32-bit RISC Microcontroller

TXZ Family

Reference Manual

Flash Memory

(Code Flash: 512KB/256KB/128KB)
(Data Flash: 32KB)
(Flash512_32-A)

Revision 2.1

2022-07

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Contents

List of Figures

List of Tables

Preface

Related documents

Document name
Clock Control and Operation Mode
Exception
Input/Output Ports
Product Information
Asynchronous Serial Communication Circuit

Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 – Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 – It is possible to omit the "0b" when the number of bit can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m: n].
  Example: S[3: 0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "n" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1]***, ***[XYZ2]***, ***[XYZ3]*** → ***[XYZn]***
- "x" substitutes suffix number or character of units and channels in the Register List.
  In case of unit, "x" means A, B, and C ...
  Example: ***[ADACR0]***, ***[ADBCR0]***, ***[ADCCR0]*** → ***[ADxCR0]***
  In case of channel, "x" means 0, 1, and 2 ...
  Example: ***[T32A0RUNA]***, ***[T32A1RUNA]***, ***[T32A2RUNA]*** → ***[T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> =0x01 (hexadecimal), ***[XYZn]***<VW> =1 (binary)
- Word and Byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and Write are possible
- Unless otherwise specified, register access supports only word access.
- The register defined as reserved must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "-" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "-", follow the definition of each register.
- Reserved bits of the Write-only register should be written with their default value.
  In the cases that default is "-", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

Terms and Abbreviation

Some of abbreviations used in this document are as follows:

ACK	Acknowledgement
Addr	Address
Adr	Address
BLK	Block
KB	Kilo Bytes
PG	Page
SFR	Special Function Register
UART	Universal Asynchronous Receiver Transmitter

1. Outline

The code flash which stores a program code, and the data flash which stores data are explained.
A code flash stores an instruction code, and CPU reads and executes it.

There is user information area which can be accessed in a code flash by bank change. Since user information area is not erased by a chip erasing command, for example, it can be written a unique management number etc. for every chip.

A data flash stores data, and even if power supply is intercepted, it keeps data.

Table 1.1 Functional description (code flash)

Flash memory	Function classification	Function	Functional Description	Comments
Code Flash 512KB 384KB 256KB	Programming and Erasing	Automatic Programming	Data programming is performed at 4 words (16 bytes).	
		Automatic chip erasing	Erasing of all the area of a flash memory is performed automatically. Object: Code flash Data flash	Except User information area in code flash.
		Automatic area erasing	Erasing in an area unit is performed automatically.	
		Automatic block erasing	Erasing in a block unit is performed automatically.	
		Automatic page erasing	Erasing in a page unit is performed automatically.	
	Program/erase protection	Protection	Programming and erasing can be prohibited per block.	
	Security	Security	Prohibition of read-out from the flash memory by a flash writer and using a debugging tools.	
	Memory swap	Automatic memory swap	Swap /swap release /swap size specification of a code flash block is performed automatically.	
	Execute Instruction	Execute Instruction	Instructions can be executed.	
	program/erase to other Flash area	program/erase to data Flash area	Basic operation to a data flash can be performed.	Dual mode

Table 1.2 Functional description (user information area)

Flash memory	Function classification	Function	Functional Description	Comments
User information area (Code Flash) 4KB	Programming and Erasing	Automatic Programming	Data programming is performed at 4 words (16 bytes).	
		Automatic page erasing	Erasing all the User information area is performed automatically.	
	Security	Security	Prohibition of read-out of the flash memory by a flash writer and the usage restrictions of a debugging function can be carried out.	It is controlled by the operation on the code flash side.
	Execute Instruction	-	-	Execution of instruction cannot be performed.

Table 1.3 Functional description (data flash)

Flash memory	Function classification	Function	Functional Description	Comments
Data Flash 32KB	Programming and Erasing	Automatic Programming	Data programming is performed at 1 words (4 bytes).	
		Automatic area erasing	Erasing in an area unit is performed automatically.	
		Automatic block erasing	Erasing in a block unit is performed automatically.	
		Automatic page erasing	Erasing in a page unit is performed automatically.	
	Program/erase protect	Protection	Programming and erasing can be prohibited per block.	
	Security	Security	Prohibition of read-out of the flash memory by a flash writer and the usage restrictions of a debugging function can be carried out.	It becomes effective simultaneously by the operation on the code flash side.
	Execute Instruction	Execute Instruction	Instructions can be executed.	No prefetch buffer
	program/erase to other Flash area	program/erase to code Flash area	Basic operation to a code flash can be performed.	Dual mode

1.1. Memory map

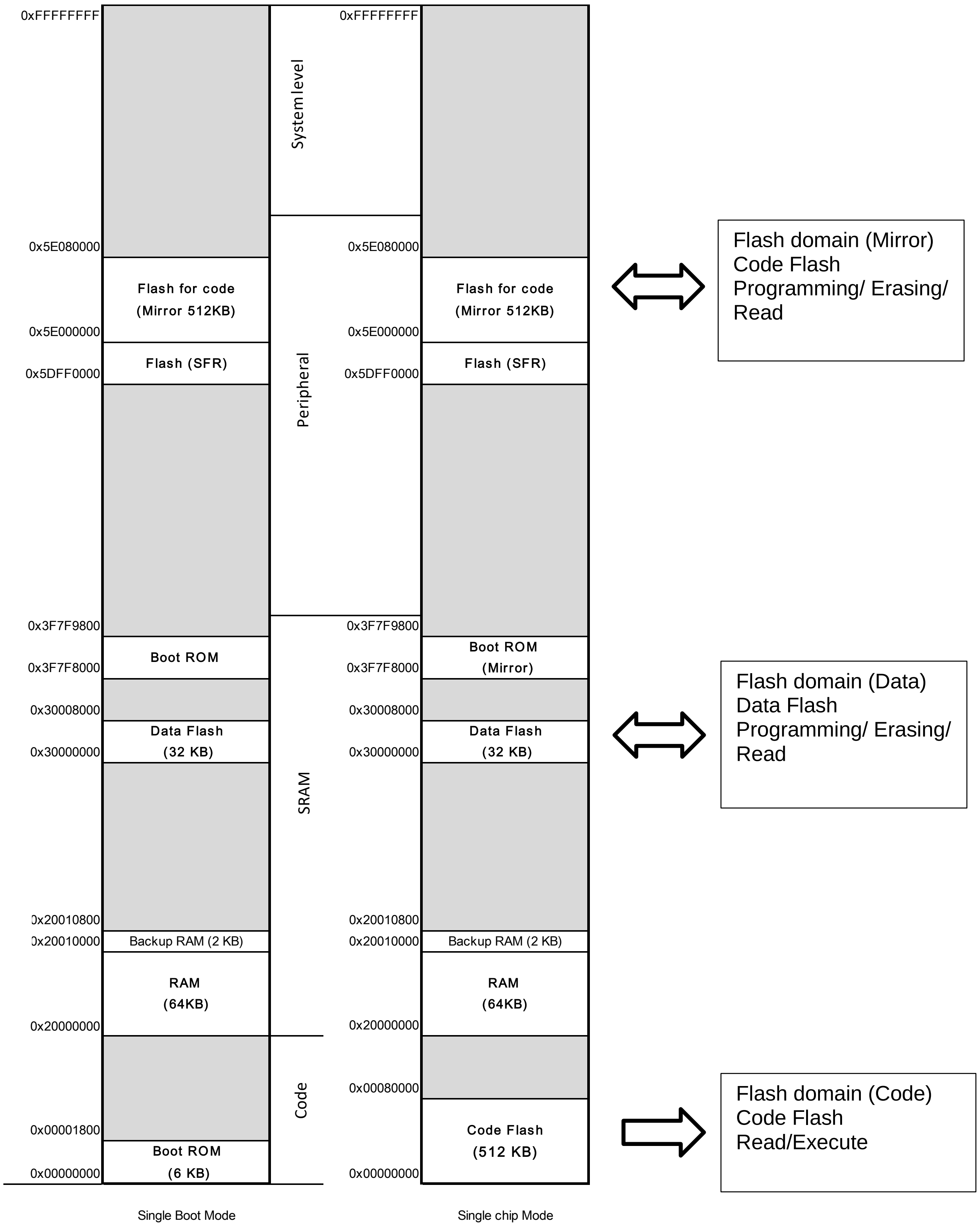


Figure 1.1 The example of a memory map

2. Configuration

2.1. Block Diagrams

The Block Diagrams of a Flash memory and a signal list are shown.

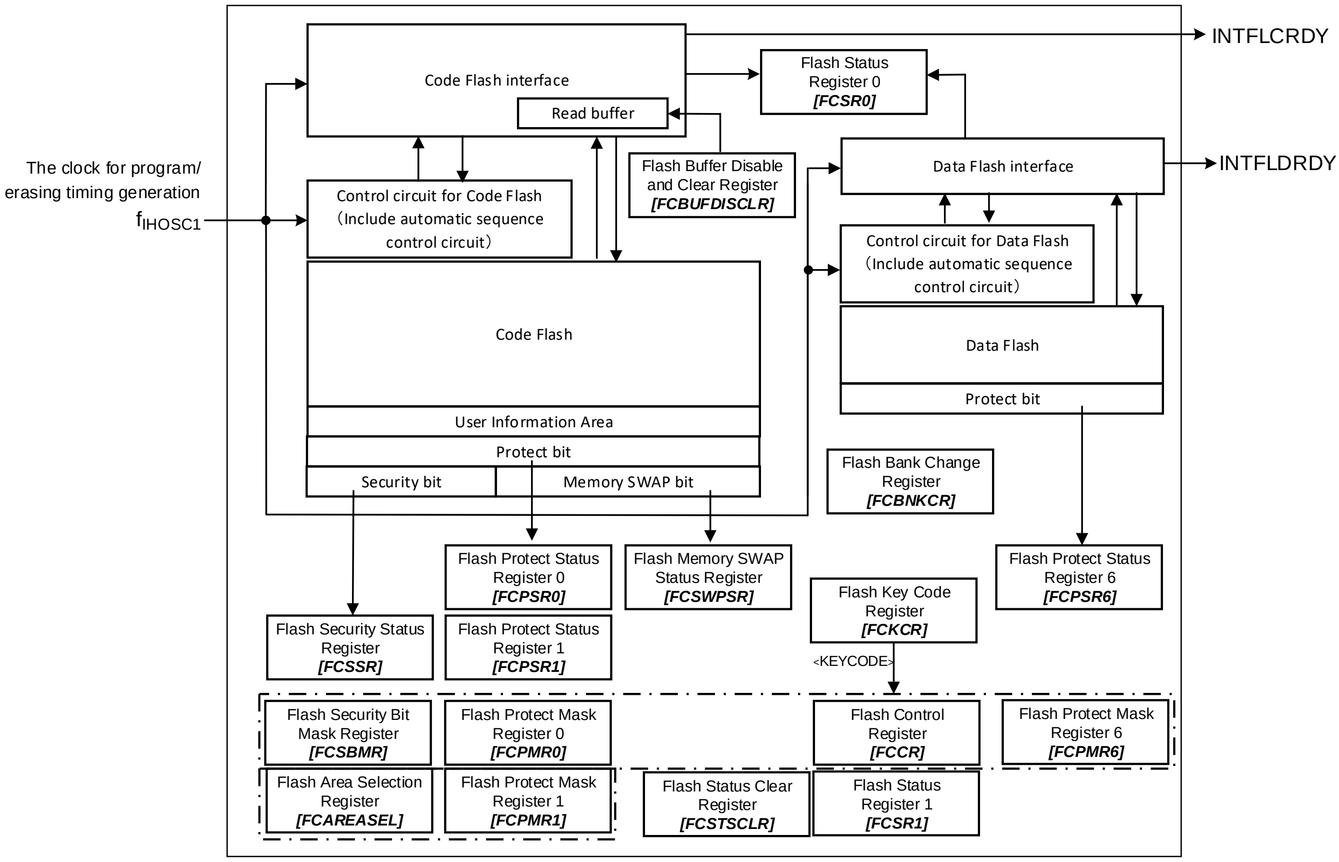


Figure 2.1 The Block Diagrams of a flash memory

Table 2.1 Signal list

No	Symbol	Signal name	I/O	Related reference manual
1	f_{IHOSC1}	The clock for program/erasing timing generation	Input	Clock Control and Operation Mode
2	INTFLCRDY	Code FLASH Ready interruption	Output	Exception
3	INTFLDRDY	Data FLASH Ready interruption	Output	Exception

2.2. Configuration of Code Flash

2.2.1. Unit of the composition

There are "Area", "Block", and "Page" as a unit of the composition of a code flash.

- Area
  It is used by an erase function.
  One area size is a maximum of 512 KB. It changes with memory sizes of a product.

- Block
  It is used by the erase function and a protection function.
  One block size is 32 KB.

- Page
  It is used by the erase function and a protection function.
  One page size is 4096 byte.

2.2.2. Block Configuration

Table 2.2 Block Configuration of 512KB code flash

Area	Block name		Code execution address	Program/erase/read address	Block size (KB)
0	Block0 (Note)	PG0	0x00000000-0x00000FFF	0x5E000000-0x5E000FFF	4
		PG1	0x00001000-0x00001FFF	0x5E001000-0x5E001FFF	4
		PG2	0x00002000-0x00002FFF	0x5E002000-0x5E002FFF	4
		PG3	0x00003000-0x00003FFF	0x5E003000-0x5E003FFF	4
		PG4	0x00004000-0x00004FFF	0x5E004000-0x5E004FFF	4
		PG5	0x00005000-0x00005FFF	0x5E005000-0x5E005FFF	4
		PG6	0x00006000-0x00006FFF	0x5E006000-0x5E006FFF	4
		PG7	0x00007000-0x00007FFF	0x5E007000-0x5E007FFF	4
	Block1		0x00008000-0x0000FFFF	0x5E008000-0x5E00FFFF	32
	Block2		0x00010000-0x00017FFF	0x5E010000-0x5E017FFF	32
	Block3		0x00018000-0x0001FFFF	0x5E018000-0x5E01FFFF	32
	Block4		0x00020000-0x00027FFF	0x5E020000-0x5E027FFF	32
	Block5		0x00028000-0x0002FFFF	0x5E028000-0x5E02FFFF	32
	Block6		0x00030000-0x00037FFF	0x5E030000-0x5E037FFF	32
	Block7		0x00038000-0x0003FFFF	0x5E038000-0x5E03FFFF	32
	Block8		0x00040000-0x00047FFF	0x5E040000-0x5E047FFF	32
	Block9		0x00048000-0x0004FFFF	0x5E048000-0x5E04FFFF	32
	Block10		0x00050000-0x00057FFF	0x5E050000-0x5E057FFF	32
	Block11		0x00058000-0x0005FFFF	0x5E058000-0x5E05FFFF	32
	Block12		0x00060000-0x00067FFF	0x5E060000-0x5E067FFF	32
	Block13		0x00068000-0x0006FFFF	0x5E068000-0x5E06FFFF	32
	Block14		0x00070000-0x00077FFF	0x5E070000-0x5E077FFF	32
	Block15		0x00078000-0x0007FFFF	0x5E078000-0x5E07FFFF	32

Note: Block0 is a generic name for PG0 to PG7 and is accessible as PG0 to PG7.

Table 2.3 Block Configuration of 384KB code flash

Area	Block name		Code execution address	Program/erase/read address	Block size (KB)
0	Block0 (Note)	PG0	0x00000000-0x00000FFF	0x5E000000-0x5E000FFF	4
		PG1	0x00001000-0x00001FFF	0x5E001000-0x5E001FFF	4
		PG2	0x00002000-0x00002FFF	0x5E002000-0x5E002FFF	4
		PG3	0x00003000-0x00003FFF	0x5E003000-0x5E003FFF	4
		PG4	0x00004000-0x00004FFF	0x5E004000-0x5E004FFF	4
		PG5	0x00005000-0x00005FFF	0x5E005000-0x5E005FFF	4
		PG6	0x00006000-0x00006FFF	0x5E006000-0x5E006FFF	4
		PG7	0x00007000-0x00007FFF	0x5E007000-0x5E007FFF	4
	Block1		0x00008000-0x0000FFFF	0x5E008000-0x5E00FFFF	32
	Block2		0x00010000-0x00017FFF	0x5E010000-0x5E017FFF	32
	Block3		0x00018000-0x0001FFFF	0x5E018000-0x5E01FFFF	32
	Block4		0x00020000-0x00027FFF	0x5E020000-0x5E027FFF	32
	Block5		0x00028000-0x0002FFFF	0x5E028000-0x5E02FFFF	32
	Block6		0x00030000-0x00037FFF	0x5E030000-0x5E037FFF	32
	Block7		0x00038000-0x0003FFFF	0x5E038000-0x5E03FFFF	32
	Block8		0x00040000-0x00047FFF	0x5E040000-0x5E047FFF	32
	Block9		0x00048000-0x0004FFFF	0x5E048000-0x5E04FFFF	32
	Block10		0x00050000-0x00057FFF	0x5E050000-0x5E057FFF	32
	Block11		0x00058000-0x0005FFFF	0x5E058000-0x5E05FFFF	32

Note: Block0 is a generic name for PG0 to PG7 and is accessible as PG0 to PG7.

Table 2.4 Block Configuration of 256KB code flash

Area	Block name		Code execution address	Program/erase/read address	Block size (KB)
0	Block0 (Note)	PG0	0x00000000-0x00000FFF	0x5E000000-0x5E000FFF	4
		PG1	0x00001000-0x00001FFF	0x5E001000-0x5E001FFF	4
		PG2	0x00002000-0x00002FFF	0x5E002000-0x5E002FFF	4
		PG3	0x00003000-0x00003FFF	0x5E003000-0x5E003FFF	4
		PG4	0x00004000-0x00004FFF	0x5E004000-0x5E004FFF	4
		PG5	0x00005000-0x00005FFF	0x5E005000-0x5E005FFF	4
		PG6	0x00006000-0x00006FFF	0x5E006000-0x5E006FFF	4
		PG7	0x00007000-0x00007FFF	0x5E007000-0x5E007FFF	4
	Block1		0x00008000-0x0000FFFF	0x5E008000-0x5E00FFFF	32
	Block2		0x00010000-0x00017FFF	0x5E010000-0x5E017FFF	32
	Block3		0x00018000-0x0001FFFF	0x5E018000-0x5E01FFFF	32
	Block4		0x00020000-0x00027FFF	0x5E020000-0x5E027FFF	32
	Block5		0x00028000-0x0002FFFF	0x5E028000-0x5E02FFFF	32
	Block6		0x00030000-0x00037FFF	0x5E030000-0x5E037FFF	32
	Block7		0x00038000-0x0003FFFF	0x5E038000-0x5E03FFFF	32

Note: Block0 is a generic name for PG0 to PG7 and is accessible as PG0 to PG7.

2.2.3. Page Configuration

Table 2.5 shows example of page configuration of 512KB code flash.

Table 2.5 Page Configuration of 512KB code flash (1/4)

Area	Page name	Code execution address	Program/erase/read address	Page size (KB)
0	Page0	0x00000000-0x00000FFF	0x5E000000-0x5E000FFF	4
	Page1	0x00001000-0x00001FFF	0x5E001000-0x5E001FFF	4
	Page2	0x00002000-0x00002FFF	0x5E002000-0x5E002FFF	4
	Page3	0x00003000-0x00003FFF	0x5E003000-0x5E003FFF	4
	Page4	0x00004000-0x00004FFF	0x5E004000-0x5E004FFF	4
	Page5	0x00005000-0x00005FFF	0x5E005000-0x5E005FFF	4
	Page6	0x00006000-0x00006FFF	0x5E006000-0x5E006FFF	4
	Page7	0x00007000-0x00007FFF	0x5E007000-0x5E007FFF	4
	Page8	0x00008000-0x00008FFF	0x5E008000-0x5E008FFF	4
	Page9	0x00009000-0x00009FFF	0x5E009000-0x5E009FFF	4
	Page10	0x0000A000-0x0000AFFF	0x5E00A000-0x5E00AFFF	4
	Page11	0x0000B000-0x0000BFFF	0x5E00B000-0x5E00BFFF	4
	Page12	0x0000C000-0x0000CFFF	0x5E00C000-0x5E00CFFF	4
	Page13	0x0000D000-0x0000DFFF	0x5E00D000-0x5E00DFFF	4
	Page14	0x0000E000-0x0000EFFF	0x5E00E000-0x5E00EFFF	4
	Page15	0x0000F000-0x0000FFFF	0x5E00F000-0x5E00FFFF	4
	Page16	0x00010000-0x00010FFF	0x5E010000-0x5E010FFF	4
	Page17	0x00011000-0x00011FFF	0x5E011000-0x5E011FFF	4
	Page18	0x00012000-0x00012FFF	0x5E012000-0x5E012FFF	4
	Page19	0x00013000-0x00013FFF	0x5E013000-0x5E013FFF	4
	Page20	0x00014000-0x00014FFF	0x5E014000-0x5E014FFF	4
	Page21	0x00015000-0x00015FFF	0x5E015000-0x5E015FFF	4
	Page22	0x00016000-0x00016FFF	0x5E016000-0x5E016FFF	4
	Page23	0x00017000-0x00017FFF	0x5E017000-0x5E017FFF	4
	Page24	0x00018000-0x00018FFF	0x5E018000-0x5E018FFF	4
	Page25	0x00019000-0x00019FFF	0x5E019000-0x5E019FFF	4
	Page26	0x0001A000-0x0001AFFF	0x5E01A000-0x5E01AFFF	4
	Page27	0x0001B000-0x0001BFFF	0x5E01B000-0x5E01BFFF	4
	Page28	0x0001C000-0x0001CFFF	0x5E01C000-0x5E01CFFF	4
	Page29	0x0001D000-0x0001DFFF	0x5E01D000-0x5E01DFFF	4
	Page30	0x0001E000-0x0001EFFF	0x5E01E000-0x5E01EFFF	4
	Page31	0x0001F000-0x0001FFFF	0x5E01F000-0x5E01FFFF	4

Table 2.6 Page Configuration of 512KB code flash (2/4)

Area	Page name	Code execution address	Program/erase/read address	Page size (KB)
0	Page32	0x00020000-0x00020FFF	0x5E020000-0x5E020FFF	4
	Page33	0x00021000-0x00021FFF	0x5E021000-0x5E021FFF	4
	Page34	0x00022000-0x00022FFF	0x5E022000-0x5E022FFF	4
	Page35	0x00023000-0x00023FFF	0x5E023000-0x5E023FFF	4
	Page36	0x00024000-0x00024FFF	0x5E024000-0x5E024FFF	4
	Page37	0x00025000-0x00025FFF	0x5E025000-0x5E025FFF	4
	Page38	0x00026000-0x00026FFF	0x5E026000-0x5E026FFF	4
	Page39	0x00027000-0x00027FFF	0x5E027000-0x5E027FFF	4
	Page40	0x00028000-0x00028FFF	0x5E028000-0x5E028FFF	4
	Page41	0x00029000-0x00029FFF	0x5E029000-0x5E029FFF	4
	Page42	0x0002A000-0x0002AFFF	0x5E02A000-0x5E02AFFF	4
	Page43	0x0002B000-0x0002BFFF	0x5E02B000-0x5E02BFFF	4
	Page44	0x0002C000-0x0002CFFF	0x5E02C000-0x5E02CFFF	4
	Page45	0x0002D000-0x0002DFFF	0x5E02D000-0x5E02DFFF	4
	Page46	0x0002E000-0x0002EFFF	0x5E02E000-0x5E02EFFF	4
	Page47	0x0002F000-0x0002FFFF	0x5E02F000-0x5E02FFFF	4
	Page48	0x00030000-0x00030FFF	0x5E030000-0x5E030FFF	4
	Page49	0x00031000-0x00031FFF	0x5E031000-0x5E031FFF	4
	Page50	0x00032000-0x00032FFF	0x5E032000-0x5E032FFF	4
	Page51	0x00033000-0x00033FFF	0x5E033000-0x5E033FFF	4
	Page52	0x00034000-0x00034FFF	0x5E034000-0x5E034FFF	4
	Page53	0x00035000-0x00035FFF	0x5E035000-0x5E035FFF	4
	Page54	0x00036000-0x00036FFF	0x5E036000-0x5E036FFF	4
	Page55	0x00037000-0x00037FFF	0x5E037000-0x5E037FFF	4
	Page56	0x00038000-0x00038FFF	0x5E038000-0x5E038FFF	4
	Page57	0x00039000-0x00039FFF	0x5E039000-0x5E039FFF	4
	Page58	0x0003A000-0x0003AFFF	0x5E03A000-0x5E03AFFF	4
	Page59	0x0003B000-0x0003BFFF	0x5E03B000-0x5E03BFFF	4
	Page60	0x0003C000-0x0003CFFF	0x5E03C000-0x5E03CFFF	4
	Page61	0x0003D000-0x0003DFFF	0x5E03D000-0x5E03DFFF	4
	Page62	0x0003E000-0x0003EFFF	0x5E03E000-0x5E03EFFF	4
	Page63	0x0003F000-0x0003FFFF	0x5E03F000-0x5E03FFFF	4

Table 2.7 Page Configuration of 512KB code flash (3/4)

Area	Page name	Code execution address	Program/erase/read address	Page size (KB)
0	Page64	0x00040000-0x00040FFF	0x5E040000-0x5E040FFF	4
	Page65	0x00041000-0x00041FFF	0x5E041000-0x5E041FFF	4
	Page66	0x00042000-0x00042FFF	0x5E042000-0x5E042FFF	4
	Page67	0x00043000-0x00043FFF	0x5E043000-0x5E043FFF	4
	Page68	0x00044000-0x00044FFF	0x5E044000-0x5E044FFF	4
	Page69	0x00045000-0x00045FFF	0x5E045000-0x5E045FFF	4
	Page70	0x00046000-0x00046FFF	0x5E046000-0x5E046FFF	4
	Page71	0x00047000-0x00047FFF	0x5E047000-0x5E047FFF	4
	Page72	0x00048000-0x00048FFF	0x5E048000-0x5E048FFF	4
	Page73	0x00049000-0x00049FFF	0x5E049000-0x5E049FFF	4
	Page74	0x0004A000-0x0004AFFF	0x5E04A000-0x5E04AFFF	4
	Page75	0x0004B000-0x0004BFFF	0x5E04B000-0x5E04BFFF	4
	Page76	0x0004C000-0x0004CFFF	0x5E04C000-0x5E04CFFF	4
	Page77	0x0004D000-0x0004DFFF	0x5E04D000-0x5E04DFFF	4
	Page78	0x0004E000-0x0004EFFF	0x5E04E000-0x5E04EFFF	4
	Page79	0x0004F000-0x0004FFFF	0x5E04F000-0x5E04FFFF	4
	Page80	0x00050000-0x00050FFF	0x5E050000-0x5E050FFF	4
	Page81	0x00051000-0x00051FFF	0x5E051000-0x5E051FFF	4
	Page82	0x00052000-0x00052FFF	0x5E052000-0x5E052FFF	4
	Page83	0x00053000-0x00053FFF	0x5E053000-0x5E053FFF	4
	Page84	0x00054000-0x00054FFF	0x5E054000-0x5E054FFF	4
	Page85	0x00055000-0x00055FFF	0x5E055000-0x5E055FFF	4
	Page86	0x00056000-0x00056FFF	0x5E056000-0x5E056FFF	4
	Page87	0x00057000-0x00057FFF	0x5E057000-0x5E057FFF	4
	Page88	0x00058000-0x00058FFF	0x5E058000-0x5E058FFF	4
	Page89	0x00059000-0x00059FFF	0x5E059000-0x5E059FFF	4
	Page90	0x0005A000-0x0005AFFF	0x5E05A000-0x5E05AFFF	4
	Page91	0x0005B000-0x0005BFFF	0x5E05B000-0x5E05BFFF	4
	Page92	0x0005C000-0x0005CFFF	0x5E05C000-0x5E05CFFF	4
	Page93	0x0005D000-0x0005DFFF	0x5E05D000-0x5E05DFFF	4
	Page94	0x0005E000-0x0005EFFF	0x5E05E000-0x5E05EFFF	4
	Page95	0x0005F000-0x0005FFFF	0x5E05F000-0x5E05FFFF	4

