

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

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

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

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

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

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

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
  - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
  - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

# M16C/28, M16C/29 Group

## Multimaster I<sup>2</sup>C-BUS Interface

---

### 1. Summary

The Multimaster I<sup>2</sup>C-BUS Interface is a circuit to perform serial communication based on the I<sup>2</sup>C-BUS data transfer format of Philips. Its arbitration-lost detect function makes multimaster communication possible. This application note describes how to use the Multimaster I<sup>2</sup>C-BUS Interface functions.

Note: I<sup>2</sup>C-BUS is a registered trademark of Royal Philips Electronics of the Netherlands.

### 2. Introduction

The example application described here applies for use with the following microcomputers, subject to conditions stipulated below.

- Microcomputers: M16C/28 Group  
M16C/29 Group

This program can be used for the other M16C Families which have the same SFR (Special Function Register) as the one in the M16C/28 and M16C/29 groups. However, since some functions may be modified such as added functions, check it in a manual. Execute sufficient evaluation when using this application note.

### 3. Overview

#### 3.1 Multimaster System

The arbitration-lost detect function and the synchronous function are incorporated to make multimaster system communication possible.

#### 3.2 General Call

A general call <sup>Note</sup> whose address data are all 0s can be detected.

Note: A general call means that the master transmits general call address "00<sub>16</sub>" to all slaves.

#### 3.3 Addressing Format

The 7-bit addressing format is supported.

Only the 7 high-order bits of the I<sup>2</sup>C address register (i.e., the slave address) is compared with the address data.

#### 3.4 Multimaster I<sup>2</sup>C-BUS Interface Related Pins

- SCLmm pins      Clock input/output pins of the Multimaster I<sup>2</sup>C-BUS Interface
- SDAm pins      Data input/output pins of the Multimaster I<sup>2</sup>C-BUS Interface

### 3.5 Selectable Functions

The Multimaster I<sup>2</sup>C-BUS Interface permits the following functions to be selected.

#### (1) Communication mode

There are following four communication modes in which data communication is performed.

- **Master transmission**      Start and stop conditions are generated (master mode). The address and control data are output on to the SDA pin synchronously with the clock on the SCL pin that the master itself generates.
- **Master reception**          The data from the transmitting device is received synchronously with the clock on the SCL pin that the master itself generates.
- **Slave transmission**        The start and stop conditions generated by the master device are received (slave mode). Control data is output synchronously with the clock generated by the master device.
- **Slave reception**            The data from the transmitting device is received synchronously with the clock generated by the master device.

#### (2) SCL mode

SCL mode can be selected from the following two modes (BCLK = 20 MHz).

- **Standard clock mode**      The SCL frequency can be selected in the range 16.1 to 100 kHz.
- **High-speed clock mode**    The SCL frequency can be selected in the range 32.3 to 400 kHz.

#### (3) ACK clock

This is the function to specify whether an acknowledge signal should be generated for a data transfer. Acknowledge mode can be selected from the following two.

- **ACK clock unavailable**    No ACK clocks are generated after a data transfer.
- **ACK clock available**      The master generates an ACK clock each time one byte of data transfer finishes.

#### (4) Data format

This is the function to specify whether a slave address should be recognized. The data format can be selected from the following two.

- **Addressing format**        The address data is recognized. Transfer processing can only be performed when the slave address and the address data have matched when a general call is received.
- **Free data forma**            No slave addresses are recognized.

### 3.6 Multimaster I<sup>2</sup>C-BUS Interface Related Registers

Figure 1 shows a memory map of the registers associated with the Multimaster I<sup>2</sup>C-BUS Interface.

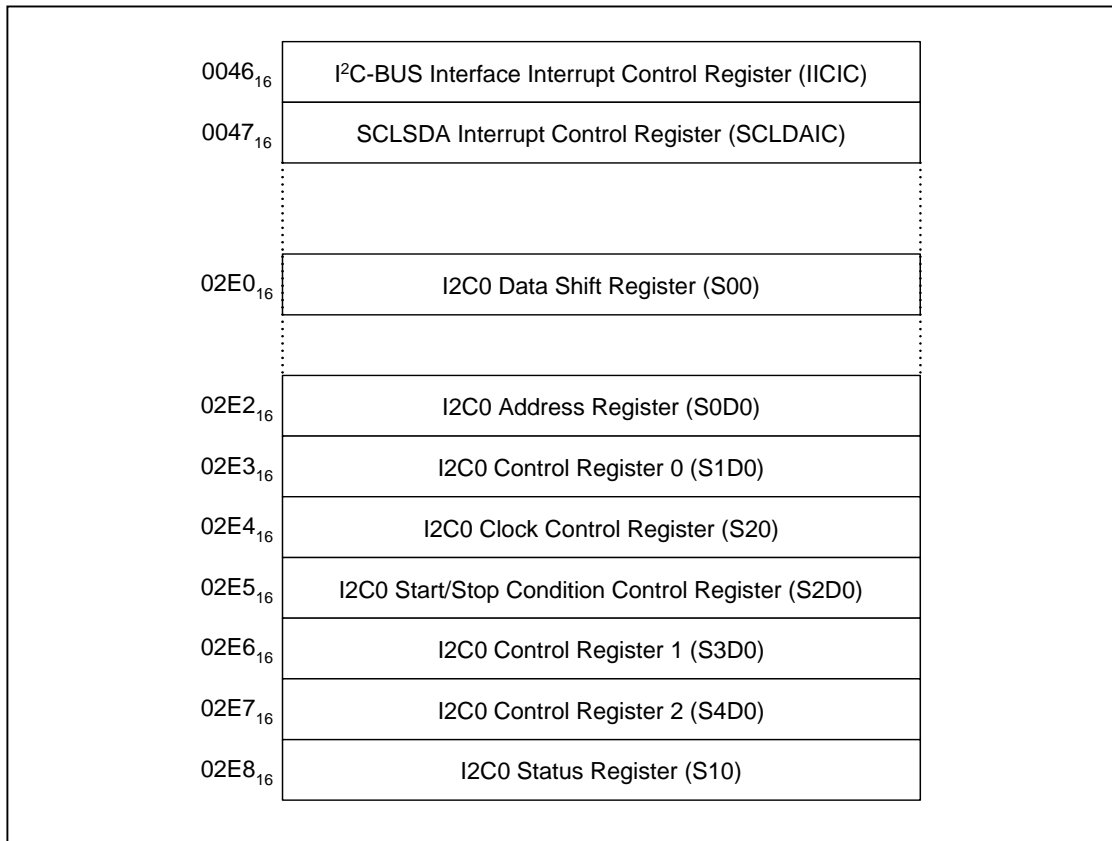


Figure 1. Memory map of the registers associated with the Multimaster I<sup>2</sup>C-BUS Interface

