RL78/G13

Serial Interface IICA (for Slave Transmission/Reception)

Introduction

This application note describes slave transmission and reception implemented via the serial interface IICA. Using IICA, the single master system described here performs slave operation (address reception, and data transmission and reception).

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

This application note describes a method for using a serial interface IICA for master transmission/reception (address transmission data transmission and reception) in a single-master configuration. It is assumed that each slave has a register to specify an address in the slave.

When the slave address 0b1010000 is specified, the slave becomes capable of transmitting and receiving data.

The following is a summary of the slave specifications assumed in this application note.

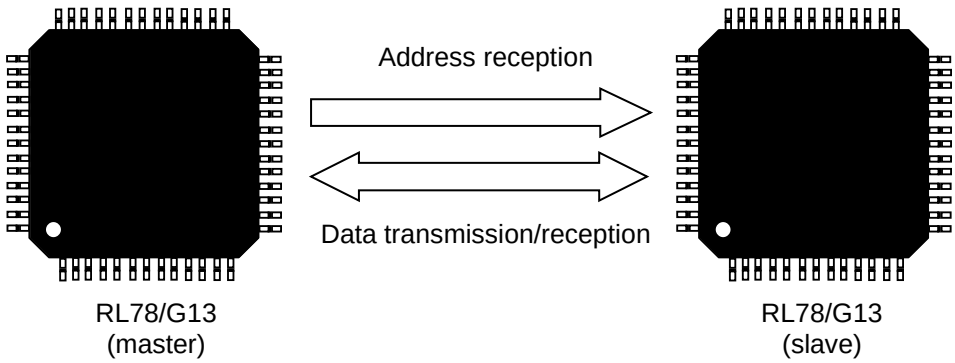
- Slave address: 0b1010000[note]
- At the specified address, an arbitrary number of data bytes can be read out or written.
- The slave serial RAM area is register addresses 0x80 to 0xFF (128 bytes). Slave operation is specified using the command register at register address 0x00.
- When 0x01 to 0x7F is specified as the register address, NACK is returned and there is disengagement from communication.
- When the register address exceeds 0xFF, the serial RAM area is selected, and only the lower 7 bits of the specified register address are handled as valid.

Caution: This sample code corresponds to the Application Note for the RL78/G13 serial interface IICA (master transmission/reception) (R01AN2759E).

Peripheral functions used and applications are shown in Table 1.1, and the IIC communication is summarized in Figure 1.1. IIC communication timing charts appear in Figure 1.2 to Figure 1.8.

Table 1.1 Peripheral Function to be Used and Its Use

Peripheral Function	Use
Serial interface IICA	IIC communication in a single master system (using the SCLA0 and SDAA0 pins)

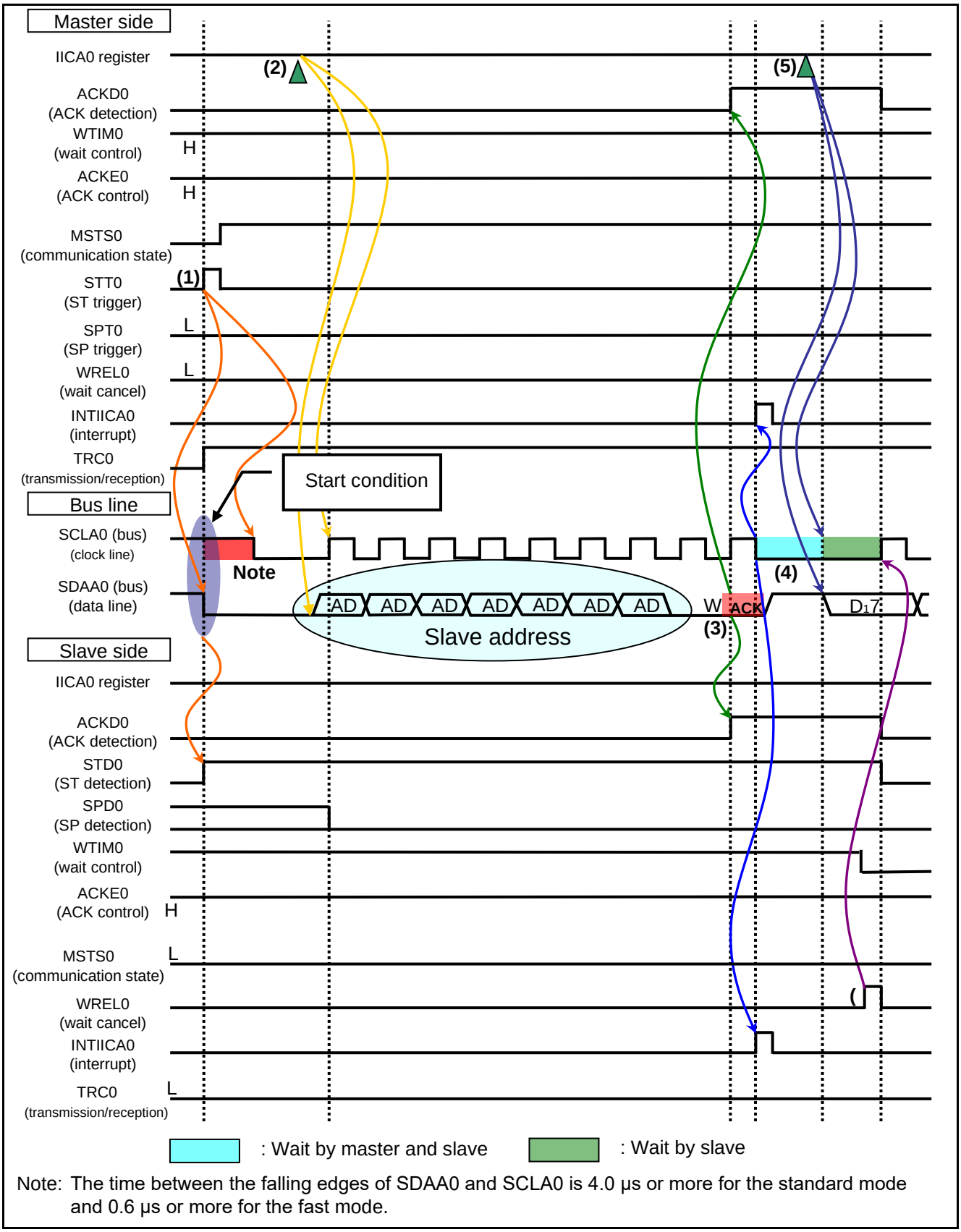


Address reception: Each device connected to the IIC bus has a unique address. Each device receives the address of a transfer destination (slave) from the master. That is, it receives one byte of data consisting of 7 bits (indicating the address) and 1 bit (indicating the transfer direction). The slave generates an acknowledgement after receiving one byte of data.
Data transmission/reception: The slave sends/receives data to/from the master after receiving the address.

Figure 1.1 Overview of IIC Communication

1.1 IIC Communication Timing Chart

(1) Master-to-slave communication 1 (start condition – address – data)



Note: The time between the falling edges of SDAA0 and SCLA0 is 4.0 µs or more for the standard mode and 0.6 µs or more for the fast mode.

Figure 1.2 IIC Communication Timing Chart (Master-to-Slave Communication Example) (1/4)

(1) The start condition trigger is set (STT0 = 1) on the master side. Then, the SDAA0 line falls, thereby generating a start condition. Later, when the start condition is detected (STD0 = 1), the master enters a master device communication state (MSTS0 = 1). The SCLA0 line falls at the end of the hold period. This completes preparations for communication.

(2) The values of the address and data direction bit W (transmission) are written to the IICA0 register on the master side. Then, the slave address is transmitted.

(3) If the received address and slave address match Note, the slave hardware sends ACK0 to the master. When the ninth clock signal rises, the master detects ACK (ACKD0 = 1).

(4) When the ninth clock signal falls, an address transmission end interrupt (INTIICA0) occurs on the master side. If the addresses match, an address match interrupt (INTIICA0) occurs on the slave side. Both the master and the slave which has the matching address enter a wait state (SCLA0 line: Low)Note.

(5) The master writes transmit data to the IICA0 register and cancels the wait.

(6) The slave selects an 8-clock wait (WTIM0 = 0) because it receives data. When the slave cancels the wait (WREL0 = 1), the master starts transferring data to the slave.

Note: When there is a mismatch between a received address and the local address, the slave side does not return ACK to the master side (NACK). Moreover, a slave-side INTIICAO interrupt (address match interrupt) does not occur, and a wait state is not entered on the slave side. However, on the master side an INTIICAO interrupt (address transmission complete interrupt) occurs, regardless of whether the result is ACK or NACK.

In the RL78 family, the local address (7 bits) is represented by the upper 7 bits in the SVA0 register. The lowermost bit in the SVA0 register is fixed at 0.

(2) Master-to-slave communication 2 (address – data – data)

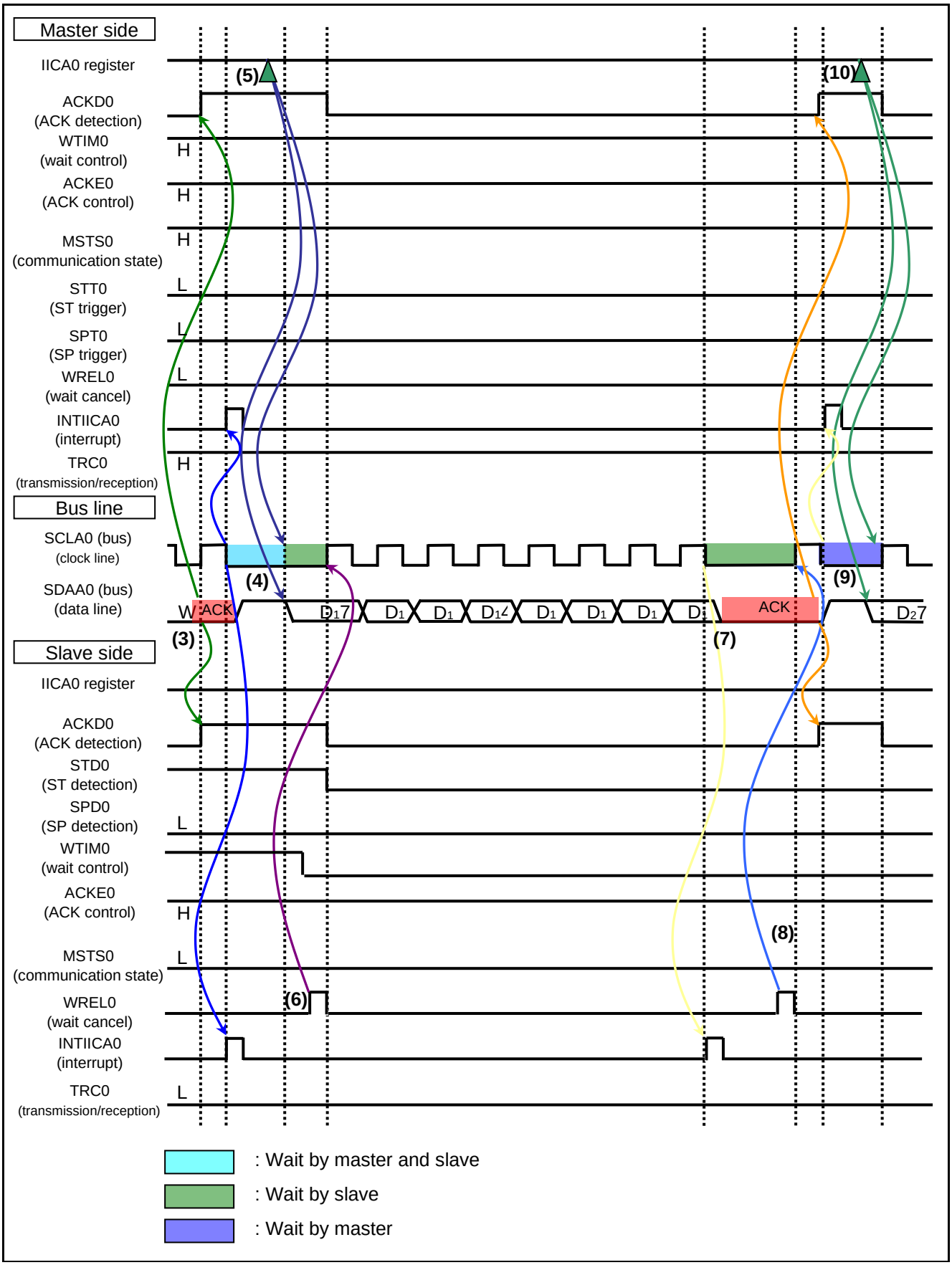


Figure 1.3 IIC Communication Timing Chart (Master-to-Slave Communication Example) (2/4)

(3) If the received address and slave address match, the slave hardware sends ACK to the master. When the ninth clock signal rises, the master detects ACK (ACKD0 = 1).

(4) When the ninth clock signal falls, an address transmission end interrupt (INTIICA0) occurs on the master side. If the addresses match, an address match interrupt (INTIICA0) occurs on the slave side. Both the master and the slave which has the matching address enter a wait state (SCLA0 line: Low).

(5) The master writes transmit data to the IICA0 register and cancels the wait.

(6) The slave selects an 8-clock wait (WTIM0 = 0) because it receives data. When the slave cancels the wait (WREL0 = 1), the master starts transferring data to the slave.

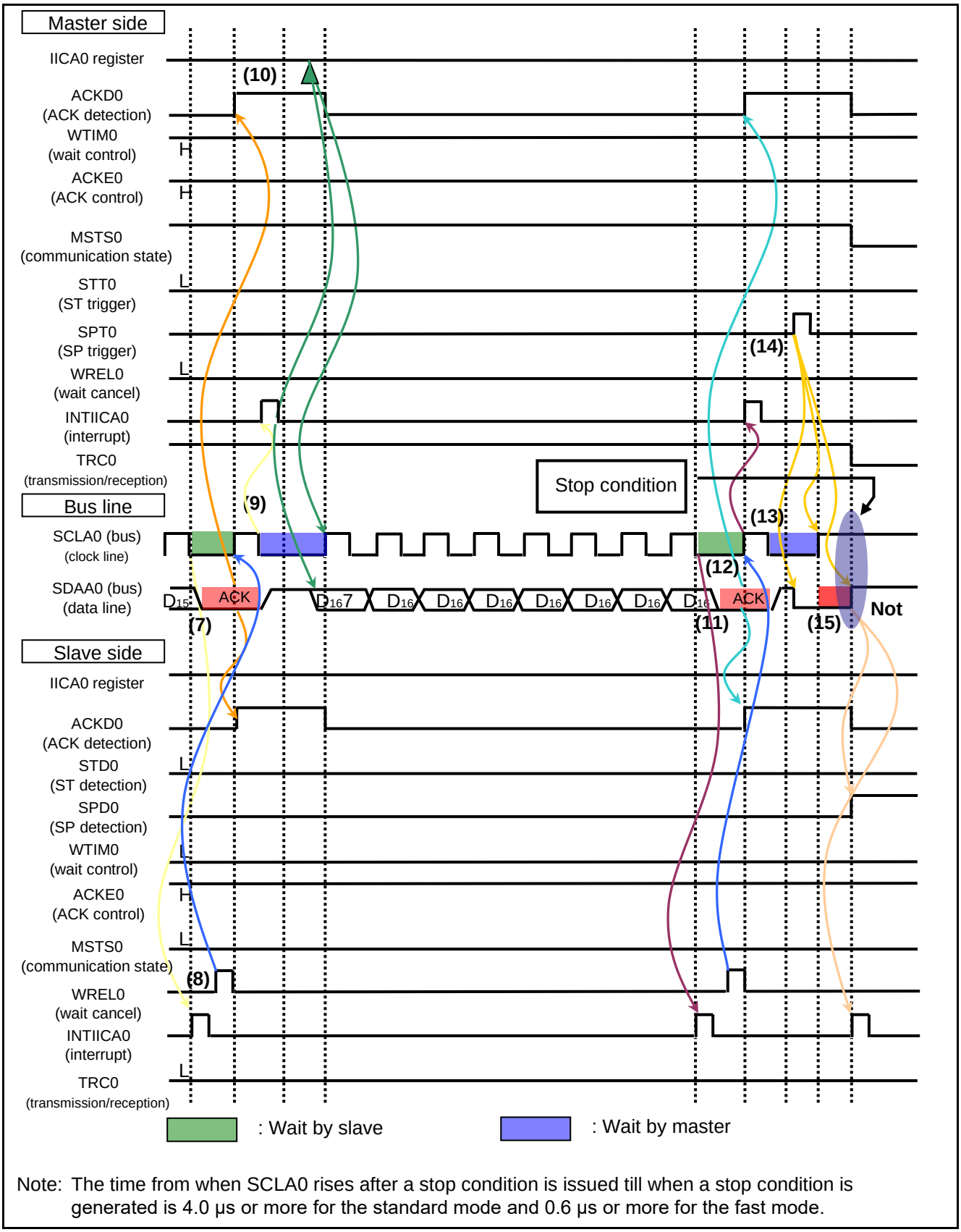
(7) When the eighth clock signal falls after the data transfer, the slave hardware generates a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the slave side.

