To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

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

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

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.

Manual for Using Sample Program Functions Serial Communication (CSIB) (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2)

This manual explains the sample program functions of the clocked serial interface B (CSIB) for the V850E/IA4 microcontroller.

The explanations are based on usage with the V850E/IA4 microcontroller. Refer to this manual when using the V850E/IA3, V850ES/IK1, and V850ES/IE2 microcontrollers.

Caution

This sample program is provided for reference purposes only and operations are therefore not subject to guarantee by NEC Electronics Corporation. When using this sample program, customers are kindly advised to sufficiently evaluate this product based on their system before usage.

Document No. U18234EJ1V0AN00 (1st edition)
Date Published November 2006 N CP(K)

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

③ PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ STATUS BEFORE INITIALIZATION

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

⑤ POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

These commodities, technology or software, must be exported in accordance with the export administration regulations of the exporting country.
Diversion contrary to the law of that country is prohibited.

- **The information in this document is current as of July, 2006. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC 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 NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
  "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.
  "Special": 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, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

INTRODUCTION

Cautions 1. Download the program used in this manual from the NEC Electronics Website (http://www.necel.com/).

2. When using this sample program, reference the following startup file and link directive file, and adjust them if as necessary.

- **Startup file: IA4_start.s**
- **Link directive file: IA4_link.dir**

Conventions The function lists are structured as follows.

Hardware name (symbol)

[Function]	Function description
[Function name]	Name of sample function
[Argument]	Type and overview of argument
[Processing content]	Processing content of sample function
[SFR(s) used]	Register name and setting content
[call function(s)]	Name and function of call function(s)
[Variable(s)]	Type, name, and overview of variable(s) used in sample function
[Interrupt(s)]	Name of function
[Interrupt source(s)]	Name
[File name]	Name of corresponding sample program file
[Caution(s)]	Caution(s) upon function usage

Interrupt function(s)

[Function name]	Name of interrupt function
[Processing content]	Processing content of interrupt function
[SFR(s) used]	Register name and setting content
[call function(s)]	None
[Variable(s)]	Name of variable, function
[File name]	Name of corresponding sample program file
[Caution(s)]	None

Product Differences The differences between the V850E/IA4 and the V850E/IA3, V850ES/IK1, and V850ES/IE2 related to the clocked serial interface B (CSIB) are shown below.

Item	V850E/IA4	V850E/IA3	V850ES/IK1	V850ES/IE2
Channel	2 channels (of which 1 channel has an alternate function as UARTA)	2 channels (of which 1 channel has an alternate function as UARTA)	1 channel	
Communication clock (during master mode)	$f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$	$f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$	$f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, $f_{XX}/64$, $f_{XX}/128$	

Remark f_{XX}: Peripheral clock frequency

Related Documents The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Documents related to V850E/IA3, V850E/IA4, V850ES/IK1, and V850ES/IE2

Document Name	Document No.
V850E1 Architecture User's Manual	U14559E
V850E/IA3, V850E/IA4 Hardware User's Manual	U16543E
V850ES Architecture User's Manual	U15943E
V850ES/IK1 Hardware User's Manual	U16910E
V850ES/IE2 Hardware User's Manual	U17716E
Inverter Control by V850 Series Vector Control by Hole Sensor Application Note	U17338E
Inverter Control by V850 Series Vector Control by Encoder Application Note	U17324E
Inverter Control by V850 Series 120° Excitation Method Control by Zero-Cross Detection Application Note	U17209E
Manual for Using Sample Program Functions Serial Communication (UARTA) (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2) Application Note	U18233E
Manual for Using Sample Program Functions Serial Communication (CSIB) (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2) Application Note	This manual
Manual for Using Sample Program Functions DMA Functions (V850E/IA3, V850E/IA4) Application Note	U18235E
Manual for Using Sample Program Functions Timer M (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2) Application Note	U18236E
Manual for Using Sample Program Functions Watchdog Timer (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2) Application Note	U18237E
Manual for Using Sample Program Functions Timer P (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2) Application Note	U18238E
Manual for Using Sample Program Functions Timer Q (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2) Application Note	U18239E
Manual for Using Sample Program Functions Timer ENC (V850E/IA3, V850E/IA4) Application Note	U18240E
Manual for Using Sample Program Functions Port Functions (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2) Application Note	U18241E
Manual for Using Sample Program Functions Clock Generator (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2) Application Note	U18242E
Manual for Using Sample Program Functions Standby Functions (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2) Application Note	U18243E
Manual for Using Sample Program Functions Interrupt Functions (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2) Application Note	U18244E
Manual for Using Sample Program Functions A/D Converters 0 and 1 (V850E/IA3, V850E/IA4, V850ES/IK1, V850ES/IE2) Application Note	U18245E
Manual for Using Sample Program Functions A/D Converter 2 (V850E/IA3, V850E/IA4) Application Note	U18246E

CONTENTS

Clocked serial interface B (CSIBn) (n = 0, 1)

Continuous transfer mode (master mode, transmission mode)

[Function]	Sets communication mode to master mode and transfer direction mode to MSB first, and performs data transmission for ten times in continuous transfer mode. Validates communication start trigger and sets communication clock to fxx/256, and transfer data length to 8 bits.	
[Function name]	csib1_main	
[Argument]	None	
[Processing content]	Sets transmission count (count_tx) to initial value 0. Starts transmission after calling each setting function.	
[SFRs used]	CB0TIC:	0x07 (Clears CSIB0 transmission enable interrupt request signal (INTCB0T), releases mask, sets to priority level 7.)
	CB0STR.CB0TSF	Communication status flag
[call function]	csib_port, csib_set, csib_start, csib_end	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
	unsigned char count:	Transmit data generating variable
[Interrupt]	csib_int_send	
[Interrupt source]	INTCB0T	
[File name]	csib1.c	
[Caution]	None	

[Function name]	csib_port	
[Processing content]	Sets port 4 as CSIB0 I/O pin.	
[SFR used]	PMC4:	0x06 (Sets $\overline{\text{SCKB0}}$ I/O and SOB0 output.)
[call function]	None	
[Variable]	None	
[File name]	csib1.c	
[Caution]	None	

[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x06 (Sets to communication type 1 and sets communication clock to fxx/256 (0.25 MHz).) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x43 (Enables CSIB0 transmission operation, sets to MSB first and continuous transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib1.c
[Caution]	CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

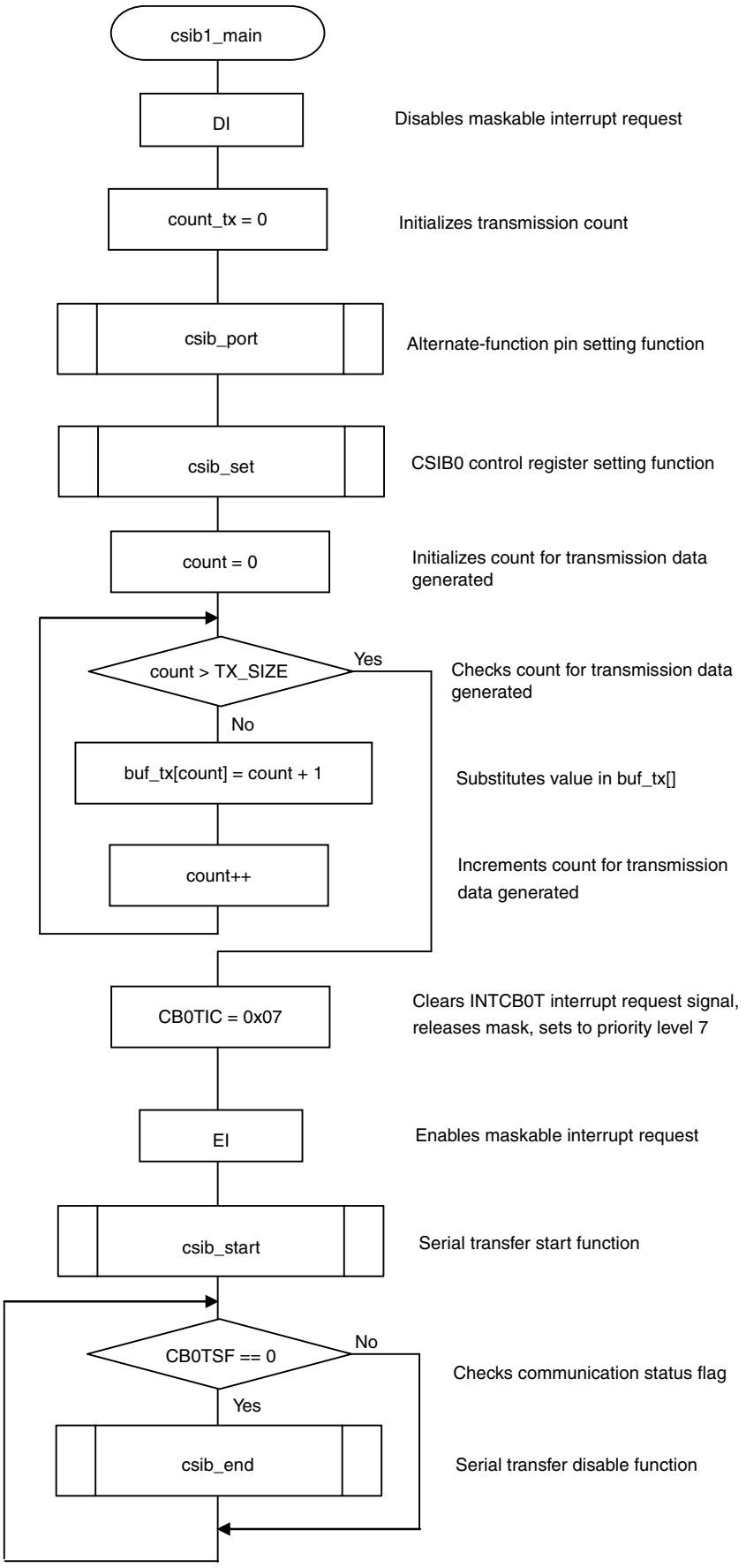
[Function name]	csib_start	
[Processing content]	Enables CSIB0 operation and writes a value to transmit data register.	
[SFRs used]	CB0CTL0.CB0PWR:	1 (Enables CSIB0 operation.)
	CB0TX	Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	count_tx:	Transmission count variable
[File name]	csib1.c	
[Caution]	None	

[Function name]	csib_end	
[Processing content]	Disables CSIB0 operation and transmission operation.	
[SFRs used]	CB0CTL0.CB0PWR:	0 (Disables CSIB0 operation.)
	CB0CTL0.CB0TXE:	0 (Disables CSIB0 transmission operation.)
[call function]	None	
[Variable]	None	
[File name]	csib1.c	
[Caution]	None	

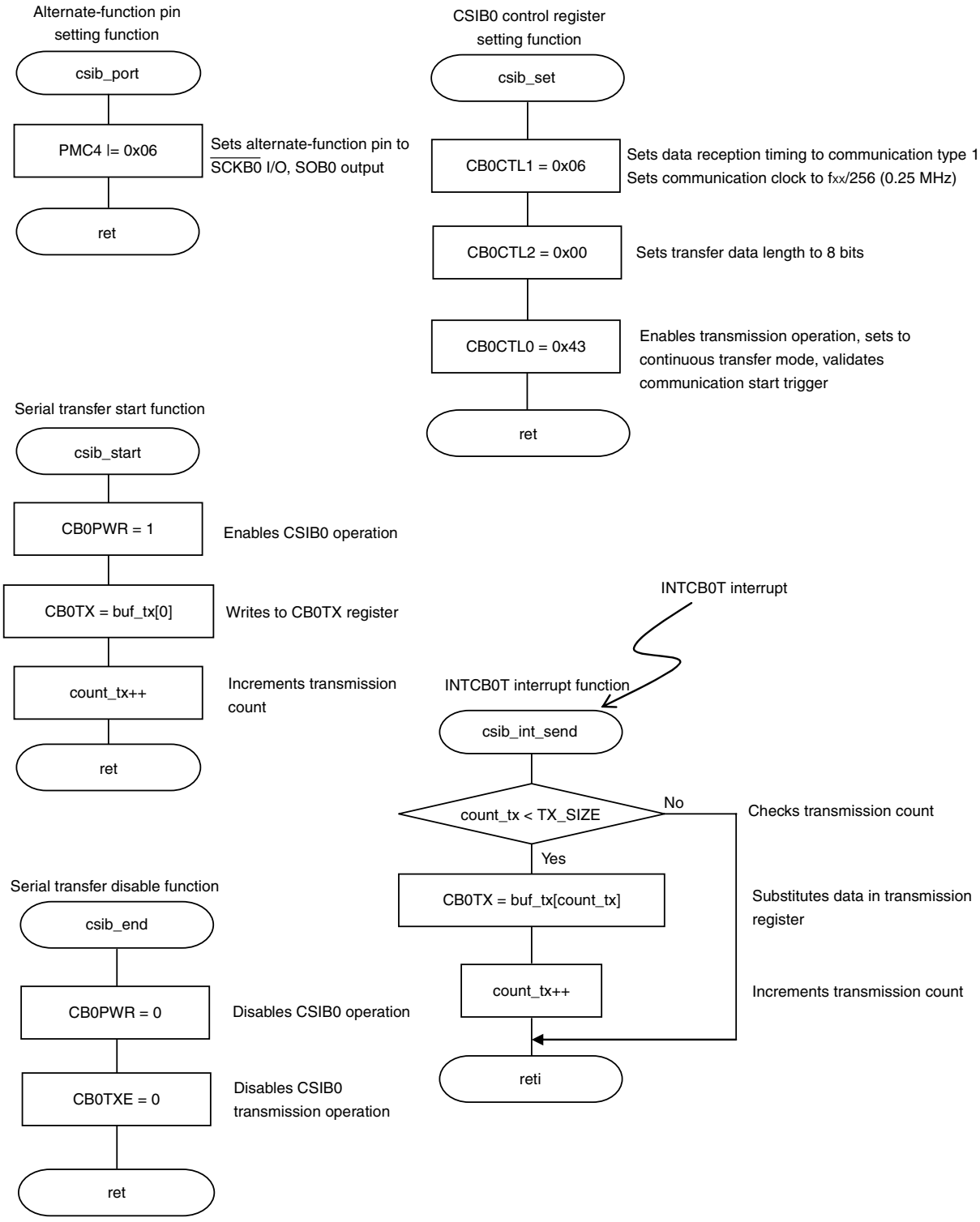
Interrupt function

[Function name]	csib_int_send	
[Processing content]	Sets new data for transmitting next data.	
[SFR used]	CB0TX	Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
[File name]	csib1.c	
[Caution]	None	

Continuous transfer mode (master mode, transmission mode) (1/2)



Continuous transfer mode (master mode, transmission mode) (2/2)



Clocked serial interface B (CSIBn) (n = 0, 1)

Continuous transfer mode (master mode, reception mode)

[Function]	Sets communication mode to master mode and transfer direction mode to MSB first, and performs data reception for ten times in continuous transfer mode. Validates communication start trigger and sets communication clock to fxx/256, and transfer data length to 8 bits.
[Function name]	csib2_main
[Argument]	None
[Processing content]	Sets reception count (count_rx) to initial value 0. Starts reception after calling each setting function.
[SFR used]	CB0RIC: 0x07 (Clears CSIB0 reception end interrupt request signal (INTCB0R), releases mask, sets to priority level 7.) CB0REIC: 0x07 (Clears CSIB0 reception error interrupt request signal (INTCB0RE), releases mask, sets to priority level 7.) CB0STR.CB0TSF Communication status flag
[call function]	csib_port, csib_set, csib_start, csib_end
[Variables]	unsigned char buf_rx[ ]: Receive data storing buffer volatile unsigned char count_rx: Reception count variable
[Interrupts]	csib_int_receive, csib_error
[Interrupt sources]	INTCB0R, INTCB0RE
[File name]	csib2.c
[Caution]	None

[Function name]	csib_port
[Processing content]	Sets port 4 as CSIB0 I/O pin.
[SFR used]	PMC4: 0x05 (Sets $\overline{\text{SCKB0}}$ I/O and SIB0 input.)
[call function]	None
[Variable]	None
[File name]	csib2.c
[Caution]	None

[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x06 (Sets to communication type 1 and sets communication clock to f_{XX}/256 (0.25 MHz).) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x23 (Enables CSIB0 reception operation, sets to MSB first and continuous transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib2.c
[Caution]	CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

[Function name]	csib_start	
[Processing content]	Enables CSIB0 operation and performs dummy read on receive data register.	
[SFRs used]	CB0CTL0.CB0PWR:	1 (Enables CSIB0 operation.)
	CB0RX	Receive data register
[call function]	None	
[Variable]	unsigned char buf_rx[ ]:	Receive data storing buffer
[File name]	csib2.c	
[Caution]	None	

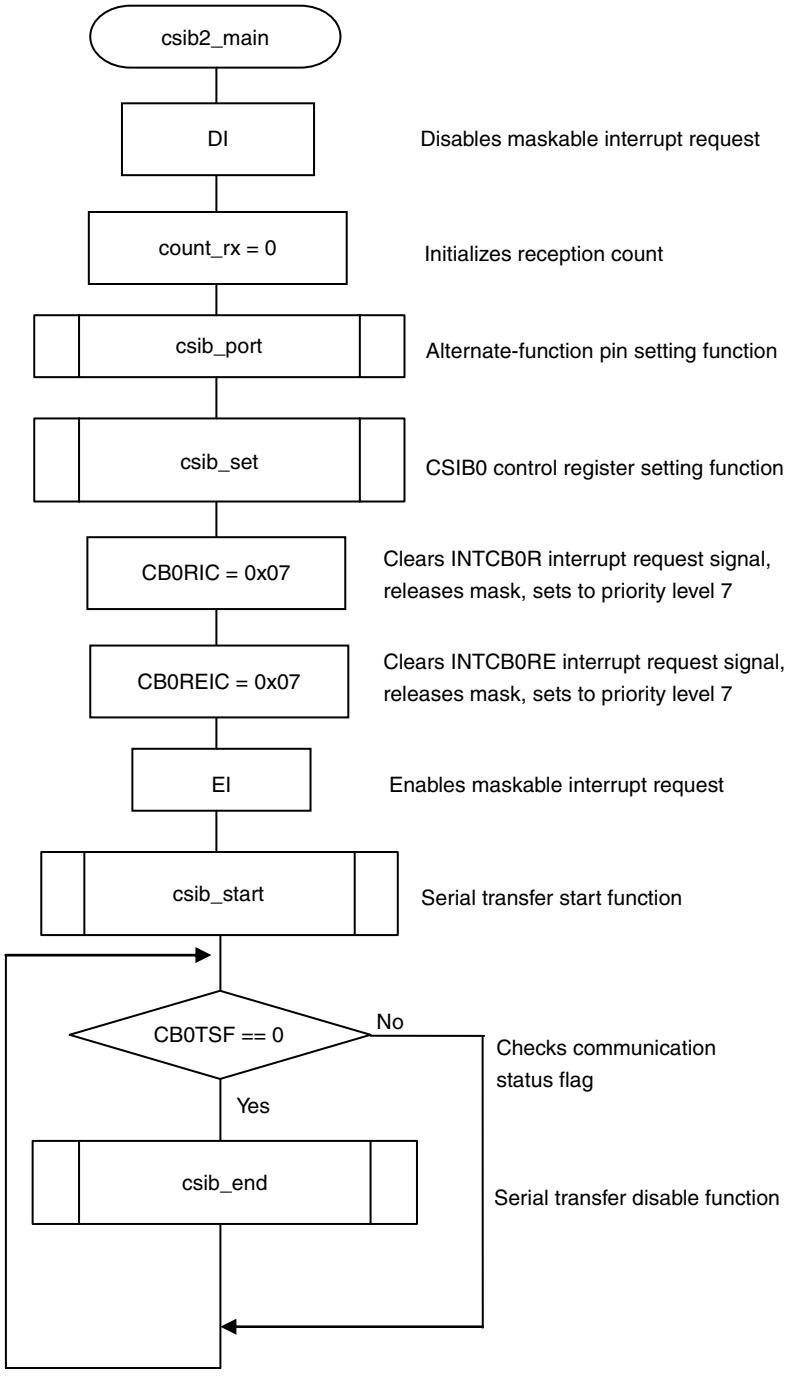
[Function name]	csib_end	
[Processing content]	Disables CSIB0 operation and reception operation.	
[SFRs used]	CB0CTL0.CB0PWR:	0 (Disables CSIB0 operation.)
	CB0CTL0.CB0RXE:	0 (Disables CSIB0 reception operation.)
[call function]	None	
[Variable]	None	
[File name]	csib2.c	
[Caution]	None	