4. Example for Master Transmission

Table 1. Set Contents of Master Transmission

Set item	Set content	Set item	Set content
Addressing format	○ 7 bits	Data reception completed interrupt enable bit	○ Disabled
			○ Enabled
Data format	○ Addressing	Stop condition interrupt enable bit	○ Disabled
			○ Enabled
SCL mode	○ High-speed mode	ACK clock	○ Available
	○ Standard mode		○ Not available
Communication speed	○ 100 kHz	Timeout detect function enable bit	○ Disabled
Communication mode	○ Master transmission		
		Master reception	
		Slave transmission	
		Slave reception	

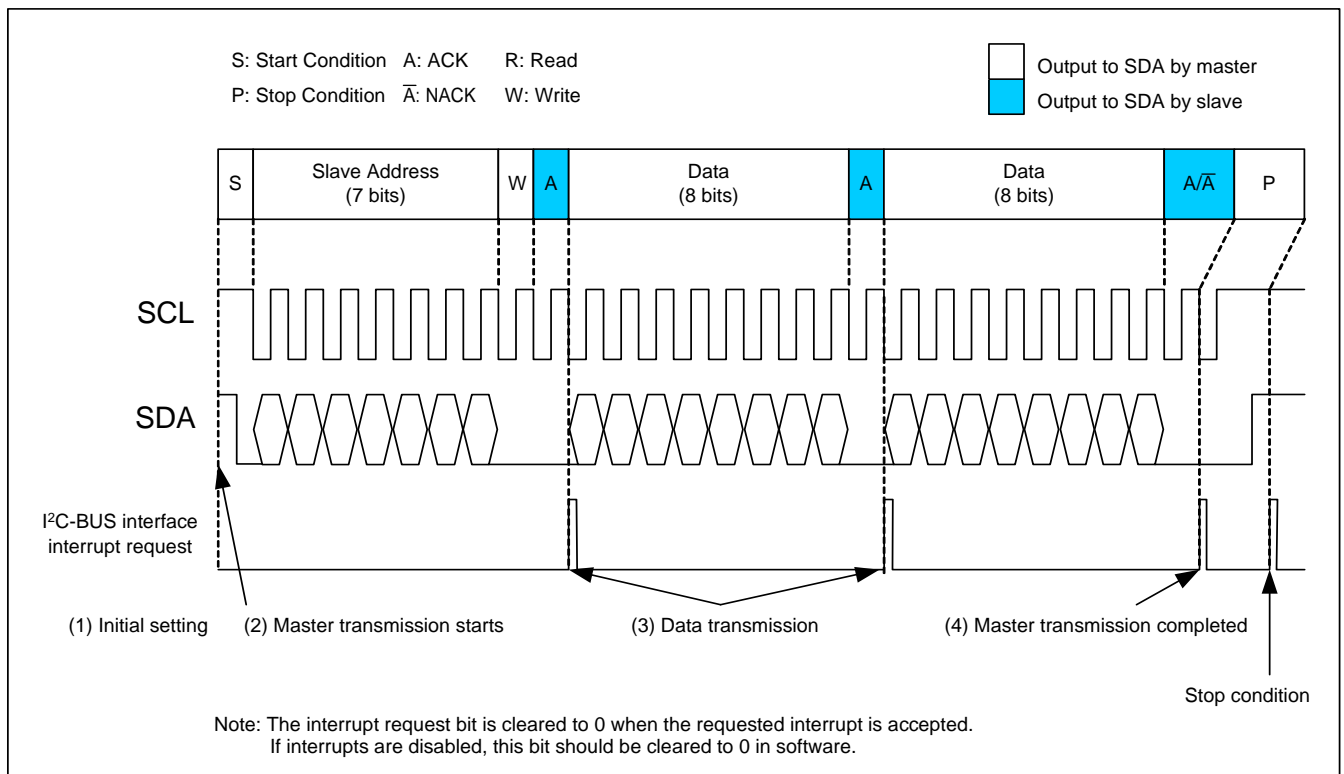


Figure 2. Operation timing of master transmission

(1) Initial setting

- 1) Set the local address in the 7 high-order bits of the I2C0 Address Register (S0D0).  
This setting is unnecessary when the I2C-BUS interface is used in a single-master system.
- 2) Set "85<sub>16</sub>" in the I2C0 Clock Control Register (S20).  
Set "011<sub>2</sub>" in the ICK4-ICK2 bits of I2C0 Control Register 2 (S4D0).  
Set "03<sub>16</sub>" in I2C0 Control Register 1 (S3D0).  
These settings select ACK clock = available, SCL = 100 kHz, and V<sub>IIC</sub> = 4 MHz (when f<sub>1</sub> = 20 MHz).
- 3) Set "00<sub>16</sub>" in the I2C0 Status Register (S10) to initialize transmit/receive modes.

- 4) Set "98<sub>16</sub>" in the I2C0 Start/Stop Condition Control Register (S2D0).
- 5) Set "08<sub>16</sub>" in I2C0 Control Register 0 (S1D0) to get the I<sup>2</sup>C-BUS interface enabled for communication.

## (2) Master transmission starts

- 1) Inspect the BB flag of the I2C0 Status Register (S10) to check whether the I<sup>2</sup>C-BUS interface is in a bus-free state.
- 2) Set "E0<sub>16</sub>" in the I2C0 Status Register (S10) to place the I<sup>2</sup>C-BUS interface in a start condition standby state.
- 3) Set the address data showing the destination of transmission in the 7 high-order bits and value "0" in the least significant bit of the I2C0 Data Shift Register (S00). This causes the I<sup>2</sup>C-BUS interface to generate a start condition. At this time, an SCL for one byte of data and an ACK clock are automatically generated.

## (3) Data transmission

Set the transmit data in the I2C0 Data Shift Register (S00) during an interrupt handling process of the I<sup>2</sup>C-BUS interface. At this time, the SCL and an ACK clock are automatically generated.

## (4) Master transmission completed

- 1) If ACK is not returned from the receiving slave side or when transmission is completed, set "C0<sub>16</sub>" in the I2C0 Status Register (S10) to place the I<sup>2</sup>C-BUS interface in a stop condition standby state.
- 2) Write dummy data to the I2C0 Data Shift Register (S00) to generate stop condition.

