector erase resume command is issued during the sector erase, it is ignored.*
- *It takes 2ms or longer before the sector erase operation is resumed after the sector erase resume command is issued. Therefore, when an erase resume and erase stop are repeated at intervals that are shorter than this time length, the timing limit is exceeded without progress to the erase operation. When the sector erase suspend command is issued again after the sector erase resume command is issued, a time interval of 2ms or longer needs to be placed after the sector erase resume command is issued.*

1.3.4 Cautions When Using Flash Memory

This section explains the cautions when using flash memory.

- If this device is reset during the write, the data that is written cannot be guaranteed.
  Moreover, it is necessary to prevent an unexpected reset like Watchdog Timer from occurring during the writing and deleting.
- If the CPU programming mode is configured (ASZ=01) in the ASZ[1:0] bits of the flash access size register (FASZR), do not execute any programs in the flash memory. The correct values will not be retrieved and the program will run out of control.
- If the CPU programming mode is configured (ASZ=01) in the ASZ[1:0] bits of the flash access size register (FASZR) and the interrupt vector table is in the flash memory, ensure that no interrupt requests occur. The correct values will not be retrieved and the program will run out of control.
- If the CPU programming mode is configured (ASZ=01) in the ASZ[1:0] bits of the flash access size register (FASZR), do not transition to low power consumption mode.
- If the CPU ROM mode is configured (ASZ=10) in the ASZ[1:0] bits of the flash access size register (FASZR), do not write to the flash memory.
- If the CPU programming mode is configured (ASZ=01) in the ASZ[1:0] bits of the flash access size register (FASZR), always write to the flash memory in half-words. Do not write in bytes.
- Immediately after writing command of the automatic algorithm to the flash memory, always perform a dummy read before reading the data that is actually read. If data is read immediately after a write, the read value cannot be guaranteed.
- When the device is transferred to the low-power consumption mode in this device, make sure that the flash memory automatic algorithm operation is finished.
  For details of the low-power consumption mode, refer to the chapter "Low-power consumption mode" in "FM3 Peripheral Manual".

1.4 Registers

This section explains the registers.

- List of Registers

Abbreviated Register Name	Register Name	Reference
FASZR	Flash Access Size Register	1.4.1
FSTR	Flash Status Register	1.4.2
FSYNDN	Flash Sync Down Register	1.4.3
CRTRMM	CR Trimming Data Mirror Register	1.4.4

1.4.1 FASZR (Flash Access Size Register)

This section explains the FASZR.

This register configures the access size for flash memory. After reset is released, ASZ is set to "10" (32-bit read), and the flash memory enters CPU ROM mode. To put the flash memory into CPU programming mode, set ASZ to "01".

bit	7	6	5	4	3	2	1	0
Field	Reserved						ASZ	
Attribute							RW	RW
Initial value							1	0

Field	bit	Description
ASZ	1:0	Flash Access Size Specifies the access size of the flash memory. 00: Setting prohibited 01: 16-bit read/write (CPU programming mode) 10: 32-bit read (CPU ROM mode: Initial value) 11: Setting prohibited

Notes:

- *When ASZ is set to "01", always perform writes to flash using half-word access (16-bit access).*
- *Do not change this register using an instruction that is contained in the flash memory. Overwrite this register from a program in any other area except for flash memory area.*
- *Perform a dummy read to this register, after changing this register.*

1.4.2 FSTR (Flash Status Register)

This section explains the FSTR.

This is a status register of flash memory.

bit	7	6	5	4	3	2	1	0
Field	Reserved						HNG	RDY
Attribute							R	R
Initial value							0	X

[bit7:2] Reserved bits

The read values are undefined. Ignored on write.

[bit1] HNG : Flash Hang

Indicates whether the flash memory is in the HANG state. Flash memory enters the HANG state if the timing is exceeded (See "[bit5] TLOV: Timing Limit Exceeded Flag Bit"). If this bit becomes "1", issue a reset command (See Section "1.3.2.1 Command Sequences").
Because the correct value might not be read out immediately after issuing an automatic algorithm command, ignore the value of this bit as read out the first time after a command is issued.

