SH7455 Group, SH7456 Group

Example of EEPROM Control Using the RSPI

R01AN0191EJ0100
Rev. 1.00
Mar. 2, 2012

Abstract

This application note describes using the Renesas Serial Peripheral Interface (RSPI) in the SH7455 Group and SH7456 Group to control the S-93C46B serial EEPROM manufactured by Seiko Instruments Inc. An explanation of the sample code for controlling the EEPROM when connected to the four-wire serial interface of the RSPI is provided below.

Products

SH7455 Group, SH7456 Group

Contents

1. Overview

1.1 Specifications

The sample code uses the RSPI in the SH7455 Group and SH7456 Group to read, erase, and write to the entire address space (16 bits × 64) of the EEPROM connected externally according to the specifications listed below. Timeouts are provided for read, erase, and write operations. The timer unit (TMU) of the SH7455 Group and SH7456 Group is used as the timeout counter.

- A single-master/single-slave configuration is used, with the SH7455 Group and SH7456 Group as the master device and the EEPROM as the slave device.
- The RSPI of the SH7455 Group and SH7456 Group is connected to the EEPROM via a four-wire serial interface. Figure 1.1 shows a connection example.
- The EEPROM (S-93C46B) has a capacity of 1 kilobits, configured as 16 bits × 64.

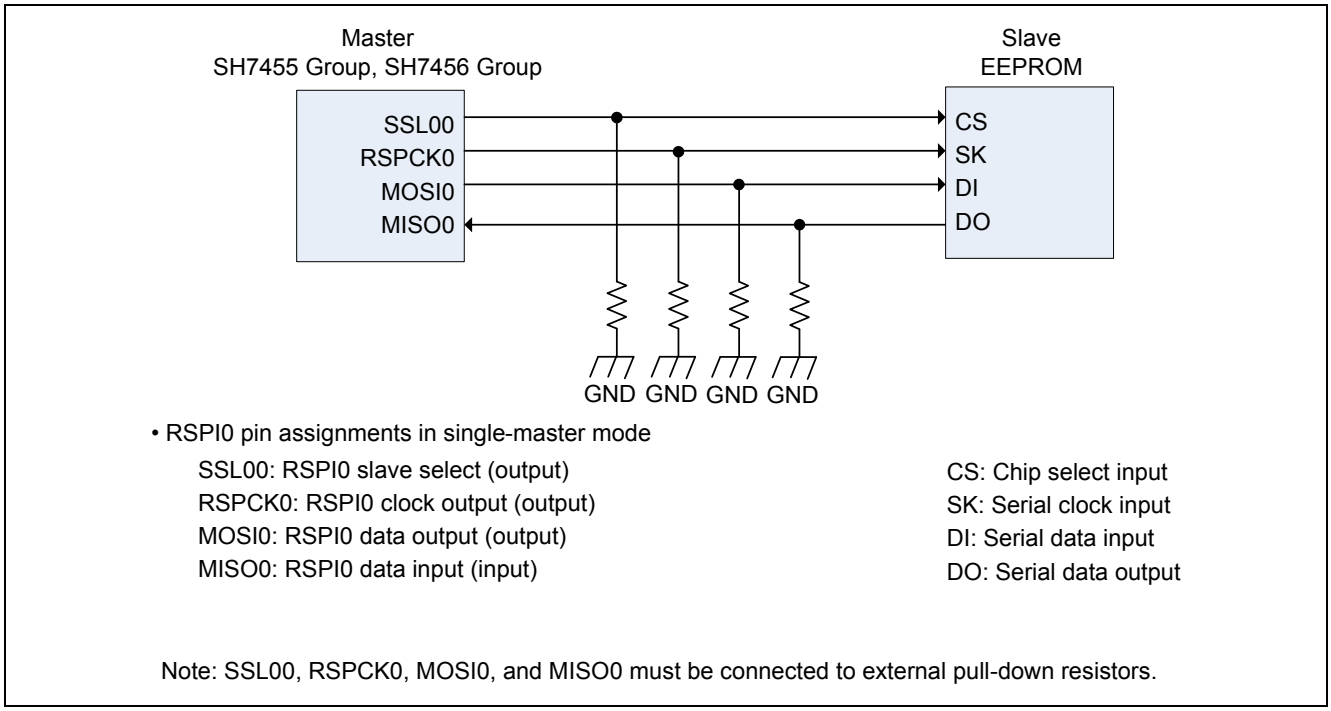


Figure 1.1 RSPI and EEPROM Connection Example

1.2 Operation Confirmation Conditions

Table 1.1 lists the conditions that apply to the sample code.

Table 1.1 Operation Confirmation Conditions

Item		Contents
MCU		SH7455 Group, SH7456 Group
	Input clock frequency	20 MHz
	Internal clock frequency setting	CPU clock (Ick): 160 MHz SHwy clock (SHck): 80 MHz Peripheral clock (Pck): 40 MHz
	Operating mode	Single-chip mode
Development tool		Renesas Electronics High-performance Embedded Workshop Version 4.08.00.011
	C/C++ Compiler	Renesas Electronics C/C++ Compiler Package for SuperH RISC Engine Family V.9.03 Release 02

2. Hardware

2.1 Pins Used

Table 2.1 lists the pin functions when RSPI channel 0 is set to single-master mode, which is used by the sample code. Pins SSL01 to SSL03 are not used.

Table 2.1 Pins Used and Their Functions

Pin Name	I/O	Function
RSPCK0	Output	RSPI0 clock output
MOSI0	Output	RSPI0 data output
MISO0	Input	RSPI0 data input
SSL00	Output	RSPI0 slave select

2.2 Renesas Serial Peripheral Interface (RSPI)

The RSPI in the SH7455 Group and SH7456 Group has three channels (RSPI0 to RSPI2), enabling high-speed serial communication in full-duplex synchronous mode between multiple processors and peripheral devices. Table 2.2 provides an overview of the RSPI.

Table 2.2 Overview of the RSPI

Item	Description
Transfer functions	• Selectable between SPI (four-wire) and clock-synchronous (three-wire) serial communication • Selectable between master and slave mode • Mode fault error and overrun error detection • Selectable serial transfer clock polarity and phase
Data format	• Switchable between MSB-first and LSB-first • Variable transfer bit length (selectable among 8 to 16, 20, 24, and 32 bits) • 128-bit transmit and receive buffers • Transfer of up to four frames (maximum frame size: 32 bits) in a single transmit or receive operation
Buffer configuration	• Double-buffer transmit and receive buffer configuration
SSL control functions	• Four SSL signals for each RSPI channel • RSPCK output and SSL assert/negate delay setting Setting range: 1 to 8 RSPCK* cycles; setting unit: 1 RSPCK cycle • Selectable SSL polarity
Control during master mode transfer	• Transfers composed of up to four commands can be executed sequentially as loops • Initiation of transfers by the CPU or DMAC • Setting of MOSI signal level at SSL negation
Interrupt sources	• Maskable interrupt sources: RSPI receive interrupt (receive buffer full) RSPI transmit interrupt (transmit buffer empty) RSPI error interrupt (mode fault, overrun)
Other	• Loopback mode • CMOS/open drain output switching function • RSPI disable (initialization) function

Note: * RSPCK: Serial transfer clock

3. Software

3.1 Operation Overview

The sample code uses the RSPI to read, erase, and write to the entire address space of the EEPROM. Figure 3.1 outlines the overall process.

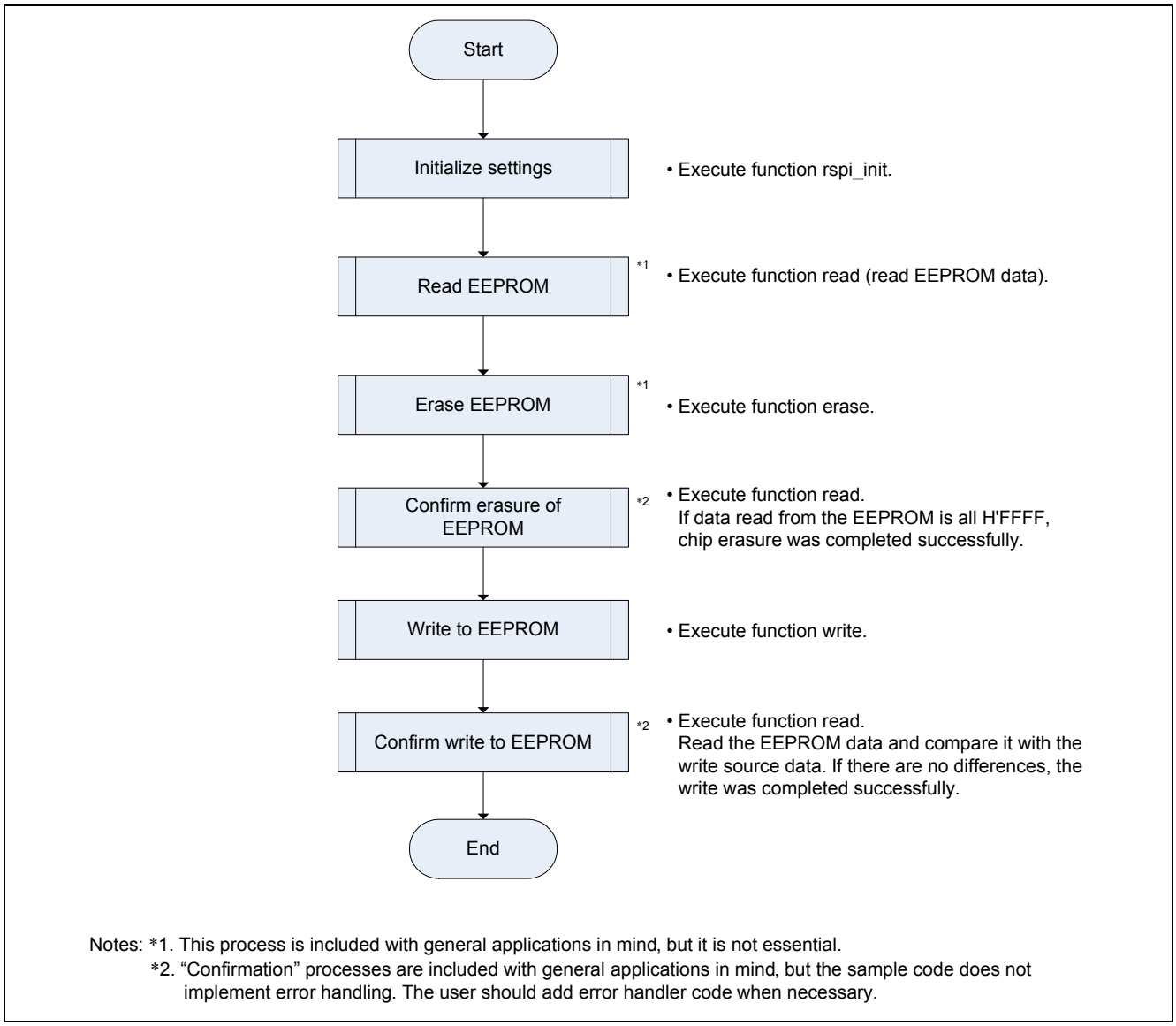


Figure 3.1 Outline of the Overall Process

3.2 EEPROM Instruction Set

Table 3.1 lists the EEPROM instructions used in the sample code. The instruction set codes are taken from the EEPROM datasheet.

Table 3.1 EEPROM Instruction Set

Instruction	Data Length	Code	Description
EWEN	9 bits	H'130	Write enable
EWDS	9 bits	H'100	Write disable
READ	9 bits	H'180 (lower 6 bits are address data)	Read data
WRITE	9 bits	H'140 (lower 6 bits are address data)	Write data
ERAL	9 bits	H'120	Chip erase (erases entire address space)

3.3 RSPI Register Settings

Table 3.2 lists the RSPI channel 0 settings used in the sample code. The setting values listed are the values used in the sample code and differ from the initial settings. When changing setting values, set values according to the standards for the target device.

In the text, the names of the bits in the registers are notated in the format “register name + dot + bit name.”

Example: SPB bit in the SP0CMD0 register → SP0CMD0.SPB

Table 3.2 RSPI Operation Overview

Function	Setting
Master/slave mode	Master mode
RSPI mode	SPI operation (four-wire)
Serial transfer clock frequency (RSPCK)	2 MHz
Data format	MSB-first
Data length	9 bits (EEPROM instruction transmit)
	16 bits (EEPROM data read/write)
Frame count/sequence length	1 (EEPROM instruction transmit, EEPROM data write)
	4 (EEPROM data read)
RSPI data register access width	16 bits
RSPI data register read value	Receive buffer
RSPI output pins	CMOS output
SSL signal active polarity	"H"
Idle time RSPCK polarity	"L"
Idle time MOSI fixed value	"L"
RSPCK delay	1 RSPCK
SSL negation delay	1 RSPCK
Next access delay	1 RSPCK + 2 Pck
RSPCK phase	Master transmit (SP0CMDi.CPHA = 0 (i = 0 to 3)) Data change: Even edge (falling edge) Data sampling: Odd edge (rising edge) Master receive (SP0CMDi.CPHA = 1 (i = 0 to 3)) Data change: Odd edge (rising edge) Data sampling: Even edge (falling edge)
SSL signal level hold	During READ instruction transmit processing During EEPROM data read processing During WRITE instruction transmit processing From transfer end to the next access start, the SSL signal level from the previous data transfer is maintained. Other than the above All SSL signals are negated at transfer end.
Generation of RSPI interrupts	Disabled

(1) RSPI0 Control Register (SP0CR)

The SP0CR register sets the operating mode of RSPI0. The setting value is H'00 when SP0CR is initialized in function rspi_init , H'08 when RSPI functions are disabled after execution of function rspi_init, and H'48 when RSPI functions are enabled after execution of function rspi_init (refer to section 4.1 for details on the rspi_init function).

Address	Setting Value	Bit		Description
H'FFFF B000	H'00 or H'08 or H'48	7	SPRIE	0: Disables generation of RSPI0 receive interrupt requests.
		6	SPE	0: Disables RSPI0 functions. 1: Enables RSPI0 functions.
		5	SPTIE	0: Disables generation of RSPI0 transmit interrupt requests.
		4	SPEIE	0: Disables generation of RSPI0 error interrupt requests.
		3	MSTR	0: Slave mode 1: Master mode
		2	MODFEN	0: Disables mode fault error detection.
		1	—	0: Reserved bit (The write value should always be "0".)
		0	SPMS	0: SPI operation (four-wire)

(2) RSPI0 Slave Select Polarity Register (SP0SSLP)

SP0SSLP sets the polarity of the SSL00 and SSL01 signals of RSPI0.

Address	Setting Value	Bit		Description
H'FFFF B001	H'01	7 to 2	—	0: Reserved bits (The write value should always be "0".)
		1	SSL1P	0: SSL01 signal is “L” active.
		0	SSL0P	1: SSL00 signal is “H” active.

(3) RSPI0 Pin Control Register (SP0PCR)

SP0PCR sets the modes of the RSPI0 pins.

Address	Setting Value	Bit		Description
H'FFFF B002	H'20	7, 6	—	0: Reserved bits (The write value should always be "0".)
		5	MOIFE	1: MOSI0 output value equals MOIFV bit setting value.
		4	MOIFV	0: MOSI0 idle fixed value equals “L” level.
		3	—	0: Reserved bit (The write value should always be "0".)
		2	SPOM	0: CMOS output
		1	—	0: Reserved bit (The write value should always be "0".)
		0	SPLP	0: Normal mode

(4) RSPI0 Clock Delay Register (SP0CKD)

Sets the period (RSPCK delay) from the beginning of SSL signal assertion to RSPCK oscillation.

Address	Setting Value	Bit		Description
H'FFFF B00C	H'00	7 to 3	—	0: Reserved bits (The write value should always be "0".)
		2 to 0	SCKDL	000: 1 RSPCK

(5) RSPI0 Slave Select Negation Delay Register (SP0SSLND)

SP0SSLND sets the period (SSL negation delay) from the transmission of the final RSPCK edge to the negation of the SSL signal during a serial transfer by the RSPI0 in master mode.

Address	Setting Value	Bit		Description
H'FFFF B00D	H'00	7 to 3	—	0: Reserved bits (The write value should always be "0".)
		2 to 0	SLNDL	000: 1 RSPCK

(6) RSPI0 Next-Access Delay Register (SP0ND)

Specifies the non-active period (next-access delay) of the SSL signal after the end of a serial transfer.

Address	Setting Value	Bit		Description
H'FFFF B00E	H'00	7 to 3	—	0: Reserved bits (The write value should always be "0".)
		2 to 0	SPNDL	000: 1 RSPCK + 2 Pck

(7) RSPI0 Command Register 0 (SP0CMD0)

SP0CMD0 sets the transfer format of RSPI command register 0. The setting value is H'E701 after execution of function rspi_init; H'E800 during transmission of the EWEN instruction, during transmission of the EWDS instruction, during transmission of the ERAL instruction, or during transmission of data (H'000) for EEPROM chip erase and obtaining the data write status; H'E880 during transmission of the READ instruction or WRITE instruction; H'EF00 when writing data to the EEPROM; and H'EF81 when reading data from the EEPROM (refer to section 4.1 for details on the rspi_init function).

