User's Manual

16

RL78/G10

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev3.20 Mar 2023

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

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/

Trademarks

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

© 2023 Renesas Electronics Corporation. All rights reserved.

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.

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the RL78/G10 and design and develop application systems and programs for these devices.
The target products are as follows.
• 10-pin: R5F10Y1x (x = 4, 6, 7) • 16-pin: R5F10Y4x (x = 4, 6, 7)

Purpose

This manual is intended to give users an understanding of the functions described in the **Organization** below.

Organization

The RL78/G10 manual is separated into two parts: this manual and the software edition (common to the RL78 family).

RL78/G10 **User's Manual** **Hardware**	**RL78 family** **User's Manual** **Software**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, and microcontrollers.

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.
- How to interpret the register format:
  → For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler.
- To know details of the RL78/G10 Microcontroller instructions:
  → Refer to the separate document **RL78 Family User's Manual: Software (R01US0015E)**.

Conventions	Data significance:	Higher digits on the left and lower digits on the right	
	Active low representations:	$\overline{\times\times\times}$ (overscore over pin and signal name)	
	Note:	Footnote for item marked with **Note** in the text	
	Caution:	Information requiring particular attention	
	Remark:	Supplementary information	
	Numerical representations:	Binary	···×××× or ××××B
		Decimal	···××××
		Hexadecimal	···××××H

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

Documents Related to Devices

Document Name	Document No.
RL78/G10 User's Manual: Hardware	R01UH0384E
RL78 Family User's Manual: Software	R01US0015E

Documents Related to Flash Memory Programming

Document Name		Document No.
PG-FP5 Flash Memory Programmer User's Manual		—
	RL78, 78K, V850, RX100, RX200, RX600 (Except RX64x), R8C, SH	R20UT2923E
	Common	R20UT2922E
	Setup Manual	R20UT0930E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

Other Documents

Document Name	Document No.
Renesas Microcontrollers RL78 Family	R01CP0003E
Semiconductor Package Mount Manual	R50ZZ0003E
Semiconductor Reliability Handbook	R51ZZ0001E

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

All trademarks and registered trademarks are the property of their respective owners.
EEPROM is a trademark of Renesas Electronics Corporation.
Windows, Windows NT and Windows XP are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
PC/AT is a trademark of International Business Machines Corporation.
SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

CONTENTS

CHAPTER 1 OUTLINE **1**

1.1 Features **1**

1.2 List of Part Numbers **3**

1.3 Pin Configuration (Top View) **4**

1.3.1 10-pin products 4

1.3.2 16-pin products 5

1.4 Pin Identification **6**

1.5 Block Diagram **7**

1.5.1 10-pin products 7

1.5.2 16-pin products 8

1.6 Outline of Functions **9**

CHAPTER 2 PIN FUNCTIONS **11**

2.1 Port Functions **11**

2.1.1 10-pin products 11

2.1.2 16-pin products 12

2.2 Functions other than port pins **13**

2.2.1 Functions for each product 13

2.2.2 Description of functions 14

2.3 Connection of Unused Pins **15**

2.4 Block Diagrams of Pins **16**

CHAPTER 3 CPU ARCHITECTURE **24**

3.1 Memory Space **25**

3.1.1 Internal program memory space 28

3.1.2 Mirror area 30

3.1.3 Internal data memory space 31

3.1.4 Special function register (SFR) area 31

3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area 31

3.1.6 Data memory addressing 32

3.2 Processor Registers **33**

3.2.1 Control registers 33

3.2.2 General-purpose registers 35

3.2.3 ES and CS registers 36

3.2.4 Special function registers (SFRs) 37

3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers) 40

3.3 Instruction Address Addressing **43**

3.3.1 Relative addressing 43
3.3.2 Immediate addressing 43
3.3.3 Table indirect addressing 44
3.3.4 Register indirect addressing 44
3.4 Addressing for Processing Data Addresses 45
3.4.1 Implied addressing 45
3.4.2 Register addressing 45
3.4.3 Direct addressing 46
3.4.4 Short direct addressing 47
3.4.5 SFR addressing 48
3.4.6 Register indirect addressing 49
3.4.7 Based addressing 50
3.4.8 Based indexed addressing 54
3.4.9 Stack addressing 55

CHAPTER 4 PORT FUNCTIONS 59
4.1 Port Functions 59
4.2 Port Configuration 59
4.2.1 Port 0 60
4.2.2 Port 4 60
4.2.3 Port 12 60
4.2.4 Port 13 60
4.3 Registers Controlling Port Function 61
4.3.1 Port mode registers 0, 4 (PM0, PM4) 62
4.3.2 Port registers 0, 4, 12, 13 (P0, P4, P12, P13) 63
4.3.3 Pull-up resistor option registers 0, 4, 12 (PU0, PU4, PU12) 64
4.3.4 Port output mode register 0 (POM0) 65
4.3.5 Port mode control register 0 (PMC0) 66
4.3.6 Peripheral I/O redirection register (PIOR) 67
4.4 Port Function Operations 68
4.4.1 Writing to I/O port 68
4.4.2 Reading from I/O port 68
4.4.3 Operations on I/O port 68
4.5 Register Settings When an Alternate Function Is Used 69
4.5.1 Basic concepts on using an alternate function 69
4.5.2 Register settings for alternate functions that do not use an output function 70
4.5.3 Example of register settings for port and alternate functions used 70
4.6 Cautions When Using Port Function 74
4.6.1 Cautions on 1-bit manipulation instruction for port register n (Pn) 74
4.6.2 Notes on specifying the pin settings 75

CHAPTER 5 CLOCK GENERATOR 76

5.1 Functions of Clock Generator 76

5.2 Configuration of Clock Generator 77

5.3 Registers Controlling Clock Generator 79

5.3.1 Clock operation mode control register (CMC) 80

5.3.2 System clock control register (CKC) 81

5.3.3 Clock operation status control register (CSC) 82

5.3.4 Oscillation stabilization time counter status register (OSTC) 83

5.3.5 Oscillation stabilization time select register (OSTS) 85

5.3.6 Peripheral enable register 0 (PER0) 86

5.3.7 Operation speed mode control register (OSMC) 87

5.3.8 High-speed on-chip oscillator frequency selection register (HOCODIV) 88

5.4 System Clock Oscillator 89

5.4.1 X1 oscillator (16-pin products only) 89

5.4.2 High-speed on-chip oscillator 92

5.4.3 Low-speed on-chip oscillator 92

5.5 Clock Generator Operation 92

5.6 Controlling Clock 94

5.6.1 Example of setting high-speed on-chip oscillator 94

5.6.2 Example of setting X1 oscillation clock 95

5.6.3 CPU clock status transition diagram 96

5.6.4 Condition before changing CPU clock and processing after changing CPU clock 99

5.6.5 Time required for switchover of CPU clock and main system clock 100

5.6.6 Conditions before clock oscillation is stopped 100

5.7 Resonator and Oscillator Constants 101

CHAPTER 6 TIMER ARRAY UNIT 103

6.1 Functions of Timer Array Unit 105

6.1.1 Independent channel operation function 105

6.1.2 Simultaneous channel operation function 107

6.1.3 8-bit timer operation function (channels 1 and 3 only) 109

6.2 Configuration of Timer Array Unit 110

6.2.1 Timer counter register 0n (TCR0n) 114

6.2.2 Timer data register 0n (TDR0n) 116

6.3 Registers Controlling Timer Array Unit 118

6.3.1 Peripheral enable register 0 (PER0) 119

6.3.2 Timer clock select register 0 (TPS0) 120

6.3.3 Timer mode register 0n (TMR0n) 121

6.3.4 Timer status register 0n (TSR0n) 126

6.3.5 Timer channel enable status register 0 (TE0, TEH0 (8-bit mode)) 127

6.3.6 Timer channel start register 0 (TS0, TSH0 (8-bit mode)) 128
6.3.7 Timer channel stop register 0 (TT0, TTH0 (8-bit mode)) 129
6.3.8 Timer output enable register 0 (TOE0) 130
6.3.9 Timer output register 0 (TO0) 131
6.3.10 Timer output level register 0 (TOL0) 132
6.3.11 Timer output mode register 0 (TOM0) 133
6.3.12 Noise filter enable register 1 (NFEN1) 134
6.3.13 Input switch control register (ISC) 135
6.3.14 Registers controlling port functions of pins to be used for timer I/O 136
6.4 Basic Rules of Timer Array Unit 137
6.4.1 Basic rules of simultaneous channel operation function 137
6.4.2 Basic rules of 8-bit timer operation function (only channels 1 and 3) 139
6.5 Operation of Counter 140
6.5.1 Count clock (f_{TCLK}) 140
6.5.2 Start timing of counter 142
6.5.3 Counter operation 143
6.6 Channel Output (TO0n pin) Control 148
6.6.1 TO0n pin output circuit configuration 148
6.6.2 TO0n pin output setting 149
6.6.3 Cautions on channel output operation 150
6.6.4 Collective manipulation of TO0n bit 154
6.6.5 Timer interrupt and TO0n pin output at count operation start 155
6.7 Timer Input (TI0n) Control 156
6.7.1 TI0n input circuit configuration 156
6.7.2 Noise filter 156
6.7.3 Cautions on channel input operation 157
6.8 Independent Channel Operation Function of Timer Array Unit 158
6.8.1 Operation as interval timer/square wave output 158
6.8.2 Operation as external event counter 164
6.8.3 Operation as frequency divider (only channels 0 and 3) 169
6.8.4 Operation as input pulse interval measurement 174
6.8.5 Operation as input signal high-/low-level width measurement 179
6.8.6 Operation as delay counter 184
6.9 Simultaneous Channel Operation Function of Timer Array Unit 189
6.9.1 Operation as one-shot pulse output 189
6.9.2 Two-channel input with one-shot pulse output function 198
6.9.3 Operation as PWM output function 207
6.9.4 Operation as multiple PWM output function 215
6.10 Cautions When Using Timer Array Unit 224
6.10.1 Cautions when using timer output 224

CHAPTER 7 12-BIT INTERVAL TIMER 225

7.1 Functions of 12-bit Interval Timer 225

7.2 Configuration of 12-bit Interval Timer 225

7.3 Registers Controlling 12-bit Interval Timer 226

7.3.1 Peripheral enable register 0 (PER0) 226

7.3.2 Operation speed mode control register (OSMC) 227

7.3.3 Interval timer control register (ITMCH, ITMCL) 228

7.4 12-bit Interval Timer Operation 229

7.4.1 12-bit interval timer operation timing 229

7.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode 230

CHAPTER 8 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER 231

8.1 Functions of Clock Output/Buzzer Output Controller 231

8.2 Configuration of Clock Output/Buzzer Output Controller 232

8.3 Registers Controlling Clock Output/Buzzer Output Controller 232

8.3.1 Clock output select register 0 (CKS0) 233

8.3.2 Registers controlling port functions of clock output/buzzer output pin 234

8.4 Operations of Clock Output/Buzzer Output Controller 235

8.4.1 Operation as output pin 235

CHAPTER 9 WATCHDOG TIMER 236

9.1 Functions of Watchdog Timer 236

9.2 Configuration of Watchdog Timer 237

9.3 Register Controlling Watchdog Timer 238

9.3.1 Watchdog timer enable register (WDTE) 238

9.4 Operation of Watchdog Timer 239

9.4.1 Controlling operation of watchdog timer 239

9.4.2 Setting time of watchdog timer 240

CHAPTER 10 A/D CONVERTER 241

10.1 Function of A/D Converter 241

10.2 Configuration of A/D Converter 243

10.3 Registers Used in A/D Converter 244

10.3.1 Peripheral enable register 0 (PER0) 245

10.3.2 A/D converter mode register 0 (ADM0) 246

10.3.3 A/D converter mode register 2 (ADM2) 250

10.3.4 A/D conversion result higher-order bit storage register (ADCRH) 250

10.3.5 A/D conversion result lower-order bit storage register (ADCRL) 251

10.3.6 Analog input channel specification register (ADS) 252

10.3.7 A/D test register (ADTES) ... 253
10.3.8 Registers controlling port function of analog input pins ... 253
10.4 A/D Converter Conversion Operations ... 254
10.5 Input Voltage and Conversion Results ... 256
10.6 A/D Converter Operation Modes ... 257
10.7 A/D Converter Setup Flowchart ... 258
10.7.1 Setting up A/D conversion of voltages on ANI0 to ANI6 ... 258
10.7.2 Setting up A/D conversion of the internal reference voltage (16-pin products only) ... 259
10.8 How to Read A/D Converter Characteristics Table ... 260
10.8.1 Resolution ... 260
10.8.2 Overall error ... 260
10.8.3 Quantization error ... 260
10.8.4 Zero-scale error ... 261
10.8.5 Full-scale error ... 261
10.8.6 Integral linearity error ... 261
10.8.7 Differential linearity error ... 261
10.8.8 Conversion time ... 262
10.8.9 Sampling time ... 262
10.9 Cautions for A/D Converter ... 263
10.9.1 Operating current in STOP mode ... 263
10.9.2 Input range of ANI0 to ANI6 pins ... 263
10.9.3 Conflicting operations ... 263
10.9.4 Noise countermeasures ... 263
10.9.5 Analog input (ANIn) pins ... 264
10.9.6 Input impedance of analog input (ANIn) pins ... 264
10.9.7 Interrupt request flag (ADIF) ... 265
10.9.8 Conversion results just after A/D conversion start ... 265
10.9.9 A/D conversion result register (ADCRH, ADCRL) read operation ... 265
10.9.10 Internal equivalent circuit ... 265
10.9.11 Starting the A/D converter ... 265

CHAPTER 11 COMPARATOR ... 266

11.1 Comparator Functions ... 266
11.2 Comparator Configuration ... 266
11.3 Registers Controlling the Comparator ... 268
11.3.1 Peripheral Enable Register 0 (PER0) ... 268
11.3.2 Comparator Mode Setting Register (COMPMDR) ... 269
11.3.3 Comparator Filter Control Register (COMPFIR) ... 270
11.3.4 Comparator Output Control Register (COMPOCR) ... 271
11.3.5 Registers Controlling Port Functions of Comparator I/O Pins ... 271
11.4 Comparator Operation ... 272

11.4.1 Comparator 0 Digital Filter Operation 273
11.4.2 Comparator 0 Interrupt Operation 273
11.4.3 Comparator 0 Output 273
11.5 Comparator Setting Flowchart 273
11.5.1 Enabling Comparator Operation 274
11.5.2 Disabling Comparator Operation 275

CHAPTER 12 SERIAL ARRAY UNIT 276

12.1 Functions of Serial Array Unit 277
12.1.1 Simplified SPI (CSI00, CSI01) 277
12.1.2 UART (UART0) 278
12.1.3 Simplified I^2C (IIC00) 279
12.2 Configuration of Serial Array Unit 280
12.2.1 Shift register 282
12.2.2 Serial data register 0nL (SDR0nL) 282
12.3 Registers Controlling Serial Array Unit 283
12.3.1 Peripheral enable register 0 (PER0) 284
12.3.2 Serial clock select register 0 (SPS0) 285
12.3.3 Serial mode register 0n (SMR0nH, SMR0nL) 286
12.3.4 Serial communication operation setting register 0n (SCR0nH, SCR0nL) 288
12.3.5 Serial data register 0n (SDR0nH, SDR0nL) 290
12.3.6 Serial flag clear trigger register 0n (SIR0n) 291
12.3.7 Serial status register 0n (SSR0n) 292
12.3.8 Serial channel start register 0 (SS0) 294
12.3.9 Serial channel stop register 0 (ST0) 295
12.3.10 Serial channel enable status register 0 (SE0) 296
12.3.11 Serial output enable register 0 (SOE0) 297
12.3.12 Serial output register 0 (SO0) 298
12.3.13 Serial clock output register 0 (CKO0) 299
12.3.14 Serial output level register 0 (SOL0) 300
12.3.15 Noise filter enable register 0 (NFEN0) 301
12.3.16 Input switch control register (ISC) 302
12.3.17 Registers controlling port functions of serial input/output pins 303
12.4 Operation Stop Mode 304
12.4.1 Stopping the operation by units 304
12.4.2 Stopping the operation by channels 305
12.5 Operation of Simplified SPI (CSI00, CSI01) Communication 306
12.5.1 Master transmission 307
12.5.2 Master reception 317
12.5.3 Master transmission/reception 326
12.5.4 Slave transmission 336

12.5.5 Slave reception 346
12.5.6 Slave transmission/reception 353
12.5.7 Calculating transfer clock frequency 363
12.5.8 Procedure for processing errors that occurred during simplified SPI (CSI00, CSI01) communication 365
12.6 Operation of UART (UART0) Communication 366
12.6.1 UART transmission 367
12.6.2 UART reception 377
12.6.3 Calculating baud rate 384
12.6.4 Procedure for processing errors that occurred during UART (UART0) communication 388
12.7 Operation of Simplified I²C (IIC00) Communication 389
12.7.1 Address field transmission 390
12.7.2 Data transmission 396
12.7.3 Data reception 400
12.7.4 Stop condition generation 405
12.7.5 Calculating transfer rate 406
12.7.6 Procedure for processing errors that occurred during simplified I²C (IIC00) communication 408

CHAPTER 13 SERIAL INTERFACE IICA 409

13.1 Functions of Serial Interface IICA 409
13.2 Configuration of Serial Interface IICA 412
13.3 Registers Controlling Serial Interface IICA 415
13.3.1 Peripheral enable register 0 (PER0) 416
13.3.2 IICA control register 00 (IICCTL00) 416
13.3.3 IICA status register 0 (IICS0) 421
13.3.4 IICA flag register 0 (IICF0) 423
13.3.5 IICA control register 01 (IICCTL01) 425
13.3.6 IICA low-level width setting register 0 (IICWL0) 427
13.3.7 IICA high-level width setting register 0 (IICWH0) 427
13.3.8 Registers controlling port functions of IICA serial input/output pins 428
13.4 I²C Bus Mode Functions 429
13.4.1 Pin configuration 429
13.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers 430
13.5 I²C Bus Definitions and Control Methods 431
13.5.1 Start conditions 431
13.5.2 Addresses 432
13.5.3 Transfer direction specification 432
13.5.4 Acknowledge ($\overline{ACK}$) 433
13.5.5 Stop condition 434
13.5.6 Clock stretch 435
13.5.7 Canceling clock stretch 437

13.5.8 Interrupt request (INTIICA0) generation timing and clock stretch control 438
13.5.9 Address match detection method 439
13.5.10 Error detection 439
13.5.11 Extension code 439
13.5.12 Arbitration 440
13.5.13 Wakeup function 442
13.5.14 Communication reservation 445
13.5.15 Cautions 449
13.5.16 Communication operations 450
13.5.17 Timing of I^2C interrupt request (INTIICA0) occurrence 460
13.6 Timing Charts 481

CHAPTER 14 INTERRUPT FUNCTIONS 496

14.1 Interrupt Function Types 496
14.2 Interrupt Sources and Configuration 496
14.3 Registers Controlling Interrupt Functions 501
14.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L) 503
14.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L) 505
14.3.3 Priority specification flag registers (PR00L, PR00H, PR10L, PR10H, PR01L, PR11L) 506
14.3.4 External interrupt rising edge enable register 0 (EGP0),
external interrupt falling edge enable register 0 (EGN0) 508
14.3.5 Program status word (PSW) 509
14.4 Interrupt Servicing Operations 510
14.4.1 Maskable interrupt request acknowledgment 510
14.4.2 Software interrupt request acknowledgment 512
14.4.3 Multiple interrupt servicing 512
14.4.4 Interrupt request hold 516

CHAPTER 15 KEY INTERRUPT FUNCTION 517

15.1 Functions of Key Interrupt 517
15.2 Configuration of Key Interrupt 517
15.3 Register Controlling Key Interrupt 519
15.3.1 Key interrupt control register (KRCTL) 519
15.3.2 Key return mode register (KRM0) 520
15.3.3 Key return flag register (KRF) 521
15.3.4 Registers controlling port functions of key interrupt input pins 521
15.4 Key Interrupt Operation 522
15.4.1 When not using the key interrupt flag (KRMD = 0) 522
15.4.2 When using the key interrupt flag (KRMD = 1) 523

CHAPTER 16 STANDBY FUNCTION 525

16.1 Overview 525

16.2 Registers controlling standby function 526

16.3 Standby Function Operation 526

16.3.1 HALT mode 526

16.3.2 STOP mode 530

CHAPTER 17 RESET FUNCTION 534

17.1 Timing of Reset Operation 536

17.2 States of Operation During Reset Periods 538

17.3 Register for Confirming Reset Source 540

17.3.1 Reset Control Flag Register (RESF) 540

CHAPTER 18 SELECTABLE POWER-ON-RESET CIRCUIT 542

18.1 Functions of Selectable Power-on-reset Circuit 542

18.2 Configuration of Selectable Power-on-reset Circuit 542

18.3 Operation of Selectable Power-on-reset Circuit 543

18.4 Cautions for Selectable Power-on-reset Circuit 544

CHAPTER 19 OPTION BYTE 545

19.1 Functions of Option Bytes 545

19.1.1 User option byte (000C0H to 000C2H) 545

19.1.2 On-chip debug option byte (000C3H) 545

19.2 Format of User Option Byte 546

19.3 Format of On-chip Debug Option Byte 548

19.4 Setting of Option Byte 549

CHAPTER 20 FLASH MEMORY 550

20.1 Serial Programming by Using Flash Memory Programmer 551

20.1.1 Programming environment 552

20.1.2 Communication mode 552

20.2 Writing to Flash Memory by Using External Device (that Incorporates UART) 553

20.2.1 Programming Environment 553

20.2.2 Communication Mode 553

20.3 Connection of Pins on Board 554

20.3.1 P40/TOOL0 pin 554

20.3.2 $\overline{\text{RESET}}$ pin 554

20.3.3 Port pins 555

20.3.4 X1 and X2 pins (16-pin products only) 555

20.3.5 Power supply 555

20.4 Serial Programming Method 556
20.4.1 Serial programming procedure 556
20.4.2 Flash memory programming mode 557
20.4.3 Communication mode 558
20.4.4 Communication commands 558
20.5 Processing Time for Each Command When Dedicated Flash Memory Programmer is in Use (Reference Value) 558

CHAPTER 21 ON-CHIP DEBUG FUNCTION 560

21.1 Connecting E1, E2, E2 Lite, E20 On-chip Debugging Emulator 560
21.2 On-Chip Debug Security ID 561
21.3 Securing of User Resources 562

CHAPTER 22 BCD CORRECTION CIRCUIT 563

22.1 BCD Correction Circuit Function 563
22.2 Registers Used by BCD Correction Circuit 563
22.2.1 BCD correction result register (BCDADJ) 563
22.3 BCD Correction Circuit Operation 564

CHAPTER 23 INSTRUCTION SET 566

23.1 Conventions Used in Operation List 567
23.1.1 Operand identifiers and specification methods 567
23.1.2 Description of operation column 568
23.1.3 Description of flag operation column 569
23.1.4 PREFIX instruction 569
23.2 Operation List 570

CHAPTER 24 ELECTRICAL SPECIFICATIONS 587

24.1 Absolute Maximum Ratings 588
24.2 Oscillator Characteristics 589
24.2.1 X1 oscillator characteristics 589
24.2.2 On-chip oscillator characteristics 589
24.3 DC Characteristics 590
24.3.1 Pin characteristics 590
24.3.2 Supply current characteristics 592
24.4 AC Characteristics 595
24.5 Serial Interface Characteristics 598
24.5.1 Serial array unit 598
24.5.2 Serial interface IICA 602
24.6 Analog Characteristics 603

24.6.1 A/D converter characteristics 603
24.6.2 Comparator characteristics 604
24.6.3 Internal reference voltage characteristics 604
24.6.4 SPOR circuit characteristics 605
24.6.5 Power supply voltage rising slope characteristics 605
24.7 RAM Data Retention Characteristics 605
24.8 Flash Memory Programming Characteristics 606
24.9 Dedicated Flash Memory Programmer Communication (UART) 606
24.10 Timing of Entry to Flash Memory Programming Modes 607

CHAPTER 25 PACKAGE DRAWINGS 608

25.1 10-pin products 608
25.2 16-pin products 609

APPENDIX A REVISION HISTORY 610

A.1 Major Revisions in This Edition 610
A.2 Revision History of Preceding Editions 612

CHAPTER 1 OUTLINE

1.1 Features

Ultra-low power consumption technology

- V_{DD} = single power supply voltage of 2.0 to 5.5 V
  (Use this product within the voltage range from 2.25 to 5.5 V because the detection voltage (V_{SPOR}) of the selectable power-on-reset (SPOR) circuit should also be considered.)
- HALT mode
- STOP mode

RL78 CPU core

- CISC architecture with 3-sage pipeline
- Minimum instruction execution time: Can be changed from high speed (0.05 μs: @ 20 MHz operation with high-speed on-chip oscillator) to low speed (1.0 μs: @ 1 MHz operation)
- Address space: 1 MB
- General-purpose registers: 8-bit register × 8
- On-chip RAM: 128 to 512 B

Code flash memory

- Code flash memory: 1 to 4 KB
- On-chip debug function

High-speed on-chip oscillator

- Select from 20 MHz, 10 MHz, 5 MHz, 2.5 MHz, and 1.25 MHz
- High accuracy: ±2.0 % (V_{DD} = 2.0 to 5.5 V, T_A = -20 to +85°C)

Operating ambient temperature

- T_A = -40 to +85°C

Power management and reset function

- On-chip selectable power-on-reset (SPOR) circuit

Serial interface

- Simplified SPI (CSI [Note 1]): 1/2 [Note 2] channels
- UART: 1 channel
- Simplified I^2C communication: 1 channel
- I^2C communication: 1 channel [Note 2]

Timer

- 8-/16-bit timer: 2/4 [Note 2] channels
- 12-bit interval timer [Note 2]: 1 channel
- Watchdog timer: 1 channel (operable with the dedicated low-speed on-chip oscillator)

A/D converter

- 8/10-bit resolution A/D converter (V_{DD} = 2.4 to 5.5 V)
- Analog input: 4/7 Note 2 channels
- Internal reference voltage (0.815 V (typ.)) Note 2

Comparator Note 2

- 1 channel
- Operation mode: High-speed mode, low-speed mode
- External reference voltage or internal reference voltage can be selected as the reference voltage.

I/O port

- I/O port: 8/14 Note 2 (N-ch open drain output [V_{DD} withstand voltage]: 2/4 Note 2)
- Can be set to N-ch open drain and on-chip pull-up resistor
- On-chip key interrupt function
- On-chip clock output/buzzer output controller

Others

- On-chip BCD (binary-coded decimal) correction circuit

ROM, RAM capacities

Flash ROM	RAM	10 pins	16 pins
4 KB	512 B	R5F10Y17	R5F10Y47
2 KB	256 B	R5F10Y16	R5F10Y46
1 KB	128 B	R5F10Y14	R5F10Y44

<R> **Note 1.** Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

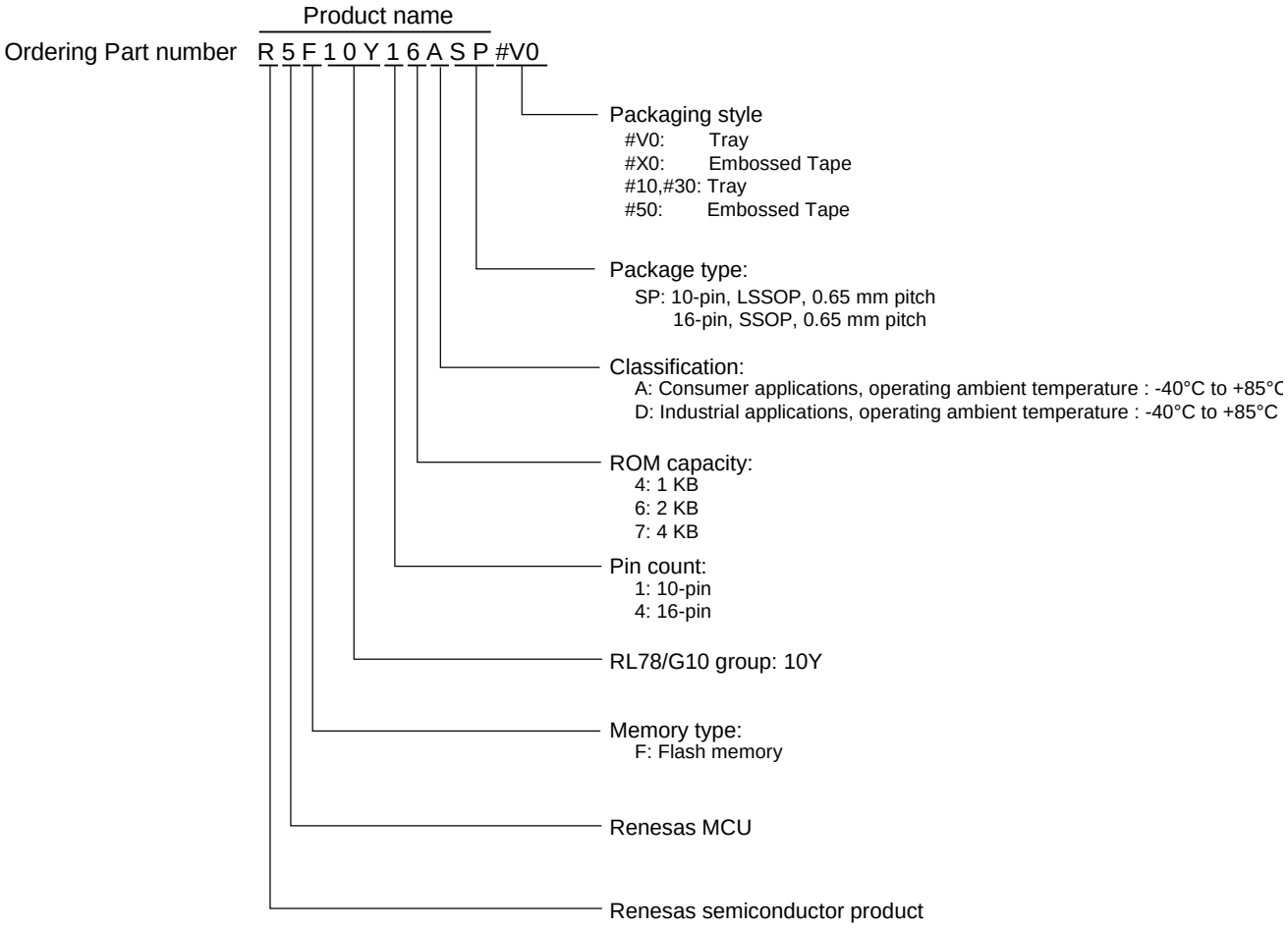
Note2. 16-pin products only

Remark The functions mounted depend on the product. See **1.6 Outline of Functions**.

1.2 List of Part Numbers

<R>

Figure 1-1. Part Number, Memory Size, and Package of RL78/G10



<R>

Table 1-1. List of Ordering Part Numbers

Pin count	Package	Fields of Application Note	Ordering Part Number		RENESAS Code
			Product Name	Packaging Specifications	
10 pins	10-pin plastic LSSOP (4.4 × 3.6 mm, 0.65 mm pitch)	A	R5F10Y14ASP, R5F10Y14ASP R5F10Y16ASP, R5F10Y16ASP	#V0,#X0	PLSP0010JA-A
			R5F10Y17ASP, R5F10Y17ASP	#10,#30,#50	
		D	R5F10Y14DSP, R5F10Y14DSP R5F10Y16DSP, R5F10Y16DSP R5F10Y17DSP, R5F10Y17DSP	#10,#30,#50	
16 pins	16-pin plastic SSOP (4.4 × 5.0 mm, 0.65 mm pitch)	A	R5F10Y44ASP, R5F10Y44ASP R5F10Y46ASP, R5F10Y46ASP R5F10Y47ASP, R5F10Y47ASP	#10,#30,#50	PRSP0016JC-B
		D	R5F10Y44DSP, R5F10Y44DSP R5F10Y46DSP, R5F10Y46DSP R5F10Y47DSP, R5F10Y47DSP		

(**Note** and **Caution** are listed on the next page.)

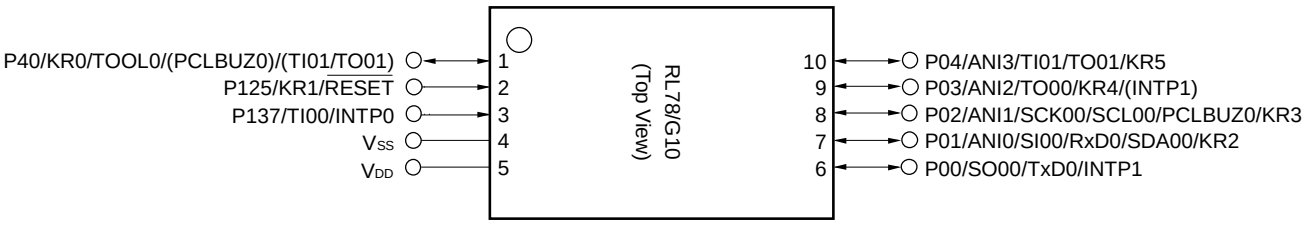
Note For the fields of application, refer to **Figure 1-1 Part Number, Memory Size, and Package of RL78/G10**.

Caution The part number represents the number at the time of publication.
Be sure to review the latest part number through the target product page in the Renesas Electronics Corp.website.

1.3 Pin Configuration (Top View)

1.3.1 10-pin products

- 10-pin plastic LSSOP (4.4 × 3.6 mm, 0.65 mm pitch)



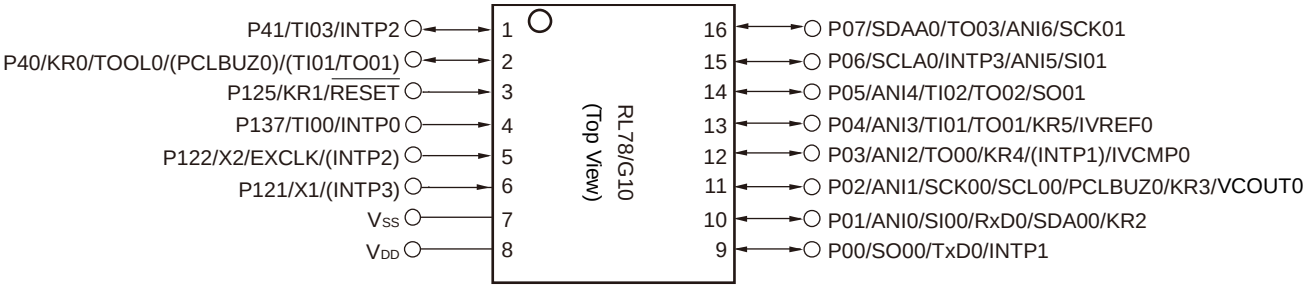
<R> **Table 1-2. Alternate Function of 10-pin products**

Pin No.	I/O	Power supply, system clock, debug	Analog		HMI		Timer	Communication interface	
10LSSOP	Digital port		A/D converter	Comparator	Interrupt function	key interrupt function (KR)	Timer array unit	Serial array unit	Serial interface IICA
1	P40	TOOL0 (PCLBUZ0)				KR0	(TI01/TO01)		
2	P125	$\overline{\text{RESET}}$				KR1			
3	P137				INTP0		TI00		
4		V_{SS}							
5		V_{DD}							
6	P00				INTP1			SO00/TxD0	
7	P01		ANI0			KR2		SI00/RxD0/SDA00	
8	P02	PCLBUZ0	ANI1			KR3		SCK00/SCL00	
9	P03		ANI2		(INTP1)	KR4	TO00		
10	P04		ANI3			KR5	TI01/TO01		

Remarks 1. For pin identification, see **1.4 Pin Identification**.
2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). See **Figure 4-6 Format of Peripheral I/O Redirection Register (PIOR)**.

1.3.2 16-pin products

- 16-pin plastic SSOP (4.4 × 5.0 mm, 0.65 mm pitch)



<R>

Table 1-3. Alternate Function of 16-pin products

Pin No.	I/O	Power supply, system clock, debug	Analog		HMI		Timer	Communication interface	
16SSOP	Digital port		A/D converter	Comparator	Interrupt function	Key interrupt function	Timer array unit	Serial array unit	Serial interface IICA
1	P41				INTP2		TI03		
2	P40	TOOL0 (PCLBUZ0)				KR0	(TI01/TO01)		
3	P125	$\overline{\text{RESET}}$				KR1			
4	P137				INTP0		TI00		
5	P122	X2 EXCLK			(INTP2)				
6	P121	X1			(INTP3)				
7		V_{SS}							
8		V_{DD}							
9	P00				INTP1			SO00/TxD0	
10	P01		ANI0			KR2		SI00/RxD0/SDA00	
11	P02	PCLBUZ0	ANI1	VCOUT0		KR3		SCK00/SCL00	
12	P03		ANI2	IVCMP0	(INTP1)	KR4	TO00		
13	P04		ANI3	IVREF0		KR5	TI01/TO01		
14	P05		ANI4				TI02/TO02	SO01	
15	P06		ANI5		INTP3			SI01	SCLA0
16	P07		ANI6				TO03	SCK01	SDAA0

Remarks 1. For pin identification, see **1.4 Pin Identification**.

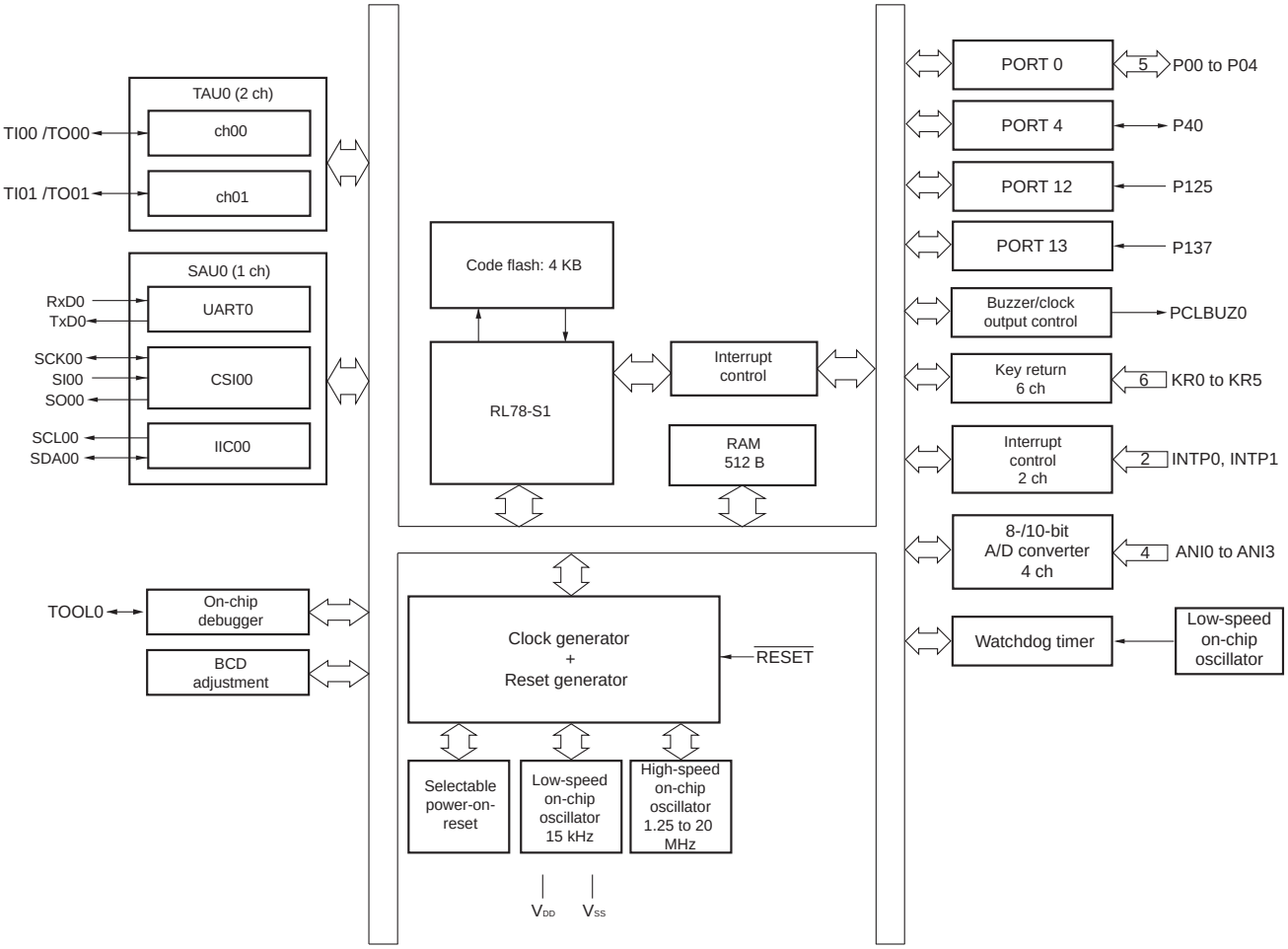
2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). See **Figure 4-6 Format of Peripheral I/O Redirection Register (PIOR)**.

1.4 Pin Identification

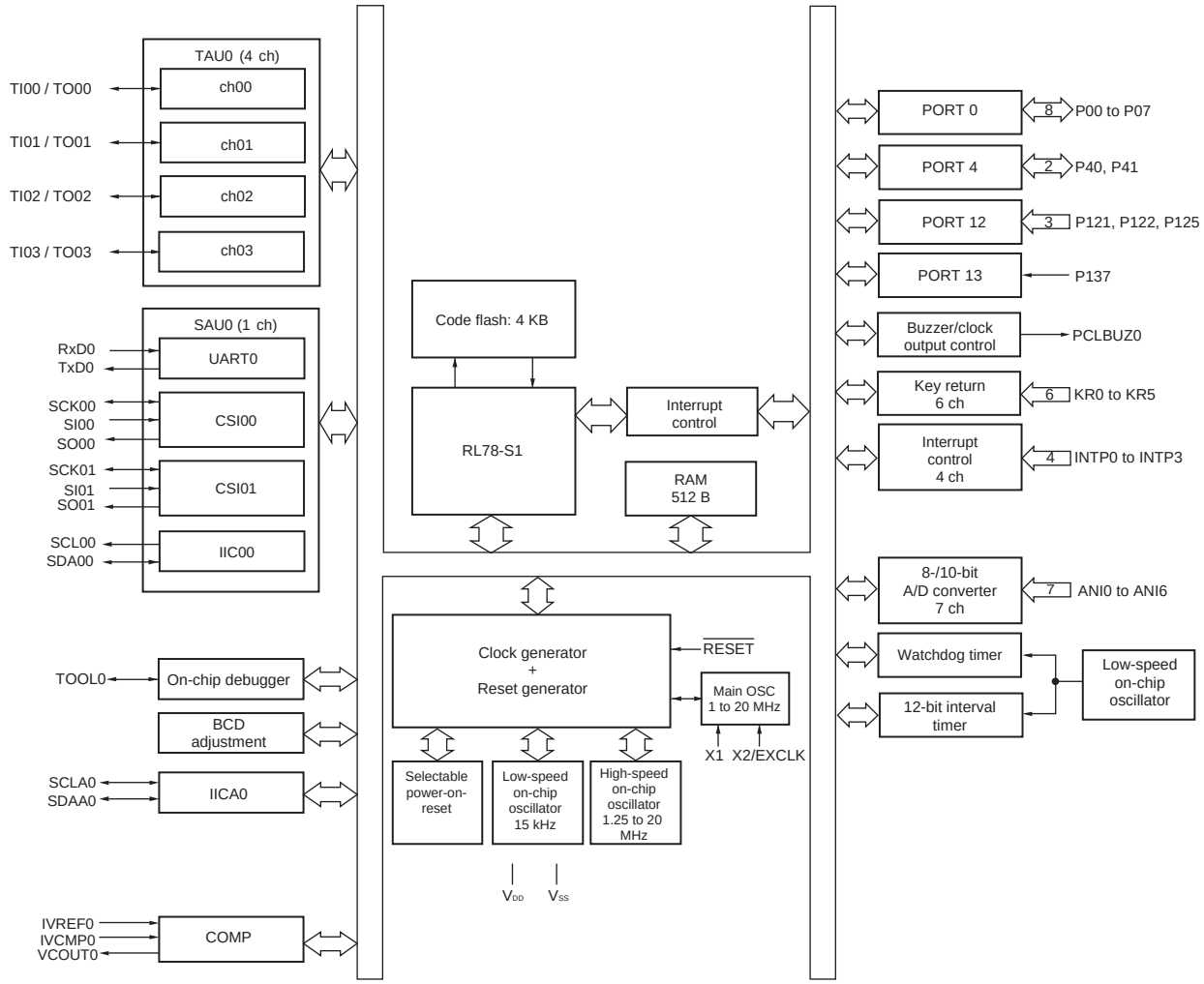
ANI0 to ANI6	: Analog Input
INTP0 to INTP3	: Interrupt Request From Peripherals
KR0 to KR5	: Key Return
P00 to P07	: Port 0
P40, P41	: Port 4
P121, P122, P125	: Port 12
P137	: Port 13
PCLBUZ0	: Programmable Clock Output/ Buzzer Output
EXCLK	: External Clock Input
X1, X2	: Crystal Oscillator (Main System Clock)
IVCMP0	: Comparator Input
VCOUT0	: Comparator Output
IVREF0	: Comparator Reference Input
$\overline{\text{RESET}}$	: Reset
RxD0	: Receive Data
SCK00, SCK01	: Serial Clock Input/Output
SCL00, SCLA0	: Serial Clock Output
SDA00, SDAA0	: Serial Data Input/Output
SI00, SI01	: Serial Data Input
SO00, SO01	: Serial Data Output
TI00 to TI03	: Timer Input
TO00 to TO03	: Timer Output
TOOL0	: Data Input/Output for Tool
TxD0	: Transmit Data
V_{DD}	: Power Supply
V_{SS}	: Ground

1.5 Block Diagram

1.5.1 10-pin products



1.5.2 16-pin products



1.6 Outline of Functions

This outline describes the function at the time when Peripheral I/O redirection register (PIOR) is set to 00H.

Item		10-pin			16-pin		
		R5F10Y14	R5F10Y16	R5F10Y17	R5F10Y44	R5F10Y46	R5F10Y47
Code flash memory		1 KB	2 KB	4 KB	1 KB	2 KB	4 KB
RAM		128 B	256 B	512 B	128 B	256 B	512 B
Main system clock	High-speed system clock	—			X1, X2 (crystal/ceramic) oscillation, external main system clock input (EXCLK): 1 to 20 MHz: V_{DD} = 2.7 to 5.5 V 1 to 5 MHz: V_{DD} = 2.0 to 5.5 V Note 3		
	High-speed on-chip oscillator clock	• 1.25 to 20 MHz (V_{DD} = 2.7 to 5.5 V) • 1.25 to 5 MHz (V_{DD} = 2.0 to 5.5 V Note 3)					
Low-speed on-chip oscillator clock		15 kHz (TYP)					
General-purpose register		8-bit register × 8					
Minimum instruction execution time		0.05 μs (20 MHz operation)					
Instruction set		• Data transfer (8 bits) • Adder and subtractor/logical operation (8 bits) • Multiplication (8 bits × 8 bits) • Rotate, barrel shift, and bit manipulation (set, reset, test, and Boolean operation), etc.					
I/O port	Total	8			14		
	CMOS I/O	6 (N-ch open-drain output (V_{DD} tolerance): 2)			10 (N-ch open-drain output (V_{DD} tolerance): 4)		
	CMOS input	2			4		
Timer	16-bit timer	2 channels			4 channels		
	Watchdog timer	1 channel					
	12-bit interval timer	—			1 channel		
	Timer output	2 channels (PWM output: 1)			4 channels (PWM outputs: 3 Note 1)		
Clock output/buzzer output		1					
		2.44 kHz to 10 MHz: (Peripheral hardware clock: f_{MAIN} = 20 MHz operation)					
Comparator		—			1		
8-/10-bit resolution A/D converter		4 channels			7 channels		
Serial interface		[10-pin products] Simplified SPI (CSI): 1 channel/simplified I²C: 1 channel/UART: 1 channel [16-pin products] Simplified SPI (CSI): 2 channels/simplified I²C: 1 channel/UART: 1 channel					
	I²C bus	—			1 channel		
Vectored interrupt sources	Internal	8			14		
	External	3			5		
Key interrupt		6					
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by selectable power-on-reset • Internal reset by illegal instruction execution Note 2 • Internal reset by data retention lower limit voltage					
Selectable power-on-reset circuit		• Detection voltage Rising edge (V_{SPOR}): 2.25 V/2.68 V/3.02 V/4.45 V (max.) Falling edge (V_{SPDR}): 2.20 V/2.62 V/2.96 V/4.37 V (max.)					

Item	10-pin			16-pin		
	R5F10Y14	R5F10Y16	R5F10Y17	R5F10Y44	R5F10Y46	R5F10Y47
On-chip debug function	Provided					
Power supply voltage	V_{DD} = 2.0 to 5.5 V Note 3					
Operating ambient temperature	T_A = - 40 to + 85 °C					

Notes 1. The number of outputs varies, depending on the setting of channels in use and the number of the master (see **6.9.4 Operation as multiple PWM output function**).

2. The illegal instruction is generated when instruction code FFH is executed. Reset by the illegal instruction execution not issued by emulation with the on-chip debug emulator.

3. Use this product within the voltage range from 2.25 to 5.5 V because the detection voltage (V_{SPOR}) of the selectable power-on-reset (SPOR) circuit should also be considered.

CHAPTER 2 PIN FUNCTIONS

2.1 Port Functions

The input or output, buffer, and pull-up resistor settings are also valid for the alternate functions.

2.1.1 10-pin products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-1-2	I/O	Input port	SO00/TxD0/INTP1	Port 0. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P00 and P01 can be set to N-ch open-drain output (V_{DD} tolerance). P01 to P04 can be set to analog input Note.
P01	7-3-2		Analog input port	ANI0/SI00/RxD0/ SDA00/KR2	
P02	7-3-1			ANI1/SCK00/SCL00/ PCLBUZ0/KR3	
P03				ANI2/TO00/KR4/ (INTP1)	
P04				ANI3/TI01/TO01/KR5	
P40	7-1-1	I/O	Input port	KR0/TOOL0/ (PCLBUZ0)/ (TI01/TO01)	Port 4 1-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P125	3-1-1	Input	Input port	KR1/$\overline{\text{RESET}}$	Port 12 1-bit input port. Use of an on-chip pull-up resistor can be specified by a software setting. P125 is also used for the input pin for external reset ($\overline{\text{RESET}}$). To use the pin for external reset, set the PORTSELB bit in the option byte (000C1H) to 1.
P137	2-1-2	Input	Input port	TI00/INTP0	Port 13 1-bit input only port.

Note Setting digital or analog to each pin can be done in the port mode control register 0 (PMC0) (can be set in 1-bit units).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Register (PIOR)**.

2.1.2 16-pin products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-1-2	I/O	Input port	SO00/TxD0/INTP1	Port 0. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P00, P01, P06, and P07 can be set to N-ch open-drain output (V_{DD} tolerance). P01 to P07 can be set to analog input Note.
P01	7-3-2		Analog input port	ANI0/SI00/RxD0/ SDA00/KR2	
P02	7-3-1			ANI1/SCK00/SCL00/ PCLBUZ0/KR3/ VCOUT0	
P03	7-9-1			ANI2/TO00/KR4/ (INTP1)/IVCMP0	
P04				ANI3/TI01/TO01/KR5/ IVREF0	
P05	7-3-1			ANI4/TI02/TO02/SO01	
P06	7-3-2			ANI5/SCLA0/INTP3/ SI01	
P07				ANI6/SDAA0/TO03/ SCK01	
P40	7-1-1	I/O	Input port	KR0/TOOL0/ (PCLBUZ0)/ (TI01/TO01)	Port 4 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P41				TI03/INTP2	
P121	2-2-1	Input	Input port	X1/(INTP3)	Port 12 3-bit input port. For P125, use of an on-chip pull-up resistor can be specified by a software setting. P125 is also used for the input pin for external reset ($\overline{\text{RESET}}$). To use the pin for external reset, set the PORTSELB bit in the option byte (000C1H) to 1.
P122				X2/EXCLK/(INTP2)	
P125	3-1-1			KR1/$\overline{\text{RESET}}$	
P137	2-1-2	Input	Input port	TI00/INTP0	Port 13 1-bit input only port.

Note Setting digital or analog to each pin can be done in the port mode control register 0 (PMC0) (can be set in 1-bit units).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Register (PIOR)**.

2.2 Functions other than port pins

2.2.1 Functions for each product

Function Name	16-pin products	10-pin products
ANI0	√	√
ANI1	√	√
ANI2	√	√
ANI3	√	√
ANI4	√	—
ANI5	√	—
ANI6	√	—
IVCMP0	√	—
IVREF0	√	—
VCOUT0	√	—
INTP0	√	√
INTP1	√	√
INTP2	√	—
INTP3	√	—
KR0	√	√
KR1	√	√
KR2	√	√
KR3	√	√
KR4	√	√
KR5	√	√
PCLBUZ0	√	√
TOOL0	√	√
$\overline{\text{RESET}}$	√	√
X1	√	—
X2	√	—
EXCLK	√	—
V_{DD}	√	√
V_{SS}	√	√

Function Name	16-pin products	10-pin products
RxD0	√	√
TxD0	√	√
SCL00	√	√
SDA00	√	√
SCK00	√	√
SCK01	√	—
SI00	√	√
SI01	√	—
SO00	√	√
SO01	√	—
SCLA0	√	—
SDAA0	√	—
TI00	√	√
TO00	√	√
TI01	√	√
TO01	√	√
TI02	√	—
TO02	√	—
TI03	√	—
TO03	√	—

2.2.2 Description of functions

Function Name	I/O	Functions
ANI0 to ANI6	input	Analog input pins of A/D converter (See **Figure 10-23 Analog Input Pin Connection**.)
IVCOUT0	output	Comparator output
IVCMP0	input	Analog input for the comparator
IVREF0	input	Reference voltage input for the comparator
INTP0 to INTP3	input	External interrupt request input Specified available edge: Rising edge, falling edge, or both rising and falling edges
KR0 to KR5	input	Key interrupt input Specified available edge: Rising edge or falling edge
PCLBUZ0	output	Clock/buzzer output
$\overline{\text{RESET}}$	input	This is the active-low system reset input pin. When the external reset pin is not used, connect this pin directly or via a resistor to V_{DD}.
RxD0	input	Serial data input pin of serial interface UART0
TxD0	output	Serial data output pin of serial interface UART0
SCK00, SCK01	I/O	Serial clock I/O pins of serial interface CSI00 and CSI01
SI00, SI01	input	Serial data input pins of serial interface CSI00 and CSI01
SO00, SO01	output	Serial data output pins of serial interface CSI00 and CSI01
SCL00	output	Serial clock output pin of serial interface simple I^2C (IIC00)
SDA00	I/O	Serial data I/O pin of serial interface simple I^2C (IIC00)
SCLA0	I/O	Clock I/O pin of serial interface IICA0
SDAA0	I/O	Serial data I/O pin of serial interface IICA0
TI00 to TI03	input	Inputting an external count clock/capture trigger to 16-bit timers 00 to 03
TO00 to TO03	output	Timer output pins of 16-bit timers 00 to 03
X1, X2	–	Resonator connection for main system clock
EXCLK	input	External clock input for main system clock
V_{DD}	–	Positive power supply
V_{SS}	–	Ground potential
TOOL0	I/O	Data I/O pin for a flash memory programmer/debugger

Caution After reset release, the relationships between P40/TOOL0 and the operating mode are as follows.

Table 2-1. Relationships Between P40/TOOL0 and Operation Mode After Reset Release

P40/TOOL0	Operating mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

For details, see 20.4.2 Flash memory programming mode.

Remark Use bypass capacitors (about 0.1 μF) as noise and latch up countermeasures with relatively thick wires at the shortest distance to V_{DD} to V_{SS} lines.

2.3 Connection of Unused Pins

Table 2-2 shows the connections of unused pins.

Remark The pins mounted depend on the product. Refer to **1.3 Pin Configuration (Top View)** and **2.1 Port Functions**.

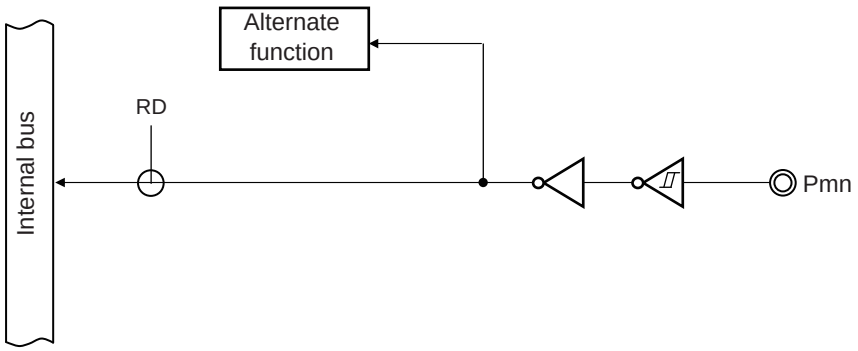
Table 2-2. Connection of Unused Pins

Pin Name	I/O	Recommended Connection of Unused Pins
P00 to P07	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P40/TOOL0		Input: Independently connect to V_{DD} via a resistor. Output: Leave open.
P41		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121, P122	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P125/$\overline{\text{RESET}}$	Input	Leave open, or connect to V_{DD} while PORTSELB = 1.
P137	Input	Independently connect to V_{DD} or V_{SS} via a resistor.

2.4 Block Diagrams of Pins

Figures 2-1 to 2-8 show the block diagrams of the pins described in **2.1.1 10-pin products** and **2.1.2 16-pin products**.

Figure 2-1. Pin Block Diagram for Pin Type 2-1-2



Remark For alternate functions, see **2.1 Port Functions**.

Figure 2-2. Pin Block Diagram for Pin Type 2-2-1

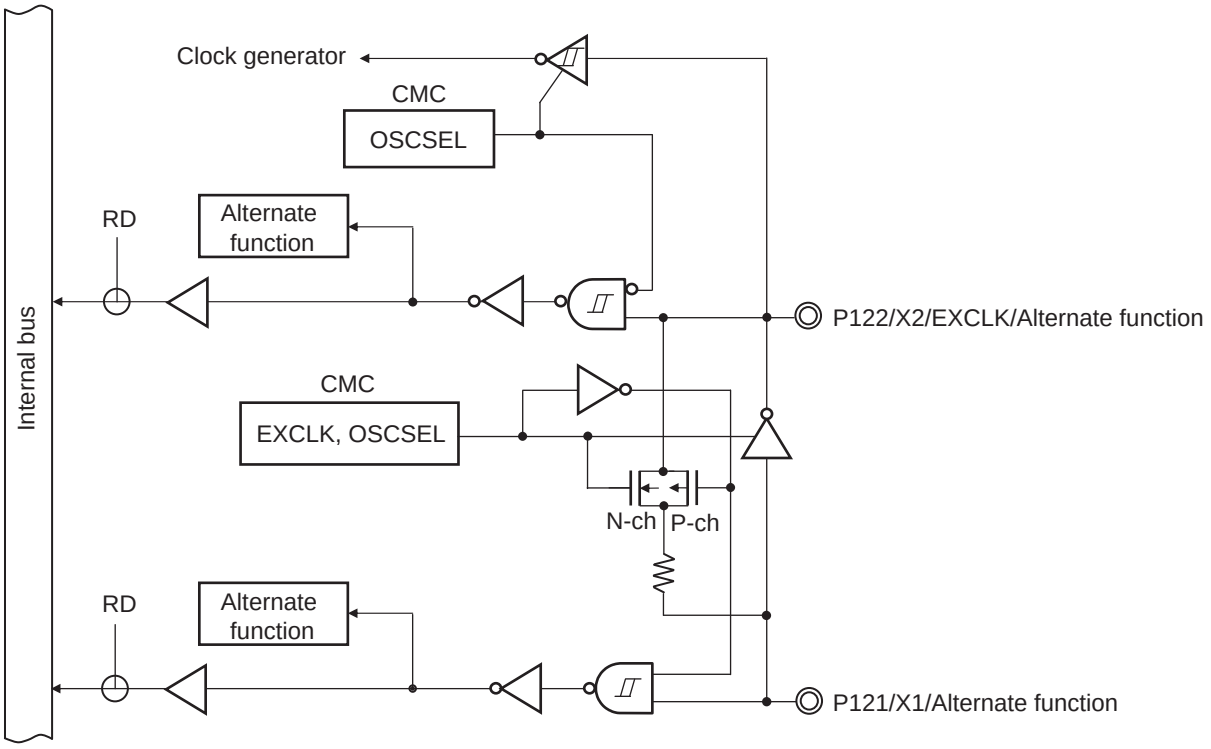
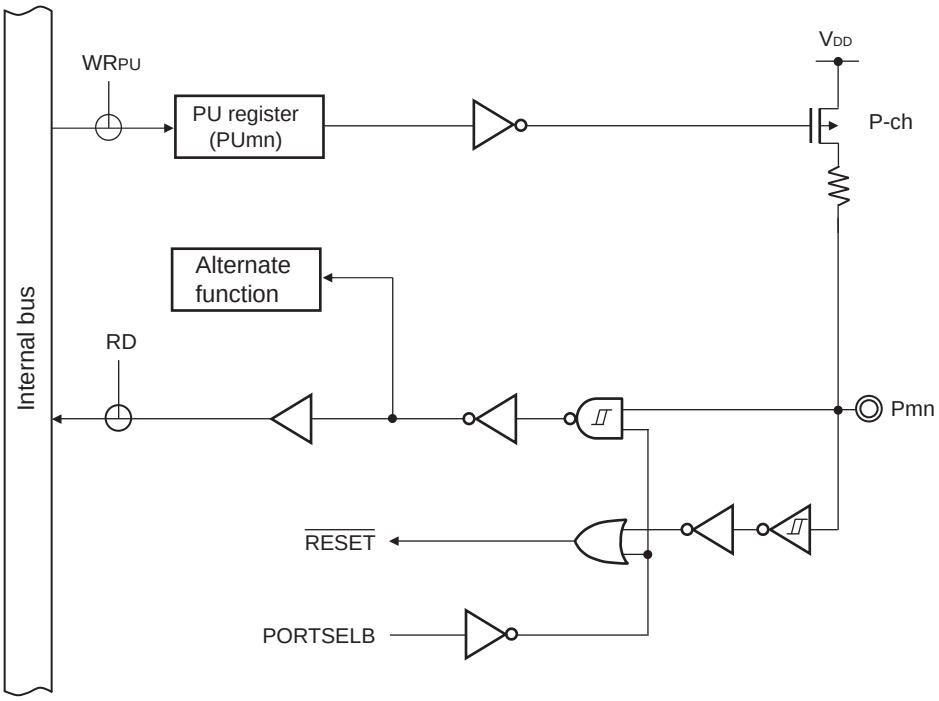
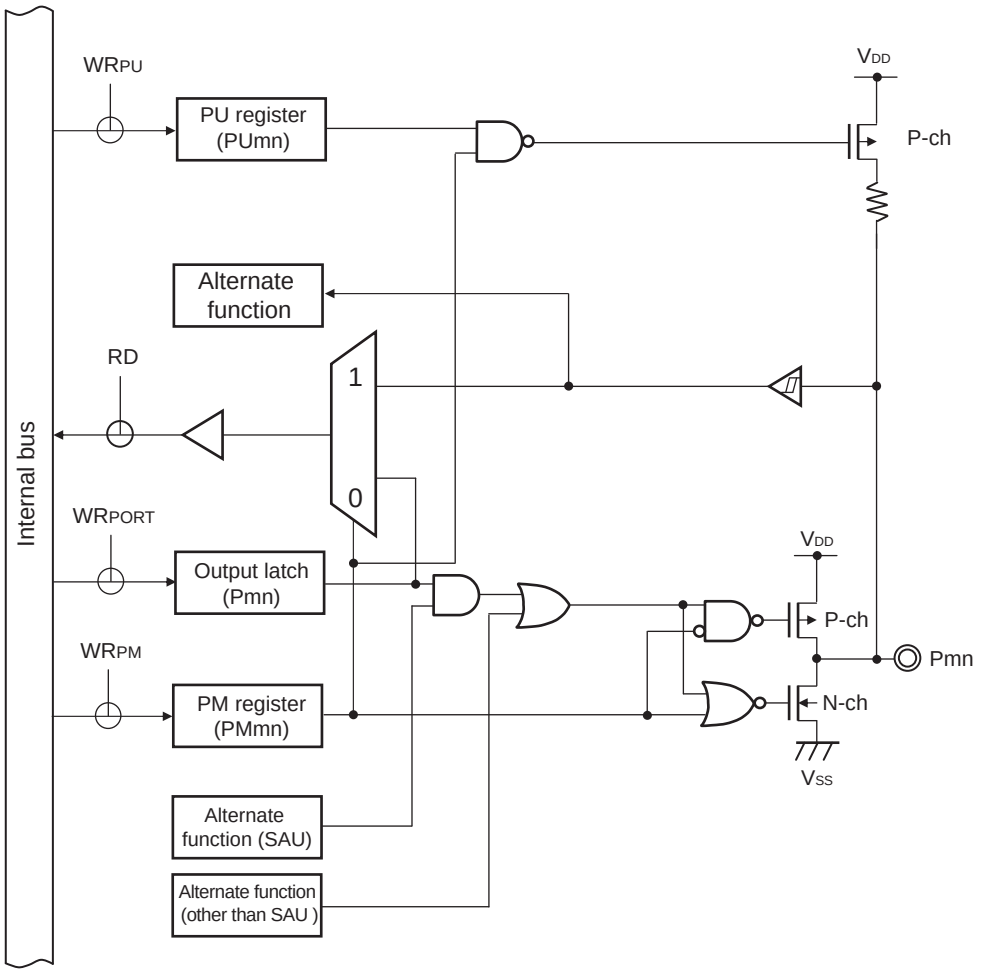


Figure 2-3. Pin Block Diagram for Pin Type 3-1-1



Remark For alternate functions, see **2.1 Port Functions**.

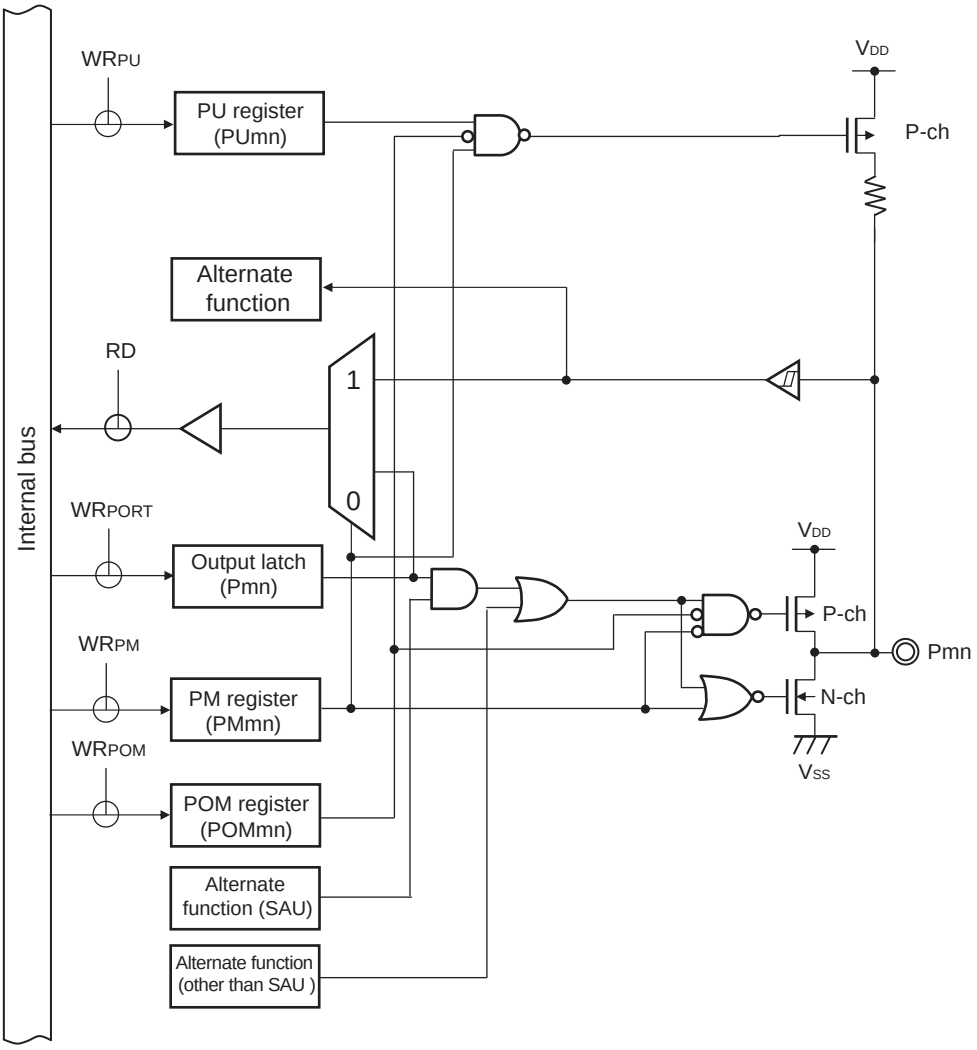
Figure 2-4. Pin Block Diagram for Pin Type 7-1-1



Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

Figure 2-5. Pin Block Diagram for Pin Type 7-1-2

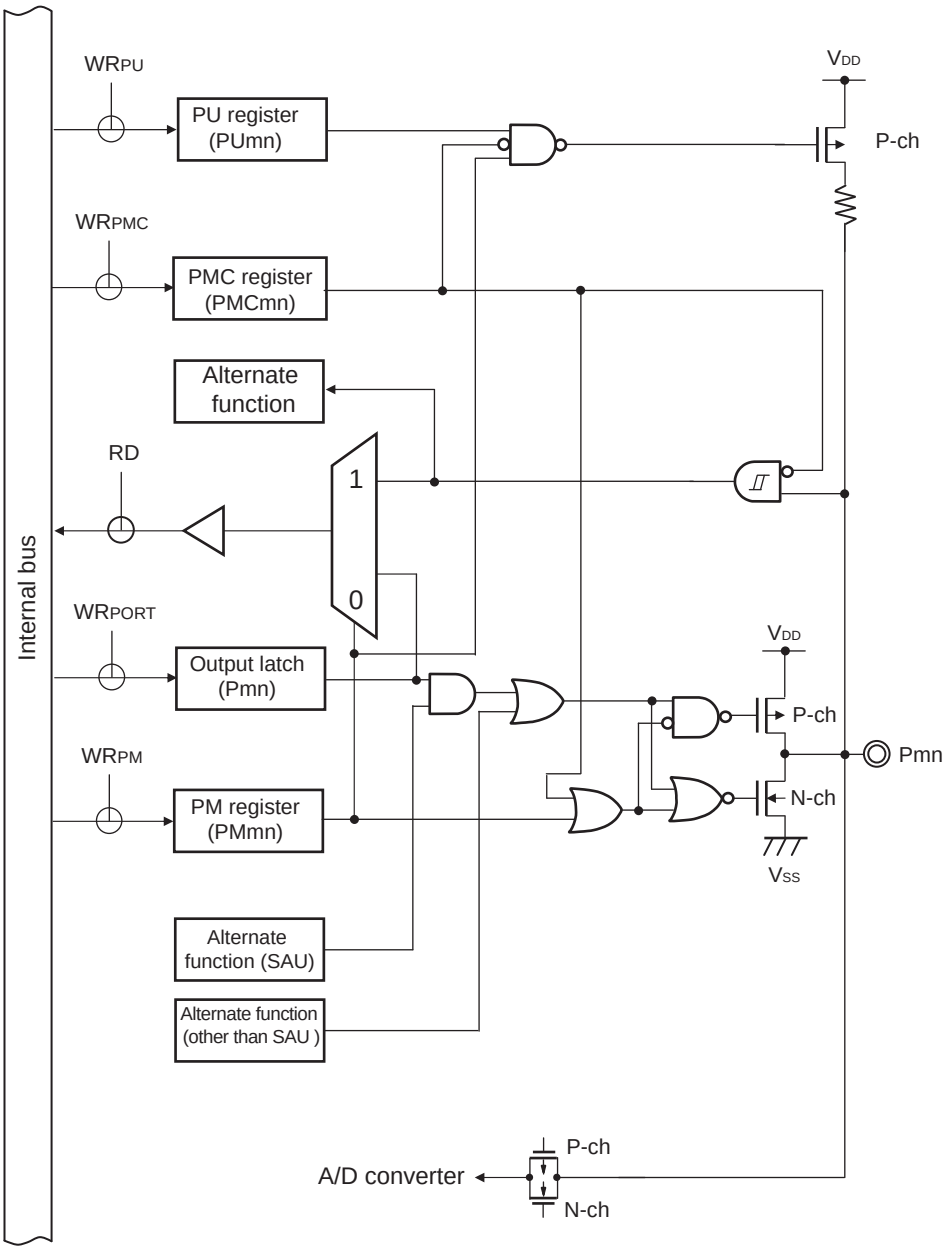


Caution The input buffer is enabled even if the type 7-1-2 pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMxx). This may lead to a through current flowing through the type 7-1-2 pin when the voltage level on this pin is intermediate. Changing the output level when the N-ch open drain output mode is selected may cause a glitch (V_{DD} level).

Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

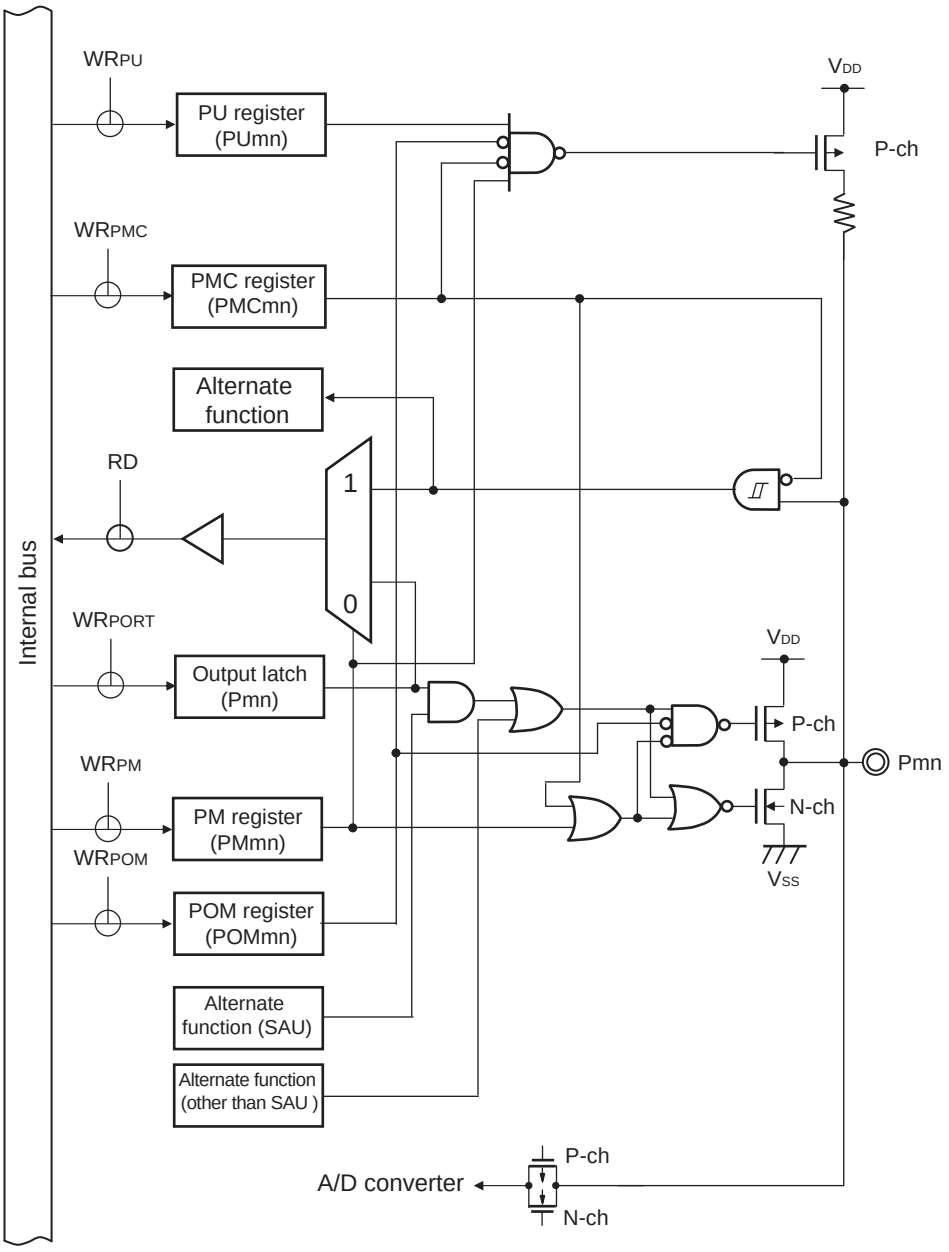
Figure 2-6. Pin Block Diagram for Pin Type 7-3-1



Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

Figure 2-7. Pin Block Diagram for Pin Type 7-3-2

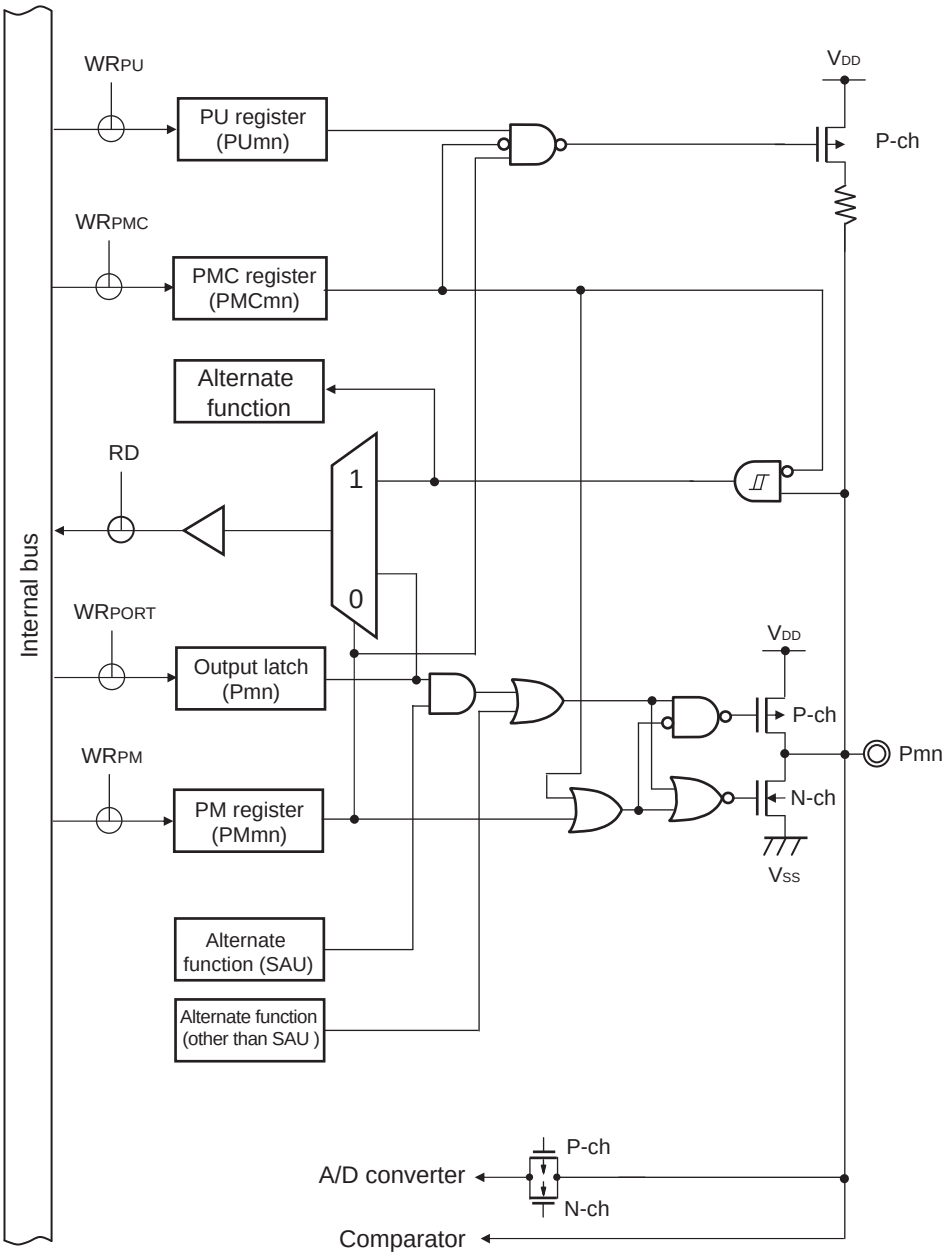


Caution The input buffer is enabled even if the type 7-3-2 pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMxx). This may lead to a through current flowing through the type 7-3-2 pin when the voltage level on this pin is intermediate. Changing the output level when the N-ch open drain output mode is selected may cause a glitch (V_{DD} level).

Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

Figure 2-8. Pin Block Diagram for Pin Type 7-9-1



Remarks 1. For alternate functions, see **2.1 Port Functions**.

2. SAU: Serial array unit

CHAPTER 3 CPU ARCHITECTURE

The RL78/G10 has the RL78-S1 core.

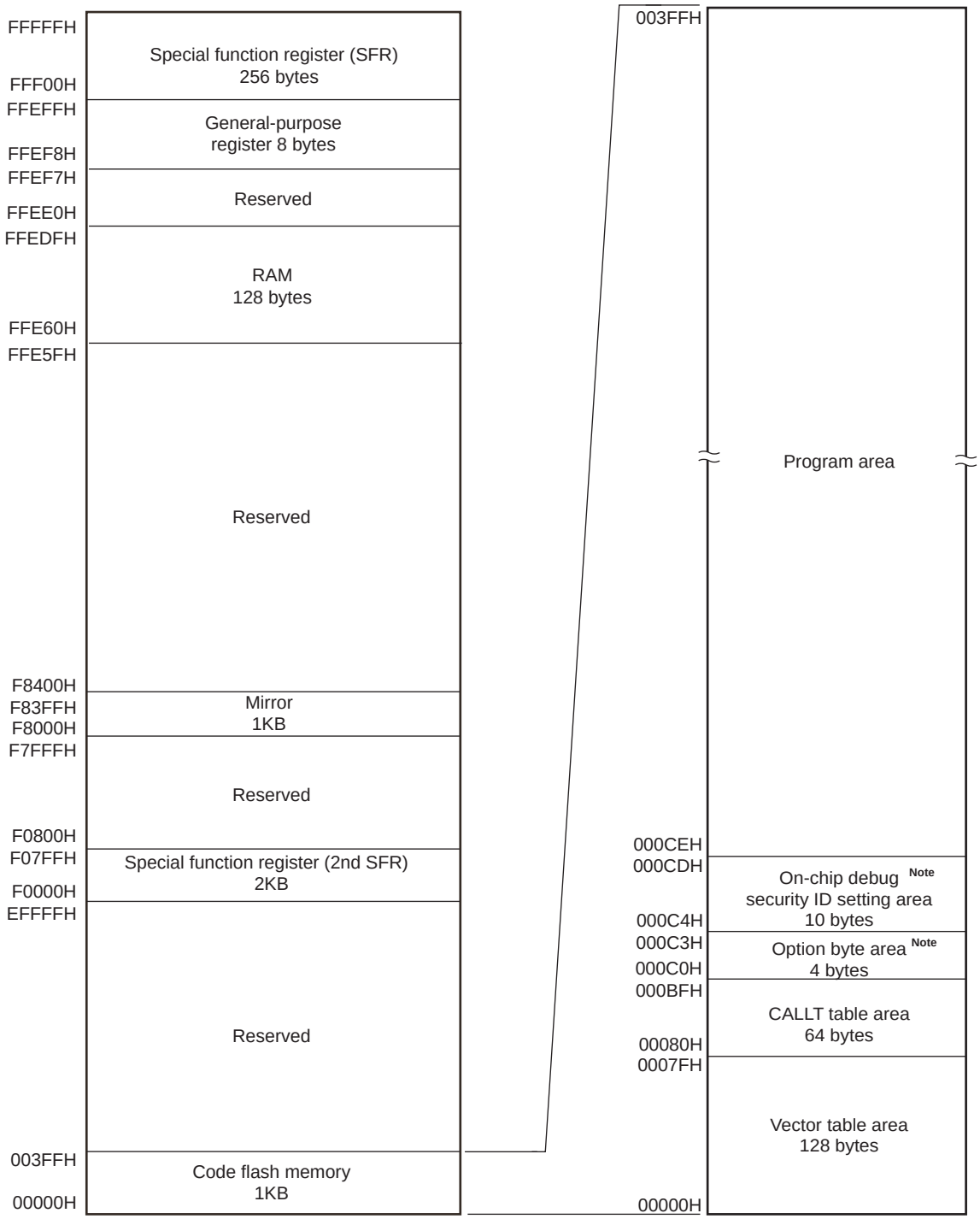
The features of the RL78-S1 core are as follows.

- CISC architecture with 3-stage pipeline
- Address space: 1 MB
- General-purpose register : 8-bit register × 8
- The RL78-S2 and RL78-S1 cores have a common instruction set. Note, however, the following instructions require a different number of clock cycles. For details, see **CHAPTER 23 INSTRUCTION SET**.
  - 16-bit data transfer (MOVW, XCHW, ONEW, CLRW)
  - 16-bit operation (ADDW, SUBW, CMPW)
  - Multiply (MULU)
  - 16-bit increment/decrement (INCW, DECW)
  - 16-bit shift (SHRW, SHLW, SARW)
  - 16-bit rotate (ROLWC)
  - Call/return (CALL, CALLT, BRK, RET, RETI, RETB)
  - Stack manipulate (PUSH, POP, MOVW, ADDW, SUBW)

3.1 Memory Space

Products in the RL78/G10 can access a 1 MB address space. Figures 3-1 to 3-3 show the memory maps.

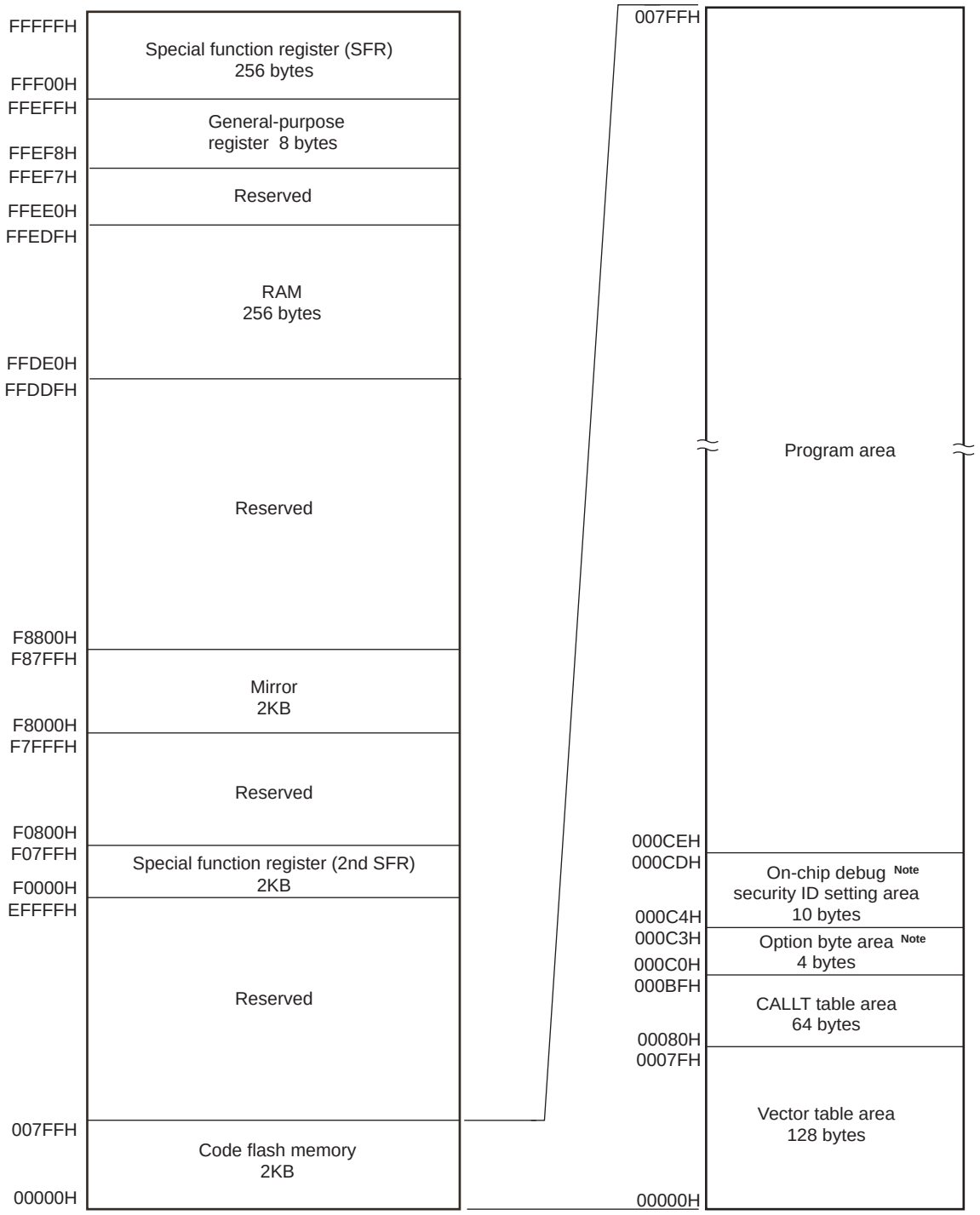
Figure 3-1. Memory Map for the R5F10Y14 and R5F10Y44



Note Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

Caution Access to the reserved area is prohibited.

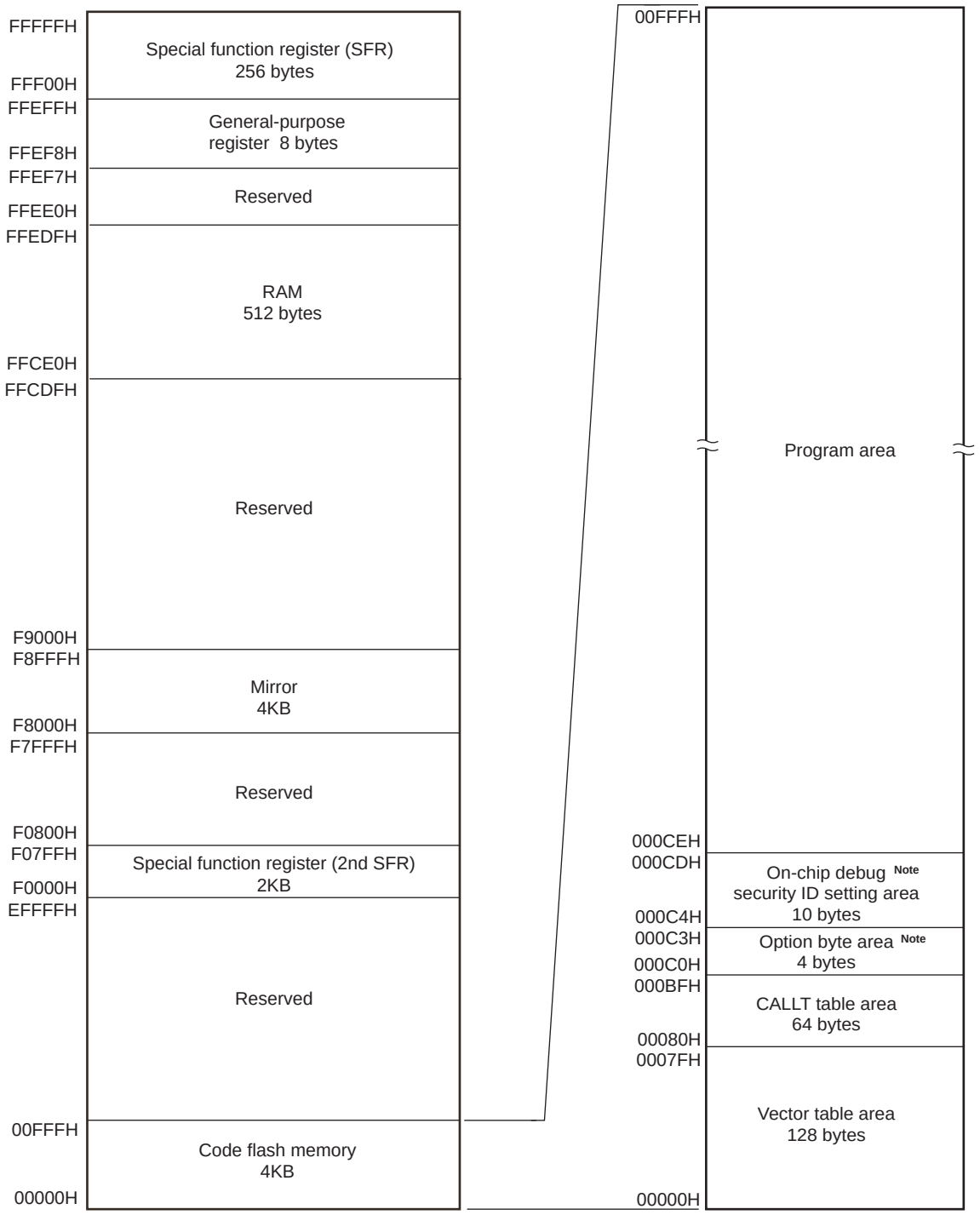
Figure 3-2. Memory Map for the R5F10Y16 and R5F10Y46



Note Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

Caution Access to the reserved area is prohibited.