Field	bit	Description
HNG	1	Flash Hang 0: The flash memory HANG state has not been detected. 1: The flash memory HANG state has been detected.

[bit0] RDY : Flash Rdy

Indicates whether a flash memory write or erase operation using the automatic algorithm is in progress or finished. While an operation is in progress, data cannot be written and the flash memory cannot be erased.

Field	bit	Description
RDY	0	Flash Rdy 0: Operation in progress (cannot write or erase) 1: Operation finished (can write or erase)

Because the correct value might not be read immediately after an automatic algorithm command is issued, ignore the value of this bit as read the first time after a command is issued.

1.4.3 FSYNDN (Flash Sync Down Register)

This section explains the FSYNDN.

The wait cycle is inserted in the read access to the flash memory at the CPU ROM mode. Current consumption can be reduced by decreasing the access clock frequency of the flash memory.

bit	7	6	5	4	3	2	1	0
Field	Reserved					SD		
Attribute						RW	RW	RW
Initial value						0	0	0

Field	bit	Description
SD	2:0	0b000: 0(Initial value) 0b001: +1 wait 0b010: Setting prohibited 0b011: +3 wait 0b100: Setting prohibited 0b101: +5 wait 0b110: Setting prohibited 0b111: +7 wait

Note:

- *After changing this register, perform a dummy read to the register.*

1.4.4 CRTRMM (CR Trimming Data Mirror Register)

This section explains the CRTRMM.

This register is a mirror register of the CR trimming data.
This register value can be used in the User Mode or Serial Writer Mode.

bit	31	10	9	0
Field	Reserved		TRMM	
Attribute			R	
Initial value			*	

[bit9:0] TRMM: CR Trimming Data Mirror Register

After the reset release, store the data of the address 0x0010_0004 in the Flash area to this register.
For details of the high-speed CR trimming data, refer to the chapter "High-speed CR Trimming Function" in "FM3 Peripheral Manual".

Field	bit	Description
TRMM	9:0	*: Lower 10bits in the address 0x0010_0004 should be read out.

Note:

- *The stored CR trimming data is lost because this register is cleared when a reset is issued within the Chip after the Flash memory erase. Therefore, before this register is cleared, save the CR trimming data stored in the register to the RAM or elsewhere.*

2. Flash Security

The flash security feature provides possibilities to protect the content of the flash memory. This chapter describes the overview and operations of the flash security.

1. Overview
2. Operation Explanation

2.1 Overview

This section explains the overview of the flash security.

If the protection code of 0x0001 is written in the security code area of flash memory, access to the flash memory is restricted. Once the flash memory is protected, performing the chip erase operation only can unlock the function otherwise read/write access to the data of the flash memory from any external pins is not possible.

This function is suitable for applications requiring security of self-containing program and data stored in the flash memory.

Table 2-1 shows the address and the protection code of the security code.

Table 2-1. Address and Protection Code of Security Code

Address	Protection Code
0x0010_0000	0x0001

2.2 Operation Explanation

This section explains the operation of the flash security.

- Setting Security

Write the protection code 0x0001 in address of the security code. The security is enabled and set after all the reset factors are generated or after turning on the power again.

- Releasing Security

The security is released by all the reset factors or turning on the power again after performing the chip erase.

- Operation with Security Enabled

The operations with security enabled vary depending on each mode.

Table 2-2 shows the security operations in each mode.

Table 2-2. Flash Operation with Security Enabled

Mode	Mode Pin MD[1:0]	Access to Flash			Access from JTAG Pins
		Chip Erase	Other Commands	Read	
User mode	"00"	Enabled	Enabled	Valid data	Disabled
Serial writer mode	"01"	Enabled	Disabled	Invalid data	Disabled

Notes:

- *Writing the protection code is generally recommended to take place at the end of the flash programming. This is to avoid unnecessary protection during the programming.*
- *In user mode, there is no limit to flash memory even during security is enabled. However, JTAG pins are fixed not to access internally from these pins during security is enabled. To release security, perform the chip erase operation using a serial writer because the security cannot be released through JTAG pins.*
- *When the CR trimming data is erased, the flash security code is also erased. When the flash security code is erased, the CR trimming data is also erased. Either flash security code or the CR trimming data cannot be erased.*
- *When security enabled, the obstruction analysis of the flash memory cannot be performed.*

■

3. Serial Programming Connection

This series supports serial onboard write (Cypress standard) to flash memory.
This chapter explains the basic configuration for serial write to flash memory by using the Cypress Serial Programmer.