(8) When the slave reads the receive data and cancels the wait (WREL0 = 1), the slave sends ACK to the master. When the ninth clock signal rises, the master detects ACK (ACKD0 = 1).

(9) When the ninth clock signal falls, the master generates a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the master side.

(10) The master writes transmit data to the IICA0 register and cancels the wait. Then, the master starts transferring data to the slave.

(3) Master-to-slave communication 3 (data – data – stop condition)

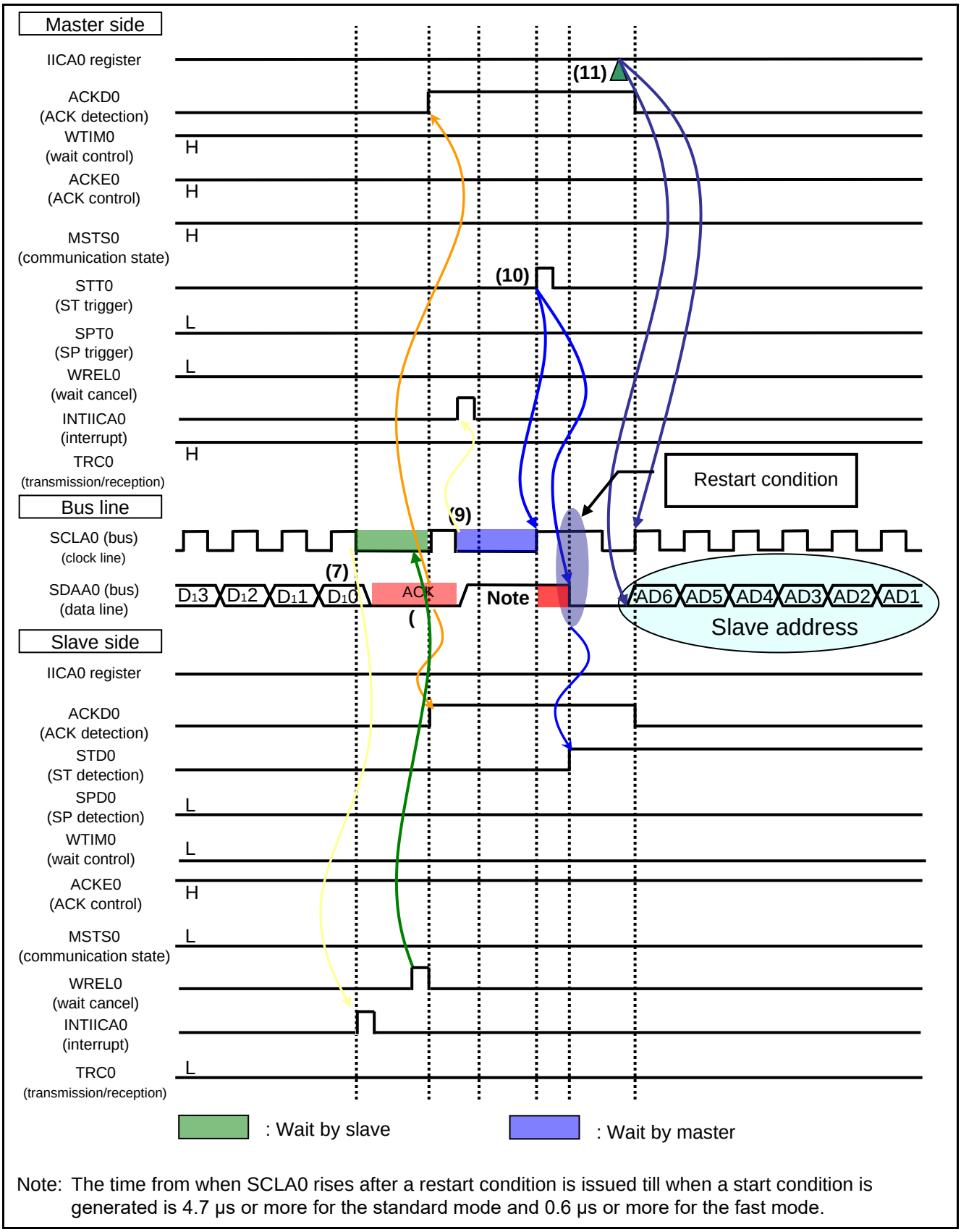


Note: The time from when SCLA0 rises after a stop condition is issued till when a stop condition is generated is 4.0 µs or more for the standard mode and 0.6 µs or more for the fast mode.

Figure 1.4 IIC Communication Timing Chart (Master-to-Slave Communication Example) (3/4)

(7) When the eighth clock signal falls after the data transfer, the slave hardware generates a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the slave side.
(8) When the slave reads the receive data and cancels the wait (WREL0 = 1), the slave sends ACK to the master. When the ninth clock signal rises, the master detects ACK (ACKD0 = 1).
(9) When the ninth clock signal falls, the master generates a wait (SCLA0 line: Low) and an address transmission end interrupt (INTIICA0) occurs on the master side.
(10) The master writes transmit data to the IICA0 register and cancels the wait. Then, the master starts transferring the data to the slave.
(11) When the eighth clock signal falls after the data transfer, the slave hardware generates a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the slave side.
(12) When the slave reads the receive data and cancels the wait (WREL0 = 1), the slave sends ACK to the master. When the ninth clock signal rises, the master detects ACK (ACKD0 = 1).
(13) When the ninth clock signal falls, the master generates a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the master side.
(14) When the stop condition trigger is set (SPT0 = 1), the SDAA0 line falls and the SCLA0 line rises. Upon the elapse of the stop condition setup time, the SDAA0 line rises, thereby generating a stop condition.
(15) When the stop condition is generated, the slave detects it (SPD0 = 1) and a IICA0 interrupt (stop condition interrupt) occurs on the slave side.

(4) Master-to-slave communication 4 (data – restart condition – address)



Note: The time from when SCLA0 rises after a restart condition is issued till when a start condition is generated is 4.7 μs or more for the standard mode and 0.6 μs or more for the fast mode.

Figure 1.5 IIC Communication Timing Chart (Master-to-Slave Communication Example) (4/4)

(7) When the eighth clock signal falls after the data transfer, the slave hardware generates a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the slave side.

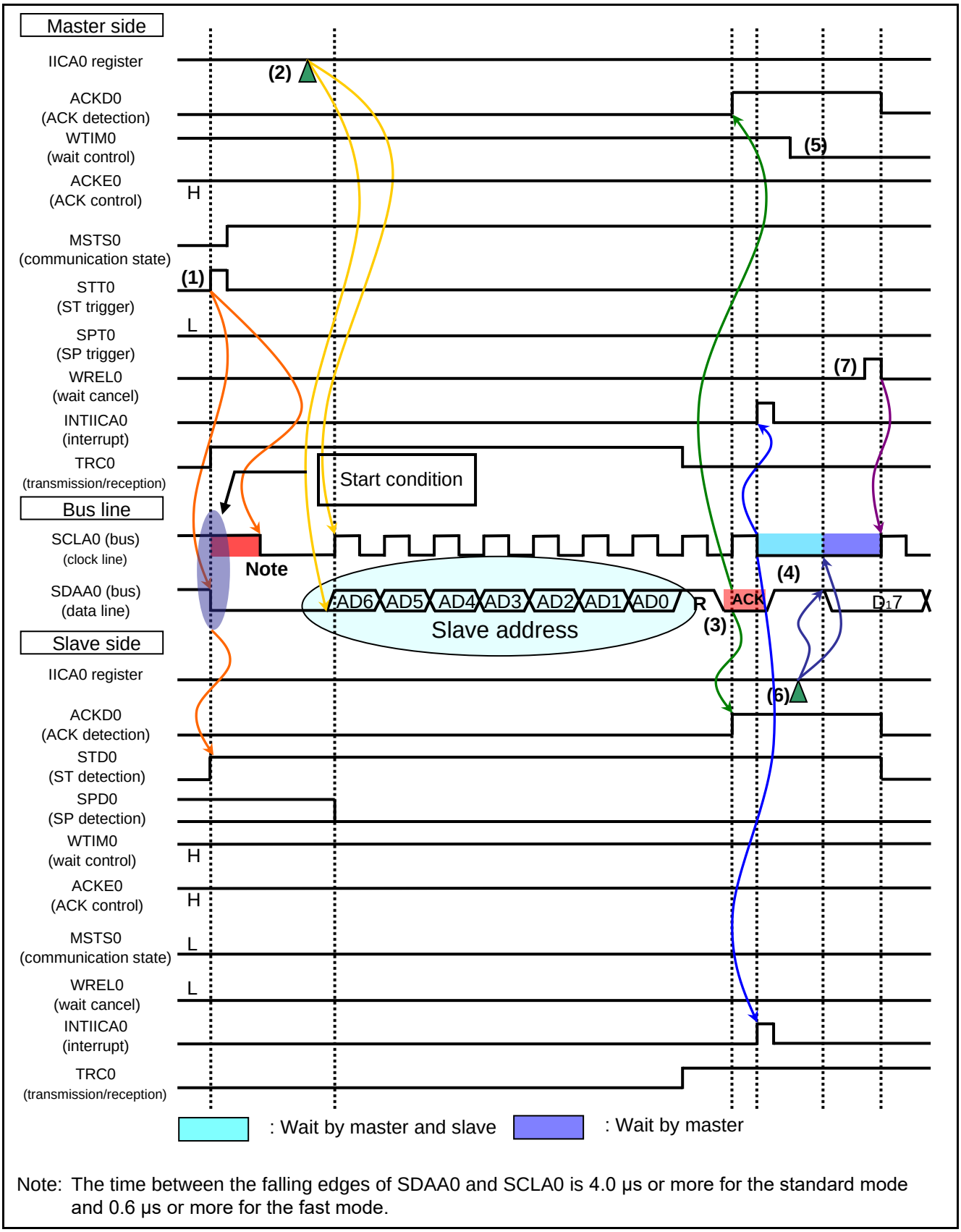
(8) The slave reads the receive data and cancels the wait (WREL0 = 1). Then, the slave sends ACK to the master. When the ninth clock signal rises, the master detects ACK (ACKD0 = 1).

(9) When the ninth clock signal falls, the master generates a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the master side.

(10) The start condition trigger is set (STT0 = 1) on the master side again. Then, the SCLA0 line rises. Upon the elapse of the restart condition setup time, the SDAA0 line falls, thereby generating a start condition. Later, at the end of the hold period after the start condition is detected (STD0 = 1), the bus clock line falls, thereby completing preparations for communication.

(11) The master writes the slave address to the IICA0 register and starts transferring the address to the slave.

(5) Slave-to-master communication 1 (start condition – address – data)



Note: The time between the falling edges of SDAA0 and SCLA0 is 4.0 µs or more for the standard mode and 0.6 µs or more for the fast mode.

Figure 1.6 IIC Communication Timing Chart (Slave-to-Master Communication Example) (1/3)

(1) The start condition trigger is set (STT0 = 1) on the master side. Then, the SDAA0 line falls, thereby generating a start condition. Later, when the start condition is detected (STD0 = 1), the master enters a master device communication state (MSTS0 = 1). The SCLA0 line falls at the end of the hold period. This completes preparations for communication.

(2) The values of the address and data direction bit R (reception) are written to the IICA0 register on the master side. Then, the slave address is transmitted.

(3) If the received address and slave address match Note, the slave hardware sends ACK to the master. When the ninth clock signal rises, the master detects ACK (ACKD0 = 1).

(4) When the ninth clock signal falls, an address transmission end interrupt (INTIICA0) occurs on the master side. If the addresses match, an address match interrupt (INTIICA0) occurs on the slave side. Both the master and the slave which has the matching address enter a wait state (SCLA0 line: Low).

(5) The master selects an 8-clock wait (WTIM0 = 0) because it receives data.

(6) The slave writes transmit data to the IICA0 register and cancels the wait.

(7) When the master cancels the wait (WREL0 = 1), the slave starts transferring data to the master.

Note: When there is a mismatch between a received address and the local address, the slave side does not return ACK to the master side (NACK). Moreover, a slave-side INTIICAO interrupt (address match interrupt) does not occur, and a wait state is not entered on the slave side. However, on the master side an INTIICAO interrupt (address transmission complete interrupt) occurs, regardless of whether the result is ACK or NACK.

In the RL78 family, the local address (7 bits) is represented by the upper 7 bits in the SVA0 register. The lowermost bit in the SVA0 register is fixed at 0.

(6) Slave-to-master communication 2 (address – data – data)

