RL78/G13

Handshake-based SPI Master Transmission/Reception

Introduction

This application note describes how the serial array unit (SAU) performs master transmission/reception by the simple SPI (CSI). The SAU uses the chip select ($\overline{CS}$) signal to select a slave device and perform single transmission/reception. The SAU also performs handshake processing using the BUSY signal.

Target Device

RL78/G13

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Contents

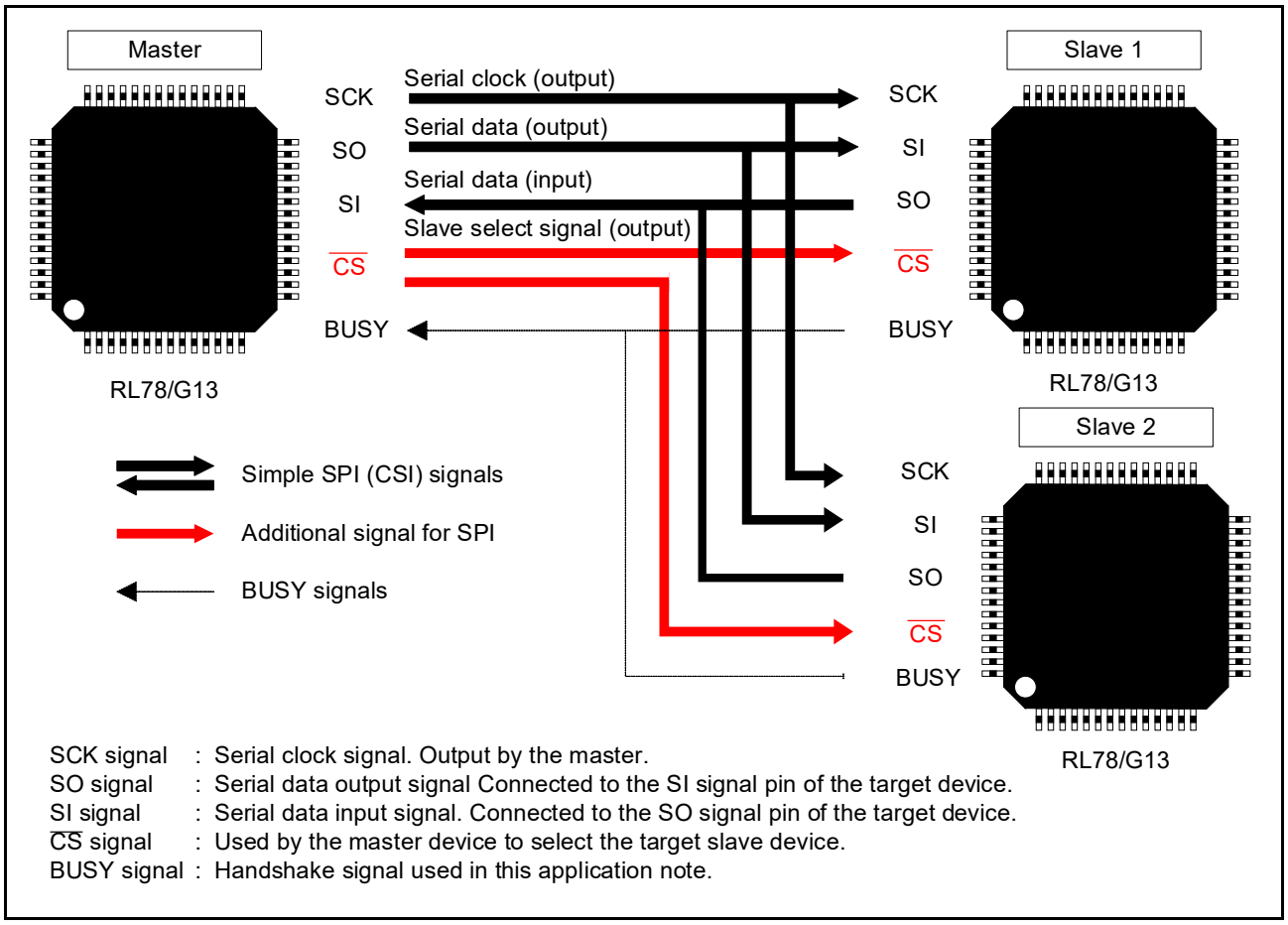
1. Specifications

The serial array unit (SAU) described in this application note performs CSI master transmission/reception. The chip select ($\overline{CS}$) signal uses the port.$\overline{CS}$ Handshake processing using the BUSY signal is also performed for the slave device selected by the $\overline{CS}$ signal.

1.1 Outline of CSI Communication

CSI communication is clock-synchronous serial communication using three signal lines, namely, serial clock (SCK), serial data input (SI), and serial data output (SO). SPI (Serial Peripheral Interface) uses an additional chip select ($\overline{CS}$) signal to select the slave device. The relationship among these signals is shown in Figure 1-1.

Figure 1-1 Outline of CSI Communication



The master first selects the slave with which it wants to communicate with the $\overline{CS}$ signal. Then, the master outputs data to the SCK signal line and the SO signal line in synchronization with the SCK signal, and inputs data from the SI signal line.

In SPI/CSI communication, the slave needs to become ready for communication by the time the master starts communication (sending the SCK signals). In this application note, the BUSY signal is used to confirm that the slave is ready for communication. The master detects a low-level BUSY signal and then initiates a communication session.

1.2 Outline of Communication

In this application note, a command and communication for the command are performed at intervals of 1 ms. A set of a command and communication for the command is defined as a slot. Figure 1-2 shows an outline of slot processing and Table 1-1 lists the commands to be used.

Figure 1-2 Outline of Slots

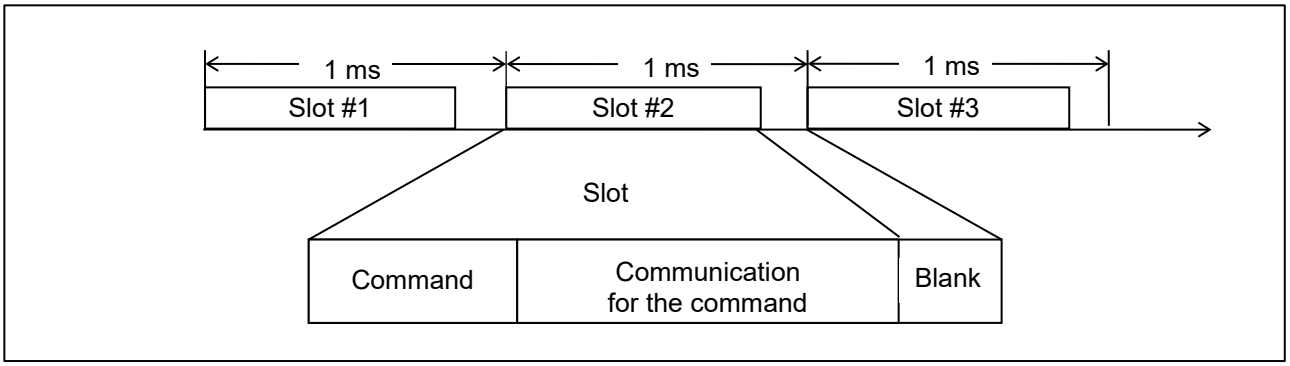


Table 1-1 Commands to be Used

Command	Outline of Operation
Status check	Checks the number of data characters that the slave can transmit or receive.
Receive	Receives data from the slave.
Transmit	Transmits data to the slave.
Transmit/receive	Transmits and receives data to and from the slave.

Table 1-2 lists the peripheral functions and their uses. Figure 1-3 and Figure 1-4 show the CSI communication operations.

Table 1-2 Peripheral Functions and Their Uses

Peripheral Function	Use
Serial array unit 0	Performs CSI master communication using the SCK00 signal (clock output), SI00 signal (receive data), and SO00 signal (transmit data).
Port	Uses P52 for $\overline{CS1}$ signal, P53 for $\overline{CS2}$ signal, and P54 for BUSY signal.
Timer array unit 0 Channel 3	The upper 8 bits are used as a 1-ms interval timer. The lower 8 bits are used as a 16-µs interval timer.

Figure 1-3 Timing chart for status check commands

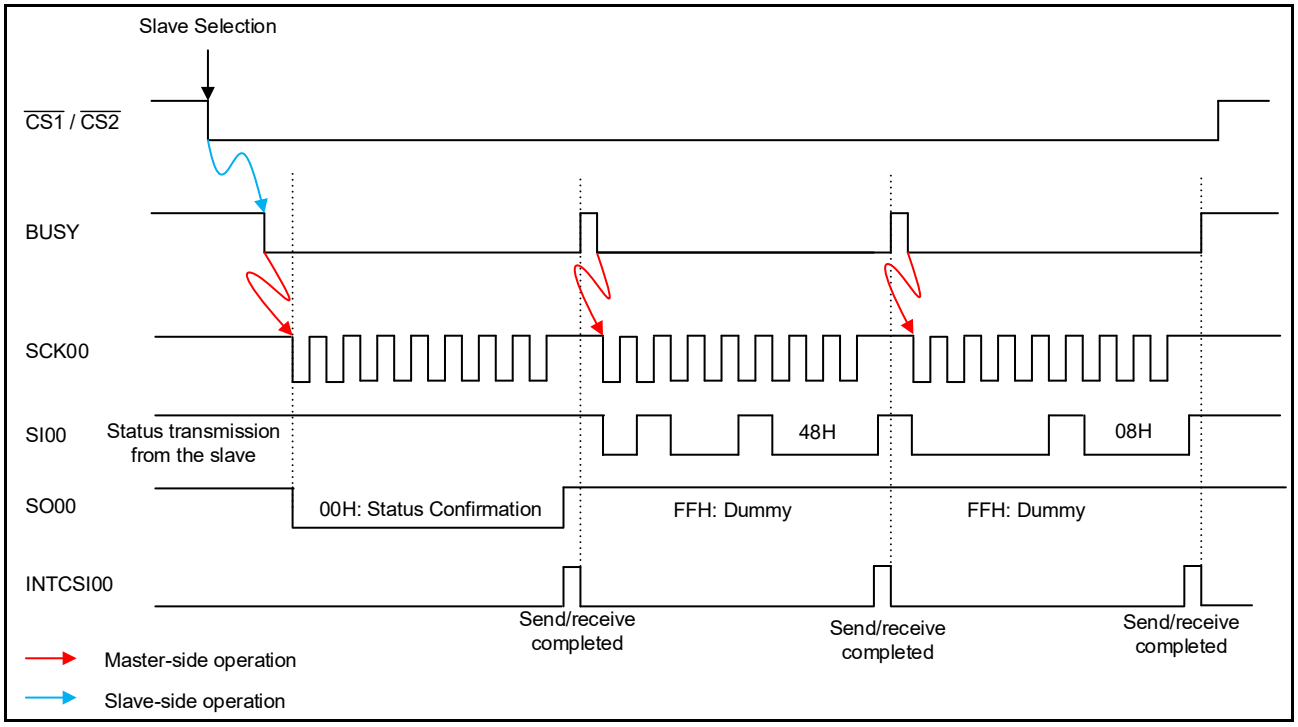
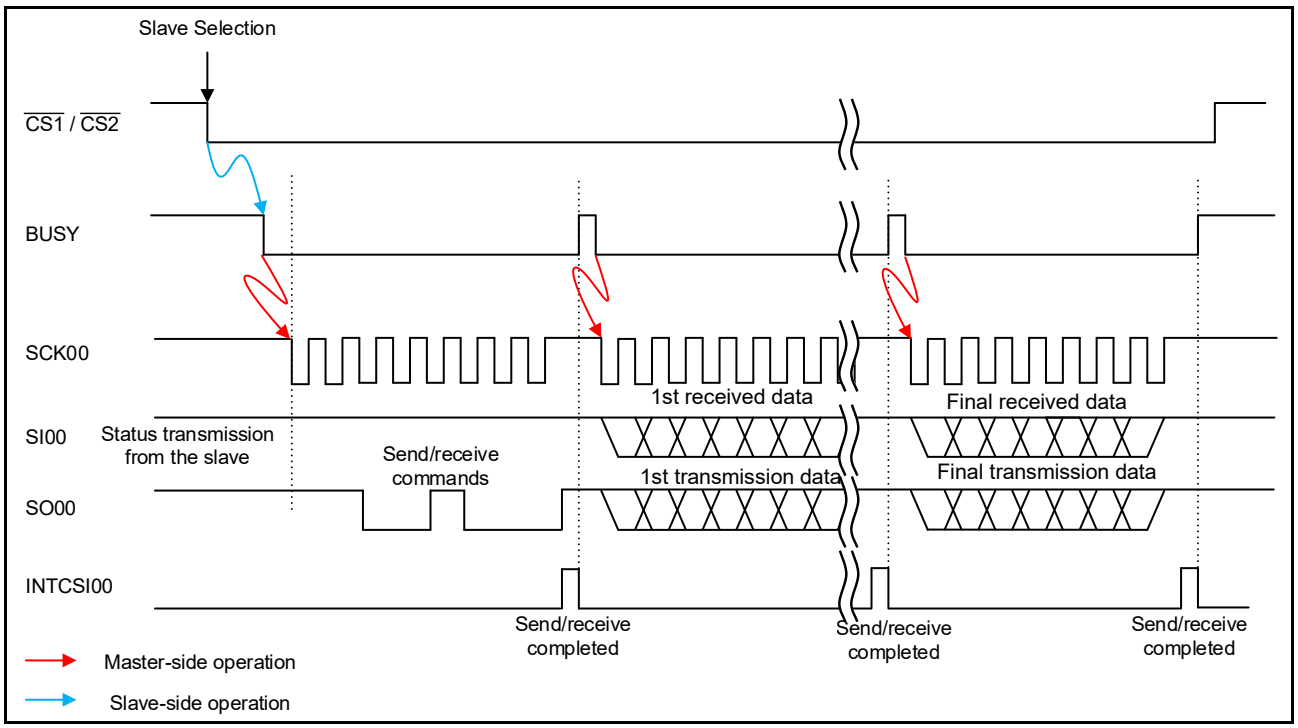


