RZ/A1H Group

Example of Booting from Serial Flash Memory

Abstract

This application note describes an example of booting from the serial flash memory via the SPI multi-I/O bus controller (hereinafter called "SPIBSC") of by using the boot mode 3 (serial flash boot) function.

Products

RZ/A1H

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Specifications

1.1 Booting from Serial Flash Memory

In boot mode 1, the RZ/A1H boots from the serial flash memory allocated to the SPI multi-I/O bus space (hereinafter called "serial flash boot"). Figure 1.1 shows the Conceptual Diagram of Serial Flash Boot Operation.

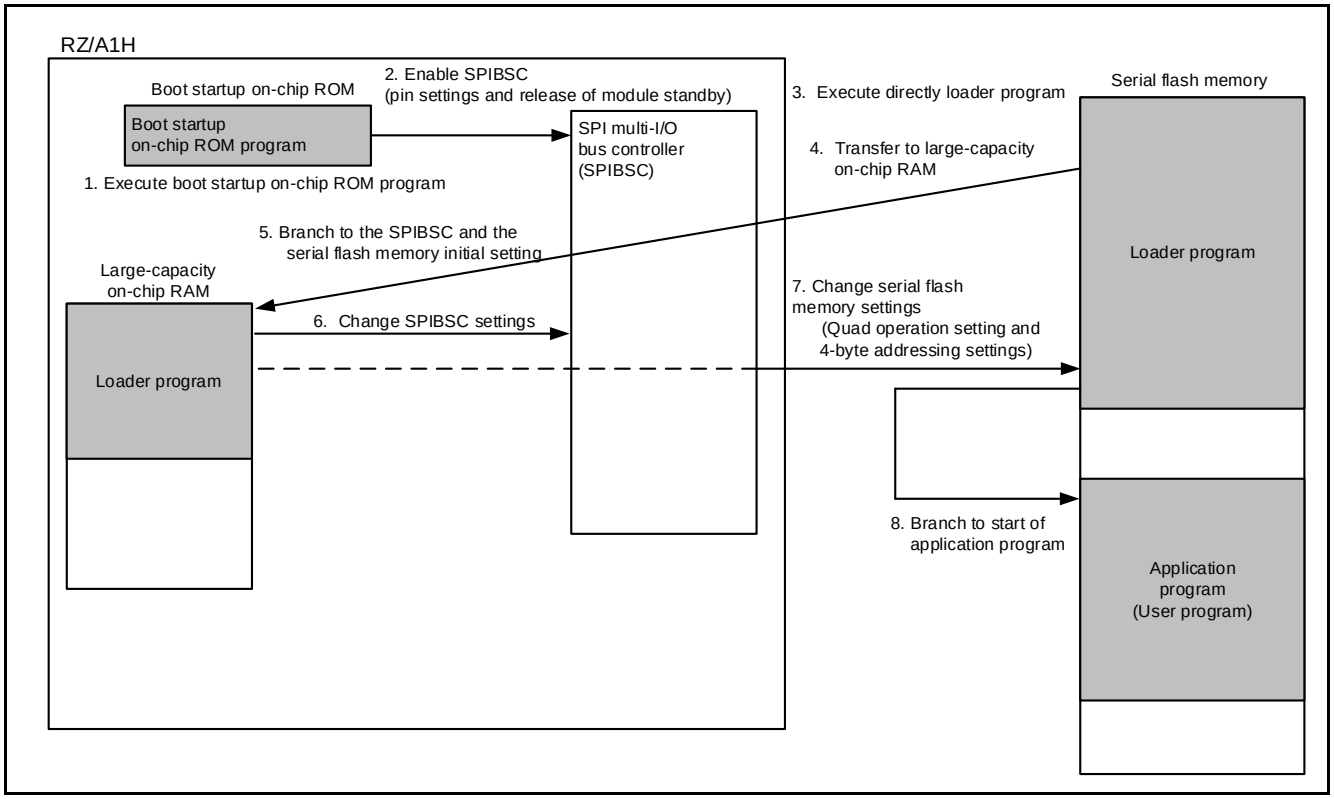


Figure 1.1 Conceptual Diagram of Serial Flash Boot Operation

The conceptual diagram of serial flash boot operation is described below.

(1) When the RZ/A1H starts up by serial flash boot, the boot startup on-chip ROM program runs after power-on reset is canceled.
(2) The boot startup on-chip ROM program sets the SPIBSC to external address space read mode to enable to directly run programs allocated to the SPI multi-I/O bus space.
(3) Execute directly the loader program stored in the serial flash memory.
(4) The loader program is transferred from the serial flash memory to the large-capacity on-chip RAM by section initialization of the loader program.
(5) Branch to the SPIBSC and the serial flash memory initial setting transferred to the large-capacity on-chip RAM.
(6) The loader program changes the SPIBSC settings.
(7) The loader program changes the serial flash memory settings.
(8) Execution branches to the start address of the application program.

The boot startup on-chip ROM program makes settings to allow common access to typical serial flash memory devices, so it is necessary to provide the optimal settings to the serial flash memory used by the customer. This application note describes how to allocate the loader program to the start address (H'1800_0000) of the SPI multi-I/O bus space branched by the boot startup on-chip ROM program, and then branch to the customer-created application program (user program) after the loader program are provided optimal settings to the serial flash memory used by the customer

This application provides a method for providing optimal settings to the serial flash memory used by the customer using the Loader program and a method for creating the application program (user program).

1.2 Peripheral Functions Used

This sample code not only configures the SPIBSC but also initializes the clock pulse oscillator, interrupt controller, bus state controller, general-purpose input/output ports, memory management unit, primary cache (L1 cache), and secondary cache (L2 cache).

In this application note, the Clock pulse generator is referred to as the CPG, the Interrupt controller as the INTC, the Bus state controller as the BSC, the OS timer as the OSTM, the Serial communication interface with FIFO as the SCIF, the General I/O ports as the PORT, the Power-down modes as the STB, and the Memory management unit as the MMU.

Table 1.1 summarizes Peripheral Functions and Their Applications, and Figure 1.2 shows Operating Environment for the sample code.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
SPI multi-I/O bus controller (SPIBSC)	When set to external address space read mode, it generates signals that enable the CPU to directly read from serial flash memory connected to the SPI multi-I/O bus space.
Clock pulse generator (CPG)	Generate the operating frequency of the RZ/A1H.
Interrupt controller (INTC)	Control OSTM channel 0 interrupt.
Bus state controller (BSC)	Generate signals for using the SDRAM in the CS3 space (Note).
OS timer (OSTM)	Generate, from the OSTM channel 0 timer, the intervals at which the LED are turned on and off.
Serial communication interface with FIFO (SCIF)	Communicate between SCIF channel 0 and the host PC.
General I/O ports (PORT)	Switch multiplexed pin functions for SPIBSC, CS3, and SCIF channel 0. Control pin for LED on/off.
Power-down modes (STB)	Cancel the module standby state of the RZ/A1H's peripheral I/O. Enable writing to the on-chip data retention RAM.
Memory management unit (MMU), L1 cache, and L2 cache	Generate conversion tables such as specifying valid area of L1 cache or specifying memory type in the RZ/A1H external address area. Enable the L1 and L2 caches.

Note: RZ/A1H board (Renesas Start Kit+ for RZ/A1H) implements the SDRAM (Samsung K4S561632D) in the CS3 space. The settings of the BSC and multiplexed pins for using the SDRAM, which lie in the configuration section of the source code, are initially disabled. When using the SDRAM, modify the source code accordingly. When two serial flash memory chips are to be used (in 8-bit access mode), the SDRAM is not available because it uses pins that are shared with the SDRAM control signals.

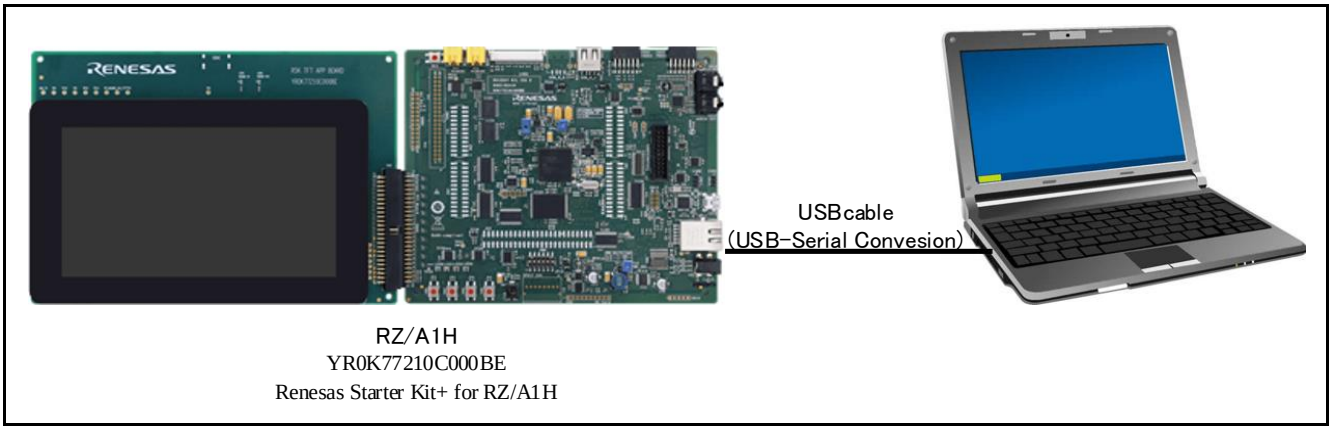


Figure 1.2 Operating Environment

2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	RZ/A1H
Operating frequency*	CPU clock (Iϕ): 400 MHz Internal bus clock (Bϕ): 133.33MHz Peripheral clock 1 (P1ϕ): 66.67MHz Peripheral clock 0 (P0ϕ): 33.33MHz
Operating voltage	Power supply voltage (I/O): 3.3V Power supply voltage (internal): 1.18V
Integrated development environment	e2 studio v7.8.0
C compiler	GNU ARM Embedded Toolchain 6-2017-q2-update
Operating mode	Boot mode 3 (Serial flash boot)
Board used	RZ/A1H board YR0K77210C000BE (hereinafter called "Renesas Start Kit+ for RZ/A1H")
Communications settings of the terminal software	• Baud rate: 115200bps • Data length: 8 bits • Parity: None • Stop bits: 1 bit • Flow control: None
Devices used (functions used on the board)	• Serial flash memory allocated to SPI multi-I/O bus space - Manufacturer: CYPRESS Inc. - Product No.: S25FL512S • RL78/G1C (Convert between USB communication and serial communication to communicate with the host PC.) • LED0

Note: * The operating frequency used in clock mode 0 (Clock input of 13.33MHz from EXTAL pin)

3. Reference Application Notes

For additional information associated with this document, refer to the following application notes.

- RZ/A1H Group I/O definition header file <iodefine.h> (R01AN1860EJ)

4. Hardware

4.1 Hardware Configuration

Figure 4.1 shows the Connection Example for Booting from Serial Flash Memory by using boot mode 3.

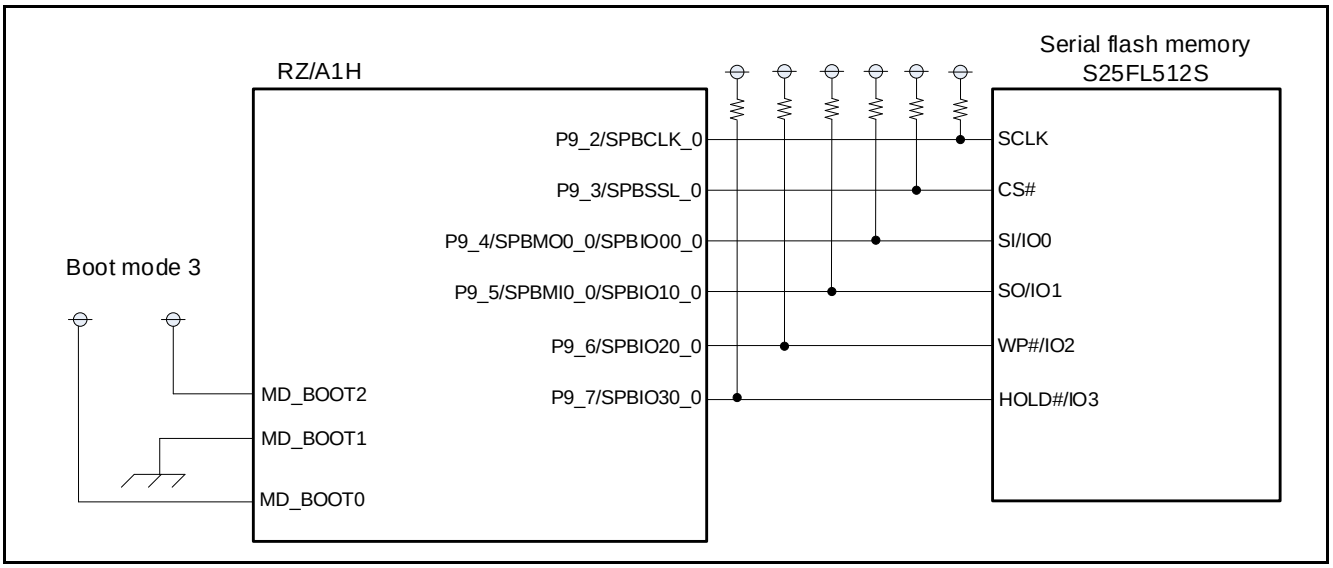


Figure 4.1 Connection Example for Booting from Serial Flash Memory

4.2 Pins Used

Table 4.1 lists the Pins Used and Their Functions.

Table 4.1 Pins Used and Their Functions

Pin Name	I/O	Function
SPBCLK_0	Output	Clock output
SPBSSL_0	Output	Slave select
SPBMO0_0/SPBIO00_0	I/O	Master send data: data 0
SPBMI0_0/SPBIO10_0	I/O	Master input data: data 1
SPBIO20_0	I/O	Data 2
SPBIO30_0	I/O	Data 3
MD_BOOT0 SW6-1	Input	Select boot mode MD_BOOT0: 1, MD_BOOT1: 0, MD_BOOT2: 1 (Set to boot mode 3)
MD_BOOT1 SW6-2	Input	
MD_BOOT2 SW6-3	Input	
P7_1	Output	Turns on and off LED0.
TxD2(P3_0)	Output	Serial transmit data signal

5. Software

5.1 Operation Overview

This section provides an overview of the sample code operation presented in this application note.

5.1.1 Terms Related to Serial Flash Boot

Table 5.1 lists the Terms Related to Serial Flash Boot Operation described in this application note.

Table 5.1 Terms Related to Serial Flash Boot Operation

Term	Description
Boot startup on-chip ROM program	This program provides settings to directly execute the programs stored in the serial flash memory connected to the SPI multi-I/O bus space when started up in boot mode 3 (serial flash boot). The RZ/A1H branches to the address of H'1800_0000 which is the start address of the SPI multi-I/O bus space after the boot startup on-chip ROM program has been executed. Note that the boot startup on-chip ROM program makes settings to enable common access to typical serial flash memory devices. Since this program is stored in the on-chip ROM of the RZ/A1H, it does not need to be created by the customer.
Loader program	This program is executed after the processing of the boot startup on-chip ROM program has been completed. The Loader program makes settings to the SPIBSC and to the registers in the serial flash memory corresponding to the serial flash memory used by the customer, and then branches to the start address of the application program. SPIBSC The Loader program should be created by the customer according to the specifications of the serial flash memory to be used while referring to this application note. In the sample code, the initial settings are optimized for use with the CYPRESS serial flash memory (product No.: S25FL512S).
Application program (User program)	This program should be created by customers depending on their system to be used.

5.1.2 Operation Overview of Sample Code Overall

The sample code comprises the Loader program executed from the boot startup on-chip ROM program and the application program.

(1) Loader program

The Loader program provides optimal settings to the serial flash memory used (CYPRESS serial flash memory (product No.: S25FL512S)). The Loader program is located at the start address (H'1808_0000) of the SPI multi-I/O bus space, which is branched from the boot startup on-chip ROM program. After the Loader program runs, it branches to the start address of the application program.

(2) Application program (User program)

This is an application program to be executed after optimal settings for the serial flash memory are provided in the Loader program. In the sample code, the application program is located at address H'1808_0000.

Figure 5.1 shows the Operation Overview of Sample Code presented in this application note.