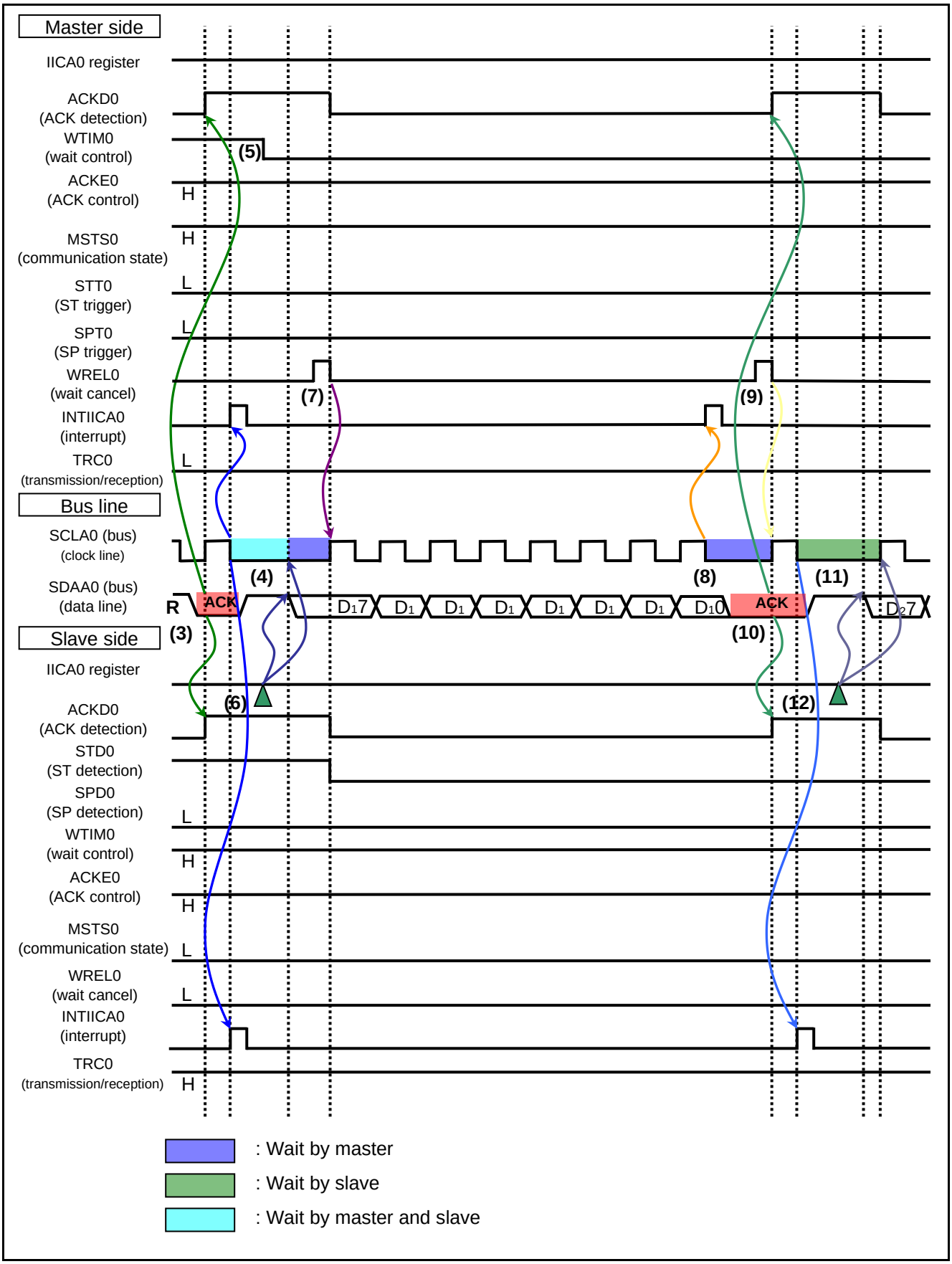
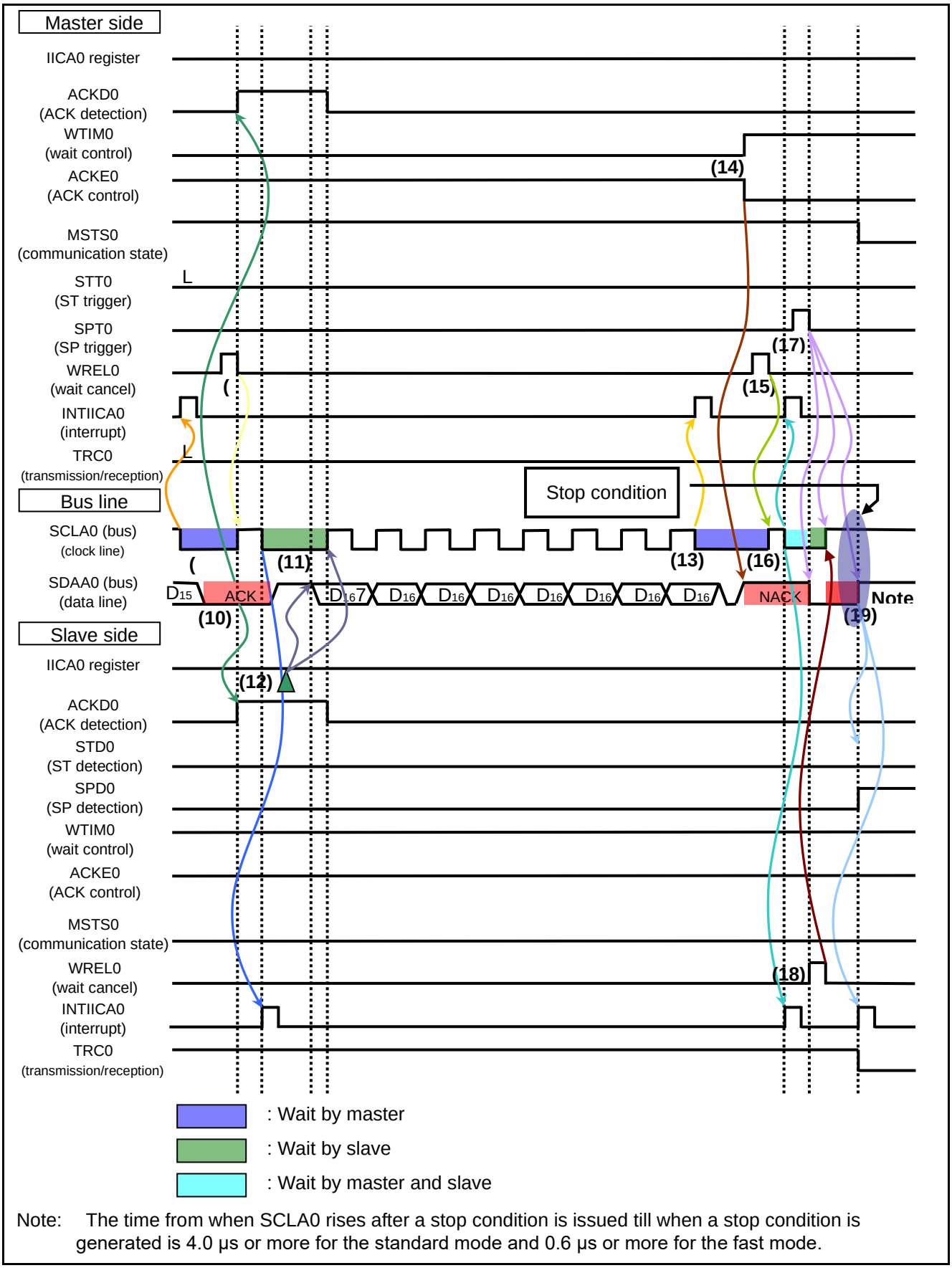


Figure 1.7 IIC Communication Timing Chart (Slave-to-Master Communication Example) (2/3)

(3) If the received address and slave address match, the slave hardware sends ACK to the master. When the ninth clock signal rises, the master detects ACK (ACKD0 = 1).
(4) When the ninth clock signal falls, an address transmission end interrupt (INTIICA0) occurs on the master side. If the addresses match, an address match interrupt (INTIICA0) occurs on the slave side. Both the master and the slave which has the matching address enter a wait state (SCLA0 line: Low).
(5) The master selects an 8-clock wait (WTIM0 = 0) because it receives data.
(6) The slave writes transmit data to the IICA0 register and cancels the wait.
(7) When the master cancels the wait (WREL0 = 1), the slave starts transferring data to the master.
(8) When the eighth clock signal falls, the master generates a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the master side. The master hardware sends ACK to the slave.
(9) The master reads the receive data and cancels the wait (WREL0 = 1).
(10) When the ninth clock signal rises, the slave detects ACK (ACKD0 = 1).
(11) When the ninth clock signal falls, the slave generates a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the slave side.
(12) The slave writes transmit data to the IICA0 register and cancels the wait. Then, the slave starts transferring data to the master.

(7) Slave-to-master communication 3 (data – data – stop condition)



Note: The time from when SCLA0 rises after a stop condition is issued till when a stop condition is generated is 4.0 μs or more for the standard mode and 0.6 μs or more for the fast mode.

Figure 1.8 IIC Communication Timing Chart (Slave-to-Master Communication Example) (3/3)

(8) When the eighth clock signal falls, the master generates a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the master side. The master hardware sends ACK to the slave.

(9) The master reads the receive data and cancels the wait (WREL0 = 1).
(10) When the ninth clock signal rises, the slave detects ACK (ACKD0 = 1).
(11) When the ninth clock signal falls, the slave generates a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the slave side.
(12) The slave writes transmit data to the IICA0 register and cancels the wait. Then, the slave starts transferring data to the master.
(13) When the eighth clock signal falls, a transfer end interrupt (INTIICA0) occurs on the master side and the master generates a wait (SCLA0 line: Low). The master hardware sends ACK to the slave.
(14) The master sets a NACK response (ACKE0 = 0) to inform the slave that the master has sent the last data (at the end of communication). Then, the master changes the wait time to 9 clock periods (WTIM0 = 1).
(15) After the master cancels the wait (WREL0 = 1), the slave detects NACK (ACKD0 = 0) at the rising edge of the ninth clock signal.
(16) When the ninth clock signal falls, the master and slave generate a wait (SCLA0 line: Low) and a transfer end interrupt (INTIICA0) occurs on the master and slave sides.
(17) When the master issues a stop condition (SPT0 = 1), the SDAA0 line falls, thereby canceling the wait on the master side. Later, the master waits until the SCLA0 line rises.
(18) The slave cancels the wait (WREL0 = 1) to terminate communication. Then, the SCLA0 line rises.
(19) The master confirms that the SCLA0 line has risen. Upon the elapse of the stop condition setup time after this confirmation, the master makes the SDAA0 line rise and issues a stop condition. When the stop condition is generated, the slave detects the stop condition (SPD0 = 1) and a stop condition interrupt (INTIICA0) occurs on the master and slave sides.

1.2 Control of Serial RAM

1.2.1 Command Settings

In this application note, commands are used to specify slave operations. The command setting sequence is shown in **Figure 1.9**, and the command setting timing chart appears in **Figure 1.10**

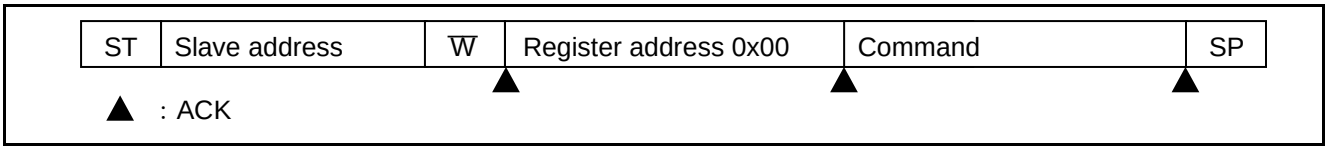


Figure 1.9 Command Setting Sequence

Upon receiving the slave address 0b1010000 following a start condition (ST), the slave returns ACK[note] after receiving a transfer direction specification bit. When the transfer direction $\overline{W}$ is specified, it is determined that the second-next received data will be written to the address specified by the next received data. Upon receiving the register address 0×00, the slave returns ACK, and determines that the next received data is a command. Upon receiving a stop condition (SP), communication is ended.

Upon receiving an address and data, the slave returns ACK. If the address or data cannot be confirmed, the SDA maintains a high level state (NACK response).

Note: In the RL78 family, if the local address and the slave address match, an ACK response is sent automatically.

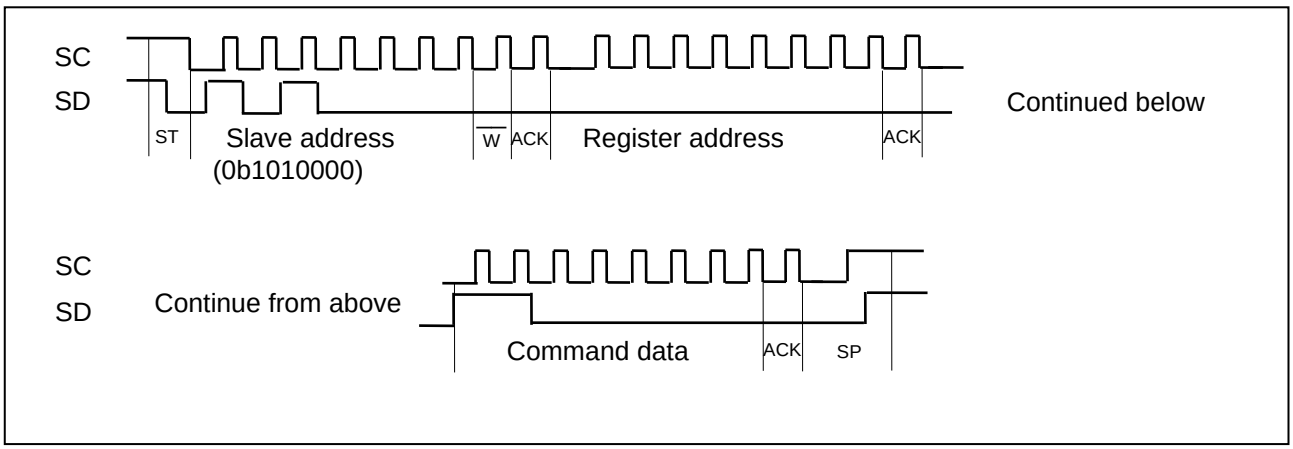


Figure 1.10 Command Setting Timing Chart

A list of command functions appears in **Table 1.2**

Bit 7 (the MSB) of the data written to register address 0x00 (command register) indicates whether the command is valid or invalid. When bit 7 is 1, the slave judges that the command is valid, and when bit 7 is 0, the slave ignores the command as invalid. Bit 6 indicates whether memory functions are used or not. When memory functions are used, bit 6 is set to 1. Bits 5 to 3 are unused, and so are all set to 0. Bit 2 indicates whether writing is forbidden or not. While bit 2 is set to 1 (from when writing is prohibited until writing is again permitted), upon write data reception the slave sends a NACK response and disengages from communication. Bit 1 indicates whether memory is initialized or not. When bit 1 is set to 1, the slave memory is initialized to initialization data specified by bit 0. After the completion of initialization, the slave sets bit 1 of the command register to 0.

Table 1.2 Command functions

Bit	Meaning	Explanation
7	Command setting	1: command valid, 0: command invalid
6	Memory function selection	1: use memory functions, 0: do not use memory functions
5 to 3	Unused	Fixed at 0
2	Write selection	1: writing forbidden, 0: writing permitted
1	Initialization selection	1: initialize memory (serial RAM area), 0: do nothing
0	Initialization data selection	1: value of lowest 7 bits of register address[note], 0: 0x00

Note: At each address in the serial RAM area, the value of the lowest 7 bits of the address is written. For example, at register addresses 0x80, 0x81, 0x82, the respective values 0x00, 0x01, 0x02 are written.

1.2.2 Continuous Data Writing

Regarding cases in which a register address is specified and data is written continuously to the serial RAM of a slave, the sequence is shown in Figure 1.11, and the timing chart appears in Figure 1.12.

Upon receiving the slave address 0b1010000 following a start condition (ST), the slave returns ACK[note] after receiving a transfer direction specification bit. When the transfer direction $\overline{W}$ is specified, it is determined that the second-next received data will be written to the address specified by the next received data. Upon receiving the register address, the slave returns ACK and stores the register address value.

Then, received data is written in sequence from the specified register address (sequential write). Upon receiving a stop condition (SP), communication is ended.

Upon receiving the address and data, the slave returns ACK. If the address or data cannot be confirmed, the SDA maintains a high level state (NACK response).

Note: In the RL78 family, if the local address and the slave address match, an ACK response is sent automatically.

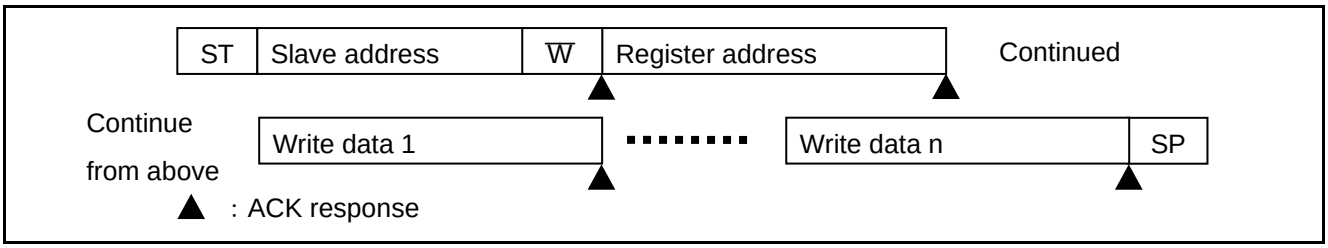


Figure 1.11 Sequence for Continuous Data Writing by Register Address Specification

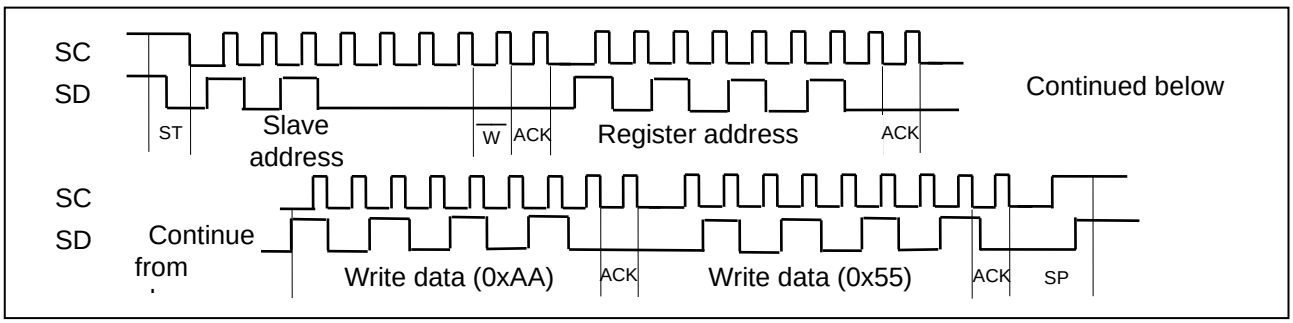


Figure 1.12 Timing Chart for Continuous Data Writing by Register Address Specification

The timing chart for data writing when writing is forbidden is shown in Figure 1.13. The slave returns ACK in response to the slave address and a register address, but returns NACK in response to write data, and disengages from communication. The master confirms the NACK response and generates a stop condition, ending communication.