Figure 1-4 Timing chart of commands sent and received



1.3 Communication Format

Table 1-3 lists the characteristics of the CSI communication format that is used in the sample code.

Table 1-3 Communication Format

Item	Specification	Remarks
Communication speed	1 Mbps	About 200 kbps at minimum
Data bit length	8 bits/character	
Transfer order	MSB first	
Communication type	Type 1	
Communication mode	Single transfer	
Communication direction	Receive/transmit/transmit and receive	
Maximum number of characters transferred	63 characters/slot	32 characters by default

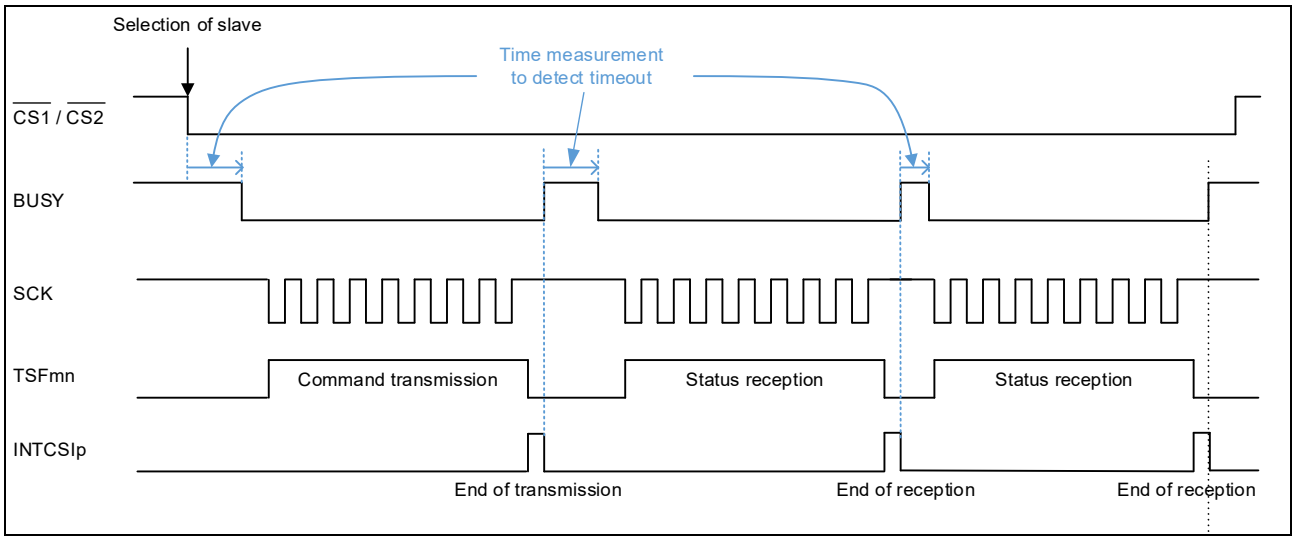
1.4 Handshake

In this application note, handshaking using the BUSY signal is performed to secure the setup time required for the communication operation on the slave side. A timeout of 16 µs is set up in case no response using the BUSY signal is returned from the slave. If no response is returned from the slave within this period, the master assumes that the slave cannot establish a communication (or does not exist) and terminates communication processing.

Figure 1-5 shows an example of handshaking for status checking. The master measures the time from the falling edge of the $\overline{CS}$ signal until the BUSY signal goes low. When detecting that the BUSY signal goes low without the timeout, the master sends the command. After confirming that command transmission is completed through the transfer completion interrupt, the master measures the time until the BUSY signal goes low again. In this way, the master synchronizes with the slave by checking the BUSY signal each time communication starts.

However, if the slave supports continuous reception or can secure enough time to prepare communication data, synchronization using the BUSY signal is not necessary.

Figure 1-5 Handshaking Example



Generally, the BUSY signal is not required for SPI communication-dedicated slave devices such as EEPROM because they are always ready for communication. When connecting such SPI communication-dedicated slave devices, pull down the BUSY signal input pin. Additionally, the timeout processing can be removed by deleting the wait() function from the program. In such a case, perform communication according to the commands that are defined for the SPI communication-dedicated slave devices and their communication protocol.

1.5 Specification Details

After completion of initialization, this sample code selects a slave, checks that slave's status, and then transmits and receives data, switches slaves, and repeats the same operation.

(1) Initialize the port.

<Conditions for setting the port>

- Use P52 controlling the $\overline{CS1}$ signal as an output port to output a high-level signal.
- Use P53 controlling the $\overline{CS2}$ signal as an output port to output a high-level signal.
- Use P54 detecting the BUSY signal as an input port.

(2) Initialize the timer.

<Conditions for setting the timer>

- Run Channel 3 as two 8-bit interval timers.
- Set the operating clock frequency to 125 kHz (derived by dividing fCLK by 256).
- Use the upper TM03H as a 1-ms interval timer.
- Use the lower TM03 as a 16-µs interval timer.
- Set the priority of the interrupts (INTTM03H, INTTM03) to the lowest level (3, by default).

(3) Initialize the CSI.

<Conditions for setting the CSI>

- Use SAU0 channel 0 as CSI00.
- Use CK00 as the transfer clock.
- Assign the clock output to the P10/SCK00 pin, the data input to the P11/SI00 pin, and the data output to the P12/SO00 pin.
- Use single transfer mode as the transfer mode.
- Set the data length to 8 bits.
- Set the phase between the data and clock to type 1.
- Set the order of data transfer mode to MSB first.
- Set the transfer rate to 1 Mbps.
- Use transmission end and reception end interrupts as the interrupt (INTCSI00).
- Set the priority of the interrupt (INTCSI00) to the lowest level (3, by default).

(4) After initialization is completed, the master performs communication with the slave as shown in the following steps.

① The master waits in HALT mode for a 1-ms interval timer interrupt (INTTM03H).

② When the master is released from HALT mode by an INTTM03 interrupt, it issues the $\overline{CS}$ signal to select the slave that is specified in the g_slave_select_flag flag, and then waits for a response from the slave.

③ When the BUSY signal goes low, the master proceeds to step 4. When a timeout is detected, the master deselects the slave and proceeds to step 9.

④ The master transmits a status check command and receives the slave status. When a timeout is detected, or the slave status cannot be received, the master deselects the slave and proceeds to step 9. When the slave status is received, the master deselects the $\overline{CS}$ signal.

⑤ The master again waits in HALT mode for a 1-ms interval timer interrupt (INTTM03H).

⑥ When the master is released from HALT mode by an INTTM03 interrupt again, it issues the $\overline{CS}$ signal to select the slave that is specified in the g_slave_select_flag flag, and then waits for a response from the slave.

⑦ When the master detects that the BUSY signal goes low, it proceeds to step 8. When a timeout is detected, the master deselects the slave and proceeds to step 9.

⑧ The master transmits and receives the number of data characters according to the status checked in step 4. When a timeout is detected, the master deselects the slave.

⑨ The master changes the g_slave_select_flag flag to switch the target slave. These steps are subsequently repeated from step 1.

(5) Commands

Each communication operation begins with the transmission of a 1-byte command. Table 1-4 lists the command formats. The master transmits a status check command and receives the response from the slave in the first slot of a communication sequence. The master Confirm that the number of data characters that the target slave can transmit or receive is equal to or greater than the specified number of data characters to be transmitted or received. Then, the master transmits/receives data for the specified number of data characters to/from the slave.

Table 1-4 Command Formats

Command Code		Command Outline
Status check	00000000B	Checks the number of data characters that the slave can transmit or receive. The following responses can be made by the slave: 01xxxxxxB: The number of characters that the slave can transmit is xxxxxxB. 00xxxxxxB: The number of characters that the slave can receive is xxxxxxB.
Reception	01xxxxxxB	The master receives xxxxxxB bytes of data.
Transmission	10xxxxxxB	The master transmits xxxxxxB bytes of data.
Transmission/reception	11xxxxxxB	The master transmits and receives xxxxxxB bytes of data.

(6) Switching communication commands

Receive, transmit, and transmit/receive commands can be switched by changing the comment-out lines in the r_main.c file. The transmit/receive command is set by default.

```
/*-------------------------------------------------------
    Send Receive command
-------------------------------------------------------*/

            CSI00_Send_Receive();                   /*call Send Receive function*/
            /* CSI00_Send(); */
            /* CSI00_Receive(); */
            g_slave_select_flag ^= 1U;              /*change slave number*/
            g_status_confirmation_flag = 0;         /* clear g_status_confirmation_flag */

```

2. Operation Confirmation Conditions

The operation of the sample code provided with this application note has been tested under the following conditions.