Interrupt functions

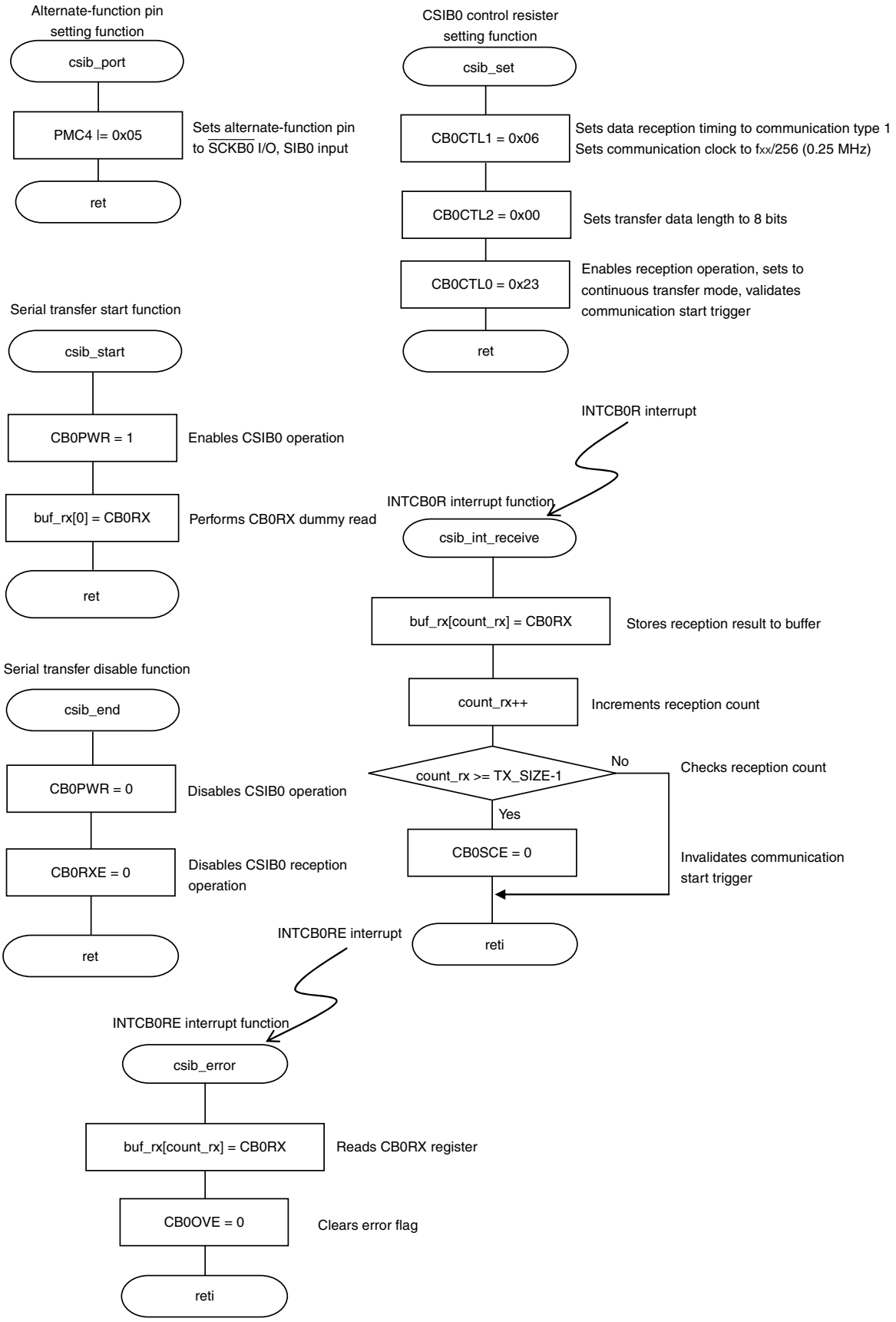
[Function name]	csib_int_receive	
[Processing content]	Stores receive data to buffer.	
[SFRs used]	CB0RX	Receive data register
	CB0CTL0.CB0SCE:	0 (Invalidates communication start trigger.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib2.c	
[Caution]	None	

[Function name]	csib_error	
[Processing content]	Clears reception error flag.	
[SFRs used]	CB0RX	Receive data register
	CB0STR.CB0OVE:	0 (Clears overrun error flag.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib2.c	
[Caution]	None	

Continuous transfer mode (master mode, reception mode) (1/2)



Continuous transfer mode (master mode, reception mode) (2/2)



Clocked serial interface B (CSIBn) (n = 0, 1)

Continuous transfer mode (master mode, transmission/reception mode)

[Function]	Sets communication mode to master mode and transfer direction mode to MSB first, and performs transmission/reception for ten times each in continuous transfer mode. Validates communication start trigger and sets communication clock to fxx/256, and transfer data length to 8 bits.
[Function name]	csib3_main
[Argument]	None
[Processing content]	Sets transmission count (count_tx) to initial value 0. Sets reception count (count_rx) to initial value 0 and starts transmission/reception after calling each setting function.
[SFRs used]	CB0REIC: 0x07 (Enables CSIB0 reception error interrupt (INTCB0RE) servicing.) CB0RIC: 0x07 (Enables CSIB0 reception end interrupt (INTCB0R) servicing.) CB0TIC: 0x07 (Enables CSIB0 transmission enable interrupt (INTCB0T) servicing.) CB0STR.CB0TSF Communication status flag
[call functions]	csib_port, csib_set, csib_start, csib_end
[Variables]	unsigned char buf_tx[ ]: Transmit data storing buffer unsigned char buf_rx[ ]: Receive data storing buffer volatile unsigned char count_tx: Transmission count variable volatile unsigned char count_rx: Reception count variable unsigned char count: Transmit data generating variable
[Interrupts]	csib_error, csib_int_send, csib_int_receive
[Interrupt sources]	INTCB0RE, INTCB0R, INTCB0T
[File name]	csib3.c
[Caution]	None

[Function name]	csib_port
[Processing content]	Sets port 4 as CSIB0 I/O pin.
[SFR used]	PMC4: 0x07 (Sets $\overline{\text{SCKB0}}$ I/O, SOB0 output and SIB0 input.)
[call function]	None
[Variable]	None
[File name]	csib3.c
[Caution]	None

[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x06 (Sets to communication type 1 and sets communication clock to fxx/256 (0.25 MHz).) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x63 (Enables CSIB0 transmission/reception operation, sets to MSB first and continuous transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib3.c
[Caution]	CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

[Function name]	csib_start	
[Processing content]	Enables CSIB0 operation and writes a value to transmit data register.	
[SFRs used]	CB0CTL0.CB0PWR: CB0TX	1 (Enables CSIB0 operation) Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]: volatile char count_tx:	Transmit data storing buffer Transmission count variable
[File name]	csib3.c	
[Caution]	None	

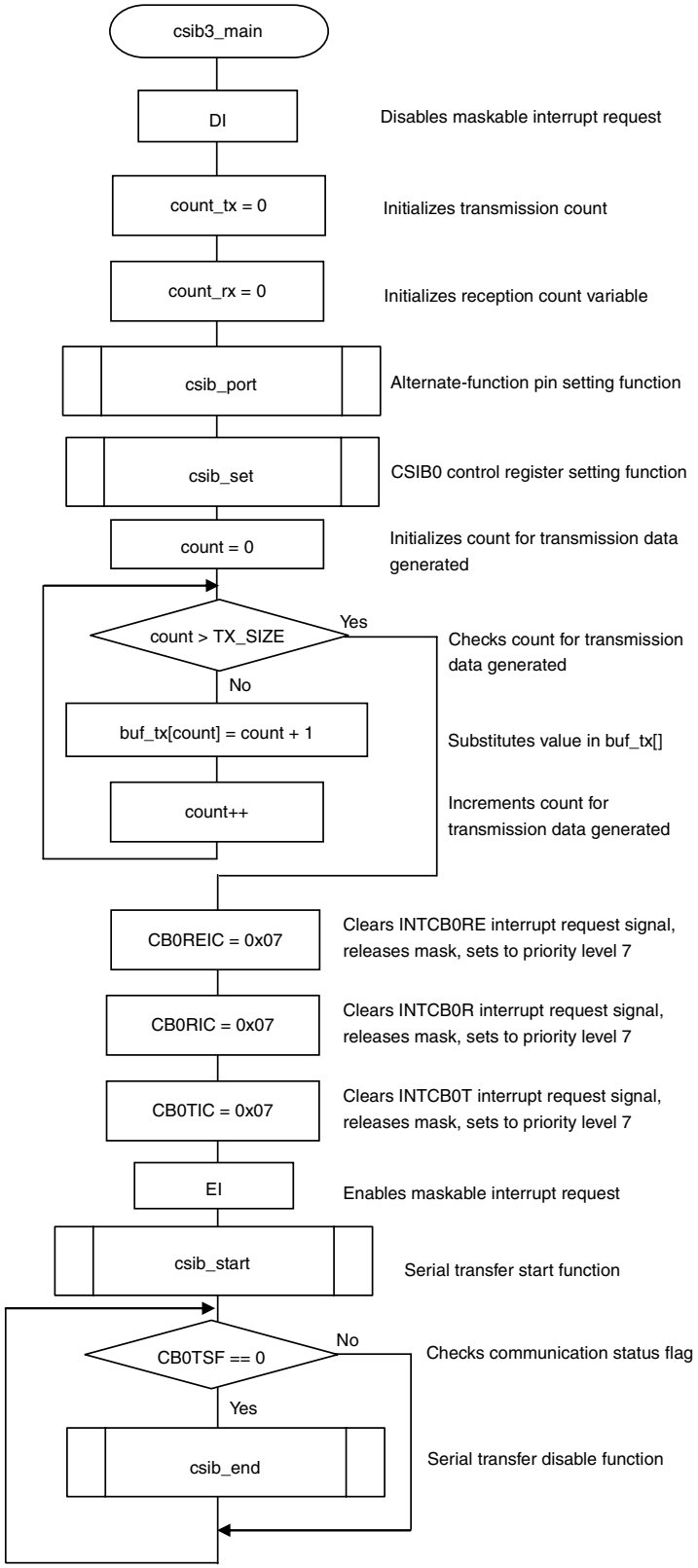
[Function name]	csib_end	
[Processing content]	Disables CSIB0 operation and transmission/reception operation.	
[SFRs used]	CB0CTL0.CB0PWR: CB0CTL0.CB0RXE: CB0CTL0.CB0TXE:	0 (Disables CSIB0 operation.) 0 (Disables CSIB0 reception operation.) 0 (Disables CSIB0 transmission operation.)
[call function]	None	
[Variable]	None	
[File name]	csib3.c	
[Caution]	None	

[Function name]	csib_error	
[Processing content]	Clears reception error flag.	
[SFRs used]	CB0RX	Receive data register
	CB0STR.CB0OVE:	0 (Clears overrun error flag.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib3.c	
[Caution]	None	

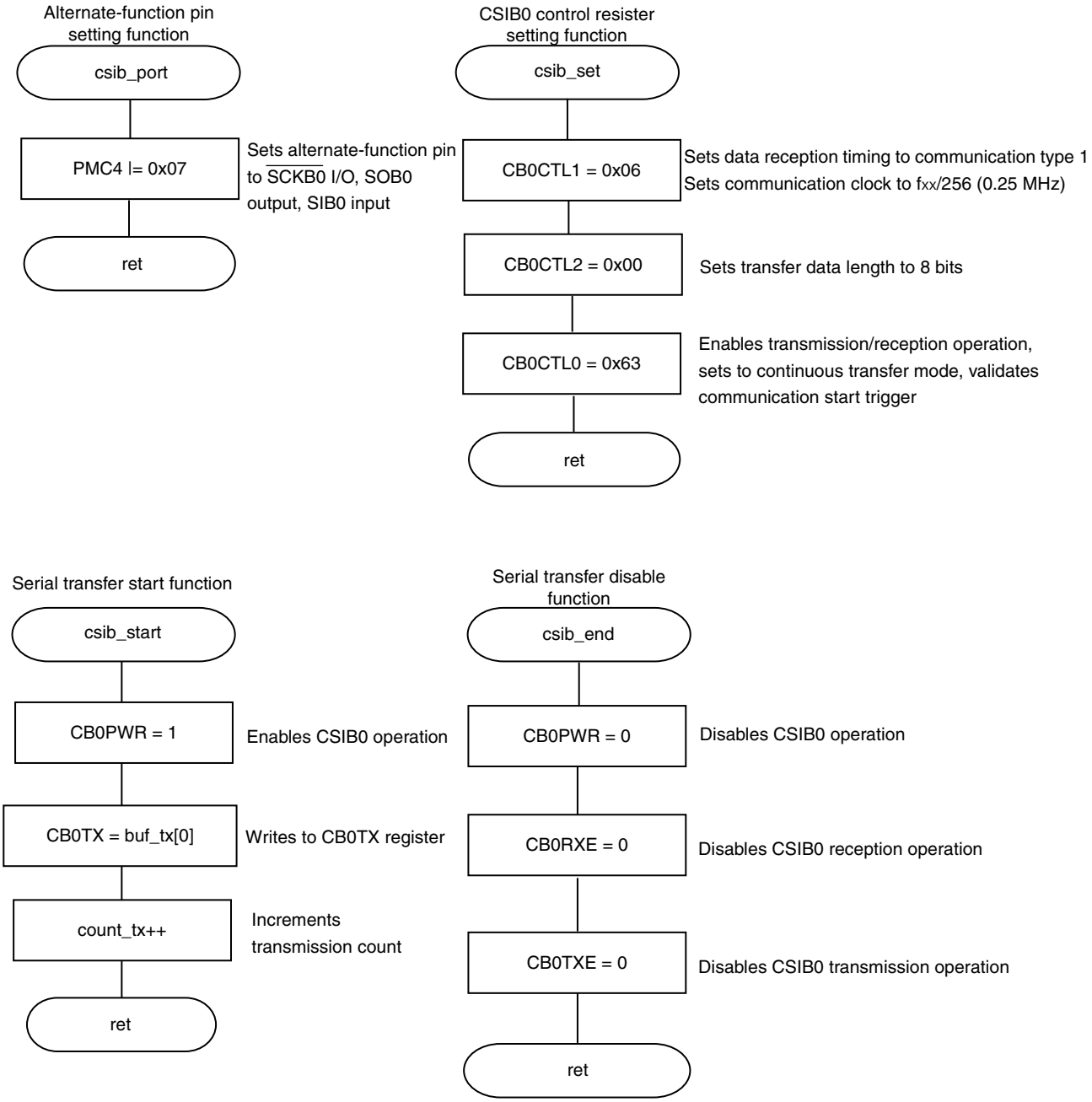
[Function name]	csib_int_send	
[Processing content]	Sets new data for transmitting next data.	
[SFR used]	CB0TX	Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
[File name]	csib3.c	
[Caution]	None	

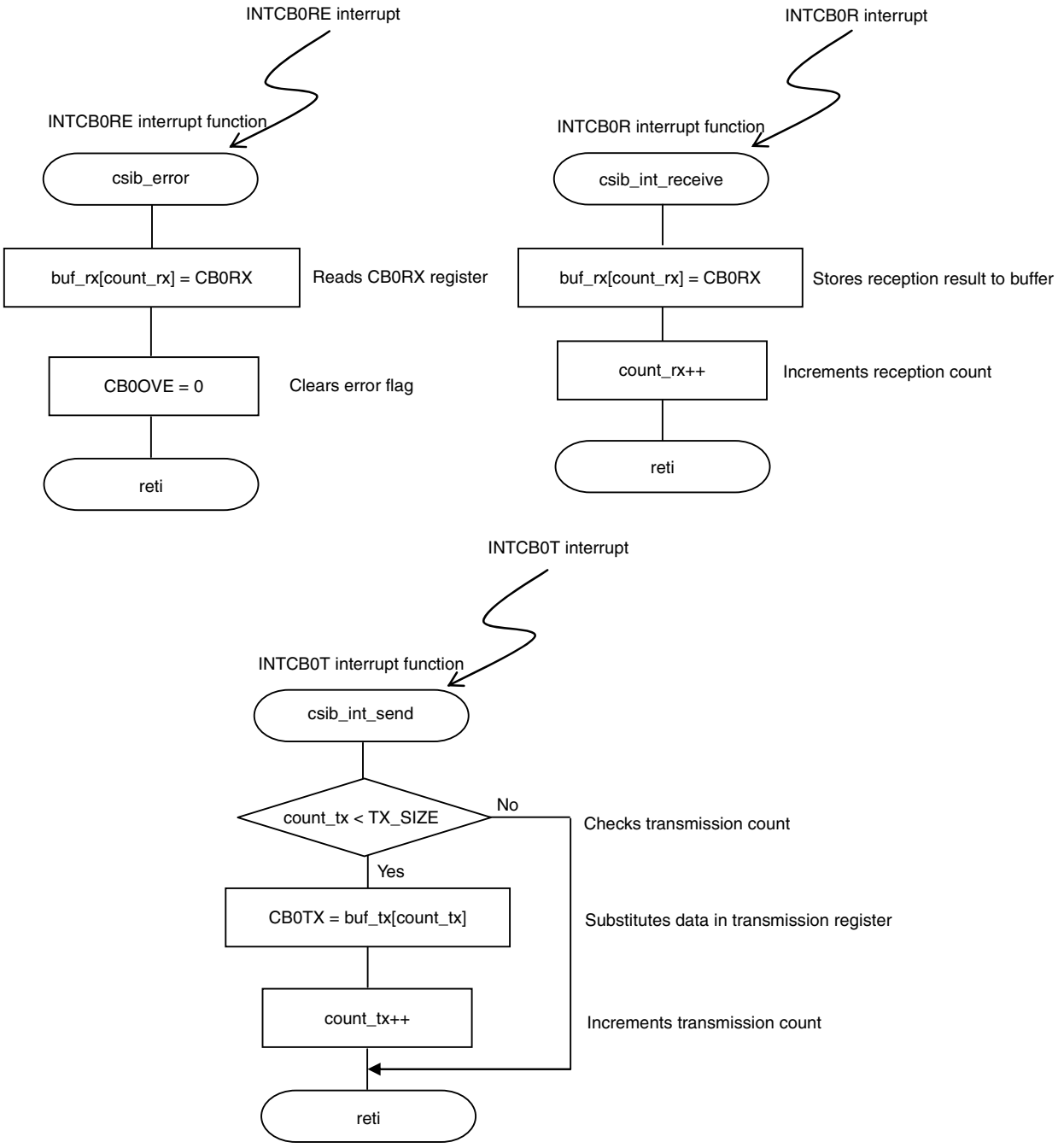
[Function name]	csib_int_receive	
[Processing content]	Stores receive data to buffer.	
[SFR used]	CB0RX	Receive data register
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib3.c	
[Caution]	None	

