R32C/100 Series

Transmitting/Receiving Data Using the Serial Bus Interface in Synchronous Serial Communication Mode

R01AN0101EJ0100
Rev. 1.00
June 29, 2012

Abstract

This document describes a program for transmitting/receiving data using the serial bus interface (SBI) in synchronous serial communication mode.

Products

R32C/120 Group
R32C/121 Group
R32C/145 Group
R32C/151 Group
R32C/152 Group
R32C/153 Group
R32C/156 Group
R32C/157 Group
R32C/160 Group
R32C/161 Group

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

This document explains how to transmit and receive data using the SBI in synchronous serial communication mode with two R32C/121 Group MCUs.
Transmission from the master device starts when the master device receives an INT0 interrupt request. 8-bit (one frame) data is transmitted from the master device three times and is received by the slave device. Then one frame of data is transmitted from the slave device three times and is received by the master device.

Table 1.1 lists the Peripheral Function and Its Application. Figure 1.1 shows a Connection Example.

Table 1.1 Peripheral Function and Its Application

Peripheral Function	Application
Serial bus interface (SBI0)	Transmit and receive data

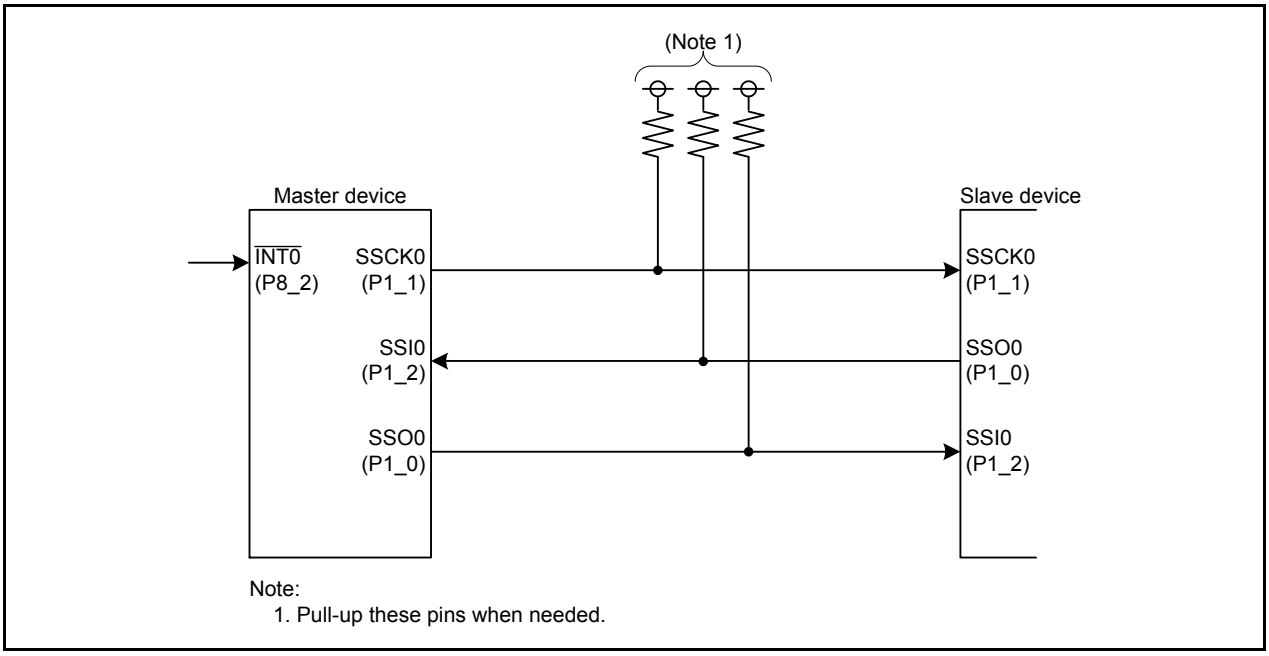


Figure 1.1 Connection Example

Figure 1.2 shows the Sample Code Outline.

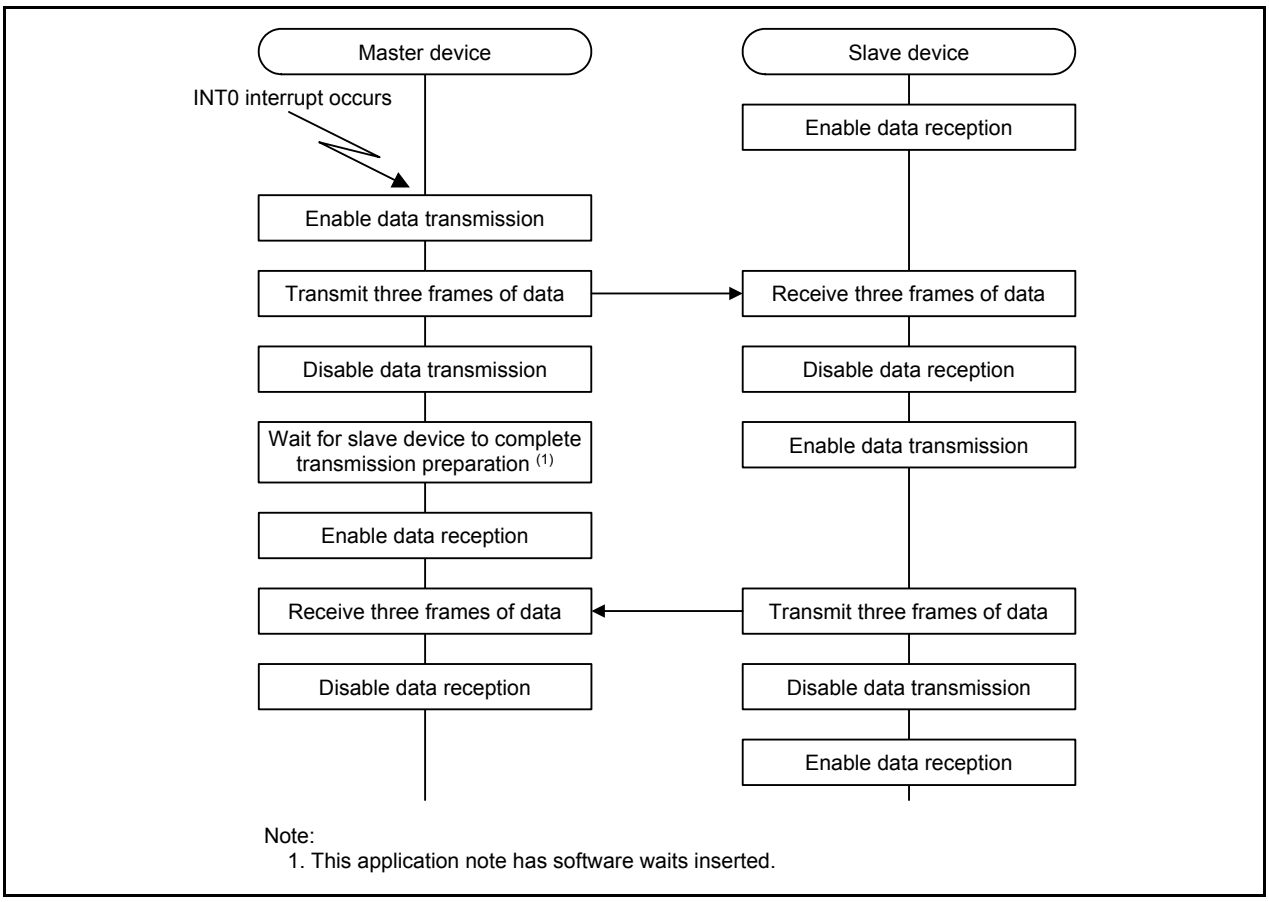


Figure 1.2 Sample Code Outline

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	R5F64219JFB (R32C/121 Group)
Operating frequencies	• Main clock: 8 MHz • PLL clock: 128 MHz • Base clock: 64 MHz • CPU clock: 64 MHz • Peripheral bus clock: 32 MHz • Peripheral function clock: 32 MHz
Operating voltage	5 V
Integrated development environment	Renesas Electronics Corporation High-performance Embedded Workshop Version 4.08
C compiler	Renesas Electronics Corporation R32C/100 Series C Compiler V.1.02 Release 01
	Compile options -D__STACKSIZE__=0X300 -D__ISTACKSIZE__=0X300 -DVECTOR_ADR=0x0FFFFFBDC -c -finfo -dir "$(CONFIGDIR)"
Operating mode	Single-chip mode
Sample code version	Version 1.00

3. Reference Application Note

The application note associated with this application note is listed below. Refer to this application note for additional information.