Figure 2-1 Operation Confirmation Conditions

Item	Description
MCU used	RL78/G13 (R5F100LE)
Operating frequency	● High-speed on-chip oscillator clock (fIH): 32 MHz ● CPU/peripheral hardware clock: 32 MHz
Operating voltage	During VDD operation: 5.0 V（2.7V～5.5V） LVD detection voltage: Reset mode. At rising edge TYP. 1.88V (1.84 V ～ 1.91 V) At falling edge TYP. 1.84V (1.80 V ～ 1.87 V)
Integrated development environment (CS+)	CS+ V8.09.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.12.00 from Renesas Electronics Corp
Integrated development environment (e2studio)	e2 studio Version: 2023-01 (23.1.0) from Renesas Electronics Corp.
C compiler (e2studio)	CC-RL V1.12.00 from Renesas Electronics Corp.
Integrated development environment (IAR)	IAR Embedded Workbench for Renesas RL78 V4.21.2 from IAR Systems Corp.
C compiler (IAR)	IAR C/C++ Compiler for Renesas RL78 V4.21.2.2420 from IAR Systems Corp.
Board used	RL78/G13 (R5F100LE) Target Board（QB-R5F100LE-TB）

3. Related Application Notes

See also the following application notes, which are related to this application note:

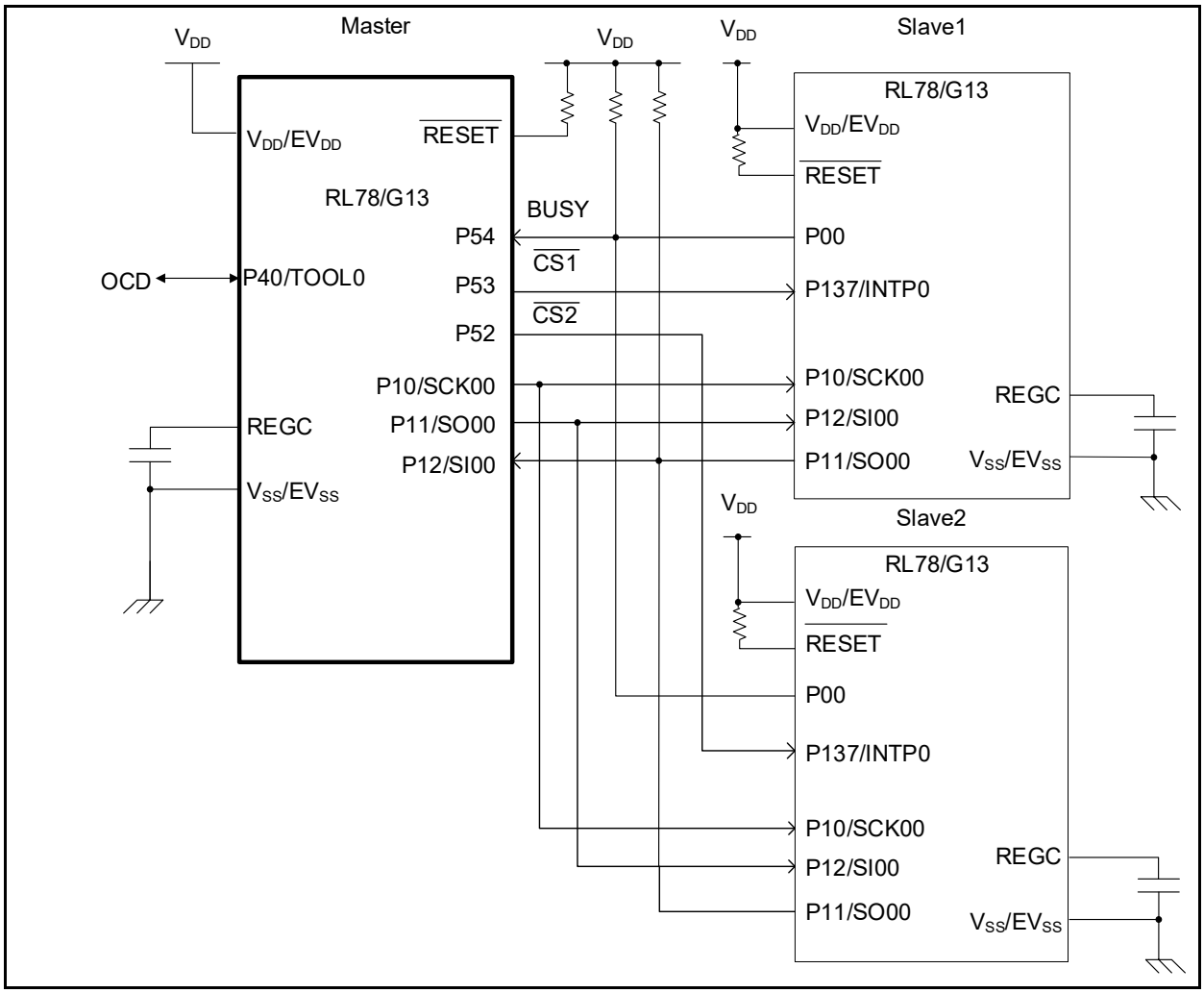
RL78/G13 Handshake-based SPI Slave Transmission/Reception (R01AN6884E) APPLICATION NOTE

4. Hardware Descriptions

4.1 Example of Hardware Configuration

Figure 4-1shows an example of the hardware configuration used in the application note.

Figure 4-1 Hardware Configuration



Note 1. This schematic circuit diagram is simplified to show the outline of connections. When creating circuits, design them so that they meet electrical characteristics by properly performing pin processing. (Connect input-only ports to V_{DD} or V_{SS} individually through a resistor.)

Note 2. Connect pins (with a name beginning with EV_{SS}), if any, to V_{SS}, and connect pins (with a name beginning with EV_{DD}), if any, to V_{DD}.

4.2 List of Pins to be Used

Table 4-1 lists the pins to be used and their functions

Table 4-1 Pins to be Used and Their Functions

Pin Name	I/O	Description
P10/SCK00/SCL00/(TI07)/(TO07)	Output	Serial clock output pin
P11/SI00/RxD0/ TOOLRxD/SDA00/ (TI06)/(TO06)	Input	Data reception pin
P12/SO00/TxD0/ TOOLTxD/(INTP5)/(TI05)/(TO05)	Output	Data transmission pin
P54	Input	BUSY signal input from slaves
P53/(INTP11)	Output	Slave 2 select signal
P52/(INTP10)	Output	Slave 1 select signal

Caution In this application note, only the used pins are processed. When actually designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.

5. Description of the Software

5.1 List of Option Byte Settings

Table 5-1 summarizes the settings of the option bytes.

Table 5-1 Option Byte Settings

Address	Setting Value	Description
000C0H	1110 1111B (EFH)	Stops the watchdog timer operation. (Stops counting after the release of the reset state.)
000C1H	0011 1111B (3FH)	LVD reset mode Detection Voltage: On the rising edge: TYP. 1.88 V On the falling edge: TYP. 1.84 V
000C2H	11101000B (E8H)	HS mode, HOCO: 32 MHz
000C3H	10000101B (85H)	Enables the on-chip debugging function.

5.2 List of Constants

Table 5-2 lists the constants that are used in the sample code.

Table 5-2 Constants Used in the Sample Code

Constant Name	Definition location	Setting Value	Description
CS1_pin	r_cg_userdefine.h	P5_bit.no2	Port register to control the $\overline{CS1}$ signal
CS2_pin	r_cg_userdefine.h	P5_bit.no3	Port register to control the $\overline{CS2}$ signal
BUSYIN	r_cg_userdefine.h	P5_bit.no4	Port register to detect the BUSY signal
TX_NUM	r_main.c	32	Number of data characters to be transmitted
RX_NUM	r_main.c	32	Number of data characters to be received
TX_RX_NUM	r_main.c	32	Number of data characters to be transmitted/received
data_length	r_main.c	1	Data length
MODE[]	r_main.c	*1	Command format
TX_DATA[]	r_main.c	*2	Stores 63 characters of transmit data, the maximum number of characters transferred.

Notes: 1. For details, see Table 1-4.
2. In this application note, ASCII codes from 0x20 to 0x5F are stored.

5.3 List of Variables

Table 5-3 lists the global variables that are used in this sample code.

Table 5-3 Global Variables Used in the Sample Code

Type	Variable Name	Contents	Function Used
uint8_t	g_tx_data	Buffer for transmit data	r_main.c
uint8_t	g_rx_data	Buffer for receive data	r_main.c
uint8_t	g_slave_select_flag	Slave select flag	r_main.c
uint8_t	g_status_confirmation_flag	Status check flag	r_main.c
uint8_t	g_timeout_flag	Timeout flag	r_main.c
uint8_t	g_num	Number of data characters that the slave can transmit	r_main.c
uint8_t	g_rx_data_stored[]	Stores receive data.	r_main.c

5.4 List of Functions

Table 5-4 lists the functions that are used in the sample code.

Table 5-4 Functions

Function Name	Outline	Source file
R_Systeminit	Initialization process	r_systeminit.c
R_CGC_Get_ResetSource	Reset source reading	r_cg_cgc_user.c
R_CGC_Create	CPU clock Configuration	r_cg_cgc.c
R_PORT_Create	Port Configuration	r_cg_port.c
R_SAU0_Create	SAU Configuration	r_cg_serial.c
R_CSI00_Create	CSI00 Configuration	r_cg_serial.c
R_TAU0_Create	TAU0 Configuration	r_cg_timer.c
main	Main processing	r_main.c
R_MAIN_UserInit	User Main Initialization	r_main.c
R_TAU0_Channel3_Higher8bits_Start	TAU0 channel 3 upper 8 bits start operating	r_cg_timer.c
R_CSI00_Start	CSI00 start operating	r_cg_serial.c
CSI00_Status_check	CSI status check	r_main.c
CSI00_Send_Receive	CSI transmission/reception	r_main.c
CSI00_Send	CSI transmission	r_main.c
CSI00_Receive	CSI reception	r_main.c
R_CSI00_Send_Receive	CSI00 Transmit/Receive Processing	r_cg_serial.c
r_csi00_interrupt	INICSI00 interrupt processing	r_cg_serial_user.c
wait	Wait for slave response	r_main.c

5.5 Function Specifications

This section describes the specifications for the functions that are used in the sample code.

[Function Name] R_Systeminit

Synopsis	Initialization process
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_timer.h r_cg_userdefine.h
Declaration	void R_Systeminit (void)
Explanation	Initialize each peripheral function.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] R_CGC_Get_ResetSource

Synopsis	Reset source reading
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_userdefine.h
Declaration	void R_CGC_Get_ResetSource(void)
Explanation	Reads the reset source.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] R_CGC_Create

Synopsis	CPU clock Configuration
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_userdefine.h
Declaration	void R_CGC_Create (void)
Explanation	Set the CPU clock.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] R_PORT_Create

Synopsis	Port Configuration
Header	r_cg_macrodriver.h r_cg_port.h r_cg_userdefine.h
Declaration	void R_CGC_Create (void)
Explanation	Configure the port settings.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] R_SAU0_Create

Synopsis	SAU Configuration
Header	r_cg_macrodriver.h r_cg_serial.h r_cg_userdefine.h
Declaration	void R_SAU_Create (void)
Explanation	Configure SAU settings.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] R_CSI00_Create

Synopsis	CSI00 Configuration
Header	r_cg_macrodriver.h r_cg_serial.h r_cg_userdefine.h
Declaration	void R_CSI00_Create (void)
Explanation	Configure CSI00 settings.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] R_TAU0_Create

Synopsis	TAU0 Configuration
Header	r_cg_macrodriver.h r_cg_timer.h r_cg_userdefine.h
Declaration	void R_CSI00_Create (void)
Explanation	Configure TAU0 settings.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] main

Synopsis	Main processing
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_timer.h r_cg_userdefine.h
Declaration	void main(void)
Explanation	Starts the operation of CSI00 and TAU03H. Selects a slave. Switches the slave select flag, g_slave_select_flag. When slave 1 is selected: g_slave_select_flag = 0 When slave 2 is selected: g_slave_select_flag = 1 Sets the status check flag, g_status_confirmation_flag, to 0 (initial value) after detection of a timeout or completion of transmission/reception.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] R_MAIN_UserInit

Synopsis	User Main Initialization
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_timer.h r_cg_userdefine.h
Declaration	void R_MAIN_UserInit(void)
Explanation	Main user initial settings.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] R_TAU0_Channel3_Higher8bits_Start

Synopsis	TAU0 channel 3 upper 8 bits start operating
Header	r_cg_macrodriver.h r_cg_timer.h r_cg_userdefine.h
Declaration	void R_TAU0_Channel3_Higher8bits_Start (void)
Explanation	Starts operation of the upper 8 bits of TAU0 channel 3.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] R_CSI00_Start

