

CONFIDENTIAL

RH850/D1x Device Family
Renesas Graphics Library
Serial Flash Memory Interface A
(SFMA) driver

User's Manual: Software

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

CONFIDENTIAL

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
 2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
 3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
 4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
 5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
 6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
 7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
 8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
 9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
 10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
 11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
 12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.
- (Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
- (Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

CONFIDENTIAL

Trademark

- Green Hills, the Green Hills logo, INTEGRITY, MULTI, DoubleCheck, EventAnalyzer, Integrate, SuperTrace, ResourceAnalyzer, CodeFactor, INTEGRITY MULTIVisor, GMART, GSTART, G-Cover, PathAnalyzer, GHNet, TimeMachine, μ -veLOsity, Padded Cell, TotalDeveloper, and Optimizing Compiler are trademarks or registered trademarks of Green Hills Software in the US and/or internationally.
- This software contains the technology owned by TES Electronic Solutions GmbH. All rights reserved for TES Electronic Solutions GmbH
- Trademarks and trademark symbols (® or ™) are omitted in the text of this manual.

CONFIDENTIAL

How to Use This Manual

1. Purpose and Target Readers

This manual is designed to provide the user with an understanding the functions of SFMA driver. This manual is written for engineers who use SFMA driver.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

Please refer to documents of drivers and hardware for a target system implementing SFMA as necessary.

The following documents are related documents. Make sure to refer to the latest versions of these documents.

Document Type	Description	Document Title	Document No.
User's manual for Hardware	Hardware specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description	RH850/D1L/D1M Group User's Manual: Hardware	R01UH0451EJ0220
User's manual for Software	Description of RGL overview	Renesas Graphics Library User's Manual: Software	R01US0181ED0400
	Description of WM	Renesas Graphics Library Window Manager (WM) Driver User's Manual: Software	LLWEB-10035990
	Description of SPEA	Renesas Graphics Library Sprite Engine A (SPEA) Driver User's Manual: Software	LLWEB-10035991
	Description of VDCE	Renesas Graphics Library Video Data Controller E (VDCE) Driver User's Manual: Software	LLWEB-10035992
	Description of VOWE	Renesas Graphics Library Video Output Warping Engine (VOWE) Driver User's Manual: Software	LLWEB-10035993
	Description of JCUA	Renesas Graphics Library JPEG Codec Unit A (JCUA) Driver User's Manual: Software	LLWEB-10035994
	Description of SFMA	Renesas Graphics Library Serial Flash Memory Interface A (SFMA) Driver User's Manual: Software	LLWEB-10064753 (This manual)
	Description of HYPB	Renesas Graphics Library HyperBus Controller (HYPB) Driver User's Manual: Software	LLWEB-10064754
	Description of OCTA	Renesas Graphics Library OctaBus Controller (OCTA) Driver User's Manual: Software	LLWEB-10064755
	Description of VOCA	Renesas Graphics Library Video Output Checker A (VOCA) Driver User's Manual: Software	LLWEB-10063801

CONFIDENTIAL

	Description of DISCOM	Renesas Graphics Library Display Output Comparator (DISCOM) Driver User's Manual: Software	LLWEB-10063802
	Description of DRW2D	Renesas Graphics Library 2D Graphics (DRW2D) Driver User's Manual: Software	LLWEB-10059472
Porting Layer Guide	Description of porting layer of RGL	Renesas Graphics Library Porting Layer Guide	LLWEB-10035995

CONFIDENTIAL

2. Notation of Numbers and Symbols

This manual uses the following notation.

Binary 0bXXXXXXXX (X=0 or 1)
Decimal XXX (X=0-9)
Hex 0XXXXXXXX (X=0-9,A-F)

CONFIDENTIAL

3. List of Abbreviations and Acronyms

Abbreviation	Full Form
API	Application Programming Interface
DDR	Double Data Rate.
H/W	Hardware
I/O	Input / Output.
SDR	Single Data Rate.
SFMA	Serial Flash Memory Interface A.
SPI	Serial Peripheral Interface.

All trademarks and registered trademarks are the property of their respective owners.

Table of Contents

1. Overview	3
1.1 Feature and Scope	3
1.2 Component Structure	3
2. Basic Specification	4
2.1 Summary Specification	4
2.2 Reserved Word	4
2.3 Interrupt Handler List	5
2.4 Error Handling	5
2.4.1 Return code	5
2.4.1.1 Parameter level	5
2.4.1.2 Timing level	5
2.4.1.3 System level	5
2.4.1.4 Hardware level	5
2.4.1.5 Device level	5
2.5 State Transition	6
3. Function Description	8
3.1 Fundamental Concepts	8
3.1.1 SFMA unit	8
3.1.2 System Configuration	8
3.1.3 Operating Mode	11
3.1.3.1 External address space read	11
3.1.3.2 SPI operating	11
3.1.4 Dependence command of the serial flash memory	12
3.2 Using the API	14
3.2.1 Initialization / De-Initialization	14
3.2.2 External address space mode	15
3.2.3 SPI operating mode	16
3.2.4 Access Addresses	17
3.2.5 Calibration	17
3.2.6 Data Erase	18
3.3 Device difference	19
3.4 Header File List	20
4. Functions	21
4.1 Function List	21
4.2 SFMA API Functions	22
4.2.1 Basic functions	22
4.2.1.1 R_SFMA_Init	22
4.2.1.2 R_SFMA_DeInit	24
4.2.1.3 R_SFMA_Open	26
4.2.1.4 R_SFMA_Close	28
4.2.1.5 R_SFMA_AccessAddressSet	30
4.2.1.6 R_SFMA_AccessAddressGet	32
4.2.1.7 R_SFMA_ProtectionModeSet	34
4.2.1.8 R_SFMA_DataErase	36
4.2.1.9 R_SFMA_DataWrite	38
4.2.1.10 R_SFMA_DataRead	40
4.2.1.11 R_SFMA_JEDECRead	42
4.2.1.12 R_SFMA_VersionStringGet	44
4.2.1.13 R_SFMA_MacroVersionGet	45

4.2.1.14	R_SFMA_GetCal.....	46
4.2.1.15	R_SFMA_GetStatus.....	47
4.2.1.16	R_SFMA_GetConfig	49
4.2.1.17	R_SFMA_WriteStatusConfig	51
4.2.1.18	R_SFMA_WriteReset	53
4.2.1.19	R_SFMA_UnitWindowSelect.....	55
4.2.1.20	R_SFMA_UnitNumberGet	57
4.2.2	Interrupt functions	59
5.	Types	60
5.1	Basic Types	60
5.2	Definition	60
5.3	Enumerated Type	61
5.3.1	r_sfma_Error_t	61
5.3.2	r_sfma_MemoryNum_t	62
5.3.3	r_sfma_Mode_t	63
5.3.4	r_sfma_DataTransferMode_t	64
5.3.5	r_sfma_ProtectionMode_t	65
5.3.6	r_sfma_AddressMode_t	66
5.3.7	r_sfma_AccessRange_t	67
5.3.8	r_sfma_FlashRegister_t	68
5.3.9	r_sfma_DummyCycle_t	69
5.3.10	r_sfma_AddressBitSize_t	71
5.3.11	r_sfma_CacheMode_t	72
5.3.12	r_sfma_WindowMode_t	73
5.4	Structure Type	74
5.4.1	r_sfma_FlashRegInfo_t	74
5.4.2	r_sfma_FlashRegSetParam_t	75
5.4.3	r_sfma_FlashCommand_t	76
5.4.4	r_sfma_Config_t	78

1. Overview

1.1 Feature and Scope

The Serial Flash Memory Interface outputs control signals to the serial flash memory connected to the SPI multi I/O bus space, thus enabling direct connection of the serial flash memory.

This module allows the connected serial flash memory to be accessed by directly reading the SPI multi I/O bus space, or using SPI operating mode to transmit and receive data.

1.2 Component Structure

The component structure of SFMA is shown in [Figure 1-1](#).

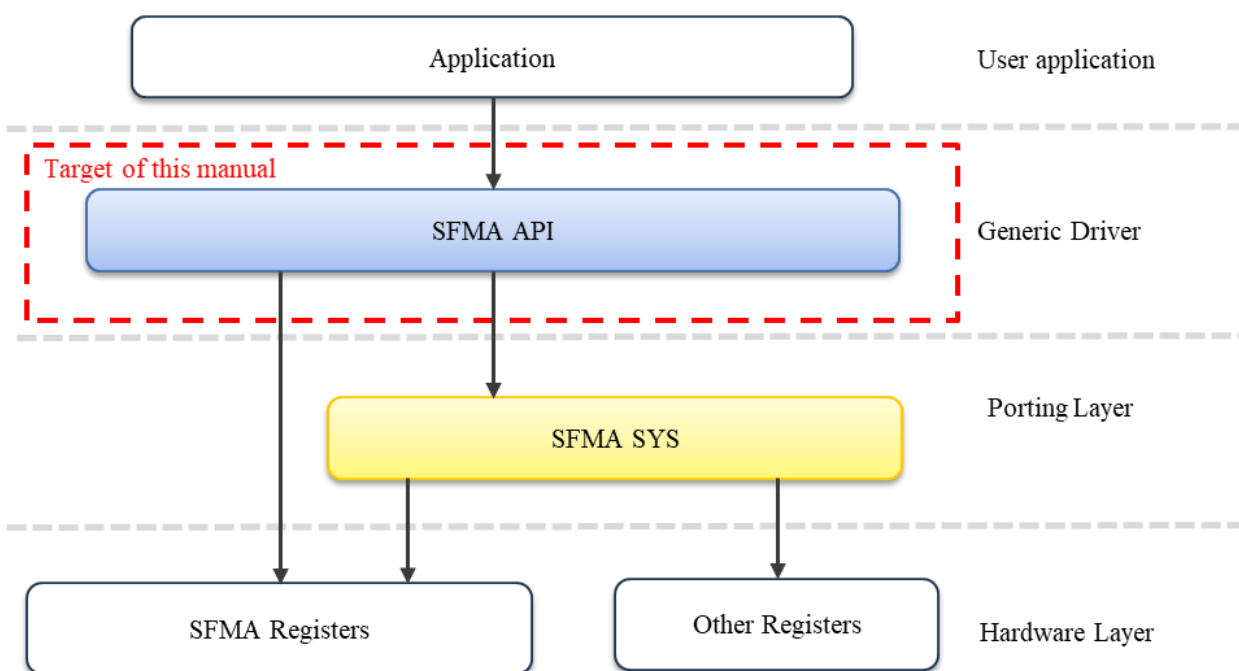


Figure 1-1 Component Structure

For the details of the API, please refer to [Chapter 4](#).

2. Basic Specification

2.1 Summary Specification

The summary of specification is described in [Table 2-1](#).

Table 2-1 Summary Specification

Items	Description
Target LSI	RH850/D1L2(H), RH850/D1M1(H), RH850/D1M1-V2, RH850/D1M1A, RH850/D1M2(H)
Main Feature	<ul style="list-style-type: none"> • Number of connected devices <ul style="list-style-type: none"> ○ Up to two serial flash memories per unit can be connected. • Data bus width <ul style="list-style-type: none"> ○ A data bus size of 1 bit, 2 bits, or 4 bits can be selected for one serial flash memory device. • Data transfer mode <ul style="list-style-type: none"> ○ SDR / DDR • Operating mode <ul style="list-style-type: none"> ○ External Address Space Read Mode <ul style="list-style-type: none"> - A maximum of 8-Gbyte address space is supported (when two serial flash memories are connected) - The SPBSSL pin can be automatically controlled through access address monitoring - Efficient data reception due to built-in read cache (64-bit line x 16 entries) ○ SPI Operating Mode <ul style="list-style-type: none"> - Desired read/write access to serial flash memory possible • Bit rate <ul style="list-style-type: none"> ○ SPBCLK is generated by frequency division of BΦ by internal baud rate generator ○ SPBCLK frequency division ratio can be set from 2 to 4080
Semaphore / Mutex	N/A. This can be implemented with porting layer.
Interrupts	N/A.