- R32C/100 Series Configuring PLL Mode (REJ05B1221)

4. Master Device Hardware

4.1 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
P1_0/SSO0	Output	Output serial data
P1_1/SSCK0	Output	Output clock
P1_2/SSI0	Input	Input serial data
P8_2/$\overline{\text{INT0}}$	Input	Input INT0 interrupt

5. Master Device Software

Transmission from the master device starts when the master device receives an INT0 interrupt request. One frame of data is transmitted from the master device three times. Then one frame of data is transmitted from the slave device three times.
The master device settings are below.

Serial bus interface 0 settings
- Transmit/receive clock: 1 MHz (peripheral bus clock (fBCLK) divided by 32)
- Clock polarity: High when clock is stopped
- Data transfer direction: MSB first

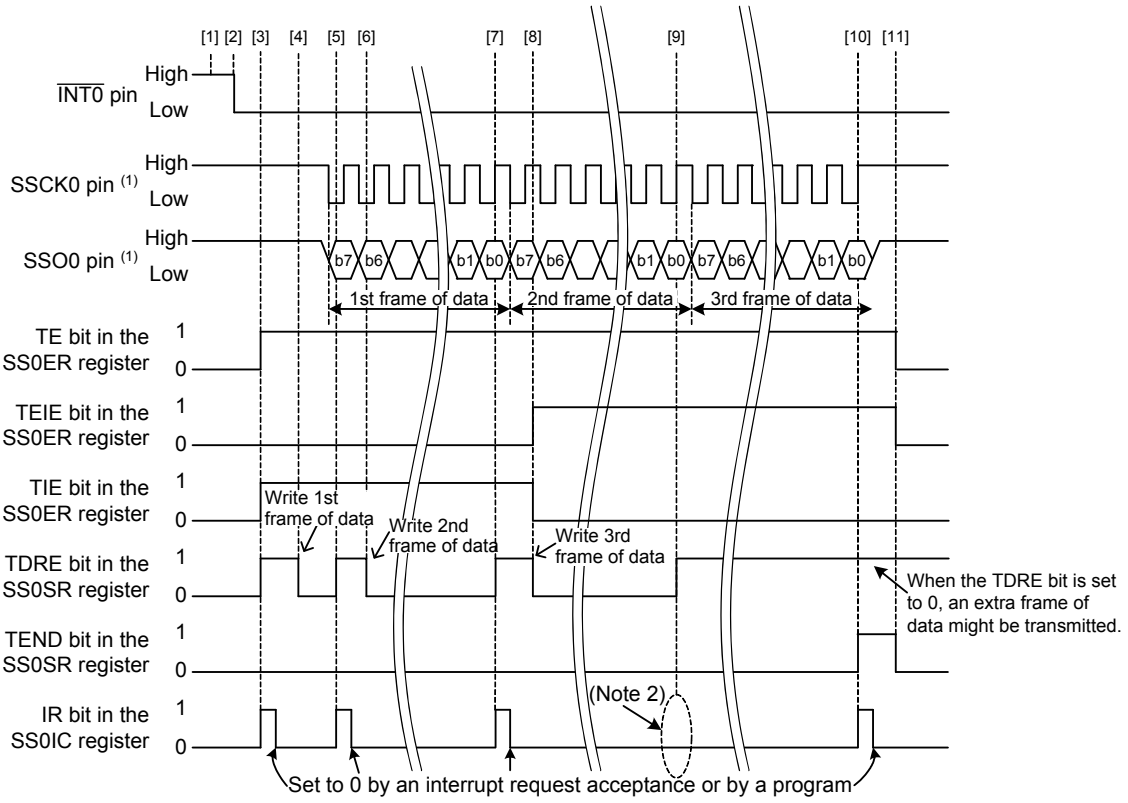
Interrupt settings
- Serial bus interface 0 interrupt
  Interrupt priority level: 2
  Interrupt request sources: Transmit data register empty, transmit end
- INT0 interrupt
  Interrupt priority level: 1
  Interrupt request source: Falling edge

5.1 Operation Overview

5.1.1 Master Transmission

(1) Perform the initial setting of the serial bus interface (SBI0).
(2) When a falling edge is input to the $\overline{\text{INT0}}$ pin, an INT0 interrupt occurs.
(3) In the INT0 interrupt handling, set the TE bit in the SS0ER register to 1 (enable data transmission) and set the TIE bit to 1 (enable transmit data register empty interrupt request). When setting these bits, the TDRE bit in the SS0SR register becomes 1 (no data left in the SS0TDR register) and a transmit data register empty interrupt occurs.
(4) In the transmit data register empty interrupt handling, write the first frame of data to the SS0TDR register. During the write operation, the TDRE bit becomes 0 (unsent data left in the SS0TDR register) and the transmit and receive clocks are output.
(5) When the first frame of data is transferred to the transmit/receive shift register, the TDRE bit becomes 1 and a transmit data register empty interrupt occurs.
(6) In the transmit data register empty interrupt handling, write the second frame of data to the SS0TDR register.
(7) When the second frame of data is transferred to the transmit/receive shift register, the TDRE bit becomes 1 and a transmit data register empty interrupt occurs.
(8) In the transmit data register empty interrupt handling, set the TIE bit in the SS0ER register to 0 (disable transmit data register empty interrupt request), set the TEIE bit to 1 (enable transmit end interrupt request), and write the last frame of data to the SS0TDR register.
(9) When the last frame of data is transferred to the transmit/receive shift register, the TDRE bit becomes 1, but the IR bit in the SS0IC register does not become 1.
(10) When the last bit is transmitted, if the TDRE bit is 1, the TEND bit in the SS0SR register becomes 1 (transmission is completed), and a transmit end interrupt occurs.
(11) In the transmit end interrupt handling, set the TEND bit to 0 (transmission continues), set the TE bit to 0 (disable data transmission), and set the TEIE bit to 0 (disable transmit end interrupt request). Then prepare for the next reception.

Figure 5.1 shows the Master Transmission Timing Diagram.



The above diagram assumes the following:

- The SSUMS bit in the SS0MR2 register is 0 (synchronous serial communication mode).
- The BIDE bit in the SS0MR2 register is 0 (standard mode).
- The MSS bit in the SS0CRH register is 1 (master mode).
- The CPHS bit in the SS0MR register is 0 (change data at odd edges (load data at even edges)).
- The CPOS bit in the SS0MR register is 0 (high when clock is stopped).
- The MLS bit in the SS0MR register is 0 (MSB first).
- Bits CSS1 and CSS0 in the SS0MR2 register are 00b (no function as chip select).

Notes:

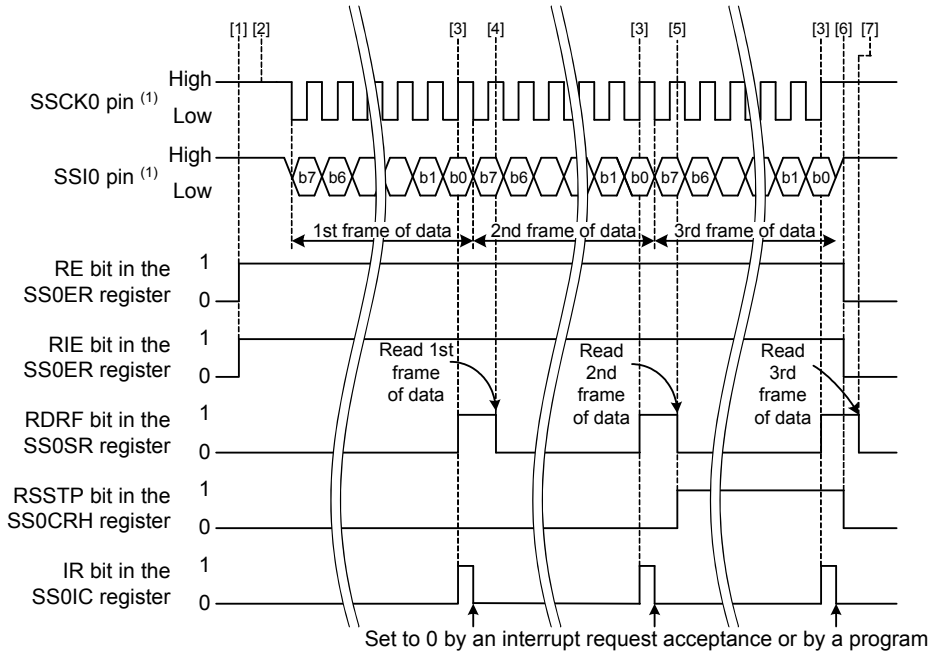
1. Example of output when pins are pulled-up.
2. The TIE bit is set to 0 (disable transmit data register empty interrupt request) in [8], so no interrupt occurs.