Address	Setting Value	Bit		Description
H'FFFF B010	H'E701 or H'E800 or H'E880 or H'EF00 or H'EF81	15	SCKDEN	1: RSPCK delay equals the RSPI0 clock delay register (SP0CKD) setting value.
		14	SLNDEN	1: SSL negation delay equals the RSPI0 slave select negation delay register (SP0SSLND) setting value.
		13	SPNDEN	1: Next-access delay equals the RSPI0 next-access delay register (SP0ND) setting value.
		12	LSBF	0: MSB-first
		11 to 8	SPB	0100 to 0111: RSPI transfer data length = 8 bits 1000: RSPI transfer data length = 9 bits 1111: RSPI transfer data length = 16 bits
		7	SSLKP	0: Negate all SSL signals at transfer end. 1: Maintain the SSL signal level from transfer end until the start of the next access.
		6 to 4	SSLA	000: SSL00 setting
		3, 2	BRDV	00: Select the base bit rate.
		1	CPOL	0: RSPCK equals 0 when idle.
		0	CPHA	0: Data sampling at odd edge, data change at even edge 1: Data change at odd edge, data sampling at even edge

(8) RSPI0 Command Register 1 (SP0CMD1)

SP0CMD1 sets the transfer format of RSPI command register 1. The setting value is H'E701 after execution of function rspi_init and H'EF81 when reading data from the EEPROM (refer to section 4.1 for details on the rspi_init function). For details of the register configuration and the functions of the bits, see (7) RSPI0 Command Register 0 (SP0CMD0).

Address	Setting Value	Bit	Description
H'FFFF B012	H'E701 or H'EF81	See (7) RSPI0 Command Register 0 (SP0CMD0).	

(9) RSPI0 Command Register 2 (SP0CMD2)

SP0CMD2 sets the transfer format of RSPI command register 2. The setting value is H'E701 after execution of function rspi_init and H'EF81 when reading data from the EEPROM (refer to section 4.1 for details on the rspi_init function). For details of the register configuration and the functions of the bits, see (7) RSPI0 Command Register 0 (SP0CMD0).

Address	Setting Value	Bit	Description
H'FFFF B014	H'E701 or H'EF81	See (7) RSPI0 Command Register 0 (SP0CMD0).	

(10) RSPI0 Command Register 3 (SP0CMD3)

SP0CMD3 sets the transfer format of RSPI command register 3. The setting value is H'E701 after execution of function rspi_init (4.1) and H'EF01 when reading final address data from the EEPROM, and H'EF81 when reading data from the EEPROM (refer to section 4.1 for details on the rspi_init function). For details of the register configuration and the functions of the bits, see (7) RSPI0 Command Register 0 (SP0CMD0).

Address	Setting Value	Bit	Description
H'FFFF B016	H'E701 or H'EF01 or H'EF81	See (7) RSPI0 Command Register 0 (SP0CMD0).	

(11) RSPI0 Bit Rate Register (SP0BR)

SP0BR sets the bit rate when the RSPI0 is in master mode.

Address	Setting Value	Bit		Description
H'FFFF B00A	H'09	7 to 0	SPR7 to SPR0	These bits set the bit rate in master mode. The setting value specifies a serial transfer clock frequency of 2 MHz (Pck = 40 MHz).

An equation for calculating the bit rate from the SP0BR setting value and the setting value of the BRDV bits in RSPI0 command registers 0 to 3 (SP0CMD0 to SP0CMD3) is given below.

$$\text{Bit rate} = \frac{f(Pck)}{2 \times (SP0BR + 1) \times 2^{BRDV}}$$

The relationship between the SP0BR setting value, the setting value of the BRDV bits, and the bit rate when Pck equals 40 MHz is shown in table 3.3.

Table 3.3 Relationship between SP0BR Setting Value, Setting Value of BRDV Bits, and Bit Rate (Pck = 40 MHz)

SP0BR Register Setting Value	Setting Value of BRDV Bits	Division Ratio	Serial Transfer Clock Frequency
1	0	4	10 MHz
2	0	6	6.67 MHz
3	0	8	5 MHz
4	0	10	4 MHz
5	0	12	3.33 MHz
	1	24	1.67 MHz
	2	48	0.83 MHz
	3	96	0.42 MHz
9	0	20	2 MHz
255	3	4096	9.77 kHz

(12) RSPI0 Sequence Control Register (SP0SCR)

SP0SCR sets the sequence control method when the RSPI operates in master mode. The setting value is H'00 during EEPROM instruction transmission or when writing data to the EEPROM, and H'03 when reading data from the EEPROM.

Address	Setting Value	Bit		Description
H'FFFF B008	H'00 or H'03	7 to 2	—	0: Reserved bits (The write value should always be "0".)
		1, 0	SPSLN	00: Sequence length = 1 (referenced SP0CMD register: 0 → 0 →...) 11: Sequence length = 4 (referenced SP0CMD register: 0 → 1 → 2 → 3 → 0 →...)

(13) RSPI0 Data Control Register (SP0DCR)

SP0DCR specifies the function of the RSPI data register (SP0DR). The setting value is H'00 during EEPROM instruction transmission or when writing data to the EEPROM, and H'03 when reading data from the EEPROM.

Address	Setting Value	Bit		Description
H'FFFF B00B	H'00 or H'03	7, 6	—	0: Reserved bits (The write value should always be "0".)
		5	SPLW	0: Word (16-bit) access to SP0DR register
		4	SPRDTD	0: Reading SP0DR register returns receive buffer value.
		3, 2	—	0: Reserved bits (The write value should always be "0".)
		1, 0	SPFC	00: Number of frames that can be stored in SP0DR = 1 11: Number of frames that can be stored in SP0DR = 4

Table 3.4 lists the setting value combinations for the SPFC bits and the SPSLN bits in the RSPI0 sequence control register (SP0SCR), using the frame configuration example shown in figure 3.2. Subsequent operation cannot be guaranteed if a setting value combination other than those listed in table 3.4 is used.

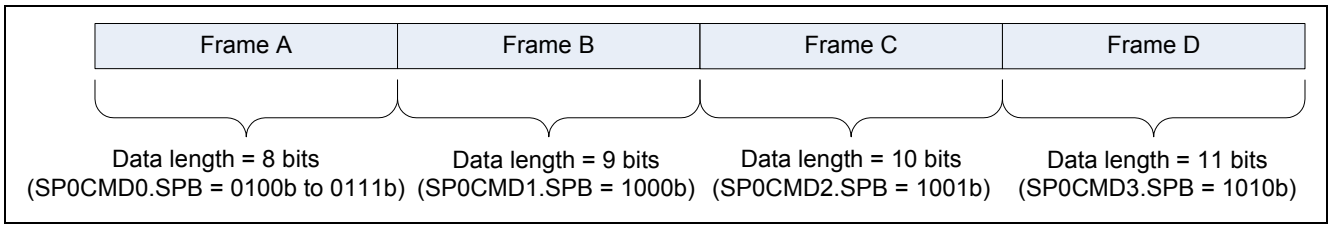


Figure 3.2 Frame Configuration Example

Table 3.4 Setting Value Combinations for SPSLN and SPFC Bits

SPSLN Bits	SPFC Bits	Transfer Frame Count	Transfer Frame Sequence	Transfer Bit Count
00b (sequence length = 1)	00b (frame count = 1)	1	A	8 bits
	01b (frame count = 2)	2	A → A	16 bits
	10b (frame count = 3)	3	A → A → A	24 bits
	11b (frame count = 4)	4	A → A → A → A	32 bits
01b (sequence length = 2)	01b (frame count = 2)	2	A → B	17 bits
	11b (frame count = 4)	4	A → B → A → B	34 bits
10b (sequence length = 3)	10b (frame count = 3)	3	A → B → C	27 bits
11b (sequence length = 4)	11b (frame count = 4)	4	A → B → C → D	38 bits

3.4 I/O Port Register Settings

The register settings used in the sample code with relation to I/O port G are described below. The setting values listed are the values used in the sample code and differ from the initial settings. When changing setting values, make settings according to the standards for the target device.

(1) Port GHJ Input Threshold Value Switching Register (PGLVR)

PGLVR sets the input threshold values of ports G, H, and J.

Address	Setting Value	Bit		Description
H'FFFF 5B00	H'000B	15 to 12	—	0: Reserved bits (The write value should always be "0".)
		11	PJPIEN	0000: Input prohibited state
		10	PJSCSEL	
		9	PJSEL0	
		8	PJSEL1	
		7	PHPIEN	0000: Input prohibited state
		6	PHSCSEL	
		5	PHSEL0	
		4	PHSEL1	
		3	PGPIEN	1011: Port G input level = CMOS input, 0.70 Vcc
		2	PGSCSEL	
		1	PGSEL0	
		0	PGSEL1	

(2) Port G Control Register 1 (PGCR1)

PGCR1 is used to select RSPI-related settings as the functions of the multiplexed pins of port G.

Address	Setting Value	Bit		Description
H'FFFF 5816	H'3111	15	—	0: Reserved bit (The write value should always be "0".)
		14 to 12	PG3MD	011: SSL00 input/output (RSPI)
		11	—	0: Reserved bit (The write value should always be "0".)
		10 to 8	PG2MD	001: RSPCK0 input/output (RSPI)
		7	—	0: Reserved bit (The write value should always be "0".)
		6 to 4	PG1MD	001: MISO0 input/output (RSPI)
		3	—	0: Reserved bit (The write value should always be "0".)
		2 to 0	PG0MD	001: MOSI0 input/output (RSPI)

3.5 TMU Register Settings

The register settings used in the sample code with relation to TMU channel 0 are described below. The setting values listed are the values used in the sample code and differ from the initial settings.

(1) TM Start Register (TMSTR)

TMSTR selects whether the TM0CNT counter operates or is stopped.

Address	Setting Value	Bit		Description
H'FFFF D004	H'01	7 to 3	—	0: Reserved bits (The write value should always be "0".)
		2	STR2	0: The TM2CNT counter is stopped.
		1	STR1	0: The TM1CNT counter is stopped.
		0	STR0	1: The TM0CNT counter operates.

(2) TM0 Control Register (TM0CR)

TM0CR selects the counter clock and controls the generation of interrupts.

Address	Setting Value	Bit		Description
H'FFFF D010	H'0000	15 to 9	—	0: Reserved bits (The write value should always be "0".)
		8	UNF	0: Indicates that a TM0CNT counter underflow has not occurred. 1: Indicates that a TM0CNT counter underflow has occurred.
		7, 6	—	0: Reserved bits (The write value should always be "0".)
		5	UNIE	0: The underflow interrupt (TUNI) is disabled.
		4, 3	—	0: Reserved bits (The write value should always be "0".)
		2 to 0	TPSC	000: Set Pck/4 as the TM0CNT count clock.

(3) TM0 Counter (TM0CNT)

TM0CNT is a down counter that is decremented by the input clock signal selected by the TPSC bits in the TM0CR register.

Address	Setting Value	Bit		Description
H'FFFF D00C	H'FFFF FFFF	31 to 0	TM0CNT	32-bit counter value

(4) TM0 Constant Register (TM0COR)

The value of the TM0COR register is loaded into the TM0CNT counter when an underflow occurs as the result of decrementing the TM0CNT counter, and decrementing of the TM0CNT counter then continues from the loaded value.

Address	Setting Value	Bit		Description
H'FFFF D008	H'FFFF FFFF	31 to 0	TM0COR	32-bit value that will be loaded into the TM0CNT counter when the TM0CNT counter underflows

3.6 File Composition

Table 3.5 lists the files used in the sample code.

Table 3.5 Files Used in the Sample Code

File Name	Outline
dbsct.c	Section B and section D setting file
env.inc	Exception event register and interrupt event register address definition file
main.c	Main function program
resetprg.c	Reset program
rspi.c, rspi.h	RSPI control program and header file
sh7455_iodefine_20101029.h	SH7455 Group and SH7456 Group peripheral function register definition file
stacksct.h	Stack size definition file
typedefine.h	Type declaration file
vect.inc	Vector definition file
vecttbl.src	Interrupt vector table definition file
vhandler.src	Interrupt handler program

3.7 Section Information

Table 3.6 lists section information.

Table 3.6 Section Information

Address		Description
H'0000 0000	RSTHandler	Reset handler
H'0000 0800	INTHandler	Exception/interrupt handler
	VECTTBL	Vector table
	INTTBL	Interrupt mask table
H'0000 1000	PResetPRG	Reset program
H'0000 3000	P	Program area
	C	Constant area
	C$BSEC	Section B initialization table
	C$DSEC	Section D initialization table
	D	Initialization data area
H'E500 E000	B	Variable area
	R	Initialized variable area
H'E501 1C00	S	Stack address area

4. RSPI Functions

The functions defined in the file **rspi.c** for initialization (rspi_init), reading (read), erasing (erase), writing (write), and error handling are described below. To use these functions it is necessary to include the file **rspi.h**.

4.1 Function rspi_init

Table 4.1 provides an overview of function rspi_init, and figure 4.1 is a flowchart of the function. It is necessary to execute function rspi_init once before using function read, function erase, or function write.

Table 4.1 Overview of Function rspi_init

Function	Arguments	Return values	Description
rspi_init	None	None	RSPI channel 0 settings • Master/slave mode: Master mode • RSPI mode: SPI operation (four-wire) • Serial transfer clock frequency (RSPCK): 2 MHz • Data format: MSB-first • RSPI data register access width: 16 bits • RSPI data register read value: Receive buffer • RSPI output pins: CMOS output • SSL signal active polarity: “H” • Assert settings for SSL01 to SSL00 signals: SSL00 setting • Idle time RSPCK polarity: “L” • Idle time MOSI fixed value: “L” • RSPCK delay: 1 RSPCK • SSL negation delay: 1 RSPCK • Next-access delay: 1 RSPCK + 2 Pck • Generation of RSPI interrupts: Disabled • Frame count: 1 • Data length: 8 bits • SSL signal level hold: Negation of all SSL signals at transfer end • RSPCK phase: Data change at odd edge, data sampling at even edge Port G settings • Input threshold value: CMOS input, 0.70 Vcc • PG3 = SSL00 output, PG2 = RSPCK0 output, PG1 = MISO0 input, PG0 = MOSI0 output TMU channel 0 settings • Counter input clock: Pck/4 • Interrupt at underflow: Disabled • TM0CNT counter value: H'FFFF FFFF • TM0CNT counter setting value at underflow: H'FFFF FFFF • TM0 counter start: Counter operates

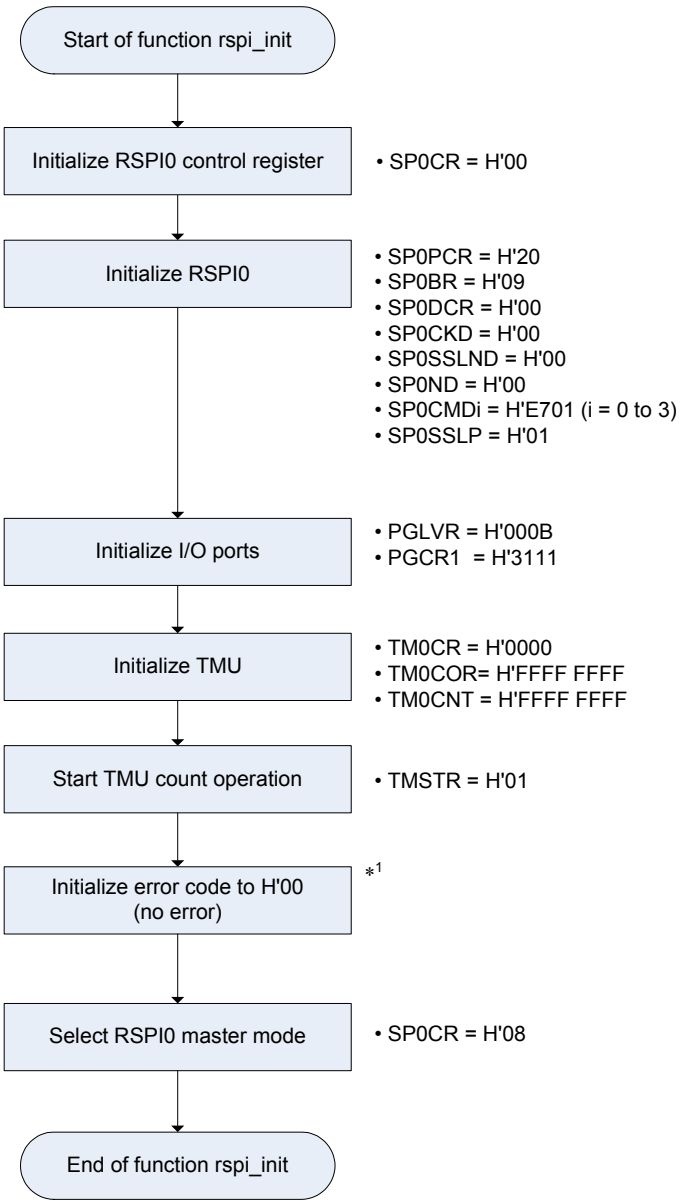


Figure 4.1 Function rspi_init Function

4.2 Function read

Table 4.2 provides an overview of function read, and figure 4.2 is an RSPI timing chart overview of the function. Before using this function, it is necessary to execute function rspi_init once.

Table 4.2 Overview of Function read

Function	Arguments	Return values	Description
read	Storage address for read data	None	• Reads the entire address space of the EEPROM, from the start address (H'00) to the end address (H'3F). • Stores the read data at the address specified by the argument.