Table 2.8 Page Configuration of 512KB code flash (4/4)

Area	Page name	Code execution address	Program/erase/read address	Page size (KB)
0	Page96	0x00060000-0x00060FFF	0x5E060000-0x5E060FFF	4
	Page97	0x00061000-0x00061FFF	0x5E061000-0x5E061FFF	4
	Page98	0x00062000-0x00062FFF	0x5E062000-0x5E062FFF	4
	Page99	0x00063000-0x00063FFF	0x5E063000-0x5E063FFF	4
	Page100	0x00064000-0x00064FFF	0x5E064000-0x5E064FFF	4
	Page101	0x00065000-0x00065FFF	0x5E065000-0x5E065FFF	4
	Page102	0x00066000-0x00066FFF	0x5E066000-0x5E066FFF	4
	Page103	0x00067000-0x00067FFF	0x5E067000-0x5E067FFF	4
	Page104	0x00068000-0x00068FFF	0x5E068000-0x5E068FFF	4
	Page105	0x00069000-0x00069FFF	0x5E069000-0x5E069FFF	4
	Page106	0x0006A000-0x0006AFFF	0x5E06A000-0x5E06AFFF	4
	Page107	0x0006B000-0x0006BFFF	0x5E06B000-0x5E06BFFF	4
	Page108	0x0006C000-0x0006CFFF	0x5E06C000-0x5E06CFFF	4
	Page109	0x0006D000-0x0006DFFF	0x5E06D000-0x5E06DFFF	4
	Page110	0x0006E000-0x0006EFFF	0x5E06E000-0x5E06EFFF	4
	Page111	0x0006F000-0x0006FFFF	0x5E06F000-0x5E06FFFF	4
	Page112	0x00070000-0x00070FFF	0x5E070000-0x5E070FFF	4
	Page113	0x00071000-0x00071FFF	0x5E071000-0x5E071FFF	4
	Page114	0x00072000-0x00072FFF	0x5E072000-0x5E072FFF	4
	Page115	0x00073000-0x00073FFF	0x5E073000-0x5E073FFF	4
	Page116	0x00074000-0x00074FFF	0x5E074000-0x5E074FFF	4
	Page117	0x00075000-0x00075FFF	0x5E075000-0x5E075FFF	4
	Page118	0x00076000-0x00076FFF	0x5E076000-0x5E076FFF	4
	Page119	0x00077000-0x00077FFF	0x5E077000-0x5E077FFF	4
	Page120	0x00078000-0x00078FFF	0x5E078000-0x5E078FFF	4
	Page121	0x00079000-0x00079FFF	0x5E079000-0x5E079FFF	4
	Page122	0x0007A000-0x0007AFFF	0x5E07A000-0x5E07AFFF	4
	Page123	0x0007B000-0x0007BFFF	0x5E07B000-0x5E07BFFF	4
	Page124	0x0007C000-0x0007CFFF	0x5E07C000-0x5E07CFFF	4
	Page125	0x0007D000-0x0007DFFF	0x5E07D000-0x5E07DFFF	4
	Page126	0x0007E000-0x0007EFFF	0x5E07E000-0x5E07EFFF	4
	Page127	0x0007F000-0x0007FFFF	0x5E07F000-0x5E07FFFF	4

2.2.4. User Information Area Configuration of Code Flash

Table 2.9 User Information Area Configuration of Code Flash

Area	User information area	Program/erase/read address	Page size (KB)
0	Page5	0x5E005000-0x5E005FFF	4

2.2.5. Program/Erase Time of Code Flash

Programming is performed in the unit of 16 bytes (4 bytes x 4 times).

Erasing is performed in the unit of Page, Block, Area, or on whole chip. An erase time varies depending on the command to be used. Please refer to “2.2.6. Memory Capacity and the Configuration for detail”.

2.2.6. Memory Capacity and the Configuration

Table 2.10 Memory capacity and the configuration

<table>
<tr><th rowspan="2">Capacity (KB)</th><th colspan="2">Area</th><th colspan="2">Block</th><th colspan="2">Page</th><th colspan="2">Programming time (Note1)</th><th colspan="4">Erasing time (Note1)</th></tr>
<tr><th>Size (KB)</th><th>pcs</th><th>Size (KB)</th><th>pcs</th><th>Size (KB)</th><th>pcs</th><th>Word (Note2)</th><th>Area</th><th>Page</th><th>Block</th><th>Area</th><th>Chip</th></tr>
<tr><td>512</td><td>512</td><td>1</td><td>32</td><td>16</td><td>4</td><td>128</td><td rowspan="3">29.5µs</td><td>3.9s</td><td rowspan="3">1.1ms</td><td rowspan="3">8.6ms</td><td rowspan="3">9.2ms</td><td rowspan="3">22.7ms (Note3)</td></tr>
<tr><td>384</td><td>384</td><td>1</td><td>32</td><td>12</td><td>4</td><td>96</td><td>2.8s</td></tr>
<tr><td>256</td><td>256</td><td>1</td><td>32</td><td>8</td><td>4</td><td>64</td><td>2s</td></tr>
</table>

Note1: The time above-mentioned is for reference only which calculated the Oscillation frequency of IHOSC1 on the standard (10MHz<Typ.>). And indicate the case of the initial value of each register after reset. A data transfer time is excluded.

Note2: Since programming is performed per 4-WORD at one time, it is required four times (above-mentioned).

Note3: It is a case where there is no block of a protection state. The erasing time of a User information area of code flash, data flash, protection bits, and a security bit are included.

2.3. Configuration of Data Flash

2.3.1. Unit of the composition

There are "Area", "Block", and "Page" as a unit of the composition of a data flash.

- Area
  It is used by an erase function.
  One area size is a maximum of 32 KB. It changes with memory sizes of a product.

- Block
  It is used by the erase function and a protection function.
  One block size is 4 KB.

- Page
  It is used by the erase function.
  One page size is 256 byte.

2.3.2. Block Configuration of Data Flash

Table 2.11 Block configuration of 32 KB data flash

Area	Block name	Program/erase/read address	Block size (KB)
4	Block0	0x30000000-0x30000FFF	4
	Block1	0x30001000-0x30001FFF	4
	Block2	0x30002000-0x30002FFF	4
	Block3	0x30003000-0x30003FFF	4
	Block4	0x30004000-0x30004FFF	4
	Block5	0x30005000-0x30005FFF	4
	Block6	0x30006000-0x30006FFF	4
	Block7	0x30007000-0x30007FFF	4

2.3.3. Page Configuration

Table 2.12 shows example page configuration of 32KB data flash.

Table 2.12 Page Configuration of 32KB data flash

Area	Page name	Program/erase/read address	Page size (Byte)
4	Page0	0x30000000-0x300000FF	256
	Page1	0x30000100-0x300001FF	256
	Page2	0x30000200-0x300002FF	256
	Page3	0x30000300-0x300003FF	256
	Page4	0x30000400-0x300004FF	256
	Page5	0x30000500-0x300005FF	256
	Page6	0x30000600-0x300006FF	256
	Page7	0x30000700-0x300007FF	256
	Page8	0x30000800-0x300008FF	256
	Page9	0x30000900-0x300009FF	256
	Page10	0x30000A00-0x30000AFF	256
	Page11	0x30000B00-0x30000BFF	256
	Page12	0x30000C00-0x30000CFF	256
	Page13	0x30000D00-0x30000DFF	256
	Page14	0x30000E00-0x30000EFF	256
	Page15	0x30000F00-0x30000FFF	256
	:	:	:
	Page124	0x30007C00-0x30007CFF	256
	Page125	0x30007D00-0x30007DFF	256
	Page126	0x30007E00-0x30007EFF	256
	Page127	0x30007F00-0x30007FFF	256

2.3.4. **Program/Erase Time of Data Flash**

Programming is performed in the unit of 4 bytes (1 word).

Erasing is performed in the unit of Page, Block, Area, or on entire chip. An erase time varies depending on the command to be used. Please refer to “2.3.5. Memory Capacity and the Configuration” for detail.

2.3.5. **Memory Capacity and the Configuration**

Table 2.13 Memory capacity and the configuration

Capacity (KB)	Area		Block		Page		Programming time (Note)		Erasing time (Note)		
	Size (KB)	pcs	Size (KB)	pcs	Size (Bytes)	pcs	Word	Area	Page	Block	Area
32	32	1	4	8	256	128	64.7μs	531ms	1ms	15.4ms	9.2ms

Note: The time above-mentioned is for reference only which calculated the Oscillation frequency of IHOSC1 on the standard (10MHz<Typ.>). And indicate the case of the initial value of each register after reset. A data transfer time is excluded.

3. Function Description and Functional Explanations

– Code flash and data flash are generally compliant with the JEDEC standards except for some specific functions. Therefore, if a user is currently using a Flash memory as an external memory, it is easy to implement the functions into this device. Furthermore, to provide easy program or erase operation, this flash memory contains a dedicated circuit to perform program or chip erase automatically.

Table 3.1 JEDEC compliant functions

JEDEC compliant functions	Modified, added, or deleted functions
- Automatic programming - Automatic chip erasing - Automatic block erasing	<Addition> Automatic area erasing, automatic page erasing, automatic memory swap/erasing <Modified> Program/erase protect (only protection of program is supported) <Deleted> Erase resume/suspend function

Precautions

(1) Make sure to set ***[CGOSCCR]***<IHOSCEN1>=1 to oscillate the internal high speed oscillator1 (IHOSC1) when data is programmed or erased code flash, data flash, user information area. Also oscillate the IHOSC1 before the operations related to the flash memory including protection and security operations. After the IHOSC1 is oscillated, confirm whether ***[CGOSCCR]***<IHOSC1F>=1 before using the flash memory. IHOSC1 is timing clock for programming/Erasing of flash memory.

(2) Please refer to reference Manual “Clock Control and Operation Mode” about IHOSC1 and ***[CGOSCCR]***<IHOSC1F>.

(3) Do not power off during Flash is busy (Programming or Erasing, ***[FCSR0]***<RDYBSY>=0).

(4) Do not enter STOP1/STOP2 mode during Flash is busy (Programming or Erasing, ***[FCSR0]***<RDYBSY>=0).

(5) Avoid reset that is occurred by SIWDT or LVD during Flash is busy (Programming or Erasing, ***[FCSR0]***<RDYBSY>=0).

3.1. Code Flash

3.1.1. Command Sequence

3.1.1.1. List of Command Sequence

This section shows addresses and data of the bus write cycle in each command of code flash.

Except the 5th bus cycle of ID-Read command, all cycles are “bus write cycles”. A bus write cycle is performed by a 32-bit (1 word) data transfer instruction. “Table 3.2 Flash memory access using the internal CPU (code flash)” only shows the lower 8 bits data.

For details of addresses, refer to “Table 3.3 Address bit configuration in the bus write cycle (Code flash)”. Use the values in the table below to Addr[11:4] where “Command” is inputted.

Note: Each command address is set to a flash area (mirror).

Table 3.2 Flash memory access using the internal CPU (code flash)

Sequence / Command	1st bus cycle	2nd bus cycle	3rd bus cycle	4th bus cycle	5th bus cycle	6th bus cycle	7th bus cycle
	Address	Address	Address	Address	Address	Address	Address
	Data	Data	Data	Data	Data	Data	Data
Read/Reset	0xYYYYXXXX	-	-	-	-	-	-
	0xF0	-	-	-	-	-	-
ID-Read	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	IA	0xYYYYXXXX	-	-
	0xAA	0x55	0x90	0x00	ID	-	-
Automatic programming	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	PA	PA	PA	PA
	0xAA	0x55	0xA0	PD0	PD1	PD2	PD3
Automatic page erasing	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	PGA	-
	0xAA	0x55	0x80	0xAA	0x55	0x40	-
Automatic block erasing	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	BA	-
	0xAA	0x55	0x80	0xAA	0x55	0x30	-
Automatic area erasing	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	AA	-
	0xAA	0x55	0x80	0xAA	0x55	0x20	-
Automatic code area erasing	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	-
	0xAA	0x55	0x80	0xAA	0x55	0x11	-
Automatic chip erasing	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	-
	0xAA	0x55	0x80	0xAA	0x55	0x10	-
Automatic protect bit programming	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	PBA	-	-	-
	0xAA	0x55	0x9A	0x9A	-	-	-
Automatic protect bit erasing	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	PBA(Note)	-
	0xAA	0x55	0x80	0xAA	0x55	0x60	-
Automatic	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	MSA	-	-	-

Command \ Sequence	1st bus cycle	2nd bus cycle	3rd bus cycle	4th bus cycle	5th bus cycle	6th bus cycle	7th bus cycle
	Address	Address	Address	Address	Address	Address	Address
	Data	Data	Data	Data	Data	Data	Data
memory swap programming	0xAA	0x55	0x9A	0x9A	-	-	-
Automatic memory swap erasing	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	MSA(Note)	-
	0xAA	0x55	0x80	0xAA	0x55	0x60	-
Automatic security bit programming	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	SBA	-	-	-
	0xAA	0x55	0x9A	0x9A	-	-	-
Automatic security bit erasing	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	SBA(Note)	-
	0xAA	0x55	0x80	0xAA	0x55	0x60	-

Note: Please refer to “Table 3.3 Address bit configuration in the bus write cycle (Code flash)”.

Supplementary explanation
IA: ID address
ID: ID data output
PGA: Page address
BA: Block address
AA: Area address
PA: Program address (write)
PD: Program data (32-bit data)
After the 4th bus cycle, 4 word data are sequentially input in address order.
PBA: Protect bit address
MSA: Memory swap address
SBA: Security bit address

3.1.1.2. Address Bit Configuration in the Bus Write Cycle (Code Flash)

Please refer to “Table 3.3 Address bit configuration in the bus write cycle (Code flash)” with “Table 3.2 Flash memory access using the internal CPU (code flash)”.

Specify addresses in the first bus cycle and later cycle based on address setting of bus write cycle of normal command.

Table 3.3 Address bit configuration in the bus write cycle (Code flash)

[Normal command]

Address	Adr [31:24]	Adr [23:19]	Adr [18:17]	Adr [16:12]	Adr [11:4]	Adr [3:0]
Normal command	Address setting of bus write cycle of normal command					
	0x5E	“00000” fixed	Area (Note)	"0” Recommended	Command	"0” Recommended

Note: Use “Area” fixed to “00”.

[Read/reset, ID-Read]

Address	Adr [31:24]	Adr [23:19]	Adr [18:16]	Adr [15:14]	Adr [13:0]
Read /reset	Address setting of 1st bus write cycle of Read/reset				
	0x5E	“00000” fixed	"0” Recommended		
ID-Read	IA: ID address (address setting of the 4th bus write cycle of ID-Read)				
	0x5E	“00000” fixed	"000” fixed	ID address	"0” Recommended

[Automatic area erasing]

Address	Adr [31:24]	Adr [23:19]	Adr [18:17]	Adr [16:0]
Area erasing	AA: Area Address (address setting of the 6th bus write cycle of area erase command)			
	0x5E	“00000” fixed	Area (Note)	"0” Recommended

Note: Use “Area” fixed to “00”.

[Automatic block erasing]

Address	Adr [31:24]	Adr [23:19]	Adr [18:17]	Adr [16:15]	Adr [14:0]
Block erasing	BA: Block address (address setting of the 6th bus write cycle of block erasing command)				
	0x5E	“00000” fixed	Area (Note)	Block address	"0” Recommended

Note: Use “Area” fixed to “00”.

[Automatic page erasing]

Address	Adr [31:24]	Adr [23:19]	Adr [18:17]	Adr [16:12]	Adr [11:0]
Page erasing	PGA: Page Address (address setting of the 6th bus write cycle of page erasing command)				
	0x5E	"00000" fixed	Area (Note)	Page address	"0" Recommended

Note: Use "Area" fixed to "00".

[Automatic programming]

Address	Adr [31:24]	Adr [23:19]	Adr [18:17]	Adr [16:4]	Adr [3:0]
Program	PA: Program address (address setting of the 4th to 7th bus write cycle of the program)				
	0x5E	"00000" fixed	Area (Note)	Program address	"0" Recommended

Note: Use "Area" fixed to "00".

[Automatic protect bit programming/erasing]

Address	Adr [31:24]	Adr [23:19]	Adr [18:17]	Adr [16:12]	Adr [11:4]	Adr [3:0]
Protect bit erasing	PBA: Protect Bit Address (address setting of the 6th bus write cycle of Protect bit erasing)					
	0x5E	"00000" fixed	Area (Note)	"00010" fixed	"0" Recommended	
Protect bit programming	PBA: Protect Bit Address (address setting of the 4th bus write cycle of Protect bit programming)					
	0x5E	"00000" fixed	Area (Note)	"00010" fixed	Protect bit selection	"0" Recommended

Note: Use "Area" fixed to "00".

[Automatic memory swap erasing/programming]

Address	Adr [31:24]	Adr [23:19]	Adr [18:17]	Adr [16:12]	Adr [11:4]	Adr [3:0]
Memory swap erasing	MSA: Memory Swap Address (address setting of the 6th bus write cycle of memory swap erasing)					
	0x5E	"00000" fixed	"00" fixed	"00011" fixed	"0" Recommended	
Memory swap programming	MSA: Memory Swap Address (address setting of the 4th bus write cycle of memory swap programming)					
	0x5E	"00000" fixed	"00" fixed	"00011" fixed	Memory swap selection	"0" Recommended

[Automatic security bit programming/erasing]

Address	Adr [31:24]	Adr [23:19]	Adr [18:17]	Adr [16:12]	Adr [11:0]
Security bit erasing	SBA: Security Bit Address (address of the 6th bus write cycle of security bit erasing)				
	0x5E	"00000" fixed	"00" fixed	"00001" fixed	"0" Recommend
Security bit programming	SBA: Security Bit Address (address of the 4th bus write cycle of security bit programming)				
	0x5E	"00000" fixed	"00" fixed	"00001" fixed	"0" Recommended

3.1.1.3. Area Address (AA), Block Address (BA): Code Flash

"Table 2.2 Block Configuration of 512KB code flash" shows area addresses and block addresses. An address of the area or block to be erased should be specified in the 6th bus write cycle of automatic area erasing command and automatic block erasing command. In single chip mode, an address of the mirror area should be specified.

3.1.1.4. Protect Bit Assignment (PBA): Code flash

A protect bit can be controlled in the unit of one bit.
Table 3.4 shows the protect bit selection of the automatic protect bit programming.

Table 3.4 Protect bit programming address

Block	Page	Register	Protect bit	PBA[11:4]						Example of address [31:0]
				Adr [11:9]	Adr [8]	Adr [7]	Adr [6]	Adr [5]	Adr [4]	
0 (Note)	0	*[FCPSR0]*	<PG0>	000	0	0	0	0	0	0x5E002000
	1		<PG1>	000	0	0	0	0	1	0x5E002010
	2		<PG2>	000	0	0	0	1	0	0x5E002020
	3		<PG3>	000	0	0	0	1	1	0x5E002030
	4		<PG4>	000	0	0	1	0	0	0x5E002040
	5		<PG5>	000	0	0	1	0	1	0x5E002050
	6		<PG6>	000	0	0	1	1	0	0x5E002060
	7		<PG7>	000	0	0	1	1	1	0x5E002070
1	8 to 15	*[FCPSR1]*	<BLK1>	000	0	1	0	0	0	0x5E002080
2	16 to 23		<BLK2>	000	0	1	0	0	1	0x5E002090
3	24 to 31		<BLK3>	000	0	1	0	1	0	0x5E0020A0
4	32 to 39		<BLK4>	000	0	1	0	1	1	0x5E0020B0
5	40 to 47		<BLK5>	000	0	1	1	0	0	0x5E0020C0
6	48 to 55		<BLK6>	000	0	1	1	0	1	0x5E0020D0
7	56 to 63		<BLK7>	000	0	1	1	1	0	0x5E0020E0
8	64 to 71		<BLK8>	000	0	1	1	1	1	0x5E0020F0
9	72 to 79		<BLK9>	000	1	0	0	0	0	0x5E002100
10	80 to 87		<BLK10>	000	1	0	0	0	1	0x5E002110
11	88 to 95		<BLK11>	000	1	0	0	1	0	0x5E002120
12	96 to 103		<BLK12>	000	1	0	0	1	1	0x5E002130
13	104 to 111		<BLK13>	000	1	0	1	0	0	0x5E002140
14	112 to 119		<BLK14>	000	1	0	1	0	1	0x5E002150
15	120 to 127		<BLK15>	000	1	0	1	1	0	0x5E002160

Note: Block0 is a generic name for PG0 to PG7.

3.1.1.5. **ID-Read Code (IA, ID): Code Flash**

"Table 3.5 ID-Read Command code assignment and the code contents" shows the code assignment and the contents of ID-Read command.

Table 3.5 ID-Read Command code assignment and the code contents

Code	ID[15:0]	IA[15:14]	Example of address [31:0]
Manufacture code	0x0098	00	0x5E000000
Device code	0x005A	01	0x5E004000
-	Reserved	10	N/A
Macro code	0x020F	11	0x5E00C000

3.1.1.6. **Memory Swap Bit Assignment (MSA)**

"Table 3.6 Setting values assigned to ***[FCSWPSR]*** using Memory Swap command, and example of address" shows the setting values of ***[FCSWPSR]***<SWP[1:0]><SIZE[5:0]> assigned in the 4th bus write cycle of the auto memory swap command.

Table 3.6 Setting values assigned to *[FCSWPSR]* using Memory Swap command, and example of address

Register		MSA[11:4]						Example of address [31:0]
		Adr [11:9]	Adr [8]	Adr [7]	Adr [6]	Adr [5]	Adr [4]	
[FCSWPSR]	<SWP[0]>	000	0	0	0	0	0	0x5E003000
	<SWP[1]>	000	0	0	0	0	1	0x5E003010
	<SIZE[0]>	000	0	0	0	1	0	0x5E003020
	<SIZE[1]>	000	0	0	0	1	1	0x5E003030
	<SIZE[2]>	000	0	0	1	0	0	0x5E003040
	<SIZE[3]>	000	0	0	1	0	1	0x5E003050
	<SIZE[4]>	000	0	0	1	1	0	0x5E003060
	<SIZE[5]>	000	0	0	1	1	1	0x5E003070

3.2. Data Flash

3.2.1. Command Sequence

3.2.1.1. List of Command Sequence

This section shows addresses and data of the bus write cycle in each command of data flash.

Except the 5th bus cycle of ID-Read command, all cycles are "bus write cycles". A bus write cycle is performed by a 32-bit (1 word) data transfer instruction. "Table 3.7 Command sequence (Data flash)" Table 3.8 Address bit configuration in the bus write cycle (data flash) only shows the lower 8 bits data.

For details of addresses, refer to "Table 3.8 Address bit configuration in the bus write cycle (data flash)". Use the values in the table below to Addr[11:4] where "Command" is inputted.

Note: Each command address is set to a flash area (data).

Table 3.7 Command sequence (Data flash)

Command \ Sequence		1st bus cycle	2nd bus cycle	3rd bus cycle	4th bus cycle	5th bus cycle	6th bus cycle	7th bus cycle
		Address	Address	Address	Address	Address	Address	Address
		Data	Data	Data	Data	Data	Data	Data
Read/reset	Address	0xYYYYXXXX	-	-	-	-	-	-
	Data	0xF0	-	-	-	-	-	-
ID-Read	Address	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	IA	0xYYYYXXXX	-	-
	Data	0xAA	0x55	0x90	0x00	ID	-	-
Automatic programming	Address	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	PA	-	-	-
	Data	0xAA	0x55	0xC0	PD0	-	-	-
Automatic page erasing	Address	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	PGA	-
	Data	0xAA	0x55	0x80	0xAA	0x55	0x40	-
Automatic block erasing	Address	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	BA	-
	Data	0xAA	0x55	0x80	0xAA	0x55	0x30	-
Automatic area erasing	Address	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	AA	-
	Data	0xAA	0x55	0x80	0xAA	0x55	0x20	-
Automatic Protect bit programming	Address	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	PBA	-	-	-
	Data	0xAA	0x55	0x9A	0x9A	-	-	-
Automatic Protect bit erasing	Address	0xYYYYX55X	0xYYYYXAAX	0xYYYYX55X	0xYYYYX55X	0xYYYYXAAX	PBA(Note)	-
	Data	0xAA	0x55	0x80	0xAA	0x55	0x60	-

Note: Refer to "Table 3.8 Address bit configuration in the bus write cycle (data flash)" in Bus Write Cycle.

Supplementary explanation
IA: ID address
ID: ID data output
PGA: Page address
BA: Block address
AA: Area address
PA: Program address (write)
PD: Program data (32-bit data)
PBA: Protect bit address

3.2.1.2. Address Configuration in the Bus Write Cycle (Data Flash)

Please refer to "Table 3.8 Address bit configuration in the bus write cycle (data flash)" with "Table 3.7 Command sequence (Data flash)".

Specify addresses in the first bus cycle and later cycle, based on address setting of bus write cycle of normal command.

Table 3.8 Address bit configuration in the bus write cycle (data flash)

[Normal command]

Address	Adr [31:24]	Adr [23:16]	Adr [15]	Adr [14:12]	Adr [11:4]	Adr [3:0]
Normal command	Address setting of bus write cycle of normal command					
	0x30	"00000000" fixed	Area (Note)	"0" Recommended	Command	"0" Recommended

Note: Use "Area" fixed to "0".

[Read/reset, ID-Read]

Address	Adr [31:24]	Adr [23:16]	Adr [15]	Adr [14:13]	Adr [12:0]
Read /reset	Address setting of 1st bus write cycle of read/reset				
	0x30	"00000000" fixed	"0" Recommended		
ID-Read	IA: ID Address (address setting of 4th bus write cycle of ID-Read)				
	0x30	"00000000" fixed	"0" fixed	ID address	"0" Recommended

[Automatic area erasing]

Address	Adr [31:24]	Adr [23:16]	Adr [15]	Adr [14:0]
Area erasing	AA: Area Address (address setting of 6th bus write cycle of area erasing command)			
	0x30	"00000000" fixed	Area (Note)	"0" Recommended

Note: Use "Area" fixed to "0".