Figure 3-3. Memory Map for the R5F10Y17 and R5F10Y47



Note Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

Caution Access to the reserved area is prohibited.

3.1.1 Internal program memory space

The internal program memory space stores the program and table data.

The RL78/G10 products incorporate internal ROM (flash memory), as shown below.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F10Y14, R5F10Y44	Flash memory	1024 × 8 bits (00000H to 003FFH)
R5F10Y16, R5F10Y46		2048 × 8 bits (00000H to 007FFH)
R5F10Y17, R5F10Y47		4048 × 8 bits (00000H to 00FFFH)

The internal program address space is divided into the following areas.

(1) Vector table area

The 128-byte area of 00000H to 0007FH is reserved as a vector table area. The program start addresses for branch upon reset or generation of each interrupt request are stored in the vector table area. Furthermore, the interrupt jump addresses are assigned to a 64 KB address area of 00000H to 0FFFFH, because the vector code is 2 bytes.

Of 16-bit addresses, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

Table 3-2. Vector Table

Vector Table Address	Interrupt Source	16-pin	10-pin
00000H	RESET, SPOR, WDT, TRAP	√	√
00004H	INTWDTI	√	√
00006H	INTP0	√	√
00008H	INTP1	√	√
0000AH	INTST0, INTCSI00, INTIIC00	√	√
0000CH	INTSR0, INTCSI01	√	**Note**
0000EH	INTSRE0	√	√
00010H	INTTM01H	√	√
00012H	INTTM00	√	√
00014H	INTTM01	√	√
00016H	INTAD	√	√
00018H	INTKR	√	√
0001AH	INTP2	√	—
0001CH	INTP3	√	—
0001EH	INTTM03H	√	—
00020H	INTIICA0	√	—
00022H	INTTM02	√	—
00024H	INTTM03	√	—
00026H	INTIT	√	—
00028H	INTCMP0	√	—
0007EH	BRK	√	√

Note INTSR0 only.

(2) CALLT instruction table area

The 64-byte area of 00080H to 000BFH can store the subroutine entry address of a 2-byte call instruction (CALLT). Set the subroutine entry address to a value in a range of 00000H to 0FFFFH (because an address code is 2 bytes).

(3) Option byte area

The 4-byte area of 000C0H to 000C3H can be used as an option byte area. For details, see **CHAPTER 19 OPTION BYTE**.

(4) On-chip debug security ID setting area

The 10-byte areas of 000C4H to 000CDH and 010C4H to 010CDH can be used as an on-chip debug security ID setting area. For details, see **CHAPTER 21 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The products with 1/2/4 KB flash memory mirror the code flash area of 00000H to 003FFH/007FFH/00FFFH to the area of F8000H to F83FFH/F87FFH/F8FFFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).

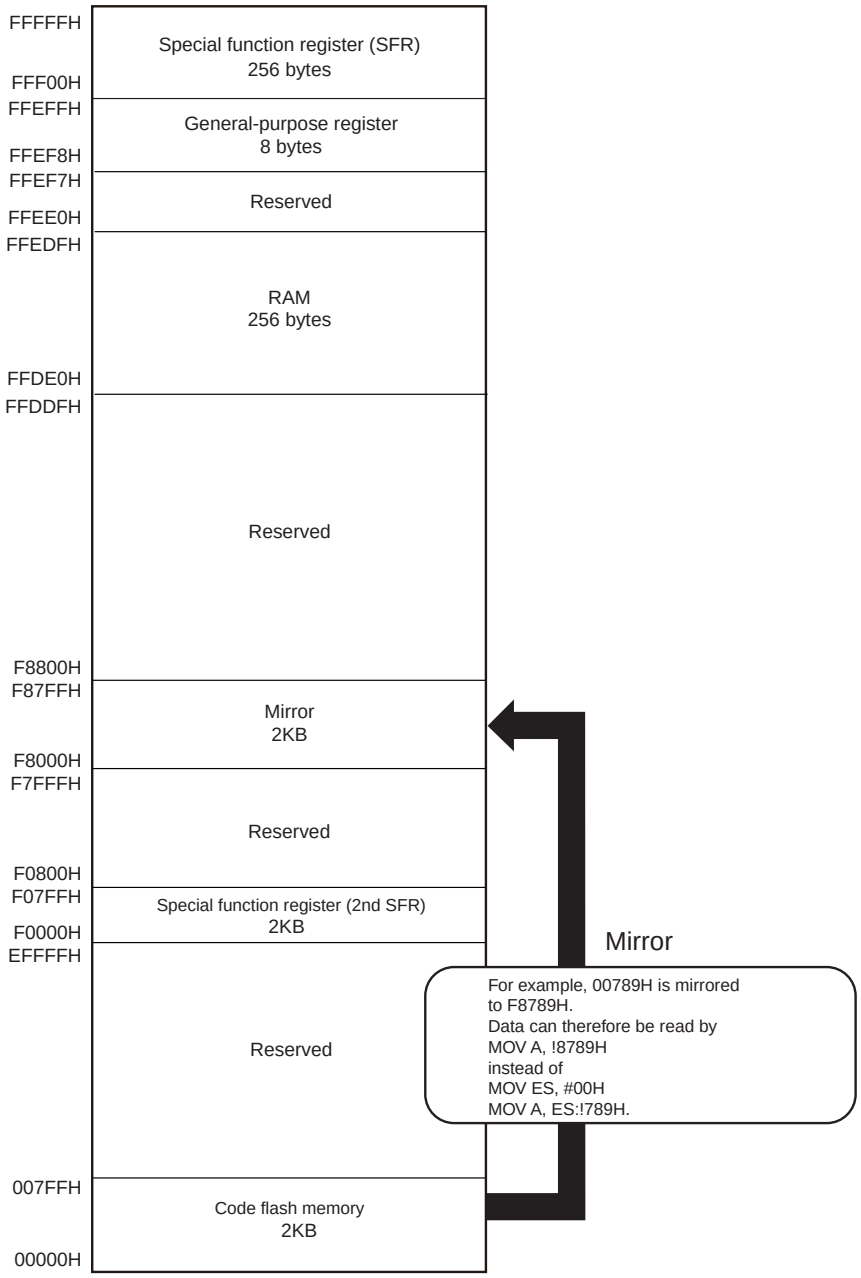
By reading data from F8000H to F83FFH/F87FFH/F8FFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code.

See **3.1 Memory Space** for the mirror area of each product.

The mirror area can only be read and no instruction can be fetched from this area.

The following shows examples.

Example R5F10Y16 (Flash memory: 2 KB)



3.1.3 Internal data memory space

The RL78/G10 products incorporate the following RAMs.

Table 3-3. Internal RAM Capacity

Part Number	Internal RAM
R5F10Y14, R5F10Y44	128 bytes (FFE60H to FFEDFH)
R5F10Y16, R5F10Y46	256 bytes (FFDE0H to FFEDFH)
R5F10Y17, R5F10Y47	512 bytes (FFCE0H to FFEDFH)

The internal RAM can be used as a data area and a program area where instructions are fetched (it is prohibited to use the general-purpose register area for fetching instructions).

The internal RAM is used as stack memory.

Caution It is prohibited to use the general-purpose register (FFEF8H to FFEFFH) space for fetching instructions or as a stack area.

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area of FFF00H to FFFFFH (see **Table 3-4** in **3.2.4 Special function registers (SFRs)**).

Caution Do not access addresses to which SFRs are not assigned.

3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area of F0000H to F07FFH (see **Table 3-5** in **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**).

SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

Caution Do not access addresses to which extended SFRs are not assigned.

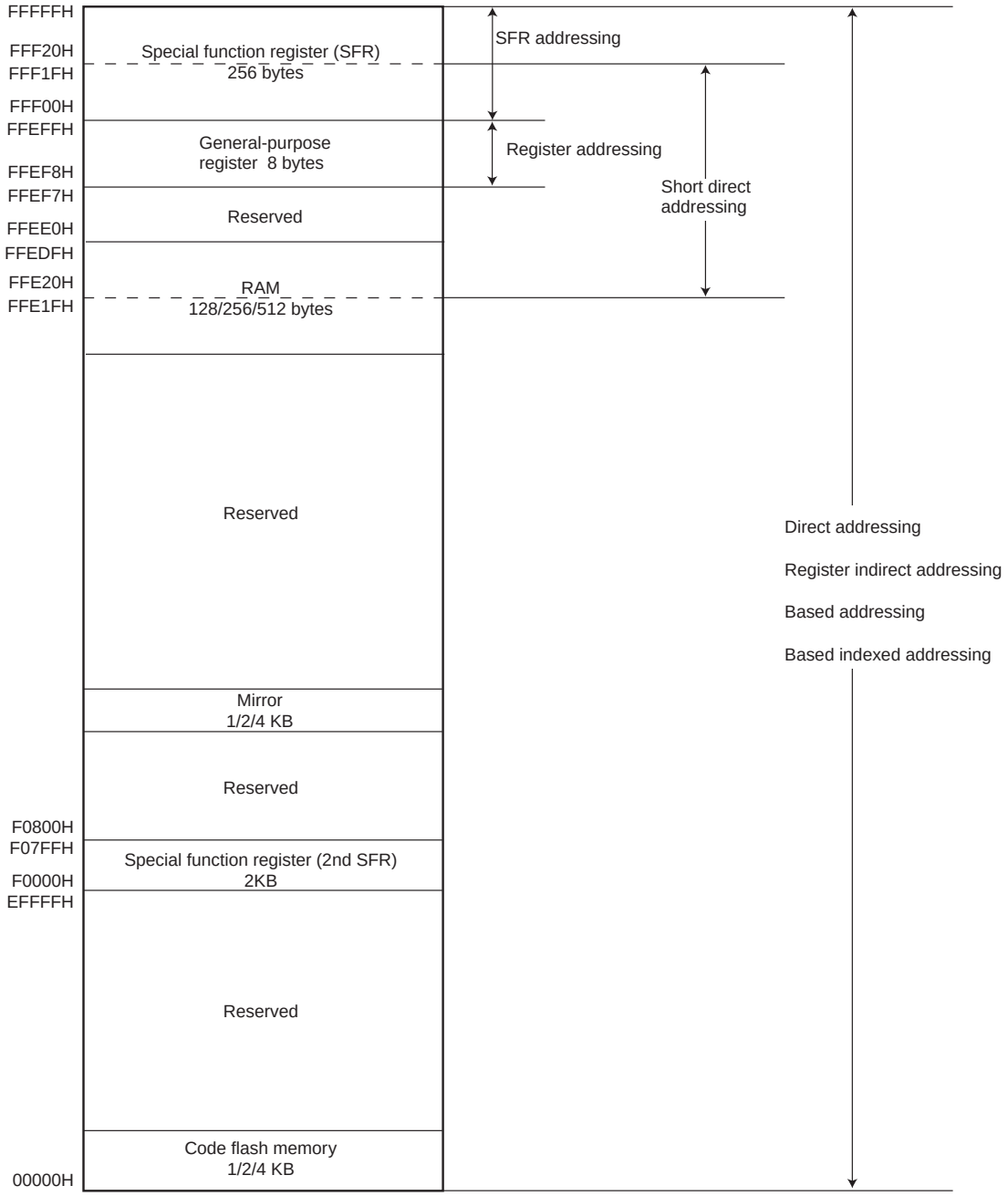
3.1.6 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78/G10, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figure 3-4 shows correspondence between data memory and addressing.

For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3-4. Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The RL78/G10 products incorporate the following processor registers.

3.2.1 Control registers

The control registers control the program sequence, statuses and stack memory. The control registers consist of a program counter (PC), a program status word (PSW) and a stack pointer (SP).

(1) Program counter (PC)

The program counter is a 20-bit register that holds the address information of the next program to be executed.

In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched.

When a branch instruction is executed, immediate data and register contents are set.

Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the 16 lower-order bits of the program counter. The four higher-order bits of the program counter are cleared to 0000.

Figure 3-5. Format of Program Counter

	19							0
PC								

(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags set/reset by instruction execution.

Program status word contents are stored in the stack area upon acknowledgment of a vectored interrupt request or PUSH PSW instruction execution, and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal generation sets the PSW register to 06H.

Figure 3-6. Format of Program Status Word

	7							0
PSW	IE	Z	0	AC	0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

This flag controls the interrupt request acknowledge operations of the CPU.

When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupt requests are disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state, and maskable interrupt request acknowledgment is controlled with an in-service priority flag (ISP1, ISP0), an interrupt mask flag for various interrupt sources, and a priority specification flag.

The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

(b) Zero flag (Z)

When the operation or comparison result is zero or equal, this flag is set (1). It is reset (0) in all other cases.

(c) Auxiliary carry flag (AC)

If the operation result has a carry from bit 3 or a borrow at bit 3, this flag is set (1). It is reset (0) in all other cases.

(d) In-service priority flags (ISP1, ISP0)

These flags manage the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PR00L, PR00H, PR10L, PR10H, PR01L, PR11L) (see **14.3.3 Priority specification flag registers (PR00L, PR00H, PR10L, PR10H, PR01L, PR11L)**) can not be acknowledged. Actual vectored interrupt request acknowledgment is controlled by the interrupt enable flag (IE).

(e) Carry flag (CY)

This flag stores overflow and underflow upon add/subtract instruction execution. It stores the shift-out value upon rotate instruction execution and functions as a bit accumulator during bit operation instruction execution.

(3) Stack pointer (SP)

This is a 16-bit register to hold the start address of the memory stack area. Only the internal RAM area can be set as the stack area.

Figure 3-7. Format of Stack Pointer

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	0

In stack addressing through a stack pointer, the SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

Cautions 1. Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

2. It is prohibited to use the general-purpose register (FFEF8H to FFEFFH) space for fetching instructions or a stack area.

3.2.2 General-purpose registers

The general-purpose registers are a bank of eight 8-bit registers (X, A, C, B, E, D, L, and H) mapped to addresses (FFEF8H to FFEFFH) of the data memory.

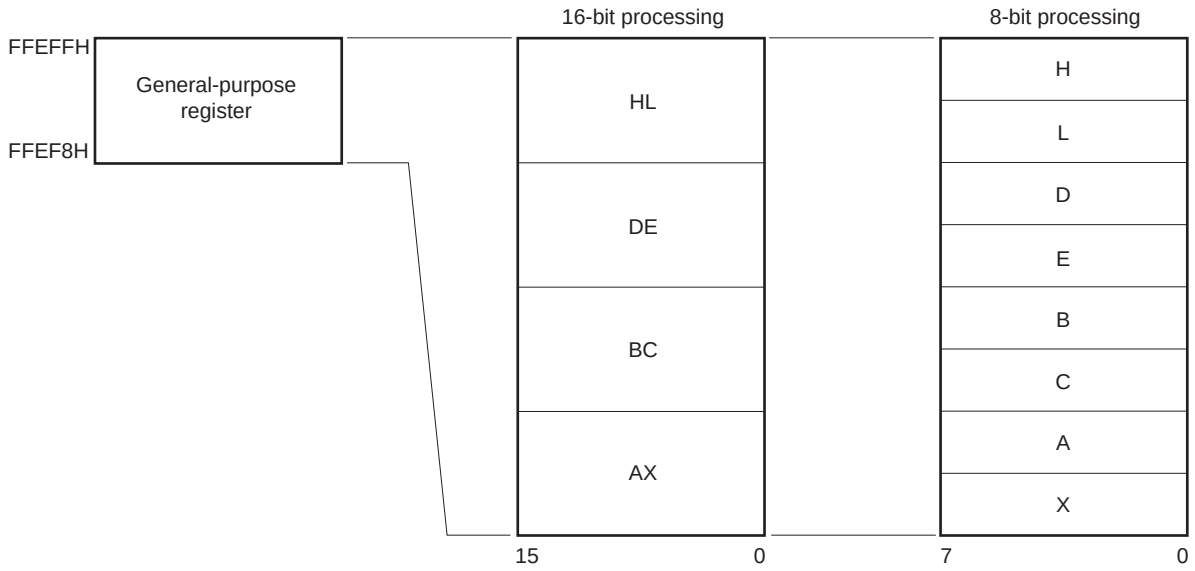
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

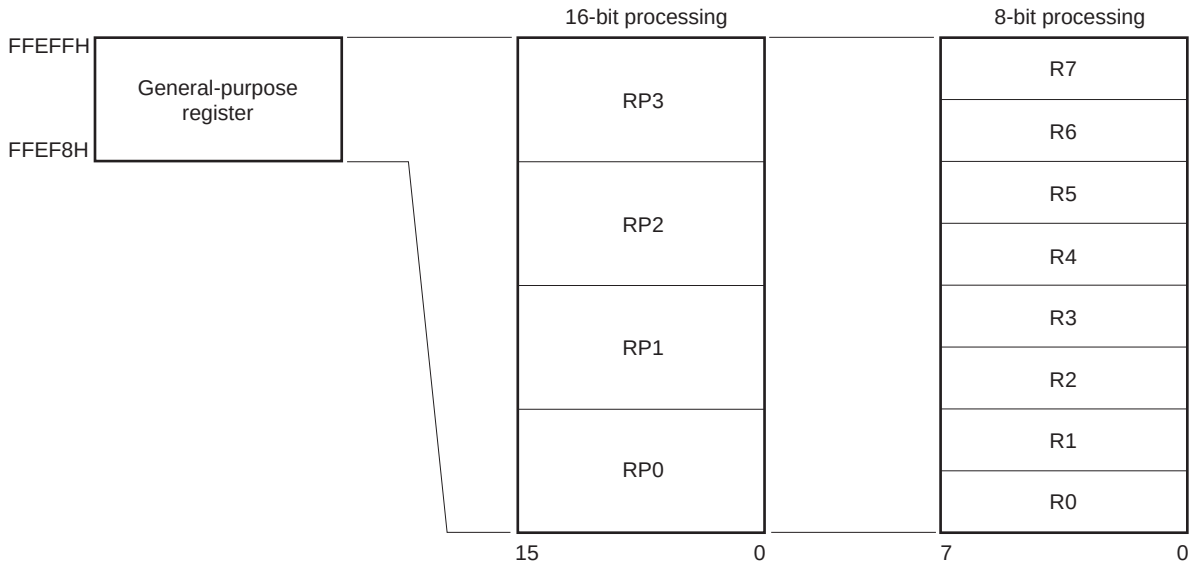
Caution **It is prohibited to use the general-purpose register (FFEF8H to FFEFFH) space for fetching instructions or as a stack area.**

Figure 3-8. Configuration of General-Purpose Registers

(a) Function name



(b) Absolute name



3.2.3 ES and CS registers

The ES register and CS register are used to specify the higher address for data access and when a branch instruction is executed (register direct addressing), respectively.

The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

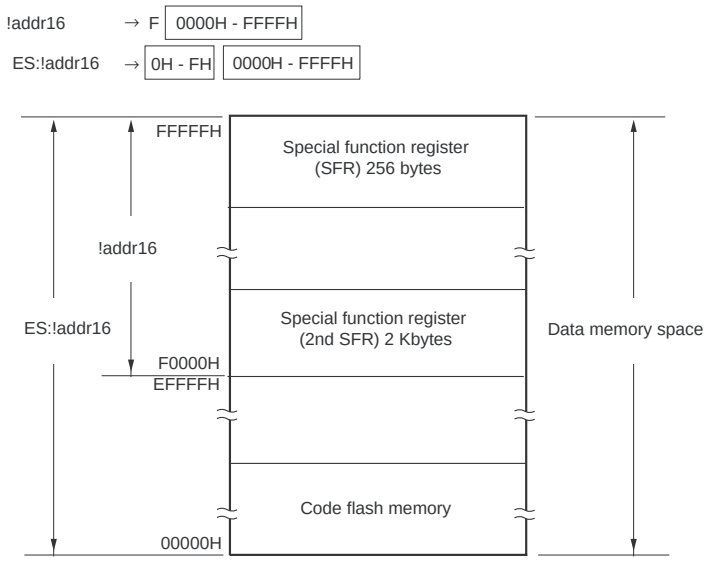
Figure 3-9. Configuration of ES and CS Registers

	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0

	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CS2	CS1	CS0

Though the data area which can be accessed with 16-bit addresses is the 64 Kbytes from F0000H to FFFFFH, using the ES register as well extends this to the 1 Mbyte from 00000H to FFFFFH.

Figure 3-10 Extension of Data Area Which Can Be Accessed



3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each SFR has a special function.

SFRs are allocated to the FFF00H to FFFFFH area.

SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1 and 8, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified with an address.

Table 3-5 gives a list of the SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of a special function register. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1 or 8). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

Caution Do not access addresses to which extended SFRs are not assigned.

Remark For extended SFRs (2nd SFRs), see **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 3-4. SFR List (1/2)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range		After Reset
					1-bit	8-bit	
FFF00H	Port register 0	P0		R/W	√	√	00H
FFF04H	Port register 4	P4		R/W	√	√	00H
FFF0CH	Port register 12	P12		R	√	√	Undefined
FFF0DH	Port register 13	P13		R	√	√	Undefined
FFF10H	Serial data register 00L	TXD0/ SIO00	SDR00L	R/W	–	√	00H
FFF11H	Serial data register 00H	–	SDR00H	R/W	–	√	00H
FFF12H	Serial data register 01L	RXD0/ SIO01 Note	SDR01L	R/W	–	√	00H
FFF13H	Serial data register 01H	–	SDR01H	R/W	–	√	00H
FFF18H	Timer data register 00L	TDR00L		R/W	–	√	00H
FFF19H	Timer data register 00H	TDR00H		R/W	–	√	00H
FFF1AH	Timer data register 01L	TDR01L		R/W	–	√	00H
FFF1BH	Timer data register 01H	TDR01H		R/W	–	√	00H
FFF1EH	A/D conversion result lower bit register	ADCRL		R	–	√	00H
FFF1FH	A/D conversion result upper bit register	ADCRH		R	–	√	00H
FFF20H	Port mode register 0	PM0		R/W	√	√	FFH
FFF24H	Port mode register 4	PM4		R/W	√	√	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	√	√	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	00H
FFF34H	Key interrupt control register	KRCTL		R/W	√	√	00H
FFF35H	Key interrupt flag register	KRF		R/W	–	√	00H
FFF37H	Key interrupt mode control register 0	KRM0		R/W	√	√	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	00H
FFF50H	IICA shift register 0 Note	IICA0		R/W	–	√	00H
FFF51H	IICA status register 0 Note	IICS0		R	√	√	00H
FFF52H	IICA flag register 0 Note	IICF0		R/W	√	√	00H
FFF60H	Comparator mode setting register Note	COMPMDR		R/W	√	√	00H
FFF61H	Comparator filter control register Note	COMPFIR		R/W	√	√	00H
FFF62H	Comparator output control register Note	COMPOCR		R/W	√	√	00H
FFF64H	Timer data register 02L Note	TDR02L		R/W	–	√	00H
FFF65H	Timer data register 02H Note	TDR02H		R/W	–	√	00H
FFF66H	Timer data register 03L Note	TDR03L		R/W	–	√	00H
FFF67H	Timer data register 03H Note	TDR03H		R/W	–	√	00H
FFF90H	Interval timer control register L Note	ITMCL		R/W	–	√	FFH
FFF91H	Interval timer control register H Note	ITMCH		R/W	–	√	0FH

Note 16-pin products only.

Table 3-4. SFR List (2/2)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range		After Reset
				1-bit	8-bit	
FFFA0H	Clock operation mode control register Note 1	CMC	R/W	–	√	00H
FFFA1H	Clock operation status control register Note 1	CSC	R/W	√	√	80H
FFFA2H	Oscillation stabilization time counter status register Note 1	OSTC	R/W	√	√	00H
FFFA3H	Oscillation stabilization time select register Note 1	OSTS	R	–	√	07H
FFFA4H	System clock control register Note 1	CKC	R/W	√	√	00H
FFFA5H	Clock output select register 0	CKS0	R/W	√	√	00H
FFFA8H	Reset control flag register	RESF	R	–	√	Undefined Note 2
FFFABH	Watchdog timer enable register	WDTE	R/W	–	√	1AH/9AH Note 3
FFFE0H	Interrupt request flag register 0L	IF0L	R/W	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H	R/W	√	√	00H
FFFE2H	Interrupt request flag register 1L Note 1	IF1L	R/W	√	√	00H
FFFE4H	Interrupt mask flag register 0L	MK0L	R/W	√	√	FFH
FFFE5H	Interrupt mask flag register 0H	MK0H	R/W	√	√	FFH
FFFE6H	Interrupt mask flag register 1L Note 1	MK1L	R/W	√	√	FFH
FFFE8H	Priority specification flag register 00L	PR00L	R/W	√	√	FFH
FFFE9H	Priority specification flag register 00H	PR00H	R/W	√	√	FFH
FFFEAH	Priority specification flag register 01L Note 1	PR01L	R/W	√	√	FFH
FFFECH	Priority specification flag register 10L	PR10L	R/W	√	√	FFH
FFFEDH	Priority specification flag register 10H	PR10H	R/W	√	√	FFH
FFFEEH	Priority specification flag register 11L Note 1	PR11L	R/W	√	√	FFH

Notes 1. 16-pin products only.

2. The reset values of the register vary depending on the reset source as shown below.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by execution of illegal instruction	Reset by WDT	Reset by SPOR	Reset by data retention lower limit voltage
RESF	TRAP bit	Cleared (0)	Set (1)	Held	Held	Cleared (0)
	WDTRF bit		Held	Set (1)	Held	
	SPORF bit		Held	Held	Set (1)	

3. The reset value of the WDTE register is determined by the setting of the option byte.

Remark For extended SFRs (2nd SFRs), see **Table 3-5 Extended SFR (2nd SFR) List.**

3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)

Unlike a general-purpose register, each extended SFR (2nd SFR) has a special function.

Extended SFRs are allocated to the F0000H to F07FFH area. SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

Extended SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1 and 8, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (!addr16.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (!addr16). This manipulation can also be specified with an address.

Table 3-5 gives a list of the extended SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of an extended SFR. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding extended SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1 or 8). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

Caution Do not access addresses to which extended SFRs are not assigned.

Remark For SFRs in the SFR area, see **3.2.4 Special function registers (SFRs)**.

Table 3-5. Extended SFR (2nd SFR) List (1/2)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range		After Reset
				1-bit	8-bit	
F0010H	A/D converter mode register 2	ADM2	R/W	√	√	00H
F0013H	A/D test register Note 1	ADTES	R/W	–	√	00H
F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	01H
F003CH	Pull-up resistor option register 12	PU12	R/W	√	√	20H
F0050H	Port output mode register 0	POM0	R/W	√	√	00H
F0060H	Port mode control register 0	PMC0	R/W	√	√	FFH
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	00H
F0073H	Input switch control register	ISC	R/W	√	√	00H
F0077H	Peripheral I/O redirection register	PIOR	R/W	–	√	00H
F00A8H	High-speed on-chip oscillator trimming register	HOCODIV	R/W	–	√	Undefined Note 2
F00F0H	Peripheral enable register 0	PER0	R/W	√	√	00H
F00F3H	Operation speed mode control register Note 1	OSMC	R/W	–	√	00H
F00FEH	BCD adjust result register	BCDADJ	R	–	√	Undefined
F0100H	Serial status register 00	SSR00	R	–	√	00H
F0102H	Serial status register 01	SSR01	R	–	√	00H
F0108H	Serial flag clear trigger register 00	SIR00	R/W	–	√	00H
F010AH	Serial flag clear trigger register 01	SIR01	R/W	–	√	00H
F0110H	Serial mode register 00L	SMR00L	R/W	–	√	20H
F0111H	Serial mode register 00H	SMR00H	R/W	–	√	00H
F0112H	Serial mode register 01L	SMR01L	R/W	–	√	20H
F0113H	Serial mode register 01H	SMR01H	R/W	–	√	00H
F0118H	Serial communication operation setting register 00L	SCR00L	R/W	–	√	87H
F0119H	Serial communication operation setting register 00H	SCR00H	R/W	–	√	00H
F011AH	Serial communication operation setting register 01L	SCR01L	R/W	–	√	87H
F011BH	Serial communication operation setting register 01H	SCR01H	R/W	–	√	00H
F0120H	Serial channel enable status register 0	SE0	R	√	√	00H
F0122H	Serial channel start register 0	SS0	R/W	√	√	00H
F0124H	Serial channel stop register 0	ST0	R/W	√	√	00H
F0126H	Serial clock select register 0	SPS0	R/W	–	√	00H
F0128H	Serial output register 0	SO0	R/W	–	√	03H
F0129H	Serial clock output register 0	CKO0	R/W	–	√	03H
F012AH	Serial output enable register 0	SOE0	R/W	√	√	00H
F0134H	Serial output level register 0	SOL0	R/W	–	√	00H
F0180H	Timer counter register 00L	TCR00L	R	–	√	FFH
F0181H	Timer counter register 00H	TCR00H	R	–	√	FFH
F0182H	Timer counter register 01L	TCR01L	R	–	√	FFH
F0183H	Timer counter register 01H	TCR01H	R	–	√	FFH

Notes 1. 16-pin products only.

2. The value after a reset is a value set in FRQSEL2 to FRQSEL0 of the option byte (000C2H).

Table 3-5. Extended SFR (2nd SFR) List (2/2)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range		After Reset
				1-bit	8-bit	
F0184H	Timer counter register 02L Note	TCR02L	R	–	√	FFH
F0185H	Timer counter register 02H Note	TCR02H	R	–	√	FFH
F0186H	Timer counter register 03L Note	TCR03L	R	–	√	FFH
F0187H	Timer counter register 03H Note	TCR03H	R	–	√	FFH
F0190H	Timer mode register 00L	TMR00L	R/W	–	√	00H
F0191H	Timer mode register 00H	TMR00H	R/W	–	√	00H
F0192H	Timer mode register 01L	TMR01L	R/W	–	√	00H
F0193H	Timer mode register 01H	TMR01H	R/W	–	√	00H
F0194H	Timer mode register 02L Note	TMR02L	R/W	–	√	00H
F0195H	Timer mode register 02H Note	TMR02H	R/W	–	√	00H
F0196H	Timer mode register 03L Note	TMR03L	R/W	–	√	00H
F0197H	Timer mode register 03H Note	TMR03H	R/W	–	√	00H
F01A0H	Timer status register 00	TSR00	R	–	√	00H
F01A2H	Timer status register 01	TSR01	R	–	√	00H
F01A4H	Timer status register 02 Note	TSR02	R	–	√	00H
F01A6H	Timer status register 03 Note	TSR03	R	–	√	00H
F01B0H	Timer channel enable status register 0	TE0	R	√	√	00H
F01B1H	Timer channel enable status register 0 (8-bit mode)	TEH0	R	√	√	00H
F01B2H	Timer channel start register 0	TS0	R/W	√	√	00H
F01B3H	Timer channel start register 0 (8-bit mode)	TSH0	R/W	√	√	00H
F01B4H	Timer channel stop register 0	TT0	R/W	√	√	00H
F01B5H	Timer channel stop register 0 (8-bit mode)	TTH0	R/W	√	√	00H
F01B6H	Timer clock select register 0	TPS0	R/W	–	√	00H
F01B8H	Timer output register 0	TO0	R/W	–	√	00H
F01BAH	Timer output enable register 0	TOE0	R/W	√	√	00H
F01BCH	Timer output level register 0	TOL0	R/W	–	√	00H
F01BEH	Timer output mode register 0	TOM0	R/W	–	√	00H
F0230H	IICA control register 00 Note	IICCTL00	R/W	√	√	00H
F0231H	IICA control register 01 Note	IICCTL01	R/W	√	√	00H
F0232H	IICA low-level width setting register 0 Note	IICWL0	R/W	–	√	FFH
F0233H	IICA high-level width setting register 0 Note	IICWH0	R/W	–	√	FFH
F0234H	Slave address register 0 Note	SVA0	R/W	–	√	00H

Note 16-pin products only.

Remark For SFRs in the SFR area, see **Table 3-4 SFR List**.

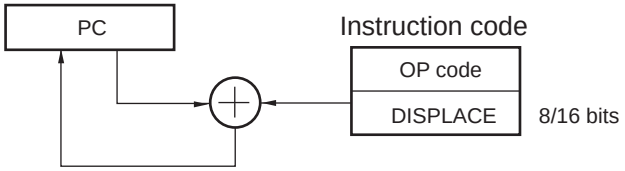
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

Relative addressing stores in the program counter (PC) the result of adding a displacement value included in the instruction word (signed complement data: –128 to +127 or –32768 to +32767) to the program counter (PC)'s value (the start address of the next instruction), and specifies the program address to be used as the branch destination. Relative addressing is applied only to branch instructions.

Figure 3-11. Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

Immediate addressing stores immediate data of the instruction word in the program counter, and specifies the program address to be used as the branch destination.

For immediate addressing, CALL !!addr20 or BR !!addr20 is used to specify 20-bit addresses and CALL !addr16 or BR !addr16 is used to specify 16-bit addresses. 0000 is set to the higher 4 bits when specifying 16-bit addresses.

Figure 3-12. Example of CALL !!addr20/BR !!addr20

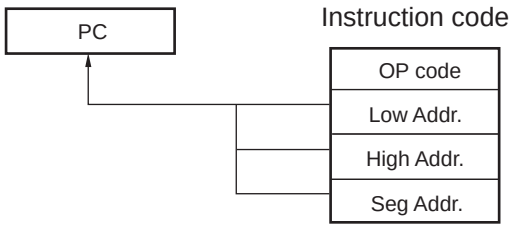
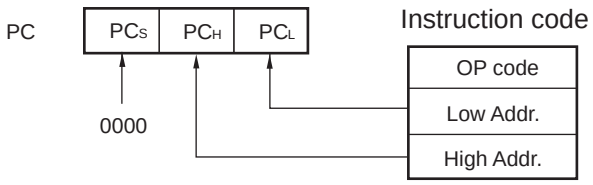


Figure 3-13. Example of CALL !addr16/BR !addr16



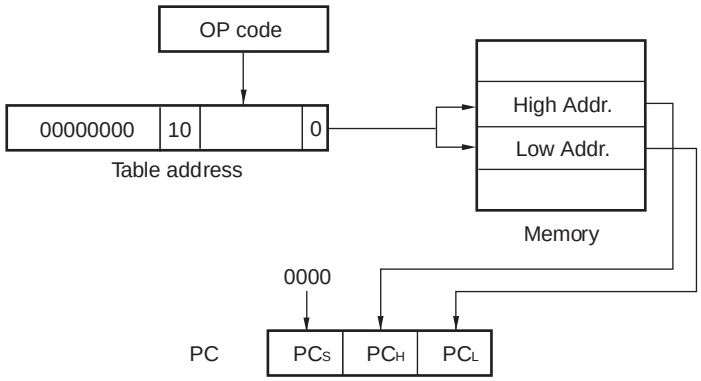
3.3.3 Table indirect addressing

[Function]

Table indirect addressing specifies a table address in the CALLT table area (0080H to 00BFH) with the 5-bit immediate data in the instruction word, stores the contents at that table address and the next address in the program counter (PC) as 16-bit data, and specifies the program address. Table indirect addressing is applied only for CALLT instructions.

In the RL78 microcontrollers, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

Figure 3-14. Outline of Table Indirect Addressing

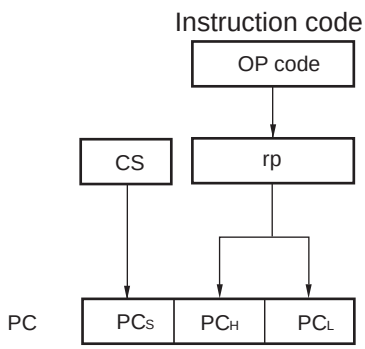


<R> ### 3.3.4 Register indirect addressing

[Function]

Register indirect addressing stores in the program counter (PC) the contents of a general-purpose register pair (AX/BC/DE/HL) and CS register of the current register bank specified with the instruction word as 20-bit data, and specifies the program address. Register indirect addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

Figure 3-15. Outline of Register Indirect Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

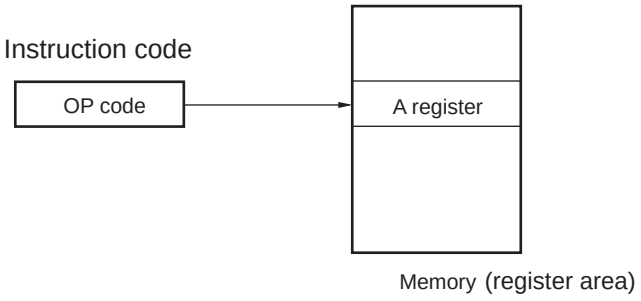
[Function]

Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[Operand format]

Implied addressing can be applied only to MULU X.

Figure 3-16. Outline of Implied Addressing



3.4.2 Register addressing

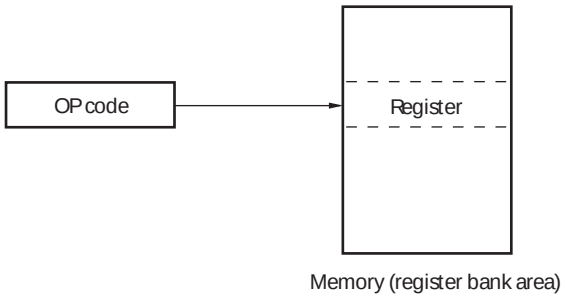
[Function]

Register addressing accesses a general-purpose register as an operand. The instruction word of 3-bit long is used to select an 8-bit register and the instruction word of 2-bit long is used to select a 16-bit register.

[Operand format]

Identifier	Description
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

Figure 3-17. Outline of Register Addressing



3.4.3 Direct addressing

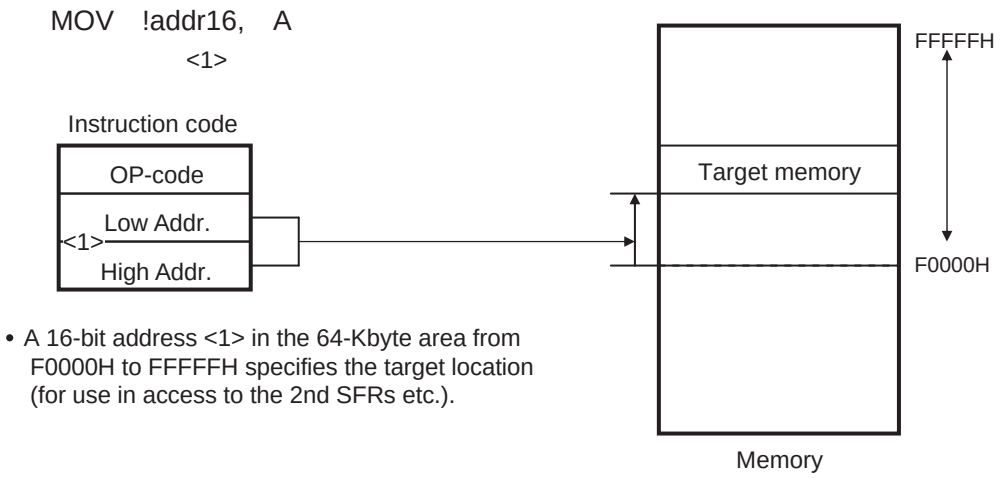
[Function]

Direct addressing uses immediate data in the instruction word as an operand address to directly specify the target address.

[Operand format]

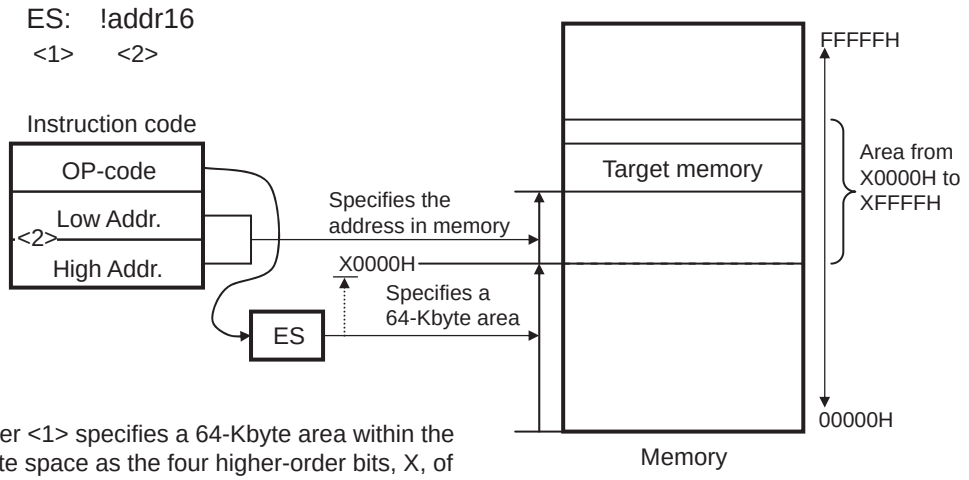
Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3-18. Example of !addr16



- A 16-bit address <1> in the 64-Kbyte area from F0000H to FFFFFH specifies the target location (for use in access to the 2nd SFRs etc.).

Figure 3-19. Example of ES: !addr16



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- A 16-bit address <2> in the area from X0000H to XFFFFH and the ES register <1> specify the target location; this is used for access to fixed data other than that in mirrored areas.

3.4.4 Short direct addressing

[Function]

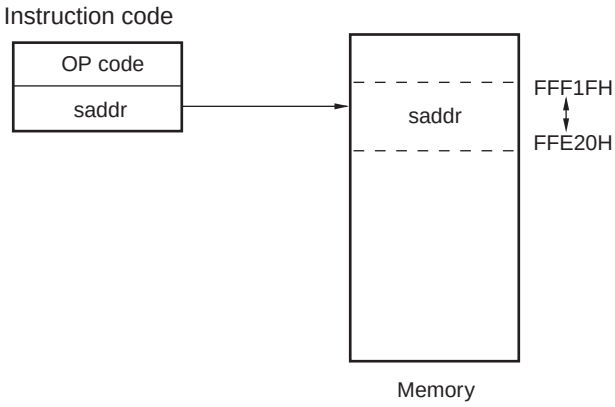
Short direct addressing directly specifies the target addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFE20H to FFF1FH.

Note that it is prohibited to use the area from FFEE0H to FFEF7H. In the products with 128 bytes of RAM, it is also prohibited to use the area from FFE20H to FFE5FH.

[Operand format]

Identifier	Description
SADDR	Label or FFE20H to FFF1FH immediate data
SADDRP	Label or FFE20H to FFF1FH immediate data (only even address is specifiable.)

Figure 3-20. Outline of Short Direct Addressing



Remark SADDR and SADDRP are used to describe the values of addresses FE20H to FF1FH with 16-bit immediate data (higher 4 bits of actual address are omitted), and the values of addresses FFE20H to FFF1FH with 20-bit immediate data.

Regardless of whether SADDR or SADDRP is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

3.4.5 SFR addressing

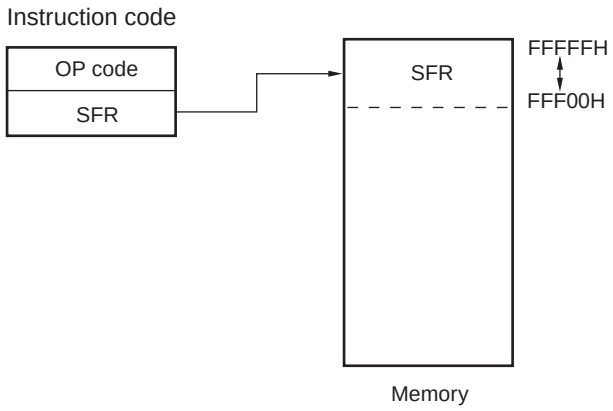
[Function]

SFR addressing directly specifies the target SFR addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFF00H to FFFFFH.

[Operand format]

Identifier	Description
SFR	SFR name
SFRP	16-bit-manipulatable SFR name (even address only)

Figure 3-21. Outline of SFR Addressing



3.4.6 Register indirect addressing

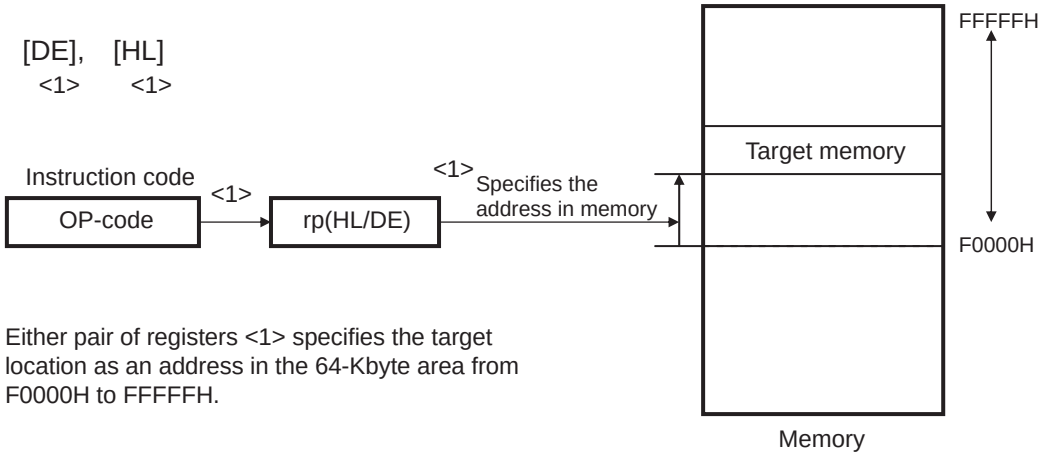
[Function]

Register indirect addressing directly specifies the target addresses using the contents of the register pair specified with the instruction word as an operand address.

[Operand format]

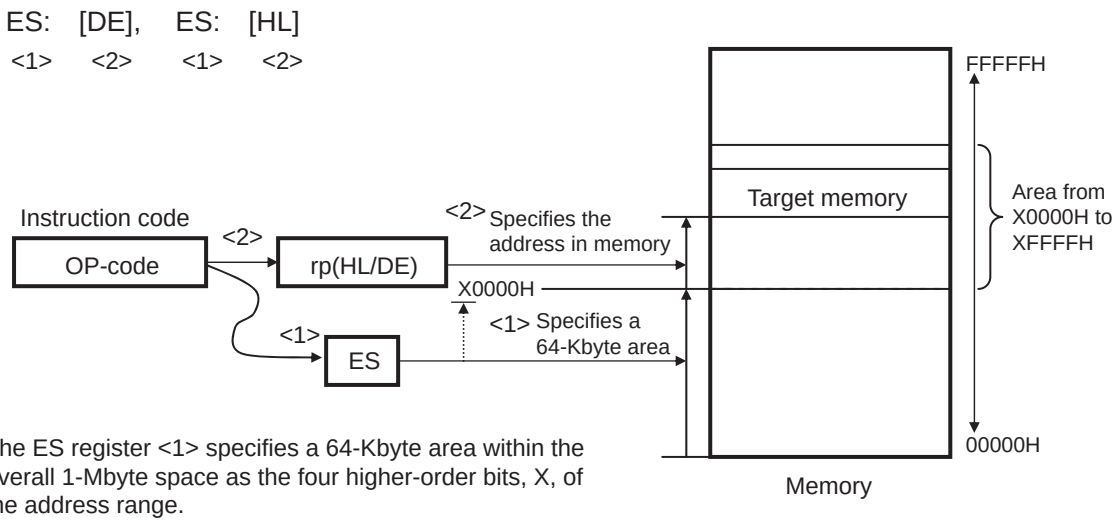
Identifier	Description
–	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

Figure 3-22. Example of [DE], [HL]



- Either pair of registers <1> specifies the target location as an address in the 64-Kbyte area from F0000H to FFFFFH.

Figure 3-23. Example of ES:[DE], ES:[HL]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> and the ES register <1> specify the target location in the area from X0000H to XFFFFH.

3.4.7 Based addressing

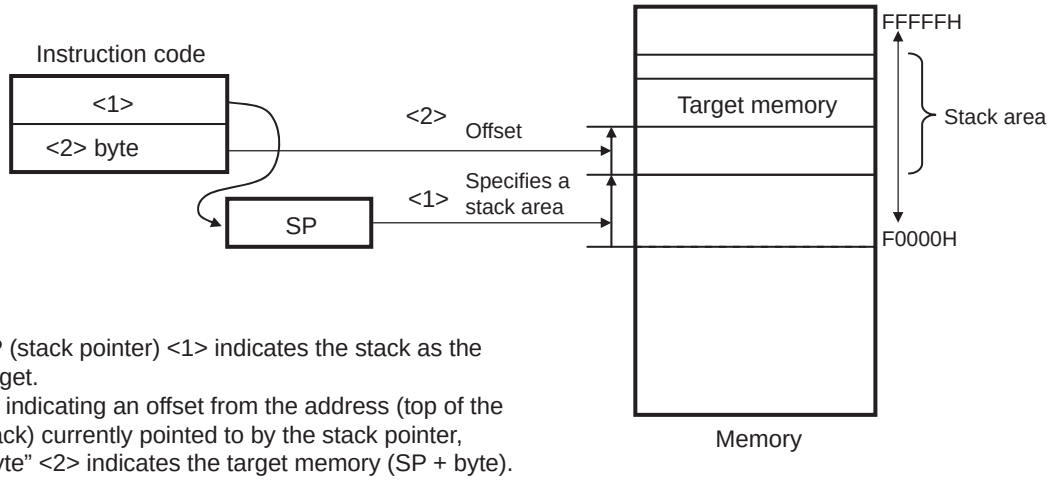
[Function]

Based addressing uses the contents of a register pair specified with the instruction word or 16-bit immediate data as a base address, and 8-bit immediate data or 16-bit immediate data as offset data. The sum of these values is used to specify the target address.

[Operand format]

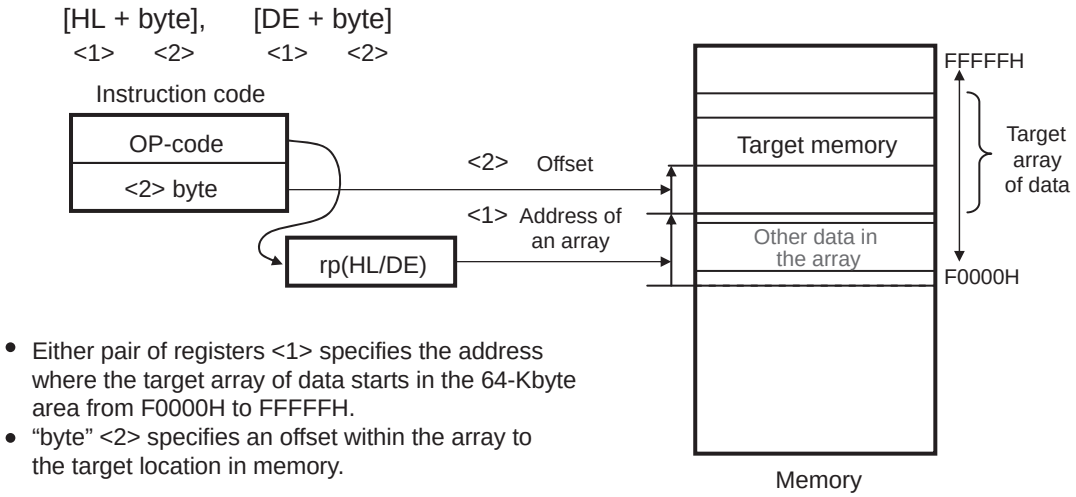
Identifier	Description
–	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
–	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
–	word[BC] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL + byte], ES:[DE + byte] (higher 4-bit addresses are specified by the ES register)
–	ES:word[B], ES:word[C] (higher 4-bit addresses are specified by the ES register)
–	ES:word[BC] (higher 4-bit addresses are specified by the ES register)

Figure 3-24. Example of [SP+byte]



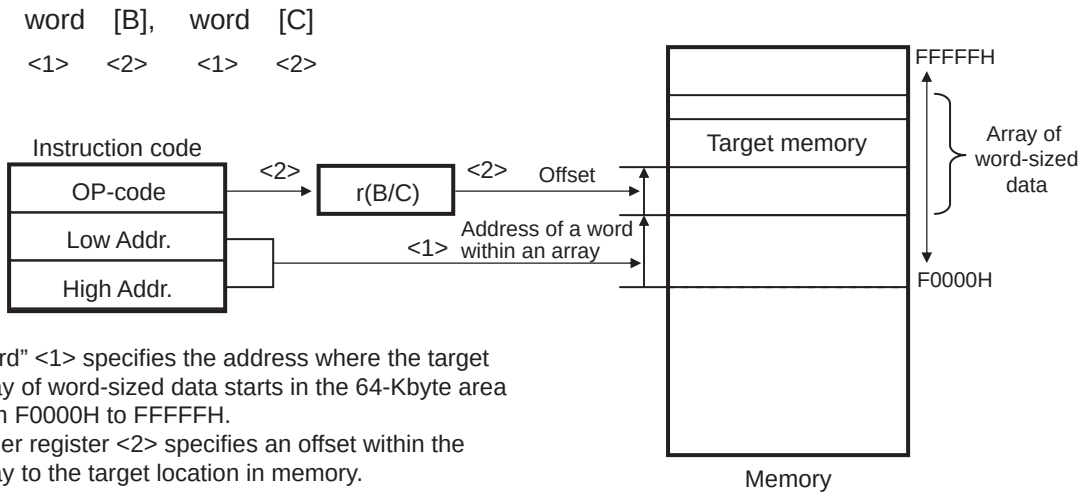
- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, “byte” <2> indicates the target memory (SP + byte).

Figure 3-25. Example of [HL + byte], [DE + byte]



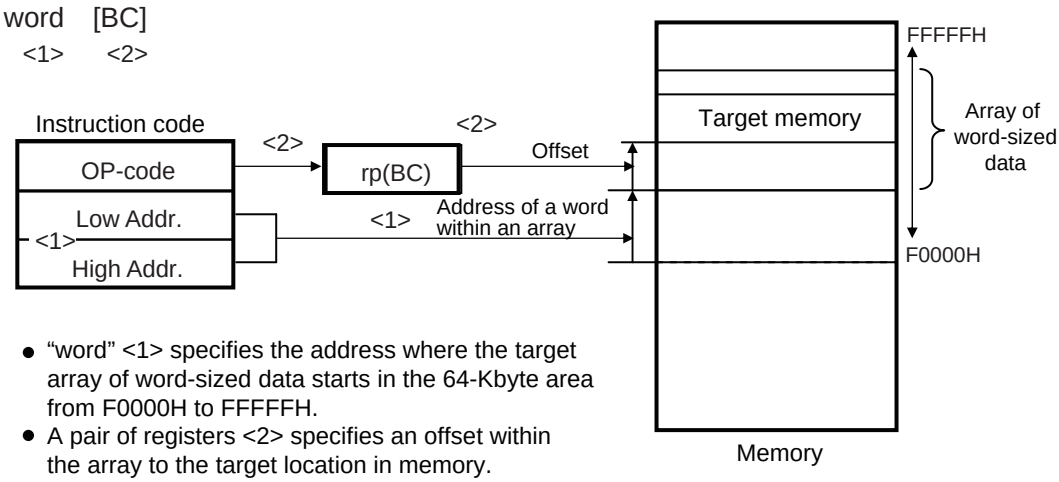
- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- "byte" <2> specifies an offset within the array to the target location in memory.

Figure 3-26. Example of word[B], word[C]



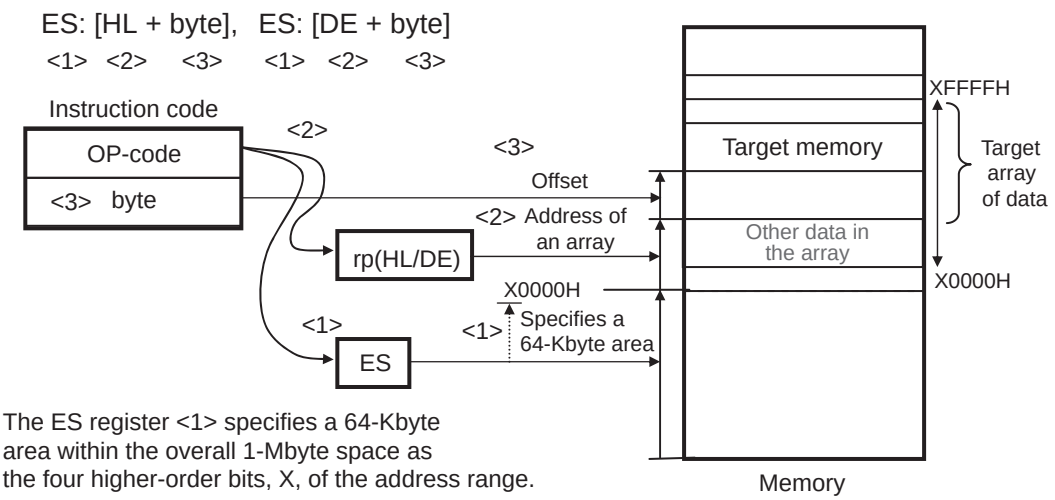
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3-27. Example of word[BC]



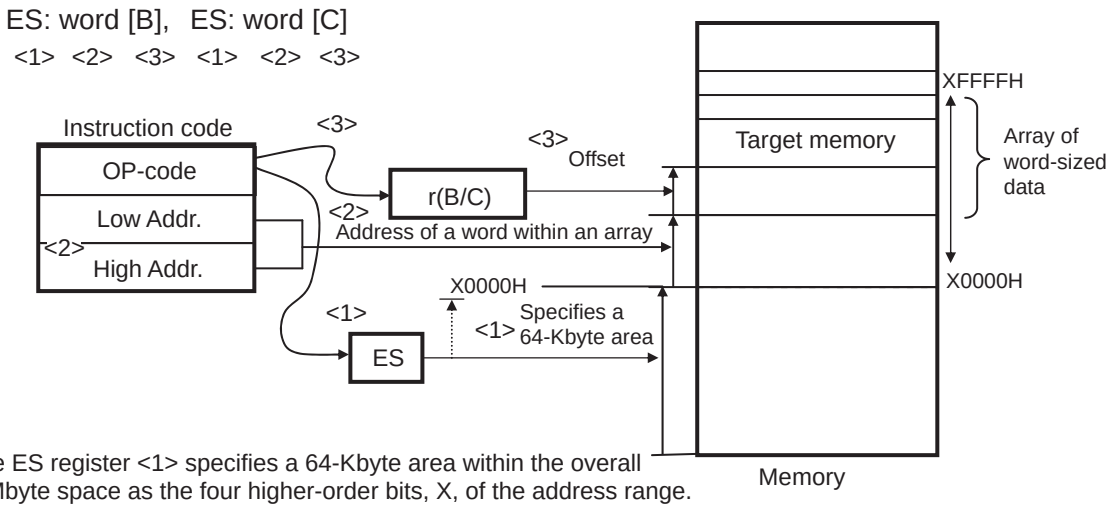
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

Figure 3-28. Example of ES:[HL + byte], ES:[DE + byte]



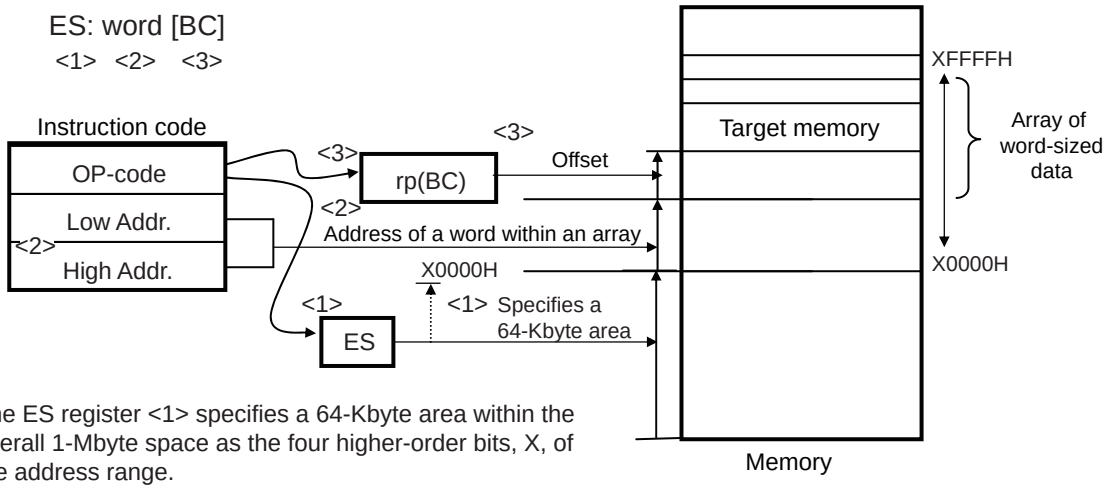
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- "byte" <3> specifies an offset within the array to the target location in memory.

Figure 3-29. Example of ES:word[B], ES:word[C]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- "word" <2> specifies the address where the target array of word-sized data starts in the 64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

Figure 3-30. Example of ES:word[BC]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- “word” <2> specifies the address where the target array of word-sized data starts in the 64-Kbyte area specified in the ES register <1>.
- A pair of registers <3> specifies an offset within the array to the target location in memory.

3.4.8 Based indexed addressing

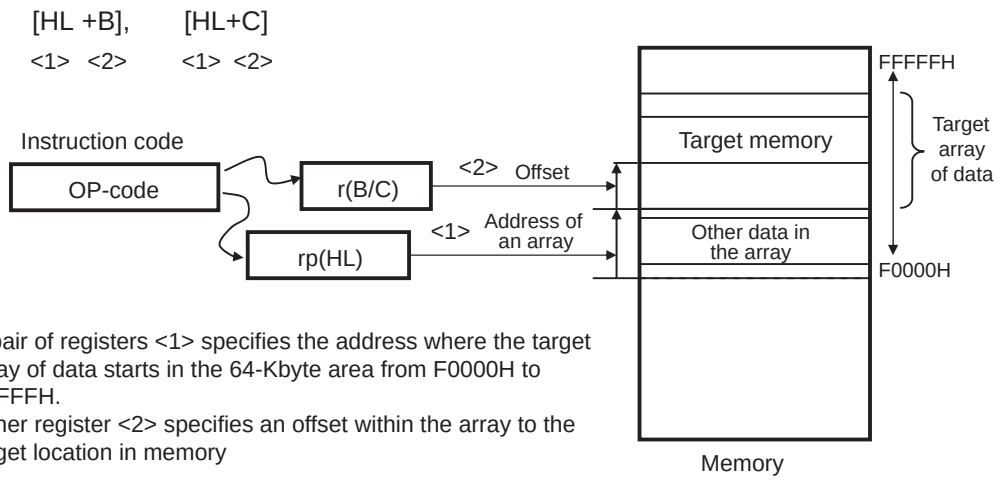
[Function]

Based indexed addressing uses the contents of a register pair specified with the instruction word as the base address, and the content of the B register or C register similarly specified with the instruction word as offset address. The sum of these values is used to specify the target address.

[Operand format]

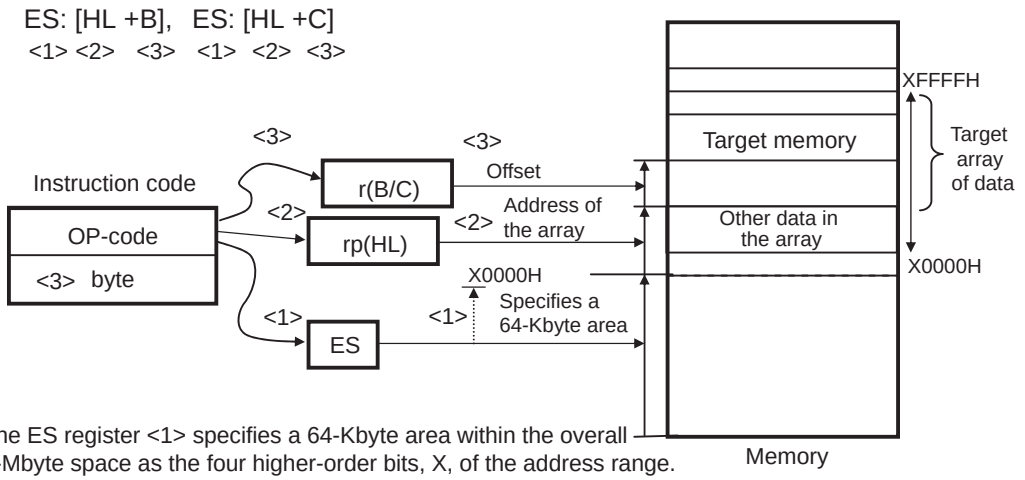
Identifier	Description
–	[HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register)

Figure 3-31. Example of [HL+B], [HL+C]



- A pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory

Figure 3-32. Example of ES:[HL+B], ES:[HL+C]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- A pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) values. This addressing is automatically employed when the PUSH, POP, subroutine call, and return instructions are executed or the register is saved/restored upon generation of an interrupt request.

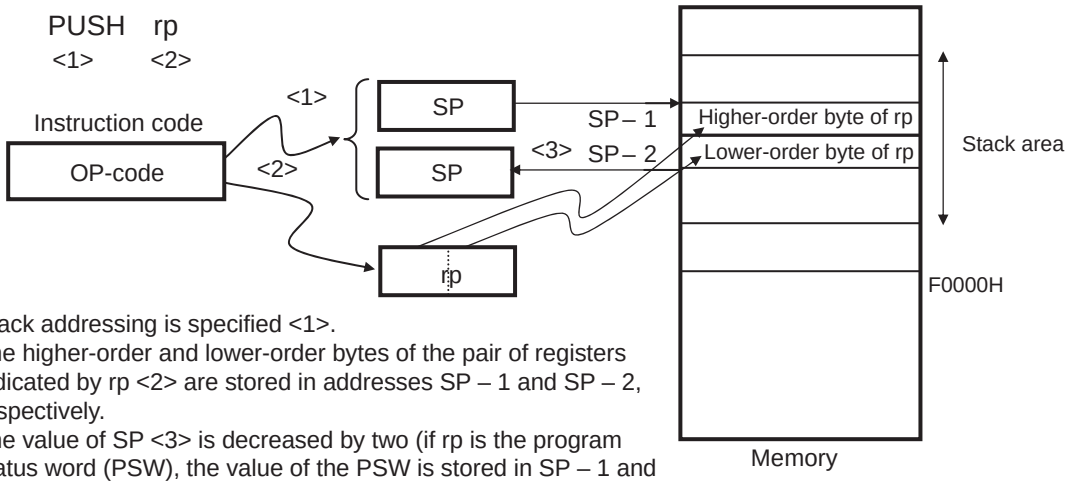
Only the internal RAM area can be set as the stack area.

[Operand format]

Identifier	Description
–	PUSH PSW AX/BC/DE/HL POP PSW AX/BC/DE/HL CALL/CALLT RET BRK RETB (Interrupt request generated) RETI

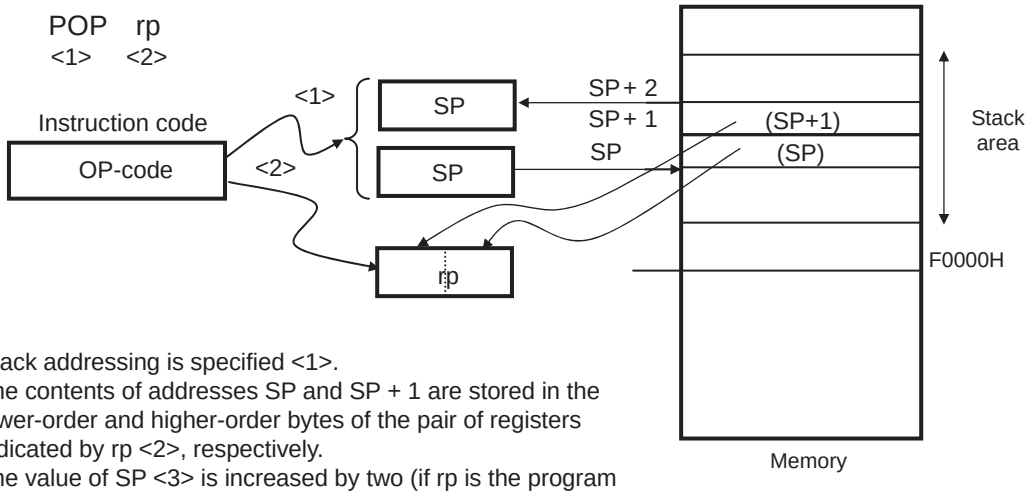
Each stack operation saves or restores data as shown in **Figures 3-33** to **3-38**.

Figure 3-33. Example of PUSH rp



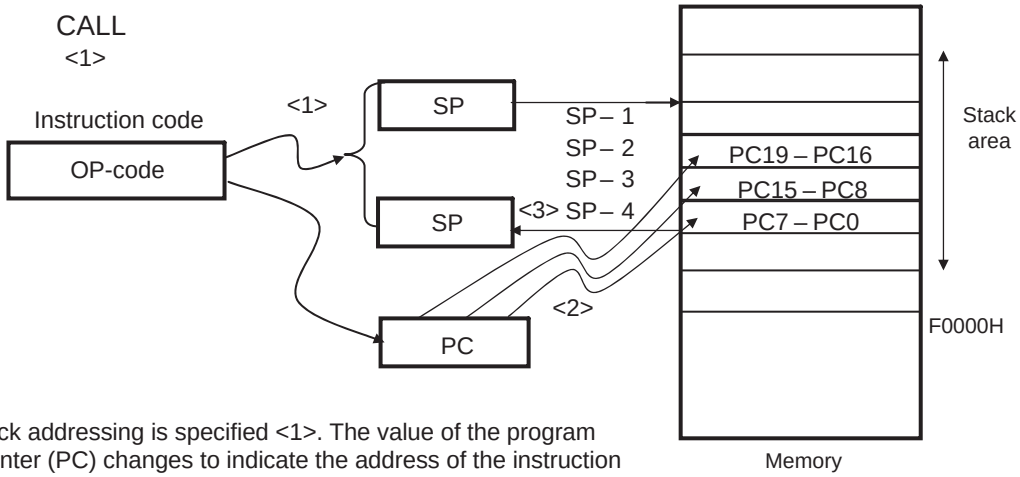
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP – 1 and SP – 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP – 1 and 0 is stored in SP – 2).

Figure 3-34. Example of POP



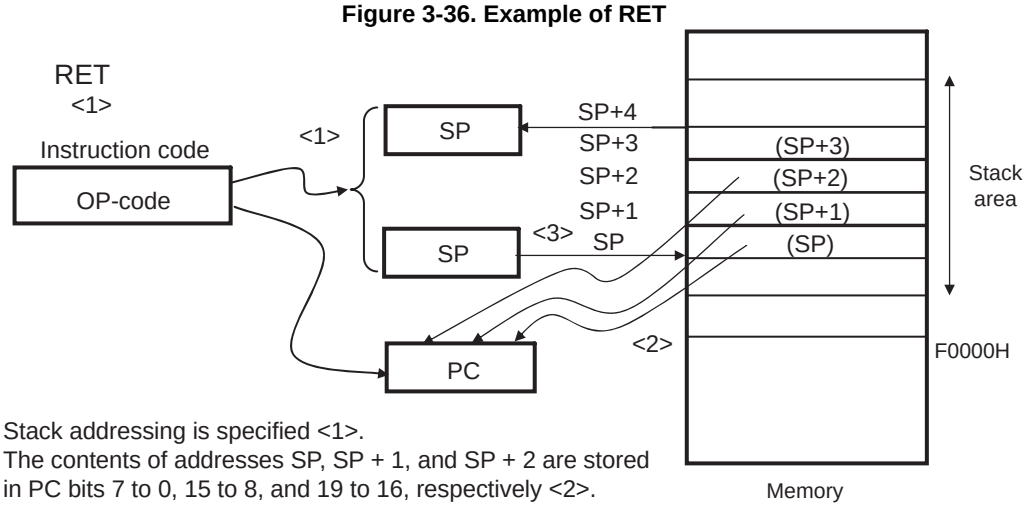
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP + 1 is stored in the PSW).

Figure 3-35. Example of CALL, CALLT



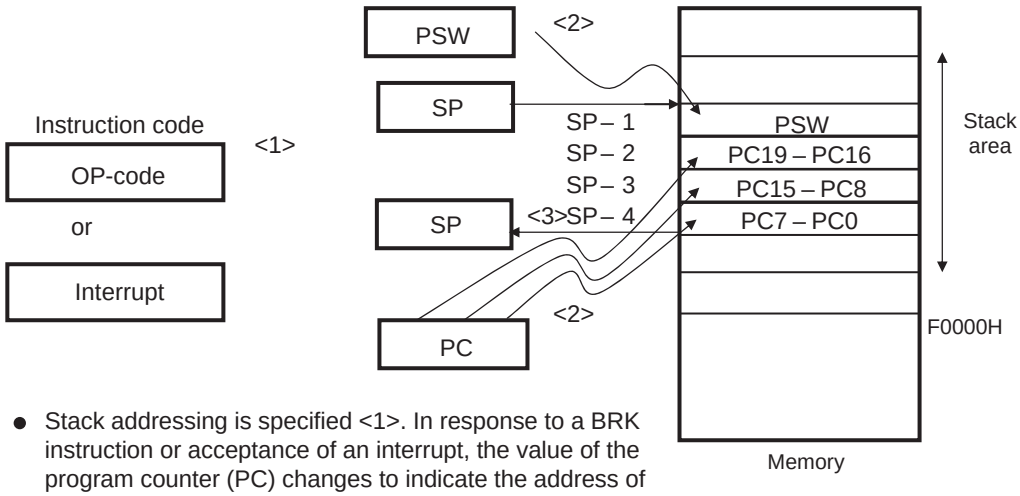
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- The values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP – 2, SP – 3, and SP – 4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3-36. Example of RET



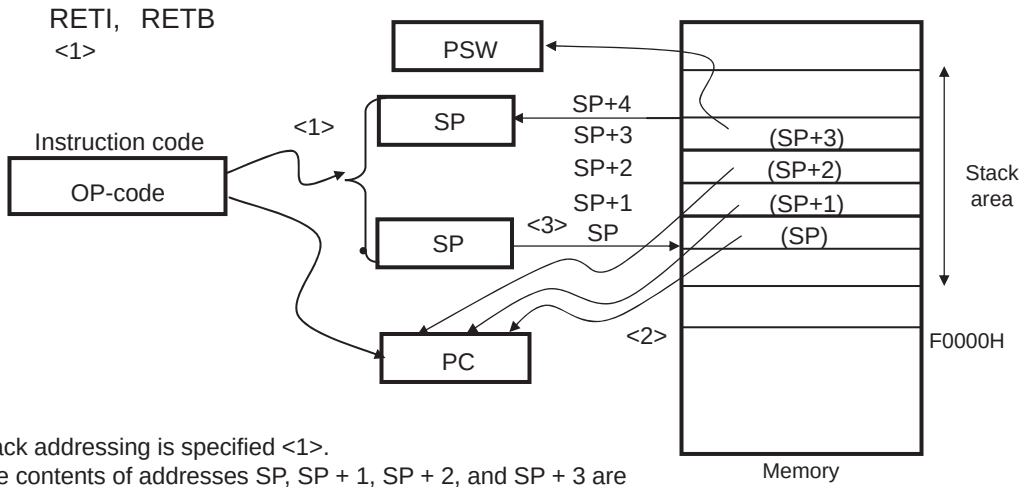
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3-37. Example of Interrupt, BRK



- Stack addressing is specified <1>. In response to a BRK instruction or acceptance of an interrupt, the value of the program counter (PC) changes to indicate the address of the next instruction.
- The values of the PSW, PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP – 1, SP – 2, SP – 3, and SP – 4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3-38. Example of RETI, RETB



- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, SP + 2, and SP + 3 are stored in PC bits 7 to 0, 15 to 8, 19 to 16, and the PSW, respectively <2>.
- The value of SP <3> is increased by four.

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The RL78 microcontrollers are provided with digital I/O ports, which enable variety of control operations.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **CHAPTER 2 PIN FUNCTIONS**.

4.2 Port Configuration

Ports include the following hardware.

Table 4-1. Port Configuration

Item	Configuration
Control registers	Port mode registers 0, 4 (PM0, PM4) Port registers 0, 4, 12, 13 (P0, P4, P12, P13) Pull-up resistor option registers 0, 4, 12 (PU0, PU4, PU12) Port output mode register 0 (POM0) Port mode control register 0 (PMC0) Peripheral I/O redirection register (PIOR)
Port	• 10-pin products Total: 8 (CMOS I/O: 6 (N-ch open-drain output (V_{DD} tolerance): 2), CMOS input: 2) • 16-pin products Total: 14 (CMOS I/O: 10 (N-ch open-drain output (V_{DD} tolerance): 4), CMOS input: 4)
On-chip pull-up resistor	• 10-pin products Total: 7 • 16-pin products Total: 11

4.2.1 Port 0

Port 0 is an I/O port with output latches. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P07 [Note] pins are used as input pins, use of the on-chip pull-up resistors can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

<R> Output from the P00, P01, P06, and P07 pins can be specified as N-ch open-drain (V_{DD} tolerant) in 1-bit units using port output mode register 0 (POM0).

This port can also be used for serial interface data I/O, clock I/O, analog input, key return input, clock/buzzer output, timer I/O, and external interrupt request input.

Reset signal generation sets P00 to input mode, and sets P01 to P07 to analog input mode.

Note For 10-pin products, P00 to P04; for 16-pin products, P00 to P07.

4.2.2 Port 4

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 and P41 pins [Note] are used as an input pin, use of the on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).

This port can also be used for data I/O for a flash memory programmer/debugger, key return input, clock/buzzer output, timer I/O, and external interrupt request input.

Note For 10-pin products, P40; for 16-pin products, P40 and P41.

4.2.3 Port 12

Port 12 is an input-only port. Use of an on-chip pull-up resistor can be specified for P125 using pull-up resistor option register 12 (PU12) (the on-chip pull-up resistor is always valid when $\overline{\text{RESET}}$ input is selected (PORTSELB = 1)).

This port can also be used for key return input, external interrupt request input, connecting resonator for main system clock, external clock input for main system clock, and reset input.

Note Once the power is turned on, P125 functions as the $\overline{\text{RESET}}$ input. The PORTSELB bit of the option byte (000C1H) defines whether this port operates as P125/KR1 or $\overline{\text{RESET}}$. When this pin is set to P125/KR1, do not input the low level to this pin during a reset by the selectable power-on-reset (SPOR) circuit and during the period from release from the reset by the SPOR circuit to the start of normal operation. If input of the low level continues during this period, the chip will remain in the reset state in response to the external reset. Accordingly, the pull-up resistor is enabled after power is turned on.

4.2.4 Port 13

Port 13 is an input-only port.

This port can also be used for timer input and external interrupt request input.

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers 0, 4 (PM0, PM4)
- Port registers 0, 4, 12, 13 (P0, P4, P12, P13)
- Pull-up resistor option registers 0, 4, 12 (PU0, PU4, PU12)
- Port output mode register 0 (POM0)
- Port mode control register 0 (PMC0)
- Peripheral I/O redirection register (PIOR)

Caution Which registers and bits are included depends on the product. For registers and bits mounted on each product, see Tables 4-2 and 4-3. Be sure to set bits that are not mounted to their initial values.

Table 4-2. Pm, PMn, PUy, POM0, PMC0 Registers and the Bits (10-pin Products)

Port		Bit name				
		Px register	PMx register	PUx register	POMx register	PMCx register
PORT0	0	P00	PM00	PU00	POM00	–
	1	P01	PM01	PU01	POM01	PMC01
	2	P02	PM02	PU02	–	PMC02
	3	P03	PM03	PU03	–	PMC03
	4	P04	PM04	PU04	–	PMC04
PORT4	0	P40	PM40	PU40	–	–
PORT12	5	P125	–	PU125	–	–
PORT13	7	P137	–	–	–	–

Table 4-3. Pm, PMn, PUy, POM0, PMC0 Registers and the Bits (16-pin Products)

Port		Bit name				
		Px register	PMx register	PUx register	POMx register	PMCx register
PORT0	0	P00	PM00	PU00	POM00	–
	1	P01	PM01	PU01	POM01	PMC01
	2	P02	PM02	PU02	–	PMC02
	3	P03	PM03	PU03	–	PMC03
	4	P04	PM04	PU04	–	PMC04
	5	P05	PM05	PU05	–	PMC05
	6	P06	PM06	PU06	POM06	PMC06
	7	P07	PM07	PU07	POM07	PMC07
PORT4	0	P40	PM40	PU40	–	–
	1	P41	PM41	PU41	–	–
PORT12	1	P121	–	–	–	–
	2	P122	–	–	–	–
	5	P125	–	PU125	–	–
PORT13	7	P137	–	–	–	–

Remark m = 0, 4, 12, 13
n = 0, 4
y = 0, 4, 12

The format of each register is described below.

4.3.1 Port mode registers 0, 4 (PM0, PM4)

These registers specify input or output mode for the port in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register by referring to **4.5 Register Settings When an Alternate Function Is Used** .

Figure 4-1. Format of Port Mode Registers 0, 4 (PM0, PM4)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FFF24H	FFH	R/W

PMmn	Pmn pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

m = 0, 4; n = 0 to 7

Caution Be sure to set bits that are not mounted to their initial values.

4.3.2 Port registers 0, 4, 12, 13 (P0, P4, P12, P13)

These registers set the output latch value of a port.

If the data is read in the input mode, the pin level is read. If it is read in the output mode, the output latch value is read [Note].

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets the P12 and P13 registers to the undefined value, and clears the other registers to 00H.

Note When a pin that is set as an analog input pin (PMC0x = 1, PM0x = 1) is read, the value read is always 0 regardless of the input signal level on the pin.
When the data bit for P125 is read while the setting for the P125/KR1/$\overline{\text{RESET}}$ pin is $\overline{\text{RESET}}$ input (PORTSELB = 1), the value read is always 1.

Figure 4-2. Format of Port Registers 0, 4, 12, 13 (P0, P4, P12, P13)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	P04	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W
P4	0	0	0	0	0	0	0	P40	FFF04H	00H (output latch)	R/W
P12	0	0	P125	0	0	0	0	0	FFF0CH	Undefined	R
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	P07	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H (output latch)	R/W
P12	0	0	P125	0	0	P122	P121	0	FFF0CH	Undefined	R
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R

Pmn	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

m = 0, 4, 12, 13; n = 0 to 7

Caution Be sure to set bits that are not mounted to their initial values.

4.3.3 Pull-up resistor option registers 0, 4, 12 (PU0, PU4, PU12)

These registers specify whether the on-chip pull-up resistors are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits that satisfy the following usage conditions for the pins to which the use of an on-chip pull-up resistor has been specified in these registers.

Usage conditions of the on-chip pull-up resistor:

- PMmn = 1 (Input mode)
- <R> PMCmn = 0 (Digital I/O)
- POM0n = 0 (Normal output mode)

On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of these registers.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PU4 to 01H, PU12 to 20H, and clears PU0 to 00H.

Figure 4-3. Format of Pull-up Resistor Option Registers 0, 4, 12 (PU0, PU4, PU12)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W
PU12	0	0	PU125 Note	0	0	0	0	0	F003CH	20H	R/W

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	01H	R/W
PU12	0	0	PU125 Note	0	0	0	0	0	F003CH	20H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

m = 0, 4, 12; n = 0 to 7

Note This bit can be only manipulated when the P125/KR1 function is selected (PORTSELB = 0) (the on-chip pull-up resistor is always valid (PU125 = 1) when the $\overline{\text{RESET}}$ input (PORTSELB = 1) is selected).

Caution Be sure to set bits that are not mounted to their initial values.

4.3.4 Port output mode register 0 (POM0)

This register sets CMOS output or N-ch open drain output in 1-bit units.

N-ch open drain output (V_{DD} tolerant) mode can be selected for the SDA00 pin during simplified I^2C communication with an external device.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Caution An on-chip pull-up resistor is not connected to a bit for which N-ch open drain output (V_{DD} tolerance) mode (POM0n =1) is set.

Figure 4-4. Format of Port Output Mode Register 0 (POM0)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	0	0	0	POM01	POM00	F0050H	00H	R/W

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	POM07	POM06	0	0	0	0	POM01	POM00	F0050H	00H	R/W

POM0n	P0n pin output mode selection
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerant) mode

n = 0, 1, 6, 7

Caution Be sure to set bits that are not mounted to their initial values.

4.3.5 Port mode control register 0 (PMC0)

This register sets the digital I/O or analog input in 1-bit units.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 4-5. Format of Port Mode Control Register 0 (PMC0)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	1	1	1	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W

PMC0n	P0n pin digital I/O/analog input selection
0	Digital I/O (alternate function other than analog input)
1	Analog input

n = 1 to 7

Cautions 1. Select input mode by using port mode register 0 (PM0) for the ports which are set by the PMC0 register as analog input.

2. Be sure to set bits that are not mounted to their initial values.

4.3.6 Peripheral I/O redirection register (PIOR)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.
This function is used to switch ports to which alternate functions are assigned.
Use the PIOR register to assign a port to the function to redirect and enable the function.
In addition, can be changed the settings for redirection until its function enable operation.
The PIOR register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 4-6. Format of Peripheral I/O Redirection Register (PIOR)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIOR	0	0	0	0	0	PIOR2	PIOR1	PIOR0	F0077H	00H	R/W

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIOR	0	0	0	PIOR4	PIOR3	PIOR2	PIOR1	PIOR0	F0077H	00H	R/W

Bit	Function	Setting value	
		0	1
PIOR4 Note	INTP3	P06	P121
PIOR3 Note	INTP2	P41	P122
PIOR2	INTP1	P00	P03
PIOR1	TI01/TO01	P04	P40
PIOR0	PCLBUZ0	P02	P40

Note 16-pin products only.

Cautions 1. It is prohibited to set PIOR0 and PIOR1 to 1 at the same time.
2. Be sure to set bits that are not mounted to their initial values.

4.4 Port Function Operations

Port operations differ depending on whether the input or output mode is set, as shown below.

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change. Therefore, byte data can be written to the ports used for both input and output.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

4.4.2 Reading from I/O port

(1) Output mode

The output latch contents are read by a transfer instruction. The output latch contents do not change.

(2) Input mode

The pin status is read by a transfer instruction. The output latch contents do not change.

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change. Therefore, byte data can be written to the ports used for both input and output.
The data of the output latch is cleared when a reset signal is generated.

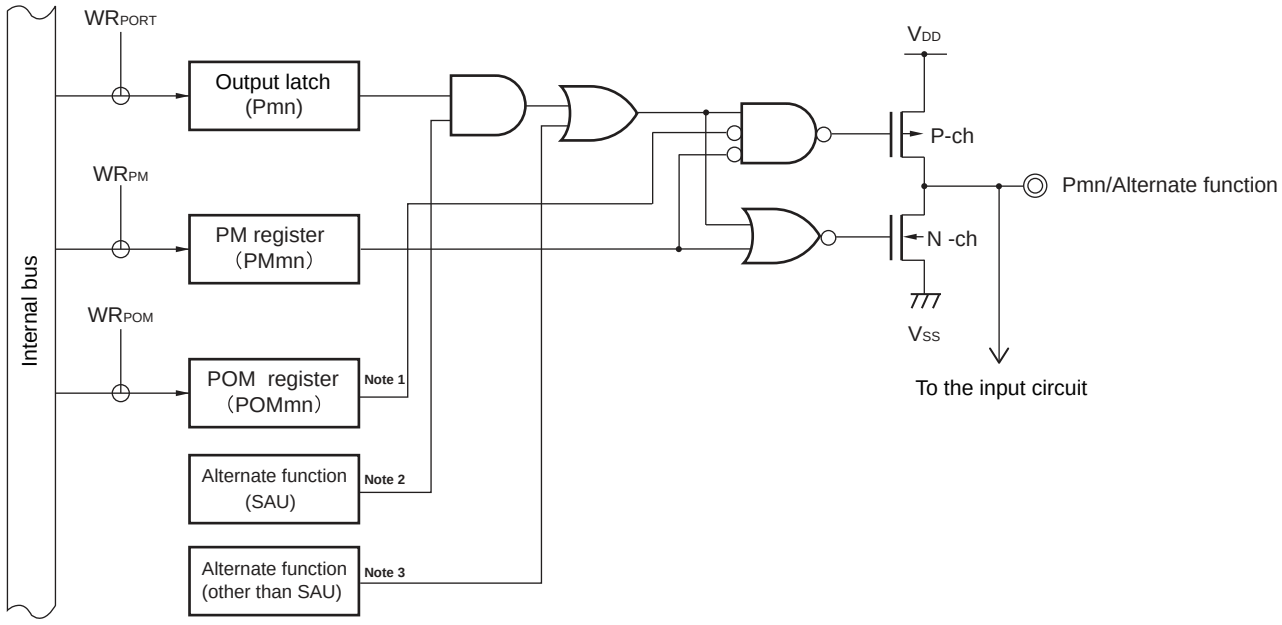
4.5 Register Settings When an Alternate Function Is Used

4.5.1 Basic concepts on using an alternate function

If a given pin is also used alternately for analog input, first in the port mode control register 0 (PMC0) specify whether the pin is to be used in analog input or digital output.

The basic configuration of an output circuit for pins that are used in digital I/O is shown in Figure 4-7. The output from the SAU function doubling as an output from the port output latch is input into the AND gate. The output from the AND gate is input into the OR gate. To the other input pin for the OR gate, the outputs from alternate non-SAU functions (TAU, clock/buzzer output, IICA, etc.) are connected. When such a pin is used as a port or alternate function, the alternate function that is not used must not interfere with the output from the function to be used. Table 4-4 summarizes underling concepts of specifying basic settings for making that distinction.

Figure 4-7. Basic Configuration of the Output Circuit for the Pins



Notes 1. In the absence of a POM register, this signal should be considered Low (0).

2. In the absence of an alternate function, this signal should be considered High (1).

3. In the absence of an alternate function, this signal should be considered Low (0).

Remark m: port number (m = 0, 4, 12, 13); n: bit number (n = 0 to 7)

Table 4-4. Basic Settings

Output function of the pin used	Output settings for alternate functions that are not used		
	Port function	SAU output function	Non-SAU output function
Port output function	–	Output: High (1)	Output: Low (0)
SAU output function	High(1)	–	Output: Low (0)
Non-SAU output function	Low(0)	Output: High (1)	Output: Low (0) Note

Note Since more than one non-SAU output function can be assigned to a given pin, the output from an alternate function that is not used must be set to Low (0). For specific settings methods, see **4.5.2 Register settings for alternate functions that do not use an output function**.

4.5.2 Register settings for alternate functions that do not use an output function

If the output from an alternate function associated with a pin is not used, the settings described below must be specified. If the pin is subject to a peripheral I/O redirect function, the output can be changed to another pin by setting the peripheral I/O redirection register (PIOR). In this manner, the port function or another alternate function that is assigned to the target pin can be used.

(1) SOp = 1/TxDq = 1 (when not using the serial output (SOp/TxDq) of SAU)

In situations where serial output (SOp/TxDq) is not used, such as when SAU is used exclusively for serial input, set the bits in the serial output enable register 0 (SOE0) associated with the output that is not used to 0 (output disabled), and the SO0n bit in the serial output register 0 (SO0) to 1 (High). This is the same as the default settings.

(2) SCKp = 1/SDAr = 1/SCLr = 1 (when not using the channel n of SAU)

When not using SAU, set the bit n (SE0n) in the serial channel enable status register 0 (SE0) to 0 (operation halted status), set the bits in the serial output enable register 0 (SOE0) associated with the output that is not used to 0 (output disabled), and the SO0n and CKO0n bits in the serial output register 0 (SO0) to 1 (High). This is the same as the default settings.

(3) TO0n = 0 (when not using the output from the channel n of TAU)

When not using the TO0n output from TAU, set the bits in the timer output enable register 0 (TOE0) associated with the output that is not used to 0 (output disabled), and the bits in the timer output register 0 (TO0) to 0 (Low). This is the same as the default settings.

(4) SDAA0 = 0/SCLA0 = 0 (when not using IICA)

When not using IICA, set the IICE0 bit in the IICA control register 00 (IICCTL00) to 0 (operation halted). This is the same as the default settings.