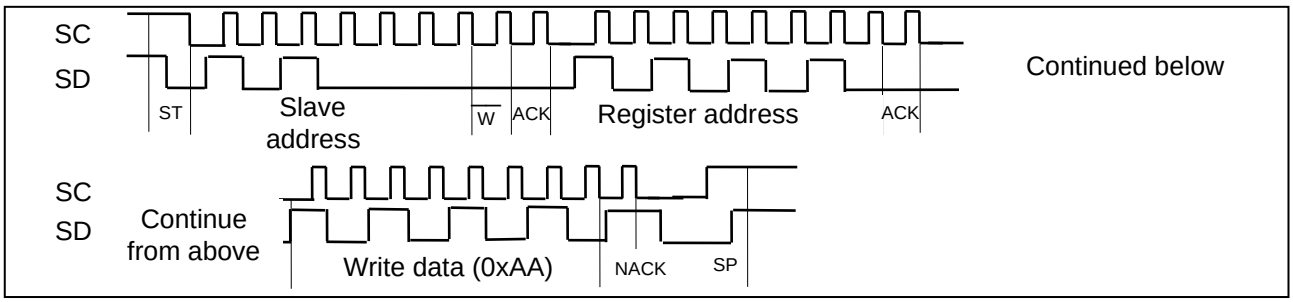


Figure 1.13 Timing Chart for Data Writing When Writing is Forbidden

1.2.3 Continuous Data Reading

Regarding cases in which a register address is specified and data is read continuously from a slave, the sequence is shown in Figure 1.14, and the timing chart appears in Figure 1.15.

Upon receiving the slave address 0b1010000 following a start condition (ST), the slave returns ACK[note] after receiving a transfer direction specification bit. When the transfer direction $\overline{W}$ is specified, it is determined that the second-next received data will be written to the address specified by the next received data. Upon receiving the register address, the slave returns ACK and stores the register address value.

However, before receiving data scheduled for writing, if after a restart condition (ST) the slave address 0b1010000 and the transfer direction R are received, the slave reads data in sequence from the previously specified register address and transmits the data to the master (sequential read).

Upon detecting a NACK from the master, the slave halts data transmission. With a stop condition (SP) from the master, communication is completed.

Upon receiving the address and data, the slave returns ACK. If the address or data cannot be confirmed, the SDA maintains a high level state (NACK response).

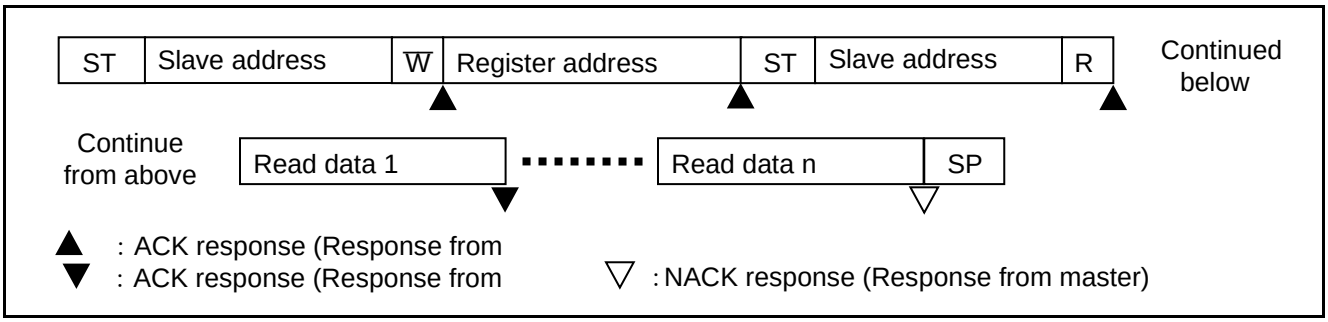


Figure 1.14 Sequence for Continuous Data Reading by Register Address Specification

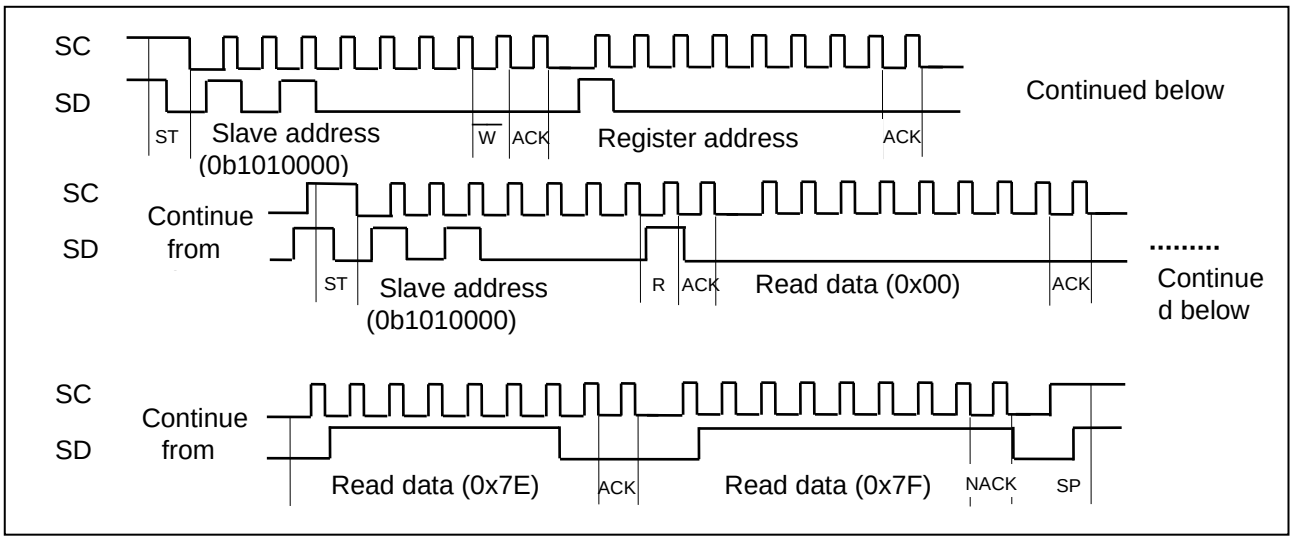


Figure 1.15 Timing Chart for Continuous Data Reading by Register Address Specification

2. Operation Check Conditions

The sample code contained in this application note has been checked under the conditions listed in the Table 2.1 below.

Table 2.1 Operation Check Conditions

Item	Description
Microcontroller used	RL78/G13 (R5F100LEA)
Operating frequency	• High-speed on-chip oscillator (HOCO) clock: 32 MHz • CPU/peripheral hardware clock: 32 MHz
Operating voltage	5.0 V (Operation is possible over a voltage range of 2.9 V to 5.5 V.) LVD operation (V_{LVD}): Reset mode which uses 2.81 V (2.76 V to 2.87 V)
Integrated development environment (CS+)	Renesas Electronics CS + for CC V8.01.00
C compiler (CS+)	Renesas Electronics CC-RL V1.08.00
Integrated development environment (e^2 studio)	Renesas Electronics e^2 studio V7.3.0
C compiler (e^2 studio)	Renesas Electronics CC-RL V1.08.00
Integrated development environment (IAR)	IAR systems IAR Embedded Workbench for Renesas RL78 V4.20.2
C compiler (IAR)	IAR systems IAR Embedded Workbench for Renesas RL78 V4.20.2.6370

3. Related Application Note

The application notes that are related to this application note are listed below for reference.

- RL78/G13 Initialization (R01AN2575E) Application Note
- RL78/G13 Serial Interface IICA (for Master Transmission/Reception) (R01AN2759E) Application Note

4. Description of the Hardware

4.1 Hardware Configuration Example

Figure 4.1 shows an example of hardware configuration that is used for this application note.

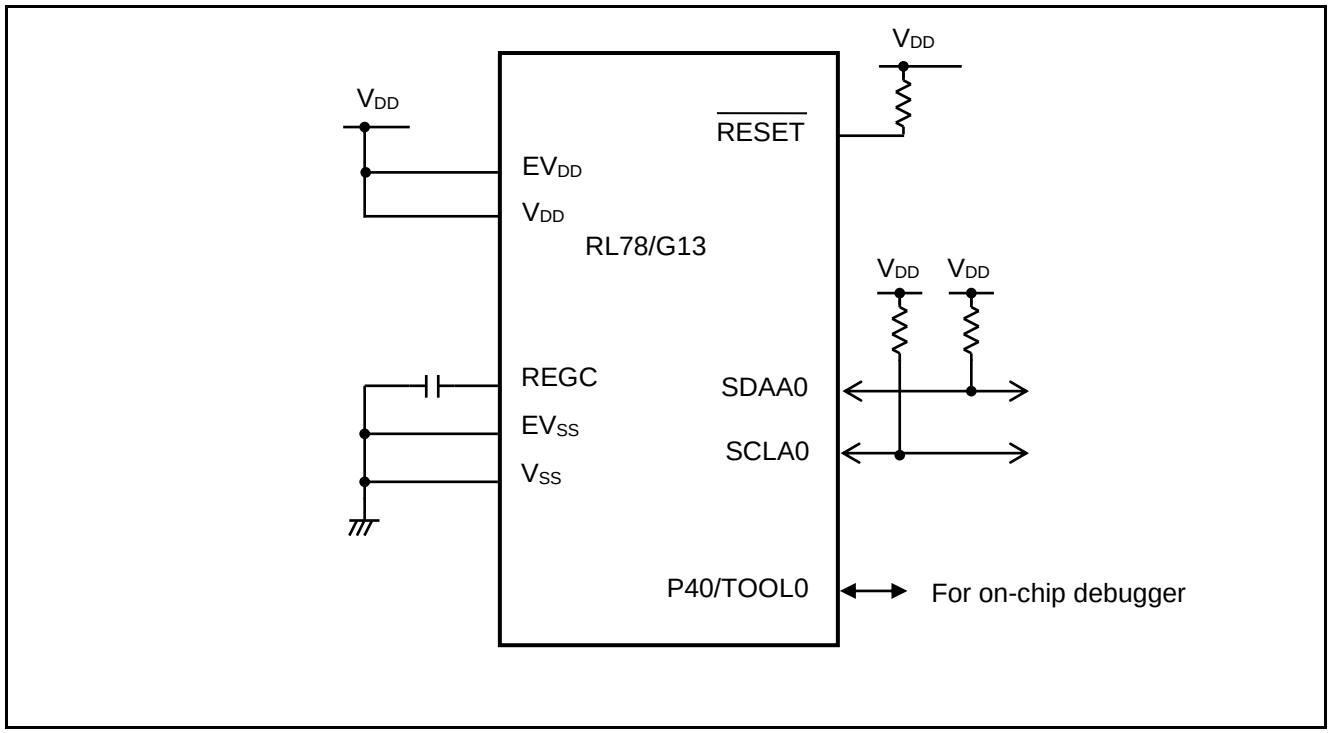


Figure 4.1 Hardware Configuration

Cautions:
1. The purpose of this circuit is only to provide the connection outline and the circuit is simplified accordingly. When designing and implementing an actual circuit, provide proper pin treatment and make sure that the hardware's electrical specifications are met (connect the input-only ports separately to V_{DD} or V_{SS} via a resistor).
2. Connect any pins whose name begins with EV_{SS} to V_{SS} and any pins whose name begins with EV_{DD} to V_{DD}, respectively.
3. V_{DD} must be held at not lower than the reset release voltage (V_{LVD}) that is specified as LVD.

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
P60/SCLA0	Input/Output	IICA0 serial clock I/O pin
P61/SDAA0	Input/Output	IICA0 serial data transmission/reception pin

5. Description of the Software

5.1 Operation Outline

The sample program covered in this application note provides IICA slave transmission and reception (address reception, and data transmission and reception) through the serial interface IICA.

(1) Initialize serial interface IICA.

<Conditions for setting>

- Select the fast mode as the operation mode.
- Set the transfer clock frequency to 400 kHz.
- To set the slave address to 0b1010000, set 0xA0 to slave address register 0 (SVA0).
- Turn the digital filter on
- Generate an interrupt in response to the ninth clock signal
- Disable stop condition interrupts.
- Use the P60/SCLA0 pin for transfer clock input and the P61/SDAA0 pin for data transmission/reception.

(2) Get the communication buffer (16 bytes) ready for use.

(3) The IICA wakeup function is used to reduce power consumption. A STOP instruction is executed, and an interrupt (INTIICA0) occurring when the local address or an extension code is received is awaited.

(4) When the local address or an extension code is received, wakeup occurs, and data communication is begun. When an extension code has been received, the slave exits from communication and returns to (3).

(5) Register address reception

1. When the received register address is 0x10 to 0x7F, the slave exits from communication. (NACK response)
2. When the register address is 0x00, ACK is returned. The next data (command) is analyzed, and ACK/NACK is returned. Moreover, processing specified by a command is performed.
3. When the register address is 0x80 to 0xFF, the register address is stored. Thereafter, an instruction from the master is awaited.

(6) Data transmission/reception

1. When the transfer direction is R, data is read in sequence from the specified register address, and data that has been read is transmitted from the slave to the master (sequential read) until a NACK response from the master is detected.
2. When the transfer direction is $\overline{W}$, received data is written in sequence (sequential write) from the specified register address (serial RAM region).

(7) When a stop condition from the master is detected, the slave exits from communication.

(8) When a start condition from the master is detected, processing is repeated from (4).

(9) The above processing (3) to (7) is repeated.

Caution This sample code is related to the RL78/G13 IICA Master Transmission/Reception (R01AN2759J) Application Note only. This sample code performs actions conforming to instructions from the master rather than actions determined in advance. The conditions for completion of communication are detection of a stop condition, and NACK detection during data transmission. When communication is completed, the next start of communication from the master is awaited. During serial RAM initialization and other communication processing, wait states (SCLA0 pin at low level) are used to synchronize with the master.

Remark Higher-level main processing is performed while checking the value of the variable g_iica0_slave_status_flag indicating the processing state of the IICA0 interrupt.

5.2 List of Option Byte Settings

Table 5.1 summarizes the settings of the option bytes.

Table 5.1 Option Byte Settings

Address	Value	Description
000C0H/010C0H	01101110B	Disables the watchdog timer. (Stops counting after the release from the reset state.)
000C1H/010C1H	01111111B	LVD reset mode, 2.81 V (2.76 V to 2.87 V)
000C2H/010C2H	11101000B	HS mode, HOCO: 32 MHz
000C3H/010C3H	10000101B	Enables the on-chip debugger.

5.3 List of Constants

Table 5.2 lists the constants that are used in this sample program.

Table 5.2 Constants for the Sample Program

Constant	Setting	Description
DATA_LENGTH	0x10	IIC transmit/receive data length
REGADDR	1	Number of bytes of register address
ON_COMM	0x00	In communication
DET_STOP	0x01	Stop condition detected
TX_END	0x02	NACK detected (data transmission ends)
TD_REQ	0x04	Data transmission request
RX_END	0x08	Reception of specified number of data items completed
DR_REQ	0x10	Received data read request
AR_END	0x20	Register address (reception) completed
NOT_SEL	0x80	After restart, address mismatch/extension code received

5.4 List of Variables

Table 5.3 lists the global variables that are used in this sample program.