Synopsis	CSI00 start operating
Header	r_cg_macrodriver.h r_cg_serial.h r_cg_userdefine.h
Declaration	void R_CSI00_Start (void)
Explanation	Starts the operation of CSI00.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] CSI00_Status_check

Synopsis	CSI status check
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_timer.h r_cg_userdefine.h
Declaration	void CSI00_Status_check (void)
Explanation	Checks the status of the slave. Sets the status check flag, g_status_confirmation_flag, to 1 when the slave status check ends normally.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] CSI00_Send_Receive

Synopsis	CSI transmission/reception
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_timer.h r_cg_userdefine.h
Declaration	void CSI00_Send_Receive (void)
Explanation	Performs the master transmission/reception processing.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] CSI00_Send

Synopsis	CSI transmission
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_timer.h r_cg_userdefine.h
Declaration	void CSI00_Send (void)
Explanation	Performs the master transmission processing.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] CSI00_Receive

Synopsis	CSI reception
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_timer.h r_cg_userdefine.h
Declaration	void CSI00_Receive (void)
Explanation	Performs the master reception processing.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] R_CSI00_Send_Receive

Synopsis	CSI00 Transmit/Receive Processing
Header	r_cg_macrodriver.h、r_cg_serial.h、r_cg_userdefine.h
Declaration	MD_STATUS R_CSI00_Send_Receive(uint8_t * const tx_buf、uint16_t tx_num、uint8_t * const rx_buf)
Explanation	Configure CSI00 data sending/receiving settings.
Arguments	• uint8_t * const tx_buf : [Address of sent data buffer] • uint16_t tx_num : [Sent data buffer size] • uint8_t * const rx_buf : [Address of received data buffer]
Return value	• [MD_OK]：Sending/receiving settings complete • [MD_ARGERROR]：Sending/receiving settings failure
Remarks	None

[Function Name] r_csi00_interrupt

Synopsis	INICSI00 interrupt processing
Header	r_cg_macrodriver.h、r_cg_serial.h、r_cg_userdefine.h
Declaration	static void __near r_csi00_interrupt(void)
Explanation	If there is unsent data, reads the received data and starts sending the unsent data. If there is no unsent data, reads the received data.
Arguments	• None
Return value	• None
Remarks	None

[Function Name] wait

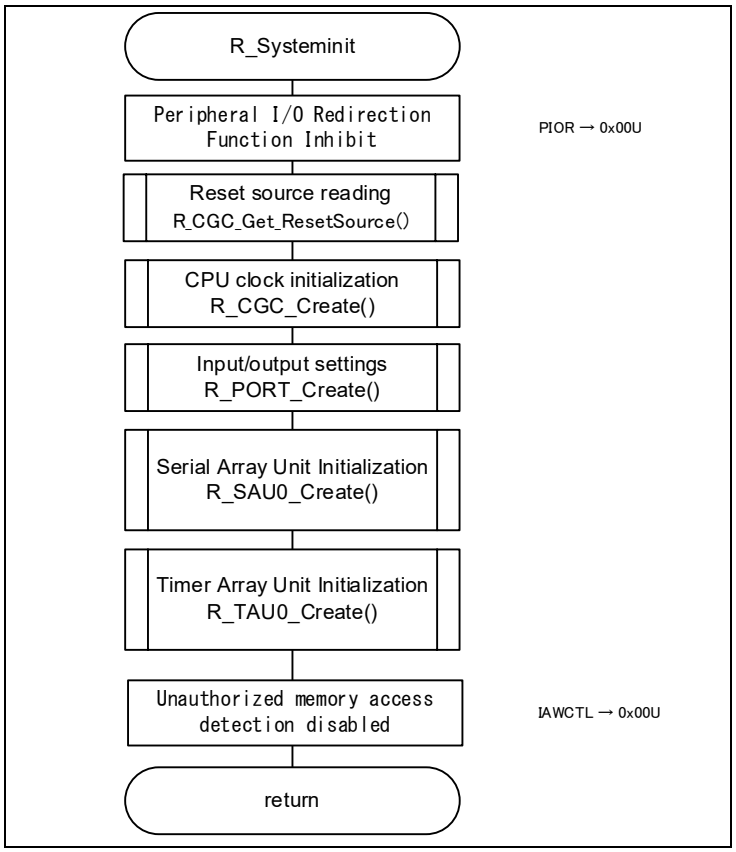
Synopsis	Wait for slave response
Header	r_cg_macrodriver.h r_cg_cgc.h r_cg_port.h r_cg_serial.h r_cg_timer.h r_cg_userdefine.h
Declaration	uint8_t wait(void)
Explanation	Waits until the BUSY signal goes low.
Arguments	• None
Return value	0: Detects that the BUSY signal goes low. 1: Detects a timeout.
Remarks	None

5.6 Flowcharts

5.6.1 Initialization process

Figure 5-1 shows the flow of Initialization process.

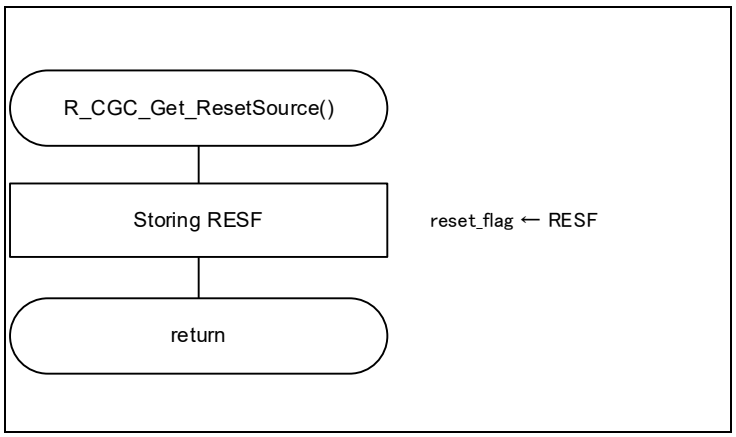
Figure 5-1 Initialization process



5.6.2 Reset source reading

Figure 5-2 shows the flow of Reset source reading

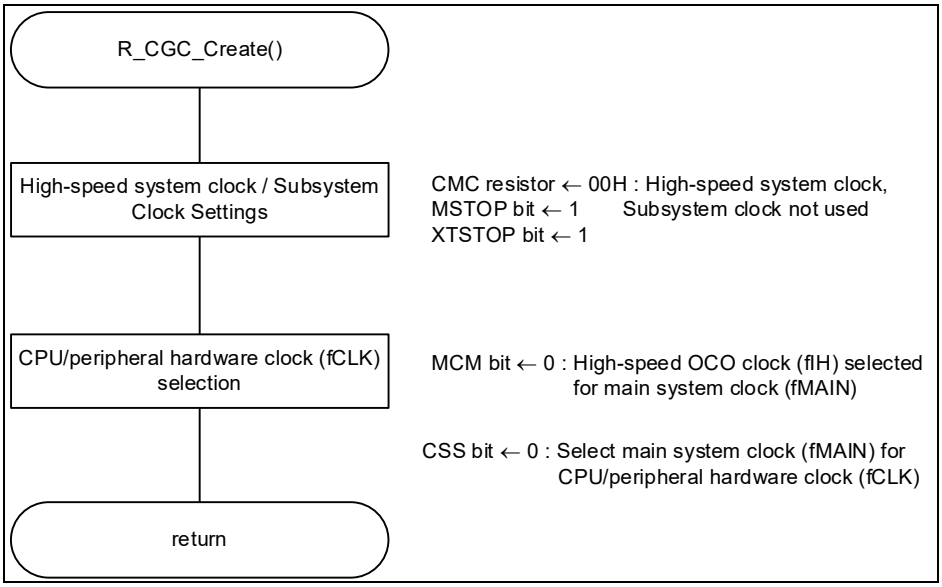
Figure 5-2 Reset source reading



5.6.3 CPU clock Configuration

Figure 5-3 shows the flow of CPU clock Configuration.

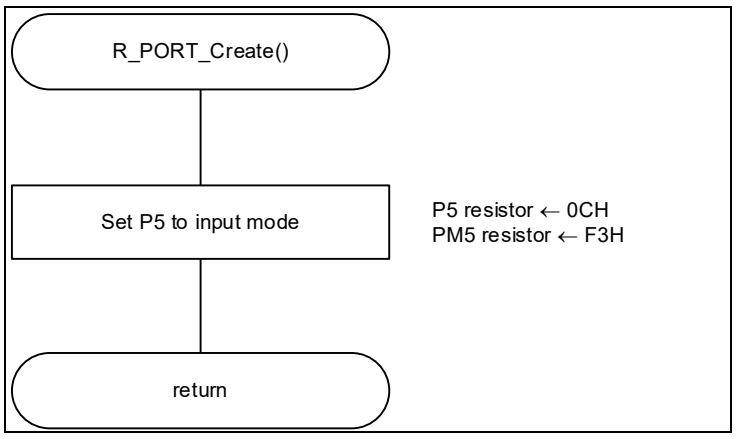
Figure 5-3 CPU clock Configuration



5.6.4 Port Configuration

Figure 5-4 shows the flow of Port Configuration.

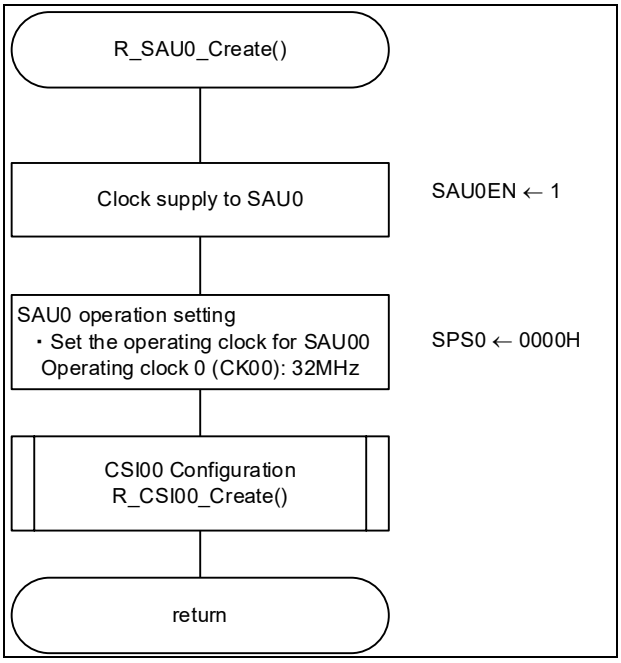
Figure 5-4 Port Configuration



5.6.5 SAU Configuration

Figure 5-5 shows the flow of SAU Configuration.