Figure 5.1 Master Transmission Timing Diagram

5.1.2 Master Reception

(1) At the same timing as (11) in Master Transmission, set the RE bit in the SS0ER register to 1 (enable data reception) and set the RIE bit to 1 (enable receive data register full or overrun error interrupt request).
(2) The transmit/receive clock is output by dummy reading the SS0RDR register.
(3) After receiving one frame of data, the RDRF bit in the SS0SR register becomes 1 (received data left in SS0RDR register), and the received data is stored in the SS0RDR register. When the RDRF bit becomes 1, a receive data register full interrupt occurs.
(4) In the receive data register full interrupt handling, read the received data from the SS0RDR register. When reading the received data, the RDRF bit becomes 0 (no data left in the SS0RDR register).
(5) In the receive data register full interrupt, set the RSSTP bit in the SS0CRH register to 1 (after receiving the current frame, end receive operation (stop the receive clock)) to receive the last frame of data. Read the received data from the SS0RDR register.
(6) In the receive data register full interrupt handling, before reading the last frame of data, set the RSSTP bit to 0 (after receiving the current frame, continue receive operation (output the receive clock)), the RE bit to 0 (disable data reception), and the RIE bit to 0 (disable receive data register full or overrun error interrupt request).
(7) Read the received data from the SS0RDR register.

Figure 5.2 shows the Master Reception Timing Diagram.



The above diagram assumes the following:
- The SSUMS bit in the SS0MR2 register is 0 (synchronous serial communication mode).
- The BIDE bit in the SS0MR2 register is 0 (standard mode).
- The MSS bit in the SS0CRH register is 1 (master mode).
- The CPHS bit in the SS0MR register is 0 (change data at odd edges (load data at even edges)).
- The CPOS bit in the SS0MR register is 0 (high when clock is stopped).
- The MLS bit in the SS0MR register is 0 (MSB first).
- Bits CSS1 and CSS0 in the SS0MR2 register are 00b (no function as chip select).

Note:
1. Example of output when pins are pulled-up.

Figure 5.2 Master Reception Timing Diagram

5.2 Constants

Table 5.1 lists the Constants Used in the Sample Code.

Table 5.1 Constants Used in the Sample Code

Constant Name	Setting Value	Contents
SND	00h	Transmission
RCV	01h	Reception
D_SBI_DATA_MAX	3	Transmit data size
D_SBI_DATA_LAST_F	D_SBI_DATA_MAX-1	Constant for determining the last frame of data
D_SBI_DATA_SECOND_LAST_F	D_SBI_DATA_MAX-2	Constant for determining the second to last frame of data
ORER	0000 0100b	Mask value for the overrun error flag

5.3 Variables

Table 5.2 lists the Global Variables.

Table 5.2 Global Variables

Type	Variable Name	Contents	Function Used
unsigned char	snd_cnt	Transmit counter	_int0, _serial_bus_0
unsigned char	f_snd_rcv	Transmit/receive flag	_int0, _serial_bus_0
unsigned char	snd_data[]	Buffer for storing transmit data	_serial_bus_0
unsigned char	rcv_data[]	Buffer for storing receive data	_serial_bus_0
unsigned char	rcv_cnt	Receive counter	_serial_bus_0

5.4 Functions

Table 5.3 lists the Functions.

Table 5.3 Functions

Function Name	Outline
peripheral_init	Initial setting for the peripheral functions
_serial_bus_0	Serial bus interface 0 interrupt handling
_int0	INT0 interrupt handling

5.5 Function Specifications

The following tables list the sample code function specifications.

peripheral_init	
Outline	Initial setting for the peripheral functions
Header	None
Declaration	void peripheral_init(void)
Description	Setting to use the serial bus interface as the master device.
Argument	None
Returned value	None
Remark	

_serial_bus_0	
Outline	Serial bus interface 0 interrupt handling
Header	None
Declaration	void _serial_bus_0(void)
Description	Read the transmit/receive flag to determine the source of the interrupt. Perform the handling below based on the determined source of the interrupt. *When transmitting • Transmit data register empty interrupt Write data to be transmitted to the SS0TDR register. When setting the last data, disable the transmit data register empty interrupt and enable the transmit end interrupt. • Transmit end interrupt Disable the transmit end interrupt, disable transmission, and prepare for reception. *When receiving • Overrun error interrupt Disable data reception. • Receive data register full interrupt Read the receive data stored in the SS0RDR register. When receiving the second to last frame of data, set the RSSTP bit in the SS0CRH register to 1 (after receiving the current frame, end receive operation (stop the receive clock)) before reading the receive data. When receiving the last frame of data, set the RE bit in the SS0ER register to 0 (disable data reception) and set the RSSTP bit to 0 (after receiving the current frame, continue receive operation (output the receive clock)) before reading the receive data.
Argument	None
Returned value	None
Remark	

_int0	
Outline	INT0 interrupt handling
Header	None
Declaration	void _int0(void)
Description	Enable data transmission.
Argument	None
Returned value	None
Remark	

5.6 Flowcharts

5.6.1 Main Processing

Figure 5.3 shows the Main Processing.

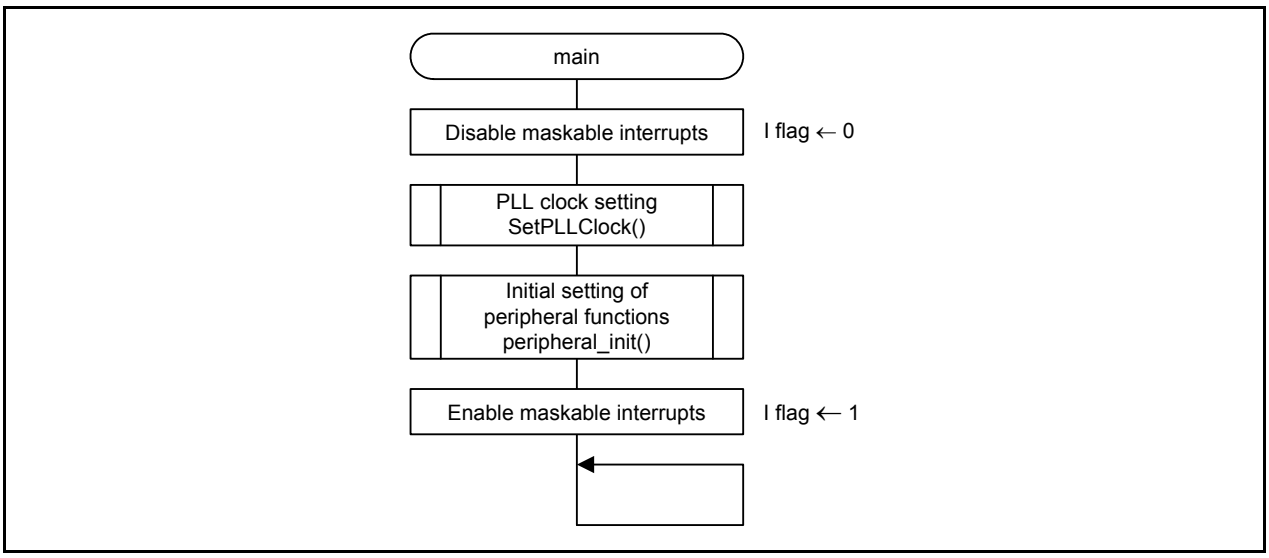


Figure 5.3 Main Processing

5.6.2 Initial Setting of the Peripheral Functions

Figure 5.4 shows the Initial Setting of the Peripheral Functions.