2.2 Reserved Word

SFMA uses the following prefixes for avoiding confusion from other software. Prefixes of SFMA is described in [Table 2-2](#).

Table 2-2 Prefixes

Prefix	Description
R_SFMA_*	Prefix for SFMA Module
r_sfma_*	

2.3 Interrupt Handler List

None.

2.4 Error Handling

2.4.1 Return code

SFMA driver has 5 types of error codes.

2.4.1.1 Parameter level

Following errors occur by a cause such as abnormality of parameter. In this case, please set valid parameter again.

- R_SFMA_ERR_PARAM_INCORRECT
- R_SFMA_ERR_RANGE_UNIT
- R_SFMA_ERR_RANGE_PARAM

2.4.1.2 Timing level

Following errors occur by a cause such as abnormality of execution timing. In this case, please call again after changing to valid state or timing.

- R_SFMA_ERR_NOT_ACCEPTABLE
- R_SFMA_ERR_NOT_SUPPORT_CLOCK
- R_SFMA_ERR_NOT_SUPPORT_TRANSFER
- R_SFMA_ERR_SFLASH_PROTECTED

2.4.1.3 System level

Following errors occur by a cause such as OS dependent error (e.g. system call error, resource shortage). In this case, please do recovery processing from a system layer, because this status cannot be restored only in this library.

- R_SFMA_ERR_FATAL_OS

2.4.1.4 Hardware level

Following errors occur when unexpected error occurs internally. In this case, please reset the RH850/D1x device.

- R_SFMA_ERR_NG
- R_SFMA_ERR_FATAL_HW

2.4.1.5 Device level

Following errors occur when the function is not supported with target device. In this case, please skip the function call.

- R_SFMA_ERR_NOT_SUPPORTED

2.5 State Transition

Each SFMA unit has following status.

Table 2-3 SFMA unit State Details

No.	State Name	Description
(1)	Uninitialized	Specifies that the SFMA driver is not initialized.
(2)	Initialized	Specifies that the SFMA driver is initialized.
(3)	Executing (SPI operating)	Specifies that SPI operating mode is enabled.
(4)	Executing (External address space read)	Specifies that External address space read mode is enabled.

The image describes state transition.

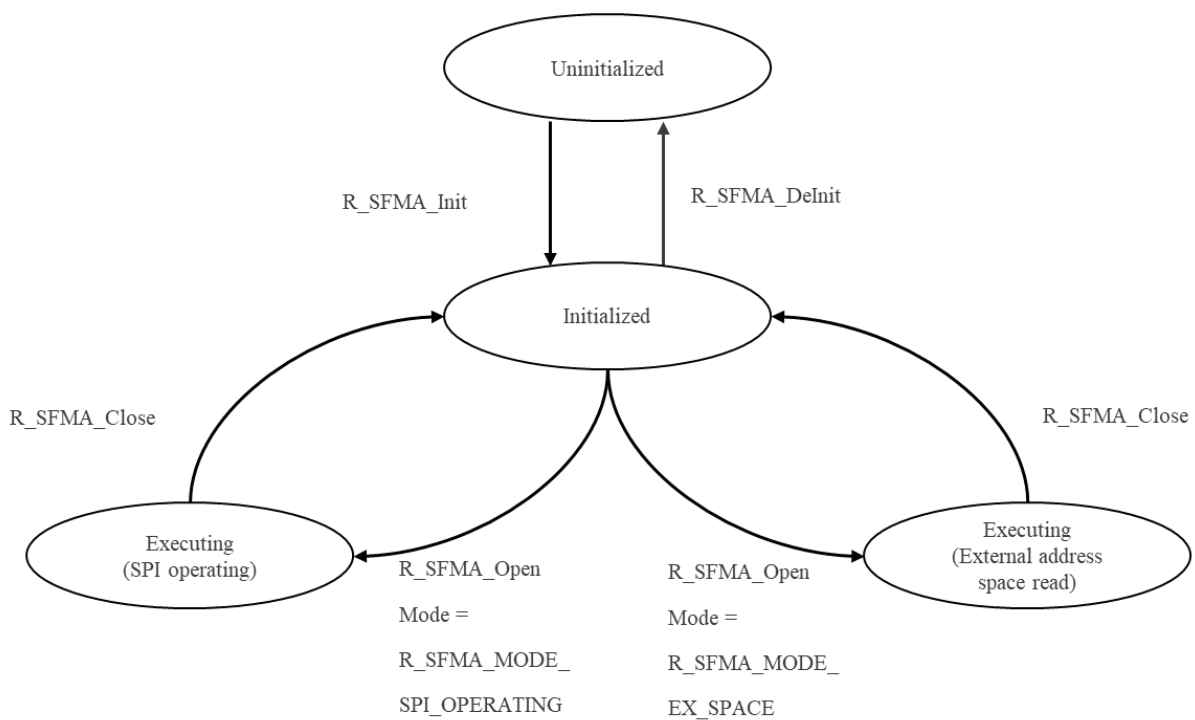


Figure 2-1 State Transition Diagram of SFMA driver

CONFIDENTIAL

Renesas Graphics Library Serial Flash Memory Interface A (SFMA) Driver

Table 2-4 State Transition Table of SFMA unit

Function Name	State			
	Uninitialized	Initialized	Executing	
			External address space read	SPI operating
R_SFMA_Init	OK	NG	NG	NG
R_SFMA_DeInit	NG	OK	NG	NG
R_SFMA_Open	NG	OK	NG	NG
R_SFMA_Close	NG	NG	OK	OK
R_SFMA_AccessAddressSet	NG	NG	OK	NG
R_SFMA_AccessAddressGet	NG	NG	OK	NG
R_SFMA_ProtectionModeSet	NG	NG	NG	OK
R_SFMA_DataErase	NG	NG	NG	OK
R_SFMA_DataWrite	NG	NG	NG	OK
R_SFMA_DataRead	NG	NG	NG	OK
R_SFMA_JEDECRead	NG	OK	NG	NG
R_SFMA_VersionStringGet	OK	OK	OK	OK
R_SFMA_MacroVersionGet	OK	OK	OK	OK
R_SFMA_GetCal	OK	OK	OK	OK
R_SFMA_GetStatus	NG	NG	NG	OK
R_SFMA_GetConfig	NG	NG	NG	OK
R_SFMA_WriteStatusConfig	NG	NG	NG	OK
R_SFMA_WriteReset	NG	NG	NG	OK
R_SFMA_UnitWindowSelect	NG	OK	NG	NG
R_SFMA_UnitNumberGet	OK	OK	OK	OK

3.Function Description

3.1 Fundamental Concepts

3.1.1 SFMA unit

RH850/D1x device has the following number of units of the SFMA.

Table 3-1 Number of units					
Feature	RH850/D1x Device Name				
	D1L2(H)	D1M1(H)	D1M1-V2	D1M1A	D1M2(H)
Number of Unit	1(Unit0)	1(Unit0)	2(Unit0, 1)	3(Unit0,1,2) *Unit1 and Unit2 are exclusive.	1(Unit0)

Almost SFMA API functions have the argument “Unit”.

User specifies the SFMA H/W unit number to be controlled. The range is 0 to 2.

3.1.2 System Configuration

This configuration has selected 1 bit data bus width. Then, SPBMO pin is the input pins and SPBMO pin is the output pins. SPBIO20 and SPBIO30 pins are not used.

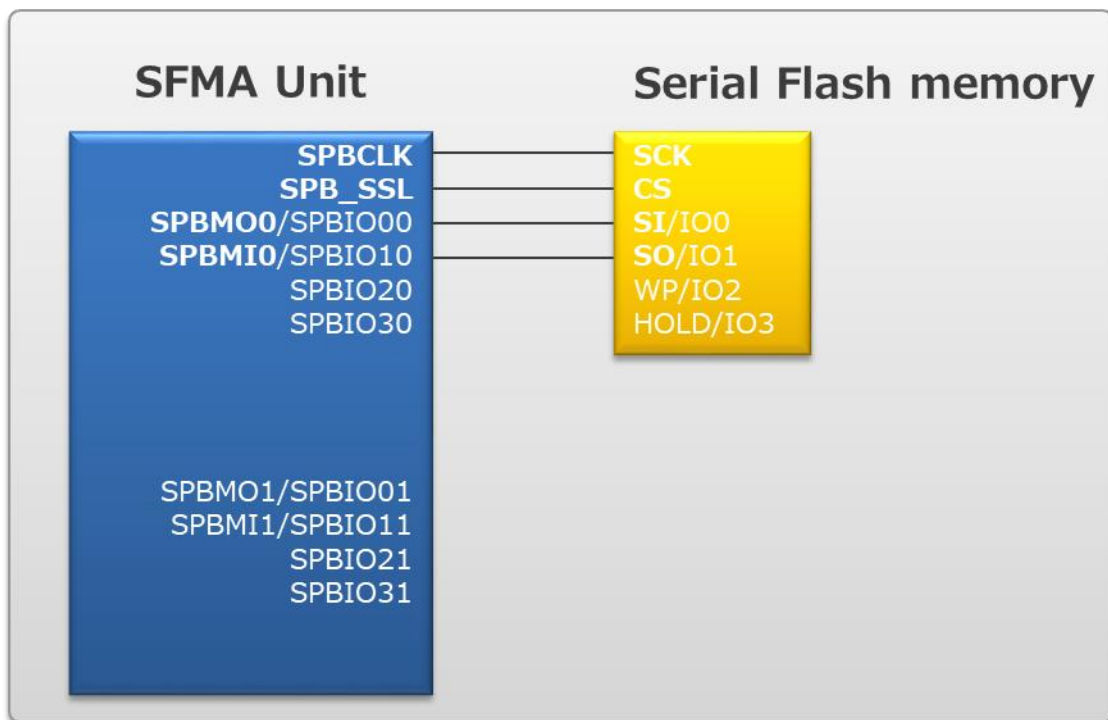


Figure 3-1 System Configuration for 1 bit data bus width

CONFIDENTIAL

Renesas Graphics Library Serial Flash Memory Interface A (SFMA) Driver

This configuration has selected 2 bits data bus width. Then, SPBIO00 and SPBIO10 pins are either the input pins or the output pins.

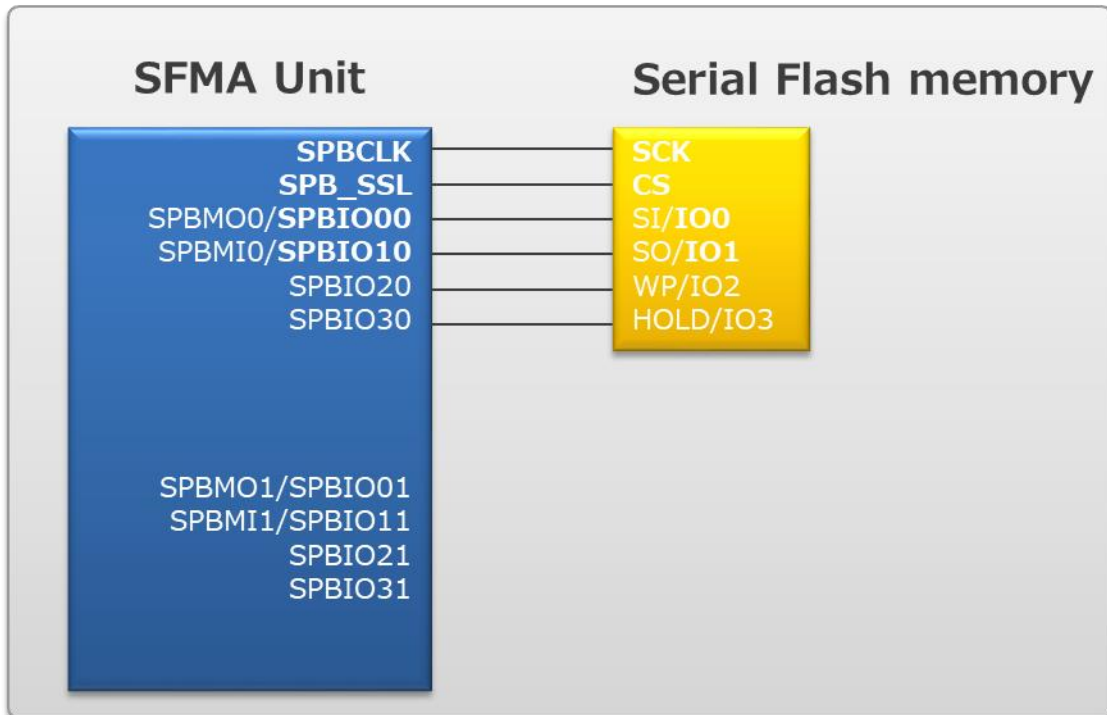


Figure 3-2 System Configuration for 2 bits data bus width

This configuration has selected 4 bits data bus width. Then, SPBIO00, SPBIO10, SPBIO20 and SPBIO30 pins are either the input pins or the output pins.