(5) PCLBUZ0 = 0 (when not using the clock output/buzzer output)

When not using the clock output/buzzer output, set the PCLOE0 bit in the clock output selection register 0 (CKS0) to 0 (output disabled). This is the same as the default settings.

<R> **Remark** p: CSI number (p = 00, 01), q: UART number (q = 0), r: IIC number (r = 00)

4.5.3 Example of register settings for port and alternate functions used

Table 4-5 shows examples of register settings for port and alternate functions that are used. Registers that control the port functions should be set as indicated in Table 4-5. For conventions used in Table 4-5, see the remarks provided below:

Remark

–:	Excluded
×:	don't care
PIOR:	Peripheral I/O redirection register
POM0:	Port output mode register 0
PMC0:	Port mode control register 0
PMn:	Port mode register n (n = 0, 4)
Pm:	Port output latch (m = 0, 4, 12, 13)

Functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

Table 4-5. Examples of Register And Output Latch Settings With Pin Functions (1/4)

Pin	Function		PIOR	POM0	PMC0	PMn	Pm	Alternate function output		16 pins	10 pins
	Name	I/O						SAU output function	Non-SAU		
P00	P00	Input	–	×	–	1	×	×	–	√	√
		Output	–	0	–	0	0/1	TxD0/SO00 = 1	–		
		N-ch open-drain output	–	1	–	0	0/1				
	SO00	Output	–	0	–	0	1	×	–	√	√
	TXD0	Output	–	0/1	–	0	1	×	–	√	√
	INTP1	Input	PIOR2 = 0	×	–	1	×	×	–	√	√
P01	P01	Input	–	×	0	1	×	×	–	√	√
		Output	–	0	0	0	0/1	SDA00 = 1	–		
		N-ch open-drain output	–	1	0	0	0/1				
	ANI0	Analog input	–	×	1	1	×	×	–	√	√
	SI00	Input	–	×	0	1	×	×	–	√	√
	RxD0	Input	–	×	0	1	×	×	–	√	√
	SDA00	I/O	–	1	0	0	1	×	–	√	√
	KR2	Input	–	×	0	1	×	×	–	√	√
P02	P02	Input	–	–	0	1	×	×	×	√	√
		Output	–	–	0	0	0/1	SCK00/SCL00 = 1	PCLBUZ0 = 0 VCOUT0 = 0 Note		
	ANI1	Analog input	–	–	1	1	×	×	×	√	√
	SCK00	Input	–	–	0	1	×	×	×	√	√
		Output	–	–	0	0	1	×	PCLBUZ0 = 0 VCOUT0 = 0 Note		
	SCL00	Output	–	–	0	0	1	×	PCLBUZ0 = 0 VCOUT0 = 0 Note	√	√
	PCLBUZ0	Output	PIOR0 = 0	–	0	0	0	SCK00/SCL00 = 1	VCOUT0 = 0 Note	√	√
	KR3	Input	–	–	0	1	×	×	×	√	√
	VCOUT0	Output	–	–	0	0	0	SCK00/SCL00 = 1	PCLBUZ0 = 0	√	–
P03	P03	Input	–	–	0	1	×	–	×	√	√
		Output	–	–	0	0	0/1	–	TO00 = 0		
	ANI2	Analog input	–	–	1	1	×	–	×	√	√
	TO00	Output	–	–	0	0	0	–	×	√	√
	KR4	Input	–	–	0	1	×	–	×	√	√
	(INTP1)	Input	PIOR2 = 1	–	0	1	×	–	×	√	√
	IVCMP0	Input	–	–	1	1	×	–	×	√	–
P04	P04	Input	–	–	0	1	×	–	×	√	√
		Output	–	–	0	0	0/1	–	TO01 = 0		
	ANI3	Analog input	–	–	1	1	×	–	×	√	√
	TI01	Input	PIOR1 = 0	–	0	1	×	–	×	√	√
	TO01	Output	PIOR1 = 0	–	0	0	0	–	×	√	√
	KR5	Input	–	–	0	1	×	–	×	√	√
	IVREF0	Input	–	–	1	1	×	–	×	√	–

Note 16-pin products only.

Table 4-5. Examples of Register And Output Latch Settings With Pin Functions (2/4)

Pin	Function		PIOR	POM0	PMC0	PMn	Pm	Alternate function output		16 pins	10 pins
	Name	I/O						SAU output function	Non-SAU		
P05	P05	Input	–	–	0	1	×	×	×	√	–
		Output	–	–	0	0	0/1	SO01 = 1	TO02 = 0		
	ANI4	Analog input	–	–	1	1	×	×	×	√	–
	TI02	Input	–	–	0	1	×	×	×	√	–
	TO02	Output	–	–	0	0	0	SO01 = 1	×	√	–
	SO01	Output	–	–	0	0	1	×	TO02 = 0	√	–
P06	P06	Input	–	×	0	1	×	–	×	√	–
		Output	–	0	0	0	0/1	–	SCLA0 = 0		
		N-ch open-drain output	–	1	0	0	0/1				
	ANI5	Analog input	–	×	1	1	×	–	×	√	–
	SI01	Input	–	×	0	1	×	–	×	√	–
<R>	SCLA0	I/O	–	1	0	0	0	–	×	√	–
	INTP3	Input	PIOR4 = 0	×	0	1	×	–	×	√	–
P07	P07	Input	–	×	0	1	×	×	×	√	–
		Output	–	0	0	0	0/1	SCK01 = 1	TO03 = 0 SDAA0 = 0		
		N-ch open-drain output	–	1	0	0	0/1				
	ANI6	Analog input	–	×	1	1	×	×	×	√	–
	TO03	Output	–	0/1	0	0	0	SCK01 = 1	SDAA0 = 0	√	–
	SCK01	Input	–	×	0	1	×	×	×	√	–
		Output	–	0	0	0	1	×	TO03 = 0 SDAA0 = 0		
<R>	SDAA0	I/O	–	1	0	0	0	SCK01 = 1	TO03 = 0	√	–
P40	P40	Input	–	–	–	1	×	–	×	√	√
		Output	–	–	–	0	0/1	–	(PCLBUZ0) = 0 (TO01) = 0		
	KR0	Input	–	–	–	1	×	–	×	√	√
	(PCLBUZ0)	Output	PIOR0 = 1	–	–	0	0	–	(TO01) = 0	√	√
	(TI01)	Input	PIOR1 = 1	–	–	1	×	–	×	√	√
	(TO01)	Output	PIOR1 = 1	–	–	0	0	–	(PCLBUZ0) = 0	√	√
P41	P41	Input	–	–	–	1	×	–	–	√	–
		Output	–	–	–	0	0/1	–	–		
	TI03	Input	–	–	–	1	×	–	–	√	–
	INTP2	Input	PIOR3 = 0	–	–	1	×	–	–	√	–

Table 4-5. Examples of Register And Output Latch Settings With Pin Functions (3/4)

Pin	Function		PIOR	POM0	PMC0	PMn	Pm	CMC	16	10
	Name	I/O						(EXCLK,OSCSEL)	pins	pins
P121	P121	Input	–	–	–	–	×	00/10/11	√	–
	X1	Input	–	–	–	–	×	01	√	–
	(INTP3)	Input	PIOR4 = 1	–	–	–	×	00/10/11	√	–
P122	P122	Input	–	–	–	–	×	00/10	√	–
	X2	Input	–	–	–	–	×	01	√	–
	EXCLK	Input	–	–	–	–	×	11	√	–
	(INTP2)	Input	PIOR3 = 1	–	–	–	×	00/10	√	–

Table 4-5. Examples of Register And Output Latch Settings With Pin Functions (4/4)

Pin	Function		PIOR	POM0	PMC0	PMn	Pm	Notes	16	10
	Name	I/O							pins	pins
P125	P125	Input	–	–	–	–	×	Optional bytes 000C1H	√	√
	KR1	Input	–	–	–	–	×	PORTSELB = 0	√	√
	$\overline{\text{RESET}}$	Input	–	–	–	–	×	Optional bytes 000C1H PORTSELB = 1	√	√
P137	P137	Input	–	–	–	–	×	–	√	√
	TI00	Input	–	–	–	–	×	–	√	√
	INTP0	Input	–	–	–	–	×	–	√	√

4.6 Cautions When Using Port Function

4.6.1 Cautions on 1-bit manipulation instruction for port register n (Pn)

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit.

Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

Example When P00 is an output port, P01 to P07 are input ports (all pin statuses are high level), and the port latch value of port 0 is 00H, if the output of output port P00 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 0 is FFH.

Explanation: The targets of writing to and reading from the Pn register of a port whose PMmn bit is 1 are the output latch and pin status, respectively.

A 1-bit manipulation instruction is executed in the following order in the RL78 microcontroller.

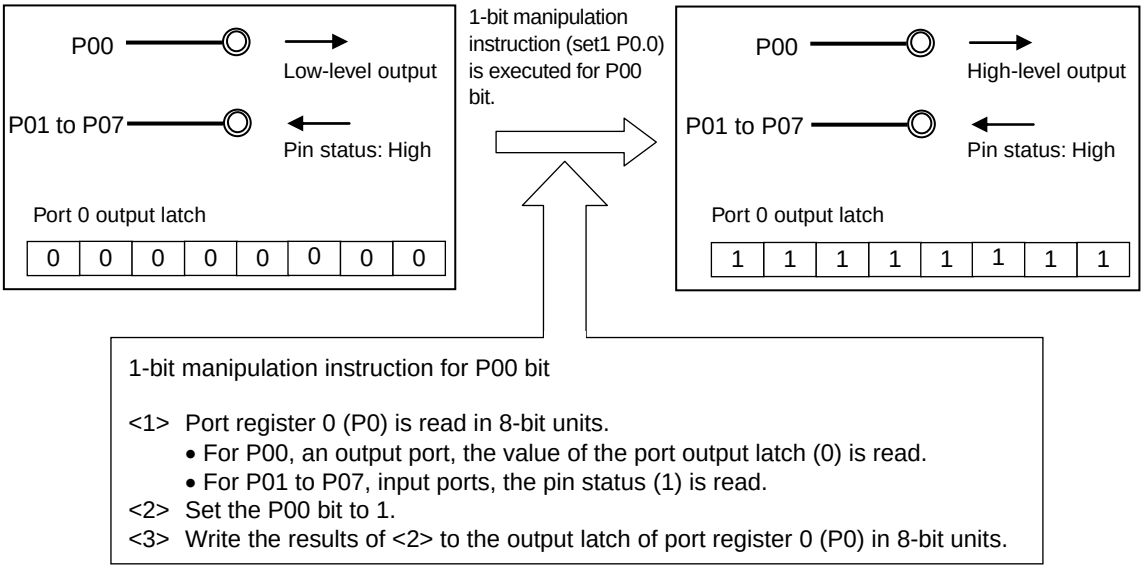
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P00, which is an output port, is read, while the pin statuses of P01 to P07, which are input ports, are read. If the pin statuses of P01 to P07 are high level at this time, the read value is FEH.

The value is changed to FFH by the manipulation in <2>.

FFH is written to the output latch by the manipulation in <3>.

Figure 4-8. Bit Manipulation Instruction (P00)



4.6.2 Notes on specifying the pin settings

For an output pin to which multiple alternate functions are assigned, the output of the unused alternate functions must be set to its initial state so as to prevent conflicting outputs. This also applies to the functions assigned by using the peripheral I/O redirection register (PIOR). For details about the alternate output function, see **4.5 Register Settings When an Alternate Function Is Used**.

No specific setting is required for input pins because the output of their alternate functions is disabled (the buffer output is Hi-Z).

Disabling the unused functions, including blocks that are only used for input or do not have I/O, is recommended for lower power consumption.

CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following three kinds of system clocks and clock oscillators are selectable.

(1) Main system clock

<1> X1 oscillator (16-pin products only)

This circuit oscillates a clock of f_X = 1 to 20 MHz by connecting a resonator to X1 and X2 pins.
The external main system clock (f_{EX} = 1 to 20 MHz) can also be supplied from EXCLK/X2/P122 pin.
Oscillation can be stopped or the external main system clock input can be disabled by executing the STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

<2> High-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{IH} = 20/10/5/2.5/1.25 MHz (typ.) by using the option byte (000C2H). After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing the STOP instruction or setting the HIOSTOP bit (bit 0 of the CSC register).
The frequency specified by using an option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 5-9 Format of High-Speed On-Chip Oscillator Frequency Selection Register (HOCODIV)**.
The frequencies that can be specified for the high-speed on-chip oscillator by using the option byte and the high-speed on-chip oscillator frequency select register (HOCODIV) are shown below.

Power Supply Voltage	Oscillation Frequency (MHz)				
	1.25	2.5	5	10	20
2.7 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√
2.0 V ≤ V_{DD} ≤ 5.5 V **Note**	√	√	√	–	–

Remark √: Can operate, -: Cannot operate

As the main system clock, a high-speed system clock (X1 clock) or high-speed on-chip oscillator clock can be selected by setting of the MCM0 bit (bit 4 of the system clock control register (CKC)).

Note Use this product within the voltage range from 2.25 to 5.5 V because the detection voltage (V_{SPOR}) of the selectable power-on-reset (SPOR) circuit should also be considered.

(2) Low Speed On-chip Oscillator clock

This circuit oscillates a clock of f_{IL} = 15 kHz (typ.).
The low speed on-chip oscillator clock cannot be used as the CPU clock.
Only the following peripheral hardware runs on the low speed on-chip oscillator clock.

- Watchdog timer
- 12-bit Interval timer [Note]

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the operation speed mode control register (OSMC) [Note], or both are set to 1.
However, when WDTON = 1, WUTMMCK0 = 0, and bit 0 (WDSTBYON) of the option byte (000C0H) is 0, oscillation of the LOCO stops if the HALT or STOP instruction is executed.

Note 16-pin products only.

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{IL}: Low speed on-chip oscillator clock frequency

5.2 Configuration of Clock Generator

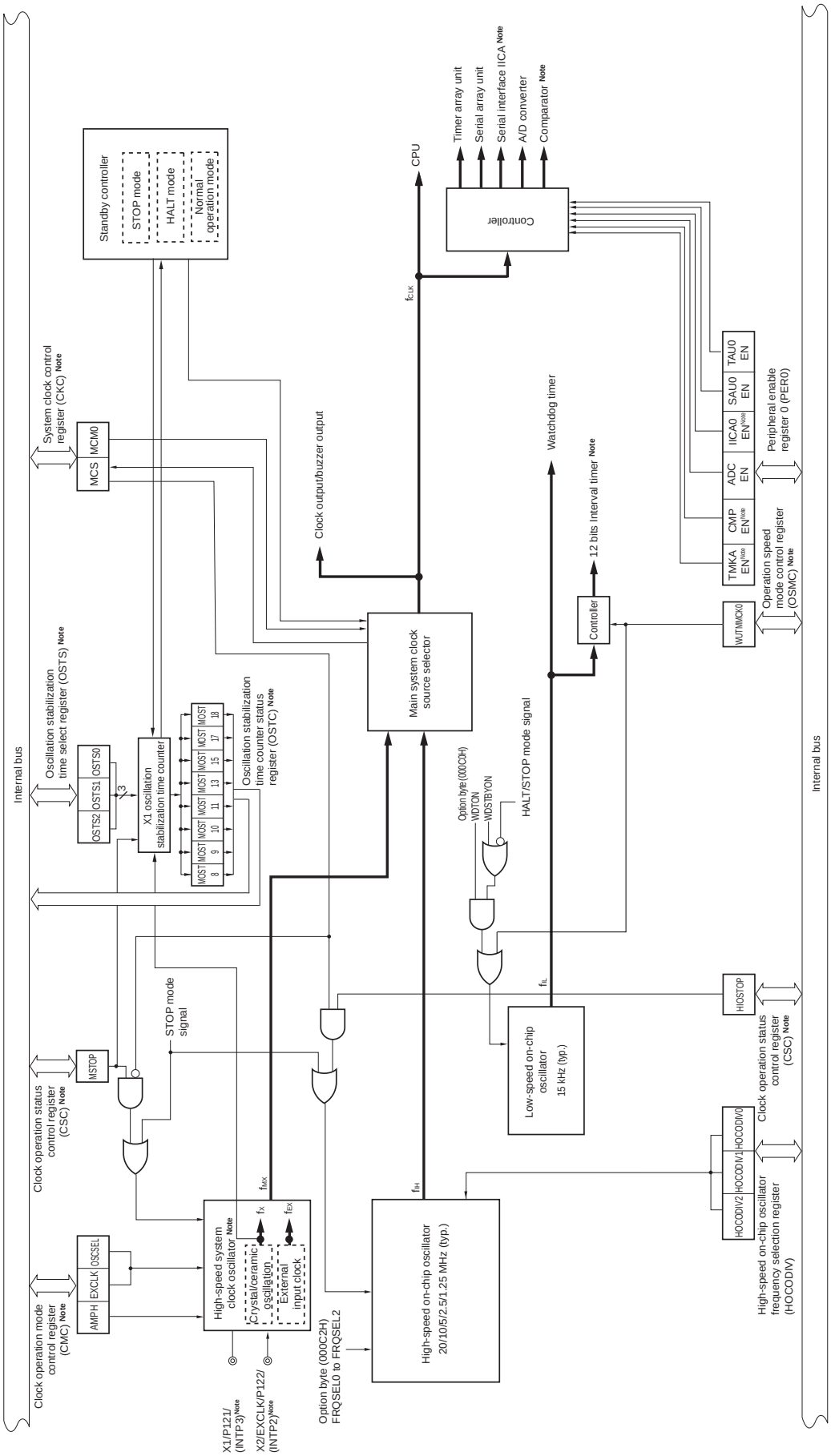
The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration	10-pin products	16-pin products
Control registers	Clock operation mode control register (CMC)	–	√
	System clock control register (CKC)	–	√
	Clock operation status control register (CSC)	–	√
	Oscillation stabilization time counter status register (OSTC)	–	√
	Oscillation stabilization time select register (OSTS)	–	√
	Peripheral enable register 0 (PER0)	√	√
	Operation speed mode control register (OSMC)	–	√
	High-speed on-chip oscillator frequency selection register (HOCODIV)	√	√
Oscillators	X1 oscillator	–	√
	High-speed on-chip oscillator	√	√
	Low-speed on-chip oscillator	√	√

Remark √: Provided, –: Not provided

Figure 5-1. Block Diagram of Clock Generator



Note 16-pin products only.

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency

5.3 Registers Controlling Clock Generator

The clock generator is controlled by the following registers depending on the products.

(1) 10-pin products
- Peripheral enable register 0 (PER0)
- High-speed on-chip oscillator frequency selection register (HOCODIV)

(2) 16-pin products
- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable register 0 (PER0)
- Operation speed mode control register (OSMC)
- High-speed on-chip oscillator frequency selection register (HOCODIV)

Caution The registers and bits mounted depend on the products. Be sure to set the initial value in the registers and bits not mounted.

5.3.1 Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/P121/(INTP3) and X2/EXCLK/P122/(INTP2) pins, and to select a gain of the oscillator.

The CMC register can be written only once by an 8-bit memory manipulation instruction after reset release. This register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5-2. Format of Clock Operation Mode Control Register (CMC)

Address: FFFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	0	0	0	0	0	AMPH

EXCLK	OSCSEL	High-speed system clock pin operation mode	X1/P121/(INTP3) pin	X2/EXCLK/P122/(INTP2) pin
0	0	Input port mode	Input port	
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

AMPH	Control of X1 clock oscillation frequency
0	1 MHz ≤ f_X ≤ 10 MHz
1	10 MHz < f_X ≤ 20 MHz

Cautions

1. **The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. When using the CMC register with its initial value (00H), be sure to set the register to 00H after a reset ends in order to prevent malfunction due to a program loop. Such a malfunction becomes unrecoverable when a value other than 00H is mistakenly written.**
2. **After reset release, set the CMC register before X1 oscillation is started as set by the clock operation status control register (CSC).**
3. **Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz. Specify the settings for the AMPH bits while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX}).**
4. **Switch the operation mode of the X1/X2 pins only when MSTOP = 1.**

Remark f_X: X1 clock frequency

5.3.2 System clock control register (CKC)

This register is used to select a main system clock.

The CKC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 5-3. Format of System Clock Control Register (CKC)

Address: FFFA4H After reset: 00H R/W [Note]

Symbol	7	6	<5>	<4>	3	2	1	0
CKC	0	0	MCS	MCM0	0	0	0	0

MCS	Status of Main system clock (f_{MAIN})
0	High-speed on-chip oscillator clock (f_{IH})
1	High-speed system clock (f_{MX})

MCM0	Main system clock (f_{MAIN}) operation control
0	Selects the high-speed on-chip oscillator clock (f_{IH}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

Note Bit 5 is read-only.

Cautions 1. Be sure to clear bits 0 to 3, 6, and 7 to 0.

2. Do not select the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN}) before the oscillation stabilization time has elapsed after oscillation of f_{MX} is started.

3. When the main system clock (f_{MAIN}) is changed, the peripheral hardware clock also changes at the same time. Only change f_{MAIN} after stopping all peripheral functions and setting the MCM0 bit.

5.3.3 Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock and high-speed on-chip oscillator clock, (except the low-speed on-chip oscillator clock).

The CSC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 80H.

Figure 5-4. Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H After reset: 80H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
CSC	MSTOP	0	0	0	0	0	0	HIOSTOP

MSTOP	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Input port mode
0	X1 oscillator operating	An external clock from the EXCLK pin enabled	Input port
1	X1 oscillator stopped	An external clock from the EXCLK pin disabled	

HIOSTOP	High-speed on-chip oscillator clock operation control
0	High-speed on-chip oscillator clock operating
1	High-speed on-chip oscillator clock stopped

Cautions

1. **After reset release, set the clock operation mode control register (CMC) before setting the CSC register.**
2. **Switch the operation mode of the X1/X2 pins only when MSTOP = 1.**
3. **When setting MSTOP bit to 0, switch the X1/X2 pins to the f_X operation mode beforehand. Setting the MSTOP flag is disabled in the input port mode.**
4. **Set the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after releasing reset. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.**
5. **To start X1 oscillation as set by the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).**
6. **When setting MSTOP bit to 1 in the f_X operation mode, make sure that MCS in the CKC register is 0 beforehand.**
7. **In the f_X operation mode, writing to the MSTOP flag is enabled but the stop control is not performed.**
8. **Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the CSC register.**
9. **The setting of the flags of the register to stop clock oscillation and the condition before clock oscillation is to be stopped are as Table 5-2.**
   Before stopping the clock oscillation, check the conditions before the clock oscillation is stopped.

Table 5-2. Condition Before Stopping Clock Oscillation and Flag Setting

Clock	Condition Before Stopping Clock	Setting of CSC Register Flags
X1 clock	CPU and peripheral hardware clocks operate with a high-speed on-chip oscillator clock. (MCS = 0)	MSTOP = 1
External main system clock		
High-speed on-chip oscillator clock	CPU and peripheral hardware clocks operate with a high-speed system clock.(MCS = 1)	HIOSTOP = 1

5.3.4 Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.

The X1 clock oscillation stabilization time can be checked in the following case:

- If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.
- If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.

When reset signal is generated, the STOP instruction and MSTOP (bit 7 of clock operation status control register (CSC)) = 1 clear the OSTC register to 00H.

Remark The oscillation stabilization time counter starts counting in the following cases.
- When oscillation of the X1 clock starts (EXCLK, OSCSEL = 0, 1 → MSTOP = 0)
- When the STOP mode is released

Figure 5-5. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

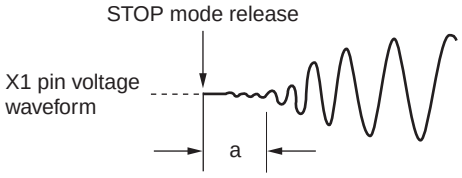
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									f_X = 10 MHz	f_X = 20 MHz
0	0	0	0	0	0	0	0	$(2^8+16)/f_X$ max.	27.2 μs max.	13.6 μs max.
1	0	0	0	0	0	0	0	$(2^8+16)/f_X$ min.	27.2 μs min.	13.6 μs min.
1	1	0	0	0	0	0	0	$(2^9+16)/f_X$ min.	52.8 μs min.	26.4 μs min.
1	1	1	0	0	0	0	0	$(2^{10}+16)/f_X$ min.	104 μs min.	52.0 μs min.
1	1	1	1	0	0	0	0	$(2^{11}+16)/f_X$ min.	206 μs min.	103 μs min.
1	1	1	1	1	0	0	0	$(2^{13}+16)/f_X$ min.	820 μs min.	410 μs min.
1	1	1	1	1	1	0	0	$(2^{15}+16)/f_X$ min.	3.27 ms min.	1.63 ms min.
1	1	1	1	1	1	1	0	$(2^{17}+16)/f_X$ min.	13.1 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$(2^{18}+16)/f_X$ min.	26.2 ms min.	13.1 ms min.

Cautions 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).
In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.**
  (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)

3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).



Remark fx: X1 clock oscillation frequency

5.3.5 Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released.

When the X1 clock is selected as the CPU clock, the operation automatically waits for the time set using the OSTS register after the STOP mode is released.

The oscillation stabilization time can be checked up to the time set using the OSTC register.

The OSTS register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets the OSTS register to 07H.

Figure 5-6. Format of Oscillation Stabilization Time Select Register (OSTS)

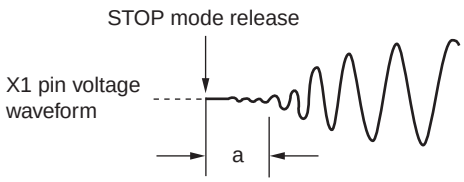
Address: FFFA3H After reset: 07H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$(2^8+16)/f_X$	27.2 μs	13.6 μs
0	0	1	$(2^9+16)/f_X$	52.8 μs	26.4 μs
0	1	0	$(2^{10}+16)/f_X$	104 μs	52.0 μs
0	1	1	$(2^{11}+16)/f_X$	206 μs	103 μs
1	0	0	$(2^{13}+16)/f_X$	820 μs	410 μs
1	0	1	$(2^{15}+16)/f_X$	3.27 ms	1.63 ms
1	1	0	$(2^{17}+16)/f_X$	13.1 ms	6.55 ms
1	1	1	$(2^{18}+16)/f_X$	26.2 ms	13.1 ms

Cautions

1. **To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.**
2. **Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.**
3. **Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.**
4. **The oscillation stabilization time counter counts up to the oscillation stabilization time set by the OSTS register.**
   In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts.
   - **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock is being used as the CPU clock.**
   - **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating. (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)**
5. **The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).**



Remark f_X: X1 clock oscillation frequency

5.3.6 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use the peripheral functions below, which are controlled by this register, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

- 12-bit Interval timer
- A/D converter
- Comparator
- Serial interface IICA
- Serial array unit
- Timer array unit

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (1/2)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER0	TMKAEN Note	CMPEN Note	ADCEN	IICA0EN Note	0	SAU0EN	0	TAU0EN

TMKAEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the 12-bit interval timer cannot be written. • The 12-bit interval timer is in the reset status.
1	Enables input clock supply. • SFR used by the 12-bit interval timer can be read and written.

CMPEN	Control of comparator input clock supply
0	Stops input clock supply. • SFR used by the comparator cannot be written. • The comparator is in the reset status.
1	Enables input clock supply. • SFR used by the comparator can be read and written.

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. • SFR used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables input clock supply. • SFR used by the A/D converter can be read and written.

Note 16-pin products only.

Caution Be sure to clear the following bits to 0.
10-pin products: Bits 1, 3, 4, 6, and 7
16-pin products: Bits 1 and 3

Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (2/2)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER0	TMKAEN Note	CMPEN Note	ADCEN	IICA0EN Note	0	SAU0EN	0	TAU0EN

IICA0EN	Control of serial interface IICA input clock supply
0	Stops input clock supply. • SFR used by the serial interface IICA cannot be written. • The serial interface IICA is in the reset status.
1	Enables input clock supply. • SFR used by the serial interface IICA can be read and written.

SAU0EN	Control of serial array unit input clock supply
0	Stops input clock supply. • SFR used by the serial array unit cannot be written. • The serial array unit is in the reset status.
1	Enables input clock supply. • SFR used by the serial array unit can be read and written.

TAU0EN	Control of timer array unit input clock supply
0	Stops input clock supply. • SFR used by timer array unit cannot be written. • Timer array unit is in the reset status.
1	Enables input clock supply. • SFR used by timer array unit can be read and written.

Note 16-pin products only.

Caution Be sure to clear the following bits to 0.
10-pin products: Bits 1, 3, 4, 6, and 7
16-pin products: Bits 1 and 3

5.3.7 Operation speed mode control register (OSMC)

The OSMC register can be used to control supply of the operation clock for the 12-bit interval timer.

When operating the 12-bit interval timer, set WUTMMCK0 = 1 beforehand and do not set WUTMMCK0 = 0 until the timer is stopped.

The OSMC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 5-8. Format of Operation Speed Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Supply of operation clock for 12-bit interval timer
0	Stops Clock supply
1	Low-speed on-chip oscillator clock (f_{IL}) supply

5.3.8 High-speed on-chip oscillator frequency selection register (HOCODIV)

This register is used to change the frequency of the high-speed on-chip oscillator clock set with the option byte (000C2H).

HOCODIV can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to the value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H).

Figure 5-9. Format of High-Speed On-Chip Oscillator Frequency Selection Register (HOCODIV)

Address: F00A8H After reset: value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H) R/W

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV 2	HOCODIV 1	HOCODIV 0

HOCODIV 2	HOCODIV 1	HOCODIV 0	High-speed on-chip oscillator clock frequency selection
0	0	1	20 MHz
0	1	0	10 MHz
0	1	1	5 MHz
1	0	0	2.5 MHz
1	0	1	1.25 MHz
Other than above			Setting prohibited

Cautions 1. Set the HOCODIV register within the operable voltage range before and after the frequency change.

2. Set the HOCODIV register with the high-speed on-chip oscillator clock (f_{IH}) selected as the CPU/peripheral hardware clock (f_{CLK}).

<R> **3. After the frequency is changed with the HOCODIV register, the frequency is switched after the following transition time has elapsed. Note that even if the same value is set in the HOCODIV register, a CPU/peripheral hardware clock wait of up to three clocks will occur.**

- **Operation for up to three clocks at the pre-change frequency**
- **CPU/peripheral hardware clock wait at the post-change frequency for up to three clocks**

5.4 System Clock Oscillator

5.4.1 X1 oscillator (16-pin products only)

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 20 MHz) connected to the X1 and X2 pins. An external clock can be input. In that case, input the clock signal to the EXCLK pin.

To use the X1 oscillator, set bits 7 and 6 (EXCLK, OSCSEL) of the clock operation mode control register (CMC) as follows.

- Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
- External clock input: EXCLK, OSCSEL = 1, 1

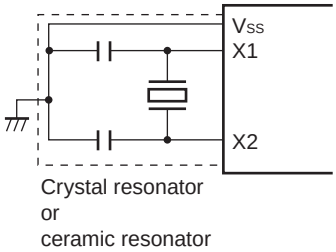
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not used as input port pins, either, see **Table 2-2 Connection of Unused Pins**.

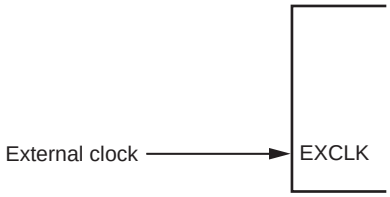
Figure 5-10 shows an example of the external circuit of the X1 oscillator.

Figure 5-10. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock input



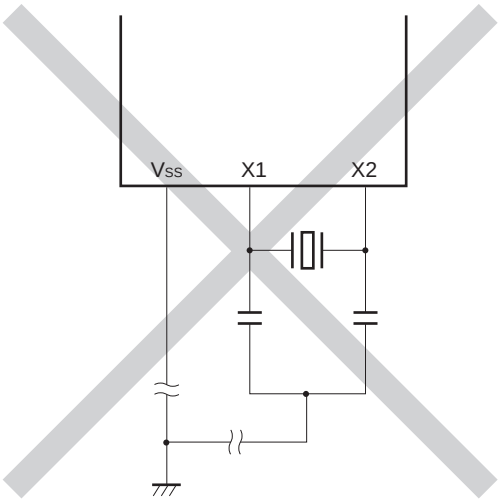
Caution When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the Figure 5-10 to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

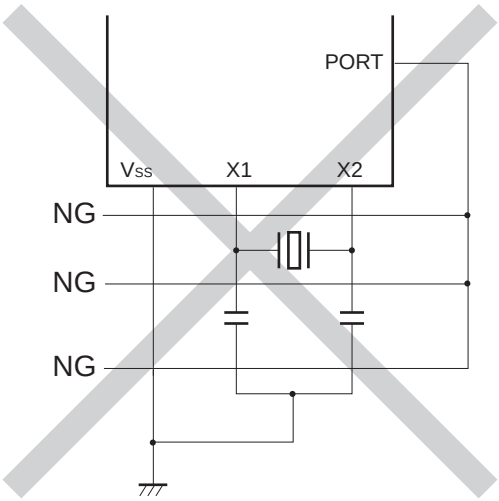
Figure 5-11 shows examples of incorrect resonator connection.

Figure 5-11. Examples of Incorrect Resonator Connection (1/2)

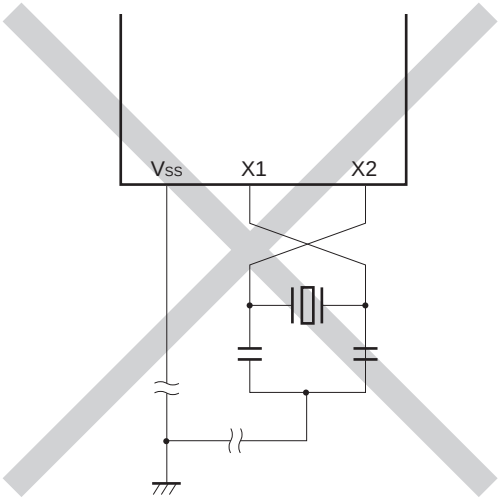
(a) Too long wiring



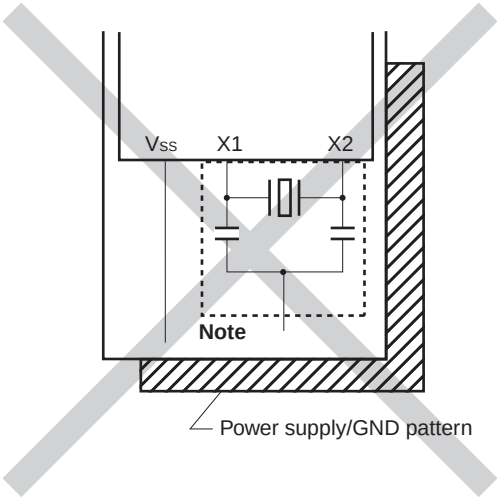
(b) Crossed signal line



(c) The X1 and X2 signal line wires cross.



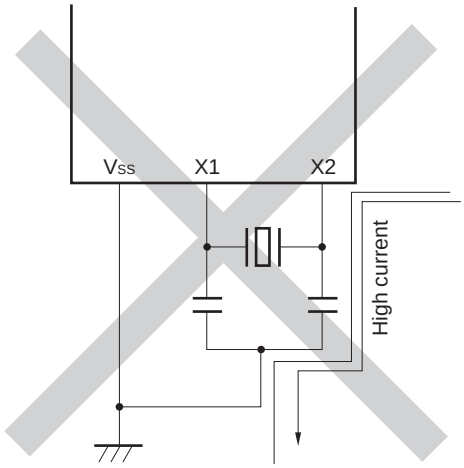
(d) A power supply/GND pattern exists under the X1 and X2 wires.



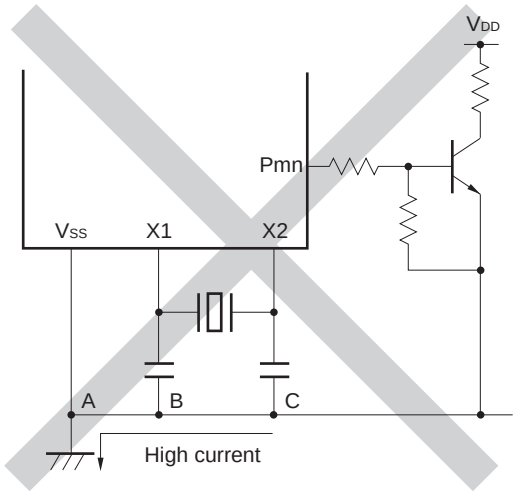
Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

Figure 5-11. Examples of Incorrect Resonator Connection (2/2)

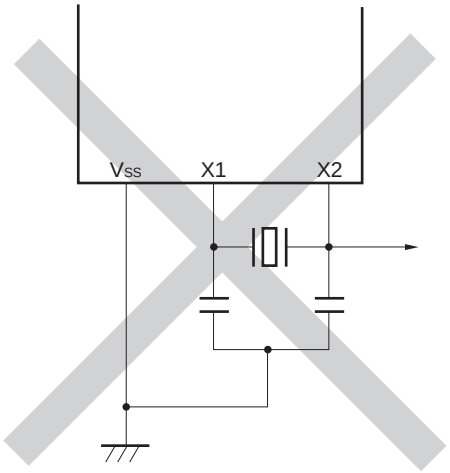
(e) Wiring near high alternating current



(f) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)



(g) Signals are fetched



5.4.2 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/G10. The frequency can be selected from among 20, 10, 5, 2.5, or 1.25 MHz by using the option byte (000C2H). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC) **Note**. The high-speed on-chip oscillator automatically starts oscillating after reset release.

Note 16-pin products only.

5.4.3 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/G10.

The low-speed on-chip oscillator clock is used only as the watchdog timer and 12-bit interval timer **Note** clock. The low-speed on-chip oscillator clock cannot be used as the CPU clock.

The low-speed on-chip oscillator runs while the watchdog timer is operating or when bit 4 (WUTMMCK0) in the operation speed mode control register (OSMC) **Note** is set to 1.

The low-speed on-chip oscillator is stopped when the watchdog timer is stopped and WUTMMCK0 is set to 0.

Note 16-pin products only.

5.5 Clock Generator Operation

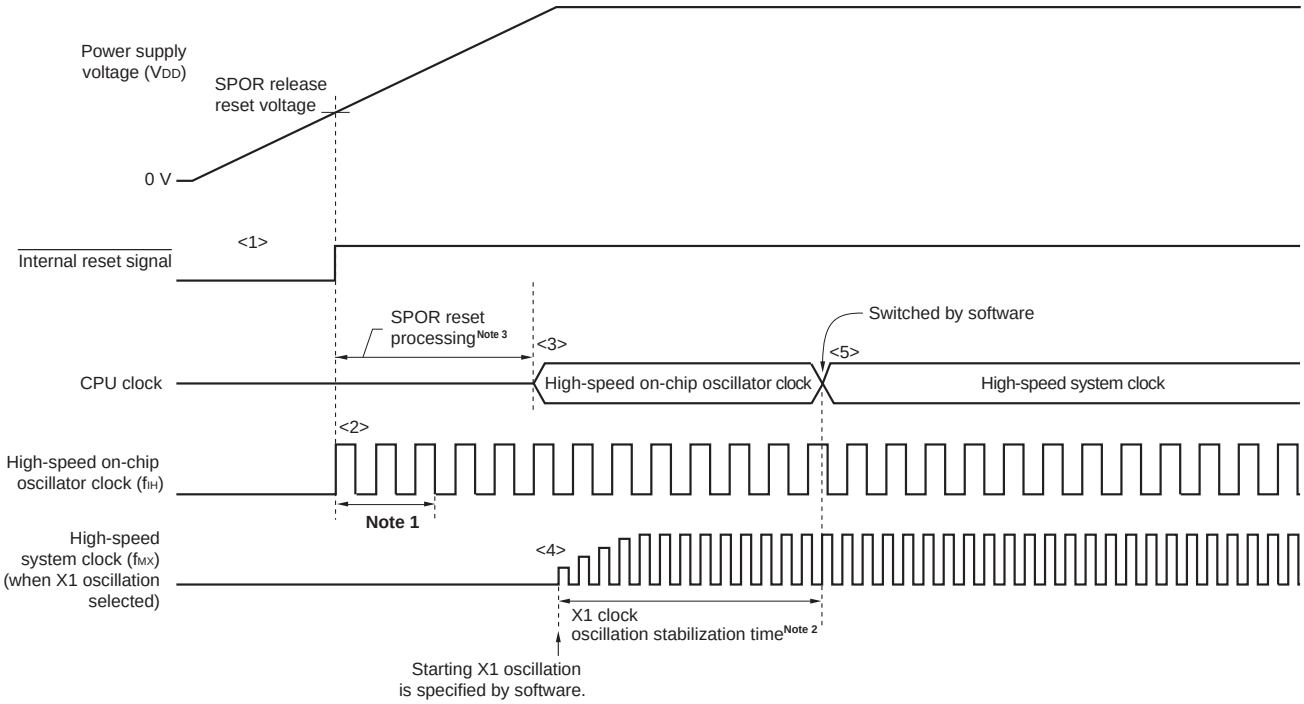
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5-1**).

- Main system clock f_{MAIN}
  - High-speed system clock **Note** f_{MX}
    X1 clock **Note** f_X
    External main system clock **Note** f_{EX}
  - High-speed on-chip oscillator clock f_{IH}
- Low-speed on-chip oscillator clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/G10. When the power supply voltage is turned on, the clock generator operation is shown in Figure 5-12.

Note 16-pin products only.

Figure 5-12. Clock Generator Operation When Power Supply Voltage Is Turned On



<1> When the power is turned on, an internal reset signal is generated by the selectable power-on-reset (SPOR) circuit.

<2> When the power supply voltage exceeds detection voltage of the SPOR circuit, the reset is released and the high-speed on-chip oscillator automatically starts oscillation.

<3> The CPU starts operation on the high-speed on-chip oscillator clock after waiting for the voltage to stabilize and an SPOR reset processing have been performed after reset release.

<4> Set the start of oscillation of the X1 clock via software (see **5.6.2 Example of setting X1 oscillation clock**).

<5> When switching the CPU clock to the X1 clock, wait for the clock oscillation to stabilize, and then switch the clock via software.

Notes 1. The reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

2. When releasing a reset, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC).

3. For SPOR reset processing time, see **CHAPTER 18 SELECTABLE POWER-ON-RESET CIRCUIT**.

Caution When an external clock input from the EXCLK pin is in use, oscillation stabilization time is unnecessary.

5.6 Controlling Clock

5.6.1 Example of setting high-speed on-chip oscillator

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected by using FRQSEL0 to FRQSEL2 of the option byte (000C2H). This frequency can be changed with the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

Option byte (000C2H)	7	6	5	4	3	2	1	0
	1	1	1	1	1	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
0	0	1	20 MHz
0	1	0	10 MHz
0	1	1	5 MHz
1	0	0	2.5 MHz
1	0	1	1.25 MHz
Other than above			Setting prohibited

[High-speed on-chip oscillator frequency selection register (HOCODIV) setting]

Address: F00A8H

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

HOCODIV 2	HOCODIV 1	HOCODIV 0	Selected frequency
0	0	1	20 MHz
0	1	0	10 MHz
0	1	1	5 MHz
1	0	0	2.5 MHz
1	0	1	1.25 MHz
Other than above			Setting prohibited

Cautions 1. Set the HOCODIV register within the operable voltage range before and after the frequency change.

2. Set the HOCODIV register with the high-speed on-chip oscillator clock (f_{IH}) selected as the CPU/peripheral hardware clock (f_{CLK}).

3. After the frequency is changed with the HOCODIV register, the frequency is switched after the following transition time has elapsed.

- **Operation for up to three clocks at the pre-change frequency**
- **CPU/peripheral hardware clock wait at the post-change frequency for up to three clocks**

5.6.2 Example of setting X1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the X1 clock, set the oscillator and start oscillation by using the oscillation stabilization time select register (OSTS) and clock operation mode control register (CMC) and clock operation status control register (CSC) and wait for oscillation to stabilize by using the oscillation stabilization time select register (OSTC). After the oscillation stabilizes, set the X1 clock to f_{CLK} by using the system clock control register (CKC).

[Register settings] Set the register in the order of <1> to <5> below.

<1> Set (1) the OSCSEL bit of the CMC register, except for the cases f_X > 10 MHz, in such cases set (1) the AMPH bit, to operate the X1 oscillator.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 1	0	0	0	0	0	AMPH 0/1

AMPH bit: Set this bit to 0 if the X1 clock is 10 MHz or less.

<2> Using the OSTS register, select the oscillation stabilization time of the X1 oscillator at releasing of the STOP mode.
Example: Setting values when a wait of at least 104 μs is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2 0	OSTS1 1	OSTS0 0

<3> Clear (0) the MSTOP bit of the CSC register to start oscillating the X1 oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP 0	0	0	0	0	0	0	HIOSTOP 0

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize.
Example: Wait until the bits reach the following values when a wait of at least 104 μs is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTC	MOST8 1	MOST9 1	MOST10 1	MOST11 0	MOST13 0	MOST15 0	MOST17 0	MOST18 0

<5> Use the MCM0 bit of the CKC register to specify the X1 clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	0	0	MCS 0	MCM0 1	0	0	0	0

5.6.3 CPU clock status transition diagram

Figure 5-13 shows the CPU clock status transition diagram of this product.

Figure 5-13. CPU Clock Status Transition Diagram

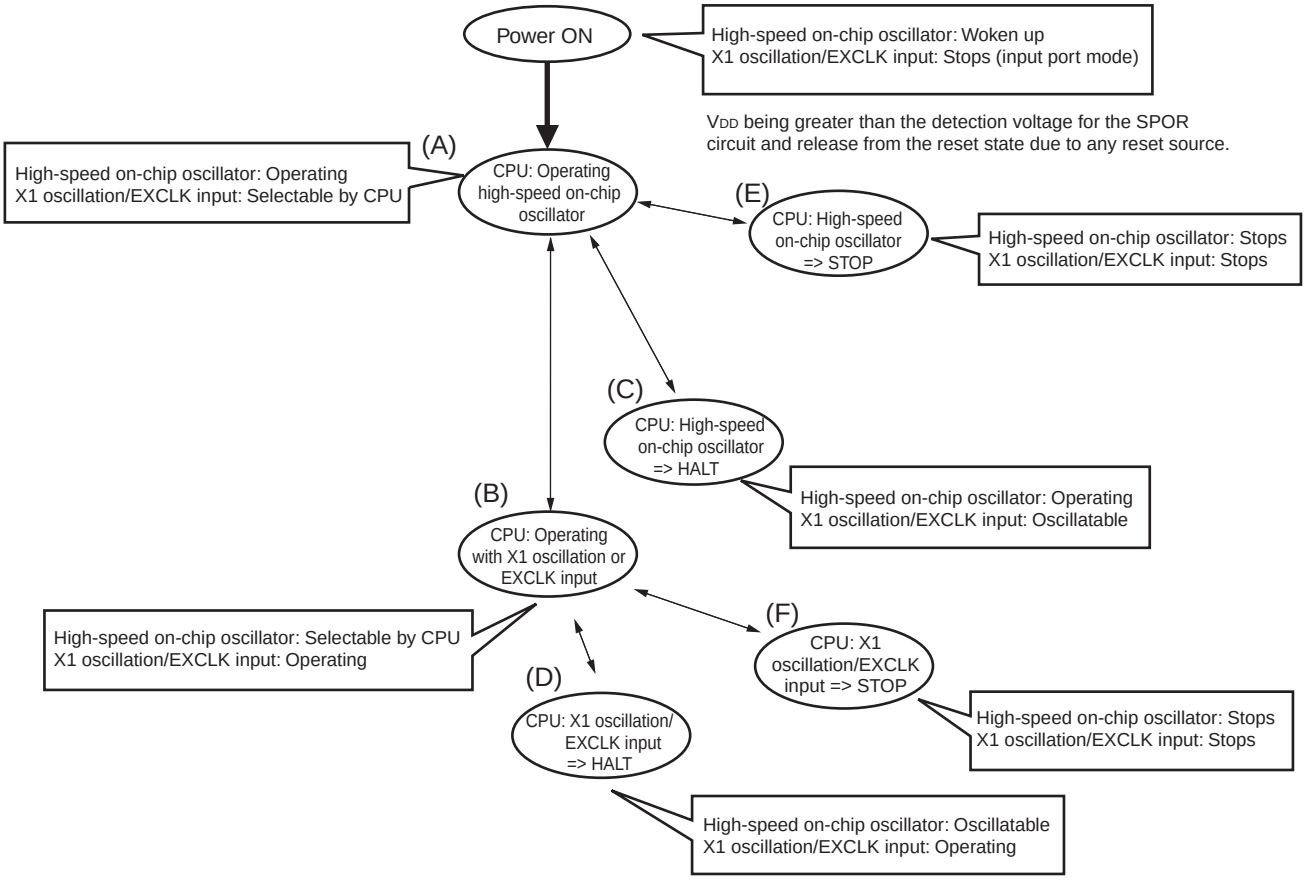


Table 5-3 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples (1/2)

(1) CPU clock changing from high-speed on-chip oscillator clock (A) to high-speed system clock (B)
(The CPU operates with the high-speed on-chip oscillator clock immediately after a reset release (A).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note1			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A) → (B) (X1 clock: 1 MHz ≤ f_X ≤ 10 MHz)	0	1	0	**Note 2**	0	Must be checked	1
(A) → (B) (X1 clock: 10 MHz < f_X ≤ 20 MHz)	0	1	1	**Note 2**	0	Must be checked	1
(A) → (B) (External main system clock)	1	1	x	**Note 2**	0	Checking is unnecessary	1

Notes 1. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release.

2. Set the oscillation stabilization time as follows.

- Desired the oscillation stabilization time counter status register (OSTC) oscillation stabilization time ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 24 ELECTRICAL SPECIFICATIONS).

Remarks 1. ×: don't care

2. (A) to (F) in Table 5-3 correspond to (A) to (F) in Figure 5-13

.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples (2/2)

(2) CPU clock changing from high-speed system clock (B) to high-speed on-chip oscillator clock (A)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		MCM0
(B) → (A)	0	27 μs (typ.)	0

Unnecessary if the CPU is operating with the high-speed on-chip oscillator clock

(3) • **HALT mode (C) set while CPU is operating with high-speed on-chip oscillator clock (A)**
• **HALT mode (D) set while CPU is operating with high-speed system clock (B)**

Status Transition	Setting
(A) → (C) (B) → (D)	Executing HALT instruction

(4) • **STOP mode (E) set while CPU is operating with high-speed on-chip oscillator clock (A)**
• **STOP mode (F) set while CPU is operating with high-speed system clock (B)**

(Setting sequence) →

Status Transition		Setting		
(A) → (E)		Stopping peripheral functions that cannot operate in STOP mode	–	Executing STOP instruction
(B) → (F)	In X1 oscillation		Sets the OSTS register	
	External clock		–	

Remark (A) to (F) in Table 5-3 correspond to (A) to (F) in Figure 5-13.

5.6.4 Condition before changing CPU clock and processing after changing CPU clock

Condition before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 5-4. Changing CPU Clock

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
High-speed on-chip oscillator clock	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	The operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock is changed.
	External main system clock	An external clock input from the EXCLK pin must be enabled. • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation accuracy stabilization time	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	External main system clock	Change is not possible	–
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation accuracy stabilization time	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	X1 clock	Change is not possible	–

5.6.5 Time required for switchover of CPU clock and main system clock

The main system clock can be switched between the high-speed on-chip oscillator clock and the high-speed system clock by specifying bit 4 (MCM0) of the system clock control register (CKC).

The actual switchover operation is not performed immediately after rewriting to the CKC register; operation continues on the pre-switchover clock for several clocks (see **Table 5-5**).

Whether the main system clock is operating on the high-speed system clock or high-speed on-chip oscillator clock can be ascertained using bit 5 (MCS) of the CKC register.

When the CPU clock is switched, the peripheral hardware is also switched.

Table 5-5. Maximum Number of Clocks Required for $f_{IH} \leftrightarrow f_{MX}$

Set Value Before Switchover		Set Value After Switchover	
MCM0		MCM0	
		0 ($f_{MAIN} = f_{IH}$)	1 ($f_{MAIN} = f_{MX}$)
0 ($f_{MAIN} = f_{IH}$)	$f_{MX} \geq f_{IH}$		1 + f_{IH}/f_{MX}
	$f_{MX} < f_{IH}$		$2f_{IH}/f_{MX}$ clock
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{IH}$	$2f_{MX}/f_{IH}$ clock	
	$f_{MX} < f_{IH}$	1 + f_{MX}/f_{IH}	

Remarks 1. Number of CPU clocks before switchover.

2. Calculate the number of clocks by rounding to the nearest whole number.

Example When switching the main system clock from the high-speed system clock to the high-speed on-chip oscillator clock (@ oscillation with f_{IH} = 5 MHz selected, f_{MX} = 10 MHz)

$2\ f_{IH}/f_{MX} = 2(10/5) = 4 \rightarrow 4$ clocks

5.6.6 Conditions before clock oscillation is stopped

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped. Before stopping the clock oscillation, check the conditions before the clock oscillation is stopped.

Table 5-6. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped	Flag Settings of SFR Register
High-speed on-chip oscillator clock	MCS = 1 (The CPU is operating on the high-speed system clock.)	HIOSTOP = 1
X1 clock	MCS = 0 (The CPU is operating on the high-speed on-chip oscillator clock.)	MSTOP = 1
External main system clock		

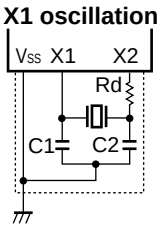
5.7 Resonator and Oscillator Constants

The resonators for which the operation is verified and their oscillator constants are shown below.

Cautions 1. The constants for these oscillator circuits are reference values based on specific environments set up for evaluation by the manufacturers. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board. Furthermore, if you are switching from a different product to this microcontroller, and whenever you change the board, again request evaluation by the manufacturer of the oscillator circuit mounted on the new board.

2. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the RL78 microcontroller so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Figure 5-14. External Oscillation Circuit Example



(1) X1 oscillation:

As of December, 2013

Manufacturer	Resonator	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants Note 1 (reference)			Oscillation Voltage Range (V)	
					C1 (pF)	C2 (pF)	Rd (kΩ)	MIN.	MAX.
Murata Manufacturing Co., Ltd. Note 2	Crystal resonator	CSTCC2M00G56-R0	SMD	2.0	(47)	(47)	0	1.6	5.5
		CSTCR4M00G55-R0	SMD	4.0	(39)	(39)	0		
		CSTLS4M00G53-B0	Lead		(15)	(15)	0		
		CSTCE8M00G52-R0	SMD	8.0	(10)	(10)	0		
		CSTLS8M00G53-B0	Lead		(15)	(15)	0		
		CSTCE16M0V53-R0	SMD	16	(15)	(15)	0		
		CSTLS16M0X51-B0	Lead		(5)	(5)	0		
		CSTCE20M0V51-R0	SMD	20	(5)	(5)	0		
		CSTCE8M00G52-R0	SMD	8.0	(10)	(10)	0	1.8	5.5
		CSTCE16M0V53-R0	SMD	16.0	(15)	(15)	0		
		CSTLS16M0X51-B0	Lead		(5)	(5)	0		
		CSTCE20M0V51-R0	SMD	20.0	(5)	(5)	0		
Kyocera Crystal Device Co., Ltd. Note 3	Crystal resonator	CX8045GB	SMD	4.0	10	10	0	2.5	5.5
		CX5032SA	SMD	8.0	9	9	0		
		CX3225SB	SMD	12.0	9	9	0		
		CX2016DB	SMD	20.0	4	4	0		

Notes 1. Values in parentheses in the C1 and C2 columns indicate an internal capacitance.

2. When using this resonator, for details about the matching, contact Murata Manufacturing Co., Ltd. (http://www.murata.com).

3. When using this resonator, for details about the matching, contact Kyocera Crystal Device Co., Ltd. (http://www.kyocera-crystal.jp/eng/index.html, http://global.kyocera.com).

CHAPTER 6 TIMER ARRAY UNIT

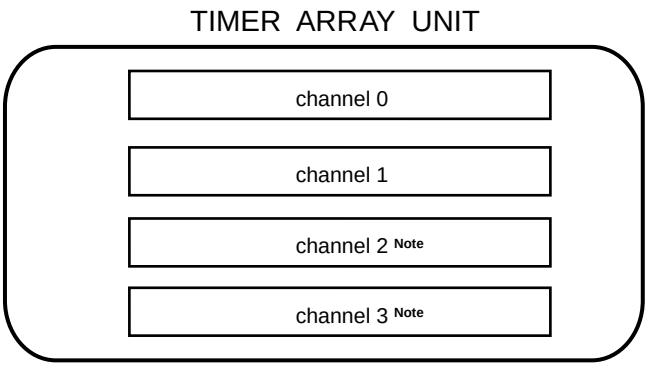
The number of units or channels of the timer array unit differs, depending on the product.

Channel	10-pin	16-pin
Channel 0	√	√
Channel 1	√	√
Channel 2	–	√
Channel 3	–	√

Remark √: Provided
–: Not provided

The timer array unit has a maximum of four 16-bit timers.

Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used in combination to create a high-accuracy timer.



Note 16-pin products only.

For details about each function, see the table below.

Independent channel operation function	Simultaneous channel operation function
• Interval timer (→ refer to **6.8.1**) • Square wave output (→ refer to **6.8.1**) • External event counter (→ refer to **6.8.2**) • Divider function Note 2 (→ refer to **6.8.3**) • Input pulse interval measurement (→ refer to **6.8.4**) • Measurement of high-/low-level width of input signal (→ refer to **6.8.5**) • Delay counter (→ refer to **6.8.6**)	• One-shot pulse output (→ refer to **6.9.1**) • Two-channel input with one-shot pulse output function Note 1 (→ refer to **6.9.2**) • PWM output function (→ refer to **6.9.3**) • Multiple PWM output function Note 1 (→ refer to **6.9.4**)

It is possible to use the 16-bit timer of channels 1 and 3 as two 8-bit timers (higher and lower). The functions that can use channels 1 and 3 as 8-bit timers are as follows:

- Interval timer (higher and lower 8-bit timers)
- Square wave output (lower 8-bit timer only)
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)
- PWM output function (lower 8-bit timer only)
- Multiple PWM output function Note 1 (lower 8-bit timer only)

Interlinked operation of channel 1 with the serial array unit operating as UART0 can be obtained by setting the ISC register. The input pulse interval measurement mode can then be used to measure the width at the baud rate of the other party in communications and make the required adjustments in response.

Notes 1. 16-pin products only.
2. Only channels 0 and 3

6.1 Functions of Timer Array Unit

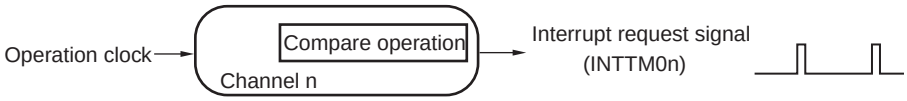
Timer array unit has the following functions.

6.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following purposes without being affected by the operation mode of other channels.

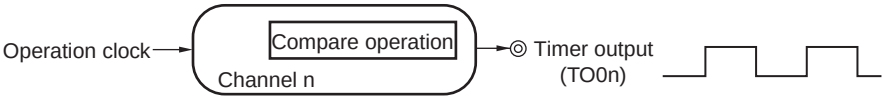
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTM0n) at fixed intervals.



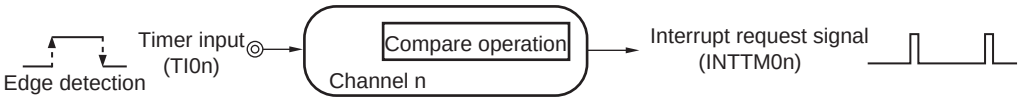
(2) Square wave output

A toggle operation is performed each time INTTM0n interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TO0n).



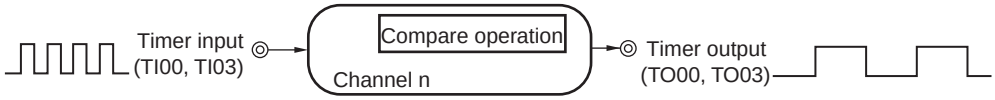
(3) External event counter

Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TI0n) has reached a specific value.



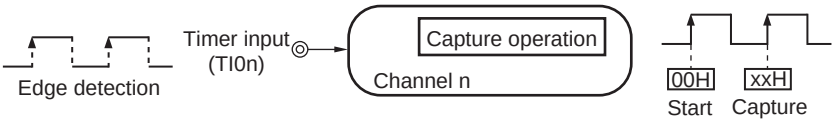
(4) Divider function (channels 0 and 3 only)

A clock input from a timer input pin (TI00, TI03) is divided and output from an output pin (TO00, TO03).



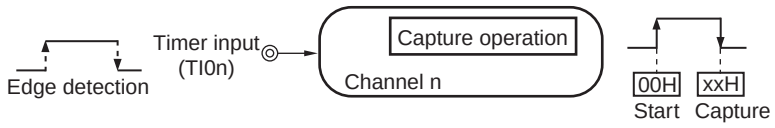
(5) Input pulse interval measurement

Counting is started by the valid edge of a pulse signal input to a timer input pin (TI0n). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



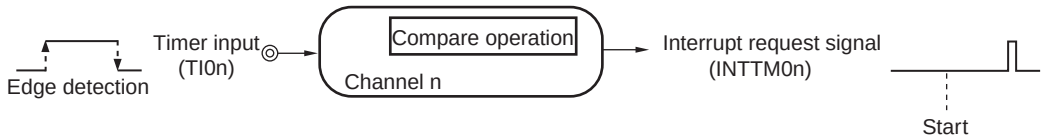
(6) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TI0n), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.



(7) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TI0n), and an interrupt is generated after any delay period.



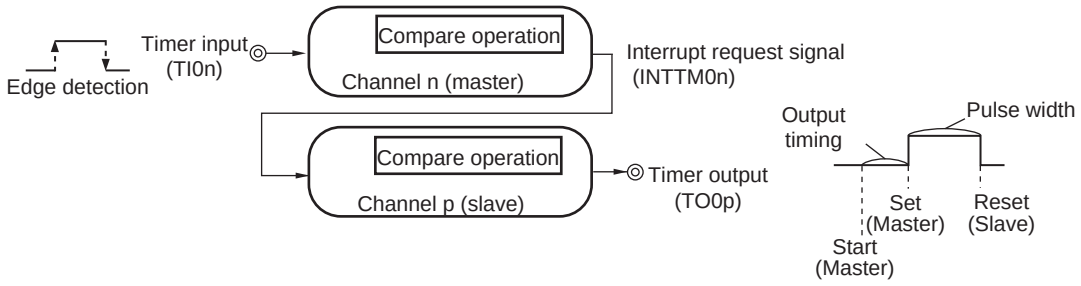
Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products).

6.1.2 Simultaneous channel operation function

By using the combination of a master channel (a reference timer mainly controlling the cycle) and a slave channel (a timer operating according to the master channel), channels can be used for the following purposes.

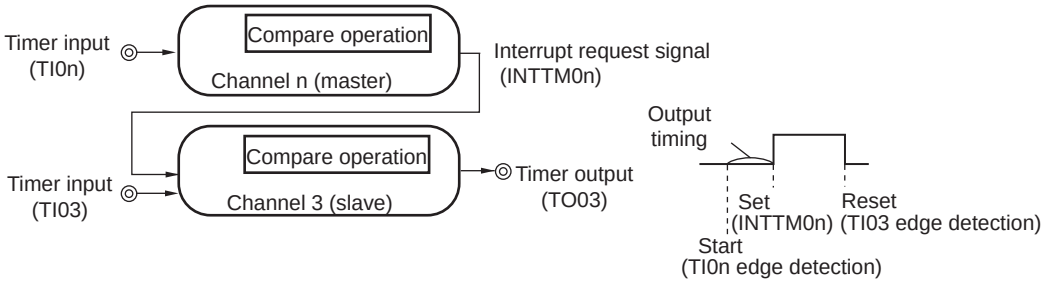
(1) One-shot pulse output

Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.



(2) Two-channel input with one-shot pulse output function (16-pin products only)

Two channels are used as a set to generate any one-shot pulse by setting or resetting the timer output pin (TO03) at a valid edge of the timer input pin (TI0n, TI03) input.

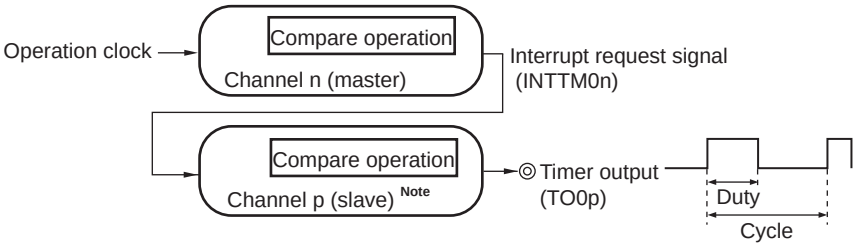


Caution There are several rules for using the simultaneous channel operation function.
For details, see 6.4.1 Basic rules of simultaneous channel operation function.

Remark n: Channel number
n = 0 (for 10-pin products); n = 0, 2 (for 16-pin products)
p: Slave channel number ($0 < p \leq 3$)

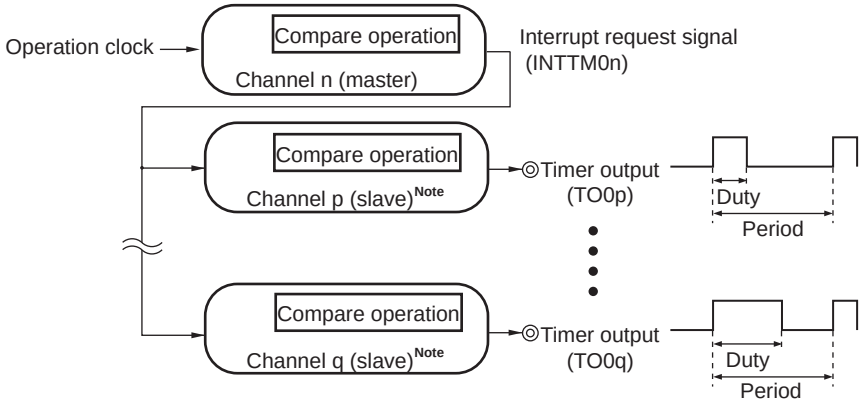
(3) PWM (Pulse Width Modulation) output function

Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



(4) Multiple PWM (Pulse Width Modulation) output function (16-pin products only)

By extending the PWM function and using one master channel and two or more slave channels, up to three types of PWM signals that have a specific period and a specified duty factor can be generated.



Note This operation can be obtained with the lower 8-bit timer of channel 1 or 3.

Caution **There are several rules for using the simultaneous channel operation function.**
For details, see 6.4.1 Basic rules of simultaneous channel operation function.

Remark n: Channel number
n = 0 (for 10-pin products); n = 0, 2 (for 16-pin products)
p, q: Slave channel number ($0 < p < q \leq 3$)

6.1.3 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer in a configuration consisting of two 8-bit timers (higher and lower).

The 16-bit timer channels 1 and 3 support the following functions as 8-bit timer operation.

- Interval timer (higher and lower 8-bit timers)
- Square wave output (lower 8-bit timer only)
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)
- PWM output function (lower 8-bit timer only)
- Multiple PWM output function [Note] (lower 8-bit timer only)

Note 16-pin products only

Caution There are several rules for using 8-bit timer operation function.
For details, see 6.4.2 Basic rules of 8-bit timer operation function (only channels 1 and 3).

6.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 6-1. Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer counter register 0n (TCR0nH, TCR0nL)
Register	Timer data register 0n (TDR0nH, TDR0nL)
Timer input	TI00 to TI03
Timer output	TO00 to TO03, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register 0 (TPS0) • Timer channel enable status register 0 (TE0, TEH0) • Timer channel start register 0 (TS0, TSH0) • Timer channel stop register 0 (TT0, TTH0) • Timer output enable register 0 (TOE0) • Timer output register 0 (TO0) • Timer output level register 0 (TOL0) • Timer output mode register 0 (TOM0)
	<Registers of each channel> • Timer mode register 0n (TMR0nH, TMR0nL) • Timer status register 0n (TSR0n) • Noise filter enable register 1 (NFEN1) • Input switch control register (ISC) • Port mode control register 0 (PMC0) • Port mode register 0, 4 (PM0, PM4) • Port register 0, 4 (P0, P4)

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Alternate port for timer I/O of the timer array unit channels varies depending on products.

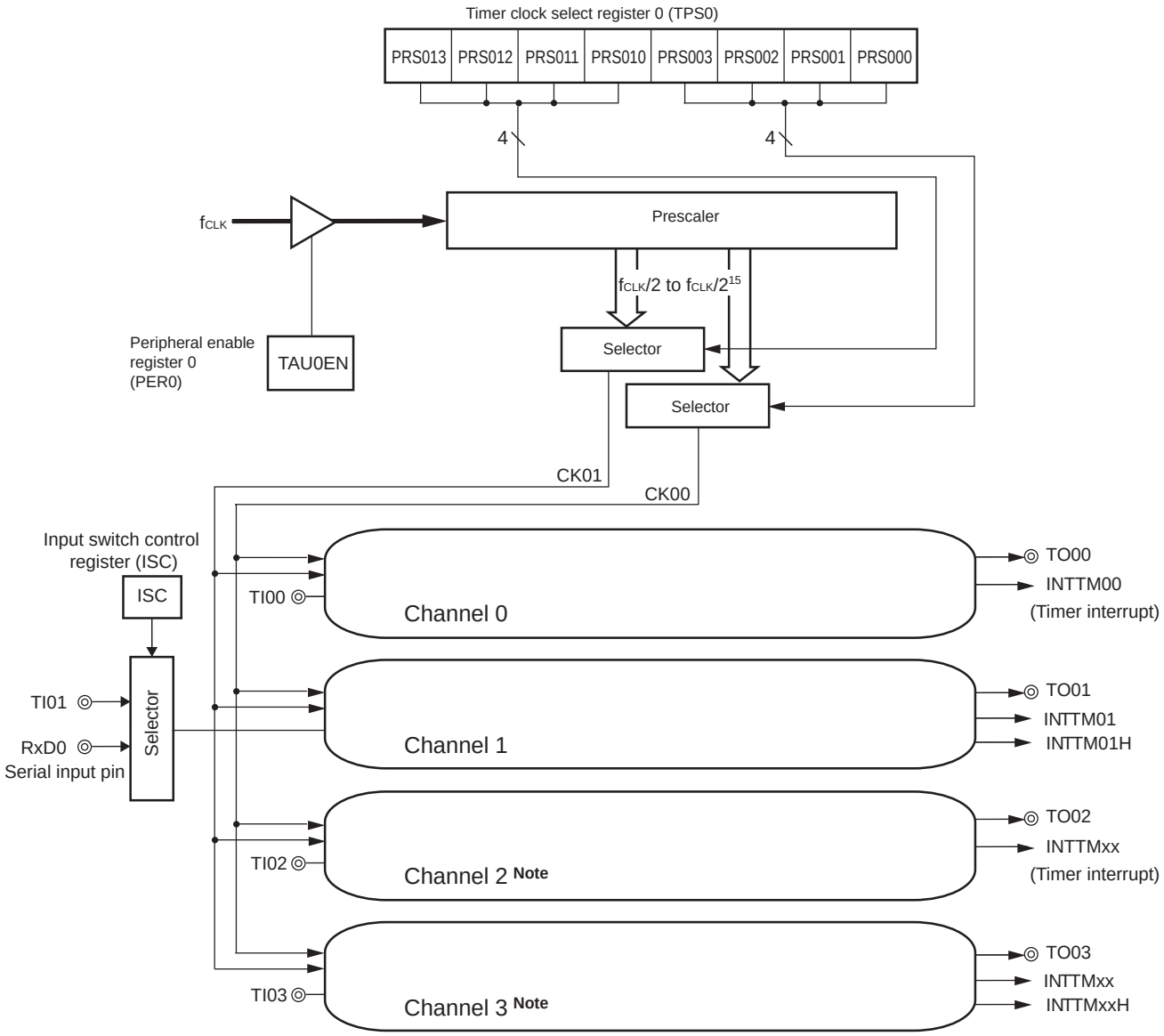
Table 6-2. Timer I/O Pins in the Products

Timer array unit channel	10-pin products	16-pin products
Channel 0	P03/TO00, P137/TI00	P03/TO00, P137/TI00
Channel 1	P04/TI01/TO01, (P40/TI01/TO01)	P04/TI01/TO01, (P40/TI01/TO01)
Channel 2	—	P05/TI02/TO02
Channel 3	—	P41/TI03, P07/TO03

Remarks 1. If a pin is to be used for both timer input and timer output, it can be used only for timer input or timer output.
2. —: Not supported
3. The pin names in parentheses indicate function-multiplexed ports while PIOR0 bit in the peripheral I/O redirection register is set to 1.

Figures 6-1 and 6-2 show the block diagrams of the timer array unit.

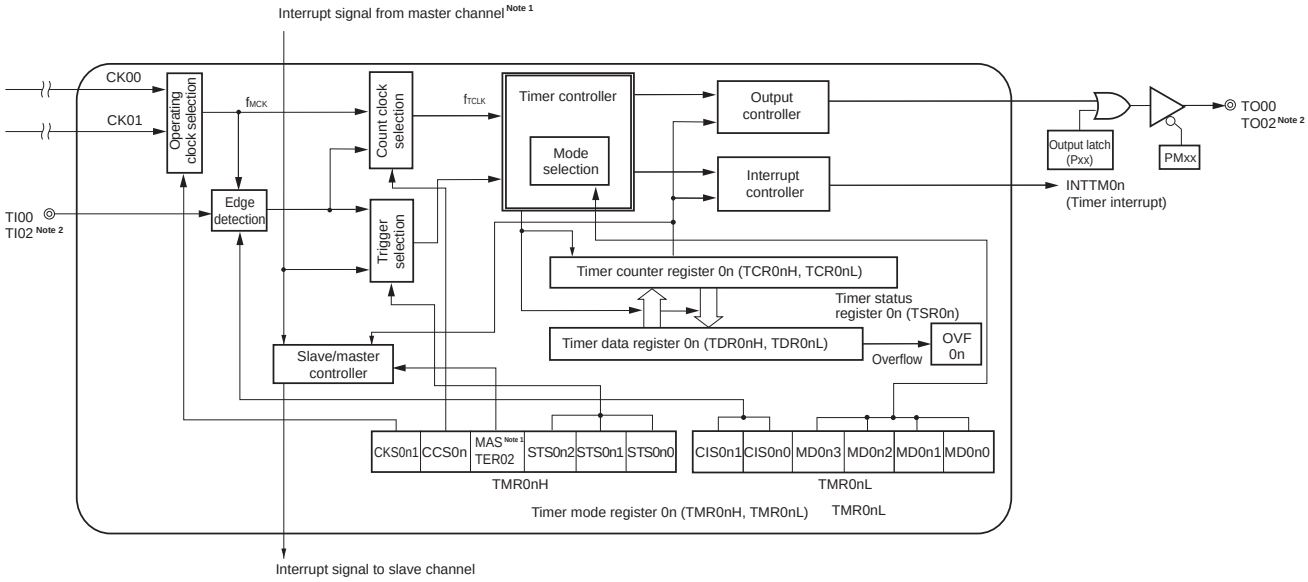
Figure 6-1. Entire Configuration of Timer Array Unit



Note 16-pin products only.

Figure 6-2. Internal Block Diagram of Channel of Timer Array Unit

(a) Channels 0 and 2

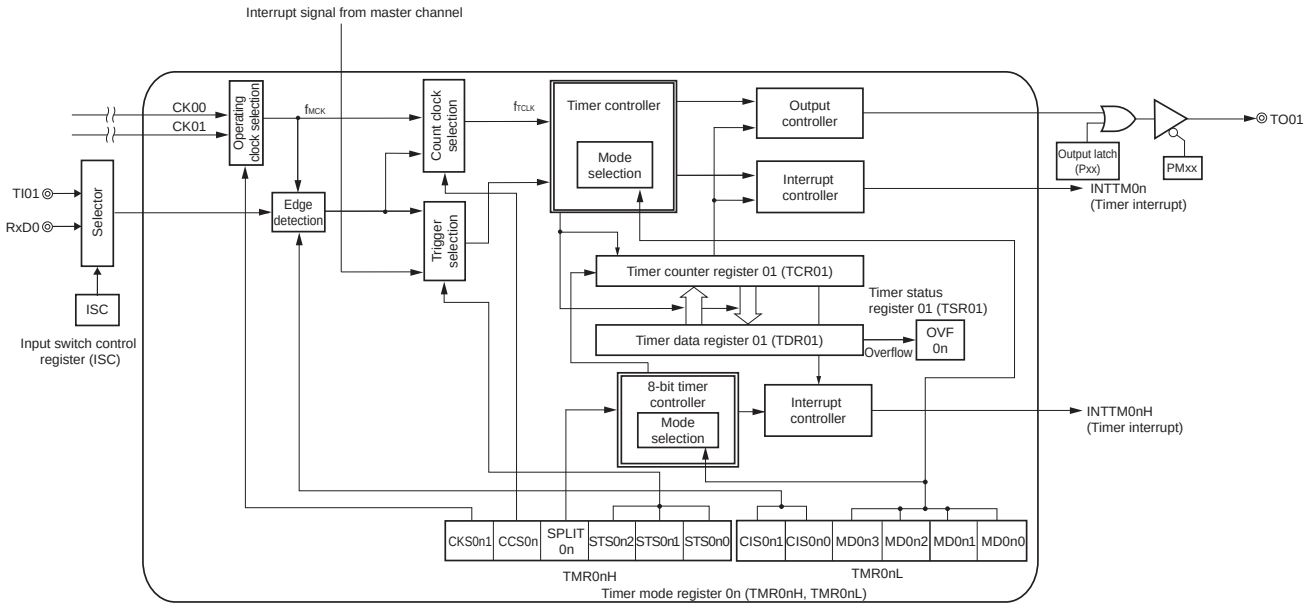


Notes **1.** Channel 2 only.

2. 16-pin products only.

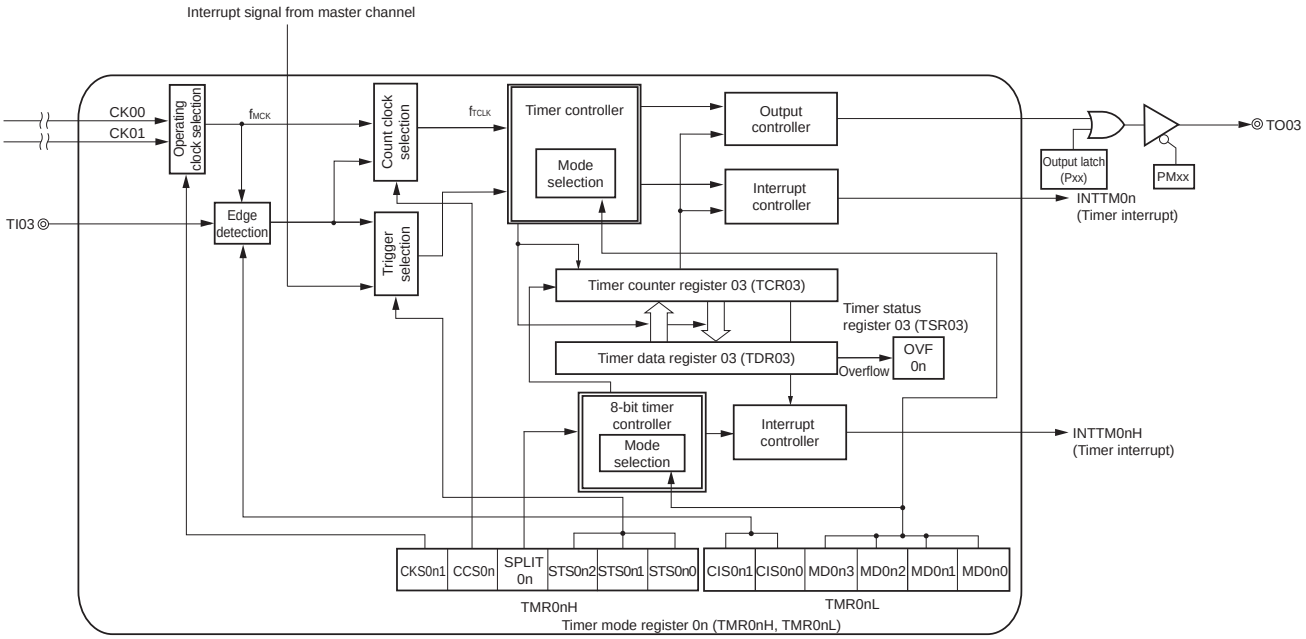
Remark n = 0, 2

(b) Channel 1



Remark n = 1

(c) Channel 3 Note



Note 16-pin products only.

Remark n = 3

6.2.1 Timer counter register 0n (TCR0n)

TCR0n register consists of two 8-bit read-only registers (TCR0nH and TCR0nL) and is used to count clocks (f_{TCLK}).

When data is read from the TCR0n register, the TCR0nH and TCR0nL registers must be accessed consecutively.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock (f_{TCLK}).

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MD0n3 to MD0n0 bits of timer mode register 0n (TMR0n) (refer to **6.3.3 Timer mode register 0n (TMR0n)**).

Figure 6-3. Format of Timer Counter Register 0n (TCR0n) (n = 0 to 3)

Address: F0180H (TCR00L), F0181H (TCR00H) After reset: FFH R
: F0182H (TCR01L), F0183H (TCR01H)
: F0184H (TCR02L), F0185H (TCR02H)
: F0186H (TCR03L), F0187H (TCR03H)

	TCR0nH								TCR0nL							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCR0n																

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Reading from the TCR0nH and TCR0nL registers must be performed successively, in order of the TCR0nL register and the TCR0nH register. If data are read from TCR0nL between the successive read, reading is not performed correctly.

Caution Consecutive reading from the TCR0nH and TCR0nL registers must be performed in the state where an interrupt is disabled by the DI instruction.

The count value can be read by reading timer counter register 0n (TCR0n).

The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAU0EN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode [Note]

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Note 16-pin products only.

Cautions
1. **The count value is not captured to timer data register 0n (TDR0n) even when the TCR0n register is read.**
2. **When channels 1 and 3 are used in 8-bit timer mode (SPLIT = 1), it is prohibited to read the TCR01H and TDR01H registers or the TCR03H and TDR03H registers.**

The TCR0n register read value differs as follows according to operation mode changes and the operating status.

Table 6-3. Timer Counter Register 0n (TCR0n) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer counter register 0n (TCR0n) Read Value[Note]			
		Value if the operation mode was changed after releasing reset	Value if the count operation paused (TT0n = 1)	Value if the operation mode was changed after count operation paused (TT0n = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	–
Capture mode	Count up	0000H	Value if stop	Undefined	–
Event counter mode	Count down	FFFFH	Value if stop	Undefined	–
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture value of TDR0n register + 1

Note Following timer operation of channel n being stopped (TE0n = 0), this is the value read from the TCR0n register at the time counter operation is enabled (TS0n = 1). The TCR0n register retains this value until counting starts.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Caution **When channels 1 and 3 are used in 8-bit timer mode (SPLIT = 1), it is prohibited to read the TCR01H and TDR01H registers or the TCR03H and TDR03H registers.**

6.2.2 Timer data register 0n (TDR0n)

The TDR0n register consists of two eight bit registers (TDR0nH, TDR0nL) for which the capture or comparison functions can be selected.

Switching between the capture and comparison functions is by using the MD0n3 to MD0n0 bits of the timer mode register 0n (TMR0n) to select the operating mode.

When using the TDR0n register as a compare register, the value of the TDR0nL and TDR0nH registers can be changed at any time.

For access to a TDR0n register, the TDR0nH and TDR0nL registers must be accessed consecutively.

In eight-bit timer mode (i.e. when the SPLIT0n bit of timer mode register 0n (TMR0n) is set to “1”), the TDR0n register can be rewritten in eight-bit units, with the higher 8 bits used as TDR0nH and the lower 8 bits used as TDR0nL.

The following points for caution apply when data are read from or written to TDR0nH and TDR0nL registers.

- In 16-bit timer mode (when channels 0 and 2 are in use, or bit 3 (SPLIT0n) of the TMR0nH register of channels 1 and 3 is cleared to “0”)
  Writing to TDR0nH and TDR0nL registers must be performed by writing in a row with data in order of that for the TDR0nH register and that for the TDR0nL register. The values of TDR0nH and TDR0nL are updated when TDR0nL is rewritten.
  Reading from the TDR0nH and TDR0nL registers must be performed in a row with data in order of that from the TDR0nL register and that from the TDR0nH register. The value of TDR0nH is updated when TDR0nL is read.
  If data are written to TDR0nH, read from TDR0nL, or read from TCR0n between the successive read or successive write operations, reading and writing is not performed correctly.
  Consecutive reading from the TDR0nH and TDR0nL registers and consecutive writing to the TDR0nH and TDR0nL registers must be performed in the state where an interrupt is disabled by the DI instruction.
- In 8-bit timer mode (when bit 3 (SPLIT0n) of the TMR0nH register of channel 1 or 3 is set to “1”)
  The data can be written to the TDR0nH and TDR0nL registers in 8-bit units in 8-bit timer mode.
  Reading from TDR0nH register must be performed in a row with data in order of that from the TDR0nL register and that from the TDR0nH register. The value of TDR0nH is updated when TDR0nL is read.
  If data are written to TDR0nH, read from TDR0nL, or read from TCR0n between the successive read operations, reading is not performed correctly.
  Consecutive reading from the TDR0nH and TDR0nL registers must be performed in the state where an interrupt is disabled by the DI instruction.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Caution **When channels 1 and 3 are used in 8-bit timer mode (SPLIT = 1), it is prohibited to read the TCR01H and TDR01H registers or the TCR03H and TDR03H registers.**

Figure 6-4. Format of Timer Data Register 0n (TDR0nH, TDR0nL) (n = 0, 2)

Address: FFF18H (TDR00L), FFF19H (TDR00H), After reset: 00H R/W
FFF64H (TDR02L), FFF65H (TDR02H)

	FFF19H (TDR00H)								FFF18H (TDR00L)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDR0n																

Figure 6-5. Format of Timer Data Register 0n (TDR0n) (n = 1, 3)

Address: FFF1AH (TDR01L), FFF1BH (TDR01H), After reset: 00H R/W
FFF66H (TDR03L), FFF67H (TDR03H)

	FFF1BH (TDR01H)								FFF1AH (TDR01L)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDR0n																

(i) When timer data register 0n (TDR0nH, TDR0nL) is used as compare register

Counting down is started from the value set to the TDR0nH and TDR0nL registers. When the count value reaches 0000H, an interrupt request signal (INTTM0n) is generated. The TDR0n register holds its value until it is rewritten.

Caution The TDR0n register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register 0n (TDR0nH, TDR0nL) is used as capture register

The count value of timer counter register 0n (TCR0n) is captured to the TDR0nH and TDR0nL registers when the capture trigger is input.

A valid edge of the TI0n pin can be selected as the capture trigger. This selection is made by timer mode register 0n (TMR0n).

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register 0 (TPS0)
- Timer channel enable status register 0 (TE0, TEH0)
- Timer channel start register 0 (TS0, TSH0)
- Timer channel stop register 0 (TT0, TTH0)
- Timer output enable register 0 (TOE0)
- Timer output register 0 (TO0)
- Timer output level register 0 (TOL0)
- Timer output mode register 0 (TOM0)
- Timer mode register 0n (TMR0nH, TMR0nL)
- Timer status register 0n (TSR0n)
- Noise filter enable register 1 (NFEN1)
- Input switch control register (ISC)
- Port mode control register 0 (PMC0)
- Port mode register 0, 4 (PM0, PM4)
- Port register 0, 4 (P0, P4)

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.3.1 Peripheral enable register 0 (PER0)

This registers is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the timer array unit is used, be sure to set bit 0 (TAU0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-6. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER0	TMKAEN[Note]	CMPEN[Note]	ADCEN	IICA0EN[Note]	0	SAU0EN	0	TAU0EN

TAU0EN	Control of timer array unit input clock
0	Stops supply of input clock. • SFR used by the timer array unit cannot be written. • The timer array unit is in the reset status.
1	Supplies input clock. • SFR used by the timer array unit can be read/written.

Note 16-pin products only.

Cautions 1. When setting the timer array unit, be sure to set the following registers first while the TAU0EN bit is set to 1. If TAU0EN = 0, the values of the registers which control the timer array unit are cleared to their initial values and writing to them is ignored (except for the noise filter enable register 1 (NFEN1), input switch control register (ISC), port mode registers 0, 4 (PM0, PM4), port registers 0, 4 (P0, P4), and port mode control register 0 (PMC0)).

- **Timer counter register 0n (TCR0nH, TCR0nL)**
- **Timer data register 0n (TDR0nH, TDR0nL)**
- **Timer clock select register 0 (TPS0)**
- **Timer channel enable status register 0 (TE0, TEH0)**
- **Timer channel start register 0 (TS0, TSH0)**
- **Timer channel stop register 0 (TT0, TTH0)**
- **Timer output enable register 0 (TOE0)**
- **Timer output register 0 (TO0)**
- **Timer output level register 0 (TOL0)**
- **Timer output mode register 0 (TOM0)**
- **Timer mode register 0n (TMR0nH, TMR0nL)**
- **Timer status register 0n (TSR0n)**

2. Be sure to clear the following bits to 0.
10-pin products: bits 1, 3, 4, 6, 7
16-pin products: bits 1, 3

6.3.2 Timer clock select register 0 (TPS0)

The TPS0 register is a 16-bit register that is used to select four types of operation clocks (CK00, CK01) that are commonly supplied to each channel from the prescaler.

Rewriting of the TPS0 register during timer operation is possible only in the following cases.

If the PRS000 to PRS003 bits can be rewritten (n = 0 to 3):
All channels for which CK00 is selected as the operation clock (CKS0n1 = 0) are stopped (TE0n = 0).

If the PRS010 to PRS013 bits can be rewritten (n = 0 to 3):
All channels for which CK01 is selected as the operation clock (CKS0n1 = 1) are stopped (TE0n = 0).

The TPS0 register can be set by a 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-7. Format of Timer Clock Select Register 0 (TPS0)

Address: F01B6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TPS0	PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000

PRS 0k3	PRS 0k2	PRS 0k1	PRS 0k0	Selection of operation clock (CK0k) Note (k = 0, 1)					
					f_{CLK} = 1.25 MHz	f_{CLK} = 2.5 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	1.25 MHz	2.5 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	625 kHz	1.25 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	313 kHz	625 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	156 kHz	313 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	78.1 kHz	156 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	39.1 kHz	78.1 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	19.5 kHz	39.1 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	$f_{CLK}/2^7$	9.77 kHz	19.5 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	$f_{CLK}/2^8$	4.88 kHz	9.77 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	2.44 kHz	4.88 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.22 kHz	2.44 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	0	0	$f_{CLK}/2^{12}$	305 Hz	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	$f_{CLK}/2^{13}$	153 Hz	305 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	$f_{CLK}/2^{14}$	76.3 Hz	153 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	$f_{CLK}/2^{15}$	38.1 Hz	76.3 Hz	153 Hz	305 Hz	610 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TT0 = 0FH, TTH0 = 0AH).

Caution **If f_{CLK} (undivided) is selected as the operation clock (CK0k) and TDR0nH and TDR0nL are cleared to 00H (n = 0 to 3), the interrupt request signal (INTTM0n) output from timer array units cannot be used.**

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. The above selected clock, but a signal which becomes high level for one f_{CLK} cycle period from its rising edge. For details, see **6.5.1 Count clock (f_{TCLK}).**

6.3.3 Timer mode register 0n (TMR0n)

The TMR0n register consists of two eight-bit registers (TMR0nH, TMR0nL) which set an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock (f_{TCLK}), select the master/slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMR0nH and TMR0nL registers is prohibited when the register is in operation (when TE0n = 1).

The TMR0nH and TMR0nL registers can be set by a 8-bit memory manipulation instruction.

Reset signal generation clears TMR0nH and TMR0nL registers to 00H.

Caution The bits mounted depend on the channels in the bit 3 of TMR0nH register.

TMR02H:	**MASTER02 bit**
TMR01H, TMR03H:	**SPLIT0n bit (n = 1, 3)**
TMR01H:	**Fixed to 0**

Figure 6-8. Format of Timer Mode Register 0n (TMR0n) (1/3)

Address: : F0190H (TMR00L), F0191H (TMR00H) After reset: 00H R/W
: F0192H (TMR01L), F0193H (TMR01H)
: F0194H (TMR02L), F0195H (TMR02H)
: F0196H (TMR03L), F0197H (TMR03H)

Symbol	7	6	5	4	3	2	1	0
TMR00H	CKS001	0	0	CCS00	0	STS002	STS001	STS000

Symbol	7	6	5	4	3	2	1	0
TMR02H	CKS021	0	0	CCS02	MASTER02	STS022	STS021	STS020

Symbol	7	6	5	4	3	2	1	0
TMR0nH (n = 1, 3)	CKS0n1	0	0	CCS0n	SPLIT0n	STS0n2	STS0n1	STS0n0

Symbol	7	6	5	4	3	2	1	0
TMR0nL (n = 0 to 3)	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

CKS0n1	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CK00 set by timer clock select register 0 (TPS0)
1	Operation clock CK01 set by timer clock select register 0 (TPS0)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCS0n bit.	