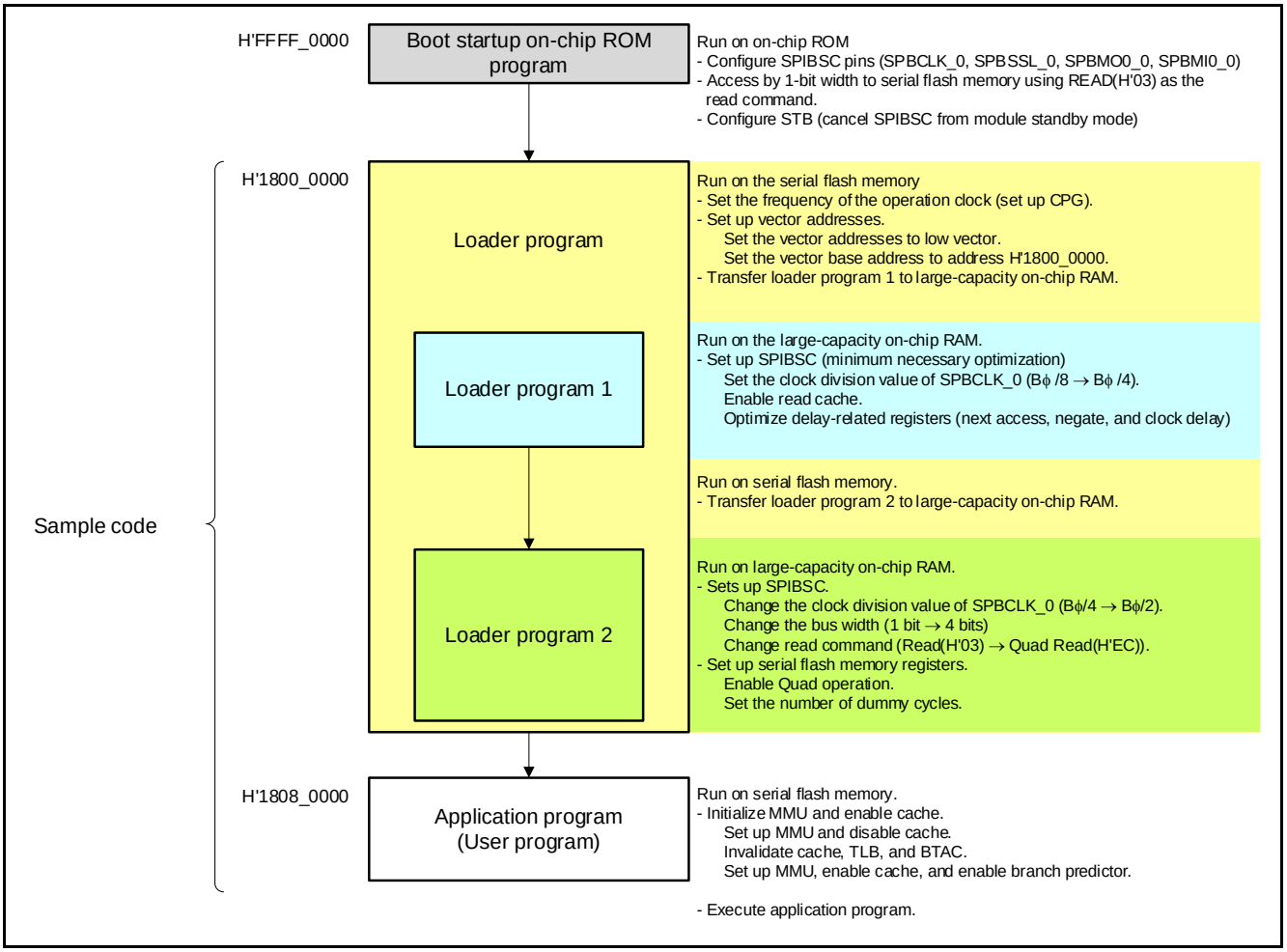


Figure 5.1 Operation Overview of Sample Code

5.1.3 Operation Overview of Loader program

The Loader program is executed after the boot startup on-chip ROM program. The Loader program should be located at the start address (H'1800_0000) of the SPI multi-I/O bus space branched from the boot startup on-chip ROM program.

The boot startup on-chip ROM program makes settings to enable the SPIBSC to operate in external address space read mode. These settings cause the RZ/A1H to convert read operations targeting the SPI multi-I/O bus space to SPI communication so that the direct read operations are enabled to the connected serial flash memory. This makes it possible for the RZ/A1H to directly run programs allocated to the SPI multi-I/O bus space. The settings for commands targeting the serial flash memory used in SPI communication conversion allow common access to typical serial flash memory devices, so it is necessary to provide the optimal settings to the serial flash memory used by the customer.

Refer to Table 5.2, Table 5.3, and Table 5.4 for the settings made after the boot startup on-chip ROM program has been executed.

The optimal settings for the serial flash memory are provided in two places: the registers in the SPIBSC module (hereinafter called "SPIBSC settings") and the registers in the serial flash memory (hereinafter called "serial flash memory settings"). In the Loader program of the sample code, the optimal settings are required when CYPRESS serial flash memory (product No.: S25FL512S) is used.

The Loader program comprises Loader program 1 and Loader program 2, as described below. Each is designed to be transferred from the SPI multi-I/O bus space to the large-capacity on-chip RAM and then executed from the large-capacity on-chip RAM.

(1) Loader program 1

The Loader program 1 sets up the SPIBSC registers so as to shorten the delay times (next access delay, SPBSSL negate delay, and clock delay), specify the transfer bit rate, and enable the read cache. The program size is comparatively compact because the content of processing is small.

(2) Loader program 2

The Loader program 2 sets up the SPIBSC registers so as to specify a data bus width of 4 bits, optimize the transfer bit rate according to the read command to be used, and output 4-byte addresses. It also sets up the registers in the serial flash memory (S25FL512S) so as to specify the number of dummy cycles for the serial flash memory, enable quad operation, and make changes for 4-byte addressing. The program size is larger than that of the initial setting program 1 because the volume of its processing is great.

The Loader program 1 and 2 cannot be set by the program allocated to the SPI multi-I/O bus space, so these programs should be executed from the large-capacity on-chip RAM. In the sample code, the Loader program 1 is transferred to the large-capacity on-chip RAM to be executed. Then the Loader program 2 is transferred to the large-capacity on-chip RAM to be executed after the possible optimal settings have been provided to the serial flash memory used. This reduces the overall running time of the Loader program.

Table 5.2, Table 5.3 and Table 5.4 list the settings made by the Loader program.

After the settings listed in Table 5.2, Table 5.3 and Table 5.4 are made, the Loader program branches to the start address of the application program. In the sample code, the application program is allocated to the area starting at H'1808_0000, which is to be the branch target.

Table 5.2 Settings for the Boot Startup On-Chip ROM Program and Loader programs (1/3)

	Item	Boot Startup On-Chip ROM Program	Loader program 1	Loader program 2
SPIBSC settings	Delay settings			
	Next access delay setting : SSLDR.SPNDL[2:0]	B'111 (8SPBCLK)	B'000 (1SPBCLK)	B'000 (1SPBCLK)
	SPBSSL negate delay setting : SSLDR.SLNDL[2:0]	B'111 (8.5SPBCLK)	B'000 (1.5SPBCLK)	B'000 (1.5SPBCLK)
	Clock delay setting : SSLDR.SCKDL[2:0]	B'111 (8SPBCLK)	B'000 (1SPBCLK)	B'000 (1SPBCLK)
	Serial clock : (@Bϕ = 133.33MHz)	Bϕ/8 = 16.67 [MHz]	Bϕ/4 = 33.33 [MHz]	Bϕ/2 = 66.67 [MHz]
	SPBCR.SPBR[7:0]	0	2	1
	SPBCR.BRDV[1:0]	3	0	0
	CPOL : CMNCR.CPOL	0	0	0
	CPHAT : CMNCR.CPHAT	0	0	0
	CPHAR : CMNCR.CPHAR	0	0	1
	SPBSSL output idle value fix :	Sets output values in SPBSSL negation period to the last bit value of the previous transfer	Sets output values in SPBSSL negation period to the last bit value of the previous transfer	Sets output values in SPBSSL negation period to Hi-z
	SPBIO30 and SPBIO31 setting	CMNCR.MOIIO3[1:0]=B'10	CMNCR.MOIIO3[1:0]=B'10	CMNCR.MOIIO3[1:0]=B'11
	SPBIO20 and SPBIO21 setting	CMNCR.MOIIO2[1:0]=B'10	CMNCR.MOIIO2[1:0]=B'10	CMNCR.MOIIO2[1:0]=B'11
	SPBIO10 and SPBIO11 setting	CMNCR.MOIIO1[1:0]=B'10	CMNCR.MOIIO1[1:0]=B'10	CMNCR.MOIIO1[1:0]=B'11
	SPBIO00 and SPBIO01 setting	CMNCR.MOIIO0[1:0]=B'10	CMNCR.MOIIO0[1:0]=B'10	CMNCR.MOIIO0[1:0]=B'11
	Fixes the output value of the terminal :	Sets the terminal output value for 1-bit/2-bit size to the last bit value of the previous transfer	Sets the terminal output value for 1-bit/2-bit size to the last bit value of the previous transfer	Sets the terminal output value for 1-bit/2-bit size to Hi-z
	SPBIO30 and SPBIO31 setting	CMNCR.IO3FV[1:0]=B'01	CMNCR.IO3FV[1:0]=B'01	CMNCR.IO3FV[1:0]=B'11
	SPBIO20 and SPBIO21 setting	CMNCR.IO2FV[1:0]=B'00	CMNCR.IO2FV[1:0]=B'00	CMNCR.IO2FV[1:0]=B'11
	SPBIO00 and SPBIO01 setting	CMNCR.IO0FV[1:0]=B'00	CMNCR.IO0FV[1:0]=B'00	CMNCR.IO0FV[1:0]=B'11
	Number of serial flash : CMNCR.BSZ[1:0]	1 B'00	1 B'00	1 B'00
	Read cache : DRCR.RBE	0 (Disabled)	1 (Enabled)	1 (Enabled)
	Read data burst length : DRCR.RBURST[3:0]	1 data unit (8 bytes) B'0000	1 data unit (8 bytes) B'0000	4 data unit (32 bytes) B'0011
	Data bus width : DRENR.DRDB[1:0]	1 [bit] B'00	1 [bit] B'00	4 [bits] B'10
	Read command : DRCMR.CMD[7:0]	Read (03H) H'03	Read (03H) H'03	Quad I/O read (ECH) H'EC
	Command : DRENR.CDE	Output enabled 1	Output enabled 1	Output enabled 1
	Optional command : DRENR.OCDE	Output disabled 0	Output disabled 0	Output disabled 0

Table 5.3 Settings for the Boot Startup On-Chip ROM Program and Loader programs (2/3)

	Item	Boot Startup On-Chip ROM Program	Loader program 1	Loader program 2
SPIBSC settings	Address specification : DRENR.ADE[3:0]	Output address[23:0] B'0111	Output address[23:0] B'0111	Output address[31:0] B'1111
	Address bit size : DRENR.ADB[1:0]	1 [bit] B'00	1 [bit] B'00	4 [bit] B'10
	Option data : DRENR.OPDE[3:0]	Output disabled B'0000	Output disabled B'0000	Output OPD3 * B'1000
	Option data bit size : DRENR.OPDB[1:0]	–	–	4 [bit] B'10
	Option data : DROPR.OPD3[7:0] DROPR.OPD2[7:0] DROPR.OPD1[7:0] DROPR.OPD0[7:0]	 – – – –	 – – – –	 H'00 – – –
	Dummy cycle enable : DRENR.DME	Insertion disabled 0	Insertion disabled 0	Insertion enabled 1
	Dummy cycle bit size : DRDMCR.DMDB[1:0]	–	–	1 [bit] B'00
	Number of dummy cycles : DRDMCR.DMCYC[2:0]	–	–	4cycles B'011
	Extended external Address : DREAR.EAC[2:0] DREAR.EAV[7:0]	External address bits [24:0] enabled (Directly accessible 32MB spaces) B'000 H'00	External address bits [24:0] enabled (Directly accessible 32MB spaces) B'000 H'00	External address bits [25:0] enabled (Directly accessible 64MB spaces) B'001 H'00
	Transfer format : DRDRENR.ADDRE DRDRENR.OPDRE DRDRENR.DRDRE	Address, option data, and data are transferred in SDR mode. 0 0 0	Address, option data, and data are transferred in SDR mode. 0 0 0	Address, option data, and data are transferred in SDR mode. 0 0 0
	AC input characteristics adjustment : CKDLY.CKDLY[3:0]	B'0100	B'0100	B'0100
	AC output characteristics adjustment : SPODLY.SPODLY[15:0]	H'0000	H'0000	H'0000

Note: * The S25FL512S transits to the MODE when H'Ax (don't care "x") is input during the High Performance Read Mode indicator cycle that follows the address cycle. Since the RZ/A1H's external address space read mode does not support the data transfer in High Performance Read Mode, the sample code makes configuration so that the S25FL512S will not switch into the High Performance Read Mode by making configuration so that H'00 is output from the OPD3 when a QuadIO Read command is issued.

Table 5.4 Settings for the Boot Startup On-Chip ROM Program and Loader programs (3/3)

	Item	Boot Startup On-Chip ROM Program	Loader program 1	Loader program 2
Multiplexed pin settings	P9_2	SPBCLK_0	SPBCLK_0	SPBCLK_0
	P9_3	SPBSSL_0	SPBSSL_0	SPBSSL_0
	P9_4	SPBMO0_0 / SPBIO00_0	SPBMO0_0 / SPBIO00_0	SPBMO0_0 / SPBIO00_0
	P9_5	SPBMI0_0 / SPBIO10_0	SPBMI0_0 / SPBIO10_0	SPBMI0_0 / SPBIO10_0
	P9_6	P9_6	P9_7	SPBIO20_0
	P9_7	P9_6	P9_7	SPBIO30_0
Serial flash memory settings	Configuration Register	No change *	No change *	Quad operation enabled. QUAD=1
	Configuration Register	No change *	No change *	LC[1:0] = B'00
Other	Operating clock settings Clock input of 13.33MHz from EXTAL pin	Iϕ=133.33[MHz] Bϕ=133.33[MHz] P1ϕ=66.67[MHz] P0ϕ=33.33[MHz]	Iϕ=400[MHz] Bϕ=133.33[MHz] P1ϕ=66.67[MHz] P0ϕ=33.33[MHz]	Iϕ=400[MHz] Bϕ=133.33[MHz] P1ϕ=66.67[MHz] P0ϕ=33.33[MHz]
	CPU exception vector position	High vector (from H' FFFF_0000)	Low vector (from H'0000_0000)	Low vector (from H'0000_0000)

Note: * In serial flash booting (boot mode 1) of RZ/A1H, the boot program sets the SPIBSC register to issue a read command (opcode: 03H, address: 3 bytes, dummy cycle: none) as the command to the serial flash memory. Therefore if the condition of the serial flash memory can not be received the above read command by the register value in serial flash booting, it is a possibility that can not be normal boot.

5.1.4 Application Program (User Program)

(1) Operation of the application program (user program)

After a reset is cancelled, the boot startup on-chip ROM program and Loader program are executed in that order. Execution then transfers to the application program that is allocated to address H'1808_0000.

The application program executes the settings for the stack pointer and the MMU. It branches to main function.

In main function, the $Sub$$main function initializes the peripheral functions such as STB, BSC, INTC, and PORT,and sets the L1 and L2 caches to enabled, and enables the IRQ and FIQ interrupts. (in the sample code, the configuration for setting up the MMU and enabling the cache is executed by the application program and not by the Loader program).

The main function outputs the character strings to the terminal on the host PC connected with the serial interface and sets the OSTM channel 0 timer to interval timer mode to activate the timer. It generates the OSTM channel 0 interrupt with a cycle of 500ms and repeats turning on/off the LED on the Renesas Start Kit+ for RZ/A1H board every 500ms using such interrupt.

(2) Notes to be observed when creating an application program (user program)

The application program should be allocated to the address branched from the Loader program. Note that the application program should be allocated to the different sector in the serial flash memory from the one in the Loader program.

The sector size of the CYPRESS serial flash memory (S25FL512S) mounted on the Renesas Start Kit+ for RZ/A1H is 4KB. In the sample code, the application program is allocated to the address of H'1808_0000 (Sector no. 16). Figure 5.2 shows the Sample Code Program Allocation.

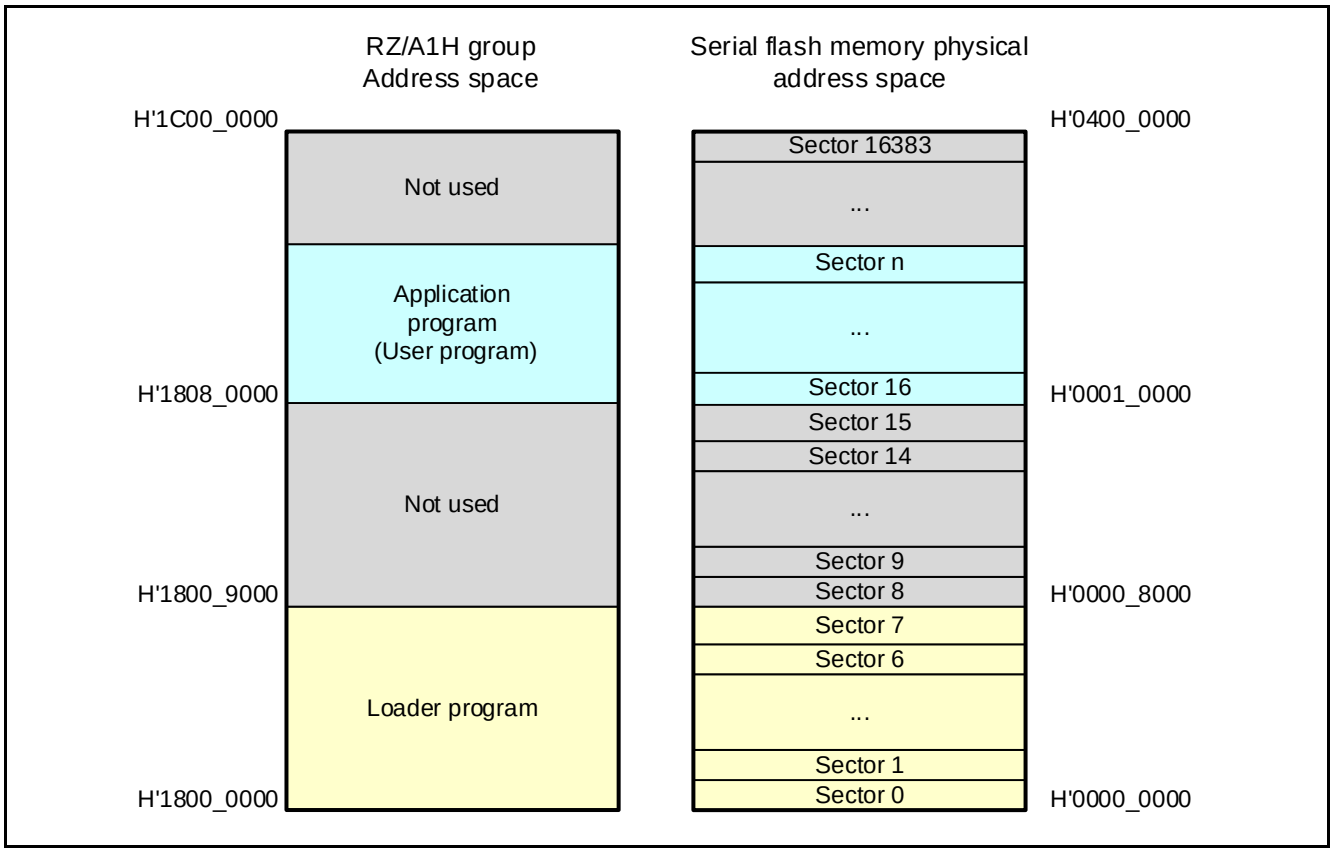


Figure 5.2 Sample Code Program Allocation

The start address of the application program can be changed by making the following changes:

- Project for the Loader program
  The branch to the starting address of the application program is executed by the loadrer program 2 (spibsc_init2.c). Specify the destination of branch with the macro definition "DEF_USER_PROGRAM_TOP" in spibsc_config.h.
- Project for the application program
  Change the allocation address so that the VECTOR_TABLE section of the application program matches the address that is specified in "DEF_USER_PROGRAM_TOP."