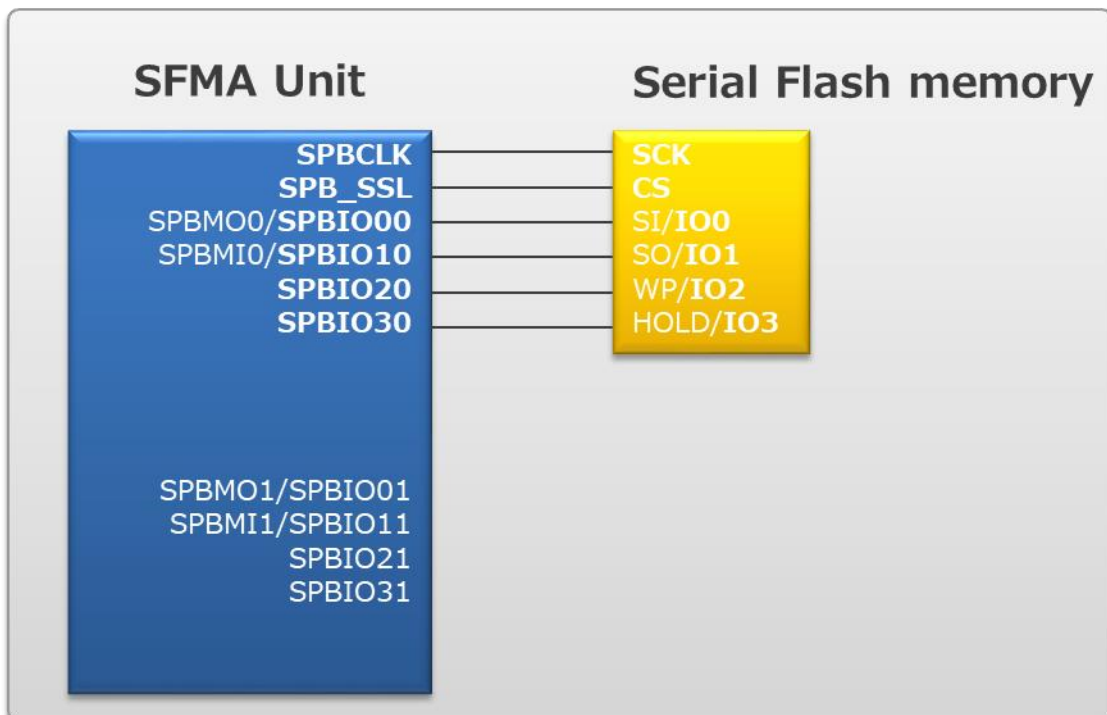


Figure 3-3 System Configuration for 4 bits data bus width

This configuration has selected 4 bits x 2 data bus width. Then, SPBIO00, SPBIO01, SPBIO10, SPBIO11, SPBIO20, SPBIO21, SPBIO30 and SPBIO31 pins are either the input pins or the output pins.

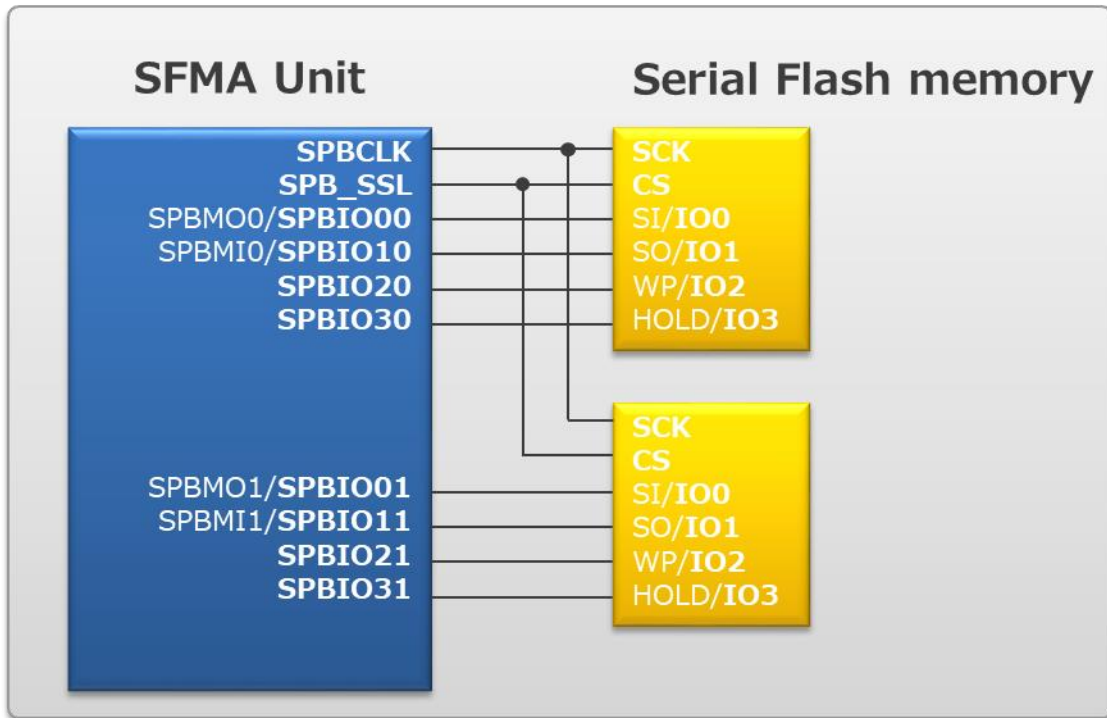


Figure 3-4 System Configuration for 4 bits x 2 data bus width

3.1.3 Operating Mode

3.1.3.1 External address space read

In external address space read mode, a read access to the SPI multi I/O bus space is converted into SPI communication and data is received. After data acquisition, data is returned to the bus master that is the issuing source. By using this mode, the serial flash memory is assigned to linear memory space and usability is improved.

The external address space read mode can access to the address space of a max 4 G bytes when one serial flash memory is connected or a max 8 G bytes when two serial flash memories are connected.

Efficient data reception due to built-in read cache (64-bit line × 16 entries)

3.1.3.2 SPI operating

In SPI operating mode, the SFMA driver can read, write, and erase data from/to the serial flash memory.

Note: The serial flash memory has protection function which prohibits writing and erasing. The control method of the protection function is different depending on serial flash memory.

3.1.4 Dependence command of the serial flash memory

The SFMA driver doesn't depend on the kind of the serial flash memory. To support various serial flash memories, the control commands, which depended on the serial flash memory, must be set. The command depending on the serial flash memory sets it to `r_sfma_FlashCommand_t` structure and it must be handed to the `Open` function.

According to the kind of the connected serial flash, there may be unsupported commands. Set `0xFF` to these commands.

The structure's members of `r_sfma_FlashCommand_t` consists of:

- SDR read command
- SDR dual read command
- SDR quad read command
- SDR dual Io read command
- SDR quad Io read command
- DDR read command
- DDR dual read command
- DDR quad read command
- DDR dual Io read command
- DDR quad Io read command
- SDR page program command
- SDR dual page program command
- SDR quad page program command
- DDR page program command
- DDR dual page program command
- DDR quad page program command
- Write enable command
- Erase sector command
- Read status register command
- Read configuration register command
- Write status register command
- Reset Enable command
- Reset command

Example

```
r_sfma_FlashCommand_t FlashCommandMX25L51245G =
{
    {
        R_SFMA_STATUS_REG, /* QuadIoMode.Reg */
        0x40 /* QuadIoMode.BitPosition */
    },
    {
        R_SFMA_STATUS_REG, /* BlockProtect.Reg */
        0x3C /* BlockProtect.BitPosition */
    },
    {
        R_SFMA_STATUS_REG, /* WriteInProgress.Reg */
        0x01 /* WriteInProgress.BitPosition */
    },
    {
        R_SFMA_CONFIG_REG, /* DummyCycle.Reg */
        0xC0, /* DummyCycle.BitMask */
        0x80 /* DummyCycle.BitSet */
    },
    R_SFMA_DUMMY_8CYC, /* ReadSdrDummyCycle */
    R_SFMA_DUMMY_8CYC, /* ReadSdrIoDualDummyCycle */
    R_SFMA_DUMMY_8CYC, /* ReadSdrIoQuadDummyCycle */
    R_SFMA_DUMMY_8CYC, /* ReadDdrDummyCycle */
    R_SFMA_DUMMY_8CYC, /* ReadDdrIoDualDummyCycle */
    R_SFMA_DUMMY_8CYC, /* ReadDdrIoQuadDummyCycle */
    R_SFMA_ADDRESS_SIZE_1BIT, /* WriteAddressBit */
    0x0C, /* ReadSdr */
    0x3C, /* ReadSdrDual */
    0x6C, /* ReadSdrQuad */
    0xBC, /* ReadSdrIoDual */
    0xEC, /* ReadSdrIoQuad */
    0x0E, /* ReadDdr */
    0xFF, /* ReadDdrDual */
    0xFF, /* ReadDdrQuad */
    0xBE, /* ReadDdrIoDual */
    0xEE, /* ReadDdrIoQuad */
    0x12, /* WriteSdr */
    0xFF, /* WriteSdrDual */
    0x3E, /* WriteSdrQuad */
    0xFF, /* WriteDdr */
    0xFF, /* WriteDdrDual */
    0xFF, /* WriteDdrQuad */
    0x06, /* WriteEnable */
    0x21, /* Erase */
    0x05, /* ReadStatus1 */
    0x15, /* ReadStatus2 */
    0x01, /* WriteStatus */
    0xFF, /* Exit external address space */
    0x66, /* ResetEnable */
    0x99, /* Reset */
    0xA5 /* Performance enhance mode indicator */
}
```

3.2 Using the API

3.2.1 Initialization / De-Initialization

R_SFMA_Init initializes the driver and the hardware as far as necessary. The Unit parameter holds a number that specifies the SFMA unit number being initialized. If it is necessary to acquire information of connected serial flash, the information on the serial flash can acquire information by calling R_SFMA_JDECRead function.

R_SFMA_DeInit function de-initializes the driver and the hardware as far as necessary.