5. Example for Master Reception

Table 2. Set Contents of Master Reception

Set item	Set content	Set item	Set content
Addressing format	<input type="radio"/> 7 bits	Data reception completed interrupt enable bit	<input type="checkbox"/> Disabled
	<input type="checkbox"/> Free data		<input type="radio"/> Enabled
Data format	<input type="radio"/> Addressing	Stop condition interrupt enable bit	<input type="checkbox"/> Disabled
	<input type="checkbox"/> Free data		<input type="radio"/> Enabled
SCL mode	<input type="checkbox"/> High-speed mode	ACK clock	<input type="radio"/> Available
	<input type="radio"/> Standard mode		<input type="checkbox"/> Not available
Communication speed	<input type="radio"/> 100 kHz	Timeout detect function enable bit	<input type="radio"/> Disabled
Communication mode	<input type="checkbox"/> Master transmission		<input type="checkbox"/> Enabled
	<input type="radio"/> Master reception		
	<input type="checkbox"/> Slave transmission		
	<input type="checkbox"/> Slave reception		

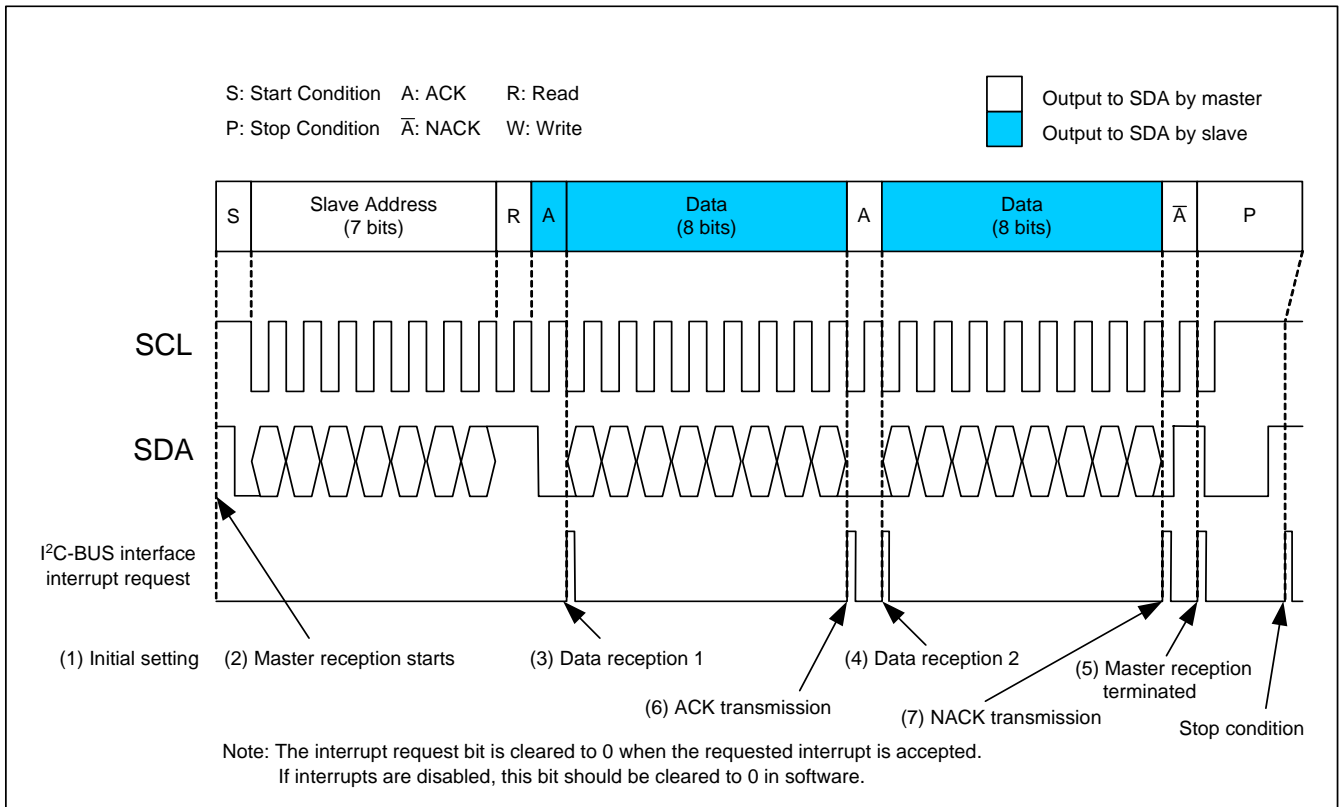


Figure 3. Operation timing of master reception

(1) Initial setting

- 1) Set a slave address in the 7 high-order bits of the I2C0 Address Register (S0D0).
- 2) Set “85<sub>16</sub>” in the I2C0 Clock Control Register (S20).  
Set “011<sub>2</sub>” in the ICK4–ICK2 bits of I2C0 Control Register 2 (S4D0).  
Set “03<sub>16</sub>” in the I2C0 Control Register 1 (S3D0).  
These settings select ACK = returned, SCL = 100 kHz, and V<sub>IIC</sub> = 4 MHz (when f<sub>1</sub> = 20 MHz).
- 3) Set “00<sub>16</sub>” in the I2C0 Status Register (S10) to initialize transmit/receive modes.



- 4) Set "98<sub>16</sub>" in the I<sup>2</sup>C0 Start/Stop Condition Control Register (S2D0).
- 5) Set "08<sub>16</sub>" in the I<sup>2</sup>C0 Control Register 0 (S1D0) to get the I<sup>2</sup>C-BUS interface enabled for communication.

## (2) Master reception starts

- 1) Inspect the BB flag of the I<sup>2</sup>C0 Status Register (S10) to check whether the I<sup>2</sup>C-BUS interface is in a bus-free state.
- 2) Set "E0<sub>16</sub>" in the I<sup>2</sup>C0 Status Register (S10) to place the I<sup>2</sup>C-BUS interface in a start condition standby state.
- 3) Set the address data showing the destination of transmission in the 7 high-order bits and value "0" in the least significant bit of the I<sup>2</sup>C0 Data Shift Register (S00). This causes the I<sup>2</sup>C-BUS interface to generate a start condition. At this time, an SCL for one byte of data and an ACK clock are automatically generated.

## (3) Data reception 1 (after transmitting a slave address)

Set "A0<sub>16</sub>" in the I<sup>2</sup>C0 Status Register (S10) during an interrupt of the I<sup>2</sup>C-BUS interface after transmitting slave data, to change communication mode to master reception. After that, set dummy data in the I<sup>2</sup>C0 Data Shift Register (S00).

## (4) Data reception 2 (data reception)

- 1) When one byte of data is received, an I<sup>2</sup>C-BUS interface interrupt request signal is generated. So read data out of the I<sup>2</sup>C0 Data Shift Register (S00) and write dummy data to the I<sup>2</sup>C0 Data Shift Register (S00).
- 2) To receive multiple bytes, repeat step 1) as necessary.