Continuous transfer mode (master mode, transmission/reception mode) (1/3)



Continuous transfer mode (master mode, transmission/reception mode) (2/3)




INTCB0RE interrupt
INTCB0RE interrupt function
csib_error
buf_rx[count_rx] = CB0RX
Reads CB0RX register
CB0OVE = 0
Clears error flag
reti
INTCB0R interrupt
INTCB0R interrupt function
csib_int_receive
buf_rx[count_rx] = CB0RX
Stores reception result to buffer
count_rx++
Increments reception count
reti
INTCB0T interrupt
INTCB0T interrupt function
csib_int_send
count_tx < TX_SIZE
No
Checks transmission count
Yes
CB0TX = buf_tx[count_tx]
Substitutes data in transmission register
count_tx++
Increments transmission count
reti

Clocked serial interface B (CSIBn) (n = 0, 1)

Continuous transfer mode (slave mode, transmission mode)

[Function]	Sets communication mode to slave mode and transfer direction mode to MSB first, and performs transmission for ten times in continuous transfer mode. Validates communication start trigger and sets communication clock to external clock, and transfer data length to 8 bits.	
[Function name]	csib4_main	
[Argument]	None	
[Processing content]	Sets transmission count (count_tx) to initial value 0. Starts transmission after calling each setting function.	
[SFRs used]	CB0TIC:	0x07 (Clears CSIB0 transmission enable interrupt request signal (INTCB0T), releases mask, sets to priority level 7.)
	CB0STR.CB0TSF	Communication status flag
[call function]	csib_port, csib_set, csib_start, csib_end	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
	unsigned char count:	Transfer data generating variable
[Interrupt]	csib_int_send	
[Interrupt source]	INTCB0T	
[File name]	csib4.c	
[Caution]	None	

[Function name]	csib_port
[Processing content]	Sets port 4 as CSIB0 I/O pin.
[SFR used]	PMC4: 0x06 (Sets $\overline{\text{SCKB0}}$ I/O and SOB0 output.)
[call function]	None
[Variable]	None
[File name]	csib4.c
[Caution]	None

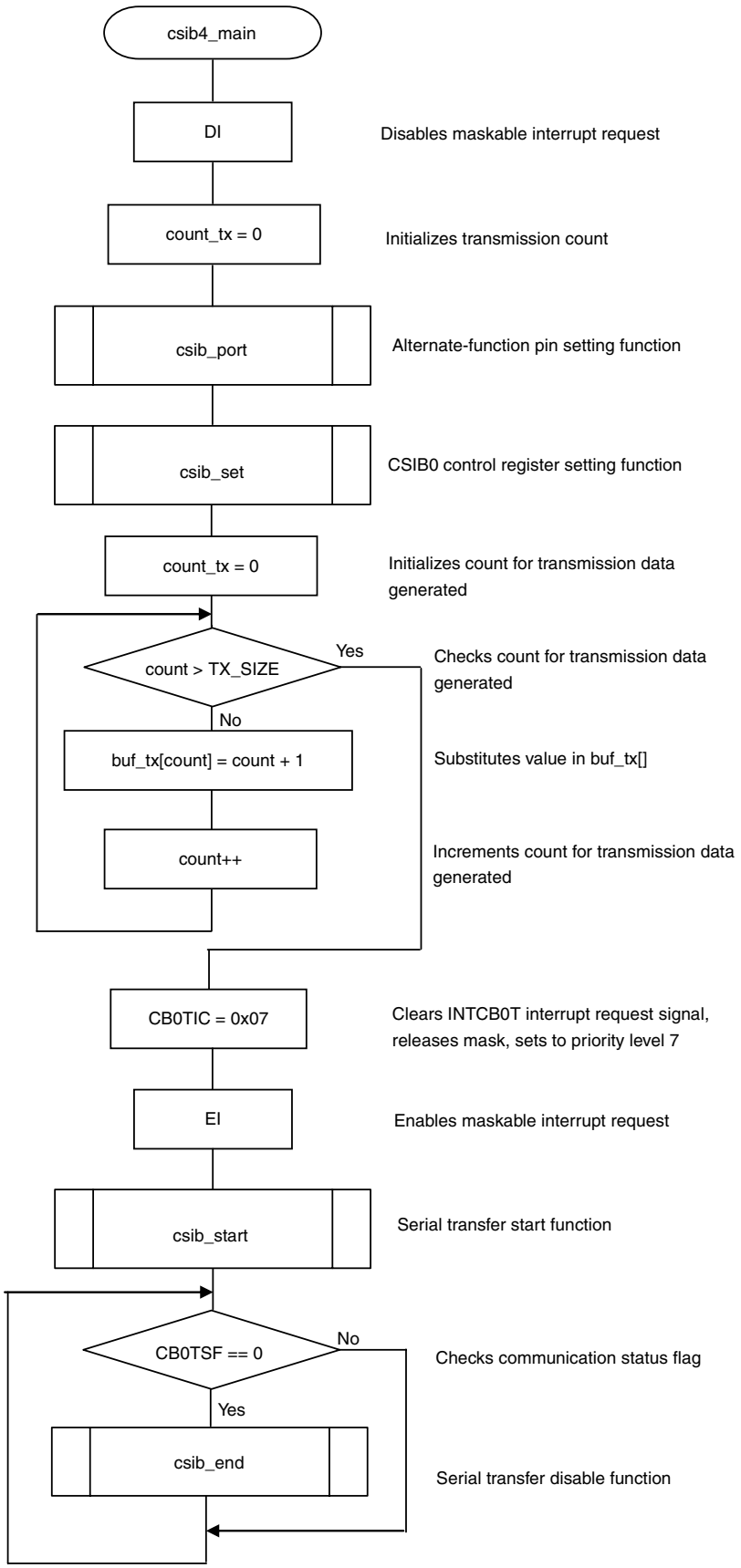
[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x07 (Sets to communication type 1 and external clock.) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x43 (Enables CSIB0 transmission operation, sets to MSB first and continuous transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib4.c
[Caution]	The CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

[Function name]	csib_start
[Processing content]	Enables CSIB0 operation and writes a value to transmit data register.
[SFRs used]	CB0CTL0.CB0PWR: 1 (Enables CSIB0 operation.) CB0TX Transmit data register
[call function]	None
[Variables]	unsigned char buf_tx[ ]: Transmit data storing buffer volatile unsigned char count_tx: Transmission count variable
[File name]	csib4.c
[Caution]	None

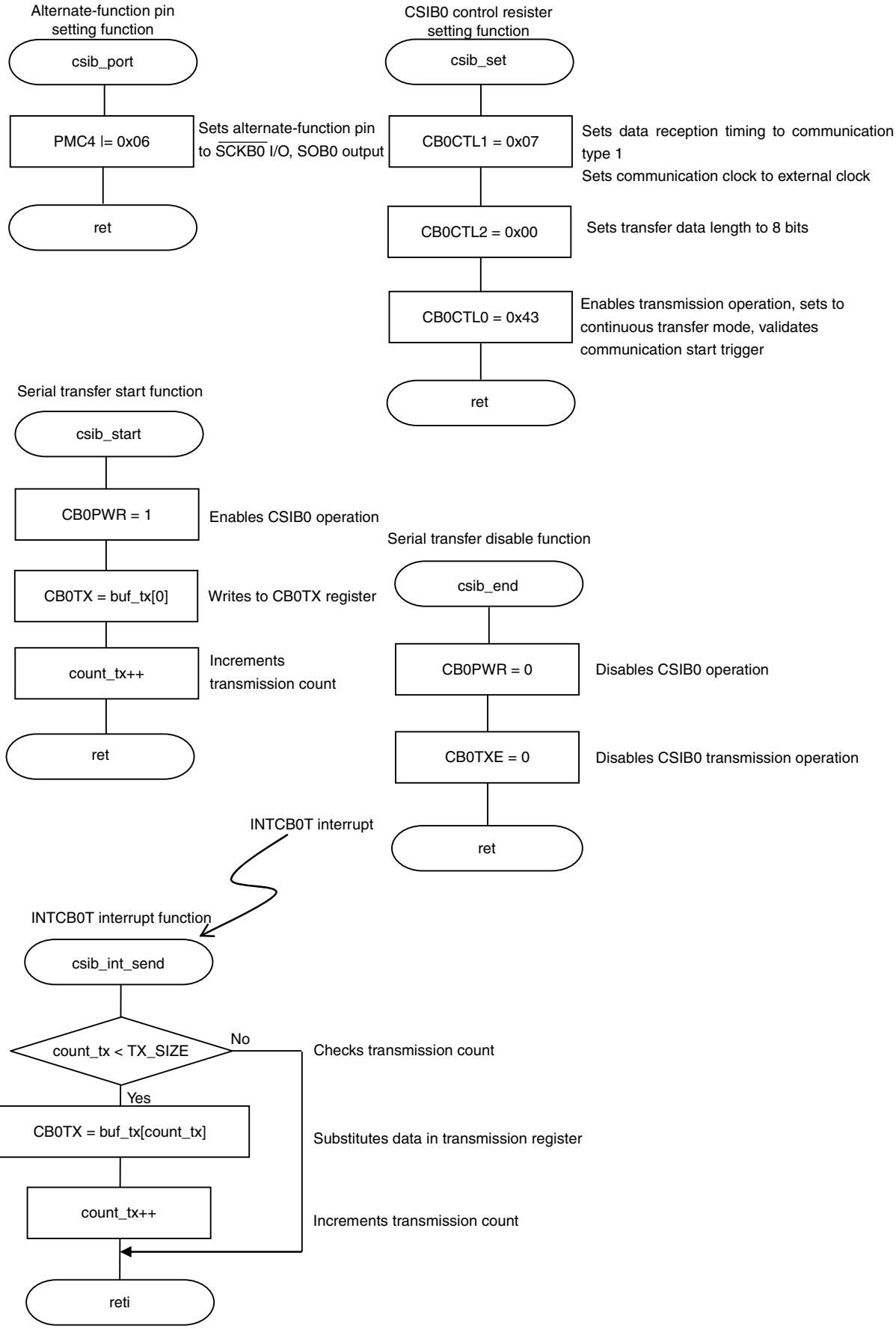
[Function name]	csib_end
[Processing content]	Disables CSIB0 operation and transmission operation.
[SFRs used]	CB0CTL0.CB0PWR: 0 (Disables CSIB0 operation.) CB0CTL0.CB0TXE: 0 (Disables CSIB0 transmission operation.)
[call function]	None
[Variable]	None
[File name]	csib4.c
[Caution]	None

[Function name]	csib_int_send	
[Processing content]	Sets new data for transmitting next data.	
[SFR used]	CB0TX	Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
[File name]	csib4.c	
[Caution]	None	

Continuous transfer mode (slave mode, transmission mode) (1/2)



Continuous transfer mode (slave mode, transmission mode) (2/2)



Clocked serial interface B (CSIBn) (n = 0, 1)

Continuous transfer mode (slave mode, reception mode)

[Function]	Sets communication mode to slave mode and transfer direction mode to MSB first, and performs reception for ten times in continuous transfer mode. Validates communication start trigger and sets communication clock to external clock, and transfer data length to 8 bits.
[Function name]	csib5_main
[Argument]	None
[Processing content]	Sets reception count (count_rx) to initial value 0. Starts reception after calling each setting function.
[SFRs used]	CB0RIC: 0x07 (Clears CSIB0 reception end interrupt request signal (INTCB0R), releases mask, sets to priority level 7.) CB0REIC: 0x07 (Clears CSIB0 reception error interrupt request signal (INTCB0RE), releases mask, sets to priority level 7.) CB0STR.CB0TSF Checks communication status flag
[call function]	csib_port, csib_set, csib_start, csib_end
[Variables]	unsigned char buf_rx[ ]: Receive data storing buffer volatile unsigned char count_rx: Reception count variable
[Interrupts]	csib_int_receive, csib_error
[Interrupt sources]	INTCB0R, INTCB0RE
[File name]	csib5.c
[Caution]	None

[Function name]	csib_port
[Processing content]	Sets port 4 as CSIB0 I/O pin.
[SFR used]	PMC4: 0x05 (Sets $\overline{\text{SCKB0}}$ I/O and SIB0 input.)
[call function]	None
[Variable]	None
[File name]	csib5.c
[Caution]	None

[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x07 (Sets to communication type 1 and sets communication clock to external clock.) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x23 (Enables CSIB0 reception operation, sets to MSB first and continuous transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib5.c
[Caution]	The CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

[Function name]	csib_start	
[Processing content]	Enables CSIB0 operation and performs dummy read on receive data register.	
[SFRs used]	CB0CTL0.CB0PWR: CB0RX	1 (Enables CSIB0 operation.) Receive data register
[call function]	None	
[Variable]	unsigned char buf_rx[ ]:	Receive data storing buffer
[File name]	csib5.c	
[Caution]	None	

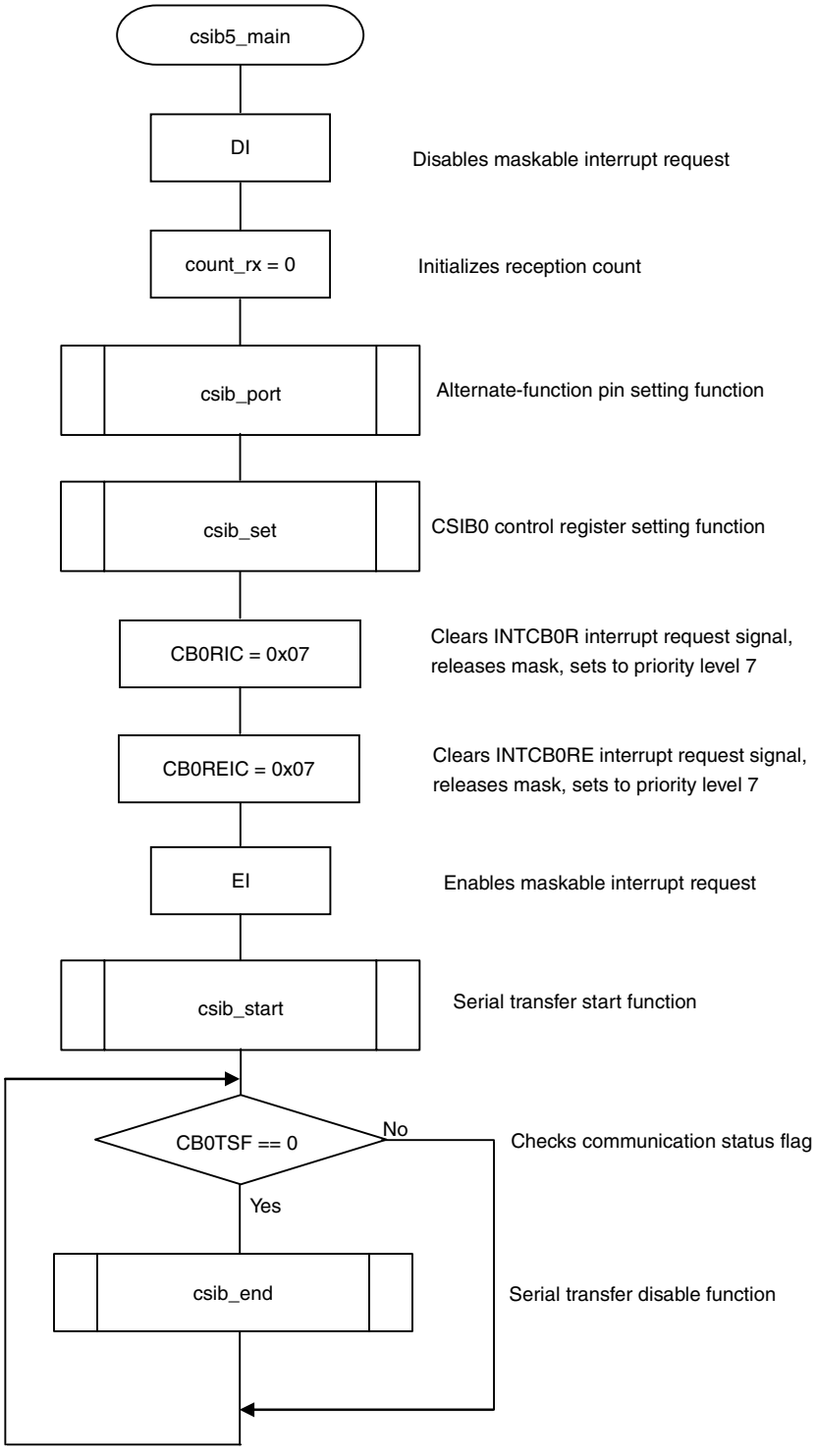
[Function name]	csib_end	
[Processing content]	Disables CSIB0 operation and reception operation.	
[SFRs used]	CB0CTL0.CB0PWR: CB0CTL0.CB0RXE:	0 (Disables CSIB0 operation.) 0 (Disables CSIB0 reception operation.)
[call function]	None	
[Variable]	None	
[File name]	csib5.c	
[Caution]	None	