[Automatic block erasing]

Address	Adr [31:24]	Adr [23:16]	Adr [15]	Adr [14:12]	Adr [11:0]
Block erasing	BA: Block Address (address setting of 6th bus write cycle of block erasing command)				
	0x30	“00000000” fixed	Area (Note)	Block address	"0” Recommended

Note: Use “Area” fixed to “0”.

[Automatic page erasing]

Address	Adr [31:24]	Adr [23:16]	Adr [15]	Adr [14:8]	Adr [7:0]
Page erasing	PGA: Page Address (address setting of 6th bus write cycle of page erasing command)				
	0x30	“00000000” fixed	Area (Note)	Page address	"0” Recommended

Note: Use “Area” fixed to “0”.

[Automatic programming]

Address	Adr [31:24]	Adr [23:16]	Adr [15]	Adr [14:2]	Adr [1:0]
Program	PA: Program Address (address setting of 4th bus write cycle of programming command)				
	0x30	“00000000” fixed	Area (Note)	Program address	"0” Recommended

Note: Use “Area” fixed to “0”.

[Automatic protect bit programming/erasing]

Address	Adr [31:24]	Adr [23:16]	Adr [15]	Adr [14:8]	Adr [7:2]	Adr [1:0]
Protect bit erasing	PBA: Protect Bit Address (address setting of 6th bus write cycle of protect bit erasing command)					
	0x30	“00000000” fixed	“0” fixed	“0000001” fixed	"0” Recommended	
Protect bit programming	PBA: Protect Bit Address (address setting of 4th bus write cycle of protect bit programming command)					
	0x30	“00000000” fixed	“0” fixed	“0000001” fixed	Protect bit selection	"0” Recommended

3.2.1.3. Area Address (AA), Block Address (BA)

“Table 2.11 Block configuration of 32 KB data flash” shows area addresses and block addresses. An address of the area or block to be erased should be specified in the 6th bus write cycle of automatic area erasing command and automatic block erasing command.

3.2.1.4. **Protect Bit Assignment (PBA)**

A protect bit can be controlled in the unit of one bit.

"Table 3.9 Protect bit program address (Data flash)" shows the protect bit selection of the automatic protect bit program.

Table 3.9 Protect bit program address (Data flash)

Block	Register	Protect bit	PBA[7:2]					Example of address [31:0]
			Adr [7:6]	Adr [5]	Adr [4]	Adr [3]	Adr [2]	
0	*[FCPSR6]*	<DBLK0>	00	0	0	0	0	0x30000100
1		<DBLK1>	00	0	0	0	1	0x30000104
2		<DBLK2>	00	0	0	1	0	0x30000108
3		<DBLK3>	00	0	0	1	1	0x3000010C
4		<DBLK4>	00	0	1	0	0	0x30000110
5		<DBLK5>	00	0	1	0	1	0x30000114
6		<DBLK6>	00	0	1	1	0	0x30000118
7		<DBLK7>	00	0	1	1	1	0x3000011C

3.2.1.5. **ID-Read Code (IA, ID): Data Flash**

"Table 3.10 ID-Read command code assignment and the contents (Data flash)" shows the code assignment and the contents of ID-Read command.

Table 3.10 ID-Read command code assignment and the contents (Data flash)

Code	ID[15:0]	IA[14:13]	Example of address [31:0]
Manufacture code	0x0098	00	0x30000000
Device code	0x005A	01	0x30002000
-	Reserved	10	N/A
Macro code	0x0240	11	0x30006000

3.3. Flowchart

This section shows examples of code flash programming.

3.3.1. Automatic Programming

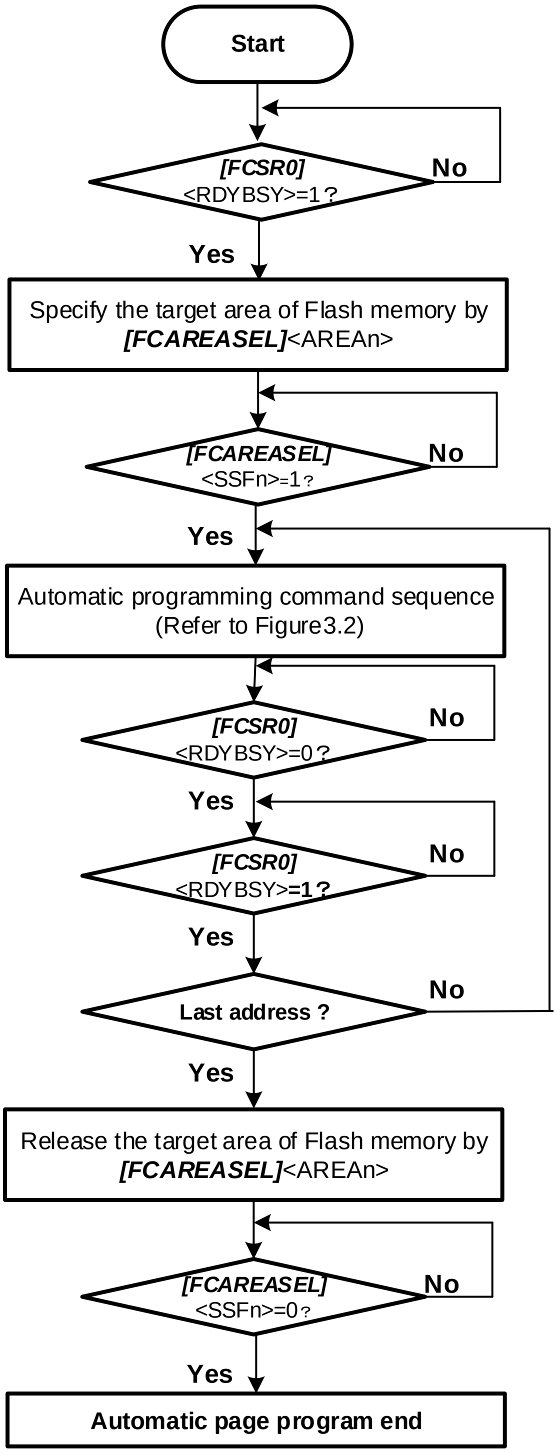


Figure 3.1 Flowchart of automatic programming (1)

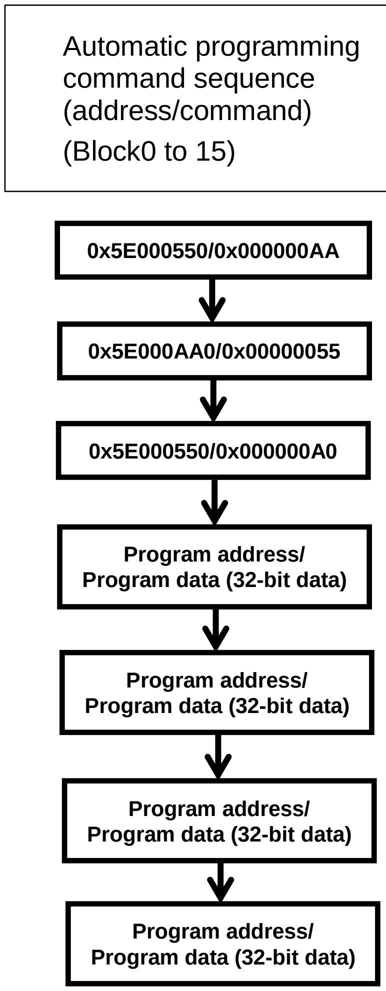


Figure 3.2 Flowchart of automatic programming (2)

3.3.2. **Automatic Erasing**

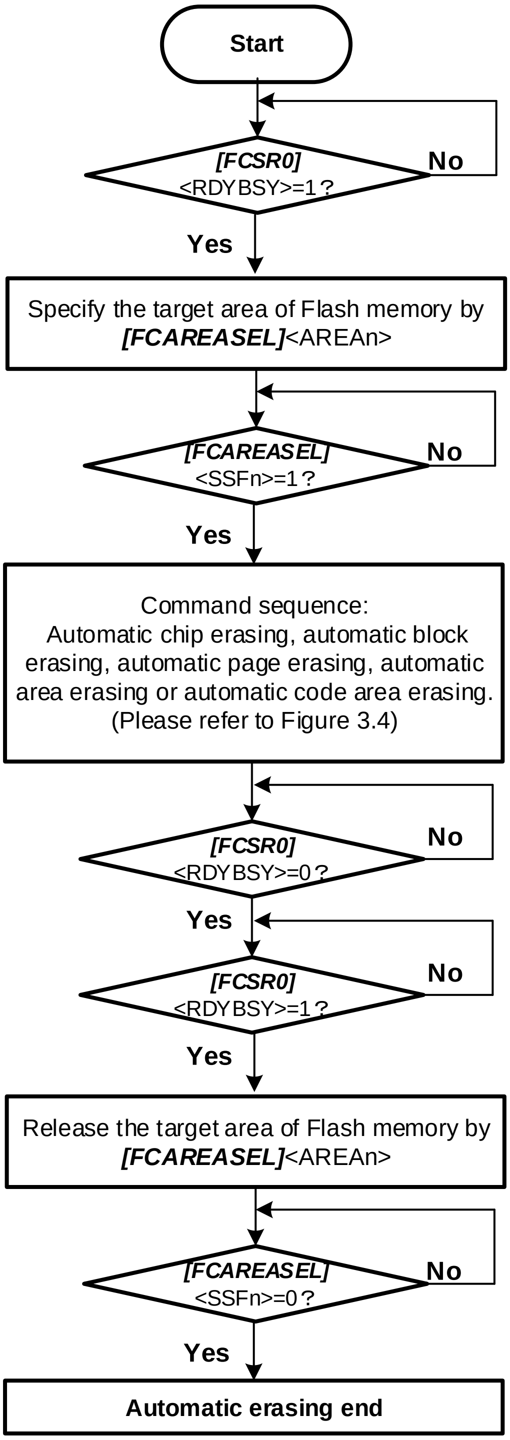


Figure 3.3 Flowchart of automatic erasing (1)

Note: Please perform blank check to confirm data was erased after automatic erasing.

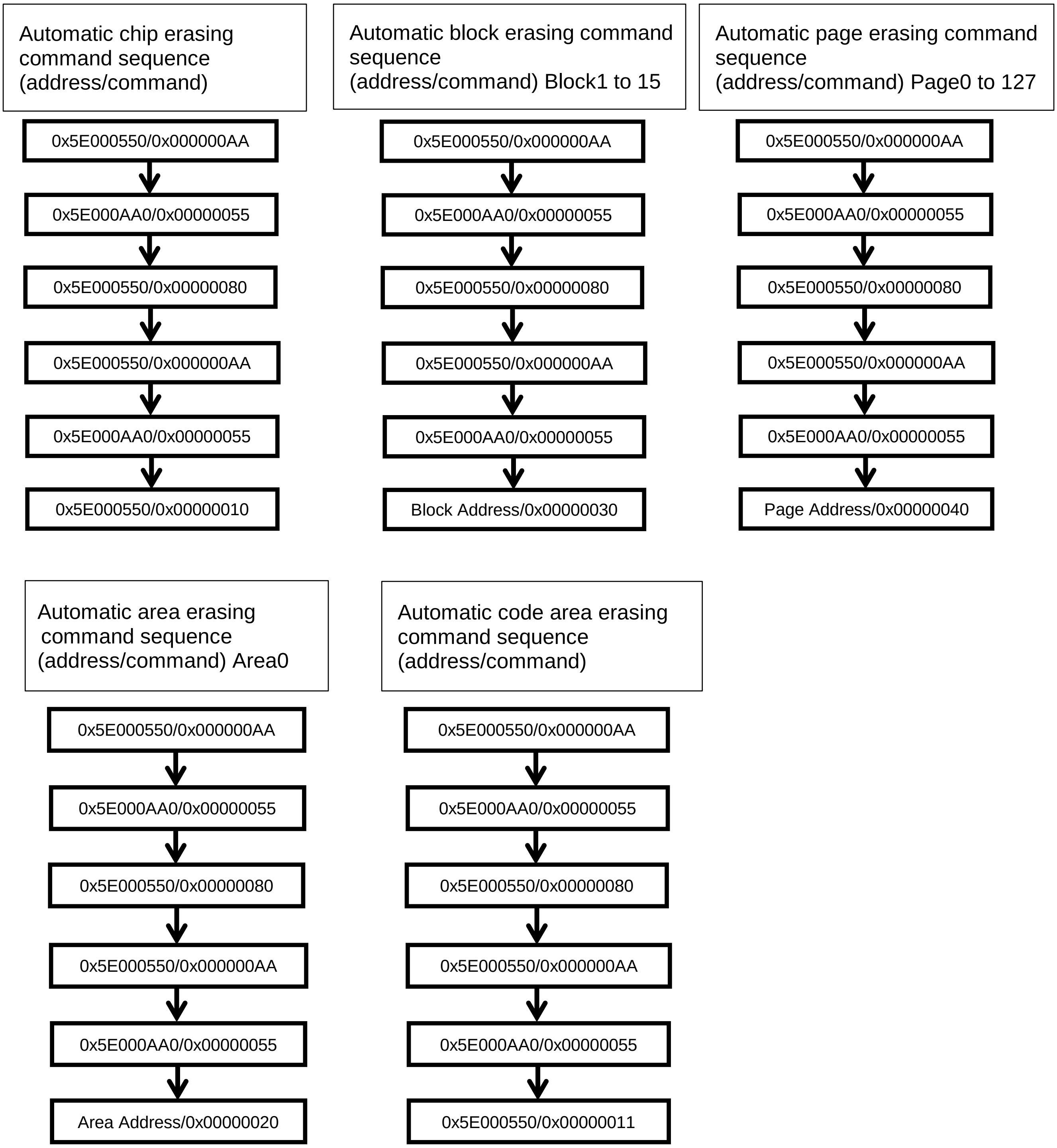


Figure 3.4 flowchart of automatic erasing (2)

3.3.3. Protect bit

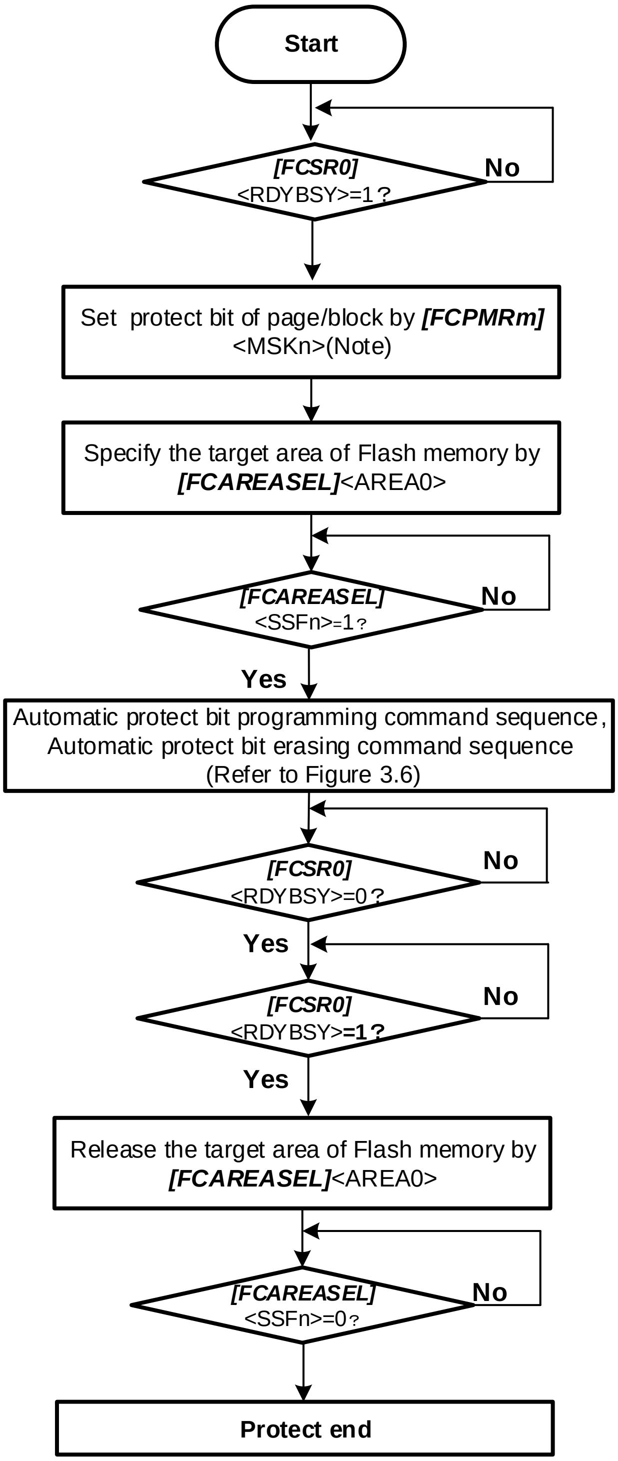


Figure 3.5 Flowchart of protect (1)

Note: <MSKn> represents <PMn>, <MSKn>, and <DMSKn>.

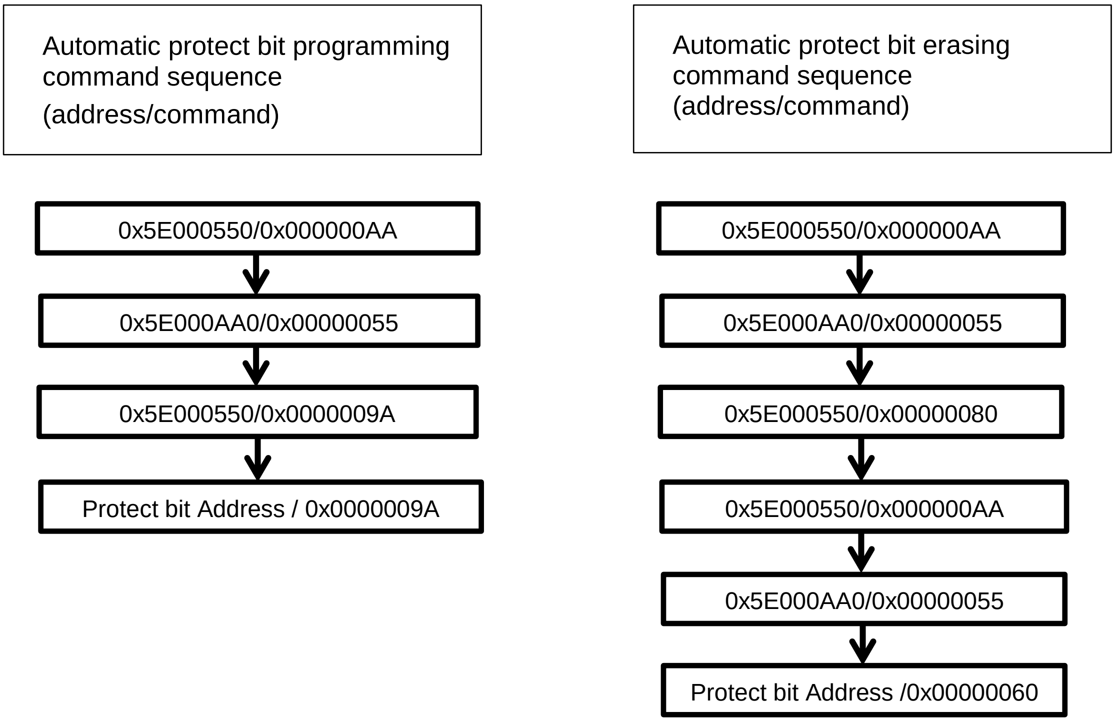


Figure 3.6 Flowchart of protect (2)

3.3.4. Security bit

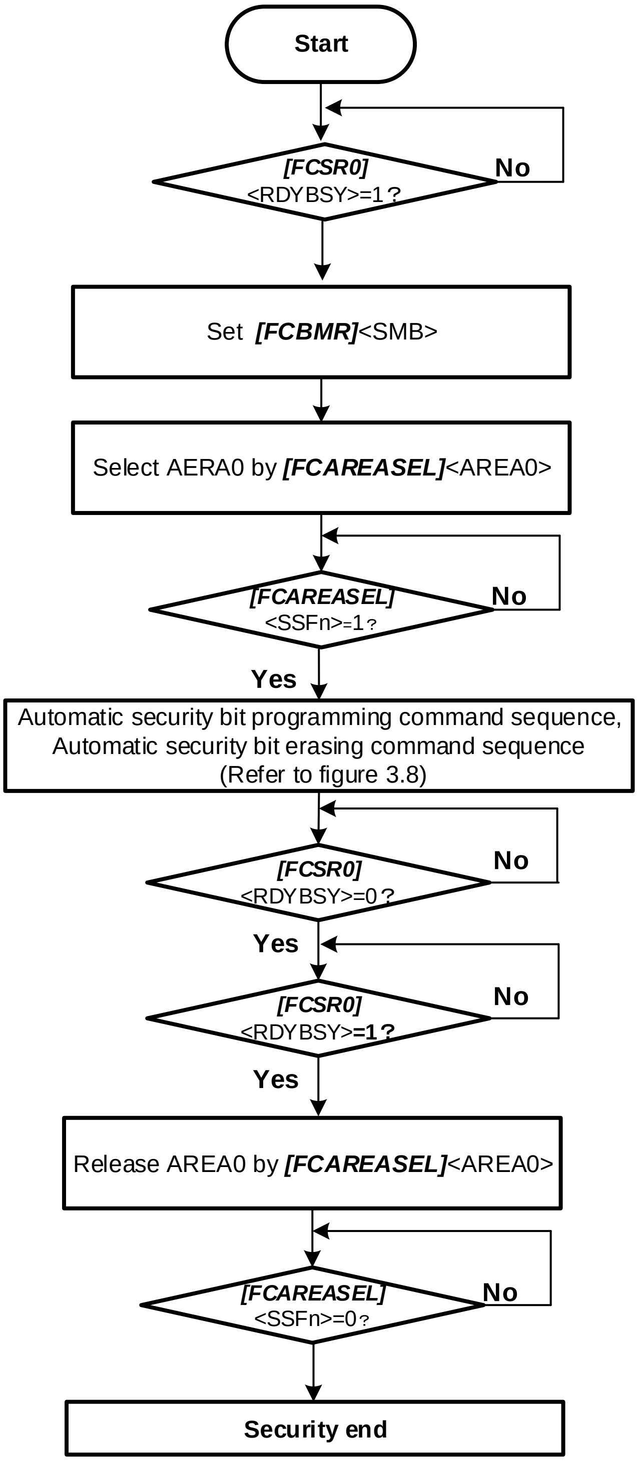


Figure 3.7 Flowchart of security (1)

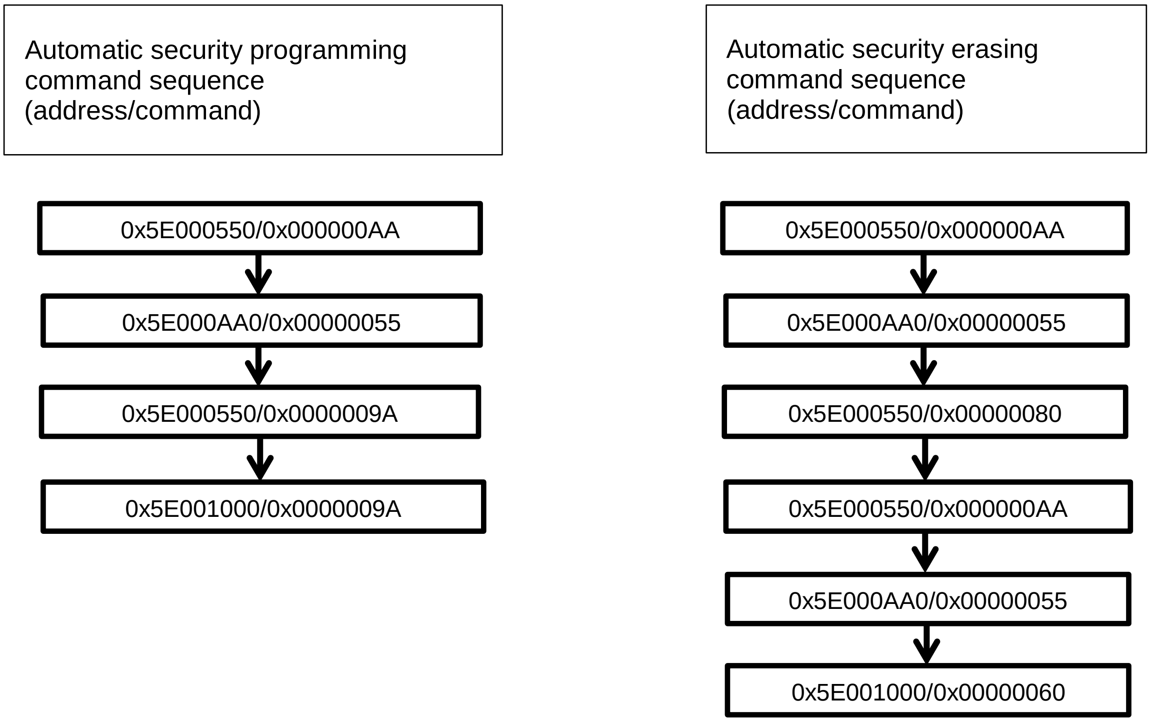


Figure 3.8 Flowchart of security (2)

3.3.5. Memory Swap

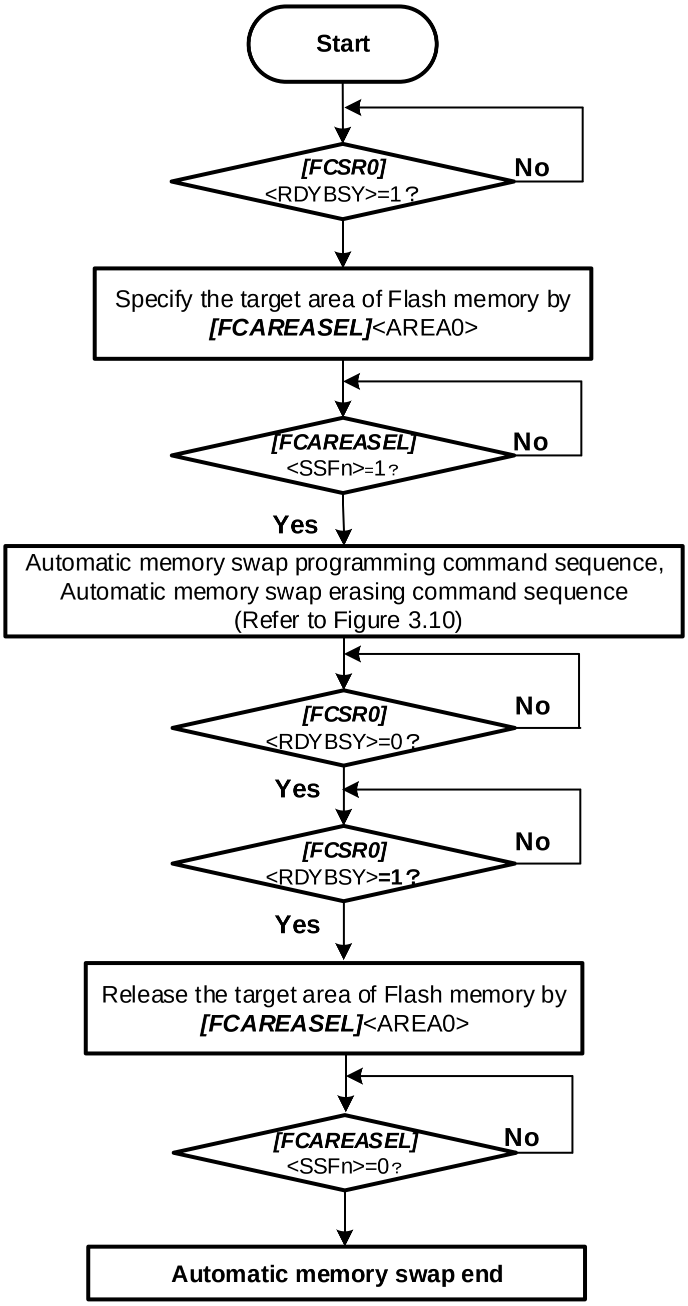


Figure 3.9 Flowchart of memory swap (1)

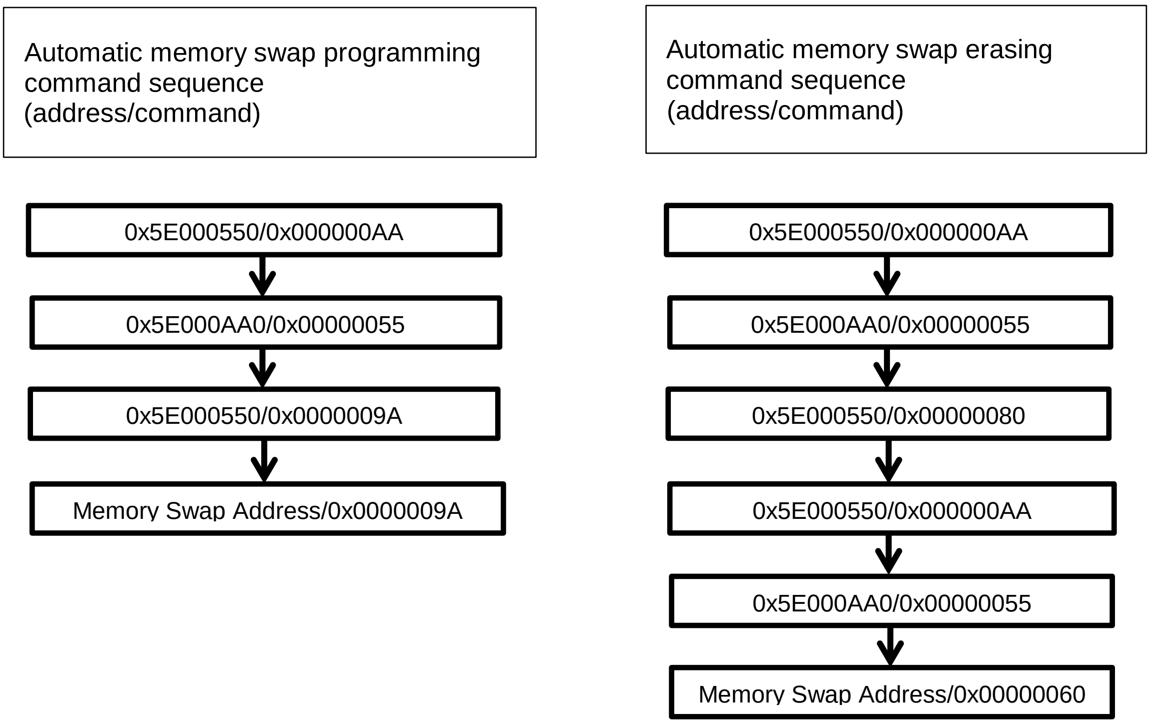


Figure 3.10 Flowchart of memory swap (2)

4. Details of Flash Memory

Flash memory is Programmed/erased data by executing a command in the control program. This programming/ erasing control program must be prepared by users in advance.