5.2 Peripheral Functions and Memory Allocation in Sample Code

5.2.1 Setting for Peripheral Functions

Table 5.5 lists the Setting for Peripheral Functions during execution of the sample code.

Table 5.5 Setting for Peripheral Functions

Module	Settings
CPG	CPU clock (Iϕ): 400 MHz Internal bus clock (Bϕ): 133.33 MHz Peripheral clock 1 (P1ϕ): 66.67 MHz Peripheral clock 0 (P0ϕ): 33.33 MHz
SPIBSC	When set to the external address space read mode, it generates the signals which enable the CPU to read directly from the serial flash memory connected to the SPI multi-I/O bus space. See Table 5.2, Table 5.3 and Table 5.4 for details.
PORT	Multiplexed pin functions are enabled on PORT9, PORT7, and PORT3. P9_2:SPBCLK_0 P9_3:SPBSSL_0 P9_4:SPBMO0_0/SPBIO00_0 P9_5:SPBMI0_0/SPBIO10_0 P9_6:SPBIO20_0 P9_7:SPBIO30_0 P7_1:LED on/off P3_0:TxD2 P3_2:RxD02*
STB	Writing to the on-chip RAM used for retention is enabled and clock supply to peripheral functions is enabled. Clock supply is enabled to all peripheral functions for which clock supply and stop control are supported by using STBCR2 to STBCR12.
OSTM	Sets the channel 0 in interval timer mode. Sets the timer counter to have interrupt request generated every 500ms when P0ϕ=33.33 MHz.
INTC	Initializes INTC, and registers and executes OSTM channel 0 interrupt (interrupt ID is 134) handler
SCIF	Sets the channel 0 in asynchronous communication mode. • Data length: 8 bits • Stop bit length: 1 bit • Parity: None Sets the clock source without frequency dividing and the bit rate value at 17. Sets the bit rate to be 115200bps when P1ϕ is 66.67MHz.

5.2.2 Memory Mapping

Figure 5.3 shows the RZ/A1H Group Address Space and Renesas Start Kit+ for RZ/A1H.

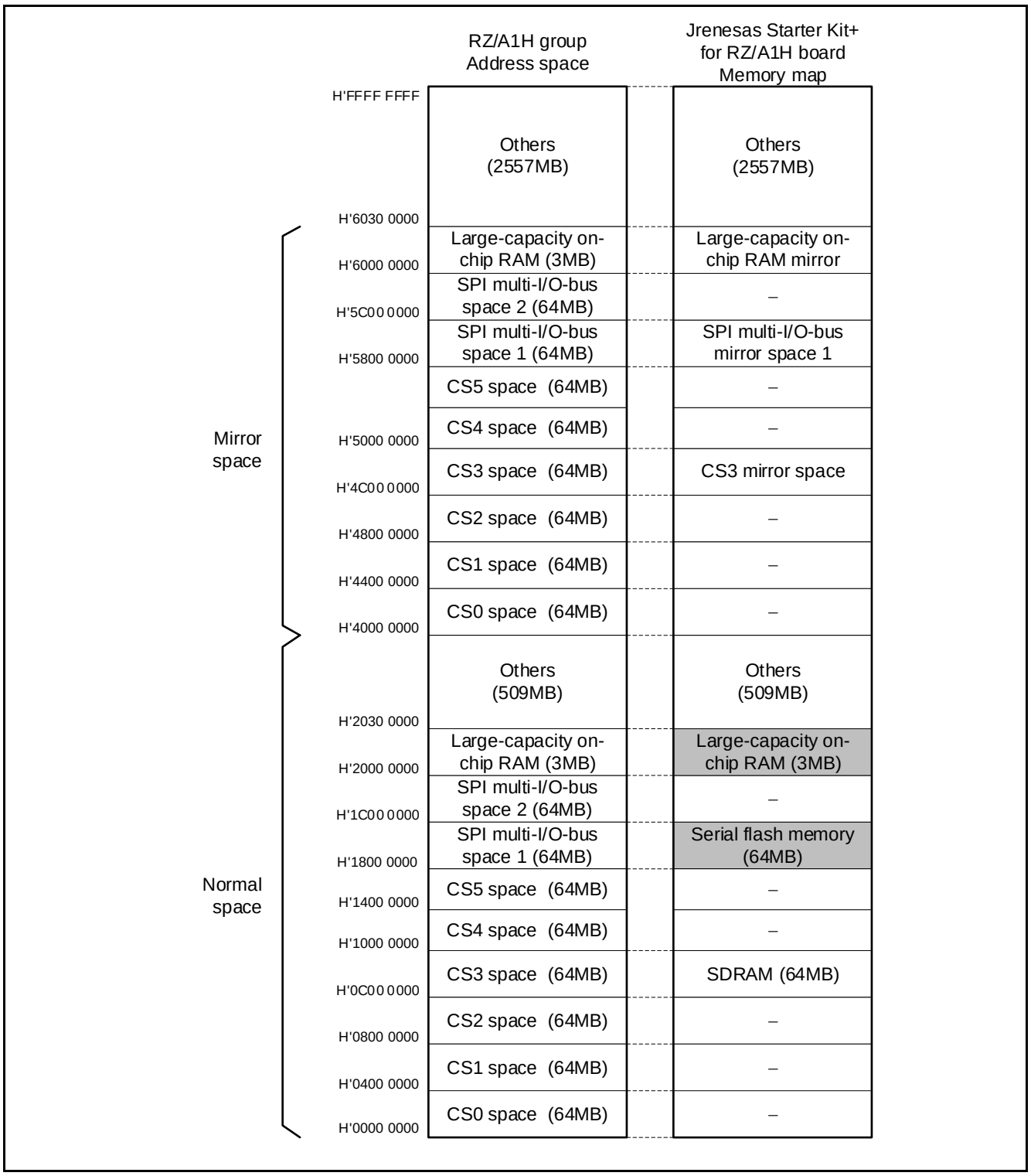


Figure 5.3 RZ/A1H Group Address Space and Renesas Start Kit+ for RZ/A1H Memory Mapping

5.2.3 Section Assignment in Sample Code

Table 5.6 shows the Sections to be Used by . Table 5.7 and Table 5.8 show the Sections to be Used by application program.

Table 5.6 Sections to be Used by Loader program

Area Name	Description	Type	Loading Area	Execution Area
VECTOR_TABLE	Exception processing vector table	Code	S-FLASH	S-FLASH
CODE_SPIBSC_INIT1	Code area for Loader program 1	Code	S-FLASH	LRAM
CODE_IO_REGRW	Program code area for read/write functions of I/O register	Code	S-FLASH	LRAM
CODE_SPIBSC_INIT2	Code area for Loader program 2	Code	S-FLASH	LRAM
DATA_SPIBSC_INIT2	DATA area for Loader program 2	RW Data	S-FLASH	LRAM
BSS_SPIBSC_INIT2	BSS area for Loader program 2	ZI Data	–	LRAM
RESET_HANDLER	Program code area of reset handler processing	Code	S-FLASH	S-FLASH
CODE	Program code area for defaults All the Code type sections which do not define section names with C source are assigned in this area.	Code	S-FLASH	S-FLASH
SVC_STACK	Stack area	ZI Data	–	LRAM

Note: "S-FLASH" and "LRAM" shown in Loading Area and Execution Area indicate the serial flash memory area and the large-capacity on-chip RAM area respectively.

Table 5.7 Sections to be Used by application program (1/2)

Area Name	Description	Type	Loading Area	Execution Area
VECTOR_TABLE	Exception processing vector table	Code	S-FLASH	S-FLASH
RESET_HANDLER	Program code area of reset handler processing This area consists of the following sections. • INITCA9CACHE (L1 cache setting) • INIT_TTB (MMU setting) • RESET_HANDLER (Reset handler)	Code	S-FLASH	S-FLASH
CODE_BASIC_SETUP	Program code area to optimize operating frequency and flash memory	Code	S-FLASH	S-FLASH
InRoot	This area consists of the sections located in the root area such as C standard library.	Code and RO Data	S-FLASH	S-FLASH
CODE_FPU_INIT	Program code area for NEON and VFP initializations This area consists of the following sections. • CODE_FPU_INIT • FPU_INIT	Code	S-FLASH	S-FLASH
CODE_RESET	Program code area for hardware initialization This area consists of the following sections. • CODE_RESET (Startup processing) • INIT_VBAR (Vector base setting)	Code	S-FLASH	S-FLASH
CODE	Program code area for defaults All the Code type sections which do not define section names with C source are assigned in this area.	Code	S-FLASH	S-FLASH
CONST	Constant data area for defaults All the RO Data type sections which do not define section names with C source are assigned in this area.	RO Data	S-FLASH	S-FLASH

Table 5.8 Sections to be Used by application program (2/2)

Area Name	Description	Type	Loading Area	Execution Area
VECTOR_MIRROR_ TABLE	Exception processing vector table (Section to transfer data to large-capacity on-chip RAM)	Code	S-FLASH	LRAM
CODE_HANDLER_ JMPTBL	Program code area for user-defined functions of IRQ interrupt handler	Code	S-FLASH	LRAM
CODE_HANDLER	Program code area of IRQ interrupt handler This area consists of the following sections. • CODE_HANDLER • IRQ_FIQ_HANDLER	Code	S-FLASH	LRAM
CODE_IO_REGRW	Program code area for read/write functions of I/O register	Code	S-FLASH	LRAM
CODE_CACHE_ OPERATION	Program code area for setting the L1 and L2 caches (see Note3)	Code	S-FLASH	LRAM
DATA_HANDLER_ JMPTBL	Registration table data area for user-defined functions of IRQ interrupt handler	RW Data	S-FLASH	LRAM
ARM_LIB_STACK	Application stack area	ZI Data	–	LRAM
IRQ_STACK	IRQ mode stack area	ZI Data	–	LRAM
FIQ_STACK	FIQ mode stack area	ZI Data	–	LRAM
SVC_STACK	Supervisor (SVC) mode stack area	ZI Data	–	LRAM
ABT_STACK	Abort (ABT) mode stack area	ZI Data	–	LRAM
TTB	MMU translation table area	ZI Data	–	LRAM
DATA	Data area with initial value for defaults All the RW Data type sections which do not define section names with C source are assigned in this area.	RW Data	S-FLASH	LRAM
BSS	Data area without initial value for defaults All the ZI Data type sections which do not define section names with C source area assigned in this area.	ZI Data	–	LRAM

Notes: 1. "S-FLASH" and "LRAM" shown in Loading Area and Execution Area indicate the serial flash memory area and the large-capacity on-chip RAM area respectively.
2. Basically the section name is set to be the same as the region's, however it consists of some sections in the areas of RESET_HANDLER, InRoot, CODE_FPU_INIT, CODE_RESET, CODE, CONST, CODE_HANDLER, DATA, and BSS. Refer to the ARM compiler toolchain manual about the region and the section.
3. This section should be placed in the cache-disabled area.

5.3 Interrupt Used

Table 5.9 lists the Interrupt Used in Sample Code (Application Program).

Table 5.9 Interrupt Used in Sample Code (Application Program)

Interrupt Source (Interrupt ID)	Priority	Processing Outline
OSTM0 (134)	5	Generates interrupts at 500 ms intervals.

5.4 Constants Used by the Loader program

Table 5.10 , Table 5.11 and Table 5.12 list the constants used by Loader program in the sample code.

Table 5.10 Constants Used in Sample Code (1/3)

Constant	Setting Value	Description
DEF_USER_PROGRAM_TOP	0x18080000	Start address of the application program
SPIBSC_1BIT	0	Sets the bit width used when issuing read commands to 1 bit.
SPIBSC_4BIT	2	Sets the bit width used when issuing read commands to 4 bits.
SPIBSC_CMNCR_BSZ_SINGLE	0	Sets the number of serial flash devices connected to SPIBSC to 1.
SPIBSC_CMNCR_BSZ_DUAL	1	Sets the number of serial flash devices connected to SPIBSC to 2.
SPIBSC_OUTPUT_ADDR_24	0x07	Specifies 24-bit address output.
SPIBSC_OUTPUT_ADDR_32	0x0f	Specifies 32-bit address output.
SPIBSC_OUTPUT_DISABLE	0	Specifies that no command, optional command, address, option data are output.
SPIBSC_OUTPUT_ENABLE	1	Specifies that a command, optional command, address, option data are output.
SPIBSC_OUTPUT_OPD_3	0x08	Specifies that option data OPD3 is output when a read command is issued.
SPIBSC_OUTPUT_OPD_32	0x0c	Specifies that option data OPD3 and OPD2 are output when a read command is issued.
SPIBSC_OUTPUT_OPD_321	0x0e	Specifies that option data OPD3, OPD2, and OPD1 are output when a read command is issued.
SPIBSC_OUTPUT_OPD_3210	0x0f	Specifies that option data OPD3, OPD2, OPD1, and OPD0 are output when a read command is issued.
SPIBSC_OUTPUT_SPID_8	0x08	Enables 8- (or 16-) bit transfer of data in SPI operation mode.
SPIBSC_OUTPUT_SPID_16	0x0c	Enables 16- (or 32-) bit transfer of data in SPI operation mode.
SPIBSC_OUTPUT_SPID_32	0x0f	Enables 32- (or 64-) bit transfer of data in SPI operation mode.
SPIBSC_SPISSL_NEGATE	0	Sets the SPBSSL signal state at transfer end to negate in SPI operation mode.
SPIBSC_SPISSL_KEEP	1	Specifies that the SPBSSL signal level is maintained from transfer end to start of next access in SPI operation mode.
SPIBSC_SPIDATA_DISABLE	0	Disables data read/write operations in SPI operation mode.
SPIBSC_SPIDATA_ENABLE	1	Enables data read/write operations in SPI operation mode.
SPIBSC_DUMMY_CYC_DISABLE	0	Specifies that no dummy cycles are inserted.
SPIBSC_DUMMY_CYC_ENABLE	1	Specifies that dummy cycles are inserted.

Table 5.11 Constants Used in Sample Code (2/3)

Constant	Setting Value	Description
SPIBSC_SDR_TRANS	0	Transfers the read from the serial flash memory in SDR mode.
SPIBSC_DDR_TRANS	1	Transfers the read from the serial flash memory in DDR mode.
SF_REQ_SERIALMODE	2	Specifies "single mode" for the serial flash memory registers.
SF_REQ_QUADMODE	3	Specifies "quad mode" for the serial flash memory registers.
SPIBSC_CKDLY_DEFAULT	0x0000A504	Defines the value to be loaded into the CKDLY register. Sets "B'0100" (initial value) to the CKDLY[3:0] bit.
SPIBSC_CKDLY_TUNING	0x0000A508	Defines the value to be loaded into the CKDLY register. Sets "B'1000" to the CKDLY[3:0] bit.
SPIBSC_SPODLY_DEFAULT	0xA5000000	Defines the value to be loaded into the SPODLY register. Sets "H'0000" (initial value) to the SPODLY[15:0] bit.
SPIBSC_SPODLY_TUNING	0xA5006363	Defines the value to be loaded into the SPODLY register. Sets "H'6363" to the SPODLY[15:0] bit.
SFLASH_MODECYC	 2 1	Specifies the period of the S25FL512S High Performance Read Mode Indicator. When SPIBSC_TRANS_MODE is set to SPIBSC_SDR_TRANS When SPIBSC_TRANS_MODE is set to SPIBSC_DDR_TRANS

Table 5.12 Constants Used in Sample Code (3/3)

Constant	Setting Value	Description
SFLASHCMD_READ_STATUS	0x05	S25FL512S Status Register Read command
SFLASHCMD_READ_CONFIG	0x35	S25FL512S Configuration Register Read command
SFLASHCMD_WRITE_STATUS	0x01	S25FL512S Write Status command
SFLASHCMD_WRITE_ENABLE	0x06	S25FL512S Write Enable command
SFLASHCMD_WRITE_DISABLE	0x04	S25FL512S write Disable command
STREG_SRWD_BIT	0x80	S25FL512S Status Register, SRWD bit
STREG_BPROTECT_BIT	0x1C	S25FL512S Status Register, BP bit
STREG_WEL_BIT	0x02	S25FL512S Status Register, WEL bit
STREG_WIP_BIT	0x01	S25FL512S Status Register, WIP bit
CFREG_LC0_BIT	0x80	S25FL512S Configuration Register, LC bit
CFREG_LC1_BIT	0x40	S25FL512S Configuration Register, LC bit
CFREG_QUAD_BIT	0x02	S25FL512S Configuration Register, QUAD bit
CFREG_FREEZE_BIT	0x01	S25FL512S Configuration Register, FREEZE bit

5.5 List of Structures/Unions Used by the Loader program

Table 5.13 to Table 5.21 list the structures used by Loader program in the sample code.