Interrupt functions

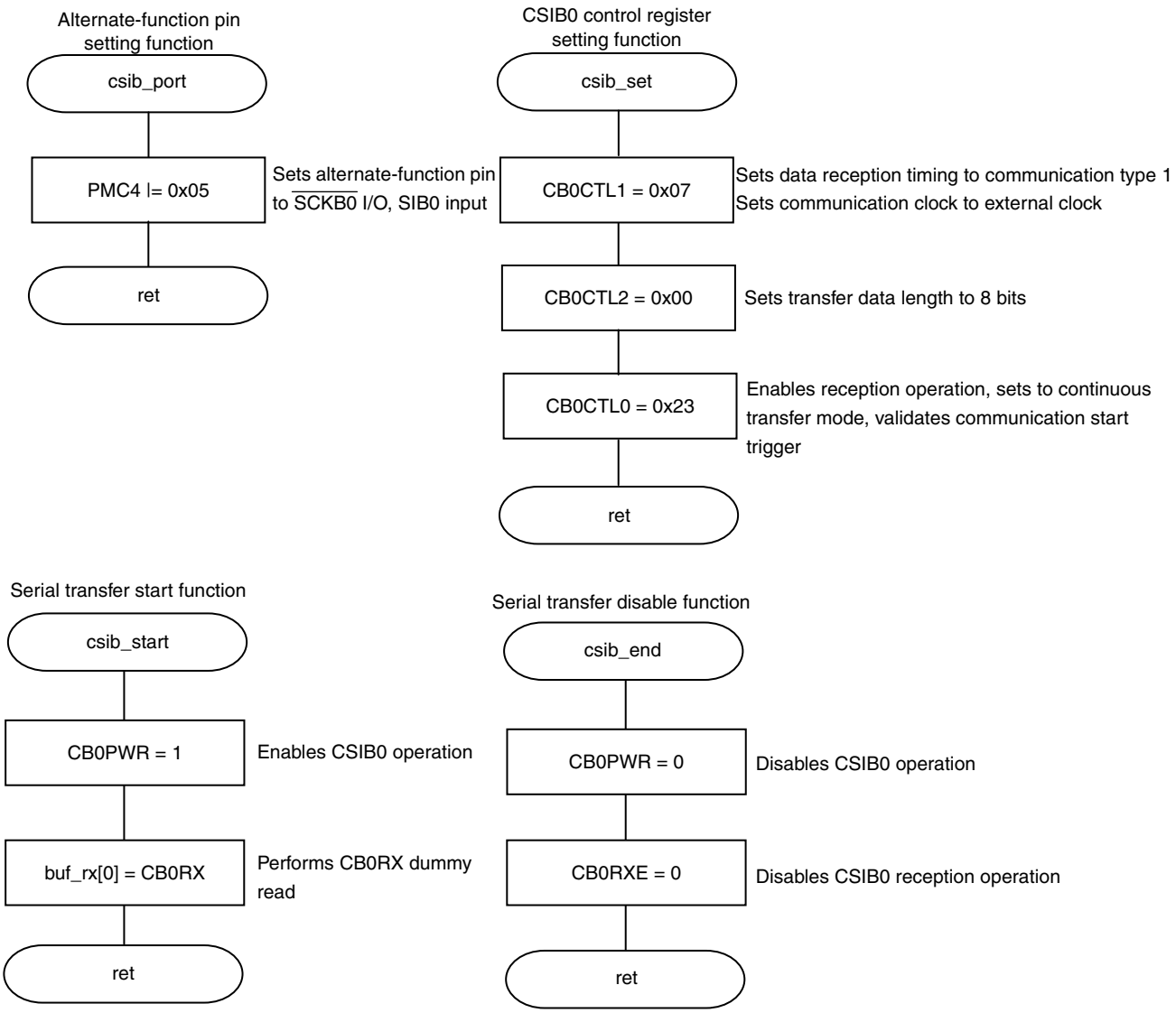
[Function name]	csib_int_receive	
[Processing content]	Stores receive data to buffer.	
[SFRs used]	CB0RX	Receive data register
	CB0CTL0.CB0SCE:	0 (Invalidates communication start trigger.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib5.c	
[Caution]	None	

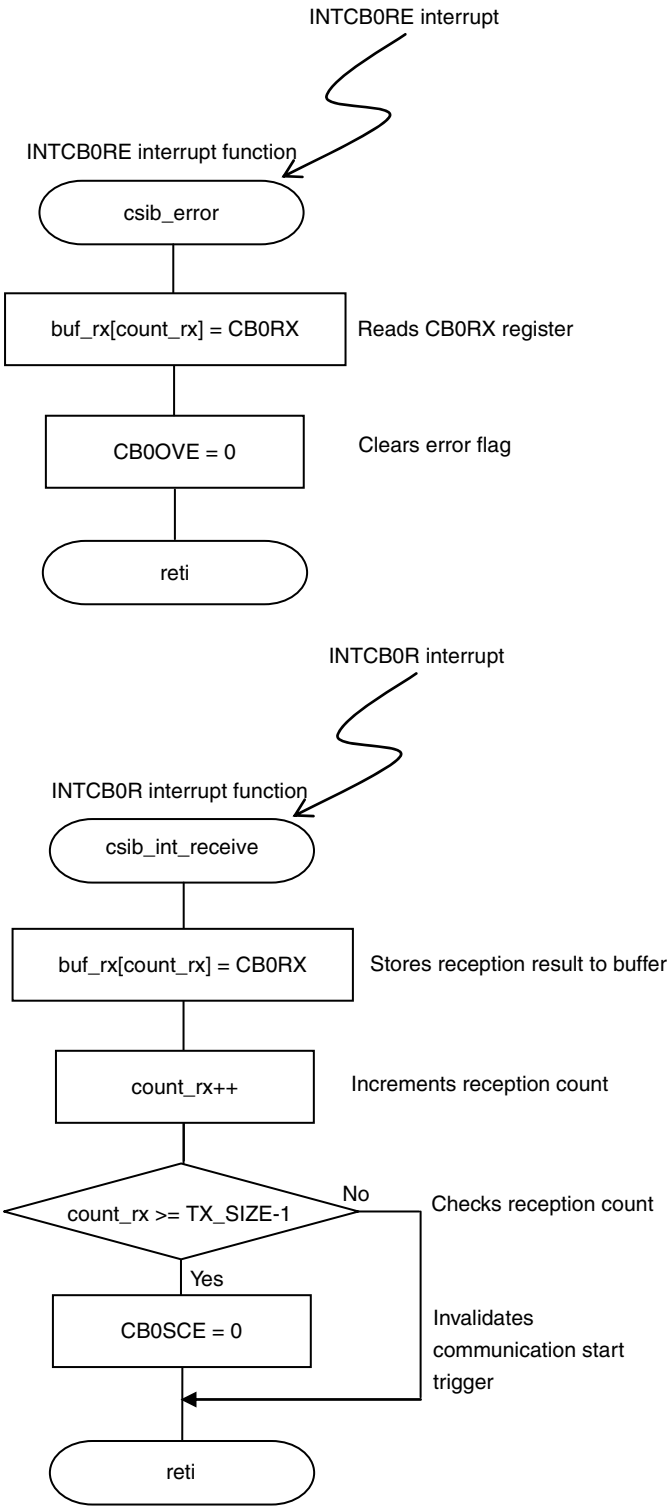
[Function name]	csib_error	
[Processing content]	Clears reception error flag.	
[SFR used]	CB0RX	Receive data register
	CB0STR.CB0OVE:	0 (Clears overrun error flag.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib5.c	
[Caution]	None	

Continuous transfer mode (slave mode, reception mode) (1/3)



Continuous transfer mode (slave mode, reception mode) (2/3)




INTCB0RE interrupt
INTCB0RE interrupt function
csib_error
buf_rx[count_rx] = CB0RX
Reads CB0RX register
CB0OVE = 0
Clears error flag
reti
INTCB0R interrupt
INTCB0R interrupt function
csib_int_receive
buf_rx[count_rx] = CB0RX
Stores reception result to buffer
count_rx++
Increments reception count
count_rx >= TX_SIZE-1
No
Checks reception count
Yes
CB0SCE = 0
Invalidates
communication start
trigger
reti

Clocked serial interface B (CSIBn) (n = 0, 1)

Continuous transfer mode (slave mode, transmission/reception mode)

[Function]	Sets communication mode to slave mode and transfer direction mode to MSB first, and performs data transmission/reception for ten times each in continuous transfer mode. Validates communication start trigger and sets communication clock to external clock, and transfer data length to 8 bits.		
[Function name]	csib6_main		
[Argument]	None		
[Processing content]	Sets transmission count (count_tx) to initial value 0. Sets reception count (count_rx) to initial value 0 and starts transmission after calling each setting function.		
[SFRs used]	CB0REIC:	0x07	(Clears CSIB0 reception error interrupt request signal (INTCB0RE), releases mask, sets to priority level 7.)
	CB0RIC:	0x07	(Clears CSIB0 reception end interrupt request signal (INTCB0R), releases mask, sets to priority level 7.)
	CB0TIC:	0x07	(Clears CSIB0 transmission enable interrupt request signal (INTCB0T), releases mask, sets to priority level 7.)
	CB0STR.CB0TSF	Communication status flag	
[call function]	csib_port, csib_set, csib_start, csib_end		
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer	
	unsigned char buf_rx[ ]:	Receive data storing buffer	
	volatile unsigned char count_tx:	Transmission count variable	
	volatile unsigned char count_rx:	Reception count variable	
	unsigned char count:	Transfer data generating variable	
[Interrupts]	csib_error, csib_int_send, csib_int_receive		
[Interrupt sources]	INTCB0RE, INTCB0T, INTCB0R		
[File name]	csib6.c		
[Caution]	None		

[Function name]	csib_port
[Processing content]	Sets port 4 as CSIB0 I/O pin.
[SFR used]	PMC4: 0x07 (Sets $\overline{\text{SCKB0}}$ I/O, SOB0 output and SIB0 input.)
[call function]	None
[Variable]	None
[File name]	csib6.c
[Caution]	None

[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x07 (Sets to communication type 1 and sets communication clock to external clock.) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x63 (Enables CSIB0 transmission and reception operation, sets to MSB first and continuous transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib6.c
[Caution]	The CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

[Function name]	csib_start
[Processing content]	Enables CSIB0 operation and writes a value to transmit data register.
[SFRs used]	CB0CTL0.CB0PWR: 1 (Enables CSIB0 operation.) CB0TX Transmit data register
[call function]	None
[Variables]	unsigned char buf_tx[ ]: Transmit data storing buffer volatile unsigned char count_tx: Transmission count variable
[File name]	csib6.c
[Caution]	None

[Function name]	csib_end	
[Processing content]	Disables CSIB0 operation and transmission/reception operation.	
[SFRs used]	CB0CTL0.CB0PWR:	0 (Disables CSIB0 operation.)
	CB0CTL0.CS0RXE:	0 (Disables CSIB0 reception operation.)
	CB0CTL0.CB0TXE:	0 (Disables CSIB0 transmission operation.)
[call function]	None	
[Variable]	None	
[File name]	csib6.c	
[Caution]	None	

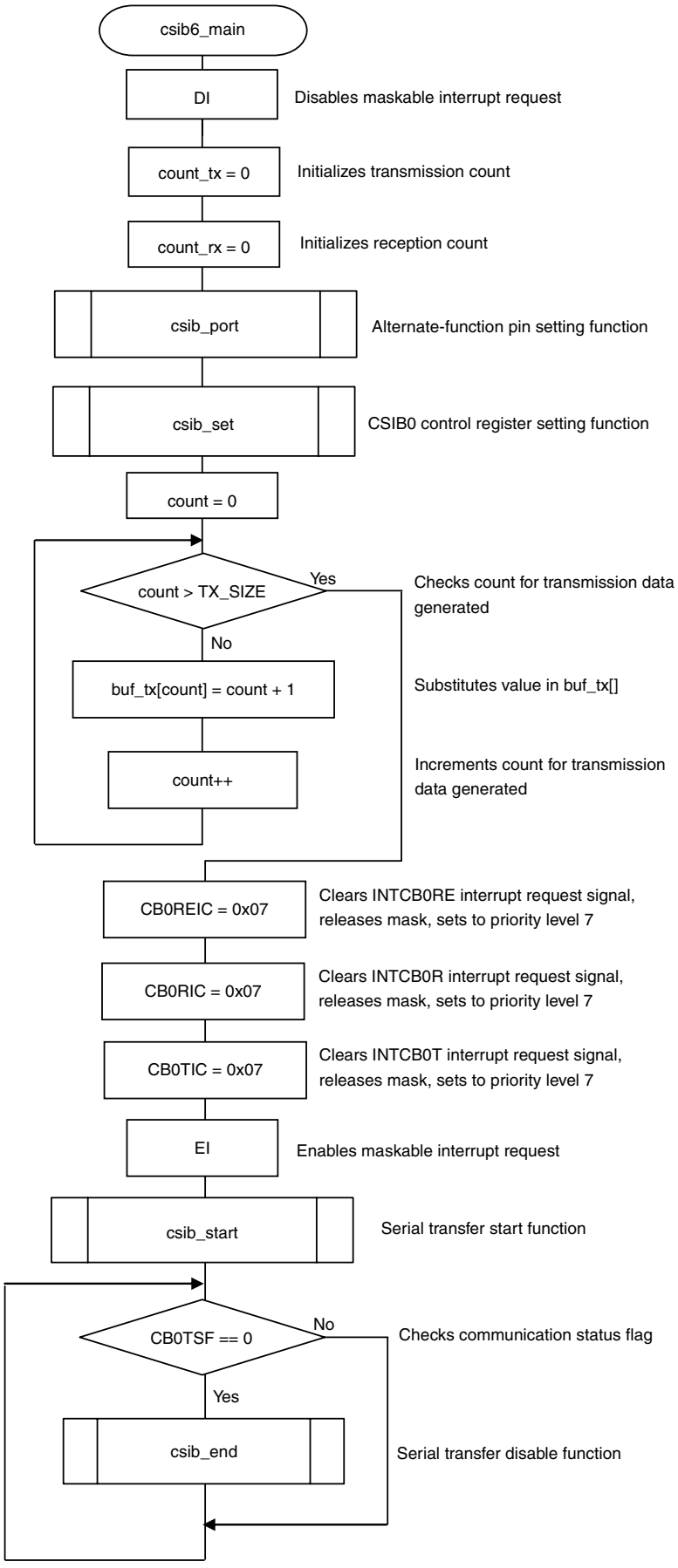
Interrupt functions

[Function name]	csib_error	
[Processing content]	Clears reception error flag.	
[SFRs used]	CB0RX	Receive data register
	CB0STR.CB0OVE:	0 (Clears overrun error flag.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib6.c	
[Caution]	None	

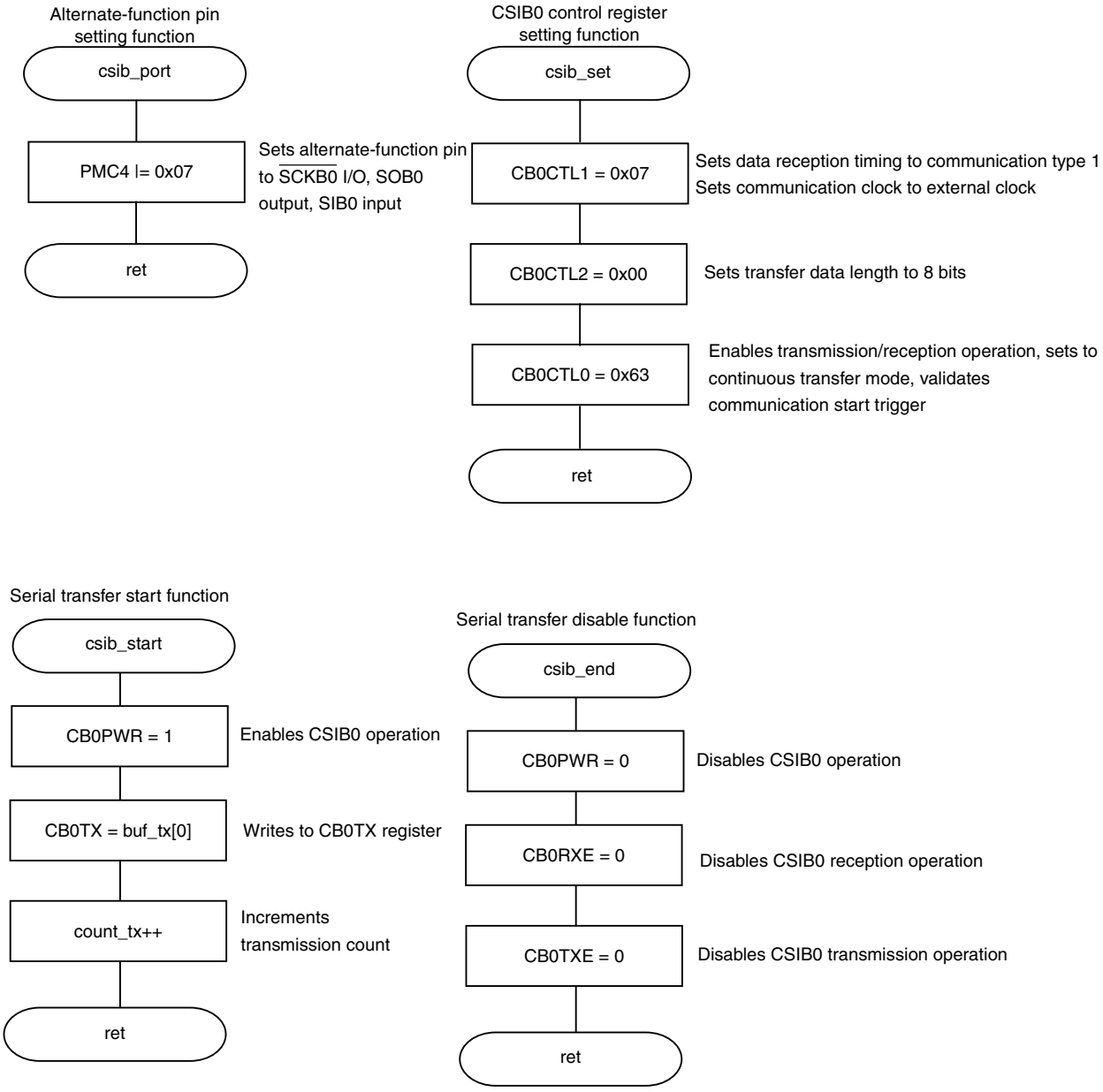
[Function name]	csib_int_send	
[Processing content]	Sets new data for transmitting next data.	
[SFR used]	CB0TX	Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
[File name]	csib6.c	
[Caution]	None	

[Function name]	csib_int_receive	
[Processing content]	Stores receive data to buffer.	
[SFR used]	CB0RX	Receive data register
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib6.c	
[Caution]	None	