While a programming is being executed on the flash memory of area 0, flash memory of another area where the instruction is not being executed (ex. Area 4: data flash) can be programmed/erased data, and vice versa can be possible. This usage is called “dual mode” in this document.

4.1. Functions

Flash memory programming and erasing operation are generally compliant with the JEDEC standards commands except for some specific functions; however address assignment of an operational command is different from standard commands.

When programming/erasing operation is performed, a command is input to the flash memory with 32-bit (one word) store instruction. After the command is input, program or erase operation is internally automatically performed.

Table 4.1 Flash memory function

Main functions	Description
Automatic programming	Code flash: Program data in 4 word unit (16 bytes) automatically. Data flash: Program data in 1 word unit (4 bytes) automatically.
Automatic chip erasing	Erases the entire flash memory at one time automatically.(Note1)
Automatic area erasing	Erases the flash memory in the unit of the area automatically.
Automatic block erasing	Erases the flash memory in the unit of the block automatically.(Note2)
Automatic page erasing	Erases the flash memory in the unit of the page automatically.
Automatic protect bit programming / erasing	Protects the flash memory from data program and erase operation.
Automatic security bit programming /erasing	Security setting to the flash memory and release security operation.
Automatic memory swap/erasing	Specifies memory swap, memory swap release, or swap size of the code flash area automatically.

Note1: Except user information area.
Note2: Block0 of code flash cannot be erased by one time. Please erase for every page by automatic page erasing command.

4.1.1. Operation Mode of the Flash Memory

The flash memory has three main operation modes:

- Read the memory data (Read mode)
- Input command for erasing/programming (Command sequence input mode)
- Erase/program data automatically (Automatic operation mode)

After power-on, or after reset, the flash memory enters read mode if the automatic operation is properly completed. Instructions described in the flash memory or data reading is executed in read mode.

The operation mode enters to Command sequence input mode after area setting. A command is inputted during this mode, the flash memory enters automatic operation mode. When a command processing is completed properly, the flash memory returns to read mode except the case that ID-Read command is handled. During the automatic operation mode, data reading or instruction on the flash memory cannot be executed.

4.1.2. Command Execution

A command is executed on the flash memory with the store instruction by inputting the command sequence after area setting. The flash memory executes an automatic operation command depending on the combination of input address and data. For details of command execution, refer to “4.1.3. Command Description”.

A cycle where the store instruction is executed on the flash memory is called “bus write cycle”. Each command takes some bus write cycle. The flash memory executes automatic operation as long as the address and data in the bus write cycle are performed in the proper order. Otherwise, the flash memory aborts executing the command, and returns to read mode.

When the user attempts to cancel the command sequence in the middle of the process, or inputs the undefined command sequence, the flash memory executes the read/reset command to enter read mode.

Note: Please perform cancellation until the 3rd bus cycle in an automatic program command, and until the last bus cycle in other commands.

When the command sequence is inputted completely, the flash memory starts the automatic operation and ***[FCSR0]***<RDYBSY>=0. When the automatic operation is completed properly, ***[FCSR0]***<RDYBSY> is set to "1"

Another command sequence is not accepted during automatic operation.
The following cautions should be exercised when executing a command.

(1) Do not perform the operation during the automatic operation below:
- Power shutdown
- All exceptions (Recommend)

(2) In order to recognize a command by the command sequencer, the flash memory must be in read mode before executing the command. Thus, confirm whether ***[FCSR0]***<RDYBSY>=1 before the flash memory entering command sequence input mode. And selecting area then execute the Read/Reset command.

(3) Execute the following command sequences on the on-chip RAM.

- Automatic chip erasing command
- ID-Read command
- Automatic security bit programming command
- Automatic security bit erasing command
- Automatic protect bit programming command
- Automatic protect bit erasing command
- Automatic memory swap command
- Automatic memory swap erasing command

(4) Set the area selection bit of the ***[FCAREASEL]*** register before executing each command. (Write “111” to <AREAn>).
Note that when the following commands are executed, set all area selection bits.

- Automatic chip erasing command
- ID-Read command
- Automatic security bit programming command
- Automatic security bit erasing command
- Automatic protect bit programming command
- Automatic protect bit erasing command
- Automatic memory swap command
- Automatic memory swap erasing command

(5) Set each bus write cycle using consecutive 1-word (32-bit) data transfer instruction.

(6) If an access is performed to the target Flash memory in each command sequence, a bus fault occurs.

(7) When issuing commands, if wrong addresses or data are inputted, make sure to issue Read/Reset command, then return to read mode.

(8) Confirmation procedure after each command completion is as follows:

1) Execute the final bus write cycle
2) Poll until ***[FCSR0]***<RDYBSY>=0(Busy).
3) Poll until ***[FCSR0]***<RDYBSY>=1(Ready).

(9) When data is read from the flash memory, clear the area selection bit of the ***[FCAREASEL]*** register. (Set “000” to <AREAn>.)

When there are two or more area, reprogramming in a dual mode is possible for command sequence other than the above. For example, when there are the area 0 and the area 1, a program can be run on the flash memory of area (area 0) other than the flash memory of the object (area 1) which performs programming/erasing (vice versa can be possible).

In the case of the dual mode, it can be used of interruption during command execution in the area 0.

4.1.3. Command Description

This section explains each command contents. For details of specific command sequences, refer to “3.1.1 Command Sequence” and “3.2.1 Command Sequence”.

4.1.3.1. Automatic Programming

(1) Operation

Code flash can be programmed in four words (16 bytes) unit with the automatic programming command sequence. Programming across 16 bytes is not possible. Data flash can be programmed in one word (four bytes) unit.

Programming data to flash memory means that data cells of “1” become those of “0”. It is not possible to become data cells of “1” from those of “0”. To become data cells of “1” from “0”, the erase operation is required.

The automatic programming is allowed only once to each programming unit already erased. Either data cells of “1” or “0” cannot be programmed data twice or more. If reprogramming to an address that has already been programmed once, the automatic program is needed to be set again after the automatic page erasing, automatic block erasing, or automatic chip erasing command is executed.

Another command sequence is not accepted during automatic operation.
After programmed, flash memory returns to command sequence input mode.

Note1: Programming execute to the same programming unit twice or more without erasing operation may damage the data.
Note2: Programming to the protected block is not possible.

(2) How to set

The 1st to 3rd bus write cycles are the automatic programming command.
In the 4th bus write cycle, the first address and data are inputted. On and after 5th bus cycle, remaining data of four words will be inputted. Data flash is programmed in one word (32 bits) unit.

If a part of 16 bytes of code flash is used, program “0xFFFFFFFF” to the unused remaining part of 16 bytes.

If a part of four bytes of data flash is used, program “0xFFFFFFFF” to the unused remaining part of four bytes.

4.1.3.2. Automatic chip erasing

(1) Operation

Automatic chip erasing erases memory cells in all addresses. It erases in order of a data flash and a code flash. If protected pages or blocks are contained, the automatic chip erasing is performed on unprotected pages or blocks (Note1). After erased, flash memory returns to command sequence input mode.

Erasing target: Code Flash, Data Flash

Since a protect bits are not erased, when erasing protect bits are required, please erase by an automatic protection bit erase command.

Another command sequence is not accepted during automatic operation. If the users attempt to stop the automatic chip erase, refer to “4.1.4 Stopping Automatic Chip Erasing”. In this case, data may not be erased properly. Thus, the automatic chip erasing must be performed again.

(2) How to set

The 1^{st} to 6^{th} bus write cycles are the automatic chip erasing command sequences. After the command sequences are input, the automatic chip erasing starts.

Note 1: When there is the block or page protected, erasing operation is repeated per page inside a flash memory. It takes the time for the number of pages until erasing operation is completed.
Note 2: Automatic chip erasing cannot be performed continuously. When re-issue the chip erasing command, after once performing a blank check.

4.1.3.3. Automatic Area Erasing

(1) Operation

The automatic area erasing command performs on the specified area. If protected pages or blocks are contained, the automatic area erasing is performed on un-protected pages or blocks (Note1). After erased, flash memory returns to command sequence input mode.

Another command sequence is not accepted during automatic operation.
After erased, flash memory returns to command sequence input mode.

(2) How to set

The 1^{st} to 5^{th} bus write cycles are the automatic area erasing command sequences. The area to be erased is specified in the 6^{th} bus write cycle.
After the command sequences are input, the automatic area erasing starts.

Note 1: When there is the block or page protected, erasing operation is repeated per page inside a flash memory. It takes the time for the number of pages until erasing operation is completed.
Note 2: Automatic area erasing cannot be performed continuously. When re-issue the chip erasing command, after once performing a blank check.

4.1.3.4. **Automatic Block Erasing**

(1) Operation

The automatic block erasing command performs on the specified block. If protected pages or blocks are contained, the automatic block erasing is not performed on these pages or blocks. And flash memory returns to command sequence input mode.

Another command sequence is not accepted during automatic operation.
After erased, flash memory returns to command sequence input mode.

(2) How to set

The 1^{st} to 5^{th} bus write cycles are the automatic block erasing command sequences. The block to be erased is specified in the 6^{th} bus write cycle.
After the command sequences are input, the automatic block erasing starts.

4.1.3.5. **Automatic Page Erasing**

(1) Operation

The automatic page erasing command performs on the specified page. If protected page is contained, the automatic page erasing is not performed on this page. And flash memory returns to command sequence input mode.

Another command sequence is not accepted during automatic operation.
After erased, flash memory returns to command sequence input mode.

(2) How to set

The 1^{st} to 5^{th} bus write cycles are the automatic page erasing command sequences. The page to be erased is specified in the 6^{th} bus write cycle.
After the command sequences are input, the automatic page erasing starts.

4.1.3.6. **Automatic Protect Bit Programming**

(1) Operation

The automatic protect bit programming set “1” to the protect bit in the unit of bit. When clear to “0” to the protect bit, use the automatic protect bit erasing command.

For details of the protection function, refer to “4.1.6 Protection Function”.

Another command sequence is not accepted during automatic operation.
After programmed, flash memory returns to command sequence input mode.

(2) How to set

The 1^{st} to 3^{rd} bus write cycles are the automatic protect bit programming command sequences. The bit to be programmed is specified in the 4^{th} bus write cycle.
After the command sequences are input, the automatic protect bit programming starts. Whether the protect bit is programmed normally, please check each bit of the ***[FCPSRn]***.

4.1.3.7. Automatic Protect Bit Erasing

(1) Operation

The automatic protect bit erasing command erase the protect bit regardless of the security state of the flash memory.

For details of the protection function, refer to “4.1.6 Protection Function”.

Another command sequence is not accepted during automatic operation.
After erased, flash memory returns to command sequence input mode.

(2) How to set

Input a automatic protect bit erasing command sequence. After the command sequences are input, the automatic protect bit erasing starts. All protect bits are erased at one time. Whether the protect bits are erased normally, please check the ***[FCPSRn]***.

4.1.3.8. Automatic Security Bit Programming

(1) Operation

The automatic security bit programming set “1” to the security bit. When clear “0” to the security bit, use the automatic security bit erasing command.

For details of the security function, refer to “4.1.7. Security Function”.

Another command sequence is not accepted during automatic operation.
After programmed, flash memory returns to command sequence input mode.

(2) How to set

Input a security bit programming command sequence. After the command sequences are input, the automatic security bit programming starts. Whether the security bit is programmed normally, please check the ***[FCSSR]***<SEC>.

4.1.3.9. Automatic Security Bit Erasing

(1) Operation

The operation of the automatic security bit erasing command varies depending on the security state of the flash memory.

- Non secured state
  Erase the security bit.

- Security state
  Erase all address of code flash and data flash, and erase security bit.

For details of the security function, refer to “4.1.7. Security Function”.

Another command sequence is not accepted during automatic operation.
After erased, flash memory returns to command sequence input mode.

(2) How to set

Input security bit erasing command sequence. After the command sequences are input, the automatic security bit erasing starts.

In non security state, all bits are erased. Whether the security bit is erased normally, please check the ***[FCSSR]***<SEC>.

In security state, if perform the security bit erasing command sequence, data of all addresses of code flash, data flash and security bit are erased. After erased, read all addresses to check whether data of flash and the security bit are erased normally. And if necessary, perform the automatic protect bit erasing command.

4.1.3.10. **ID-Read**

(1) Operation

The ID-Read command can read the information including the type of the flash memory. The information consists of a manufacturer code, device code, and macro code.

(2) How to set

The 1^{st} to 3^{rd} bus write cycles are the ID-Read command sequences. The ID address to be read is specified in the 4^{th} bus write cycle. After the 4^{th} bus write cycle, release area selection to read mode and input 5^{th} bus cycle. It can be read ID data from Flash.

If read other ID, input ID-read command sequence from 1^{st} bus cycle again.

Note: After executed ID-read, must be execute the Read/Reset command for return to read mode.

4.1.3.11. **Read/Reset Command**

(1) Operation

This command is to return the flash memory to read mode.

(2) How to set

The 1^{st} bus write cycle is the Read/Reset command sequence. After the command sequence is executed, the flash memory returns to read mode.

4.1.3.12. Automatic Memory Swap

(1) Operation

The automatic memory swap set “1” to each bit of ***[FCSWPSR]***<SWP[1:0]><SIZE[5:0]> in the unit of bit. When clear to “0” to the protect bit, use the automatic memory swap erasing command.

Another command sequence is not accepted during automatic operation.
After executed, flash memory returns to command sequence input mode.

(2) How to set

The 1st to 4th bus write cycles are the automatic memory swap command sequences. After the command sequences are input, “1” is set to the designation bit of the ***[FCSWPSR]***. Whether the memory swap is programmed normally, please check each bit of the ***[FCSWPSR]***<SWP[1:0]><SIZE[5:0]>.

4.1.3.13. Automatic Memory Swap Erasing

(1) Operation content

The automatic memory swap erasing can erases ***[FCSWPSR]***<SWP[1:0]><SIZE[5:0]> at one time.

Another command sequence is not accepted during automatic operation.
After executed, flash memory returns to command sequence input mode.

(2) How to set

Input a command sequence “Automatic memory swap erasing”. After the command sequences are input, the automatic memory swap erasing starts. Whether the memory swap is erased normally, please check the ***[FCSWPSR]***<SWP[1:0]><SIZE[5:0]>.

4.1.4. Stopping Automatic Chip Erasing

When the user attempts to cancel the automatic chip erasing in the middle of the process, cancel the automatic chip erasing as follows:
The flash memory returns to read mode.

1. Read ***[FCSR0]***<RDYBSY>.
2. If the result of Procedure 1 is “1” (Ready), end at Procedure 9. If the result is “0” (Busy), proceed to Procedure 3.
3. Write “0x7” to ***[FCCR]***<WEABORT>.
4. Write “0x0” to ***[FCCR]***<WEABORT>.
5. Poll until ***[FCSR0]***<RDYBSY>=1(Ready).
6. Read ***[FCSR1]***<WEABORT>
7. Issue the Read/reset command.
8. If the result of Procedure 6 is “0”, end at Procedure 9. If the result of Procedure 6 is “1”, perform the following operation to clear this flag:
   1) Write “0x7” to ***[FCSTSCLR]***<WEABORT>.
   2) Write “0x0” to ***[FCSTSCLR]***<WEABORT>.
   3) Poll until ***[FCSR]***<WEABORT>=0.
9. End

Note**:** Before write to ***[FCCR]***, need clear protection by ***[FCKCR]***.

4.1.5. Completion Detection of the Automatic Operation

The flash memory has an interrupt function to detect the completion of programming/erasing operation.

Table 4.2 Detection of Completion Flash programming/Erasing

	Signal name	Interruption name
Completion of the programming/erasing operation of a code flash	INTFLCRDY	Code FLASH Ready interruption
Completion of the programming/erasing operation of a data flash	INTFLDRDY	Data FLASH Ready interruption

When an automatic chip erasing command sequence is performed, first, INTFLDRDY occurs at the time of the end of programming/erasing to a data flash. And next, INTFLCRDY occurs in generating at the time of the end of programming/erasing to a code flash.

4.1.5.1. Procedure

The procedure (in the case of a data flash) which uses completion detection interruption of automatic operation is as follows.

Please refer to chapter “Interrupts” of a reference manual “Exception” for the details of interruption processing.

1. Enable INTFLDRDY interruption.
2. After issued automatic programming or erasing command to a data flash, check under automatic operation (BUSY state) by ***[FCSR0]***<RDYBSY>.
3. An INTFLDRDY interrupt occurs after the end of automatic programming or erasing of data flash.
4. When you do not program in continuously, in an interrupt handler, INTFLDRDY interruption is disabled, and perform return. When you program in continuously, issue a new command sequence after INTFLDRDY interruption without disable, and perform return.

5. When continuing program, repeat step3 to 4 in parallel performing a main process.

4.1.6. Protection Function

The protection function prohibits data program/erase operation on the flash memory in the unit of block. The protection function is set to code flash and data flash separately.

In code flash, set the protection function to page 0 to 7 in the unit of page in block0. The remaining blocks are set in the unit of block. In data flash, the protection function is set in the unit of block.
Erasing protect setting, all protect bits are erased one time.

4.1.6.1. How to Set the Protection Function

In order to enable a protection function, a protect bit is set to “1” by a protect bit programming command.
The protection function is enabled under the condition below:

1. ***[FCPMRm]***<MSKn>=1 (Note)
2. Protect bit n=1

At this time, the block n is being protected from data programming/erasing.
When check the status of protect bit, monitor ***[FCPSRm]*** after set ***[FCPMRm]***<MSKn>=1 .(Note)

Note: <MSKn> represents <PMn>, <MSKn>, and <DMSKn>.

4.1.6.2. Protection Release

Execute the protect bit erasing command, protect bits become “0” and being released block protection.

Note: All protect bits become “0” with the protect bit erasing command.

4.1.6.3. Protection Temporary Release Function

The protection function can be temporary released without erasing the protect bits.
Specified block can only be released.

When ***[FCPMRm]***<MSKn>=0, programming/erasing operation function is disabled regardless of the state of the protect bits (***[FCPSRm]***<PGn>/<BLKn>).

For details of register settings, refer to ***[FCPMRm]*** in chapter “5.2. Detail of Registers”.

Note: <MSKn> represents <PMn>, <MSKn>, and <DMSKn>.

4.1.7. Security Function

The security function can disable data reading from the flash writer, and disable the debug function.

4.1.7.1. Security Setting

In order to enable a security function, a security bit is set to “1” by a security bit program command.
The security function is enabled under the following conditions:

1. ***[FCSBMR]***<SMB>=1
2. Security bit=1

When check the status of security bit, monitor ***[FCSSR]***<SEC> after set ***[FCSBMR]***<SMB>=1.

4.1.7.2. Security Setting Release

To release the security function, perform the procedure below:

1. ***[FCSBMR]***<SMB>=0
2. Set “0” to the security bit with the security bit erasing command.

While ***[FCSBMR]***<SMB>=1 and ***[FCSSR]***<SEC>=1, if the security bit erasing command is executed, the chip erasing function is executed, and then code flash, data flash, and security bit is erased.

4.1.7.3. Operation

Table 4.3 shows the flash memory operation when the security function is enabled.

Table 4.3 Flash memory operation when the security function is enabled

Parameter	Description
Flash memory reading	Reading from the CPU is possible.
Debug mode	Debugging is disabled.
Flash writer mode	Flash memory cannot be read or written.

4.1.8. Memory Swap Function

When application program reprogramming on the code flash is suspended, the power may become off after the program code is erased, and application program reprogramming may not be continued. To avoid such case, use this memory swap function to save your program.

4.1.8.1. Memory Swap Setting

A swap region starts from Address 0 and the same size next region. A swap size is determined by ***[FCSWPSR]***<SIZE>. To change the size, set "1" to ***[FCSWPSR]***<SIZE> with the automatic memory swap programming command.

To perform memory swap, set "1" to ***[FCSWPSR]***<SWP[0]> with the automatic memory swap programming command. To release the swap condition, set "1" to ***[FCSWPSR]***<SWP[1]> with the automatic memory swap command or execute the automatic memory swap erasing command. A swap condition can be checked with ***[FCSWPSR]***<SWP>.

For details of the automatic memory swap command, refer to "4.1.3.12 Automatic Memory Swap".

4.1.8.2. Memory Swap Operation

This section explains the basic operation flow of the memory swap. For the concrete example of the memory swap operation, refer to "6.8 How to Reprogramming User Boot Program".

Release the protection function temporarily, when the protection function is valid.

For details of the protection function temporary release, refer to "4.1.6.3 Protection Temporary Release Function". If the protection function is not temporarily released, command execution is not performed in the procedure.

1. Check whether the next area (next to the area starting from Address 0) is blank. (Hereafter the area starting from 0 is called Page 0, and the next area is called Page 1.) If not, erase the area.

   Page0: Old original data
   Page1: Blank

2. Program the original data starting from Address 0 to the next region. (Both regions have the same data.)

   Page0: Old original data
   Page1: Copied data (old original data)

3. Perform memory swap.

   Page0: Copied data (old original data)
   Page1: Old original data

4. Erase old original data to be blank.

   Page0: Copied data (Old original data)
   Page1: Blank

5. Program new data to the blank region.

   Page0: Copied data (Old original data)
   Page1: New original data

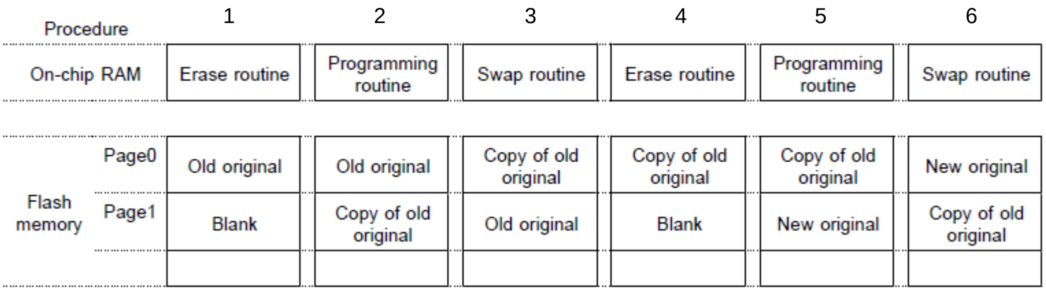
6. Release the swap state.

   Page0: New original data
   Page1: Copied data (Old original data)

7. Execute the automatic protect bit erasing command.

8. Options if required.

   - Erase copied data (old original data).
   - Reprogram the flash memory data except the swap regions.
   - Validate the protection function.
   - Validate the security function.



Erase routine: A program is to erase Flash memory.
Programming routine: A program is to program Flash memory.
Swap routine: A program is to swap Flash memory.

Figure 4.1 Example of Procedure of Memory Swap

4.1.8.3. Erasing the Memory Swap Information

After the memory swap state is released, if the user attempts to perform memory swap again, initialize the all bits of the ***[FCSWPSR]*** register with the automatic memory swap erasing command.

4.1.9. User Information Area

Instructions cannot be executed in the user information area. Data reading can be instructed by the CPU.

Data becomes accessible on bank switching with ***[FCBNKCR]***. For address assignment, refer to “Table 2.9 User Information Area Configuration of Code Flash”. After bank switching, do not access to code flash (Area 0).

The chip erasing command is not erased; therefore, it can be written the unique number for management.

User information area cannot be used with code flash (Area 0) Use this area exclusively.

4.1.9.1. Switching Procedure of the User Information Area

(1) Load the switching program on the RAM, and make Jump.

(2) Write “111” to ***[FCBUFDISCLR]***<BUFDISCLR[2:0]>.

(3) Write “111” to ***[FCBNKCR]***<BANK0[2:0]>.