Table 5.13 Structure for Configuring the SPIBSC External Address Space Read Mode (st_spibsc_cfg_t) (1/4)

Member	Description
uint8_t udef_cmd	Read command • Specifies the read command output to the serial flash memory when converting read operations targeting the SPI multi-I/O bus space to SPI communication. • The setting value of this member is set in the CMD[7:0] bit field in the data read command setting register (DRCMR).
uint8_t udef_cmd_width	Read command bit width • Specifies the bit width used when issuing read commands. • Settable values: SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width • The setting value of this member is set in the CDB[1:0] bit field in the data read enable setting register (DRENR).
uint8_t udef_opd3 uint8_t udef_opd2 uint8_t udef_opd1 uint8_t udef_opd0	Option data • Specifies the option data output to the serial flash memory when converting read operations targeting the SPI multi-I/O bus space to SPI communication. • The setting values of these members are set in the OPD3[7:0], OPD2[7:0], OPD1[7:0], and OPD0[7:0] bit fields in the data read option setting register (DROPR).
uint8_t udef_opd_enable	Option data enable • Specifies whether or not option data is issued. • Settable values: SPIBSC_OUTPUT_DISABLE: No option data output. SPIBSC_OUTPUT_OPD_3: OPD3 is output. SPIBSC_OUTPUT_OPD_32: OPD3 and OPD2 are output. SPIBSC_OUTPUT_OPD_321: OPD3, OPD2, and OPD1 are output. SPIBSC_OUTPUT_OPD_3210: OPD3, OPD2, OPD1, and OPD0 are output. • The setting value of this member is set in the OPDE[3:0] bit field in the data read enable setting register (DRENR).
uint8_t udef_opd_width	Option data bit width • Specifies the bit width used when issuing option data. • Settable values: SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width • The setting value of this member is set in the OPDB[1:0] bit field in the data read enable setting register (DRENR).

Table 5.14 Structure for Configuring the SPIBSC External Address Space Read Mode (st_spibsc_cfg_t) (2/4)

Member	Description
uint8_t udef_dmycyc_num	Number of dummy cycles • Specifies the number of dummy cycles output to the serial flash memory when converting read operations targeting the SPI multi-I/O bus space to SPI communication. • Settable values: To be loaded with the results of calculating the value to be set in the DMCYC[2:0] bit field of the data read dummy cycle setting register (DRDMCR) based on the value defined in SPIBSC_DMYCYC_SETTING. Values that will resultantly yield number of 0 to 7 are allowed. For details on the calculation, see "Table 6.2 List of Macros for Customizing the Sample Code (1/2)".
uint8_t udef_dmycyc_enable	Dummy cycle enable • Specifies whether or not dummy cycles are inserted. • Settable values: SPIBSC_DUMMY_CYC_DISABLE: Insertion of dummy cycles disabled. SPIBSC_DUMMY_CYC_ENABLE: Insertion of dummy cycles enabled. • The setting value of this member is set in the DME bit field in the data read enable setting register (DRENR).
uint8_t udef_dmycyc_width	Dummy cycle bit width • Specifies the bit width used when issuing dummy cycles. • Settable values: SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width • The setting value of this member is set in the DMDB[1:0] bit field in the data read dummy cycle setting register (DRDMCR).
uint8_t udef_data_width	Data read bit width • The serial flash memory data read bit width used when converting read operations targeting the SPI multi-I/O bus space to SPI communication. • Settable values: SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width • The setting value of this member is set in the DRDB[1:0] bit field in the data read enable setting register (DRENR).

Table 5.15 SPIBSC External Address Read Mode Settings Structure (st_spibsc_cfg_t) (3/4)

Member	Description
uint8_t udef_spbr	Bit rate • Specifies the bit rate of the serial clock (SPBCLK) output to the serial flash memory when converting read operations targeting the SPI multi-I/O bus space to SPI communication. • Settable value: Used in conjunction with the bit rate division ratio setting (udef_brdv). • The setting value of this member is set in the SPBR[7:0] bit field in the bit rate setting register (SPBCR).
uint8_t udef_brdv	Bit rate division ratio setting • Specifies the bit rate of the serial clock (SPBCLK) output to the serial flash memory when converting read operations targeting the SPI multi-I/O bus space to SPI communication. • Settable value: Used in conjunction with the bit rate (udef_spbr). • The setting value of this member is set in the BRDV[1:0] bit field in the bit rate setting register (SPBCR).
uint8_t udef_addr_width	Address bit width • Specifies the bit width used to output addresses to the serial flash memory when converting read operations targeting the SPI multi-I/O bus space to SPI communication. • Settable values: SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width • The setting value of this member is set in the ADB[1:0] bit field in the data read enable setting register (DRENR).
uint8_t udef_addr_mode	Address enable • Specifies the format of address output to the serial flash memory when converting read operations targeting the SPI multi-I/O bus space to SPI communication. • Settable values: SPIBSC_OUTPUT_ADDR_24: 24-bit address output SPIBSC_OUTPUT_ADDR_32: 32-bit address output • The setting value of this member is set in the ADE[3:0] bit field in the data read enable setting register (DRENR).

Table 5.16 Structure for Configuring the SPIBSC External Address Space Read Mode (st_spibsc_cfg_t) (4/4)

Member	Description
uint8_t udef_drdrenr_addre	Address DDR enable • Selects the SDR/DDR transfer mode for addresses to be output in external address space read mode. • Settable values: SPIBSC_SDR_TRANS: SDR transfer SPIBSC_DDR_TRANS: DDR transfer • The setting value of this member is set in the ADDRE bit of the data read DDR enable register (DRDRENR).
uint8_t udef_drdrenr_opdre	Option data DDR enable • Selects the SDR/DDR transfer mode for option data to be output in external address space read mode. • Settable value: SPIBSC_SDR_TRANS: SDR transfer SPIBSC_DDR_TRANS: DDR transfer • The setting value of this member is set in the OPDRE bit of the data read DDR enable register (DRDRENR).
uint8_t udef_drdrenr_drdre	Transfer data DDR enable • Selects the SDR/DDR transfer mode for the data to be transferred in external address space read mode. • Settable value: SPIBSC_SDR_TRANS: SDR transfer SPIBSC_DDR_TRANS: DDR transfer • The setting value of this member is set in the DRDRE bit of the data read DDR enable register (DRDRENR).

Table 5.17 SPIBSC SPI Operating Mode Settings Structure (st_spibsc_spimd_reg_t) (1/5)

Member	Description
uint32_t cdb	Command bit width • Specifies the command bit width in SPI operation mode. • Settable values: SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width • The setting value of this member is set in the CDB[1:0] bit field in the SPI mode enable setting register (SMENR).
uint32_t ocdb	Optional command bit width • Specifies the optional command bit width in SPI operation mode. • Settable values: SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width • The setting value of this member is set in the OCDB[1:0] bit field in the SPI mode enable setting register (SMENR).
uint32_t adb	Address bit width • Specifies the address bit width in SPI operation mode. • Settable values: SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width • The setting value of this member is set in the ADB[1:0] bit field in the SPI mode enable setting register (SMENR).
uint32_t opdb	Option data bit width • Specifies the option data bit width in SPI operation mode. • Settable values: SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width • The setting value of this member is set in the OPDB[1:0] bit field in the SPI mode enable setting register (SMENR).
uint32_t spidb	Transfer data bit width • Specifies the transfer data bit width in SPI operation mode. • Settable values: SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width • The setting value of this member is set in the SPIDB[1:0] bit field in the SPI mode enable setting register (SMENR).
uint32_t cde	Command enable • Specifies whether or not commands are output in SPI operation mode. • Settable values: SPIBSC_OUTPUT_DISABLE: Output disabled. SPIBSC_OUTPUT_ENABLE: Output enabled. • The setting value of this member is set in the CDE bit field in the SPI mode enable setting register (SMENR).

Table 5.18 SPIBSC SPI Operating Mode Settings Structure (st_spibsc_spimd_reg_t) (2/5)

Member	Description
uint32_t ocde	Optional command enable • Specifies whether or not the optional command is output in SPI operation mode. • Settable values: SPIBSC_OUTPUT_DISABLE: Output disabled. SPIBSC_OUTPUT_ENABLE: Output enabled. • The setting value of this member is set in the OCDE bit field in the SPI mode enable setting register (SMENR).
uint32_t ade	Address enable • Specifies whether or not the address is output in SPI operation mode. • Settable values: SPIBSC_OUTPUT_DISABLE: Output disabled. SPIBSC_OUTPUT_ADDR_24: ADR[23:0] is output SPIBSC_OUTPUT_ADDR_32: ADR[31:0] is output • The setting value of this member is set in the ADE[3:0] bit field in the SPI mode enable setting register (SMENR).
uint32_t opde	Option data enable • Specifies whether or not the option data is output in SPI operation mode. • Settable values: SPIBSC_OUTPUT_DISABLE: Output disabled. SPIBSC_OUTPUT_OPD_3: OPD3 is output SPIBSC_OUTPUT_OPD_32: OPD3 and OPD2 are output SPIBSC_OUTPUT_OPD_321: OPD3, OPD2, and OPD1 are output SPIBSC_OUTPUT_OPD_3210: OPD3, OPD2, OPD1, and OPD0 are output • The setting value of this member is set in the OPDE[3:0] bit field in the SPI mode enable setting register (SMENR).
uint32_t spide	Transfer data enable • Specifies whether or not data transfers occur in SPI operation mode. • Settable values: SPIBSC_OUTPUT_DISABLE: Transfers disabled. SPIBSC_OUTPUT_SPID_8: 8- (or 16-) bit transfers enabled. SPIBSC_OUTPUT_SPID_16: 16- (or 32-) bit transfers enabled. SPIBSC_OUTPUT_SPID_32: 32- (or 64-) bit transfers enabled. • The setting value of this member is set in the SPIDE[3:0] bit field in the SPI mode enable setting register (SMENR).
uint32_t sslkp	SPBSSL signal level hold • Specifies the SPBSSL signal state at transfer end in SPI operation mode. • Settable values: SPIBSC_SPISSL_NEGATE: Negate at transfer end. SPIBSC_SPISSL_KEEP: Maintain SPBSSL signal level from transfer end to start of next access. • The setting value of this member is set in the SSLKP bit in the SPI mode control register (SMCR).

Table 5.19 SPIBSC SPI Operating Mode Settings Structure (st_spibsc_spimd_reg_t) (3/5)

Member	Description
uint32_t spire	Data read enable • Specifies whether or not data reads occur in SPI operation mode. • Settable values: SPIBSC_SPIDATA_DISABLE: Data reads disabled. SPIBSC_SPIDATA_ENABLE: Data reads enabled. • The setting value of this member is set in the SPIRE bit in the SPI mode control register (SMCR).
uint32_t spiwe	Data write enable • Specifies whether or not data writes occur in SPI operation mode. • Settable values: SPIBSC_SPIDATA_DISABLE: Data writes disabled. SPIBSC_SPIDATA_ENABLE: Data writes enabled. • The setting value of this member is set in the SPIWE bit in the SPI mode control register (SMCR).
uint32_t dme	Dummy cycle enable • Specifies whether or not dummy cycles are inserted in SPI operation mode. • Settable values: SPIBSC_DUMMY_CYC_DISABLE: Insertion disabled. SPIBSC_DUMMY_CYC_ENABLE: Insertion enabled. • The setting value of this member is set in the DME bit field in the SPI mode enable setting register (SMENR).
uint32_t adder	Address DDR enable • Selects either SDR or DDR transfer when outputting addresses in SPI operation mode. • Settable value: SPIBSC_SDR_TRANS: SDR transfer • The setting value of this member is set in the ADDRE bit in the SPI mode DDR enable register (SMDRENR).
uint32_t opdre	Option data DDR enable • Selects either SDR or DDR transfer when outputting option data in SPI operation mode. • Settable value: SPIBSC_SDR_TRANS: SDR transfer • The setting value of this member is set in the OPDRE bit in the SPI mode DDR enable register (SMDRENR).
uint32_t spidre	Transfer data DDR enable • Selects either SDR or DDR transfer when transferring data in SPI operation mode. • Settable value: SPIBSC_SDR_TRANS: SDR transfer • The setting value of this member is set in the SPIDRE bit in the SPI mode DDR enable register (SMDRENR).

Table 5.20 SPIBSC SPI Operating Mode Settings Structure (st_spibsc_spimd_reg_t) (4/5)

Member	Description
uint8_t dmdb	Dummy cycle bit width • Specifies the bit width of dummy cycles in SPI operation mode. • Settable values: SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width • The setting value of this member is set in the DMDB[1:0] bit field in the SPI mode dummy cycle setting register (SMDMCR).
uint8_t dmcyc	Number of dummy cycles • Settable values: To be loaded with the results of calculating the value to be set in the DMCYC[2:0] bit field of the SPI mode dummy cycle setting register (SMDMCR) based on the value defined in SPIBSC_DMYCYC_SETTING. Values that will resultantly yield number of 0 to 7 are allowed. For details on the calculation, see "Table 6.2 List of Macros for Customizing the Sample Code (1/2)".
uint8_t cmd	Command • Specifies the command output in SPI operation mode. • The setting value of this member is set in the CMD[7:0] bit field in the SPI mode command setting register (SMCMR).
uint8_t ocmd	Optional command • Specifies the optional command output in SPI operation mode. • The setting value of this member is set in the OCMD[7:0] bit field in the SPI mode command setting register (SMCMR).
uint32_t addr	Address • Specifies the address output in SPI operation mode. • The setting value of this member is set in the ADR[31:0] bit field in the SPI mode address setting register (SMADR).
uint8_t opd[4]	Option data • Specifies the option data output in SPI operation mode. • The setting value of this member is set in the OPDn[7:0] bit field in the SPI mode option setting register (SMOPR). OPD3[7:0] <== opd[0] OPD2[7:0] <== opd[1] OPD1[7:0] <== opd[2] OPD0[7:0] <== opd[3]

Table 5.21 SPIBSC SPI Operating Mode Settings Structure (st_spibsc_spimd_reg_t) (5/5)

Member	Description
uint32_t smrdr[2]	Read data store buffer • The data read in SPI operation mode (SPI mode read data register n (SMRDRn)) is stored as follows: SMRDR0 ==> smrdr[0] SMRDR1 ==> smrdr[1]
uint32_t smwdr[2]	Write data store buffer • The data to be written in SPI operation mode (SPI mode write data register n (SMWDRn)) is stored as follows: SMWDR0 <== smwdr[0] SMWDR1 <== smwdr[1]

5.6 List of Variables for Loader program

Table 5.22 lists the Global Variables.

Table 5.22 Global Variables

Type	Variable Name	Contents
st_spibsc_cfg_t	g_spibsc_cfg	SPIBSC external address space read mode settings storage variable • Stores the register configuration information to be used in the SPIBSC external address space read mode.
st_spibsc_spimd_reg_t	g_spibsc_spimd_reg	SPIBSC SPI operation mode settings storage variable • Stores the SPIBSC settings used in SPI operation mode. In the sample code, these settings are also used as arguments when running serial flash control functions within API functions and user-defined functions.

5.7 List of Functions Used in the Loader program

The sample code comprises interface functions (API functions) for using peripheral functions, user-defined functions (functions called by API functions) which must be prepared by the user for the purpose of the target system, and sample functions which are necessary for the sample code to operate.

For the sample code of Loader program, Table 5.23, Table 5.24, and Table 5.25 list the Sample Functions, the API Functions, and the user-defined functions respectively. The function specification of the application program is basically the same as the function of the "RZ/A1H Group Example of Initialization". Refer to the application note for more information.

Table 5.23 Sample Functions

Function	Description
reset_handler	Reset handler
init_spibsc_init1_section	Loader program 1 deployment function Deploys the Loader program 1 from the ROM area (serial flash memory) to the large-capacity on-chip RAM so that it can run from the large-capacity on-chip RAM.
spibsc_init1	Loader program 1 execution function Makes settings (STEP1) to optimize the serial flash memory.
init_spibsc_init2_section	Loader program 2 deployment function Deploys the Loader program 2 from the ROM area (serial flash memory) to the large-capacity on-chip RAM so that it can run from the large-capacity on-chip RAM.
spibsc_init2	Loader program 2 execution function Makes settings (STEP2) to optimize the serial flash memory.

Table 5.24 API Functions

Function	Description
R_SFLASH_Exmode_Setting	SPIBSC initial setting function Makes initial settings necessary to control the serial flash memory with the SPIBSC and initial settings necessary to use the SPIBSC in external address space read mode. The function makes settings to the registers in the serial flash memory according to the initial settings in the SPIBSC-related registers. After the initial settings, the SPIBSC is set to external address space read mode.
R_SFLASH_WaitTend	Wait function for SPIBSC data transfer-end Waits until the data has been transferred from the SPIBSC.
R_SFLASH_Exmode	SPIBSC external address space read mode switching function Switches the SPIBSC from SPI operating mode to external address space read mode.
R_SFLASH_Set_Config	SPIBSC external address space read mode setting function Loads g_spibsc_cfg with the register configuration information necessary for using the SPIBSC in external address space read mode.
R_SFLASH_SpibscStop	SPIBSC stop function Negates the SSL to stop the access to the serial flash memory.
R_SFLASH_Exmode_Init	SPIBSC external address space read mode initial setting function Makes initial settings necessary for using the SPIBSC in external address space read mode. After the initial settings are made, the SPIBSC enters the external address space read mode.
R_SFLASH_Spibsc_Transfer	Serial flash control function Issues a command to the serial flash memory in the SPI operation mode according to the contents of the arguments

Table 5.25 User-Defined Functions

Function	Description
Userdef_SPIBSC_Set_Config	SPIBSC external address space read mode register information storage function This function must store the register configuration information to be set in the SPIBSC-related registers in SPIBSC external address space read mode according to the serial flash memory to be used into the area specified by the argument. This function in the sample code stores the register configuration information to be set in the SPIBSC-related registers assuming CYPRESS serial flash memory (S25FL512S).
Userdef_SFLASH_Set_Mode	Serial flash memory register setting function Implement the processing that sets to the serial flash memory registers that are necessary to use the SPIBSC in external address space read mode according to the serial flash memory to be used. The function in the sample code configures the registers for the CYPRESS serial flash memory (S25FL512S).
Userdef_SFLASH_Write_Enable	Serial flash memory write enable function Implement the processing that enables the serial flash memory registers for writes according to the serial flash memory to be used. In the sample code, this function issues a Write Status Register (WRSR) command to the CYPRESS serial flash memory (S25FL512S).
Userdef_SFLASH_Busy_Wait	Serial flash memory write completion wait function Implement the processing that waits for the write to the serial flash memory register to be completed while reading the required serial flash memory register according to the serial flash memory to be used. In the sample code, this function waits for the write to be completed by issuing a Read Status Register (RDSR) command to the CYPRESS serial flash memory (S25FL512S) and referencing the contents of the Status Register.