CCS0n	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKS0n1 bit
1	Valid edge of input signal input from the TI0n pin
Count clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Cautions 1. Be sure to clear the following bits to 0.
TMR00H register: bits 3, 5, 6
TMR01H to TMR03H registers: bits 5, 6
TMR00L to TMR03L registers: bits 4, 5

2. The timer array unit must be stopped (TT0 = 0FH, TTH0 = 0AH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock (f_{MCK}) specified by using the CKS0n1 bit or the valid edge of the signal input from the TI0n pin is selected as the count clock (f_{TCLK}).

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-8. Format of Timer Mode Register 0n (TMR0n) (2/3)

Symbol	7	6	5	4	3	2	1	0
TMR00H	CKS001	0	0	CCS00	0	STS002	STS001	STS000

Symbol	7	6	5	4	3	2	1	0
TMR02H	CKS021	0	0	CCS02	MASTER02	STS022	STS021	STS020

Symbol	7	6	5	4	3	2	1	0
TMR0nH (n = 1, 3)	CKS0n1	0	0	CCS0n	SPLIT0n	STS0n2	STS0n1	STS0n0

(Bit 3 of TMR02H)

MASTER02	Selection of independent channel operation/simultaneous channel operation (slave/master) of channel n
0	Operates as the slave channel in the independent channel operation function or the simultaneous channel operation function.
1	Operates as the master channel in the simultaneous channel operation function.
Channel 0 or 2 can be set as the master channel. Channel 2 operates as the master channel when bit 3 (MASTER02) in TMR02H is set to 1. Channel 0 operates as the master channel regardless of the setting of bit 3 in TMR00H[Note] because channel 0 is highest in the order of channels. For the channel to be used as the independent channel operation function, MASTER02 = 0.	

(Bit 3 of TMR01H and TMR03H)

SPLIT0n	Selection of 8 or 16-bit timer operation for channels 1 and 3 (n = 1, 3)
0	Operates as 16-bit timer.
1	Operates as 8-bit timer.

STS 0n2	STS 0n1	STS 0n0	Setting of start trigger or capture trigger of channel n (n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products))
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of the TI0n pin input is used as the start trigger and capture trigger.
0	1	0	Both the edges of the TI0n pin input are used as a start trigger and a capture trigger.
1	0	0	When the channel is used as a slave channel with the one-shot pulse output, PWM output function, or multiple PWM output function: The interrupt request signal of the master channel (INTTM0n) is used as the start trigger.
1	1	0	When the channel is used as a slave channel in two-channel input with one-shot pulse output function: The interrupt request signal of the master channel (INTTM0n) is used as the start trigger. A valid edge of the TI03 pin input of the slave channel is used as the end trigger
Other than above			Setting prohibited

Note Bit 3 of TMR00H register is read-only and its value is fixed to 0. Writing is ignored.

Caution Be sure to clear the following bits to 0.

TMR00H register: bits 3, 5, 6

TMR01H to TMR03H registers: bits 5, 6

Figure 6-8. Format of Timer Mode Register 0n (TMR0n) (3/3)

Symbol	7	6	5	4	3	2	1	0
TMR0nL (n = 0 to 3)	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

CIS0n1	CIS0n0	Selection of TI0n pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STS0n2 to STS0n0 bits is other than 010B, set the CIS0n1 to CIS0n0 bits to 10B.		

MD 0n3	MD 0n2	MD 0n1	Setting of operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer/Square wave output/Divider function/PWM output (master)	Down count
0	1	0	Capture mode	Input pulse interval measurement/Two-channel input with one-shot pulse output function (slave)	Up count
0	1	1	Event counter mode	External event counter	Down count
1	0	0	One-count mode	Delay counter/One-shot pulse output/Two-channel input with one-shot pulse output function (master)/PWM output (slave)	Down count
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Up count
Other than above			Setting prohibited		
The operation of each mode changes depending on the operation of MD0n0 bit (refer to the table below).					

Operation mode (Value set by the MD0n3 to MD0n1 bits)	MD 0n0	Setting of starting counting and interrupt
• Interval timer mode Note 2 (0, 0, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• Capture mode (0, 1, 0)	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode Note 2 (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode Note 1 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, a timer interrupt is not generated.
	1	Start trigger is valid during counting operation Note 2. At that time, a timer interrupt is not generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time, a timer interrupt is not generated.
Other than above		Setting prohibited

Notes 1. In one-count mode, the interrupt request signal (INTTM0n) when starting a count operation and TO0n output are not controlled.

2. If the start trigger (TS0n = 1) is issued during operation, the counter is initialized, and recounting is started (interrupt request signal (INTTM0n) is not generated).

Caution **Be sure to clear bits 4 and 5 in registers TMR00L to TMR03L to 0.**

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.3.4 Timer status register 0n (TSR0n)

The TSR0n register indicates the overflow status of the counter of channel n.

The TSR0n register is valid only in the capture mode (MD0n3 to MD0n1 = 010B) and capture & one-count mode (MD0n3 to MD0n1 = 110B). It will not be set in any other mode. See Table 6-4 for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSR0n register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-9. Format of Timer Status Register 0n (TSR0n)

Address: F01A0H (TSR00), F01A2H (TSR01) After reset: 00H R
F01A4H (TSR02), F01A6H (TSR03)

Symbol	7	6	5	4	3	2	1	0
TMR00L	0	0	0	0	0	0	0	OVF

OVF	Counter overflow status of channel n
0	Overflow does not occur.
1	Overflow occurs.
When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.	

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Table 6-4. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF bit	Set/clear conditions
• Capture mode • Capture & one-count mode	clear	When no overflow has occurred upon capturing
	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	– (Use prohibited)
	set	

Remark The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

6.3.5 Timer channel enable status register 0 (TE0, TEH0 (8-bit mode))

The TE0 and TEH0 registers are used to enable or stop the timer operation of each channel.

Each bit of the TE0 and TEH0 registers correspond to each bit of the timer channel start register 0 (TS0, TSH0) and the timer channel stop register 0 (TT0, TTH0). When a bit of the TS0 and TSH0 registers is set to 1, the corresponding bit of TE0 and TEH0 is set to 1. When a bit of the TT0 and TTH0 registers is set to 1, the corresponding bit of TE0 and TEH0 is cleared to 0.

The TE0 and TEH0 registers can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TE0 and TEH0 registers to 00H.

Figure 6-10. Format of Timer Channel Enable Status Register 0 (TE0)

Address: F01B0H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TE0	0	0	0	0	TE03[Note]	TE02[Note]	TE01	TE00

TE0n	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
Indicates operation enable/stop status of the 16-bit timer. TE01 and TE03 bits indicate whether operation of the lower 8-bit timer is enabled or stopped when channels 1 and 3 are in the 8-bit timer mode.	

Note 16-pin products only.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-11. Format of Timer Channel Enable Status Register 0 (TEH0)

Address: F01B1H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TEH0	0	0	0	0	TEH03[Note]	0	TEH01	0

TEH0n	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit indicates whether operation of the higher 8-bit timer is enabled or stopped when channels 1 and 3 are in the 8-bit timer mode.	

Note 16-pin products only.

6.3.6 Timer channel start register 0 (TS0, TSH0 (8-bit mode))

The TS0 and TSH0 registers are trigger registers that are used to initialize timer counter register 0n (TCR0n) and start the counting operation of each channel.

When a bit of these registers is set to 1, the corresponding bit of timer channel enable status register 0 (TE0, TEH0) is set to 1. The TSH0n and TS0n bits are immediately cleared to 0 when operation is enabled (TE0n = 1), because they are trigger bits.

The TS0 and TSH0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TS0 and TSH0 registers to 00H.

Figure 6-12. Format of Timer Channel Start Register 0 (TS0)

Address: F01B2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TS0	0	0	0	0	TS03[Note]	TS02[Note]	TS01	TS00

TS0n	Operation enable (start) trigger of channel n (n = 0 to 3)
0	No trigger operation
1	The TE0n bit is set to 1 and the count operation becomes enabled. The TCR0n register count operation start in the count operation enabled state varies depending on each operation mode (see Table 6-5 in **6.5.2 Start timing of counter**). TS01 and TS03 bits are the trigger to enable operation (start operation) of the lower 8-bit timer when channels 1 and 3 are in the 8-bit timer mode.

Figure 6-13. Format of Timer Channel Start Register 0 (TSH0)

Address: F01B3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TSH0	0	0	0	0	TSH03[Note]	0	TSH01	0

TSH0n	Operation enable (start) trigger of channel n (n = 1, 3)
0	No trigger operation
1	The TEH0n bit is set to 1 and the count operation becomes enabled. The TCR0n register count operation start in the interval timer mode in the count operation enabled state (see Table 6-5 in **6.5.2 Start timing of counter**).
This bit is the trigger to enable operation (start operation) of the higher 8-bit timer when channels 1 and 3 are used in the 8-bit timer mode.	

Note 16-pin products only.

Cautions 1. Be sure to clear the following bits to 0.

TS0: Bits 2 to 7 in 10-pin products, bits 4 to 7 in 16-pin products

TSH0: Bits 0, 2 to 7 in 10-pin products, bits 0, 2, 4 to 7 in 16-pin products

2. When switching from a function that does not use TI0n pin input to one that does, the following wait period is required from when timer mode register 0n (TMR0n) is set until the TS0n bit is set to 1.

When the TI0n pin noise filter is enabled (TNFEN = 1):

Four cycles of the operation clock (f_{MCK})

When the TI0n pin noise filter is disabled (TNFEN = 0):

Two cycles of the operation clock (f_{MCK})

Remark When the TS0 and TSH0 registers are read, 0 is always read.

6.3.7 Timer channel stop register 0 (TT0, TTH0 (8-bit mode))

The TT0 and TTH0 registers are trigger registers that are used to stop the counting operation of each channel.

When a bit of TT0 and TTH0 registers is set to 1, the corresponding bit of timer channel enable status register 0 (TE0, TEH0) is cleared to 0. The TT0n and TTH0n bits are immediately cleared to 0 when operation is stopped (TE0n, TEH0n = 0), because they are trigger bits.

The TT0 and TTH0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears TT0 and TTH0 registers to 00H.

Figure 6-14. Format of Timer Channel Stop Register 0 (TT0)

Address: F01B4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TT0	0	0	0	0	TT03[Note]	TT02[Note]	TT01	TT00

TT0n	Operation stop trigger of channel n (n = 0 to 3)
0	No trigger operation
1	TE0n is cleared to 0, and counting operation is stopped. TT01 and TT03 bits are the trigger to stop operation of the lower 8-bit timer when channels 1 and 3 are in the 8-bit timer mode.

Figure 6-15. Format of Timer Channel Stop Register 0 (TTH0)

Address: F01B5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TTH0	0	0	0	0	TTH03[Note]	0	TTH01	0

TTH0n	Operation stop trigger of channel n (n = 1, 3)
0	No trigger operation
1	TEH0n is cleared to 0, and counting operation is stopped (stop trigger is generated).
This bit is the trigger to stop operation of the higher 8-bit timer when channels 1 and 3 are used in the 8-bit timer mode.	

Note 16-pin products only.

Caution Be sure to clear the following bits to 0.

TT0: Bits 2 to 7 in 10-pin products, bits 4 to 7 in 16-pin products

TTH0: Bits 0, 2 to 7 in 10-pin products, bits 0, 2, 4 to 7 in 16-pin products

Remark When the TT0 and TTH0 registers are read, 0 is always read.

6.3.8 Timer output enable register 0 (TOE0)

The TOE0 register is used to enable or disable timer output of each channel.

Channel n for which timer output has been enabled becomes unable to rewrite the value of the TO0n bit of timer output register 0 (TO0) described later by software, and the value reflecting the setting of the timer output function through the count operation is output from the timer output pin (TO0n).

The TOE0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-16. Format of Timer Output Enable Register 0 (TOE0)

Address: F01BAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOE0n	0	0	0	0	TOE03[Note]	TOE02[Note]	TOE01	TOE00

TOE0n	Timer output enable/disable of channel n
0	Disable output of timer. Without reflecting on TO0n bit timer operation, to fixed the output. Writing to the TO0n bit is enabled and the level set in the TO0n bit is output from the TO0n pin.
1	Enable output of timer. Reflected in the TO0n bit timer operation, to generate the output waveform. Writing to the TO0n bit is disabled (writing is ignored).

Note 16-pin products only.

Caution **Be sure to clear bits 2 to 7 in 10-pin products and bits 4 to 7 in 16-pin products to 0.**

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.3.9 Timer output register 0 (TO0)

The TO0 register is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TO0n) of each channel.

The TO0n bit of this register can be rewritten by software only when timer output is disabled (TOE0n = 0). When timer output is enabled (TOE0n = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the TO0n alternate pin as a port function pin, set the corresponding TO0n bit to 0.

The TO0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-17. Format of Timer Output Register 0 (TO0)

Address: F01B8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TO0	0	0	0	0	TO03[Note]	TO02[Note]	TO01	TO00

TO0n	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Note 16-pin products only.

Caution **Be sure to clear bits 2 to 7 in 10-pin products and bits 4 to 7 in 16-pin products to 0.**

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.3.10 Timer output level register 0 (TOL0)

The TOL0 register is a register that controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOE0n = 1) in the Slave channel output mode (TOM0n = 1). In the master channel output mode (TOM0n = 0), this register setting is invalid.

The TOL0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-18. Format of Timer Output Level Register 0 (TOL0)

Address: F01BCH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	TOL03[Note]	TOL02[Note]	TOL01	0

TOL0n	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Negative logic output (active-low)

Note 16-pin products only.

Caution **Be sure to clear bits 0, 2 to 7 in 10-pin products and bits 0, 4 to 7 in 16-pin products to 0.**

Remarks **1.** The timer output logic is inverted when the timer output signal changes next, instead of immediately after the TOL0 register value is rewritten.

2. n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.3.11 Timer output mode register 0 (TOM0)

The TOM0 register is used to control the timer output mode of each channel.

When a channel is used for the independent channel operation function, set the corresponding bit of the channel to be used to 0.

When a channel is used for the simultaneous channel operation function (one-shot pulse output, two-channel input with one-shot pulse output function [Note], PWM output, multiple PWM output [Note]), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOE0n = 1).

The TOM0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-19. Format of Timer Output Mode Register 0 (TOM0)

Address: F01BEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	TOM03[Note]	TOM02[Note]	TOM01	0

TOM0n	Control of timer output mode of channel n
0	Used as the independent channel operation function (to produce toggle output by the interrupt request signal (INTTM0n))
1	Slave channel output mode (output is set by the interrupt request signal (INTTM00, INTTM02) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel)

Note 16-pin products only.

Caution **Be sure to clear bits 0, 2 to 7 in 10-pin products and bits 0, 4 to 7 in 16-pin products to 0.**

Remark n: Master channel number
n = 0 (for 10-pin products); n = 0, 2 (for 16-pin products)

p: Slave channel number
p = 1 (for 10-pin products); n < p ≤ 3 (for 16-pin products)
(For details of the relation between the master channel and slave channel, refer to **6.4.1 Basic rules of simultaneous channel operation function**.)

6.3.12 Noise filter enable register 1 (NFEN1)

The NFEN1 register is used to set whether the noise filter can be used for the timer input (TI0n) pin signal to each channel.

Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.

When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected. When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for the target channel [Note]. For the timer input (TI0n) operation, see **6.5.1 (2) When valid edge of input signal via the TI0n pin is selected (CCS0n = 1)**, **6.5.2 Start timing of counter**, and **6.7 Timer Input (TI0n) Control**.

The NFEN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-20. Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	0	0	0	0	TNFEN03[Note]	TNFEN02[Note]	TNFEN01	TNFEN00

TNFEN0n	Enable/disable using noise filter of TI0n pin input signal (n = 0 to 3)
0	Noise filter OFF
1	Noise filter ON

Note 16-pin products only.

Caution **The applicable pin for the noise filter set by the TNFEN01 bit can be switched by setting the ISC1 bit in the input switch control register (ISC).**

ISC1 = 0: Whether or not to use the noise filter for the TI01 pin input signal can be selected.

ISC1 = 1: Whether or not to use the noise filter for the RxD0 pin input signal can be selected.

6.3.13 Input switch control register (ISC)

The ISC register is used to implement baud rate correction by using channel 1 in association with the serial array unit.

When the ISC1 bit is set to 1, the input signal of the serial data input (RxD0) pin is selected as a timer input (TI01).

The width at the baud rate (transfer rate) of the other party in communications can be measured by using the timer array unit input pulse interval measurement mode with the input edge signal of the start bit as a trigger.

The ISC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-21. Format of Input Switch Control Register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching channel 1 input of timer array unit
0	Uses the input signal of the TI01 pin as a timer input (normal operation).
1	Uses the input signal of the RxD0 pin as a timer input (detects the wakeup signal and measures the pulse width for baud rate correction).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD0 pin as an external interrupt (wakeup signal detection).

Caution Be sure to clear bits 7 to 2 to 0.

6.3.14 Registers controlling port functions of pins to be used for timer I/O

Using the timer array unit functions requires setting of the registers that control the port functions multiplexed on the target channels (port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx)). For details, see **4.3.1 Port mode registers 0, 4 (PM0, PM4), 4.3.2 Port registers 0, 4, 12, 13 (P0, P4, P12, P13),** and **4.3.5 Port mode control register 0 (PMC0)**.

For the setting examples when the port is used as timer I/O, see **4.5.3 Example of register settings for port and alternate functions used**.

When using the ports that share the pin with the timer output (such as P04/ANI3/TI01/TO01/KR5) for timer output, set the bit in the port mode register (PMxx), the port register (Pxx), and the port mode control register (PMCxx) corresponding to each port to 0.

Example: When using P04/ANI3/TI01/TO01/KR5 for timer output

Set the PMC04 bit of port mode control register 0 to 0.
Set the PM4 bit of port mode register 0 to 0.
Set the P04 bit of port register 0 to 0.

When using the ports (such as P04/ANI3/TI01/TO01/KR5) to be shared with the timer output pin for timer input, set the bit in the port mode register (PMxx) corresponding to each port to 1. Also set the bit in the port mode control register (PMCxx) corresponding to each port to 0. At this time, the bit in the port register (Pxx) may be 0 or 1.

Example: When using P04/ANI3/TI01/TO01/KR5 for timer input

Clear the PMC04 bit of port mode control register 0 to 0.
Set the PM04 bit of port mode register 0 to 1.

6.4 Basic Rules of Timer Array Unit

6.4.1 Basic rules of simultaneous channel operation function

When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

(1) Only an even channel (channel 0, 2) can be set as a master channel.
(2) Any channel, except channel 0, can be set as a slave channel [Note].
(3) The slave channel must be lower than the master channel.
Example: If channel 2 is set as a master channel, channel 3 can be set as a slave channel.
(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example: If channels 0 and 2 are set as the master channel, only channel 1 can be set as the slave channel of master channel 0. Channel 3 cannot be set as the slave channel of master channel 0.
(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKS0n1 bit (bit 7 of timer mode register 0nH (TMR0nH)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
(7) A master channel can transmit the interrupt request signal (INTTM0n), start software trigger, and count clock (f_{TCLK}) to the lower channels.
(8) A slave channel can use the interrupt request signal (INTTM0n), a start software trigger, or the count clock (f_{TCLK}) of the master channel as a source clock, but cannot transmit its own INTTM0n, start software trigger, or count clock (f_{TCLK}) to channels with lower channel numbers.
(9) A master channel cannot use the interrupt request signal (INTTM0n), a start software trigger, or the count clock (f_{TCLK}) from the other higher master channel as a source clock.
(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TS0n) of the channels in combination must be set at the same time.
(11) During the counting operation, a TS0n bit of a master channel or TS0n bits of all channels which are operating simultaneously can be set. It cannot be applied to TS0n bits of slave channels alone.
(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TT0n) of the channels in combination must be set at the same time.

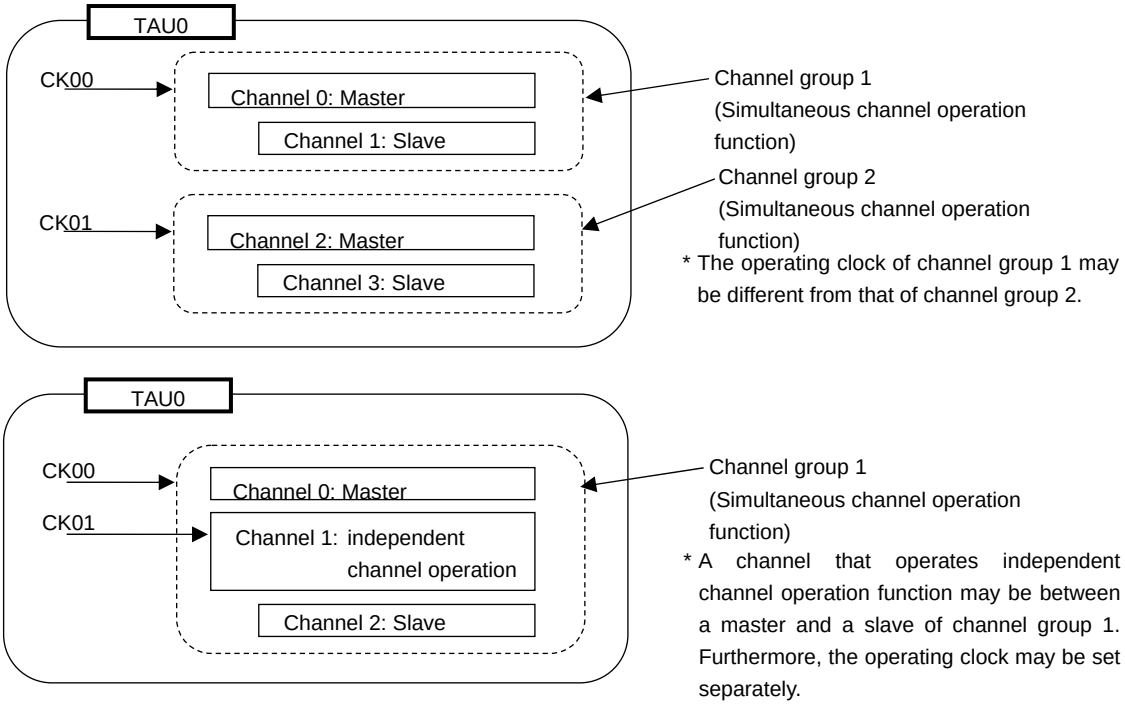
Note When channel 1 or 3 is used as an 8-bit timer, the lower 8 bits can be selected as the slave channel that operates in combination. In this case, the higher 8 bits of channel 1 or 3 can be used as an interval timer.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel operation function in **6.4.1 Basic rules of simultaneous channel operation function** do not apply to the channel groups.

Example


TAU0
CK00
Channel 0: Master
Channel 1: Slave
CK01
Channel 2: Master
Channel 3: Slave
Channel group 1
(Simultaneous channel operation function)
Channel group 2
(Simultaneous channel operation function)
* The operating clock of channel group 1 may be different from that of channel group 2.
TAU0
CK00
CK01
Channel 0: Master
Channel 1: independent channel operation
Channel 2: Slave
Channel group 1
(Simultaneous channel operation function)
* A channel that operates independent channel operation function may be between a master and a slave of channel group 1. Furthermore, the operating clock may be set separately.

6.4.2 Basic rules of 8-bit timer operation function (only channels 1 and 3)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

(1) The 8-bit timer operation function applies only to channels 1 and 3.
(2) When using 8-bit timers, set the SPLIT bit of timer mode register 0nH (TMR0nH) to 1.
(3) The higher 8 bits can be operated as the interval timer function.
(4) At the start of operation, the higher 8 bits output the interrupt request signal (INTTM01H, INTTM03H) (which is the same operation performed when MD0n0 is set to 1).
(5) The operation clock of the higher 8 bits is selected according to the CKS0n1 bit of the lower-bit TMR0nH register.
(6) For the higher 8 bits, the TSH0n bit is manipulated to start channel operation and the TTH0n bit is manipulated to stop channel operation. The channel status can be checked using the TEH0n bit.
(7) The lower 8 bits operate according to the settings of TMR0nH and TMR0nL registers. The lower 8-bit timer supports the following functions:
   - Interval timer
   - Square wave output
   - External event counter
   - Delay counter
   - PWM output function
   - Multiple PWM output function (16-pin products only)
(8) For the lower 8 bits, the TS0n bit is manipulated to start channel operation and the TT0n bit is manipulated to stop channel operation. The channel status can be checked using the TE0n bit.
(9) During 16-bit operation, manipulating the TSH0n/TTH0n bits is invalid. The TS0n and TT0n bits are manipulated to operate channel n. The TEH0n bit is not changed.
(10) For the 8-bit timer function, the simultaneous operation functions (one-shot pulse, PWM, and multiple PWM (16-pin only)) cannot be used.

Remark n: Channel number (n = 1, 3)

Caution When channels 1 and 3 are used in 8-bit timer mode (SPLIT = 1), it is prohibited to read the TCR01H and TDR01H registers or the TCR03H and TDR03H registers.

6.5 Operation of Counter

6.5.1 Count clock (f_{TCLK})

The count clock (f_{TCLK}) of the timer array unit can be selected between following by CCS0n bit of timer mode register 0n (TMR0n). .

- Operation clock (f_{MCK}) specified by the CKS0n1 bit
- Valid edge of input signal input from the TI0n pin

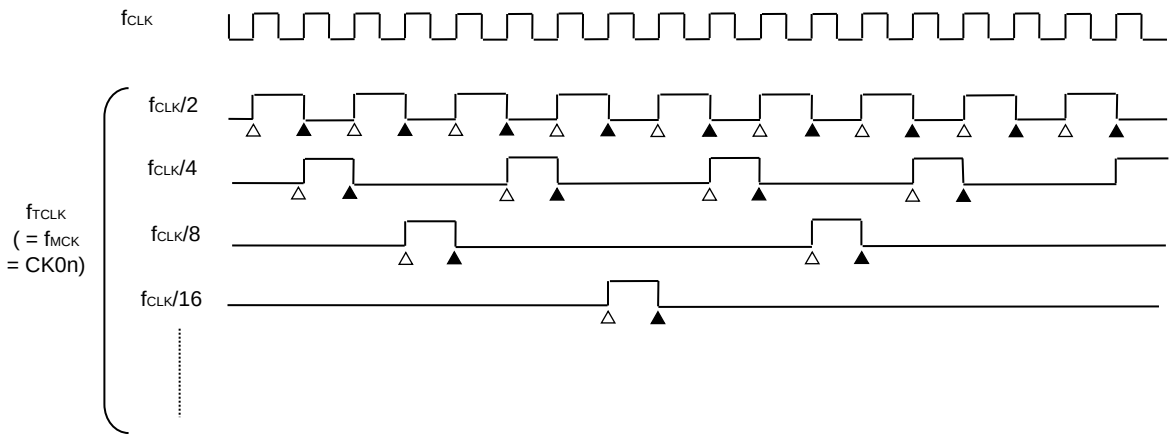
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operation clock (f_{MCK}) specified by the CKS0n1 bit is selected (CCS0n = 0)

The count clock (f_{TCLK}) is between f_{CLK} to f_{CLK} /2^{15} by setting of timer clock select register 0 (TPS0). When a divided f_{CLK} is selected, however, the clock selected in TPS0 register is at the high level for one f_{CLK} cycle period from its rising edge. When a f_{CLK} is selected, it is fixed to the high level

Counting of timer count register 0n (TCR0n) delayed by one f_{CLK} cycle period from rising edge of the count clock (f_{TCLK}), because of synchronization with f_{CLK}. But, this is described as “counting at rising edge of the count clock (f_{TCLK})”, as a matter of convenience.

Figure 6-22. Timing of f_{CLK} and Count Clock (f_{TCLK}) (When CCS0n = 0)



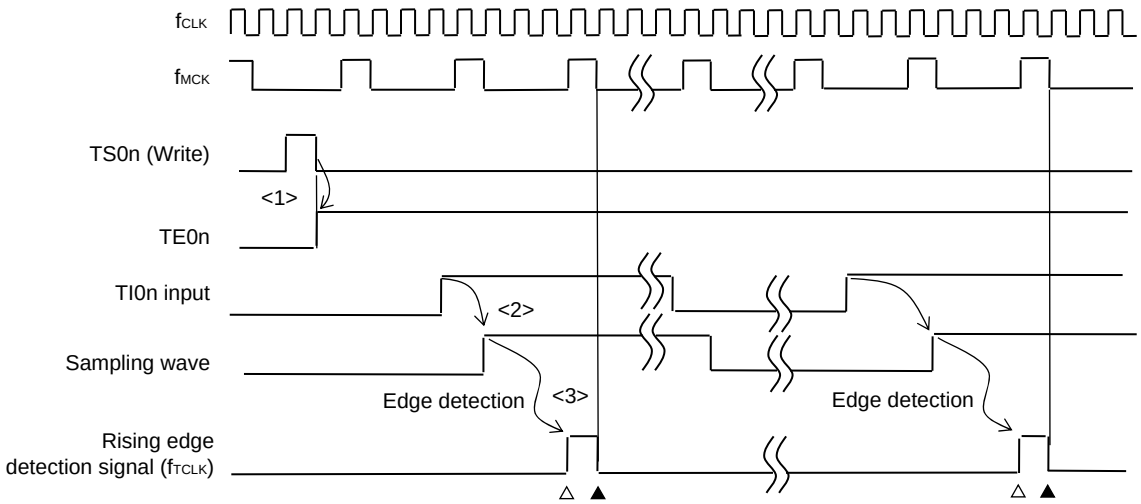
Remarks 1. △ : Rising edge of the count clock (f_{TCLK})
▲ : Synchronization, increment/decrement of counter
2. f_{CLK}: CPU/peripheral hardware clock

(2) When valid edge of input signal via the TI0n pin is selected (CCS0n = 1)

The count clock (f_{TCLK}) is the signal that detects valid edge of input signal via the TI0n pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TI0n pin (when a noise filter is used, it is delayed for 3 to 4 clock cycles).

Counting of timer count register 0n (TCR0n) delayed by one f_{CLK} cycle period from rising edge of the count clock (f_{TCLK}), because of synchronization with f_{CLK}. But, this is described as “counting at valid edge of input signal via the TI0n pin”, as a matter of convenience.

Figure 6-23. Timing of f_{CLK} and Count Clock (f_{TCLK}) (When CCS0n = 1, noise filter unused)



<1> Setting TS0n bit to 1 enables the timer to be started and the operation enters wait state for valid edge of input signal via the TI0n pin.

<2> The rise of input signal via the TI0n pin is sampled by f_{MCK}.

<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock (f_{TCLK})) is output.

Remarks 1. △ : Rising edge of the count clock (f_{TCLK})

▲ : Synchronization, increment/decrement of counter

2. f_{CLK}: CPU/peripheral hardware clock

f_{MCK}: Operation clock of channel n

3. The waveform of the input signal via TI0n pin of the input pulse interval measurement, the measurement of high/low width of input signal, and the delay counter, the one-shot pulse output are the same as that shown in **Figure 6-23**.

6.5.2 Start timing of counter

Timer count register 0n (TCR0n) operation becomes enabled by setting of TS0n bit of timer channel start register 0 (TS0).

Operations from count operation enabled state to timer count register 0n (TCR0n) count start is shown in **Table 6-5**.

Table 6-5. Operations from Count Operation Enabled State to Timer Count Register 0n (TCR0n) Count Start

Timer operation mode	Operation when TS0n = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TS0n = 1) until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see **6.5.3 (1) Interval timer mode operation**).
• Event counter mode	Writing 1 to the TS0n bit loads the value of the TDR0n register to the TCR0n register. Detection TI0n input edge, the subsequent count clock performs count down operation. (see **6.5.3 (2) Event counter mode operation**).
• Capture mode	No operation is carried out from start trigger (TS0n = 1) detection until count clock generation. The first count clock loads 0000H to the TCR0n register and the subsequent count clock performs count up operation (see **6.5.3 (3) Capture mode operation (input pulse interval measurement)**).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0n bit while the timer is stopped (TE0n = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDR0n register to the TCR0n register and the subsequent count clock performs count down operation (see **6.5.3 (4) One-count mode operation**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TS0n bit while the timer is stopped (TE0n = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCR0n register and the subsequent count clock performs count up operation (see **6.5.3 (5) Capture & one-count mode operation (high-level width is measured)**).

6.5.3 Counter operation

Here, the counter operation in each mode is explained.

(1) Interval timer mode operation

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit. Timer count register 0n (TCR0n) holds the initial value until count clock (f_{TCLK}) generation.

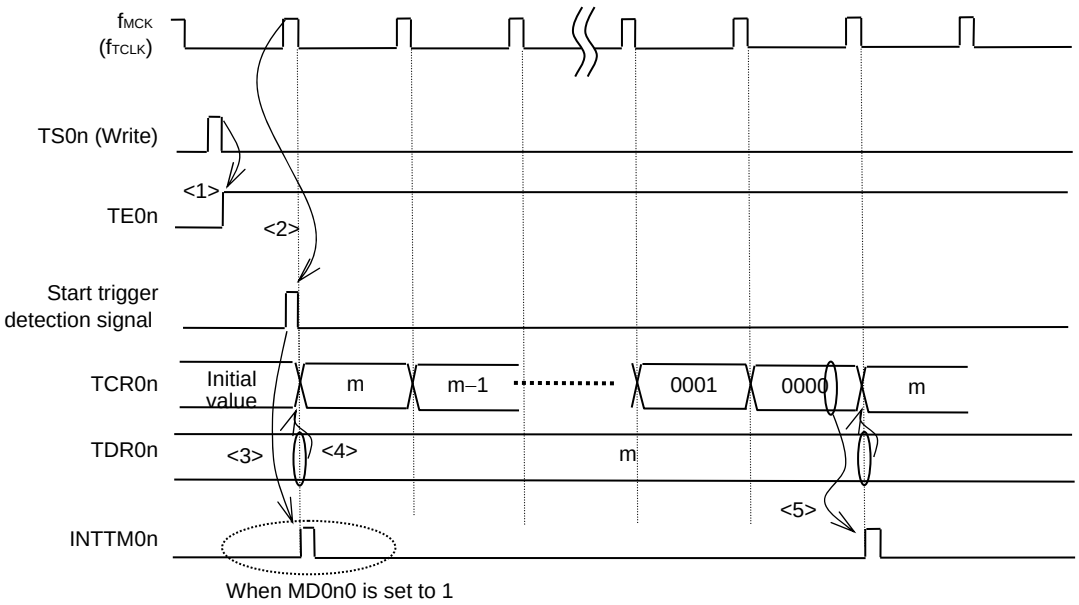
<2> A start trigger is generated at the first count clock after operation is enabled.

<3> When the MD0n0 bit is set to 1, INTTM0n is generated by the start trigger.

<4> By the first count clock after the operation enable, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting starts in the interval timer mode.

<5> When the TCR0n register counts down and its count value is 0000H, INTTM0n is generated in the next count clock and the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and counting keeps on.

Figure 6-24. Operation Timing (In Interval Timer Mode)



Caution In the first cycle operation of count clock (f_{TCLK}) after writing the TS0n bit, an error of a maximum of one cycle of the count clock (f_{TCLK}) is generated since count start delays until count clock (f_{TCLK}) has been generated. When the information on count start timing is necessary, the interrupt request signal (INTTM0n) can be generated at count start by setting MD0n0 = 1.

Remark f_{MCK}, the start trigger detection signal, and INTTM0n become active for one clock period in synchronization with f_{CLK}.

(2) Event counter mode operation

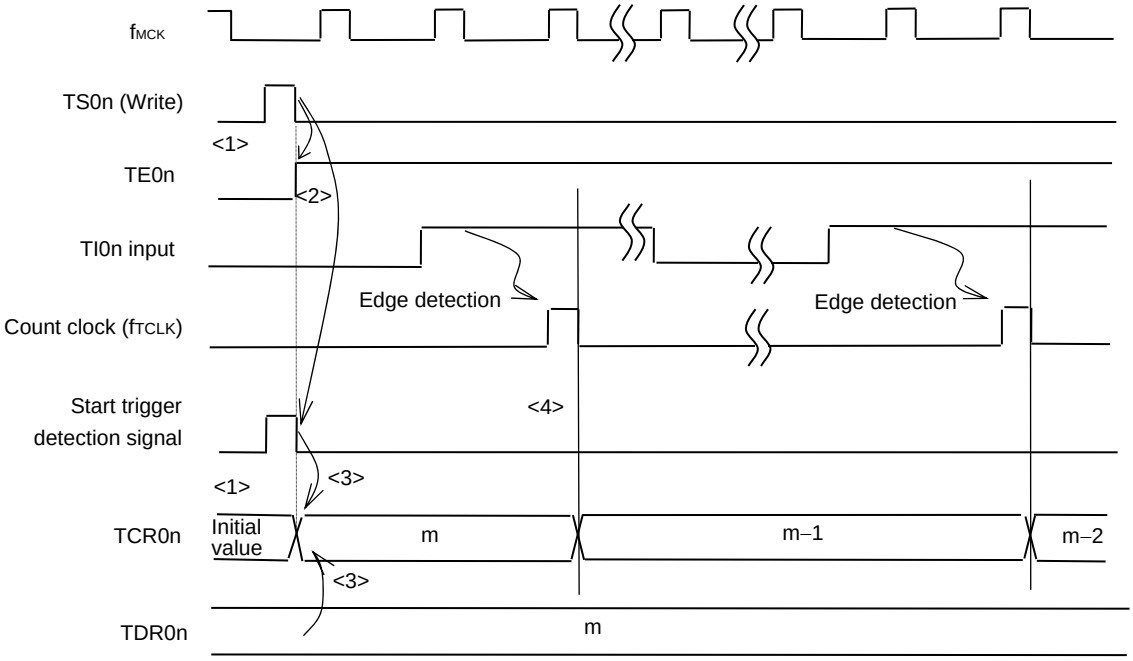
<1> Timer count register 0n (TCR0n) holds its initial value while operation is stopped (TE0n = 0).

<2> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.

<3> As soon as 1 has been written to the TS0n bit and 1 has been set to the TE0n bit, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register to start counting.

<4> After that, the TCR0n register value is counted down according to the count clock (f_{TCLK}) of the valid edge of the TI0n input.

Figure 6-25. Operation Timing (In Event Counter Mode)



Remark Figure 6-25 shows the timing when the noise filter is not used. When the noise filter is on-state, the edge detection is delayed by two cycles of the operating clock (f_{MCK}) from the TI0n input (totally 3 to 4 cycles). The error of one cycle is due to the asynchronous timing between the TI0n input and operating clock (f_{MCK}).

(3) Capture mode operation (input pulse interval measurement)

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.

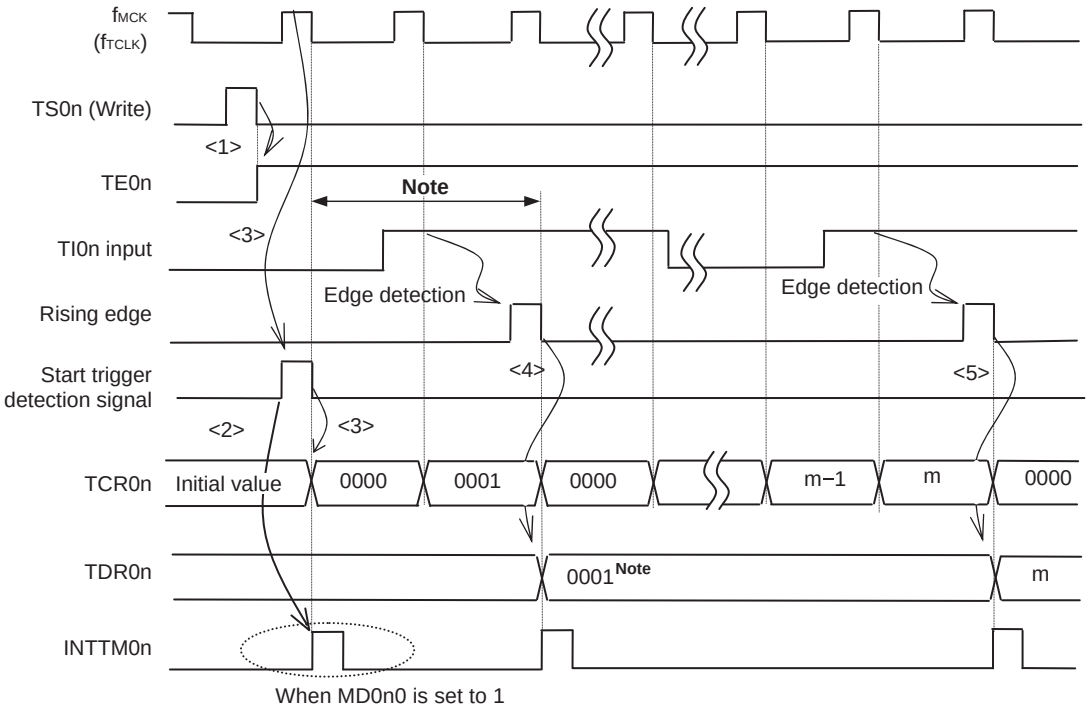
<2> Timer count register 0n (TCR0n) holds the initial value until count clock (f_{TCLK}) generation.

<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCR0n register and counting starts in the capture mode. (When the MD0n0 bit is set to 1, INTTM0n is generated by the start trigger.)

<4> On detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and the interrupt request signal (INTTM0n) is generated. However, this capture value has no meaning. The TCR0n register keeps on counting from 0000H.

<5> On next detection of the valid edge of the TI0n input, the value of the TCR0n register is captured to the TDR0n register and the interrupt request signal (INTTM0n) is generated.

Figure 6-26. Operation Timing (In Capture Mode: Input Pulse Interval Measurement)



Note If a clock has been input to TI0n (the trigger exists) when capturing starts, counting starts when a start trigger is generated (<3>) by writing to TS0n (<1>), even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

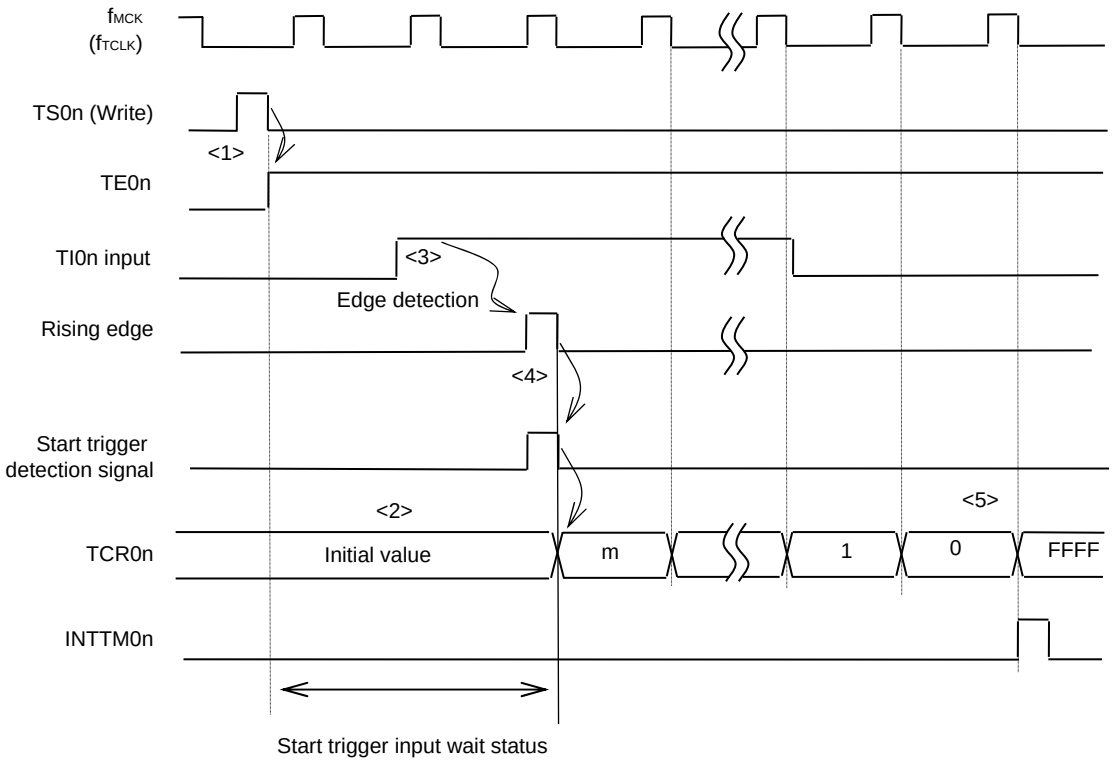
Caution In the first cycle operation of count clock (f_{TCLK}) after writing the TS0n bit, an error of a maximum of one cycle of the count clock (f_{TCLK}) is generated since count start delays until count clock (f_{TCLK}) has been generated. When the information on count start timing is necessary, the interrupt request signal (INTTM0n) can be generated at count start by setting MD0n0 = 1.

Remark Figure 6-26 shows the timing when the noise filter is not used. When the noise filter is on-state, the edge detection is delayed by two cycles of the operating clock (f_{MCK}) from the TI0n input (totally 3 to 4 cycles). The error of one cycle is due to the asynchronous timing between the TI0n input and operating clock (f_{MCK}).

(4) One-count mode operation

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit.
<2> Timer count register 0n (TCR0n) holds the initial value until start trigger generation.
<3> Rising edge of the TI0n input is detected.
<4> On start trigger detection, the value of timer data register 0n (TDR0n) is loaded to the TCR0n register and count starts.
<5> When the TCR0n register counts down and its count value is 0000H, the interrupt request signal (INTTM0n) is generated and the value of the TCR0n register becomes FFFFH and counting stops.

Figure 6-27. Operation Timing (In One-count Mode)

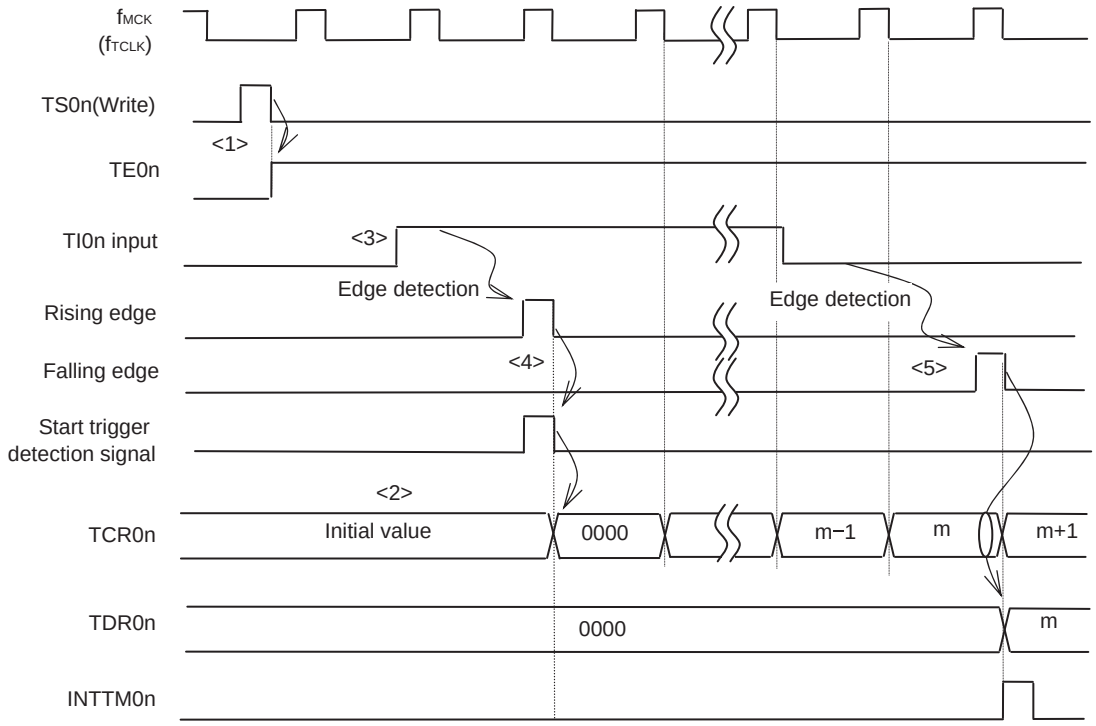


Remark Figure 6-27 shows the timing when the noise filter is not used. When the noise filter is on-state, the edge detection is delayed by two cycles of the operating clock (f_{MCK}) from the TI0n input (totally 3 to 4 cycles). The error of one cycle is due to the asynchronous timing between the TI0n input and operating clock (f_{MCK}).

(5) Capture & one-count mode operation (high-level width is measured)

<1> Operation is enabled (TE0n = 1) by writing 1 to the TS0n bit of timer channel start register 0 (TS0).
<2> Timer count register 0n (TCR0n) holds the initial value until start trigger generation.
<3> Rising edge of the TI0n input is detected.
<4> On start trigger detection, the value of 0000H is loaded to the TCR0n register and count starts.
<5> On detection of the falling edge of the TI0n input, the value of the TCR0n register is captured to timer data register 0n (TDR0n) and the interrupt request signal (INTTM0n) is generated.

Figure 6-28. Operation Timing (In Capture & One-count Mode: High-level Width Measurement)

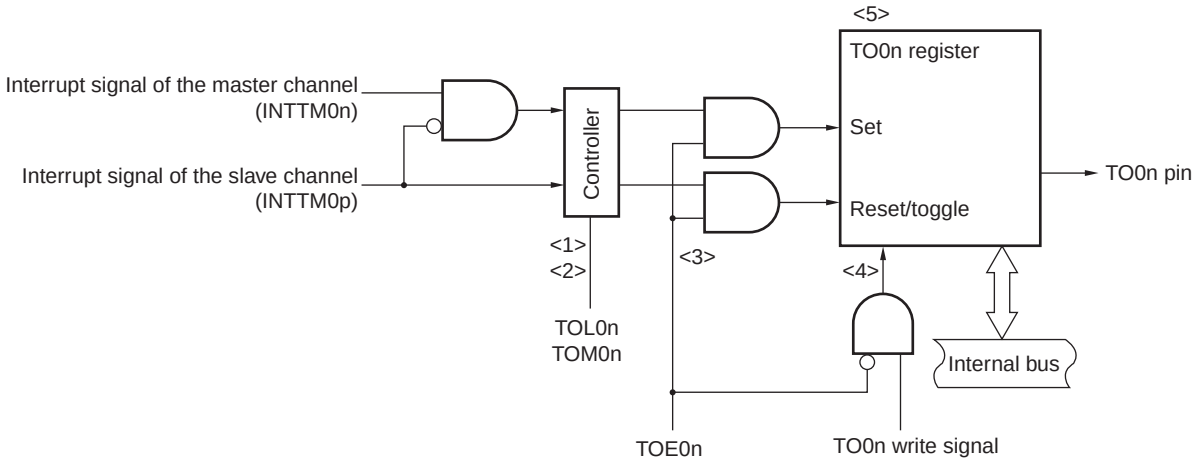


Remark Figure 6-28 shows the timing when the noise filter is not used. When the noise filter is on-state, the edge detection is delayed by two cycles of the operating clock (f_{MCK}) from the TI0n input (totally 3 to 4 cycles). The error of one cycle is due to the asynchronous timing between the TI0n input and operating clock (f_{MCK}).

6.6 Channel Output (TO0n pin) Control

6.6.1 TO0n pin output circuit configuration

Figure 6-29. Output Circuit Configuration



The following describes the TO0n pin output circuit.

<1> When TOM0n = 0 (master channel output mode), the set value of timer output level register 0 (TOL0) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register 0 (TO0).

<2> When TOM0n = 1 (slave channel output mode), both INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0 register.
At this time, the TOL0 register becomes valid and the signals are controlled as follows:

When TOL0n = 0: Positive logic output (INTTM0n → set, INTTM0p → reset)
When TOL0n = 1: Negative logic output (INTTM0n → reset, INTTM0p → set)

When INTTM0n and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTM0n (set signal) is masked.

<3> While timer output is enabled (TOE0n = 1), INTTM0n (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TO0 register. Writing to the TO0 register (TO0n write signal) becomes invalid.
When TOE0n = 1, the TO0n pin output never changes with signals other than interrupt signals.
To initialize the TO0n pin output level, it is necessary to set timer operation is stopped (TOE0n = 0) and to write a value to the TO0 register.

<4> While timer output is disabled (TOE0n = 0), writing to the TO0n bit to the target channel (TO0n write signal) becomes valid. When timer output is disabled (TOE0n = 0), neither INTTM0n (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TO0 register.

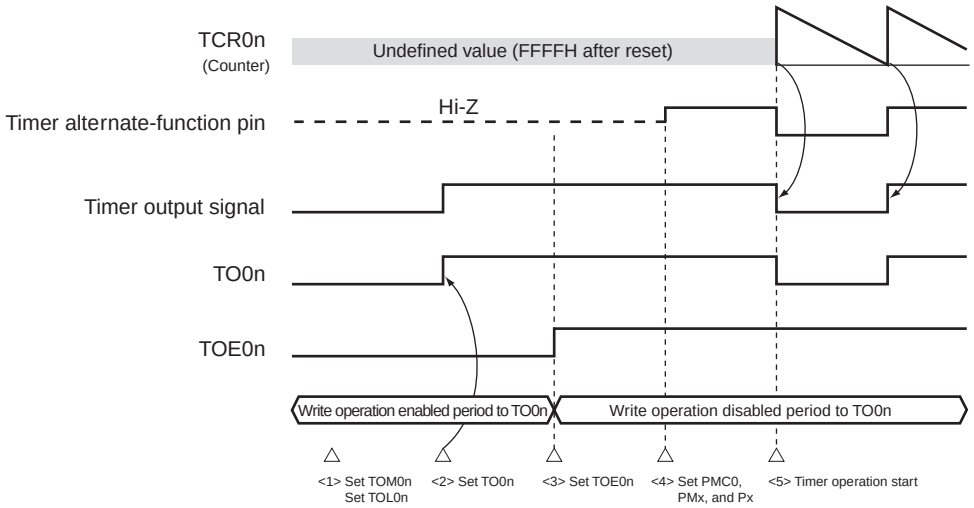
<5> The TO0 register can always be read, and the TO0n pin output level can be checked.

Remark n: Master channel number
n = 0 (for 10-pin products); n = 0, 2 (for 16-pin products)
p: Slave channel number
$n < p \leq 3$

6.6.2 TO0n pin output setting

The following figure shows the procedure and status transition of the TO0n output pin from initial setting to timer operation start.

Figure 6-30. Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOM0n bit (0: Master channel output mode, 1: Slave channel output mode)
- TOL0n bit (0: Positive logic output, 1: Negative logic output)

<2> The timer output signal is set to the initial status by setting timer output register 0 (TO0).

<3> The timer output operation is enabled by writing 1 to the TOE0n bit (writing to the TO0 register is disabled).

<4> The port is set to digital I/O by the port mode control register (PMC0). The port is set to output mode by the port mode register (PM0). The output latch for the port is set to 0 by the port register (P0) (see **6.3.14 Registers controlling port functions of pins to be used for timer I/O**).

<5> The timer operation is enabled (TS0n = 1).

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.6.3 Cautions on channel output operation

(1) Changing values set in the registers TO0, TOE0, TOL0, and TOM0 during timer operation

Since the timer operations (operations of timer count register 0n (TCR0n) and timer data register 0n (TDR0n)) are independent of the TO0n output circuit and changing the values set in timer output register 0 (TO0), timer output enable register 0 (TOE0), timer output level register 0 (TOL0), and timer output mode register 0 (TOM0) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TO0n pin by timer operation, however, set the TO0, TOE0, TOL0, and TOM0 registers to the values stated in the register setting example of each operation.
When the values set to the TOE0, TOL0, and TOM0 registers (but not the TO0 register) are changed close to the occurrence of the interrupt request signal (INTTM0n) of each channel, the waveform output to the TO0n pin might differ, depending on whether the values are changed immediately before or immediately after INTTM0n occurs.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

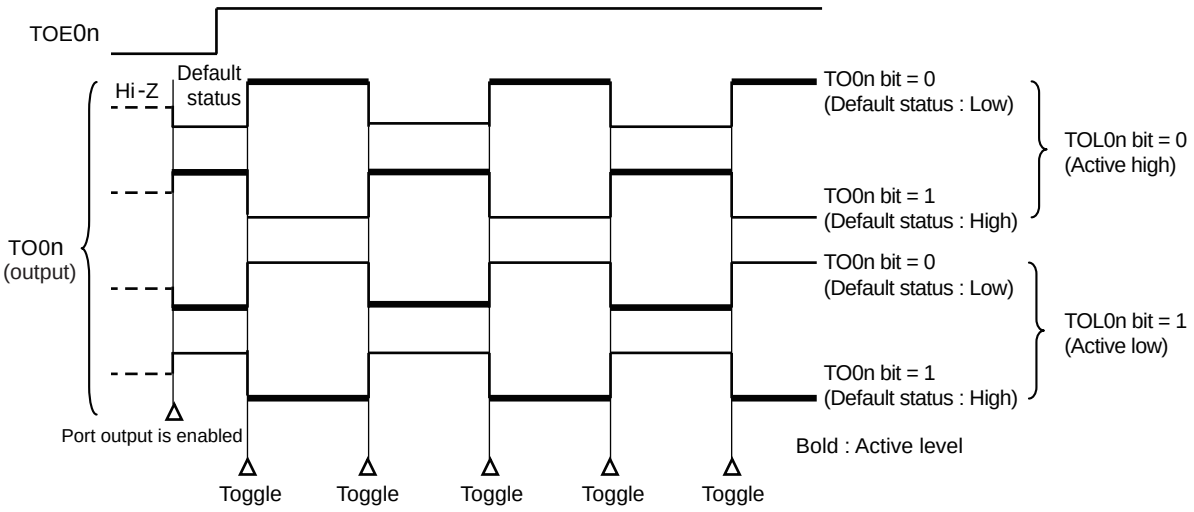
(2) Default level of TO0n pin and output level after timer operation start

The change in the output level of the TO0n pin when timer output register 0 (TO0) is written while timer output is disabled (TOE0n = 0), the initial level is changed, and then timer output is enabled (TOE0n = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOM0n = 0) setting

The setting of timer output level register 0 (TOL0) is invalid when master channel output mode (TOM0n = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TO0n pin is reversed.

Figure 6-31. TO0n Pin Output Status at Toggle Output (TOM0n = 0)

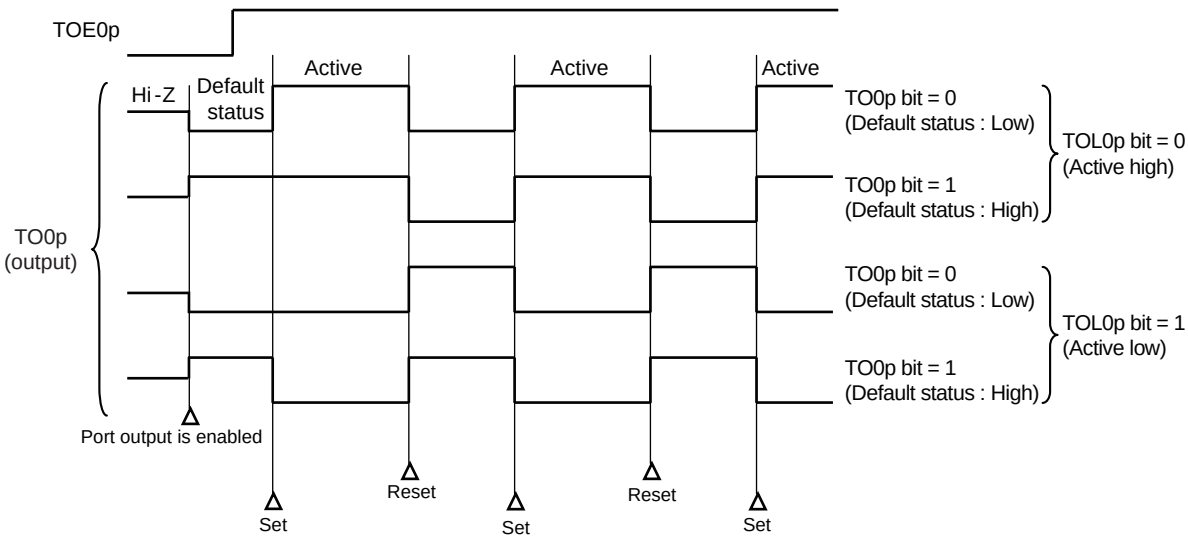


Remarks **1.** Toggle: Reverse TO0n pin output status
2. n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

(b) When operation starts with slave channel output mode (TOM0p = 1) setting (PWM output)

When slave channel output mode (TOM0p = 1), the active level is determined by timer output level register 0 (TOL0p) setting.

Figure 6-32. TO0p Pin Output Status at PWM Output (TOM0p = 1)



Remarks **1.** Set: The output signal of the TO0p pin changes from inactive level to active level.
Reset: The output signal of the TO0p pin changes from active level to inactive level.

2. p: Channel number (n < p ≤ 3)

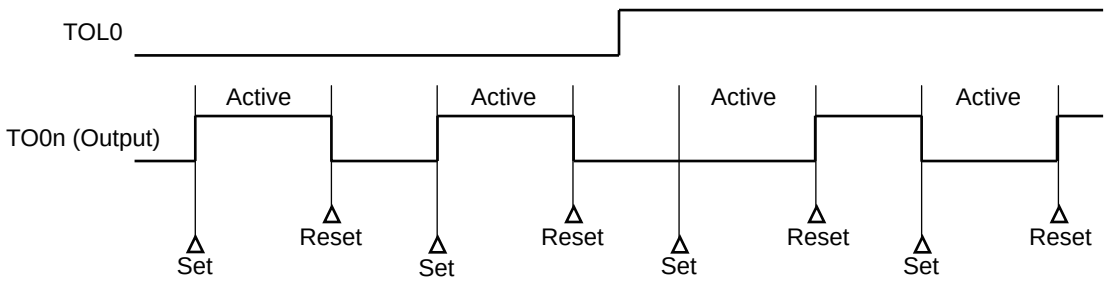
(3) Operation of TO0n pin in slave channel output mode (TOM0n = 1)

(a) When timer output level register 0 (TOL0) setting has been changed during timer operation

When the TOL0 register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TO0n pin change condition. Rewriting the TOL0 register does not change the output level of the TO0n pin.

The operation when TOM0n is set to 1 and the value of the TOL0 register is changed while the timer is operating (TE0n = 1) is shown below.

Figure 6-33. Operation when TOL0 Register Has Been Changed during Timer Operation



Remarks **1.** Set: The output signal of the TO0n pin changes from inactive level to active level.
Reset: The output signal of the TO0n pin changes from active level to inactive level.
2. n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TO0n pin/TO0n bit set timing at master channel interrupt request signal (INTTM0n) generation is delayed by one cycle of the count clock (f_{TCLK}) by the slave channel.

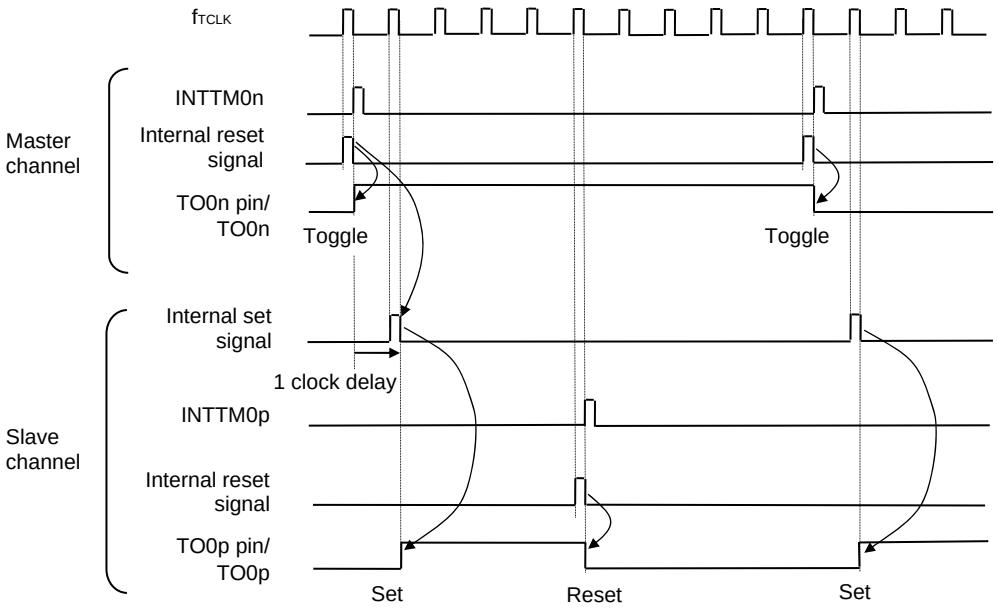
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-34 shows the set/reset operating statuses where the master/slave channels are set as follows.

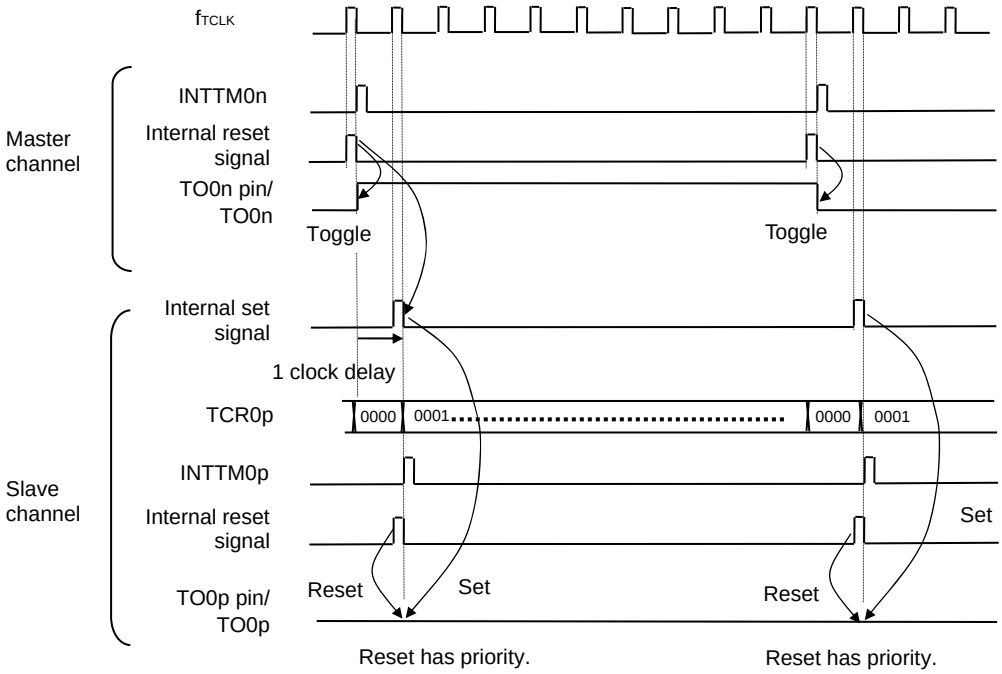
Master channel: TOE0n = 1, TOM0n = 0, TOL0n = 0
Slave channel: TOE0p = 1, TOM0p = 1, TOL0p = 0

Figure 6-34. Set/Reset Timing Operating Statuses

(a) Basic operation timing



(b) Operation timing when 0 % duty



Remarks **1.** Internal reset signal: TO0n pin reset/toggle signal
Internal set signal: TO0n pin set signal

2. n: Master channel number (for 10-pin products, n = 0; for 16-pin products, n = 0, 2)
p: Slave channel number ($n < p \leq 3$)

6.6.4 Collective manipulation of TO0n bit

In timer output register 0 (TO0), the setting bits for all the channels are located in one register in the same way as timer channel start register 0 (TS0). Therefore, the TO0n bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TO0n bits (TOE0n = 0) that correspond to the relevant bits of the channel used to perfor0 output (TO0n).

Figure 6-35. Example of TO0n Bit Collective Manipulation

Before writing

TO0	0	0	0	0	TO03 Note 0	TO02 Note 0	TO01 1	TO00 0
TOE0	0	0	0	0	TOE03 Note 0	TOE02 Note 0	TOE01 1	TOE00 0

Data to be written

TO0	0	0	0	0	1	1	0	0

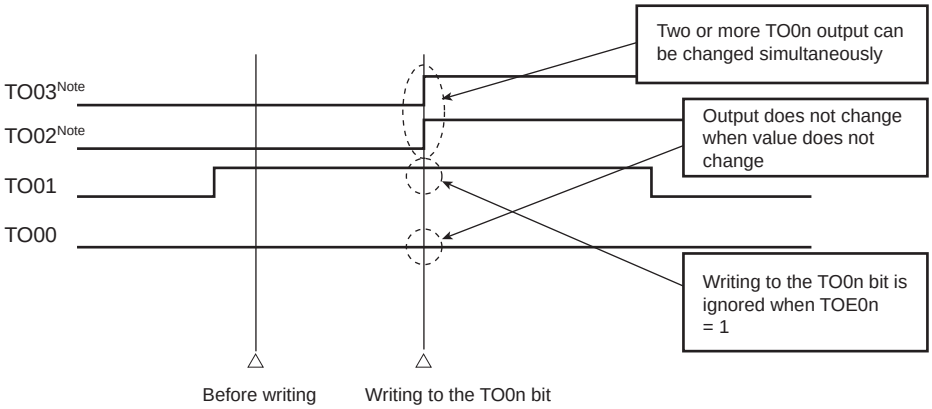
After writing

TO0	0	0	0	0	TO03 Note 1	TO02 Note 1	TO01 1	TO00 0

Writing is done only to the TO0n bit with TOE0n = 0, and writing to the TO0n bit with TOE0n = 1 is ignored.

TO0n (channel output) to which TOE0n = 1 is set is not affected by the write operation. Even if the write operation is done to the TO0n bit, it is ignored and the output change by timer operation is normally done.

Figure 6-36. TO0n Pin Statuses by Collective Manipulation of TO0n Bit



Note 16-pin products only.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.6.5 Timer interrupt and TO0n pin output at count operation start

In the interval timer mode or capture mode, the MD0n0 bit in timer mode register 0n (TMR0n) sets whether or not to generate a timer interrupt at count start.

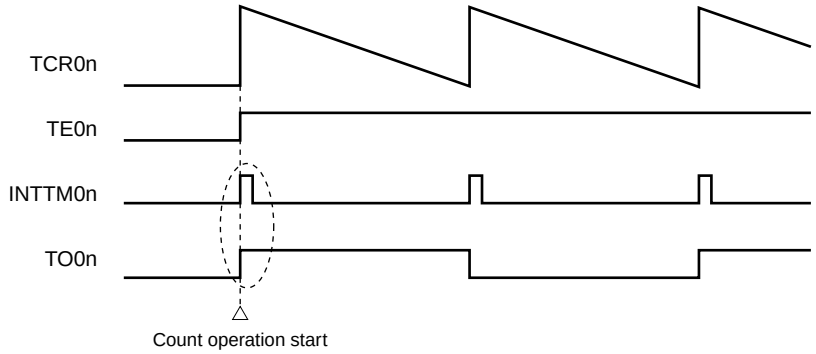
When MD0n0 is set to 1, the count operation start timing can be known by the interrupt request signal (INTTM0n) generation.

In the other modes, neither INTTM0n at count operation start nor TO0n output is controlled.

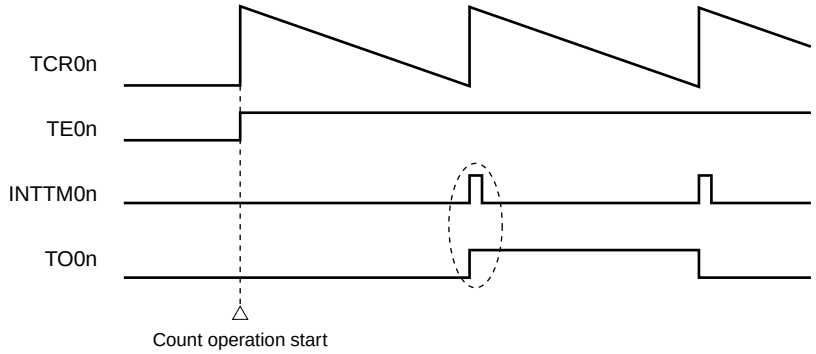
Figure 6-37 shows operation examples when the interval timer mode (TOE0n = 1, TOM0n = 0) is set.

Figure 6-37. Operation Examples of Timer Interrupt at Count Operation Start and TO0n Output

(a) When MD0n0 is set to 1



(b) When MD0n0 is set to 0



When MD0n0 is set to 1, the interrupt request signal (INTTM0n) is output at count operation start, and TO0n performs a toggle operation.

When MD0n0 is set to 0, the interrupt request signal (INTTM0n) is not output at count operation start, and TO0n does not change either. After counting one cycle, INTTM0n is output and TO0n performs a toggle operation.

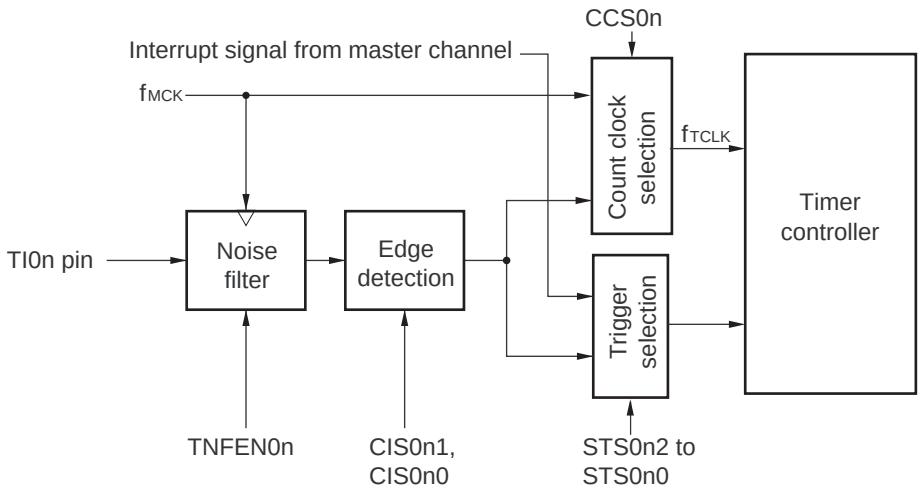
Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.7 Timer Input (TI0n) Control

6.7.1 TI0n input circuit configuration

A signal is input from a timer input pin, goes through a noise filter and an edge detector, and is sent to a timer controller. Enable the noise filter for the pin in need of noise removal. The following shows the configuration of the input circuit.

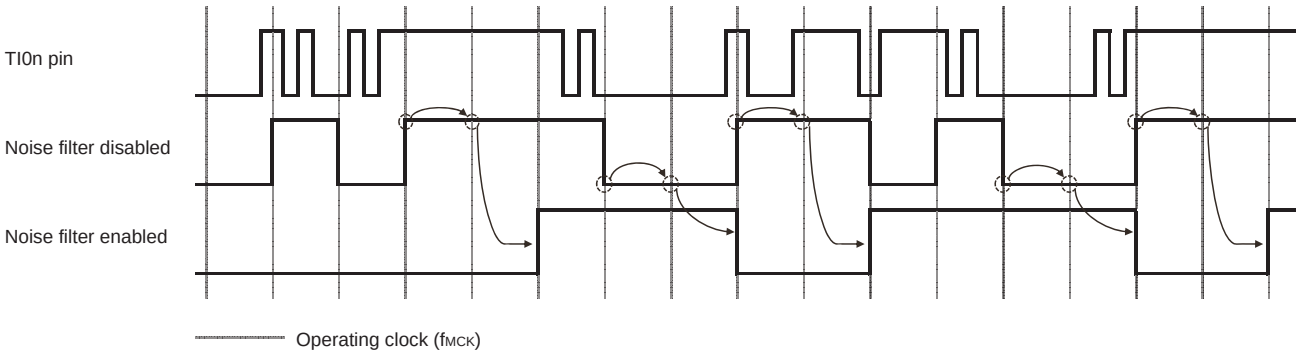
Figure 6-38. Input Circuit Configuration



6.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for channel n. When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for channel n, whether the signal keeps the same value for two clock cycles is detected. The following shows differences in waveforms output from the noise filter between when the noise filter for the TI0n pin is enabled and disabled.

Figure 6-39. Sampling Waveforms through TI0n Input Pin with Noise Filter Enabled and Disabled



6.7.3 Cautions on channel input operation

When a timer input pin is set as unused, the operating clock is not supplied to the noise filter. Therefore, after settings are made to use the timer input pin, the following wait time is necessary before a trigger is specified to enable operation of the channel corresponding to the timer input pin.

(1) Noise filter is disabled
When bits 12 (CCS0n), 9 (STS0n1), and 8 (STS0n0) in the timer mode register 0n (TMR0n) are all 0 and then one of them is set to 1, wait for at least two cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TS0).

(2) Noise filter is enabled
When bits 12 (CCS0n), 9 (STS0n1), and 8 (STS0n0) in the timer mode register 0n (TMR0n) are all 0 and then one of them is set to 1, wait for at least four cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TS0).

6.8 Independent Channel Operation Function of Timer Array Unit

6.8.1 Operation as interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates the interrupt request signal (INTTM0n) at fixed intervals.

The INTTM0n generation period can be calculated by the following expression.

Generation period of INTTM0n = Period of count clock × (Set value of TDR0n + 1)

When channel 1 or 3 is used as an 8-bit timer (SPLIT0n = 1), both the higher and lower 8-bit timers can be used as interval timers.

(2) Operation as square wave output

The TO0n pin performs a toggle operation as soon as INTTM0n has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TO0n can be calculated by the following expressions.

• Period of square wave output from TO0n pin = Period of count clock × (Set value of TDR0n + 1) × 2
• Frequency of square wave output from TO0n pin = Frequency of count clock/{(Set value of TDR0n + 1) × 2}

When channel 1 or 3 is used as an 8-bit timer (SPLIT0n = 1), only the lower 8-bit timer can be used for square wave output.

The timer count register 0n (TCR0n) operates as a down counter in the interval timer mode.

The TCR0n register loads the value of the timer data register 0n (TDR0n) at the first count clock after the channel start trigger bit (TS0n, TSH01, TSH03) of the timer channel start register 0 (TS0, TSH0) is set to 1. If the MD0n0 bit of timer mode register 0n (TMR0n) is 0 at this time, INTTM0n is not output and TO0n is not toggled. If the MD0n0 bit of the TMR0n register is 1, INTTM0n is output and TO0n is toggled.

After that, the TCR0n register count down in synchronization with the count clock.

When TCR0n = 0000H, INTTM0n is output and TO0n is toggled at the next count clock. At the same time, the TCR0n register loads the value of the TDR0n register again. After that, the same operation is repeated.

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-40. Block Diagram of Operation as Interval Timer/Square Wave Output

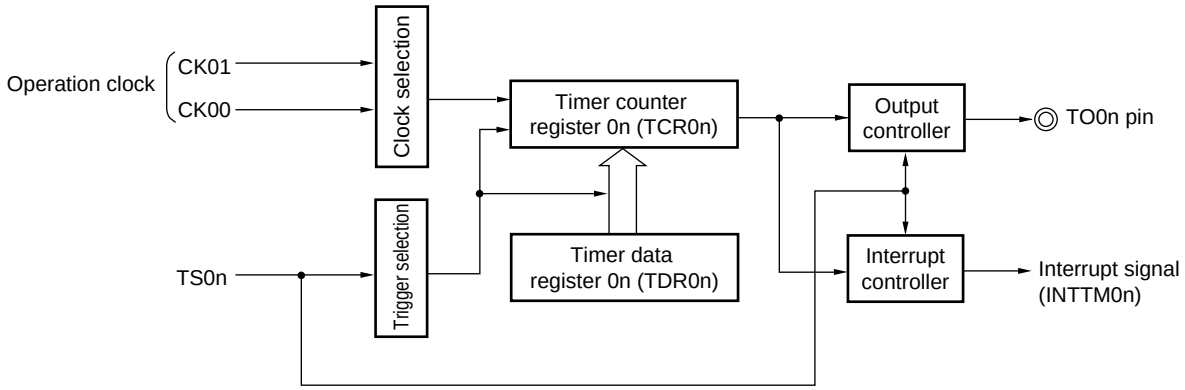
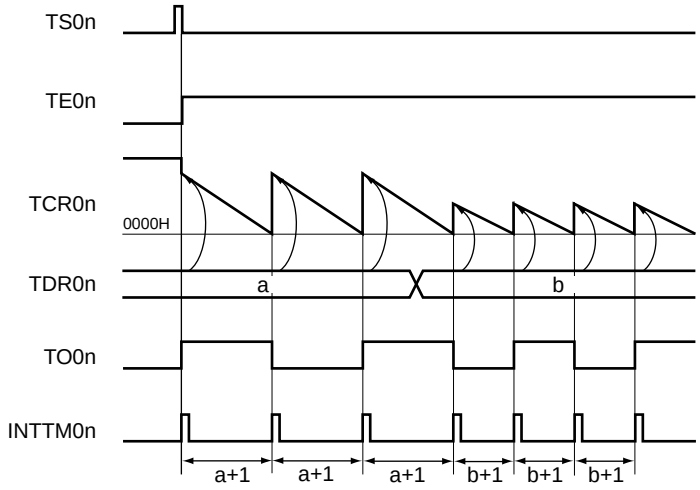


Figure 6-41. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MD0n0 = 1)



Remarks 1. n: Channel number

n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

2. TS0n: Bit n of timer channel start register 0 (TS0)

TE0n: Bit n of timer channel enable status register 0 (TE0)

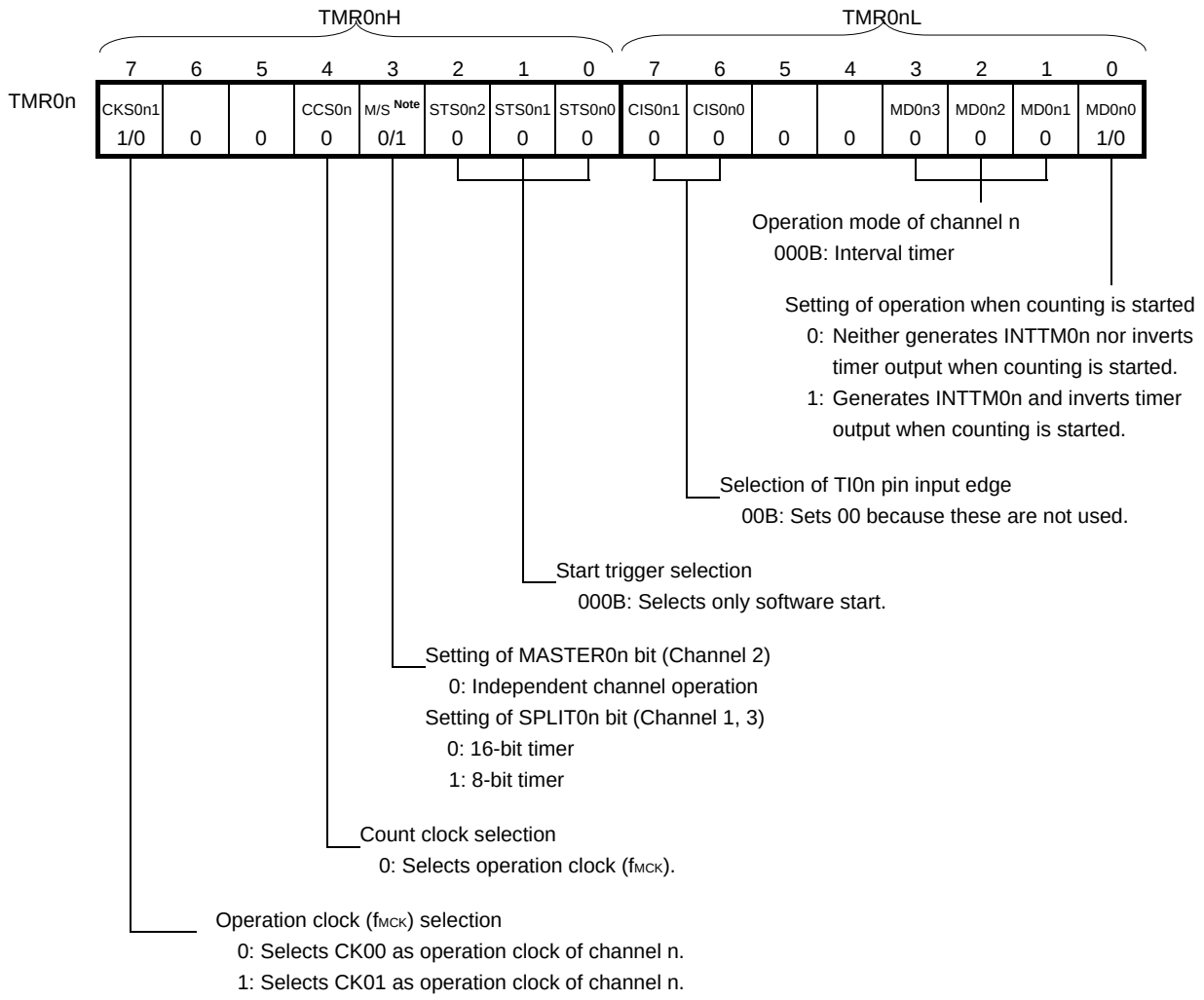
TCR0n: Timer count register 0n (TCR0n)

TDR0n: Timer data register 0n (TDR0n)

TO0n: TO0n pin output signal

Figure 6-42. Example of Set Contents of Registers for Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 1/0	0: Outputs 0 from TO0n. 1: Outputs 1 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 1/0	0: Stops the TO0n output operation by counting operation (the level set in the TO0n bit is output from the TO0n pin). 1: Enables the TO0n output operation by counting operation (output from the TO0n pin is toggled).

Note TMR02: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00: 0 fixed

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-42. Example of Set Contents of Registers for Operation as Interval Timer/Square Wave Output (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Setting is invalid because master channel output mode is set (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-43. Procedure for Operating Interval Timer/Outputting Square Wave (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to SFR of the TAU is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1 (when the TAU0EN bit is 0, read/write operation is disabled).	Power-on status. Each channel stops operating. (Clock supply is started and writing to SFR of the TAU is enabled.)
	Sets timer clock select register 0 (TPS0). Determines operating clock (CK00 and CK01) for each channel.	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode for each channel). Sets interval (period) value in the timer data register 0n (TDR0n) (for the access procedure to the TDR0nH and TDR0nL registers, see **6.2.2 Timer data register 0n (TDR0n)**).	Channel stops operating.
	Clears the target bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the target bit of the TOL0 register to 0. To use square wave output: Sets the TO0n bit and determines default level of the TO0n output. Sets the TOE0n bit to 1 and enables operation of TO0n. Clears the port register and port mode register to 0 (output mode is set).	The TO0n pin goes into Hi-Z state. (The port mode register is set to input mode.) TO0n does not change because channel stops operating (the TO0n pin is not affected even if the TO0n bit is modified). The level set in the TO0n bit is output from the TO0n pin.
Operation start	Sets the TOE0n bit to 1 and enables operation of TO0n (only if resuming square wave output operation). Sets the target bit of TS0 or TSH0 register to 1. The target bit of TS0 or TSH0 register automatically returns to 0 because it is a trigger bit.	The target bit of TE0 or TEH0 register is set to 1, and count operation starts. Value of the TDR0n register is loaded to timer count register 0n (TCR0n) at the next count clock. INTTM0n is generated and TO0n performs toggle operation if the MD0n0 bit of the TMR0nL register is 1.
During operation	The set value of the TDR0n register can be changed. The TCR0n register can always be read (for the access procedure to the TCR0nH and TCR0nL registers, see **6.2.1 Timer counter register 0n (TCR0n)**). The set values in the target bits of the TO0 and TOE0 register can be changed. The set values in the target bits of the TMR0n, TOM0, and TOL0 registers cannot be changed.	Counter (TCR0n) counts down. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again and the count operation is continued. By detecting TCR0n = 0000H, INTTM0n is generated and TO0n performs toggle operation. After that, the above operation is repeated.
Operation stop	Sets the target bit of TT0 or TTH0 register to 1. The target bit of TT0 or TTH0 register automatically returns to 0 because it is a trigger bit.	The target bit of the TE0 or TEH0 register is cleared to 0, and count operation stops. The TCR0n register holds count value and stops. The TO0n output is not initialized but holds current status.
	Clears the TOE0n bit to 0 and sets a value in the TO0n bit.	The level set in the TO0n bit is output from the TO0n pin.

Operation is resumed.

(**Remark** and **Caution** are listed on the next page.)

Figure 6-43. Procedure for Operating Interval Timer/Outputting Square Wave (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TO0n pin output level Clears the TO0n bit to 0 after the value to be held (output latch) is set in the port register. →	The TO0n pin output level is held by port function.
	Clears the TAU0EN bit of the PER0 register to 0. →	Power-off status (Clock supply is stopped and SFR of the TAU is initialized.)

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Caution When channels 1 and 3 are used in 8-bit timer mode (SPLIT = 1), it is prohibited to read the TCR01H and TDR01H registers or the TCR03H and TDR03H registers.

6.8.2 Operation as external event counter

The timer array unit can be used as an external event counter that counts the number of times the valid input edge (external event) is detected in the TI0n pin. When a specified count value is reached, the event counter generates the interrupt request signal (INTTM0n). The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDR0n + 1

When channel 1 or 3 is used as an 8-bit timer (SPLIT0n = 1), only the lower 8-bit timer can be used as the external event counter.

Timer count register 0n (TCR0n) operates as a down counter in the event counter mode.

The TCR0n register loads the value of timer data register 0n (TDR0n) by setting any channel start trigger bit (TS0n) of timer channel start register 0 (TS0) to 1.

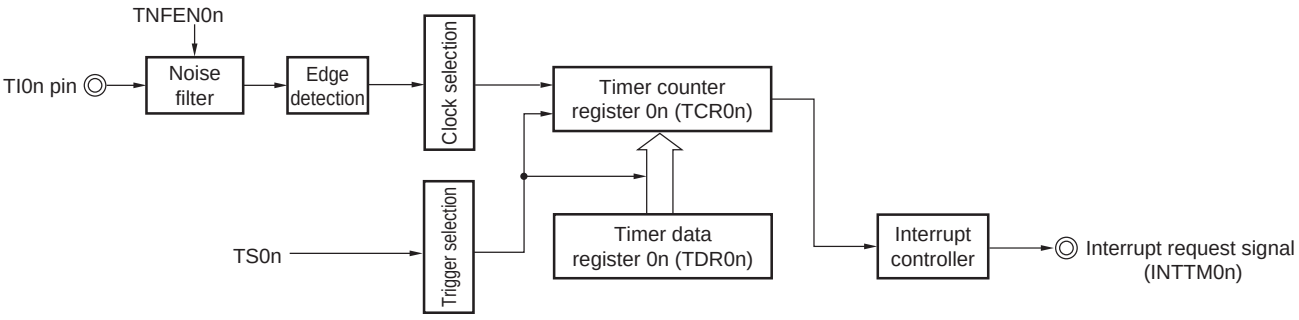
The TCR0n register counts down each time the valid input edge of the TI0n pin has been detected. When TCR0n = 0000H, the TCR0n register loads the value of the TDR0n register again, and outputs INTTM0n.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TO0n pin. Stop the output by setting the TOE0n bit of timer output enable register 0 (TOE0) to 0.

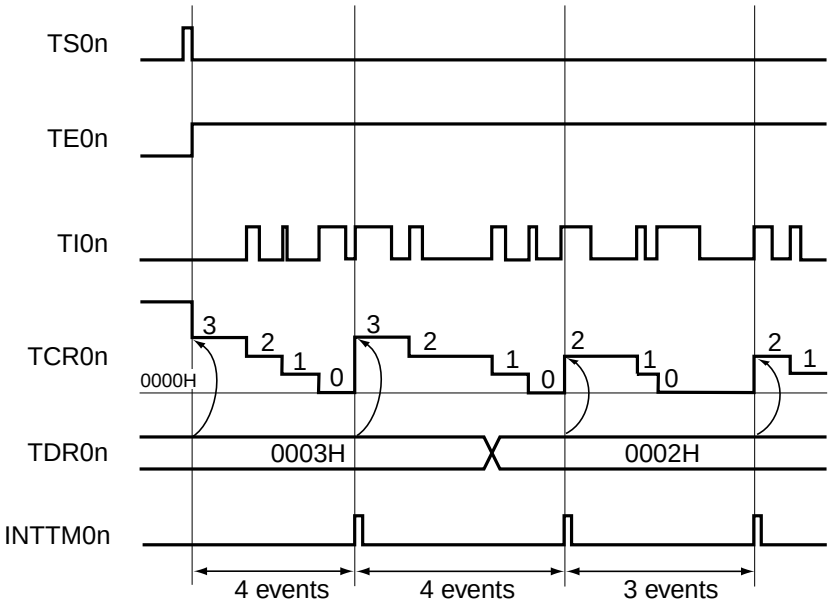
The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid during the next count period.