(4) Read ***[FCBNKCR]***<BANK0[2:0]> to confirm whether ***[FCBNKCR]***<BANK0[2:0]> is “111”.

(5) Perform the following operation in the user information area:
Data reading, data programming, data erasing

(6) Write “000” to ***[FCBNKCR]***<BANK0[2:0]>.

(7) Read ***[FCBNKCR]***<BANK0[2:0]> to confirm whether ***[FCBNKCR]***<BANK0[2:0]> is “000”.

(8) Write “000” to ***[FCBUFDISCLR]***<BUFDISCLR[2:0]>.

(9) Return to the original program.

4.1.9.2. Data programming Method for the User Information Area

Data programming on the user information memory is the same procedure as those of code flash (Area 0) by step (5) of “4.1.9.1”.

4.1.9.3. Data Erasing Method for the User Information Area

Data erasing on the user information memory is the same procedure as those of code flash (Area 0) by step (5) of “4.1.9.1”. All data are erased at one time.

5. Registers

5.1. Register List

The table below lists the registers related to flash memory.

Circumference function		channel/unit	Base address
			Type1
Flash memory	FC	-	0x5DFF0000

Register name		Address (Base+)
Flash Security Bit Mask Register	***[FCSBMR]***	0x010
Flash Security Status Register	***[FCSSR]***	0x014
Flash Key Code Register	***[FCKCR]***	0x018
Flash Status Register 0	***[FCSR0]***	0x020
Flash Protect Status Register 0	***[FCPSR0]***	0x030
Flash Protect Status Register 1	***[FCPSR1]***	0x034
Flash Protect Status Register 6	***[FCPSR6]***	0x048
Flash Protect Mask Register 0	***[FCPMR0]***	0x050
Flash Protect Mask Register 1	***[FCPMR1]***	0x054
Flash Protect Mask Register 6	***[FCPMR6]***	0x068
Flash Status Register 1	***[FCSR1]***	0x0100
Flash Memory SWAP Status Register	***[FCSWPSR]***	0x0104
Flash Area Selection Register	***[FCAREASEL]***	0x0140
Flash Control Register	***[FCCR]***	0x0148
Flash Status Clear Register	***[FCSTSCLR]***	0x014C
Flash Bank Change Register	***[FCBNKCR]***	0x0150
Flash Buffer Disable and Clear Register	***[FCBUFDISCLR]***	0x0158

Note: Do not access to the addresses where the registers are not assigned.

5.2. Detail of Registers

5.2.1. *[FCSBMR]* (Flash Security Bit Mask Register)

Bit	Bit Symbol	After reset	Type	Function
31:1	-	0	R	Read as "0"
0	SMB	1	R/W	Security mask bit 1: No masked 0: Masked (Security is temporarily released) When security is enabled (***[FCSSR]***<SEC>=1), if "0" is written to this register, security is temporarily released. This register is initialized only on the Power On Reset (at power on or at the return from STOP2 with power shutdown).

Note: To rewrite this register, follow the procedure below:

1. Write the specific code (0xA74A9D23) to ***[FCKCR]***.
2. Rewrite the data of ***[FCSBMR]***<SMB> within 16 clocks after Procedure 1.

5.2.2. *[FCSSR]* (Flash Security Status Register)

Bit	Bit Symbol	After reset	Type	Function
31:1	-	0	R	Read as "0"
0	SEC	0/1	R	Security status: Indicates security status. 1: Secured 0: Not secured The state of security is loaded by a system reset.

5.2.3. *[FCKCR]* (Flash Key Code Register)

Bit	Bit Symbol	After reset	Type	Function
31:0	KEYCODE	0x00000000	W	Locked register release key code When ***[FCSBMR]***, ***[FCPMRn]***, ***[FCCR]***, ***[FCAREASEL]*** are rewritten, write the specific code (0xA74A9D23) to this register. And then rewrite the value of the register within 16 clocks after the previous action. If invalid data is written to this register within 16 clocks, released status is reset.

5.2.4. *[FCSR0]* (Flash Status Register 0)

Bit	Bit Symbol	After reset	Type	Function
31:16	-	0	R	Read as “0”
15	-	Undefined	R	Read as “Undefined”
14:11	-	0	R	Read as “0”
10	RDYBSY2	1	R	ReadyBusy flag of Area 4 0: In automatic operation 1: Completion of automatic operation
9	-	1	R	Read as “1”
8	RDYBSY0	1	R	ReadyBusy flag of Area 0 0: In automatic operation 1: Completion of automatic operation
7:1	-	0	R	Read as “0”
0	RDYBSY	1	R	ReadyBusy flag (all flash area) 0: In automatic operation 1: Completion of automatic operation ReadyBusy flag indicate when automatic programming command or automatic erasing command is executed. This bit indicates automatic operation status. When this bit is “0”, it indicates that the flash memory is busy status where it is in automatic operation. When automatic operation is completed, this bit is set to “1”. It indicates ready status where the register can accept the next command.

5.2.5. *[FCPSR0]* (Flash Protect Status Register 0)

Bit	Bit Symbol	After reset	Type	Function
31:8	-	0	R	Read as “0”
7	PG7	0/1	R	Protect status of code flash (Area 0). 1: Protected 0: Not protected This register indicates the protected status of each page from Page0 to Page7 (Block0). If one of bits is “1”, it indicates that the corresponding page is protected. Protected page cannot be erased and programmed. The state of protection is loaded by system reset.
6	PG6	0/1	R	
5	PG5	0/1	R	
4	PG4	0/1	R	
3	PG3	0/1	R	
2	PG2	0/1	R	
1	PG1	0/1	R	
0	PG0	0/1	R	

5.2.6. *[FCPSR1]* (Flash Protect Status Register 1)

Bit	Bit Symbol	After reset	Type	Function
31:16	-	0	R	Read as "0"
15	BLK15	0/1	R	Protect status of code flash (Area 0). 1: Protected 0: Not protected This register indicates the protected status of each block from Block1 to Block15. If one of bits is "1", it indicates that the corresponding block is protected. Protected block cannot be erased and programmed. The state of protection is loaded by system reset.
14	BLK14	0/1	R	
13	BLK13	0/1	R	
12	BLK12	0/1	R	
11	BLK11	0/1	R	
10	BLK10	0/1	R	
9	BLK9	0/1	R	
8	BLK8	0/1	R	
7	BLK7	0/1	R	
6	BLK6	0/1	R	
5	BLK5	0/1	R	
4	BLK4	0/1	R	
3	BLK3	0/1	R	
2	BLK2	0/1	R	
1	BLK1	0/1	R	
0	-	0	R	Read as "0"

5.2.7. *[FCPSR6]* (Flash Protect Status Register 6)

Bit	Bit Symbol	After reset	Type	Function
31:8	-	0	R	Read as "0"
7	DBLK7	0/1	R	Protect status of data flash (Area 4). 1: Protected 0: Not protected This register indicates the protected status of each block of data flash. If one of bits is "1", it indicates that the corresponding block is protected. Protected block cannot be erased and programmed. The state of protection is loaded by system reset.
6	DBLK6	0/1	R	
5	DBLK5	0/1	R	
4	DBLK4	0/1	R	
3	DBLK3	0/1	R	
2	DBLK2	0/1	R	
1	DBLK1	0/1	R	
0	DBLK0	0/1	R	

5.2.8. *[FCPMR0]* (Flash Protect Mask Register 0)

Bit	Bit Symbol	After reset	Type	Function
31:8	-	0	R	Read as "0"
7	PM7	1	R/W	Protect mask status of code flash (Area 0). 1: Not masked (Protected) 0: Masked (Not protected) This register masks each protected page from Page0 to Page7 (block0). This register is initialized by a system reset.
6	PM6	1	R/W	
5	PM5	1	R/W	
4	PM4	1	R/W	
3	PM3	1	R/W	
2	PM2	1	R/W	
1	PM1	1	R/W	
0	PM0	1	R/W	

Note: To rewrite this register, follow the procedure below:

1. Write the specific code (0xA74A9D23) to ***[FCKCR]***.
2. Rewrite the data of ***[FCPMR0]***<PMn> within 16 clocks after Procedure 1.

5.2.9. *[FCPMR1]* (Flash Protect Mask Register 1)

Bit	Bit Symbol	After reset	Type	Function
31:16	-	1	R/W	Write as "1"
15	MSK15	1	R/W	Protect mask status of code flash (Area 0). 1: Not masked (Protected) 0: Masked (Not protected) This register masks each protected page from Block 1 to Block 15 in the unit of block. This register is initialized by a system reset.
14	MSK14	1	R/W	
13	MSK13	1	R/W	
12	MSK12	1	R/W	
11	MSK11	1	R/W	
10	MSK10	1	R/W	
9	MSK9	1	R/W	
8	MSK8	1	R/W	
7	MSK7	1	R/W	
6	MSK6	1	R/W	
5	MSK5	1	R/W	
4	MSK4	1	R/W	
3	MSK3	1	R/W	
2	MSK2	1	R/W	
1	MSK1	1	R/W	
0	-	0	R	Read as "0"

Note: To rewrite this register, follow the procedure below:

1. Write the specific code (0xA74A9D23) to ***[FCKCR]***.
2. Rewrite the data of ***[FCPMR1]***<MSKn> within 16 clocks after Procedure 1.

5.2.10. *[FCPMR6]* (Flash Protect Mask Register 6)

Bit	Bit Symbol	After reset	Type	Function
31:16	-	0	R	Read as “0”
15:8	-	1	R/W	Recommended write as “1”
7	DMSK7	1	R/W	Protect status of data flash (Area 4). 1: Not masked (Protected) 0: Masked (Not protected) This register masks each protected block of data flash memory in the unit of block. This register is initialized by a system reset.
6	DMSK6	1	R/W	
5	DMSK5	1	R/W	
4	DMSK4	1	R/W	
3	DMSK3	1	R/W	
2	DMSK2	1	R/W	
1	DMSK1	1	R/W	
0	DMSK0	1	R/W	

Note: To rewrite this register, follow the procedure below:

1. Write the specific code (0xA74A9D23) to ***[FCKCR]***.
2. Rewrite the data of ***[FCPMR6]***<DMSKn> within 16 clocks after Procedure 1

5.2.11. *[FCSR1]* (Flash Status Register 1)

Bit	Bit Symbol	After reset	Type	Function
31:25	-	0	R	Read as “0”
24	WEABORT	0	R	When ***[FCCR]***<WEABORT>=111, this bit is set to “1”.
23:0	-	0	R	Read as “0”

5.2.12. *[FCSWPSR]* (Flash Memory SWP Status Register)

Bit	Bit Symbol	After reset	Type	Function
31:14	-	0	R	Read as “0”
13:8	SIZE[5:0]	000000	R	This bit indicates the setting of memory swap size (Area 0). 000000: No swap (Initial value) 000001: 4Kbytes (Page0 ←→ Page1) 000010: 8Kbytes (Page0 to 1 ←→ Page 2 to 3) 000100: 16K bytes (Page 0 to 3 ←→ Page 4 to 7) 001000: 32K bytes (Block0 ←→ Block1) Other settings than the above are prohibited.
7:2	-	0	R	Read as “0”
1:0	SWP[1:0]	00	R	Swap setting 00: Release the swap (Initial status) 01: Swap is ongoing 10: Prohibited 11: Release the swap

Note1: Perform memory swap on the program in the RAM.

Note2: If “11” is set to <SWP[1:0]>, swap is released. If “00” is set to <SWP[1:0]>, swap condition is released and initialized with the automatic swap erasing command.
At this time, the swap size setting <SIZE[5:0]> is initialized to “000000” as well.
Note that this operation must be performed on the condition where program code is programmed in both memories to be swapped.

5.2.13. *[FCAREASEL]* (Flash Area Selection Register)

Bit	Bit Symbol	After reset	Type	Function
31	-	0	R	Read as "0"
30	SSF4	0	R	Selection of Area 4. **1:** Selects Area 4 (Command sequence input mode) **0:** Not select Area 4 (Read mode)
29:27	-	0	R	Read as "0"
26	SSF0	0	R	Selection of Area 0. **1:** Selects Area 0 (Command sequence input mode) **0:** Not select Area 0 (Read mode)
25:23	-	0	R	Read as "0"
22:20	-	000	R/W	Write as "000"
19	-	0	R	Read as "0"
18:16	AREA4[2:0]	000	R/W	Specify Area 4 of data flash as the target to enter to Command sequence input mode for data programming with flash memory operation commands. (Note1) 111**:** Selects Area 4. Others**:** Not select Area 4.
15	-	0	R	Write as "0"
14:12	-	000	R/W	Write as "000"
11	-	0	R	Read as "0"
10:8	-	000	R/W	Write as "000"
7	-	0	R	Read as "0"
6:4	-	000	R/W	Write as "000"
3	-	0	R	Read as "0"
2:0	AREA0[2:0]	000	R/W	Specify Area 0 of code flash as the target to enter to Command sequence input mode for data programming with flash memory operation commands. (Note1) 111**:** Selects Area 0. Others**:** Not select Area 0.

Note1: When rewrite <AREA0[2:0]>,<AREA4[2:0]>, please perform the next operation until read data of <SSF0>,<SSF4> is reflected setting.

Note2: To rewrite this register, follow the procedure below:

1. Write the specific code (0xA74A9D23) to ***[FCKCR]***.
2. Rewrite the data of ***[FCAREASEL]***<AREAn> within 16 clocks after the previous action.

Note3: Rewrite the contents of this register on the program code in the RAM.

5.2.14. *[FCCR]* (Flash Control Register)

Bit	Bit Symbol	After reset	Type	Function
31:3	-	0	R	Read as "0"
2:0	WEABORT[2:0]	000	R/W	Stops the automatic chip erasing. 111: Stops the automatic erasing operation. 000: Inactive Others: Prohibited

Note1: To rewrite this register, follow the procedure below:

1. Write the specific code (0xA74A9D23) to ***[FCKCR]***.
2. Rewrite data of ***[FCCR]***<WEABORT> within 16 clocks after Procedure 1.

Note2: Rewrite the contents of this register on the program code in the RAM.

5.2.15. *[FCSTSCLR]* (Flash Status Clear Register)

Bit	Bit Symbol	After reset	Type	Function
31:3	-	0	R	Read as "0"
2:0	WEABORT[2:0]	000	R/W	Clear ***[FCSR1]***<WEABORT> to "0". 111: Clears Others: Inactive

Note: Rewrite the contents of this register on the program code in the RAM.

5.2.16. *[FCBNKCR]* (Flash Bank Change Register)

Bit	Bit Symbol	After reset	Type	Function
31:7	-	0	R	Read as "0"
6:4	-	000	R/W	Write as "000"
3	-	0	R	Read as "0"
2:0	BANK0[2:0]	000	R/W	Address "0x5E005000" to "0x5E005FFF" of code flash change to the user information area. 111: User information area 000: Code Flash Others: Don't care

Note 1: Before and after BANK0 operation, code flash buffer operation is required. For detail, refer to "5.2.17. ***[FCBUFDISCLR]*** (Flash Buffer Disable and Clear Register)".
Note 2: To set this register, write the value to the register, and confirm the written value by reading the register.
Note 3: Rewrite the contents of this register on the program code in the RAM.
Note 4: Do not access to code flash (Area0) except "0x5E005000" to "0x5E005FFF" while the user information area is being used.

5.2.17. *[FCBUFDISCLR]* (Flash Buffer Disable and Clear Register)

Bit	Bit Symbol	After reset	Type	Function
31:3	-	0	R	Read as "0"
2:0	BUFDISCLR[2:0]	000	R/W	Stops the buffer of code flash, and clears the buffer. 111: Stops the buffer function and clears the buffer. 000: Start the buffer function. Others: Inactive When bank switch (***[FCBNKCR]***) is performed between code flash (Area 0) and user information area, make sure to stop and clear the buffer with this register before the switching starts. After the user information area is operated, make sure to write "000" to start the buffer operation.

Note1: When the value is set to this register, write the value to the register, and confirm the written value by reading the register.
Note2: Rewrite the contents of this register on the program code in the RAM.

6. The programming method

6.1. Initialization

Before performing programming/erasing operation to a code flash or a data flash, must be oscillate a internal high speed oscillator1 (IHOSC1). And. Please operate flash memory after oscillation start and check ***[CGOSCCR]***<IHOSC1F>=1. Please refer to the reference manual “Clock Control and Operation Mode” for detail.

6.2. Mode Description

This device provides single chip mode and single boot mode. The single chip mode contains normal mode and user boot mode. Please refer to Table 6.1 for detail.

Table 6.1 The mode and operation

Mode		Operation
Single boot Mode		After reset is released, The program of the Boot ROM (mask ROM) which is build-in will be start. "The programming/erasing program code for a flash memory" can be downloaded from the host to on-chip RAM via a communication function, and the “The programming/erasing program for a flash memory” can be run. Please refer to “6.6. How to Reprogram the Flash in Single Boot Mode”.
Single chip Mode	Normal Mode	A user's application program is run. Moreover, a on-chip flash memory can be program/erase a "flash memory programming/erasing program" in RAM. Although it can be operated to all the flash memory to build in, the application program of the user on a flash memory cannot be run during flash memory programming/erasing. When only code flash (Area 0) is built-in, only this mode is available. Please refer to “6.5. How to Reprogramming the Flash” for how to program/erase a flash memory.
	Dual Mode	The on-chip flash memory which is different area can be erasing and programming, and running a user's application program concurrently. In case of built-in two or more the Area of a code flash or data flash are available. Please refer to “6.7. How to Reprogramming using Dual Mode” for how to program/erase a flash memory.

6.3. Mode Determination

Either the single chip mode or single boot operation mode is determined by the level of the BOOT_N pin when reset is released.

Table 6.2 Operation mode setting

Operation mode	Pin	
	RESET_N	BOOT_N
Single chip mode Dual mode	0→1	1
Single boot mode	0→1	0

6.4. Memory Map in Each Mode

Refer to “Figure 1.1 The example of a memory map”.

6.5. How to Reprogramming the Flash

The user boot mode reprograms the flash memory using the program in the on-chip RAM on the user's set. This mode is used when the data transfer bus for the flash memory program code on the user application is different from the UART. It operates in single chip mode; therefore, normal mode, in which user application is activated in single chip mode, needs to switch to user boot mode for programming flash memory. For that reason, the user is required to add a mode judgment routine to the reset service routine in the user application program.

This mode switch condition is required to be constructed according to the user system set condition. A flash memory programming routine, which is uniquely made by the user, needs to be installed in the new application. This routine is used for programming after being switched to the user boot mode. It is recommended that program/erase protection is set to the necessary block to avoid accidental modification in single chip mode (normal operation mode) after reprogramming is completed. Make sure not to generate any exception in user boot mode.

The following section explains two procedures where the reprogramming routine stored in Flash memory (1-A) and the reprogramming routine is transferred from the external device (1-B). For details of the programming/erasing the flash memory, refer to "4 Details of Flash Memory".

6.5.1. (1-A) Procedure that a Programming Routine Stored in Flash memory

6.5.1.1. Step-1

A user determines the conditions (e.g., pin status) to enter the user boot mode and the I/O to be used to transfer data. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, programmed the following three program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine:	A program to determine to switch to user boot mode.
(b) Flash programming routine:	A program to download new program from the external device and reprogram Flash memory.
(c) Copy routine:	A program to copy the data described in (b) to the on-chip RAM.

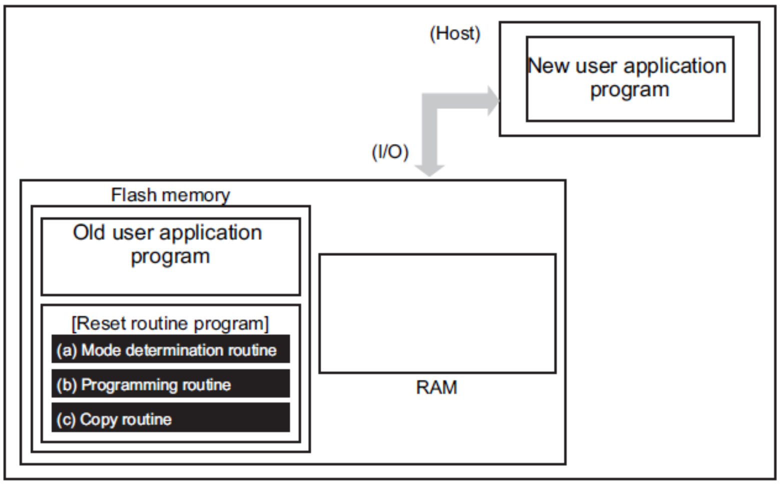


Figure 6.1 Procedure that a Programming Routine Stored in Flash memory (1)

6.5.1.2. **Step-2**

This section explains the case that a programming routine stored in the reset service routine. First, the reset routine determines to enter the user boot mode. If mode switching conditions are met, the device enters the user boot mode to reprogram data. (Make sure not to generate any exception in user boot mode.)

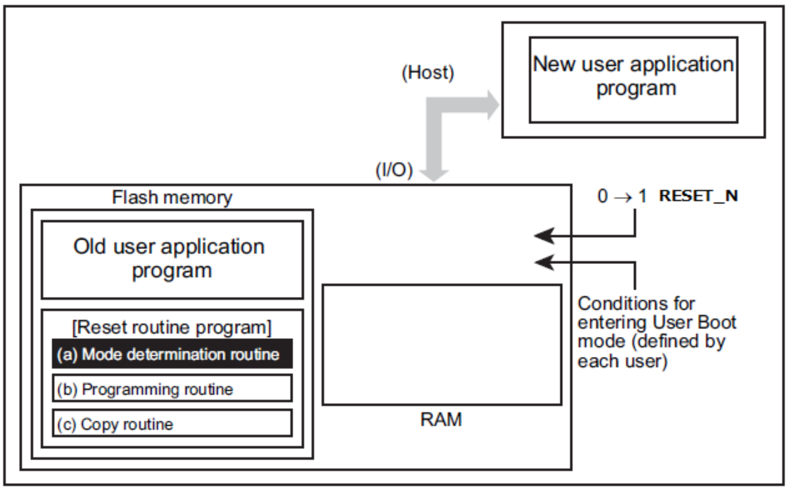


Figure 6.2 Procedure that a Programming Routine Stored in Flash memory (2)

6.5.1.3. **Step-3**

After the device enters the user boot mode, the device executes the copy routine (c) to download the flash programming routine (b) from the host controller to the on-chip RAM.

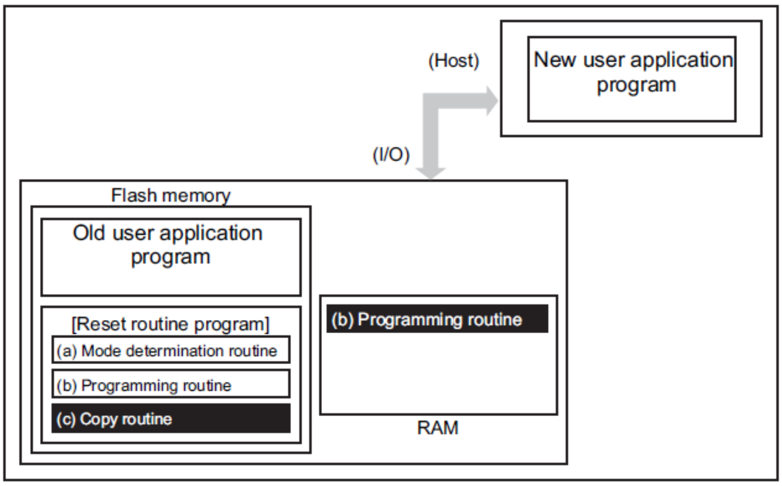


Figure 6.3 Procedure that a Programming Routine Stored in Flash memory (3)

6.5.1.4. **Step-4**

The device jumps to the programming routine (b) on the RAM to release the program/erase protection for the old application program, and to erase the flash (the units of erase is arbitrary size).

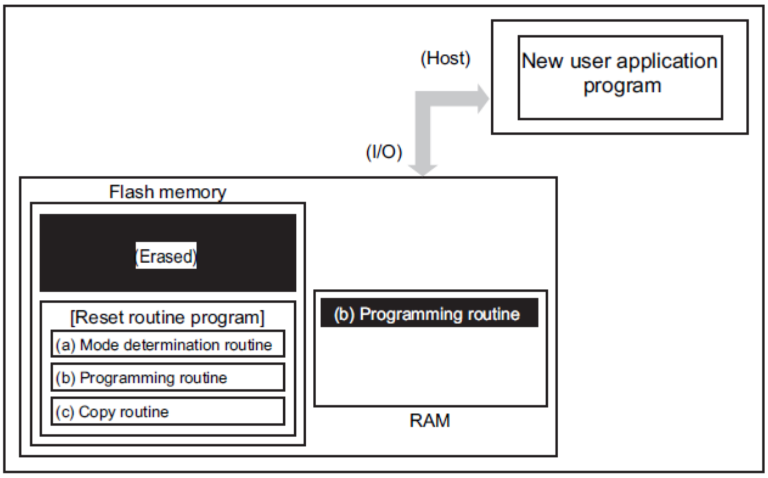


Figure 6.4 Procedure that a Programming Routine Stored in Flash memory (4)

6.5.1.5. **Step-5**

The device continues to execute the flash programming routine to download new program data from the host controller and program it into the erased flash block. When the programming is completed, set the program/erase protection of that flash area in the user's program to ON.

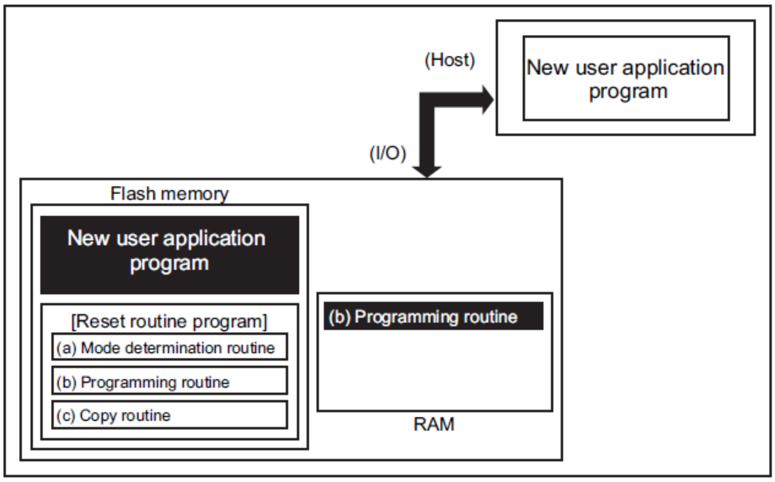


Figure 6.5 Procedure that a Programming Routine Stored in Flash memory (5)

6.5.1.6. **Step-6**

Do reset by setting “0” to the RESET_N pin. Upon reset, the flash memory is set to normal mode. After reset, the CPU will start operation along with the new application program.