Function read performs the following two processes:

[1] Transmits the data read instruction (READ instruction)
[2] Performs continuous data read

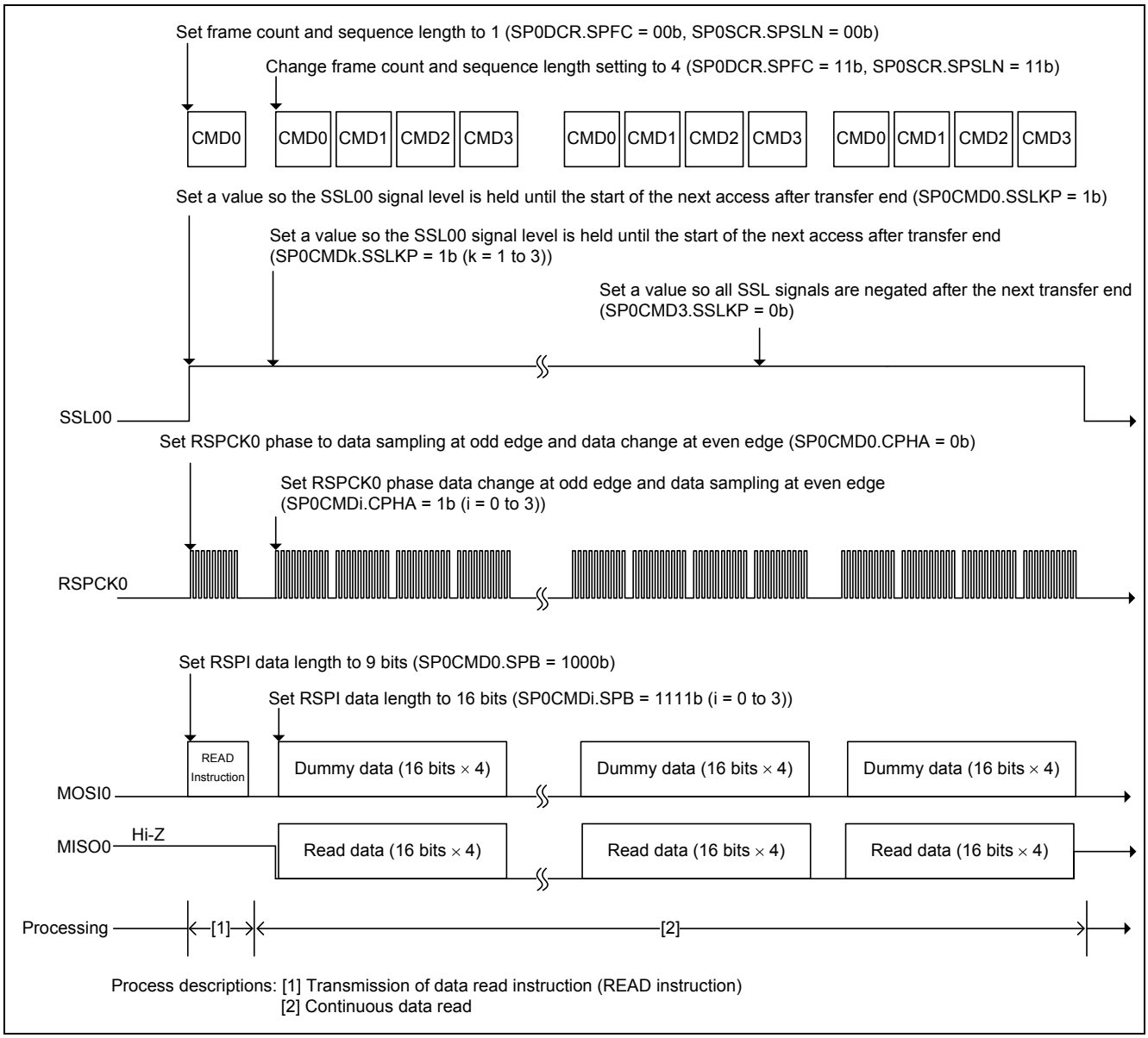


Figure 4.2 RSPI Timing Chart Overview of Function read

[1] Transmission of data read instruction (READ instruction)

To read data from the EEPROM, the function transmits a data read instruction (READ instruction) and read address to the EEPROM. An overview of the operation is shown below. Figure 4.3 is a flowchart of the function.

- H'00 (EEPROM start address) is specified as the read address.
- A READ instruction is transmitted to the EEPROM.
- While transmitting the READ instruction, the SSL00 signal (“H” output) is asserted.
- After transmitting the READ instruction, assertion of SSL00 signal (“H” output) is maintained.
- The RSPI transfer format shown in table 4.3 is used to transmit the READ instruction.

Table 4.3 RSPI Transfer Format for Transmission of READ Instruction

Item		Register	Setting Value	Description
Frame count		SP0DCR.SPFC	00b	Set frame count to 1.
Sequence length		SP0SCR.SPSLN	00b	Set sequence length to 1.
SP0CMD0	Data length	SP0CMD0.SPB	1000b	Set data length to 9 bits.
	SSL signal level hold	SP0CMD0.SSLKP	1b	Hold SSL00 signal level between transfer end and start of next access.
	RSPCK phase	SP0CMD0.CPHA	0b	Data change: Even edge (falling edge) Data sampling: Odd edge (rising edge)

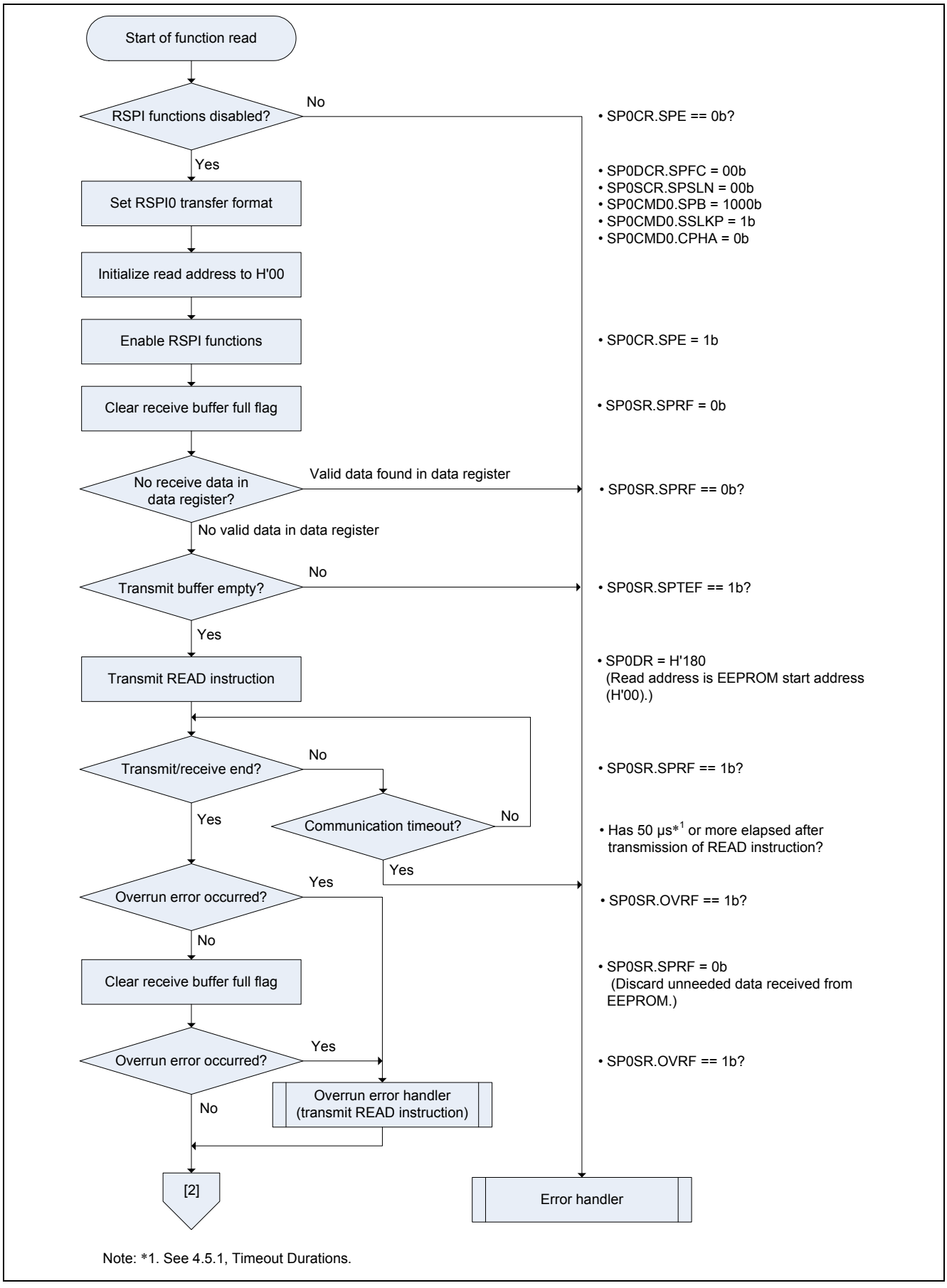


Figure 4.3 Function read (1/3)

[2] Continuous data read

The function outputs the serial transfer clock (RSPCK0) and reads data from the entire address space of the EEPROM. An overview of the operation is shown below. Figures 4.4 and 4.5 are flowcharts of the function. After the final 64 bits of data are read, the RSPI transfer format is changed during communication to negate the SSL00 signal (“L” output).

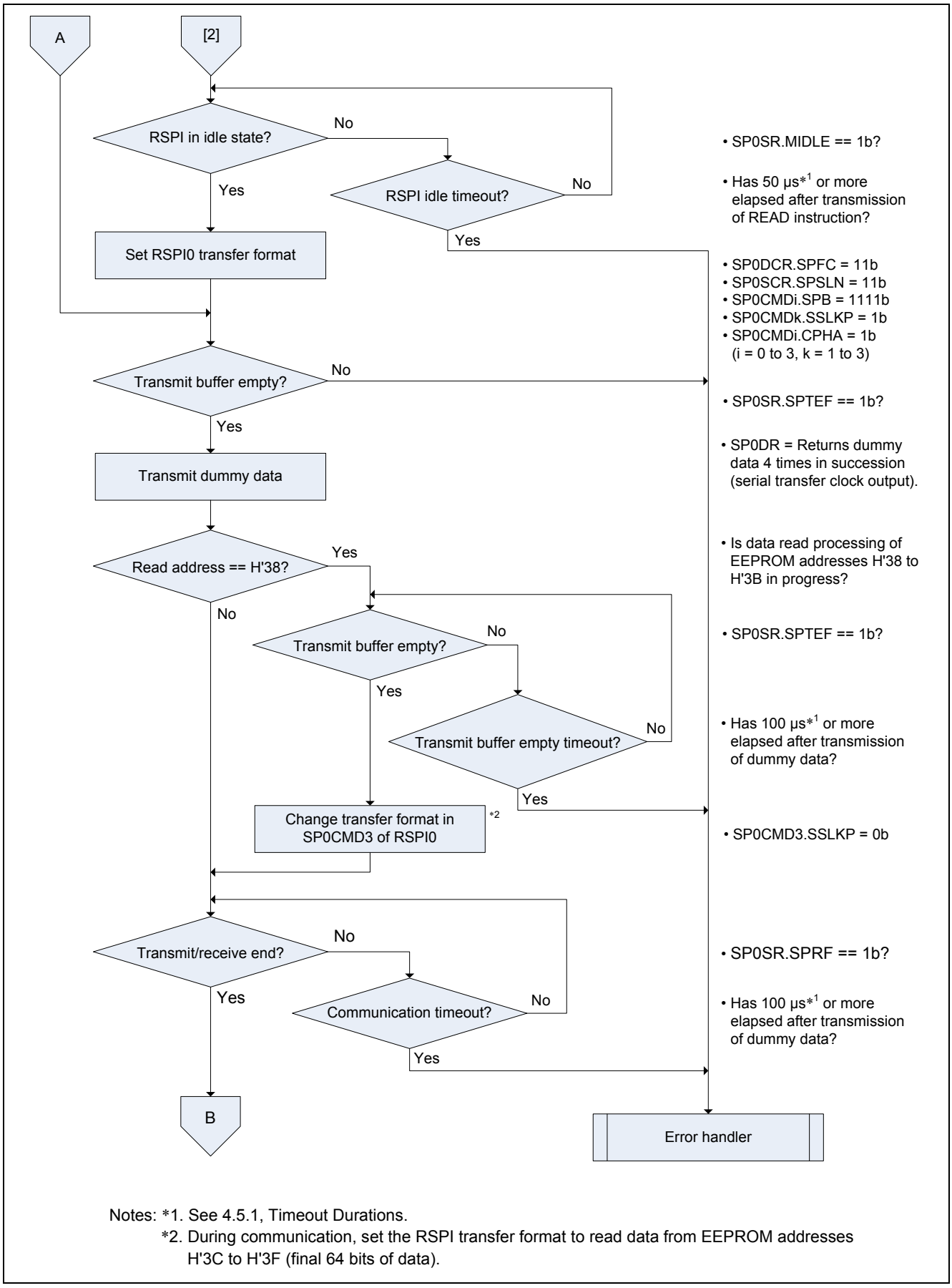
- Data is read in 64-bit units (16 bits × 4) from the entire address space of the EEPROM, from the start address (H'00) to the end address (H'3F).
- Assertion of the SSL00 signal (“H” output) is maintained during continuous data read, and the SSL00 signal (“L” output) is negated after the end of continuous data read.
- To read data, dummy data (user-defined value) is transmitted and the serial transfer clock (RSPCK0) is output.
- If the SSL00 signal is in the asserted state (“H” output), after receiving a READ instruction the EEPROM automatically increments the address by 1 each time 16 bits of data is output. It is therefore not necessary to specify the read address with subsequent READ instructions.
- To negate the SSL00 signal (“L” output) after the final 64 bits of data have been read, the RSPI transfer format for the data (the final 64 bits of data) at EEPROM addresses H'3C to H'3F (table 4.5) is set while the data at EEPROM addresses H'38 to H'3B is being read.
- To control the continuous data read process, the read address is incremented by 4 each time 64 bits of data is read.
- The RSPI transfer format shown in table 4.4 is used to read the data at EEPROM addresses H'00 to H'3B. (Items that differ from table 4.3 are shown in **bold** text.)
- The RSPI transfer format shown in table 4.5 is used to read the data at EEPROM addresses H'3C to H'3F (the final 64 bits of data). (Items that differ from table 4.4 are shown in **bold** text.)

Table 4.4 RSPI Transfer Format for Reading Data at EEPROM Addresses H'00 to H'3B

Item		Register	Setting Value	Description
Frame count		**SP0DCR.SPFC**	**11b**	**Set frame count to 4.**
Sequence length		**SP0SCR.SPSLN**	**11b**	**Set sequence length to 4.**
SP0CMDi (i = 0 to 3)	**Data length**	**SP0CMDi.SPB**	**1111b**	**Set data length to 16 bits.**
	SSL signal level hold	SP0CMDi.SSLKP	1b	Hold SSL00 signal level after transfer end to start of next access.
	RSPCK phase	**SP0CMDi.CPHA**	**1b**	**Data change: Odd edge (rising edge) Data sampling: Even edge (falling edge)**

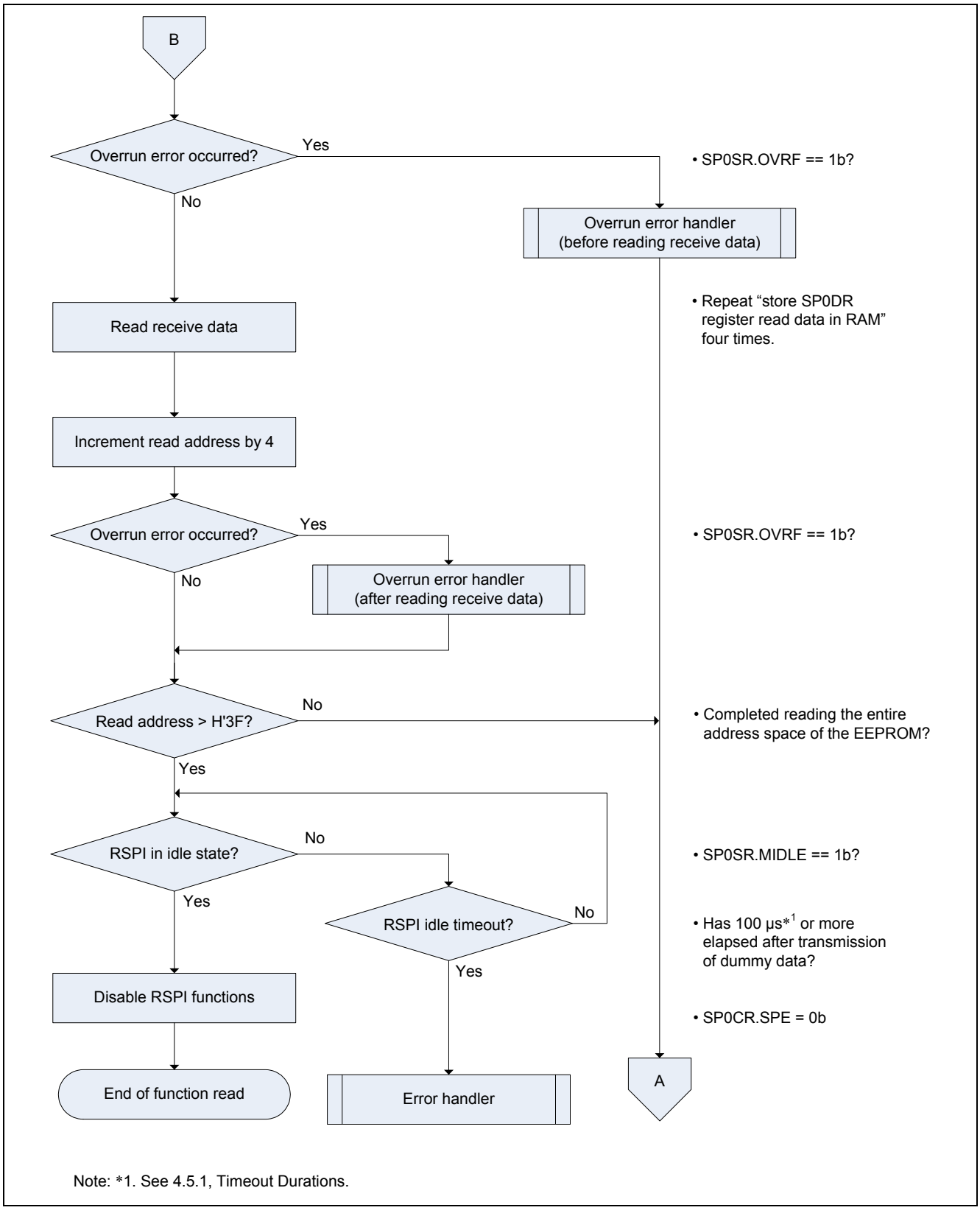
Table 4.5 RSPI Transfer Format for Reading Data at EEPROM Addresses H'3C to H'3F (Final 64 Bits of Data)