## (5) Master reception terminated

- 1) To terminate data reception, generate a stop condition instead of transmitting dummy data. First, set "C0<sub>16</sub>" in the I<sup>2</sup>C0 Status Register (S10) to place the I<sup>2</sup>C-BUS interface in a stop condition standby state.
- 2) Write dummy data to the I<sup>2</sup>C0 Data Shift Register (S00) to generate stop condition.

## (6) ACK transmission

For an ACK to be transmitted, the ACK bit of the I<sup>2</sup>C0 Clock Control Register (S20) should be set to 0. This causes the SCL pin that is fixed low to be released open.

## (7) NACK transmission

For a NACK to be transmitted, the ACK bit of the I<sup>2</sup>C0 Clock Control Register (S20) should be set to 1. This causes the SCL pin that is fixed low to be released open.

6. Example for Slave Reception

Table 3. Set Contents of Slave Reception

Set item	Set content		Set item	Set content	
Addressing format	○	7 bits	Data reception completed interrupt enable bit		Disabled
				○	Enabled
Data format	○	Addressing	Stop condition interrupt enable bit		Disabled
		Free data		○	Enabled
SCL mode		High-speed mode	ACK clock	○	Available
	○	Standard mode			Not available
Communication speed	○	100 kHz	Timeout detect function enable bit	○	Disabled
Communication mode		Master transmission			Enabled
		Master reception			
		Slave transmission			
	○	Slave reception			

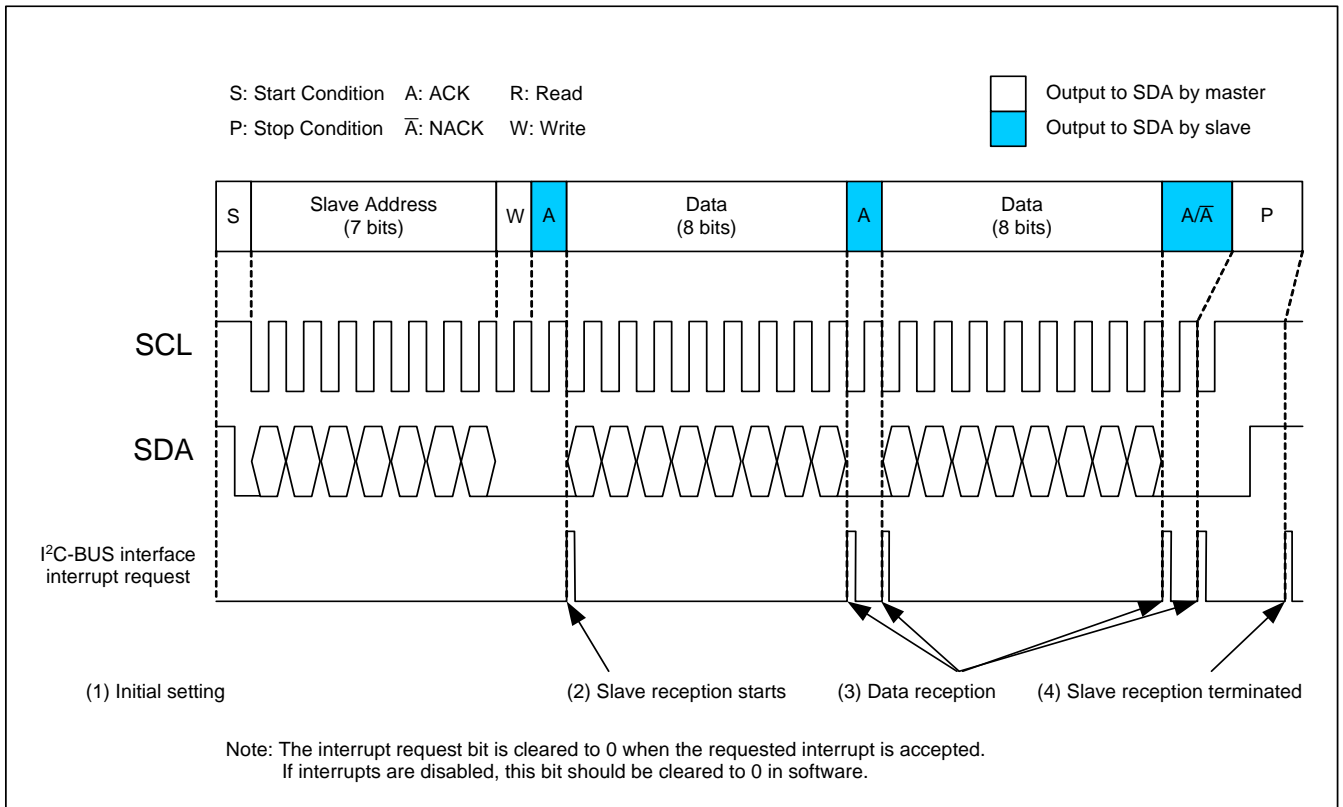


Figure 4. Operation timing of slave reception

(1) Initial setting

- 1) Set a slave address in the 7 high-order bits of the I2C0 Address Register (S0D0).
- 2) Set "85<sub>16</sub>" in the I2C0 Clock Control Register (S20).  
Set "011<sub>2</sub>" in the ICK4-ICK2 bits of I2C0 Control Register 2 (S4D0).  
Set "03<sub>16</sub>" in the I2C0 Control Register 1 (S3D0).  
These settings select ACK = returned, SCL = 100 kHz, and V<sub>IIC</sub> = 4 MHz (when f<sub>1</sub> = 20 MHz).
- 3) Set "00<sub>16</sub>" in the I2C0 Status Register (S10) to initialize transmit/receive modes.

- 4) Set "98<sub>16</sub>" in the I<sup>2</sup>C0 Start/Stop Condition Control Register (S2D0).
- 5) Set "08<sub>16</sub>" in the I<sup>2</sup>C0 Control Register 0 (S1D0) to get the I<sup>2</sup>C-BUS interface enabled for communication.

## (2) Slave reception starts

- 1) When the first data following a start condition is received, an address comparison is performed.
- 2)
  - If the transmitted address consists of all 0s (general call), the ADR0 bit of the I<sup>2</sup>C0 Status Register (S10) is set to 1 and an I<sup>2</sup>C-BUS interface interrupt request signal is generated.
  - If the transmitted address matches the address that was set in step 1) of (1) Initial setting, the AAS bit of the I<sup>2</sup>C0 Status Register (S10) is set to 1 and an I<sup>2</sup>C-BUS interface interrupt request signal is generated.
  - Otherwise, the ADR0 and AAS bits of the I<sup>2</sup>C0 Status Register (S10) both are set to 0 and no I<sup>2</sup>C-BUS interface interrupt request signal is generated.
- 3) When TRX = 0, a slave reception is assumed.
- 4) Set dummy data in the I<sup>2</sup>C0 Data Shift Register (S00).