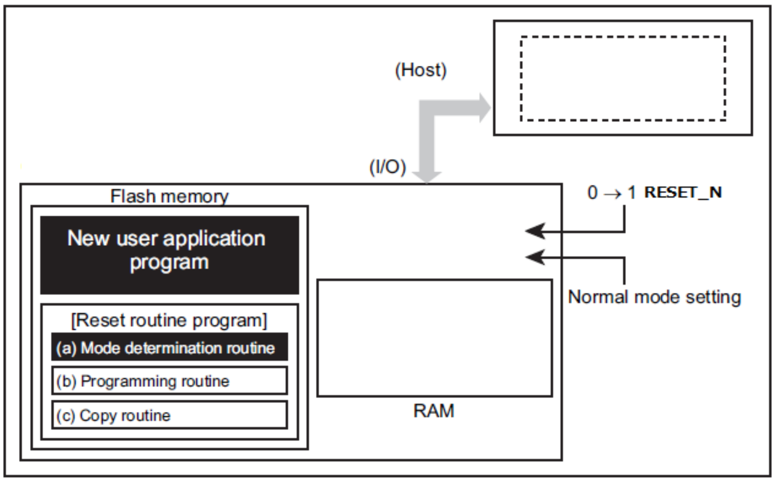


Figure 6.6 Procedure that a Programming Routine Stored in Flash memory (6)

6.5.2. (1-B) Procedure that a Programming Routine is Transferred from External Host

6.5.2.1. Step-1

The user determines the conditions (e.g., pin status) to enter user boot mode, and determines I/O used in data transfer. Then suitable circuit design and program are created. Before installing the device on a printed circuit board, programmed the following two program routines into an arbitrary flash block using programming equipment such as a flash writer.

(a) Mode determination routine:	A program to determine to switch to reprogramming operation
(b) Transfer routine:	A program to obtain a programming program (c) from the external device.

The programming routine shown below must be prepared on the host controller.

(c) Programming routine:	A program to reprogramming data

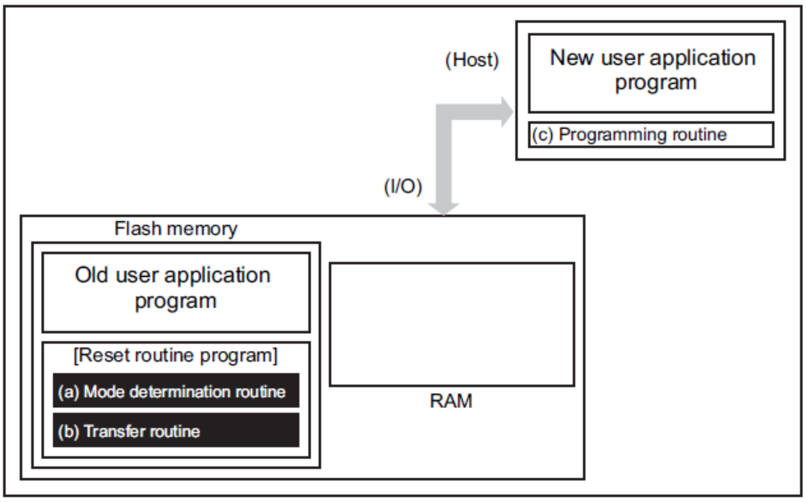


Figure 6.7 Procedure that a Programming Routine is Transferred from External Host (1)

6.5.2.2. Step-2

This section explains the case where a programming routine is stored in the reset service routine.

First, the reset service routine determines to enter user boot mode. If mode switching conditions are met, the device enters user boot mode to reprogram data. (Make sure not to generate any exception in user boot mode.)

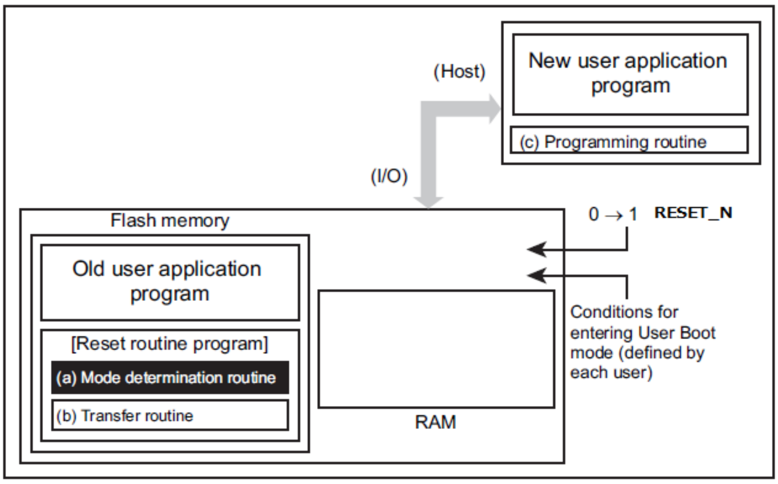


Figure 6.8 Procedure that a Programming Routine is Transferred from External Host (2)

6.5.2.3. Step-3

After the device enters user boot mode, the device executes the transfer routine (b) to download the flash programming routine (c) from the host controller to the on-chip RAM.

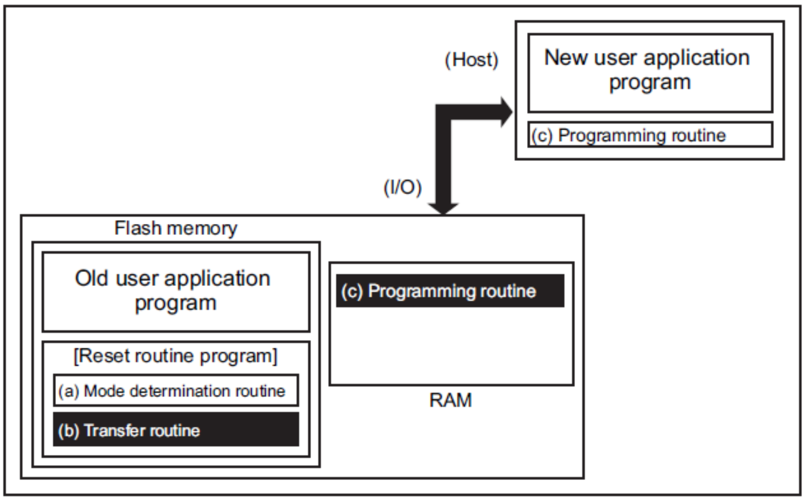


Figure 6.9 Procedure that a Programming Routine is Transferred from External Host (3)

6.5.2.4. **Step-4**

The device jumps to the programming routine on the RAM to release the program/erase protection for the old application program, and to erase the flash (the units of erase is arbitrary size).

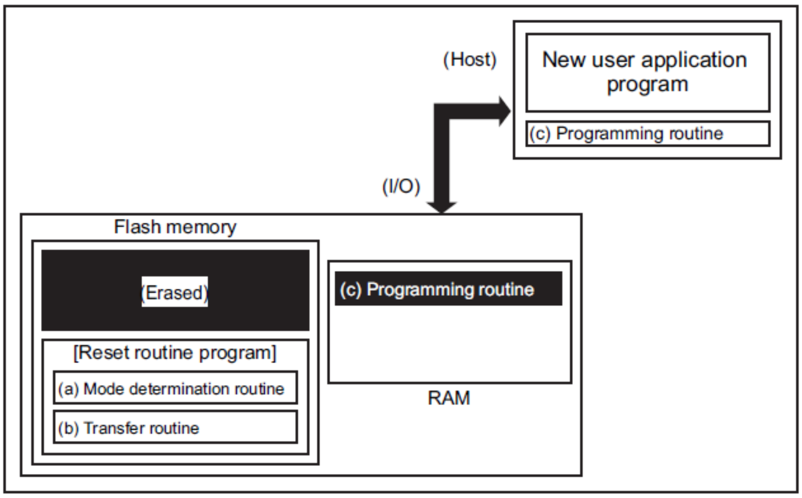


Figure 6.10 Procedure that a Programming Routine is Transferred from External Host (4)

6.5.2.5. **Step-5**

The device continues to execute the programming routine (c) on the RAM to download new program data from the host controller and programs it into the erased flash blocks. When the programming is completed, set the program/erase protection of that flash area in the user's program to ON.

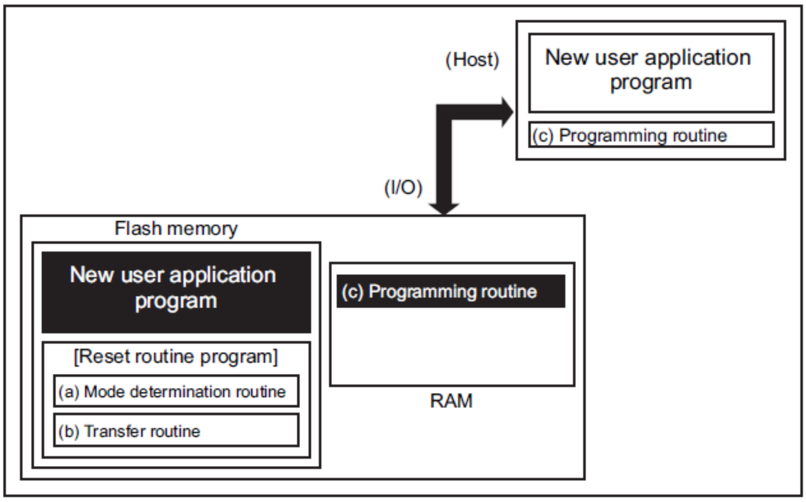


Figure 6.11 Procedure that a Programming Routine is Transferred from External Host (5)

6.5.2.6. **Step-6**

Do reset by setting “0” to the RESET_N pin. Upon reset, the flash memory is set to normal mode. After reset, the CPU will start operation along with the new application program.

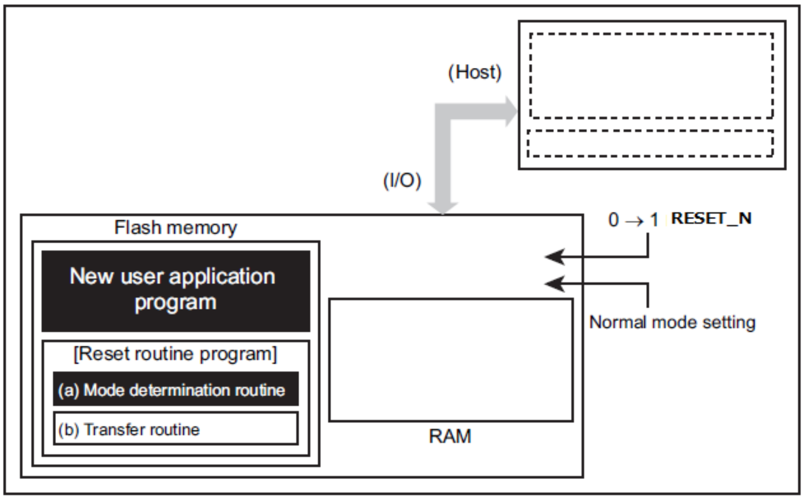


Figure 6.12 Procedure that a Programming Routine is Transferred from External Host (6)

6.6. How to Reprogram the Flash in Single Boot Mode

6.6.1. Single Boot Mode

The single boot mode utilizes a program contained in on-chip Boot ROM for reprogramming the flash memory. In this mode, the Boot ROM is mapped to the area containing interrupt vector tables, and the flash memory is mapped to another address area other than the Boot ROM area.

In the single boot mode, the flash memory is reprogrammed by the commands and data on serial transfer.

Table 6.3 Functions and Commands

Functions /Commands	Basic Operation	Description	Comment /Refer section
Communication function	Communication equipment	Use UART	
	Communication Rate	The signal sent at the rate beforehand decided from the external host controller is analyzed, and a communication rate is set up automatically.	"Table 6.7 Setting of baud rate in Single boot mode (fc=10MHz, No error)"
RAM Transfer Command	Transfer to on-chip RAM	Connecting the pins of UART to the external host, a programming routine is copied from the external host device to the on-chip RAM. A programming routine on the RAM is executed to erasing/ programming the flash memory.	
	Password	Any data (8 bytes to 255 bytes) in the flash memory can be used as a password. If did not match password, generate error and stop RAM transfer.	A part of user memory use for password.
Flash memory erasing command	Flash memory erasing	Erases on-chip flash memory except user information area, regardless of a program/ erase protect condition or security status, without a password.	Erasing for; Data Flash Code Flash Protect bits Memory swap bits Security bit

UART (Note) of a target (TXZ microcontroller) and the external host controller (hereafter controller) are connected. The "flash reprogramming program" sent from the controller is stored in on-chip RAM. The "flash reprogramming program" on RAM is run, and a flash memory is reprogrammed.

The details of communication with the controller should follow the below mentioned protocol.

In single boot mode, do not generate all exceptions to avoid abnormal program termination.

To protect the contents of the flash memory in single chip mode (normal operation mode), it is recommended to protect relevant flash blocks against accidental erasure after reprogramming is complete.

Note: For detail of UART, please refer to "Asynchronous Serial Communication Circuit" of reference manual.

6.6.2. Mode Setting

In order to execute the on-board programming, boot up this device in single boot mode. For details of single boot mode setting, refer to “6.3 Mode Determination”.

6.6.3. Interface Specifications

This section describes the serial communication format in single boot mode. The serial operation mode supports UART (asynchronous communication). In order to execute the on-board programming, set the communication format of the programming controller as well.

Communication channel:	UART channel x (depends on the product)
Serial transfer mode:	UART (asynchronous communication) mode, half-duplex communication, LSB-first
Data length:	8 bits
Parity bit:	None
STOP bit:	1 bit
Baud rate:	Arbitrary baud rate (Table 6.7 Setting of baud rate in Single boot mode (fc=10MHz, No error))
WDT:	Stops

The clock/mode control block setting of the internal boot program operates on the initial settings(fc=10MHz, Clock are supplied to using function blocks). For details of the initial setting of the clock, refer to reference manual "Clock Control and Operation Mode" of each product.

A baud rate is determined by the 32-bit timer (T32A) mentioned in “6.6.6.1 Serial Operation Mode Determination”. At this time, a baud rate needs to be within the measurable range by the timer.

The pins used in the internal boot program are shown in “Table 6.4 Pins used in the internal boot program”. Other pins are not operated in the boot program.

For details of communication terminals, refer to reference manual "Product Information" of each product.

Table 6.4 Pins used in the internal boot program

Category	Pin name
Mode setting pin	BOOT_N
Reset pin	RESET_N
Communication pins	UTxTXD
	UTxRXD

Note1: UART channels to be used vary depending on the product. For details, refer to reference manual “Product Information”.

Note2: When two UART pins exist in the same channel, either UART pin connected with the host device is automatically detected at start-up in single boot mode (depending on the product). The RXD pin not used in the channel is set to OPEN or fixed to “High” level. Do not connect both UART pins to the host device at the same time.

6.6.4. Restrictions on Memories

Note that the single boot mode places restrictions on the on-chip RAM and on-chip flash memory as shown in "Table 6.5 Restrictions on the memories in single boot mode".

Table 6.5 Restrictions on the memories in single boot mode

Memory	Restrictions
On-chip RAM	Boot program uses the memory as a work area through "0x20000000" to "0x200003FF". Store the program from "0x20000400" through the end address which can be transmitted. For the last transfer address available, refer to "Product Information" in Reference manual.
Internal flash memory	From "0x5E001000" up to the (maximum capacity -N x3 -1) of Code flash can be used as the password area. (N is password length) Data flash cannot be used as the password area.

6.6.5. Operation Command

The boot program provides the following operation commands:

Table 6.6 Operation commands in single boot mode

Operation command data	Operation mode
0x10	RAM transfer
0x40	Flash memory erasing

6.6.5.1. RAM transfer

The RAM transfer is to store data from the controller to on-chip RAM. When the transfer is complete normally, a user program starts. The memory address of "0x20000400" or later can be used for a user program except "0x20000000" to "0x200003FF" where the addresses are used for the boot program. The execution start address means the start address to store data in the RAM.

This RAM transfer function can perform user's own on-board programming control. In order to execute the on-board programming by a user program, refer to "6.5How to Reprogramming the Flash".

6.6.5.2. Flash Memory Erasing

The flash memory erasing command erases the entire blocks of the flash memory except the user information area. This command erases data flash, code flash, protect bits, and security bit regardless of a program/erase protect condition or security status, without a password.

A user information area cannot be erased by the flash memory erasing command. If a user would like to erase this area, execute this command and then perform the RAM transfer to execute the user information area erasing program.

6.6.6. **Common Operation Regardless of the Command**

This section describes common operation under the boot program execution condition.

6.6.6.1. **Serial Operation Mode Determination**

The controller must send “0x86” on the 1st byte at the desired baud rate in Table 6.7. See “Figure 6.13 Serial operation mode determination data”. If communication is impossible, please set lower baud rate.

Table 6.7 Setting of baud rate in Single boot mode (fc=10MHz, No error)

Baud Rate (Calculation)	<BRN>	<BRK>
9600 (9599)	65	57
19200 (19203)	32	29
38400 (38388)	16	46
57600 (57637)	10	10
62500 (62500)	9	0
76800 (76923)	8	55
115200 (115274)	5	37
128000 (127796)	4	7

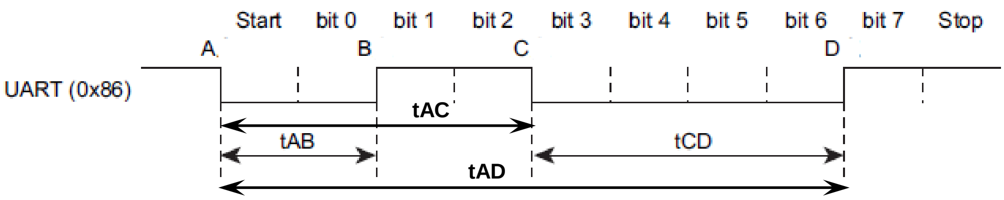


Figure 6.13 Serial operation mode determination data

“Figure 6.13 Serial operation mode determination data” shows a flowchart of internal boot program. Using 32-bit timer (T32A) with the time of tAB, tAC and tAD, the 1st byte of serial operation mode determination data “0x86” after reset is provided.

In “Figure 6.14 Reception flowchart in serial operation mode”, the CPU monitors the level of the receive pin, and obtains a timer value at the moment when the receive pin’s level is changed. Therefore, the timer values of tAB, tAC and tAD have a margin of error. Note that if a transfer goes at a high baud rate, the CPU may not be able to determine the level of receive pin.

“Figure 6.15 Serial operation mode determination flowchart” shows that the serial operation mode is determined by whether the time length of the receive pin is long or short. If the length is tAB ≤ tCD, the serial operation mode is determined as UART mode. The time of tAD is used for whether the automatic baud rate setting is enabled or not. If the length is tAB > tCD, the serial operation mode is not determined as UART mode. Note that timer values of tAB, tAC and tAD have a margin of error as mentioned before. If the baud rate goes high but operation frequency is low, each timer value becomes small. This may generate unexpected determination. (To prevent this problem, set the UART in the programming routine again if necessary.)

For example, the serial operation mode may not be determined as UART mode even when the controller attempts to use UART mode, or sometimes the data of the baud rate from the controller is not recognized. To avoid such situation, when UART mode is utilized, the controller should determine a time-out period where the time is expected to receive an echo-back "0x86" from the target board. If it fails to obtain that echo-back before the time-out, the controller should consider that the communication is disabled.

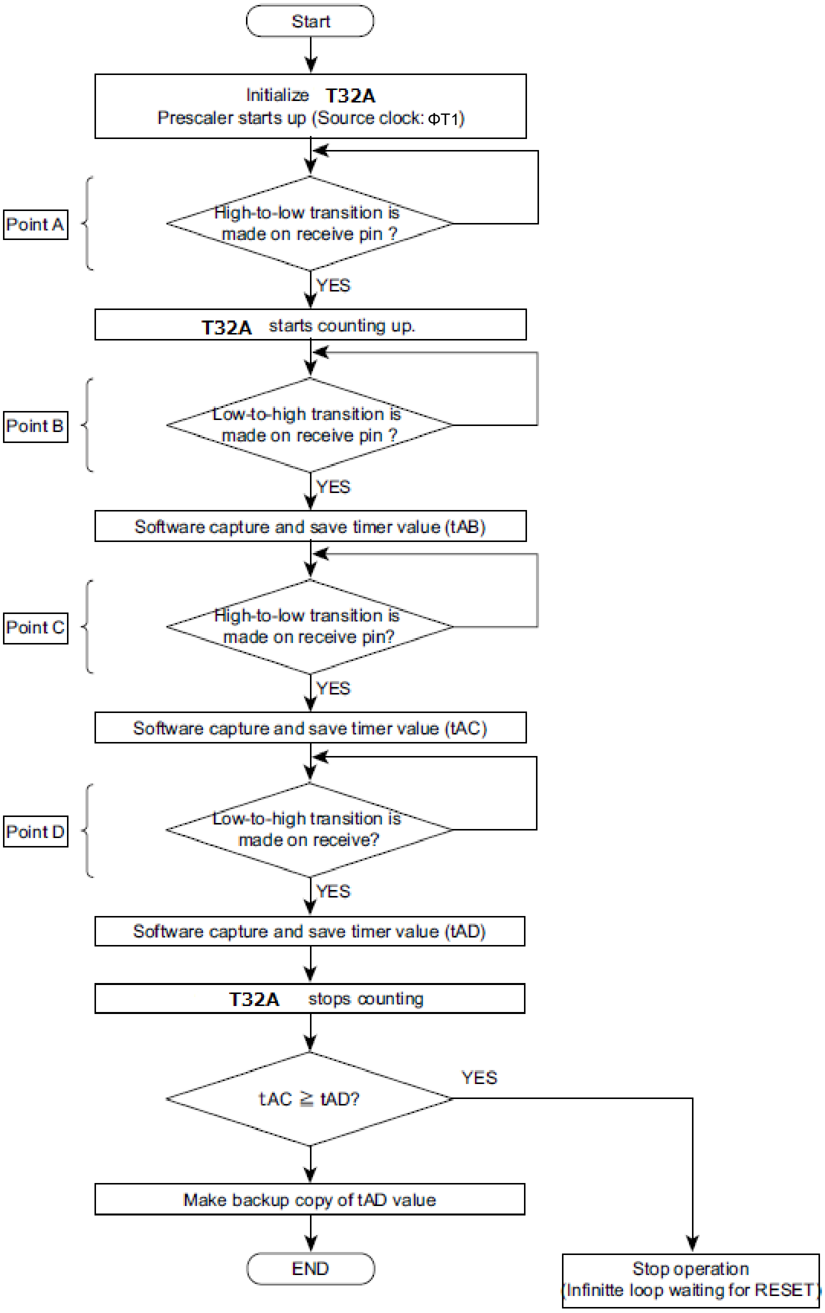


Figure 6.14 Reception flowchart in serial operation mode

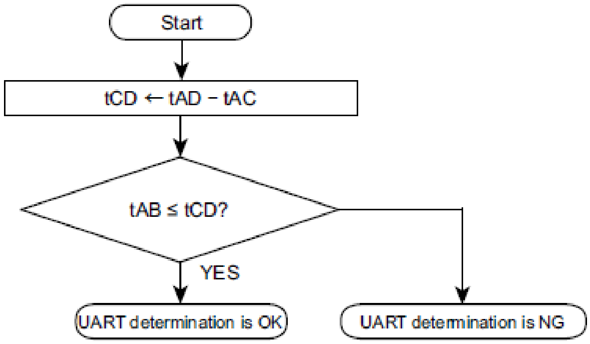


Figure 6.15 Serial operation mode determination flowchart

6.6.6.2. Acknowledgement Response Data

The internal boot program represents processing states in specific codes and sends them to the controller. From "Table 6.8 ACK response data corresponding to serial operation determination data" to "Table 6.11 ACK response data corresponding to flash memory erasing operation"Table 6.10 ACK response data corresponding to CHECKSUM dataTable 6.11 ACK response data corresponding to flash memory erasing operation response to each receive data.

The upper four bits of ACK response data are equal to the upper four bits of the operation command data. The bit 3 indicates a receive error. The bit 0 indicates an invalid operation command error, a checksum error or a password error. The bit 1 and bit 2 are always "0".

Table 6.8 ACK response data corresponding to serial operation determination data

Transmit data	Meaning
0x86	Determined that UART communication is possible. (Note)

Note: When the serial operation is determined as UART, if the baud rate setting is determined as unacceptable, the boot program aborts without sending back any response.

Table 6.9 ACK response data corresponding to operation command data

Transmit data	Meaning
0x?8(Note)	A receive error occurs in the operation command data.
0x?1(Note)	An undefined operation command data is received normally.
0x10	Determined as a RAM transfer command.
0x40	Determined as a flash memory erasing command.

Note: The upper 4 bits of the ACK response data are the same as those of the previous command data.

Table 6.10 ACK response data corresponding to CHECKSUM data

Transmit data	Meaning
0xN8(Note)	A receive error occurred in the operation command data.
0xN1(Note)	A CHECKSUM error or password error occurred.
0xN0(Note)	The CHECKSUM value was determined as correct value.

Note: The upper 4 bits of the ACK response data are the same as the operation command data.

Table 6.11 ACK response data corresponding to flash memory erasing operation

Transmit data	Meaning
0x54	Determined as a flash memory erase command.
0x4F	Erase command completed.
0x4C	Erase command completed illegally.
0x47	Flash operation command was aborted.

6.6.6.3. **Password**

Any data (a part of user memory) in the flash memory can be used as a password. Once the password is set, operation commands, including flash memory data reading or flash memory data programming in single boot mode, the password verification is required.

(1) Mechanism of Password

Arbitrary data (sequential data greater than 8 bytes or more) in the flash memory can be set a password. Password verification is performed by the comparison between the password sent from the external controller and the memory data in the MCU where the password is specified.

(2) Password Configuration

A password is comprised of four elements: PLEN, PNSA, PCSA, and a password string. See “Figure 6.16 Password configuration (Example of Transmission)”.

・PLEN (Password length data)

The length of a password is specified in the range of “0x08” to “0xFF”. A password error occurs when PLEN is set to less than “0x07”, address data of PNSA is less than “0x07”, or password length data does not match the address of PNSA.

・PNSA (Password store start address)

Use the address “0x5E001000” up to the maximum address in the unit of four bytes. Memory data of specified address is the number of bytes of password string. A password error occurs when address data of PNSA is less than “0x07”, or password length data does not match the address of PNSA. Memory data is defined as N.

・PCSA (Password compare start address)

Specify the address within “0x5E001000” up to (the maximum memory address-Nx4+1) in the unit of four bytes. Specified address is the start address to be compared with the password string. If the specified address by PCSA deviates from code flash, or specified address by PCSA+password length is over the above address, a password error occurs.

・Password string

Use “8” to “255” (=N) byte data. Memory data and password string are compared on the number of N bytes where the start address is specified by PCSA. If the comparison result is not matched, or the same data over 3 bytes are sequentially detected, a password error occurs.

・Password error

If a password error occurs, from then on the external device cannot communicate with the TXZ. To communicate with the TXZ, perform reset with the reset pin (RESET_N) to reboot the TXZ in single boot mode.