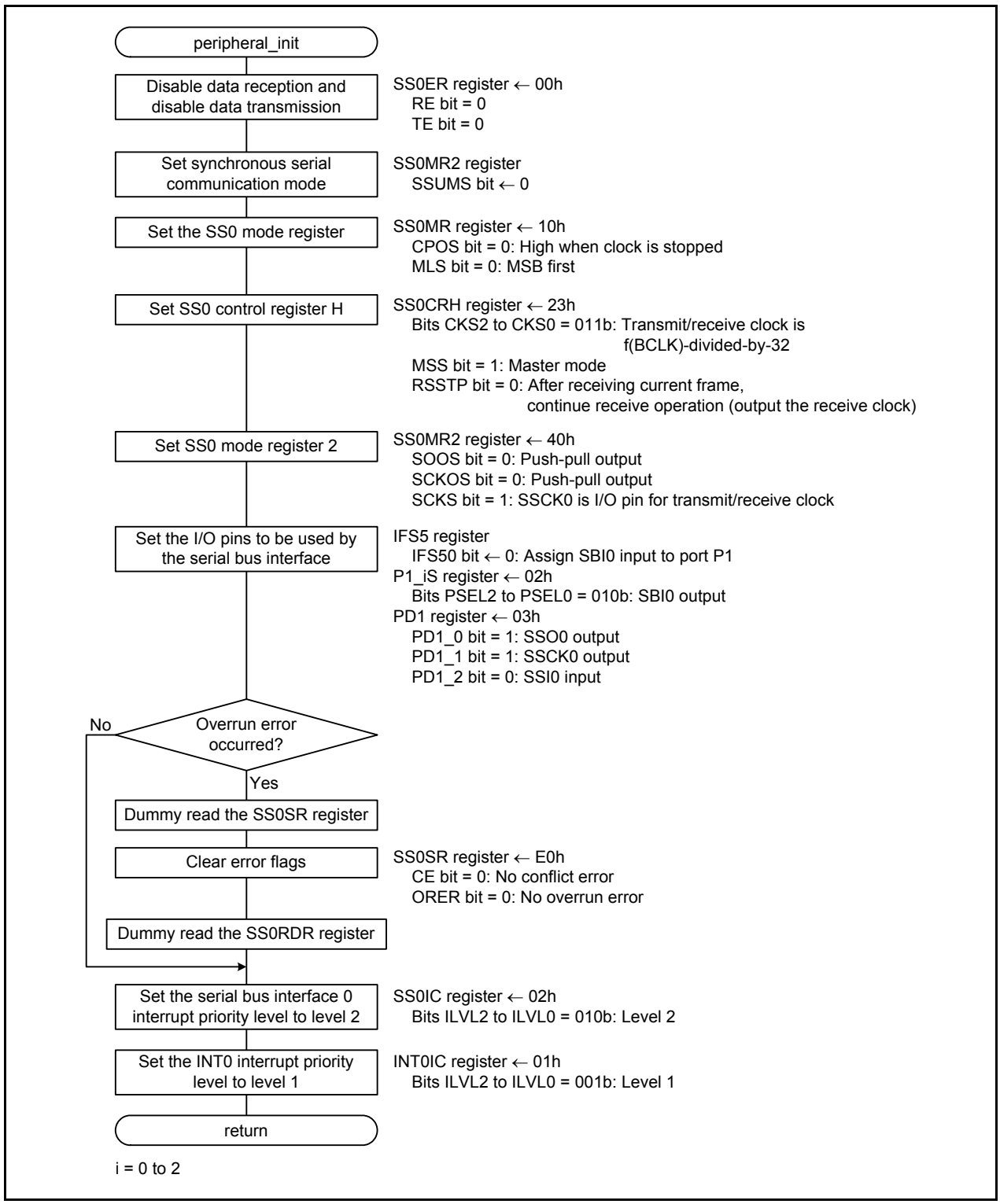


Figure 5.4 Initial Setting of the Peripheral Functions

5.6.3 Serial Bus Interface 0 Interrupt Handling

Figure 5.5 and Figure 5.6 show the Serial Bus Interface 0 Interrupt Handling.

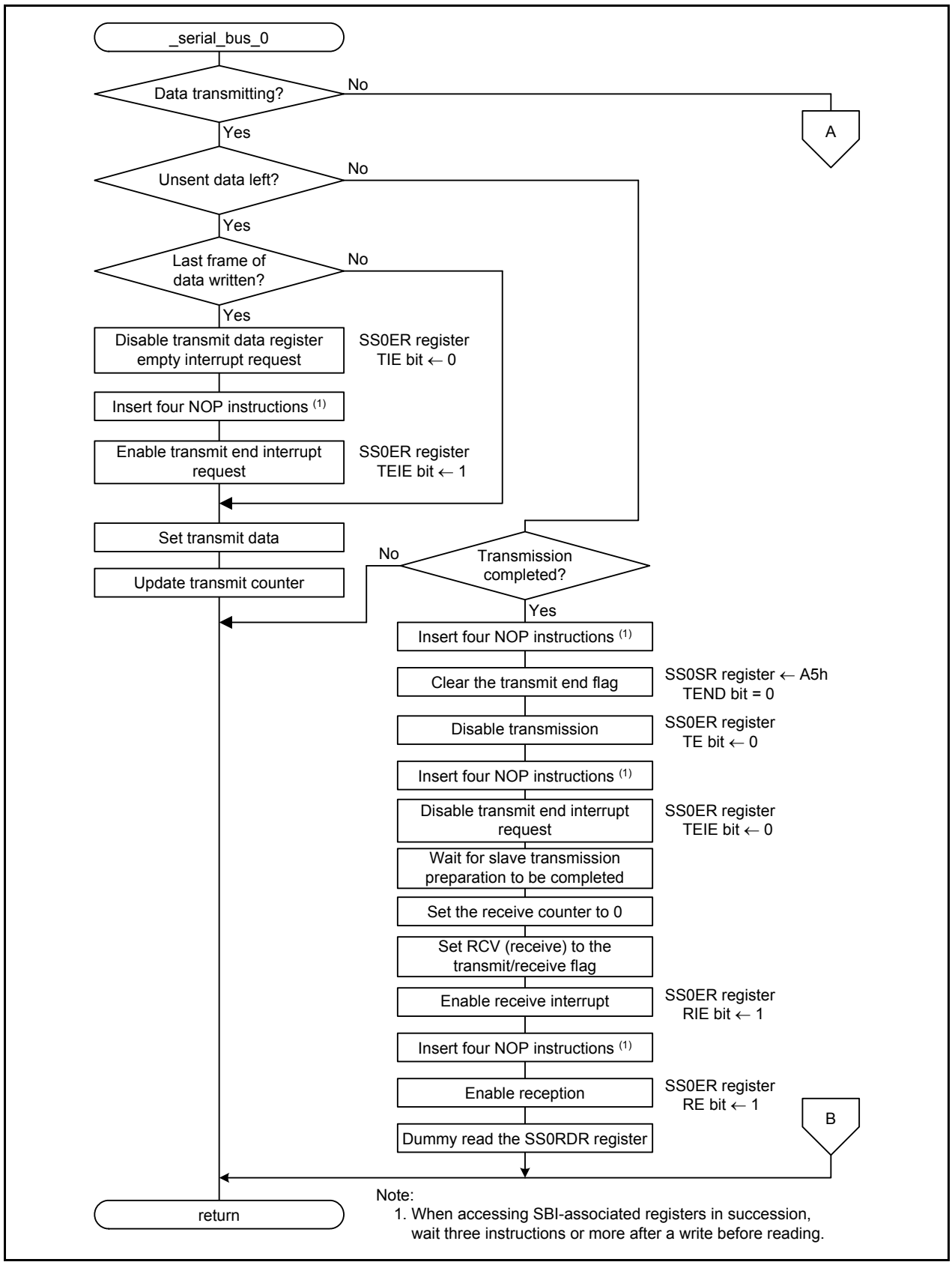


Figure 5.5 Serial Bus Interface 0 Interrupt Handling (1/2)