5.8 Loader program Functional Specifications

Specifications of the functions of the sample code are listed below.

reset_handler	
Outline	Loader program reset handler
Declaration	reset_handler
Description	The entry function of the Loader program.
Arguments	None
Return Value	None

init_spibsc_init1_section	
Outline	Loader program 1 deployment function
Declaration	void init_spibsc_init1_section (void);
Description	Deploys the Loader program 1 from the ROM area (serial flash memory) to the large-capacity on-chip RAM so that it can run from the large-capacity on-chip RAM.
Arguments	None
Return Value	None

spibsc_init1	
Outline	Loader program 1 execution function
Declaration	void spibsc_init1 (void);
Description	Makes settings (STEP1) to optimize the serial flash memory. • Specifies various SSL delay cycle counts. • Changes the SPBCLK operating frequency: Bϕ/8 → Bϕ/4 • Enables the SPIBSC read cache.
Arguments	None
Return Value	None

init_spibsc_init2_section	
Outline	Loader program 2 section initialization function
Declaration	void init_spibsc_init2_section (void);
Description	Transfers Loader program 2 to the large-capacity on-chip RAM.
Arguments	None
Return Value	None

spibsc_init2	
Outline	Loader program 2 execution function
Declaration	void spibsc_init2 (void);
Description	Makes settings (STEP2) to optimize the serial flash memory. • Changes the SPBCLK operating frequency: Bϕ/4 → Bϕ/2 • Changes the read command: H'03 → H'EC • Makes serial flash memory internal register settings. Configuration register: Sets QUAD bit to 1. Configuration register: Sets for LC[1:0]. (The setting value of LC[1:0] bit is different depending on the read command. See "Table 6.6 List of Numbers of Dummy Cycles Necessary for the Operating Frequency of the MX25L51245G".)
Arguments	None
Return Value	None

R_SFLASH_Exmode_Setting		
Outline	SPIBSC initial setting function	
Declaration	int32_t R_SFLASH_Exmode_Setting (uint32_t ch_no, uint32_t dual, st_spibsc_cfg_t *spibsccfg);	
Description	Makes initial settings necessary to control the serial flash memory with the SPIBSC and initial settings necessary to use the SPIBSC in external address space read mode. The function also configures the registers in the serial flash memory according to the initial values of the SPIBSC-related registers. The function must switch the SPIBSC mode from external address space read mode to SPI operation mode before configuring the serial flash memory registers and switch the SPIBSC mode from SPI operation mode to external address space read mode after configuring the serial flash memory registers. The SPIBSC external address space read mode initial setting function (R_SFLASH_Exmode_Init) is run from within this function.	
Arguments	uint32_t ch_no	SPIBSC channel number (only 0 is allowed.)
	uint32_t dual	Number of serial flash devices connected to the channel One device: SPIBSC_CMNCR_BSZ_SINGLE (0) Two devices: SPIBSC_CMNCR_BSZ_DUAL (1)
	st_spibsc_cfg_t *spibsccfg	SPIBSC external address space read mode settings Refer to Table 5.13 to Table 5.16 for setting details
Return Value	0: Normal end -1: Error	

R_SFLASH_WaitTend	
Outline	Wait function for SPIBSC data transfer-end
Declaration	void R_SFLASH_WaitTend(uint32_t ch_no);
Description	Waits until the data has been transferred from the SPIBSC.
Arguments	uint32_t ch_no SPIBSC channel number (only 0 is allowed.)
Return Value	None

R_SFLASH_Exmode	
Outline	SPIBSC external address space read mode switching function
Declaration	int32_t R_SFLASH_Exmode(uint32_t ch_no);
Description	Switches the SPIBSC from SPI operation mode to external address space read mode. All the entries in the read cache are cleared until the read access to the SPI multi I/O bus space after the switching to the external address space read mode.
Arguments	uint32_t ch_no SPIBSC channel number (only 0 is allowed.)
Return Value	0: Setting successful

R_SFLASH_Set_Config	
Outline	SPIBSC external address space read setting function
Declaration	void R_SFLASH_Set_Config(uint32_t ch_no, st_spibsc_cfg_t *spibsccfg);
Description	Loads g_spibsc_cfg with the register configuration information necessary for using the SPIBSC in external address space read mode according to the serial flash memory to be used. This function executes the user-defined function Userdef_SPIBSC_Set_Config (SPIBSC external address space read mode register information storage function) during its execution.
Arguments	uint32_t ch_no SPIBSC channel number (only 0 is allowed.) st_spibsc_cfg_t *spibsccfg SPIBSC external address space read mode settings Refer to Table 5.13 to Table 5.16 for setting details
Return Value	0: Normal end -1: Error

R_SFLASH_SpibscStop	
Outline	SPIBSC stop function
Declaration	int32_t R_SFLASH_SpibscStop(uint32_t ch_no);
Description	Negates the SSL to stop the access to the serial flash memory.
Arguments	uint32_t ch_no SPIBSC channel number (only 0 is allowed.)
Return Value	None

R_SFLASH_Exmode_Init		
Outline	SPIBSC external address space read mode initial setting function	
Declaration	int32_t R_SFLASH_Exmode_Init(uint32_t ch_no, uint32_t dual, st_spibsc_cfg_t *spibsccfg)	
Description	Makes initial settings necessary to use the SPIBSC in external address space read mode. After the initial settings, the SPIBSC is set to external address space read mode.	
Arguments	uint32_t ch_no	SPIBSC channel number (only 0 is allowed.)
	uint32_t dual	Number of serial flash devices connected to the channel One device: SPIBSC_CMNCR_BSZ_SINGLE (0) Two devices: SPIBSC_CMNCR_BSZ_DUAL (1)
	st_spibsc_cfg_t *spibsccfg	SPIBSC external address space read mode settings Refer to Table 5.13 to Table 5.16 for setting details
Return Value	0: Normal end -1: Error	

R_SFLASH_Spibsc_Transfer		
Outline	Serial flash control function	
Declaration	int32_t R_SFLASH_Spibsc_Transfer(uint32_t ch_no, st_spibsc_spimd_reg_t * regset);	
Description	Issues a command to the serial flash memory in SPI operation mode according to the contents of the argument regset.	
Arguments	uint32_t ch_no	SPIBSC channel number (only 0 is allowed.)
	st_spibsc_spimd_reg_t * regset	SPIBSC SPI operation mode Refer to Table 5.17 to Table 5.21 for setting details.
Return Value	0: Setting successful -1: Settings failure	
Remarks	When this function is called in external address space read mode, it switches into the SPI operation mode before issuing the command to the serial flash memory.	

Userdef_SPIBSC_Set_Config		
Outline	SPIBSC external address space read mode register information storage function	
Declaration	void Userdef_SPIBSC_Set_Config(uint32_t ch_no, st_spibsc_cfg_t *spibsccfg);	
Description	This function must store the register configuration information to be set in the SPIBSC-related registers in SPIBSC external address space read mode according to the serial flash memory to be used into the area specified by the argument spibsccfg. For the value to be specified in spibsccfg, see Table 5.13 to Table 5.16 and "6.3.1 Signal Output when a Read Command is Issued".	
Arguments	uint32_t ch_no	SPIBSC channel number (only 0 is allowed.)
	st_spibsc_cfg_t *spibsccfg	SPIBSC external address read settings Refer to Table 5.13 to Table 5.16 for setting details
Return Value	None	
Remarks	This function in the sample code stores the register configuration information to be set in the SPIBSC-related registers assuming CYPRESS serial flash memory (S25FL512S).	

Userdef_SFLASH_Set_Mode		
Outline	Serial flash memory register setting function	
Declaration	int32_t Userdef_SFLASH_Set_Mode(uint32_t ch_no, uint32_t dual, en_sf_req_t req, uint8_t data_width, uint8_t addr_mode);	
Description	Implement the processing that sets to the serial flash memory registers that are necessary to use the SPIBSC in external address space read mode according to the serial flash memory to be used.	
Arguments	uint32_t ch_no	SPIBSC channel number (only 0 is allowed.)
	uint32_t dual	Number of serial flash devices connected to the channel SPIBSC_CMNCR_BSZ_SINGLE: one device SPIBSC_CMNCR_BSZ_DUAL: two devices
	en_sf_req_t req	Register setting information SF_REQ_SERIALMODE: Specifies serial mode. SF_REQ_QUADMODE: Specifies quad mode.
	uint8_t data_width	Width of the bus for transferring data between SPIBSC and serial flash memory SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width
	uint8_t addr_mode	Bit width of the addresses to be issued to the serial flash memory Specify the bit width of the address to be output when issuing a read command. SPIBSC_OUTPUT_ADDR_24: 24-bit address output SPIBSC_OUTPUT_ADDR_32: 32-bit address output
Return Value	0: Setting successful -1: Settings failure	
Remarks	The sample code makes settings to the registers of CYPRESS serial flash memory (product No.: S25FL512S).	

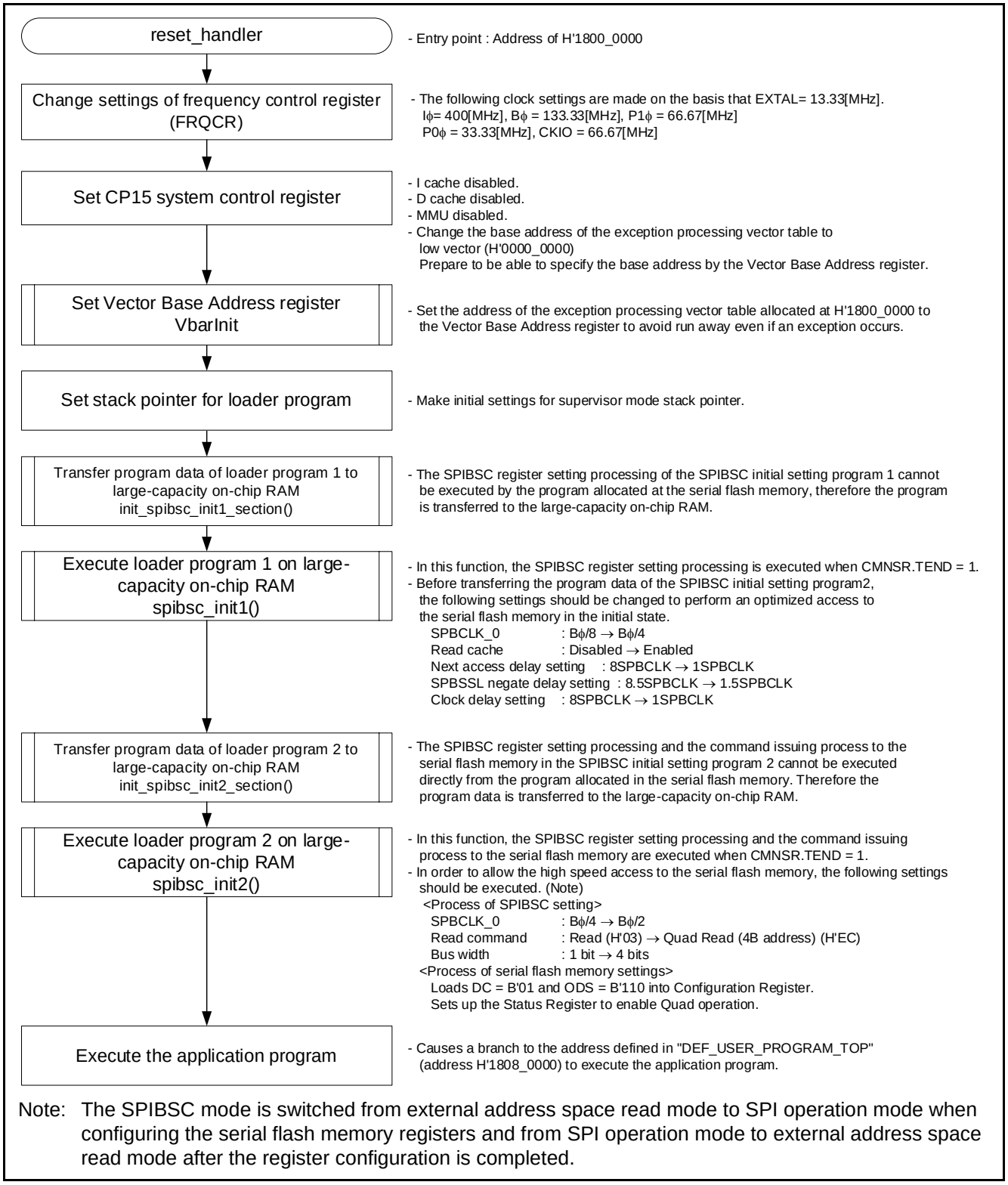
Userdef_SFLASH_Write_Enable	
Outline	Serial flash memory write enable function
Declaration	int32_t Userdef_SFLASH_Write_Enable(uint32_t ch_no);
Description	Implement the processing that enables the serial flash memory registers for writes according to the serial flash memory to be used.
Arguments	uint32_t ch_no SPIBSC channel number (only 0 is allowed.)
Return Value	0: Setting successful -1: Settings failure
Remarks	This function in the sample code issues the "Write Status Register (WRSR)" command to the CYPRESS serial flash memory (S25FL512S).

Userdef_SFLASH_Busy_Wait	
Outline	Serial flash memory write completion wait function
Declaration	int32_t Userdef_SFLASH_Busy_Wait(uint32_t ch_no, uint32_t dual, uint8_t data_width);
Description	Implement the processing that waits for the write to the serial flash memory to be completed while reading the required serial flash memory register according to the serial flash memory to be used.
Arguments	uint32_t ch_no SPIBSC channel number (only 0 is allowed.) uint32_t dual Number of serial flash devices connected to the channel SPIBSC_CMNCR_BSZ_SINGLE: one device SPIBSC_CMNCR_BSZ_DUAL: two devices uint8_t data_width Width of the bus for transferring data between SPIBSC and serial flash memory SPIBSC_1BIT: 1-bit width SPIBSC_4BIT: 4-bit width
Return Value	None
Remarks	This function in the sample code waits write processing to be completed while referencing the contents of the Status Register by issuing "Read Status Register (RDSR)" command to the CYPRESS serial flash memory (S25FL512S).

5.9 Loader program Flowcharts

5.9.1 Loader program (Overall)

Figure 5.4 shows the Flowchart of Loader program (Overall).



Note: The SPIBSC mode is switched from external address space read mode to SPI operation mode when configuring the serial flash memory registers and from SPI operation mode to external address space read mode after the register configuration is completed.

Figure 5.4 Flowchart of Loader program (Overall)

5.9.2 Loader program 1 (STEP1)

In the Loader program 1, the read cache is enabled after changing the delay cycle count to be inserted during the communication and setting the SPI clock frequency to Bϕ/4.

The process executed in the Loader program 1 cannot be performed by the program allocated to the SPI multi-I/O bus space due to the change in SPIBSC register settings, therefore this program should be transferred to the large-capacity on-chip RAM to be executed.

Figure 5.5 shows the Flowchart of Loader program 1.