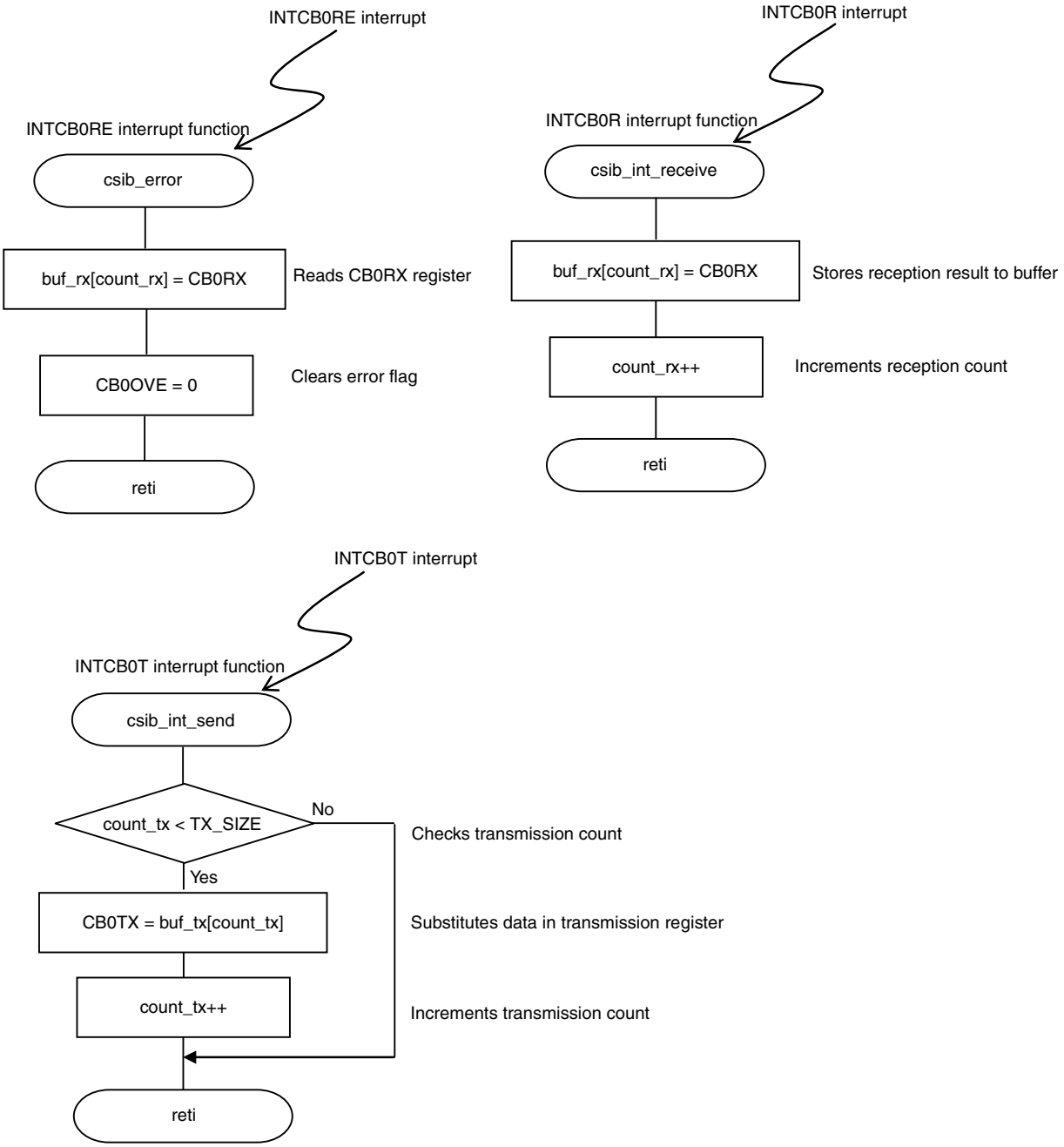
Continuous transfer mode (slave mode, transmission/reception mode) (1/3)



Continuous transfer mode (slave mode, transmission/reception mode) (2/3)



Continuous transfer mode (slave mode, transmission/reception mode) (3/3)



Clocked serial interface B (CSIBn) (n = 0, 1)

Single transfer mode (master mode, transmission mode)

[Function]	Sets communication mode to master mode and transfer direction mode to MSB first, and performs data transmission for ten times in single transfer mode. Validates communication start trigger, and sets communication clock to fxx/256 and transfer data length to 8 bits.	
[Function name]	csib7_main	
[Argument]	None	
[Processing content]	Sets transmission count (count_tx) to initial value 0. Starts transmission after calling each setting function.	
[SFR used]	CB0RIC: 0x07 (Clears CSIB0 reception end interrupt request signal (INTCB0R), releases mask, sets to priority level 7.)	
[call function]	csib_port, csib_set, csib_start, csib_end	
[Variables]	unsigned char buf_tx[ ]: volatile unsigned char count_tx: unsigned char count:	Transmit data storing buffer Transmission count variable Transfer data generating variable
[Interrupt]	csib_int_send	
[Interrupt source]	INTCB0R	
[File name]	csib7.c	
[Caution]	None	

[Function name]	csib_port
[Processing content]	Sets port 4 as CSIB0 I/O pin.
[SFR used]	PMC4: 0x06 (Sets $\overline{\text{SCKB0}}$ I/O and SOB0 output.)
[call function]	None
[Variable]	None
[File name]	csib7.c
[Caution]	None

[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x06 (Sets to communication type 1 and sets communication clock to fXX/256 (0.25 MHz).) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x41 (Enables CSIB0 transmission operation, sets to MSB first and single transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib7.c
[Caution]	The CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

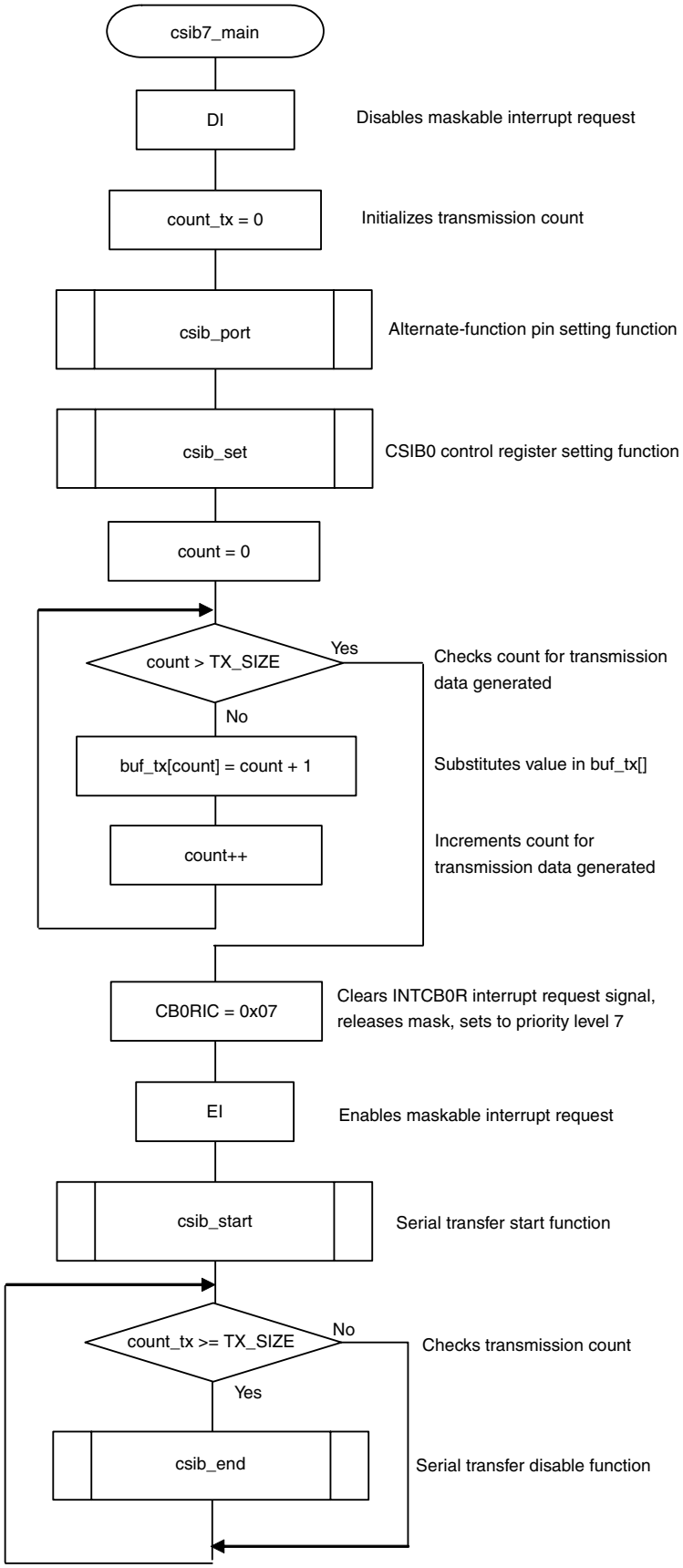
[Function name]	csib_start	
[Processing content]	Enables CSIB0 operation and writes a value to transmit data register.	
[SFRs used]	CB0CTL0.CB0PWR:	1 (Enables CSIB0 operation.)
	CB0TX	Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
[File name]	csib7.c	
[Caution]	None	

[Function name]	csib_end	
[Processing content]	Disables CSIB0 operation and transmission operation.	
[SFRs used]	CB0CTL0.CB0PWR:	0 (Disables CSIB0 operation.)
	CB0CTL0.CB0TXE:	0 (Disables CSIB0 transmission operation.)
[call function]	None	
[Variable]	None	
[File name]	csib7.c	
[Caution]	None	

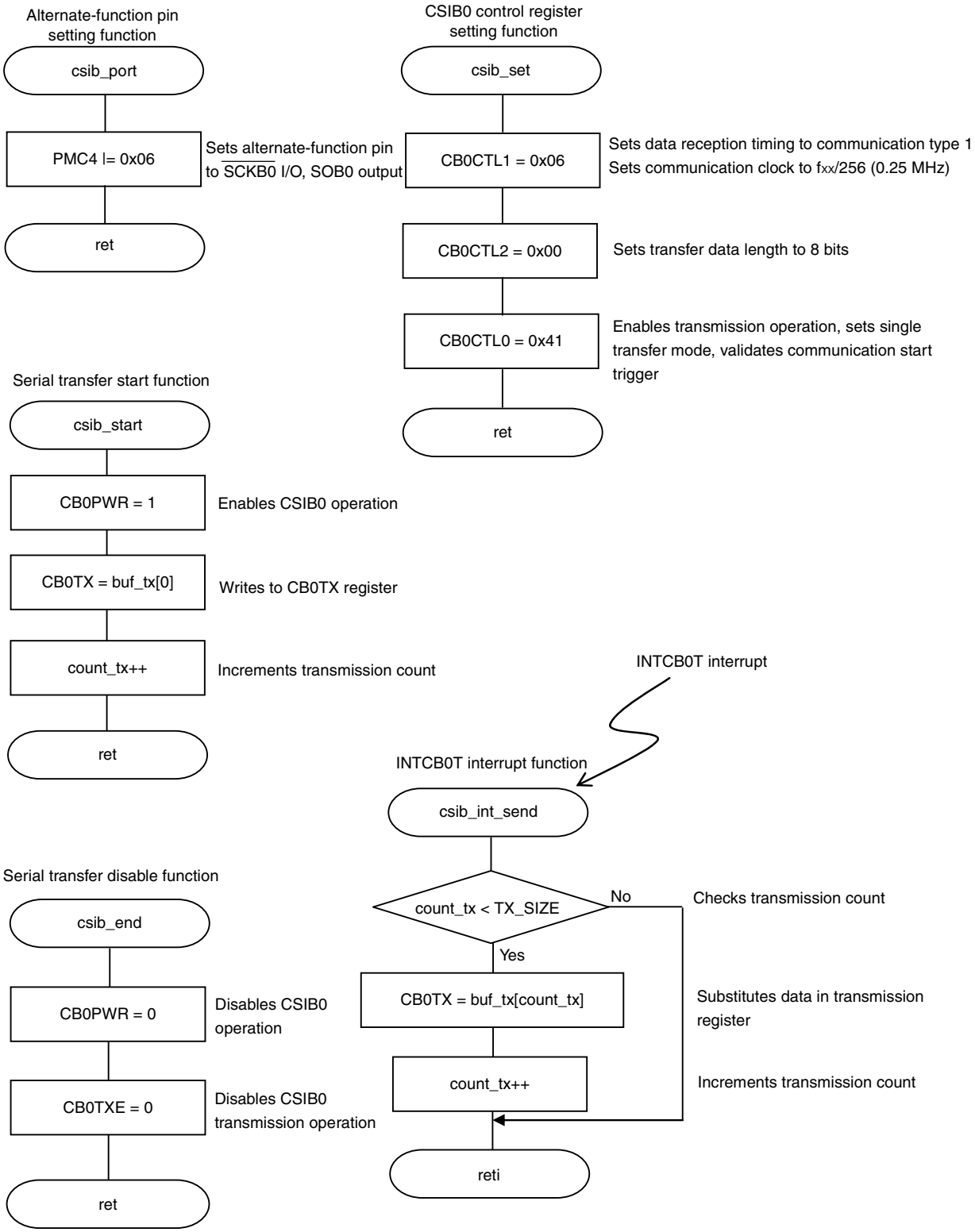
Interrupt function

[Function name]	csib_int_send	
[Processing content]	Sets new data for transmitting next data.	
[SFR used]	CB0TX Transmit data register	
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
[File name]	csib7.c	
[Caution]	None	

Single transfer mode (master mode, transmission mode) (1/2)



Single transfer mode (master mode, transmission mode) (2/2)



Clocked serial interface B (CSIBn) (n = 0, 1)

Single transfer mode (master mode, reception mode)

[Function]	Sets communication mode to master mode and transfer direction mode to MSB first, and performs data reception for ten times in single transfer mode. Validates communication start trigger and sets communication clock to fxx/256, and transfer data length to 8 bits.
[Function name]	csib8_main
[Argument]	None
[Processing content]	Sets reception count (count_rx) to initial value 0. Starts reception after calling each setting function.
[SFRs used]	CB0RIC: 0x07 (Clears CSIB0 reception end interrupt request signal (INTCB0R), releases mask, sets to priority level 7.) CB0REIC: 0x07 (Clears CSIB0 reception error interrupt request signal (INTCB0RE), releases mask, sets to priority level 7.)
[call functions]	csib_port, csib_set, csib_start, csib_end
[Variables]	unsigned char buf_rx[ ]: Receive data storing buffer volatile unsigned char count_rx: Reception count variable
[Interrupts]	csib_int_receive, csib_error
[Interrupt sources]	INTCB0R, INTCB0RE
[File name]	csib8.c
[Caution]	None

[Function name]	csib_port
[Processing content]	Sets port 4 as CSIB0 I/O pin.
[SFR used]	PMC4: 0x05 (Sets $\overline{\text{SCKB0}}$ I/O and SIB0 input.)
[call function]	None
[Variable]	None
[File name]	csib8.c
[Caution]	None

[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x06 (Sets to communication type 1 and sets communication clock to fXX/256 (0.25 MHz).) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x21 (Enables CSIB0 reception operation, sets to MSB first and single transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib8.c
[Caution]	The CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

[Function name]	csib_start	
[Processing content]	Enables CSIB0 operation and performs dummy read on receive data register.	
[SFRs used]	CB0CTL0.CB0PWR: CB0RX	1 (Enables CSIB0 operation.) Receive data register
[call function]	None	
[Variable]	unsigned char buf_rx[ ]:	Receive data storing buffer
[File name]	csib8.c	
[Caution]	None	

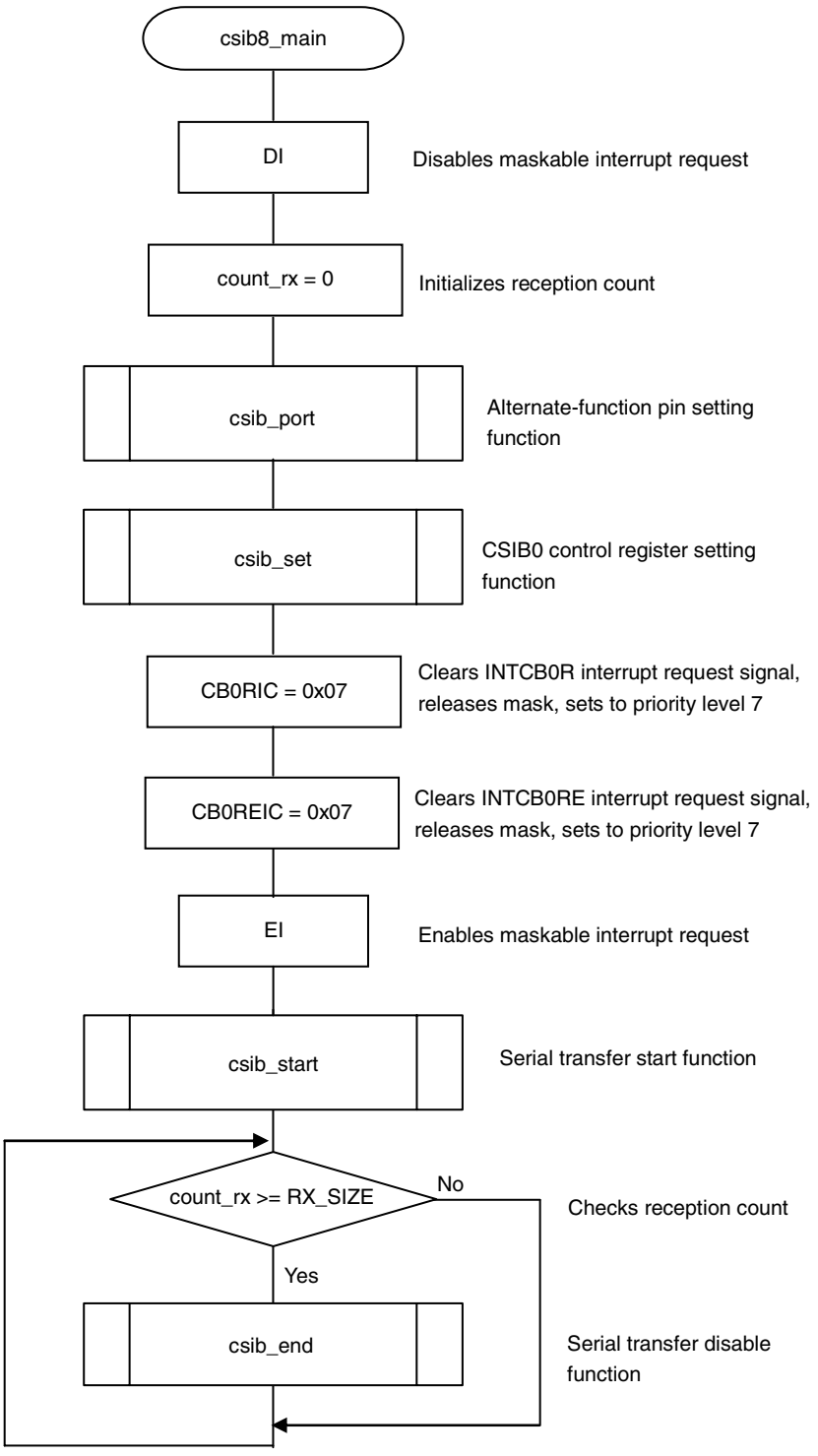
[Function name]	csib_end	
[Processing content]	Disables CSIB0 operation and reception operation.	
[SFRs used]	CB0CTL0.CB0PWR: CB0CTL0.CB0RXE:	0 (Disables CSIB0 operation.) 0 (Disables CSIB0 reception operation.)
[call function]	None	
[Variable]	None	
[File name]	csib8.c	
[Caution]	None	