Figure 6-44. Block Diagram of Operation as External Event Counter



Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-45. Example of Basic Timing of Operation as External Event Counter



Remarks 1. n: Channel number

n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

2. TS0n: Bit n of timer channel start register 0 (TS0)

TE0n: Bit n of timer channel enable status register 0 (TE0)

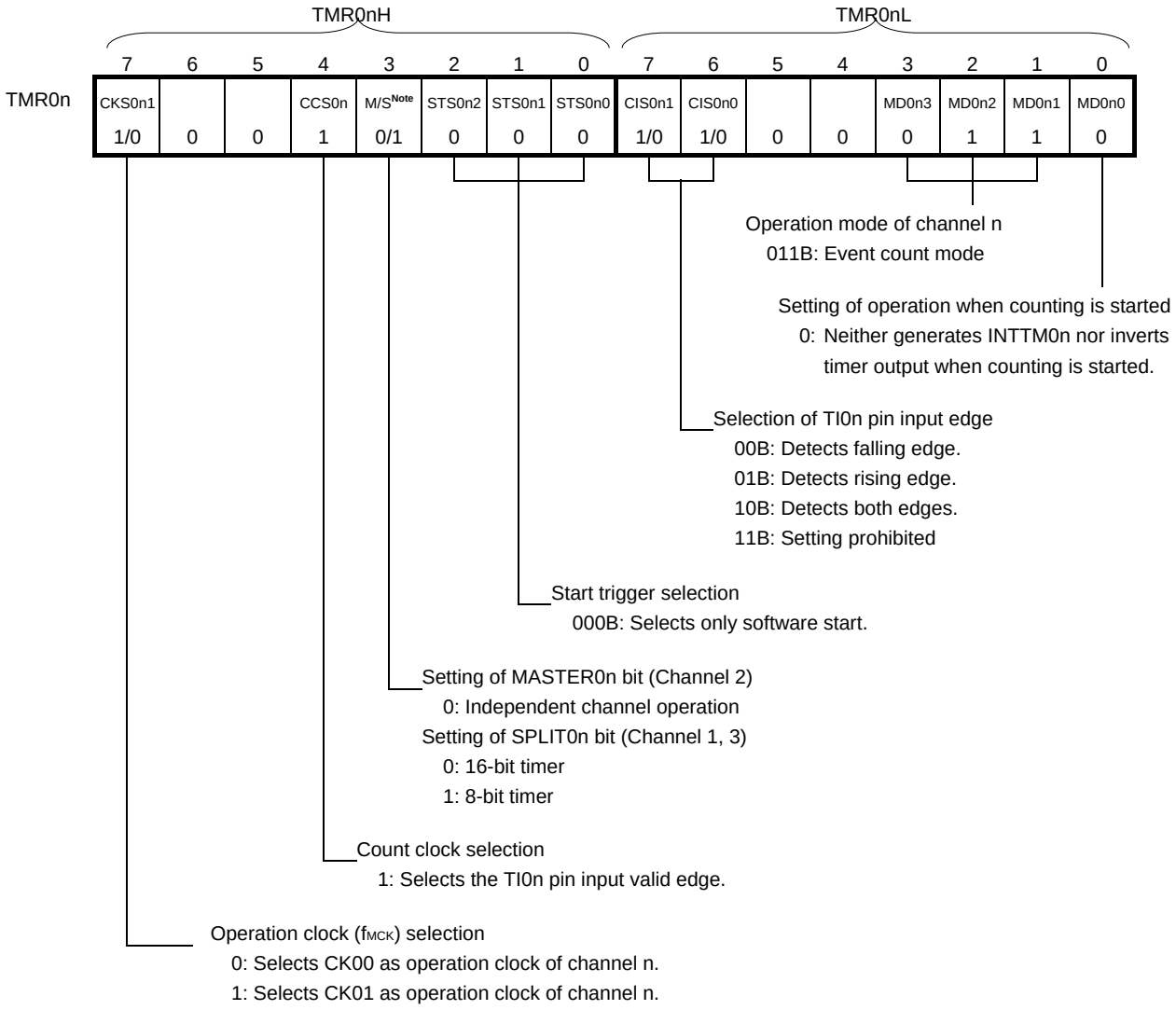
TI0n: TI0n pin input signal

TCR0n: Timer count register 0n (TCR0n)

TDR0n: Timer data register 0n (TDR0n)

Figure 6-46. Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



(b) Timer output register 0 (TO0)

	Bit n
TO0	TO0n 0

0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOE0n 0

0: Stops the TO0n output operation by counting operation.

Note TMR02: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00: 0 fixed

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-46. Example of Set Contents of Registers in External Event Counter Mode (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOL0n 0

0: Setting is invalid because master channel output mode is set (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOM0n 0

0: Sets master channel output mode.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-47. Procedure for Operating External Event Counter

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to SFR of the TAU is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1 (when the TAU0EN bit is 0, read/write operation is disabled).	Power-on status. Each channel stops operating. (Clock supply is started and writing to SFR of the TAU is enabled.)
	Sets timer clock select register 0 (TPS0). Determines operating clock (CK00 and CK01) for each channel.	
Channel default setting	Sets noise filter enable register 1 (NFEN1). Sets timer mode register 0n (TMR0n) (determines operation mode for each channel and selects the detection edge). Sets number of counts in the timer data register 0n (TDR0n) (for the access procedure to the TDR0nH and TDR0nL registers, see **6.2.2 Timer data register 0n (TDR0n)**). Clears the target bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the target bit of the TOL0 register to 0. Clears the target bit of timer output enable register 0 (TOE0) to 0.	Channel stops operating.
Operation start	Sets the target bit of TS0 register to 1. The target bit of TS0 register automatically returns to 0 because it is a trigger bit.	The target bit of TE0 register is set to 1, and count operation starts. Value of the TDR0n register is loaded to timer count register 0n (TCR0n) and detection of the TI0n pin input edge is awaited.
During operation	The set value of the TDR0n register can be changed. The TCR0n register can always be read (for the access procedure to the TCR0nH and TCR0nL registers, see **6.2.1 Timer counter register 0n (TCR0n)**). The set values in the target bits of the TMR0n, TO0, TOE0, TOM0, and TOL0 registers cannot be changed.	Counter (TCR0n) counts down each time input edge of the TI0n pin has been detected. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again, and the count operation is continued. By detecting TCR0n = 0000H, the INTTM0n is generated. After that, the above operation is repeated.
Operation stop	Sets the target bit of TT0 register to 1. The target bit of TT0 automatically returns to 0 because it is a trigger bit.	The target bit of TE0 register is cleared to 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	Clears the TAU0EN bit of the PER0 register to 0.	Power-off status (Clock supply is stopped and SFR of the TAU is initialized.)

Operation is resumed.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.8.3 Operation as frequency divider (only channels 0 and 3)

The timer array unit can be used as a frequency divider that divides a clock input to the TI0n pin and outputs the result clock from the TO0n pin.

The divided clock frequency output from TO0n can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDR0n + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDR0n + 1)

Timer count register 0n (TCR0n) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TCR0n register loads the value of timer data register 0n (TDR0n) when the TI0n valid edge is detected.

If the MD0n0 bit of timer mode register 0n (TMR0n) is 0 at this time, INTTM0n is not output and TO0n is not toggled. If the MD0n0 bit of timer mode register 0n (TMR0n) is 1, INTTM0n is output and TO0n is toggled.

After that, the TCR0n register counts down at the valid edge of the TI0n pin. When TCR0n = 0000H, it toggles TO0n. At the same time, the TCR0n register loads the value of the TDR0n register again, and continues counting.

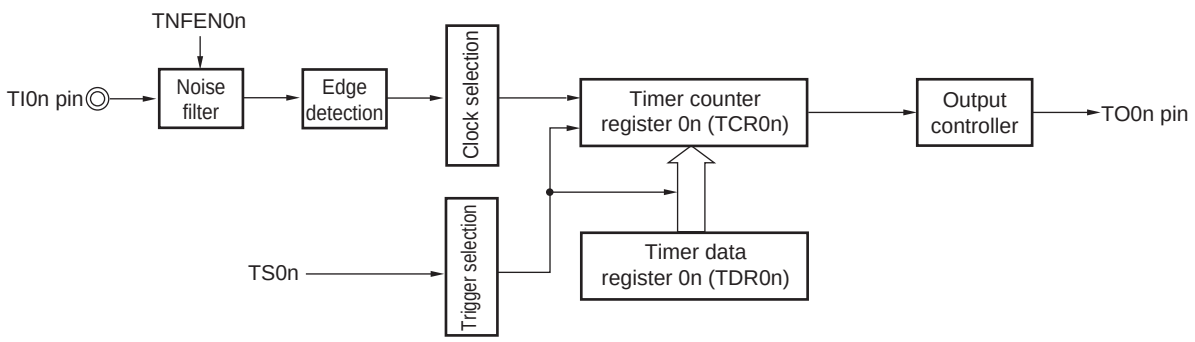
If detection of both the edges of the TI0n pin is selected, the duty factor error of the input clock affects the divided clock period of the TO0n output.

The period of the TO0n output clock includes a sampling error of a maximum of one operating clock (f_{MCK}) period.

Clock period of TO0n output = Ideal TO0n output clock period ± Operating clock period (error)

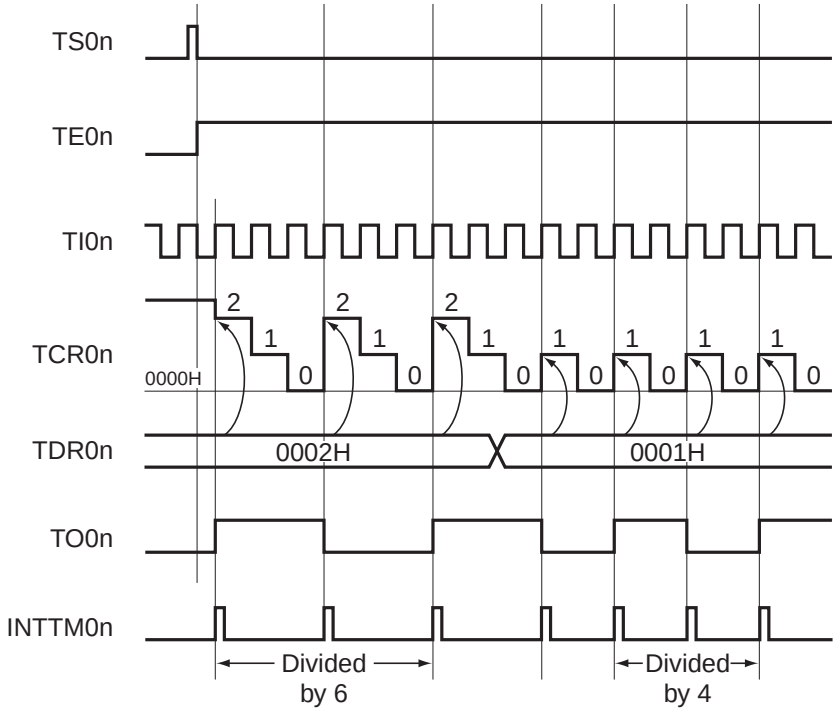
The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid during the next count period.

Figure 6-48. Block Diagram of Operation as Frequency Divider



Remark n = 0, 3

Figure 6-49. Example of Basic Timing of Operation as Frequency Divider (MD0n0 = 1)



Remark n: Channel number

n = 0 (for 10-pin products); n = 0, 3 (for 16-pin products)

TS0n: Bit n of timer channel start register 0 (TS0)

TE0n: Bit n of timer channel enable status register 0 (TE0)

TI0n: TI0n pin input signal

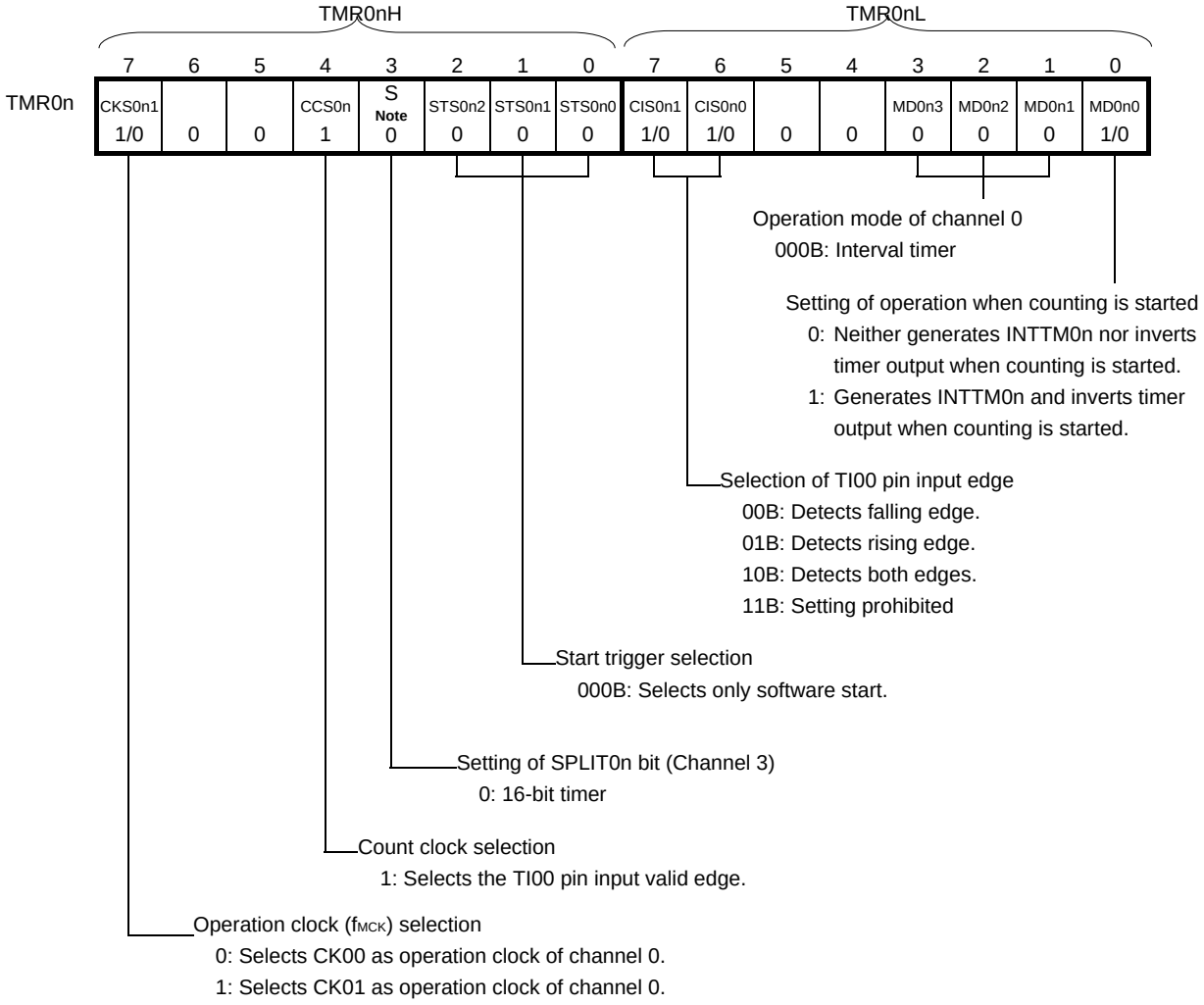
TCR0n: Timer count register 0n (TCR0n)

TDR0n: Timer data register 0n (TDR0n)

TO0n: TO0n pin output signal

Figure 6-50. Example of Set Contents of Registers During Operation as Frequency Divider (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 1/0	0: Outputs 0 from TO0n. 1: Outputs 1 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 1/0	0: Stops the TO0n output operation by counting operation (the level set in the TO0n bit is output from the TO0n pin.). 1: Enables the TO0n output operation by counting operation (output from the TO0n pin is toggled).

Note TMR03: SPLIT03 bit
TMR00: 0 fixed

Remark n: Channel number
n = 0 (for 10-pin products); n = 0, 3 (for 16-pin products)

Figure 6-50. Example of Set Contents of Registers During Operation as Frequency Divider (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOL0n 0

0: Setting is invalid because master channel output mode is set (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOM0n 0

0: Sets master channel output mode.

Remark n: Channel number
n = 0 (for 10-pin products); n = 0, 3 (for 16-pin products)

Figure 6-51. Procedure for Operating Frequency Divider

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to SFR of the TAU is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1 (when the TAU0EN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to SFR of the TAU is enabled.)
	Sets timer clock select register 0 (TPS0). Determines operating clock (CK00 and CK01) for each channel.	
Channel default setting	Sets noise filter enable register 1 (NFEN1). Sets timer mode register 0n (TMR0n) (determines operation mode for each channel and selects the detection edge). Sets interval (period) value in the timer data register 0n (TDR0n) (for the access procedure to the TDR0nH and TDR0nL registers, see **6.2.2 Timer data register 0n (TDR0n)**).	Channel stops operating.
	Clears the target bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the target bit of the TOL0 register to 0. Sets the TO0n bit and determines default level of the TO0n output. Sets the TOE0n bit to 1 and enables operation of TO0n. → Clears the port register and port mode register to 0 (output mode is set). →	The TO0n pin goes into Hi-Z state. (The port mode register is set to input mode.) TO0n does not change because channel stops operating (the TO0p pin is not affected even if the TO0p bit is modified). The level set in the TO0n bit is output from the TO0n pin.
Operation start	Sets the TOE0n bit to 1 and enables operation of TO0n (only when operation is resumed). Sets the target bit of TS0 register to 1. → The target bit of TS0 register automatically returns to 0 because it is a trigger bit.	The target bit of TE0 register is set to 1, and count operation starts. Value of the TDR0n register is loaded to timer count register 0n (TCR0n) at the next count clock. INTTM0n is generated and TO0n performs toggle operation if the MD0n0 bit of the TMR0nL register is 1.
During operation	The set value of the TDR0n register can be changed. The TCR0n register can always be read (for the access procedure to the TCR0nH and TCR0nL registers, see **6.2.1 Timer counter register 0n (TCR0n)**). The set values in the target bits of the TO0 and TOE0 registers can be changed. The set values in the target bits of the TMR0n, TOM0, and TOL0 registers cannot be changed.	Counter (TCR0n) counts down. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again, and the count operation is continued. By detecting TCR0n = 0000H, INTTM0n is generated and TO0n performs toggle operation. After that, the above operation is repeated.
Operation stop	Sets the target bit of TT0 register is to 1. → The target bit of TT0 automatically returns to 0 because it is a trigger bit.	The target bit of TE0n register is cleared to 0, and count operation stops. The TCR0n register holds count value and stops. The TO0n output is not initialized but holds current status.
	Clears the TOE0n bit to 0 and sets a value in the TO0n bit. →	The level set in the TO0n bit is output from the TO0n pin.
TAU stop	To hold the TO0n pin output level Clears the TO0n bit to 0 after the value to be held (output latch) is set to the port register. →	The TO0n pin output level is held by port function.
	Clears the TAU0EN bit of the PER0 register to 0. →	Power-off status (Clock supply is stopped and SFR of the TAU is initialized.)

Operation is resumed.

Remark n: Channel number
n = 0 (for 10-pin products); n = 0, 3 (for 16-pin products).

6.8.4 Operation as input pulse interval measurement

The count value can be captured on detection of a valid edge of TI0n pin input and the interval of the pulse input to TI0n pin can be measured. In addition, the count value can be captured by setting TS0n to 1 by software during the period of TE0n = 1.

For the UART0 baud rate correction, set bit 1 (ISC1) of the input switch control register (ISC) to 1.

In the following descriptions, read TI0n as RxD0. When the ISC1 bit is set to 1, the input signal of the serial data input (RxD0) pin is selected as a timer input (TI01). The width at the baud rate (transfer rate) of the other party in communications can be measured by using the input pulse interval measurement mode with the input edge signal of the start bit as a trigger.

The input pulse interval can be calculated by the following expression.

TI0n input pulse interval = Period of count clock × ((10000H × TSR0n: OVF) + (Capture value of TDR0n + 1))

Caution The TI0n pin input is sampled using the operating clock (f_{MCK}) selected with the CKS0n1 bit of timer mode register 0n (TMR0n), so an error of one cycle of the operating clock (f_{MCK}) occurs.

Timer count register 0n (TCR0n) operates as an up counter in the capture mode.

When the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TCR0n register counts up from 0000H in synchronization with the count clock.

When the TI0n pin input valid edge is detected, the count value of the TCR0n register is transferred (captured) to timer data register 0n (TDR0n) and, at the same time, the TCR0n register is cleared to 0000H, and the INTTM0n is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.

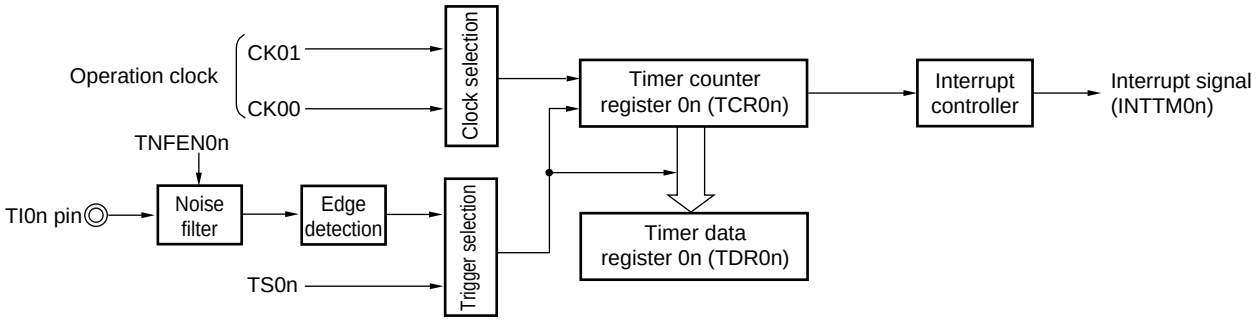
As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Set the STS0n2 to STS0n0 bits of the TMR0n register to 001B to use the valid edges of TI0n as a start trigger and a capture trigger.

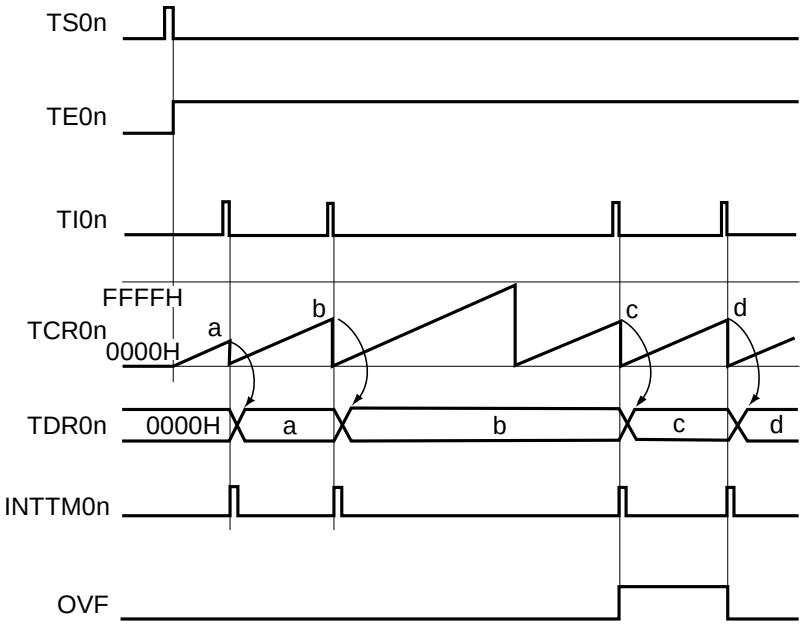
In addition, a software operation (TS0n = 1) can be used as a capture trigger, instead of using the TI0n pin input. When TS0n is set to 1 during TE0n = 1, the count value is captured in synchronization with the operating clock (f_{MCK}).

Figure 6-52. Block Diagram of Operation as Input Pulse Interval Measurement



Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

<R> **Figure 6-53. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MD0n0 = 0)**

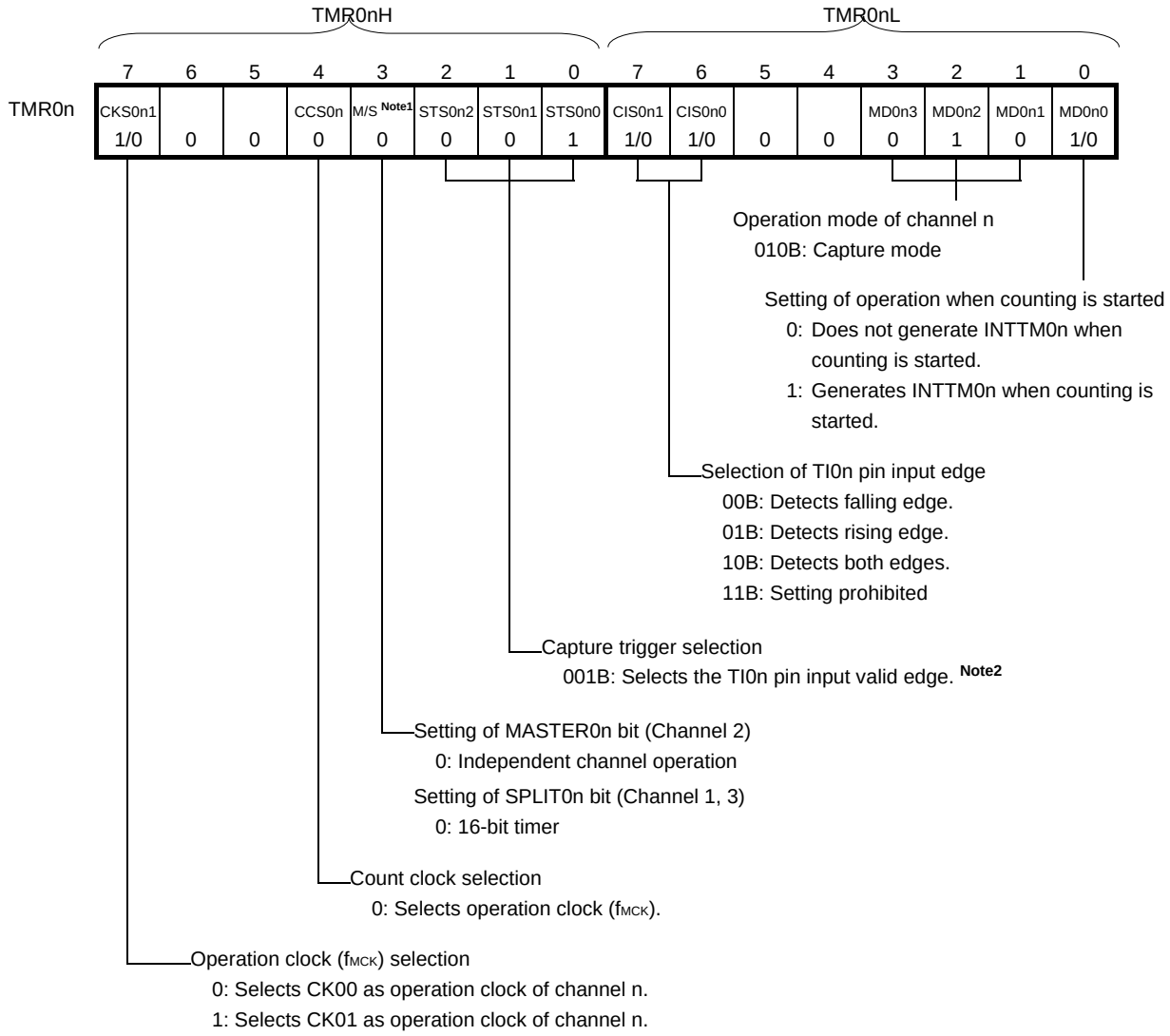


Remarks 1. n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

2. TS0n: Bit n of timer channel start register 0 (TS0)
TE0n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer count register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)
OVF: Bit 0 of timer status register 0n (TSR0n)

Figure 6-54. Example of Set Contents of Registers to Measure Input Pulse Interval (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



(b) Timer output register 0 (TO0)

	Bit n
TO0	TO0n 0

0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOE0n 0

0: Stops TO0n output operation by counting operation.

Notes **1.** TMR02: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00: 0 fixed

2. A software operation (TS0n = 1) can be used as a capture trigger, instead of using the TI0n pin input.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-54. Example of Set Contents of Registers to Measure Input Pulse Interval (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOL0n 0

0: Setting is invalid because master channel output mode is set (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOM0n 0

0: Sets master channel output mode.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-55. Procedure for Measuring Input Pulse Interval

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to SFR of the TAU is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1 (when the TAU0EN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to SFR of the TAU is enabled.)
	Sets timer clock select register 0 (TPS0). Determines operating clock (CK00 and CK01) for each channel.	
Channel default setting	Sets noise filter enable register 1 (NFEN1). Sets timer mode register 0n (TMR0n) (determines operation mode for each channel and selects the detection edge). Clears the target bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the target bit of the TOL0 register to 0. Clears the target bit of the timer output enable register (TOE0n) to 0.	Channel stops operating.
Operation start	Sets the target bit of TS0 register to 1. → The target bit of TS0 register automatically returns to 0 because it is a trigger bit.	The target bit of TE0 register is set to 1, and count operation starts. Timer count register 0n (TCR0n) is cleared to 0000H at the next count clock. When the MD0n0 bit of the TMR0n register is 1, INTTM0n is generated.
During operation	The set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. The TDR0n register can always be read (for the access procedure to the TDR0nH and TDR0nL registers, see **6.2.2 Timer data register 0n (TDR0n)**). The TCR0n register can always be read (for the access procedure to the TCR0nH and TCR0nL registers, see **6.2.1 Timer counter register 0n (TCR0n)**). The TSR0n register can always be read. The set values in the target bits of the TO0, TOE0, TOM0n, and TOL0n registers cannot be changed.	Counter (TCR0n) counts up from 0000H. When the TI0n pin input valid edge is detected or the TS0n bit is set to 1, the count value is transferred (captured) to timer data register 0n (TDR0n). At the same time, the TCR0n register is cleared to 0000H, and the INTTM0n signal is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	Sets the target bit of TT0 register is to 1. → The target bit of TT0 register automatically returns to 0 because it is a trigger bit.	The target bit of TE0 register is cleared to 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	Clears the TAU0EN bit of the PER0 register to 0. →	Power-off status (Clock supply is stopped and SFR of the TAU is initialized.)

Operation is resumed.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.8.5 Operation as input signal high-/low-level width measurement

By starting counting at one edge of the TI0n pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TI0n can be measured. The signal width of TI0n can be calculated by the following expression.

Signal width of TI0n input = Period of count clock × ((10000H × TSR0n: OVF) + (Capture value of TDR0n + 1))

Caution The TI0n pin input is sampled using the operating clock (f_{MCK}) selected with the CKS0n1 bit of timer mode register 0n (TMR0n), so an error of one cycle of the operating clock (f_{MCK}) occurs.

Timer count register 0n (TCR0n) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TE0n bit is set to 1 and the TI0n pin start edge detection wait status is set.

When the TI0n pin input start edge (rising edge of the TI0n pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TI0n pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register 0n (TDR0n) and, at the same time, INTTM0n is output. If the counter overflows at this time, the OVF bit of timer status register 0n (TSR0n) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCR0n register stops at the value “value transferred to the TDR0n register + 1”, and the TI0n pin start edge detection wait status is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDR0n register, the OVF bit of the TSR0n register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0n register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Whether the high-level width or low-level width of the TI0n pin is to be measured can be selected by using the CIS0n1 and CIS0n0 bits of the TMR0n register.

Because this function is used to measure the signal width of the TI0n pin input, the TS0n bit cannot be set to 1 while the TE0n bit is 1.

CIS0n1, CIS0n0 of TMR0n register = 10B: Low-level width is measured.

CIS0n1, CIS0n0 of TMR0n register = 11B: High-level width is measured.

Figure 6-56. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement

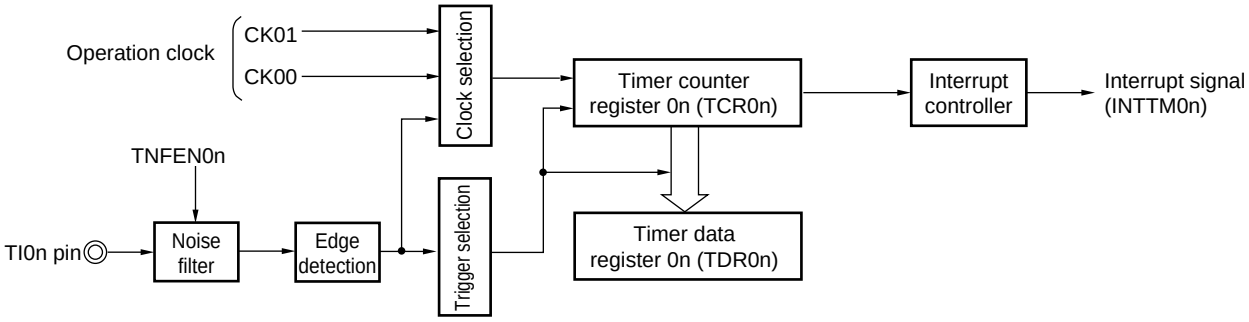
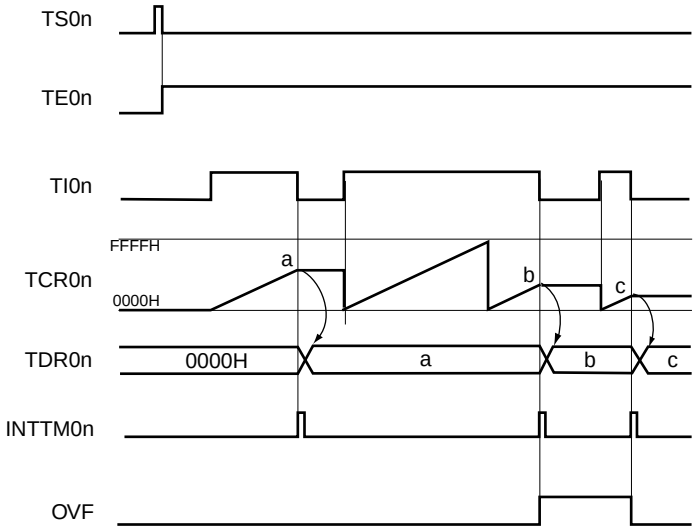


Figure 6-57. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remarks 1. n: Channel number

n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

2. TS0n: Bit n of timer channel start register 0 (TS0)

TE0n: Bit n of timer channel enable status register 0 (TE0)

TI0n: TI0n pin input signal

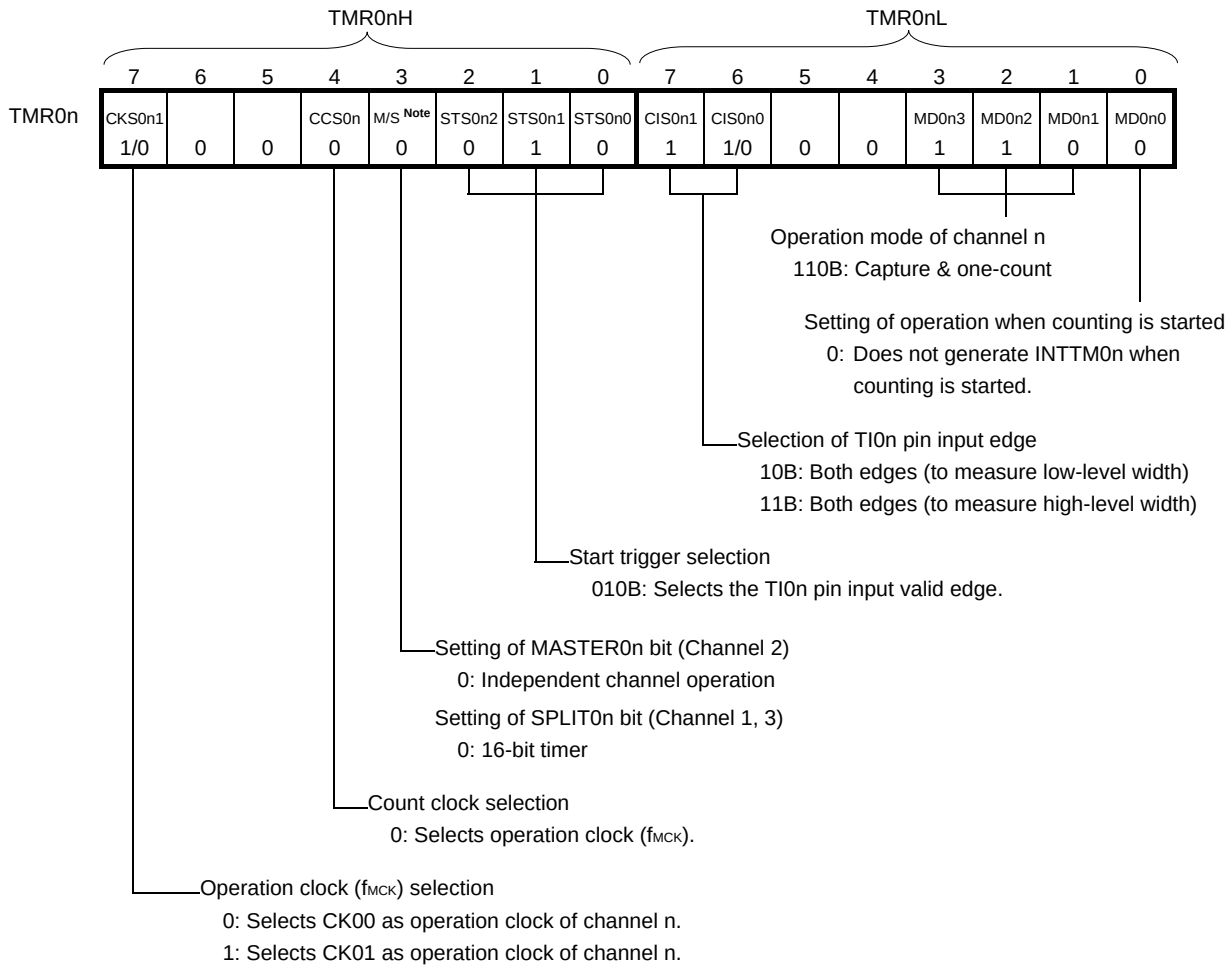
TCR0n: Timer count register 0n (TCR0n)

TDR0n: Timer data register 0n (TDR0n)

OVF: Bit 0 of timer status register 0n (TSR0n)

Figure 6-58. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Setting is invalid because master channel output mode is set (TOM0n = 0).

Note TMR02: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00: 0 fixed

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-58. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width (2/2)

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOM0n 0

0: Sets master channel output mode.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-59. Procedure for Measuring Input Signal High-/Low-Level Width

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to SFR of the TAU is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1 (when the TAU0EN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to SFR of the TAU is enabled.)
	Sets timer clock select register 0 (TPS0). Determines operating clock (CK00 and CK01) for each channel.	
Channel default setting	Sets noise filter enable register 1 (NFEN1). Sets timer mode register 0n (TMR0n) (determines operation mode for each channel and selects the detection edge). Clears the target bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the target bit of the TOL0 register to 0. Clears the target bit of the timer output enable register (TOE0n) to 0.	Channel stops operating.
Operation start	Sets the target bit of TS0 register to 1. → The target bit of TS0 register automatically returns to 0 because it is a trigger bit.	The target bit of TE0 register is set to 1, and the TI0n pin start edge detection wait status is set.
	Detects the TI0n pin input count start valid edge. →	Clears timer count register 0n (TCR0n) to 0000H and starts counting up.
During operation	The TDR0n register can always be read (for the access procedure to the TDR0nH and TDR0nL registers, see **6.2.2 Timer data register 0n (TDR0n)**). The TCR0n register can always be read (for the access procedure to the TCR0nH and TCR0nL registers, see **6.2.1 Timer counter register 0n (TCR0n)**). The TSR0n register can always be read. The set values in the target bits of the TO0, TOE0, TOM0n, and TOL0n registers cannot be changed.	When the TI0n pin start edge is detected, the counter (TCR0n) counts up from 0000H. If a capture edge of the TI0n pin is detected, the count value is transferred to timer data register 0n (TDR0n) and INTTM0n is generated. If an overflow occurs at this time, the OVF bit of timer status register 0n (TSR0n) is set; if an overflow does not occur, the OVF bit is cleared. The TCR0n register stops the count operation until the next TI0n pin start edge is detected. After that, the above operation is repeated.
Operation stop	Sets the target bit of TT0 register to 1. → The target bit of TT0 register automatically returns to 0 because it is a trigger bit.	The target bit of TE0 register is cleared to 0, and count operation stops. The TCR0n register holds count value and stops. The OVF bit of the TSR0n register is also held.
TAU stop	Clears the TAU0EN bit of the PER0 register to 0. →	Power-off status (Clock supply is stopped and SFR of the TAU is initialized.)

Operation is resumed.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.8.6 Operation as delay counter

It is possible to start counting down when the valid edge of the TI0n pin input is detected (an external event), and then generate the interrupt request signal (INTTM0n) after any specified interval.

It can also generate INTTM0n at any interval by setting TS0n to 1 by software to start the count down during the period of TE0n = 1.

The interrupt request signal (INTTM0n) generation period can be calculated by the following expression.

Generation period of interrupt request signal (INTTM0n) = Period of count clock × (Set value of TDR0n + 1)

Caution The TI0n pin input is sampled using the operating clock (f_{MCK}) selected with the CKS0n1 bit of timer mode register 0n (TMR0n), so an error of one cycle of the operation clock (f_{MCK}) occurs.

When channel 1 or 3 is used as an 8-bit timer (SPLIT0n = 1), only the lower 8-bit timer can be used as the delay counter.

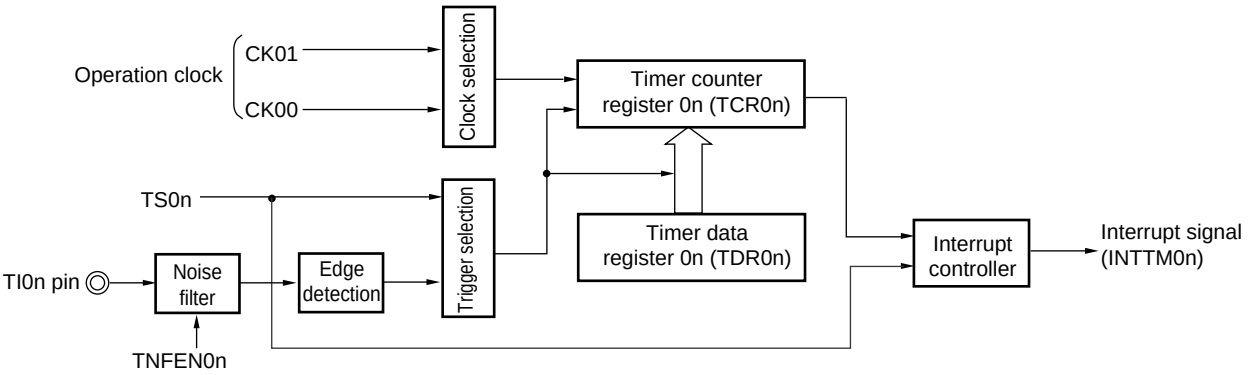
Timer count register 0n (TCR0n) operates as a down counter in the one-count mode.

When the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TE0n bit is set to 1 and the TI0n pin input valid edge detection wait status is set.

Timer count register 0n (TCR0n) starts operating upon TI0n pin input valid edge detection and loads the value of timer data register 0n (TDR0n). The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock. When TCR0n = 0000H, it outputs INTTM0n and stops counting with TCR0n = FFFFH until the next TI0n pin input valid edge is detected.

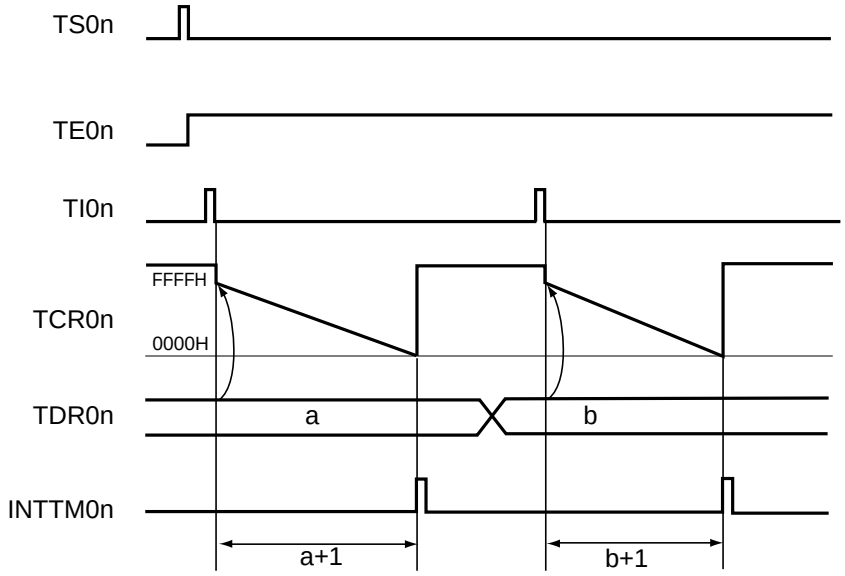
The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid from the next period.

Figure 6-60. Block Diagram of Operation as Delay Counter



Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-61. Example of Basic Timing of Operation as Delay Counter



Remarks **1.** n: Channel number

n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

2. TS0n: Bit n of timer channel start register 0 (TS0)

TE0n: Bit n of timer channel enable status register 0 (TE0)

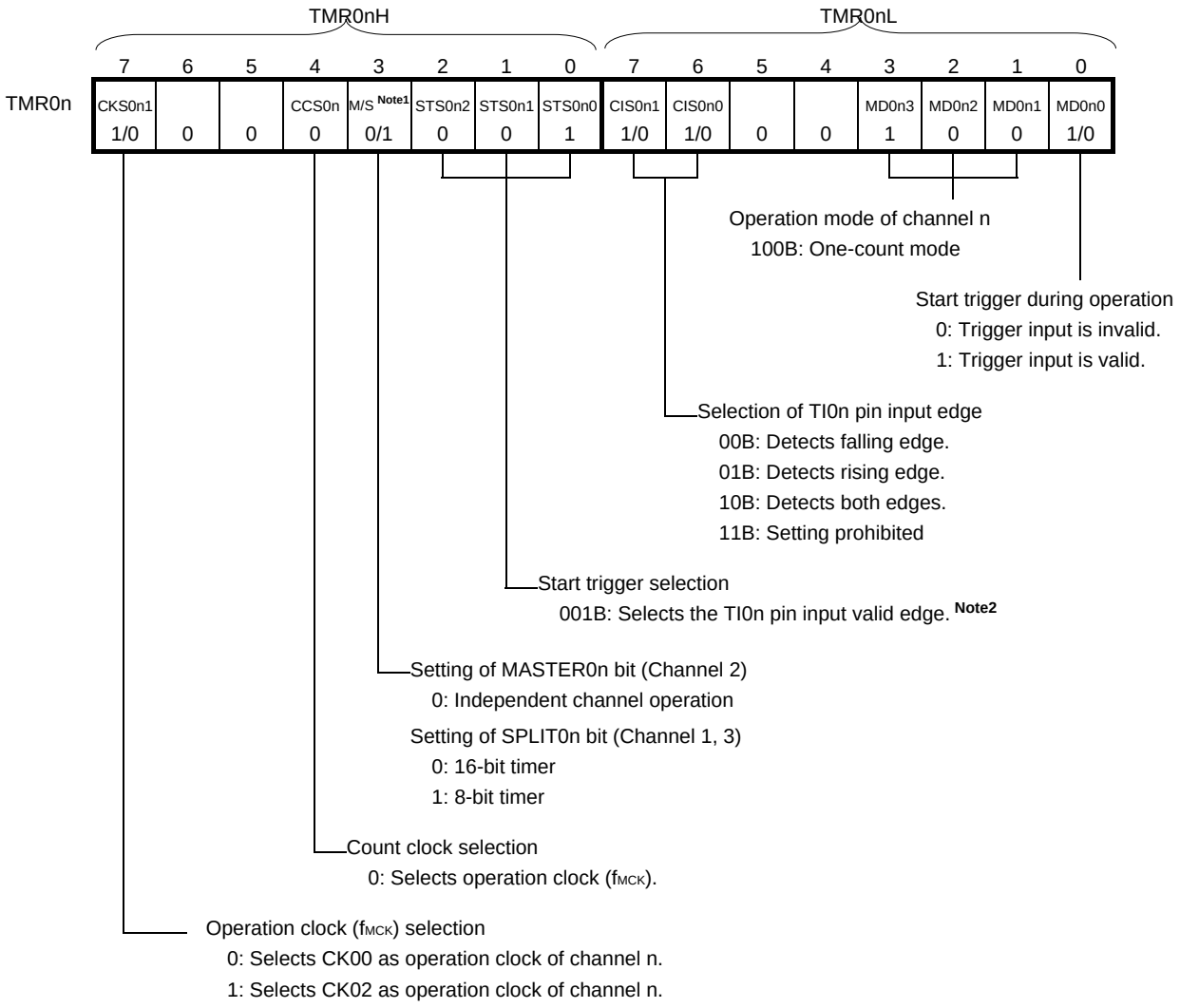
TI0n: TI0n pin input signal

TCR0n: Timer count register 0n (TCR0n)

TDR0n: Timer data register 0n (TDR0n)

Figure 6-62. Example of Set Contents of Registers to Delay Counter (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



(b) Timer output register 0 (TO0)

	Bit n
TO0	TO0n 0

0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOE0n 0

0: Stops the TO0n output operation by counting operation.

Notes 1. TMR02: MASTER0n bit
TMR01, TMR03: SPLIT0n bit
TMR00: 0 fixed

2. A software operation (TS0n = 1) can be used as a capture trigger, instead of using the TI0n pin input.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-62. Example of Set Contents of Registers to Delay Counter (2/2)

(d) Timer output level register 0 (TOL0)

	Bit n
TOL0	TOL0n 0

0: Setting is invalid because master channel output mode is set (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

	Bit n
TOM0	TOM0n 0

0: Sets master channel output mode.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

Figure 6-63. Procedure for Operating Delay Counter

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to SFR of the TAU is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1 (when the TAU0EN bit is 0, read/write operation is disabled).	Power-on status. Each channel stops operating. (Clock supply is started and writing to SFR of the TAU is enabled.)
	Sets timer clock select register 0 (TPS0). Determines operating clock (CK00 and CK01) for each channel.	
Channel default setting	Sets noise filter enable register 1 (NFEN1). Sets timer mode register 0n (TMR0n) (determines operation mode for each channel and selects the detection edge). Sets INTTM0n output delay in the timer data register 0n (TDR0n) (for the access procedure to the TDR0nH and TDR0nL registers, see **6.2.2 Timer data register 0n (TDR0n)**). Clears the target bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the target bit of the TOL0 register to 0. Clears the target bit of the timer output enable register (TOE0n) to 0.	Channel stops operating.
Operation start	Sets the target bit of TS0 register to 1. The target bit of TS0 register automatically returns to 0 because it is a trigger bit.	Target bit of TE0 register is set to 1, and the start trigger detection (the valid edge of the TI0n pin input is detected or the TS0n bit is set to 1) wait status is set.
	Count operation starts on detection of the next start trigger: - The TI0n pin input valid edge is detected. - The TS0n bit is set to 1 by software.	Value of the TDR0n register is loaded to the timer count register 0n (TCR0n), and count down operation starts.
During operation	The set value of the TDR0n register can be changed. The TCR0n register can always be read (for the access procedure to the TCR0nH and TCR0nL registers, see **6.2.1 Timer counter register 0n (TCR0n)**). The set values in the target bits of theTMR0n, TO0, TOE0, TOM0n, and TOL0n registers cannot be changed.	The counter (TCR0n) counts down. When TCR0n counts down to 0000H, INTTM0n is generated, and counting stops (which leaves TCR0n at FFFFH) until the next start trigger is detected (the valid edge of the TI0n pin input is detected or the TS0n bit is set to 1). After that, the above operation is repeated.
Operation stop	The target bit of TT0 register is set to 1. The target bit of TT0 register automatically returns to 0 because it is a trigger bit.	The target bit of TE0 register is cleared to 0, and count operation stops. The TCR0n register holds count value and stops.
TAU stop	Clears the TAU0EN bit of the PER0 register to 0.	Power-off status (Clock supply is stopped and SFR of the TAU is initialized.)

Operation is resumed.

Remark n: Channel number
n = 0, 1 (for 10-pin products); n = 0 to 3 (for 16-pin products)

6.9 Simultaneous Channel Operation Function of Timer Array Unit

6.9.1 Operation as one-shot pulse output

By using two channels as a set, a one-shot pulse having any delay (output delay time) can be generated from the signal input to the TI0n pin.

In addition, by setting TS0n to 1 by software, the count down can be started during the period of TE0n = 1.

The delay time and one-shot pulse width can be calculated by the following expressions.

Delay time = {Set value of TDR0n (master) + 2} × Count clock period
One-shot pulse width = {Set value of TDR0p (slave)} × Count clock period

Caution The TI0n pin input is sampled using the operating clock (f_{MCK}) selected with the CKS0n1 bit of timer mode register 0n (TMR0n), so an error of one cycle of the operating clock (f_{MCK}) occurs.

The master channel operates in the one-count mode and counts the delays. Timer count register 0n (TCR0n) of the master channel starts operating upon start trigger detection and loads the value of timer data register 0n (TDR0n).

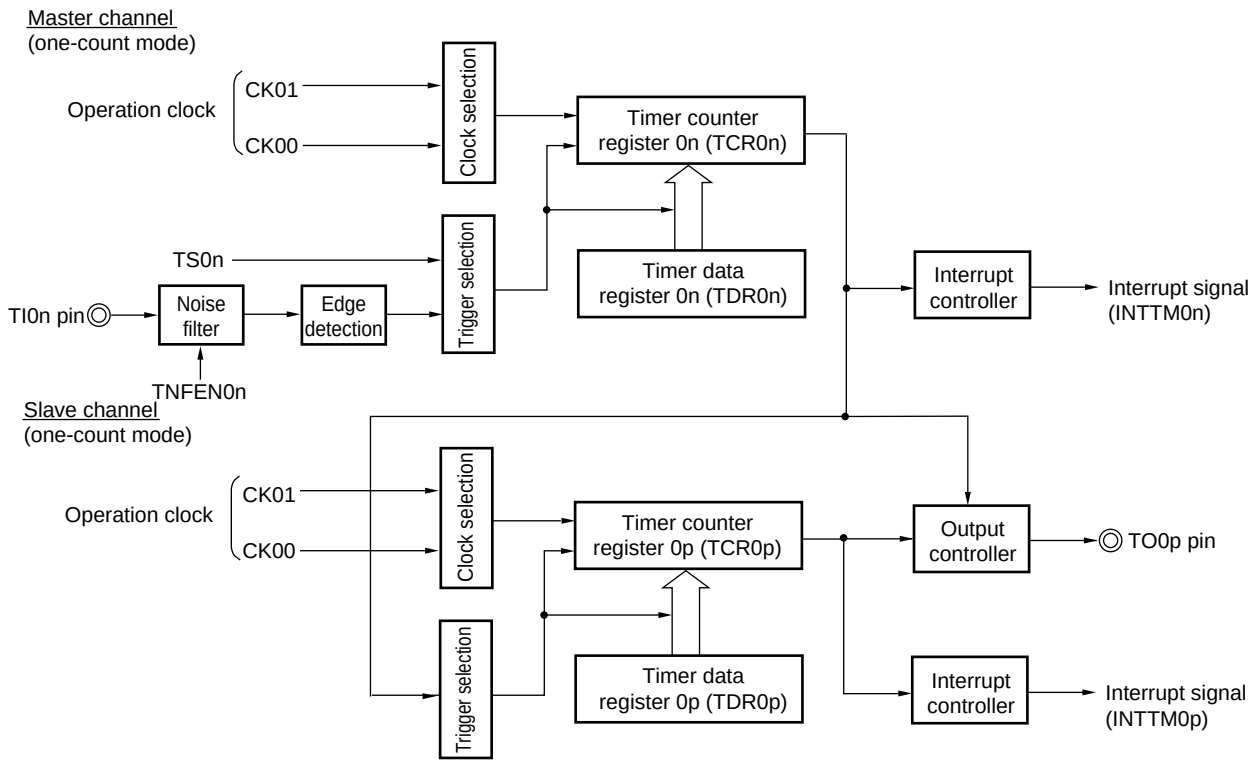
The TCR0n register counts down from the value of the TDR0n register it has loaded, in synchronization with the count clock (f_{TCLK}). When TCR0n = 0000H, it outputs INTTM0n and stops counting until the next start trigger is detected.

The slave channel operates in the one-count mode and counts the one-shot pulse width. The TCR0p register of the slave channel starts operation using INTTM0n of the master channel as a start trigger, and loads the value of the TDR0p register. The TCR0p register counts down from the value of The TDR0p register it has loaded, in synchronization with the count value (f_{TCLK}). When TCR0p = 0000H, it outputs INTTM0p and stops counting with TCR0p = FFFFH until the next start trigger (INTTM0n of the master channel) is detected. The output level of TO0p becomes active one count clock (f_{TCLK}) after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

Caution The timing of loading of timer data register 0n (TDR0n) of the master channel is different from that of the TDR0p register of the slave channel. If the TDR0n and TDR0p registers are rewritten during operation, therefore, an illegal waveform may be output. Rewrite the TDR0n register after INTTM0n is generated and the TDR0p register after INTTM0p is generated.

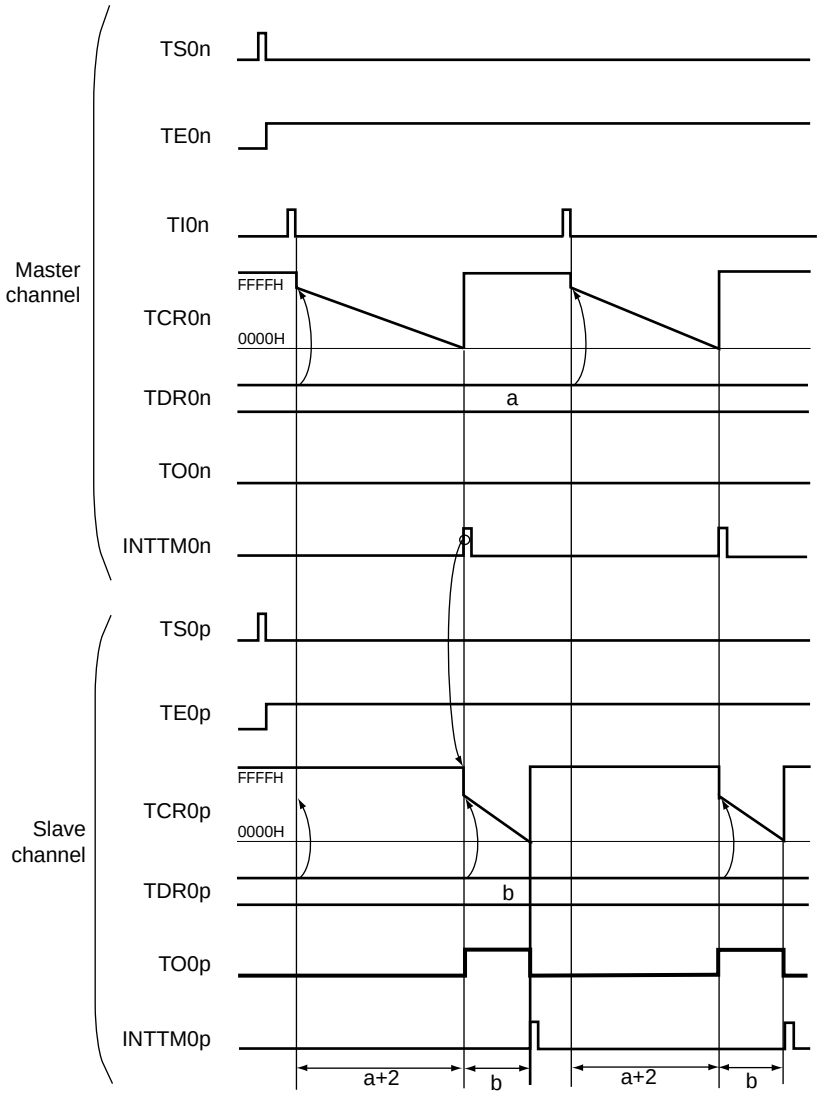
Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-64. Block Diagram of Operation for One-Shot Pulse Output



Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-65. Example of Basic Timing of Operation for One-Shot Pulse Output

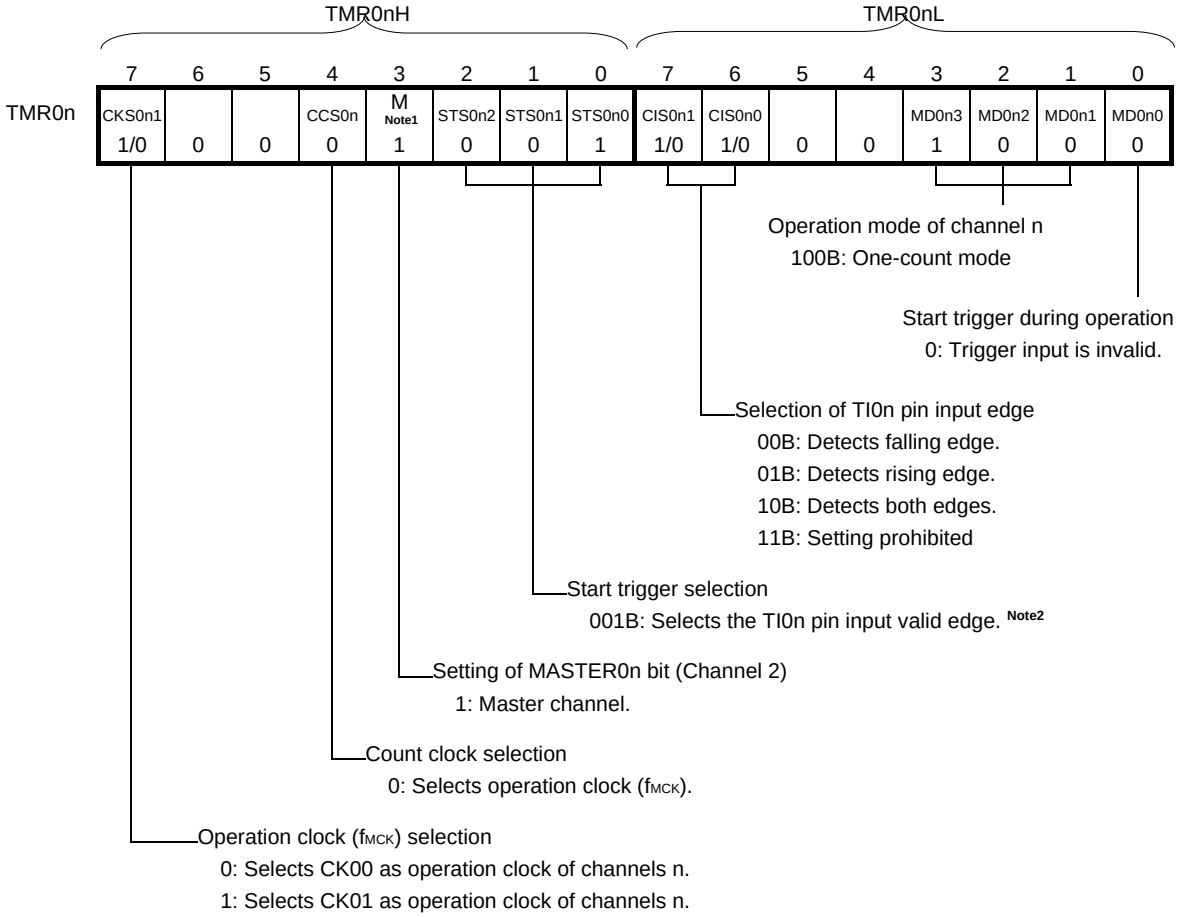


Remarks 1. n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

2. TS0n, TS0p: Bit n, p of timer channel start register 0 (TS0)
TE0n, TE0p: Bit n, p of timer channel enable status register 0 (TE0)
TI0n, TI0p: TI0n and TI0p pins input signal
TCR0n, TCR0p: Timer count registers 0n, 0p (TCR0n, TCR0p)
TDR0n, TDR0p: Timer data registers 0n, 0p (TDR0n, TDR0p)
TO0n, TO0p: TO0n and TO0p pins output signal

Figure 6-66. Example of Set Contents of Registers for One-Shot Pulse Output (Master Channel) (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Setting is invalid because master channel output mode is set (TOM0n = 0).

Notes 1. TMR02: MASTER02 bit
TMR00: 0 fixed

2. A software operation (TS0n = 1) can be used as a start trigger, instead of using the TI0n pin input.

Remark n: Master channel number (n = 0, 2)

Figure 6-66. Example of Set Contents of Registers for One-Shot Pulse Output (Master Channel) (2/2)

(e) Timer output mode register 0 (TOM0)

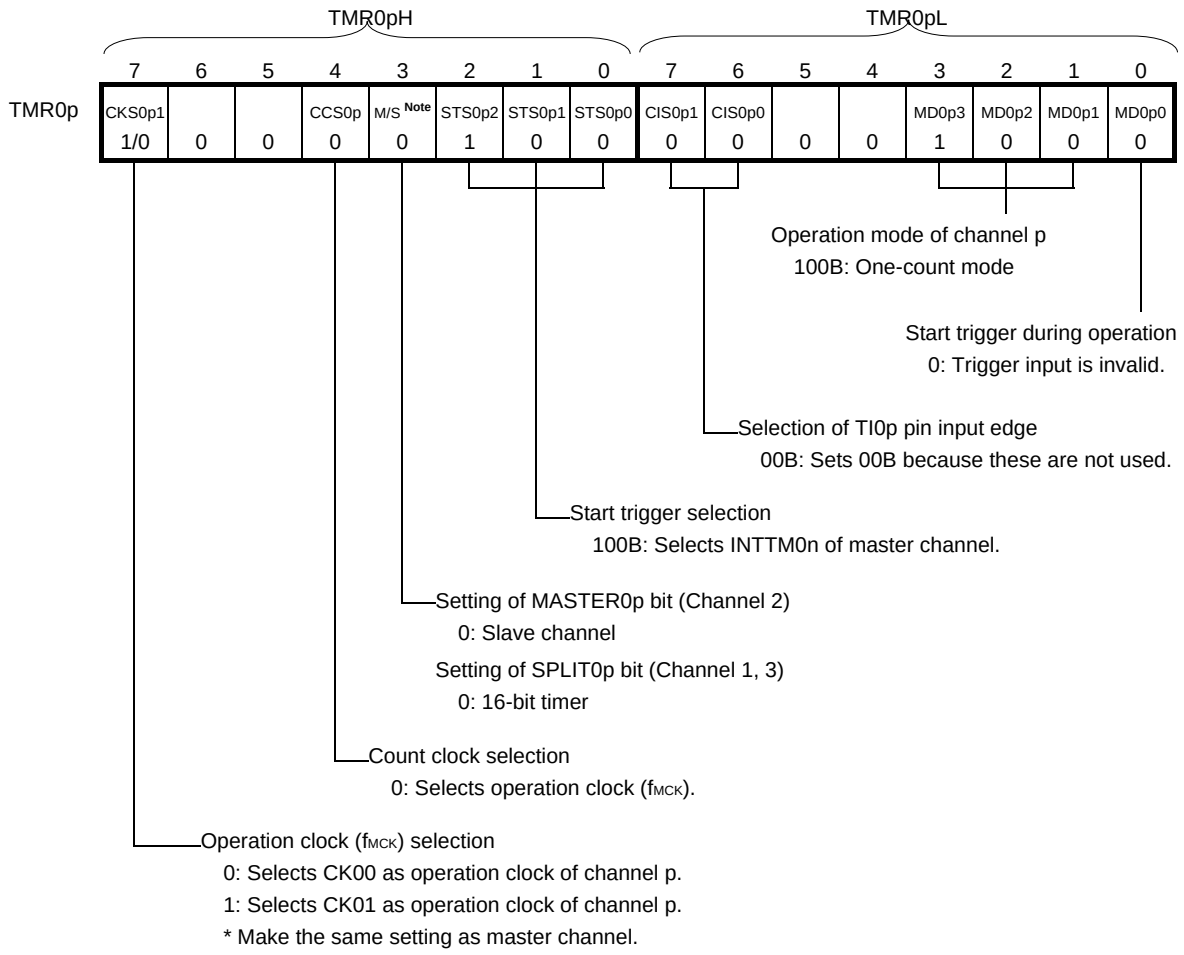
	Bit n
TOM0	TOM0n 0

0: Sets master channel output mode.

Remark n: Master channel number (n = 0, 2)

Figure 6-67. Example of Set Contents of Registers for One-Shot Pulse Output (Slave Channel) (1/2)

(a) Timer mode register 0p (TMR0pH, TMR0pL)



(b) Timer output register 0 (TO0)

	Bit p	
TO0	TO0p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p	
TOE0	TOE0p 1/0	0: Stops the TO0p output operation by counting operation (the level set in the TO0p bit is output from the TO0p pin). 1: Enables the TO0p output operation by counting operation (output from the TO0p pin is toggled).

(d) Timer output level register 0 (TOL0)

	Bit p	
TOL0	TOL0p 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

Note TMR02: MASTER0n bit
TMR01, TMR03: SPLIT0p bit

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-67. Example of Set Contents of Registers for One-Shot Pulse Output (Slave Channel) (2/2)

(e) Timer output mode register 0 (TOM0)

	Bit p
TOM0	TOM0p 1

1: Sets the slave channel output mode.

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-68. Procedure for Outputting One-Shot Pulse (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to SFR of the TAU is disabled.)
	Sets the TAU0EN bit of peripheral enable registers 0 (PER0) to 1 (when the TAU0EN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to SFR of the TAU is enabled.)
	Sets timer clock select register 0 (TPS0). Determines operating clock (CK00 and CK01) for each channel.	
Channel default setting	Sets timer mode registers 0n, 0p (TMR0n, TMR0p) (determines operation mode for each channel and selects the detection edge). Sets an output delay of the master channel in the timer data register 0n (TDR0n), and a pulse width of the slave channel in the timer data register 0p (TDR0p) (for the access procedure to the TDR0nH and TDR0nL registers, see **6.2.2 Timer data register 0n (TDR0n)**).	Channel stops operating.
	Sets master channel. Sets noise filter enable register 1 (NFEN1). Clears the target bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the target bit of the TOL0 register to 0. Clears the target bit of the timer output enable register 0 (TOE0) to 0. Sets slave channel. Sets the target bit of timer output mode register 0 (TOM0) to 1 (slave channel output mode). Sets the target bit of the TOL0 register. Sets the TO0p bit and determines default level of the TO0p output. Sets the TOE0p bit to 1 and enables operation of TO0p. →	The TO0p pin goes into Hi-Z state. (The port mode register is set to input mode.) TO0p does not change because channel stops operating (the TO0p pin is not affected even if the TO0p bit is modified).
	Clears the port register and port mode register to 0. (output mode is set) →	The level set in the TO0p bit is output from the TO0p pin.

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-68. Procedure for Outputting One-Shot Pulse (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOE0p bit of the slave channel to 1 to enable TO0p operation (only when operation is resumed). Sets the target bits of the TS0 register (master and slave) to 1 at the same time. The target bits of the TS0 register automatically return to 0 because they are trigger bits.	The target bits of the TE0 register are set to 1 and the master channel enters the TI0n pin input valid edge detection wait status.
	Count operation starts on detection of the next start triggers: - The TI0n pin input valid edge is detected. - The TS0n bit is set to 1 by software.	Value of the TDR0n register is loaded to the timer count register 0n (TCR0n) of the master channel, and count down operation starts.
During operation	Changes master channel setting. The TCR0n register can always be read (for the access procedure to the TCR0nH and TCR0nL registers, see **6.2.1 Timer counter register 0n (TCR0n)**). The set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. The set values in the target bits of theTDR0n, TO0, TOE0, TOM0, and TOL0 registers cannot be changed. Changes slave channel setting. The TCR0p register can always be read. The set values in the target bits of the TO0p, TOE0p, TOM0, and TOL0 registers can be changed. The set values of the TMR0p and TDR0p registers cannot be changed.	The master channel counter (TCR0n) performs count down operation. When the count value reaches TCR0n = 0000H, INTTM0n is generated, and the counter stops at TCR0n = FFFFH until the next start trigger is detected (the TI0n pin input valid edge is detected or TS0n bit is set to 1). The slave channel, triggered by INTTM0n of the master channel, loads the value of the TDR0p register to the TCR0p register, and the counter starts counting down. The output level of TO0p becomes active one cycle of the count clock (f_{TCLK}) after generation of INTTM0n from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped at TCR0p = FFFFH. After that, the above operation is repeated.
Operation stop	Sets the target bits of the TT0 register (master and slave) to 1 at the same time. The target bits of the TT0 register automatically return to 0 because they are trigger bits.	The target bits of the TE0 register are cleared to 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	Clears the TOE0p bit of slave channel to 0 and sets a value to the TO0p bit.	The level set in the TO0p bit is output from the TO0p pin.
TAU stop	To hold the TO0p pin output level Clears the TO0p bit to 0 after the value to be held (output latch) is set to the port register.	The TO0p pin output level is held by port function.
	Clears the TAU0EN bit of the PER0 register to 0.	Power-off status (Clock supply is stopped and SFR of the TAU is initialized.)

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

6.9.2 Two-channel input with one-shot pulse output function

By using signal input to two pins (TI0n and TI0p), a one-shot pulse having any delay pulse width can be generated. The two-channel input with one-shot pulse output function is provided only in the 16-pin products.

The output delay time and one-shot pulse width can be calculated by the following expressions.

Delay time = {Set value of TDR0n (master) + 2} × count clock period One-shot pulse active-level width = count clock period × ((10000H + TSR0p:OVF) + (capture value of TDR0p (slave) + 1)

Caution The TI0n and TI0p pin inputs are each sampled using the operating clock (f_{MCK}) selected with the CKS0n1 bit of the timer mode register (TMR0n), so an error of one cycle of the operating clock (f_{MCK}) per pin occurs.

The master channel should be operated in the one-count mode to start counting the delays (output delay time) upon detection of a valid edge of the master channel TI0n pin input used as the start trigger. Upon detection of a start trigger (valid edge of TI0n pin input), the master channel loads the value of timer data register 0n (TDR0n) to the timer count register 0n (TCR0n), and performs counting down in synchronization with the count clock (f_{TCLK}).

When TCR0n = 0000H, the master channel outputs INTTM0n and outputs the active level from the TO0p pin. It stops counting until the next start trigger is detected.

The slave channel should be operated in the capture mode to set the one-shot pulse to the inactive level upon detection of a valid edge of the slave channel TI0n pin input used as the end trigger. Upon detection of an end trigger (valid edge of TI0p pin input), the slave channel transfers (captures) the count value of the TCR0p register to the TDR0p register, and clears it to 0000H. Simultaneously, the slave channel outputs INTTM0p and the inactive level from the TO0p pin. Here, if the counter overflow has occurred, the OVF bit in the timer status register 0p (TSR0p) is set; if not, the OVF bit is cleared. After this, the same steps are repeated.

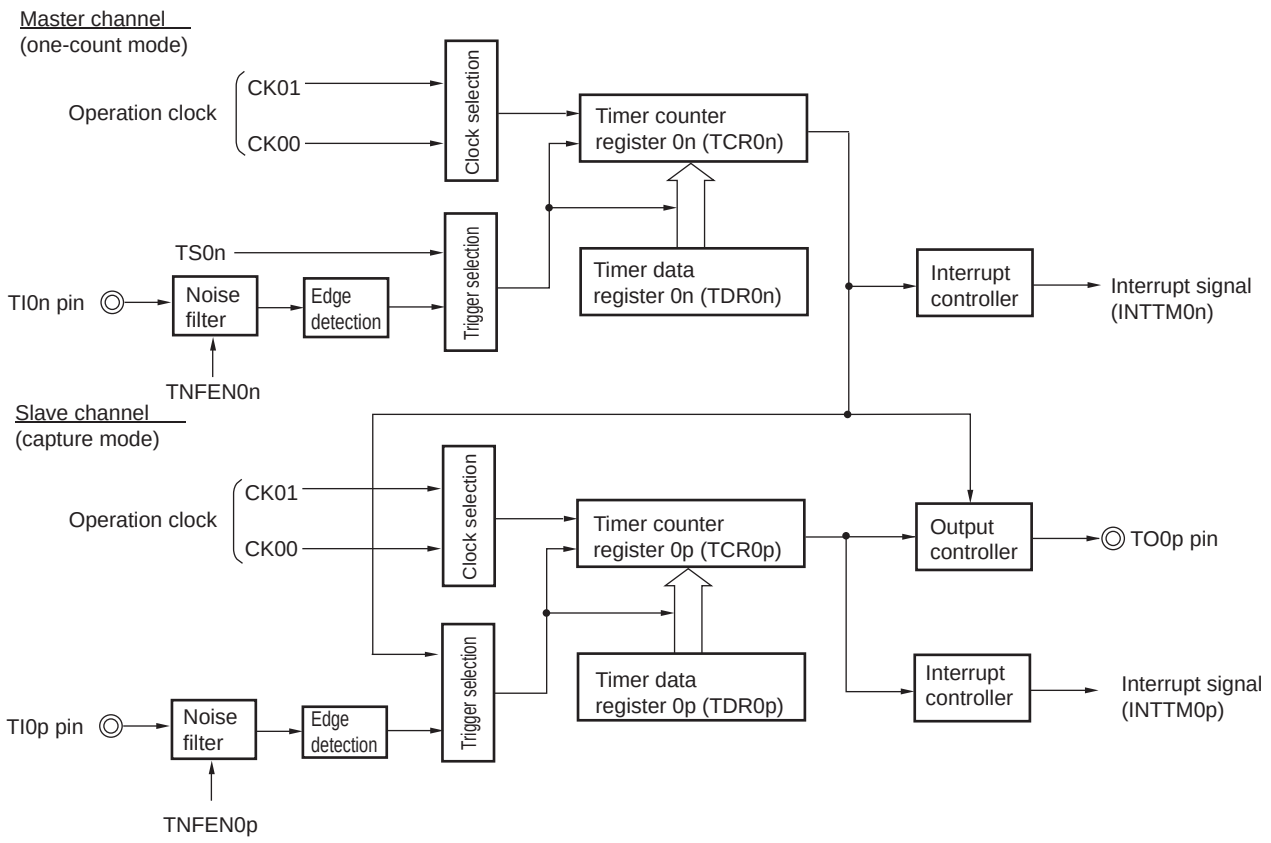
When the count value is captured to the TDR0p register, the OVF bit in the TSR0p register is updated depending on the overflow status during the active level period, which allows the overflow status of the captured value to be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSR0p register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Instead of using the TI0n pin input, the software operation (TS0n = 1) can be used as a start trigger for the master channel.

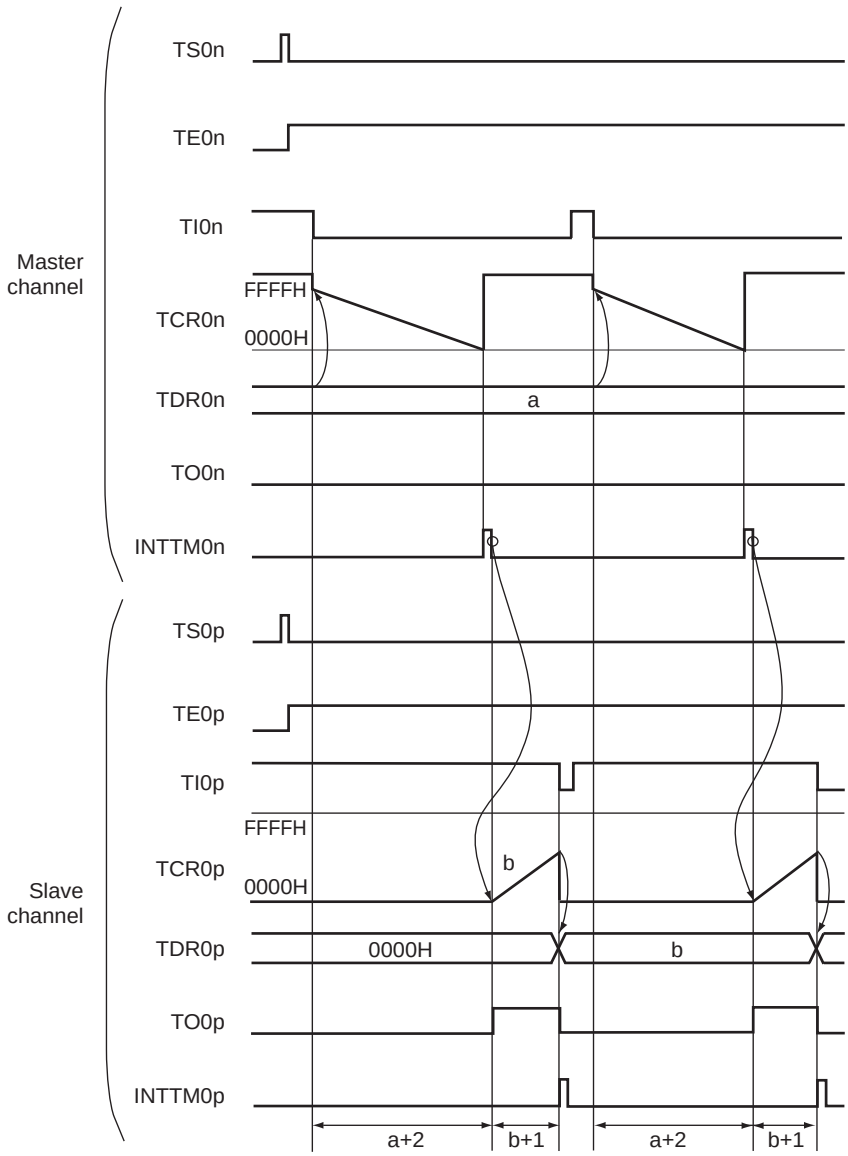
Remark n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 6-69. Block Diagram of Operation for Two-channel Input with One-shot Pulse Output Function



Remark n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 6-70. Example of Basic Timing of Operation for Two-channel Input with One-shot Pulse Output Function

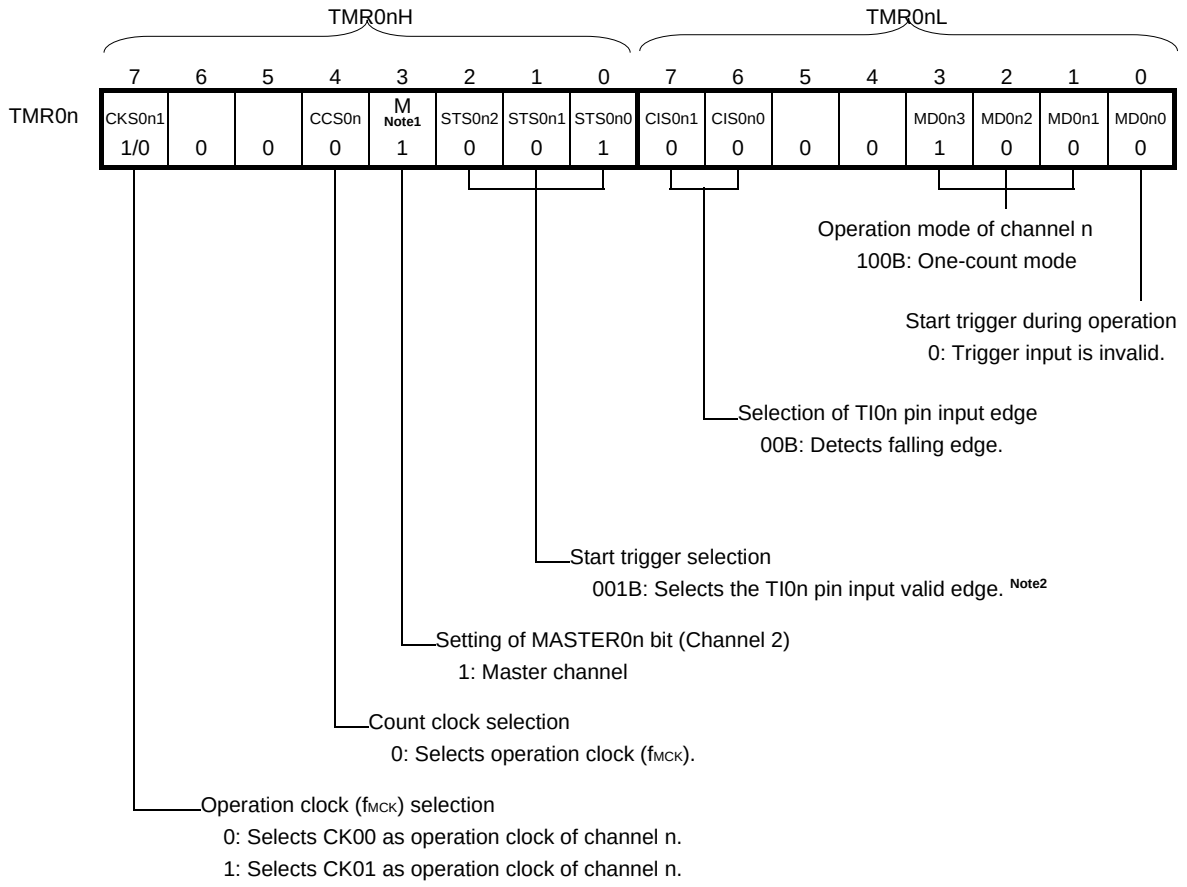


Remarks 1. n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

2. TS0n, TS0p: Bit n, p of timer channel start register 0 (TS0)
TE0n, TE0p: Bit n, p of timer channel enable status register 0 (TE0)
TI0n, TI0p: TI0n and TI0p pins input signal
TCR0n, TCR0p: Timer count registers 0n, 0p (TCR0n, TCR0p)
TDR0n, TDR0p: Timer data registers 0n, 0p (TDR0n, TDR0p)
TO0n, TO0p: TO0n and TO0p pins output signal

Figure 6-71. Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (Master Channel) (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Setting is invalid because master channel output mode is set (TOM0n = 0).

Notes 1. TMR02: MASTER02 bit
TMR00: 0 fixed

2. A software operation (TS0n = 1) can be used as a start trigger, instead of using the TI0n pin input.

Remark n: Master channel number (n = 0, 2)

Figure 6-71. Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (Master Channel) (2/2)

(e) Timer output mode register 0 (TOM0)

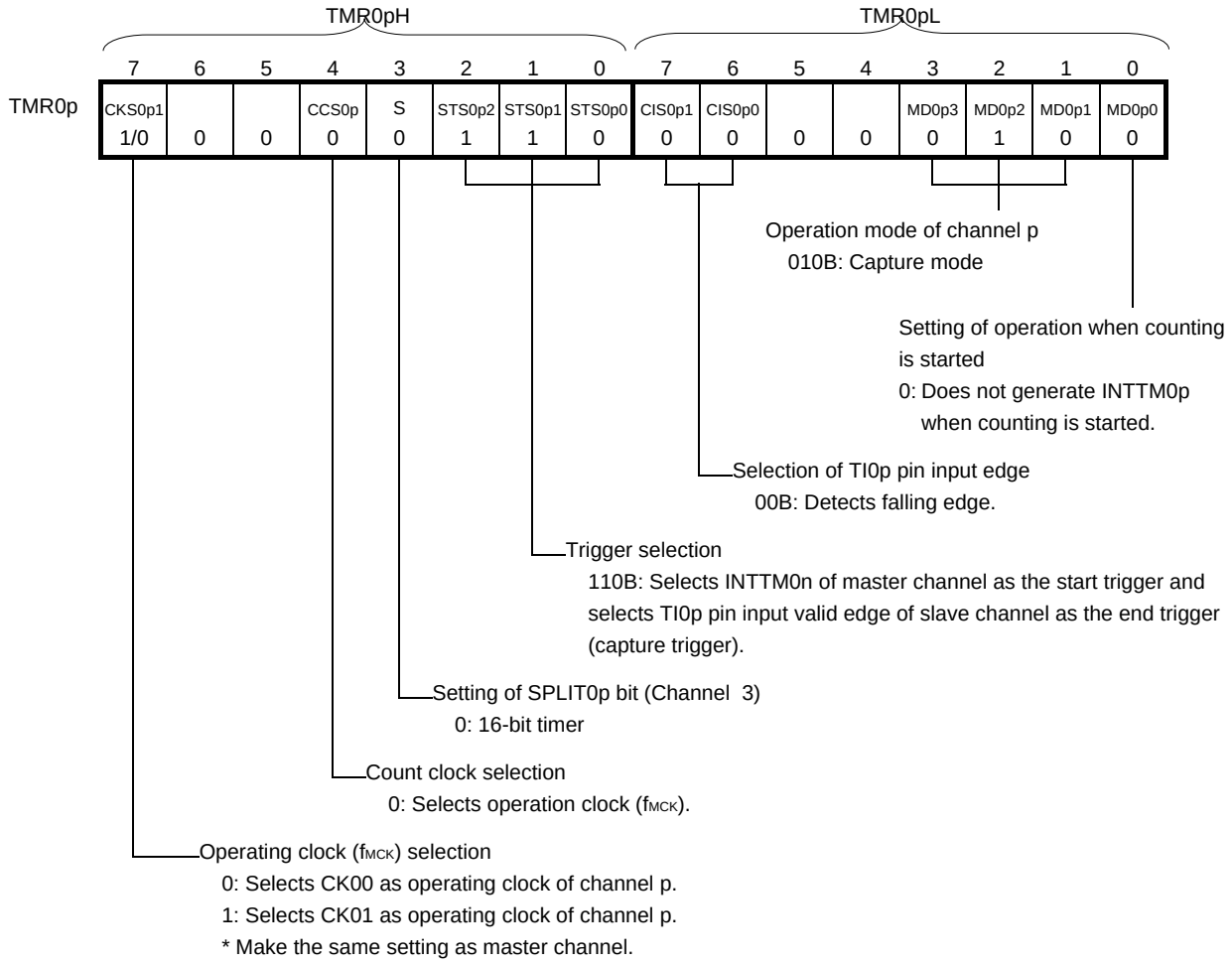
	Bit n
TOM0	TOM0n 0

0: Sets master channel output mode.

Remark n: Master channel number (n = 0, 2)

Figure 6-72. Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (Slave Channel) (1/2)

(a) Timer mode register 0p (TMR0pH, TMR0pL)



(b) Timer output register 0 (TO0)

	Bit p	
TO0	TO0p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p	
TOE0	TOE0p 1/0	0: Stops the TO0p output operation by counting operation (the level set in the TO0p bit is output from the TO0p pin). 1: Enables the TO0p output operation by counting operation (output from the TO0p pin is toggled).

(d) Timer output level register 0 (TOL0)

	Bit p	
TOL0	TOL0p 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 6-72. Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (Slave Channel) (2/2)

(e) Timer output mode register 0 (TOM0)

	Bit p
TOM0	TOM0p 1

1: Sets the slave channel output mode.

Remark p: Slave channel number (p = 3)

Figure 6-73. Procedure for Two-channel Input with One-shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to SFR of the TAU is disabled.)
	Sets the TAU0EN bit of peripheral enable registers 0 (PER0) to 1 (when the TAU0EN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to SFR of the TAU is enabled.)
	Sets timer clock select register 0 (TPS0). Determines operating clock (CK00 and CK01) for each channel.	
Channel default setting	Sets noise filter enable register 1 (NFEN1). Sets timer mode register 0n, 0p (TMR0n, TMR0p) (determines operation mode for each channel and selects the detection edge).	Channel stops operating.
	Sets master channel Sets delay (output delay time) to timer data register 0n (TDR0n) (for the access procedure to the TDR0nH and TDR0nL registers, see **6.2.2 Timer data register 0n (TDR0n)**). Clears the target bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the target bit of the TOL0 register to 0. Clears the target bit of the timer output enable register 0 (TOE0) to 0.	The TO0p pin goes into Hi-Z state. (The port mode register is set to input mode.)
	Sets slave channel. Sets the target bit of timer output mode register 0 (TOM0) to 1 (slave channel output mode). Sets the target bit of the TOL0 register. Sets the TO0p bit and determines default level of the TO0p output. Sets the TOE0p bit to 1 and enables operation of TO0p. →	TO0p does not change because channel stops operating. (The TO0p pin is not affected even if the TO0p bit is modified).
	Clears the port register and port mode register to 0. (output mode is set) →	The level set in the TO0p bit is output from the TO0p pin.