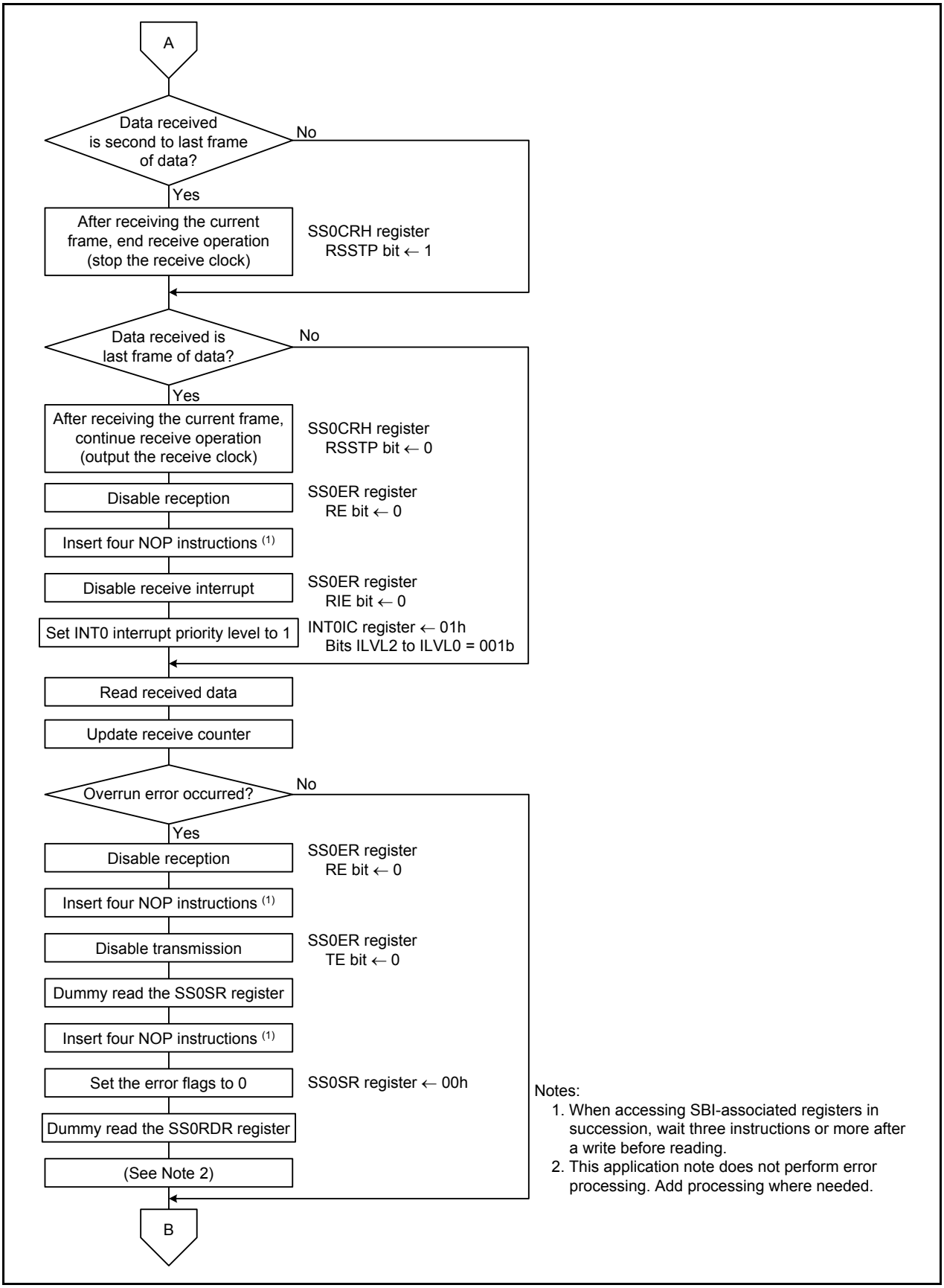


Figure 5.6 Serial Bus Interface 0 Interrupt Handling (2/2)

5.6.4 INT0 Interrupt Handling

Figure 5.7 shows the INT0 Interrupt Handling.

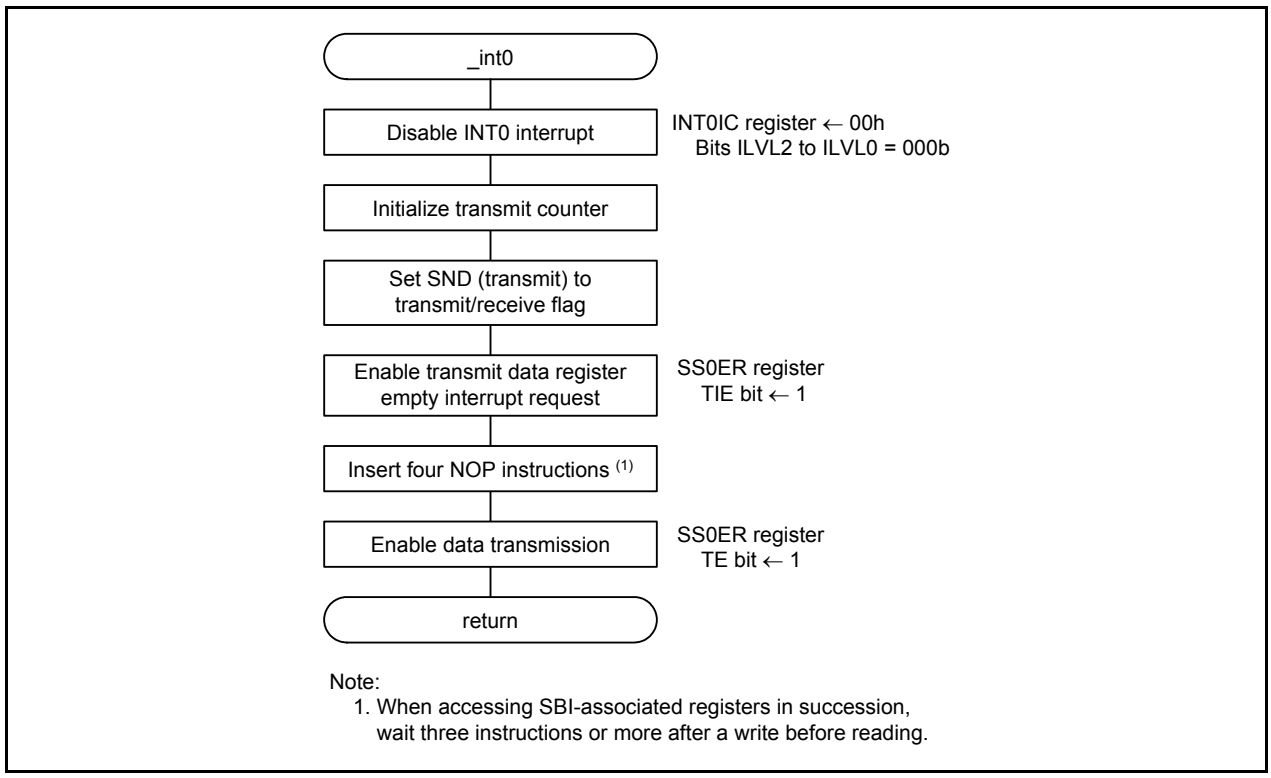


Figure 5.7 INT0 Interrupt Handling

6. Slave Device Hardware

6.1 Pins Used

Table 6.1 lists the Pins Used and Their Functions.

Table 6.1 Pins Used and Their Functions

Pin Name	I/O	Function
P1_0/SSO0	Output	Output serial data
P1_1/SSCK0	Input	Input clock
P1_2/SSI0	Input	Input serial data

7. Slave Device Software

The slave device receives one frame of data from the master device three times. Then the slave device transmits one frame of data three times. The slave device settings are below.