Interrupt functions

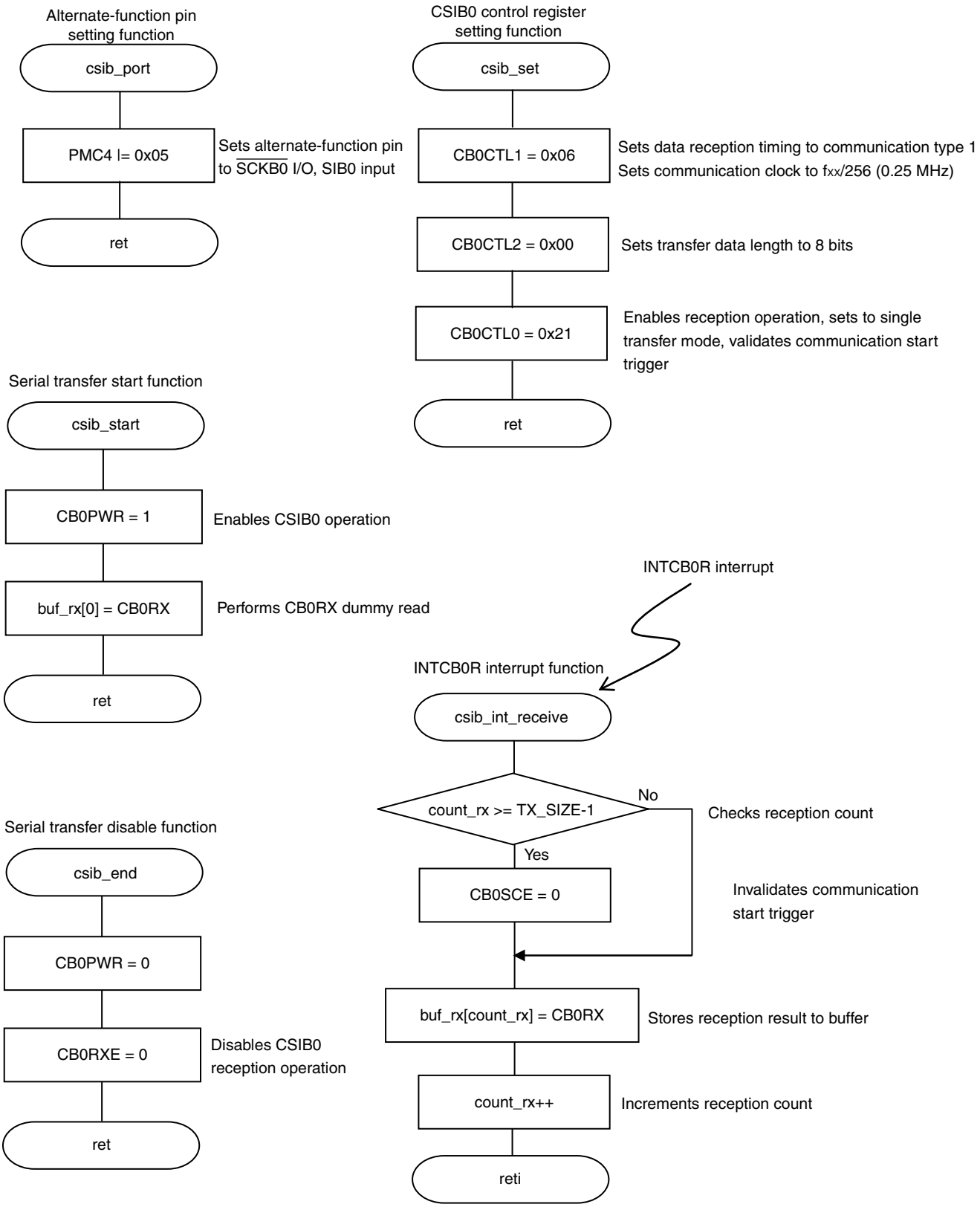
[Function name]	csib_int_receive	
[Processing content]	Stores receive data to buffer.	
[SFRs used]	CB0RX	Receive data register
	CB0CTL0.CB0SCE:	0 (Invalidates communication start trigger.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib8.c	
[Caution]	None	

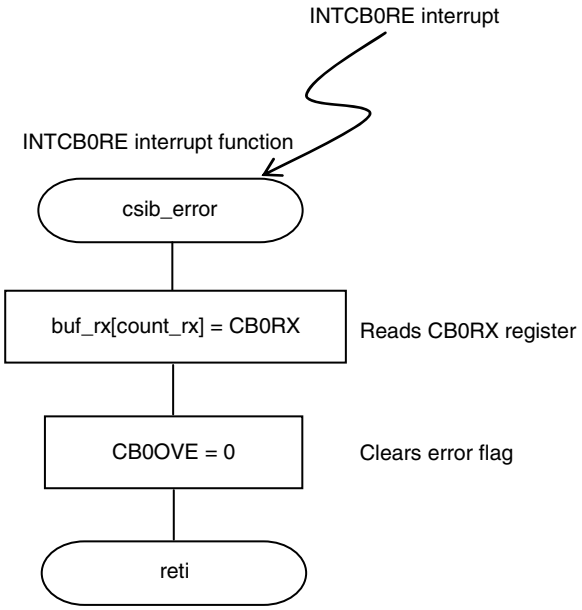
[Function name]	csib_error	
[Processing content]	Clears reception error flag.	
[SFRs used]	CB0RX	Receive data register
	CB0STR.CB0OVE:	0 (Clears overrun error flag.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib8.c	
[Caution]	None	

Single transfer mode (master mode, reception mode) (1/3)



Single transfer mode (master mode, reception mode) (2/3)




INTCB0RE interrupt
INTCB0RE interrupt function
csib_error
buf_rx[count_rx] = CB0RX
Reads CB0RX register
CB0OVE = 0
Clears error flag
reti

Clocked serial interface B (CSIBn) (n = 0, 1)

Single transfer mode (master mode, transmission/reception mode)

[Function]	Sets communication mode to master mode and transfer direction mode to MSB first, and performs transmission/reception for ten times each in single transfer mode. Validates communication start trigger and sets communication clock to fxx/256, and transfer data length to 8 bits.
[Function name]	csib9_main
[Argument]	None
[Processing content]	Sets transmission count (count_tx) to initial value 0. Sets reception count (count_rx) to initial value 0 and starts transmission/reception after calling each setting function.
[SFRs used]	CB0REIC: 0x07 (Clears CSIB0 reception error interrupt request signal (INTCB0RE), releases mask, sets to priority level 7.) CB0RIC: 0x07 (Clears CSIB0 reception end interrupt request signal (INTCB0R), releases mask, sets to priority level 7.) CB0TIC: 0x07 (Clears CSIB0 transmission enable interrupt request signal (INTCB0T), releases mask, sets to priority level 7.)
[call functions]	csib_port, csib_set, csib_start, csib_end
[Variables]	unsigned char buf_tx[ ]: Transmit data storing buffer unsigned char buf_rx[ ]: Receive data storing buffer volatile unsigned char count_tx: Transmission count variable volatile unsigned char count_rx: Reception count variable unsigned char count: Transfer data generating variable
[Interrupts]	csib_error, csib_int_send, csib_int_receive
[Interrupt sources]	INTCB0RE, INTCB0T, INTCB0R
[File name]	csib9.c
[Caution]	None

[Function name]	csib_port
[Processing content]	Sets port 4 as CSIB0 I/O pin.
[SFR used]	PMC4: 0x07 (Sets $\overline{\text{SCKB0}}$ I/O, SOB0 output and SIB0 input.)
[call function]	None
[Variable]	None
[File name]	csib9.c
[Caution]	None

[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x06 (Sets to communication type 1 and sets communication clock to fxx/256 (0.25 MHz).) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x61 (Enables CSIB0 transmission and reception operation, sets to MSB first and single transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib9.c
[Caution]	The CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

[Function name]	csib_start	
[Processing content]	Enables CSIB0 operation and writes a value to transmit data register.	
[SFRs used]	CB0CTL0.CB0PWR:	1 (Enables CSIB0 operation.)
	CB0TX	Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
[File name]	csib9.c	
[Caution]	None	

[Function name]	csib_end	
[Processing content]	Disables CSIB0 operation and transmission/reception operation.	
[SFRs used]	CB0CTL0.CB0PWR:	0 (Disables CSIB0 operation.)
	CB0CTL0.CB0RXE:	0 (Disables CSIB0 reception operation.)
	CB0CTL0.CB0TXE:	0 (Disables CSIB0 transmission operation.)
[call function]	None	
[Variable]	None	
[File name]	csib9.c	
[Caution]	None	

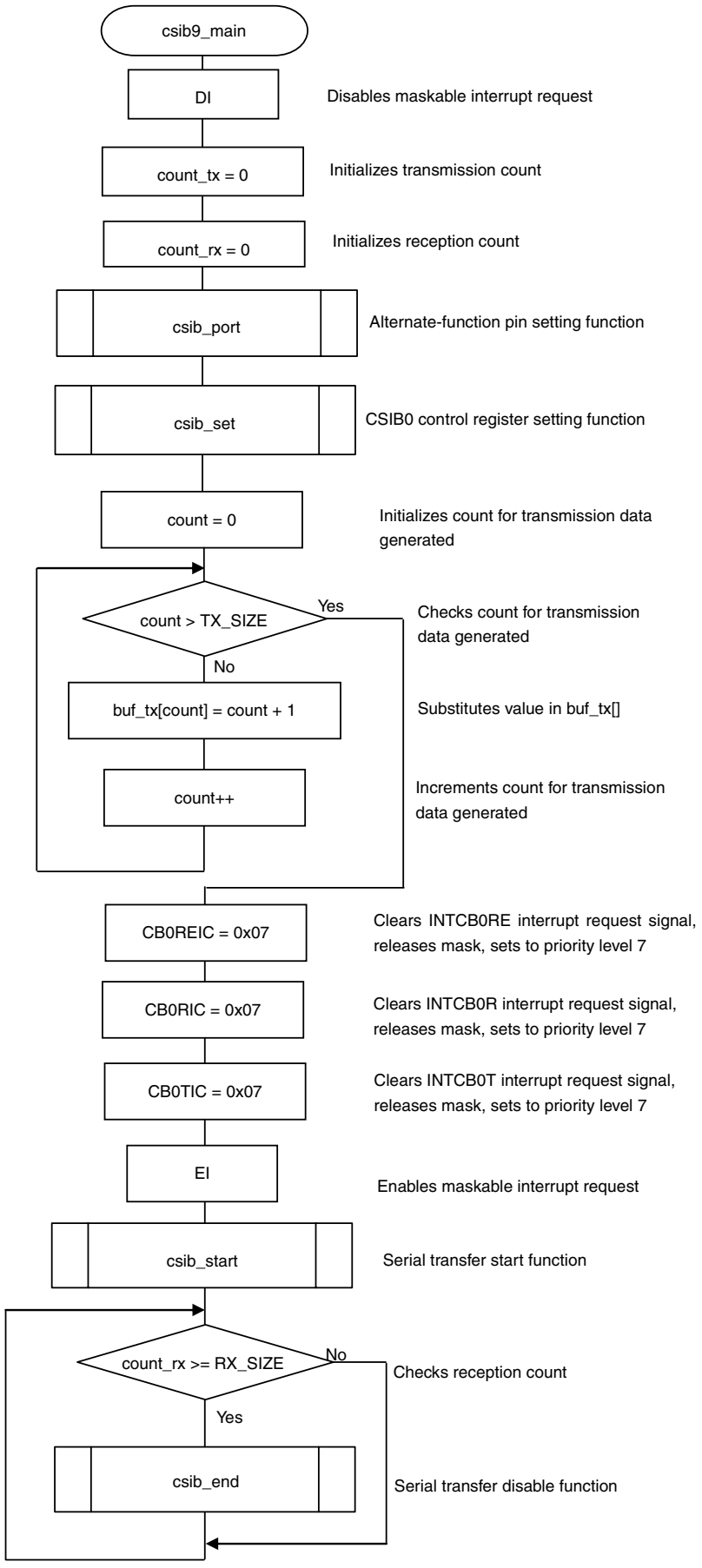
Interrupt functions

[Function name]	csib_error	
[Processing content]	Clears reception error flag.	
[SFRs used]	CB0RX	Transmit data register
	CB0STR.CB0OVE:	0 (Clears overrun error flag.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib9.c	
[Caution]	None	

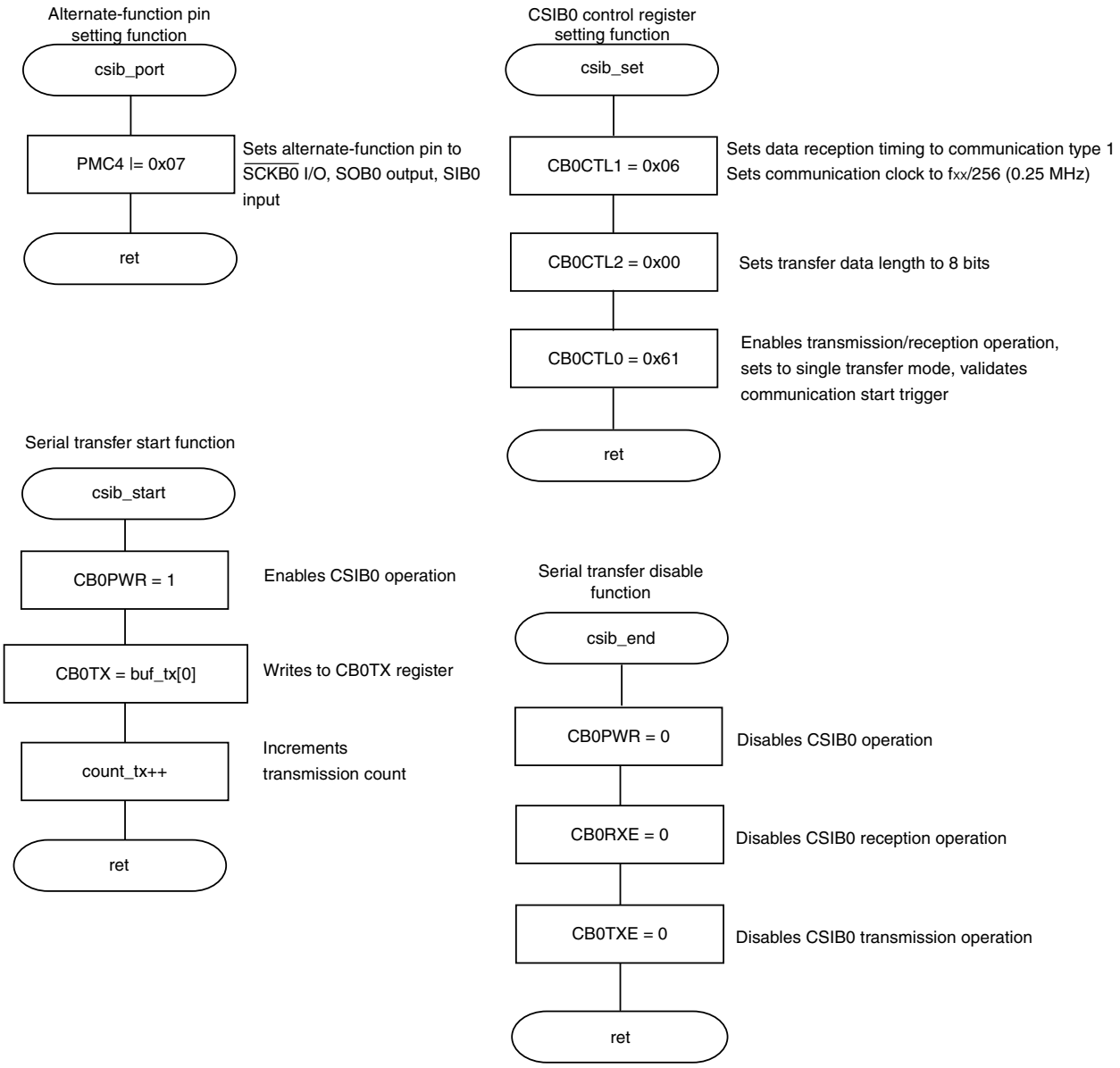
[Function name]	csib_int_send	
[Processing content]	Sets new data for transmitting next data.	
[SFR used]	CB0TX	Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
[File name]	csib9.c	
[Caution]	None	

[Function name]	csib_int_receive	
[Processing content]	Stores receive data to buffer.	
[SFR used]	CB0RX	Receive data register
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib9.c	
[Caution]	None	

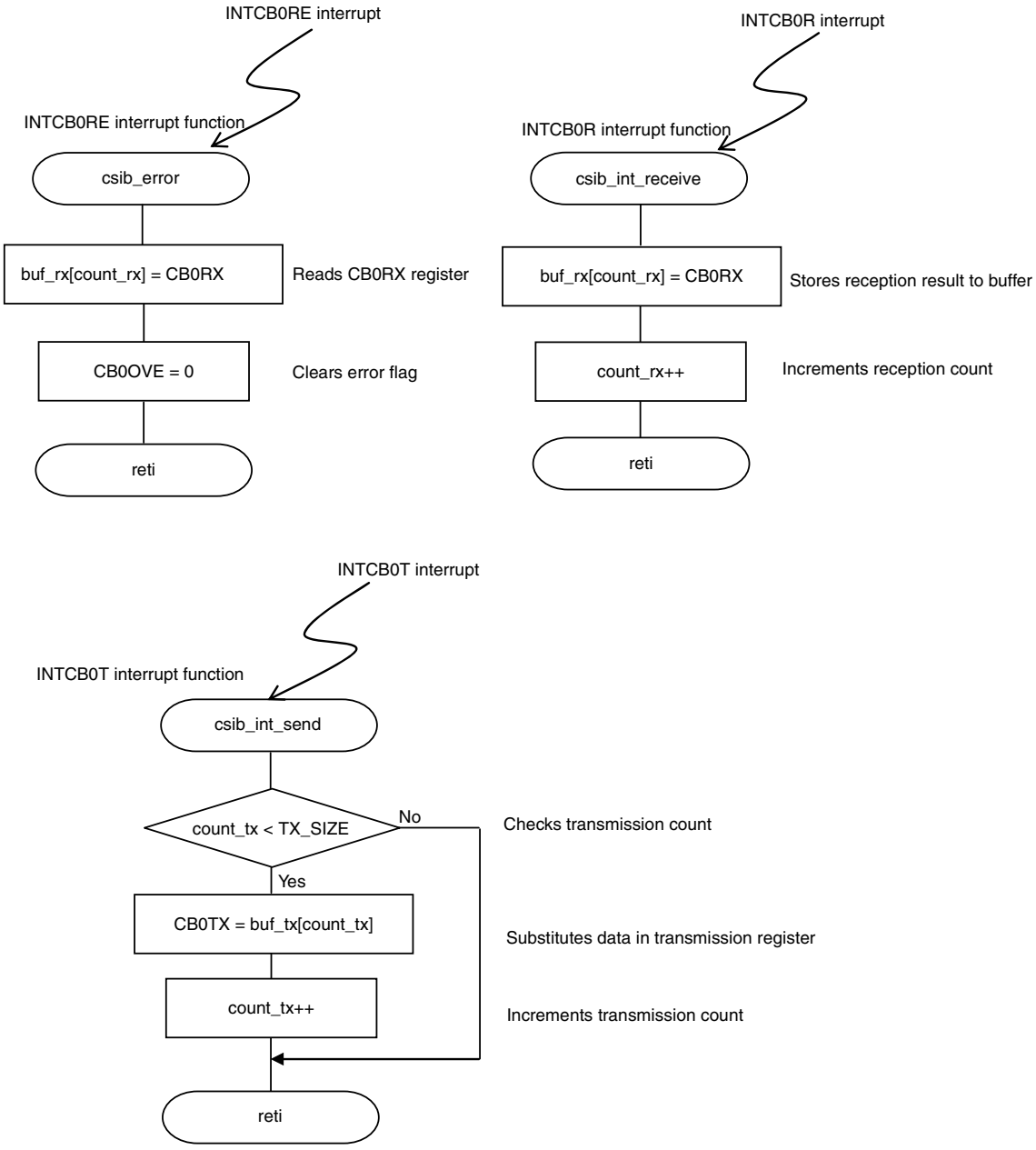
Single transfer mode (master mode, transmission/reception mode) (1/3)