## (3) Data reception

- 1) When one byte of data is received while WIT flag = 1, an I<sup>2</sup>C-BUS interface interrupt request signal is generated. So write 1 or 0 to the ACK bit to output an ACK. (In the sample program, ACK is always returned.)
- 2) An I<sup>2</sup>C-BUS interface interrupt request signal will be generated again after ACK clock is output. So read data from the I<sup>2</sup>C0 Data Shift Register (S00) and set dummy data in the I<sup>2</sup>C0 Data Shift Register (S00).
- 3) To receive multiple types of control data, repeat steps 1) and 2).

Note: When WIT = 0, ACK is automatically returned when one byte of data is received, and an I<sup>2</sup>C-BUS interface interrupt request signal is generated on falling edge of the ACK clock.

## (4) Slave reception terminated

- 1) Communication is terminated when a stop condition is detected.

## 7. Example for Slave Transmission

Table 4. Set Contents of Slave Transmission

Set item	Set content	Set item	Set content
Addressing format	<input type="radio"/> 7 bits	Data reception completed interrupt enable bit	<input type="checkbox"/> Disabled
	<input type="checkbox"/> Free data		<input type="radio"/> Enabled
Data format	<input type="radio"/> Addressing	Stop condition interrupt enable bit	<input type="checkbox"/> Disabled
	<input type="checkbox"/> Free data		<input type="radio"/> Enabled
SCL mode	<input type="checkbox"/> High-speed mode	ACK clock	<input type="radio"/> Available
	<input type="radio"/> Standard mode		<input type="checkbox"/> Not available
Communication speed	<input type="radio"/> 100 kHz	Timeout detect function enable bit	<input type="radio"/> Disabled
Communication mode	<input type="checkbox"/> Master transmission		<input type="checkbox"/> Enabled
	<input type="checkbox"/> Master reception		
	<input type="radio"/> Slave transmission		
	<input type="checkbox"/> Slave reception		

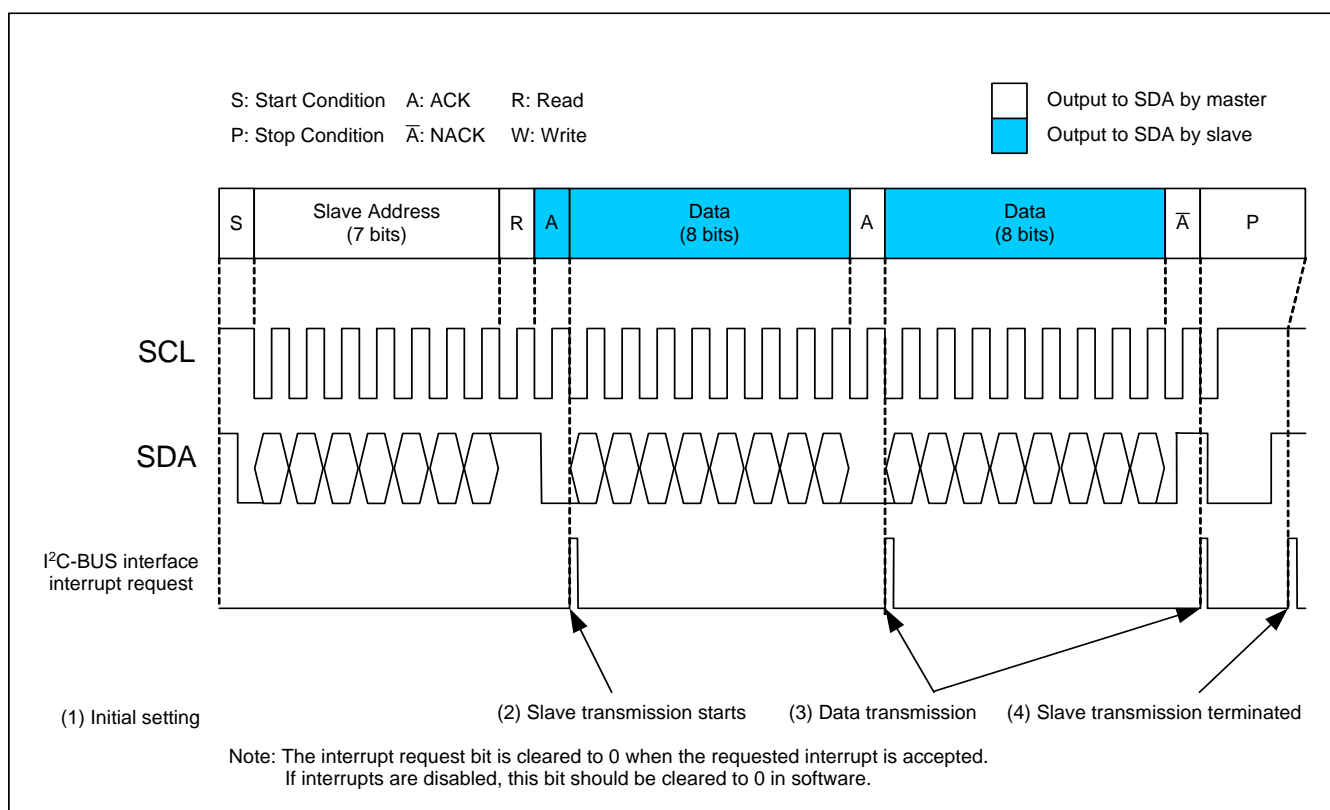


Figure 5. Operation timing of slave transmission

### (1) Initial setting

- 1) Set a slave address in the 7 high-order bits of the I2C0 Address Register (S0D0).
- 2) Set "85<sub>16</sub>" in the I2C0 Clock Control Register (S20).  
Set "011<sub>2</sub>" in the ICK4-ICK2 bits of I2C0 Control Register 2 (S4D0).  
Set "03<sub>16</sub>" in the I2C0 Control Register 1 (S3D0).  
These settings select ACK = returned, SCL = 100 kHz, and V<sub>IIC</sub> = 4 MHz (when f<sub>1</sub> = 20 MHz).
- 3) Set "00<sub>16</sub>" in the I2C0 Status Register (S10) to initialize transmit/receive modes.

- 4) Set "98<sub>16</sub>" in the I<sup>2</sup>C0 Start/Stop Condition Control Register (S2D0).
- 5) Set "08<sub>16</sub>" in the I<sup>2</sup>C0 Control Register 0 (S1D0) to get the I<sup>2</sup>C-BUS interface enabled for communication.