Table 5.3 Global Variables for the Sample Program

Type	Variable Name	Contents	Function Used
uint8_t	g_iica0_slave_status_flag	IICA0 slave flag	R_IICA0_Slave_Receive(), iica0_slave_handler(),main()
uint8_t *	gp_iica0_rx_address	IICA0 receive buffer address	R_IICA0_Slave_Receive(), iica0_slave_handler()
uint16_t	g_iica0_rx_len	IICA0 receive data length	R_IICA0_Slave_Receive(), iica0_slave_handler()
uint16_t	g_iica0_rx_cnt	IICA0 receive data count	R_IICA0_Slave_Receive(), iica0_slave_handler()
uint8_t *	g_iica0_tx_address	IICA0 transmit buffer address	R_IICA0_Slave_Send(), iica0_slave_handler()
uint16_t	g_iica0_tx_cnt	IICA0 transmit data count	R_IICA0_Slave_Send(), iica0_slave_handler()
static uint8_t	g_rx_data[DATA_LENGTH]	Data receive buffer	R_IICA0_Slave_Receive()
static uint8_t	g_serial_RAM[128]	Serial RAM area	main(), R_Command()
static uint8_t	g_com_stat	Command status	main(), R_Command()
uint16_t	g_reg_addr	Internal address (Address of serial ram area)	main(),R_Increment_Address (), iica0_slave_handler ()

5.5 Variable Specifications

Table 5.4 lists the Variable Specifications that are used in this sample program.

Table 5.4 Variable Specifications

[Variable name] g_iica0_slave_status_flag

Summary	IICA0 slave flag
Explanation	Indicates the slave operation state

Bit	Meaning
7	1: Received other than slave address 0b1010000 (address mismatch after restart) 0: Received slave address 0b1010000
6	Fixed to 0
5	1: Register address set (includes update of same value) 0: Register address not set
4	1: Received data exceeding specified number of times 0: Did not receive data exceeding specified number of times
3	1: Received data the specified number of times (also used for register address setting) 0: Did not receive data the specified number of times
2	1: Data transmission request from master 0: No data transmission request from master
1	1: NACK detected (master has completed reception of last data) 0: NACK not detected
0	1: Stop condition detected 0: Stop condition not detected

5.6 List of Functions

Table 5.5 summarizes the functions that are used in this sample program.

Table 5.5 Functions

Function Name	Outline
R_Increment_Address	Internal address update processing
R_Command	Command execution processing
R_IICA0_Slave_Receive	Slave reception setting
r_iica0_interrupt	IICA0 interrupt processing
iica0_slave_handler	Slave processing within IICA 0 interrupt

5.7 Function Specifications

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

[Function Name] R_Increment_Address	
Synopsis	Internal address update processing
Header	-
Declaration	void R_Increment_Address(void)
Explanation	Updates the serial RAM address. After 0xFF returns to 0x80.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_Command	
Synopsis	Command execution processing
Header	-
Declaration	void R_Command(void)
Explanation	Performs processing corresponding to the command written to the command register.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_IICA0_Slave_Receive		
Synopsis	Slave reception setting	
Header	r_cg_serial.h	
Declaration	void R_IICA0_Slave_Receive(uint8_t * const rx_buf and uint16_t rx_num)	
Explanation	Specifies the slave reception mode.	
Arguments	rx_buf	Receive data buffer address
	rx_num	Receive data length
Return value	None	
Remarks	None	

[Function Name] r_iica0_interrupt	
Synopsis	IICA0 interrupt processing
Header	-
Declaration	static void __near r_iica0_interrupt(void)
Explanation	Interrupt handler for IICA0 interrupt.
Arguments	None
Return value	None
Remarks	In this sample code, in order to meet the specifications, processing to stop the wakeup function operation is added to the beginning of this function, which is generated by the code generator. Even when code is generated, a setting for code generation is used so as to output only an initialization function so as not to exert an influence; interrupt processing is provided in the "r_serial_user.c" file.

[Function Name] iica0_slave_handler	
Synopsis	Slave processing within IICA0 interrupt
Header	r_cg_serial.h
Declaration	static void iica0_slave_handler(void)
Explanation	Performs slave transmission and reception during IICA0 interrupt handling.
Arguments	None
Return value	None
Remarks	In this sample code, code generated by the code generator is not used.

5.8 Flowcharts

Figure 5.1 the overall flow of the sample program described in this application note.

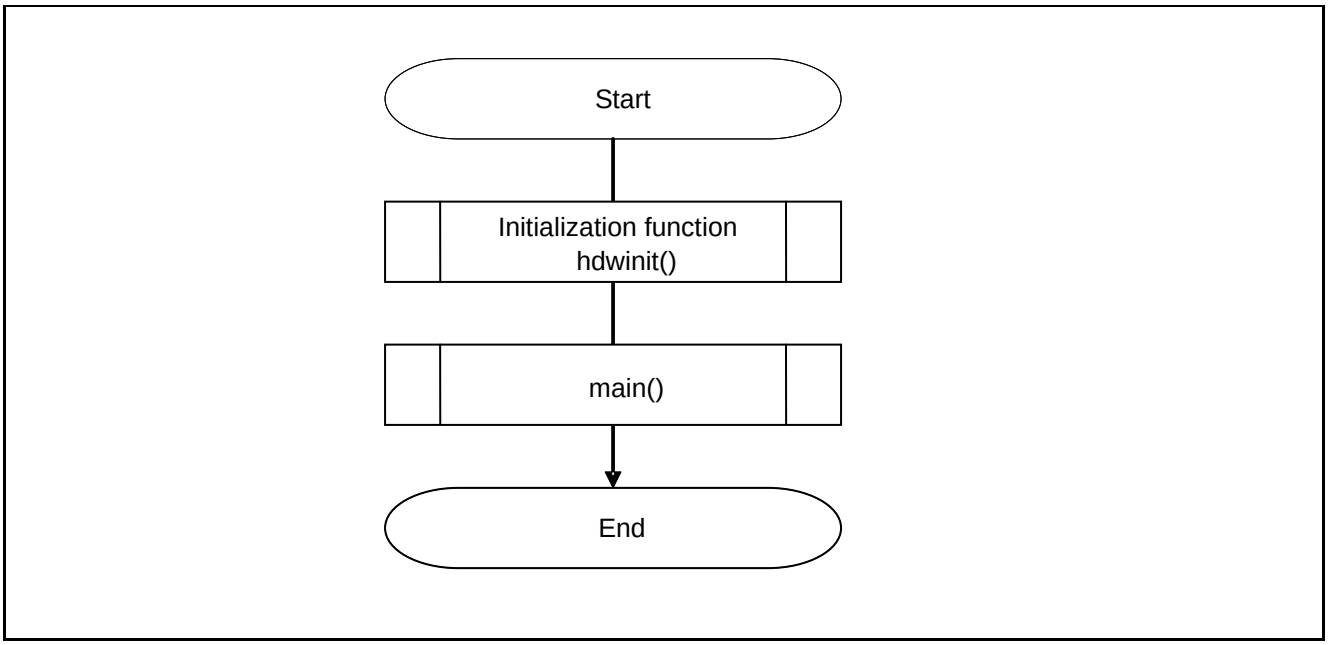


Figure 5.1 Overall Flow

5.8.1 Initialization Function

Figure 5.2 shows the flowchart for the initialization function.

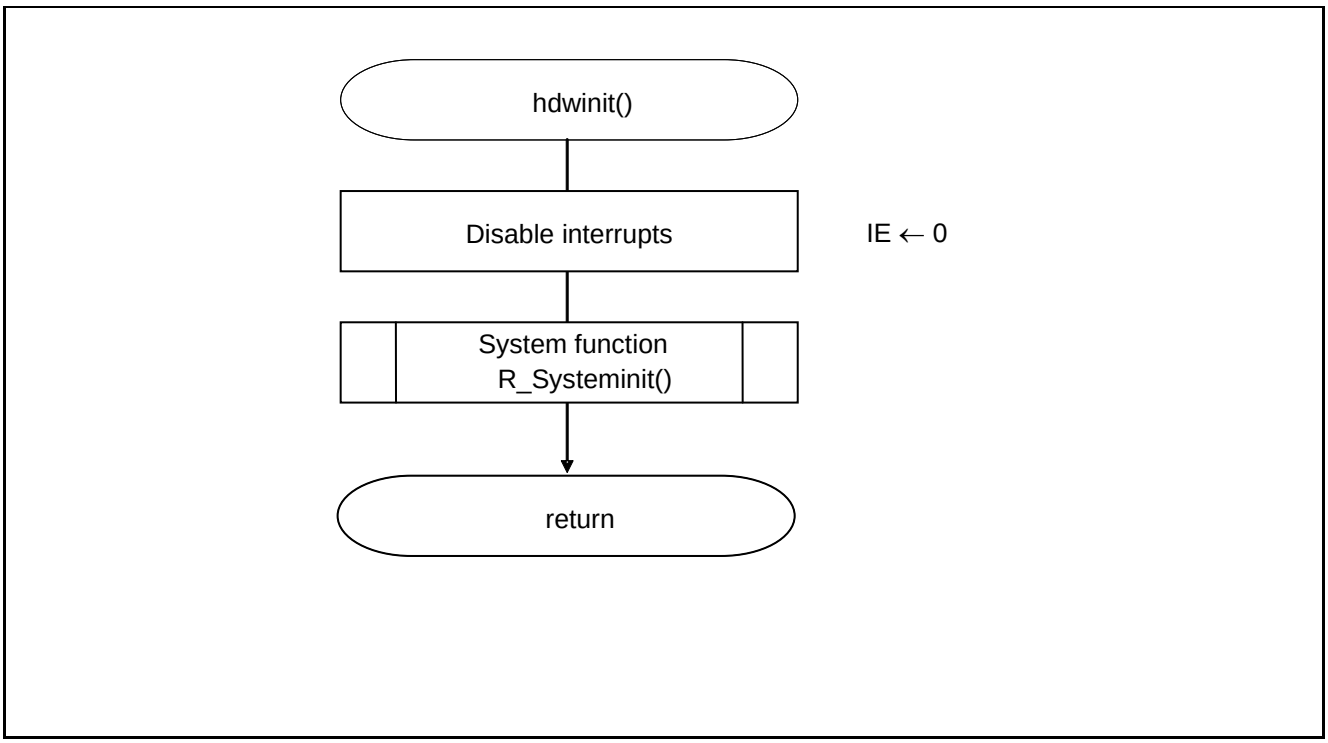


Figure 5.2 Initialization Function

5.8.2 System Function

Figure 5.3 shows the flowchart for the system function.

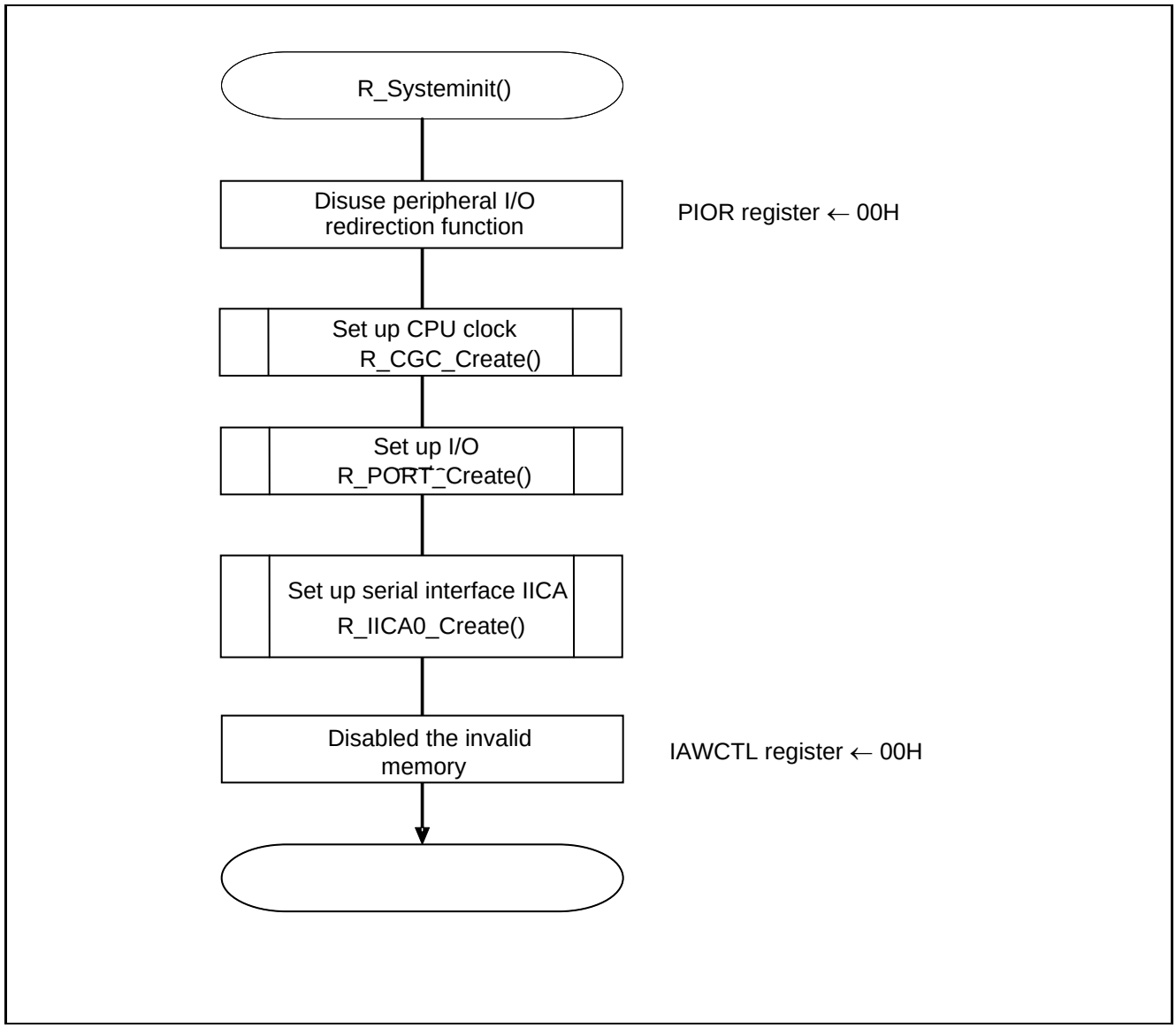


Figure 5.3 System Function

5.8.3 CPU Clock Setup

Figure 5.4 shows the flowchart for the CPU clock setup.

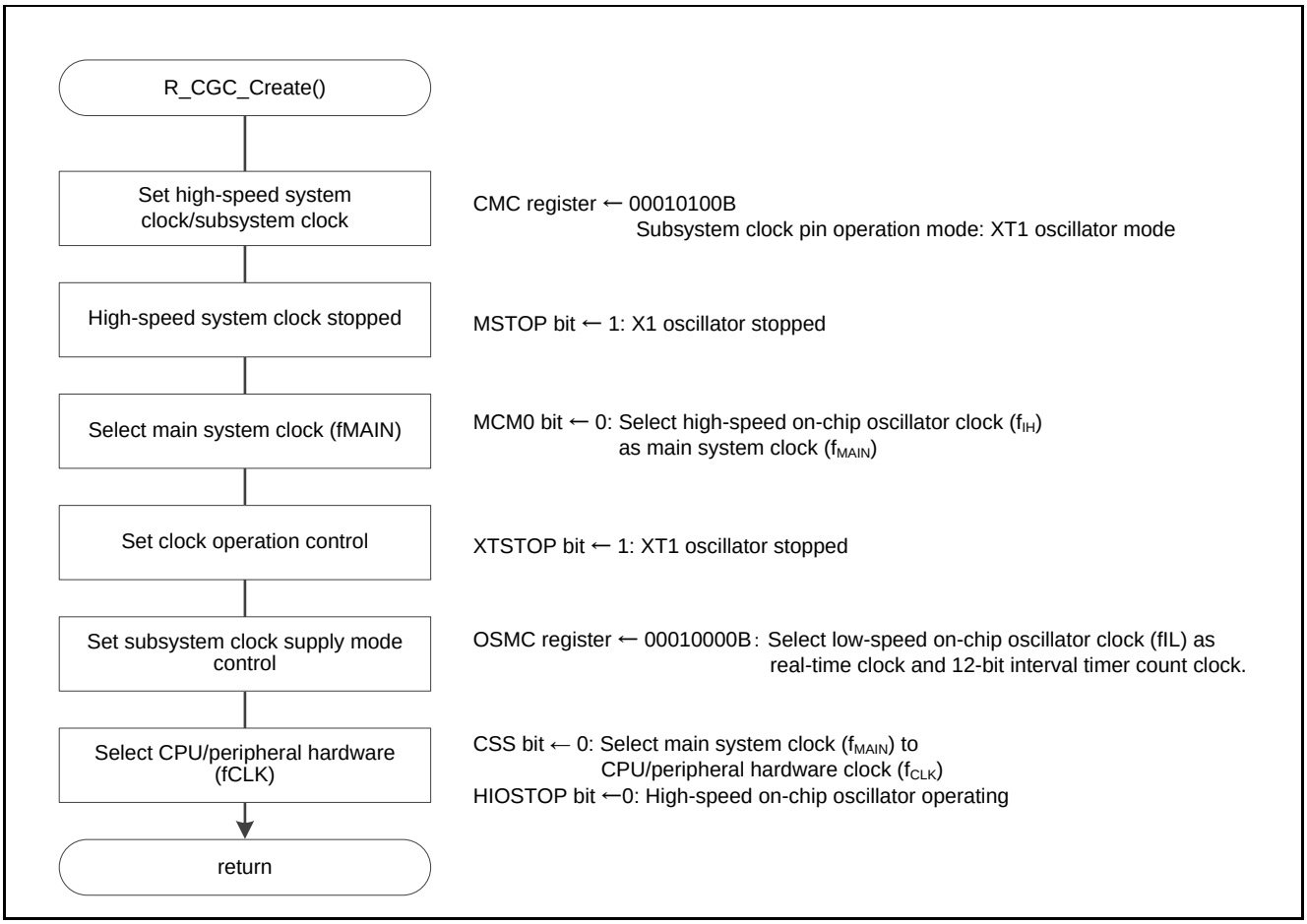


Figure 5.4 CPU Clock Setup

Caution: For details on the procedure for setting up the CPU clock (R_CGC_Create ()), refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN2575E).