Single transfer mode (master mode, transmission/reception mode) (2/3)



Single transfer mode (master mode, transmission/reception mode) (3/3)



Clocked serial interface B (CSIBn) (n = 0, 1)

Single transfer mode (slave mode, transmission mode)

[Function]	Sets communication mode to slave mode and transfer direction mode to MSB first, and performs transmission for ten times in single transfer mode. Validates communication start trigger and sets communication clock to external clock, and transfer data length to 8 bits.	
[Function name]	csib10_main	
[Argument]	None	
[Processing content]	Sets transmission count (count_tx) to initial value 0. Starts transmission after calling each setting function.	
[SFR used]	CB0RIC: 0x07 (Clears CSIB0 reception end interrupt request signal (INTCB0R), releases mask, sets to priority level 7.)	
[call functions]	csib_port, csib_set, csib_start, csib_end	
[Variables]	unsigned char buf_tx[ ]: volatile unsigned char count_tx: unsigned char count:	Transmit data storing buffer Transmission count variable Transfer data generating variable
[Interrupt]	csib_int_send	
[Interrupt source]	INTCB0R	
[File name]	csib10.c	
[Caution]	None	

[Function name]	csib_port
[Processing content]	Sets port 4 as CSIB0 I/O pin.
[SFR used]	PMC4: 0x06 (Sets $\overline{\text{SCKB0}}$ I/O and SOB0 outputs.)
[call function]	None
[Variable]	None
[File name]	csib10.c
[Caution]	None

[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x07 (Sets to communication type 1 and external clock.) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x41 (Enables CSIB0 transmission operation, sets to MSB first and single transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib10.c
[Caution]	The CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

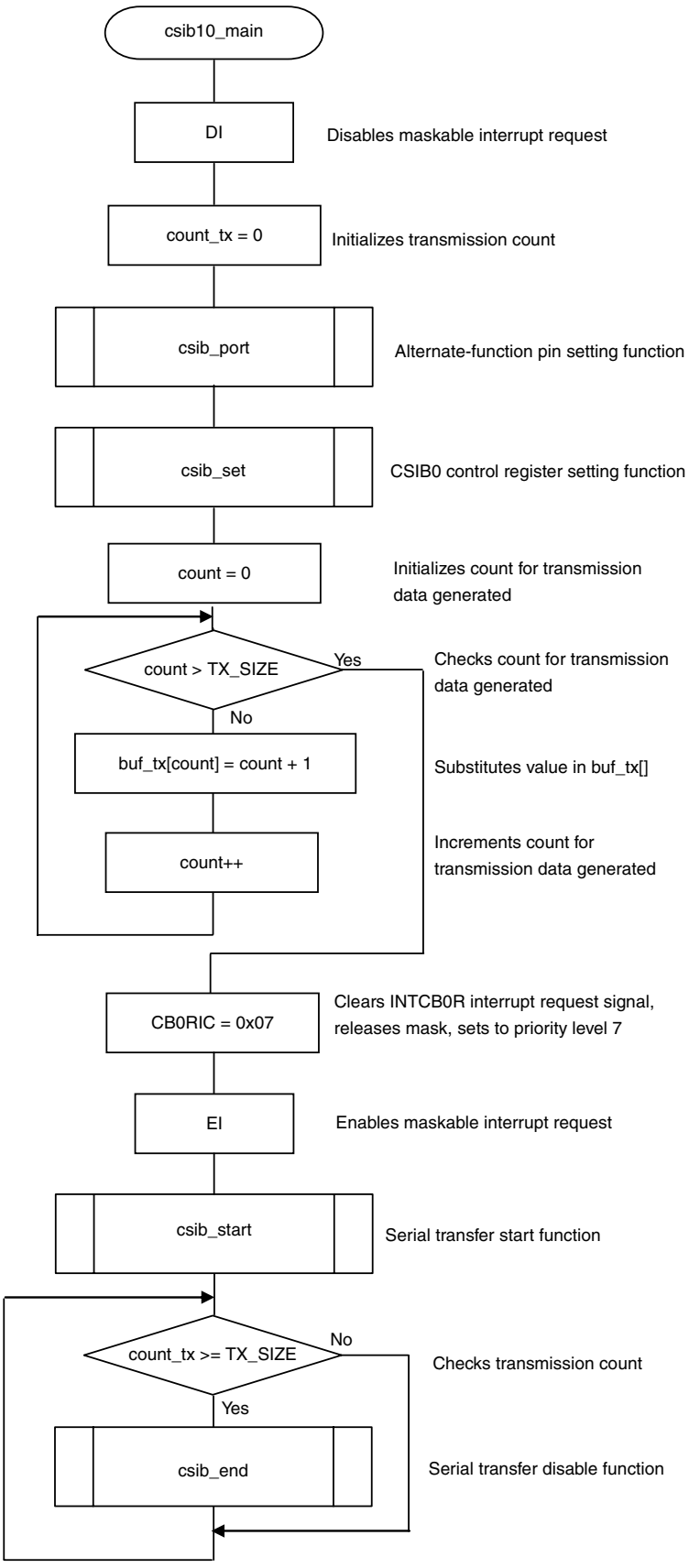
[Function name]	csib_start
[Processing content]	Enables CSIB0 operation and writes a value to transmit data register.
[SFRs used]	CB0CTL0.CB0PWR: 1 (Enables CSIB0 operation.) CB0TX Transmit data register
[call function]	None
[Variables]	unsigned char buf_tx[ ]: Transmit data storing buffer volatile unsigned char count_tx: Transmission count variable
[File name]	csib10.c
[Caution]	None

[Function name]	csib_end
[Processing content]	Disables CSIB0 operation and transmission operation.
[SFRs used]	CB0CTL0.CB0PWR: 0 (Disables CSIB0 operation.) CB0CTL0.CB0TXE: 0 (Disables CSIB0 transmission operation.)
[call function]	None
[Variable]	None
[File name]	csib10.c
[Caution]	None

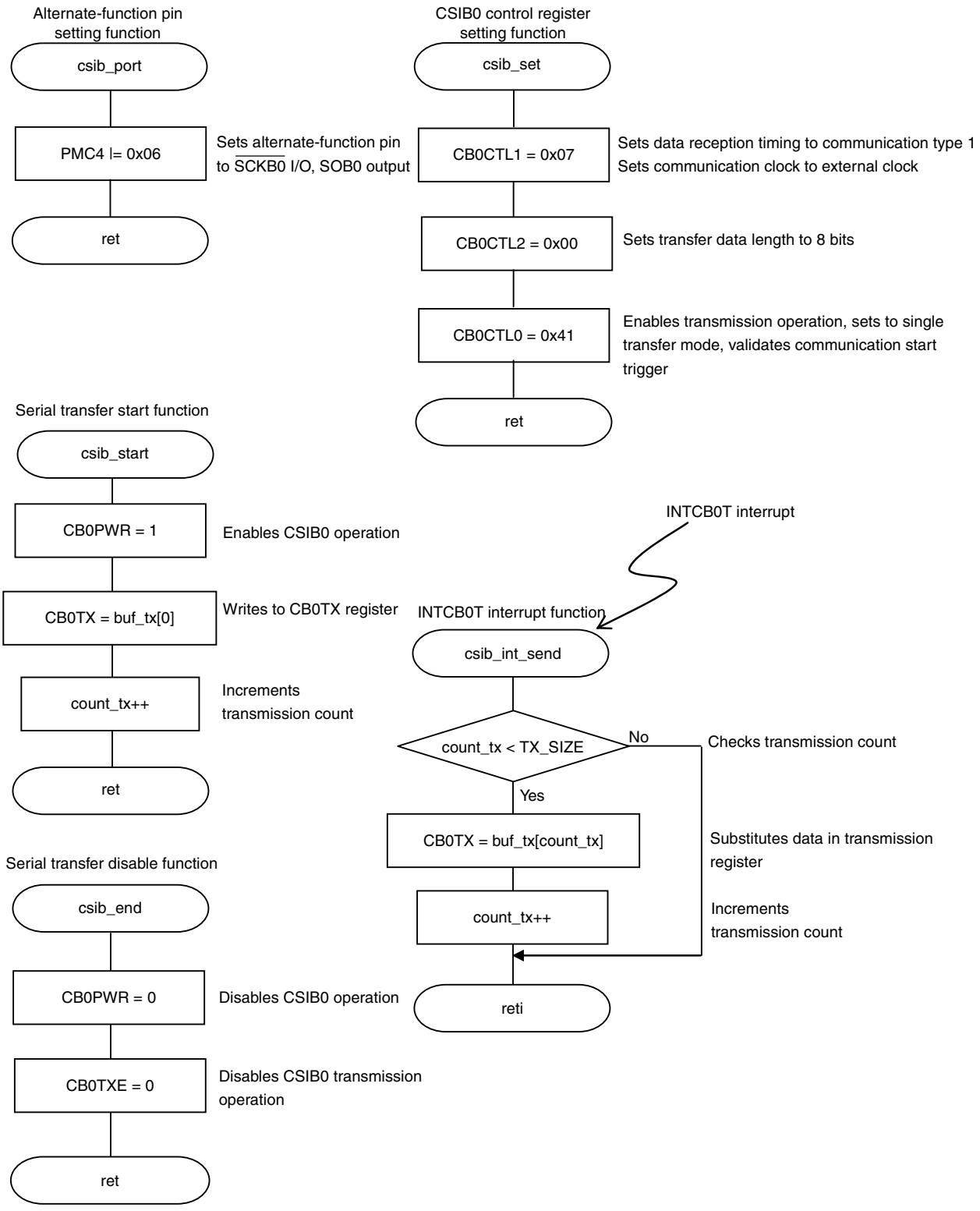
Interrupt function

[Function name]	csib_int_send	
[Processing content]	Sets new data for transmitting next data.	
[SFR used]	CB0TX	Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
[File name]	csib10.c	
[Caution]	None	

Single transfer mode (slave mode, transmission mode) (1/2)



Single transfer mode (slave mode, transmission mode) (2/2)



Clocked serial interface B (CSIBn) (n = 0, 1)

Single transfer mode (slave mode, reception mode)

[Function]	Sets communication mode to slave mode and transfer direction mode to MSB first, and performs reception for ten times in single transfer mode. Validates communication start trigger and sets communication clock to external clock, and transfer data length to 8 bits.
[Function name]	csib11_main
[Argument]	None
[Processing content]	Sets reception count (count_rx) to initial value 0. Starts reception after calling each setting function.
[SFRs used]	CB0RIC: 0x07 (Clears CSIB0 reception end interrupt request signal (INTCB0R), releases mask, sets to priority level 7.) CB0REIC: 0x07 (Clears CSIB0 reception error interrupt request signal (INTCB0RE), releases mask, sets to priority level 7.)
[call functions]	csib_port, csib_set, csib_start, csib_end
[Variables]	unsigned char buf_rx[ ]: Receive data storing buffer volatile unsigned char count_rx: Reception count variable
[Interrupts]	csib_int_receive, csib_error
[Interrupt sources]	INTCB0R, INTCB0RE
[File name]	csib11.c
[Caution]	None

[Function name]	csib_port
[Processing content]	Sets port 4 as CSIB0 I/O pin.
[SFR used]	PMC4: 0x05 (Sets $\overline{\text{SCKB0}}$ I/O and SIB0 input.)
[call function]	None
[Variable]	None
[File name]	csib11.c
[Caution]	None

[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x07 (Sets to communication type 1 and sets communication clock to external clock.) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x21 (Enables CSIB0 reception operation, sets to MSB first and single transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib11.c
[Caution]	The CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

[Function name]	csib_start	
[Processing content]	Enables CSIB0 operation and performs dummy read on receive data register.	
[SFRs used]	CB0CTL0.CB0PWR: CB0RX	1 (Enables CSIB0 operation.) Receive data register
[call function]	None	
[Variable]	unsigned char buf_rx[ ]:	Receive data storing buffer
[File name]	csib11.c	
[Caution]	None	

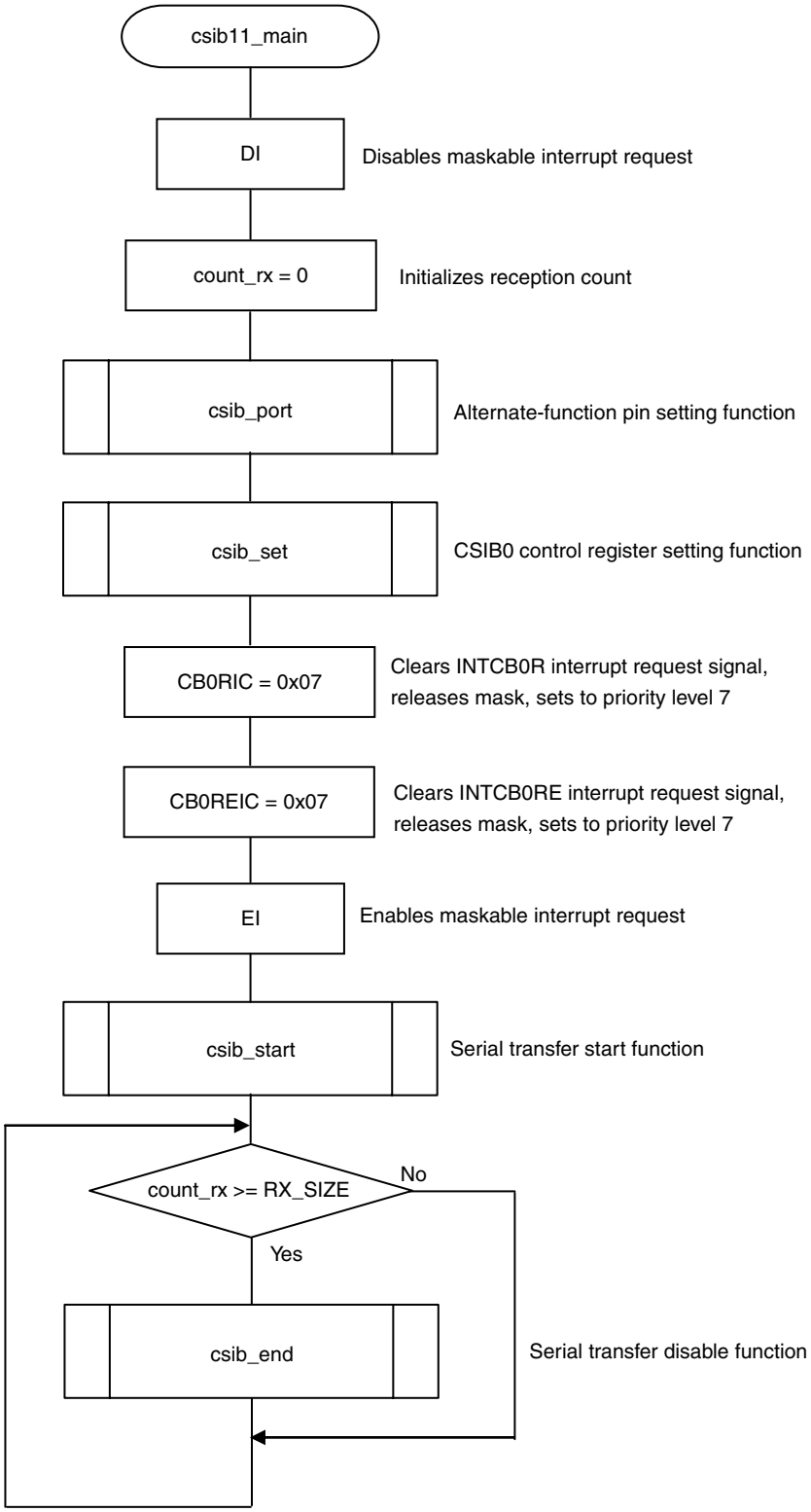
[Function name]	csib_end	
[Processing content]	Disables CSIB0 operation and reception operation.	
[SFRs used]	CB0CTL0.CB0PWR: CB0CTL0.CB0RXE:	0 (Disables CSIB0 operation.) 0 (Disables CSIB0 reception operation.)
[call function]	None	
[Variable]	None	
[File name]	csib11.c	
[Caution]	None	

Interrupt functions

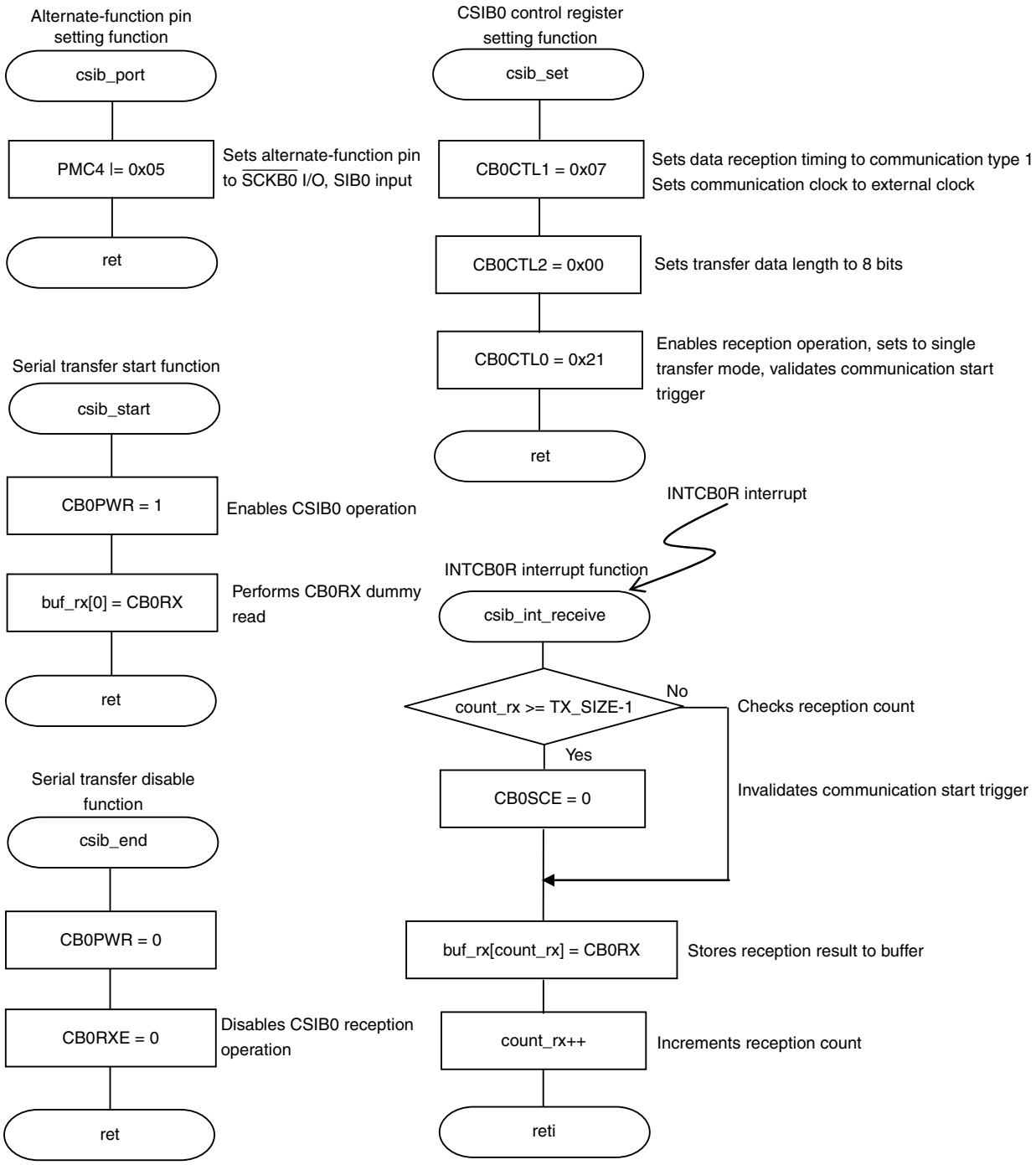
[Function name]	csib_int_receive	
[Processing content]	Stores receive data to buffer.	
[SFRs used]	CB0RX	Receive data register
	CB0CTL0.CB0SCE:	0 (Invalidates communication start trigger.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib11.c	
[Caution]	None	