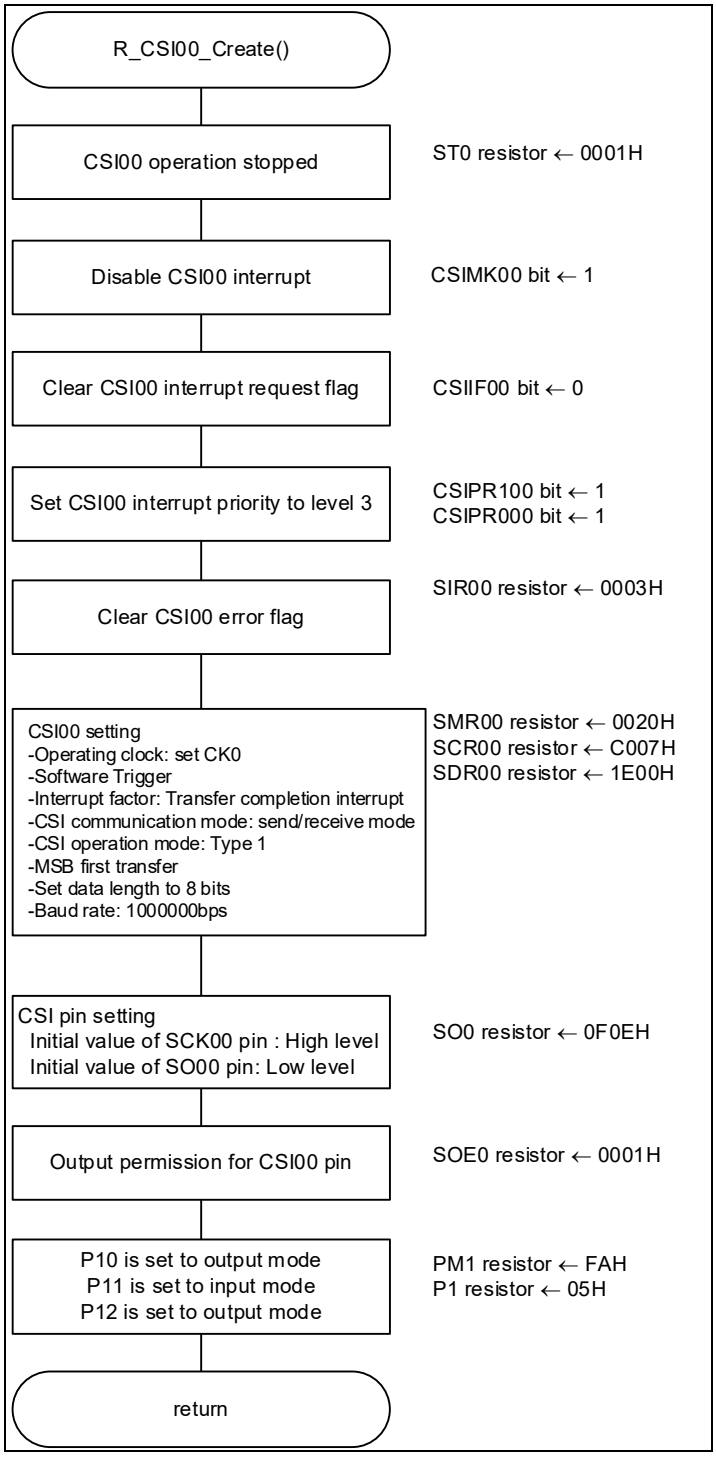
Figure 5-5 SAU Configuration



5.6.6 CSI00 Configuration

Figure 5-6 shows the flow of CSI00 Configuration.

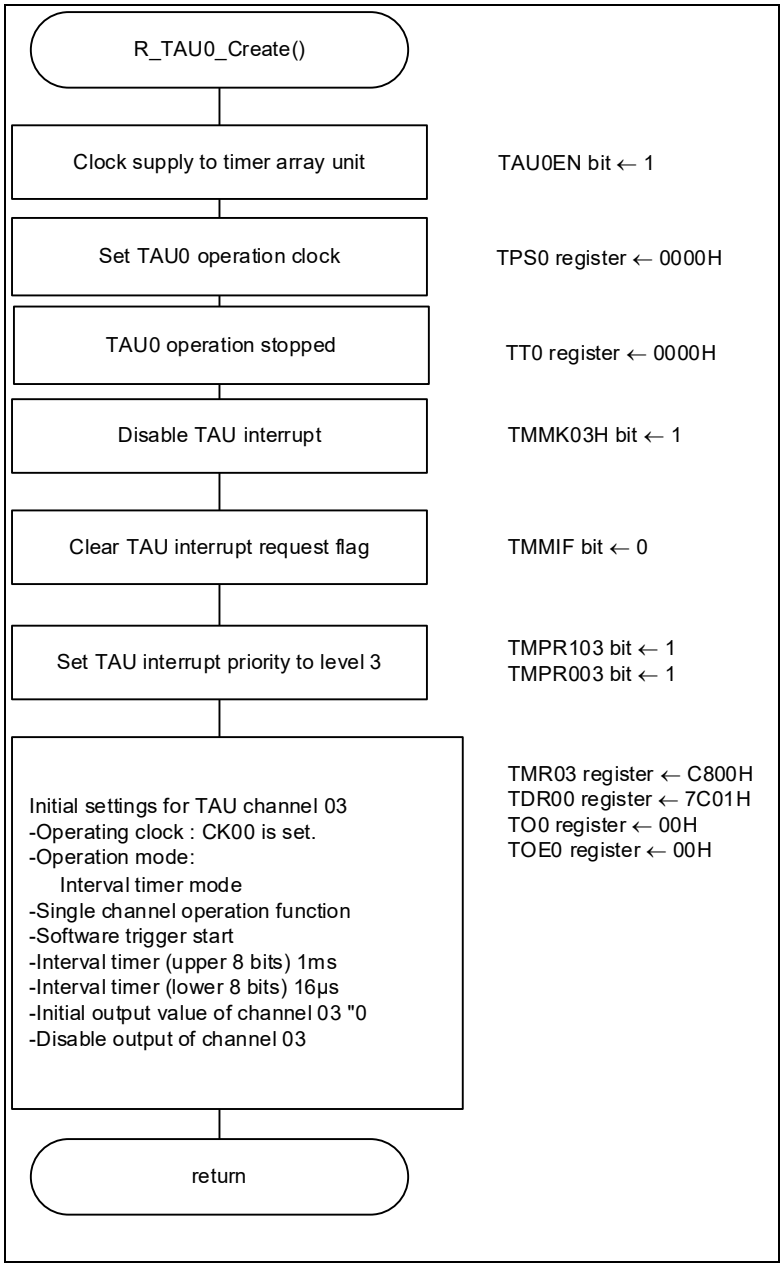
Figure 5-6 CSI00 Configuration



5.6.7 TAU0 Configuration

Figure 5-7 shows the flow of TAU0 Configuration.

Figure 5-7 TAU0 Configuration



5.6.8 Flowchart of Main Processing

Figure 5-8 to Figure 5-9 show the overall flow of processing in this application note.

Figure 5-8 Main Processing 1/2

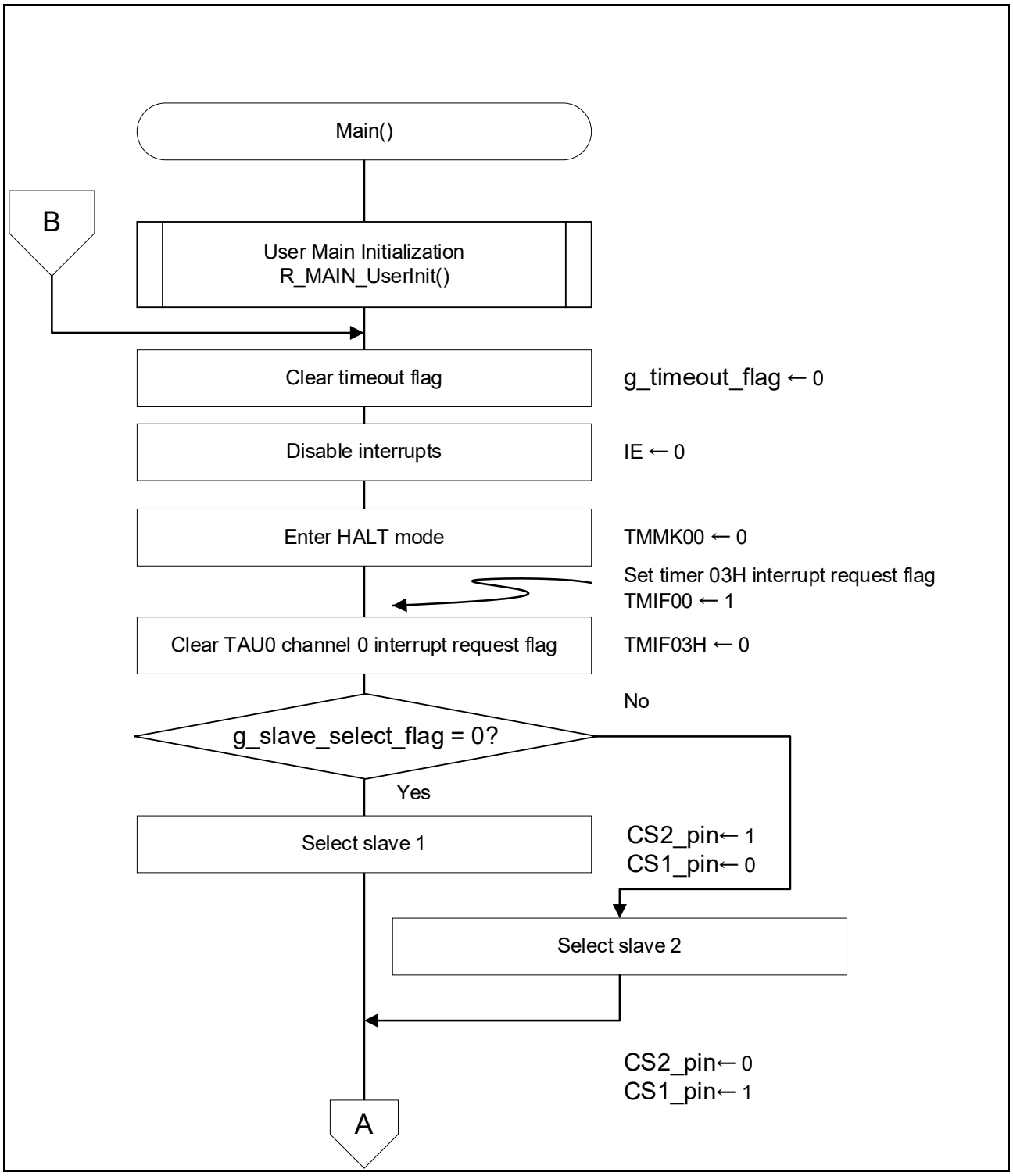
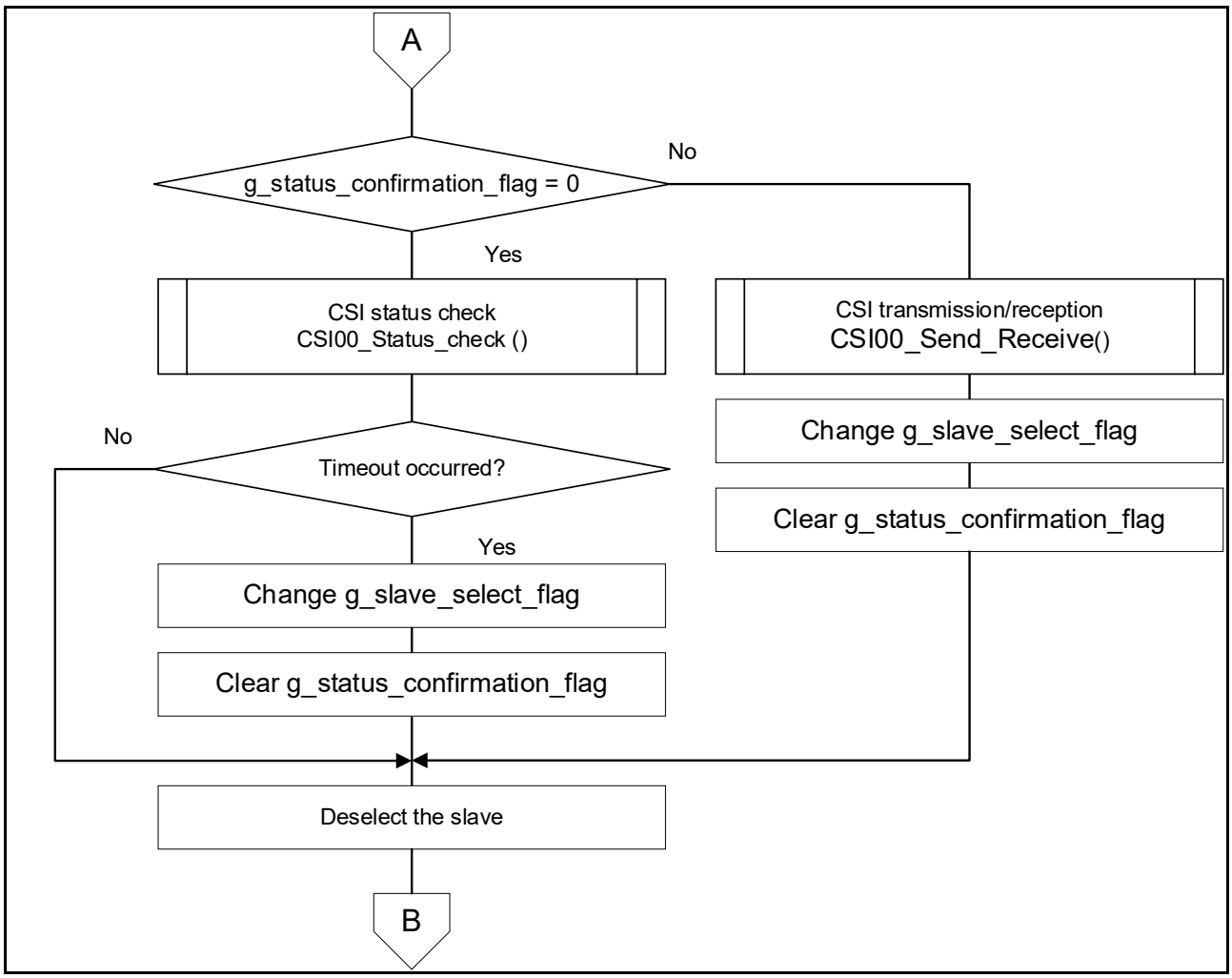