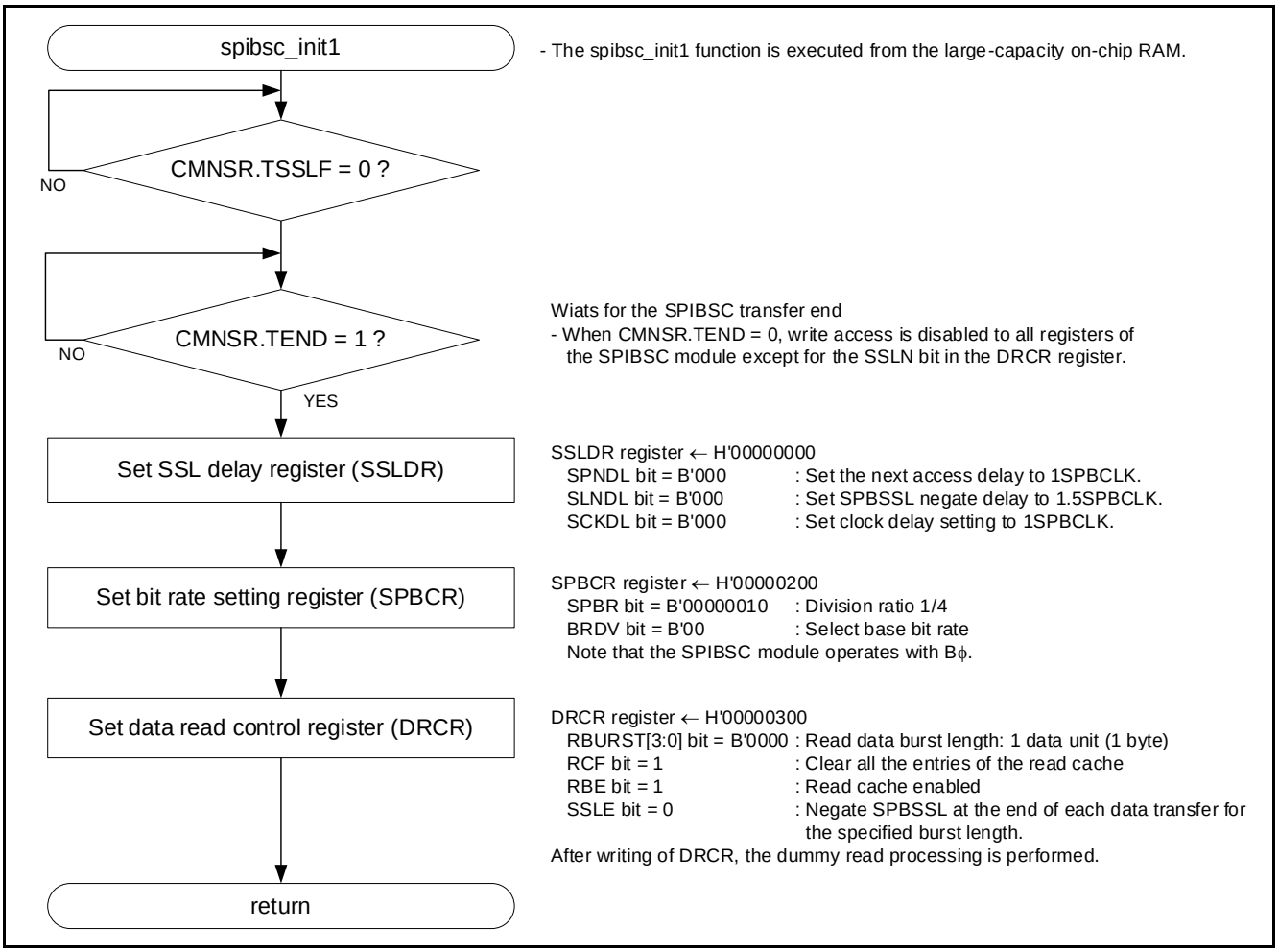


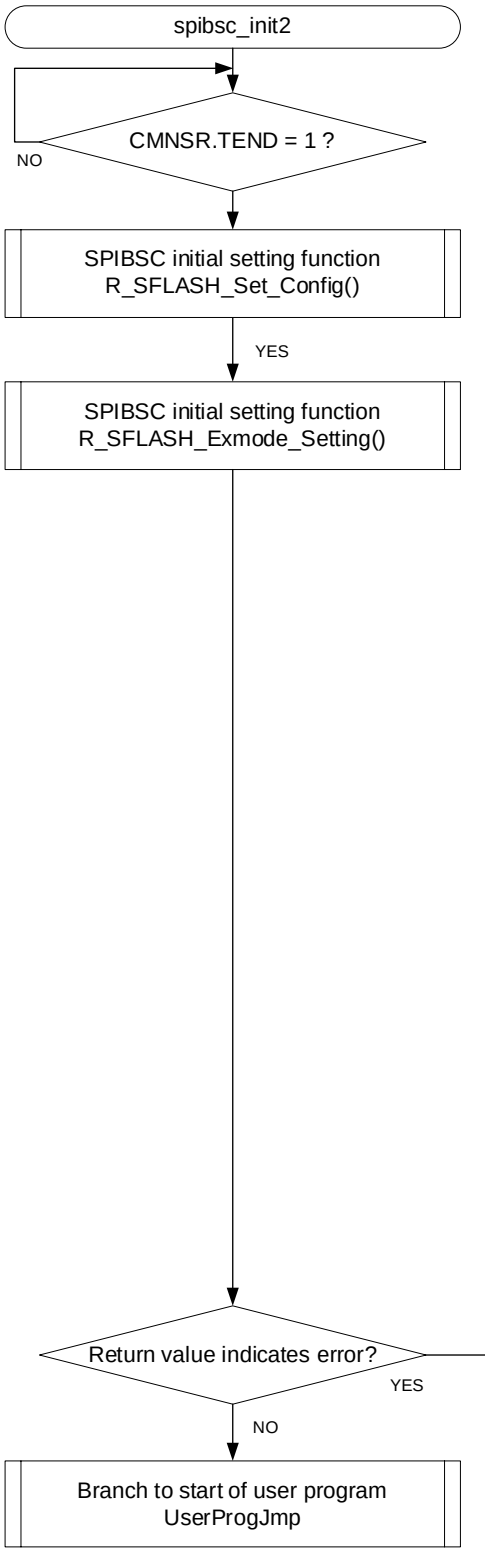
Figure 5.5 Flowchart of Loader program 1

5.9.3 Loader program 2 (STEP2)

The Loader program 2 sets up the serial flash memory registers (Configuration Register, QE bit and LC[1:0]) and changes the bus bit width from 1 bit to 4 bits so that the serial flash memory can be accessed at a higher speed. After setting up the serial flash memory registers, the program sets the type of read command to be issued to the serial flash memory when using the SPIBSC in external address space read mode to "Quad Read (4B address)" (H'EC) and changes the SPI clock frequency to Bϕ/2.

Since the Loader program 2 modifies the SPIBSC registers during its processing, it cannot run in the SPI multi-I/O bus space. Accordingly, it is expanded in large-capacity on-chip RAM for execution.

Figure 5.6 shows the Flowchart of Loader program 2.



- The spibsc_init2 function is executed on the large-capacity on-chip RAM.

- When wait for end of SPIBSC transfer (R_SFLASH_WaitTend function call)
  If CMNSR.TEND = 0, write accesses to all registers of the SPIBSC module are disabled except the SSLN bit of the DRCR register.

- In order to the high speed access to the serial flash memory, the following setting should be executed for the SPIBSC register and the serial flash memory.

<Process of SPIBSC register settings>
SPBCR register
  SPBR bit = B'00000001 : Sets a division ratio to 1/2
  BRDV bit = B'00 : Select the base bit rate.
  (SPBCLK_0 = 66.67MHz (Bϕ/2))
CMNCR register
  CPHAT bit = 0 : Data transmission at even edge during SDR transfer.
  CPHAR bit = 1 : Data reception at even edge during SDR transfer.
  CPOL bit = 0 : SPBCLK_0 output is 0 during SPBSSL negation period
  SSLP bit= 0 : The SPBSSL signal is low active.
  BSZ bit = B'00 : Number of serial flash memory devices to be connected is 1.

<Process of serial flash memory settings (Issue a Write Status command) >
Sets up the registers using the user-defined function Userdef_SFLASH_Set_Mode.
Configuration Register
  QUAD bit = 1 : Enables Quad operation.
  LC[1:0] bit = B'00 : Sets to the dummy cycle number (shortest) of Quad Read (4-byte address) command (H'EC)

<Process of SPIBSC register settings>
DRCR register
  RBURST[3:0] bit = B'0011 : Read data burst length: 4 data units (32 bytes)
 DRENR register
  ADB bit = B'10 : Address bit size: 4 bits
  OPDB bit = B'10 : Option data bit size: 4 bits
  DRDB bit = B'10 : Data read bit size: 4 bits
  DME bit = 1 : Dummy cycle insertion enabled
  ADE bit = B'1111 : Output 32 address bits.
  OPDE bit = B'1000 : Output the option data OPD3.
DRCMR register
  CMD bit = H'EC : Sets Quad Read (4-bytes address) command (H'EC).
CKDLY register ← H'0000A504
  CKDLY bit = B'0100 : Initial value
SPODLY register ← H'A5000000
  SPODLY bit = H'0000 : Initial value
DREAR register ← H'00000001
  EAV bit = B'00000000
  EAC bit = B'001 : External address bits [25:0] enabled.
  SPIBSC channel 0 space can be directly accessed the full space of 64MB.
DRDRENR register
  ADDRE bit = 0 : SDR transfer mode for addresses
  OPDRE bit = 0 : SDR transfer mode for option data
  DRDRE bit = 0 : SDR transfer mode for data reads

- Branch to the address defined in the macro
  "DEF_USER_PROGRAM_TOP" (address H'1808_0000)

Note: The Configuration Register are configured to change the serial flash memory access width from 1 bit to 4 bits. To make these register settings, it is necessary to switch the SPIBSC from external address space read mode to SPI operation mode before issuing WRSR command to the serial flash memory. After the registers are configured with the WRSR commands, the SPIBSC is switched from SPI operation mode to external address space read mode.

Figure 5.6 Flowchart of Loader program 2

6. Application Example

6.1 Operation of the Sample Code Used in its Initial State

The sample code in its initial state accesses the CYPRESS serial flash memory (product No.: S25FL512S) according to the settings that are summarized in Table 6.1.

Table 6.1 Sample Code Access Settings

Item	Setting
Serial flash memory	• CYPRESS serial flash memory • Model name: S25FL512S • Read command used: H'EC 4-bit bus, SDR transfer Number of dummy cycles required: 4 (When running at a maximum operating frequency of 80 MHz.)
Number of serial flash memory devices to be connected to the SPI multi-I/O bus space	1
Data bus width	4 bits
Number of address bytes (Number of bytes to be issued when specifying an address)	4 bytes
Transfer format in read mode	SDR transfer mode
Output level of the SPBCLK pin in idle state	SPBCLK is set to 0 when SPBSSL is negated. (CMNCR.CPOL = 0)
SPIBSC data transmission timing	Data transmission at even edge during SDR transfer. (CMNCR.CPHAT = 0)
SPIBSC data reception timing	Data reception at even edge during SDR transfer. (CMNCR.CPHAR = 1)

Figure 6.1 shows the Read Operation in SDR Transfer Mode (Initial State of the Sample Code). In a read operation, the command, address, and dummy cycles are output from the SPIBSC before a data cycle begins and read data is output from the S25FL512S.

The S25FL512S samples the input data at the rising edge of the clock in SDR transfer mode. The SPIBSC begins data output with respect to the falling edge of the clock and continues data output processing at the rising edge. CPOL=0 and CPHAT=0 are set so that the S25FL512S can sample the MSB of the output data from the SPIBSC at the rising edge of the first SPBCLK.

The S25FL512S also begins data output with respect to the falling edge of the clock in SDR transfer mode. To keep the output data present, in the data cycle, till the falling edge of the next clock with respect to the falling edge of the last clock of the dummy cycle, CPHAR=1 is set in the SPIBSC so that the input data can be sampled at an even edge (falling edge when CPOL=0).

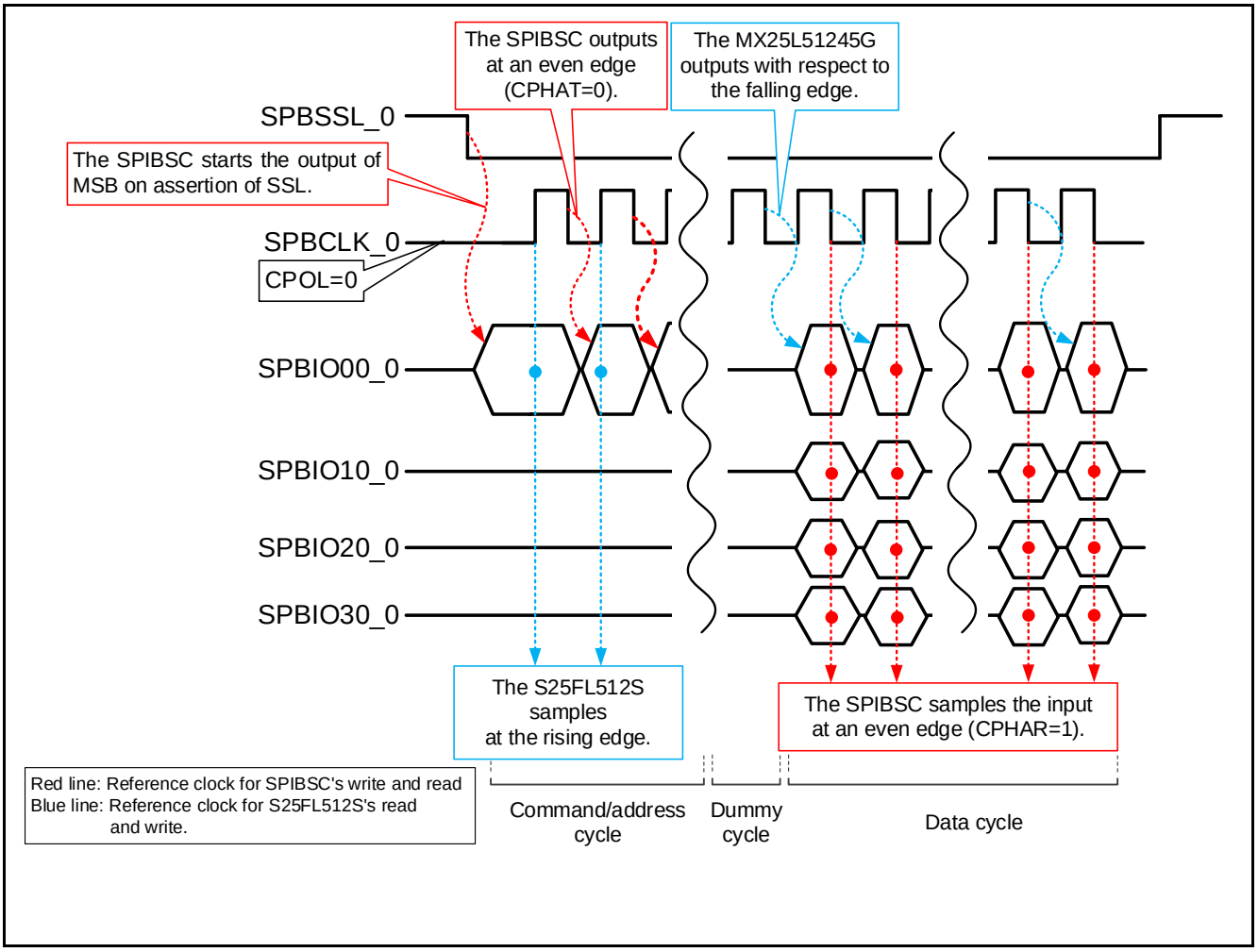


Figure 6.1 Read Operation in SDR Transfer Mode (Initial State of the Sample Code)

6.2 Changing the Sample Code without Changing the Serial Flash Memory

The sample code can be subject to customization by changing the macro definitions (defined in spibsc_config.h) which are shown in Table 6.2 and Table 6.3. " " in the tables denotes a setting for using the sample code in its initial state.

The user -defined function Userdef_SFLASH_Set_Mode is called to set up the serial flash memory registers within the R_SFLASH_Exmode_Setting function which is executed during initialization. The Userdef_SFLASH_Set_Mode function configures the Configuration Register of the S25FL512S, shown in Table 6.2 and Table 6.3, so that their values match the values of the SPIBSC macros which are defined according to the specifications for the read command to be used. This enables read accesses to the S25FL512S shown in Table 6.4 and Table 6.5.

Table 6.2 List of Macros for Customizing the Sample Code (1/2)

Macro Name	Value	Setting
DEF_SPIBSC_DUAL_MODE	SPIBSC_CMNCR_BSZ_SINGLE: 1	CMNCR.BSZ[1:0] = B'00
	SPIBSC_CMNCR_BSZ_DUAL: 2	CMNCR.BSZ[1:0] = B'01
SPIBSC_BUS_WITDH (Note 2)	SPIBSC_1BIT: 1 bit	DRENR.ADB[1:0] = B'00 DRENR.OPDB[1:0] = B'00 DRENR.DRDB[1:0] = B'00
	SPIBSC_4BIT: 4 bits	DRENR.ADB[1:0] = B'10 DRENR.OPDB[1:0] = B'10 DRENR.DRDB[1:0] = B'10
SPIBSC_OUTPUT_ADDR (Note 2)	SPIBSC_OUTPUT_ADDR_24: 3 bytes	DRENR.ADE[3:0] = B'0111
	SPIBSC_OUTPUT_ADDR_32: 4 bytes	DRENR.ADE[3:0] = B'1111
SPIBSC_TRANS_MODE (Note 2)	SPIBSC_SDR_TRANS: SDR transfer	DRDRENR.ADDRE = 0 DRDRENR.OPDRE = 0 DRDRENR.DRDRE = 0
	SPIBSC_DDR_TRANS: DDR transfer	DRDRENR.ADDRE = 1 DRDRENR.OPDRE = 1 DRDRENR.DRDRE = 1
SPIBSC_READCMD_SETTING (Note 1) (Note 2)	Serial flash memory read command to be specified in DRCMR.CMD ("0xEC" in initial state)	DRCMR.CMD[7:0] = H'EC
SPIBSC_DMYCYC_SETTING (Note 2)	Number of dummy cycles (4 in initial state)	DRDMCR.DMCYC[2:0] = B'001 (Note 3)

Notes: 1. The sample code supports the commands H'0B (FAST READ), H'0C (FAST READ4B), H'EB (4READ), H'EC (4READ4B) as the commands that can be issued to the S25FL512S.
2. SPIBSC_DMYCYC_SETTING, SPIBSC_BUS_WITDH, SPIBSC_TRANS_MODE, and SPIBSC_OUTPUT_ADDR need be configured according to the contents of the read command to be defined in SPIBSC_READCMD_SETTING.
3. The value to be specified in RDMCR.DMYCYC[2:0] is dependent on the read command to be used and must be calculated from the number of dummy cycles and "High Performance Read Mode indicator" that are necessary for theS25FL512S. In the sample code, the number of dummy cycles is defined in SPIBSC_DMYCYC_SETTING.

Table 6.3 List of Macros for Customizing the Sample Code (2/2)

Macro Name	Value	Setting
SPIBSC_OUTPUT_OPTION_SETTING	Specifies the option data to be output from among OPD3, OPD3/OPD2, OPD3/OPD2/OPD1, OPD3/OPD2/OPD1/OPD0, and "output disabled".	
	SPIBSC_OUTPUT_DISABLE	DRENR.OPDE[3:0] = B'0000
	SPIBSC_OUTPUT_OPD_3	DRENR.OPDE[3:0] = B'1000
	SPIBSC_OUTPUT_OPD_32	DRENR.OPDE[3:0] = B'1100
	SPIBSC_OUTPUT_OPD_321	DRENR.OPDE[3:0] = B'1110
	SPIBSC_OUTPUT_OPD_3210	DRENR.OPDE[3:0] = B'1111
SPIBSC_OUTPUT_OPTION_DATA_OPD3	0x00	DROPR.OPD3[7:0] = H'00
SPIBSC_OUTPUT_OPTION_DATA_OPD2	0x00	DROPR.OPD2[7:0] = H'00
SPIBSC_OUTPUT_OPTION_DATA_OPD1	0x00	DROPR.OPD1[7:0] = H'00
SPIBSC_OUTPUT_OPTION_DATA_OPD0	0x00	DROPR.OPD0[7:0] = H'00
SPIBSC_CPOL_SETTING	SPIBSC_CMNCR_CPOL_LOW: SPBCLK is set to 0 when SPBSSL is negated.	CMNCR.CPOL = 0
	SPIBSC_CMNCR_CPOL_HIGH: SPBSSL is set to 1 when SPBCLK is negated.	CMNCR.CPOL = 1
SPIBSC_CPHAT_SETTING	SPIBSC_CMNCR_CPHAT_EVEN: Data transmission at even edge during SDR transfer. Data transmission starts at even edge during DDR transfer.	CMNCR.CPHAT = 0
	SPIBSC_CMNCR_CPHAT_ODD: Data transmission at odd edge during SDR transfer. Data transmission starts at odd edge during DDR transfer.	CMNCR.CPHAT = 1
SPIBSC_CPHAR_SETTING	SPIBSC_CMNCR_CPHAR_ODD: Data reception at odd edge during SDR transfer. Data reception starts at odd edge during DDR transfer.	CMNCR.CPHAR = 0
	SPIBSC_CMNCR_CPHAR_EVEN: Data reception at even edge during SDR transfer. Data reception starts at even edge during DDR transfer	CMNCR.CPHAR = 1
SPIBSC_CKDLY_SETTING	SPIBSC_CKDLY_DEFAULT: Initial value	CKDLY.CKDLY[3:0] = B'0100
	SPIBSC_CKDLY_TUNING: Makes the data input setup time shorter and the data input hold time longer.	CKDLY.CKDLY[3:0] = B'1000
SPIBSC_SPODLY_SETTING	SPIBSC_SPODLY_DEFAULT: Initial value	SPODLY.SPODLY[15:0] = H'0000
	SPIBSC_SPODLY_TUNING: The delay, hold, buffer on and buffer off times for data output are lengthened.	SPODLY.SPODLY[15:0] = H'6363

Table 6.4 shows S25FL512S Status Register and Table 6.5 shows S25FL512S Configuration Register. The Userdef_SFLASH_Set_Mode function is used to set the bits denoted by "　　" in the tables and the other bits are left to hold their initial value.