Example

```
#define LOC_SFMA_UNIT (0)

/* Serial flash commands. */
r_sfma_FlashCommand_t loc_FlashCommand_A = {};
r_sfma_FlashCommand_t loc_FlashCommand_B = {};
r_sfma_FlashCommand_t* loc_FlashCommand;

void Initialize(void)
{
    uint8_t ManufactureID, MemoryType, Capacity;

    /* Initialize the SFMA driver. */
    R_SFMA_Init(LOC_SFMA_UNIT);

    /* Read JEDEC information. */
    R_SFMA_JEDECRead(LOC_SFMA_UNIT, &ManufactureID, &MemoryType, &Capacity);

    /* Select serial flash command. */
    if (ManufactureID == ID_A)
    {
        loc_FlashCommand = &loc_FlashCommand_A;
    }
    Else
    {
        loc_FlashCommand = &loc_FlashCommand_B;
    }
}
```

3.2.2 External address space mode

The setting method of External address space mode is shown below. Incidentally, set the value which was in the division at 32M to the access address. After setting completion, the serial flash connected is allocated for 0x10000000 to 0x11FFFFFF of SPI multi I/O bus space.

Example

```
#define LOC_ACCESS_ADDRESS (0x02000000) /* access address is from 32M */

void SetupExternalAddressReadMode(void)
{
    r_sfma_Config_t cfg;
    uint8_t data[100];
    uint32_t addr = 0;
    uint32_t size = 100;
    uint8_t* checkPointer;
    uint32_t i;

    /* Opens the SFMA driver at External address space read mode. */
    cfg.Mode = R_SFMA_MODE_EX_SPACE;
    cfg.MemoryNum = R_SFMA_MEMORY_SINGLE;
    cfg.DataTransferMode = R_SFMA_SDR_QUAD_IO;
    cfg.AddressMode = R_SFMA_ADDRESS_32BIT;
    cfg.SerialFlashPageSize = 256; /* byte */
    cfg.SerialFlashMemoryMaxClock = (80 * 1000 * 1000); /* Hz */
    cfg.SerialFlashMemorySectorSize = (4 * 1024) /* byte */
    cfg.SerialFlashMemorySize = LOC_SERIAL_FLASH_MEMORY_SIZE; /* byte */
    cfg.Command = loc_FlashCommand;
    cfg.CacheMode = R_SFMA_CACHE_OFF;
    cfg.Calibration = 0x00030001ul;
    cfg.PerformanceEnMode = R_SFMA_PER_EN_MODE_DISABLE;

    R_SFMA_Open(LOC_SFMA_UNIT, &cfg);

    /* Sets the address */
    R_SFMA_AccessAddressSet(LOC_SFMA_UNIT, addr, R_SFMA_ACCESS_RANGE_32MB);

    /* Read data */
    checkPointer = (uint8_t*)0x10000000; /* Serial flash Internal Address */
    for (i = 0; i < 100; i++)
    {
        data[i] = *checkPointer;
        checkPointer++;
    }
}
```

3.2.3 SPI operating mode

The setting method of SPI operating mode is shown below.

Example

```
void SetupSpiOperatingMode(void)
{
    r_sfma_Config_t cfg;
    uint8_t data[100];
    uint32_t addr = 0;
    uint32_t size = 100;

    /* Opens the SFMA driver at SPI operating mode. */
    cfg.Mode = R_SFMA_MODE_SPI_OPERATING;
    cfg.MemoryNum = R_SFMA_MEMORY_SINGLE;
    cfg.DataTransferMode = R_SFMA_SDR_QUAD_IO;
    cfg.AddressMode = R_SFMA_ADDRESS_32BIT;
    cfg.SerialFlashPageSize = 256; /* byte */
    cfg.SerialFlashMemoryMaxClock = (80 * 1000 * 1000); /* Hz */
    cfg.SerialFlashMemorySectorSize = (4 * 1024) /* byte */
    cfg.SerialFlashMemorySize = LOC_SERIAL_FLASH_MEMORY_SIZE; /* byte */
    cfg.Command = loc_FlashCommand;
    cfg.CacheMode = R_SFMA_CACHE_OFF;
    cfg.Calibration = 0x00030001ul;
    cfg.PerformanceEnMode = R_SFMA_PER_EN_MODE_DISABLE;

    R_SFMA_Open(LOC_SFMA_UNIT, &cfg);

    /* Sets protection mode */
    R_SFMA_ProtectionModeSet(LOC_SFMA_UNIT, R_SFMA_REQ_UNPROTECT);

    /* Reads the data */
    R_SFMA_DataRead(LOC_SFMA_UNIT, addr, data, size);

    /* Make write data */
    for (int i = 0; i < size; i++)
    {
        data[i] = i;
    }

    /* Erases the data */
    R_SFMA_DataErase(Unit, addr, size);

    /* Writes the data */
    R_SFMA_DataWrite(Unit, addr, data, size);
}
```

3.2.4 Access Addresses

In the external address space mode is possible to access MAX 8 Gbytes space of serial flash memory. But access able space which mapped to CPU address space is MAX 512 Mbytes. R_SFMA_AccessAddressSet function sets the access address and the access range.

When the R_SFMA_Open function is executed, the access address is automatically set to 0x00000000. And the access range is automatically set to 32 Mbytes.

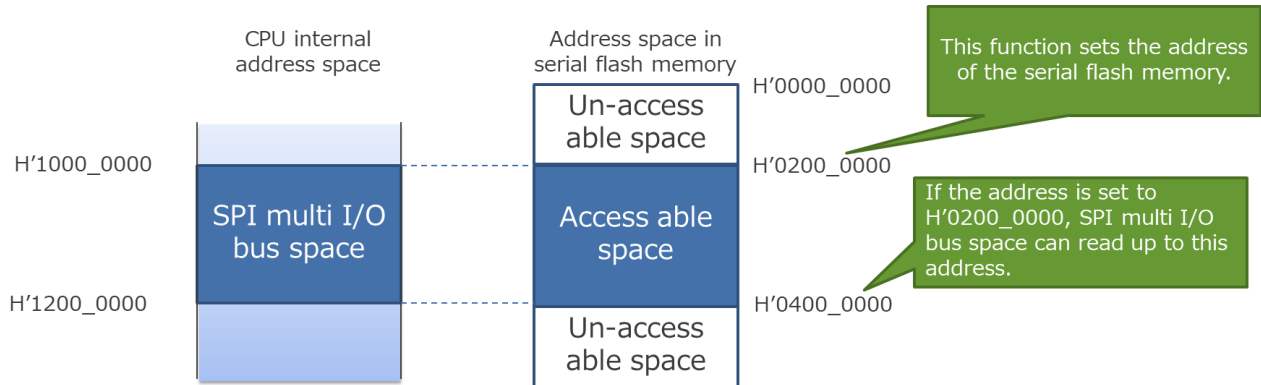


Figure 3-5 Access address map(example for Range = R_SFMA_ACCESS_RANGE_32MB)

3.2.5 Calibration

SFMA driver needs to set the phase value as argument of R_SFMA_Open to calibrate the phase between SPBCLK, sampling point, and input / output data. The phase value depends on the connected serial flash memory, board design and so on.

If you want to be executed the calibration, please calibrate with data read / data write after calling the R_SFMA_Open function by setting the phase value to Calibration member of r_sfma_Config_t structure.

Refer to RH850/D1L/D1M Group User's Manual: Hardware for details about the phase value.

3.2.6 Data Erase

R_SFMA_DataErase erases the data of designated address in the serial flash memory.

Erasing of data is performed on the sector unit. Therefore, this function erases data of the sector including the size from the address.

Erase sector size become double size when MemoryNum of the r_sfma_Config_t structure is R_SFMA_MEMORY_DUAL.

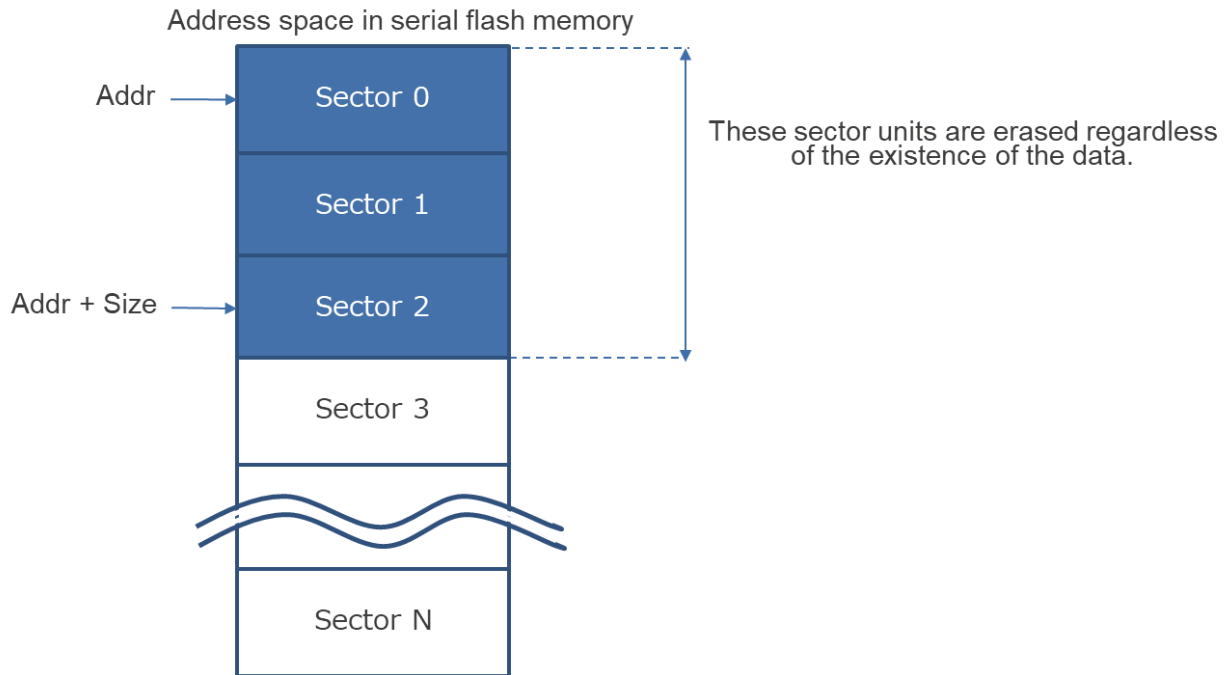


Figure 3-6 Erasing of data example

CONFIDENTIAL

3.3 Device difference

The following table shows the function differences depending on the device.

Table 3-2 Difference functions

Feature			RH850/D1x Device Name					
			D1L2	D1L2H	D1M1(H)	D1M1-V2	D1M1A	D1M2(H)
Number of Unit			1 Unit0)	1 (Unit0)	1 (Unit0)	2 (Unit0, 1)	3 ¹ (Unit0, 1, 2)	1 (Unit0)
Number of serial flash memory to connect			1	2	2	2	2 (Unit0, 1) 1 (Unit2) ²	2
Maximum Clock	Unit0	SDR	120MHz	120MHz	120MHz	120MHz	120MHz	120MHz
		DDR	80MHz	80MHz	80MHz	80MHz	80MHz	80MHz
	Unit1	SDR				40MHz	40MHz	
		DDR				40MHz	40MHz	
	Unit2	SDR					80MHz	
		DDR					80MHz	

¹ Unit1 and Unit2 are exclusive.

² If Unit2 is used, only 1 serial flash memory can be connected to Unit0.

CONFIDENTIAL

Renesas Graphics Library Serial Flash Memory Interface A (SFMA) Driver

The following table shows the function differences depending on the device.
“NG” function will fail and return error code.