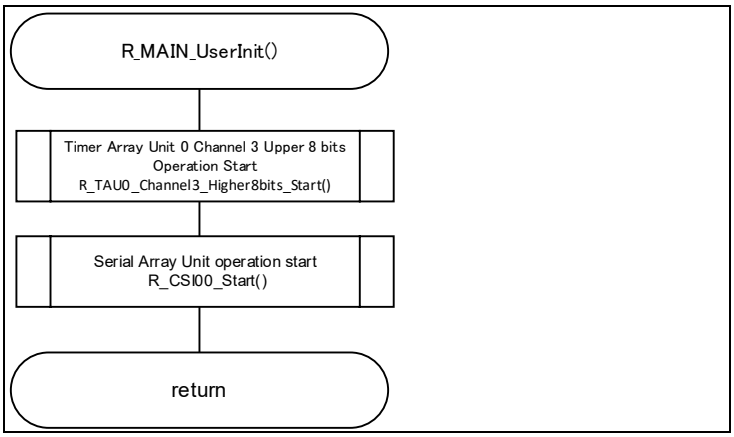
Figure 5-9 Main Processing 2/2



5.6.9 User Main Initialization

Figure 5-10 shows the flow of User Main Initialization.

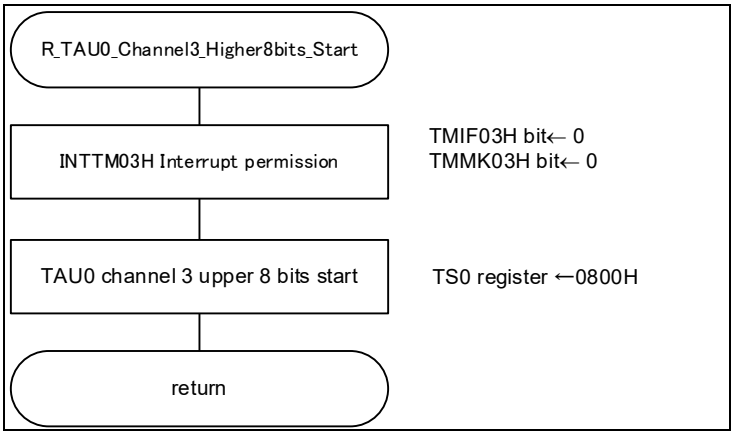
Figure 5-10 User Main Initialization



5.6.10 TAU0 channel 3 upper 8 bits start operation

Figure 5-11 shows the flow of TAU0 channel 3 upper 8 bits start operation.

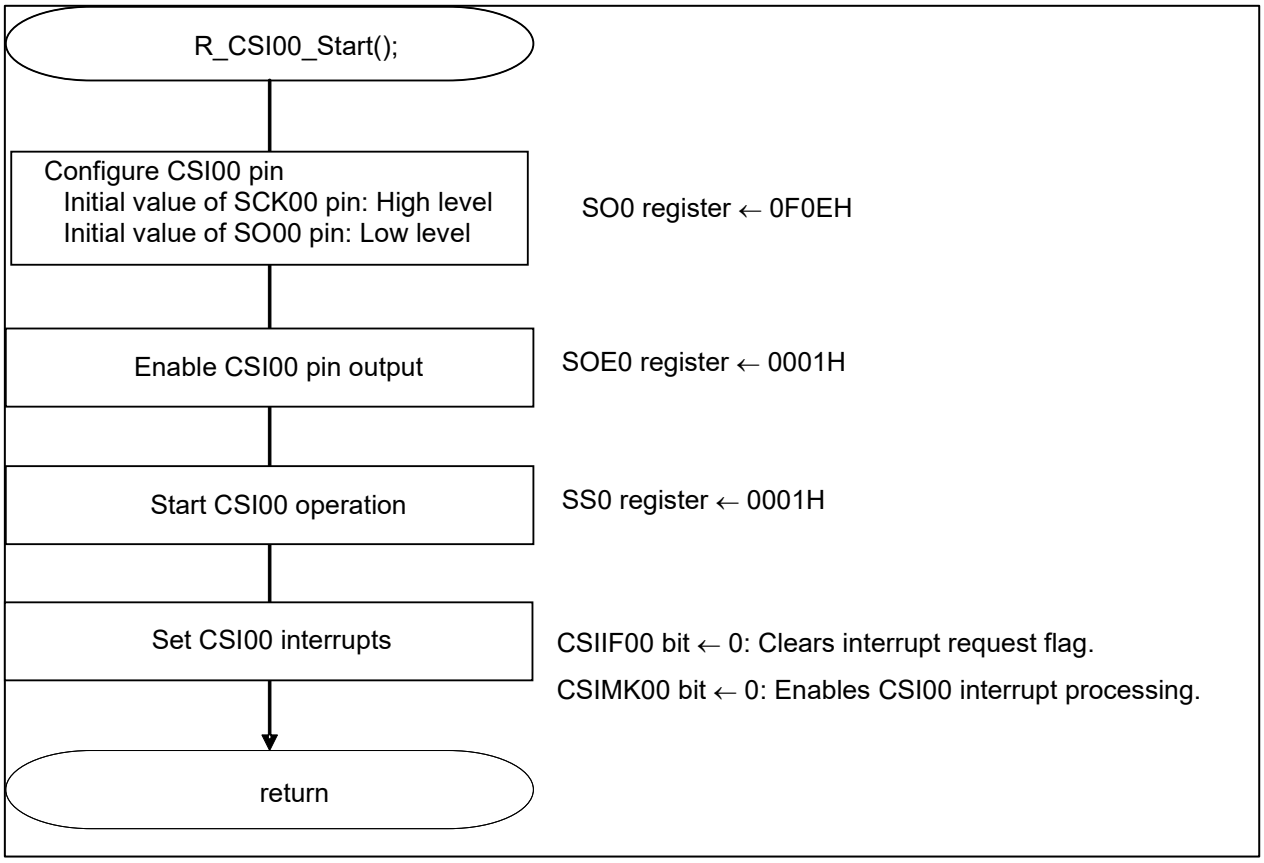
Figure 5-11 TAU0 channel 3 upper 8 bits start operation



5.6.11 CSI00 start operation

Figure 5-12 shows the flow of CSI00 start operation.

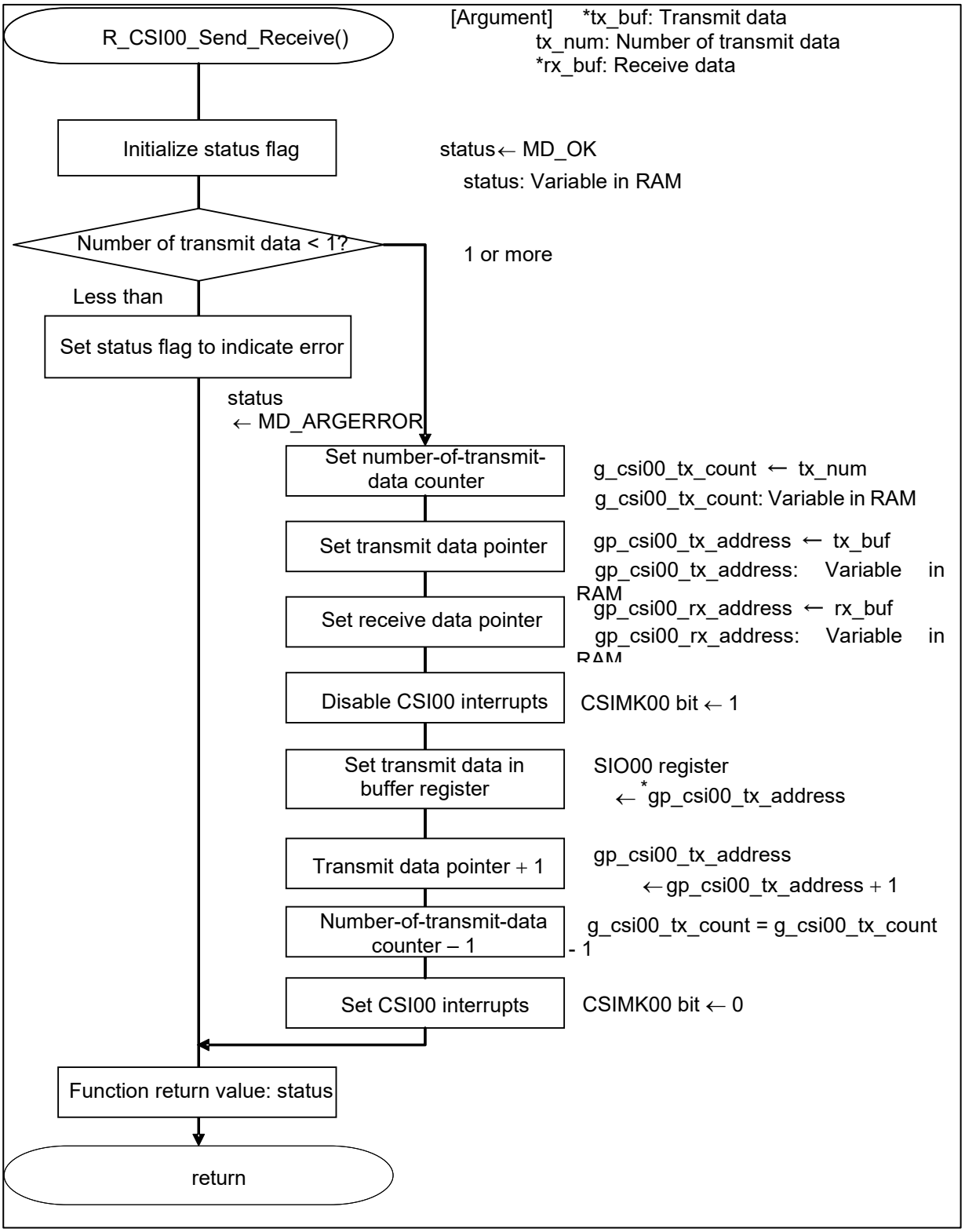
Figure 5-12 CSI00 start operation



5.6.12 CSI00 Transmit/Receive Processing

Figure 5-13 shows the flow of CSI00 Transmit/Receive Processing.