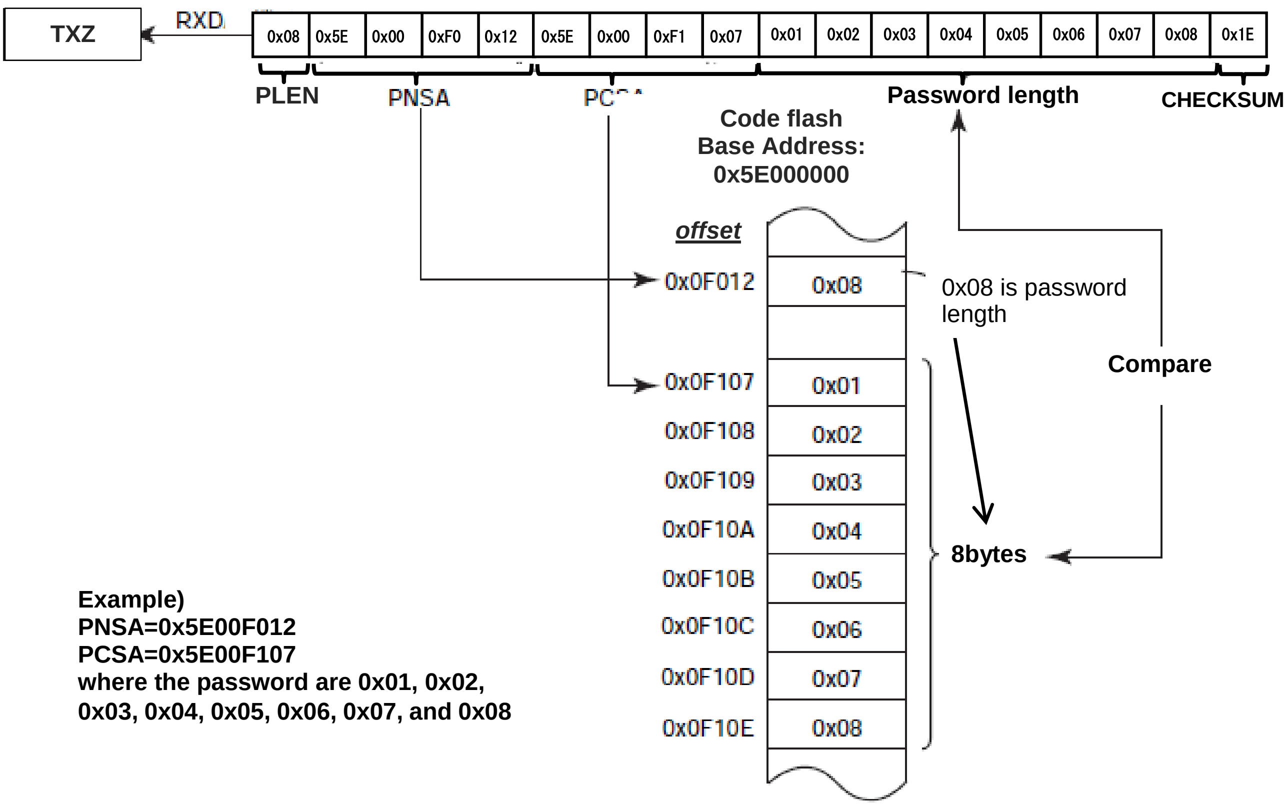


Figure 6.16 Password configuration (Example of Transmission)

(3) Password Setting/Releasing/Verification

・Password setting

Password system uses a part of a user program. Therefore, special process is not required for password setting. At the time when a password is programmed to the code flash, a password is set.

・Password releasing

To release a password, chip erasing (entire erasing) of code flash is required. A password is released at the time when the code flash is initialized to “0xFF” except the user information memory.

・The case where password verification is unnecessary.

When the entire area of the code flash and data flash are “0xFF”, the product is determined as a blank product. At this time, password verification is not performed.

For example, even if code flash area is all “0xFF” a password error occurs as long as data remains in data flash. In this case, perform chip erasing.

(4) Password Setting Values and Setting Ranges

A password must be set according to the condition described in “Table 6.12 Password setting values and setting ranges”. Unless the condition is met, a password error occurs.

Table 6.12 Password setting values and setting ranges

Password	Blank product (Note 1)	Non blank product
PNSA (Address where the number of passwords is stored)	0x5E001000 ≤PNSA ≤ Maximum memory address	0x5E001000 ≤ PNSA ≤ (Maximum memory address)
PCSA (Address where the start address used for password comparison)	0x5E001000 ≤ PCSA ≤Maximum memory address	0x5E001000 ≤ PCSA ≤ (Maximum memory address) – (N x 4) + 1
N (The number of passwords)	Necessary (Note 2)	8 ≤ N
Password	Necessary (Note 2)	Necessary (Note 1)

Note1: Over the same three bytes consecutive data cannot be used as a password string.
Note2: When the flash memory erasing command is used, a dummy password string should be sent to the blank product.

6.6.6.4. Password Determination

(1) Password verification using RAM transfer command

This item explains about the password determination No.5 described in “6.6.8. Communication Rules of RAM Transfer Command”.

If password area data deviates from the range of address, a password address error occurs. Also, if the same three bytes data or more are continued, or the case where data is not all “0xFF”, a password error occurs as shown in “Figure 6.17 Password check flowchart”. If a password address error or a password area error is determined, an ACK response is “0x11” regardless of the result of the password verification.

Then, received data (password data) is verified. Unless all N-byte data match the password in the flash memory, a password error occurs. If a password error occurs, an ACK response is a password error.

When the security function is enabled, password verification is performed.

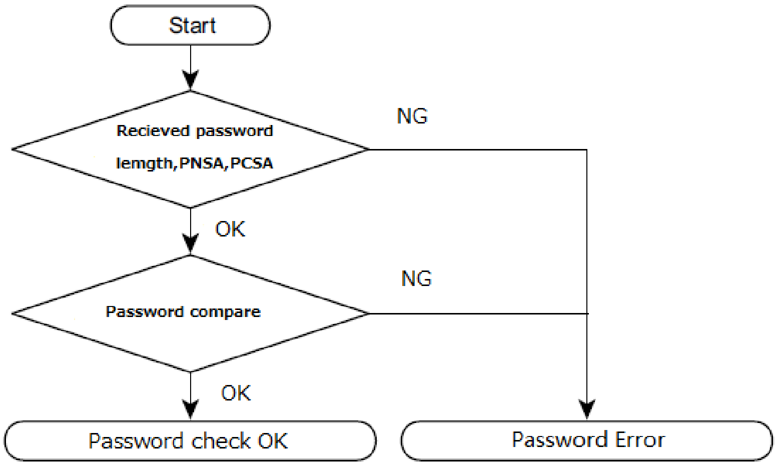


Figure 6.17 Password check flowchart

6.6.6.5. CHECKSUM Calculation

The CHECKSUM is calculated by 8-bit addition (ignoring the overflow) to transmit data and taking the two’s complement of the sum of lower 8 bits. Use this calculation when the controller transmits the CHECKSUM value.

Example calculation of CHECKSUM

To calculate the CHECKSUM for 2 bytes data (“0xE5” and “0xF6”), perform 8-bit addition without signed.

0xE5 + 0xF6 = 0x1DB

Take the two’s complement of the sum to the lower 8-bit, and that is a checksum value. So, “0x25” is sent to the controller.

0 - 0xDB = 0x25

6.6.7. Communication Rules for Determination of Serial Operation Mode

This section describes the communication rule for determination of serial operation mode. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): Controller to TXZ
Transfer direction (T→C): TXZ from controller

Table 6.13 Communication rules for determination of serial operation mode

No	Transfer direction	Transfer data	Description
1	C→T	Serial operation mode and baud rate setting	Controller transmits data to determine the serial operation mode. For details of mode determination of the target, refer to "6.6.6.1Serial Operation Mode Determination".
		0x86	Controller transmits "0x86". If the target determines UART mode is OK, the target successively determines whether baud rate setting is possible or not. If not, the program stops and communication is shutdown.
2	T→C	ACK response to serial operation mode	Receive data from the controller is ACK response data responding to the 1st byte of serial operation mode setting data. If the target determines the setting is possible, the target sets the UART. Data reception should be enabled before data is programmed to the transmit buffer.
		Normal state: 0x86	If the target determines the setting is possible, the target transmits "0x86". If the target determines the setting is not possible, the target transmits nothing, and stops the operation. Set a time out time (5 sec.) after the controller finished transmitting the 1st byte of data. If the controller does not receive data "0x86" properly within a time-out time, it should be determined as a communication failure.
3	-	-	Controller transmits operation command data. For details of transfer format of each operation command, refer to "6.6.8. Communication Rules of RAM Transfer Command" or "6.6.9. Communication Rules of Flash memory Erasing".

6.6.8. Communication Rules of RAM Transfer Command

This section shows communication rules of RAM transfer. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): From Controller to TXZ
Transfer direction (T→C): From TXZ to Controller

Table 6.14 Communication Rules of RAM Transfer Command

No	Transfer direction	Transfer data	Description
1	C→T	Operation command data (0x10)	Controller transmits RAM transfer command data “0x10”.
2	T→C	ACK response to the operation command Normal: 0x10 Abnormal: 0x11 Communication error: 0x18	The target checks received data, and it sends ACK response data. If a receive error exists, the target sends ACK response data “0x18” indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target checks the data against operation command data described in “Table 6.6 Operation commands in single boot mode”. If checking is failed, the target sends ACK response data “0x11” indicating abnormal state, and then returns to the initial state waiting for operation command data. If checking is succeeded, the target sends ACK response data “0x10” indicating normal state, and then it waits for next data.
3	C→T	Password length (PLEN) (1 byte)	The controller transmits password length data of the code flash.
4	C→T	Password length store address (PNSA) (4 bytes)	The controller transmits the address data where the password length is stored.
5	C→T	Password store start address (PCSA) (4 bytes)	The controller transmits the start address where the password is stored.
6	C→T	Password string (8 bytes to 255 bytes)	The controller transmits password data of the code flash. If it has been erased, the controller transmits dummy data.
7	C→T	CHECKSUM value of transmit data (No.3 to 6)	The controller calculates the CHECKSUM value of transmit data (No.3 to 6), and sends them. For details of CHECKSUM calculation, refer to “6.6.8 Communication Rules of RAM Transfer Command”.
8	T→C	Password length error check, password store address error check, password verification, ACK response to CHECKSUM value. - Blank: 0x14 - Normal: 0x10 - Abnormal: 0x11 - Communication error: 0x18	The target checks received data, and then it sends ACK response data. If a receive error exists, the target sends ACK response data “0x18” indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target checks a CHECKSUM value and password. For details of password verification, refer to “6.6.6.4. Password Determination”. If password determination is failed, the target sends ACK response data "0x11" indicating abnormal state, and then returns to the initial state waiting for operation command data. If password determination is succeeded, the target sends ACK response data “0x10” indicating normal state, and then it waits for next transmit data. In the case of blank products, ACK response data"0x14" is replied, and it waits

			for next transmit data.
9	C→T	RAM store start address (31 to 24)	The controller transmits the RAM start address to be stored in RAM store data by dividing into 4 times as a next transmit data. Transmission order is as follows: 1st byte corresponds to bit 31 to bit 24 and 4th byte corresponds to bit 7 to bit 0 of transfer address. These addresses should be placed in “0x20000400” through the last address of RAM address. The target checks receive data. If a receive error exists, the target sends ACK response data “0x18” indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target transmits nothing, and waits for next transmit data.
10	C→T	RAM store start address (23 to 16)	
11	C→T	RAM store start address (15 to 8)	
12	C→T	RAM store start address (7 to 0)	
13	C→T	The number of bytes where the RAM stores data (15 to 8)	The controller transmits the number of bytes to be block-transferred. Transmission order is as follows: 1st byte corresponds to bit 15 to bit 8 and 2nd byte corresponds to bit 7 to bit 0 of transfer address. These addresses should be placed in “0x20000400” through the last address of RAM address. The target checks receive data. If a receive error exists, the target sends ACK response data “0x18” indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target transmits nothing, and waits for next transmit data.
14	C→T	The number of bytes where the RAM stores data (7 to 0)	
15	C→T	A CHECKSUM value of transmit data (No.9 to 14)	The controller transmits a CHECKSUM value of transmit data (No.9 to 14).
16	T→C	ACK response to a CHECKSUM value Normal: 0x10 Abnormal: 0x11 Communication error: 0x18	The target checks receive data, and it sends ACK response data. If a receive error exists, the target sends ACK response data “0x18” indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target checks a CHECKSUM value. If checking is failed, the target sends ACK response data “0x11” indicating abnormal state, and then returns to the initial state waiting for operation command data. If checking is succeeded, the target sends ACK response data “0x10” indicating normal state, and then it waits for next data.
17	C→T	RAM store data	The controller transmits data to be stored in RAM from the controller. The target receives data to be stored in RAM.
18	C→T	A CHECKSUM value of transmit data (No.17)	The controller transmits a CHECKSUM value of transmit data (No.17).

19	T→C	ACK response to CHECKSUM verification - Normal: 0x10 - Abnormal: 0x11 - Communication error: 0x18	The target checks receive data, and it sends ACK response data. If a receive error exists, the target sends ACK response data "0x18" indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target checks a CHECKSUM value. If checking is failed, the target responds ACK response data "0x11" indicating abnormal state, and then returns to the initial state waiting for operation command data. If checking is succeeded, the target sends ACK response data "0x10" indicating normal state and jumps to RAM store start address (No.9 to 12) as a branch address.

Note: A setup of the functions (a port, UART, a timer, RAM, etc.) which the Boot ROM program used is not initialized.

6.6.9. **Communication Rules of Flash memory Erasing**

This section shows a communication format of flash memory erasing command. Transfer directions in the table are indicated as follows:

Transfer direction (C→T): From Controller to TXZ
Transfer direction (T→C): From TXZ to Controller

Table 6.15 Communication Rules of Flash memory Erasing

No	Transfer direction	Transfer data	Description
1	C→T	Operation command data (0x40)	The controller transmits flash memory erasing command data "0x40".
2	T→C	ACK response to operation command Normal: 0x40 Abnormal: 0x41 Communication error: 0x48	The target checks receive data, and it sends ACK response data. If a receive error exists, the target sends ACK response data "0x48" indicating communication error, and then returns to the initial state waiting for operation command data. If a receive error does not exist, the target checks a CHECKSUM value according to the operation commands shown in "Table 6.6 Operation commands in single boot mode". If checking is failed, the target responds ACK response data "0x41" indicating abnormal state, and then returns to the initial state waiting for operation command data. If checking is succeeded, the target sends ACK response data "0x40" indicating normal state, and waits for next data.
3	C→T	Erase enable command data (0x54)	The controller transmits erase enable command data "0x54".
4	T→C	ACK response to erase enable command data - Normal: 0x54 - Abnormal: 0x51 - Communication error: 0x58	The target checks receive data and, it sends ACK response data. If receive error exists, the target sends ACK response data "0x58" indicating abnormal communication, and then returns to the initial state waiting for operation command data. If receive error does not exist, the target checks an erase enable command "0x54". If checking is failed, the target responds ACK response data "0x51" indicating abnormal state, and then returns to the initial state waiting for operation command data. If checking is succeeded, the target sends ACK response data "0x54" indicating normal state, and performs chip erasing.
5	-	-	Chip erasing in progress.
6	T→C	ACK response to the checking for chip erasing - Erasing completed: 0x4F - Abnormal end (blank check error): 0x4C - Abnormal end (time-out error): 0x47	The target sends the result of chip erasing process. If any problems occur, the target sends ACK response data "0x4F" indicating normal state. If a blank check error occurs, the target sends ACK response data "0x4C" indicating abnormal state. If chip erasing command is aborted, the target sends ACK response data "0x47" indicating abort and then returns to the initial state waiting for operation command data.

6.6.10. Internal Boot Program General Flowchart

This section shows an internal boot program general flowchart.

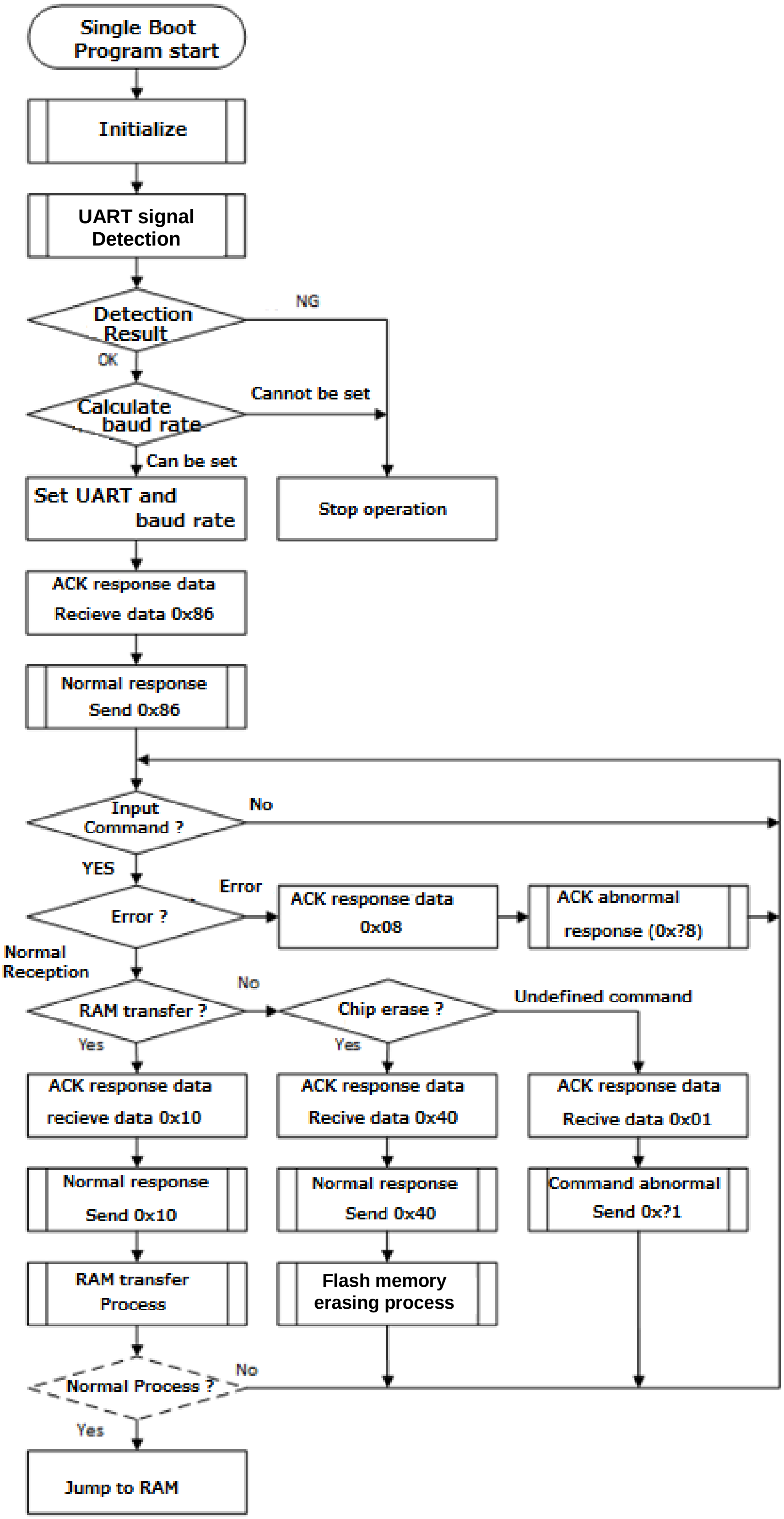


Figure 6.18 Boot program general flowchart

6.6.11. Reprogramming Procedure of the Flash Using Reprogramming Algorithm in Boot ROM

This section describes the reprogramming procedure of the flash using reprogramming algorithm in the on-chip Boot ROM. (The Following example is using UART)

6.6.11.1. Step-1

The condition of the flash memory does not care whether a former user program has been programmed or erased. Since a programming routine and programming data are transferred via the UART, the UART of this device must be connected to an external host. A programming routine (a) is prepared on the host.

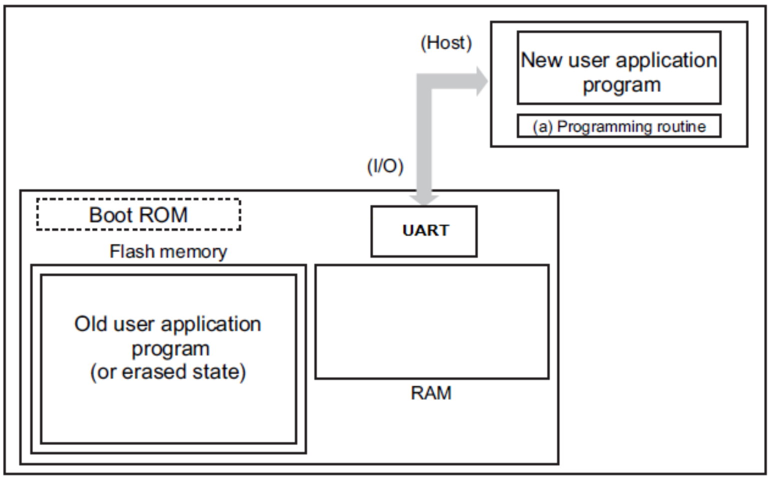


Figure 6.19 Procedure of Using Reprogramming Algorithm in Boot ROM (1)

6.6.11.2. Step-2

The user releases the reset by the pin condition setting for single boot mode and boots up on the Boot ROM. According to the procedure of boot mode, the user transfers the programming routine (a) via the UART from the source (host). Password verification is performed against the password in the user application program first. For details, refer to “(4) Password Setting Values and Setting Ranges” in section “6.6.6.3. Password”.

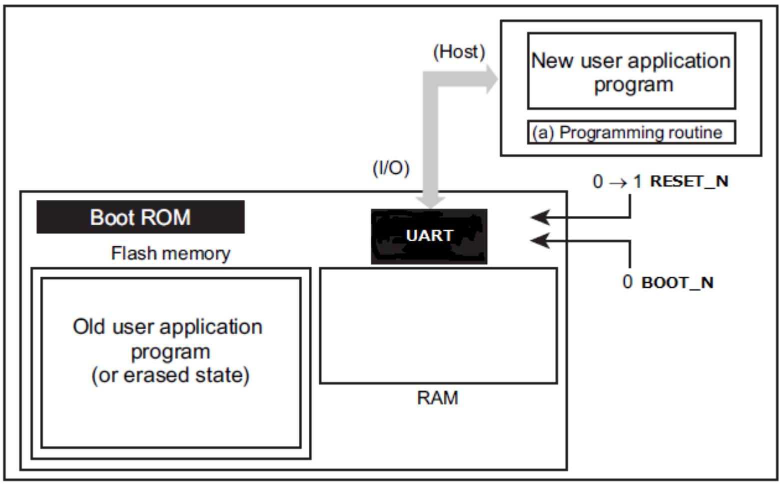


Figure 6.20 Procedure of Using Reprogramming Algorithm in Boot ROM (2)

6.6.11.3. **Step-3**

When the password verification is completed, the boot program transfers a programming routine (a) from the host into the on-chip RAM. The Boot ROM loads this routine to the on-chip RAM. The programming routine must be stored in the range from “0x20000400” to the end address which can be transmitted of the on-chip RAM.

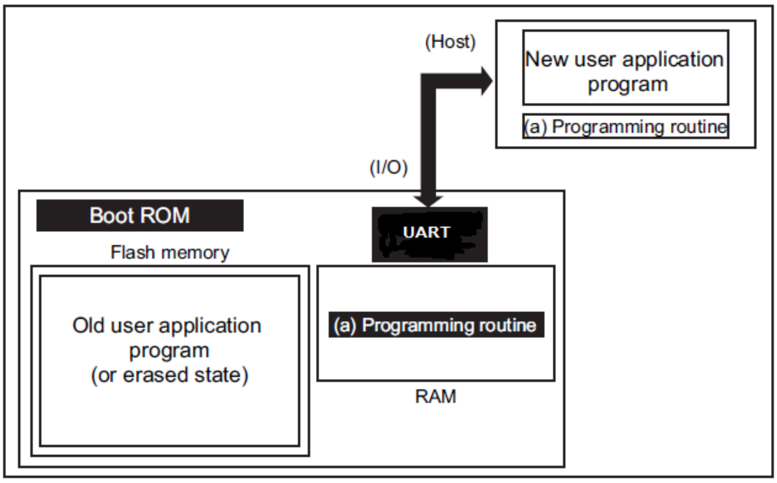


Figure 6.21 Procedure of Using Reprogramming Algorithm in Boot ROM (3)

6.6.11.4. **Step-4**

The boot program jumps to the programming routine (a) in the on-chip RAM to erase the flash block containing old application program codes (the units of erase is arbitrary size).

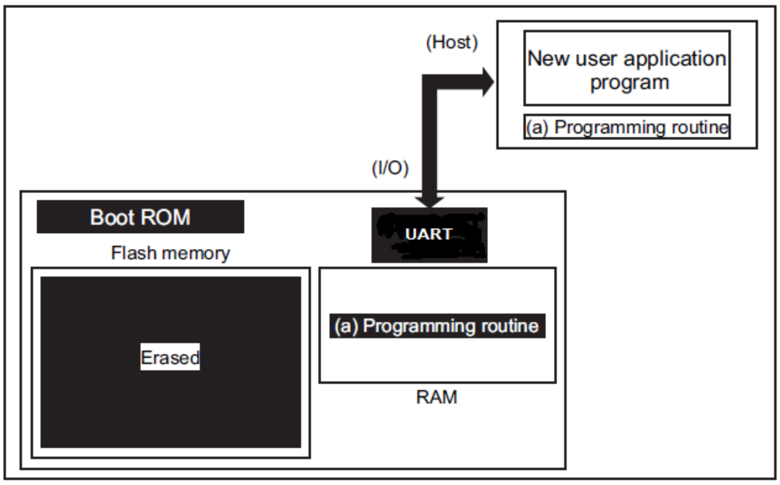


Figure 6.22 Procedure of Using Reprogramming Algorithm in Boot ROM (4)

6.6.11.5. **Step-5**

The boot program executes the programming routine (a) to download new application program codes from the host and programs it into the erased flash area. When the programming is completed, set the programming or erasing protection of that flash area in the user's program to ON.

In the example below, new program codes come from the same host via the same UART used when the programming routine has been transferred. However, once the programming routine starts operation, it is free to change the transfer path and the source of the transfer. The user can create a hardware board and programming routine to suit your particular needs.

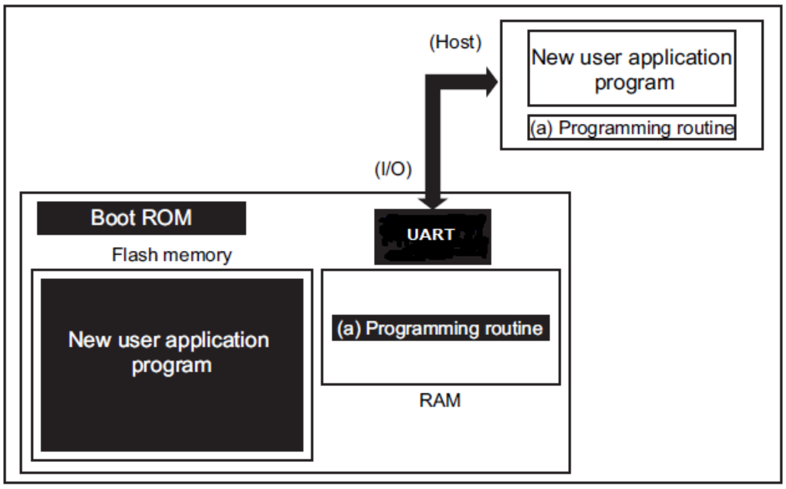


Figure 6.23 Procedure of Using Reprogramming Algorithm in Boot ROM (5)

6.6.11.6. **Step-6**

When programming of Flash memory is completed, the user shuts the power once and disconnects the cable connected with the host. The user then turns on the power again, so that the device re-boots in single-chip mode (normal mode) to execute the new program.