5.8.4 I/O Port Setup

Figure 5.5 shows the flowchart for the I/O port setup.

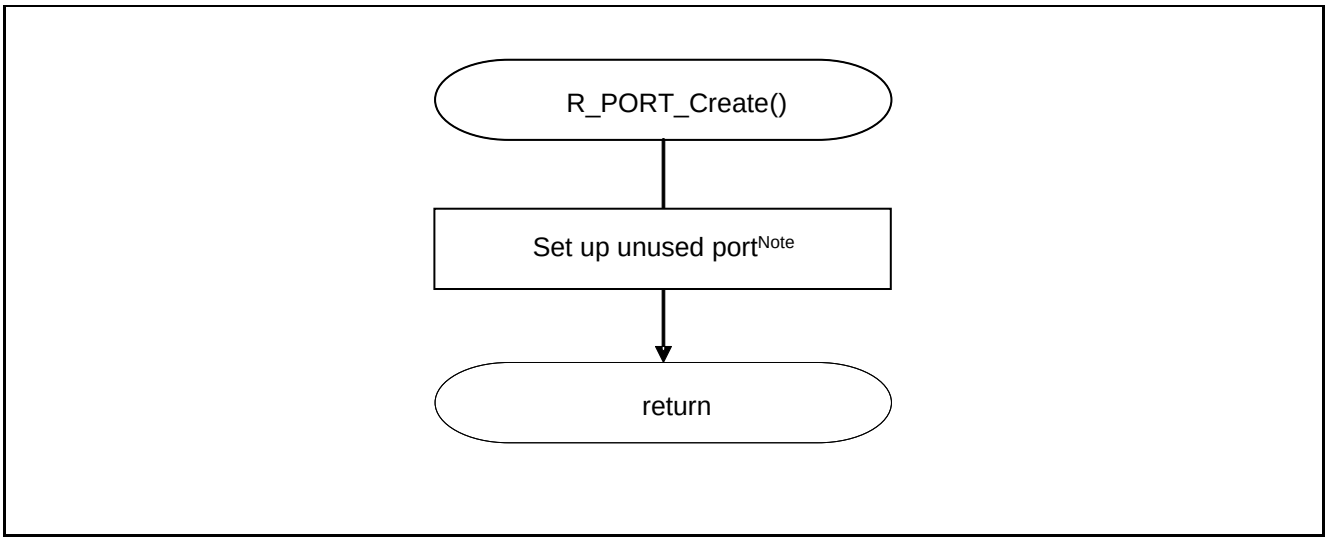


Figure 5.5 I/O Port Setup

Note: Refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN2575E) for the configuration of the unused ports.

Caution: Provide proper treatment for unused pins so that their electrical specifications are observed. Connect each of any unused input-only ports to V_{DD} or V_{SS} via separate resistors.

5.8.5 Serial Interface IICA Setup

Figure 5.6 shows the flowchart for the serial interface IICA setup.

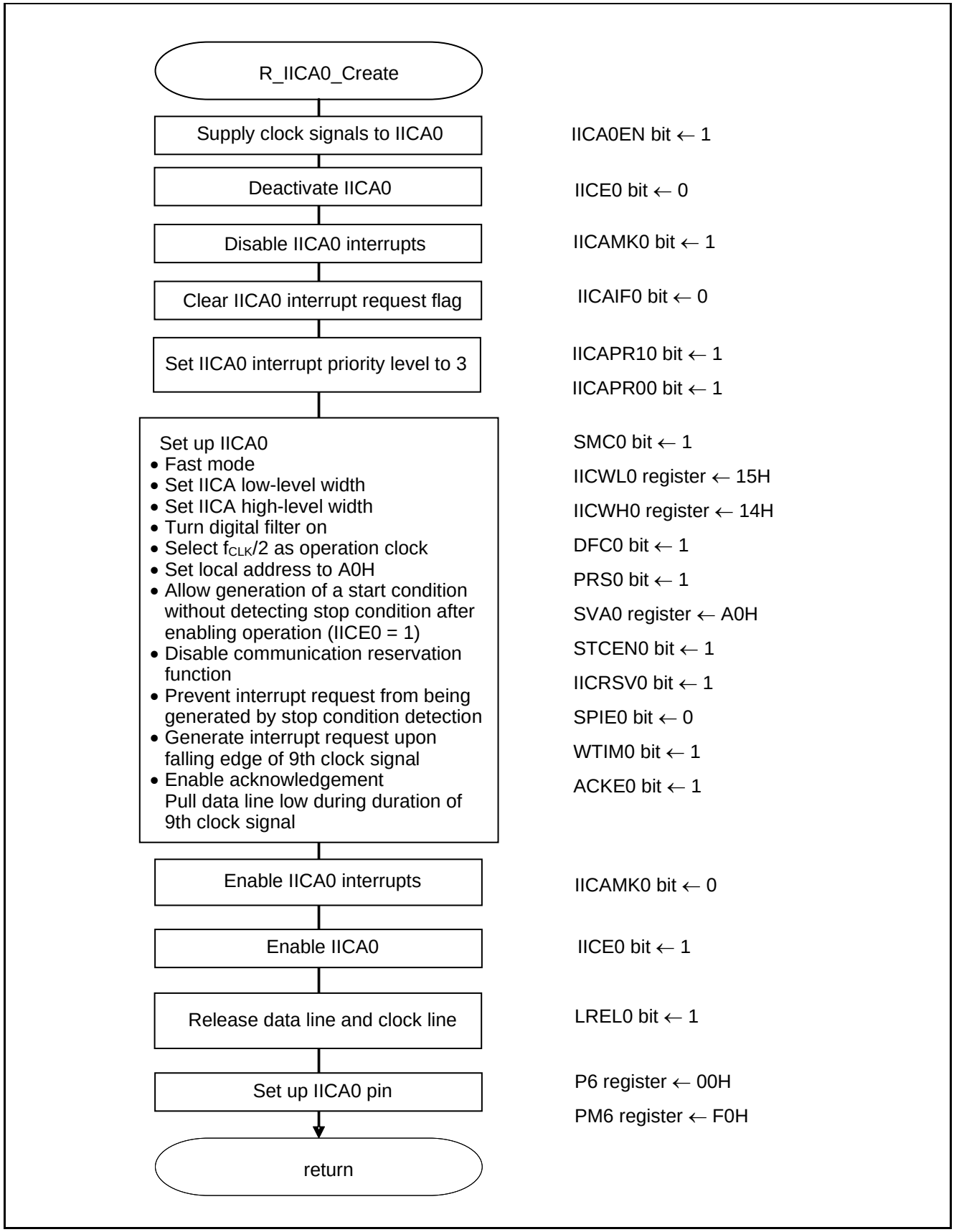


Figure 5.6 Serial Interface IICA Setup

Starting clock signal supply to serial interface IICA0

- Peripheral enable register 0 (PER0)
  Start supplying clock signals to IICA0 by using IICAEN.

Symbol: PER0

7	6	5	4	3	2	1	0
RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
x	x	x	**1**	x	x	x	x

Bit 4

IICA0EN	Serial interface IICA0 input clock control
0	Stops supply of input clock.
1	**Enables supply of input clock.**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Setting up the IICA0 operation mode

- IICA control register 01 (IICCTL01)
  Select an operation clock frequency.
  Turn the digital filter on.
  Select the fast mode.
  Disable the wakeup function.

Symbol: IICCTL01

7	6	5	4	3	2	1	0
WUP0	0	CLD0	DAD0	SMC0	DFC0	0	PRS0
0	0	x	x	**1**	**1**	0	**1**

Bit 7

WUP0	Address match wakeup control
0	**Disable the address match wakeup function in STOP mode.**
1	Enable the address match wakeup function in STOP mode.

Bit 3

SMC0	Operation mode selection
0	Standard mode
1	**Fast mode**

Bit 2

DFC0	Digital filter operation control
0	Turns the digital filter off.
1	**Turns the digital filter on.**

Bit 0

PRS0	Operation clock frequency selection
0	Selects f_{CLK} as the operation clock frequency.
1	**Selects f_{CLK}/2 as the operation clock frequency.**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Configuring the transfer clock

- IICA low-level width setting register 0 (IICWL0)
- IICA high-level width setting register 0 (IICWH0)
  Set the low-level width and high-level width of the SCLA0 pin signal.

Symbol: IICWL0

7	6	5	4	3	2	1	0
0	**0**	**0**	**1**	**0**	**1**	**0**	**1**

Symbol: IICWH0

7	6	5	4	3	2	1	0
0	**0**	**0**	**1**	**0**	**1**	**0**	**0**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Setting the local address

- Slave address register 0 (SVA0)
  Set the local address.

Symbol: SVA0

7	6	5	4	3	2	1	0
1	**0**	**1**	**0**	**0**	**0**	**0**	**0**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Setting up the IICA operation

- IICA control register 00 (IICCTL00)
  Enable I^2C operation.
  Disable stop condition interrupts.
  Set the wait and interrupt request generation timing.
  Enable acknowledgement output.

Symbol: IICCTL00

7	6	5	4	3	2	1	0
IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0
1	**1**	x	**0**	**1**	**1**	x	x

Bit 7

IICE0	Enabling/disabling of I^2C operation
0	Stops operation. Resets the IICA status register 0 (IICS0). Also stops internal operation.
1	**Enables operation.**

Bit 6

LREL0	Transition from the communication state
0	Normal operation
1	**Makes a transition from the current communication state to the standby state. Automatically cleared to 0 after the transition.**

Bit 4

SPIE0	Enabling/disabling generation of interrupt requests due to stop condition detection
0	**Disabled**
1	Enabled

Bit 3

WTIM0	Wait/interrupt request control
0	Interrupt request is generated at the falling edge of the eighth clock signal. Waits with the clock output remaining at low level, after eight clock signals are output.
1	**Interrupt request is generated at the falling edge of the ninth clock signal.** **Waits with the clock output remaining at low level, after nine clock signals are output.**

Bit 2

ACKE0	Acknowledgement control
0	Disables acknowledgements.
1	**Enables acknowledgements.**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Setting up the IICA pins

- Port register 6 (P6)
- Port mode register 6 (PM6)
  Use P60 for SCLA0 and P61 for SDAA0 in output mode.

Symbol: P6

7	6	5	4	3	2	1	0
P67	P66	P65	P64	P63	P62	P61	P60
x	x	x	x	x	x	**0**	**0**

Bit 1

P61	Output data control
0	**Output 0**
1	Output 1

Bit 0

P60	Output data control
0	**Output 0**
1	Output 1

Symbol: PM6

7	6	5	4	3	2	1	0
PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60
x	x	x	x	x	x	**0**	**0**

Bit 1

PM61	P61 I/O mode selection
0	**Output mode (output buffer on)**
1	Input mode (output buffer off)

Bit 0

PM60	P60 I/O mode selection
0	**Output mode (output buffer on)**
1	Input mode (output buffer off)

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

5.8.6 Main Processing

Figure 5.7 through Figure 5.8 shows the flowchart for the main processing.

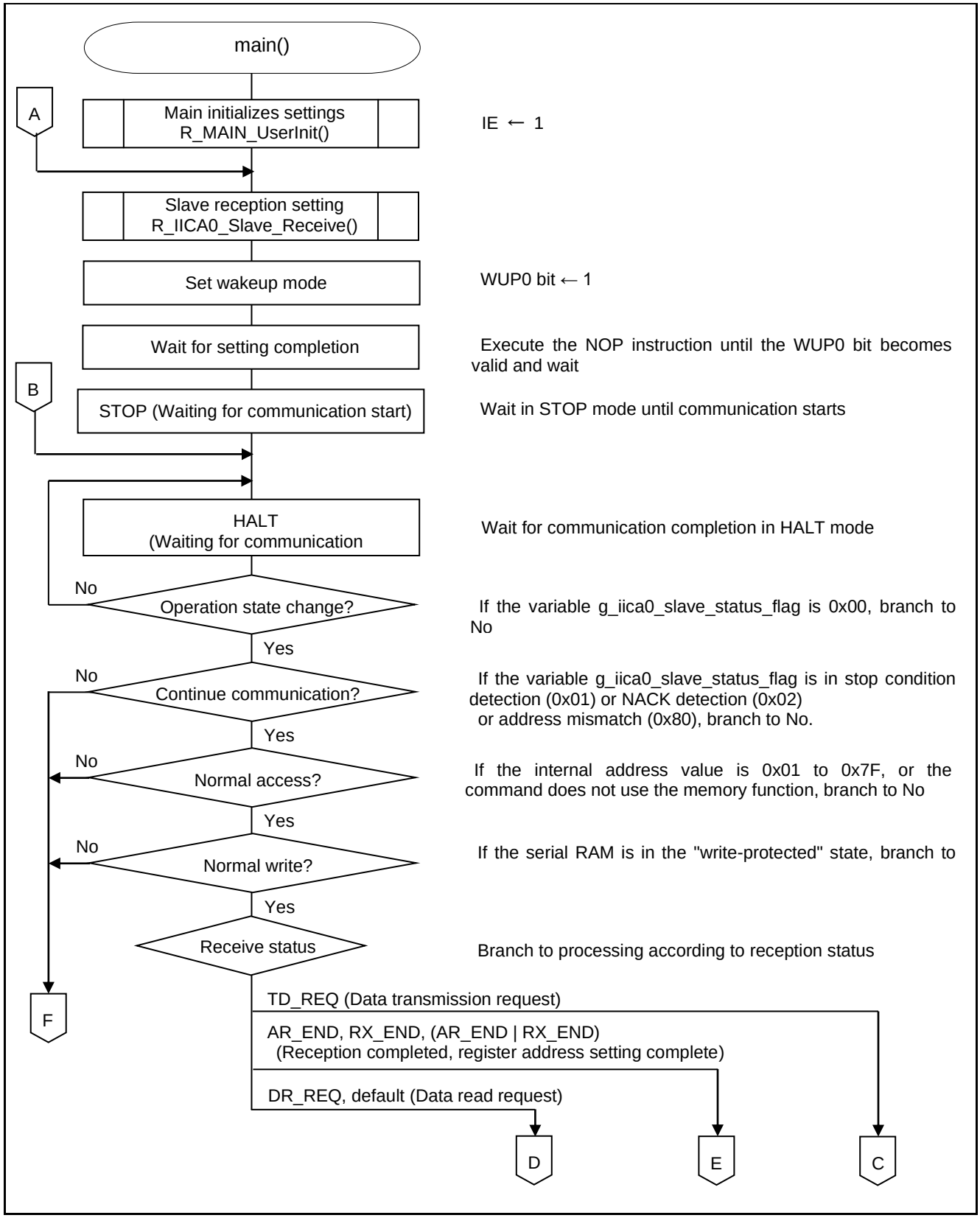


Figure 5.7 Main Processing (1/2)