Item		Register	Setting Value	Description
Frame count		SP0DCR.SPFC	11b	Set frame count to 4.
Sequence length		SP0SCR.SPSLN	11b	Set sequence length to 4.
SP0CMDj (j = 0 to 2)	Data length	SP0CMDj.SPB	1111b	Set data length to 16 bits.
	SSL signal level hold	SP0CMDj.SSLKP	1b	Hold SSL00 signal level after transfer end to start of next access.
	RSPCK phase	SP0CMDj.CPHA	1b	Data change: Odd edge (rising edge) Data sampling: Even edge (falling edge)
SP0CMD3	Data length	SP0CMD3.SPB	1111b	Set data length to 16 bits.
	SSL signal level hold	**SP0CMD3.SSLKP**	**0b**	**Negate all SSL signals at transfer end.**
	RSPCK phase	SP0CMD3.CPHA	1b	Data change: Odd edge (rising edge) Data sampling: Even edge (falling edge)



Notes: *1. See 4.5.1, Timeout Durations.
*2. During communication, set the RSPI transfer format to read data from EEPROM addresses H'3C to H'3F (final 64 bits of data).

Figure 4.4 Function read (2/3)



Note: *1. See 4.5.1, Timeout Durations.

Figure 4.5 Function read (3/3)

4.3 Function erase

Table 4.6 provides an overview of function erase, and figure 4.6 is an RSPI timing chart overview of the function. Before using this function, it is necessary to execute function rspi_init once.

Table 4.6 Overview of Function erase

Function	Arguments	Return values	Description
erase	None	None	Erases the entire address space of the EEPROM. After erasure, the data value of the entire address space of the EEPROM is H'FFFF.

Function erase performs the following four processes:

[1] Transmission of write enable instruction (EWEN instruction)
[2] Transmission of chip erase instruction (ERAL instruction)
[3] Waiting for chip erase to complete
[4] Transmission of write disable instruction (EWDS instruction)

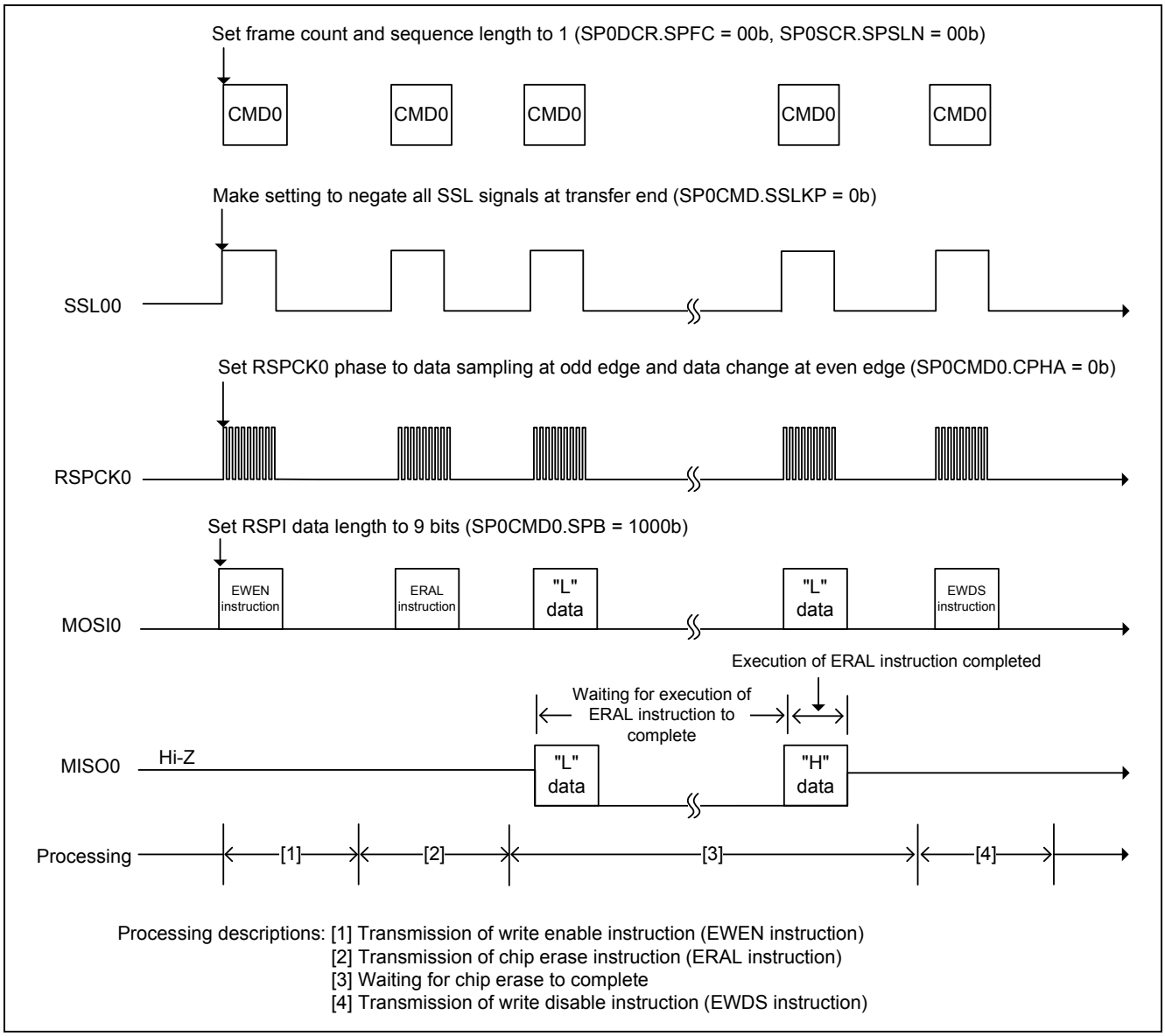


Figure 4.6 RSPI Timing Chart Overview of Function erase

[1] Transmission of write enable instruction (EWEN instruction)

To enable writing to the EEPROM, the function transmits a write enable instruction (EWEN instruction) to the EEPROM. An overview of the operation is shown below. Figure 4.7 is a flowchart of the function.

- An EWEN instruction is transmitted to the EEPROM.
- The SSL00 signal is asserted ("H" output) during transmission of the EWEN instruction.
- After transmission of the EWEN instruction, the SSL00 signal is negated ("L" output).
- The RSPI transfer format shown in table 4.7 is used to transmit the EWEN instruction.

Table 4.7 RSPI Transfer Format for Function erase

Item		Register	Setting Value	Description
Frame count		SP0DCR.SPFC	00b	Set frame count to 1.
Sequence length		SP0SCR.SPSLN	00b	Set sequence length to 1.
SP0CMD0	Data length	SP0CMD0.SPB	1000b	Set data length to 9 bits.
	SSL signal level hold	SP0CMD0.SSLKP	0b	Negate all SSL signals at transfer end.
	RSPCK phase	SP0CMD0.CPHA	0b	Data change: Even edge (falling edge) Data sampling: Odd edge (rising edge)

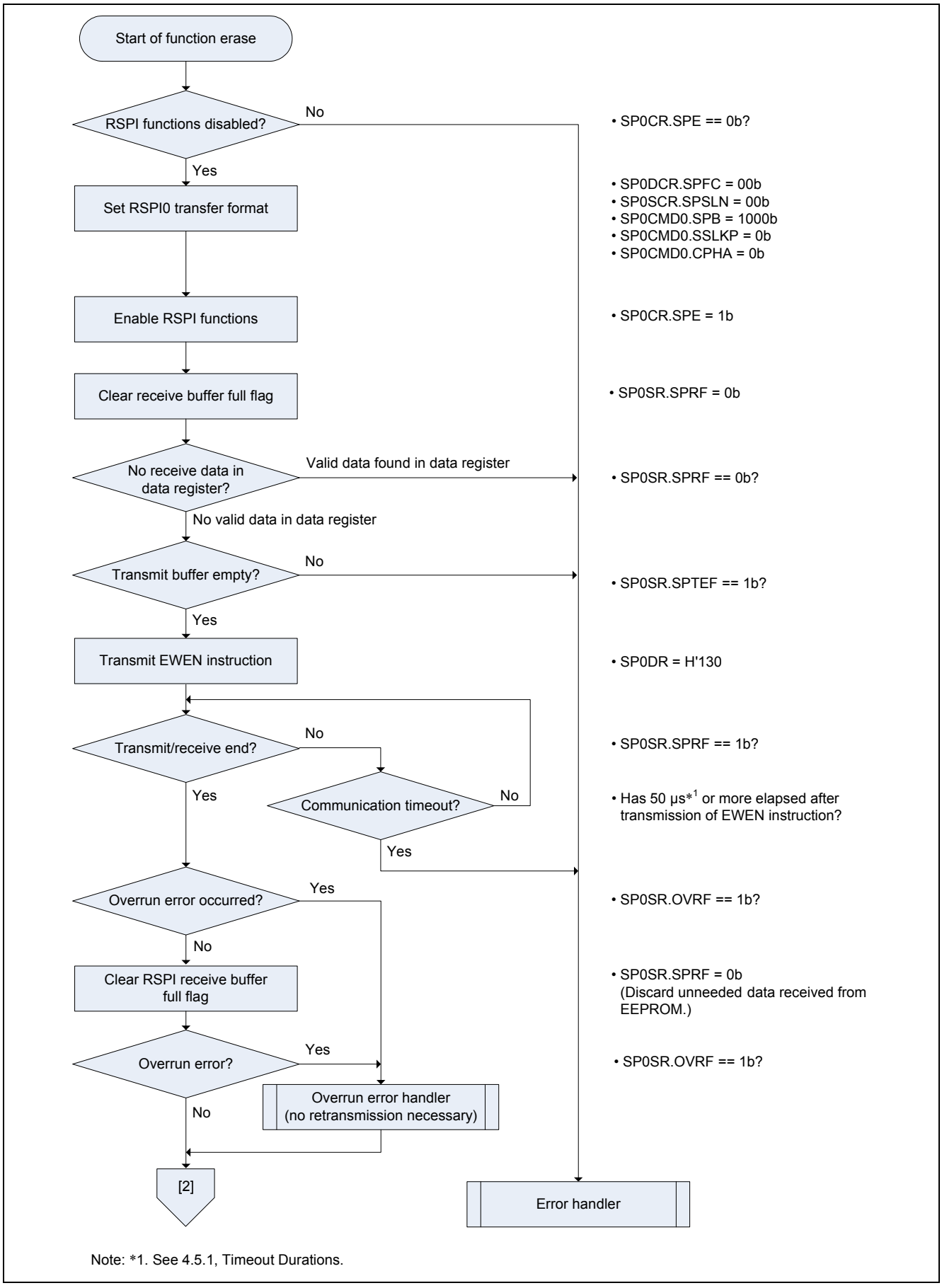


Figure 4.7 Function erase (1/4)

[2] Transmission of chip erase instruction (ERAL instruction)

To erase the entire address space of the EEPROM, the function transmits a chip erase instruction (ERAL instruction) to the EEPROM. An overview of the operation is shown below. Figure 4.8 is a flowchart of the function.

- An ERAL instruction is transmitted to the EEPROM.
- The SSL00 signal is asserted ("H" output) during transmission of the ERAL instruction.
- After transmission of the ERAL instruction, the SSL00 signal is negated ("L" output).
- The same RSPI transfer format shown in table 4.7 of "[1] Transmission of write enable instruction (EWEN instruction)" is used to transmit the ERAL instruction.

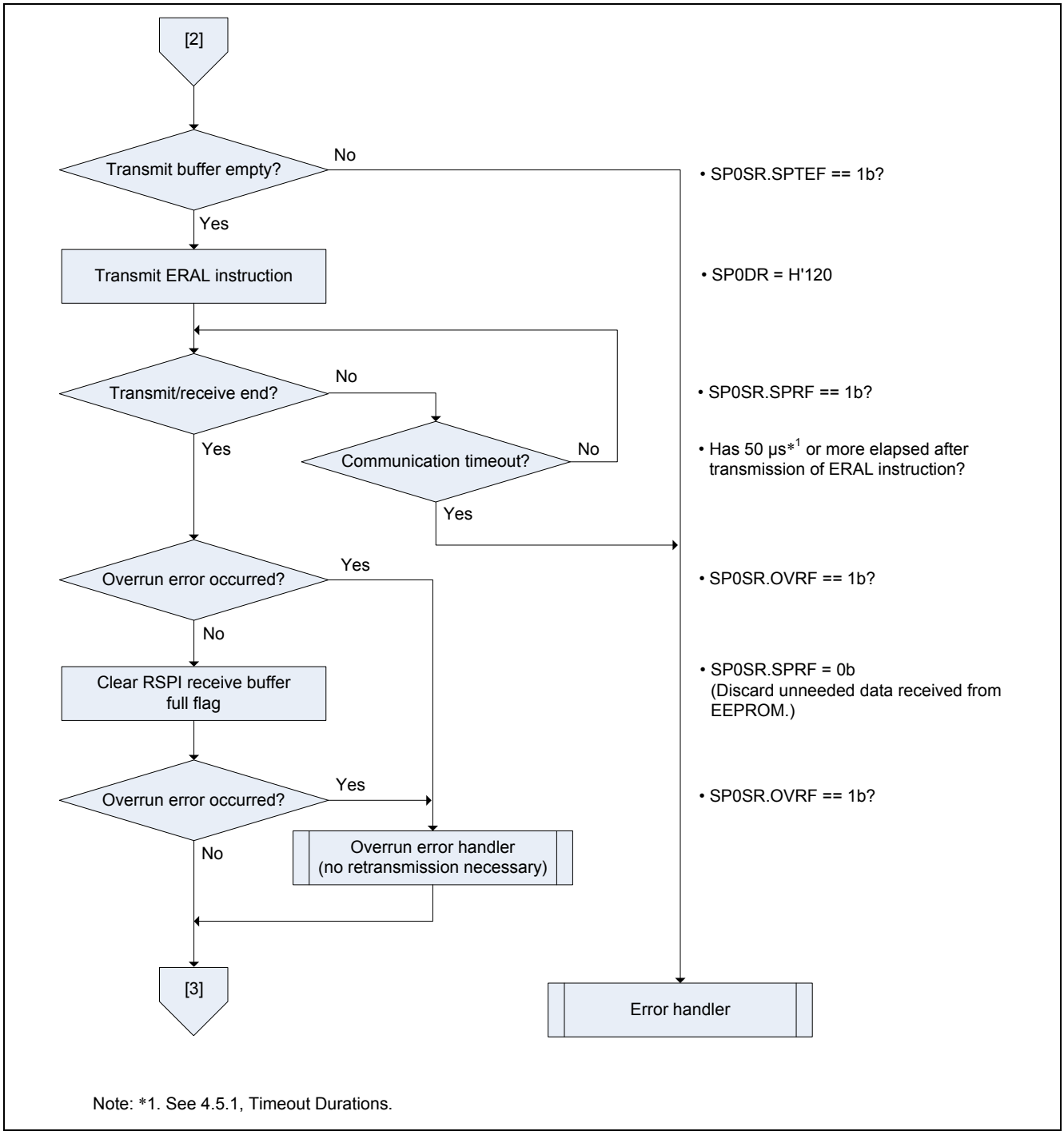


Figure 4.8 Function erase (2/4)