Table 3-3 Differences APIs

Function Name	RH850/D1x Device Name			
	D1L2(H)	D1M1(H)	D1M1A, D1M1-V2	D1M2(H)
R_SFMA_Init	OK	OK	OK	OK
R_SFMA_DeInit	OK	OK	OK	OK
R_SFMA_Open	OK	OK	OK	OK
R_SFMA_Close	OK	OK	OK	OK
R_SFMA_AccessAddressSet	OK	OK	OK	OK
R_SFMA_AccessAddressGet	OK	OK	OK	OK
R_SFMA_ProtectionModeSet	OK	OK	OK	OK
R_SFMA_DataErase	OK	OK	OK	OK
R_SFMA_DataWrite	OK	OK	OK	OK
R_SFMA_DataRead	OK	OK	OK	OK
R_SFMA_JEDECRead	OK	OK	OK	OK
R_SFMA_VersionStringGet	OK	OK	OK	OK
R_SFMA_MacroVersionGet	OK	OK	OK	OK
R_SFMA_GetCal	OK	OK	OK	OK
R_SFMA_GetStatus	OK	OK	OK	OK
R_SFMA_GetConfig	OK	OK	OK	OK
R_SFMA_WriteStatusConfig	OK	OK	OK	OK
R_SFMA_WriteReset	OK	OK	OK	OK
R_SFMA_UnitWindowSelect	NG	NG	NG	NG
R_SFMA_UnitNumberGet	NG	NG	NG	NG

3.4 Header File List

Table 3-4 Header File List

No.	Header File Name	Description
(1)	r_sfma_api.h	Header file for SFMA API.
(2)	r_typedefs.h	Header file for predefined data types.

4.Functions

4.1 Function List

This section describes about the SFMA API functions which are in [Table 4-1](#) and executable state of each function is described in the specification of each function.

Table 4-1 List of SFMA API Functions

Function Name	Purpose
<i>R_SFMA_Init</i>	This function initializes the SFMA driver.
<i>R_SFMA_DeInit</i>	This function de-initializes the SFMA driver.
<i>R_SFMA_Open</i>	This function opens the unit by specified configuration mode.
<i>R_SFMA_Close</i>	This function closes the unit.
<i>R_SFMA_AccessAddressSet</i>	This function sets the address of the serial flash memory.
<i>R_SFMA_AccessAddressGet</i>	This function gets the address of the serial flash memory.
<i>R_SFMA_ProtectionModeSet</i>	This function sets protection mode of the serial flash memory.
<i>R_SFMA_DataErase</i>	This function erases the data of designated address in the serial flash memory.
<i>R_SFMA_DataWrite</i>	This function writes data to the serial flash memory.
<i>R_SFMA_DataRead</i>	This function reads data from the serial flash memory.
<i>R_SFMA_JEDECRead</i>	This function reads basic JEDEC data.
<i>R_SFMA_VersionStringGet</i>	This function returns the version string of this SFMA driver.
<i>R_SFMA_MacroVersionGet</i>	This function returns the major and minor version of the H/W macro.
<i>R_SFMA_GetCal</i>	This function retrieves the calibration value.
<i>R_SFMA_GetStatus</i>	This function retrieves the status for the device in SPI operating mode.
<i>R_SFMA_GetConfig</i>	This function retrieves the configuration for the device in SPI operating mode.
<i>R_SFMA_WriteStatusConfig</i>	This function writes the Status and configuration for the device in SPI operating mode.
<i>R_SFMA_WriteReset</i>	This function writes a software reset for the device in SPI operating mode.
<i>R_SFMA_UnitWindowSelect</i>	This function is reserved for future use.
<i>R_SFMA_UnitNumberGet</i>	This function is reserved for future use.

4.2 SFMA API Functions

This chapter describes the application interface functions, which are required for general use of the driver.

4.2.1 Basic functions

The section describes driver functions, which are required for general use of the driver, but which are related to a specific functionality of the macro itself.

4.2.1.1 R_SFMA_Init

Function Prototypes

```
r_sfma_Error_t R_SFMA_Init(const unit32_t    Unit)
```

Input Parameter

Table 4-2 Input parameter of R_SFMA_Init

Parameter	Description
Unit	Specifies the SFMA unit number.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.

Description

This function initializes the driver and the hardware as far as necessary.
SFMA unit status will become Initialized state after the execution of this function.

Reentrancy

Non-reentrant

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

r_sfma_Error_t

CONFIDENTIAL

4.2.1.2 R_SFMA_DeInit

Function Prototypes

```
r_sfma_Error_t R_SFMA_DeInit(const uint32_t Unit)
```

Input Parameter

Table 4-3 Input parameter of R_SFMA_DeInit

Parameter	Description
Unit	Specifies the SFMA unit number.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_FATAL_OS	- Fatal error has occurred at OS interface.

Description

This function de-initializes the driver and the hardware.
SFMA unit status will become Uninitialized state after executing this function.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

r_sfma_Error_t

4.2.1.3 R_SFMA_Open**Function Prototypes**

```
r_sfma_Error_t R_SFMA_Open( const uint32_t          Unit,  
                             const r_sfma_Config_t * const Config)
```

Input Parameter**Table 4-4 Input parameter of R_SFMA_Open**

Parameter	Description
Unit	Specifies the SFMA unit number.
Config	This is a pointer to the r_sfma_Config_t structure to the configuration of the unit.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_SFMA_ERR_OK	- No error has occurred.
R_SFMA_ERR_PARAM_INCORRECT	- A parameter provided to a function is incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_RANGE_UNIT	- The unit-number is the outside of the range.
R_SFMA_ERR_FATAL_OS	- Fatal Error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.
R_SFMA_ERR_NOT_SUPPORT_CLOCK	- The set value of the serial clock isn't supported.
R_SFMA_ERR_NOT_SUPPORT_TRANSFER	- The set transfer mode isn't supported in the connected serial flash.

Description

This function opens the SFMA driver at an operating mode, number of memories that is specified.

If the function successfully executes, the return code will be R_SFMA_ERR_OK. And the status will be changed to Executing (External address space read) if R_SFMA_MODE_EX_SPACE is specified. The status will be changed to Executing (SPI operating) if R_SFMA_MODE_SPI_OPERATING is specified.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

r_sfma_Error_t
r_sfma_Config_t

4.2.1.4 R_SFMA_Close**Function Prototypes**

```
r_sfma_Error_t R_SFMA_Close( const uint32_t Unit)
```

Input Parameter**Table 4-5 Input parameter of R_SFMA_Close**

Parameter	Description
Unit	Specifies the SFMA unit number.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.

Description

This function closes the SFMA driver

If the function successfully executes, the return code will be R_SFMA_ERR_OK and the state will be in the Initialize state.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

r_sfma_Error_t

4.2.1.5 R_SFMA_AccessAddressSet**Function Prototypes**

```

r_sfma_Error_t R_SFMA_AccessAddressSet( const uint32_t      Unit,
                                         const uint64_t      Addr,
                                         const r_sfma_AccessRange_t Range)

```

Input Parameter**Table 4-6 Input parameter of R_SFMA_AccessAddressSet**

Parameter	Description
Unit	Specifies the SFMA unit number.
Addr	The parameter specifies the access address of the serial flash memory. This parameter aligns in the access range of "Range" of the argument. e.g. if "Addr" is "0x2000000" and "Range" is 64MB, the setting address will be "0x0".
Range	The parameter specifies the access range of the serial flash memory.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_NG	- An error has occurred, but no specific error code is defined for it.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_RANGE_UNIT	- The unit-number is the outside of the range.
R_SFMA_ERR_RANGE_PARAM	- A parameter is the outside of the range.
R_SFMA_ERR_FATAL_OS	- Fatal Error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.

Description

This function sets the address of the serial flash memory.

If the function successfully executes, the return code will be R_SFMA_ERR_OK.

When the R_SFMA_Open function is executed, the access address and the access range is automatically set to 0x00000000 and 32 Mbytes.

See [3.2.4](#) about Access addresses.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

`r_sfma_Error_t`

`r_sfma_AccessRange_t`

4.2.1.6 R_SFMA_AccessAddressGet**Function Prototypes**

```

r_sfma_Error_t R_SFMA_AccessAddressGet( const uint32_t          Unit,
                                         uint64_t             * const Addr,
                                         r_sfma_AccessRange_t * const Range)

```

Input Parameter**Table 4-7 Input parameter of R_SFMA_AccessAddressGet**

Parameter	Description
Unit	Specifies the SFMA unit number.

Input-Output Parameter

None

Output Parameter**Table 4-8 Output parameter of R_SFMA_AccessAddressGet**

Parameter	Description
Addr	This is a pointer to the access address of the serial flash memory.
Range	This is a pointer to the access range of the serial flash memory.

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_PARAM_INCORRECT	- A parameter provided to a function is incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_FATAL_OS	- Fatal error has occurred at OS interface.

Description

This function gets the address of the serial flash memory.
 If the function successfully executes, the return code will be R_SFMA_ERR_OK.

Reentrancy

Non-reentrant as default.
 If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

r_sfma_Error_t

r_sfma_AccessRange_t

4.2.1.7 R_SFMA_ProtectionModeSet**Function Prototypes**

```
r_sfma_Error_t R_SFMA_ProtectionModeSet(const uint32_t Unit,  
                                         const r_sfma_ProtectionMode_t Mode)
```

Input Parameter**Table 4-9 Input parameter of R_SFMA_ProtectionModeSet**

Parameter	Description
Unit	Specifies the SFMA unit number.
Mode	The parameter specifies the protection mode of the serial flash memory.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_PARAM_INCORRECT	- Parameter was incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.

Description

This function sets protection mode of the serial flash memory.
The write and erase access to serial flash memory is prohibited if protection is set.

Reentrancy

Non-reentrant as default.
If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

`r_sfma_Error_t`
`r_sfma_ProtectionMode_t`

4.2.1.8 R_SFMA_DataErase**Function Prototypes**

```
r_sfma_Error_t R_SFMA_DataErase( const uint32_t Unit,
                                   const uint64_t Addr,
                                   const int32_t  Size )
```

Input Parameter**Table 4-10 Input parameter of R_SFMA_DataErase**

Parameter	Description
Unit	Specifies the SFMA unit number.
Addr	The parameter specifies the erase address of the serial flash memory.
Size	The parameter specifies the data size (in bytes) to erase.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_PARAM_INCORRECT	- A parameter provided to a function is incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_RANGE_PARAM	- A parameter is the outside of the range.
R_SFMA_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.
R_SFMA_ERR_SFLASH_PROTECTED	- The serial flash memory is protected.

Description

This function erases the data of designated address in the serial flash memory.

This function has the possibility that the processing takes time. Therefore, R_SFMA_Sys_Relax is sometimes executed.

Refer to Porting Layer specification for details.

See [3.2.6](#) about Data Erase.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

`r_sfma_Error_t`

4.2.1.9 R_SFMA_DataWrite**Function Prototypes**

```

r_sfma_Error_t R_SFMA_DataWrite(const uint32_t Unit,
                                const uint64_t Addr,
                                const uint8_t *Buf,
                                const int32_t Size)

```

Input Parameter**Table 4-11 Input parameter of R_SFMA_DataWrite**

Parameter	Description
Unit	Specifies the SFMA unit number.
Addr	The parameter specifies the write address of the serial flash memory. This parameter aligns in the page size of the serial flash memory.
Buf	This is a pointer to the address of buffer.
Size	The parameter specifies the data size (in bytes) to write.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_PARAM_INCORRECT	- A parameter provided to a function is incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_RANGE_PARAM	- A parameter is the outside of the range.
R_SFMA_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.
R_SFMA_ERR_NOT_SUPPORT_TRANSFER	- The set transfer mode isn't supported in the connected serial flash.
R_SFMA_ERR_SFLASH_PROTECTED	- The serial flash memory is protected.

Description

This function writes data to the serial flash memory.

In order to write data, the data of the sector must have been previously erased.

This function has the possibility that the processing takes time. Therefore, R_SFMA_Sys_Relax is sometimes executed.

This function always executes by the SDR transfer mode.

If the function successfully executes, the return code will be R_SFMA_ERR_OK.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

r_sfma_Error_t

4.2.1.10 R_SFMA_DataRead

Function Prototypes

```
r_sfma_Error_t R_SFMA_DataRead(const uint32_t Unit,
                                const uint64_t Addr,
                                uint8_t *Buf,
                                const int32_t Size)
```

Input Parameter

Table 4-12 Input parameter of R_SFMA_DataRead

Parameter	Description
Unit	Specifies the SFMA unit number.
Addr	The parameter specifies the read address of the serial flash memory.
Size	The parameter specifies the data size (in bytes) to read.