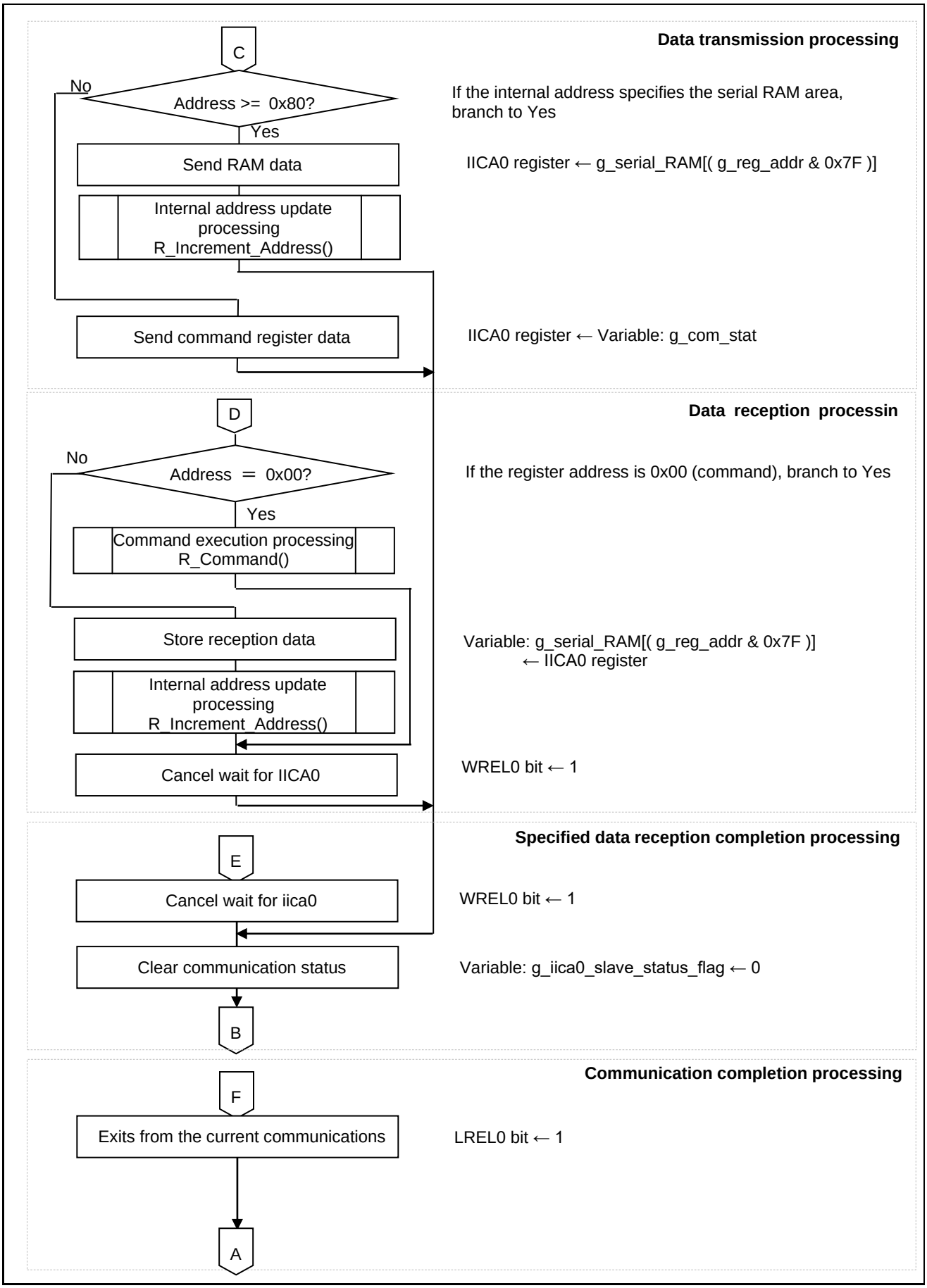


Figure 5.8 Main Processing (2/2)

Setting up the wakeup function

- IICA control register 01 (IICTL01)
  Enable the wakeup function.

Symbol: IICCTL01

7	6	5	4	3	2	1	0
WUP0	0	CLD0	DAD0	SMC0	DFC0	0	PRS0
1	0	x	x	1	1	0	1

Bit 7

WUP0	Address match wakeup control
0	Disables the address match wakeup function in STOP mode
1	**Enables the address match wakeup function in STOP mode**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Communication termination setting

- IICA control register 00 (IICCTL00)
  Terminate the current communication, and make the system enter a communication standby

Symbol: IICCTL00

7	6	5	4	3	2	1	0
IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0
1	**1**	x	0/1	0/1	0/1	0	0

Bit 6

LREL0	Transition from the communication state
0	Normal operation
1	**Makes a transition from the current communication state to the standby state**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

IIC bus wait cancellation setting

・IICA Control Register00（IICCTL00）
End the current wait state and set the communication state

Symbol: IICCTL00

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0
1	0	**1**	0/1	0/1	0/1	0	0

Bit 5

WREL0	**Wait cancellation**
0	Do not cancel wait
1	**Cancel wait. This setting is automatically cleared after wait is canceled.**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

5.8.7 Main Initial Setting

Figure 5.9 show the flowchart for the main initial setting.

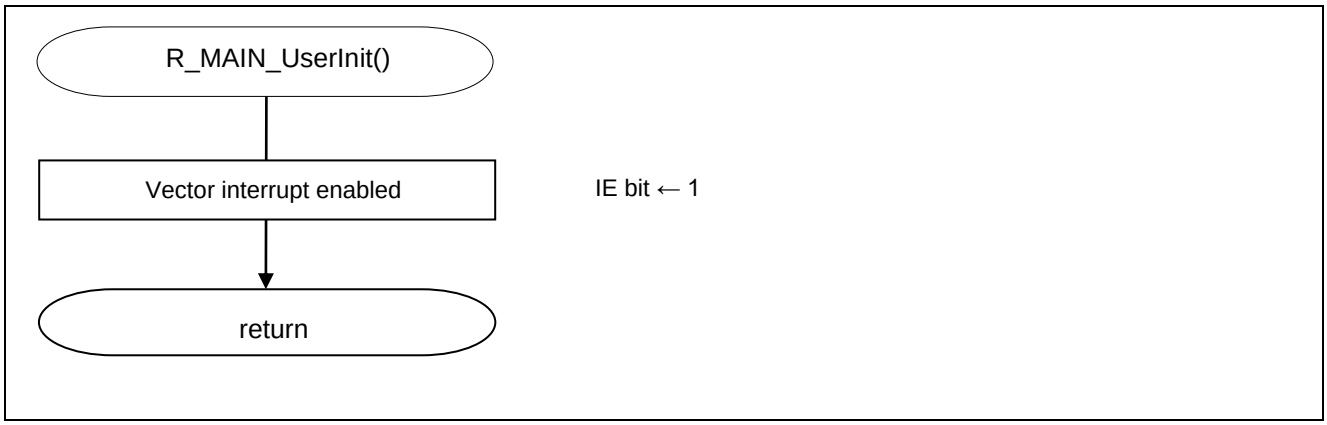


Figure 5.9 Main Initial Setting

5.8.8 Internal Address Update Processing

Figure 5.10 shows the flowchart for the internal address update processing.

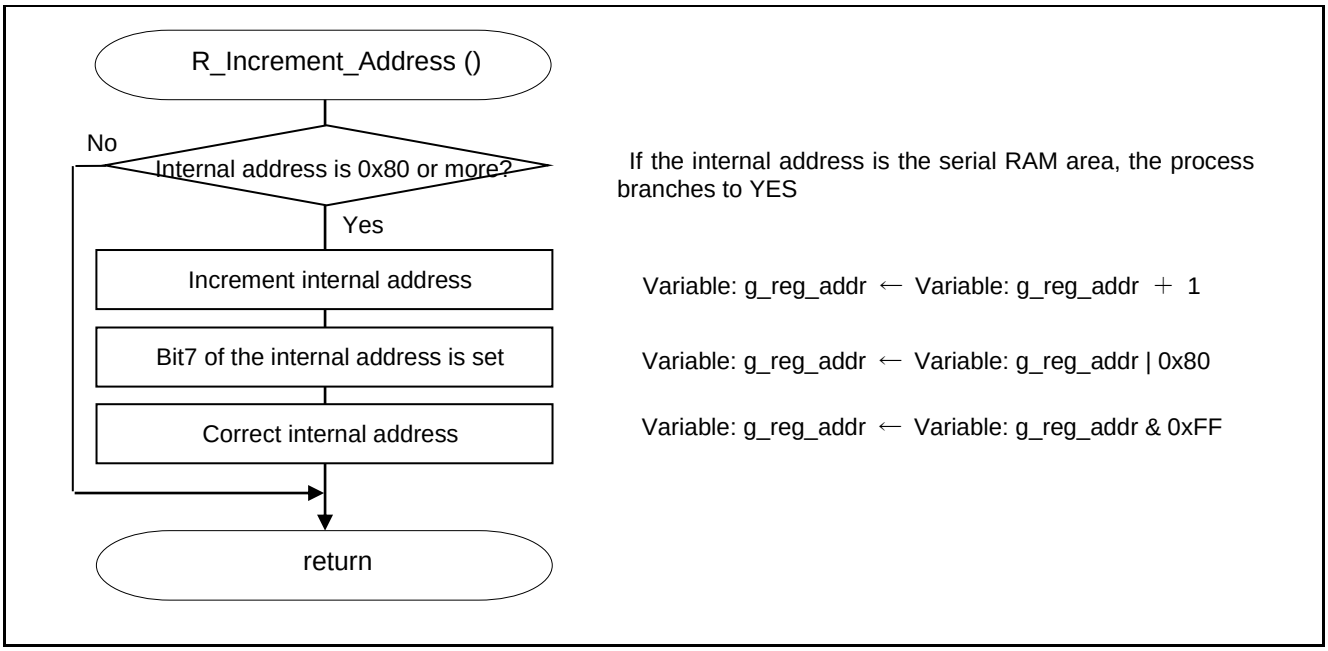


Figure 5.10 Internal Address Update Processing

5.8.9 Command Execution Processing

Figure 5.11 shows the flowchart for the command execution processing.

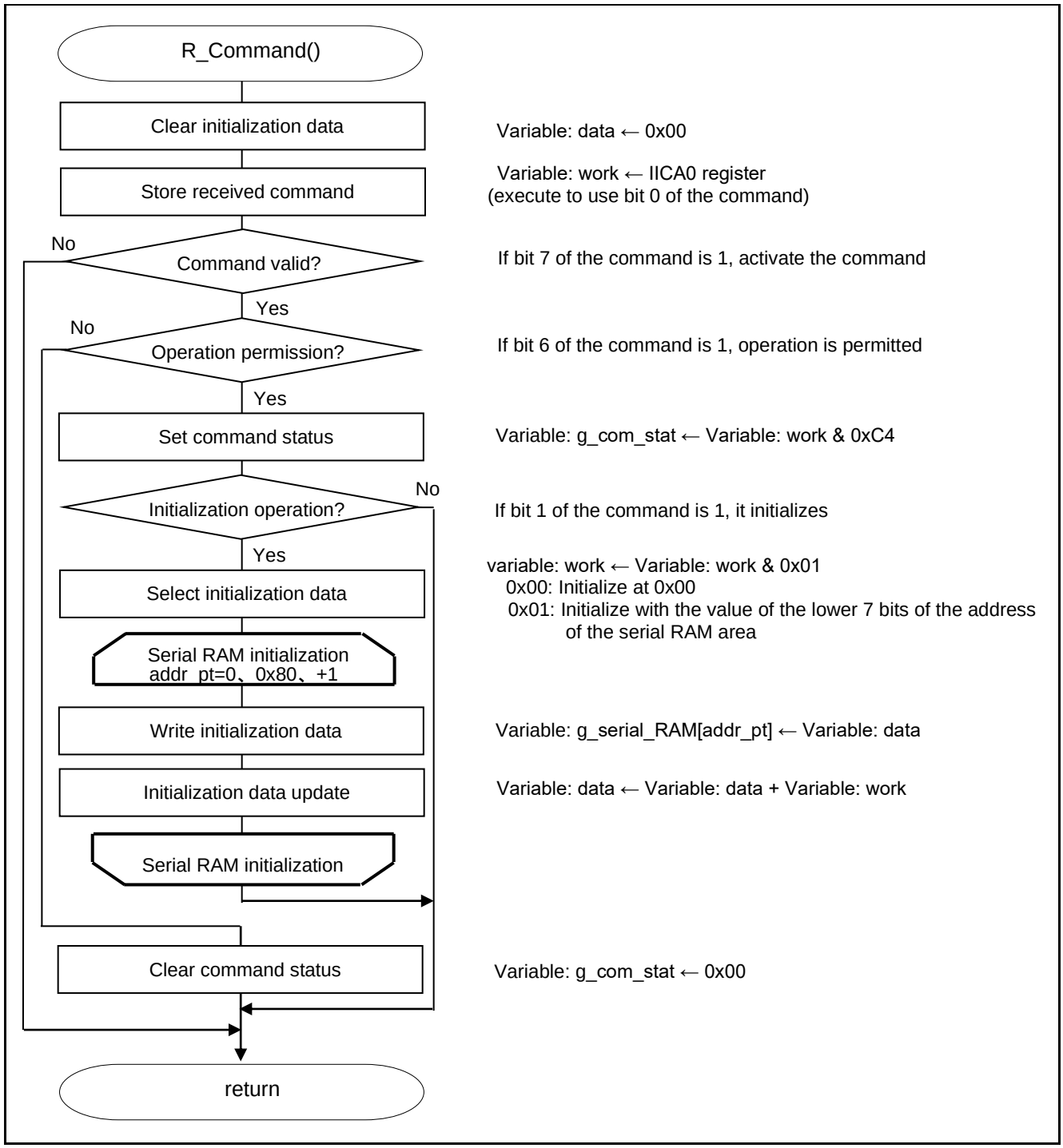


Figure 5.11 Command Execution Processing

5.8.10 Slave Reception Setting

Figure 5.12 shows the flowchart for the slave reception setting.

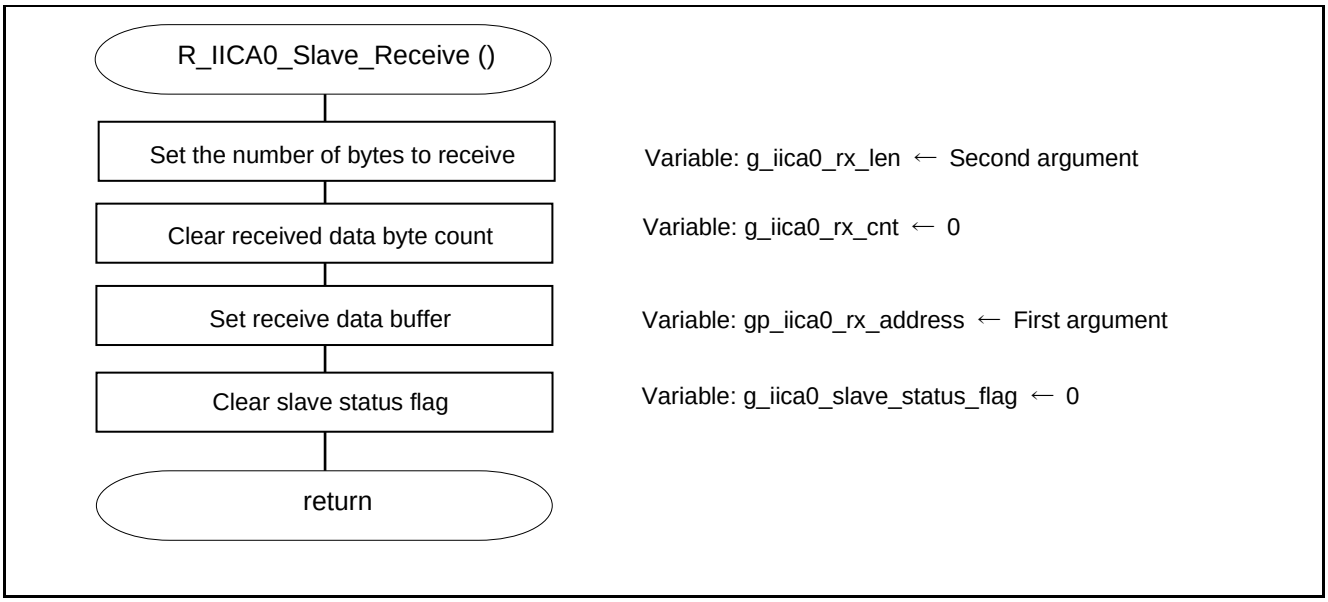


Figure 5.12 Slave Reception Setting

5.8.11 IICA0 Interrupt Processing

Figure 5.13 shows the flowchart for IICA0 interrupt processing.