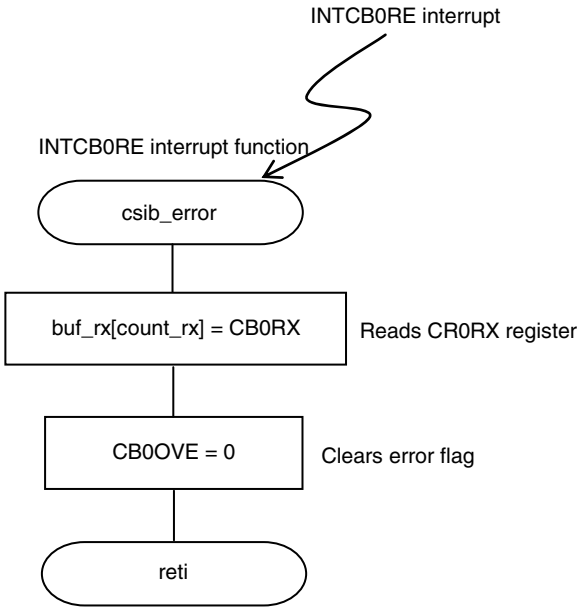
[Function name]	csib_error	
[Processing content]	Clears reception error flag.	
[SFRs used]	CB0RX	Receive data register
	CB0STR.CB0OVE:	0 (Clears overrun error flag.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib11.c	
[Caution]	None	

Single transfer mode (slave mode, reception mode) (1/3)



Single transfer mode (slave mode, reception mode) (2/3)




INTCB0RE interrupt
INTCB0RE interrupt function
csib_error
buf_rx[count_rx] = CB0RX
Reads CR0RX register
CB0OVE = 0
Clears error flag
reti

Clocked serial interface B (CSIBn) (n = 0, 1)

Single transfer mode (slave mode, transmission/reception mode)

[Function]	Sets communication mode to slave mode and transfer direction mode to MSB first, and performs data transmission/reception for ten times each in single transfer mode. Validates communication start trigger and sets communication clock to external clock, and transfer data length to 8 bits.
[Function name]	csib12_main
[Argument]	None
[Processing content]	Sets transmission count (count_tx) to initial value 0. Sets reception count (count_rx) to initial value 0 and starts transmission/reception after calling each setting function.
[SFRs used]	CB0REIC: 0x07 (Clears CSIB0 reception error interrupt request signal (INTCB0RE), releases mask, sets to priority level 7.) CB0RIC: 0x07 (Clears CSIB0 reception end interrupt request signal (INTCB0R), releases mask, sets to priority level 7.) CB0TIC: 0x07 (Clears CSIB0 transmission enable interrupt request signal (INTCB0T), releases mask, sets to priority level 7.)
[call functions]	csib_port, csib_set, csib_start, csib_end
[Variables]	unsigned char buf_tx[ ]: Transmit data storing buffer unsigned char buf_rx[ ]: Receive data storing buffer volatile unsigned char count_tx: Transmission count variable volatile unsigned char count_rx: Reception count variable unsigned char count: Transfer data generating variable
[Interrupts]	csib_error, csib_int_send, csib_int_receive
[Interrupt sources]	INTCB0RE, INTCB0T, INTCB0R
[File name]	csib12.c
[Caution]	None

[Function name]	csib_port
[Processing content]	Sets port 4 as CSIB0 I/O pin.
[SFR used]	PMC4: 0x07 (Sets $\overline{\text{SCKB0}}$ I/O, SOB0 output and SIB0 input.)
[call function]	None
[Variable]	None
[File name]	csib12.c
[Caution]	None

[Function name]	csib_set
[Processing content]	Sets CSIB0 control register.
[SFRs used]	CB0CTL1: 0x07 (Sets to communication type 1 and external clock.) CB0CTL2: 0x00 (Sets transfer data length to 8 bits.) CB0CTL0: 0x61 (Enables CSIB0 transmission and reception operation, sets to MSB first and single transfer mode, and validates communication start trigger.)
[call function]	None
[Variable]	None
[File name]	csib12.c
[Caution]	The CB0TXE and CB0RXE bits of the CB0CTL0 register are rewritable only when the CB0PWR bit is 0. However, the CB0PWR bit can be set to 1 simultaneously.

[Function name]	csib_start	
[Processing content]	Enables the CSIB0 operation and writes a value to the transmit data register.	
[SFRs used]	CB0CTL0.CB0PWR:	1 (Enables CSIB0 operation.)
	CB0TX	Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
[File name]	csib12.c	
[Caution]	None	

[Function name]	csib_end	
[Processing content]	Disables CSIB0 operation and transmission/reception operation.	
[SFRs used]	CB0CTL0.CB0PWR:	0 (Disables CSIB0 operation.)
	CB0CTL0.CB0RXE:	0 (Disables CSIB0 reception operation.)
	CB0CTL0.CB0TXE:	0 (Disables CSIB0 transmission operation.)
[call function]	None	
[Variable]	None	
[File name]	csib12.c	
[Caution]	None	

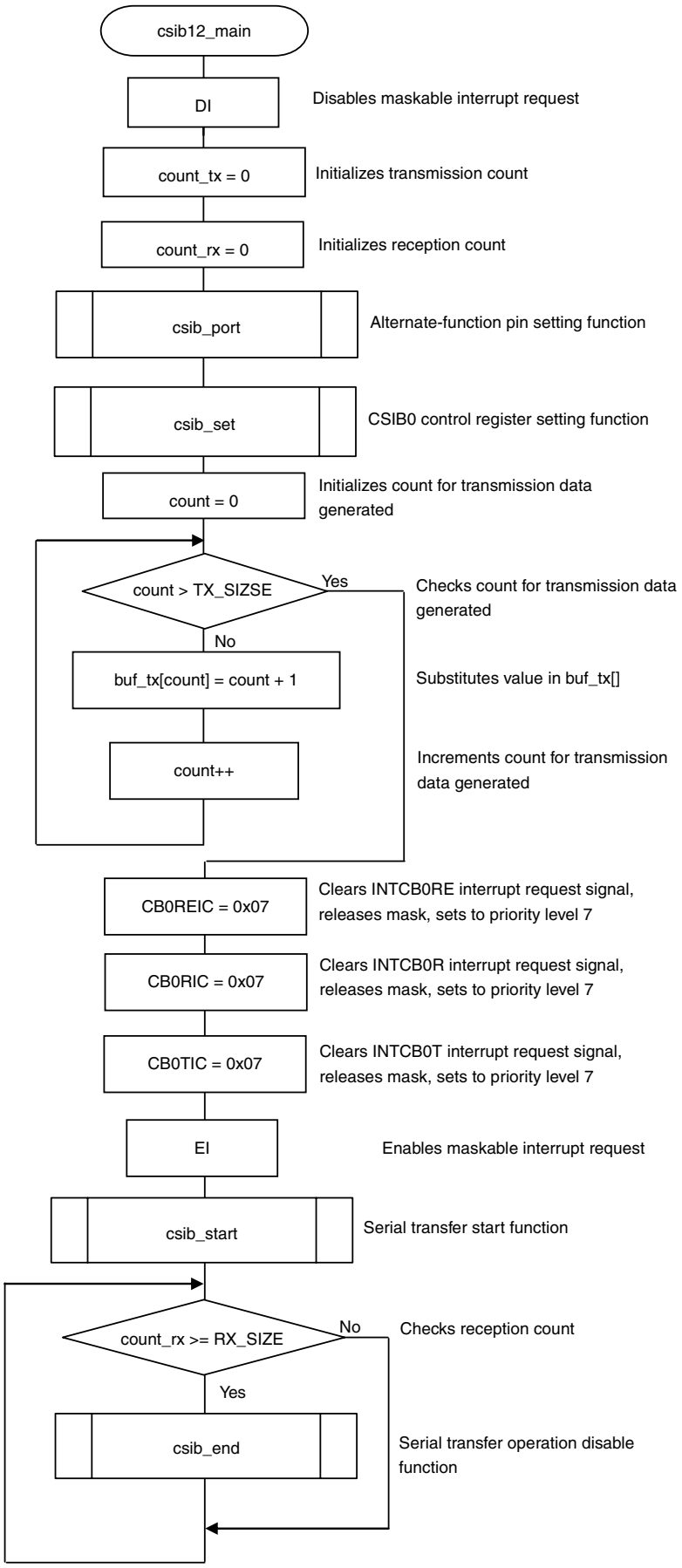
Interrupt functions

[Function name]	csib_error	
[Processing content]	Clears reception error flag.	
[SFRs used]	CB0RX	Receive data register
	CB0STR.CB0OVE:	0 (Clears overrun error flag.)
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib12.c	
[Caution]	None	

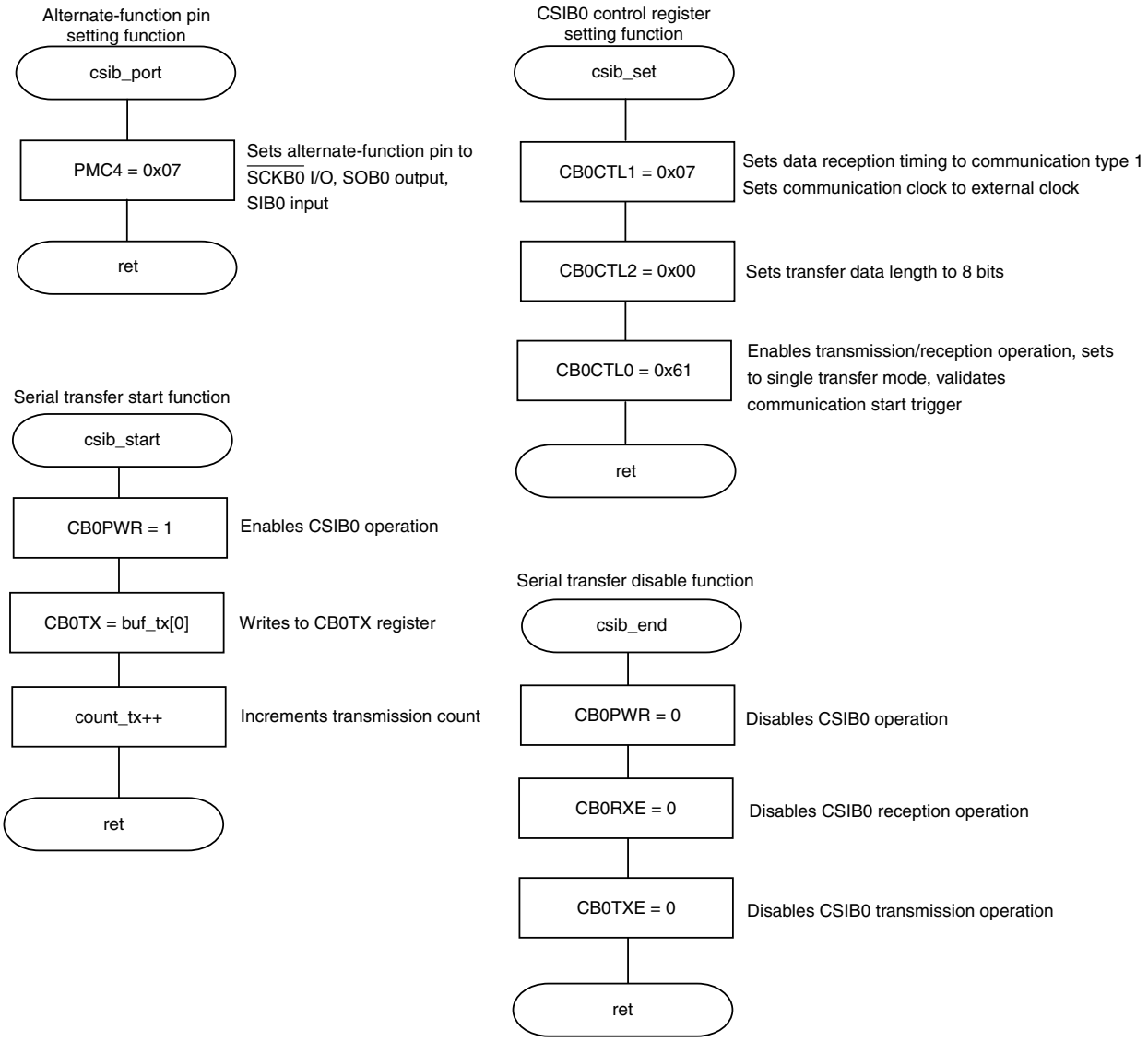
[Function name]	csib_int_send	
[Processing content]	Sets new data for transmitting next data.	
[SFR used]	CB0TX	Transmit data register
[call function]	None	
[Variables]	unsigned char buf_tx[ ]:	Transmit data storing buffer
	volatile unsigned char count_tx:	Transmission count variable
[File name]	csib12.c	
[Caution]	None	

[Function name]	csib_int_receive	
[Processing content]	Stores receive data to buffer.	
[SFR used]	CB0RX Receive data register	
[call function]	None	
[Variables]	unsigned char buf_rx[ ]:	Receive data storing buffer
	volatile unsigned char count_rx:	Reception count variable
[File name]	csib12.c	
[Caution]	None	

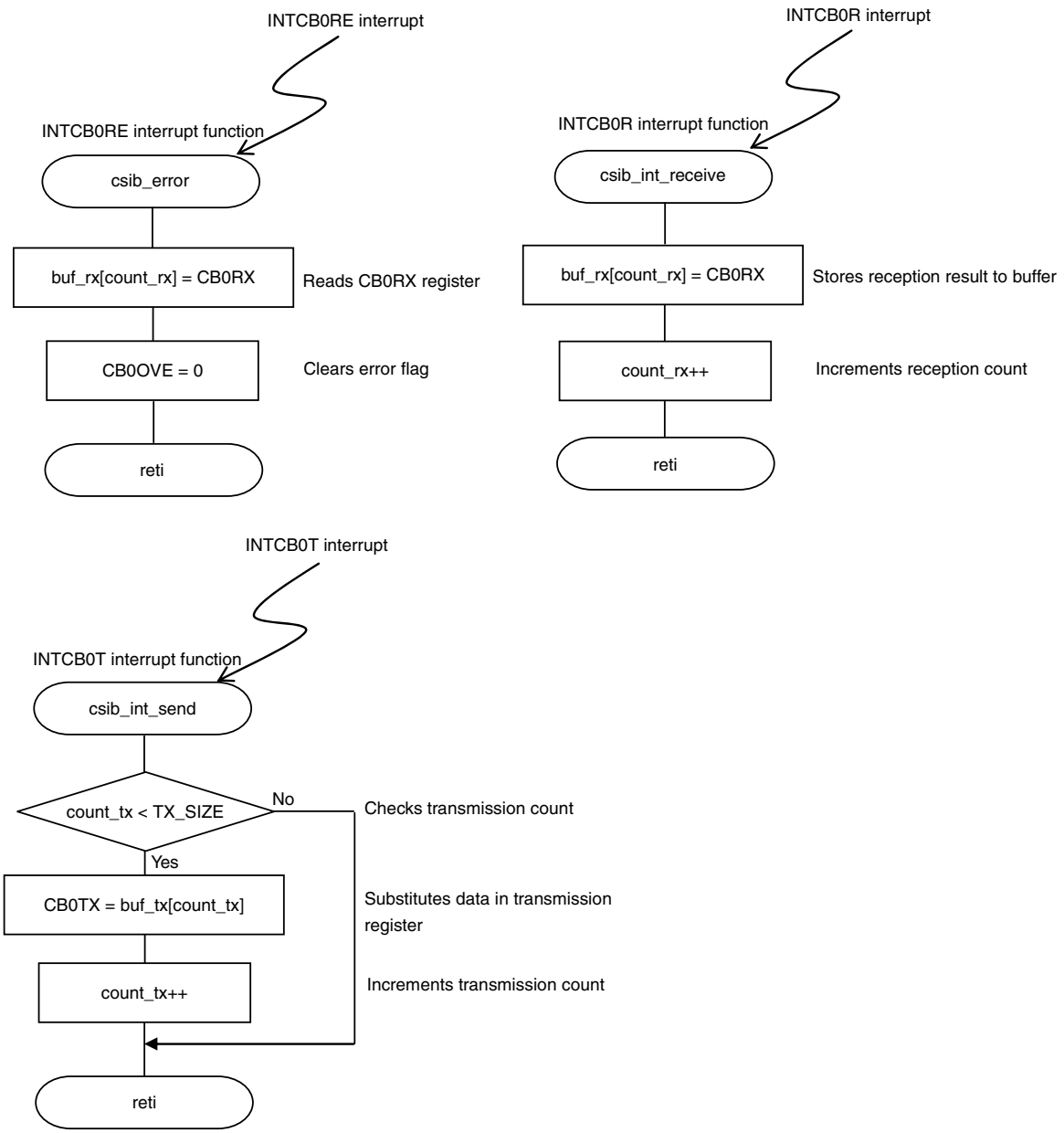
Single transfer mode (slave mode, transmission/reception mode) (1/3)



Single transfer mode (slave mode, transmission/reception mode) (2/3)



Single transfer mode (slave mode, transmission/reception mode) (3/3)



For further information,
please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

NEC Electronics Shanghai Ltd.
Room 2511-2512, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai P.R. China P.C:200120
Tel: 021-5888-5400
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
12/F., Cityplaza 4,
12 Taikoo Wan Road, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/

G06.11A