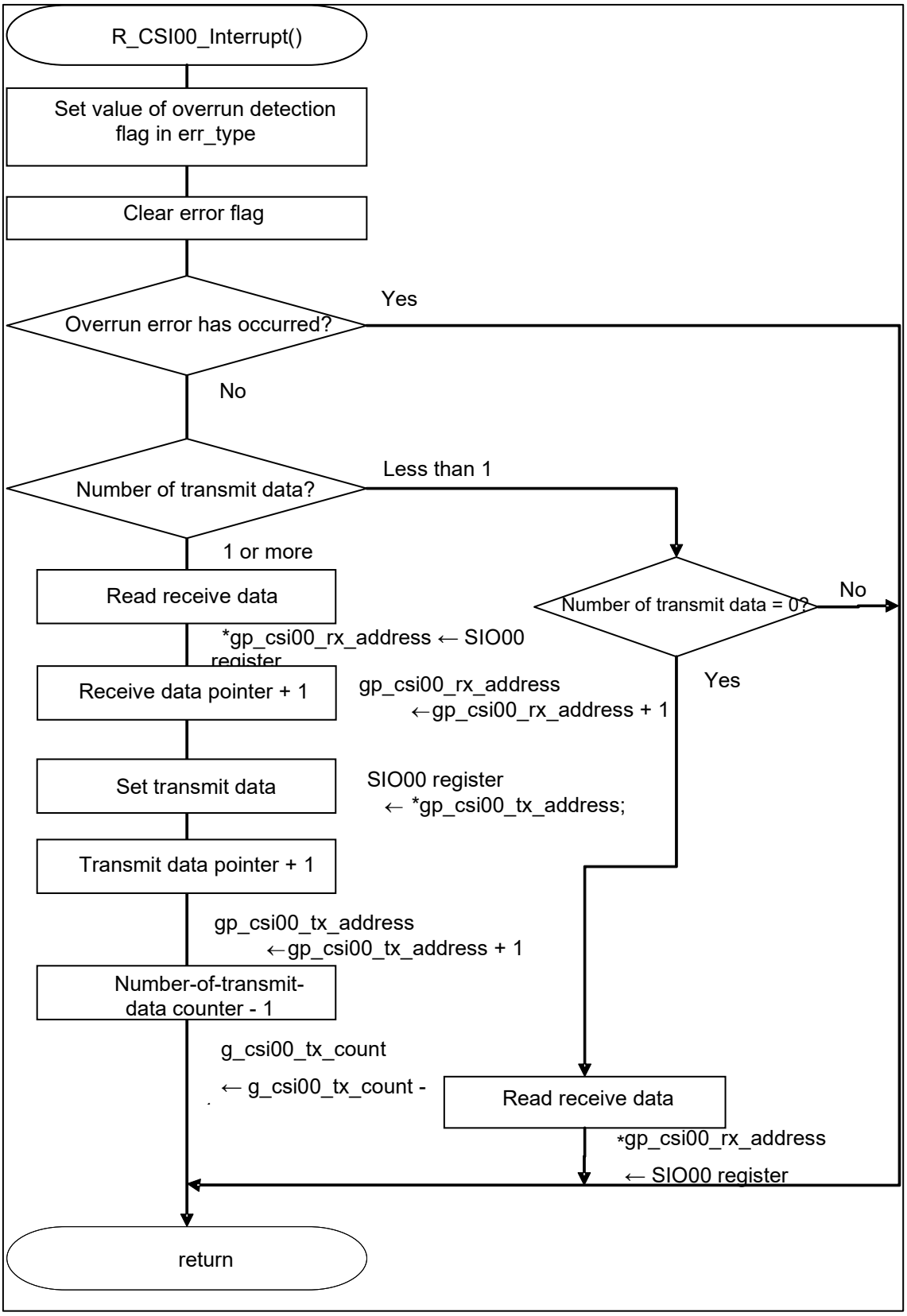
Figure 5-13 CSI00 Transmit/Receive Processing



5.6.13 INICSI00 interrupt processing

Figure 5-14 shows the flow of INICSI00 interrupt processing.

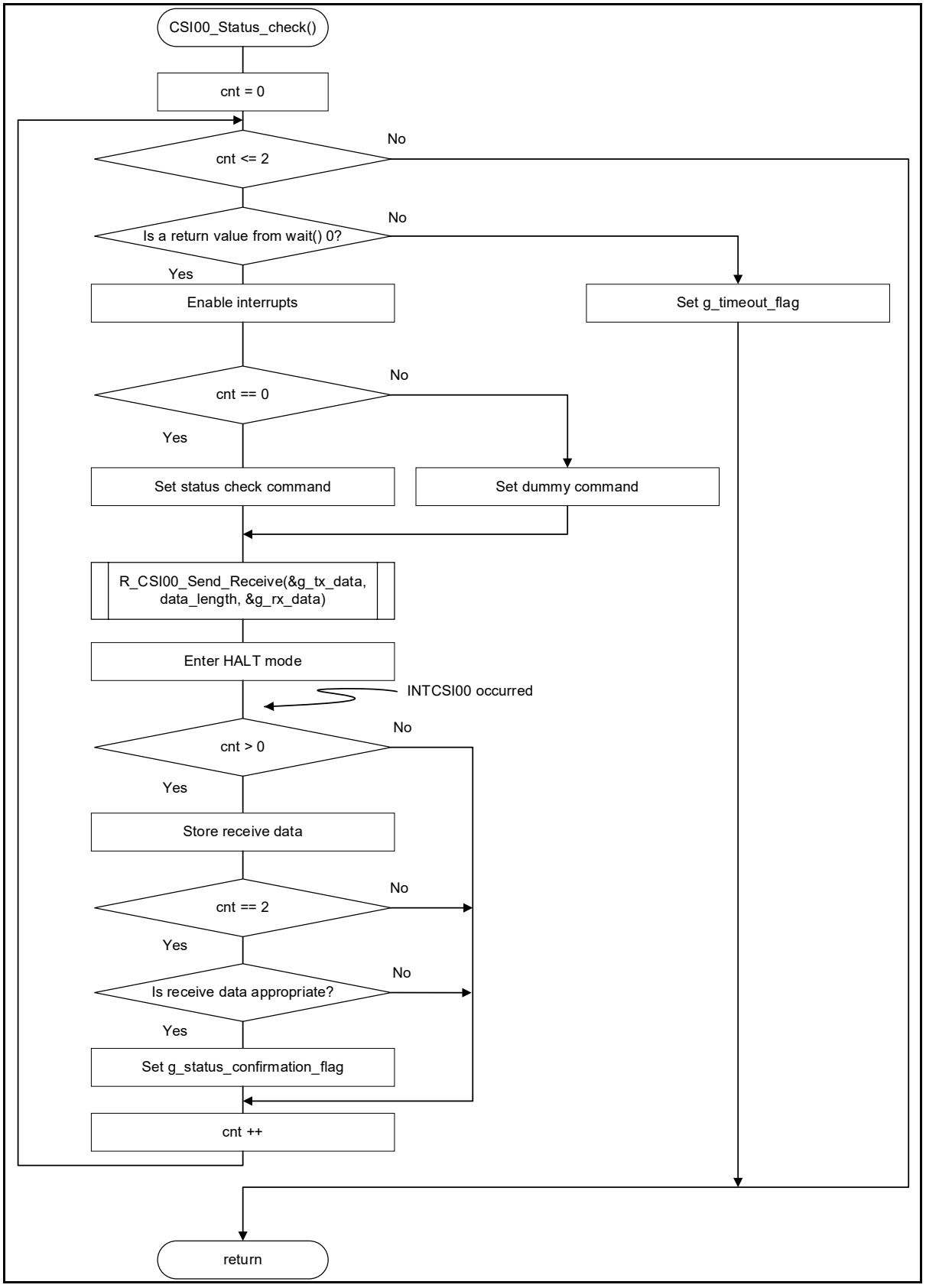
Figure 5-14 INICSI00 interrupt processing



5.6.14 Flowchart of CSI Status Check

Figure 5-15 shows the flow of status checking.

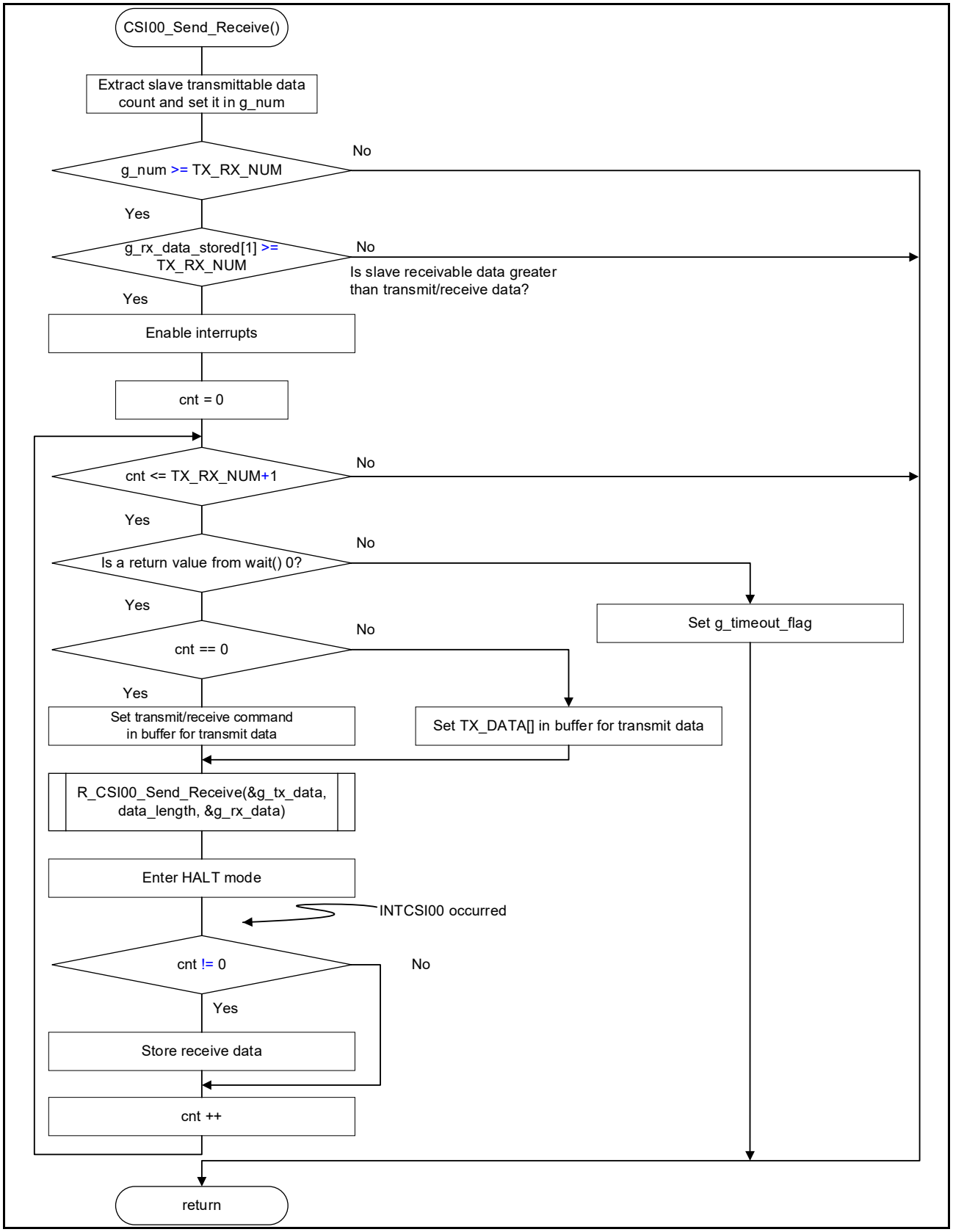
Figure 5-15 CSI Status Check



5.6.15 Flowchart of CSI Transmission/Reception

Figure 5-16 shows the flow of transmission and reception.