Table 6.4 S25FL512S Status Register

Bit Position	Bit Name	Attribute (Note)	Description
7	SRWD	NV	Status register write protect 1 = Status register write disabled 0 = Status register write enabled
6	P_ERR	V, Read only	1 = Error occurred 0 = No Error
5	E_ERR	V Read only	1 = Error occurred 0 = No Error
4,3,2	BP2,BP1,BP0	NV	Level of protected block
1	WEL	V	Write enable latch 1 = Write enable 0 = Not write enable
0	WIP	V	Write in progress bit 1 = Write operation 0 = Not in write operation

Note: "NV" in the attribute column denotes "Non-volatile bit" and "V" denotes "Volatile bit."

Table 6.5 S25FL512S Configuration Register

Bit Position	Bit Name	Attribute (Note 1)	Description
7,6	LC1, LC0	NV	Dummy cycle 1, Dummy cycle 0 LC[1:0] = B'00（注 2）
5	TBPROT	OTP	0 = BP starts at bottom (Low address) 1 = BP starts at top(High address)
4	RFU		Reserved Futue Use
3	BPNV	OTP	1 = Volatile 0 = Nonvolatile
2	RFU		Reserved Future use
1	QUAD	NV	1 = QUAD 0 = Dual or Serial
0	FREEZE	V	1 = block Protection and OTP locked 0 = block Protection and OTP un-locked

Notes: 1. "NV" in the attribute column denotes "Non-volatile bit" and "OTP" denotes "One-time programmable bit".

2. As seen from Table 6.6 the number of dummy cycles differs depending on the read command to be used. The sample code uses H'EC as the read command and therefore loads LC[1:0] bit with a value of B'00 so that an optimum dummy cycle count can be obtained.

Table 6.6 shows a List of Numbers of Dummy Cycles Necessary for the Operating Frequency of the MX25L51245G. The number of necessary dummy cycles differ depending on the read command and operating frequency to be used. Since the sample code uses the H'EC command as the read command and sets SPBCLK to 66.66 MHz, its optimum setting is LC[1:0] = B'00 which yields a dummy cycle count of 4 cycles.

When changing the read command to be used, resets the number of dummy cycles according to the new read command and the frequency of SPBCLK.

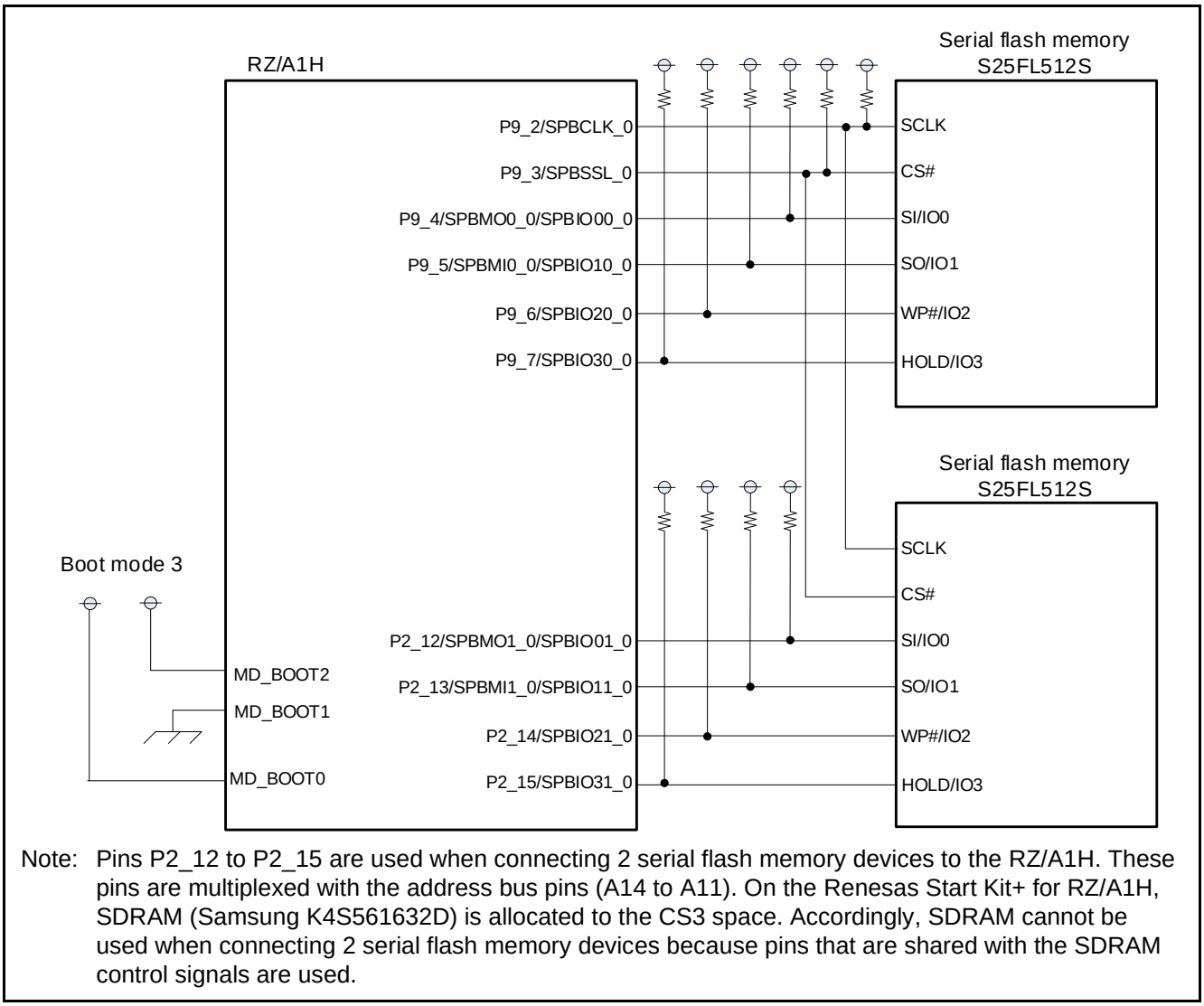
Table 6.6 List of Numbers of Dummy Cycles Necessary for the Operating Frequency of the MX25L51245G

Read Command	C[1:0]			
	B'00	B'01	B'10	B'11
FAST READ（H'0B） FAST READ4B（H'0C）	8cycles /80MHz	8cycles /90MHz	8cycles /104MHz	0cycles /50MHz
4READ（H'EB） 4READ4B（H'EC）	4cycles /80MHz	4cycels /90MHz	5cycels /104MHz	1cycels /50MHz

6.2.1 Changing to 2-Serial-Flash-Memory-Device Configuration (8-bit Access Mode)

Serial flash memory can be made accessible in 8-bit width by using 2 serial flash memory devices when connecting to the SPI multi-I/O bus space.

Figure 6.2 shows an Example of Connecting 2 Serial Flash Memory Devices (8-bit Access Width).



Note: Pins P2_12 to P2_15 are used when connecting 2 serial flash memory devices to the RZ/A1H. These pins are multiplexed with the address bus pins (A14 to A11). On the Renesas Start Kit+ for RZ/A1H, SDRAM (Samsung K4S561632D) is allocated to the CS3 space. Accordingly, SDRAM cannot be used when connecting 2 serial flash memory devices because pins that are shared with the SDRAM control signals are used.

Figure 6.2 Example of Connecting 2 Serial Flash Memory Devices (8-bit Access Width)

To connect 2 flash memory devices, it is necessary to configure the pins P2_12 to P2_15 for pin functions SPBMO1_0/SPBIO01_0, SPBMI1_0/SPBIO11_0, SPBIO21_0, and SPBIO31_0, respectively. In the sample code, the function "io_spibsc_port_setting" is used to configure the multiplexed pin functions.

When changing the configuration for connecting to the SPI multi-I/O bus space to 2-serial-flash-memory-device configuration, change the macro definitions listed in Table 6.7 from the initial values defined for the sample code.

Table 6.7 List of Macros for Customizing the Sample Coe (2-Device Configuration)

Macro Name	Value	Setting
DEF_SPIBSC_DUAL_MODE	SPIBSC_CMNCR_BSZ_DUAL: 2	CMNCR.BSZ[1:0] = B'01

The addresses that are allocated to the SPI multi-I/O bus space for the serial flash memory differs for the configuration in which one serial flash memory device is connected and the configuration in which two serial flash memory devices are connected. After the boot startup on-chip ROM program is executed, however, the SPIBSC causes a branch to address H'1800_0000 (address H'0000_0000 of the serial flash memory) when the number of address bytes is 3 bytes (16MB area maximum) and one serial flash memory device is connected. The Loader program need be allocated within the 16MB area starting at address H'0000_0000 of the serial flash memory which is connected to SPBMO0_0/SPBIO00_0 and SPBMI0_0/SPBIO10_0. In the sample code, it is allocated to sectors 0 to 7 (addresses H'0000_0000 to H'0000_7FFF) of the serial flash memory.

Since the application program starts execution after the Loader program set the number of address bytes to 4 and the number of serial flash memory devices connected to 2, it can be allocated to an area that is available in the 2-serial-flash-memory configuration.

Figure 6.3 shows an Example of Allocating Programs in 2-Serial-Flash-Memory Configuration.

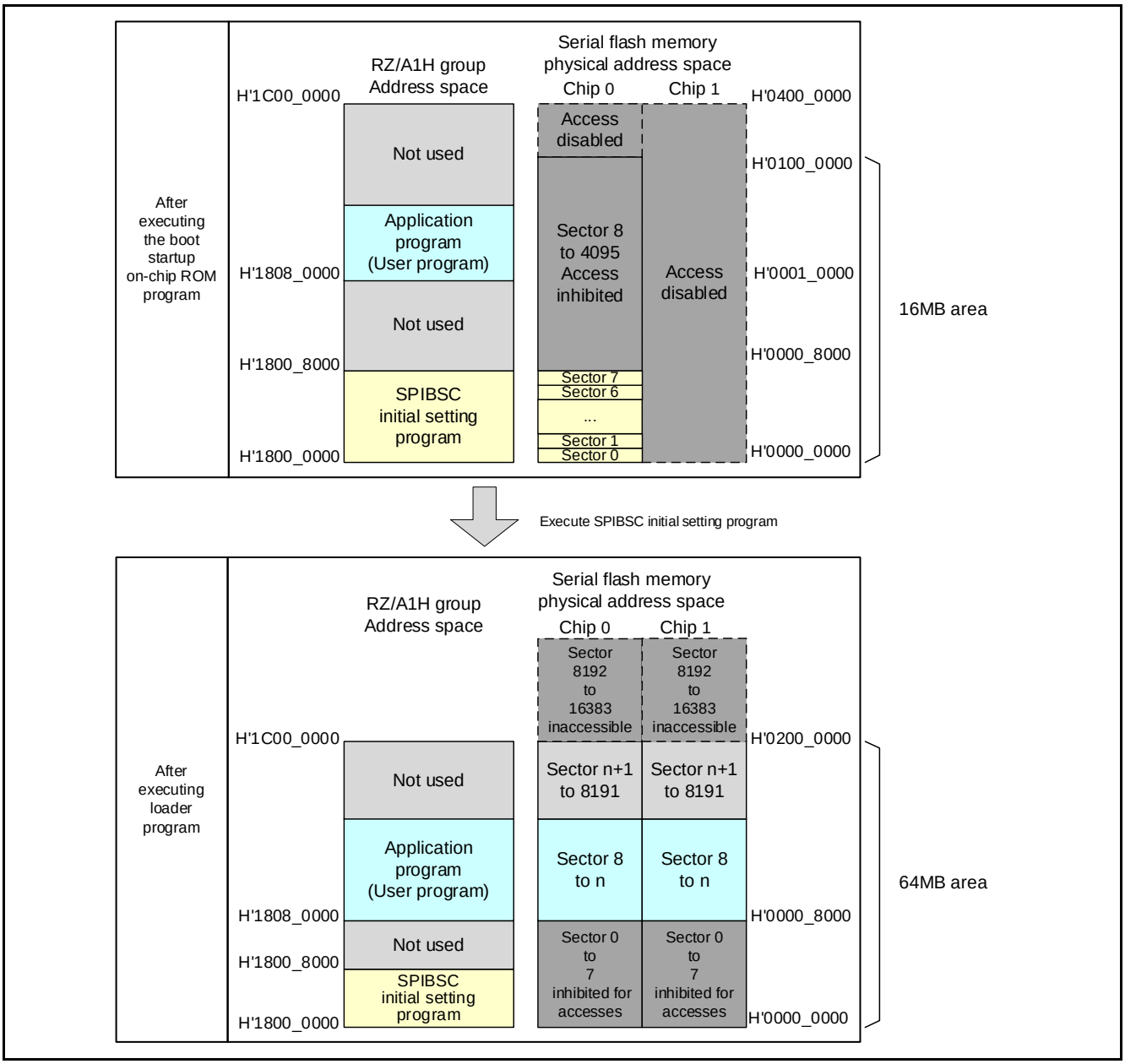


Figure 6.3 Example of Allocating Programs in 2-Serial-Flash-Memory Configuration

6.3 Changing the Sample Code When Changing the Serial Flash Memory

When changing the serial flash memory, the sample code should be changed according to the specifications of the serial flash memory to be used.

Table 6.8 lists the Points for Changing Sample Code.

Table 6.8 Points for Changing Sample Code

Change Point	Description
Signal Output when a Read Command is Issued	Change the output signal to be sent to the serial flash memory when a read command is issued in external address space read mode according to the specifications for the serial flash memory read command to be used.
Setting up the Serial Flash Memory Registers	The registers in the serial flash memory required to use the SPIBSC in external address read mode are made according to the serial flash memory to be used.
Serial flash memory write enable	The settings for the registers in the serial flash memory are made to enable write operations according to the serial flash memory to be used. (Note)
Serial Flash Memory Write	The registers in the serial flash memory are read according to the serial flash memory to be used. Read the registers in the flash memory and wait for the write to the serial flash memory to be completed.

Note: In some cases, it is necessary to enable write operations to the serial flash memory in order to make settings to the registers in the serial flash memory.

Processing summarized in Table 6.2 is executed by the Loader program 2 (spibsc_init2 function). It can be handled by changing the processing of the user-defined function in the sample code according to the serial flash memory to be used. Figure 6.4 shows the Hierarchical Module Diagram of the Loader program 2. Subsections 6.3.1 to 6.3.4 show the outline of the processing executed by the sample program.