## (2) Slave transmission starts

- 1) When the first data following a start condition is received, an address comparison is performed.
- 2)
  - If the transmitted address consists of all 0s (general call), the ADR0 bit of the I<sup>2</sup>C0 Status Register (S10) is set to 1 and an I<sup>2</sup>C-BUS interface interrupt request signal is generated. (The sample program does not support general calls.)
  - If the transmitted address matches the address that was set in step 1) of (1) Initial setting in Section 6, the AAS bit of the I<sup>2</sup>C0 Status Register (S10) is set to 1 and an I<sup>2</sup>C-BUS interface interrupt request signal is generated.
  - Otherwise, the ADR0 and AAS bits of the I<sup>2</sup>C0 Status Register (S10) both are set to 0 and no I<sup>2</sup>C-BUS interface interrupt request signal is generated.
- 3) When TRX = 1, a slave transmission is assumed.
- 4) Set transmit data in the I<sup>2</sup>C0 Data Shift Register (S00).

## (3) Data transmission

- 1) When one byte of data is transmitted, an I<sup>2</sup>C-BUS interface interrupt request signal will be generated. When this interrupt is generated, set transmit data in the I<sup>2</sup>C0 Data Shift Register (S00).
- 2) To transmit multiple bytes, repeat step 1) as necessary.

## (4) Slave transmission terminated

- 1) If ACK is not returned from the transmitting master side, set dummy data in the I<sup>2</sup>C0 Data Shift Register (S00) to terminate the slave transmission.

## 8. Arbitration Lost

The following describes the operation of the I<sup>2</sup>C-BUS interface when an arbitration-lost occurs.

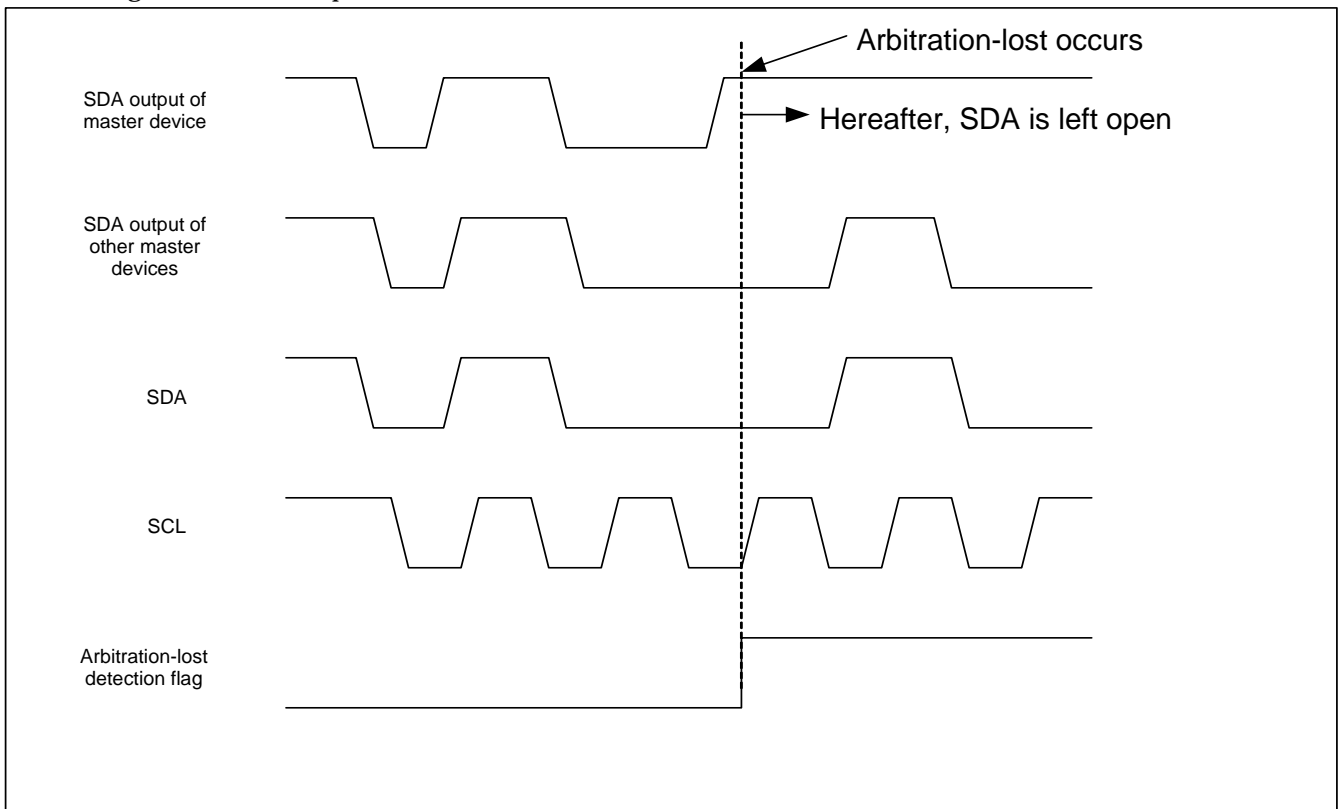


Figure 6. Operation timing of the arbitration-lost detection flag

When an arbitration-lost occurs, the arbitration-lost detection flag is set to 1.

(1) If a slave address is being transmitted when an arbitration-lost has occurred

When an arbitration-lost is detected, communication mode automatically changes to slave reception enabling the slave address to be received. If the selected data format is an addressing format, the slave address can be resolved by inspecting the AAS bit of the S10 register.

(2) If any data following a slave address is being transmitted when an arbitration-lost has occurred

When an arbitration-lost is detected, communication mode automatically changes to slave reception enabling the data to be received.

### 8.1 Precautions to Take when Clearing the Arbitration-lost Detection Flag

There are following three methods to clear the arbitration-lost detection flag, each of which requires caution.

(1) Writing to the S00 register

When the S00 register is accessed for write for other than transmitting a start condition, a short low-level pulse may be output from the SCL pin. Therefore, except when transmitting a start condition, switch the SCL pin for port output using the port function select bit (PEC bit of the S3D0 register) before writing to the S00 register.

Before switching back to the SCL output function from the port output function, wait 3 VIIC clock cycles or more after writing to the S00 register.

(2) Setting the ES0 bit of the S1D0 register to 0 (I<sup>2</sup>C-BUS interface disabled)

The bus busy flag is cleared at the same time the arbitration-lost detection flag is cleared.

When the I<sup>2</sup>C-BUS interface is used in a multimaster system, it is possible that although the bus remains in a busy state until a stop condition is detected, because there are no means of detecting a busy state of the bus, a start condition will be sent out while the bus remains busy.