Settings

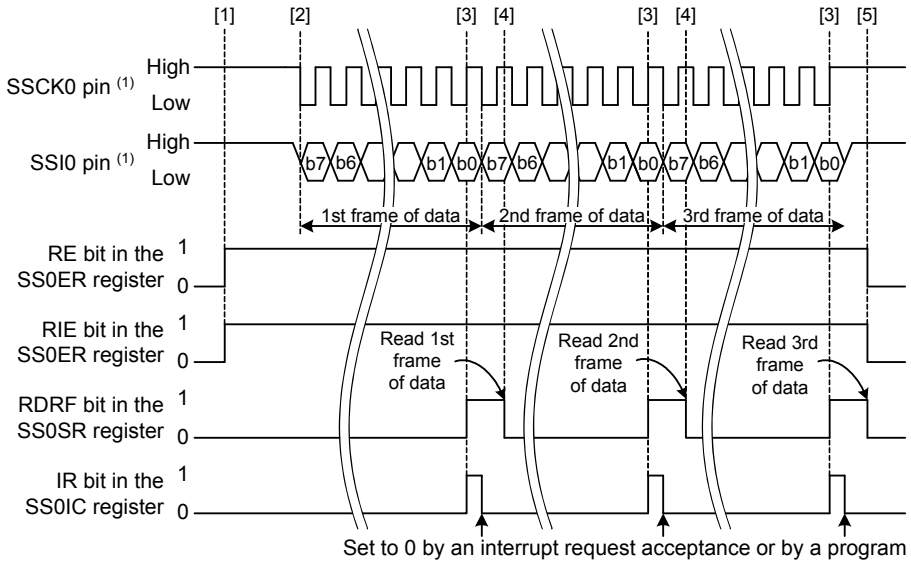
- Channel used: Channel 0
- Clock polarity: High when clock is stopped
- Data transfer direction: MSB first
- Interrupt: Serial bus interface 0 interrupt (interrupt priority level: 2)

7.1 Operation Overview

7.1.1 Slave Reception

(1) To initialize the serial bus interface, set the RE bit in the SS0ER register to 1 (enable data reception), the RIE bit to 1 (enable receive data register full or overrun error interrupt request), then dummy read the SS0RDR register.
(2) Data is received by inputting a clock to the SSCK0 pin.
(3) After receiving one frame of data, the RDRF bit in the SS0SR register becomes 1 (received data left in the SS0RDR register), and the data received is stored in the SS0RDR register. When storing the received data, a receive data register full interrupt occurs.
(4) In the receive data register full interrupt handling, read the received data from the SS0RDR register. When reading, the RDRF bit becomes 0 (no data left in the SS0RDR register).
(5) When the last frame of data is received, set the RIE bit to 0 (disable receive data register full or overrun error interrupt request), the RE bit to 0 (disable data reception), and prepare for transmission.

Figure 7.1 shows the Slave Reception Timing Diagram.



The above diagram assumes the following:
- The SSUMS bit in the SS0MR2 register is 0 (synchronous serial communication mode).
- The BIDE bit in the SS0MR2 register is 0 (standard mode).
- The MSS bit in the SS0CRH register is 0 (slave mode).
- The CPHS bit in the SS0MR register is 0 (change data at odd edges (load data at even edges)).
- The CPOS bit in the SS0MR register is 0 (high when clock is stopped).
- The MLS bit in the SS0MR register is 0 (MSB first).
- Bits CSS1 and CSS0 in the SS0MR2 register are 00b (no function as chip select).

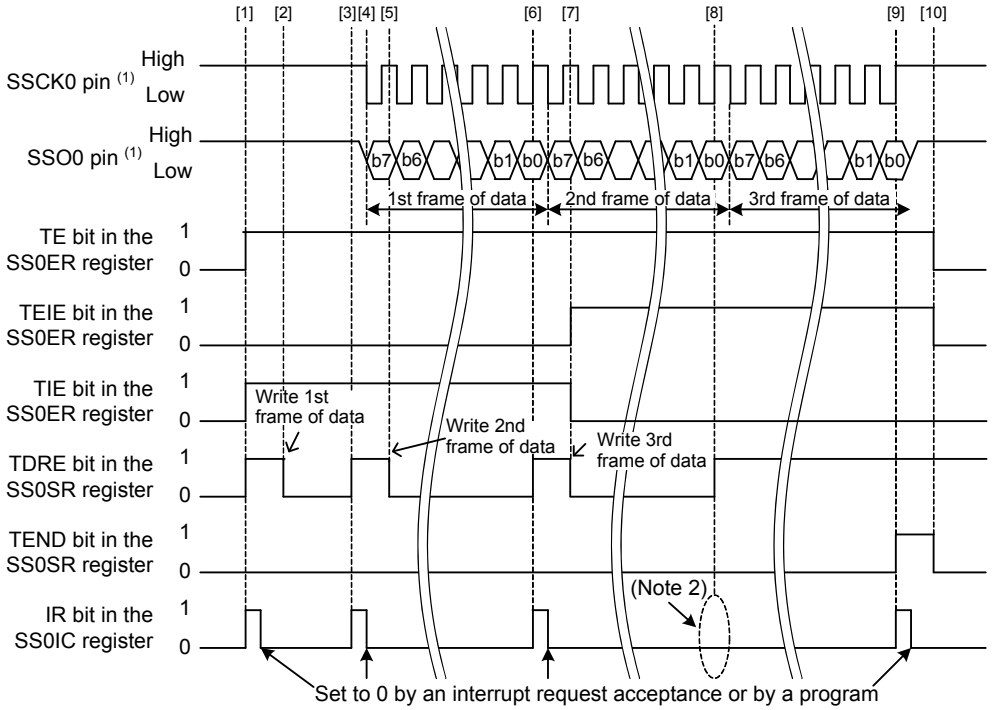
Note:
1. Example of output when pins are pulled-up.

Figure 7.1 Slave Reception Timing Diagram

7.1.2 Slave Transmission

(1) Set the TIE bit in the SS0ER register to 1 (enable transmit data register empty interrupt request) and the TE bit to 1 (enable data transmission). When setting these bits, the TDRE bit in the SS0SR register becomes 1 (no data left in the SS0TDR register), and at the same timing as (5) in Slave Reception, a transmit data register empty interrupt occurs.
(2) In the transmit data register empty interrupt handling, write the first frame of data to the SS0TDR register. When writing, the TDRE bit becomes 0 (unsent data left in the SS0TDR register).
(3) After transferring the first frame of data to the transmit/receive shift register, the TDRE bit becomes 1, and a transmit data register empty interrupt occurs.
(4) The data is transmitted when a clock is input to the SSCK0 pin.
(5) In the transmit data register empty interrupt handling, write the second frame of data to the SS0TDR register.
(6) After transferring the second frame of data to the transmit/receive shift register, the TDRE bit becomes 1, and a transmit data register empty interrupt occurs.
(7) In the transmit data register empty interrupt handling, set the TIE bit to 0 (disable transmit data register empty interrupt request) and the TEIE bit in the SS0ER register to 1 (enable transmit end interrupt request). Write the last frame of data to the SS0TDR register.
(8) After transferring the last frame of data to the transmit/receive shift register, the TDRE bit becomes 1, but because the TIE bit is 0, a transmit data register empty interrupt does not occur.
(9) When the last bit is transmitted, if the TDRE bit is 1, the TEND bit in the SS0SR register becomes 1 (transmission is completed), and a transmit end interrupt occurs.
(10) In the transmit end interrupt handling, set the TEND bit to 0 (transmission continues), the TE bit to 0 (disable data transmission), and the TEIE bit to 0 (disable transmit end interrupt request).

Figure 7.2 shows the Slave Transmission Timing Diagram.



The above diagram assumes the following:
- The SSUMS bit in the SS0MR2 register is 0 (synchronous serial communication mode).
- The BIDE bit in the SS0MR2 register is 0 (standard mode).
- The MSS bit in the SS0CRH register is 0 (slave mode).
- The CPHS bit in the SS0MR register is 0 (change data at odd edges (load data at even edges)).
- The CPOS bit in the SS0MR register is 0 (high when clock is stopped).
- The MLS bit in the SS0MR register is 0 (MSB first).
- Bits CSS1 and CSS0 in the SS0MR2 register are 00b (no function as chip select).

Notes:
1. Example of output when pins are pulled-up.
2. The TIE bit is set to 0 (disable transmit data register empty interrupt request) in [7], so no interrupt occurs.

Figure 7.2 Slave Transmission Timing Diagram

7.2 Constants

Table 7.1 lists the Constants Used in the Sample Code.

Table 7.1 Constants Used in the Sample Code

Constant Name	Setting Value	Contents
SND	00h	Transmission
RCV	01h	Reception
D_SBI_DATA_MAX	3	Transmit data size
D_SBI_DATA_LAST_F	D_SBI_DATA_MAX-1	Constant for determining the last frame of data
ORER	0000 0100b	Mask value for the overrun error flag

7.3 Variables

Table 7.2 lists the Global Variables.

Table 7.2 Global Variables

Type	Variable Name	Contents	Function Used
unsigned char	rcv_cnt	Receive counter	peripheral_init, _serial_bus_0
unsigned char	f_snd_rcv	Transmit/receive flag	peripheral_init, _serial_bus_0
unsigned char	snd_cnt	Transmit counter	_serial_bus_0
unsigned char	snd_data[]	Buffer for storing transmit data	_serial_bus_0
unsigned char	rcv_data[]	Buffer for storing receive data	_serial_bus_0

7.4 Functions

Table 7.3 lists the Functions.