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 6-73. Procedure for Two-channel Input with One-shot Pulse Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOE0p bit of the slave channel to 1 to enable TO0p operation (only when operation is resumed). Sets the target bits of the TS0 register (master and slave) to 1 at the same time. The target bits of the TS0 register automatically return to 0 because they are trigger bits.	The target bits of the TE0 register are set to 1 and the master channel enters the TI0n pin input valid edge detection wait status.
	Count operation starts on detection of the next start triggers: - The TI0n pin input valid edge is detected. - The TS0n bit is set to 1 by software.	Value of the TDR0n register is loaded to the timer count register 0n (TCR0n) of the master channel, and count down operation starts.
During operation	Changes master channel setting. The TCR0n register can always be read (for the access procedure to the TCR0nH and TCR0nL registers, see **6.2.1 Timer counter register 0n (TCR0n)**). The set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. The set values in the target bits of the TDR0n, TO0, TOE0, TOM0, and TOL0 registers cannot be changed. Changes slave channel setting. The TDR0p register can always be read. The TCR0p register can always be read. The TSR0p register can always be read. The set values of only the CIS0p1 and CIS0p0 bits of the TMR0p register can be changed. The set values in the target bits of the TO0p, TOE0p, TOM0, and TOL0 registers cannot be changed.	The master channel counter (TCR0n) performs count down operation. When the count value reaches TCR0n = 0000H, INTTM0n is generated, and the counter stops at TCR0n = FFFFH until the next start trigger is detected (the TI0n pin input valid edge is detected or TS0n bit is set to 1). The slave channel, triggered by INTTM0p of the master channel, clears the timer counter register 0p (TCR0p) to 0000H. The counter (TCR0p) starts counting up from 0000H, and when the TI0n pin input valid edge is detected, the count value is transferred to the timer data register 0p (TDR0p) (capture) and TCR0p register is cleared to 0000H. At this time, INTTM0p is generated, which sets the TO0p output level to inactive After that, the above operation is repeated.
Operation stop	Sets the target bits of the TT0 register (master and slave) to 1 at the same time. The target bits of the TT0 register automatically return to 0 because they are trigger bits.	The target bits of the TE0 register are cleared to 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	Clears the TOE0p bit of slave channel to 0 and sets a value to the TO0p bit.	The level set in the TO0p bit is output from the TO0p pin.
TAU stop	To hold the TO0p pin output level Clears the TO0p bit to 0 after the value to be held (output latch) is set to the port register.	The TO0p pin output level is held by port function.
	Clears the TAU0EN bit of the PER0 register to 0.	Power-off status Clock supply is stopped and SFR of the TAU is initialized.

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n = 3)

6.9.3 Operation as PWM output function

Two channels can be used as a set to generate a pulse of any period and duty factor.

When channel 1 or 3 is used as an 8-bit timer (SPLIT0n = 1), only the lower 8-bit timer can be used as the slave channel for the PWM output function.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR0n (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDR0p (slave)}/{Set value of TDR0n (master) + 1} × 100
0% output: Set value of TDR0p (slave) = 0000H (00H in 8-bit timer mode)
100% output: Set value of TDR0p (slave) ≥ {Set value of TDR0n (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDR0p (slave) > (set value of TDR0n (master) + 1), the actually output PWM waveform has a 100% duty factor.

The master channel operates in the interval timer mode. If the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the interrupt request signal (INTTM0n) is output, the value set to timer data register 0n (TDR0n) is loaded to timer count register 0n (TCR0n), and the counter counts down in synchronization with the count clock (f_{TCLK}). When TCR0n reaches 0000H, INTTM0n is output, the value of the TDR0n register is loaded again to the TCR0n register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TT0n) of timer channel stop register 0 (TT0) is set to 1.

During the PWM output function operation, the period until the master channel counts down to 0000H is the PWM output (TO0p) cycle.

The slave channel operates in one-count mode. By using INTTM0n from the master channel as a start trigger, the TCR0p register loads the value of the TDR0p register and TCR0p counts down to 0000H. When TCR0p reaches 0000H, it outputs INTTM0p, and stops counting with TCR0p = FFFFH until the next start trigger (INTTM0n from the master channel) is generated.

During the PWM output function operation, the period until the slave channel counts down to 0000H is the PWM output (TO0p) duty.

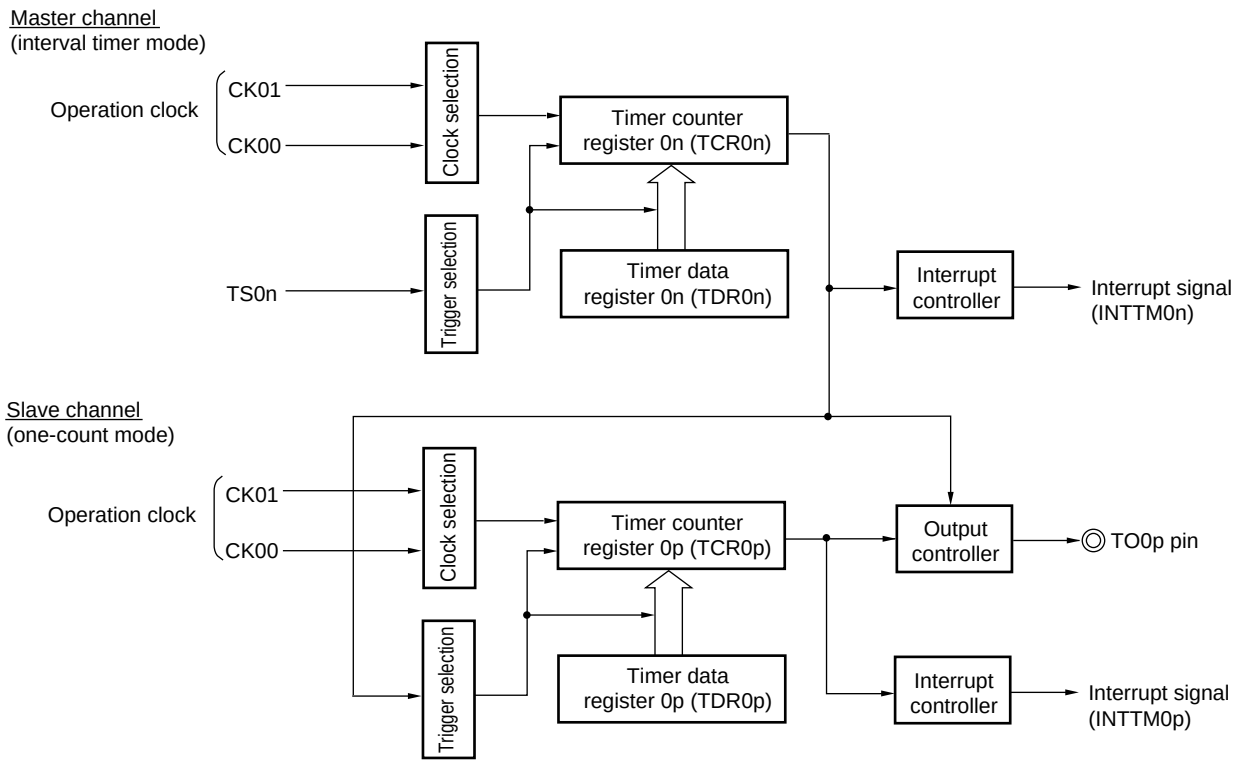
PWM output (TO0p) goes to the active level one count clock (f_{TCLK}) after the master channel generates INTTM0n and goes to the inactive level when the TCR0p register of the slave channel becomes 0000H.

Cautions 1. To rewrite both timer data register 0n (TDR0nH, TDR0nL) of the master channel and the TDR0pH and TDR0pL registers of the slave channel, a write access is necessary at least four times. The timing at which the values of the TDR0nH, TDR0nL, TDR0pH, and TDR0pL registers are loaded to the TCR0nH, TCR0nL, TCR0pH, and TCR0pL registers is upon generation of INTTM0n of the master channel. Thus, when rewriting is performed split before and after generation of INTTM0n of the master channel, the TO0p pin cannot output the expected waveform. To rewrite all of the TDR0nH, TDR0nL, TDR0pH, and TDR0pL registers, therefore, be sure to consecutively rewrite the four registers immediately after INTTM0n is generated from the master channel.

2. To use the PWM output function in 8-bit timer mode, set 00H in TDR0nH of the master channel and set the pulse period for the 8-bit timer. The TDR0nL register value should be set within the range from 00H to FEH (0% to 100% output).

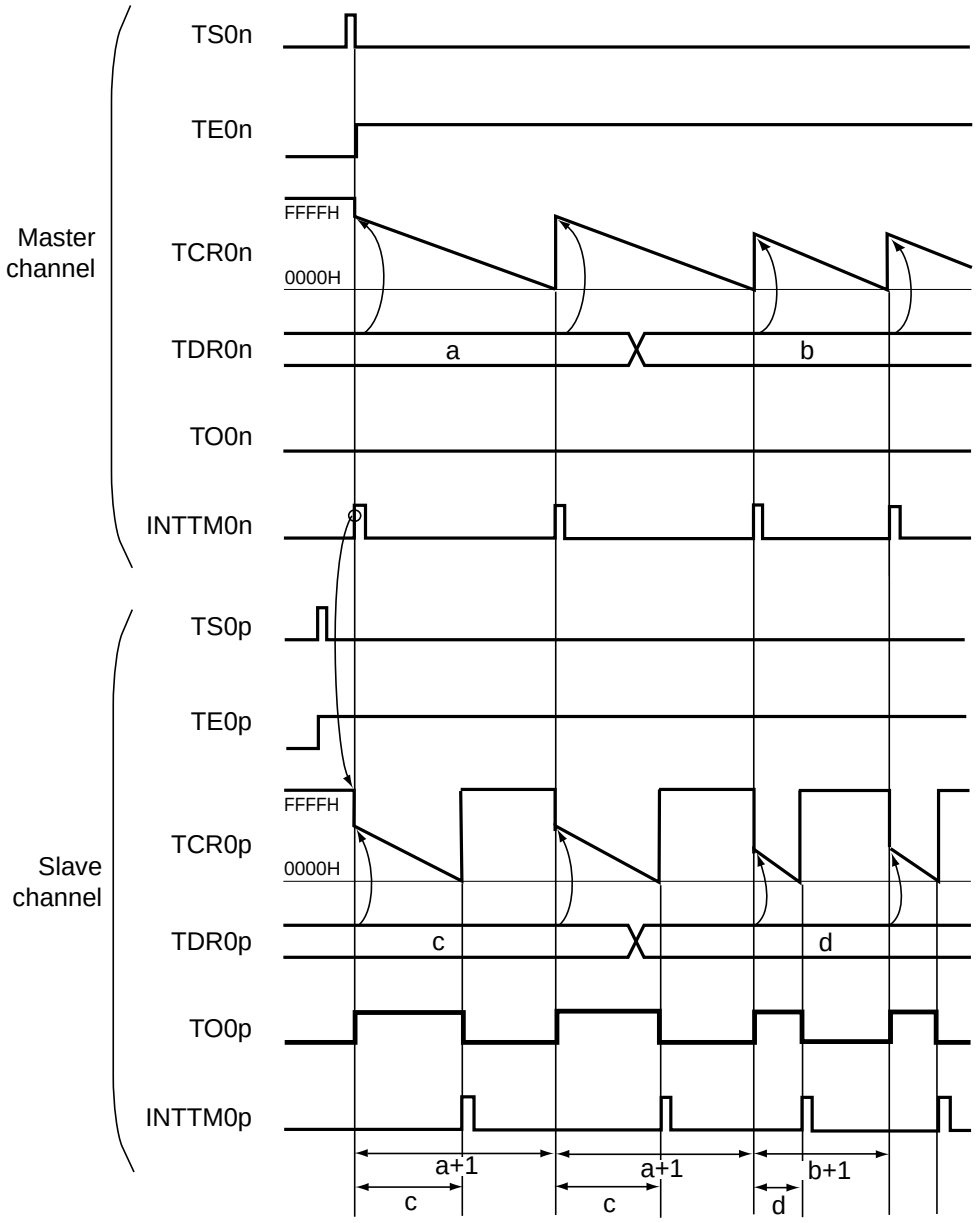
Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-74. Block Diagram of Operation as PWM Output Function



Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-75. Example of Basic Timing of Operation as PWM Output Function

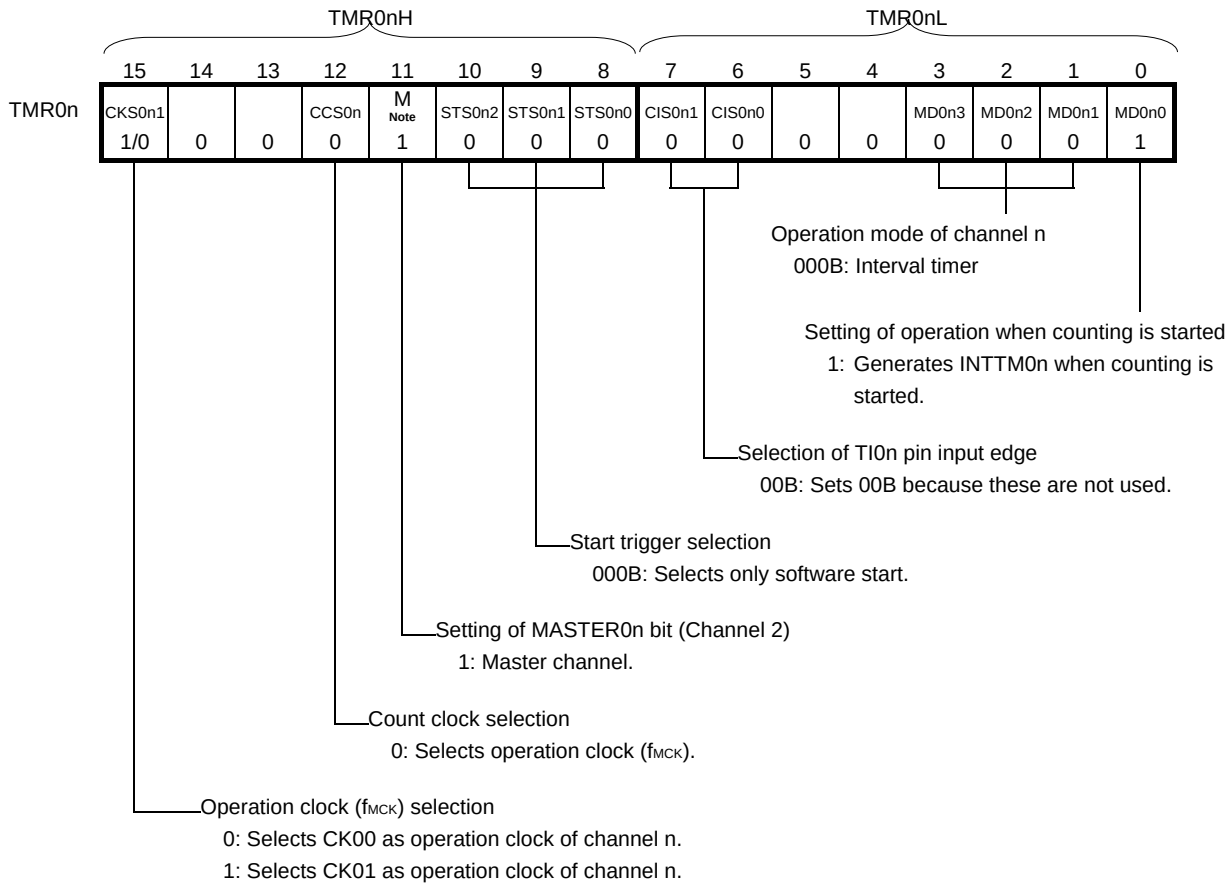


Remark 1. n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

2. TS0n, TS0p: Bit n, p of timer channel start register 0 (TS0)
TE0n, TE0p: Bit n, p of timer channel enable status register 0 (TE0)
TCR0n, TCR0p: Timer count registers 0n, 0p (TCR0n, TCR0p)
TDR0n, TDR0p: Timer data registers 0n, 0p (TDR0n, TDR0p)
TO0n, TO0p: TO0n and TO0p pins output signal

Figure 6-76. Example of Set Contents of Registers for PWM Output Function (Master Channel)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n 0	0: Outputs 0 from TO0n.

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n 0	0: Stops the TO0n output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n 0	0: Setting is invalid because master channel output mode is set (TOM0n = 0).

(e) Timer output mode register 0 (TOM0)

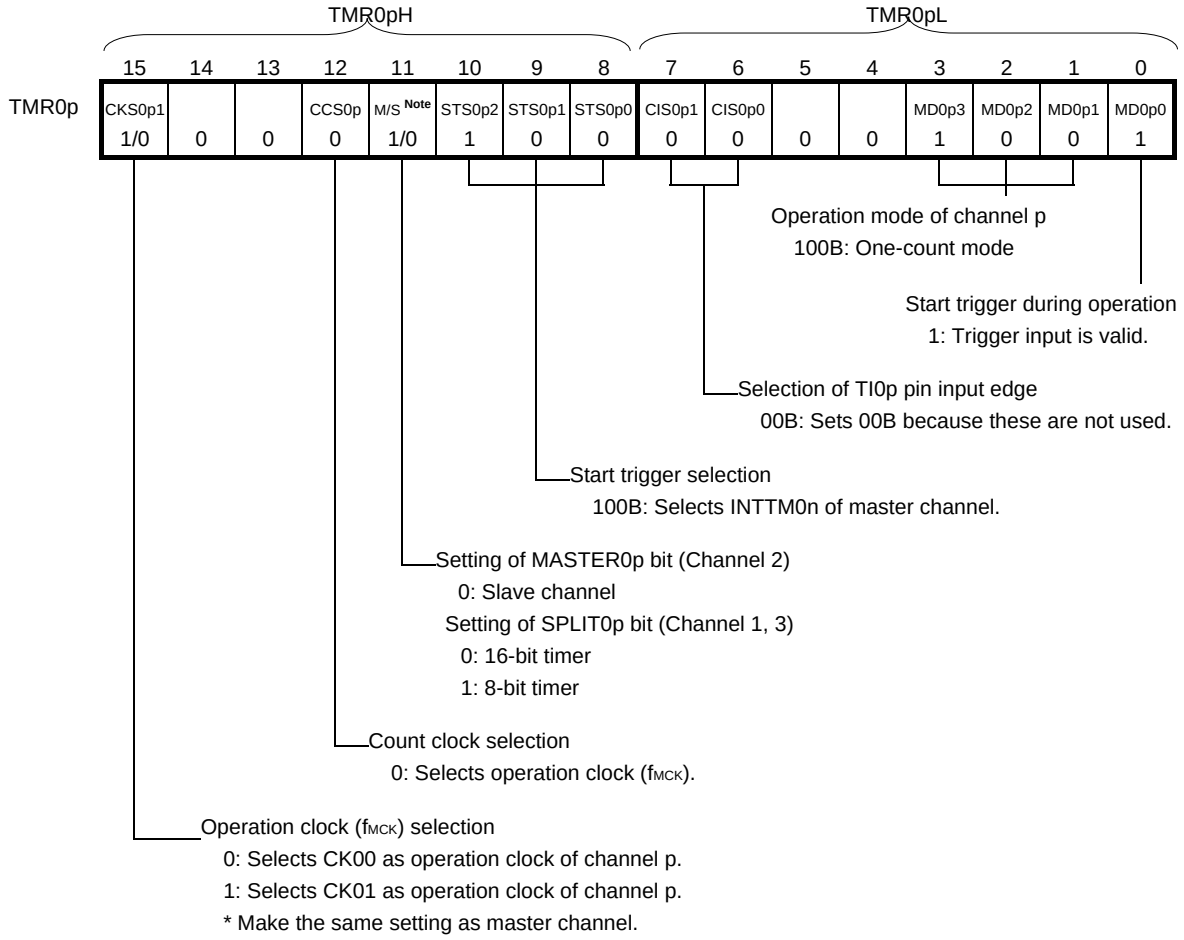
	Bit n	
TOM0	TOM0n 0	0: Sets master channel output mode.

Note TMR02: MASTER02 bit
TMR00: 0 fixed

Remark n: Master channel number (n = 0, 2)

Figure 6-77. Example of Set Contents of Registers for PWM Output Function (Slave Channel) (1/2)

(a) Timer mode register 0p (TMR0pH, TMR0pL)



(b) Timer output register 0 (TO0)

	Bit p	
TO0	TO0p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p	
TOE0	TOE0p 1/0	0: Stops the TO0p output operation by counting operation (the level set in the TO0p bit is output from the TO0p pin). 1: Enables the TO0p output operation by counting operation (output from the TO0p pin is toggled).

(d) Timer output level register 0 (TOL0)

	Bit p	
TOL0	TOL0p 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

Note TMR02: MASTER0p bit
TMR01, TMR03: SPLIT0p bit

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-77. Example of Set Contents of Registers for PWM Output Function (Slave Channel) (2/2)

(e) Timer output mode register 0 (TOM0)

	Bit p	
TOM0	TOM0p 1	1: Sets the slave channel output mode.

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-78. Procedure for Using PWM Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to SFR of the TAU is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1 (when the TAU0EN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to SFR of the TAU is enabled.)
	Sets timer clock select register 0 (TPS0). Determines operating clock (CK00 and CK01) for each channel.	
Channel default setting	Sets timer mode registers 0n, 0p (TMR0n, TMR0p) (determines operation mode for each channel). Sets an interval (period) value of the master channel and a duty factor of the slave channel in the timer data registers 0n and 0p (TDR0n and TDR0p) (for the access procedure to the TDR0nH and TDR0nL registers, see **6.2.2 Timer data register 0n (TDR0n)**).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets master channel Clears the target bit of the timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the target bit of the TOL0 register to 0. Clears the target bit of the timer output enable register 0 (TOE0) to 0. Sets slave channel Sets the target bit of the timer output mode register 0 (TOM0) to 1 (slave channel output mode). Sets the target bit of the TOL0 register. Sets the TO0p bit and determines default level of the TO0p output. Sets the TOE0p bit to 1 and enables operation of TO0p. →	The TO0p pin goes into Hi-Z state. (The port mode register is set to input mode.) TO0p does not change because channel stops operating (The TO0p pin is not affected even if the TO0p bit is modified).
	Clears the port register and port mode register to 0. (output mode is set.) →	The level set in the TO0p bit is output from the TO0p pin.

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-78. Procedure for Using PWM Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOE0p bit of the slave register to 1 and enables operation of TO0n (only when operation is resumed). Sets the target bits (master and slave) of the TS0 register to 1 at the same time. The target bits of the TS0 register automatically return to 0 because they are trigger bits.	The target bit of the TE0 register is set to 1, and the timer counter register 0n (TCR0n) of the master channel is loaded with the TDR0n register value and starts counting down.
During operation	Changes master channel setting. The set values of the TDR0n register can be changed after INTTM0n of the master channel is generated. The TCR0n register can always be read (for the access procedure to the TCR0nH and TCR0nL registers, see **6.2.1 Timer counter register 0n (TCR0n)**). The set values in the target bits of the TMR0n, TO0, TOE0, TOM0, and TOL0 registers cannot be changed. Changes slave channel setting. The set values of the TDR0p register can be changed after INTTM0n of the master channel is generated. The TCR0p register can always be read. The set values in the target bits of the TO0, TOE0, and TOL0 registers can be changed. The set values in the target bits of the TMR0p and TOM0 registers cannot be changed.	The timer counter register 0n (TCR0n) of the master channel performs count down operation. When the count value reaches TCR0n = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to the TCR0n register, and the counter starts counting down again. At the slave channel, the value of the TDR0p register is loaded to the TCR0p register, triggered by INTTM0n of the master channel, and the counter starts counting down. The output level of TO0p becomes active one count clock (f_{TCLK}) after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped with TCR0p = FFFFH. After that, the above operation is repeated.
Operation stop	Sets the target bits of the TT0 registers (master and slave) to 1 at the same time. The target bits of the TT0 registers automatically return to 0 because they are trigger bits.	The target bits of the TE0 register are cleared to 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	Clears the TOE0p bit of slave channel to 0 and sets a value to the TO0p bit.	The level set in the TO0p bit is output from the TO0p pin.
TAU stop	To hold the TO0p pin output level Clears the TO0p bit to 0 after the value to be held (output latch) is set to the port register.	The TO0p pin output level is held by port function.
	Clears the TAU0EN bit of the PER0 register to 0.	Power-off status (Clock supply is stopped and SFR of the TAU is initialized.)

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

6.9.4 Operation as multiple PWM output function

By extending the PWM output function and using multiple slave channels, many PWM waveforms with different duty values can be output. The multiple PWM output function is provided only in the 16-pin products.

When channel 1 or 3 is used as an 8-bit timer (SPLIT0n = 1), only the lower 8-bit timer can be used as the slave channel for the PWM output function.

For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDR0n (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDR0p (slave 1)}/{Set value of TDR0n (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDR0q (slave 2)}/{Set value of TDR0n (master) + 1} × 100

Remark Although the duty factor exceeds 100% if the set value of TDR0p (slave 1) > {set value of TDR0n (master) + 1} or if the {set value of TDR0q (slave 2)} > {set value of TDR0n (master) + 1}, the actually output PWM waveform has a 100% duty factor.

The master channel counts the pulse periods. When operated in interval timer mode, it loads the TDR0n value to the TCR0n register to start counting down.

The slave channel 1 counts the duty factor, and outputs any PWM waveform from the TO0p pin. When operated in one-count mode, it loads the TDR0p register value to the TCR0p register using INTTM0n from the master channel as a start trigger, and performs count down operation until TCR0p reaches 0000H. When TCR0p = 0000H, TCR0p outputs INTTM0p and stops counting with TCR0p = FFFFH until the next start trigger (INTTM0n from the master channel) has been input.

In the same way as the slave channel 1, the slave channel 2 counts the duty factor, and outputs a desired PWM waveform from the TO0q pin. When operated in one-count mode, the counter loads the TDR0q register value to the TCR0q register using INTTM0n from the master channel as a start trigger, and performs counting down until TCR0q reaches 0000H. When TCR0q = 0000H, the TCR0q register outputs INTTM0q and stops counting with TCR0q = FFFFH until the next start trigger (INTTM0n from the master channel) has been input.

The PWM output level (TO0p or TO0q) becomes active one count clock (f_{TCLK}) after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H or TCR0q = 0000H.

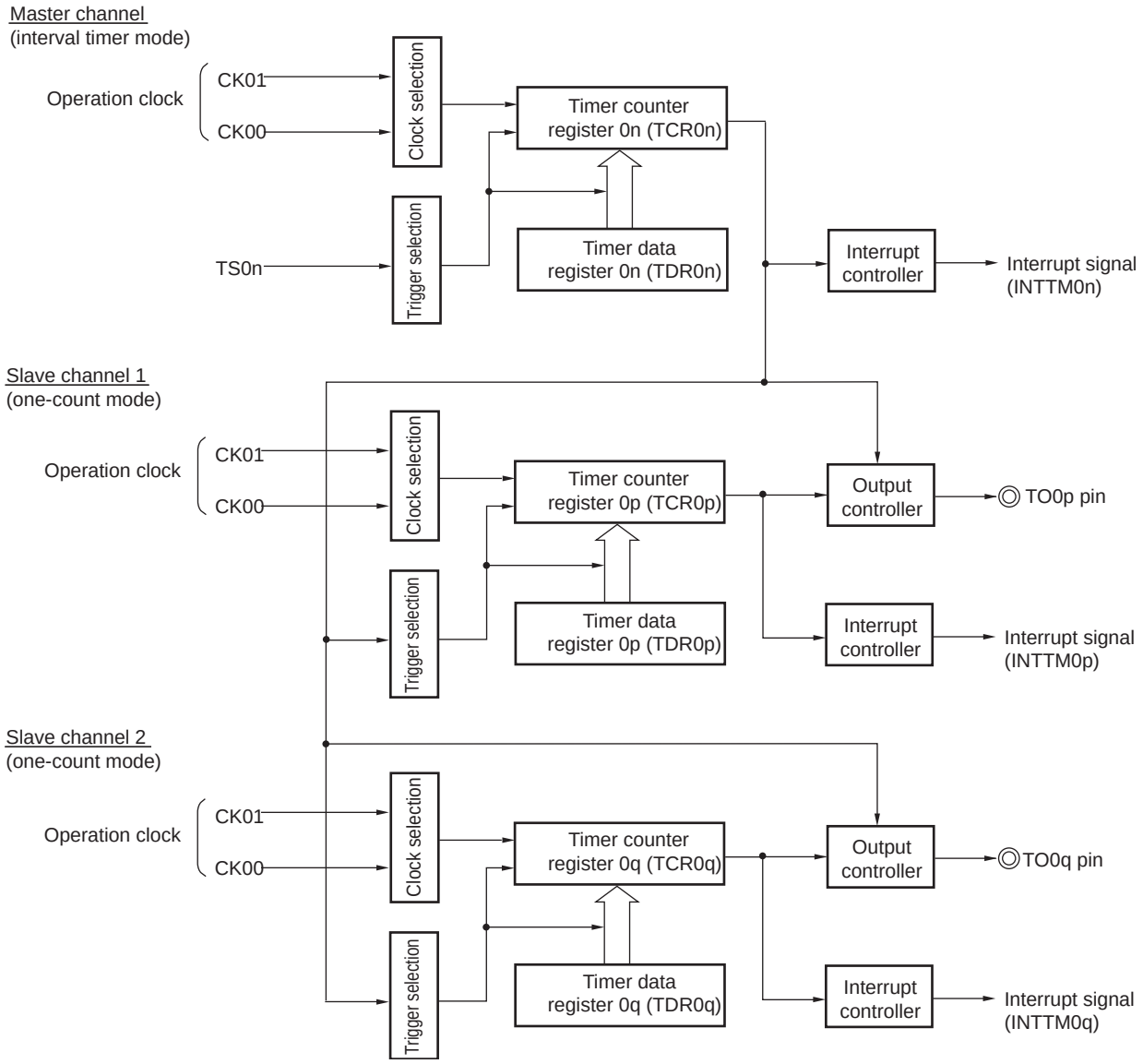
When channel 0 is used as the master channel as above, up to three types of PWM signals can be output at the same time.

Cautions 1. To rewrite both timer data register 0n (TDR0nH, TDR0nL) of the master channel and the TDR0pH and TDR0pL registers of the slave channel, write access is necessary at least four times. Since the values of the TDR0nH, TDR0nL, TDR0pH, and TDR0pL registers are loaded to the TCR0nH, TCR0nL, TCR0pH, and TCR0pL registers after INTTM0n is generated from the master channel, if rewriting is performed separately before and after generation of INTTM0n from the master channel, the TO0p pin cannot output the expected waveform. To rewrite all of the TDR0nH, TDR0nL, TDR0pH, and TDR0pL registers, be sure to consecutively rewrite the four registers immediately after INTTM0n is generated from the master channel.

2. To use the multiple PWM output function in 8-bit timer mode, set 00H in TDR0nH of the master channel and set the pulse period for the 8-bit timer. The TDR0nL register value should be set within the range from 00H to FEH (0% to 100% output).

Remark n: Master channel number (n = 0)
p: Channel number of slave channel 1, q: Channel number of slave channel 2
n < p < q ≤ 3 (Where p and q are consecutive integers greater than n)

Figure 6-79. Block Diagram of Operation as Multiple PWM Output Function (Output Two Types of PWMs)

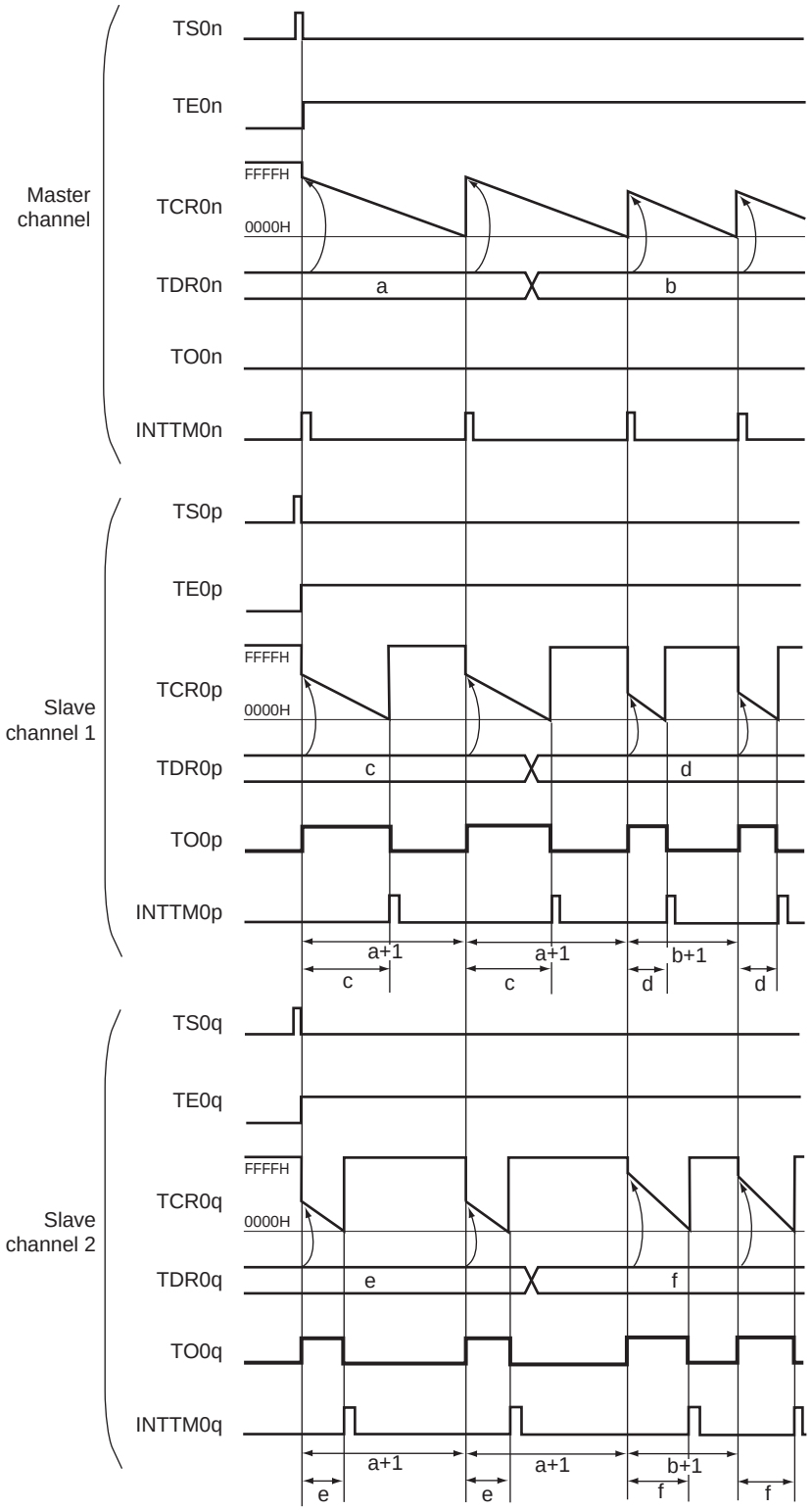


Remark n: Channel number (n = 0)

p: Channel number of slave channel 1, q: Channel number of slave channel 2

$n < p < q \leq 3$ (Where p and q are consecutive integers greater than n)

Figure 6-80. Example of Basic Timing of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



(**Remarks** are listed on the next page.)

Remarks 1. n: Channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 3 (Where p and q are consecutive integers greater than n)

2. TS0n, TS0p, TS0q: Bit n, p, q of timer channel start register 0 (TS0)
TE0n, TE0p, TE0q: Bit n, p, q of timer channel enable status register 0 (TE0)
TCR0n, TCR0p, TCR0q: Timer count registers 0n, 0p, 0q (TCR0n, TCR0p, TCR0q)
TDR0n, TDR0p, TDR0q: Timer data registers 0n, 0p, 0q (TDR0n, TDR0p, TDR0q)
TO0n, TO0p, TO0q: TO0n, TO0p, and TO0q pins output signal

Figure 6-81. Example of Set Contents of Registers for Multiple PWM Output Function (Master Channel) (1/2)

(a) Timer mode register 00 (TMR00)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR00	CKS001			CCS00		STS002	STS001	STS000	CIS001	CIS000			MD003	MD002	MD001	MD000
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel 0
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTM00 when counting is started.

Selection of TI00 pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
0: Selects CK00 as operation clock of channel 0.
1: Selects CK01 as operation clock of channel 0.

Figure 6-81. Example of Set Contents of Registers for Multiple PWM Output Function (Master Channel) (2/2)

(b) Timer output register 0 (TO0)

	Bit 0	
TO0	TO00 0	0: Outputs 0 from TO00.

(c) Timer output enable register 0 (TOE0)

	Bit 0	
TOE0	TOE00 0	0: Stops the TO00 output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit 0	
TOL0	TOL00 0	0: Setting is invalid because master channel output mode is set (TOM00 = 0).

(e) Timer output mode register 0 (TOM0)

	Bit 0	
TOM0	TOM00 0	0: Sets master channel output mode.

Figure 6-82. Example of Set Contents of Registers for Multiple PWM Output Function (Slave Channel) (Output Two Types of PWMs) (1/2)

(a) Timer mode register 0p, 0q (TMR0p, TMR0q)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0p	CKS0p1			CCS0p	M/S Note	STS0p2	STS0p1	STS0p0	CIS0p1	CIS0p0			MD0p3	MD0p2	MD0p1	MD0p0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1
TMR0q	CKS0q1			CCS0q	M/S Note	STS0q2	STS0q1	STS0q0	CIS0q1	CIS0q0			MD0q3	MD0q2	MD0q1	MD0q0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TI0p and TI0q pins input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTM00 of master channel.

Setting of MASTER0p and MASTER0q bits (Channel 2)
0: Slave channel
Setting of SPLIT0p and SPLIT0q bits (Channel 1, 3)
0: 16-bit timer
1: 8-bit timer

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
0: Selects CK00 as operation clock of channel p, q.
1: Selects CK01 as operation clock of channel p, q.
* Make the same setting as master channel.

Note TMR02: MASTER0p and MASTER0q bits
TMR01, TMR03: SPLIT0p and SPLIT0q bits

Remark p: Channel number of slave channel 1, q: Channel number of slave channel 2
$0 < p < q \leq 3$ (Where p and q are consecutive integers greater than 0)

Figure 6-82. Example of Set Contents of Registers for Multiple PWM Output Function (Slave Channel) (Output Two Types of PWMs) (2/2)

(b) Timer output register 0 (TO0)

	Bit q	Bit p	
TO0	TO0q 1/0	TO0p 1/0	0: Outputs 0 from TO0p or TO0q. 1: Outputs 1 from TO0p or TO0q.

(c) Timer output enable register 0 (TOE0)

	Bit q	Bit p	
TOE0	TOE0q 1/0	TOE0p 1/0	0: Stops the TO0p or TO0q output operation by counting operation (the level set in the TO0p or TO0q bit is output from the TO0p or TO0q pin). 1: Enables the TO0p or TO0q output operation by counting operation (output from the TO0p or TO0q pin is toggled).

(d) Timer output level register 0 (TOL0)

	Bit q	Bit p	
TOL0	TOL0q 1/0	TOL0p 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit q	Bit p	
TOM0	TOM0q 1	TOM0p 1	1: Sets the slave channel output mode.

Remark p: Channel number of slave channel 1, q: Channel number of slave channel 2
$0 < p < q \leq 3$ (Where p and q are consecutive integers greater than 0)

Figure 6-83. Procedure for Using Multiple PWM Output Function (Output Two Types of PWMs) (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to SFR of the TAU is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1 (when the TAU0EN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to SFR of the TAU is enabled.)
	Sets timer clock select register 0 (TPS0). Determines operating clock (CK00 and CK01) for each channel.	
Channel default setting	Sets timer mode registers 00, 0p, 0q (TMR00, TMR0p, TMR0q) (determines operation mode for each channels). Sets an interval (period) value of the master channel and a duty factor of the slave channels to the timer data registers 00, 0p, and 0q (TDR00, TDR0p, and TDR0q) (for the access procedure to the TDR0nH and TDR0nL registers, see **6.2.2 Timer data register 0n (TDR0n)**).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets master channel. Clears the target bit of the timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the target bit of the TOL0 register to 0. Clears the target bit of the timer output enable register 0 (TOE0) to 0. Sets slave channel. Sets the target bit of the timer output mode register 0 (TOM0) to 1 (slave channel output mode). Sets the target bit of the TOL0 register. Sets the TO0p and TO0q bits and determines default level of the TO0p and TO0q outputs. Sets the TOE0p and TOE0q bits to 1 and enables TO0p and TO0q outputs based on count operation. →	The TO0p and TO0q pins go into Hi-Z state. (The port mode register is in output mode.) TO0p and TO0q do not change because channels stop operating. (The TO0p pin is not affected even if the TO0p or TO0q bit is modified).
	Clears the port register and port mode register to 0. (output mode is set.) →	The levels set in the TO0p and TO0q bits are output from the TO0p and TO0q pins.

Remark p: Channel number of slave channel 1, q: Channel number of slave channel 2
$n < p < q \leq 3$ (Where p and q are a consecutive integer greater than n)

Figure 6-83. Procedure for Using Multiple PWM Output Function (Output Two Types of PWMs) (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOE0p and TOE0q bits of the slave register to 1 and enables the TO0p and TO0q outputs by the count operation (only when resuming operation). Sets the target bits of the TS0 register (master and slave) to 1 at the same time. The target bits of TS0 register automatically return to 0 because they are trigger bits.	The target bit of the TE0 register is set to 1, and the timer counter register 00 (TCR00) of the master channel is loaded with the TDR00 register value and starts counting down.
During operation	Changes master channel setting. The set values of the TDR00 register can be changed after INTTM00 of the master channel is generated. The TCR00 register can always be read (for the access procedure to the TCR00H and TCR00L registers, see **6.2.1 Timer counter register 0n (TCR0n)**). The set values in the target bits of the TMR00, TO0, TOE0, TOM0, and TOL0 registers cannot be changed. Changes slave channel setting. The set values of the TDR0p and TDR0q registers can be changed after INTTM00 of the master channel is generated. The TCR0p and TCR0q register can always be read. The set values in the target bits of the TO0, TOE0, and TOL0 registers can be changed. The set values in the target bits of the TMR0p, TMR0q, and TOM0 registers cannot be changed.	The timer counter register 00 (TCR00) of the master channel performs count down operation. When the count value reaches TCR00 = 0000H, INTTM00 is generated. At the same time, the value of the TDR00 register is loaded to the TCR00 register, and the counter starts counting down again. At the slave channel, the values of the TDR0p and TDR0q registers are loaded to the TCR0p and TCR0q registers, triggered by INTTM00 of the master channel, and the counter starts counting down. The output levels of TO0p and TO0q become active one count clock (f_{TCLK}) after generation of the INTTM00 output from the master channel. They become inactive when TCR0p = 0000H and TCR0q = 0000H, and the counting operation is stopped at TCR0p = FFFFH and TCR0q = FFFFH. After that, the above operation is repeated.
Operation stop	Sets the target bits (master and slave) of the TT0 register to 1 at the same time. The target bits of the TT0 register automatically return to 0 because they are trigger bits.	The target bits of the TE0 register are cleared to 0, and count operation stops. The TCR00, TCR0p, and TCR0q registers hold count value and stop. The TO0p and TO0q outputs are not initialized but hold current status.
	Clears the TOE0p and TOE0q bits of slave channels to 0 and sets a value to the TO0p and TO0q bits.	The levels set in the TO0p and TO0q bit are output from the TO0p and TO0q pins.
TAU stop	To hold the TO0p and TO0q pin output levels Clears the TO0p and TO0q bits to 0 after the value to be held (output latch) is set to the port register.	The TO0p and TO0q pin output levels are held by port function.
	Clears the TAU0EN bit of the PER0 register to 0.	Power-off status (Clock supply is stopped and SFR of the TAU is initialized.)

Remark p: Channel number of slave channel 1, q: Channel number of slave channel 2
$0 < p < q \leq 3$ (Where p and q are a consecutive integer greater than 0)

6.10 Cautions When Using Timer Array Unit

6.10.1 Cautions when using timer output

Depending on the product, a timer output and other alternate functions may be assigned to some pins. In such case, the outputs of the other alternate functions must be set to their initial states.

For details, see **4.5 Register Settings When an Alternate Function Is Used**.

CHAPTER 7 12-BIT INTERVAL TIMER

Note 16-pin products have a single 12-bit interval timer.

7.1 Functions of 12-bit Interval Timer

An interrupt request signal (INTIT) is generated at any previously specified time interval. It can be utilized as the trigger for waking up from STOP mode and HALT mode.

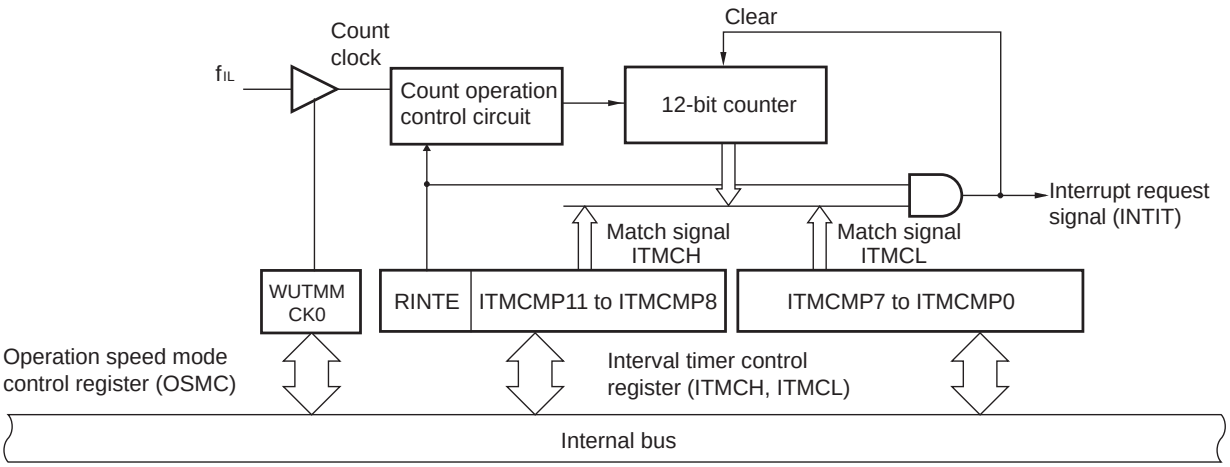
7.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 7-1. Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 0 (PER0)
	Operation speed mode control register (OSMC)
	Interval timer control register H (ITMCH) Interval timer control register L (ITMCL)

Figure 7-1. Block Diagram of 12-bit Interval Timer



7.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Operation speed mode control register (OSMC)
- Interval timer control register (ITMC)

7.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

When using the 12-bit interval timer, be sure to set bit 7 (TMKAEN) to 1 at first.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER0	TMKAEN Note	CMPEN Note	ADCEN	IICA0EN Note	0	SAU0EN	0	TAU0EN

TMKAEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the 12-bit interval timer cannot be written. • The 12-bit interval timer is in the reset status.
1	Enables input clock supply. • SFR used by the 12-bit interval timer can be read and written.

Note 16-pin products only.

Cautions 1. Set the WUTMMCK0 bit of the OSMC register to 1 to determine the clock source for counting before supplying an input clock signal to the 12-bit interval timer (TMKAEN = 1).

2. When setting the 12-bit interval timer, be sure to first set the TMKAEN bit to 1 and then set the following registers, while oscillation of the count clock is stable. If TMKAEN = 0, writing to the 12-bit interval timer is ignored, and all read values are default values (except for the operation speed mode control register (OSMC)).

- **Interval timer control register H (ITMCH)**
- **Interval timer control register L (ITMCL)**

3. Be sure to clear the following bits to 0.

10-pin products: Bits 1, 3, 4, 6, and 7

16-pin products: Bits 1 and 3

7.3.2 Operation speed mode control register (OSMC)

The WUTMMCK0 bit can be used to control supply of the 12-bit interval timer count clock.

Set the WUTMMCK0 bit to 1 before operating the 12-bit interval timer.

Do not clear WUTMMCK0 to 0 before counter operation has stopped.

The OSMC register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-3. Format of Operation Speed Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Supply of count clock for 12-bit interval timer
0	Clock supply stop.
1	Low-speed on-chip oscillator clock (f_{IL}) supply

7.3.3 Interval timer control register (ITMCH, ITMCL)

This register is used to set up the starting and stopping of the 12-bit interval timer operation and to specify the timer compare value.

Set the eight lower-order bits (ITCMP7 to ITCMP0) of the value for comparison in the ITMCL register and then set the four higher-order bits (ITCMP11 to ITCMP8) of the value for comparison and make the setting to stop or start counter operation in the ITMCH register.

The ITMCH and ITMCL registers can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets the ITMCH and ITMCL registers to 0FH and FFH, respectively.

Figure 7-4. Format of Interval Timer Control Register (ITMCH, ITMCL)

Address: FFF91H After reset: 0FH R/W

Symbol	7	6	5	4	3 to 0
ITMCH	RINTE	0	0	0	ITCMP11 to ITCMP8

Address: FFF90H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ITMCL	ITCMP7 to ITCMP0							

RINTE	12-bit interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITCMP11 to ITCMP0	Specification of the 12-bit interval timer compare value
001H	These bits generate an interrupt at the fixed cycle (count clock cycles x (ITCMP setting + 1)).
• • •	
FFFH	
000H	Setting prohibited
Example interrupt cycles when 001H or FFFH is specified for ITCMP11 to ITCMP0 • ITCMP11 to ITCMP0 = 001H, count clock: when f_{IL} = 15 kHz 1/15 [kHz] × (1 + 1) ÷ 0.1333 [ms] = 133.3 [μs] • ITCMP11 to ITCMP0 = FFFH, count clock: when f_{IL} = 15 kHz 1/15 [kHz] × (4095 + 1) ÷ 273 [ms]	

Cautions 1. Set the TMKAMK flag to 1 to disable processing of the INTIT interrupt before stopping the counter (by clearing the RINTE bit to 0). Clear the TMKAIF flag to 0 to enable INTIT interrupt processing before restarting counter operation (by setting the RINTE bit to 1).

2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit to 1.

3. When setting the RINTE bit after returned from standby mode and entering standby mode again, confirm that the written value of the RINTE bit is reflected, or wait that more than one clock of the count clock has elapsed after returned from standby mode. Then enter standby mode.

4. Only change the setting of the ITCMP11 to ITCMP0 bits when the counting operation is stopped (RINTE = 0).
However, it is possible to change the settings of the ITCMP11 to ITCMP8 bits at the same time as when changing the setting of the RINTE bit from 0 to 1 or 1 to 0.

7.4 12-bit Interval Timer Operation

7.4.1 12-bit interval timer operation timing

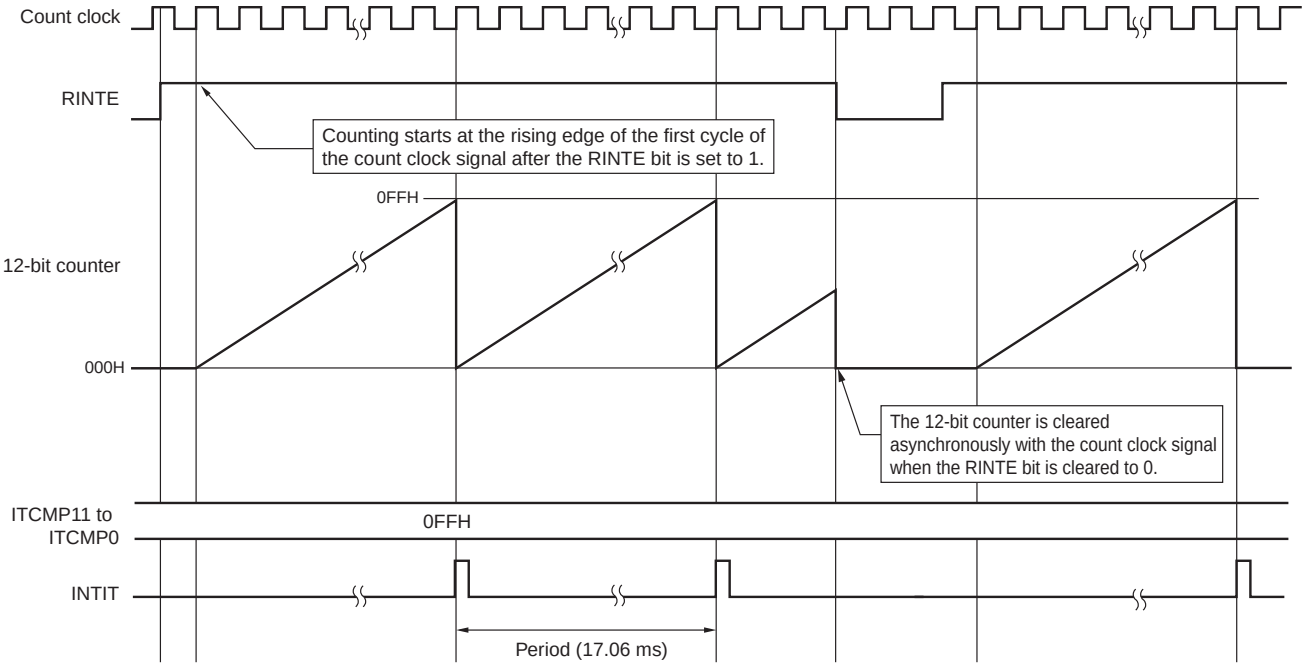
The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate an 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

When the RINTE bit is set to 1, the 12-bit counter starts counting.

When the 12-bit counter value matches the value specified for the ITCMP11 to ITCMP0 bits, the 12-bit counter value is cleared to 0, counting continues, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the 12-bit interval timer is shown in Figure 7-5.

Figure 7-5. 12-bit Interval Timer Operation Timing
(ITCMP11 to ITCMP0 = 0FFH, count clock: f_{IL} = 15 kHz)

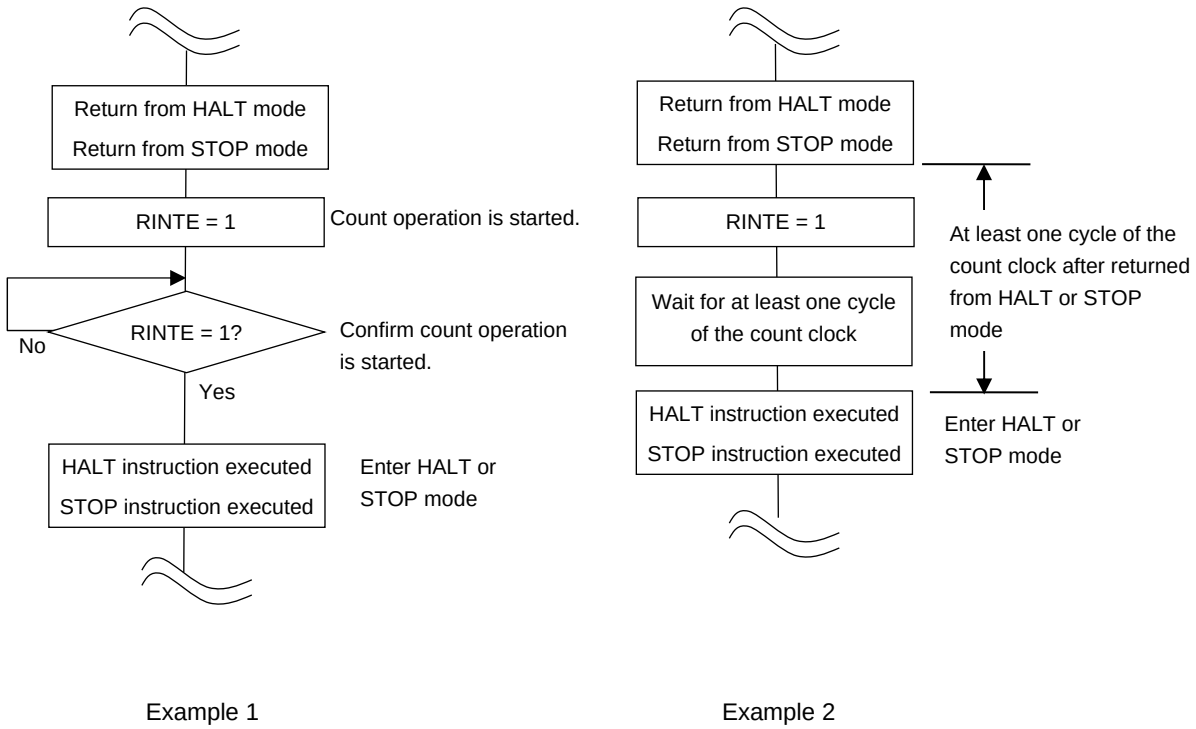


7.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

When setting the RINTE bit after returned from HALT or STOP mode and entering HALT or STOP mode again, write 1 to the RINTE bit, and confirm the written value of the RINTE bit is reflected or wait for at least one cycle of the count clock. Then, enter HALT or STOP mode.

- After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter HALT or STOP mode (see Example 1 in Figure 7-6).
- After setting RINTE to 1, wait for at least one cycle of the count clock and then enter HALT or STOP mode (see Example 2 in Figure 7-6).

Figure 7-6. Procedure of entering to HALT or STOP mode after setting RINTE to 1



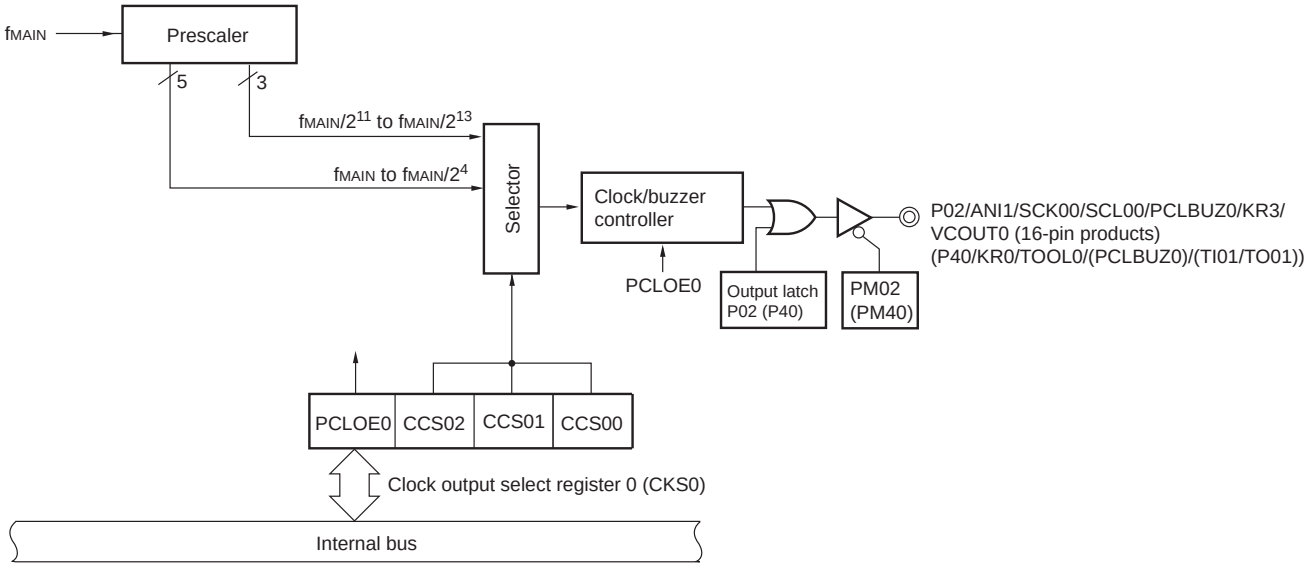
Example 1 Example 2

CHAPTER 8 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

8.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for clock output for supply to peripheral ICs.
Buzzer output is a function to output a square wave of buzzer frequency.
One pin can be used to output a clock or buzzer sound.
The PCLBUZ0 pin outputs a clock selected by clock output select register 0 (CKS0).
Figure 8-1 shows the block diagram of clock output/buzzer output controller.

Figure 8-1. Block Diagram of Clock Output/Buzzer Output Controller



Caution For the frequency the PCLBUZ0 pin can output, refer to 24.4 AC Characteristics.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR).

8.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 8-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select register 0 (CKS0) Port mode register 0 (PM0) [Port mode register 4 (PM4)] Port register 0 (P0) [Port register 4 (P4)] Port mode control register 0 (PMC0) Peripheral I/O redirection register (PIOR)

Remark Functions in brackets in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR).

8.3 Registers Controlling Clock Output/Buzzer Output Controller

The following registers are used to control the clock output/buzzer output controller.

- Clock output select register 0 (CKS0)
- Port mode register 0 (PM0) [Port mode register 4 (PM4)]
- Port mode control register 0 (PMC0)
- Peripheral I/O redirection register (PIOR)

Remark Functions in brackets can be assigned via settings in the peripheral I/O redirection register (PIOR).

8.3.1 Clock output select register 0 (CKS0)

This register sets output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZ0), and sets the output clock.

The CKS0 register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-2. Format of Clock Output Select Register 0 (CKS0)

Address: FFFA5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKS0	PCLOE0	0	0	0	0	CCS02	CCS01	CCS00

PCLOE0	PCLBUZ0 pin output enable/disable specification
0	Output disable (default)
1	Output enable

CCS02	CCS01	CCS00	PCLBUZ0 pin output clock selection					
				f_{MAIN} (MHz)				
				1.25	2.5	5	10	20
0	0	0	f_{MAIN}	1.25 MHz	2.5 MHz	5 MHz Note	10 MHz Note	Setting prohibited Note
0	0	1	$f_{MAIN}/2$	625 kHz	1.25 MHz	2.5 MHz	5 MHz Note	10 MHz Note
0	1	0	$f_{MAIN}/2^2$	312.5 kHz	625 kHz	1.25 MHz	2.5 MHz	5 MHz Note
0	1	1	$f_{MAIN}/2^3$	156.3 kHz	312.5 kHz	625 kHz	1.25 MHz	2.5 MHz
1	0	0	$f_{MAIN}/2^4$	78.1 kHz	156.3 kHz	312.5 kHz	625 kHz	1.25 MHz
1	0	1	$f_{MAIN}/2^{11}$	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	0	$f_{MAIN}/2^{12}$	305 Hz	610 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	1	$f_{MAIN}/2^{13}$	153 Hz	305 Hz	610 Hz	1.22 kHz	2.44 kHz

Note The available output clock varies depending on the operating voltage range. For detail, refer to **24.4 AC Characteristics**.

Cautions 1. Change the output clock after disabling the PCLBUZ0 pin output (PCLOE0 = 0).

2. To shift to STOP mode, execute the STOP instruction when at least 1.5 cycles of the clock used for the PCLBUZ0 pin output have elapsed after the PCLBUZ0 pin output has been disabled.

Remark f_{MAIN}: Main system clock frequency

8.3.2 Registers controlling port functions of clock output/buzzer output pin

Using the port pin for the clock output/buzzer output controller requires setting of the registers that control the port function multiplexed on the clock output/buzzer output pin (PCLBUZ0 pin): (port mode registers 0, 4 (PM0, PM4), port registers 0, 4 (P0, P4), port mode control register 0 (PMC0), peripheral I/O redirection register (PIOR)).

For details on the registers that control the port functions, see **4.3.1 Port mode registers 0, 4 (PM0, PM4)**, **4.3.2 Port registers 0, 4, 12, 13 (P0, P4, P12, P13)**, **4.3.5 Port mode control register 0 (PMC0)**, and **4.3.6 Peripheral I/O redirection register (PIOR)**.

When you intend to use the PCLBUZ0 pin, set the corresponding bits in the port mode register (PM0) and port mode control register 0 (PMC0) to 0 and the corresponding bits in the port register (P0) and port output mode register (POM0) to 1.

For details, see **4.5.3 Example of register settings for port and alternate functions used**.

PCLBUZ0 pin output can be assigned to pin P40 by setting the PIOR bit in the peripheral I/O redirection register (PIOR) to 1.

8.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.
The PCLBUZ0 pin outputs a clock/buzzer selected by the clock output select register 0 (CKS0).

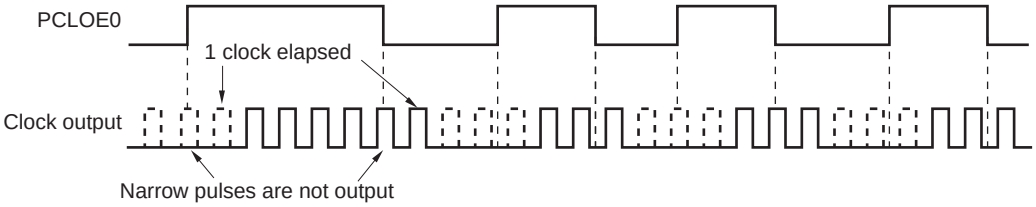
8.4.1 Operation as output pin

The PCLBUZ0 pin is output as the following procedure.

<1> Set the bits in the port mode register (PM0/PM4), port register (P0/P4), and port mode control register 0 (PMC0) that correspond to the pin on which the PCLBUZ0 function is multiplexed to 0.
<2> Select the output frequency with bits 0 to 2 (CCS00 to CCS02) of the clock output select register (CKS0) of the PCLBUZ0 pin (output in disabled).
<3> Set bit 7 (PCLOE0) of the CKS0 register to 1 to enable clock/buzzer output.

Remark The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOE0 bit) is switched. At this time, pulses with a narrow width are not output. Figure 8-3 shows enabling or stopping output using the PCLOE0 bit and the timing of outputting the clock.

Figure 8-3. Timing of Clock Output from PCLBUZ0



Caution Entry to STOP mode within 1.5 clock cycles of the PCLBUZ0 pin output being disabled (PCLOE0 = 0) will shorten the width of the PCLBUZ0 pin output pulse. In such cases, only execute the STOP instruction when at least 1.5 cycles of the clock used for PCLBUZ0 output have elapsed after the PCLBUZ0 pin output has been disabled.

CHAPTER 9 WATCHDOG TIMER

9.1 Functions of Watchdog Timer

The count operation is specified by the user option byte (000C0H) in the watchdog timer.

The watchdog timer operates on the low-speed on-chip oscillator clock.

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of the RESF register, see **CHAPTER 17 RESET FUNCTION**.

When 75% of the overflow time is reached, an interval interrupt is generated.

9.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 9-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

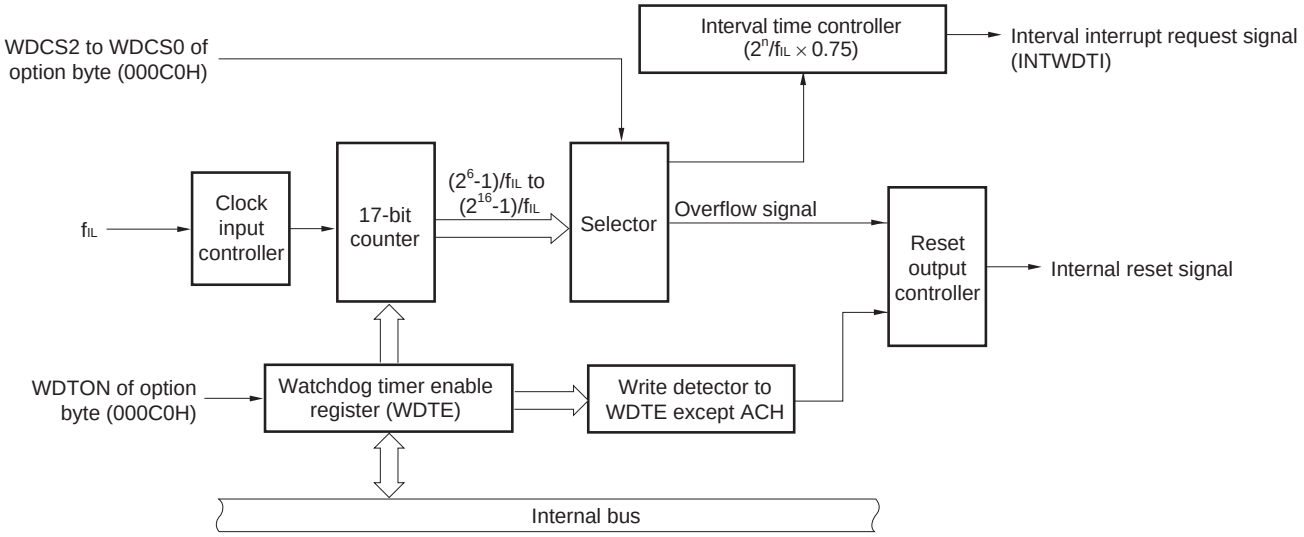
How the counter operation is controlled and overflow time are set by the option byte.

Table 9-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

Remark For the option byte, see **CHAPTER 19 OPTION BYTE**.

Figure 9-1. Block Diagram of Watchdog Timer



9.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

9.3.1 Watchdog timer enable register (WDTE)

Writing “ACH” to the WDTE register clears the watchdog timer counter and starts counting again.
This register can be set by an 8-bit memory manipulation instruction.
Reset signal generation sets this register to 1AH or 9AH[Note].

Figure 9-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH After reset: 1AH/9AH[Note] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Cautions 1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.
2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.
3. The value read from the WDTE register is 1AH/9AH (this differs from the written value (ACH)).

9.4 Operation of Watchdog Timer

9.4.1 Controlling operation of watchdog timer

<1> When the watchdog timer is used, its operation is specified by the option byte (000C0H).

- Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 19**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

- Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **9.4.2** and **CHAPTER 19**).

<2> After a reset release, the watchdog timer starts counting.

<3> By writing “ACH” to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.

<4> If the overflow time expires without “ACH” written to the WDTE register, an internal reset signal is generated.
An internal reset signal is generated in the following cases.

- If a 1-bit manipulation instruction is executed on the WDTE register
- If data other than “ACH” is written to the WDTE register

Cautions 1. If the watchdog timer is cleared by writing “ACH” to the WDTE register, the actual overflow time may become shorter than the overflow time set by the option byte by up to one clock cycle of f_{IL}.

2. The watchdog timer can be cleared immediately before the count value overflows.

3. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

WDSTBYON = 0 : Watchdog timer operation stops.
WDSTBYON = 1 : Watchdog timer operation continues.

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts.

When operating with the X1 oscillation clock Note after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

Note 16-pin products only.

9.4.2 Setting time of watchdog timer

Set the overflow time and interval interrupt time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to the watchdog timer enable register (WDTE) before the overflow time.

When 75% of the overflow time is reached, an interval interrupt is generated.

The following overflow time and interval interrupt time can be set.

Table 9-3. Setting of Overflow Time and Interval Interrupt Time

WDCS2	WDCS1	WDCS0	Overflow Time (when f_{IL} = 17.25 kHz (MAX.))	Interval Interrupt Time (when f_{IL} = 17.25 kHz (MAX.))
0	0	0	$(2^6 -1)/f_{IL}$ (3.65 ms)	$2^6/f_{IL} \times 0.75$ (2.78 ms)
0	0	1	$(2^7 -1)/f_{IL}$ (7.36 ms)	$2^7/f_{IL} \times 0.75$ (5.56 ms)
0	1	0	$(2^8 -1)/f_{IL}$ (14.7 ms)	$2^8/f_{IL} \times 0.75$ (11.1 ms)
0	1	1	$(2^9 -1)/f_{IL}$ (29.6 ms)	$2^9/f_{IL} \times 0.75$ (22.2 ms)
1	0	0	$(2^{11} -1)/f_{IL}$ (118 ms)	$2^{11}/f_{IL} \times 0.75$ (89.0 ms)
1	0	1	$(2^{13} -1)/f_{IL}$ (474 ms)	$2^{13}/f_{IL} \times 0.75$ (356 ms)
1	1	0	$(2^{14} -1)/f_{IL}$ (949 ms)	$2^{14}/f_{IL} \times 0.75$ (712 ms)
1	1	1	$(2^{16} -1)/f_{IL}$ (3799 ms)	$2^{16}/f_{IL} \times 0.75$ (2849 ms)

Cautions 1. When operating with the X1 oscillation clock Note after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock Note and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

2. The watchdog timer continues counting even after INTWDTI is generated (until ACH is written to the watchdog timer enable register (WDTE)). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

3. The watchdog timer always generates an interval interrupt when the specified time is reached unless this is specifically disabled. If the interval interrupt from the watchdog timer is not to be used, be sure to disable the interrupt by setting the WDTIMK bit to 1.

Note 16-pin products only.

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

CHAPTER 10 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

- 10-pin products: 4 channels (ANI0 to ANI3)
- 16-pin products: 7 channels (ANI0 to ANI6), internal reference voltage [Note] (0.815 V (typ.))

Note The internal reference voltage cannot be used for the A/D converter and comparator simultaneously. When the internal reference voltage is selected as the target for conversion by the A/D converter, do not select the internal reference voltage as the reference voltage of the comparator.

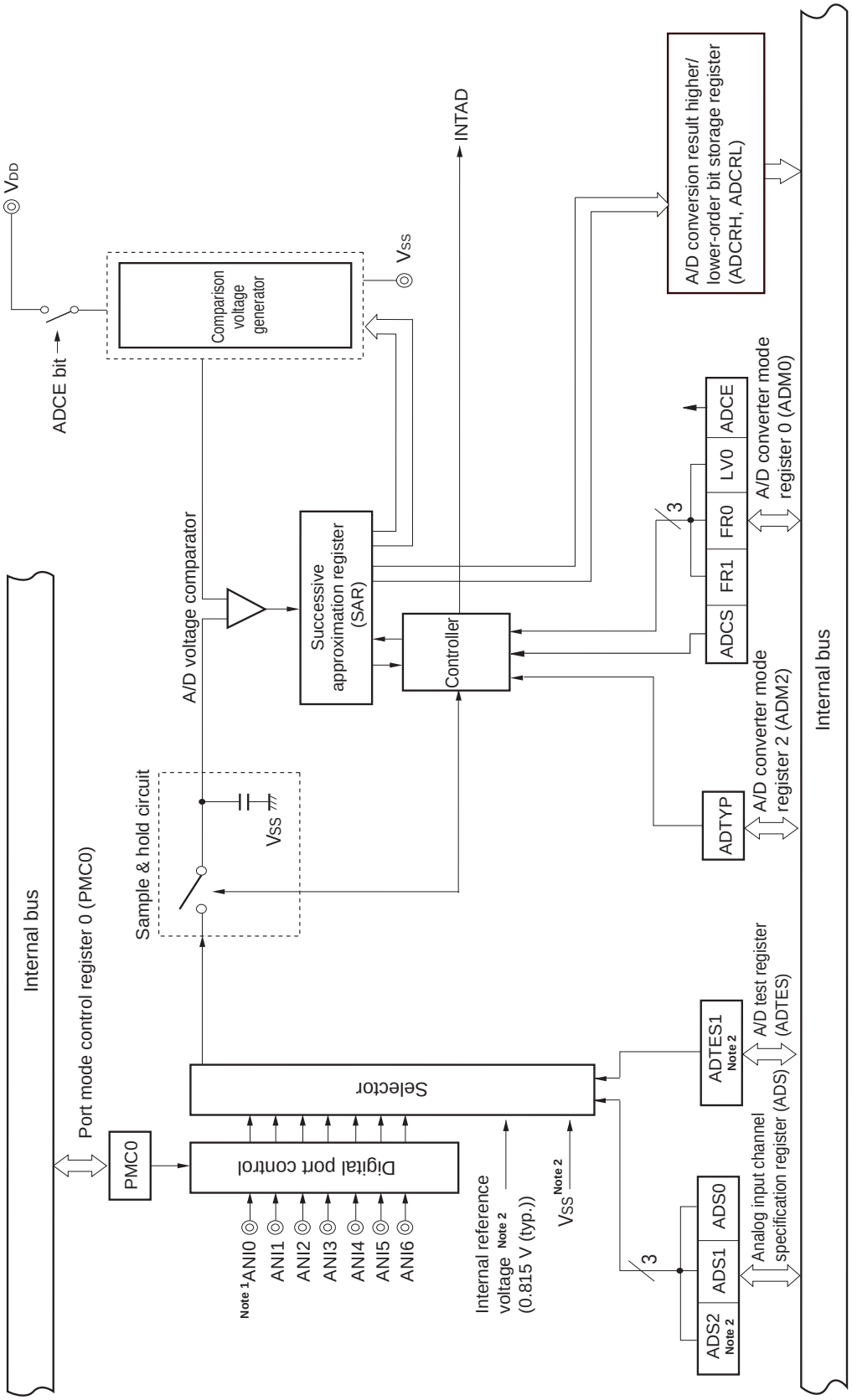
10.1 Function of A/D Converter

The A/D converter is used to convert analog input signals into digital values, and is configured to control up to 7 channels of A/D converter analog inputs. Ten-bit or eight-bit resolution can also be selected by using the ADTYP bit of A/D converter mode register 2 (ADM2).

The A/D converter has the following function.

- 10-bit/8-bit resolution A/D conversion
  Following the selection of one analog input channel from among ANI0 to ANI6, software initiates A/D conversion with 10-bit or 8-bit resolution. An A/D conversion end interrupt request signal (INTAD) is generated on completion of A/D conversion. The range of operating voltage for the A/D converter is from 2.4 to 5.5 V.

Figure 10-1. Block Diagram of A/D Converter



Notes 1. For 10-pin products, ANI0 to ANI3.

2. 16-pin products only.

10.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI6 [Note] pins

These are the analog input pins of the 7 channels of the A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.
In addition to the voltages on analog input pins from ANI0 to ANI6, the internal reference voltage (0.815 (typ.)) can be selected as the target for conversion by the A/D converter.

Note For 10-pin products, only ANI0 to ANI3.

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) A/D voltage comparator

This A/D voltage comparator compares the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 V_{DD}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 V_{DD}), the MSB bit of the SAR is reset.
After that, bit 8 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 9, to which the result has been already set.

Bit 9 = 0: (1/4 V_{DD})
Bit 9 = 1: (3/4 V_{DD})

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 8 = 1
Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.
When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 2 of the SAR register.

(4) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR register is a register that sets voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result higher-order bit storage register (ADCRH) and the A/D conversion result lower-order bit storage register (ADCRL). When A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(6) A/D conversion result higher-order bit storage register (ADCRH)

ADCRH is an 8-bit register which holds the result of A/D conversion. The conversion result is loaded from the successive approximation register, and the eight higher-order bits of the A/D conversion result are stored in ADCRH.

The two lower-order bits of the result of 10-bit A/D conversion are stored in ADCRL.

(7) A/D conversion result lower-order bit storage register (ADCRL)

ADCRL is an 8-bit register which holds the two lower-order bits (ADCR1, ADCR0) of the result of 10-bit A/D conversion. The six lower-order bits of this register are fixed to 0.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates an A/D conversion end interrupt request signal (INTAD).

10.3 Registers Used in A/D Converter

The A/D converter is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register 0 (ADM0)
- A/D converter mode register 2 (ADM2)
- A/D conversion result higher-order bit storage register (ADCRH)
- A/D conversion result lower-order bit storage register (ADCRL)
- Analog input channel specification register (ADS)
- A/D test register (ADTES)
- Port mode register 0 (PM0)
- Port mode control register 0 (PMC0)

10.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER0	TMKAEN[Note]	CMPEN[Note]	ADCEN	IICA0EN[Note]	0	SAU0EN	0	TAU0EN

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. • SFR used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables input clock supply. • SFR used by the A/D converter can be read/written.

Note 16-pin products only.

Cautions 1. When setting the A/D converter, be sure to set the following registers while the ADCEN bit is set to 1 first. If ADCEN = 0, the values of the A/D converter control registers are cleared to their initial values and writing to them is ignored (except for the port mode register 0 (PM0) and the port mode control register 0 (PMC0)).

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 2 (ADM2)**
- **A/D conversion result higher-order bit storage register (ADCRH)**
- **A/D conversion result lower-order bit storage register (ADCRL)**
- **Analog input channel specification register (ADS)**
- **A/D test register (ADTES)**

2. Be sure to clear the following bits to 0.

10-pin products: Bits 1, 3, 4, 6, and 7

16-pin products: Bits 1 and 3

10.3.2 A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-3. Format of A/D Converter Mode Register 0 (ADM0)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	0	0	FR1 Note 1	FR0 Note 1	0	LV0 Note 1	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation (conversion stopped/standby status)
1	Enables conversion operation (conversion operation status)
<Clear conditions> • 0 is written to ADCS. • The bit is automatically cleared to 0 when A/D conversion ends. <Set condition> • 1 is written to ADCS when the ADCE bit is 1.	

ADCE	A/D voltage comparator operation control Note 2
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Notes 1. For details of the FR1, FR0, and LV0 bits and A/D conversion, see **Table 10-2 10-Bit Resolution A/D Conversion Time Selection** or **Table 10-3 8-Bit Resolution A/D Conversion Time Selection**.

2. The operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 0.1 μs from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 0.1 μs or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. If the ADCS bit is set to 1 to perform A/D conversion without waiting for at least 0.1 μs, ignore data of the conversion.

Cautions 1. Rewrite the values of the FR1, FR0, and LV0 bits in the conversion standby status (ADCS = 0, ADCE = 1) or in the conversion stopped status (ADCS = 0, ADCE = 0). Rewriting the values of the FR1, FR0, and LV0 bits, and ADCS bits by an 8-bit manipulation instruction at the same time is prohibited.

2. Setting ADCS =1 and ADCE = 0 is prohibited. When 1 is written to the ADCS bit in the conversion stopped status (ADCE = 0, ADCS = 0), the ADCS bit is not set to 1.

3. Changing the ADCE and ADCS bits from 0 to 1 at the same time by using an 8-bit manipulation instruction is prohibited. Be sure to set these bits in the order described in 10.7 A/D Converter Setup Flowchart.

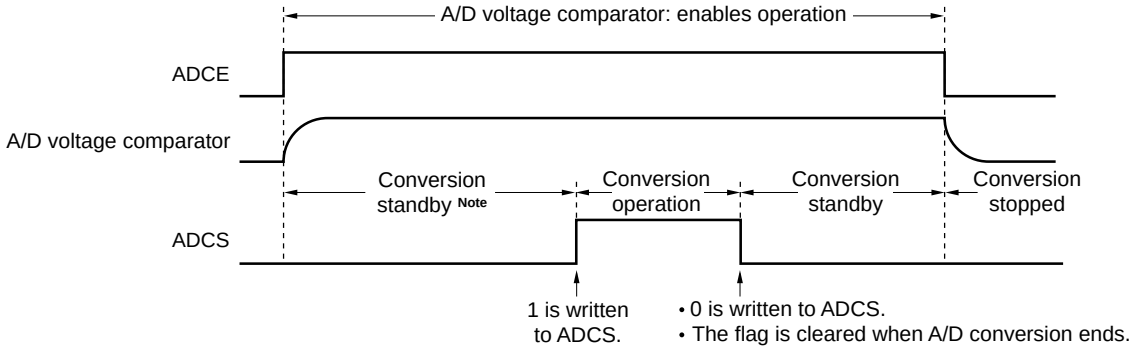
4. Be sure to clear bits 2, 5, and 6 to 0.

5. Setting the ADCS bit to 1 during conversion (ADCS = 1) is prohibited. When restarting the conversion for the same channel is required, stop conversion once (ADCS = 0), and then restart the A/D conversion (ADCS = 1).

Table 10-1. Settings of ADCS and ADCE Bits

ADCS	ADCE	A/D Conversion Operation
0	0	Conversion stopped state
0	1	Conversion standby state
1	0	Setting prohibited
1	1	Conversion-in-progress state

Figure 10-4. Timing Chart When A/D Voltage Comparator Is Used



Note It requires at least 0.1 μs to stabilize the internal circuit until the A/D conversion operation is started (ADCS = 1) after the operation of the A/D voltage comparator is enabled (ADCE = 1). If the ADCS bit is set to 1 without waiting for at least 0.1 μs, ignore data of the first conversion.

Table 10-2. 10-Bit Resolution A/D Conversion Time Selection

A/D Converter Mode Register 0 (ADM0)			Conversion Clock	Number of Conversion Clock	Conversion Time	Conversion Time Selection (μs)				
FR1	FR0	LV0 Note 2				f_{CLK} = 1.25 MHz	f_{CLK} = 2.5 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz Note 1
0	0	0	$f_{CLK}/8$	23 f_{AD} (Number of sampling clock: 9 f_{AD})	$184/f_{CLK}$	Setting prohibited	Setting prohibited	Setting prohibited	18.4	9.2
0	1		$f_{CLK}/4$		$92/f_{CLK}$			18.4	9.2	4.6
1	0		$f_{CLK}/2$		$46/f_{CLK}$		18.4	9.2	4.6	Setting prohibited
1	1		f_{CLK}		$23/f_{CLK}$	18.4	9.2	4.6	Setting prohibited	
0	0	1 Note 1	$f_{CLK}/8$	17 f_{AD} (Number of sampling clock: 3 f_{AD})	$136/f_{CLK}$	Setting prohibited	Setting prohibited	Setting prohibited	13.6	6.8
0	1		$f_{CLK}/4$		$68/f_{CLK}$			13.6	6.8	3.4
1	0		$f_{CLK}/2$		$34/f_{CLK}$		13.6	6.8	3.4	Setting prohibited
1	1		f_{CLK}		$17/f_{CLK}$	13.6	6.8	3.4	Setting prohibited	

Notes 1. Setting prohibited when 2.4 V ≤ V_{DD} < 2.7 V. Can be selected when 2.7 V ≤ V_{DD} ≤ 5.5 V.

2. When the internal reference voltage is selected as the target for conversion by the A/D converter, be sure to clear the LV0 bit to 0.

Table 10-3. 8-Bit Resolution A/D Conversion Time Selection

A/D Converter Mode Register 0 (ADM0)			Conversion Clock	Number of Conversion Clock	Conversion Time	Conversion Time Selection (μs)				
FR1	FR0	LV0 Note 2				f_{CLK} = 1.25 MHz	f_{CLK} = 2.5 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz Note 1
0	0	0	$f_{CLK}/8$	21 f_{AD} (Number of sampling clock: 9 f_{AD})	$168/f_{CLK}$	Setting prohibited	Setting prohibited	Setting prohibited	16.8	8.4
0	1		$f_{CLK}/4$		$84/f_{CLK}$			16.8	8.4	4.2
1	0		$f_{CLK}/2$		$42/f_{CLK}$		16.8	8.4	4.2	Setting prohibited
1	1		f_{CLK}		$21/f_{CLK}$	16.8	8.4	4.2	Setting prohibited	
0	0	1 Note 1	$f_{CLK}/8$	15 f_{AD} (Number of sampling clock: 3 f_{AD})	$120/f_{CLK}$	Setting prohibited	Setting prohibited	Setting prohibited	12.0	6.0
0	1		$f_{CLK}/4$		$60/f_{CLK}$			12.0	6.0	3.0
1	0		$f_{CLK}/2$		$30/f_{CLK}$		12.0	6.0	3.0	Setting prohibited
1	1		f_{CLK}		$15/f_{CLK}$	12.0	6.0	3.0	Setting prohibited	

Notes 1. Setting prohibited when 2.4 V ≤ V_{DD} < 2.7 V. Can be selected when 2.7 V ≤ V_{DD} ≤ 5.5 V.

2. When the internal reference voltage is selected as the target for conversion by the A/D converter, be sure to clear the LV0 bit to 0.

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 24.6.1 A/D converter characteristics.

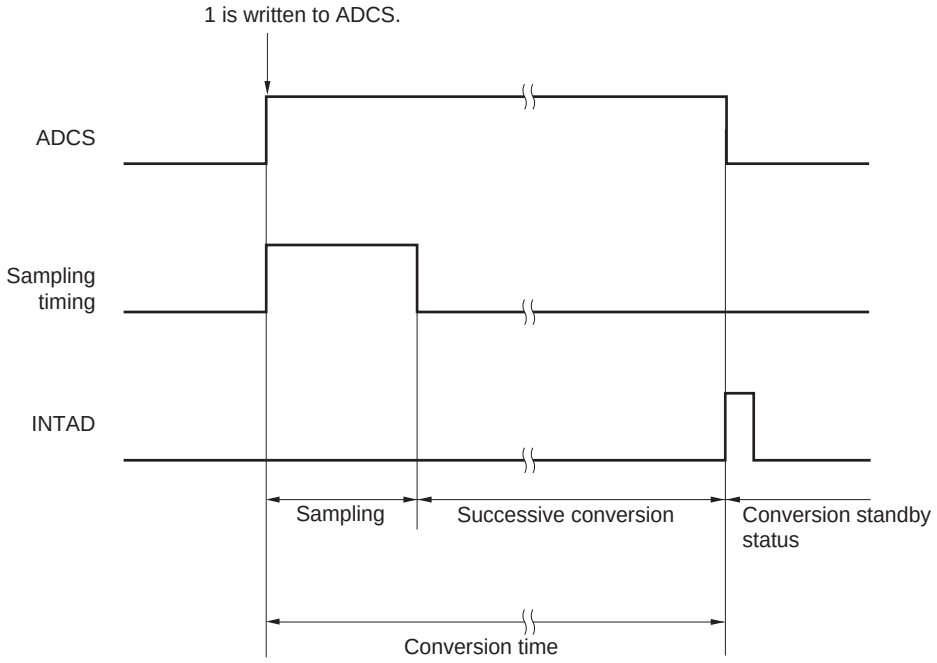
2. When the internal reference voltage is selected as the target for conversion by the A/D converter, the internal reference voltage cannot be used as the reference voltage of the comparator.

3. Rewrite the values of the FR1, FR0, and LV0 bits to other than the same data in the conversion standby status (ADCS = 0, ADCE = 1) or in the conversion stopped status (ADCS = 0, ADCE = 0). Rewriting the values of the FR1, FR0, and LV0 bits, and ADCS bits by an 8-bit manipulation instruction at the same time is prohibited.

4. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 10-5. A/D Converter Sampling and A/D Conversion Timing



10.3.3 A/D converter mode register 2 (ADM2)

This register is used to set the resolution of the A/D converter.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-6. Format of A/D Converter Mode Register 2 (ADM2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
ADM2	0	0	0	0	0	0	0	ADTYP

ADTYP	Resolution of A/D conversion
0	10-bit resolution
1	8-bit resolution

Caution Rewrite the value of the ADM2 register in the conversion stopped status (while the ADCS and ADCE bits are set to 0).

10.3.4 A/D conversion result higher-order bit storage register (ADCRH)

ADCRH is an 8-bit register which holds the result of A/D conversion. The conversion result is loaded from the successive approximation register after A/D conversion ends. When 10-bit resolution is selected, the eight higher-order bits of the A/D conversion result are stored in this register and the two lower-order bits of the A/D conversion result are stored in ADCRL.

The ADCRH register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-7. Format of A/D Conversion Result Higher-Order Bit Storage Register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH	ADCR9	ADCR8	ADCR7	ADCR6	ADCR5	ADCR4	ADCR3	ADCR2

Caution When writing to the A/D converter mode register 0 (ADM0) and the analog input channel specification register (ADS), the contents of the ADCRH/ADCRL register may become undefined. Read the conversion result following conversion completion before writing to the ADM0 and ADS registers. Using timing other than the above may cause an incorrect conversion result to be read.

10.3.5 A/D conversion result lower-order bit storage register (ADCRL)

This register is an 8-bit register that holds the two lower-order bits of the result of 10-bit A/D conversion. The six lower-order bits are fixed to 0.

The ADCRL register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-8. Format of A/D Conversion Result Lower-Order Bit Storage Register (ADCRL)

Address: FFF1EH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRL	ADCR1	ADCR0	0	0	0	0	0	0

Figure 10-9 shows the state after the result of 10-bit resolution A/D conversion has been stored. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The eight higher-order bits of the conversion result are stored in ADCRH and the two lower-order bits of the result are stored in ADCRL.

Figure 10-9. The State after Storage of the Result of 10-bit Resolution A/D Conversion

ADCRH								ADCRL							
ADC R9	ADC R8	ADC R7	ADC R6	ADC R5	ADC R4	ADC R3	ADC R2	ADC R1	ADC R0	0	0	0	0	0	0

Cautions 1. When writing to the A/D converter mode register 0 (ADM0) and analog input channel specification register (ADS), the contents of the ADCRH/ADCRL registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0 and ADS registers. Using timing other than the above may cause an incorrect conversion result to be read.

2. When the ADCRL register is read while 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1), 0 is read from the two higher-order bits (ADCR1 and ADCR0). Note that, when ADCRL register is read before completion of A/D conversion while 8-bit resolution A/D conversion is selected, 0 may not be read from the two higher-order bits (ADCR1, ADCR0).