Input-Output Parameter

None

Output Parameter

Table 4-13 Output parameter of R_SFMA_DataRead

Parameter	Description
Buf	This is a pointer to the address of buffer.

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_NG	- An error has occurred, but no specific error code is defined for it.
R_SFMA_ERR_PARAM_INCORRECT	- A parameter provided to a function is incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_RANGE_UNIT	- The unit-number is the outside of the range.
R_SFMA_ERR_RANGE_PARAM	- A parameter is the outside of the range.
R_SFMA_ERR_FATAL_OS	- Fatal Error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.
R_SFMA_ERR_NOT_SUPPORT_TRANSFER	- The set transfer mode isn't supported in the connected serial flash.

Description

This function reads data from the serial flash memory.

This function has the possibility that the processing takes time. Therefore, R_SFMA_Sys_Relax is sometimes executed. If the function successfully executes, the return code will be R_SFMA_ERR_OK.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

r_sfma_Error_t

4.2.1.11 R_SFMA_JEDECRead

Function Prototypes

```
r_sfma_Error_t R_SFMA_JEDECRead(const uint32_t      Unit,
                                uint8_t * const ManufacturerID,
                                uint8_t * const MemoryType,
                                uint8_t * const Capacity)
```

Input Parameter

Table 4-14 Input parameter of R_SFMA_JEDECRead

Parameter	Description
Unit	Specifies the SFMA unit number.

Input-Output Parameter

None

Output Parameter

Table 4-15 Output parameter of R_SFMA_JEDECRead

Parameter	Description
ManufacturerID	This is a pointer to the manufacturer ID.
MemoryType	This is a pointer to the memory type.
Capacity	This is a pointer to the memory capacity.

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_PARAM_INCORRECT	- A parameter provided to a function is incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_RANGE_UNIT	- The unit-number is the outside of the range.
R_SFMA_ERR_RANGE_PARAM	- A parameter is the outside of the range.
R_SFMA_ERR_FATAL_OS	- Fatal Error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.

Description

This function reads basic JEDEC data.

The function operates at the maximum clock frequency depending on the device and unit. See [Table 3-2](#) about maximum clock.

If the function successfully executes, the return code will be R_SFMA_ERR_OK.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

r_sfma_Error_t

4.2.1.12 R_SFMA_VersionStringGet

Function Prototypes

```
const uint8_t* R_SFMA_VersionStringGet(void)
```

Input Parameter

None

Input-Output Parameter

None

Output Parameter

None

Return Codes

Version string.

Description

This function returns version string of the SFMA driver.

Reentrancy

Reentrant.

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

None

4.2.1.13 R_SFMA_MacroVersionGet**Function Prototypes**

```
r_sfma_Error_t R_SFMA_MacroVersionGet(uint32_t * const Major,  
                                       uint32_t * const Minor)
```

Input Parameter

None

Input -Output Parameter

None

Output Parameter**Table 4-16 Output parameter of R_SFMA_MacroVersionGet**

Parameter	Description
Major	The major version.
Minor	The minor version.

Return Codes

R_SFMA_ERR_OK	- No error has occurred.
R_SFMA_ERR_PARAM_INCORRECT	- Either parameter Major or parameter Minor was R_NULL

Description

This function returns the major and minor version of the H/W macro.

Reentrancy

Reentrant.

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

r_sfma_Error_t

4.2.1.14 R_SFMA_GetCal**Function Prototypes**

```
r_sfma_Error_t R_SFMA_GetCal(const uint32_t Unit,  
                             uint32_t * Cal)
```

Input Parameter**Table 4-17 Input parameter of R_SFMA_GetCal**

Parameter	Description
Unit	Specifies the SFMA unit number.

Input-Output Parameter

None

Output Parameter**Table 4-18 Output parameter of R_SFMA_GetCal**

Parameter	Description
Cal	This is a pointer to store the calibration value.

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_PARAM_INCORRECT	- Parameter was incorrect.

Description

This function retrieves the calibration value.

If the function successfully executes, the return code will be R_SFMA_ERR_OK.

Reentrancy

Reentrant

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

r_sfma_Error_t

4.2.1.15 R_SFMA_GetStatus**Function Prototypes**

```

r_sfma_Error_t R_SFMA_GetStatus(const uint32_t      Unit,
                                uint8_t * const Status1,
                                uint8_t * const Status2)

```

Input Parameter**Table 4-19 Input parameter of R_SFMA_GetStatus**

Parameter	Description
Unit	Specifies the SFMA unit number.

Input-Output Parameter

None

Output Parameter**Table 4-20 Output parameter of R_SFMA_GetStatus**

Parameter	Description
Status1	This is a pointer to store the Status1 value.
Status2	This is a pointer to store the Status2 value. When two serial flash memories are connected, Status1 stores the parameter of the first serial flash memory and Status2 stores the parameter of the second serial flash memory. When two serial flash memories are not connected, Status2 returns 0.

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_PARAM_INCORRECT	- Parameter was incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.

Description

This function retrieves the Status register value for the serial flash memory in SPI operating mode.
If the function successfully executes, the return code will be R_SFMA_ERR_OK.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

`r_sfma_Error_t`

4.2.1.16 R_SFMA_GetConfig

Function Prototypes

```
r_sfma_Error_t R_SFMA_GetConfig(const uint32_t      Unit,
                                uint8_t * const Config1,
                                uint8_t * const Config2)
```

Input Parameter

Table 4-21 Input parameter of R_SFMA_GetConfig

Parameter	Description
Unit	Specifies the SFMA unit number.

Input-Output Parameter

None

Output Parameter

Table 4-22 Output parameter of R_SFMA_GetConfig

Parameter	Description
Config1	This is a pointer to store the Config1 value.
Config2	This is a pointer to store the Config2 value. When two serial flash memories are connected, Config1 stores the parameter of the first serial flash memory and Config2 stores the parameter of the second serial flash memory. When two serial flash memories are not connected, Config2 returns 0.

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_PARAM_INCORRECT	- Parameter was incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.

Description

This function retrieves the Configuration register value for the serial flash memory in SPI operating mode.
If the function successfully executes, the return code will be R_SFMA_ERR_OK.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

`r_sfma_Error_t`

4.2.1.17 R_SFMA_WriteStatusConfig**Function Prototypes**

```
r_sfma_Error_t R_SFMA_WriteStatusConfig(const uint32_t Unit,  
                                         uint8_t   Status,  
                                         uint8_t   Config)
```

Input Parameter**Table 4-23 Input parameter of R_SFMA_WriteStatusConfig**

Parameter	Description
Unit	Specifies the SFMA unit number.
Status	This is the value to write to the Status register.
Config	This is the value to write to the Configuration register.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_PARAM_INCORRECT	- Parameter was incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.

Description

This function writes the Status register value and the Configuration register value to the serial flash memory in SPI operating mode.

If the function successfully executes, the return code will be R_SFMA_ERR_OK.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

`r_sfma_Error_t`

4.2.1.18 R_SFMA_WriteReset**Function Prototypes**

```
r_sfma_Error_t R_SFMA_WriteReset(const uint32_t Unit)
```

Input Parameter**Table 4-24 Input parameter of R_SFMA_WriteReset**

Parameter	Description
Unit	Specifies the SFMA unit number.

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_PARAM_INCORRECT	- Parameter was incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	- Fatal error has occurred at H/W.

Description

This function writes a software reset for the serial flash memory in SPI operating mode.
If the function successfully executes, the return code will be R_SFMA_ERR_OK.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

`r_sfma_Error_t`

4.2.1.19 R_SFMA_UnitWindowSelect**Function Prototypes**

```
r_sfma_Error_t R_SFMA_UnitWindowSelect(const uint32_t Unit,  
                                       const r_sfma_WindowMode_t Window)
```

Input Parameter**Table 4-25 Input parameter of R_SFMA_UnitWindowSelect**

Parameter	Description
Unit	Specifies the SFMA unit number.
Window	The parameter specifies the window for SFMA

Input-Output Parameter

None

Output Parameter

None

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_RANGE_UNIT	- The unit-number was outside the range.
R_SFMA_ERR_PARAM_INCORRECT	- Parameter was incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_FATAL_OS	- Fatal error has occurred at OS interface.
R_SFMA_ERR_NOT_SUPPORTED	- The device isn't supported.

Description

This function is reserved for future use.

This function selects the Window for SFMA0 and SFMA1.

The CPU internal address space have the Primary Data Read Window and Programming Window for SFMA. This function assigns the Window by specifying the unit number and the Window. When one of the units assigns a Window, the other unit assigns another Window.

If the function successfully executes, the return code will be R_SFMA_ERR_OK.

Reentrancy

Non-reentrant as default.

If user implements following functions to prevent multiple executions, this function will become re-entrant.

- R_SFMA_Sys_Lock
- R_SFMA_Sys_Unlock

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

r_sfma_Error_t
r_sfma_WindowMode_t

4.2.1.20 R_SFMA_UnitNumberGet**Function Prototypes**

```
r_sfma_Error_t R_SFMA_UnitNumberGet(const r_sfma_WindowMode_t Window,  
                                     uint32_t * const Unit);
```

Input Parameter**Table 4-26 Input parameter of R_SFMA_UnitNumberGet**

Parameter	Description
Window	The parameter specifies the window for SFMA

Input-Output Parameter

None

Output Parameter**Table 4-27 Output parameter of R_SFMA_UnitNumberGet**

Parameter	Description
Unit	Specifies the SFMA unit number.

Return Codes

R_SFMA_ERR_OK	- No error occurred.
R_SFMA_ERR_PARAM_INCORRECT	- Parameter was incorrect.
R_SFMA_ERR_NOT_ACCEPTABLE	- A function was called in an incorrect state.
R_SFMA_ERR_NOT_SUPPORTED	- The device isn't supported.

Description

This function is reserved for future use.

If the function successfully executes, the return code will be R_SFMA_ERR_OK.

Reentrancy

Reentrant

Sync/Async

Synchronous

Call from Interrupt

Prohibited.

Preconditions

See [Table 2-4](#) about SFMA unit status conditions.

See also

`r_sfma_Error_t`
`r_sfma_WindowMode_t`

4.2.2 Interrupt functions

None.

5.Types

5.1 Basic Types

This section shows the basic types used on this library.

Table 5-1 Basic type

Types	Definition	Basic types
char_t	typedef char char_t	signed char
int8_t	typedef signed char int8_t	signed char
int16_t	typedef signed short int16_t	signed short
int32_t	typedef signed int int32_t	signed int
int64_t	typedef signed long long int64_t	signed long long
uint8_t	typedef unsigned char uint8_t	unsigned char
uint16_t	typedef unsigned short uint16_t	unsigned short
uint32_t	typedef unsigned int uint32_t	unsigned int
uint64_t	typedef unsigned long long uint64_t	unsigned long long
float32_t	typedef float float32_t	float
float64_t	typedef double float64_t	double

5.2 Definition

This section shows the definitions used in SFMA API.

Table 5-2 Definition of SFMA API

Name	Description
R_SFMA_VERSION_HI	MSB byte of the version information. It is major version information. This value is changed with release version.
R_SFMA_VERSION_LO	LSB byte of the version information. It is minor version information. This value is changed with release version.

Table 5-3 Definition of the performance enhance mode option

Name	Description
R_SFMA_PER_EN_MODE_DISABLE	SFMA option for performance enhance mode. The option is disabled.
R_SFMA_PER_EN_MODE_ENABLE	SFMA option for performance enhance mode. The option is enabled.

5.3 Enumerated Type

This section shows the enumerated types used in SFMA API Function.

5.3.1 r_sfma_Error_t

Description

SFMA driver error code.

If an error occurs, these enumerations give information about the reason.