Table 7.3 Functions

Function Name	Outline
peripheral_init	Initial setting of the peripheral functions
_serial_bus_0	Serial bus interface 0 interrupt handling

7.5 Function Specifications

The following tables list the sample code function specifications.

peripheral_init	
Outline	Initial setting for the peripheral interrupts
Header	None
Declaration	void peripheral_init(void)
Description	Setting to use the serial bus interface as the slave device.
Argument	None
Returned value	None
Remark	

_serial_bus_0	
Outline	Serial bus interface 0 interrupt handling
Header	None
Declaration	void _serial_bus_0(void)
Description	Read the transmit/receive flag to determine the source of the interrupt. Perform the handling below based on the determined source of the interrupt. *When transmitting • Transmit data register empty interrupt Write data to be transmitted to the SS0TDR register. When setting the last frame of data, disable the transmit data register empty interrupt and enable the transmit end interrupt. • Transmit end interrupt Disable the transmit end interrupt, disable transmission, and prepare for reception. *When receiving • Overrun error interrupt Disable data reception. • Receive data register full interrupt Read the receive data stored in the SS0RDR register. When receiving the last frame of data, disable reception, disable the receive interrupts, and prepare for transmission.
Argument	None
Returned value	None
Remark	

7.6 Flowcharts

7.6.1 Main Processing

Figure 7.3 shows the Main Processing.

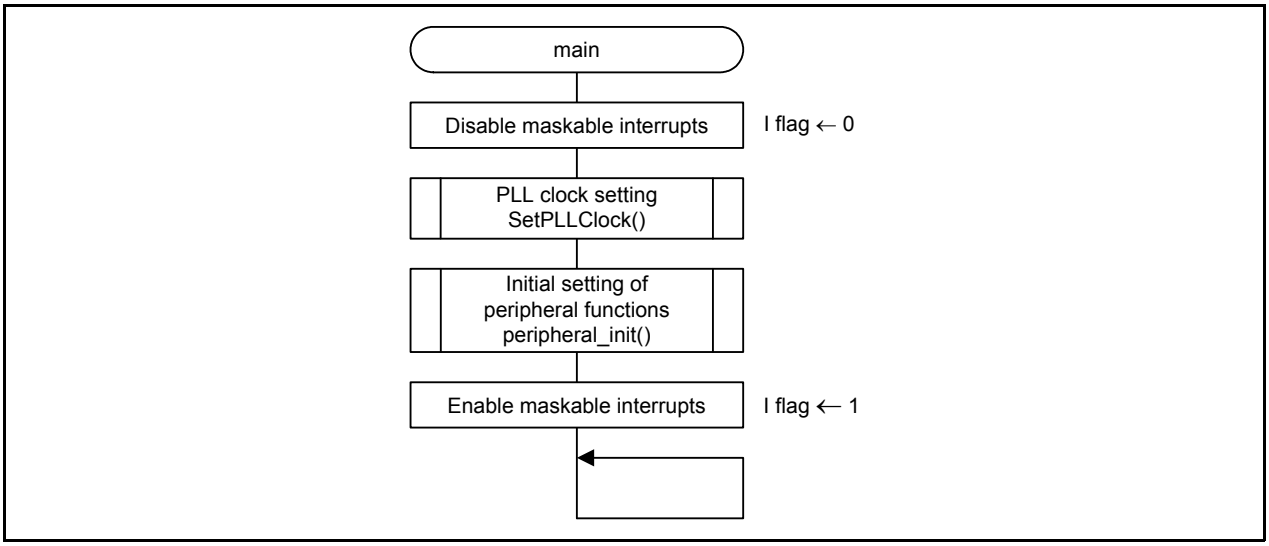


Figure 7.3 Main Processing

7.6.2 Initial Setting of the Peripheral Functions

Figure 7.4 shows the Initial Setting of the Peripheral Functions.

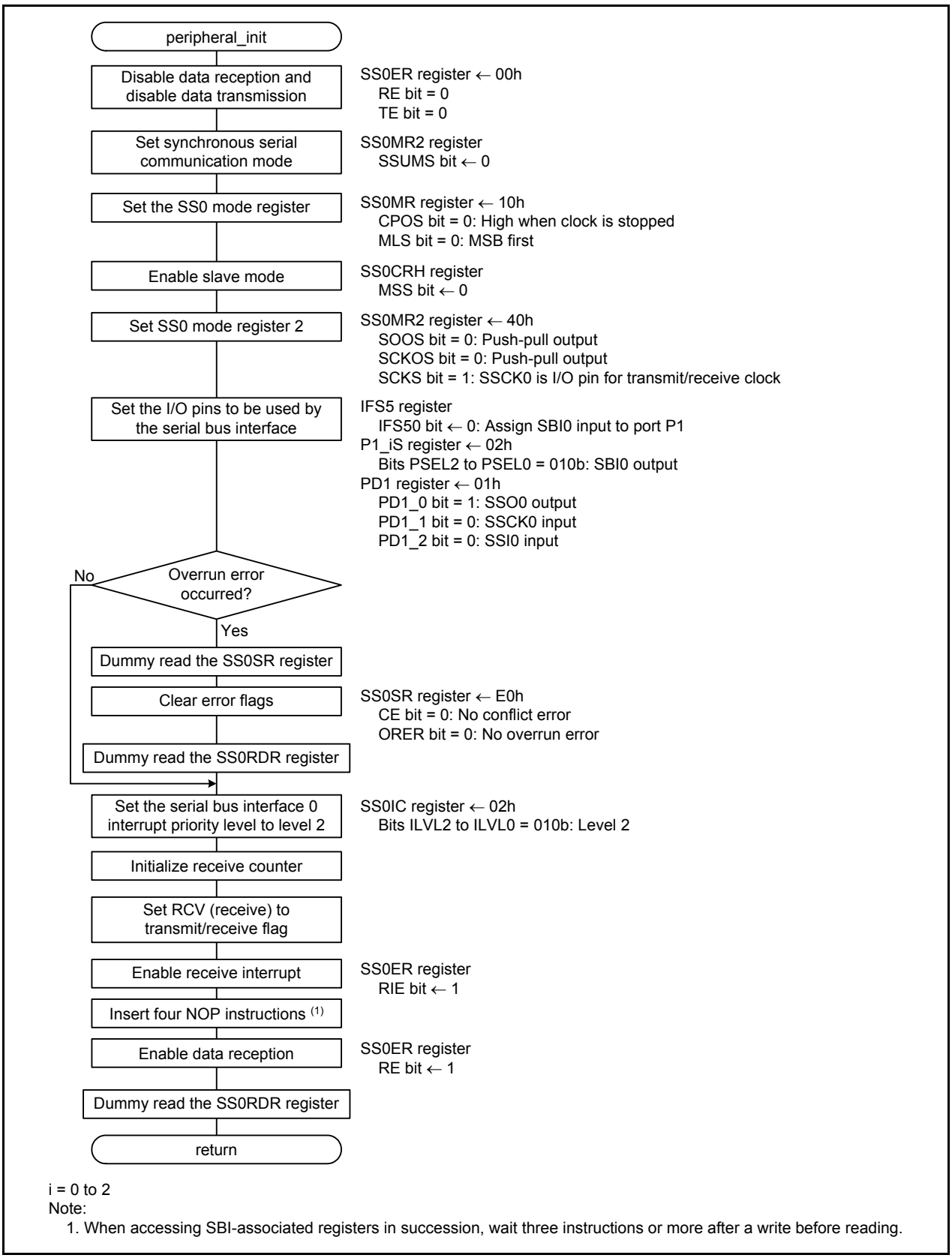


Figure 7.4 Initial Setting of the Peripheral Functions

7.6.3 Serial Bus Interface 0 Interrupt Handling

Figure 7.5 and Figure 7.6 show the Serial Bus Interface 0 Interrupt Handling.