- (3) Setting the IHR bit of the S1D0 register to 1 (reset)  
Same as described in (2).

## 8.2 Procedure for Clearing the Arbitration-lost Detection Flag

To clear the arbitration-lost detection flag, follow the procedure described below.

- (1) Set bit 1 in the Port 2 Register to 0 or 1.
- (2) Set the port function select bit (PEC) to 1.
- (3) Write dummy data to the S00 register (AL bit cleared).
- (4) Wait for 3 VIIC clock cycles after writing to the S00 register. (Equivalent to 15 NOPs when VIIC = 4 MHz, Xin = 20 MHz)
- (5) Set the port function select bit (PEC) to 0.

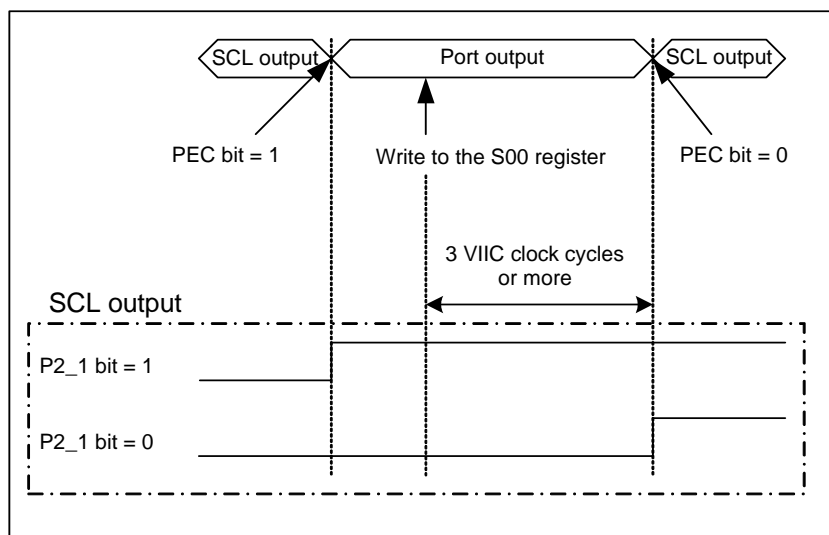


Figure 7. Procedure for Clearing the Arbitration-lost Detection Flag

## 9. Interrupts

The I<sup>2</sup>C-BUS interface has the following four causes of interrupt.

(1) Interrupt when 9-bit transmission/reception is completed (including ACK/NACK)

The causes of interrupt can be determined by inspecting the WIT bit of the S3D0 register. When WIT bit = 0, it means that the generated interrupt is attributable to this cause of interrupt.

(2) Interrupt when 8 bits are received

Setting the WIT bit of the S3D0 register enables this cause of interrupt.

The causes of interrupt can be determined by inspecting the WIT bit of the S3D0 register. When WIT bit = 1, it means that the generated interrupt is attributable to this cause of interrupt.

If no determination is made of ACK–NACK transmissions, there is no need to use this interrupt.

(3) Interrupt when a stop condition is detected

Setting the SIM bit of the S3D0 register enables this cause of interrupt.

The causes of interrupt can be determined by inspecting the SCPIN bit of the S4D0 register. When a stop condition is detected, the SCPIN bit is set to 1.

(4) Interrupt when the SCL clock remains stuck high for more than a predetermined time during communication

Setting the TOE bit of the S4D0 register enables this cause of interrupt.

The causes of interrupt can be determined by inspecting the TOF bit of the S4D0 register. When the SCL clock remains stuck high for more than a predetermined time during communication, the TOF bit is set to 1.



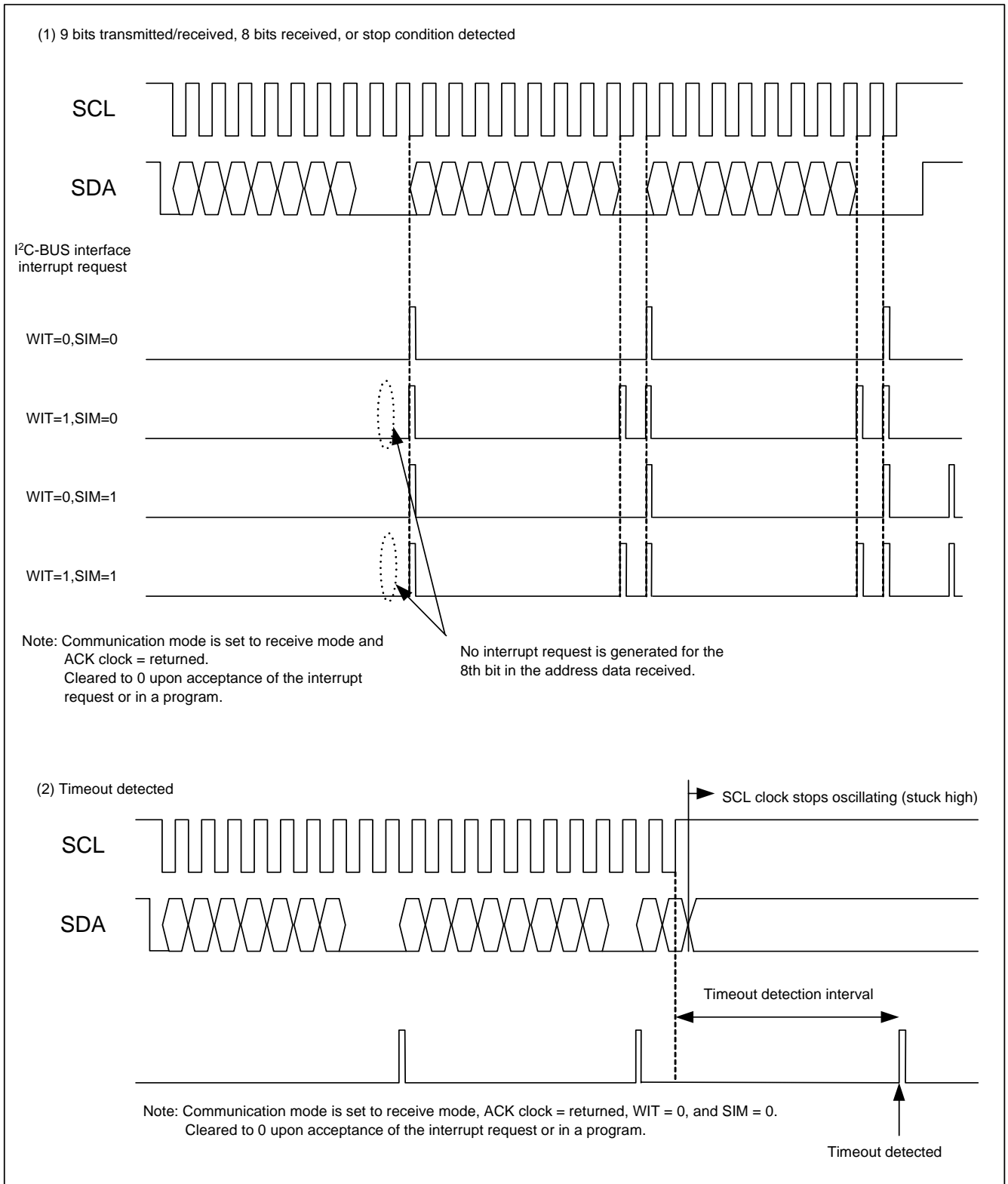


Figure 8. Timing with which I<sup>2</sup>C-BUS interface interrupt requests are generated

## 10. Sample Program

This sample program is provided for reference purposes only, and is not guaranteed to operate properly in all systems.