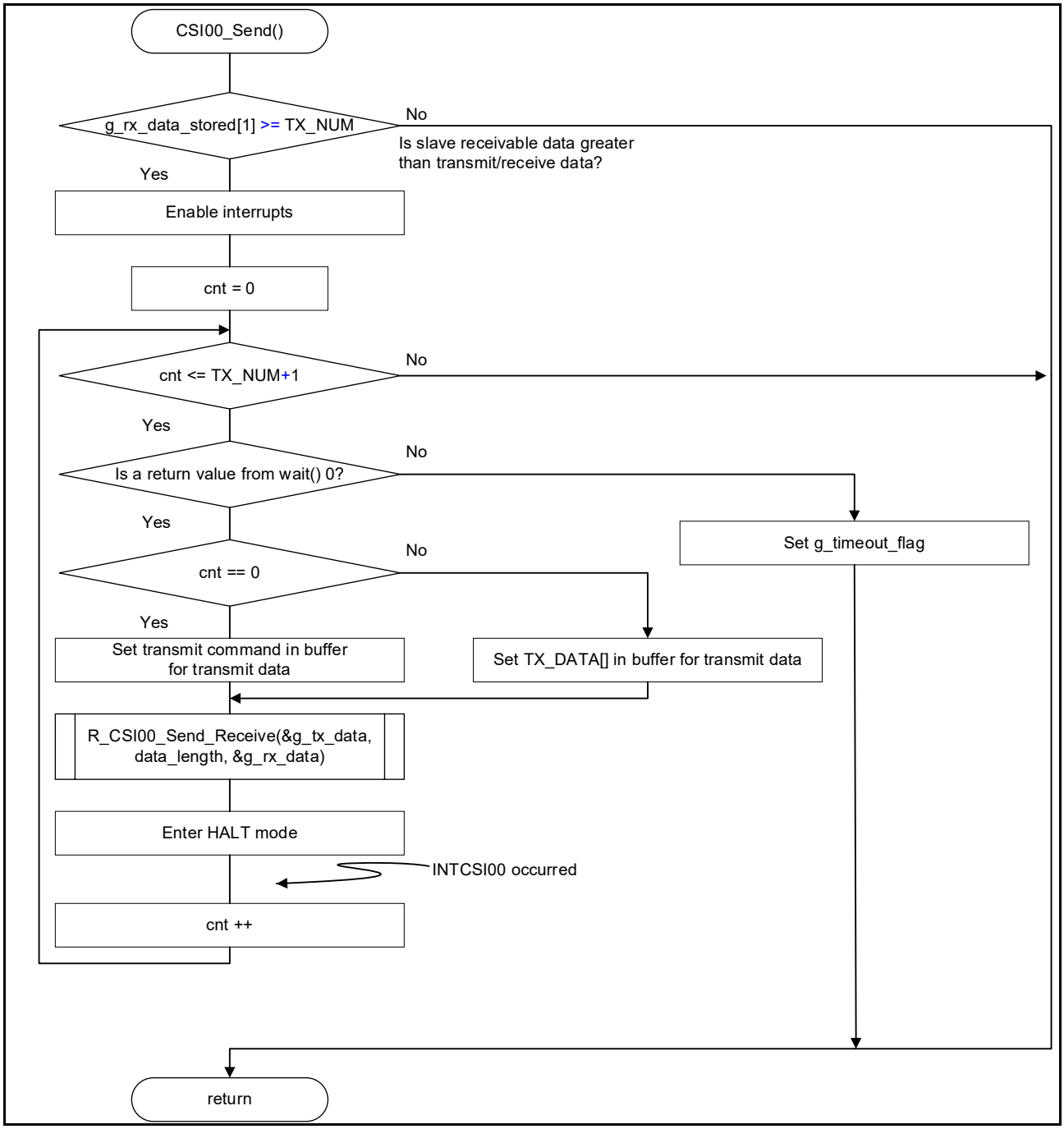
Figure 5-16 CSI Transmission/Reception



5.6.16 Flowchart of CSI Transmission

Figure 5-17 shows the flow of transmission.

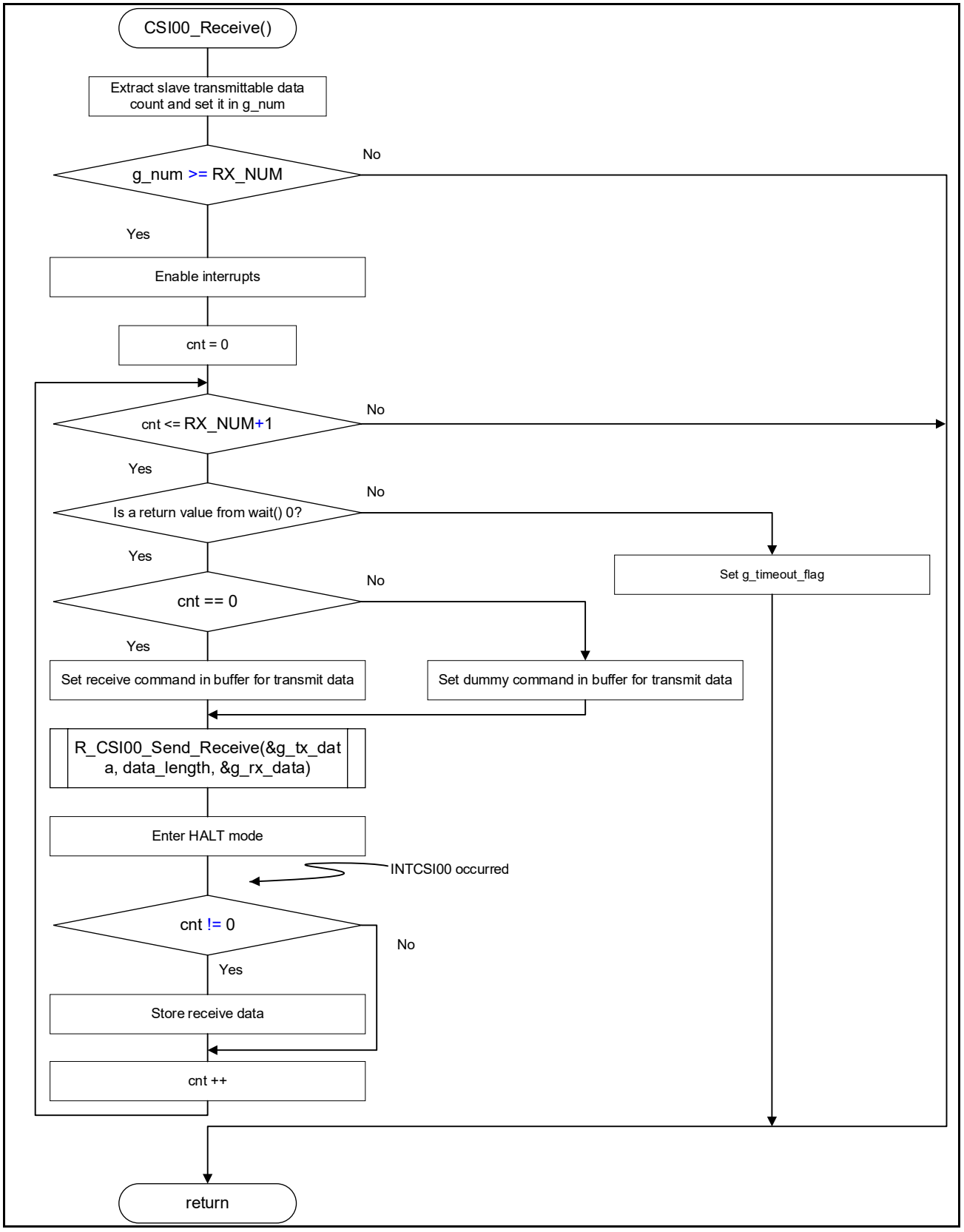
Figure 5-17 CSI Transmission



5.6.17 Flowchart of CSI Reception

Figure 5-18 shows the flow of reception.

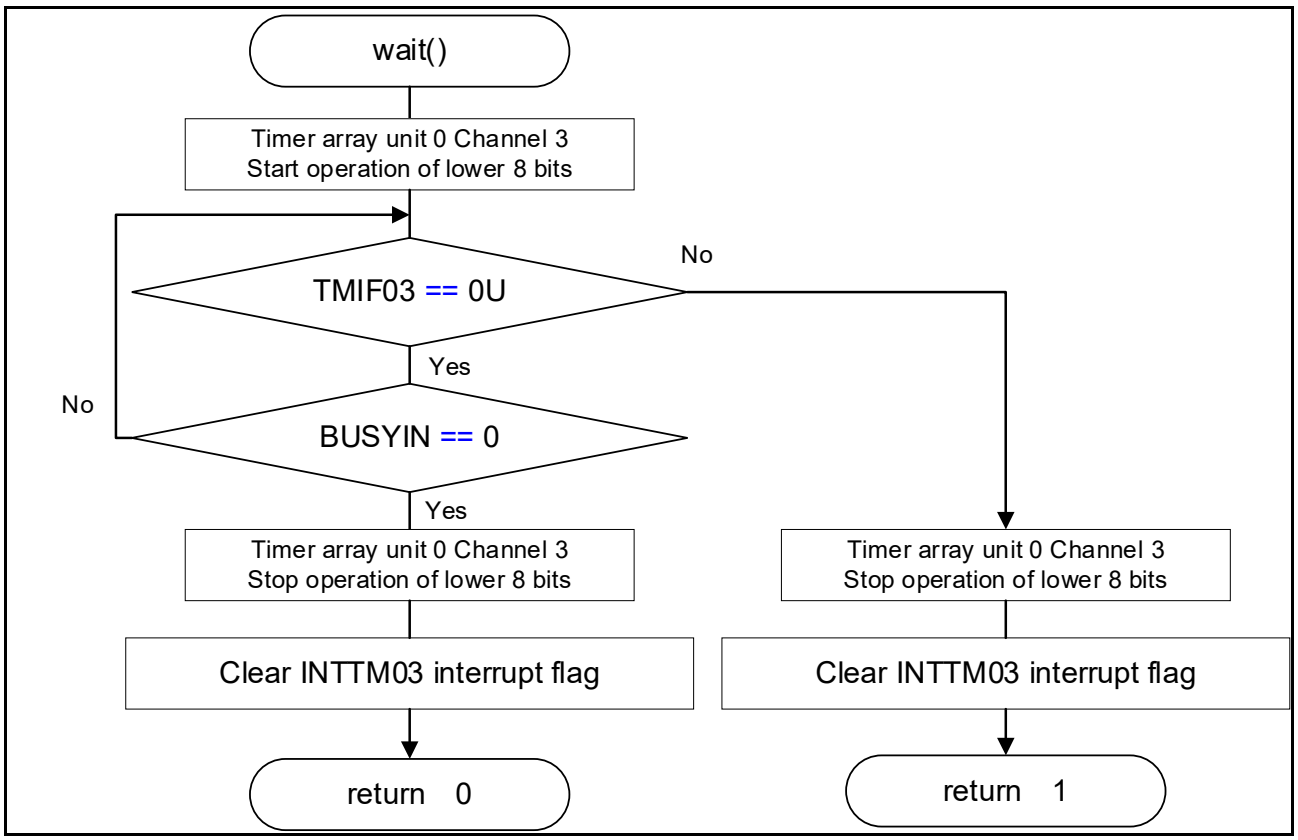
Figure 5-18 CSI Reception



5.6.18 Flowchart of Wait for Slave Response

Figure 5-19 shows the flow of wait for slave response.

Figure 5-19 Wait for Slave Response



6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

RL78/G13 User’s Manual: Hardware (R01UH0146J)
RL78 family user's manual software (R01US0015)
The latest versions can be downloaded from the Renesas Electronics website.

Technical update
The latest versions can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2023.6.15	-	First Edition

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

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

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can 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 must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. 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. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the 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.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. 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, for example, 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.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.