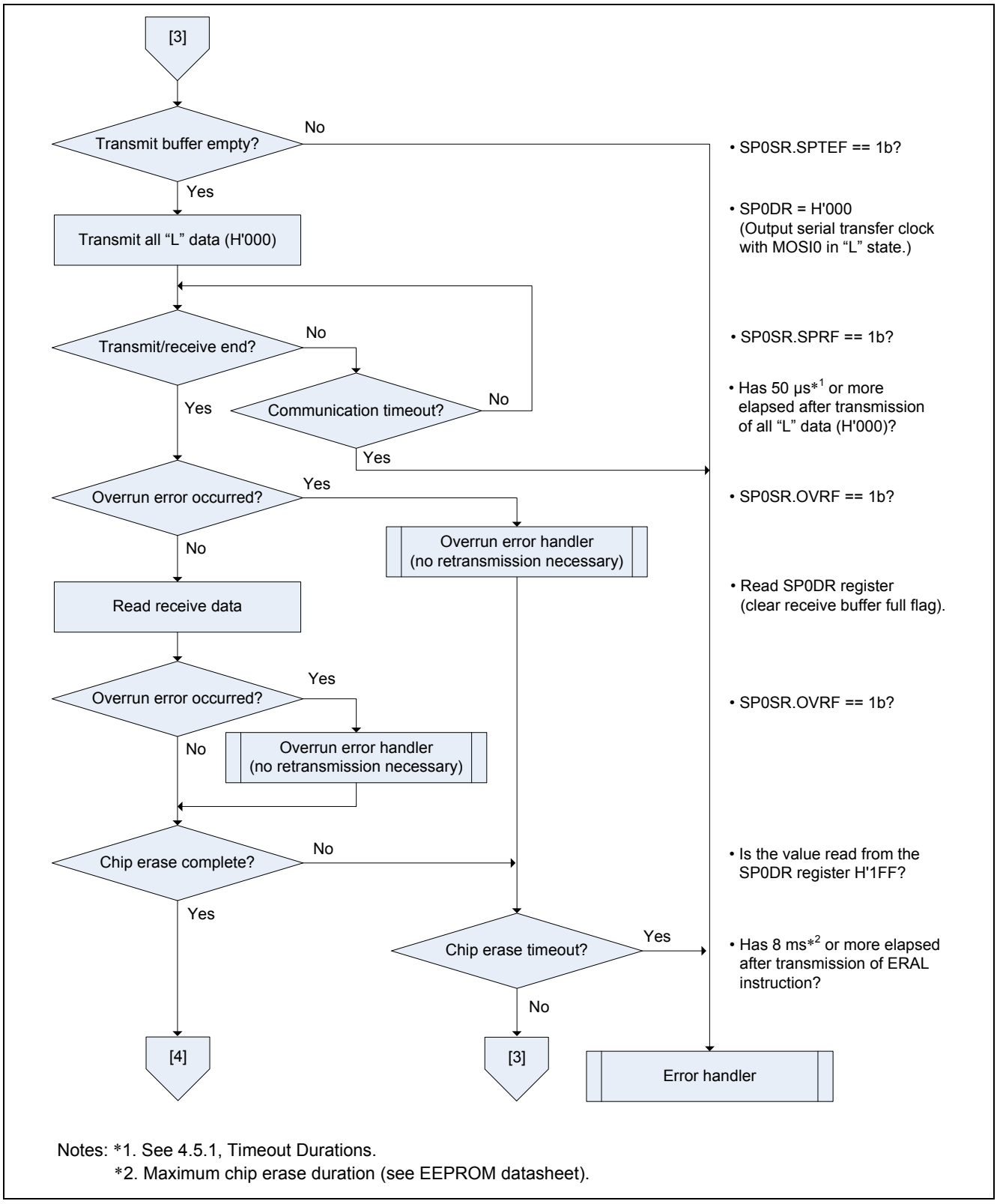
[3] Waiting for chip erase to complete

The function waits for erasure of the EEPROM chip to complete. An overview of the operation is shown below. Figure 4.9 is a flowchart of the function.

- All “L” data (H'0000)*[1] is transmitted to the EEPROM.
- The SSL00 signal is asserted (“H” output) during transmission of all “L” data (H'000).*[1]
- After transmission of all “L” data (H'000),*[1] the SSL00 signal is negated (“L” output).
- When the chip erase is in progress, data other than all “H” data (H'1FF)*[2] is received from the EEPROM.
- After the chip erase is completed, all “H” data (H'1FF)*[2] is received from the EEPROM.
- The same RSPI transfer format shown in table 4.7 of “[1] Transmission of write enable instruction (EWEN instruction)” is used to transmit all “L” data (H'000).*[1]

Notes: *1. The data (H'000) transmitted when checking the EEPROM’s chip erase state is the value specified in the EEPROM datasheet.

*2. The EEPROM outputs “L” data when the chip erase is in progress and “H” data after the chip erase is completed.



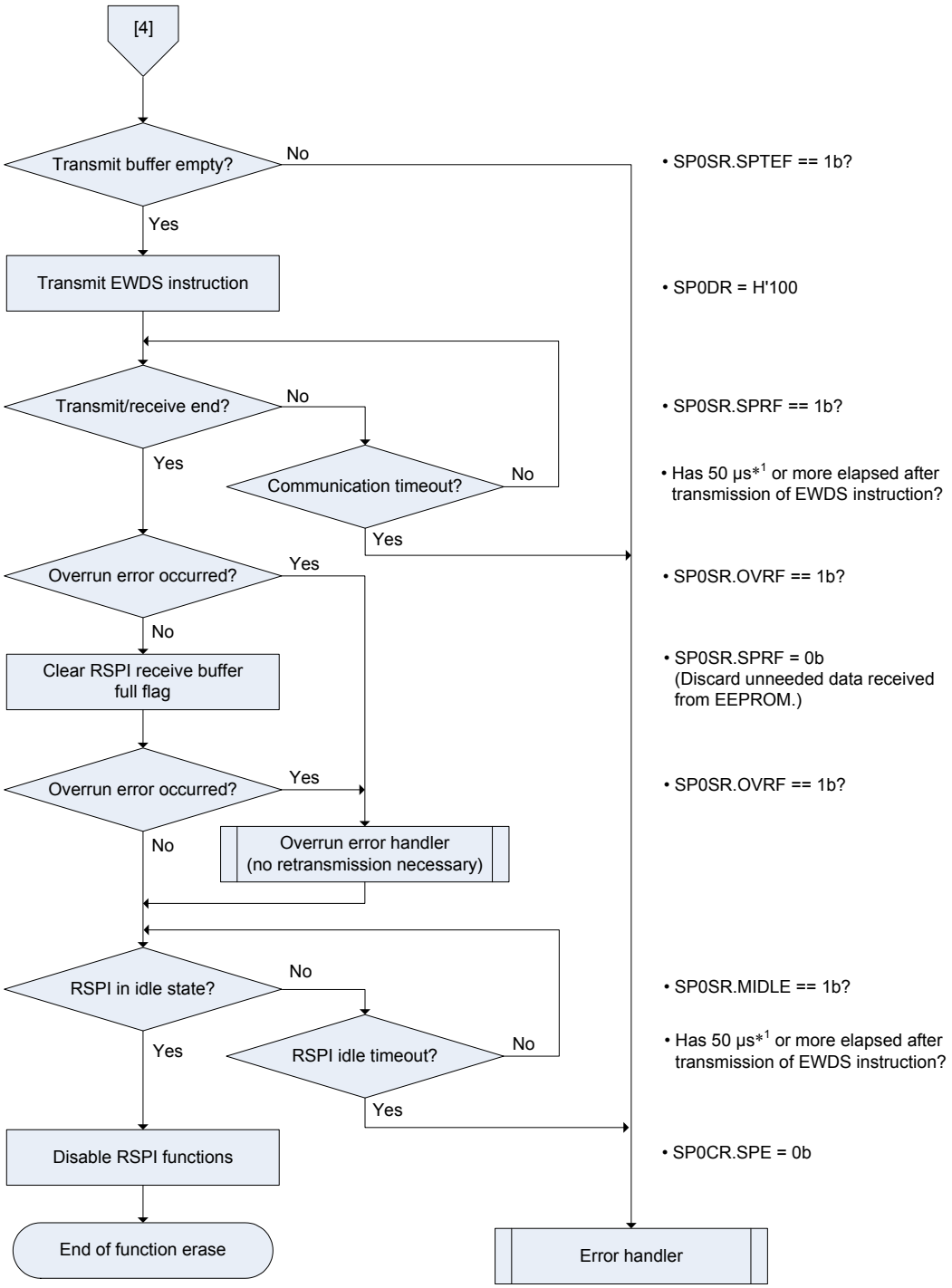
Notes: *1. See 4.5.1, Timeout Durations.
*2. Maximum chip erase duration (see EEPROM datasheet).

Figure 4.9 Function erase (3/4)

[4] Transmission of write disable instruction (EWDS instruction)

To disable writing to the EEPROM, the function transmits a write disable instruction (EWDS instruction) to the EEPROM. An overview of the operation is shown below. Figure 4.10 is a flowchart of the function.

- An EWDS instruction is transmitted.
- The SSL00 signal is asserted ("H" output) during transmission of the EWDS instruction.
- After transmission of the EWDS instruction, the SSL00 signal is negated ("L" output).
- The same RSPI transfer format shown in table 4.7 of "[1] Transmission of write enable instruction (EWEN instruction)" is used to transmit the EWDS instruction.



Note: *1. See 4.5.1, Timeout Durations.

Figure 4.10 Function erase (4/4)

4.4 Function write

Table 4.8 shows an overview of function write, and figure 4.11 is an RSPI timing chart overview of the function. Before using this function, it is necessary to execute function rspi_init once. User-defined data (H'A5A5 in the sample code; this value has no particular significance) is written to the entire address space of the EEPROM.

Table 4.8 Overview of Function write

Function	Arguments	Return values	Description
write	Write data address	None	Writes the write data stored at the address specified by the argument sequentially to the entire address space of the EEPROM, from the start address (H'00) to the end address (H'3F).

Function write performs the following five processes:

[1] Transmission of the write enable instruction (EWEN instruction)
[2] Transmission of the data write instruction (WRITE instruction)
[3] Transmission of the write data
[4] Waiting for data write to complete
[5] Transmission of write disable instruction (EWDS instruction)

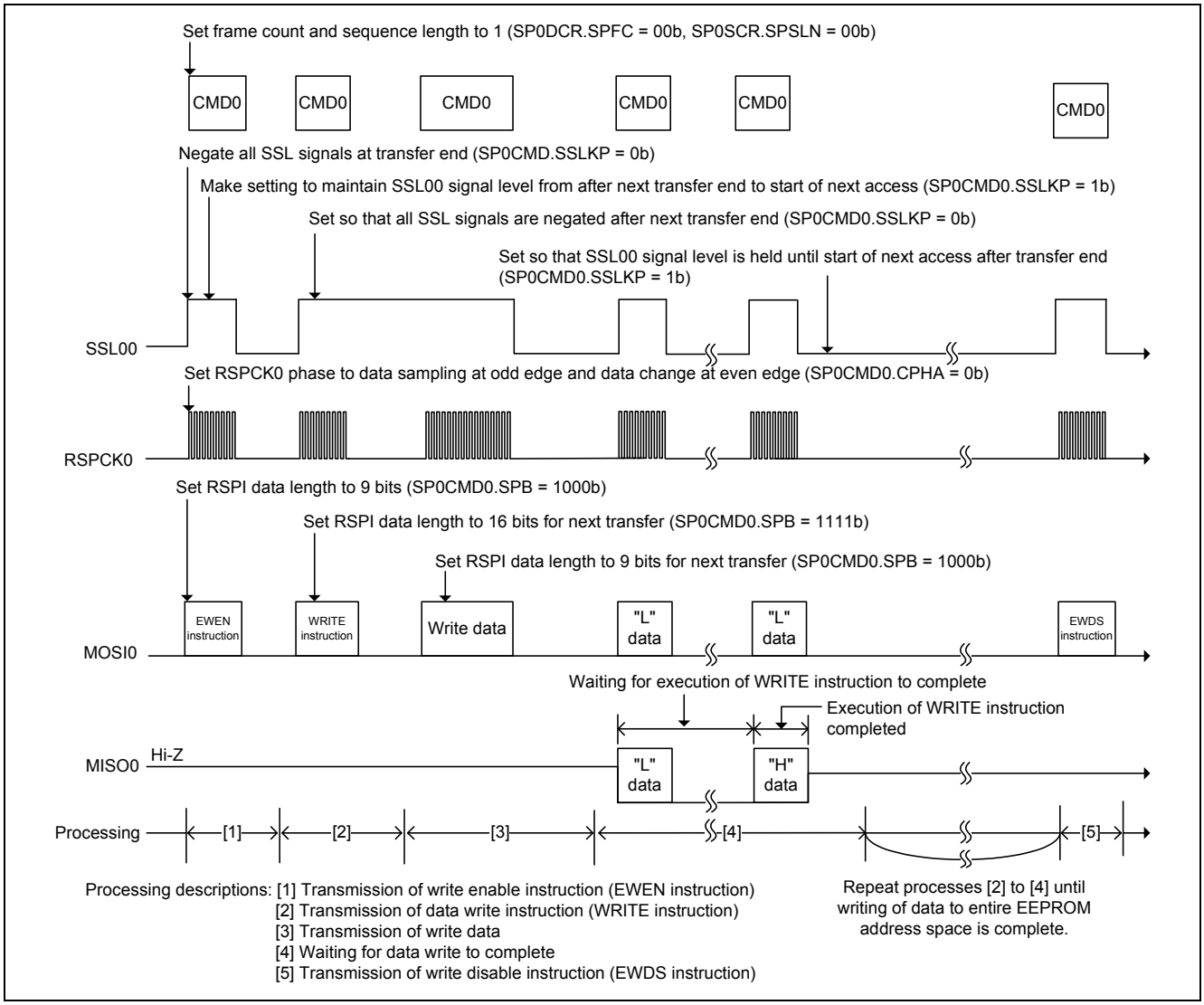


Figure 4.11 RSPI Timing Chart Overview of Function write

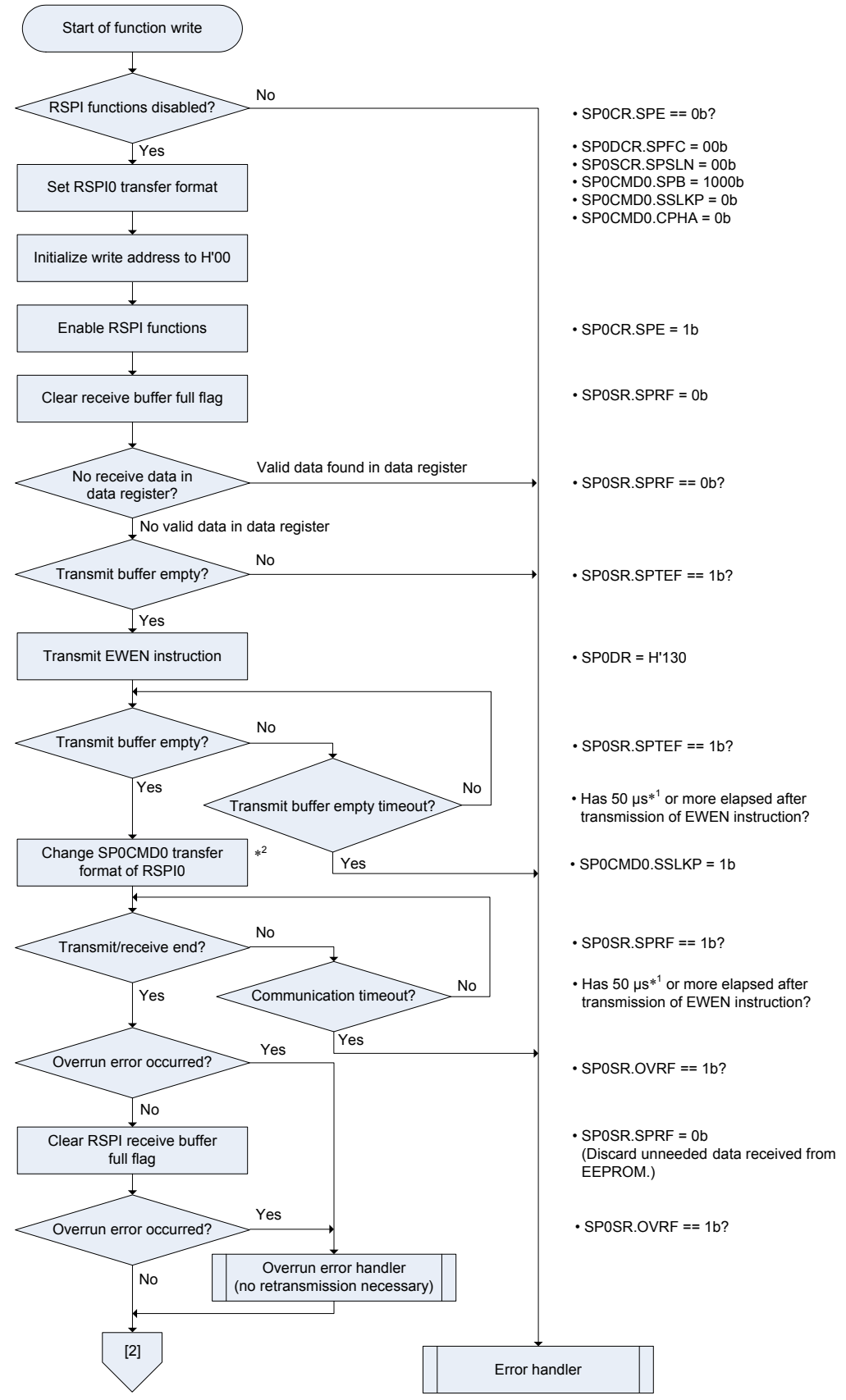
[1] Transmission of write enable instruction (EWEN instruction)

To enable writing to the EEPROM, the function transmits a write enable instruction (EWEN instruction) to the EEPROM. An overview of the operation is shown below. Figure 4.12 is a flowchart of the function.