When it is to be incorporated into a system, a careful examination should be made before being put to use.

Furthermore, since its functionality as integral part of a system cannot be evaluated with this program alone, evaluation with the final system is indispensable.

### 10.1 Operating Conditions

Oscillator frequency: 20 MHz

Table 5. Set Contents of the Sample Program

Set item	Set content		Set item	Set content	
Addressing format	<input type="radio"/>	7 bits	Data reception completed interrupt enable bit	<input type="checkbox"/>	Disabled
	<input type="checkbox"/>			<input type="radio"/>	Enabled
Data format	<input type="radio"/>	Addressing	Stop condition interrupt enable bit	<input type="checkbox"/>	Disabled
	<input type="checkbox"/>	Free data		<input type="radio"/>	Enabled
SCL mode	<input type="checkbox"/>	High-speed mode	ACK clock	<input type="radio"/>	Available
	<input type="radio"/>	Standard mode		<input type="checkbox"/>	Not available
Communication speed	<input type="checkbox"/>	100 kHz	Timeout detect function enable bit	<input type="radio"/>	Disabled
Communication mode	<input type="checkbox"/>	Master transmission		<input type="checkbox"/>	Enabled
	<input type="checkbox"/>	Master reception	<input type="checkbox"/>		
	<input type="checkbox"/>	Slave transmission	<input type="checkbox"/>		
	<input type="checkbox"/>	Slave reception	<input type="checkbox"/>		

### 10.2 Functions

#### ■ Initialization function

```
char iic_ini(char SWITCH,char SUBADDRESS);
```

**Description:** This function initializes the I<sup>2</sup>C-BUS interface to permit transmit/receive operations to be performed via the I<sup>2</sup>C-BUS. When the initialization is finished and interrupts are found enabled, the I<sup>2</sup>C-BUS interface is ready to operate as a slave device. Furthermore, it can be made to operate as a master device by calling the function to start a master transmission/reception that is described below.

**Parameters:** SWITCH    0: Disables the I<sup>2</sup>C-BUS function  
                               1: Enables the I<sup>2</sup>C-BUS function  
                               SUBADDRESS: Set the subaddress value of the master device

**Return value:**            0: Failed  
                               1: Succeeded

**Other:**

When the I<sup>2</sup>C-BUS function is disabled, the functions described below cannot be used.

#### ■ Master control start function

```
char iic_master_start(char SLAVE, char RW, char * BUF, char LEN);
```

**Description:** This function causes master control to start. Before this function can be used, the I<sup>2</sup>C-BUS must be enabled for use by iic\_ini.

**Parameters:** SLAVE 0x00–0x7f: Address of the slave device to be specified

RW                    0: Master transmit operation  
                         1: Master receive operation  
\*BUF:                Pointer to the transmit or receive buffer  
LEN    0x00–0xff: Communication data length  
Return value        0: Master control failed to start  
                         1: Master control started successfully

Note: The function “fclr i” is executed at the beginning of the master control start function to disable interrupts. The processing performed here permits a start condition to be output as early as possible after determination of a bus busy state.

#### [I<sup>2</sup>C-BUS functions created by user]

Call the functions that indicate transmit/receive status and its data bytes in their parameters, as described below.

These functions must be provided by the user.

#### ■ Master control complete function

```
void iic_master_end(char STATUS);
```

Description: This is the function called by the firmware after master control is completed. The status with which master communication has terminated is notified to the user by the parameters shown below.

Parameters: STATUS

4 high-order bits	1: Master transmission
	2: Master reception
4 low-order bits	0: Terminated normally
	1: Lost in bus contention
	2: Terminated with NACK

Return value: None

Other:

This function is called from within an I<sup>2</sup>C-BUS interrupt handling process.

#### ■ Slave check function

```
* char iic_id_chk(char RW);
```

Description: This is the function called by the firmware after one byte is received. The content of a master-to-slave request is notified to the user by the parameters shown below. When NULL pointer is returned, slave specification is denied; when the pointer to the communication buffer is returned, a slave operation is started.

Parameters: RW            0: Master requests reception (so that slave receives)  
                                 1: Master requests transmission (so that slave transmits)

Return value NULL pointer: Slave specification denied  
                 pointer:        Pointer to the transmit or receive buffer

#### ■ Slave control complete function

```
void iic_slave_end(char STATUS,char IIC_INDEX);
```

Description: This is the function called by the firmware after slave control is completed. The status with which slave communication has terminated is notified to the user by the parameters shown below.

Parameters: STATUS    4 high-order bits    1: Slave reception  
   2: Slave transmission  
                                 4 low-order bits    0: Terminated normally  
                         IIC\_INDEX            0x00–0xff: Number of data bytes received

Return value: None

### 10.3 Using the Sample Program

- Add the interrupt vectors listed below.  
Software interrupt number 6 (Multimaster I<sup>2</sup>C-BUS Interface)  
.glb\_iic\_int  
.lword\_iic\_int
- Set the IFSR27 bit of Interrupt Source Select Register 2 to 1 to select the Multimaster I<sup>2</sup>C-BUS Interface for the interrupt source.

## 11. Reference

Renesas Technology Corporation M16C Family Home Page

<http://www.renesas.com/en/m16c>

E-mail Support

E-mail: [csc@renesas.com](mailto:csc@renesas.com)

Hardware Manual

M16C/28 and M16C/29 Group Hardware Manual

(Use the latest version on the home page: <http://www.renesas.com>)

TECHNICAL UPDATE/TECHNICAL NEWS

(Use the latest information on the home page: <http://www.renesas.com>)

## REVISION HISTORY

Rev.	Date	Description	
		Page	Summary
1.00	2005.06.23	-	First edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.  
The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.  
Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.  
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.