Definition

```
typedef enum
{
    R_SFMA_ERR_OK = 0,
    R_SFMA_ERR_NG,
    R_SFMA_ERR_PARAM_INCORRECT,
    R_SFMA_ERR_RANGE_UNIT,
    R_SFMA_ERR_RANGE_PARAM,
    R_SFMA_ERR_NOT_ACCEPTABLE,
    R_SFMA_ERR_FATAL_OS,
    R_SFMA_ERR_FATAL_HW,
    R_SFMA_ERR_NOT_SUPPORT_CLOCK,
    R_SFMA_ERR_NOT_SUPPORT_TRANSFER,
    R_SFMA_ERR_SFLASH_PROTECTED,
    R_SFMA_ERR_NOT_SUPPORTED
} r_sfma_Error_t;
```

Table 5-4 Enumerator of r_sfma_Error_t

Name	Description
R_SFMA_ERR_OK	No error occurred.
R_SFMA_ERR_NG	An error has occurred, but no specific error code is defined for it.
R_SFMA_ERR_PARAM_INCORRECT	A parameter provided to a function was incorrect.
R_SFMA_ERR_RANGE_UNIT	The unit-number was outside the range.
R_SFMA_ERR_RANGE_PARAM	Parameter is the outside the range.
R_SFMA_ERR_NOT_ACCEPTABLE	A function was called in an incorrect state.
R_SFMA_ERR_FATAL_OS	Fatal error has occurred at OS interface.
R_SFMA_ERR_FATAL_HW	Fatal error has occurred at H/W.
R_SFMA_ERR_NOT_SUPPORT_CLOCK	The set value of the serial clock isn't supported.
R_SFMA_ERR_NOT_SUPPORT_TRANSFER	The set transfer mode isn't supported in the connected serial flash.
R_SFMA_ERR_SFLASH_PROTECTED	The serial flash memory is protected.
R_SFMA_ERR_NOT_SUPPORTED	The device isn't supported.

See also

None

5.3.2 r_sfma_MemoryNum_t**Description**

This type describes the number of serial flash memory to be connected.

Definition

```
typedef enum
{
    R_SFMA_MEMORY_SINGLE = 0,
    R_SFMA_MEMORY_DUAL
} r_sfma_MemoryNum_t;
```

Table 5-5 Enumerator of r_sfma_MemoryNum_t

Name	Description
R_SFMA_MEMORY_SINGLE	One serial flash memory is connected.
R_SFMA_MEMORY_DUAL	Two serial flash memories are connected.

See also

None

5.3.3 r_sfma_Mode_t**Description**

This type describes the operating mode.

Definition

```
typedef enum
{
    R_SFMA_MODE_EX_SPACE = 0,
    R_SFMA_MODE_SPI_OPERATING
} r_sfma_Mode_t;
```

Table 5-6 Enumerator of r_sfma_Mode_t

Name	Description
R_SFMA_MODE_EX_SPACE	External address space read mode.
R_SFMA_MODE_SPI_OPERATING	SPI operating mode.

See also

None

5.3.4 r_sfma_DataTransferMode_t**Description**

This type describes the data transfer mode.

Definition

```
typedef enum
{
    R_SFMA_SDR_SINGLE_IO = 0,
    R_SFMA_SDR_DUAL_IO,
    R_SFMA_SDR_QUAD_IO,
    R_SFMA_DDR_SINGLE_IO,
    R_SFMA_DDR_DUAL_IO,
    R_SFMA_DDR_QUAD_IO
} r_sfma_DataTransferMode_t;
```

Table 5-7 Enumerator of r_sfma_DataTransferMode_t

Name	Description
R_SFMA_SDR_SINGLE_IO	SDR with Single I/O.
R_SFMA_SDR_DUAL_IO	SDR with Dual I/O.
R_SFMA_SDR_QUAD_IO	SDR with Quad I/O.
R_SFMA_DDR_SINGLE_IO	DDR with Single I/O.
R_SFMA_DDR_DUAL_IO	DDR with Dual I/O.
R_SFMA_DDR_QUAD_IO	DDR with Quad I/O.

See also

None

5.3.5 r_sfma_ProtectionMode_t**Description**

This type describes the protection mode of the serial flash memory.

Definition

```
typedef enum
{
    R_SFMA_MODE_PROTECT = 0,
    R_SFMA_MODE_UNPROTECT
} r_sfma_ProtectionMode_t;
```

Table 5-8 Enumerator of r_sfma_ProtectionMode_t

Name	Description
R_SFMA_MODE_PROTECT	Protection mode.
R_SFMA_MODE_UNPROTECT	Un-protection mode.

See also

None

5.3.6 r_sfma_AddressMode_t**Description**

This type describes the format of the address output to the serial flash memory.

Definition

```
typedef enum
{
    R_SFMA_ADDRESS_24BIT = 0,
    R_SFMA_ADDRESS_32BIT
} r_sfma_AddressMode_t;
```

Table 5-9 Enumerator of r_sfma_AddressMode_t

Name	Description
R_SFMA_ADDRESS_24BIT	24 bit address output.
R_SFMA_ADDRESS_32BIT	32 bit address output.

See also

None

5.3.7 r_sfma_AccessRange_t**Description**

This type describes the access range of the serial flash memory.

Definition

```
typedef enum
{
    R_SFMA_ACCESS_RANGE_32MB = 0,
    R_SFMA_ACCESS_RANGE_64MB,
    R_SFMA_ACCESS_RANGE_128MB,
    R_SFMA_ACCESS_RANGE_256MB,
    R_SFMA_ACCESS_RANGE_512MB
} r_sfma_AccessRange_t;
```

Table 5-10 Enumerator of r_sfma_AccessRange_t

Name	Description
R_SFMA_ACCESS_RANGE_32MB	Access range is 32 Mbytes.
R_SFMA_ACCESS_RANGE_64MB	Access range is 64 Mbytes.
R_SFMA_ACCESS_RANGE_128MB	Access range is 128 Mbytes.
R_SFMA_ACCESS_RANGE_256MB	Access range is 256 Mbytes.
R_SFMA_ACCESS_RANGE_512MB	Access range is 512 Mbytes.

See also

None

5.3.8 r_sfma_FlashRegister_t**Description**

This type describes the register of the serial flash memory.

Definition

```
typedef enum
{
    R_SFMA_STATUS_REG = 0,
    R_SFMA_CONFIG_REG,
    R_SFMA_NONE_REG
} r_sfma_FlashRegister_t;
```

Table 5-11 Enumerator of r_sfma_FlashRegister_t

Name	Description
R_SFMA_STATUS_REG	Status Register.
R_SFMA_CONFIG_REG	Configuration Register.
R_SFMA_NONE_REG	Unused Register.

See also

None

5.3.9 r_sfma_DummyCycle_t

Description

This type describes the data read dummy cycles.

Definition

```
typedef enum
{
    R_SFMA_DUMMY_1CYC = 0,
    R_SFMA_DUMMY_2CYC,
    R_SFMA_DUMMY_3CYC,
    R_SFMA_DUMMY_4CYC,
    R_SFMA_DUMMY_5CYC,
    R_SFMA_DUMMY_6CYC,
    R_SFMA_DUMMY_7CYC,
    R_SFMA_DUMMY_8CYC,
    R_SFMA_DUMMY_9CYC,
    R_SFMA_DUMMY_10CYC,
    R_SFMA_DUMMY_11CYC,
    R_SFMA_DUMMY_12CYC,
    R_SFMA_DUMMY_13CYC,
    R_SFMA_DUMMY_14CYC,
    R_SFMA_DUMMY_15CYC,
    R_SFMA_DUMMY_16CYC,
    R_SFMA_DUMMY_0CYC,
} r_sfma_DummyCycle_t;
```

Table 5-12 Enumerator of r_sfma_DummyCycle_t

Name	Description
R_SFMA_DUMMY_1CYC	1cycle.
R_SFMA_DUMMY_2CYC	2cycles.
R_SFMA_DUMMY_3CYC	3cycles.
R_SFMA_DUMMY_4CYC	4cycles.
R_SFMA_DUMMY_5CYC	5cycles.
R_SFMA_DUMMY_6CYC	6cycles.
R_SFMA_DUMMY_7CYC	7cycles.
R_SFMA_DUMMY_8CYC	8cycles.
R_SFMA_DUMMY_9CYC	9cycles.
R_SFMA_DUMMY_10CYC	10cycles.
R_SFMA_DUMMY_11CYC	11cycles.
R_SFMA_DUMMY_12CYC	12cycles.
R_SFMA_DUMMY_13CYC	13cycles.
R_SFMA_DUMMY_14CYC	14cycles.
R_SFMA_DUMMY_15CYC	15cycles.
R_SFMA_DUMMY_16CYC	16cycles.
R_SFMA_DUMMY_0CYC	Dummy cycle insertion disabled.

CONFIDENTIAL

Renesas Graphics Library Serial Flash Memory Interface A (SFMA) Driver

See also

None

5.3.10 r_sfma_AddressBitSize_t**Description**

This type describes the address bit width for a command of serial flash memory.

Definition

```
typedef enum
{
    R_SFMA_ADDRESS_SIZE_1BIT = 0,
    R_SFMA_ADDRESS_SIZE_2BIT,
    R_SFMA_ADDRESS_SIZE_4BIT
} r_sfma_AddressBitSize_t;
```

Table 5-13 Enumerator of r_sfma_AddressBitSize_t

Name	Description
R_SFMA_ADDRESS_SIZE_1BIT	The address bit width is 1bit.
R_SFMA_ADDRESS_SIZE_2BIT	The address bit width is 2bits.
R_SFMA_ADDRESS_SIZE_4BIT	The address bit width is 4bits.

See also

None

5.3.11 r_sfma_CacheMode_t

Description

This type describes the cache off and cache on with the burst length for the SFMA interface. 1 data unit is 64 bits long.

Definition

```
typedef enum
{
    R_SFMA_CACHE_OFF    = 0,
    R_SFMA_CACHE_BL1    = 1,
    R_SFMA_CACHE_BL2    = 2,
    R_SFMA_CACHE_BL3    = 3,
    R_SFMA_CACHE_BL4    = 4,
    R_SFMA_CACHE_BL5    = 5,
    R_SFMA_CACHE_BL6    = 6,
    R_SFMA_CACHE_BL7    = 7,
    R_SFMA_CACHE_BL8    = 8,
    R_SFMA_CACHE_BL9    = 9,
    R_SFMA_CACHE_BL10   = 10,
    R_SFMA_CACHE_BL11   = 11,
    R_SFMA_CACHE_BL12   = 12,
    R_SFMA_CACHE_BL13   = 13,
    R_SFMA_CACHE_BL14   = 14,
    R_SFMA_CACHE_BL15   = 15,
    R_SFMA_CACHE_BL16   = 16
} r_sfma_CacheMode_t;
```

Table 5-14 Enumerator of r_sfma_CacheMode_t

Name	Description
R_SFMA_CACHE_OFF	SFMA cache off.
R_SFMA_CACHE_BL1	SFMA cache on, with the burst length 1 data unit.
R_SFMA_CACHE_BL2	SFMA cache on, with the burst length 2 data units.
R_SFMA_CACHE_BL3	SFMA cache on, with the burst length 3 data units.
R_SFMA_CACHE_BL4	SFMA cache on, with the burst length 4 data units.
R_SFMA_CACHE_BL5	SFMA cache on, with the burst length 5 data units.
R_SFMA_CACHE_BL6	SFMA cache on, with the burst length 6 data units.
R_SFMA_CACHE_BL7	SFMA cache on, with the burst length 7 data units.
R_SFMA_CACHE_BL8	SFMA cache on, with the burst length 8 data units.
R_SFMA_CACHE_BL9	SFMA cache on, with the burst length 9 data units.
R_SFMA_CACHE_BL10	SFMA cache on, with the burst length 10 data units.
R_SFMA_CACHE_BL11	SFMA cache on, with the burst length 11 data units.
R_SFMA_CACHE_BL12	SFMA cache on, with the burst length 12 data units.
R_SFMA_CACHE_BL13	SFMA cache on, with the burst length 13 data units.
R_SFMA_CACHE_BL14	SFMA cache on, with the burst length 14 data units.
R_SFMA_CACHE_BL15	SFMA cache on, with the burst length 15 data units.
R_SFMA_CACHE_BL16	SFMA cache on, with the burst length 16 data units.