1. Serial Programmer

3.1 Serial Programmer

Cypress Serial Programmer (software) is an onboard programming tool for all microcontrollers with built-in flash memory.

3.1.1 Basic Configuration

This section explains the basic configuration.

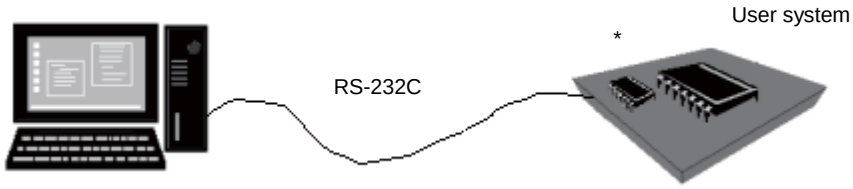
- Basic Configuration of FLASH MCU Programmer (Clock Asynchronous Serial Write)

FLASH MCU Programmer writes data, through clock asynchronous serial communication, to built-in flash memory of a microcontroller installed in the user system when the PC and the user system are connected through RS-232C cable.
In these series, also when a main clock does not exist, serial programming (UART communication mode) is possible.

If flash erase is executed to the flash memory that its flash security is enabled when built-in CR oscillator is used instead of external crystal oscillator, serial communication will be disconnected after the erase operation and then CR trimming data will be lost. When flash security is enabled, use external crystal oscillator.

Figure 3-1 shows the basic configuration of FLASH MCU Programmer, and Table 3-1 lists the system configuration.