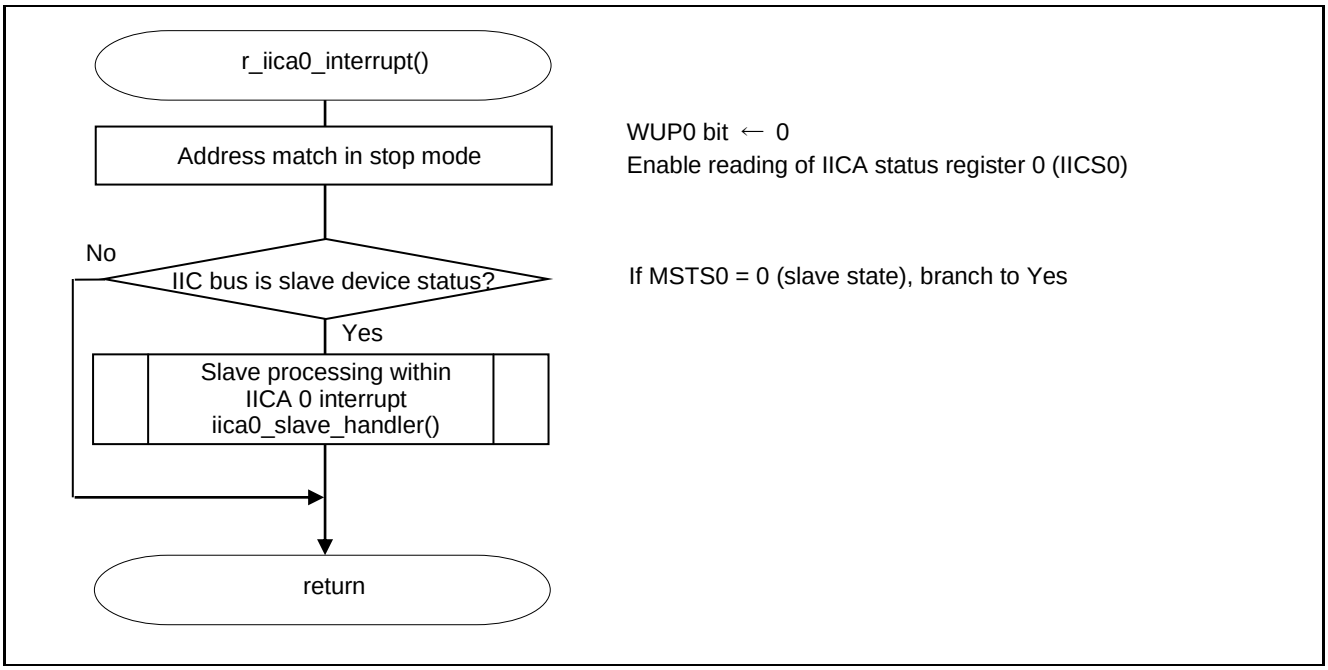


Figure 5.13 IICA0 Interrupt Processing

5.8.12 IICA0 Slave Handler

Figure 5.14 through Figure 5.16 show the flowcharts for the IICA0 slave handler.

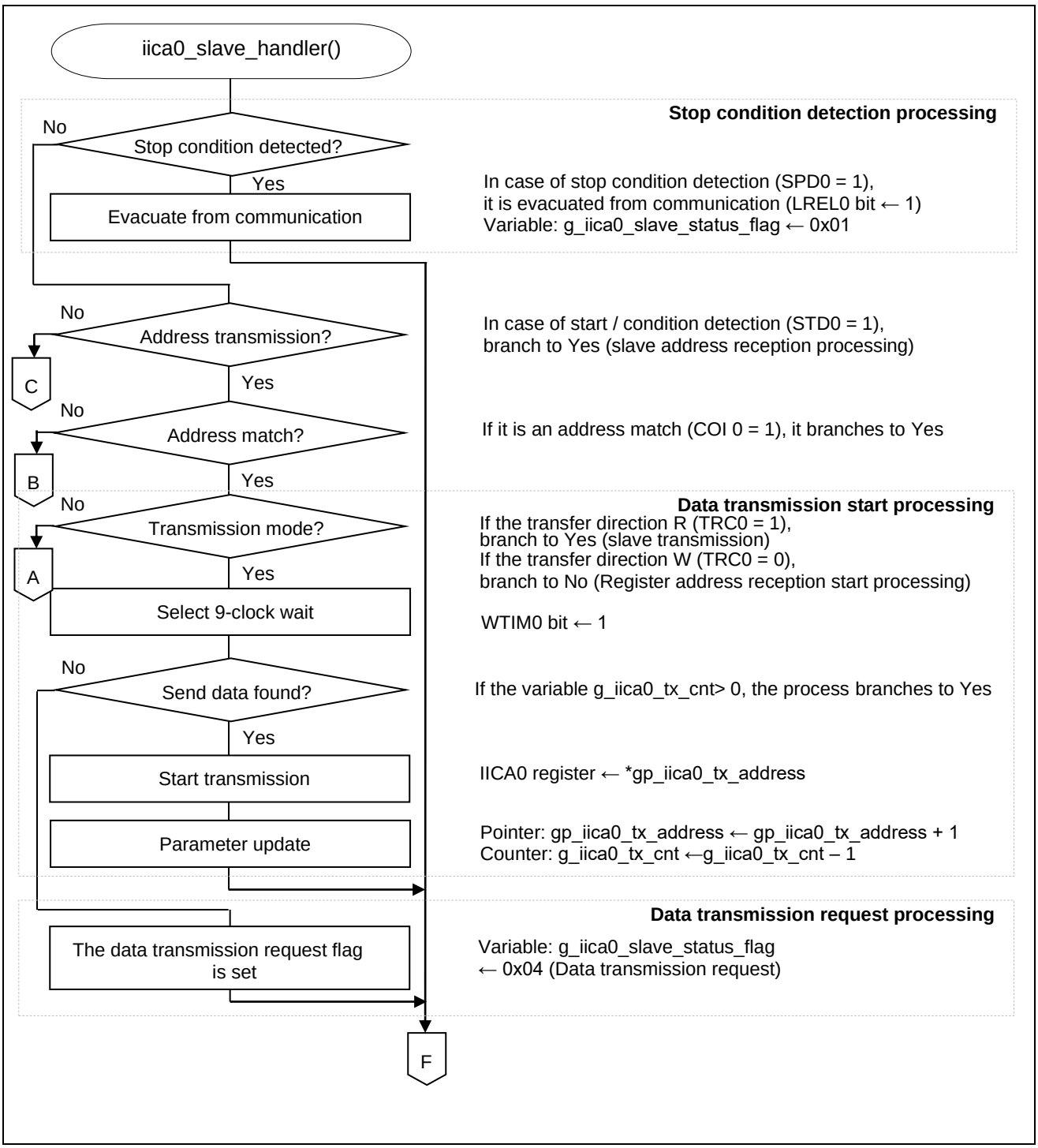


Figure 5.14 IICA0 Slave Handler (1/3)

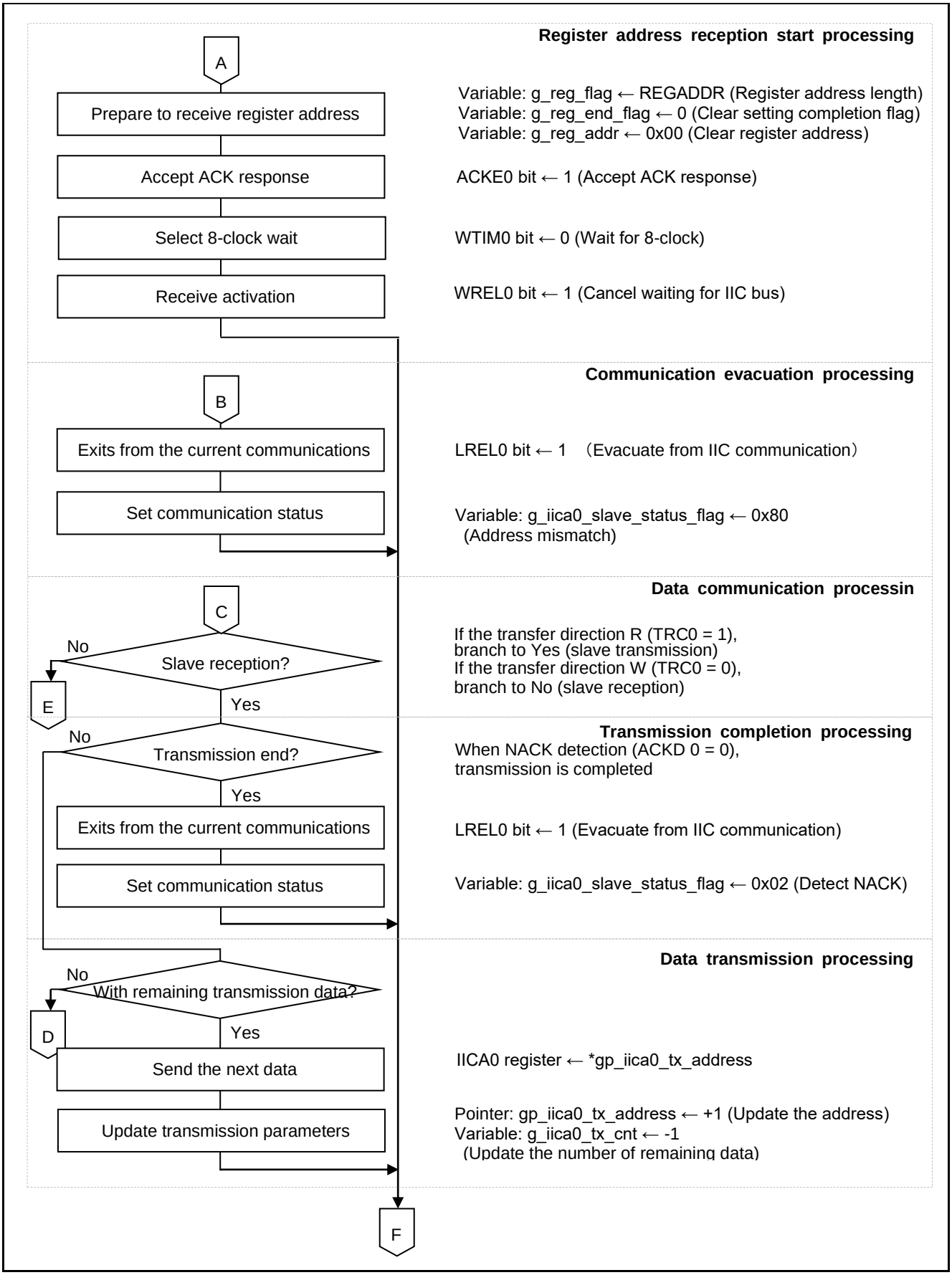


Figure 5.15 IICA0 Slave Handler (2/3)

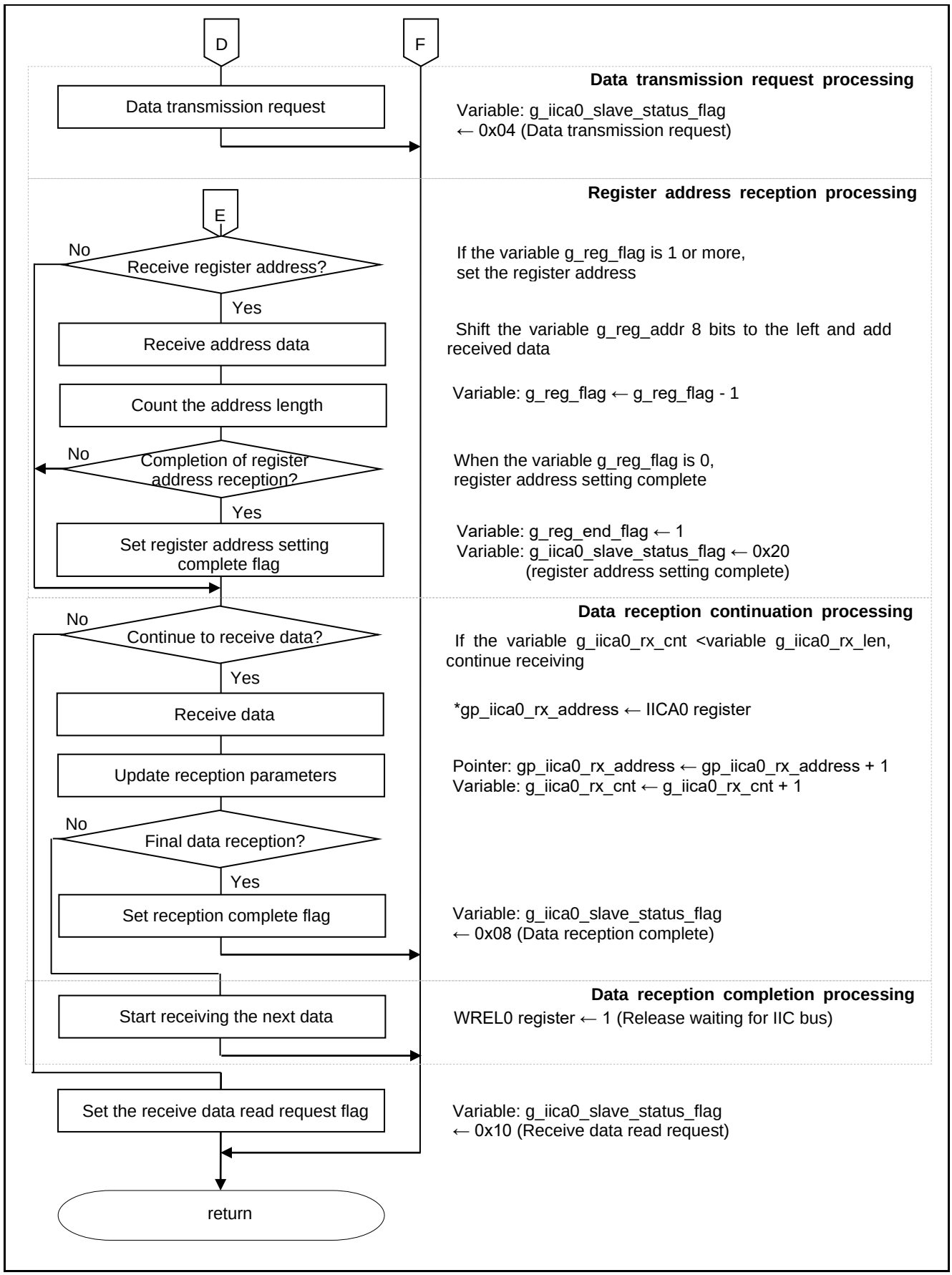


Figure 5.16 IICA0 Slave Handler (3/3)

6. Sample Code

The sample code is available on the Renesas Electronics Website.

7. Documents for Reference

User's Manual:

RL78/G13 User's Manual: Hardware (R01UH0146EJ)
RL78 Family User's Manual: Software (R01US0015EJ)
The latest version can be downloaded from the Renesas Electronics website.

Technical Updates/Technical News

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

Website and Support

Renesas Electronics Website

http://www.renesas.com/index.jsp

Inquiries

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

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Apr. 16, 2015	—	First edition issued
2.00	Apr. 1, 2019		Completely revised
2.01	Dec. 4, 2020		Add project for IAR Embedded Workbench

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/.