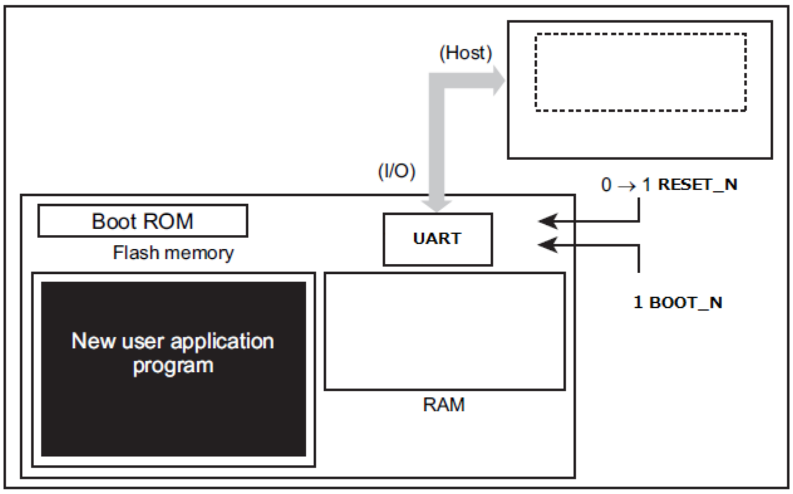


Figure 6.24 Procedure of Using Reprogramming Algorithm in Boot ROM (6)

6.7. How to Reprogramming using Dual Mode

The dual mode executes flash reprogramming using the flash memory reprogramming routine located in specified block on the users' set.

For example, while a program is executing on Area 0, another area (such as Area 4: data flash) of the flash memory, on which instructions are not executed, can be programmed/erased. (The opposite case is also possible.) Programming/erasing of the flash memory cannot be executed on the same area of the flash memory. Use different areas for programming/erasing of the flash memory.

When you use an exception in a dual mode, please mind not to perform accidentally the area which performs programming/erasing of a flash memory.

6.7.1. Example of Flash Memory Reprogramming Procedure

6.7.1.1. Step-1

A user determines the conditions (e.g., pin status) to enter the on-board programming and the target area in Flash memory to be programmed or erased. Then suitable circuit design and program are created along to the users' conditions.

(a)	Mode determination routine:	A program to determine to switch to user boot mode
(b)	Programming routine:	A program to download new program from the host controller and reprogram Flash memory.

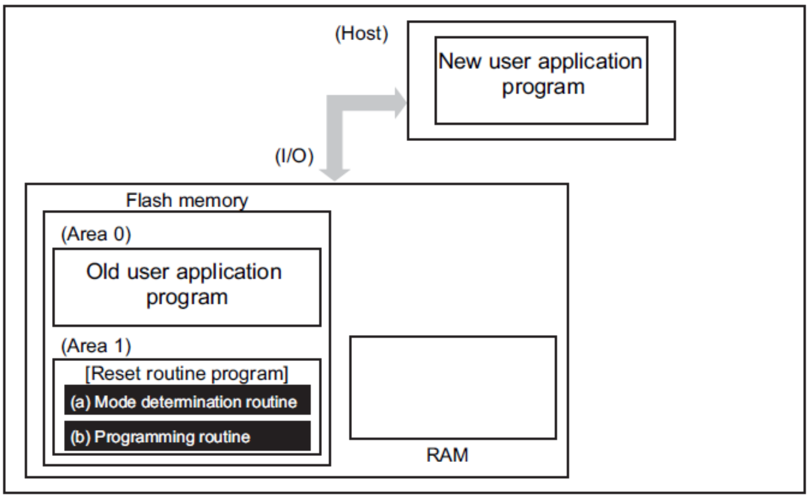


Figure 6.25 Reprogramming using Dual Mode (1)

6.7.1.2. Step-2

This section explains the case where a programming routine is stored in the reset routine. The reset routine determines to enter the dual mode. If mode switching conditions are met, the program jumps to the flash reprogramming routine to transfer to dual mode.

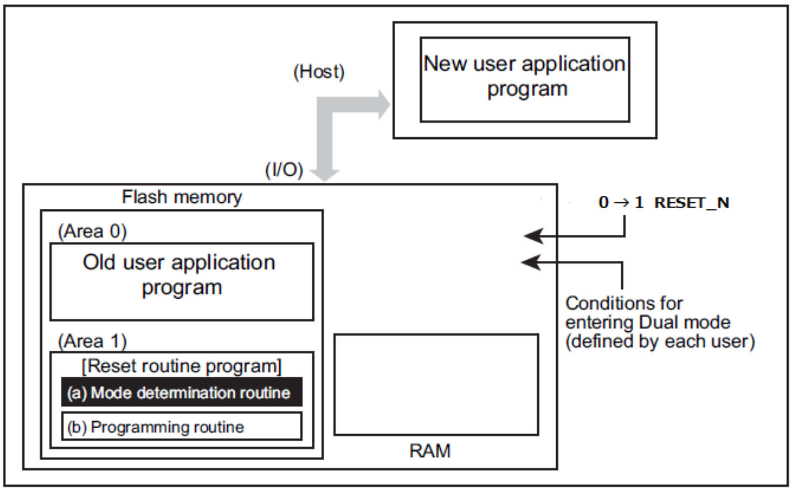


Figure 6.26 Reprogramming using Dual Mode (2)

6.7.1.3. Step-3

After the program jumps to the flash reprogramming routine, the program releases the program/erase protection in the old user program area and erases the areas in unit of the area, block, or page.

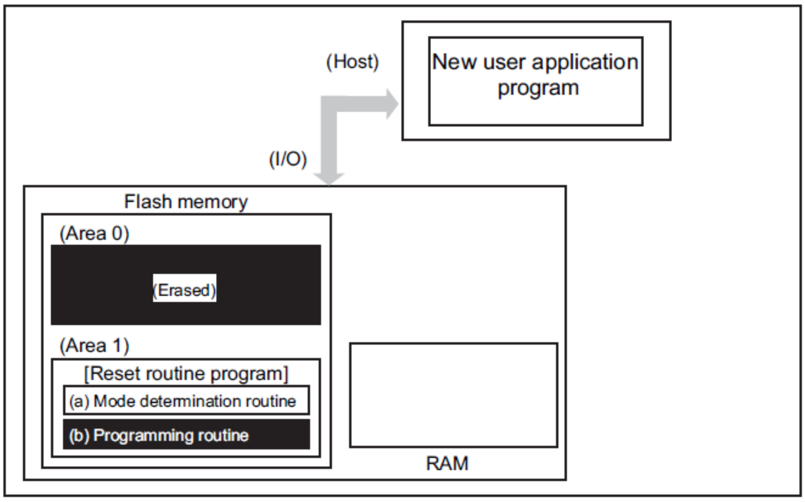


Figure 6.27 Reprogramming using Dual Mode (3)

6.7.1.4. **Step-4**

Subsequently, confirm whether the erased area of the flash are blank, and then downloads a new user's application program data from the transfer source (Host) to develop it on the RAM.

Developed data on the RAM is written to the erased area of the flash memory. When all data programming is completed, set the program/erase protection of that flash block in the user program area to ON.

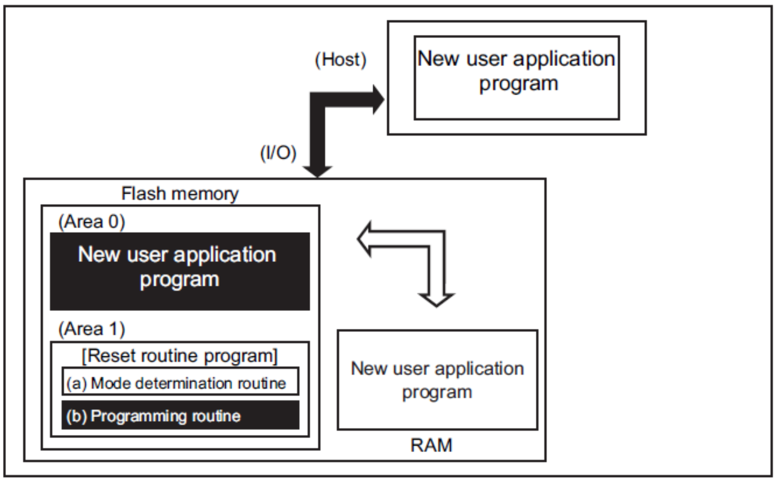


Figure 6.28 Reprogramming using Dual Mode (4)

6.7.1.5. **Step-5**

Assert reset by setting "0" to the RESET_N pin. Upon reset, the flash memory is set to normal mode. After reset, the CPU will start operation along with the new application program.

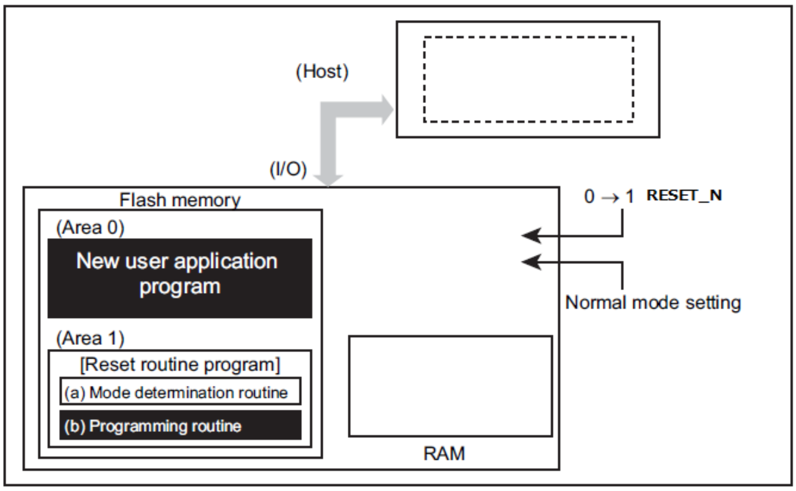


Figure 6.29 Reprogramming using Dual Mode (5)

6.8. How to Reprogramming User Boot Program

This method switches the Page 0 area to Page 1 area to hold a user boot program using the memory swap function when Flash memory is reprogrammed.

The following is an example of reprogramming procedure of user boot program.
(Assumed conditions: Swap size is 4K bytes. Page 1 program is copied from Page 0.)

6.8.1. Example of Flash Memory Reprogramming Procedure

6.8.1.1. Step-1

The user confirms whether "00" is read from ***[FCSWPSR]***<SWP[1:0]>.

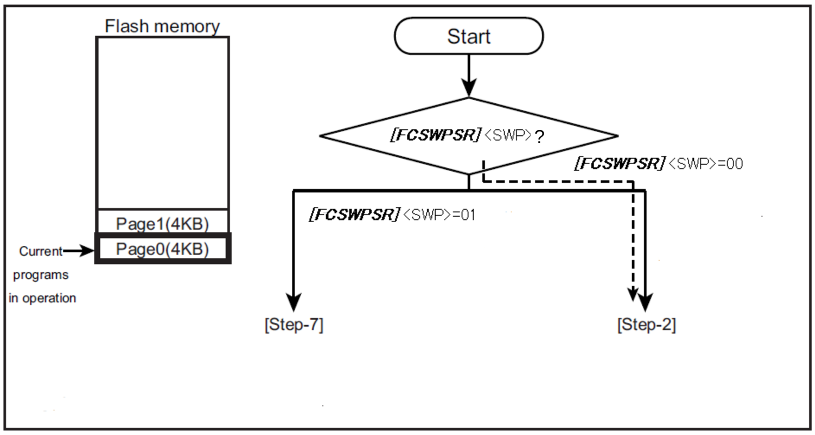


Figure 6.30 Reprogram by User Boot Program (1)

6.8.1.2. Step-2

The user checks ***[FCPSR0]***<PG1>=0. If protection status enabled then write "0" to ***[FCPMR0]***<PM1> for temporary release protection.

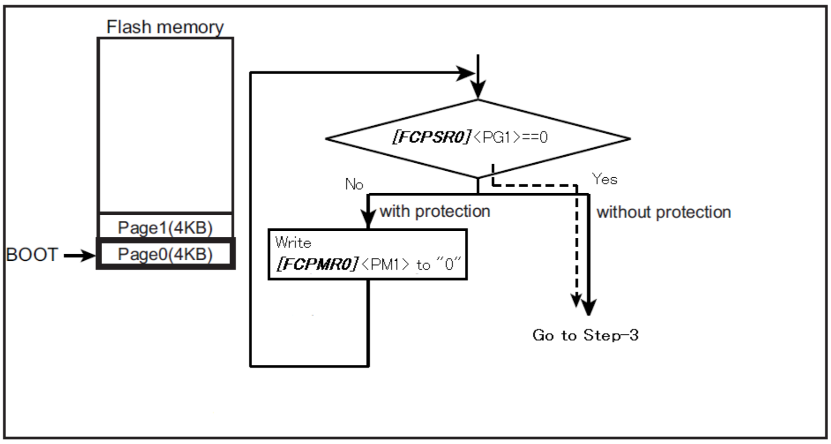


Figure 6.31 Reprogram by User Boot Program (2)

6.8.1.3. **Step-3**

The user transfers the reprogramming routine to the on-chip RAM, and moves the PC (Program Counter) to the transferred program.

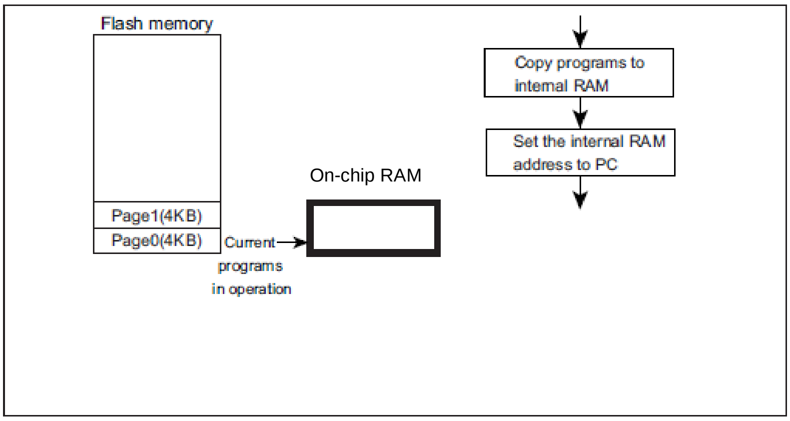


Figure 6.32 Reprogram by User Boot Program (3)

6.8.1.4. **Step-4**

The user erases Page 1, and then copy a program of Page 0 to program of Page 1.

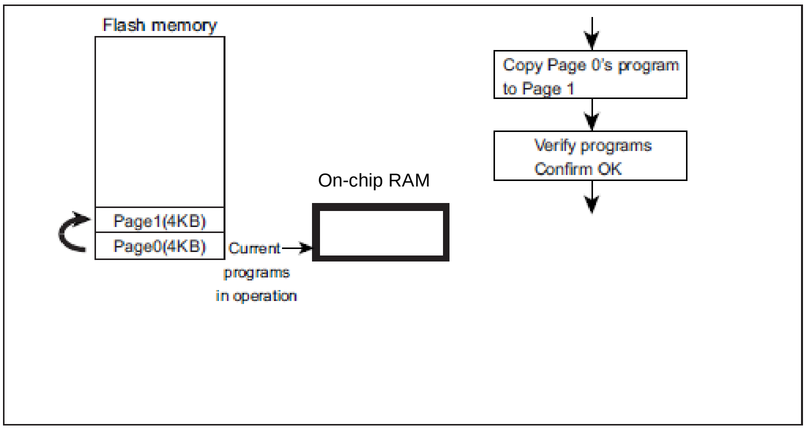


Figure 6.33 Reprogram by User Boot Program (4)

6.8.1.5. **Step-5**

The automatic memory swap command sets “01” to ***[FCSWPSR]***<SWP[1:0]> to swap Page 0 with Page 1.

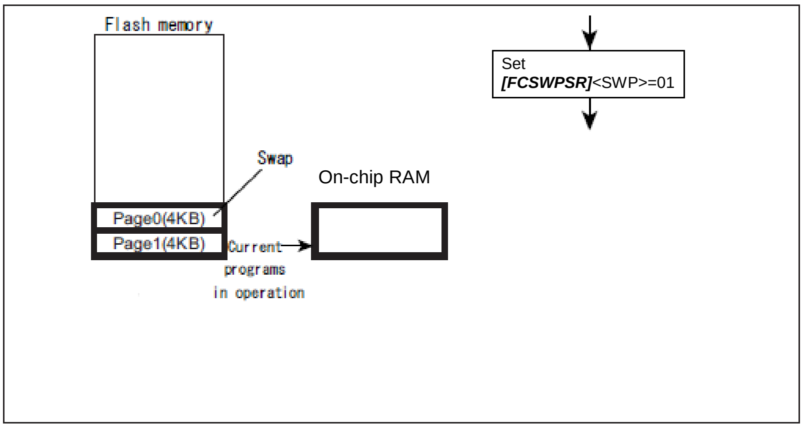


Figure 6.34 Reprogram by User Boot Program (5)

6.8.1.6. **Step-6**

The user performs a reset or releases a reset condition.
Page 1 is assigned to address 0 and the flash memory boots up at Page1.
A program branches to the conditioning routine where ***[FCSWPSR]***<SWP[1:0]> is set to “01” (To [Step-7]).

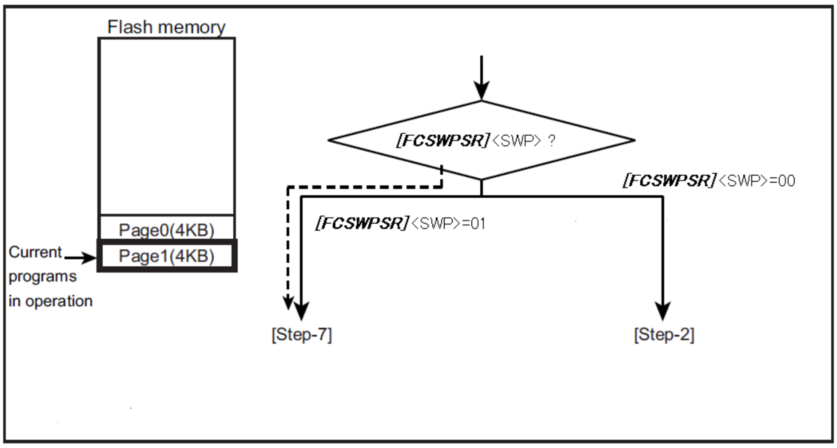


Figure 6.35 Reprogram by User Boot Program (6)

6.8.1.7. Step-7

The user checks ***[FCPSR0]***<PG1>=0. If protection status enabled then write “0” to ***[FCPMR0]***<PM1> for temporary release protection.

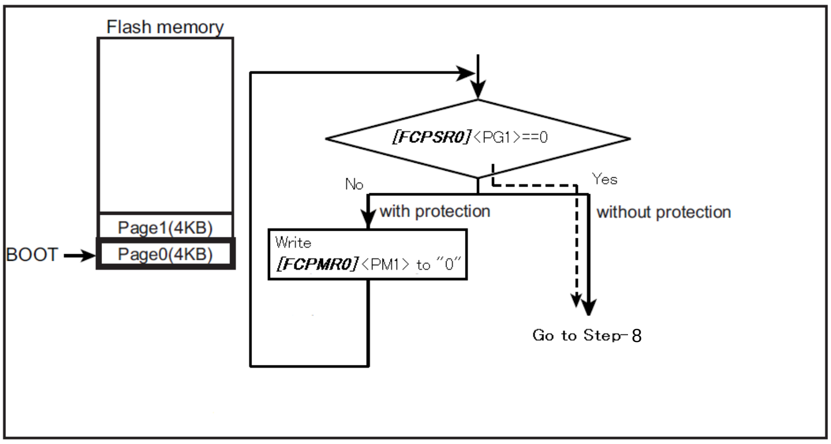


Figure 6.36 Reprogram by User Boot Program (7)

Note: Protection function performs to address. Then when memory swapped between Page0 and Page1, <PG0>/<PM0> is Page1 and <PG1>/<PM1> is Page0.

6.8.1.8. Step-8

The user transfers the flash reprogramming routine to the on-chip RAM, and then sets the on-chip RAM address to PC (Program Counter).

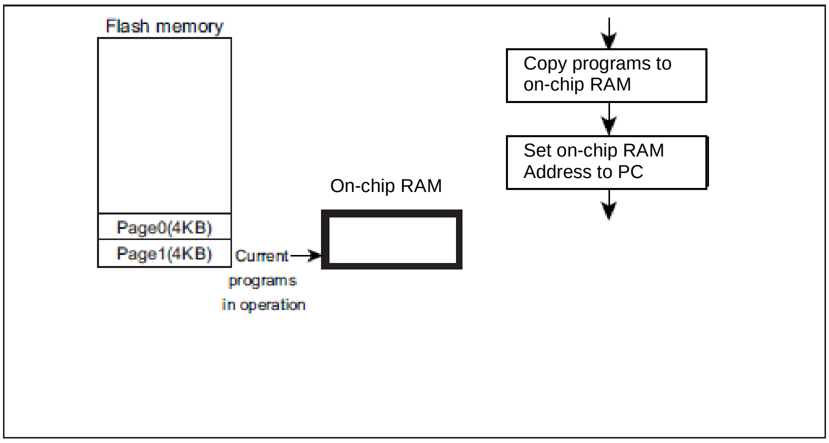


Figure 6.37 Reprogram by User Boot Program (8)

6.8.1.9. **Step-9**

The user programs a new boot program to Page 0.

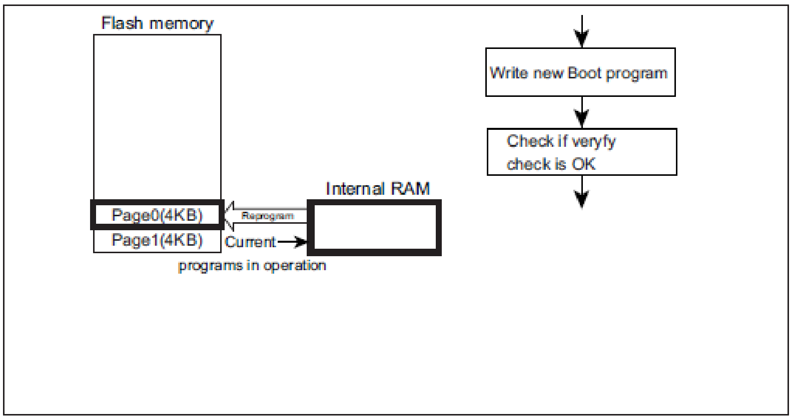


Figure 6.38 Reprogram by User Boot Program (9)

6.8.1.10. **Step-10**

The automatic memory swap command sets “11” to ***[FCSWPSR]***<SWP[1:0]> to swap release Page 0 and Page 1.

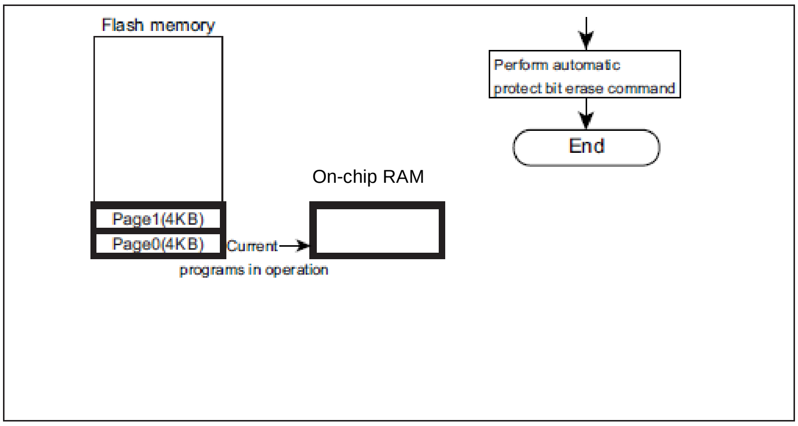


Figure 6.39 Reprogram by User Boot Program (10)

7. General Precautions

- Do not perform any operation that is not described in this document.
- Do not access the addresses described in this document that is not assigned to the registers.

8. Revision History

Table 8.1 Revision history

Revision	Date	Description
1.0	2017-10-12	First release
2.0	2018-7-20	-1.: Modified description. “which saves data” to “which stores data”. -1.1.: Modified description “Read” to “Read/Execute” in Figure1.1. -2.2.2.: Modified description Note of Table2.2/2.3/2.4. -3.: Modified description in “Precautions”. (1)“high-speed” to “high speed”, “Make sure to” to “Make sure to set ***[CGOSCCR]***<IHOSCEN1>=1 to”, (5) “Do not reset” to “Avoid reset that”. -3.1.1.1.: Added “(Note)” in Table 3.2. -3.1.1.2.: Added description protect bit/memory swap/security bit. -3.2.1.2.: Corrected in Table3.8 “Adr[16:0]” to “Adr[14:0] area erasing, “Adr[5:2]” to “Adr[7:2] protect bit, explanation of protect bit, -3.3.1.: Modified Figure3.1/3.2, Modified Name of Figure3.1/3.2. -3.3.2.: Modified Figure3.3/3.4 Modified Name of Figure3.3/3.4. -3.3.3./3.3.4./3.3.5.: Added these sections. -4.1.6.3.:Modified referring section. -4.1.7.2.: Corrected “and security bits are erased” to “and security bit is erased” -4.1.7.3.: Added “or written” in Table4.3 -4.1.9.1./4.1.9.2./4.1.2.3: Corrected “User Information Memory” to “User Information Area”. -4.1.9.2./4.1.9.3.: Added description. “by step (5) of “4.1.9.1”” -5.1.: Added section, Deleted Table name and Note. -5.2.: Modified section name. -5.2.1.:Modified description “on the power-on reset” to “on the Power On Reset” -5.2.4.: Modified Table -5.2.9.:Corrected Write as “1” -5.2.10.: Modified Table. Separated [31:8] to [31:16]/[15:8]. -5.2.12.: Modified Table -5.2.13.: Modified Table. Description in <SSF4><SSF0> , Separated [15:12] to [15]/[14:12]. -5.2.14: Changed Note order -5.2.17 Modified explanation of function: “user information memory” to “use information area” -6.6.4: Added description in Table 6.5 -6.6.5.1.: Added description. -6.6.6.2.: “CHECK SUM” to “CHECKSUM” in Table6.10. -6.6.6.3.: “Checksum” to “CHECKSUM” in Figure6.16. -6.6.6.5./6.6.8./6.6.9.: “CHECK SUM” to “CHECKSUM” -6.6.8./6.6.9.: Modified Table6.14/6.15 -6.6.10.:Modified Figure6.18
2.1	2022-07-07	-Table 1.2 Functional Description (User Information Area) Program/erase protection is deleted.

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS**.

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION
https://toshiba.semicon-storage.com/