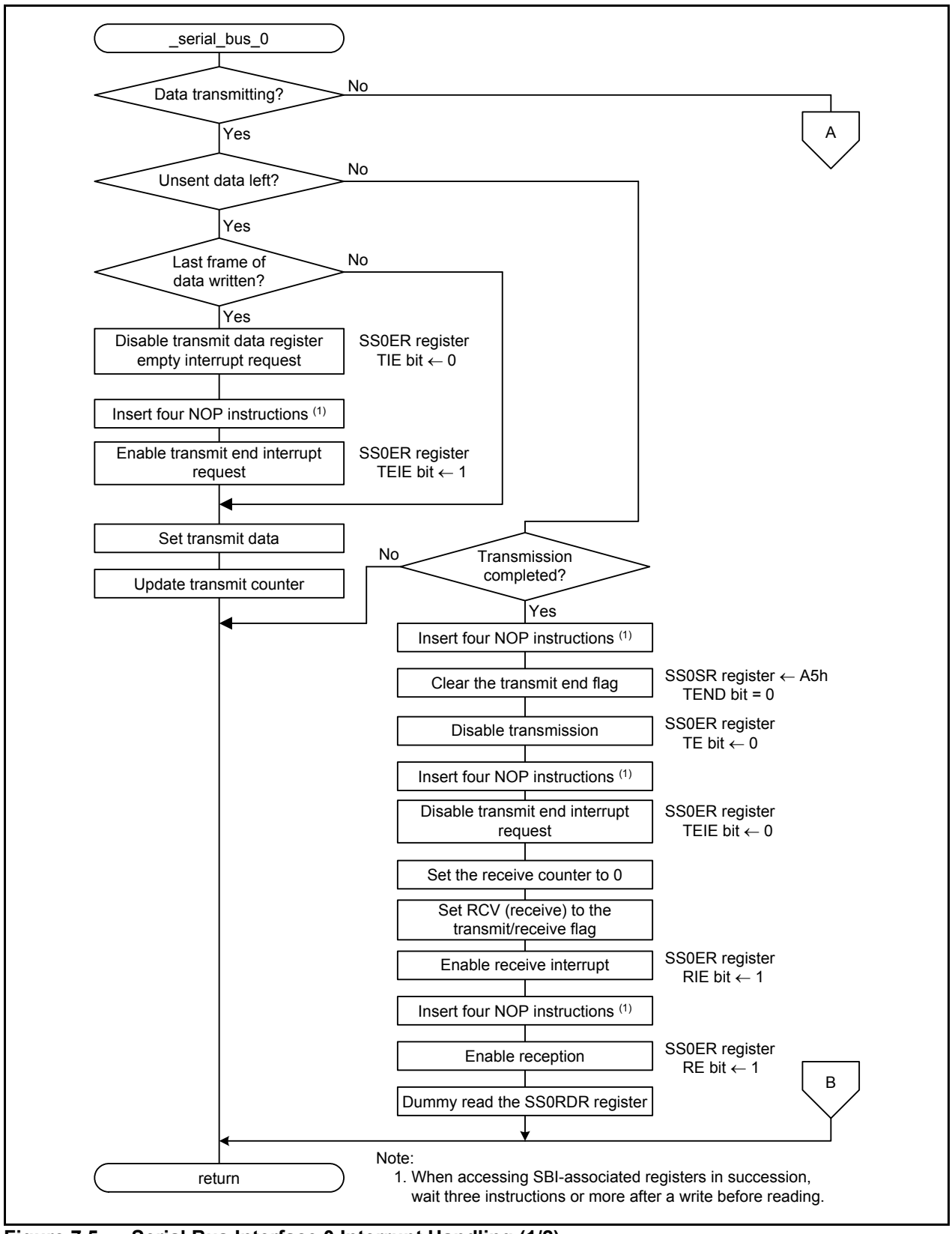


Figure 7.5 Serial Bus Interface 0 Interrupt Handling (1/2)

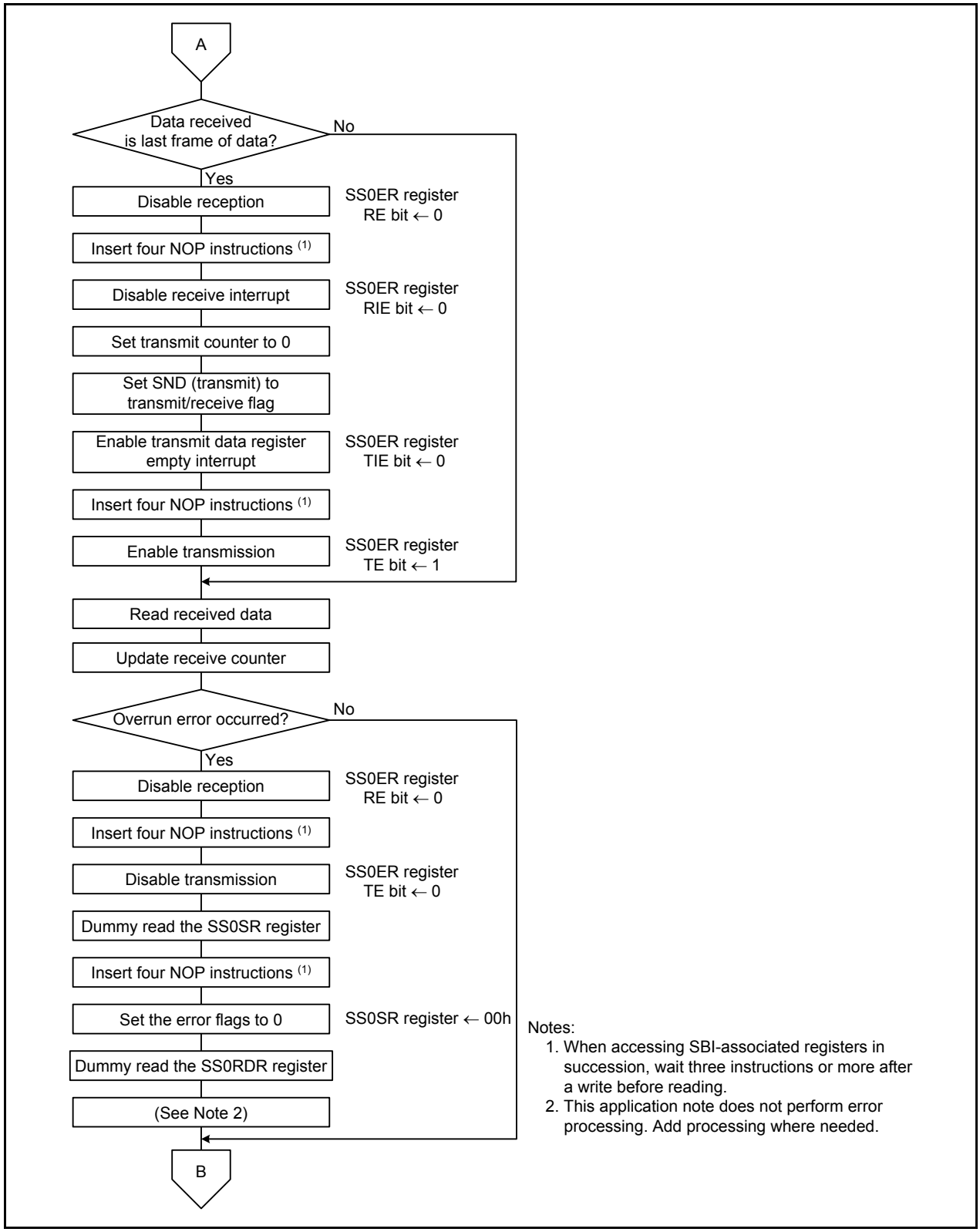


Figure 7.6 Serial Bus Interface 0 Interrupt Handling (2/2)

8. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

9. Reference Documents

R32C/120 Group User's Manual: Hardware Rev.1.20
R32C/121 Group User's Manual: Hardware Rev.1.20
R32C/151 Group User's Manual: Hardware Rev.1.10
R32C/152 Group User's Manual: Hardware Rev.1.10
R32C/153 Group User's Manual: Hardware Rev.1.10
R32C/156 Group User's Manual: Hardware Rev.1.10
R32C/157 Group User's Manual: Hardware Rev.1.10
R32C/160 Group User's Manual: Hardware Rev.1.02
R32C/161 Group User's Manual: Hardware Rev.1.02
The latest versions can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

C Compiler Manual
R32C/100 Series C Compiler Package V.1.02
C Compiler User's Manual Rev.2.00
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/contact/

Revision History	R32C/100 Series Transmitting/Receiving Data Using the Serial Bus Interface in Synchronous Serial Communication Mode

Rev.	Date	Description	
		Page	Summary
1.00	June 29, 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 part number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

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.
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 2.0