10.3.6 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-10. Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	0	ADS2 Note 1	ADS1	ADS0

10-pin products

ADS1	ADS0	Target of A/D conversion	Analog input pin
0	0	ANI0	P01/ANI0 pin
0	1	ANI1	P02/ANI1 pin
1	0	ANI2	P03/ANI2 pin
1	1	ANI3	P04/ANI3 pin

16-pin products

ADS2	ADS1	ADS0	Target of A/D conversion	Analog input pin
0	0	0	ANI0	P01/ANI0 pin
0	0	1	ANI1	P02/ANI1 pin
0	1	0	ANI2	P03/ANI2 pin
0	1	1	ANI3	P04/ANI3 pin
1	0	0	ANI4	P05/ANI4 pin
1	0	1	ANI5	P06/ANI5 pin
1	1	0	ANI6	P07/ANI6 pin
1	1	1	Internal reference voltage (0.815 V (typ.)) Note 2	–

Notes 1. 16-pin products only.

2. When the internal reference voltage is selected as the target for conversion by the A/D converter, be sure to clear the LV0 bit in the A/D converter mode register 0 (ADM0) to 0.

Cautions 1. Rewrite the ADS register in the conversion standby status (ADCS = 0, ADCE = 1) or in the conversion stopped status (ADCS = 0, ADCE = 0).

2. Set the port used as an analog input port to the input mode by using the port mode register 0 (PM0) and to the analog input by using the port mode control register 0 (PMC0). Do not set the pin that is set by port mode control register 0 (PMC0) as digital I/O by the ADS register.

3. The internal reference voltage cannot be used for the A/D converter and comparator simultaneously. When the internal reference voltage is selected as the target for conversion by the A/D converter (ADS2 to ADS0 = 111B), it cannot be selected as the reference voltage of the comparator.

4. Be sure to clear the following bits to 0.

10-pin products: Bits 2 to 7

16-pin products: Bits 3 to 7

10.3.7 A/D test register (ADTES)

This register is used to select VSS as the analog input to be A/D converted. When the internal reference voltage (0.815 V (typ.)) is selected as the target of A/D conversion, the sampling capacitor must be discharged before A/D conversion of this voltage proceeds.

The ADTES register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 10-11. Format of A/D Test Register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	0

ADTES1	Selection of target for A/D conversion
0	ANIx/internal reference voltage (0.815 V (typ.)) (This is specified using the analog input channel specification register (ADS).)
1	V_{SS} (discharging the sampling capacitor)

Caution **When A/D converting the internal reference voltage (0.815 V (typ.)), follow the procedure described in 10.7.2 Setting up A/D conversion of the internal reference voltage (16-pin products only), including discharge of the sampling capacitor once.**

Remark Be sure to clear bits 0 and 2 to 7 to 0.

10.3.8 Registers controlling port function of analog input pins

Set up the registers for controlling the functions of the ports shared with the analog input pins of the A/D converter (port mode register 0 (PM0) and port mode control register 0 (PMC0)). For details, see **4.3.1 Port mode registers 0, 4 (PM0, PM4)** and **4.3.5 Port mode control register 0 (PMC0)**.

For an example of settings when using a port pin for analog input of the A/D converter, see **4.5.3 Example of register settings for port and alternate functions used**.

When using the ANI0 to ANI6 pins for analog input of the A/D converter, set the bits in the port mode register 0 (PM0) and port mode control register 0 (PMC0) corresponding to each port to 1.

10.4 A/D Converter Conversion Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) V_{DD} by the tap selector.

<4> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) V_{DD}, the MSB bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) V_{DD}, the MSB bit is reset to 0.

<5> Next, bit 8 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) V_{DD}
- Bit 9 = 0: (1/4) V_{DD}

The voltage tap and sampled voltage are compared and bit 8 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

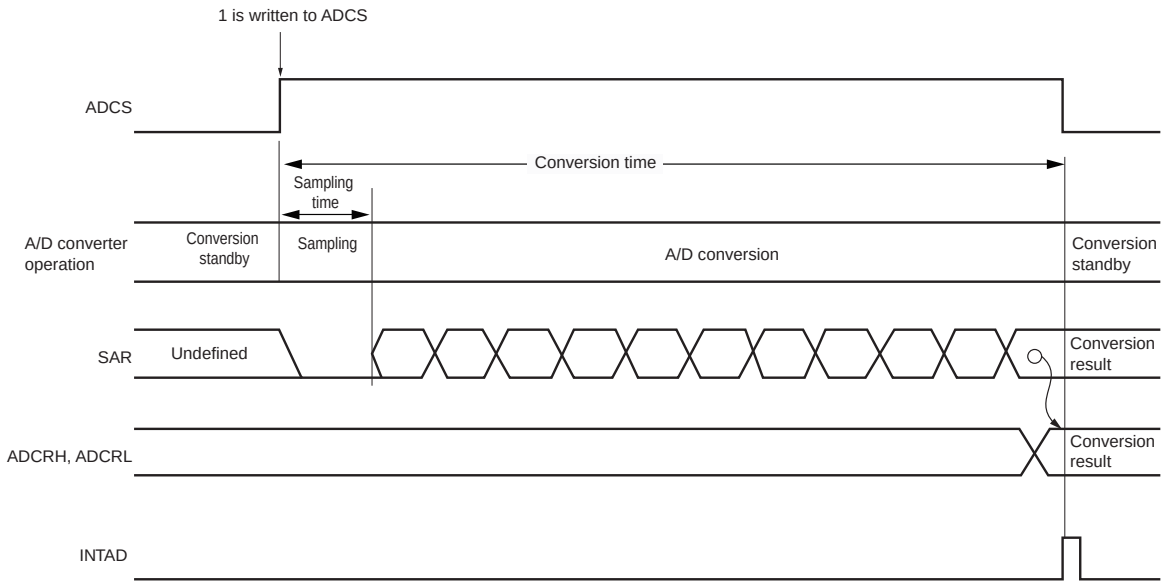
<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCRH, ADCRL) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) is generated.
After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

Remark Two types of the A/D conversion result registers are available.

- ADCRH register (8 bits): Store eight higher-order bits of the result of 10-bit resolution A/D conversion or the result of 8-bit resolution A/D conversion.
- ADCRL register (2 bits): Store two lower-order bits of the result of 10-bit resolution A/D conversion.

Figure 10-12. Conversion Operation of A/D Converter



A/D conversion is performed once when the bit 7 (ADCS) of the A/D converter mode register 0 (ADM0) is set to 1 by software. The ADCS bit is automatically cleared to 0 after A/D conversion ends.

Reset signal generation clears the A/D conversion result register (ADCRH, ADCRL) to 00H.

10.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI6 **Note**, internal reference voltage) and the theoretical A/D conversion result (stored in the A/D conversion result register (ADCR = ADCRH + ADCRL)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{V_{DD}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{V_{DD}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{V_{DD}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

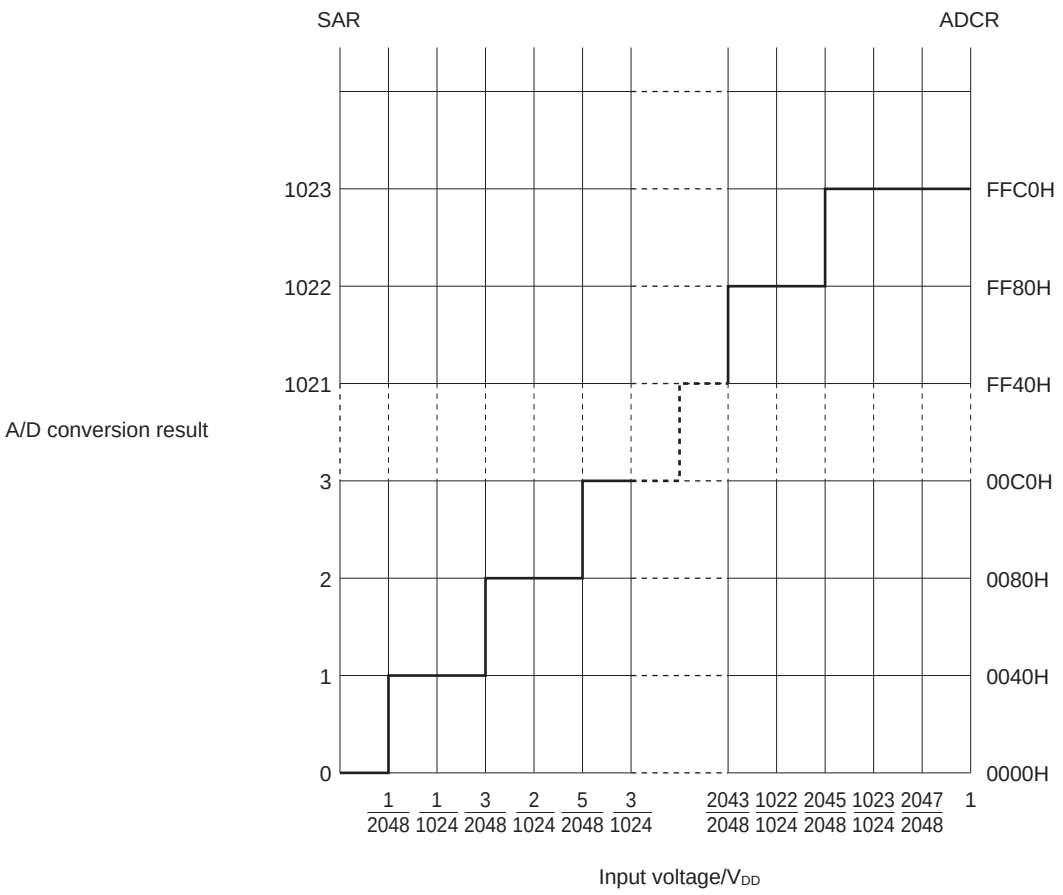
ADCR: A/D conversion result register (ADCRH + ADCRL) value

SAR: Successive approximation register

Note For 10-pin products, ANI0 to ANI3.

Figure 10-13 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-13. Relationship Between Analog Input Voltage and A/D Conversion Result



10.6 A/D Converter Operation Modes

The operation of the A/D converter is described below. In addition, the setting procedure is described in **10.7 A/D Converter Setup Flowchart**.

<1> In the conversion stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the conversion standby status.

<2> After the software counts up to the stabilization wait time (0.1 μs), the ADCS bit of the ADM0 register is set to 1 to start the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

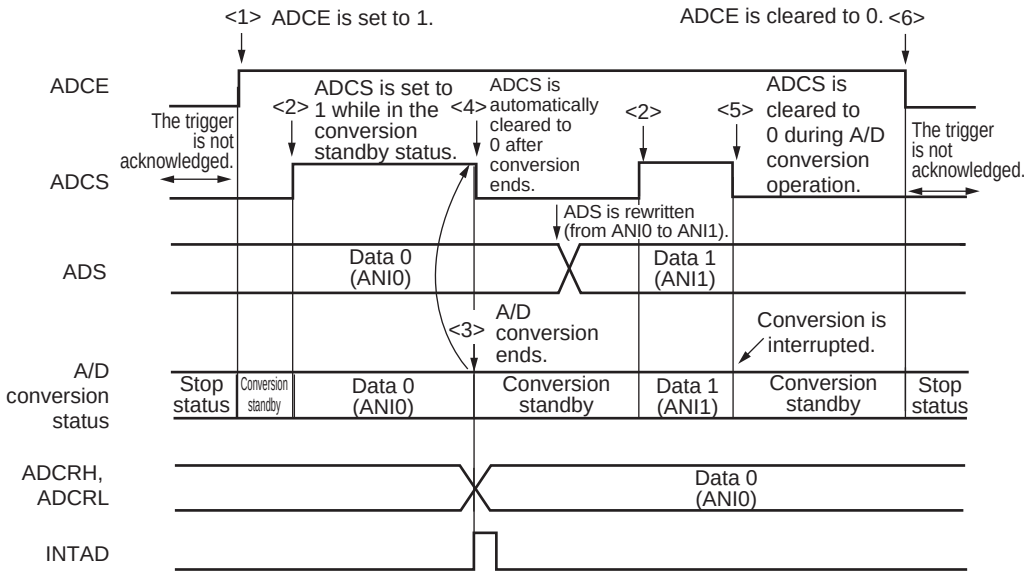
<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCRH, ADCRL), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the conversion standby status.

<5> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the conversion standby status.

<6> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the conversion stop status. Setting ADCS =1 and ADCE = 0 is prohibited. Specifying 1 for ADCS in the conversion stopped status (ADCS =0, ADCE = 0) is ignored and A/D conversion does not start.

Figure 10-14. Example of Operation Timing

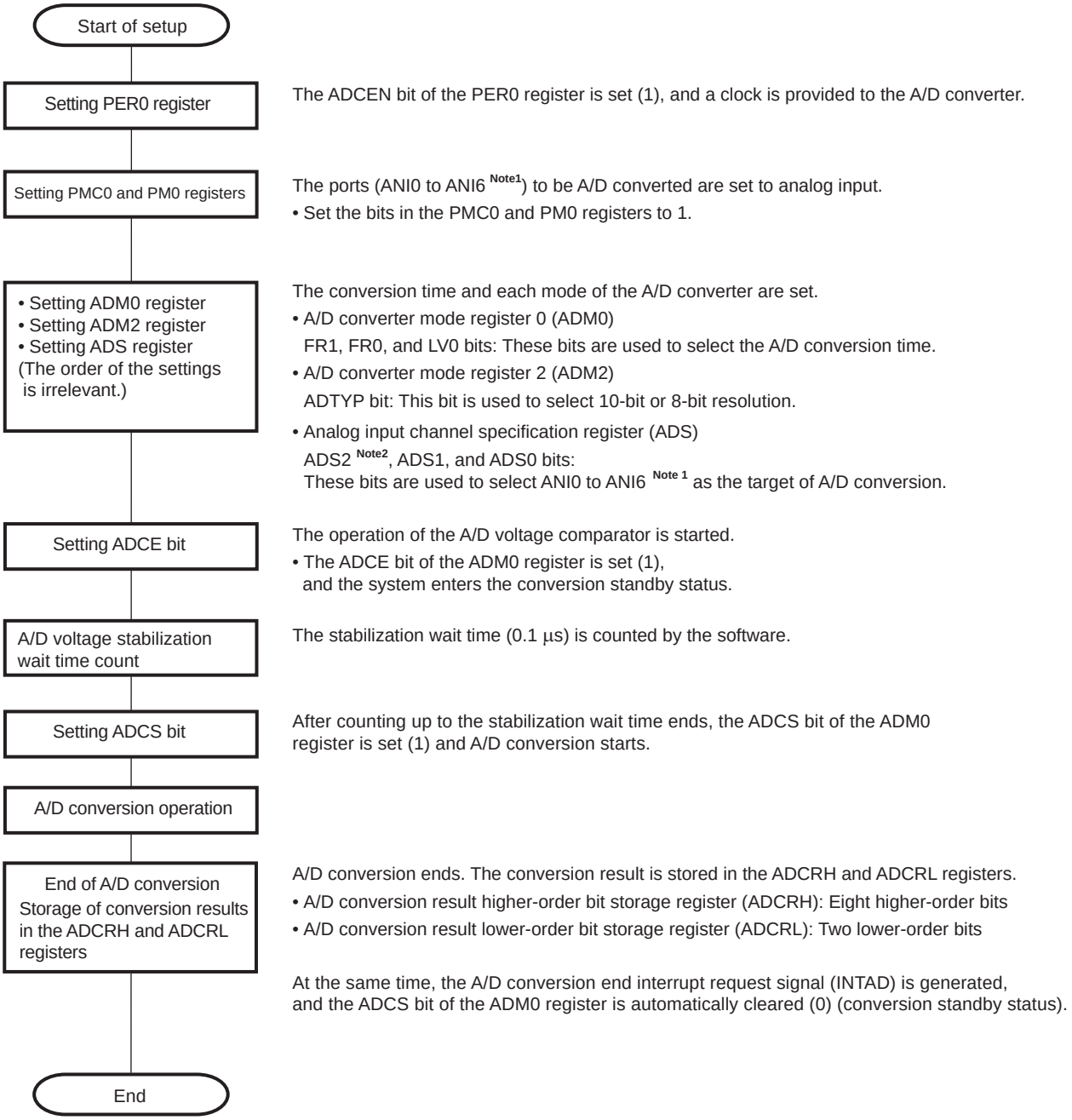


10.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart is described below.

10.7.1 Setting up A/D conversion of voltages on ANI0 to ANI6

Figure 10-15. Setting Up A/D Conversion of Voltages on ANI0 to ANI6

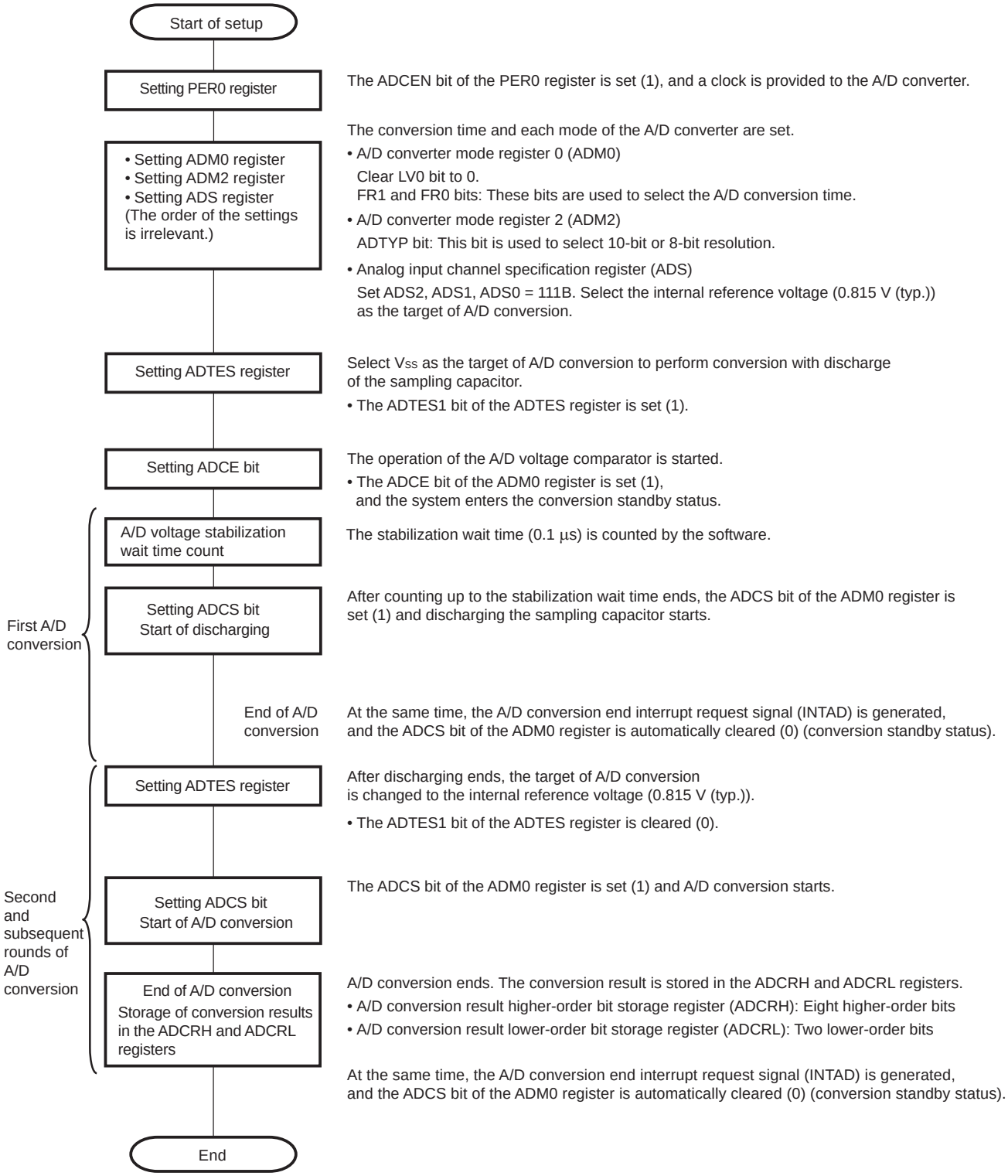


Notes 1. For 10-pin products, ANI0 to ANI3.

2. 16-pin products only.

10.7.2 Setting up A/D conversion of the internal reference voltage (16-pin products only)

Figure 10-16. Setting Up A/D Conversion of Internal Reference Voltage



10.8 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

10.8.1 Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$
$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

10.8.2 Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

10.8.3 Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 10-17. Overall Error

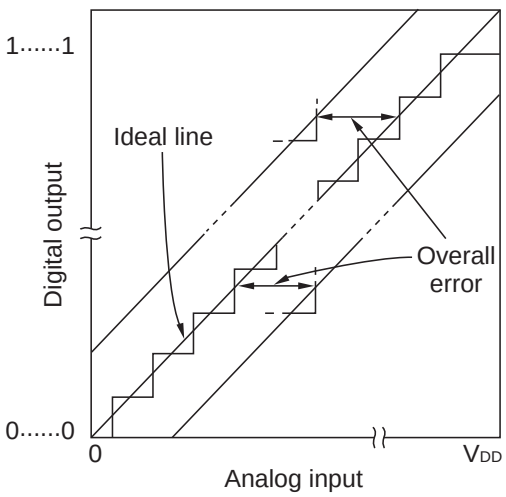
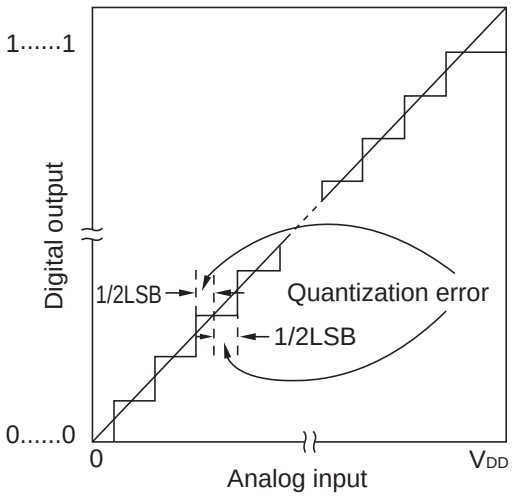


Figure 10-18. Quantization Error



10.8.4 Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0.....000 to 0.....001.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

10.8.5 Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (full-scale – 3/2LSB) when the digital output changes from 1.....110 to 1.....111.

10.8.6 Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

10.8.7 Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 10-19. Zero-Scale Error

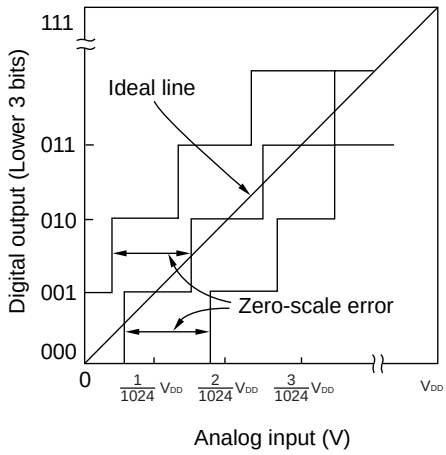


Figure 10-20. Full-Scale Error

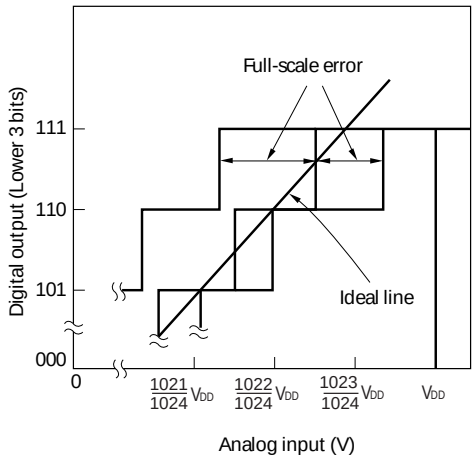


Figure 10-21. Integral Linearity Error

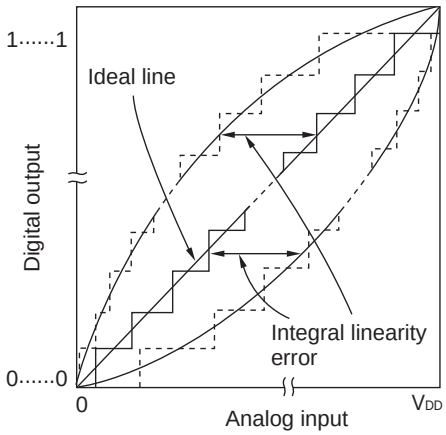
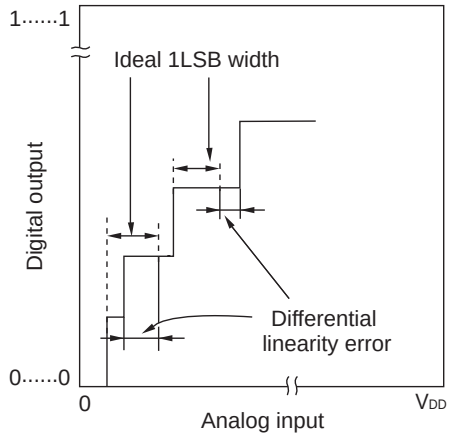


Figure 10-22. Differential Linearity Error



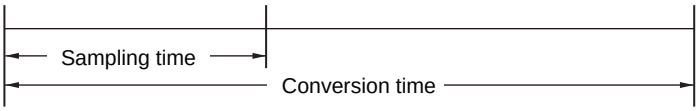
10.8.8 Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

10.8.9 Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



10.9 Cautions for A/D Converter

10.9.1 Operating current in STOP mode

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the ADM0 register to 0 at the same time.

10.9.2 Input range of ANI0 to ANI6 pins

Observe the rated range of the ANI0 to ANI6 [Note] pins input voltage. If a voltage exceeding V_{DD} or equal to or lower than V_{SS} (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

10.9.3 Conflicting operations

<1> Reading from the ADCRH or ADCRL register has priority if conflict between writing to the A/D conversion result register (ADCRH, ADCRL) and reading from ADCRH or ADCRL register by software operation occurs at the end of conversion. After the read operation, the new conversion result is written to the ADCRH or ADCRH register.

<2> Writing to the ADM0 register has priority if conflict between writing to the ADCRH or ADCRL register and writing to the A/D converter mode register 0 (ADM0) occurs at the end of conversion. Writing to the ADCRH or ADCRL register is not performed, nor is the A/D conversion end interrupt signal (INTAD) generated.

10.9.4 Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the V_{DD} and ANI0 to ANI6 [Note] pins.

<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

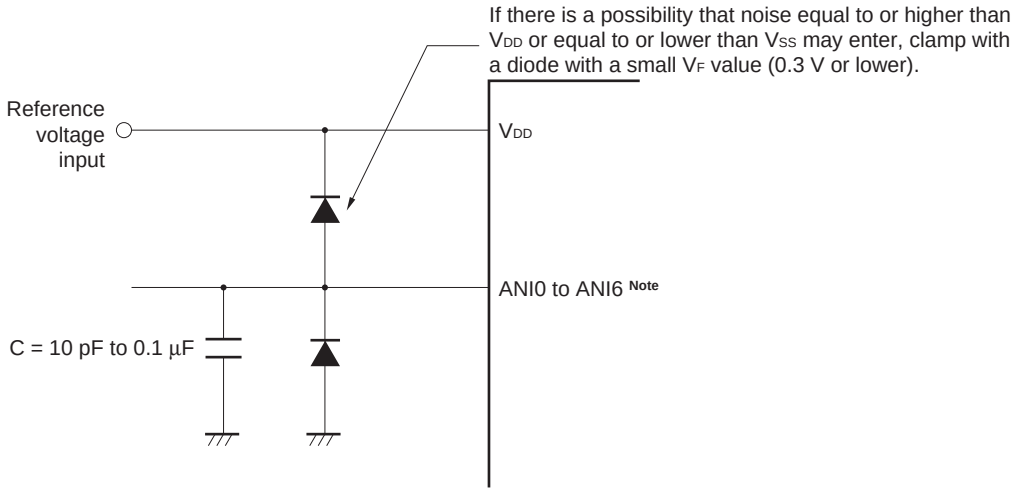
<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external C as shown in Figure 10-23 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Note For 10-pin products, ANI0 to ANI3.

Figure 10-23. Analog Input Pin Connection



Note For 10-pin products, ANI0 to ANI3.

10.9.5 Analog input (ANIn) pins

<1> The analog input pins (ANI0 to ANI6 Note 1) are also used as input port pins (P01 to P07 Note 2).
When A/D conversion is performed with any of the ANI0 to ANI6 pins selected, do not change output value to alternate port P01 to P07 Note 2 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to prevent such a pulse from being input or output.

Notes 1. For 10-pin products, ANI0 to ANI3.
2. P05 to P07 are provided only in the 16-pin products.

10.9.6 Input impedance of analog input (ANIn) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.

Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.

To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 1 kΩ, and to connect a capacitor of about 0.1 μF to the ANI0 to ANI6 Note pins (see **Figure 10-23**).

Note For 10-pin products, ANI0 to ANI3.

10.9.7 Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed. When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

10.9.8 Conversion results just after A/D conversion start

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 0.1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request signal (INTAD) and removing the first conversion result.

10.9.9 A/D conversion result register (ADCRH, ADCRL) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and port mode control register (PMC0), the contents of the ADCRH and ADCRL registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, or PMC0 register.

10.9.10 Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 10-24. Internal Equivalent Circuit of ANIn Pin

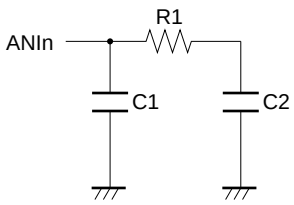


Table 10-4. Resistance and Capacitance Values of Equivalent Circuit

AV_{REFP}, V_{DD}	ANIn Pins	R1 [kΩ]	C1 [pF]	C2 [pF]
2.7 V ≤ V_{DD} ≤ 5.5 V	ANI0 to ANI6 Note	40	8	1.7
2.4 V ≤ V_{DD} < 2.7 V	ANI0 to ANI6 Note	200		

Note For 10-pin products, ANI0 to ANI3.

Remark The resistance and capacitance values shown in Table 10-4 are not guaranteed values.

10.9.11 Starting the A/D converter

The range of operating voltage for the A/D converter is from 2.4 to 5.5 V. Start the A/D converter after the V_{DD} voltage stabilizes.

CHAPTER 11 COMPARATOR

Caution The 16-pin product has one comparator channel.

11.1 Comparator Functions

The comparator integrates the following functions:

- The comparator response speed can be selected.
  High-speed mode: Decreased response delay time, with increased power consumption.
  Low-speed mode: Increased response delay time, with decreased power consumption.
- The comparator reference voltage can be either the externally input reference voltage or the internal reference voltage [Note] (0.815 V (typ.)).
- The integrated digital filter for noise elimination allows selecting the noise elimination width.
- The inverted/non-inverted comparator output can be output from the VCOUT0 pin.
- An interrupt (INTCMP0) can be generated upon detection of the effective edge of the comparator output signal.

Note The internal reference voltage cannot be used for the A/D converter and comparator simultaneously. When the internal reference voltage is selected as the reference voltage of the comparator, do not select the internal reference voltage as the target for conversion by the A/D converter.

11.2 Comparator Configuration

The comparator consists of the following hardware:

(1) IVCMP0 Pin

The comparator analog input pin. An analog signal to be compared by the comparator is input to this pin.

(2) IVREF0 Pin

A input pin to supply the reference voltage externally. The reference voltage of the comparator and analog input that is input to the IVCMP0 pin are compared.

In addition to the voltage supplied to the IVREF0 pin externally, the internal reference voltage (0.815 V (typ.)) can be selected for the comparator reference voltage.

For details on setting the comparator reference voltage, see **11.3.2 Comparator Mode Setting Register (COMPMDR)**.

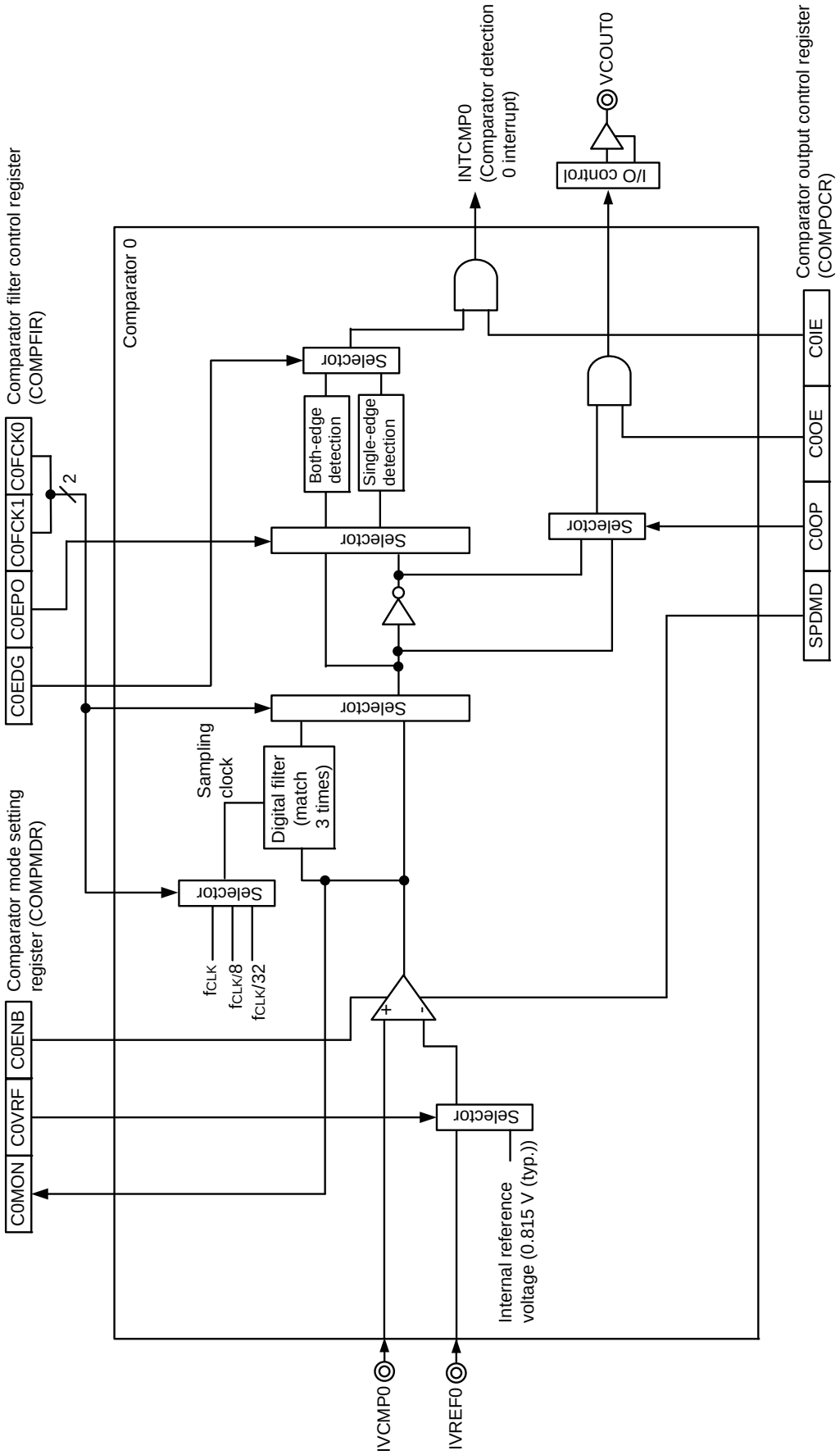
(3) VCOUT0 Pin

A pin to output the comparator comparison results. The inverted/non-inverted comparator output can be output from the VCOUT0 pin.

For use of the VCOUT0 pin as a comparator output, see **11.4.3 Comparator 0 Output**.

Figure 11-1 shows the block diagram of the comparator.

Figure 11-1. Block Diagram of the Comparator



11.3 Registers Controlling the Comparator

The following lists the registers to control the comparator.

- Peripheral enable register 0 (PER0)
- Comparator mode setting register (COMPMDR)
- Comparator filter control register (COMPFIR)
- Comparator output control register (COMPOCR)
- Port mode control register 0 (PMC0)
- Port mode register 0 (PM0)
- Port register 0 (P0)

11.3.1 Peripheral Enable Register 0 (PER0)

This register enables or disables clock supply to each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the comparator is used, be sure to set bit 5 (CMPEN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER0	TMKAEN Note	CMPEN Note	ADCEN	IICA0EN Note	0	SAU0EN	0	TAU0EN

CMPEN	Control of comparator input clock
0	Stops input clock supply. • SFR used by the comparator cannot be written. • The comparator is in the reset status.
1	Supplies input clock. • SFR used by the comparator can be read/written.

Note These bits are only available in the 16-pin products.

Cautions 1. When setting the comparator, be sure to set the CMPEN bit to 1 first before setting the registers shown below. If CMPEN = 0, control registers of the comparator are cleared to their initial values and writing to them is ignored (except for port mode register 0 (PM0), port register 0 (P0), and port mode control register 0 (PMC0)).

• Comparator mode setting register (COMPMDR)

• Comparator filter control register (COMPFIR)

• Comparator output control register (COMPOCR)

2. Be sure to clear the following bits to 0.

10-pin products: Bits 1, 3, 4, 6, and 7

16-pin products: Bits 1 and 3

11.3.2 Comparator Mode Setting Register (COMPMDR)

This register selects the comparator reference voltage, starts/stops the comparison operation, and indicates the comparison result state.

The COMPMDR register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that the C0MON bit can only be read.

Reset signal generation clears this register to 00H.

Figure 11-3. Format of Comparator Mode Setting Register (COMPMDR)

Address: FFF60H After reset: 00H R/W [Note 1]

Symbol	7	6	5	4	<3>	2	1	<0>
COMPMDR	0	0	0	0	C0MON	C0VRF	0	C0ENB

C0MON [Note 2]	Comparator 0 monitor flag
0	IVCMP0 < comparator 0 reference voltage
1	IVCMP0 > comparator 0 reference voltage

C0VRF	Comparator 0 reference voltage selection
0	Supplied from the IVREF0 pin.
1	Supplied from the internal reference voltage (0.815 V (typ.)) [Note 3]

C0ENB	Comparator 0 operation control
0	Comparator 0 operation disabled
1	Comparator 0 operation enabled

Notes
1. Bit 3 is a read-only bit.
2. After the comparator 0 operation is enabled (C0ENB = 1), the IVREF0 pin state can be read from the C0MON bit setting. When the comparator 0 operation is then disabled (C0ENB = 0), the C0MON bit value is undefined.
3. When the internal reference voltage (0.815 V (typ.)) is selected as the comparator 0 reference voltage, the internal reference voltage cannot be selected for the A/D converter.

11.3.3 Comparator Filter Control Register (COMPFIR)

This register selects the effective edge for the comparator interrupt signal, and enables or disables the digital filter.

If noise elimination is required, set the C0FCK1 and C0FCK0 bits so that the digital filter is enabled. When the digital filter is enabled, the comparator output is checked if its level remains the same for three consecutive digital filter sampling clock cycles.

The COMPFIR register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-4. Format of Comparator Filter Control Register (COMPFIR)

Address: FFF61H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
COMPFIR	0	0	0	0	C0EDG	C0EPO	C0FCK1	C0FCK0

C0EDG	C0EPO	Effective edge selection for comparator 0 interrupt signal Note 1
0	0	Rising edge
0	1	Falling edge
1	×	Both rising and falling edges

C0FCK1	C0FCK0	Comparator 0 digital filter enable/disable Notes 1, 2, 3
0	0	Digital filter disabled
0	1	Digital filter enabled, sampling clock: f_{CLK}
1	0	Digital filter enabled, sampling clock: $f_{CLK}/8$
1	1	Digital filter enabled, sampling clock: $f_{CLK}/32$

Notes 1. If bit C0EDG, C0EPO, C0FCK1, or C0FCK0 is changed while operation of the comparator 0 is enabled, a comparator 0 interrupt (INTCMP0) may be generated. Change these bits only after clearing the C0IE bit in the COMPOCR register to 0 to disable an interrupt request. Also, be sure to clear bit 2 (CMPIF0) in the interrupt request flag register 1L (IF1L) to 0 after changing these bits.

2. If bit C0FCK1 or C0FCK0 is changed, a wait period of four cycles of the sampling clock is required to update the digital filter. To use the comparator 0 interrupt (INTCMP0), set the C0IE bit in the COMPOCR register to 1 after this wait period.

3. To use the comparator in STOP mode, disable the digital filter (C0FCK1 and C0FCK0 = 00B).

Remark ×: Don't care

11.3.4 Comparator Output Control Register (COMPOCR)

This register selects the comparator response speed, controls the VCOUT0 output, and enables or disables the interrupt request signal.

The COMPOCR register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-5. Format of Comparator Output Control Register (COMPOCR)

Address: FFF62H After reset: 00H R/W

Symbol	<7>	6	5	4	3	<2>	<1>	<0>
COMPOCR	SPDMD	0	0	0	0	C0OP	C0OE	C0IE

SPDMD Note 1	Comparator speed selection
0	Low-speed mode
1	High-speed mode

C0OP	VCOUT0 output polarity selection
0	Non-inverted comparator 0 output is output from the VCOUT0 pin.
1	Inverted comparator 0 output is output from the VCOUT0 pin.

C0OE	VCOUT0 pin output enable/disable
0	Comparator 0 VCOUT0 pin output disabled
1	Comparator 0 VCOUT0 pin output enabled

C0IE	Comparator 0 interrupt request enable/disable
0	Comparator 0 interrupt request disabled
1	Comparator 0 interrupt request enabled

Note When rewriting the SPDMD bit, be sure to clear the C0ENB bit in the COMPMDR register to 0 in advance.

11.3.5 Registers Controlling Port Functions of Comparator I/O Pins

The port mode register (PM0), the port register (P0), and the port mode control register (PMC0) should be appropriately set to control the functions of the port pins that are also used for input and output of the comparator. For details, refer to **4.3.1 Port mode registers 0, 4 (PM0, PM4)**, **4.3.2 Port registers 0, 4, 12, 13 (P0, P4, P12, P13)**, and **4.3.5 Port mode control register 0 (PMC0).** For an example of settings when using a port pin for input and output of the comparator, see **4.5.3 Example of register settings for port and alternate functions used**.

When using the IVCMP0 and IVREF0 pins as analog inputs of the comparator, the appropriate bits should be set to 1 in the port mode register (PM0) and the port mode control register (PMC0) that correspond to each port.

When using the VCOUT0 pin as a comparator output, bits should be cleared to 0 in the port mode register (PM0), the port register (P0) and the port mode control register (PMC0). For details on the VCOUT0 pin setting, follow the setting procedure in **11.4.3 Comparator 0 Output.**

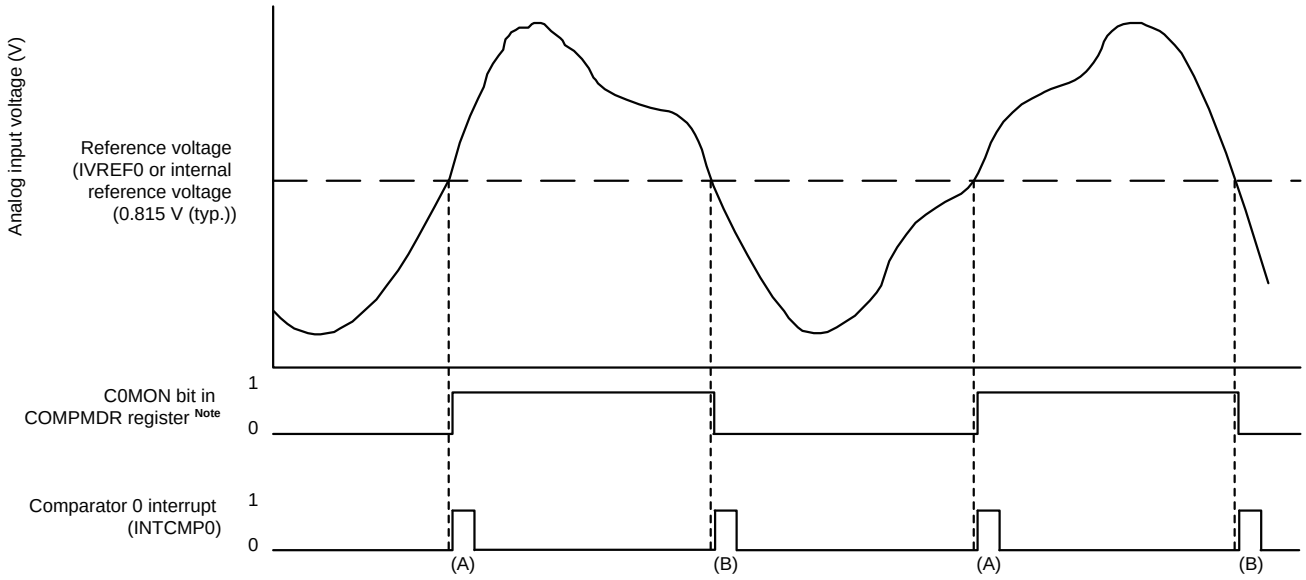
11.4 Comparator Operation

The C0MON bit in the COMPMDR register is set to 1 when the analog input voltage on the IVCMP0 pin is higher than the reference voltage. When lower, the C0MON bit is set to 0.

When using the comparator 0 interrupt (INTCMP0), set the C0IE bit in the COMPOCR register to 1 (interrupt request enabled). If the comparison result changes at this time, a comparator 0 interrupt request signal is generated. For details on the comparator 0 interrupt request, refer to **11.4.2 Comparator 0 Interrupt Operation**.

Figure 11-6 shows an example of the comparator 0 operation (no digital filter (C0FCK1 and C0FCK0 in COMPFIR = 00B), both-edge detection on an interrupt (C0EDG = 1).

Figure 11-6. Example of Comparator 0 Operation (No Digital Filter, Both-Edge Detection on Interrupt)



Note The output delay time depends on the comparator operating mode. For details, see **24.6.2 Comparator characteristics**.

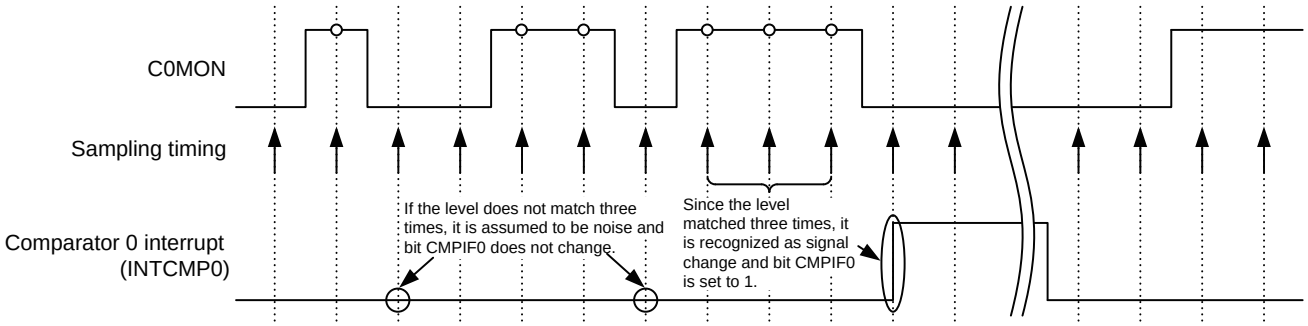
Caution When the rising edge is specified as an effective edge for an interrupt (C0EDG = 0 and C0EPO = 0), INTCMP0 only changes at (A).
When the falling edge is specified as an effective edge for an interrupt (C0EDG = 0 and C0EPO = 1), INTCMP0 only changes at (B).

11.4.1 Comparator 0 Digital Filter Operation

Comparator 0 incorporates a digital filter. The sampling clock is selected by bits C0FCK1 and C0FCK0 in the COMPFIR register. The comparator 0 output signal is sampled every sampling clock, and when the level of the output signal matches three times, the digital filter output changes at the next sampling clock.

Figure 11-7 shows the comparator 0 digital filter and interrupt operation example.

Figure 11-7. Comparator 0 Digital Filter and Interrupt Operation Example



Remark The operation example in Figure 11-7 applies when the digital filter is enabled (bits C0FCK1 and C0FCK0 in the COMPFIR register is 01B, 10B, or 11B).

11.4.2 Comparator 0 Interrupt Operation

When using the comparator 0 interrupt, set the C0IE bit in the COMPOCR register to 1 (interrupt request enabled). The condition for interrupt request generation can be set by the COMPFIR register. The comparator outputs can also be passed through the digital filter.

For details on the register setting, refer to **11.3.3 Comparator Filter Control Register (COMPFIR)** and **11.3.4 Comparator Output Control Register (COMPOCR)**.

11.4.3 Comparator 0 Output

The comparison result from the comparator can be output from the VCOUT0 pin. Bits C0OP and C0OE in the COMPOCR register are used to set the output polarity (inverted or non-inverted output) of the VCOUT0 pin and enable or disable the VCOUT0 pin output, respectively. For details on the register settings, refer to **11.3.4 Comparator Output Control Register (COMPOCR)**.

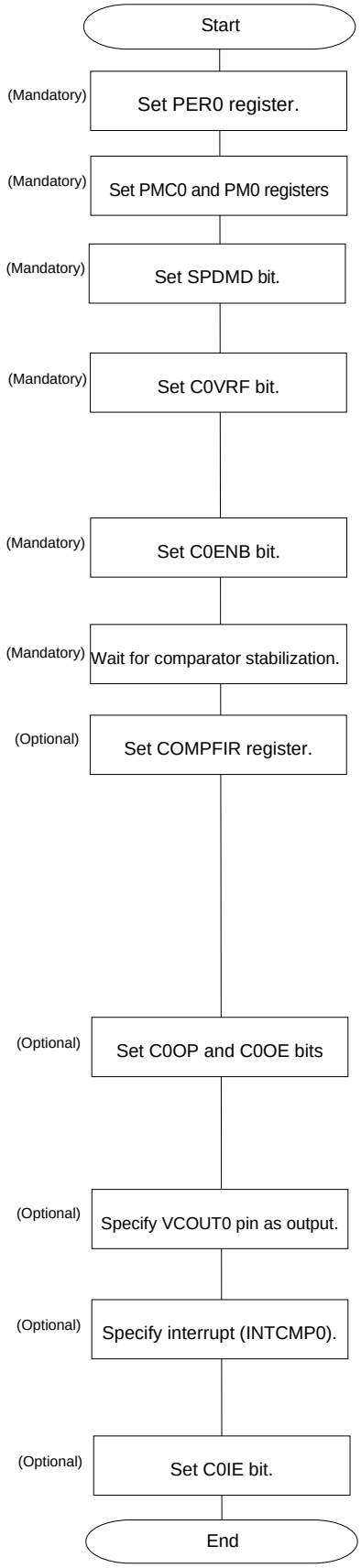
To output the comparator comparison result from the VCOUT0 pin, follow the procedure shown in Figure 11-8, Procedure for Enabling Comparator Operation.

11.5 Comparator Setting Flowchart

Figure 11-8 shows the comparator setting flowchart.

11.5.1 Enabling Comparator Operation

Figure 11-8. Procedure for Enabling Comparator Operation



Set bit CMPEN in PER0 to 1 to supply clock to comparator.

Set the ports for IVCMP0 and IVREF0 pins to analog input function.

- Set bits in PMC0 and PM0 to 1.

Specify comparator response speed.

- Set bit SPDMD in COMPOCR to 1: High-speed mode
- Clear bit SPDMD in COMPOCR to 0: Low-speed mode

Specify comparator reference voltage.

- Set bit C0VRF in COMPMDR to 1: Supplied from internal reference voltage (0.815 V (typ.))
- Clear bit C0VRF in COMPMDR to 0: Supplied from IVREF0 pin

Enable the comparator operation.

- Set bit C0ENB in COMPMDR to 1 to enable comparator 0 operation.

Count the comparator output stabilization signals (t_{CMP}).
For information on t_{CMP}, refer to **24.6.2 Comparator characteristics.**

Enable or disable the digital filter of the comparator and set the effective edge for an interrupt.

- Set bits C0FCK1 and C0FCK0 in COMPFIR:
  - 00B: Digital filter disabled
  - 01B: Digital filter enabled, sampling clock: f_{CLK}
  - 10B: Digital filter enabled, sampling clock: $f_{CLK}/8$
  - 11B: Digital filter enabled, sampling clock: $f_{CLK}/32$
- Set bits C0EPO and C0EDG in COMPFIR:
  - 00B: Rising edge
  - 01B: Falling edge
  - 1xB: Both rising and falling edges

Output comparator comparison result from VCOUT0 pin.

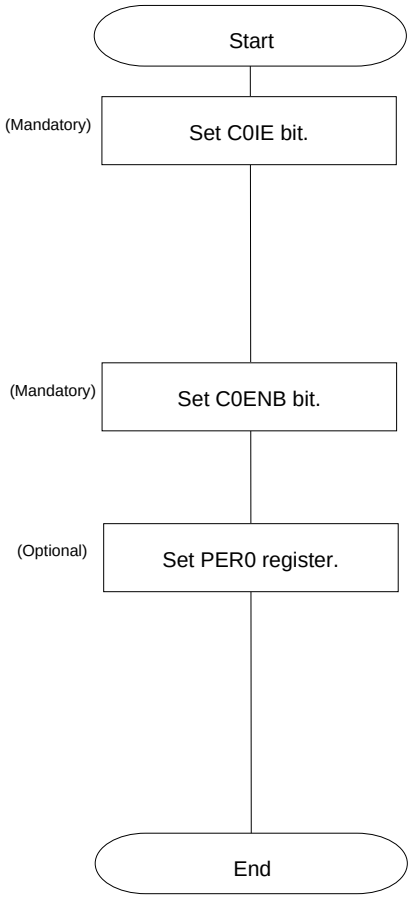
- Set bit C0OP in COMPOCR:
  - 0B: Non-inverted comparator result is output from VCOUT0 pin.
  - 1B: Inverted comparator result is output from VCOUT0 pin.
- Set bit C0OE in COMPOCR to 1 to enable output.
- Set the port for VCOUT0 pin to output function.
  - Set bits in PMC0 and PM0 to 1.
  - Set bits in P0 to 1.

Specify the comparator interrupt.

- Specify the interrupt priority level by bits CMPPR00 and CMPPR10 in PR1L.
- Initialize interrupt request by clearing bit CMPIF0 in IF1L to 0.
- Enable vector interrupt by clearing bit CMPMK0 in MK1L to 0.
- Enable interrupt request by setting bit C0IE in COMPOCR to 1.

11.5.2 Disabling Comparator Operation

Figure 11-9. Procedure for Disabling Comparator Operation



Disable comparator interrupt request.

- Clear bit C0IE in COMPOCR to 0 to disable interrupt request.

Clear comparator interrupt request flag.

- Clear bit CMPIF0 in IF1L to 0 to initialize interrupt request (unnecessary interrupt request cleared).

Disable comparator operation.

- Clear bit C0ENB in COMPMDR to 0 to disable comparator 0 operation.

Clear bit CMPEN in PER0 to 0 to stop clock supply to comparator.

Note that, stopping clock supply by so setting PER0 initializes all control registers of the comparator (except for port mode register 0 (PM0), port register 0 (P0), and port mode control register 0 (PMC0)).
To use the comparator again, set control registers according to Figure 11-8 Procedure for Enabling Comparator Operation.

CHAPTER 12 SERIAL ARRAY UNIT

Serial array unit 0 has maximum of two serial channels. Each channel can achieve simplified SPI (CSI [Note]), UART, and simplified I^2C communication.

Function assignment of each channel supported by the RL78/G10 is as shown below.

<R> **Note** Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

- 10-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	–		–

- 16-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		–

A single channel cannot be used under multiple communication methods. When a different communication method is to be configured, use another channel.

12.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/G10 has the following features.

12.1.1 Simplified SPI (CSI00, CSI01)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel. Simplified SPI communication is clocked communication performed by using three communication lines: one for the serial clock (SCK), one for transmitting serial data (SO), one for receiving serial data (SI).

For details about the settings, see **12.5 Operation of Simplified SPI (CSI00, CSI01) Communication.**

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate **Note**

  During master communication: Max. $f_{CLK}/4$

  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**.

12.1.2 UART (UART0)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

For details about the settings, see **12.6 Operation of UART (UART0) Communication**.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

The ISC register can be used to set up the input signal on the RxD0 pin of UART0 as an external interrupt input or as a timer input for the timer array unit. The input pulse interval measurement mode of the timer array unit can then be used to measure the width at the baud rate of the other party in communications and make the required adjustments in response.

12.1.3 Simplified I²C (IIC00)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see **12.7 Operation of Simplified I²C (IIC00) Communication.**

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function [Note 1] and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Overrun error
- ACK error

* [Functions not supported by simplified I²C [Note 2]]

- Slave transmission, slave reception
- Multi-master function (arbitration loss detection function)
- Clock stretch detection functions

Notes 1. When receiving the last data, 0 is written to the SOE00 bit of the serial output enable register 0 (SOE0) and serial communication data output is stopped, disabling ACK output. See **12.7.3 (2) Processing flow** for details.

2. Full I²C functions are described in **CHAPTER 13 SERIAL INTERFACE IICA** (16-pin products only).

12.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 12-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits
Buffer register	Serial data register 0nL (SDR0nL [Note 2])
Serial clock I/O	SCK00 and SCK01 [Note1] pins (for simplified SPI), SCL00 pin (for simplified I^2C)
Serial data input	SI00 and SI01 [Note1] pins (for simplified SPI), RxD0 pin (for UART)
Serial data output	SO00 and SO01 [Note1] pins (for simplified SPI), TxD0 pin (for UART)
Serial data I/O	SDA00 pin (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register 0 (SPS0) • Serial channel enable status register 0 (SE0) • Serial channel start register 0 (SS0) • Serial channel stop register 0 (ST0) • Serial output enable register 0 (SOE0) • Serial output register 0 (SO0) • Serial clock output register 0 (CKO0) • Serial output level register 0 (SOL0) • Noise filter enable register 0 (NFEN0) • Input switch control register (ISC)
	<Registers of each channel> • Serial data register 0n (SDR0nH, SDR0nL[Note2]) • Serial mode register 0n (SMR0nH, SMR0nL) • Serial communication operation setting register 0n (SCR0n) • Serial status register 0n (SSR0n) • Serial flag clear trigger register 0n (SIR0n)
	<Registers of port function block> • Port output mode register 0 (POM0) • Port mode control register 0 (PMC0) • Port mode register 0 (PM0) • Port register 0 (P0)

Notes 1. Only 16-pin products handle CSI01 transfer.

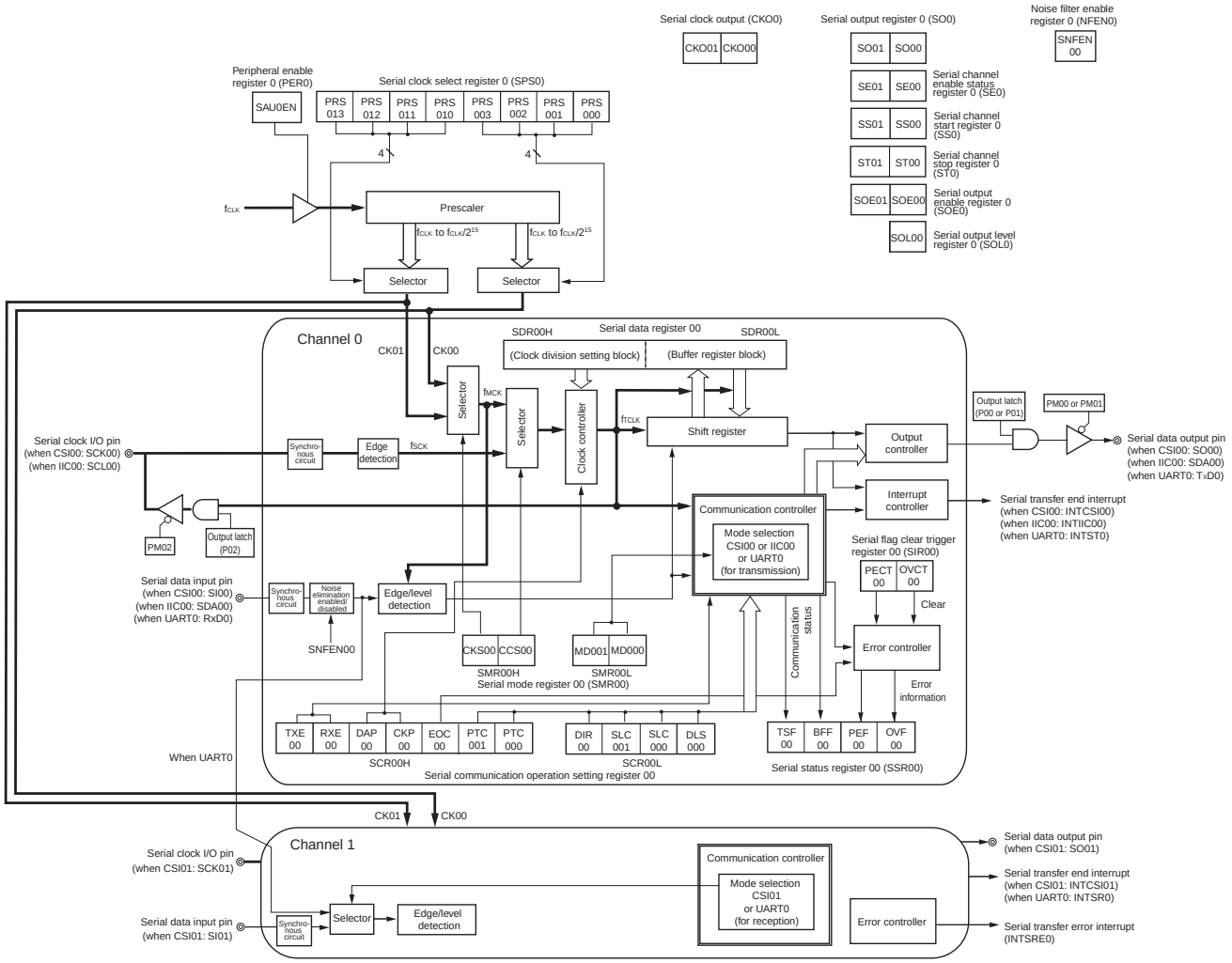
2. The serial data register 0nL (SDR0nL) can be read or written as the following SFR, depending on the communication mode.

- During CSIp communication: SIOp (CSIp data register)
- During UART0 reception: RXD0 (UART0 receive data register)
- During UART0 transmission: TXD0 (UART0 transmit data register)
- During IIC0 communication: SIO0 (IIC0 data register)

Remark n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), q: UART number (q = 0)

Figure 12-1 shows the block diagram of the serial array unit 0.

Figure 12-1. Block Diagram of Serial Array Unit 0



Note Only 16-pin products handle CSI01 transfer.

12.2.1 Shift register

This is a 9-bit register that converts parallel data into serial data or vice versa.

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write to the shift register, use the lower 8 bits of serial data register 0nL (SDR0nL).

	8	7	6	5	4	3	2	1	0
Shift register									

12.2.2 Serial data register 0nL (SDR0nL)

The SDR0nL register is used as a transmit/receive buffer register of channel n.

When data is received, parallel data converted by the shift register is stored in the SDR0nL register. When data is to be transmitted, set transmit data to be transferred to the shift register in the SDR0nL register.

The length of data stored in the SDR0nL register is as follows, depending on the setting of bit 0 (DLS0n0) of serial communication operation setting register 0n (SCR0nL), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDR0nL register)
- 8-bit data length (stored in bits 0 to 7 of SDR0nL register)

The SDR0nL register is set by an 8-bit memory manipulation instruction when the operation is enabled (SE0n = 1).

Writing to the SDR0nL register is prohibited when the operation is stopped (SE0n = 0).

Reset signal generation clears the SDR0nL register to 00H.

Eight-bit memory manipulation instructions are used to set the SDR0nL register as a buffer for an SFR listed below, in accord with the communications protocol in use.

- During CSIp communication: SIOp (CSIp data register)
- During UART0 reception: RXD0 (UART0 receive data register)
- During UART0 transmission: TXD0 (UART0 transmit data register)
- During IIC00 communication: SIO00 (IIC00 data register)

Remark n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-2. Format of Serial Data Register 0nL (SDR0nL) (n = 0, 1)

Address: FFF10H (SDR00L), FFF12H (SDR01L) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SDR0nL								

Remark For the function of the SDR0nH register, see **12.3 Registers Controlling Serial Array Unit**.

12.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Serial clock select register 0 (SPS0)
- Serial mode register 0n (SMR0nH, SMR0nL)
- Serial communication operation setting register 0n (SCR0nH, SCR0nL)
- Serial data register 0n (SDR0nH, SDR0nL)
- Serial flag clear trigger register 0n (SIR0n)
- Serial status register 0n (SSR0n)
- Serial channel start register 0 (SS0)
- Serial channel stop register 0 (ST0)
- Serial channel enable status register 0 (SE0)
- Serial output enable register 0 (SOE0)
- Serial output level register 0 (SOL0)
- Serial output register 0 (SO0)
- Serial clock output register 0 (CKO0)
- Noise filter enable register 0 (NFEN0)
- Input switch control register (ISC)
- Port output mode register 0 (POM0)
- Port mode control register 0 (PMC0)
- Port mode register 0 (PM0)
- Port register 0 (P0)

Remark n: Channel number (n = 0, 1)

12.3.1 Peripheral enable register 0 (PER0)

PER0 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PER0 register to 00H.

Figure 12-3. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER0	TMKAEN[Note]	CMPEN[Note]	ADCEN	IICA0EN[Note]	0	SAU0EN	0	TAU0EN

SAU0EN	Control of serial array unit 0 input clock supply
0	Stops supply of input clock. • SFR used by serial array unit 0 cannot be written. • Serial array unit 0 is in the reset status.
1	Enables input clock supply. • SFR used by serial array unit 0 can be read/written.

Note 16-pin products only.

Cautions 1. When setting serial array unit 0, be sure to set the following registers while the SAU0EN bit is set to 1 first. If SAU0EN = 0, control registers of serial array unit 0 become default values and writing to them is ignored (except for the noise filter enable register 0 (NFEN0), input switch control register (ISC), port output mode register 0 (POM0), port mode register 0 (PM0), port mode control register 0 (PMC0), and port register 0 (P0)).

- **Serial clock select register 0 (SPS0)**
- **Serial mode register 0n (SMR0nH, SMR0nL)**
- **Serial communication operation setting register 0n (SCR0nH, SCR0nL)**
- **Serial data register 0n (SDR0nH, SDR0nL)**
- **Serial flag clear trigger register 0n (SIR0n)**
- **Serial status register 0n (SSR0n)**
- **Serial channel start register 0 (SS0)**
- **Serial channel stop register 0 (ST0)**
- **Serial channel enable status register 0 (SE0)**
- **Serial output enable register 0 (SOE0)**
- **Serial output level register 0 (SOL0)**
- **Serial output register 0 (SO0)**
- **Serial clock output register 0 (CKO0)**

2. Be sure to clear the following bits to 0.
10-pin products: Bits 1, 3, 4, 6, and 7
16-pin products: Bits 1 and 3

12.3.2 Serial clock select register 0 (SPS0)

The SPS0 register is an 8-bit register that is used to select two types of operation clocks (CK00, CK01) that are commonly supplied to each channel. CK01 is selected by bits 7 to 4 of the SPS0 register, and CK00 is selected by bits 3 to 0.

Rewriting the SPS0 register is prohibited when the operation is enabled (when SE0n = 1).

The SPS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the SPS0 register to 00H.

Figure 12-4. Format of Serial Clock Select Register 0 (SPS0)

Address: F0126H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SPS0	PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000

PRS 0n3	PRS 0n2	PRS 0n1	PRS 0n0	Section of operation clock (CKn) Note					
					f_{CLK} = 1.25 MHz	f_{CLK} = 2.5 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	1.25 MHz	2.5 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	625 kHz	1.25 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	313 kHz	625 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	156 kHz	313 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	78 kHz	156 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	39 kHz	78 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	19.5 kHz	39 kHz	78 kHz	156 kHz	313 kHz
0	1	1	1	$f_{CLK}/2^7$	9.8 kHz	19.5 kHz	39 kHz	78 kHz	156 kHz
1	0	0	0	$f_{CLK}/2^8$	4.9 kHz	9.8 kHz	19.5 kHz	39 kHz	78 kHz
1	0	0	1	$f_{CLK}/2^9$	2.5 kHz	4.9 kHz	9.8 kHz	19.5 kHz	39 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.22 kHz	2.5 kHz	4.9 kHz	9.8 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	625 Hz	1.22 kHz	2.5 kHz	4.9 kHz	9.8 kHz
1	1	0	0	$f_{CLK}/2^{12}$	313 Hz	625 Hz	1.22 kHz	2.5 kHz	4.9 kHz
1	1	0	1	$f_{CLK}/2^{13}$	152 Hz	313 Hz	625 Hz	1.22 kHz	2.5 kHz
1	1	1	0	$f_{CLK}/2^{14}$	78 Hz	152 Hz	313 Hz	625 Hz	1.22 kHz
1	1	1	1	$f_{CLK}/2^{15}$	39 Hz	78 Hz	152 Hz	313 Hz	625 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register 0 (ST0) = 03H) the operation of the serial array unit (SAU).

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. n: Channel number (n = 0, 1)

12.3.3 Serial mode register 0n (SMR0nH, SMR0nL)

The SMR0nH and SMR0nL registers are registers that set an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, an operation mode (Simplified SPI (CSI), UART, or I^2C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMR0nH and SMR0nL registers is prohibited when the operation is enabled (when SE0n = 1). However, the MD0n0 bit can be rewritten even when the operation is enabled.

The SMR0nH and SMR0nL registers can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets the SMR0nH and SMR0nL registers to 00H and 20H, respectively.

Figure 12-5. Format of Serial Mode Register 0n (SMR0nH, SMR0nL) (1/2)

Address: F0111H (SMR00H), F0113H (SMR01H)
After reset: 00H R/W
Symbol: SMR0nH

15	14	13	12	11	10	9	8
CKS0n	CCS0n	0	0	0	0	0	STS0n Note 1

Address: F0110H (SMR00L), F0112H (SMR01L)
After reset: 20H R/W
Symbol: SMR0nL

7	6	5	4	3	2	1	0
0	SIS0n0 Note 2	1	0	0	MD0n2	MD0n1	MD0n0

CKS0n	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CK00 set by the SPS0 register
1	Operation clock CK01 set by the SPS0 register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCS0n bit and the SDR0nH register, a transfer clock (f_{TCLK}) is generated.	

CCS0n	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKS0n bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in simplified SPI (CSI) mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCS0n = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDR0nH register.	

STS0n Note 1	Selection of start trigger source
0	Only software trigger is valid (selected for simplified SPI (CSI), UART transmission, and simplified I^2C).
1	Valid edge of the R_xD0 pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SS0 register.	

Notes 1. Provided in the SMR01H register only.
2. Provided in the SMR01L register only.

Caution Do not change the initial values of the following bits.
SMR00H: Be sure to clear bits 0 to 5 to 0.
SMR01H: Be sure to clear bits 1 to 5 to 0.
SMR00L: Be sure to clear bits 3, 4, 6, and 7 to 0, and set bit 5 to 1.
SMR01L: Be sure to clear bits 3, 4, and 7 to 0, and set bit 5 to 1.

Remark n: Channel number (n = 0, 1)

Figure 12-5. Format of Serial Mode Register 0n (SMR0nH, SMR0nL) (2/2)

Address: F0111H (SMR00H), F0113H (SMR01H)
After reset: 00H R/W
Symbol: SMR0nH

7	6	5	4	3	2	1	0
CKS 0n	CCS 0n	0	0	0	0	0	STS 0n Note 1

Address: F0110H (SMR00L), F0112H (SMR01L)
After reset: 20H R/W
Symbol: SMR0nL

7	6	5	4	3	2	1	0
0	SIS 0n0 Note 2	1	0	0	MD 0n2	MD 0n1	MD 0n0

SIS0n0 Note 2	Controls inversion of level of receive data of UART0
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MD0n2	MD0n1	Setting of operation mode of channel n
0	0	Simplified SPI (CSI) mode
0	1	UART mode
1	0	Simplified I^2C mode
1	1	Setting prohibited

MD0n0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDR0nL register to the shift register.)
For successive transmission, the next transmit data is written by setting the MD0n0 bit to 1 when SDR0nL data has run out.	

Notes 1. Provided in the SMR01H register only.
2. Provided in the SMR01L register only.

Caution Do not change the initial values of the following bits.
SMR00H: Be sure to clear bits 0 to 5 to 0.
SMR01H: Be sure to clear bits 1 to 5 to 0.
SMR00L: Be sure to clear bits 3, 4, 6, and 7 to 0, and set bit 5 to 1.
SMR01L: Be sure to clear bits 3, 4, and 7 to 0, and set bit 5 to 1.

Remark n: Channel number (n = 0, 1)

12.3.4 Serial communication operation setting register 0n (SCR0nH, SCR0nL)

The SCR0nH and SCR0nL registers are communication operation setting registers of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCR0nH and SCR0nL registers is prohibited when the operation is enabled (when SE0n = 1).

The SCR0nH and SCR0nL registers can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets the SCR0nH and SCR0nL registers to 00H and 87H, respectively.

Figure 12-6. Format of Serial Communication Operation Setting Register 0n (SCR0nH, SCR0nL) (1/2)

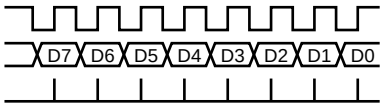
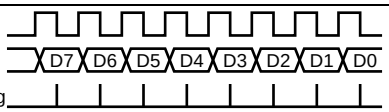
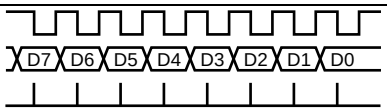
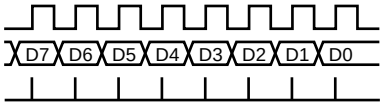
Address: F0119H (SCR00H) , F011BH (SCR01H)
After reset: 00H R/W
Symbol: SCR0nH

7	6	5	4	3	2	1	0
TXE 0n	RXE 0n	DAP 0n	CKP 0n	0	EOC 0n	PTC 0n1	PTC 0n0

Address: F0118H (SCR00L) , F011AH (SCR01L)
After reset: 87H R/W
Symbol: SCR0nL

7	6	5	4	3	2	1	0
DIR 0n	0	SLC 0n1 Note 1	SLC 0n0	0	1	1	DLS 0n0

TXE0n	RXE0n	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP0n	CKP0n	Selection of data and clock phase in simplified SPI (CSI) mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Be sure to set DAP0n, CKP0n = 0, 0 in the UART mode and simplified I²C mode.			

EOC0n	Selection of masking of error interrupt signal (INTSREx (x = 0, 1))
0	Disables generation of error interrupt INTSREx (INTSRx is generated).
1	Enables generation of error interrupt INTSREx (INTSRx is not generated if an error occurs).
Set EOC0n = 0 in the simplified SPI (CSI) mode, simplified I²C mode, and during UART transmission Note 2.	

Notes 1. Provided in the SCR00L register only.

2. If EOC0n is not cleared for CSI0n, error interrupt INTSREn may be generated.

(**Caution** and **Remark** are listed on the next page.)

Figure 12-6. Format of Serial Communication Operation Setting Register 0n (SCR0nH, SCR0nL) (2/2)

Address: F0119H (SCR00H) , F011BH (SCR01H)

After reset: 00H R/W

Symbol: SCR0nH

7	6	5	4	3	2	1	0
TXE 0n	RXE 0n	DAP 0n	CKP 0n	0	EOC 0n	PTC 0n1	PTC 0n0

Address: F0118H (SCR00L) , F011AH (SCR01L)

After reset: 87H R/W

Symbol: SCR0nL

7	6	5	4	3	2	1	0
DIR 0n	0	SLC 0n1 Note 1	SLC 0n0	0	1	1	DLS 0n0

PTC0n1	PTC0n0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity Note 2.	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTC0n1, PTC0n0 = 0, 0 in the simplified SPI (CSI) mode and simplified I^2C mode.			

DIR0n	Selection of data transfer sequence in simplified SPI (CSI) and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIR0n = 0 in the simplified I^2C mode.	

SLC0n1 Note 1	SLC0n0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits (n = 0 only)
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set the stop bit length to 1 bit (SLC0n1, SLC0n0 = 0, 1) during UART reception and in the simplified I^2C mode. Set no stop bit (SLC0n1, SLC0n0 = 0, 0) in the simplified SPI (CSI) mode. Set the stop bit length to 1 bit (SLC0n1, SLC0n0 = 0, 1) or 2 bits (SLC0n1, SLC0n0 = 1, 0) during UART transmission.		

DLS0n0	Setting of data length in simplified SPI (CSI) and UART modes
0	7-bit data length (stored in bits 0 to 6 of the SDR0nL register)
1	8-bit data length (stored in bits 0 to 7 of the SDR0nL register)
Be sure to set DLS0n0 = 1 in the simplified I^2C mode.	

Notes 1. Provided in the SCR00L register only.

2. 0 is always added regardless of the data contents.

Caution Do not change the initial values of the following bits.

SCR0nH: Be sure to clear bit 3 to 0.

SCR00L: Be sure to clear bits 3 and 6 to 0, and set bits 1 and 2 to 1.

SCR01L: Be sure to clear bits 3, 5, and 6 to 0, and set bits 1 and 2 to 1.

Remark n: Channel number (n = 0, 1)

12.3.5 Serial data register 0n (SDR0nH, SDR0nL)

The SDR0nH and SDR0nL registers are the transmit/receive data registers of channel n.

The SDR0nH and SDR0nL registers are set by an 8-bit memory manipulation instruction.

Reset signal generation clears the SDR0nH and SDR0nL registers to 00H.

The SDR0nH register is used as a register that sets the division ratio of the operating clock (f_{MCK}).

If the CCS0n bit of the SMR0nH register is cleared to 0, the clock set by dividing the operating clock by the SDR0nH register is used as the transfer clock.

If the CCS0n bit of the SMR0nH register is set to 1, set the SDR0nH register to 00000000B. The input clock f_{SCK} from the SCKp pin (slave transmission in the simplified SPI (CSI) mode) is used as the transfer clock.

The SDR0nH register is set by an 8-bit memory manipulation instruction when the operation is stopped (SE0n = 0). Writing to the SDR0nH register is ignored when the operation is enabled (SE0n = 1), and 0 is always read from the SDR0nH register.

The SDR0nL register functions as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the SDR0nL register. During transmission, the data to be transmitted to the shift register is set to the SDR0nL register.

The SDR0nL register is set by an 8-bit memory manipulation instruction when the operation is enabled (SE0n = 1). Writing to the SDR0nL register is prohibited when the operation is stopped (SE0n = 0).

Figure 12-7. Format of Serial Data Register 0n (SDR0n)

Address: FFF11H (SDR00H) , FFF13H (SDR01H)
After reset: 00H R/W

Division ratio setting register

Symbol	7	6	5	4	3	2	1	0
SDR0nH								0

Address: FFF10H (SDR00L) , FFF12H (SDR01L)
After reset: 00H R/W

Transmit/receive buffer register

Symbol	7	6	5	4	3	2	1	0
SDR0nL								

SDR0nH[7:1]							Transfer clock setting by dividing the operating clock (f_{MCK})
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

Cautions 1. Setting SDR0nH[7:1] = (0000000B, 0000001B) is prohibited when UART is used.
2. Setting SDR0nH[7:1] = 0000000B is prohibited when simplified I^2C is used.

Remarks 1. For the function of the SDR0nL register, see **12.2 Configuration of Serial Array Unit**.
2. n: Channel number (n = 0, 1)

12.3.6 Serial flag clear trigger register 0n (SIR0n)

The SIR0n register is a trigger register that is used to clear each error flag of channel n.

When each bit (FECT0n, PECT0n, OVCT0n) of this register is set to 1, the corresponding bit (FEF0n, PEF0n, OVF0n) of serial status register 0n (SSR0n) is cleared to 0. Because the SIR0n register is a trigger register, it is cleared immediately when the corresponding bit of the SSR0n register is cleared.

The SIR0n register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the SIR0n register to 00H.

Figure 12-8. Format of Serial Flag Clear Trigger Register 0n (SIR0n)

Address: F0108H (SIR00) , F010AH (SIR01) , After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SIR0n	0	0	0	0	0	FECT0n[Note]	PECT0n	OVCT0n

FECT0n[Note]	Clear trigger of framing error flag of channel n
0	Not cleared
1	Clears the FEF0n bit of the SSR0n register to 0.

PECT0n	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEF0n bit of the SSR0n register to 0.

OVCT0n	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVF0n bit of the SSR0n register to 0.

Note Provided in the SIR01 register only.

Caution Be sure to clear the following bits to 0.
SIR00 register: Bits 2 to 7
SIR01 register: Bits 3 to 7

Remarks 1. n: Channel number (n = 0, 1)
2. When the SIR0n register is read, 00H is always read.

12.3.7 Serial status register 0n (SSR0n)

The SSR0n register indicates the communication status and error occurrence status of channel n. The errors indicated by this register are framing errors, parity errors, and overrun errors.

The SSR0n register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears the SSR0n register to 00H.

Figure 12-9. Format of Serial Status Register 0n (SSR0n) (1/2)

Address: F0100H (SSR00) , F0102H (SSR01) , After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
SSR0n	0	TSF0n	BFF0n	0	0	FEF0n Note	PEF0n	OVF0n

TSF0n	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The ST0n bit of the ST0 register is set to 1 (communication is stopped) or the SS0n bit of the SS0 register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFF0n	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDR0nL register.
1	Valid data is stored in the SDR0nL register.
<Clear conditions> • Transferring transmit data from the SDR0nL register to the shift register ends during transmission. • Reading receive data from the SDR0nL register ends during reception. • The ST0n bit of the ST0 register is set to 1 (communication is stopped) or the SS0n bit of the SS0 register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDR0nL register while the TXE0n bit of the SCR0nH register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDR0nL register while the RXE0n bit of the SCR0nH register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Note Provided in the SSR01 register only.

Caution **If data is written to the SDR0nL register when BFF0n = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVF0n = 1) is detected.**

Remark n: Channel number (n = 0, 1)

Figure 12-9. Format of Serial Status Register 0n (SSR0n) (2/2)

Address: F0100H (SSR00), F0102H (SSR01) , After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
SSR0n	0	TSF0n	BFF0n	0	0	FEF0n Note	PEF0n	OVF0n

FEF0n Note	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear condition> • 1 is written to the FECT0n bit of the SIR0n register. <Set condition> • A stop bit is not detected when UART reception ends.	

PEF0n	Parity / ACK error detection flag of channel n
0	No error occurs.
1	Parity error occurs (during UART reception) or ACK is not detected (during I^2C transmission).
<Clear condition> • 1 is written to the PECT0n bit of the SIR0n register. <Set condition> • The parity of the transmit data and the parity bit do not match when UART reception ends (parity error). • No ACK signal is returned from the slave channel at the ACK reception timing during I^2C transmission (ACK is not detected).	

OVF0n	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> • 1 is written to the OVCT0n bit of the SIR0n register. <Set condition> • Even though receive data is stored in the SDR0nL register, that data is not read and transmit data or the next receive data is written while the RXE0n bit of the SCR0nH register is set to 1 (reception or transmission and reception mode in each communication mode). • Transmit data is not ready for slave transmission or transmission and reception in simplified SPI (CSI) mode.	

Note Provided in the SSR01 register only.

Remark n: Channel number (n = 0, 1)

12.3.8 Serial channel start register 0 (SS0)

The SS0 register is a trigger register that is used to enable communication/count for each channel.

When 1 is written to a bit of this register (SS0n), the corresponding bit (SE0n) of serial channel enable status register 0 (SE0) is set to 1 (operation is enabled). Because the SS0n bit is a trigger bit, it is cleared immediately when SE0n = 1.

The SS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the SS0 register to 00H.

Figure 12-10. Format of Serial Channel Start Register 0 (SS0)

Address: F0122H (SS0) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01	SS00

SS0n	Operation start trigger of channel n
0	No trigger operation
1	Sets the SE0n bit to 1 and enters the communication wait status Note.

Note If SS0n is set to 1 during transfer operations, transfer stops and the interface enters the state of waiting. At this time, the control registers and the shift register, the SCK0n and SO0n pins, and the FEF0n, PEF0n, and OVF0n flags retain their values.

Cautions
1. **Be sure to clear bits 2 to 7 to 0.**
2. **For the UART reception, set the RXE0n bit of SCR0nH register to 1, and then be sure to set SS0n to 1 after 4 or more f_{MCK} clocks have elapsed.**

Remarks
1. n: Channel number (n = 0, 1)
2. When the SS0 register is read, 00H is always read.

12.3.9 Serial channel stop register 0 (ST0)

The ST0 register is a trigger register that is used to enable stopping communication/count for each channel.

When 1 is written to a bit of this register (ST0n), the corresponding bit (SE0n) of serial channel enable status register 0 (SE0) is cleared to 0 (operation is stopped). Because the ST0n bit is a trigger bit, it is cleared immediately when SE0n = 0.

The ST0 register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the ST0 register to 00H.

Figure 12-11. Format of Serial Channel Stop Register 0 (ST0)

Address: F0124H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	ST01	ST00

ST0n	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SE0n bit to 0 and stops the communication operation Note

Note The control registers and the shift register, the SCK0n and SO0n pins, and the FEF0n, PEF0n, and OVF0n flags retain their values.

Caution **Be sure to clear bits 2 to 7 to 0.**

Remarks **1.** n: Channel number (n = 0, 1)

2. When the ST0 register is read, 00H is always read.

12.3.10 Serial channel enable status register 0 (SE0)

The SE0 register indicates whether the data transmission/reception operation of each channel is enabled or disabled.

When 1 is written to a bit of serial channel start register 0 (SS0), the corresponding bit of this register is set to 1. When 1 is written to a bit of serial channel stop register 0 (ST0), the corresponding bit of this register is cleared to 0.

If the operation of channel n is enabled, the value of the CKO0n bit (serial clock output of channel n) of serial output register 0 (SO0) cannot be rewritten by software, and a value is output from the serial clock pin according to the communication operation.

If the operation of channel n is disabled, the value of the CKO0n bit of the SO0 register can be set by software and its value is output from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SE0 register can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the SE0 register to 00H.

Figure 12-12. Format of Serial Channel Enable Status Register 0 (SE0)

Address: F0120H (SE0) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	SE01	SE00

SE0n	Indication of operation enable/disable status of channel n
0	Operation is disabled (stopped)
1	Operation is enabled.

Caution Be sure to clear bits 2 to 7 to 0.

Remark n: Channel number (n = 0, 1)

12.3.11 Serial output enable register 0 (SOE0)

The SOE0 register is used to enable or disable output of the serial communication operation of each channel.

If serial output is enabled for channel n, the value of the SO0n bit of serial output register 0 (SO0) cannot be rewritten by software, and a value is output from the serial data output pin according to the communication operation.

If serial output is disabled for channel n, the SO0n bit value of the SO0 register can be set by software, and its value is output from the serial data output pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SOE0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the SOE0 register to 00H.

Figure 12-13. Format of Serial Output Enable Register 0 (SOE0)

Address: F012AH (SOE0) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 Note	SOE00

SOE0n	Serial output enable/disable of channel n
0	Disables output by serial communication operation.
1	Enables output by serial communication operation.

Note 16-pin products only.

Caution Be sure to clear the following bits to 0.
10-pin products: Bits 1 to 7
16-pin products: Bits 2 to 7

Remark n: Channel number (n = 0, 1)

12.3.12 Serial output register 0 (SO0)

The SO0 register is a buffer register for serial output of each channel.

The value of the SO0n bit of this register is output from the serial data output pin of channel n.

The SO0n bit of this register can be rewritten by software only when serial output is disabled (SOE0n = 0). When serial output is enabled (SOE0n = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding SO0n bit to 1.

The SO0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the SO0 register to 03H.

Figure 12-14. Format of Serial Output Register 0 (SO0)

Address: F0128H (SO0) After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	SO01 Note	SO00

SO0n	Serial data output of channel n
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Note 16-pin products only.

Caution **For 10-pin products, be sure to set bit 1 to 1 and clear bits 2 to 7 to 0.**
For 16-pin products, be sure to clear bits 2 to 7 to 0.

Remark n: Channel number (n = 0, 1)

12.3.13 Serial clock output register 0 (CKO0)

The CKO0 register is a buffer register for serial clock output of each channel.

The value of the CKO0n bit of this register is output from the serial clock output pin of channel n.

The CKO0n bit of this register can be rewritten by software only when channel operation is disabled (SE0n = 0). When channel operation is enabled (SE0n = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding CKO0n bit to 1.

The CKO0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the CKO0 register to 03H.

Figure 12-15. Format of Serial Clock Output Register 0 (CKO0)

Address: F0129H (CKO0) After reset: 03H R/W

Symbol	7	6	5	4	3	2	1	0
CKO0	0	0	0	0	0	0	CKO01 Note	CKO00

CKO0n	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

Note 16-pin products only.

Caution **For 10-pin products, be sure to set bit 1 to 1 and clear bits 2 to 7 to 0.**
For 16-pin products, be sure to clear bits 2 to 7 to 0.

Remark n: Channel number (n = 0, 1)

12.3.14 Serial output level register 0 (SOL0)

The SOL0 register is used to set inversion of the data output level of channel 0.

This register can be set only in the UART mode. Be sure to set 0 to corresponding bit in the simplified SPI (CSI) mode and simplified I^2C mode.

Inverting channel 0 by using this register is reflected on pin output only when serial output is enabled (SOE00 = 1). When serial output is disabled (SOE00 = 0), the value of the SO00 bit is output as is.

Rewriting the SOL0 register is prohibited when the operation is enabled (SE00 = 1).

The SOL0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears the SOL0 register to 00H.

Figure 12-16. Format of Serial Output Level Register 0 (SOL0)

Address: F0134H (SOL0) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	SOL00

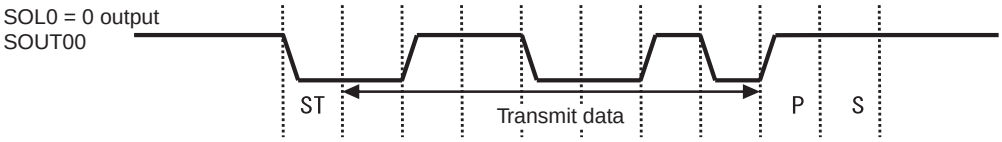
SOL00	Selects inversion of the level of the transmit data of channel 0 in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Be sure to clear bits 1 to 7 to 0.

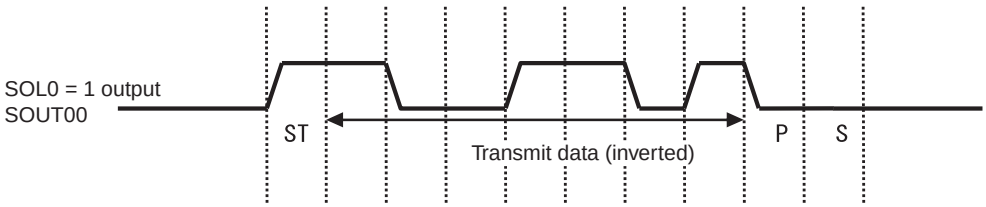
Figure 12-17 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 12-17. Examples of Reverse Transmit Data

(a) Non-reverse Output (SOL00 = 0)



(b) Reverse Output (SOL00 = 1)



12.3.15 Noise filter enable register 0 (NFEN0)

The NFEN0 register is used to set whether the noise filter can be used for the input signal from the serial data input pin of UART.

Disable the noise filter of the pin used for simplified SPI (CSI) or simplified I^2C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected.

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for the target channel.

The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN0 register to 00H.

Figure 12-18. Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0070H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	0	0	SNFEN00

SNFEN00	Use of noise filter of RxD0 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to use the RxD0 pin. Clear the SNFEN00 bit to 0 to use other than the RxD0 pin.	

Caution Be sure to clear bits 1 to 7 to 0.

12.3.16 Input switch control register (ISC)

The ISC1 and ISC0 bits in the ISC register are used to handle the combination of the external interrupt and the timer array unit at the time of baud rate correction of UART0.

When bit 0 is set to 1, the input signal on the serial data input (RxD0) pin is input to the external interrupt input (INTP0), making detection of the input edge signal of the start bit in the form of the INTP0 interrupt possible.

When bit 1 is set to 1, the input signal on the serial data input (RxD0) pin is input to the timer input pin (TI01). The width at the baud rate (transfer rate) of the other party can be measured by using the timer array unit input pulse interval measurement mode with the input edge signal of the start bit as a trigger.

The ISC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the ISC register to 00H.

Figure 12-19. Format of Input Switch Control Register (ISC)

Address: F00730H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching the channel 1 of the timer array unit
0	Select the input signal on TI01 pin as the timer input (normal operation)
1	Select the input signal on RxD0 pin as the timer input (Detection of the wake-up signal and pulse-width-measurement for baud rate correction)

ISC0	Switching the external interrupt (INTP0)
0	Select the input signal on INTP0 pin as the external interrupt input (normal operation)
1	Select the input signal on RxD0 pin as the external interrupt input (detection of the wake-up signal)

Caution Be sure to clear bits 2 to 7 to 0.