- An EWEN instruction is transmitted to the EEPROM.
- The SSL00 signal is asserted ("H" output) during transmission of the EWEN instruction.
- After transmission of the EWEN instruction, the SSL00 signal is negated ("L" output).
- The RSPI transfer format shown in table 4.9 is used to transmit the EWEN instruction.
- During transmission of the EWEN instruction, the RSPI transfer format (table 4.10) is set for "[2] Transmission of data write instruction (WRITE instruction)."

Table 4.9 RSPI Transfer Format for Transmission of the EWEN Instruction

Item		Register	Setting Value	Description
Frame count		SP0DCR.SPFC	00b	Set frame count to 1.
Sequence length		SP0SCR.SPSLN	00b	Set sequence length to 1.
SP0CMD0	Data length	SP0CMD0.SPB	1000b	Set data length to 9 bits.
	SSL signal level hold	SP0CMD0.SSLKP	0b	Negate all SSL signals at transfer end.
	RSPCK phase	SP0CMD0.CPHA	0b	Data change: Even edge (falling edge) Data sampling: Odd edge (rising edge)



Notes: *1. See 4.5.1, Timeout Durations.
*2. During communication, the RSPI transfer format is set for "[2] Transmission of data write instruction (WRITE instruction)."

Figure 4.12 Function write (1/6)

[2] Transmission of data write instruction (WRITE instruction)

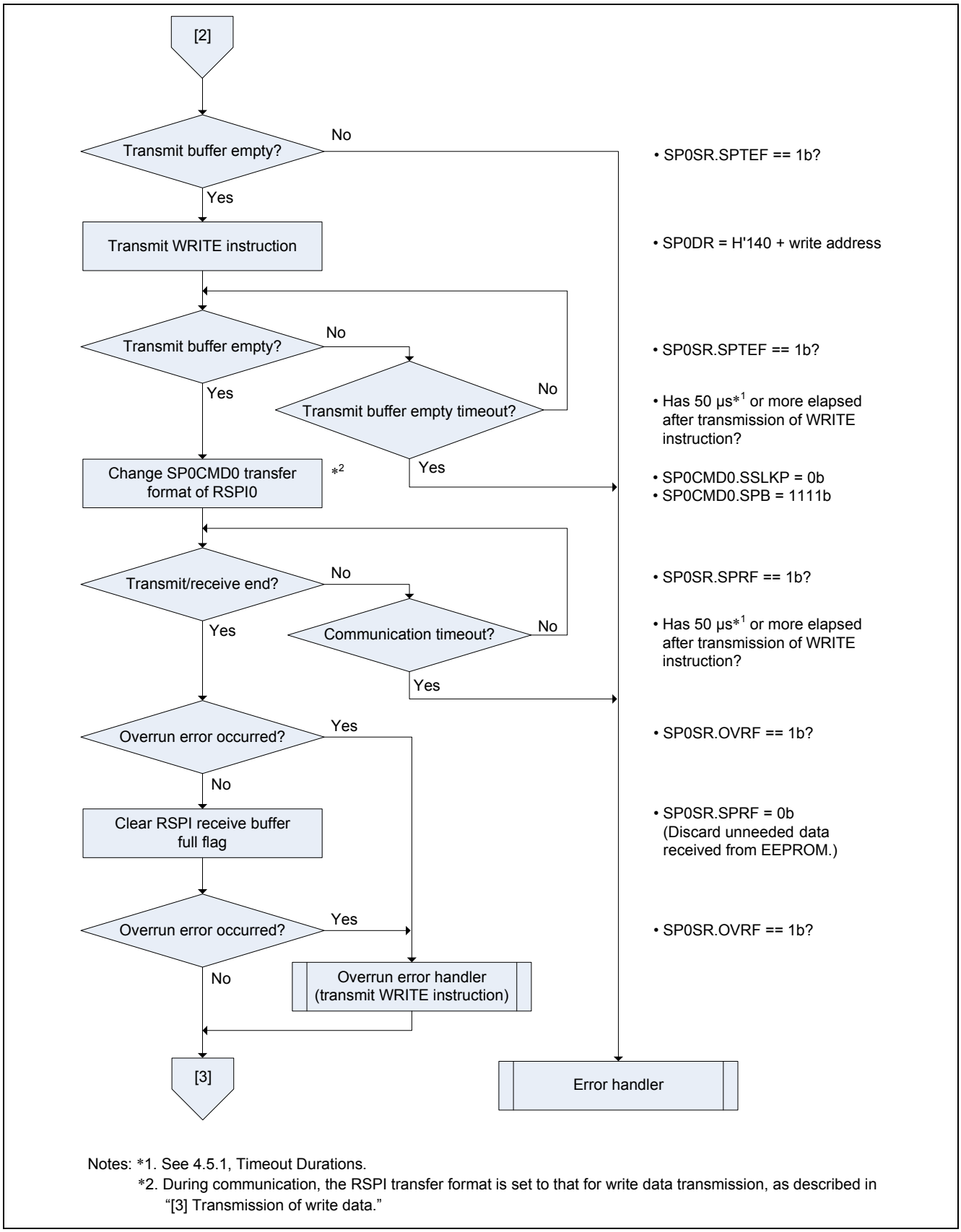
To write data to the EEPROM, the function transmits a data write instruction (WRITE instruction) to the EEPROM. An overview of the operation is shown below. Figure 4.13 is a flowchart of the function.

- A WRITE instruction is transmitted to the EEPROM.
- Each time 16-bit data is write completed, the write address increments by 1 with the specified write address changing sequentially from the start address of the EEPROM (H'00) to the end address (H'3F).
- The SSL00 signal is asserted ("H" output) during transmission of the WRITE instruction.
- The SSL00 signal continues to be asserted ("H" output) even after transmission of the WRITE instruction completes.
- The RSPI transfer format shown in table 4.10 is used to transmit the WRITE instruction. (Items that differ from tables 4.9 and 4.12 are shown in **bold** text.)
- During transmission of the WRITE instruction, the RSPI transfer format (table 4.11) is set for "[3] Transmission of write data."

Table 4.10 RSPI Transfer Format for Transmission of the WRITE Instruction*[1]

Item		Register	Setting Value	Description
Frame count		SP0DCR.SPFC	00b	Set frame count to 1.
Sequence length		SP0SCR.SPSLN	00b	Set sequence length to 1.
SP0CMD0	Data length	SP0CMD0.SPB	1000b	Set data length to 9 bits.
	SSL signal level hold	**SP0CMD0.SSLKP**	**1b**	**Hold SSL00 signal level after transfer end to start of next access.**
	RSPCK phase	SP0CMD0.CPHA	0b	Data change: Even edge (falling edge) Data sampling: Odd edge (rising edge)

Note: *1. This RSPI transfer format is set in "[1] Transmission of write enable instruction (EWEN instruction)" and "[4] Waiting for data write to complete."



Notes: *1. See 4.5.1, Timeout Durations.
*2. During communication, the RSPI transfer format is set to that for write data transmission, as described in "[3] Transmission of write data."

Figure 4.13 Function write (2/6)

[3] Transmission of write data

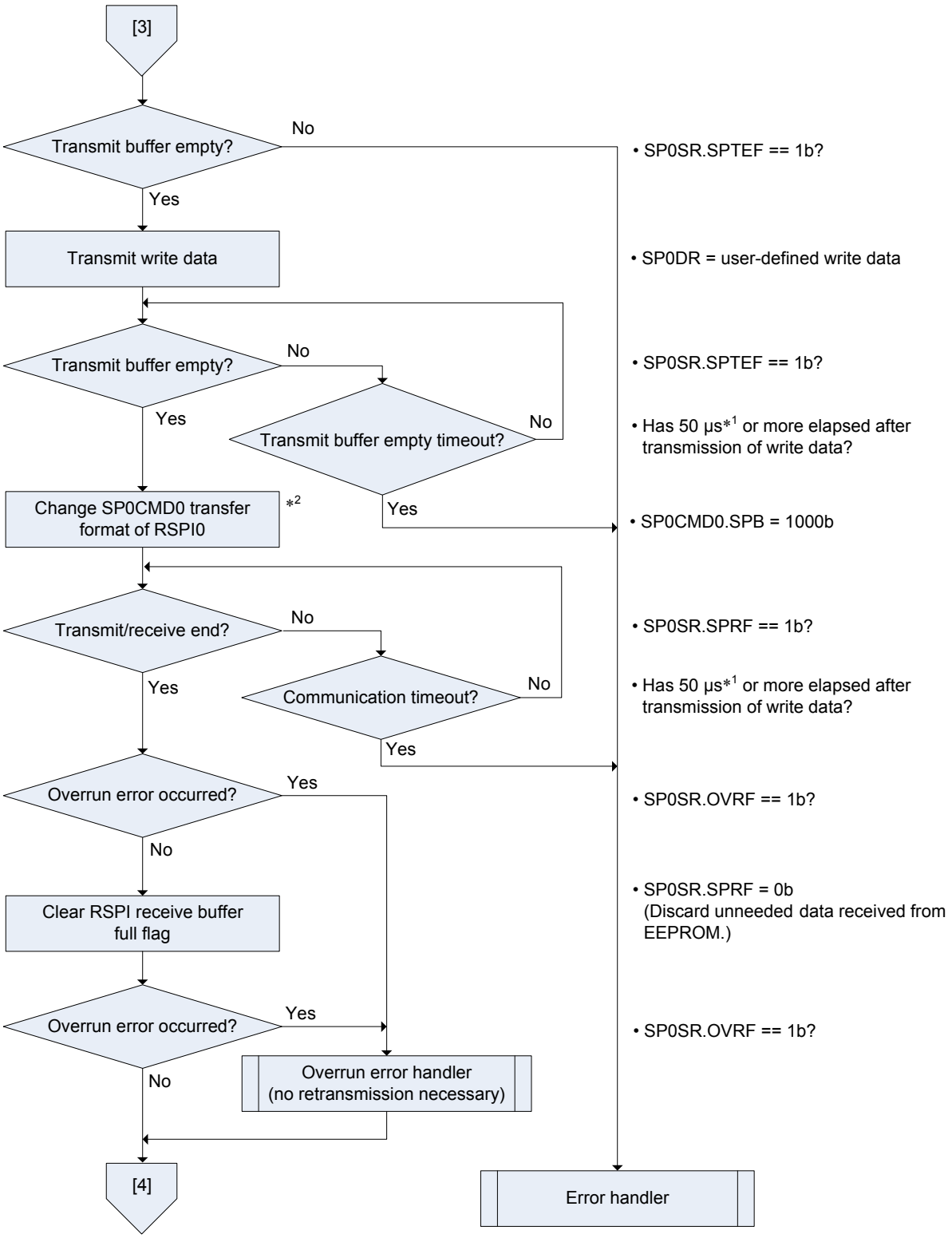
The function transmits the write data to the EEPROM. An overview of the operation is shown below. Figure 4.14 is a flowchart of the function.

- 16 bits of user-defined write data (H'A5A5 in the sample code) is written to the EEPROM.
- The SSL00 signal is asserted ("H" output) during transmission of the write data.
- After transmission of the write data, the SSL00 signal is negated ("L" output).
- The RSPI transfer format shown in table 4.11 is used to transmit the write data. (Items that differ from table 4.10 are shown in **bold** text.)
- During transmission of the write data, the RSPI transfer format (table 4.12) is set for "[4] Waiting for data write to complete."

Table 4.11 RSPI Transfer Format for Transmission of the Write Data*1

Item		Register	Setting Value	Description
Frame count		SP0DCR.SPFC	00b	Set frame count to 1.
Sequence length		SP0SCR.SPSLN	00b	Set sequence length to 1.
SP0CMD0	**Data length**	**SP0CMD0.SPB**	**1111b**	**Set data length to 16 bits.**
	SSL signal level hold	**SP0CMD0.SSLKP**	**0b**	**Negate all SSL signals at transfer end.**
	RSPCK phase	SP0CMD0.CPHA	0b	Data change: Even edge (falling edge) Data sampling: Odd edge (rising edge)

Note: *1. This RSPI transfer format is set in "[2] Transmission of data write instruction (WRITE instruction)".



Notes: *1. See 4.5.1, Timeout Durations.
*2. During communication, the RSPI transfer format is set to that for waiting for the data write to complete, as described in "[4] Waiting for data write to complete."

Figure 4.14 Function write (3/6)

[4] Waiting for data write to complete

The function waits for writing of data to the EEPROM to complete. An overview of the operation is shown below. Figures 4.15 and 4.16 are flowcharts of the function.

- All “L” data (H'000)*[1] is transmitted to the EEPROM.
- The SSL00 signal is asserted (“H” output) during transmission of all “L” data (H'000).*[1]
- After transmission of all “L” data, the SSL00 signal is negated (“L” output).
- During writing of the data, data other than all “H” data (H'1FF)*[2] is received from the EEPROM.
- After writing of the data is completed, all “H” data (H'1FF)*[2] is received from the EEPROM.
- The RSPI transfer format shown in table 4.12 is used while waiting for the data write to complete. (Items that differ from table 4.11 are shown in **bold** text.)
- When data writing continues to the next address, the RSPI transfer format (table 4.10) is set for “[2] Transmission of data write instruction (WRITE instruction)” after waiting for the data write to complete.

Notes: *1. The data (H'000) transmitted when checking the EEPROM’s data write state is the value specified in the EEPROM datasheet.
*2. The EEPROM outputs “L” data when data writing is in progress and “H” data after the data write completes.

Table 4.12 RSPI Transfer Format for Waiting for the Data Write to Complete*[1]

Item		Register	Setting Value	Description
Frame count		SP0DCR.SPFC	00b	Set frame count to 1.
Sequence length		SP0SCR.SPSLN	00b	Set sequence length to 1.
SP0CMD0	**Data length**	**SP0CMD0.SPB**	**1000b**	**Set data length to 9 bits.**
	SSL signal level hold	SP0CMD0.SSLKP	0b	Negate all SSL signals at transfer end.
	RSPCK phase	SP0CMD0.CPHA	0b	Data change: Even edge (falling edge) Data sampling: Odd edge (rising edge)

Note: *1. This RSPI transfer format is set in “[3] Transmission of write data.”