See also

None

5.3.12 r_sfma_WindowMode_t**Description**

This type describes the Window for SFMA address map.

Definition

```
typedef enum
{
    R_SFMA_WINDOWMODE_PRIMARY = 0,
    R_SFMA_WINDOWMODE_PROGRAM,
} r_sfma_WindowMode_t;
```

Table 5-15 Enumerator of r_sfma_WindowMode_t

Name	Description
R_SFMA_WINDOWMODE_PRIMARY	Primary Data Read Window.
R_SFMA_WINDOWMODE_PROGRAM	Programming Window.

See also

None

5.4 Structure Type

This section shows the structure used in SFMA API Function.

5.4.1 r_sfma_FlashRegInfo_t

Description

This type describes the register information of the serial flash memory.

Definition

```
typedef struct
{
    r_sfma_FlashRegister_t Reg;
    uint8_t BitPosition;
} r_sfma_FlashRegInfo_t;
```

Table 5-16 Member of r_sfma_FlashRegInfo_t

Name	Description
Reg	The register name of serial flash memory.
BitPosition	The bit position of status register or configuration register when 'Reg' is selected R_SFMA_STATUS_REG or R_SFMA_CONFIG_REG. The value is ignored when 'Reg' is selected R_SFMA_NONE_REG.

See also

r_sfma_FlashRegister_t

5.4.2 r_sfma_FlashRegSetParam_t**Description**

This type describes the register information and setting parameter of the serial flash memory.

Definition

```
typedef struct
{
    r_sfma_FlashRegister_t Reg;
    uint8_t BitMask;
    uint8_t BitSet;
} r_sfma_FlashRegSetParam_t;
```

Table 5-17 Member of r_sfma_FlashRegSetParam_t

Name	Description
Reg	The register name of serial flash memory.
BitMask	The bit mask of status register or configuration register when 'Reg' is selected R_SFMA_STATUS_REG or R_SFMA_CONFIG_REG. The value is ignored when 'Reg' is selected R_SFMA_NONE_REG.
BitSet	The bit value to status register or configuration register when 'Reg' is selected R_SFMA_STATUS_REG or R_SFMA_CONFIG_REG. The value is ignored when 'Reg' is selected R_SFMA_NONE_REG.

See also

r_sfma_FlashRegister_t

5.4.3 r_sfma_FlashCommand_t**Description**

This type describes the command information of the serial flash memory.
 This has the register information and the read dummy cycle information
 Some commands are not supported depending on the kind of serial flash memory connected.
 For incompatible commands, specify 0xFF.

Definition

```
typedef struct
{
    r_sfma_FlashRegInfo_t      QuadIoMode;
    r_sfma_FlashRegInfo_t      BlockProtect;
    r_sfma_FlashRegInfo_t      WriteInProgress;
    r_sfma_FlashRegSetParam_t   DummyCycle;
    r_sfma_DummyCycle_t         ReadSdrDummyCycle;
    r_sfma_DummyCycle_t         ReadSdrIoDualDummyCycle;
    r_sfma_DummyCycle_t         ReadSdrIoQuadDummyCycle;
    r_sfma_DummyCycle_t         ReadDdrDummyCycle;
    r_sfma_DummyCycle_t         ReadDdrIoDualDummyCycle;
    r_sfma_DummyCycle_t         ReadDdrIoQuadDummyCycle;
    r_sfma_AddressBitSize_t     WriteAddressBit;
    uint8_t                     ReadSdr;
    uint8_t                     ReadSdrDual;
    uint8_t                     ReadSdrQuad;
    uint8_t                     ReadSdrIoDual;
    uint8_t                     ReadSdrIoQuad;
    uint8_t                     ReadDdr;
    uint8_t                     ReadDdrDual;
    uint8_t                     ReadDdrQuad;
    uint8_t                     ReadDdrIoDual;
    uint8_t                     ReadDdrIoQuad;
    uint8_t                     WriteSdr;
    uint8_t                     WriteSdrDual;
    uint8_t                     WriteSdrQuad;
    uint8_t                     WriteDdr;
    uint8_t                     WriteDdrDual;
    uint8_t                     WriteDdrQuad;
    uint8_t                     WriteEnable;
    uint8_t                     Erase;
    uint8_t                     ReadStatus1;
    uint8_t                     ReadStatus2;
    uint8_t                     WriteStatus;
    uint8_t                     ExitExtAddrSpc;
    uint8_t                     ResetEnable;
    uint8_t                     Reset;
    uint8_t                     PerfEnhanceInd;
} r_sfma_FlashCommand_t;
```

CONFIDENTIAL

Renesas Graphics Library Serial Flash Memory Interface A (SFMA) Driver

Table 5-18 Member of `r_sfma_FlashCommand_t`

Name	Description
QuadIoMode	Quad I/O operation.
BlockProtect	Block Protection.
WriteInProgress	Write in progress.
DummyCycle	Dummy cycle.
ReadSdrDummyCycle	Alpha mode to use.
ReadSdrIoDualDummyCycle	Read SDR Dummy cycles.
ReadSdrIoQuadDummyCycle	Read SDR Dual Dummy cycles
ReadDdrDummyCycle	Read SDR Quad Dummy cycles
ReadDdrIoDualDummyCycle	Read DDR Dummy cycles.
ReadDdrIoQuadDummyCycle	Read DDR Dual Dummy cycles
WriteAddressBit	Read DDR Quad Dummy cycles
ReadSdr	SDR read.
ReadSdrDual	SDR dual read.
ReadSdrQuad	SDR quad read.
ReadSdrIoDual	SDR dual I/O read.
ReadSdrIoQuad	SDR quad I/O read.
ReadDdr	DDR read.
ReadDdrDual	DDR dual read.
ReadDdrQuad	DDR quad read.
ReadDdrIoDual	DDR dual I/O read.
ReadDdrIoQuad	DDR quad I/O read.
WriteSdr	SDR program page.
WriteSdrDual	SDR program page dual.
WriteSdrQuad	SDR program page quad.
WriteDdr	DDR program page.
WriteDdrDual	DDR program page dual.
WriteDdrQuad	DDR program page quad.
WriteEnable	Write enable.
Erase	Erase sector.
ReadStatus1	Read status register.
ReadStatus2	Read config register.
WriteStatus	Write status register.
ExitExtAddrSpc	Exit external address space.
ResetEnable	Reset enable.
Reset	Reset device.
PerfEnhanceInd	Performance enhance indicator. e.g. Sets to "0xA5" in case of connecting MX25L51245G

See also

`r_sfma_FlashRegInfo_t`
`r_sfma_FlashRegSetParam_t`
`r_sfma_DummyCycle_t`
`r_sfma_AddressBitSize_t`

5.4.4 r_sfma_Config_t

Description

This type describes the configuration of the unit

Definition

```
typedef struct
{
    r_sfma_Mode_t           Mode;
    r_sfma_MemoryNum_t      MemoryNum;
    r_sfma_DataTransferMode_t DataTransferMode;
    r_sfma_AddressMode_t    AddressMode;
    uint32_t                SerialFlashMemoryPageSize;
    uint32_t                SerialFlashMemoryMaxClock;
    uint32_t                SerialFlashMemorySectorSize;
    uint64_t                SerialFlashMemorySize;
    r_sfma_FlashCommand_t   *Command;
    r_sfma_CacheMode_t      CacheMode;
    uint32_t                Calibration;
    uint32_t                PerformanceEnMode;
} r_sfma_Config_t;
```

Table 5-19 Member of r_sfma_Config_t

Name	Description
Mode	The operating mode.
MemoryNum	The number of serial flash memory to connect.
DataTransferMode	The data transfer mode.
AddressMode	The format of address output to the serial flash memory.
SerialFlashMemoryPageSize	Page size of connected serial flash memory (Byte).
SerialFlashMemoryMaxClock	Specify maximum clock speed of connected the serial flash memory (Hz). Bit rate of SPBCLK is decided by SerialFlashMemoryMaxClock and "BΦ" which is input to SFMA H/W macro. And setting of "BΦ" is outside RGL. For example, in case "BΦ" is set to 160 MHz, if SerialFlashMemoryMaxClock is set to 80 MHz, "BΦ" is divided by 2 and bit rate of SPBCLK is set to 80 MHz. But if SerialFlashMemoryClock is set to 60 MHz, "BΦ" is divided by 4 (because division ratio can be set only to even number) then bit rate of SPBCLK is set to 40 MHz. See Table 3-2 about available maximum clock. Refer to RH850/D1L/D1M Group User's Manual: Hardware about "BΦ" and SPBCLK.
SerialFlashMemorySectorSize	Erase Sector Size of connected the serial flash memory (Byte). If the MemoryNum of the r_sfma_Config_t structure is R_SFMA_MEMORY_DUAL, this size must set double size of sector size of the serial flash memory. (e.g. This size is 8 Kbytes when connecting two MX25L51245G.)
SerialFlashMemorySize	Size of connected serial flash memory (Byte). This size is total size of connected serial flash memory. (e.g. This size is 128 MBytes(128*1024*1024), when connecting two MX25L51245G)
Command	The serial flash commands.
CacheMode	The cache settings for the SFMA interface.
Calibration	Calibration setting for the phase between SPBCLK, sampling point, and input / output data. This value sets to CKDLYOC[2:0] and CKDLYRX[2:0] of CKDLY(Clock phase adjust register). Specifies the value depending on the device. Refer to RH850/D1L/D1M Group User's Manual: Hardware about CKDLY. See 3.2.5 for details.
PerformanceEnMode	The option to enable the performance enhance mode. See Table 5-3 .

See also

- r_sfma_Mode_t
- r_sfma_MemoryNum_t
- r_sfma_DataTransferMode_t
- r_sfma_AddressMode_t
- r_sfma_FlashCommand_t
- r_sfma_CacheMode_t

Revision History	Renesas Graphics Library Serial Flash Memory Interface A (SFMA) driver User's Manual: Software
------------------	--

Rev.	Date	Description	
		Page	Summary
0.1	Nov 22, 2019	-	First edition.
1.0	April 24, 2020	-	Update Revision.

Renesas Graphics Library
Serial Flash Memory Interface A (SFMA) driver
User's Manual: Software

Publication Date: Rev.0.1 Nov 22, 2019
 Rev.1.0 April 24, 2020

Published by: Renesas Electronics Corporation



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics Corporation

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc. Milpitas Campus

1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.

Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics America Inc. San Jose Campus

6024 Silver Creek Valley Road, San Jose, CA 95138, USA

Tel: +1-408-284-8200, Fax: +1-408-284-2775

Renesas Electronics Canada Limited

9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3

Tel: +1-905-237-2004

Renesas Electronics Europe GmbH

Arcadiastrasse 10, 40472 Düsseldorf, Germany

Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.

Room 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, China

Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai 200333, China

Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited

Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong

Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan

Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.

80 Bendemeer Road, #06-02 Singapore 339949

Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.

Unit No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia

Tel: +60-3-5022-1288, Fax: +60-3-5022-1290

Renesas Electronics India Pvt. Ltd.

No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India

Tel: +91-80-67208700

Renesas Electronics Korea Co., Ltd.

17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea

Tel: +82-2-558-3737, Fax: +82-2-558-5338



ルネサスエレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<https://www.renesas.com/contact/>

Renesas Graphics Library
Serial Flash Memory Interface A (SFMA) driver



Renesas Electronics Corporation