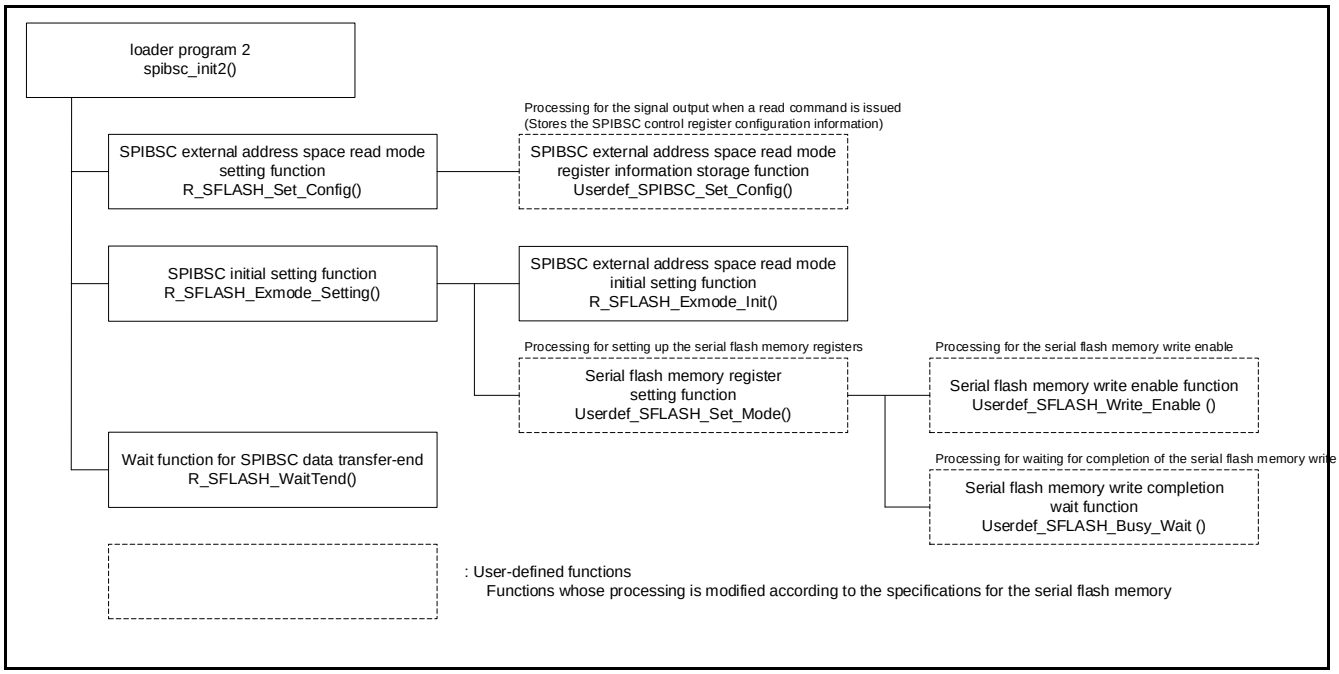


Figure 6.4 Hierarchical Module Diagram of the Loader program 2

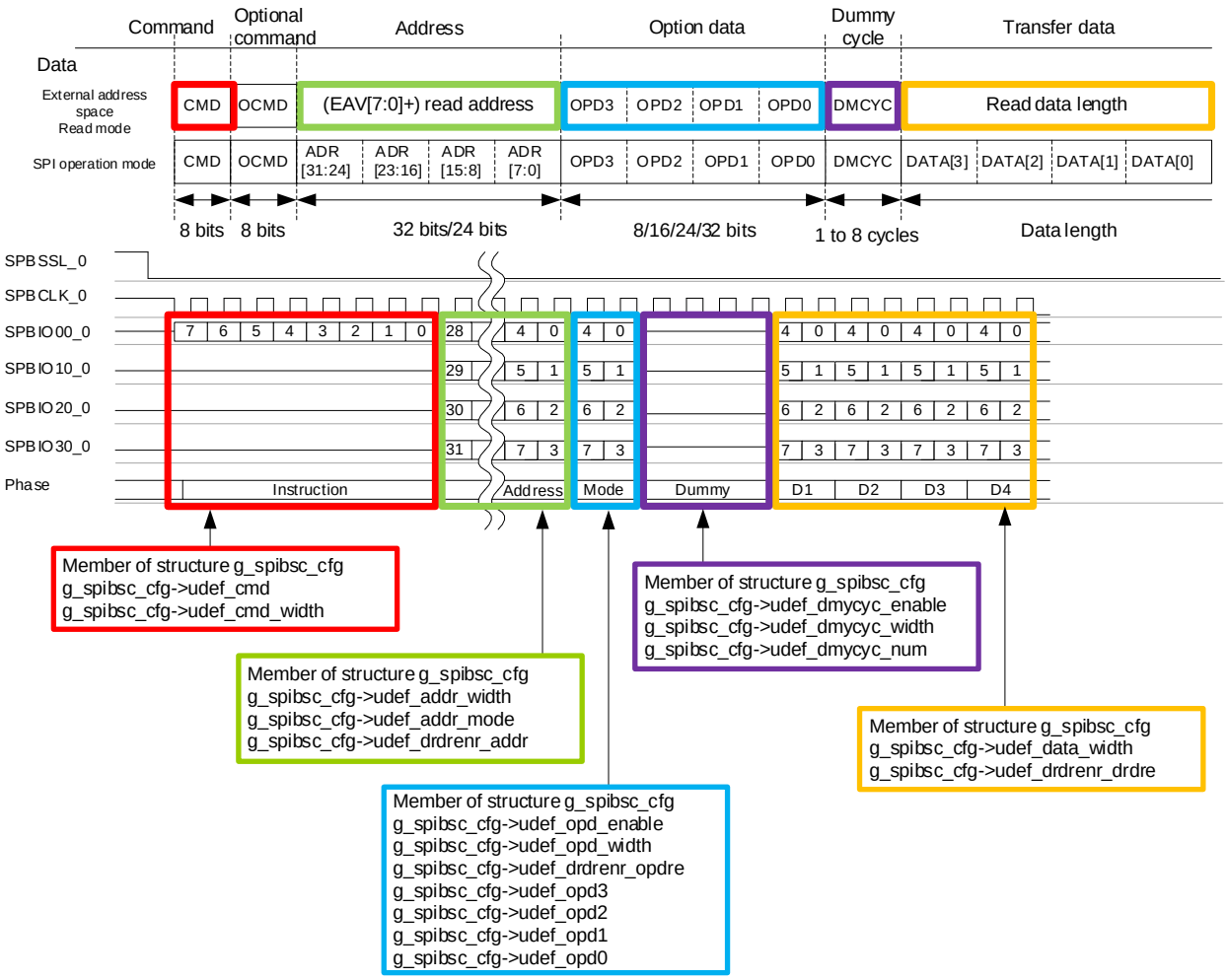
6.3.1 Signal Output when a Read Command is Issued

In external address space read mode, a read access to the SPI multi-I/O bus space is initiated by sending a signal, which is converted for SPI communication, to the serial flash memory when issuing the read command. When changing the serial flash memory to be used, change the signal to be output when issuing a read command according to the specifications for the serial flash memory read command.

The SPIBSC allows the signal to be output to the serial flash memory by setting up the SPIBSC control register in the external address space read mode.

In the sample code, the value to be specified in the SPIBSC control register can be changed through a global variable (variable for storing the SPIBSC external address space read mode settings: g_spibsc_cfg). g_spibsc_cfg can be configured using the user-defined function (the SPIBSC external address space read mode register information storage function: Userdef_SPIBSC_Set_Config) which is the executed by the SPIBSC external address space read mode setting function (R_SFLASH_Set_Config). Modify the implementation of the Userdef_SPIBSC_Set_Config function according to the contents of Table 5.13 to Table 5.21 and the specifications for the serial flash memory to be used.

Figure 6.5 shows the Correspondence between SPIBSC Control Register Settings and Waveforms Output to Serial Flash Memory during SPIBSC External Address Read Operation. Refer to these example settings when determining the g_spibsc_cfg settings that match the read command of the serial flash memory used.



The sample code defines the members of the structures as follows when the CYPRESS's serial flash memory (S25FL512S) is used:

g_spibsc_cfg->udef_cmd	:SPIBSC_READCMD_SETTING (H'EC)
g_spibsc_cfg->udef_cmd_width	:SPIBSC_1BIT
g_spibsc_cfg->udef_addr_width	:SPIBSC_BUS_WITDH (SPIBSC_4BIT)
g_spibsc_cfg->udef_addr_mode	:SPIBSC_OUTPUT_ADDR (SPIBSC_OUTPUT_ADDR_32)
g_spibsc_cfg->udef_drdrenr_addr	:SPIBSC_TRANS_MODE (SPIBSC_SDR_TRANS)
g_spibsc_cfg->udef_opd_enable	:SPIBSC_OUTPUT_OPD_3
g_spibsc_cfg->udef_opd_width	:SPIBSC_BUS_WITDH (SPIBSC_4BIT)
g_spibsc_cfg->udef_drdrenr_opdre	:SPIBSC_TRANS_MODE (SPIBSC_SDR_TRANS)
g_spibsc_cfg->udef_opd3	:SPIBSC_OUTPUT_OPTION_DATA_OPD3 (H'00)
g_spibsc_cfg->udef_opd2	:SPIBSC_OUTPUT_OPTION_DATA_OPD2 (OPD2 is not output.)
g_spibsc_cfg->udef_opd1	:SPIBSC_OUTPUT_OPTION_DATA_OPD1 (OPD1 is not output.)
g_spibsc_cfg->udef_opd0	:SPIBSC_OUTPUT_OPTION_DATA_OPD0 (OPD 0 is not output.)
g_spibsc_cfg->udef_dmycyc_enable	:SPIBSC_DUMMY_CYC_ENABLE
g_spibsc_cfg->udef_dmycyc_width	:SPIBSC_1BIT
g_spibsc_cfg->udef_dmycyc_num	:Calculated from the value defined in SPIBSC_DMYCYC_SETTING.
g_spibsc_cfg->udef_data_width	:SPIBSC_BUS_WITDH (SPIBSC_4BIT)
g_spibsc_cfg->udef_drdrenr_drdre	:SPIBSC_TRANS_MODE (SPIBSC_SDR_TRANS)

Note: Mode cycle in the figure for the refers to the cyclel of the High Performance Read Mode Indicator for the S25FL512S.

Figure 6.5 Correspondence between SPIBSC Control Register Settings and Waveforms Output to Serial Flash Memory during SPIBSC External Address Read Operation

6.3.2 Setting up the Serial Flash Memory Registers

In the sample code, the user-defined function Userdef_SFLASH_Set_Mode is called by the R_SFLASH_Exmode_Setting function which is executed during the initialization processing in order to set up the serial flash memory.

Set up the registers in the serial flash memory according to the contents of the global variable g_spibsc_cfg which is described in "6.3.1 Signal Output when a Read Command is Issued".

The write_status function which is called from the Userdef_SFLASH_Set_Mode function calls Userdef_SFLASH_Write_Enable to issue a Write Enable command so that the serial flash memory can be enabled for writes before writing data into Status Register and Configuration Register. Subsequently the Userdef_SFLASH_Busy_Wait function is called to verify that the memory is enabled for writes. Implement the Userdef_SFLASH_Set_Mode function according to the specifications for the serial flash memory to be used.

Figure 6.6 shows the Userdef_SFLASH_Set_Mode Function Processing Flow of the sample code.

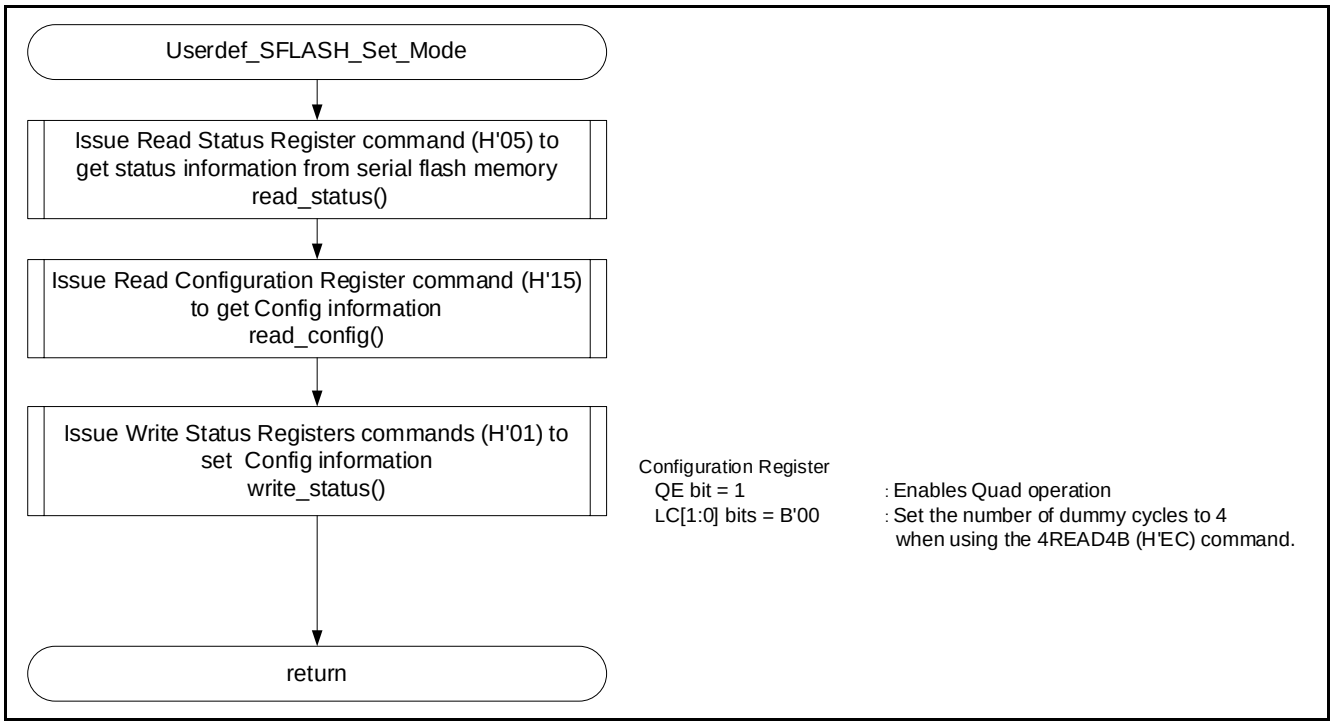


Figure 6.6 Userdef_SFLASH_Set_Mode Function Processing Flow

6.3.3 Serial Flash Memory Write Enable

It is necessary to enable the serial flash memory for writes before writing data to the registers (Status Register and Configuration Register) of the serial flash memory.

Implement the Userdef_SFLASH_Write_Enable function according to the specifications for the serial flash memory to be used so that it can be enabled for writes.

The sample code uses the serial flash control function (R_SFLASH_Spibsc_Transfer) to issue a Write Enable command (H'06) whereby enabling writes (setting the WEL bit of the Status Register to 1).

Figure 6.7 shows the Userdef_SFLASH_Write_Enable of the sample code.

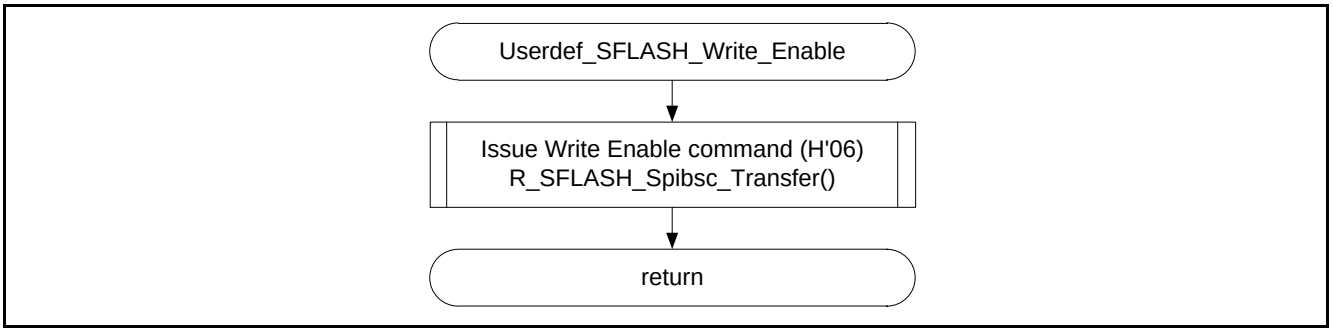


Figure 6.7 Userdef_SFLASH_Write_Enable Function Processing Flow

6.3.4 Serial Flash Memory Write Completion Wait

The serial flash memory switches into the busy state when a write is performed on its register (Status Register or Configuration Register). A wait need be inserted between the time the serial flash memory enters the busy state and the time the written data is reflected in the register.

Implement the Userdef_SFLASH_Busy_Wait function so that the sample code waits until the write to the serial flash memory gets completed according to the specifications for the serial flash memory to be used.

The sample code waits for the completion of the write by reading the WIP bit of the Status Register.

Figure 6.8 shows the Userdef_SFLASH_Busy_Wait of the sample code.

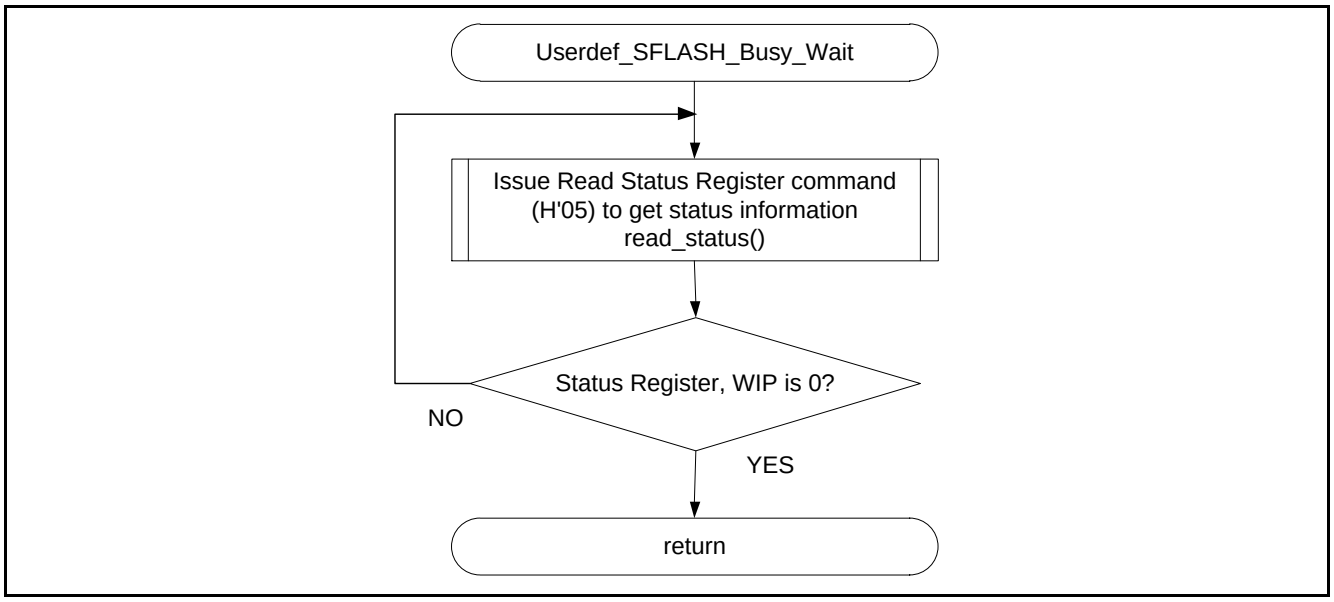


Figure 6.8 Userdef_SFLASH_Busy_Wait Function Processing Flow

7. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

8. Reference Documents

User's Manual: Hardware

RZ/A1L Group, RZ/A1H Group User's Manual: Hardware
The latest version can be downloaded from the Renesas Electronics website.

RZ/A1H AVB board YR0K77210C000BE(Renesas Starter Kit+ for RZ/A1H) User's Manual
The latest version can be downloaded from the Renesas Electronics website.

ARM Architecture Reference Manual ARMv7-A and ARMv7-R edition Issue C
The latest version can be downloaded from the ARM website.

ARM Generic Interrupt Controller Architecture Specification Architecture version 1.0
The latest version can be downloaded from the ARM website.

ARM CortexTM-A9 (Revision: r3p0) Technical Reference Manual
The latest version can be downloaded from the ARM website.

ARM CoreLinkTM Level 2 Cache Controller L2C-310 (Revision: r3p2) Technical Reference Manual
The latest version can be downloaded from the ARM website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website
http://www.renesas.com

Inquiries
http://www.renesas.com/contact/

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
Rev.1.00	Aug. 19, 2020	—	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

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

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

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 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.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, P. R. China 200333
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, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 5.0