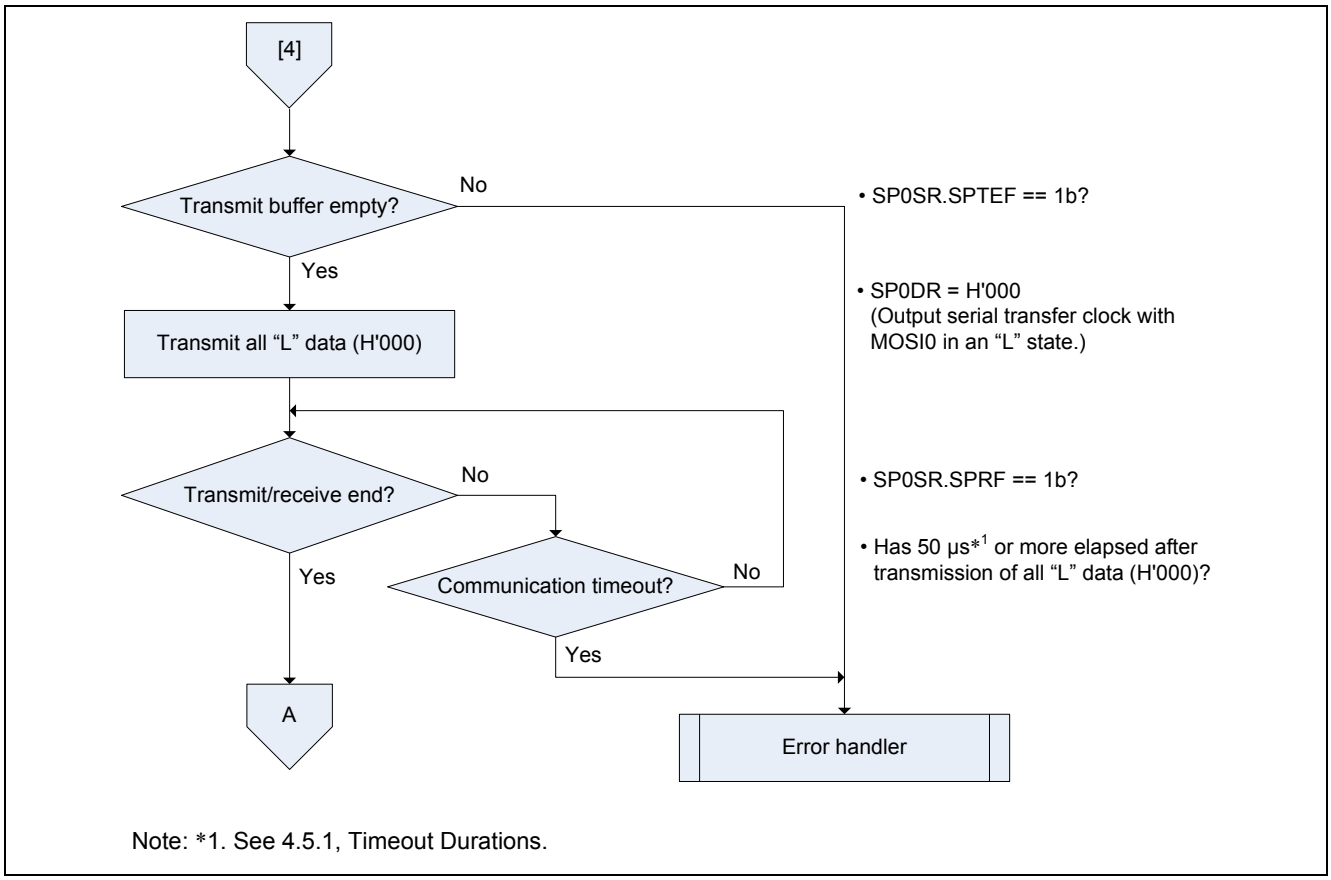
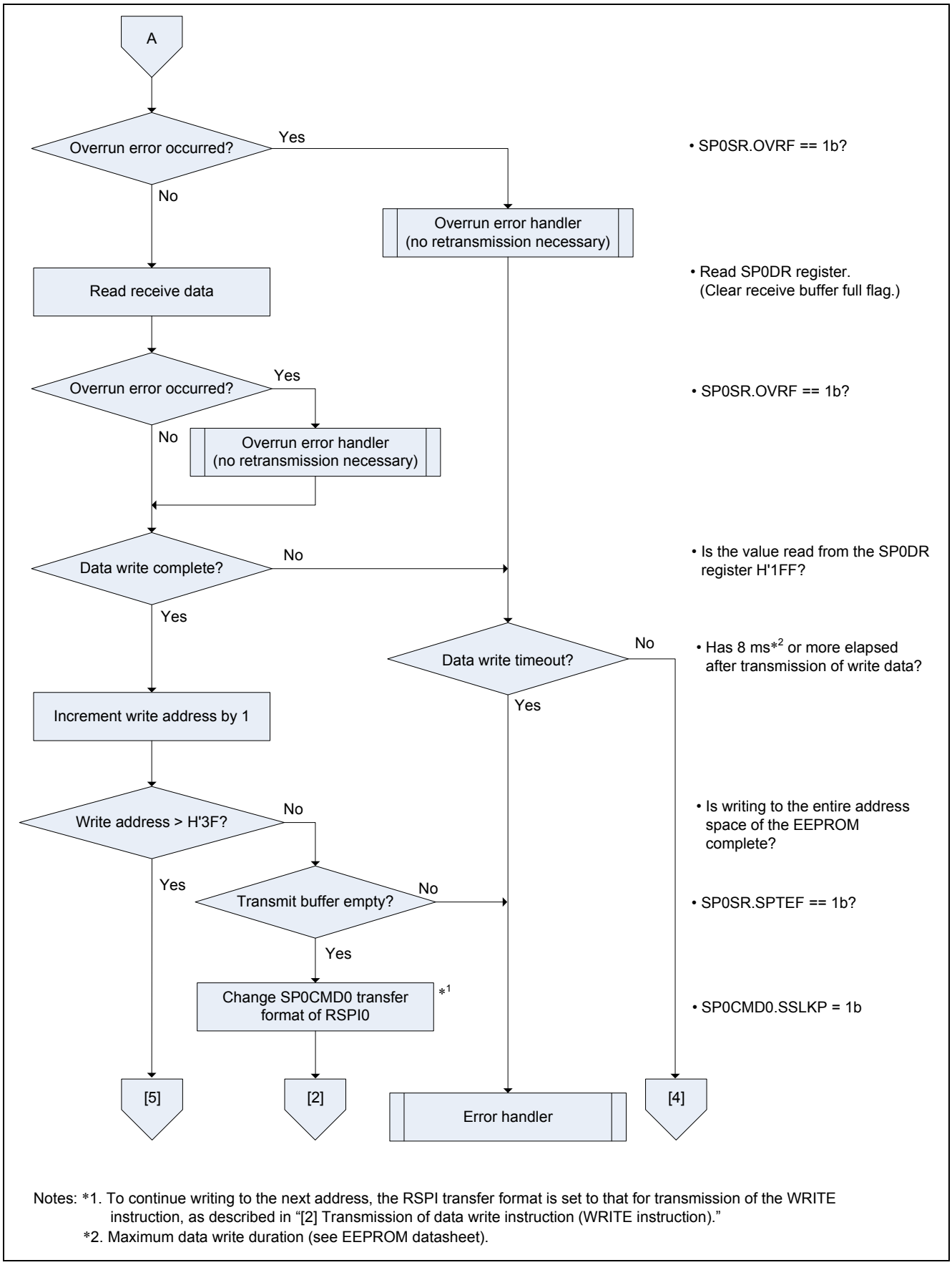


Figure 4.15 Function write (4/6)



Notes: *1. To continue writing to the next address, the RSPI transfer format is set to that for transmission of the WRITE instruction, as described in "[2] Transmission of data write instruction (WRITE instruction)."
*2. Maximum data write duration (see EEPROM datasheet).

Figure 4.16 Function write (5/6)

[5] Transmission of write disable instruction (EWDS instruction)

To disable writing to the EEPROM, the function transmits a write disable instruction (EWDS instruction) to the EEPROM. An overview of the operation is shown below. Figure 4.17 is a flowchart of the function.

- An EWDS instruction is transmitted.
- The SSL00 signal is asserted ("H" output) during transmission of the EWDS instruction.
- After transmission of the EWDS instruction, the SSL00 signal is negated ("L" output).
- The RSPI transfer format shown in table 4.13 is used to transmit the EWDS instruction.

Table 4.13 RSPI Transfer Format for Transmission of the EWDS Instruction*1

Item		Register	Setting Value	Description
Frame count		SP0DCR.SPFC	00b	Set frame count to 1.
Sequence length		SP0SCR.SPSLN	00b	Set sequence length to 1.
SP0CMD0	Data length	SP0CMD0.SPB	1000b	Set data length to 9 bits.
	SSL signal level hold	SP0CMD0.SSLKP	0b	Negate all SSL signals at transfer end.
	RSPCK phase	SP0CMD0.CPHA	0b	Data change: Even edge (falling edge) Data sampling: Odd edge (rising edge)

Note: *1. This table is identical to table 4.12.

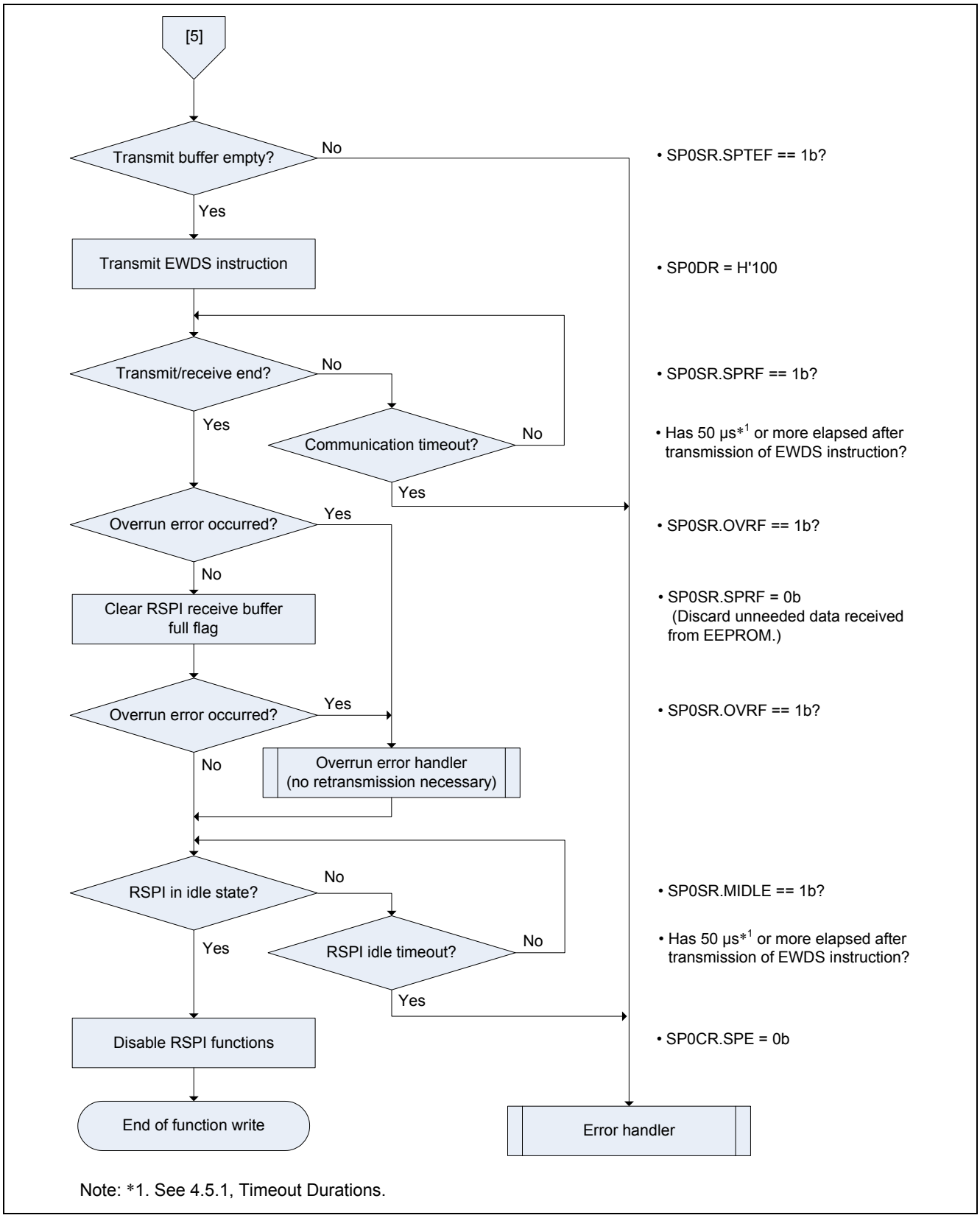


Figure 4.17 Function write (6/6)

4.5 Regarding Error Handling

The timeout durations, error handler, and overrun error handler are described below.

4.5.1 Timeout Durations

The timeout durations related to the RSPI are designed to provide sufficient time while taking into account the serial transfer clock frequency, data length, RSPCK delay, SSL negation delay, and next-access delay. When making changes to the RSPI transfer settings, the timeout durations should be modified as needed.

The value of the EEPROM chip erase and data write timeout durations are taken from the EEPROM datasheet. If the EEPROM is changed, the timeout durations should be modified after referring to the datasheet.

4.5.2 Error Handler

After disabling the RSPI functions, the error handler stores the error code in a global variable (gucErrCode) and halts the program (with an infinite loop). The error codes are listed in tables 4.14 and 4.15. Error codes not listed in these tables are reserved values. Figure 4.18 is a flowchart of the error handler.

Table 4.14 Error Codes (Upper 4 Bits)

Error Code	Process Generating Error
H'0	No error
H'1	read processing
H'2	erase processing
H'3	write processing

Table 4.15 Error Codes (Lower 4 Bits)

Error Code	Process Generating Error
H'0	No error
H'1*1	RSPI functions already enabled at start of processing of function read, function erase, or function write.
H'2	Buffer state other than “transmit buffer empty” before transmission.
H'3	Buffer state other than “transmit buffer empty,” and RSPI transfer format cannot be switched.
H'4	Cannot enter state “valid receive data in SP0DR register”.
H'5*1	RSPI cannot enter idle state.
H'6	After clearing RSPI receive buffer full flag, cannot enter state “no valid receive data in SP0DR register”.
H'7	EEPROM chip erase did not complete within the specified time.
H'8	EEPROM data write did not complete within the specified time.
H'9	After overrun error flag cleared by overrun error handler, cannot enter state “no overrun error”.
H'A	After RSPI receive buffer full flag cleared by overrun error handler, cannot enter state “no valid receive data in SP0DR register”.

Note: *1. This error is specific to the sample source code and is not listed in the hardware manual.

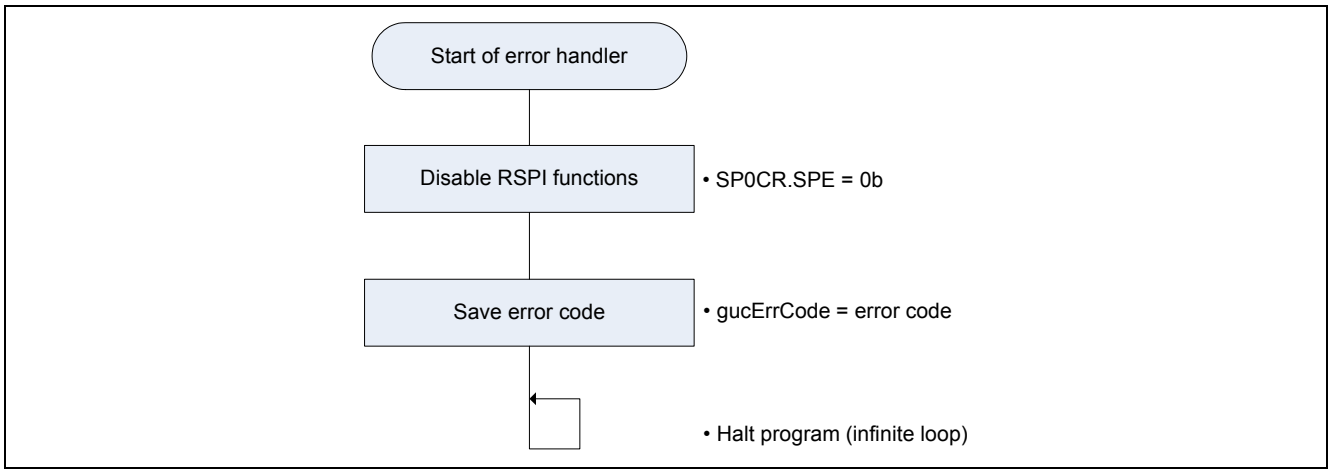


Figure 4.18 Error Handler

4.5.3 Overrun Error Handler (No Retransmission Necessary: Case Where Retransmission of EEPROM Command Instruction Not Needed)

Figure 4.19 illustrates the process for recovering from an overrun error when retransmission is not necessary.