Figure 3-1. Basic Configuration of FLASH MCU Programmer

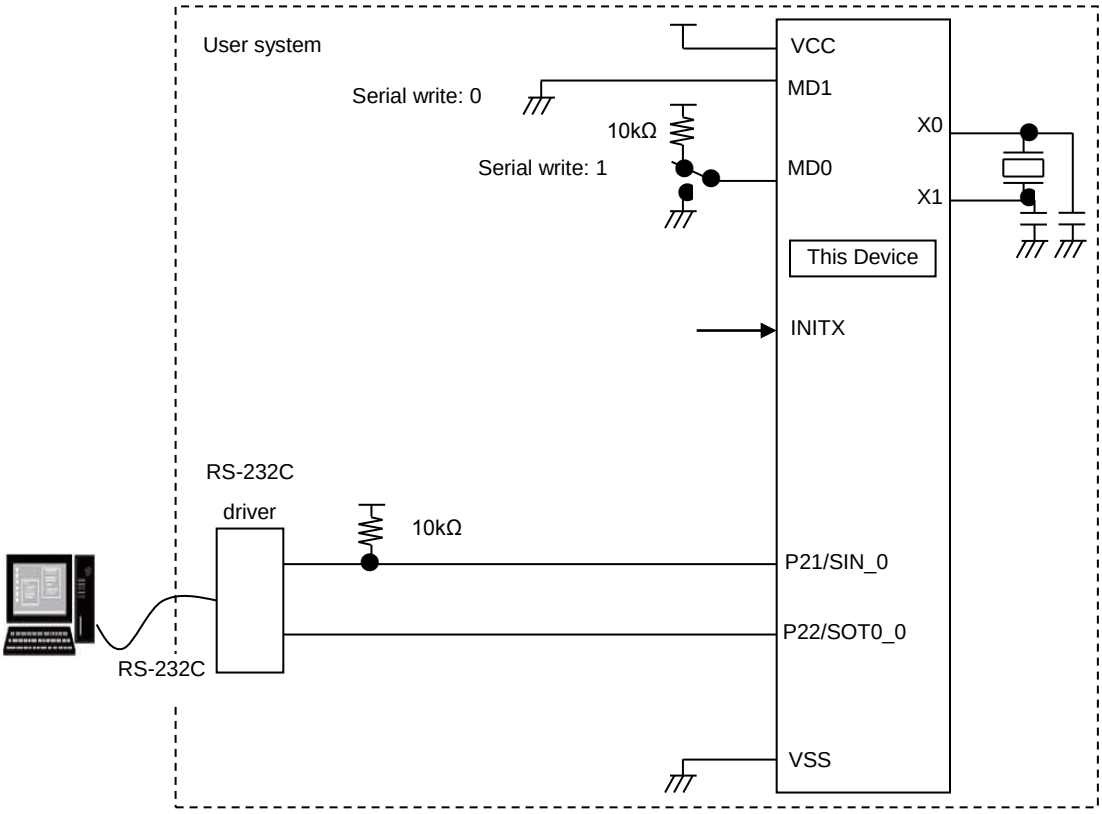


* RS-232C driver IC is required separately.

Table 3-1. System Configuration of FLASH MCU Programmer

Name	Specifications
FLASH MCU Programmer	Software (In case you request the data, contact to Cypress sales representatives.)
RS-232C cable	Sold on the market.

Figure 3-2. Connection Example using FLASH MCU Programmer



Note: The pull-up resistance values shown are for example. Select the most appropriate resistance values for each system.

Table 3-2. Oscillating Frequency and Communication Baud Rate Available for Clock Asynchronous Serial Communication

Source Oscillating Frequency	Communication Baud Rate
4 MHz	9600 bps
8 MHz	19200 bps
16 MHz	38400 bps
20 MHz	48000 bps

3.1.2 Pins Used

This section explains the used pins.

Table 3-3. Pins Used for Serial Write

Pins	Function	Supplement
MD0, MD1	Mode pin	Performing an external reset or turning on the power after setting MD0=H and MD1=L enters the serial write mode. When attaching a pull-up or pull-down resistor, avoid long wiring.
X0, X1	Oscillation pin	See the "Data Sheet" for the source oscillation clock (main clock) frequencies that can be used in serial write mode. (Restrictions apply to clock asynchronous communication. For details, see Table 3-2.)
P22/SOT0_0	UART serial data output pin	When the communication mode is set to UART, this pin becomes a serial data output pin when communication begins after the serial write mode is activated.
P21/SIN0_0	Clock synchronous/ asynchronous select pin/UART serial data input pin	Setting the input level of this pin to "H" until the start of communication enables the clock asynchronous communication mode, and setting it to "L" enables the clock synchronous communication mode. When the communication mode is set to UART, this pin can be used as a serial data input pin when communication begins after the serial write mode is activated.
INITX	Reset pin	-
VCC	Power supply pin	For writing, supply power to the microcontroller from the user system.
VSS	GND pin	-

Note:

- *Note that initial states of un-used pins in the serial writer mode are the same as initial states of them in the user mode. Refer to "Pin Status in Each CPU State" section in "Data Sheet" of the product used and relevant chapters in the "FM Family Peripheral Manual".*

Revision History

Document Revision History

Document Title: FM3 CY9AAA0N/1A0N/A30N/130N/130L Series, Flash Programming Specifications Document Number: 002-04750			
Revision	**Issue Date**	**Origin of Change**	**Description of Change**
**	03/31/2014	AKIH	Rev 1.0 Initial release Rev 2.0 MB9A130LA series, MB9A130N series, and MB9AA30N series were added. MB9A130LA series, MB9A130N series, and MB9AA30N series were added. MB9AFA31 was added to the memory map (Figure 1-1). MB9AFA32 was added to the memory map (Figure 1-2). Rev 2.1 Company name and layout design change Rev 3.0 MB9A130LB series was added. Rev 4.0 MB9AAA0N series and MB9A1A0N series was added.
*A	01/29/2016	AKIH	Migrated Spansion Guide from MB9A130L-MN706-00011-4v0-E to Cypress format
*B	07/26/2017	YSAT	Adapted Cypress new logo
*C	02/27/2018	NOSU	Modified link to Microcontroller Support Information Page Add a note about address notation in command sequences. Add a note for initial states of un-used pins in the serial writer mode.
*D	11/14/2018	HTER	Series name MB9AAA0N/1A0N/A30N/130N/130L -> CY9AAA0N/1A0N/A30N/130N/130L Changed title and category