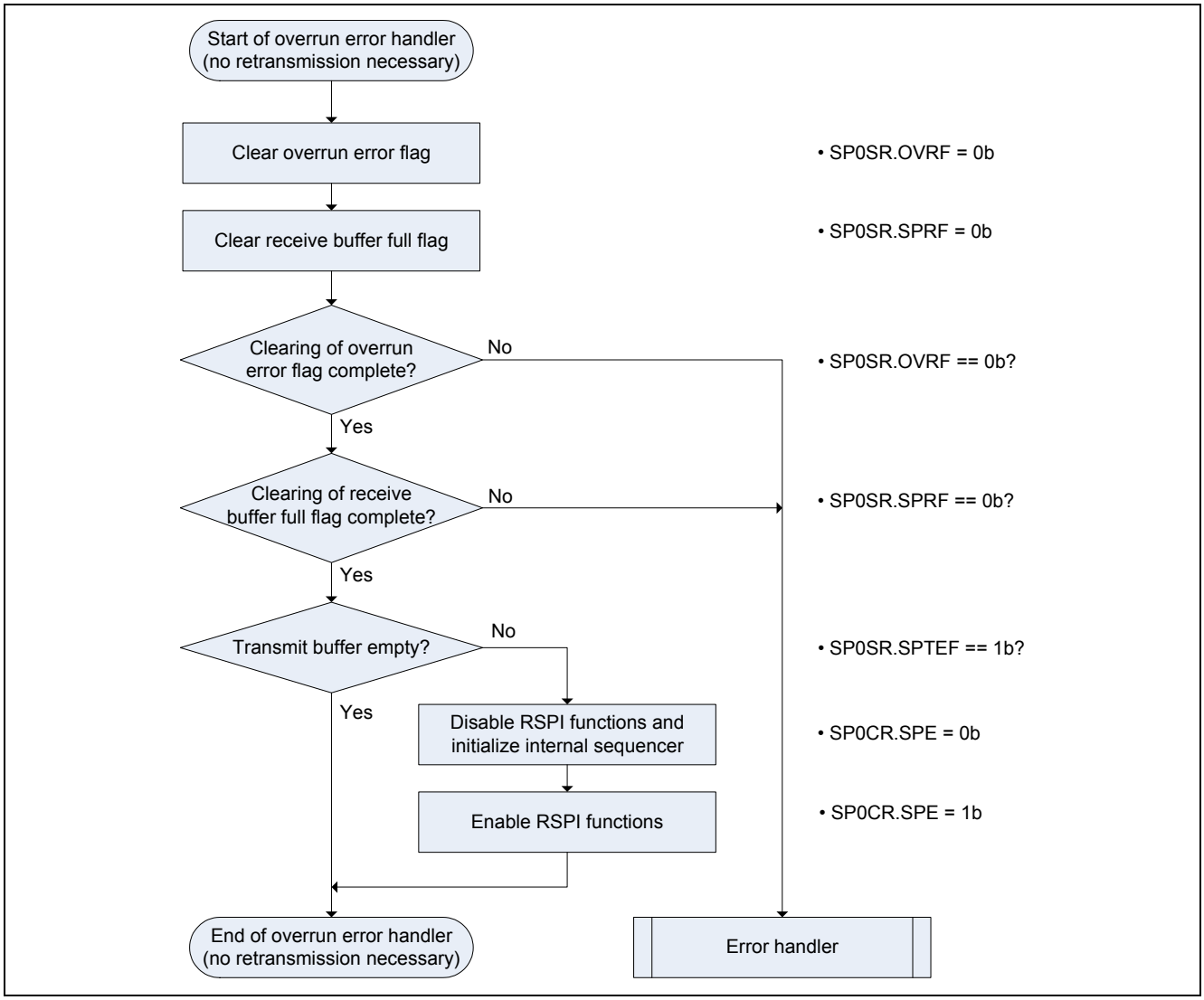
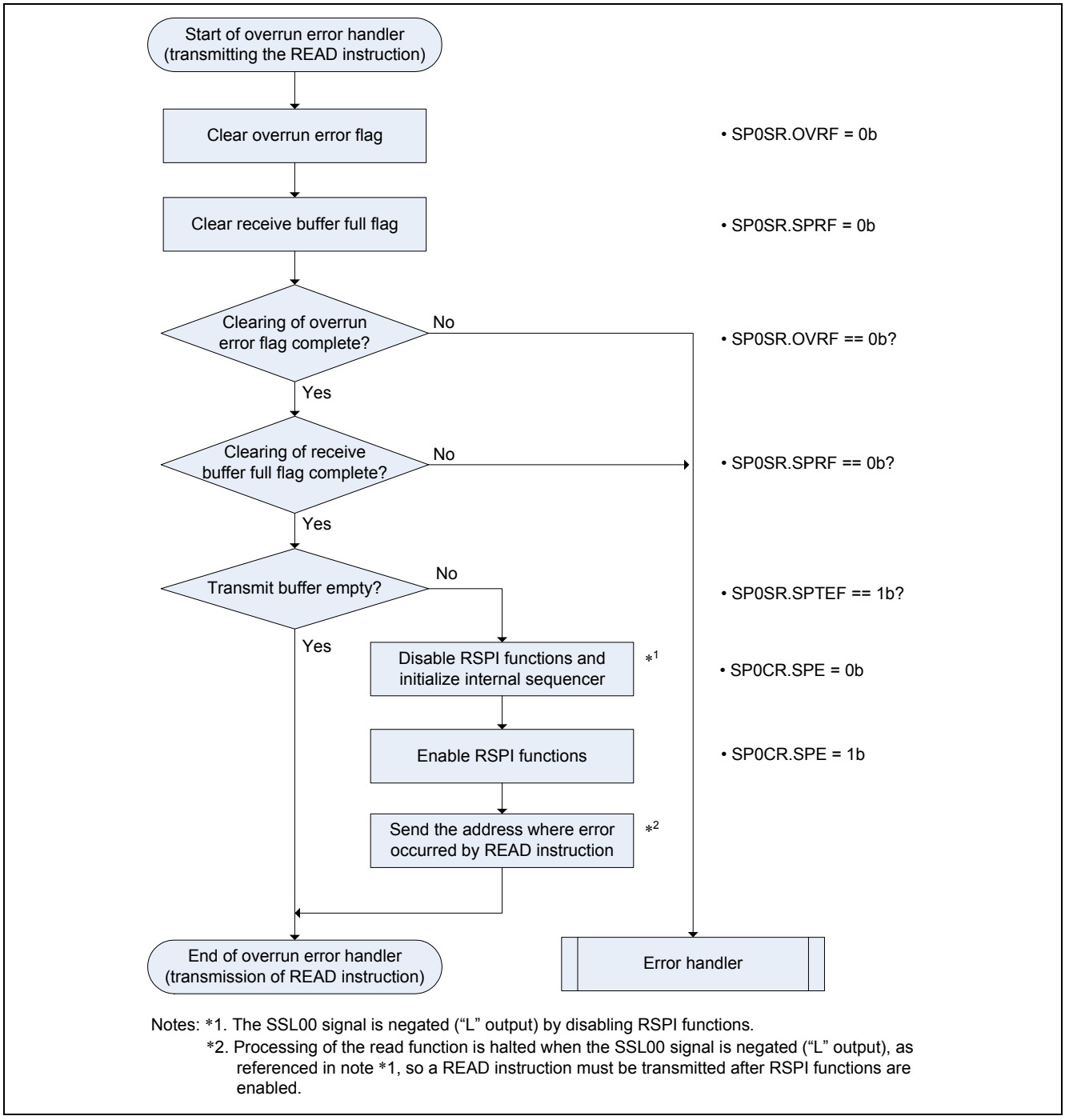


Figure 4.19 Overrun Error Handler (No Retransmission Necessary)

4.5.4 Overrun Error Handler (Transmitting the READ Instruction)

Figure 4.20 illustrates the process for recovering from an overrun error that occurs during transmission of the READ instruction.

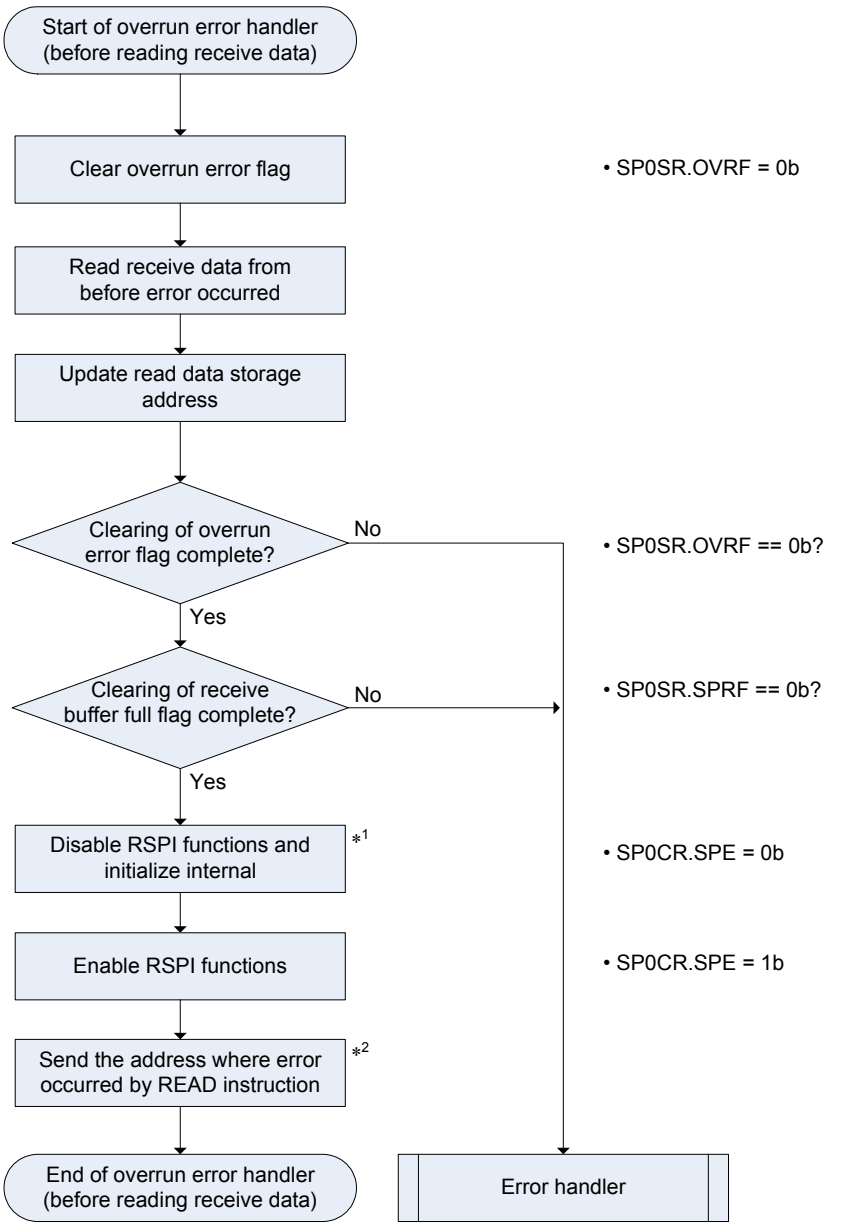


Notes: *1. The SSL00 signal is negated (“L” output) by disabling RSPI functions.
*2. Processing of the read function is halted when the SSL00 signal is negated (“L” output), as referenced in note *1, so a READ instruction must be transmitted after RSPI functions are enabled.

Figure 4.20 Overrun Error Handler (Transmission of READ Instruction)

4.5.5 Overrun Error Handler (Before Reading Receive Data)

Figure 4.21 illustrates the process for recovering from an overrun error that occurs before reading receive data.

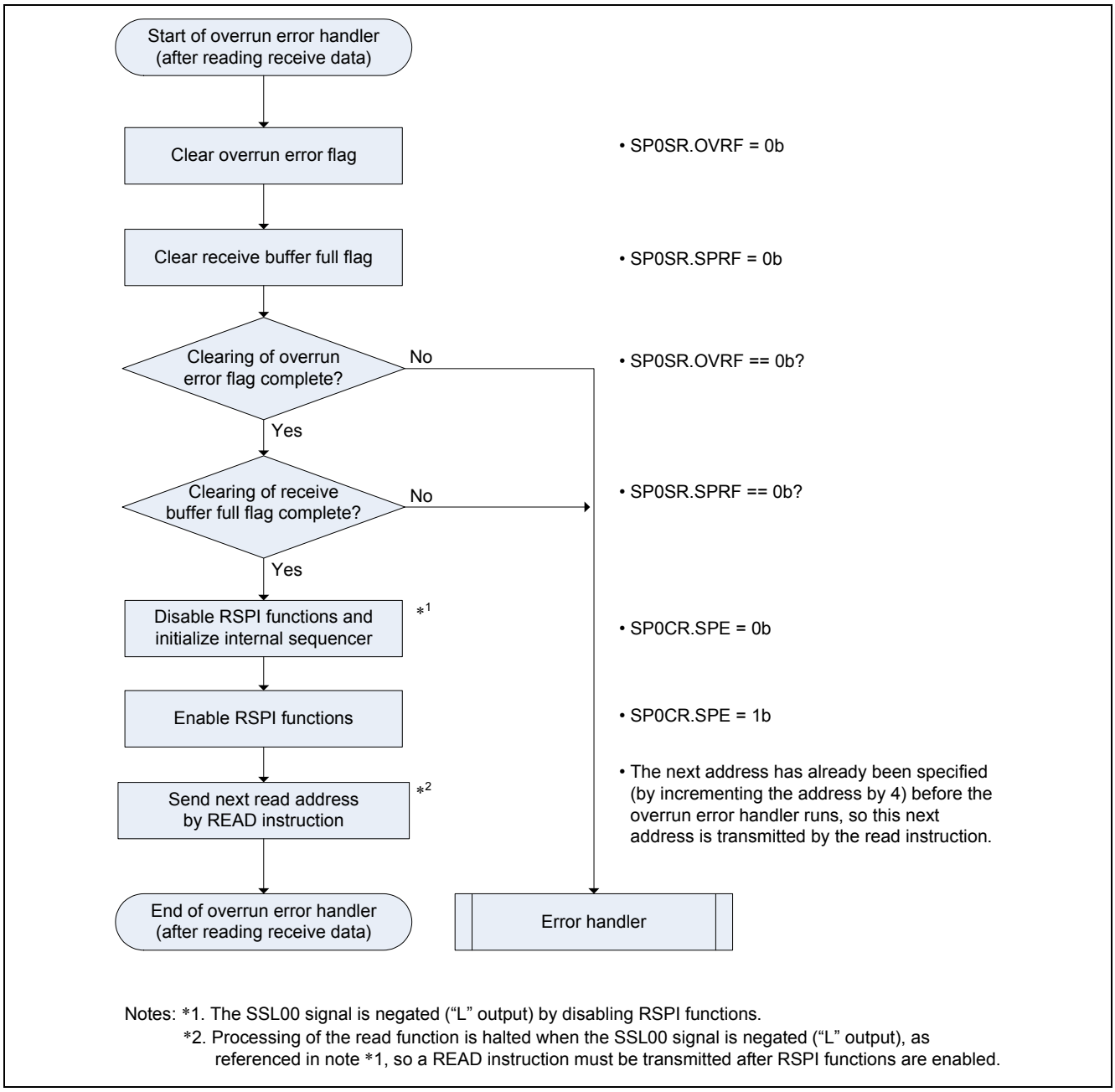


Notes: *1. The SSL00 signal is negated (“L” output) by disabling RSPI functions.
*2. Processing of the read function is halted when the SSL00 signal is negated (“L” output), as referenced in note *1, so a READ instruction must be transmitted after RSPI functions are enabled.

Figure 4.21 Overrun Error Handler (Before Reading Receive Data)

4.5.6 Overrun Error Handler (After Reading Receive Data)

Figure 4.22 illustrates the process for recovering from an overrun error that occurs after reading receive data.

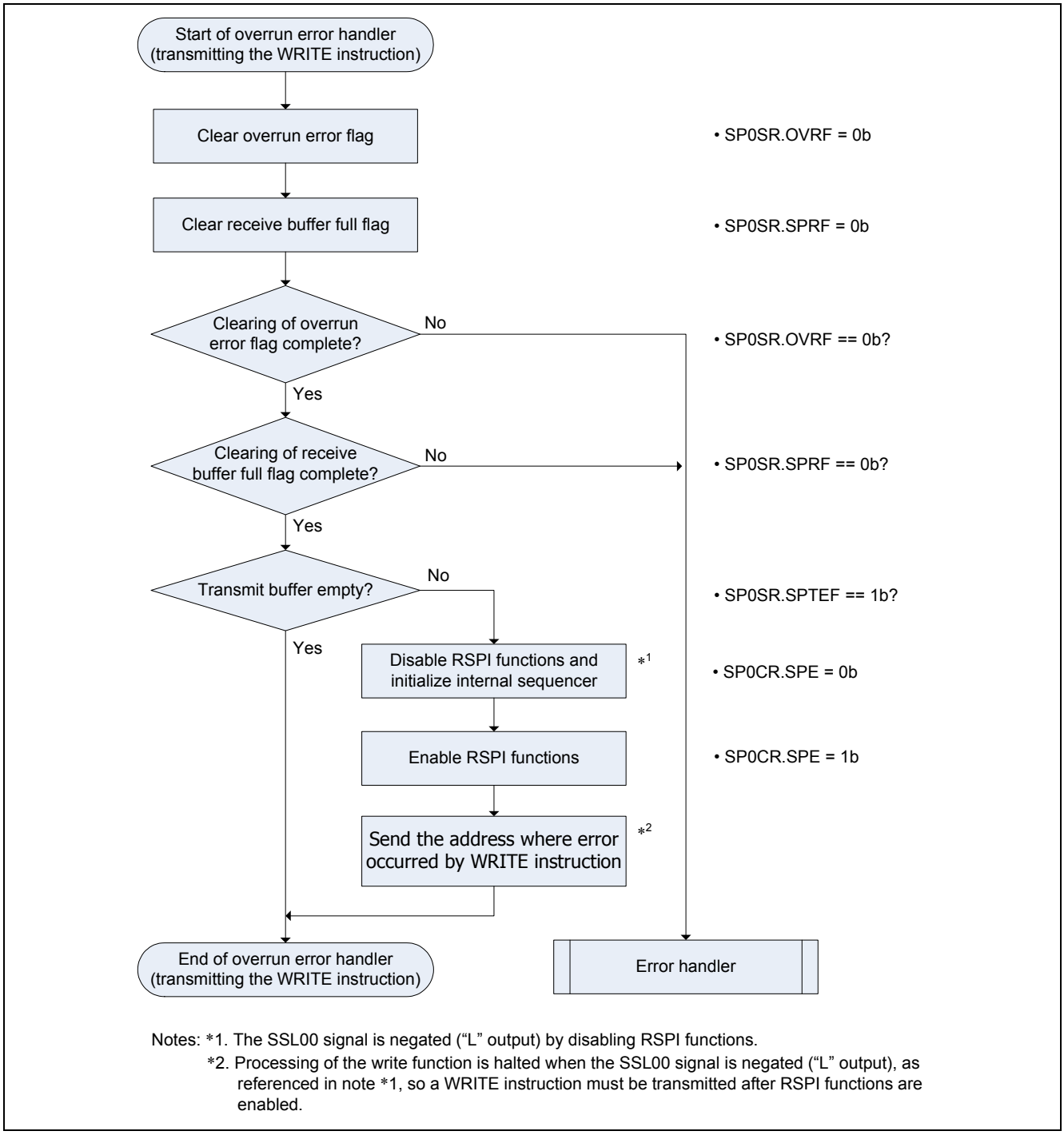


Notes: *1. The SSL00 signal is negated (“L” output) by disabling RSPI functions.
*2. Processing of the read function is halted when the SSL00 signal is negated (“L” output), as referenced in note *1, so a READ instruction must be transmitted after RSPI functions are enabled.

Figure 4.22 Overrun Error Handler (After Reading Receive Data)

4.5.7 Overrun Error Handler (Transmitting the WRITE Instruction)

Figure 4.23 illustrates the process for recovering from an overrun error that occurs when transmitting of the WRITE instruction.



Notes: *1. The SSL00 signal is negated (“L” output) by disabling RSPI functions.
*2. Processing of the write function is halted when the SSL00 signal is negated (“L” output), as referenced in note *1, so a WRITE instruction must be transmitted after RSPI functions are enabled.

Figure 4.23 Overrun Error Handler (Transmitting the WRITE Instruction)

5. Reference Documents

SH7455 Group, SH7456 Group User's Manual: Hardware Rev.1.10 (R01UH0030EJ0110)
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

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

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

Revision History	Example of EEPROM Control Using the RSPI

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 2, 2012	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord 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 one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - "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.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. 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.
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.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

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: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, 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 Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1