12.3.17 Registers controlling port functions of serial input/output pins

Using the serial array unit requires setting of the registers that control the port functions multiplexed on the target channel (port mode register (PM0), port register (P0), port output mode register (POM0), port mode control register (PMC0)).

For details, see **4.3.1 Port mode registers 0, 4 (PM0, PM4)**, **4.3.2 Port registers 0, 4, 12, 13 (P0, P4, P12, P13), 4.3.4 Port output mode register 0 (POM0)**, and **4.3.5 Port mode control register 0 (PMC0)**.

For an example of settings when using a port pin for serial input and output, see **4.5.3 Example of register settings for port and alternate functions used**.

Specifically, using a port pin with a multiplexed serial data or serial clock output function (e.g. P00/SO00/TxD0/INTP1) for serial data or serial clock output, requires setting the corresponding bits in the port mode control register (PMC0) and port mode register (PM0) to 0, and the corresponding bit in the port register (P0) to 1.

When using the port pin in N-ch open-drain output (V_{DD} tolerance) mode, set the corresponding bit in the port output mode register (POM0) to 1.

Example: When P00/SO00/TxD0/INTP1 is to be used for serial data output
Set the PMC00 bit of port mode control register 0 to 0.
Set the PM00 bit of port mode register 0 to 0.
Set the P00 bit of port register 0 to 1.

Specifically, using a port pin with a multiplexed serial data or serial clock input function (e.g. P01/ANI0/SI00/RxD0/SDA00/KR2) for serial data or serial clock input, requires setting the corresponding bit in the port mode register (PM0) to 1, and the corresponding bit in the port mode control register (PMC0) to 0. In this case, the corresponding bit in the port register (P0) can be set to 0 or 1.

Example: When P01/ANI0/SI00/RxD0/SDA00/KR2 is to be used for serial data input
Set the PMC01 bit of port mode control register 0 to 0.
Set the PM01 bit of port mode register 0 to 1.

12.4 Operation Stop Mode

Each serial interface of serial array unit has the operation stop mode.
In this mode, serial communication cannot be executed, thus reducing the power consumption.
In addition, the serial interface function alternate pins can be used as port function pins in this mode.

12.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

The PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

Figure 12-20. Peripheral Enable Register 0 (PER0) Setting When Stopping Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAU0 to be stopped to 0.

	<7>	<6>	<5>	<4>	3	<2>	1	<0>
PER0	TMKAEN Note	CMPEN Note	ADCEN	IICA0EN Note		SAU0EN		TAU0EN
	×	×	×	×	0	0/1	0	×

Control of SAU0 input clock
0: Stops supply of input clock
1: Supplies input clock

Note 16-pin products only.

Cautions 1. When setting serial array unit 0, be sure to first set the control registers of the serial array unit 0 with the SAU0EN bit set to 1. If SAU0EN = 0, control registers of serial array unit 0 become default values and writing to them is ignored (except for the noise filter enable register 0 (NFEN0), input switch control register (ISC), port output mode register 0 (POM0), port mode register 0 (PM0), port mode control register 0 (PMC0), and port register 0 (P0)).

2. Be sure to clear the following bits to 0.
10-pin products: Bits 1, 3, 4, 6, and 7
16-pin products: Bits 1 and 3

Remark ▒: Setting disabled (fixed by hardware)
×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user.

12.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 12-21. Each Register Setting When Stopping Operation by Channels

(a) Serial channel stop register 0 (ST0) … This register is a trigger register that is used to enable stopping communication/count by each channel.

Symbol:	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	ST01 0/1	ST00 0/1

1: Clears the SE0n bit to 0 and stops the communication operation

* Because the ST0n bit is a trigger bit, it is cleared immediately when SE0n = 0.

(b) Serial Channel Enable Status Register 0 (SE0) … This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

Symbol:	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	SE01 0/1	SE00 0/1

0: Operation stops

* The SE0 register is a read-only status register. Operation is stopped by using the ST0 register.
For a channel whose operation is disabled, the value of the CKO0n bit of the SO0 register can be set by software.

(c) Serial output enable register 0 (SOE0) … This register is a register that is used to enable or stop output of the serial communication operation of each channel.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 Note 0/1	SOE00 0/1

0: Stops output by serial communication operation

* For channel n whose serial output is stopped, the SO0n bit value of the SO0 register can be set by software.

(d) Serial clock output register 0 (CKO0) …This register is a buffer register for serial output of each channel.

Symbol:	7	6	5	4	3	2	1	0
CKO0	0	0	0	0	0	0	CKO01 Note 0/1	CKO00 0/1

1: Serial clock output value is "1"

* When using pins corresponding to each channel as port function pins, set the corresponding CKO0n bit to "1".

(e) Serial output register 0 (SO0) …This register is a register that is used to set a value of serial output of each channel.

Symbol:	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	SO01 Note 0/1	SO00 0/1

1: Serial data output value is "1"

* When using pins corresponding to each channel as port function pins, set the corresponding SO0n bit to "1".

Note 16-pin products only.

Remarks 1. n: Channel number (n = 0, 1)

2. ☐ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

12.5 Operation of Simplified SPI (CSI00, CSI01 Note 1) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]

- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate Note 2

During master communication: Max. $f_{CLK}/4$

During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Notes 1. 16-pin products only.

2. Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics.
For details, see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**.

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01Note		–

Simplified SPI (CSI00, CSI01 Note) performs the following seven types of communication operations.

- Master transmission (See **12.5.1**.)
- Master reception (See **12.5.2**.)
- Master transmission/reception (See **12.5.3**.)
- Slave transmission (See **12.5.4**.)
- Slave reception (See **12.5.5**.)
- Slave transmission/reception (See **12.5.6**.)

Note 16-pin products only.

12.5.1 Master transmission

Master transmission is that the RL78/G10 outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00	CSI01 Note 1
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SO00	SCK01, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	7 or 8 bits	
Transfer rate Note 2	Max. $f_{CLK}/4$ [Hz] (SDR0nH[7:1] = 1 or more) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]	
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data output starts at the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-reverse (data output at the falling edge and data input at the rising edge of SCK) • CKP0n = 1: Reverse (data output at the rising edge and data input at the falling edge of SCK)	
Data direction	MSB or LSB first	

Notes 1. 16-pin products only.

2. Use this operation within a range that satisfies the conditions above and the peripheral function characteristics in the electrical specifications (see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{CLK}: System clock frequency

2. n = 0, 1

(1) Register setting

Figure 12-22. Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00, CSI01 Note) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol: SMR0nH

15	14	13	12	11	10	9	8
CKS0n	CCS0n						STS0n
0/1	0	0	0	0	0	0	0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Symbol: SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0/1

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

15	14	13	12	11	10	9	8
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	0	0/1	0/1	0	0	0	0

Selection of the data and clock phase (For details about the setting, see **12.3 Registers Controlling Serial Array Unit**.)

Symbol: SCR0nL

7	6	5	4	3	2	1	0
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

15	14	13	12	11	10	9	8
Baud rate setting (Operation clock (f_{MCK}) division setting)							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Transmit data (Transmit data setting)							

SIOp

(d) Serial clock output register 0 (CKO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
CKO0							CKO01 Note	CKO00
	0	0	0	0	0	0	0/1	0/1

Communication starts when these bits are 1 if the clock phase is non-inversion (the CKP0n bit of the SCR0n = 0). If the clock phase is inverted (CKP0n = 1), communication starts when these bits are 0.

(**Note** and **Remarks** are listed on the next page.)

Figure 12-22. Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00, CSI01 Note) (2/2)

(e) Serial output register 0 (SO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
SO0							SO01 Note	SO00
	0	0	0	0	0	0	0/1	0/1

(f) Serial output enable register 0 (SOE0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SOE0							SOE01 Note	SOE00
	0	0	0	0	0	0	0/1	0/1

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0							SS01	SS00
	0	0	0	0	0	0	0/1	0/1

Note 16-pin products only.

Remarks 1. n = 0, 1, p: CSI number (p = 00, 01)

2. □: Setting is fixed in the simplified SPI (CSI) master transmission mode, ▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user.

(2) Operation procedure

Figure 12-23. Initial Setting Procedure for Master Transmission

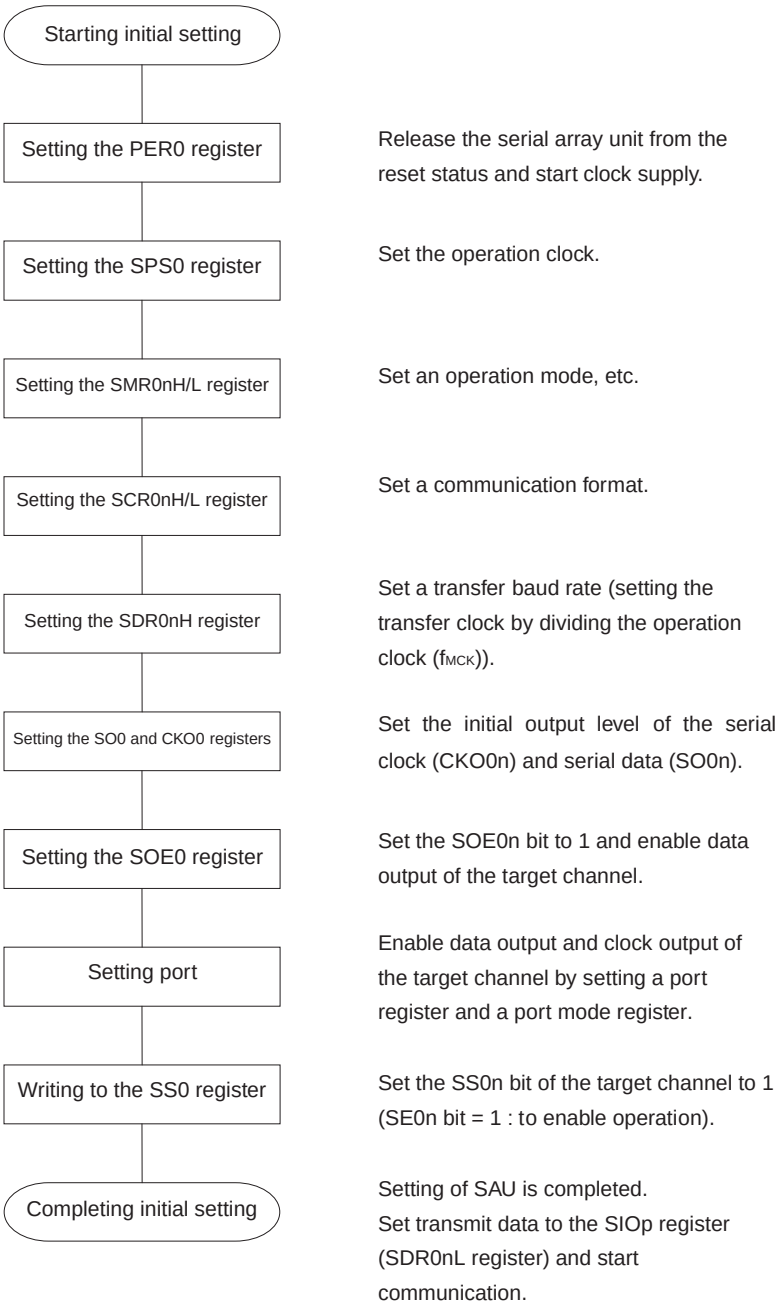


Figure 12-24. Procedure for Stopping Master Transmission

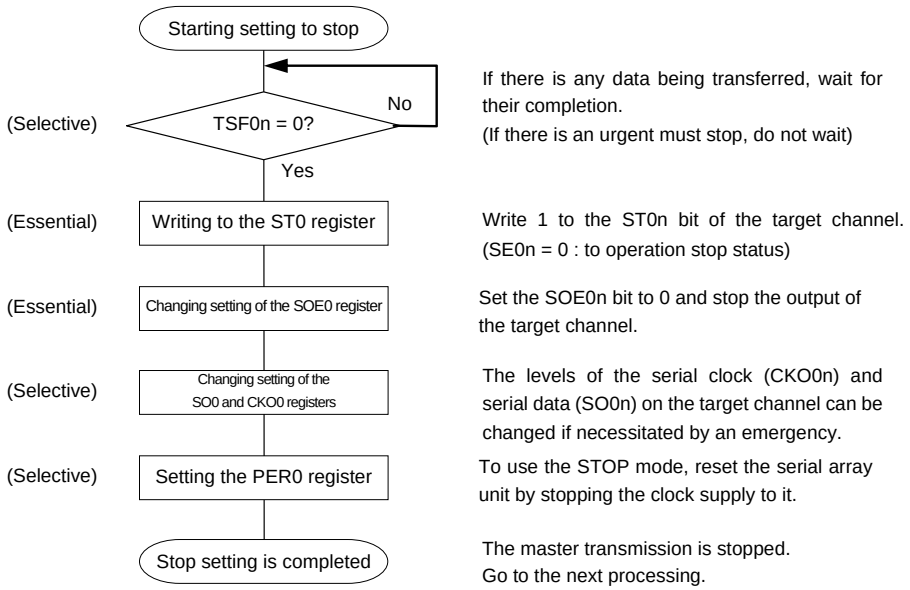
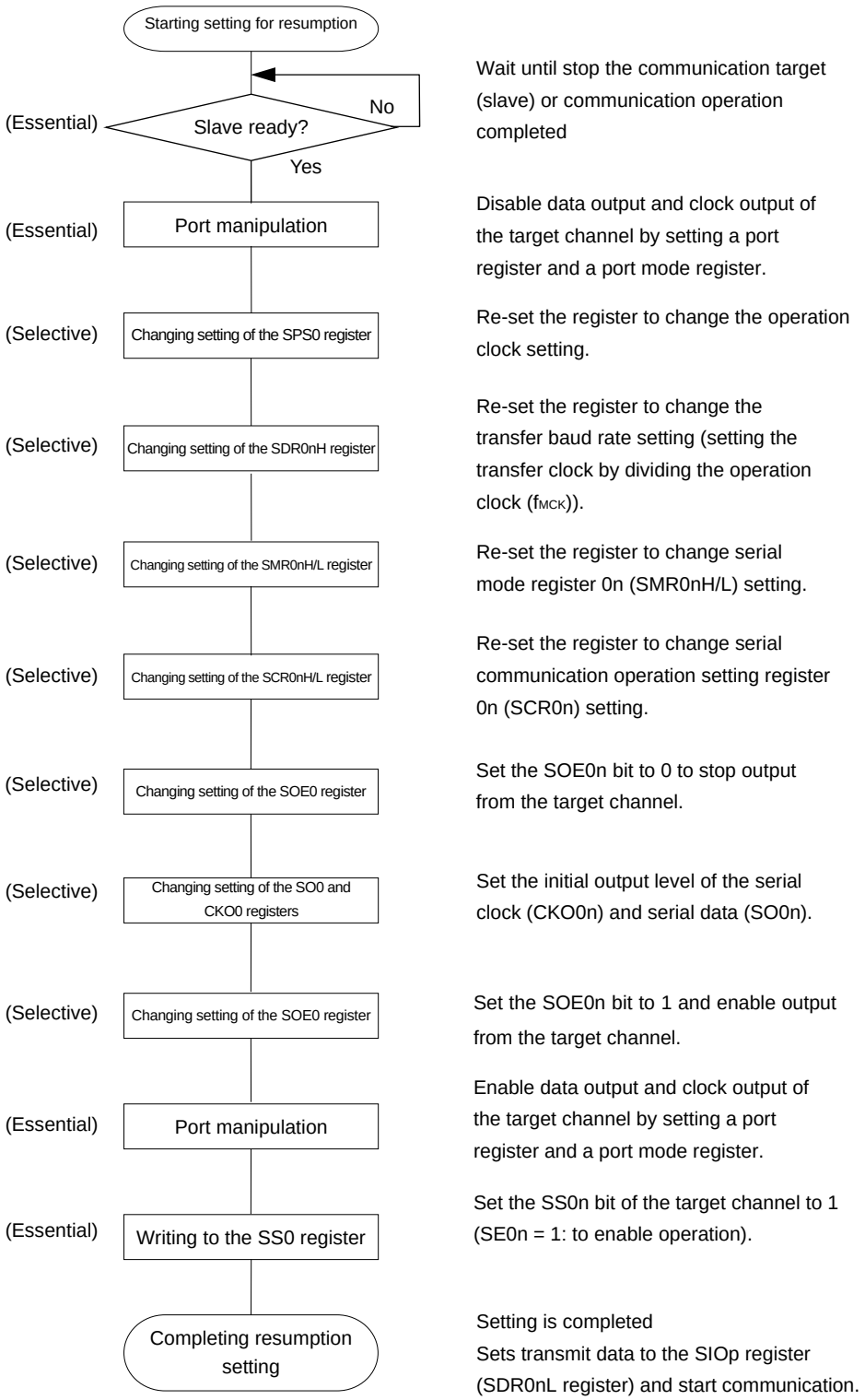


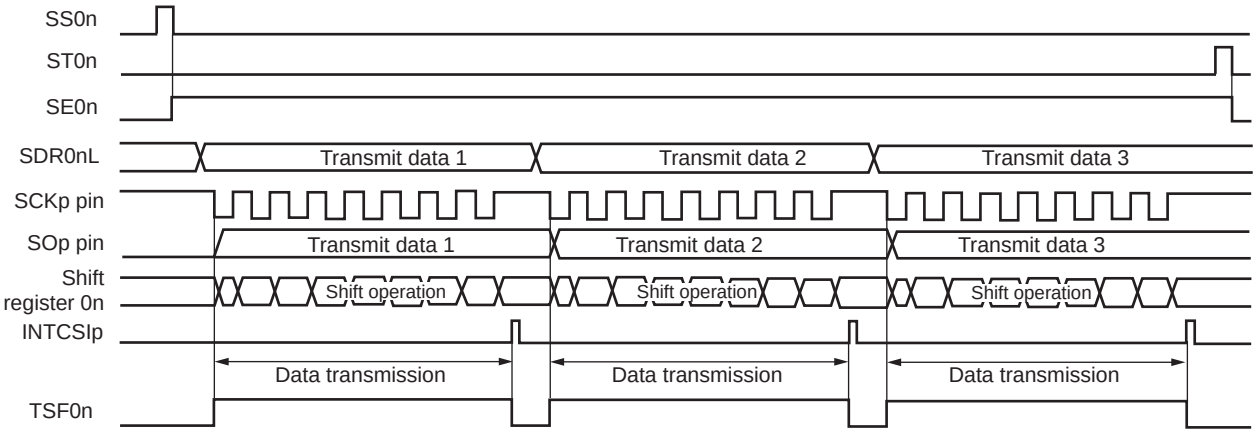
Figure 12-25. Procedure for Resuming Master Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

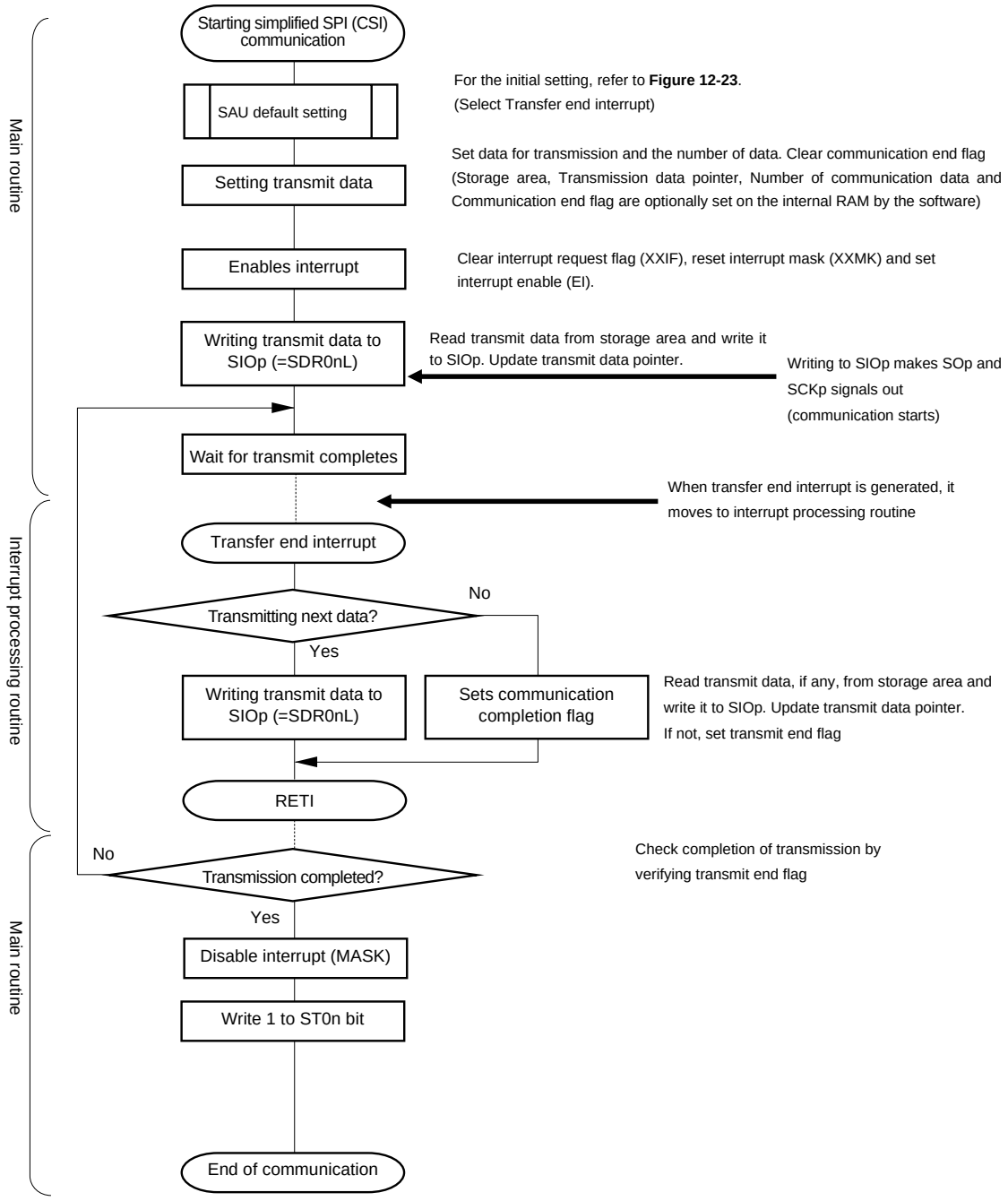
(3) Processing flow (in single-transmission mode)

Figure 12-26. Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAP0n = 0, CKP0n = 0)



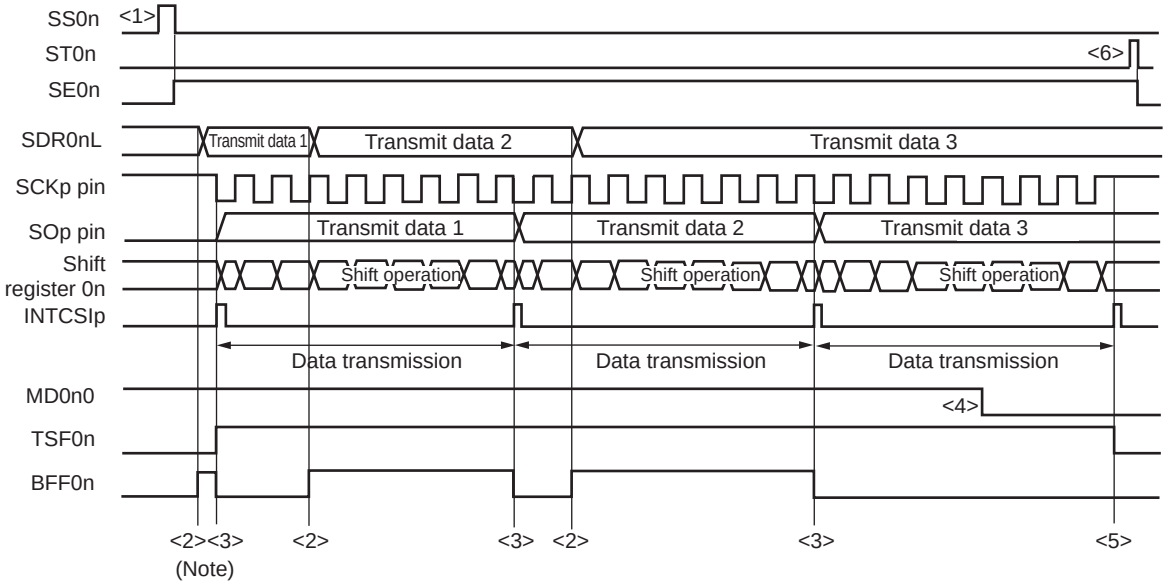
Remark n = 0, 1, p: CSI number (p = 00, 01)

Figure 12-27. Flowchart of Master Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 12-28. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAP0n = 0, CKP0n = 0)

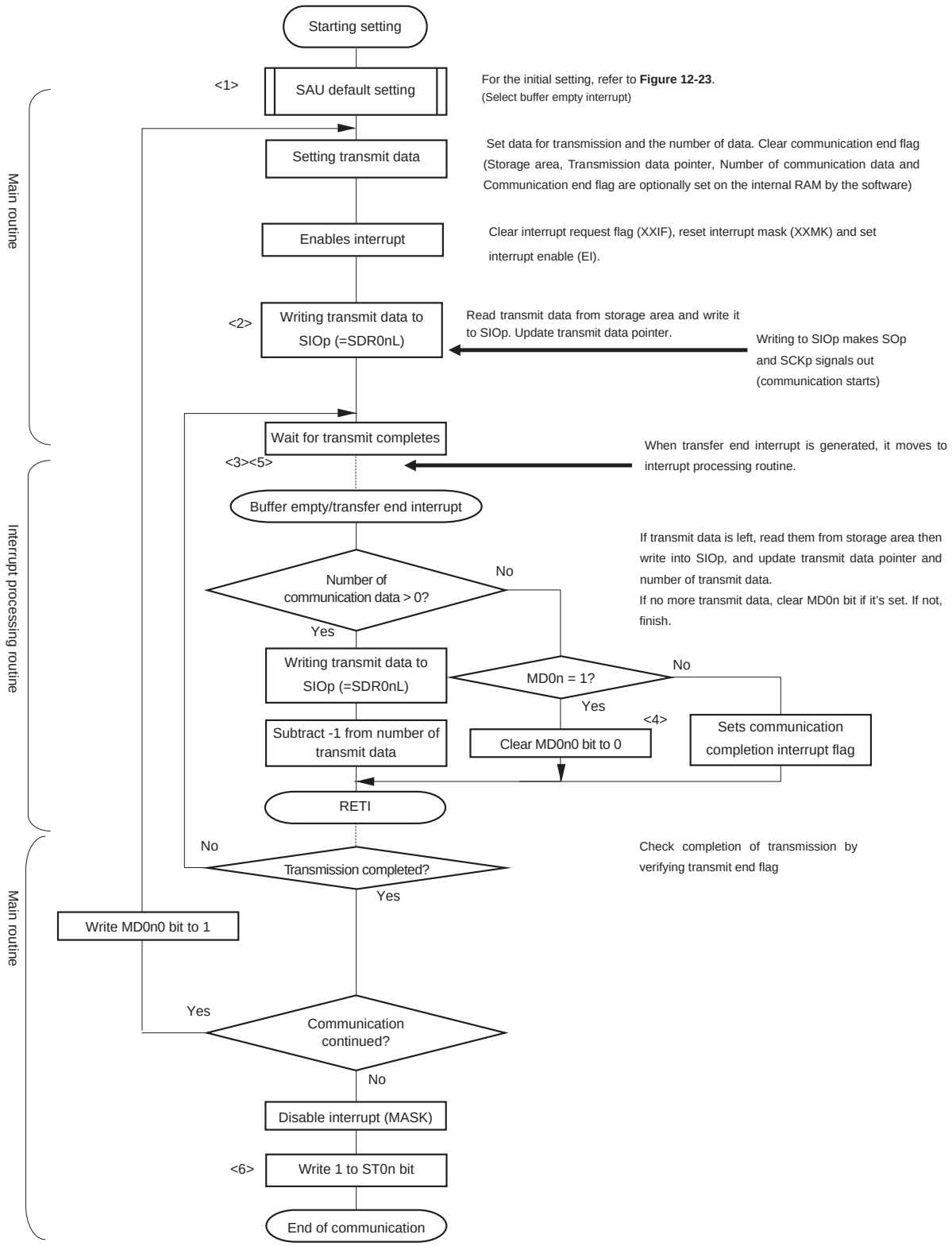


Note If transmit data is written to the SDR0nL register while the BFF0n bit of serial status register 0n (SSR0n) is 1 (valid data is stored in serial data register 0n (SDR0nL)), the transmit data is overwritten.

Caution **The MD0n0 bit of serial mode register 0n (SMR0nH/L) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.**

Remark n = 0, 1, p: CSI number (p = 00, 01)

Figure 12-29. Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-28 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

12.5.2 Master reception

Master reception is that the RL78/G10 outputs a transfer clock and receives data from other device.

Simplified SPI	CSI00	CSI01[Note 1]
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00	SCK01, SI01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVF0n) only	
Transfer data length	7 or 8 bits	
Transfer rate [Note 2]	Max. $f_{CLK}/4$ [Hz] (SDR0nH[7:1] = 1 or more) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]	
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data input starts at the start of the operation of the serial clock. • DAP0n = 1: Data input starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-inversion • CKP0n = 1: Inverted	
Data direction	MSB or LSB first	

Notes 1. 16-pin products only.

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{CLK}: System clock frequency

2. n = 0, 1

(1) Register setting

Figure 12-30. Example of Contents of Registers for Master Reception of Simplified SPI (CSI00, CSI01 Note) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol: SMR0nH

15	14	13	12	11	10	9	8
CKS0n	CCS0n						STS0n
0/1	0	0	0	0	0	0	0

Symbol: SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

15	14	13	12	11	10	9	8
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
0	1	0/1	0/1	0	0	0	0

Symbol: SCR0nL

7	6	5	4	3	2	1	0
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **12.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

15	14	13	12	11	10	9	8
Baud rate setting (Operation clock (f_{MCK}) division setting)							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Receive data (Write FFH as dummy data.)							

SIOp

(d) Serial clock output register 0 (CKO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
CKO0							CKO01 Note	CKO00
	0	0	0	0	0	0	0/1	0/1

Communication starts when these bits are 1 if the clock phase is non-inversion (the CKP0n bit of the SCR0n = 0). If the clock phase is inverted (CKP0n = 1), communication starts when these bits are 0.

(e) Serial output register 0 (SO0) … The register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
SO0							SO01 Note	SO00
	0	0	0	0	0	0	×	×

(**Note** and **Remarks** are listed on the next page.)

Figure 12-30. Example of Contents of Registers for Master Reception of Simplified SPI (CSI00, CSI01 Note) (2/2)

(f) Serial output enable register 0 (SOE0) … The register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
SOE0							SOE01 Note	SOE00
	0	0	0	0	0	0	×	×

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0							SS01	SS00
	0	0	0	0	0	0	0/1	0/1

Note 16-pin products only.

Remarks 1. n = 0, 1, p: CSI number (p = 00, 01)

2. □: Setting is fixed in the simplified SPI (CSI) master transmission mode, ■: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-31. Initial Setting Procedure for Master Reception

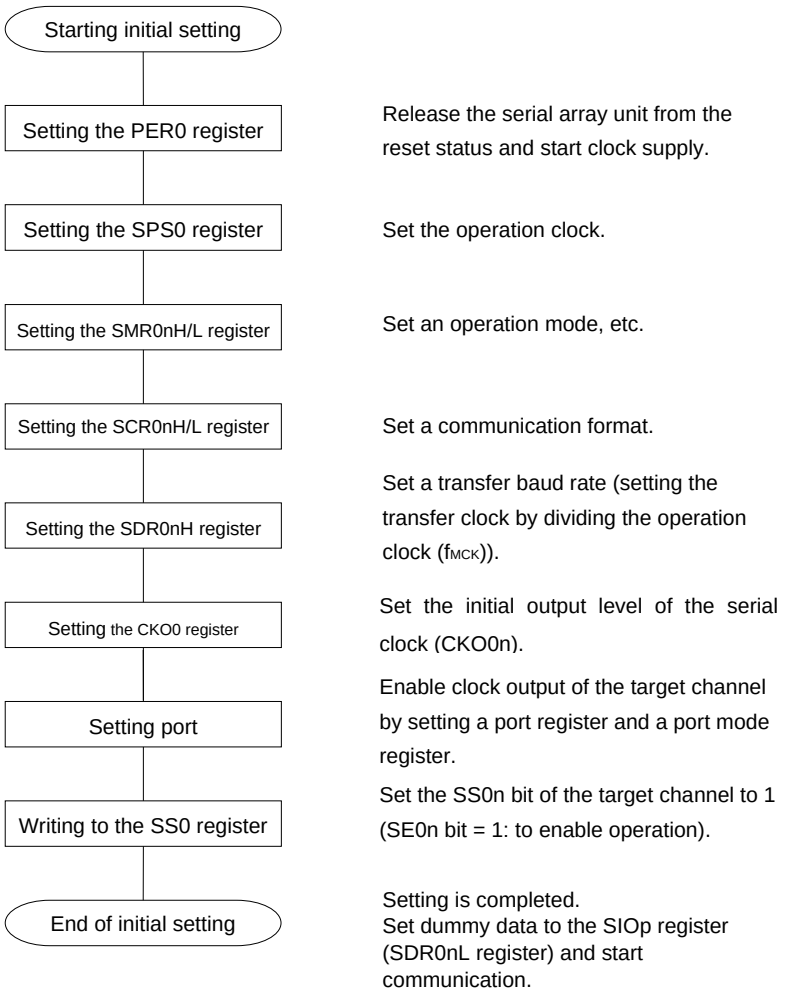


Figure 12-32. Procedure for Stopping Master Reception

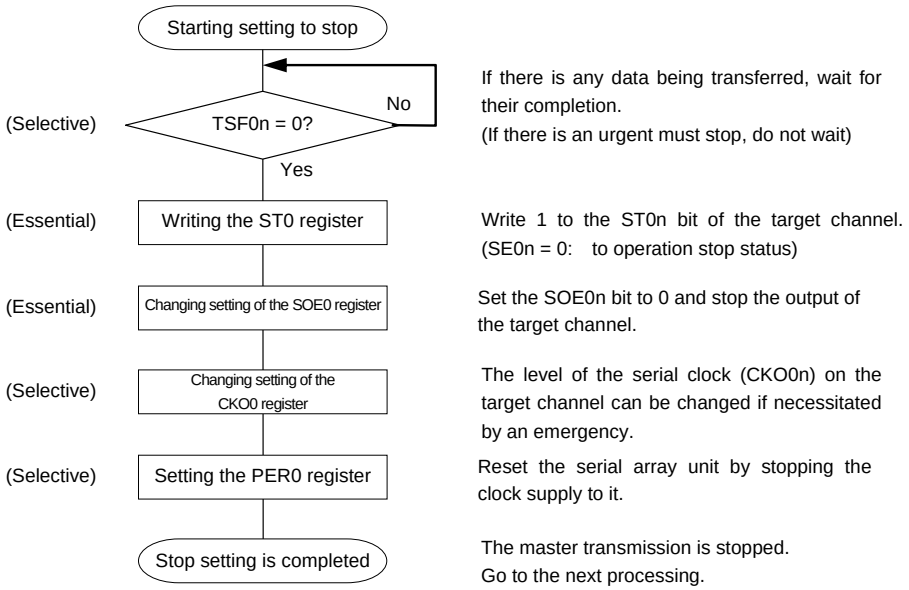
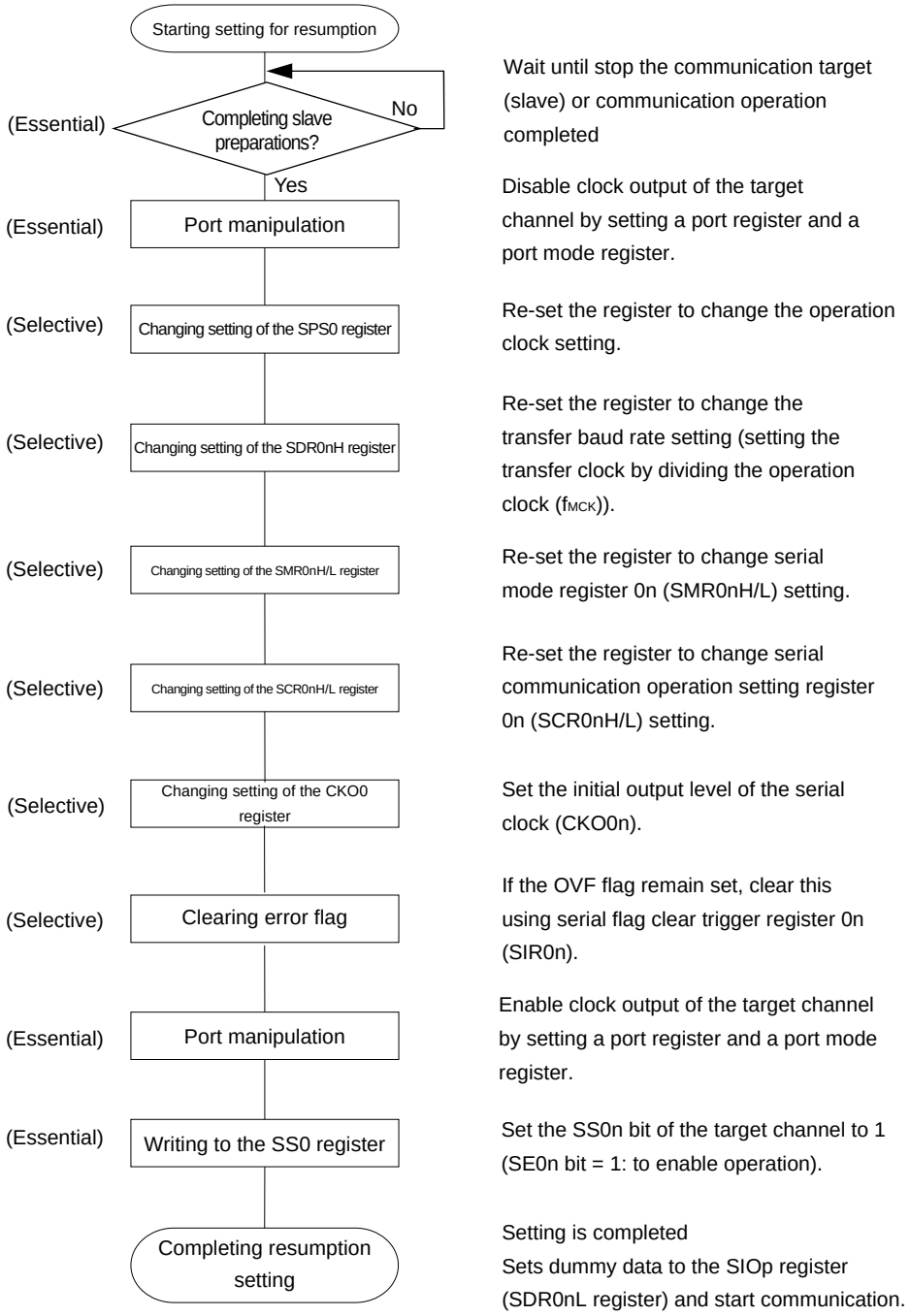


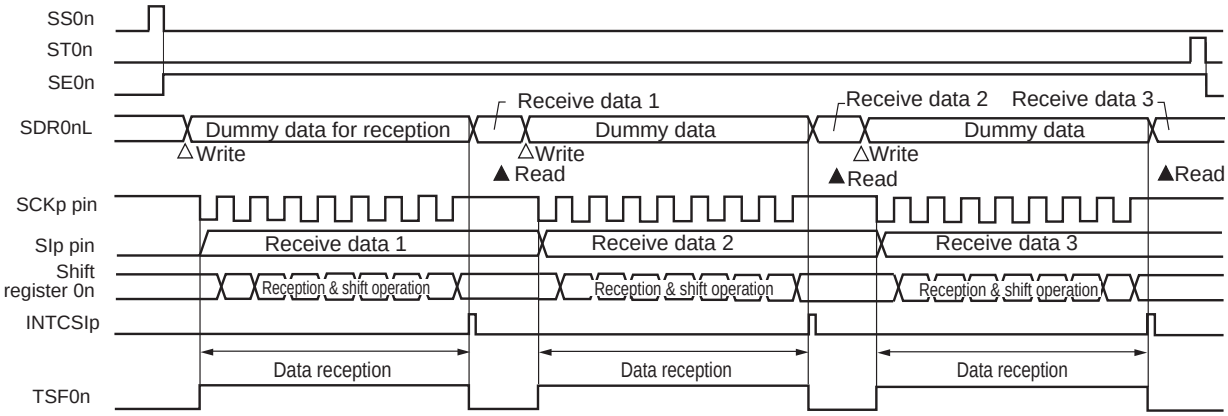
Figure 12-33. Procedure for Resuming Master Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

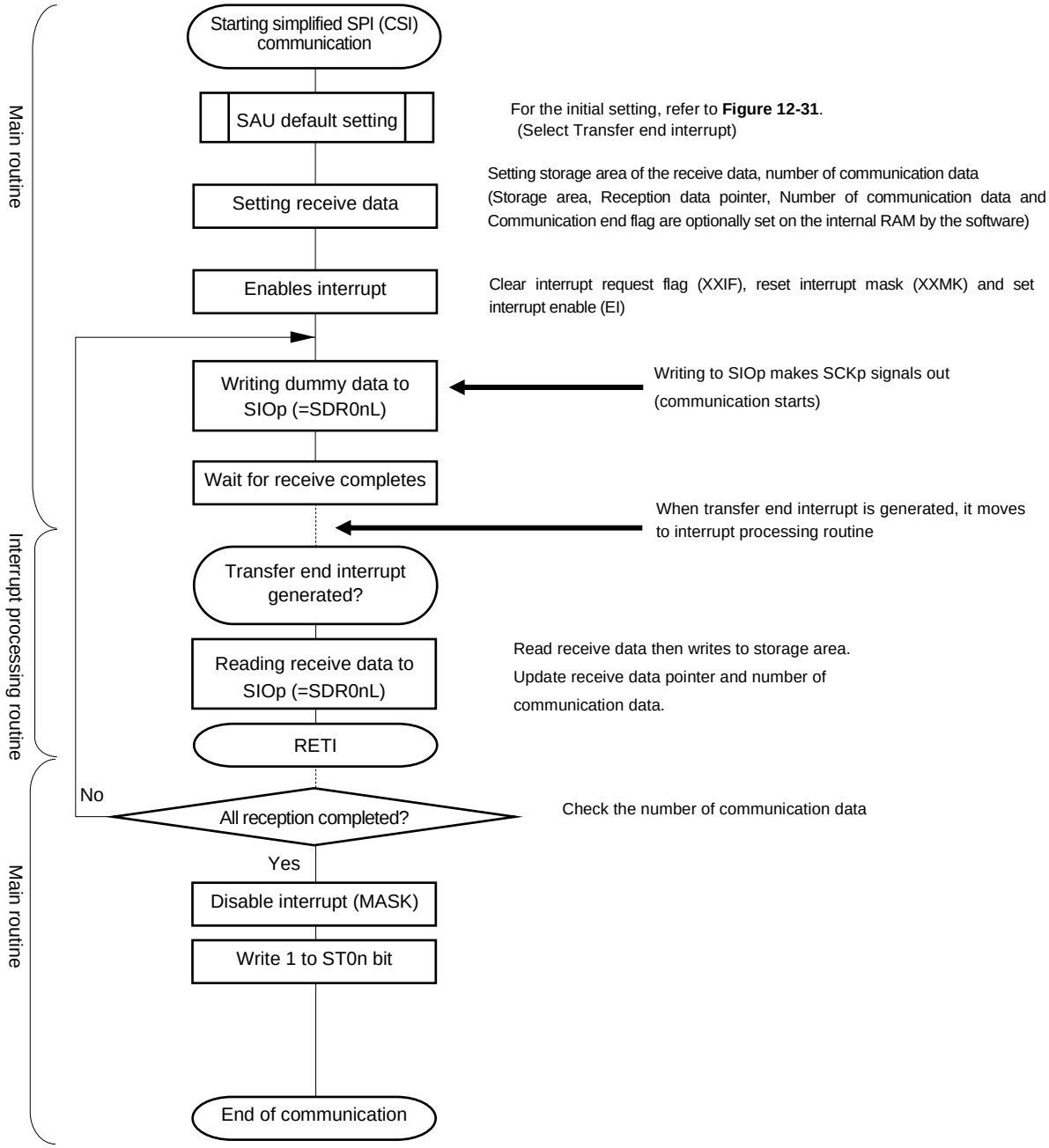
(3) Processing flow (in single-reception mode)

Figure 12-34. Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAP0n = 0, CKP0n = 0)



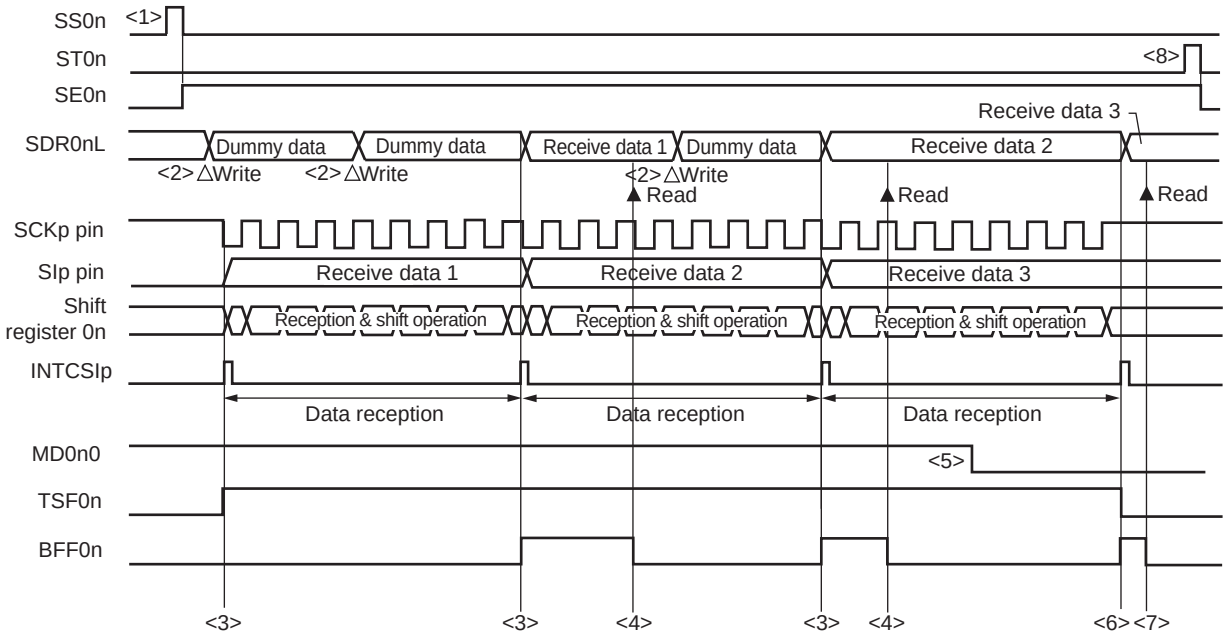
Remark n = 0, 1, p: CSI number (p = 00, 01)

Figure 12-35. Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing flow (in continuous reception mode)

Figure 12-36. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAP0n = 0, CKP0n = 0)

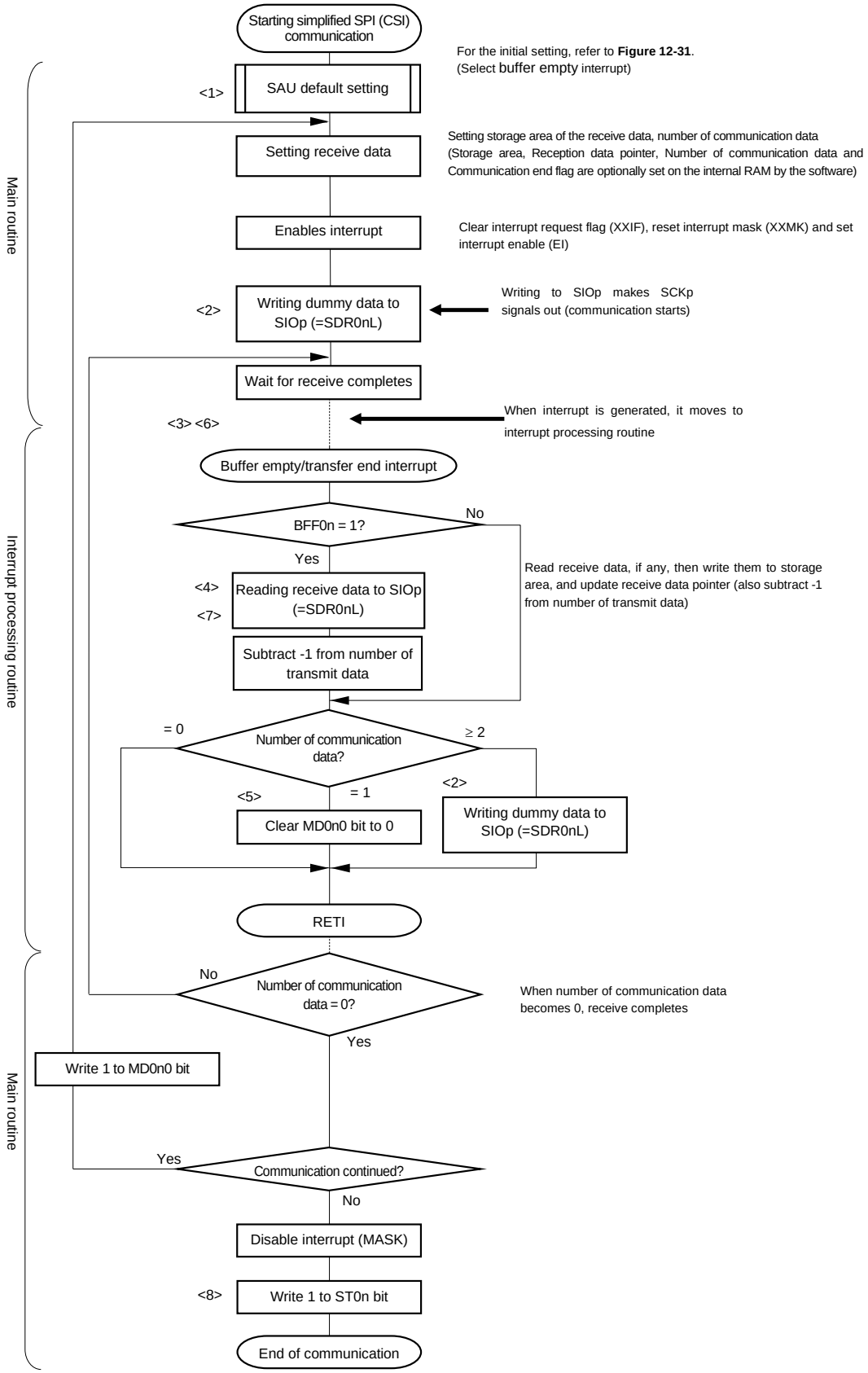


Caution The MD0n0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-37 Flowchart of Master Reception (in Continuous Reception Mode)**.

2. n = 0, 1, p: CSI number (p = 00, 01)

Figure 12-37. Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-36 Timing Chart of Master Reception (in Continuous Reception Mode)**.

12.5.3 Master transmission/reception

Master transmission/reception is that the RL78/G10 outputs a transfer clock and transmits/receives data to/from other device.

Simplified SPI	CSI00	CSI01 Note 1
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVF0n) only	
Transfer data length	7 or 8 bits	
Transfer rate Note 2	Max. $f_{CLK}/4$ [Hz] (SDR0nH[7:1] = 1 or more) Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz]	
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data I/O starts at the start of the operation of the serial clock. • DAP0n = 1: Data I/O starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-inversion • CKP0n = 1: Inverted	
Data direction	MSB or LSB first	

Notes **1.** 16-pin products only.
2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**).

Remarks **1.** f_{CLK}: System clock frequency
2. n = 0, 1

(1) Register setting

Figure 12-38. Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01 Note) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol:SMR0nH

15	14	13	12	11	10	9	8
CKS0n	CCS0n						STS0n
0/1	0	0	0	0	0	0	0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Symbol:SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0/1

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol:SCR0nH

15	14	13	12	11	10	9	8
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	1	0/1	0/1	0	0	0	0

Selection of the data and clock phase (For details about the setting, see **12.3 Registers Controlling Serial Array Unit**.)

Symbol:SCR0nL

7	6	5	4	3	2	1	0
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 0n (SDR0nH, SDR0nL)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data setting/receive data register							

SIOp

(d) Serial clock output register 0 (CKO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
CKO0							CKO01 Note	CKO00
	0	0	0	0	0	0	0/1	0/1

Communication starts when these bits are 1 if the clock phase is non-inversion (the CKP0n bit of the SCR0n = 0). If the clock phase is inverted (CKP0n = 1), communication starts when these bits are 0.

(e) Serial output register 0 (SO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
SO0							SO01 Note	SO00
	0	0	0	0	0	0	0/1	0/1

(**Note** and **Remarks** are listed on the next page.)

Figure 12-38. Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01 Note) (2/2)

(f) Serial output enable register 0 (SOE0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SOE0							SOE01 Note	SOE00
	0	0	0	0	0	0	0/1	0/1

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0							SS01	SS00
	0	0	0	0	0	0	0/1	0/1

Note 16-pin products only.

Remarks 1. n = 0, 1, p: CSI number (p = 00, 01)

2. ☐: Setting is fixed in the simplified SPI (CSI) master transmission/reception mode

▒ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-39. Initial Setting Procedure for Master Transmission/Reception

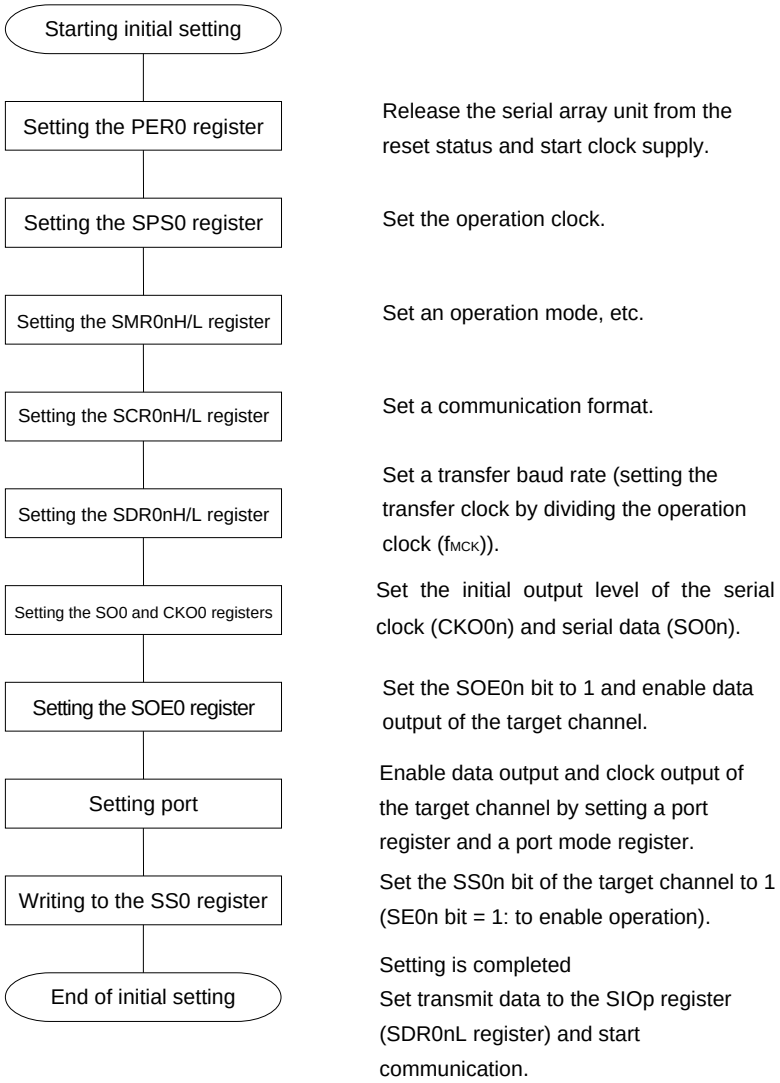


Figure 12-40. Procedure for Stopping Master Transmission/Reception

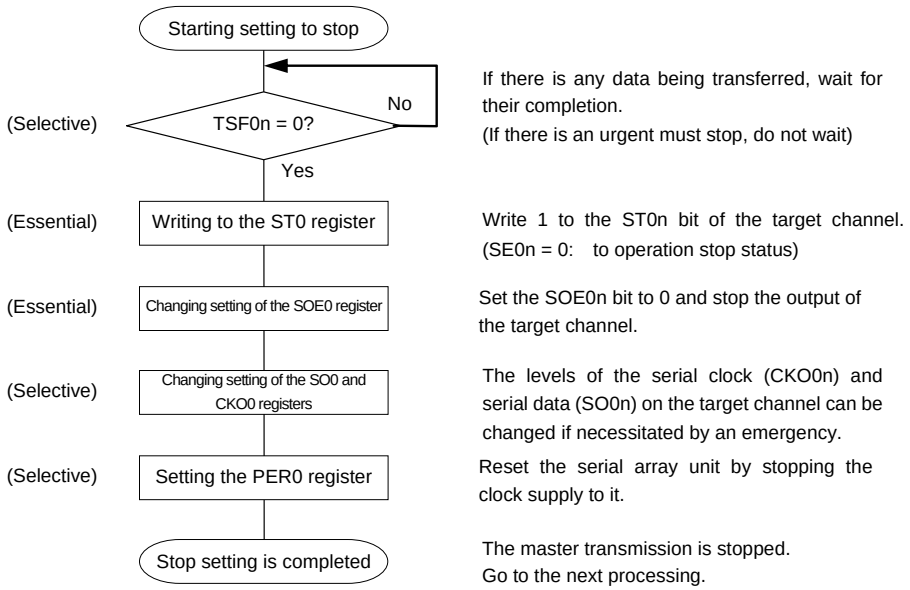
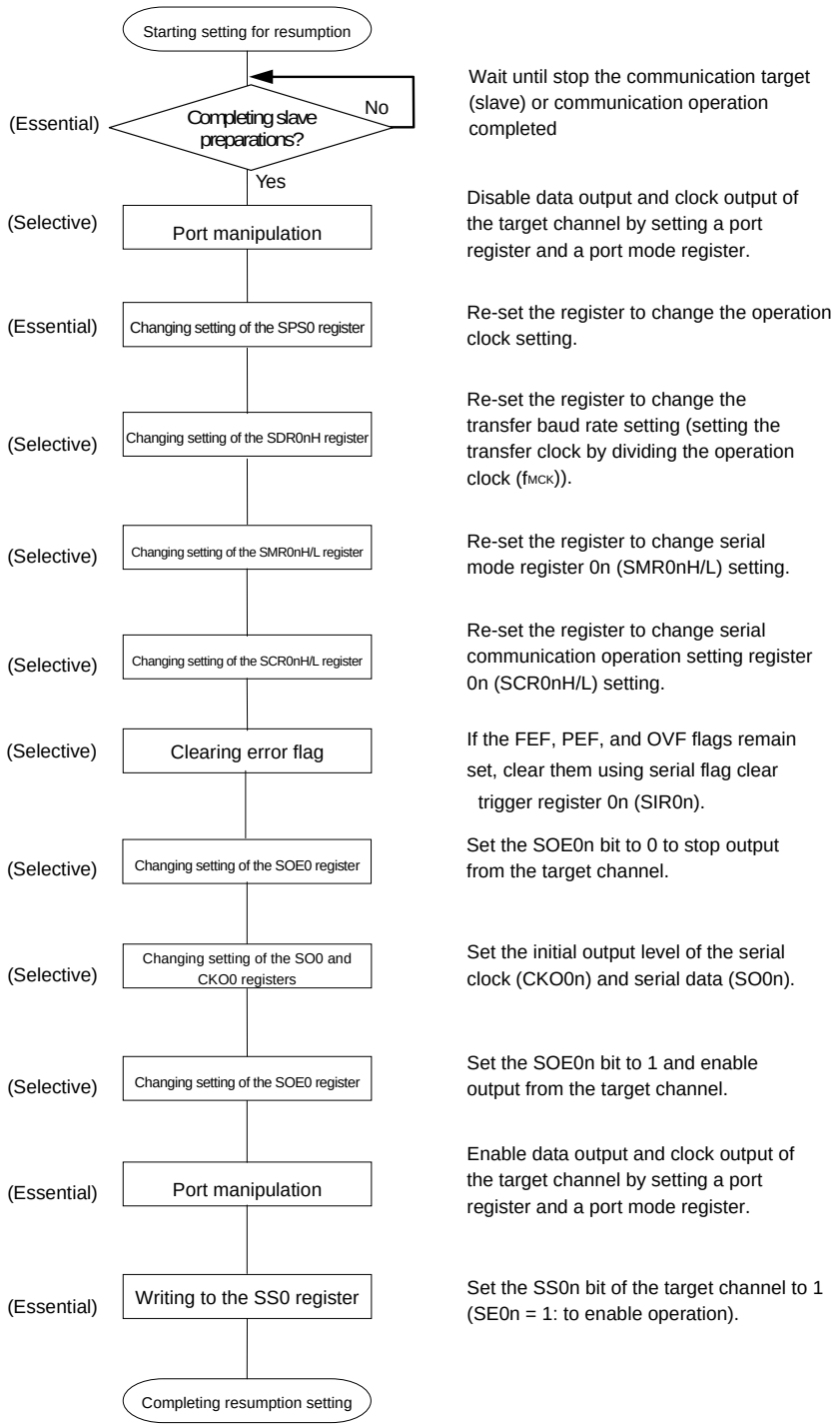


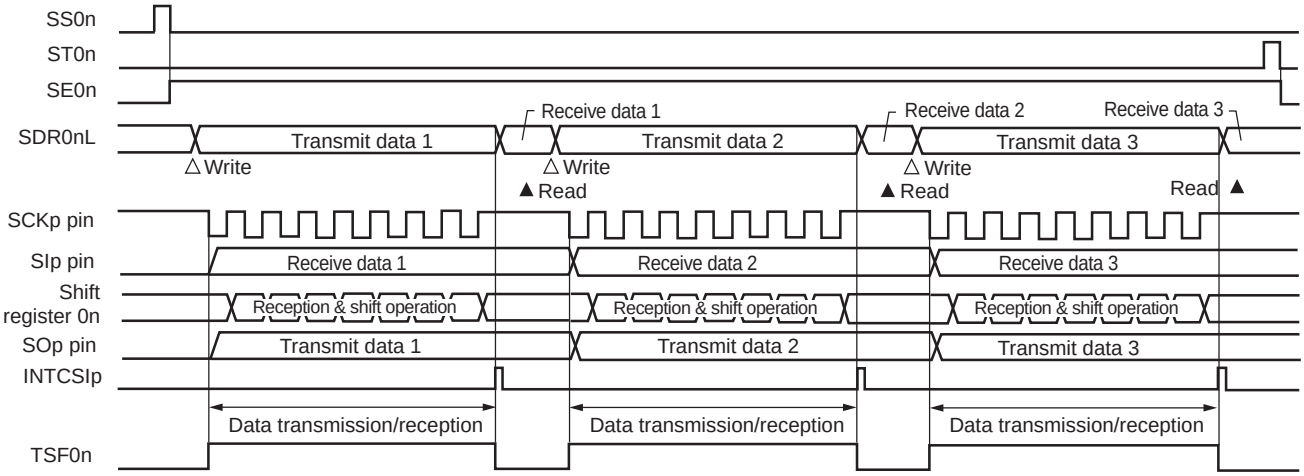
Figure 12-41. Procedure for Resuming Master Transmission/Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

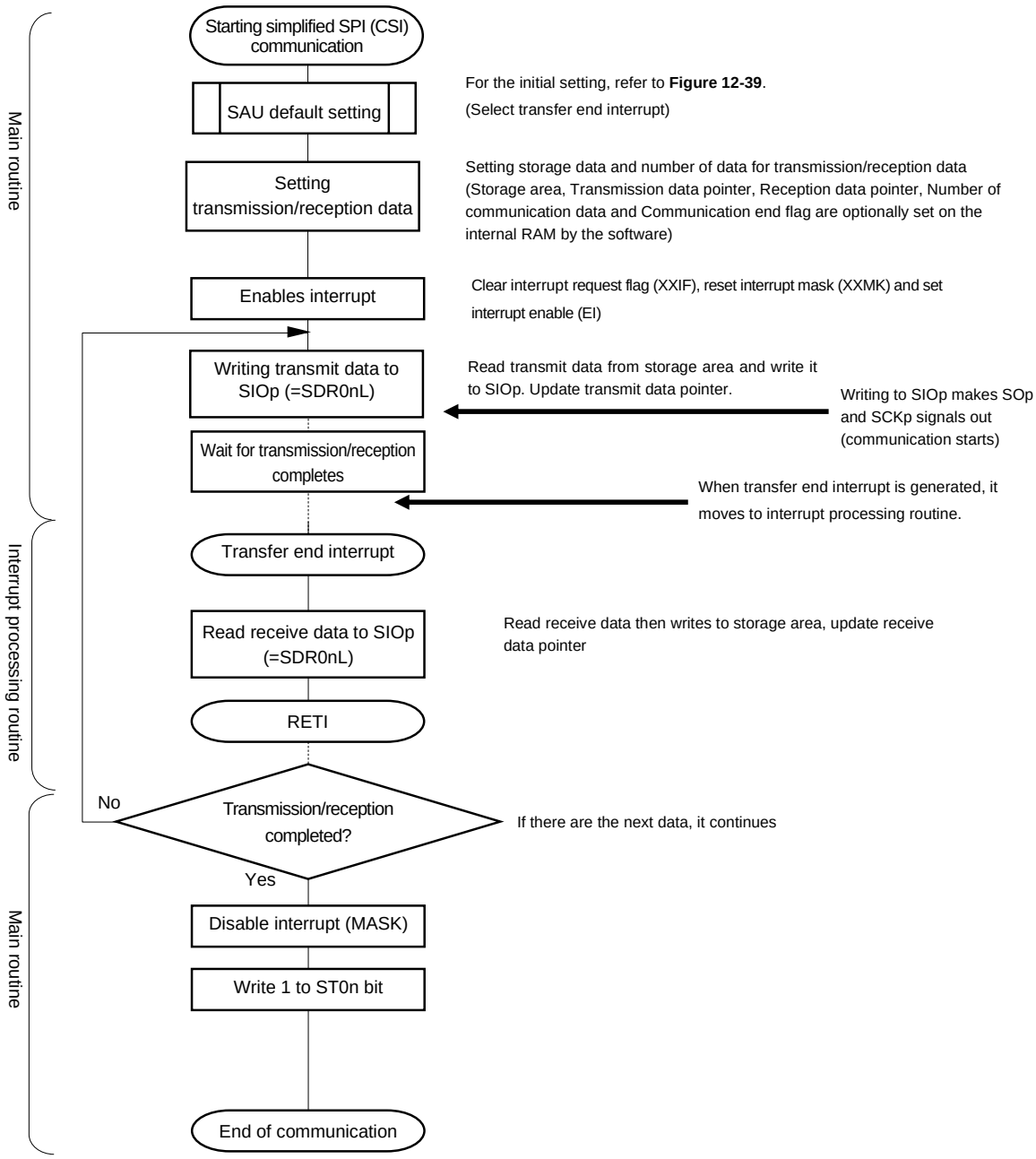
(3) Processing flow (in single-transmission/reception mode)

Figure 12-42. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAP0n =0, CKP0n = 0)



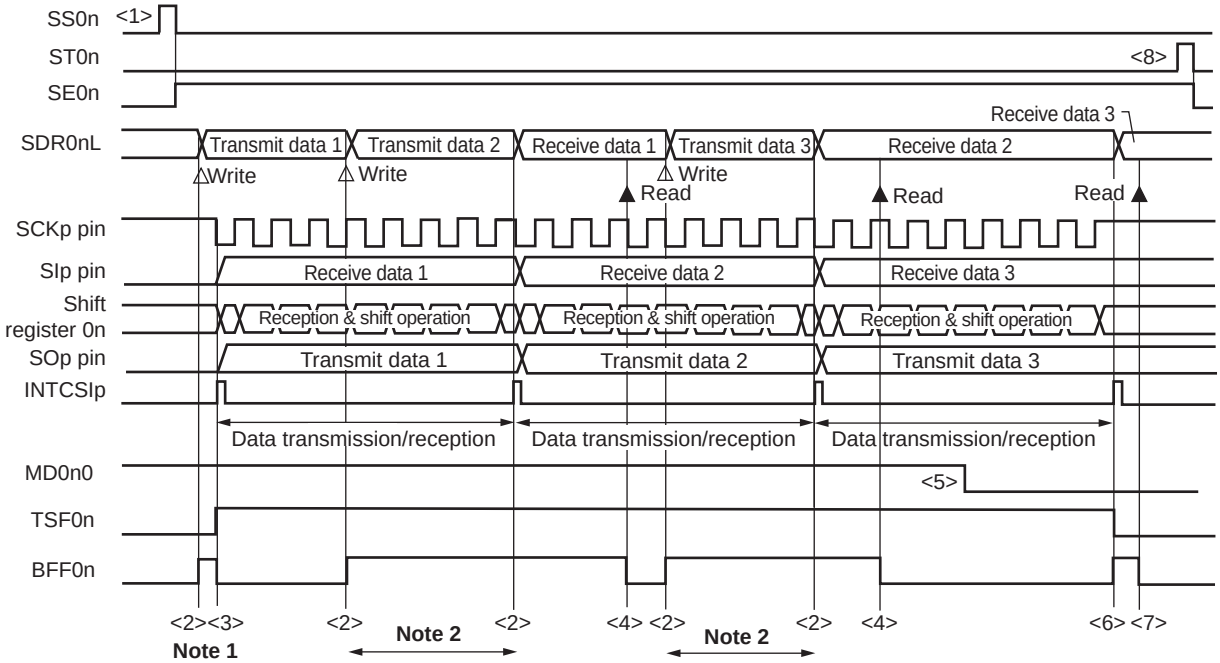
Remark n = 0, 1, p: CSI number (p = 00, 01)

Figure 12-43. Flowchart of Master Transmission/Reception (in Single-Transmission/Reception Mode)



(4) Processing flow (in continuous transmission/reception mode)

Figure 12-44. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAP0n = 0, CKP0n = 0)



Notes 1. If transmit data is written to the SDR0nL register while the BFF0n bit of serial status register 0n (SSR0n) is 1 (valid data is stored in serial data register 0n (SDR0nL)), the transmit data is overwritten.

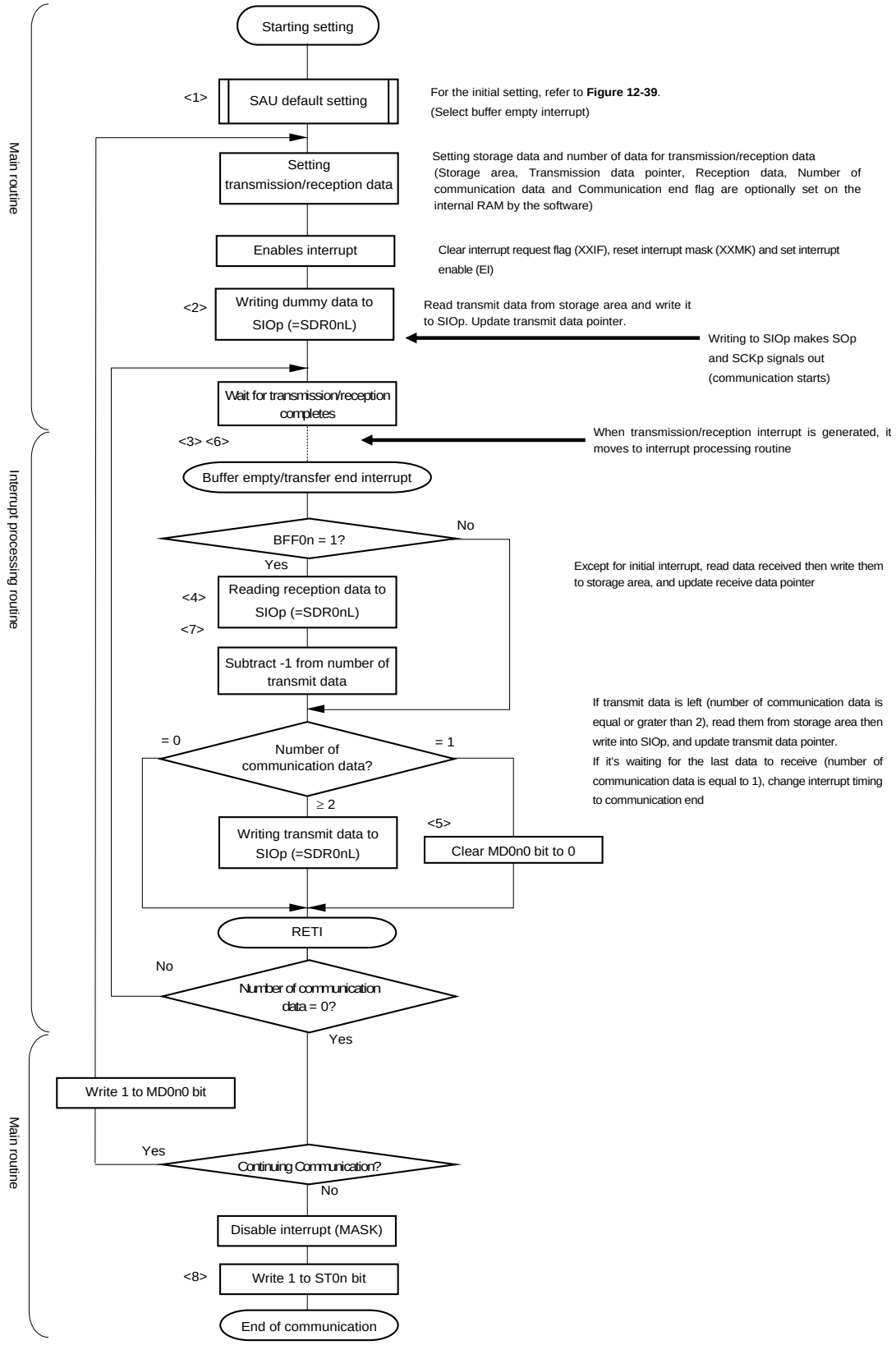
2. The transmit data can be read by reading the SDR0nL register during this period. At this time, the transfer operation is not affected.

Caution The MD0n0 bit of serial mode register 0n (SMR0nL) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-45 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode**).

2. n = 0, 1, p: CSI number (p = 00, 01)

Figure 12-45. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-44 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

12.5.4 Slave transmission

Slave transmission is that the RL78/G10 transmits data to another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01 Note 1
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SO00	SCK01, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVF0n) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. $f_{MCK}/6$ [Hz] Notes 2, 3	
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data output starts at the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-inversion • CKP0n = 1: Inverted	
Data direction	MSB or LSB first	

Notes **1.** 16-pin products only.

2. Because the external serial clock input to the SCK00 and SCK01 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

3. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**).

Remarks **1.** f_{MCK}: Operation clock frequency of target channel

2. n = 0, 1

(1) Register setting

Figure 12-46. Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01 Note) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol: SMR0nH

15	14	13	12	11	10	9	8
CKS0n	CCS0n						STS0n
0/1	1	0	0	0	0	0	0

Symbol: SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

15	14	13	12	11	10	9	8
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	0	0/1	0/1	0	0	0	0

Symbol: SCR0nL

7	6	5	4	3	2	1	0
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **12.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

15	14	13	12	11	10	9	8
0000000 Baud rate setting							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Transmit data setting							

SIOp

(d) Serial clock output register 0 (CKO0) … The register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
CKO0							CKO01 Note	CKO00
	0	0	0	0	0	0	×	×

(e) Serial output register 0 (SO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
SO0							SO01 Note	SO00
	0	0	0	0	0	0	0/1	0/1

(**Note** and **Remarks** are listed on the next page.)

Figure 12-46. Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01 Note) (2/2)

(f) Serial output enable register 0 (SOE0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SOE0							SOE01 Note	SOE00
	0	0	0	0	0	0	0/1	0/1

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0							SS01	SS00
	0	0	0	0	0	0	0/1	0/1

Note 16-pin products only.

Remarks 1. n = 0, 1 p: CSI number (p = 00, 01)

2. □: Setting is fixed in the simplified SPI (CSI) master transmission mode, ▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-47. Initial Setting Procedure for Slave Transmission

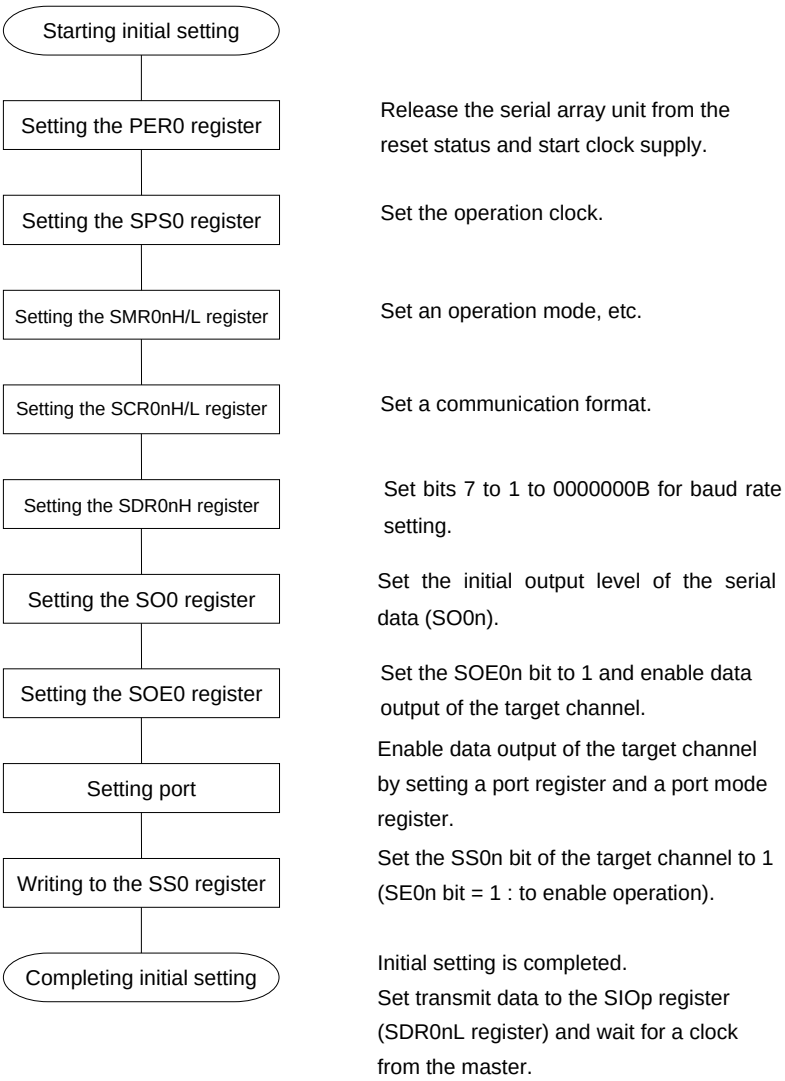


Figure 12-48. Procedure for Stopping Slave Transmission

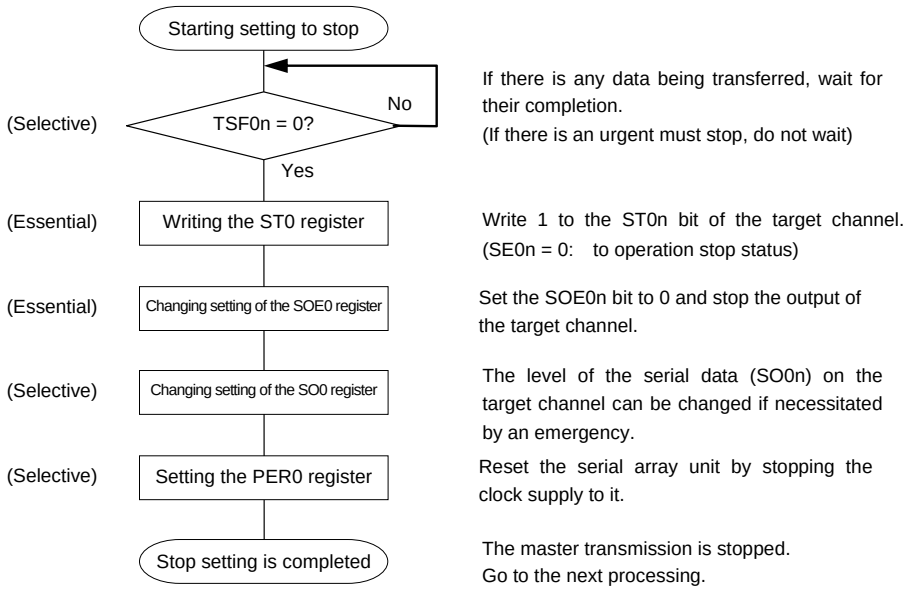
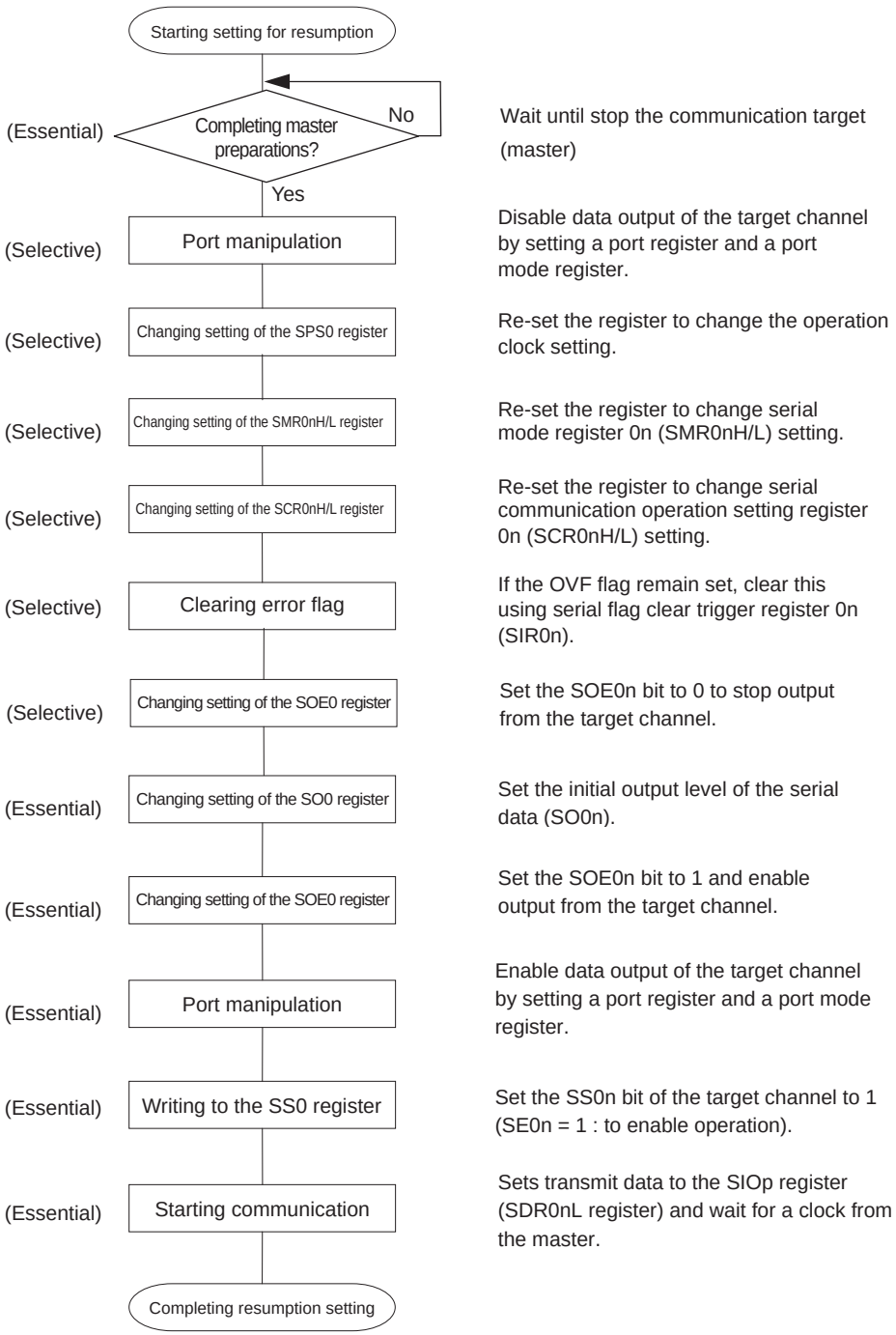


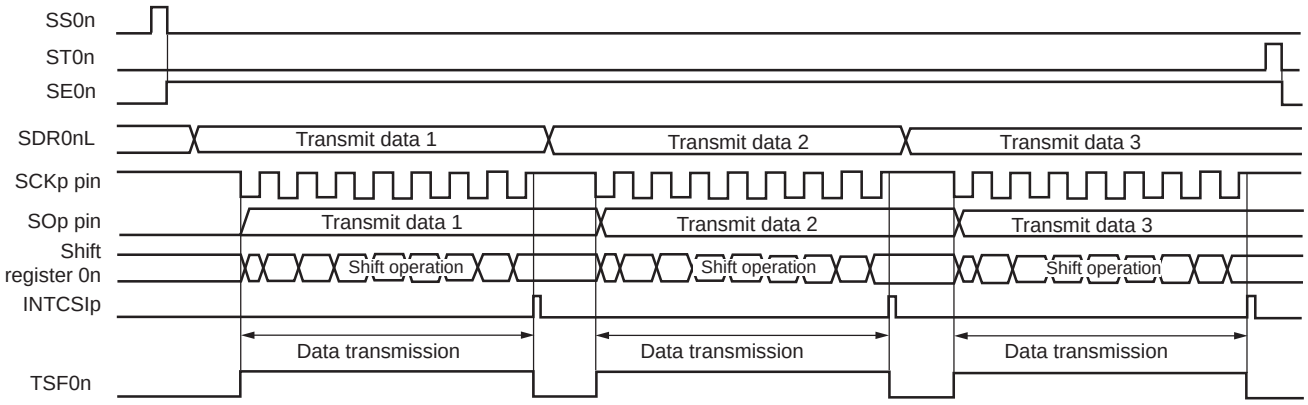
Figure 12-49. Procedure for Resuming Slave Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

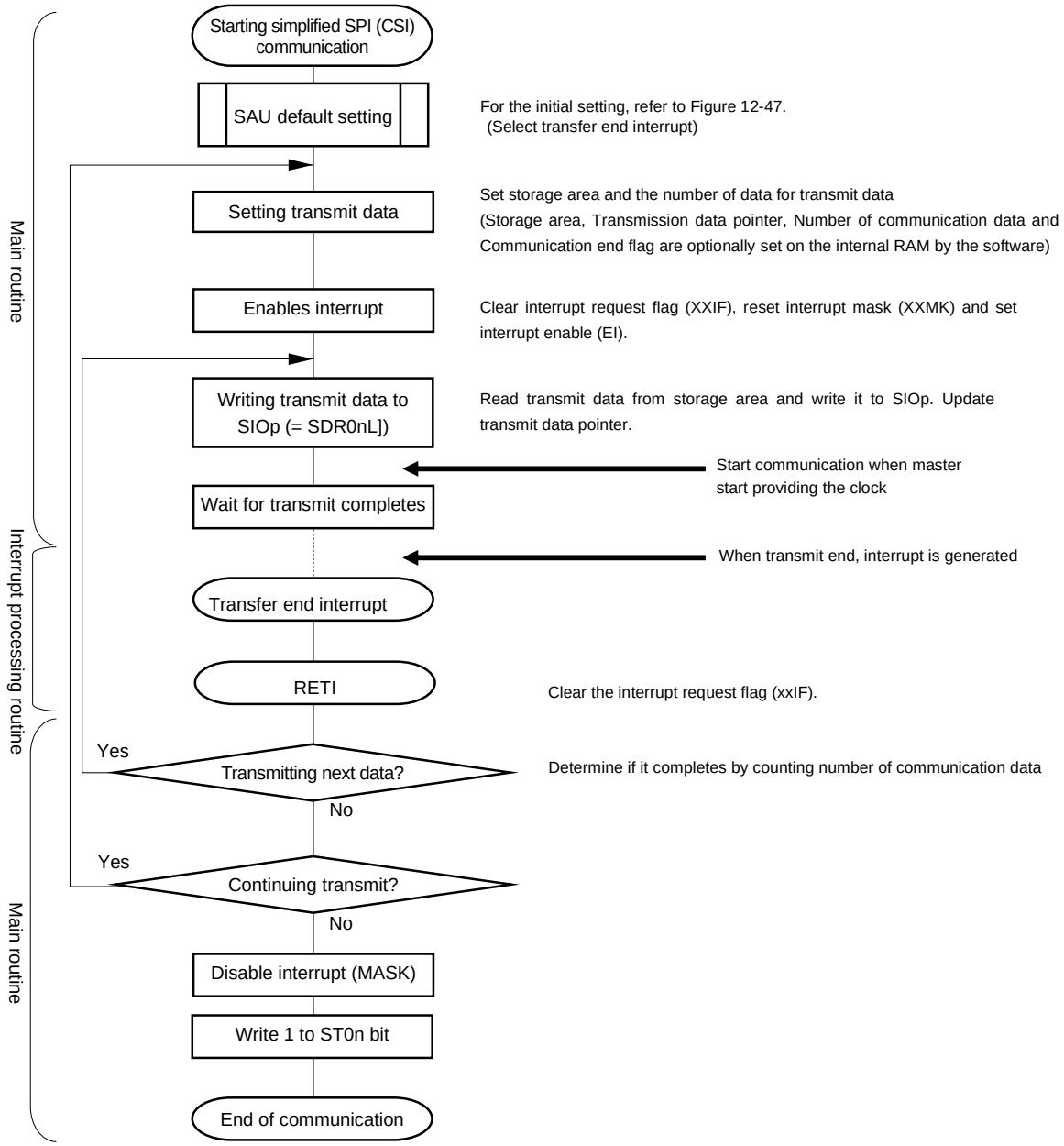
(3) Processing flow (in single-transmission mode)

Figure 12-50. Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAP0n = 0, CKP0n = 0)



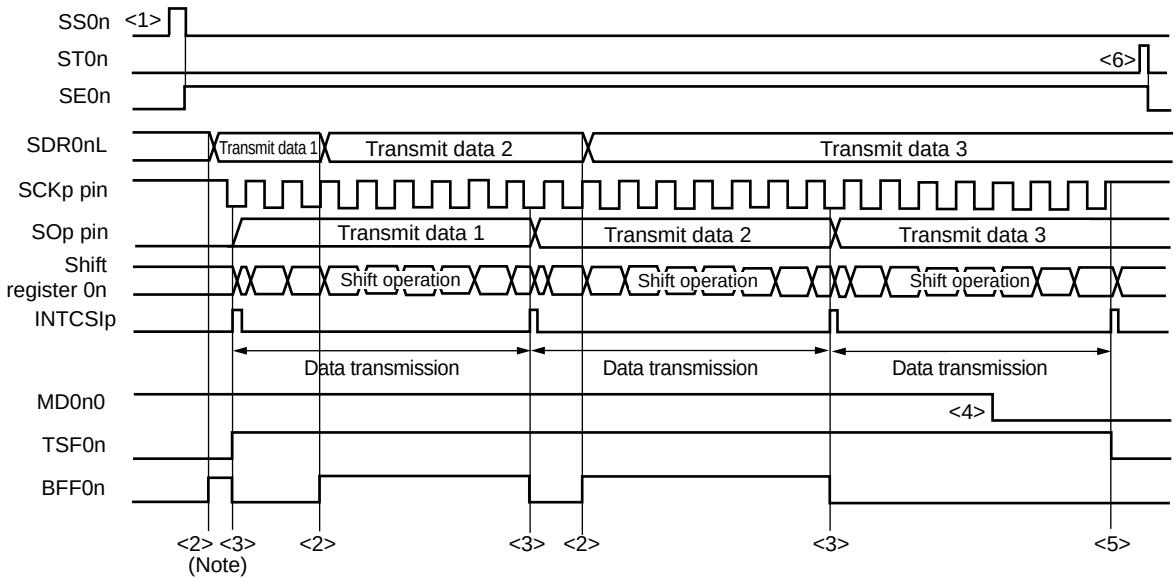
Remark n = 0 , 1, p: CSI number (p = 00, 01)

Figure 12-51. Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 12-52. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAP0n = 0, CKP0n = 0)

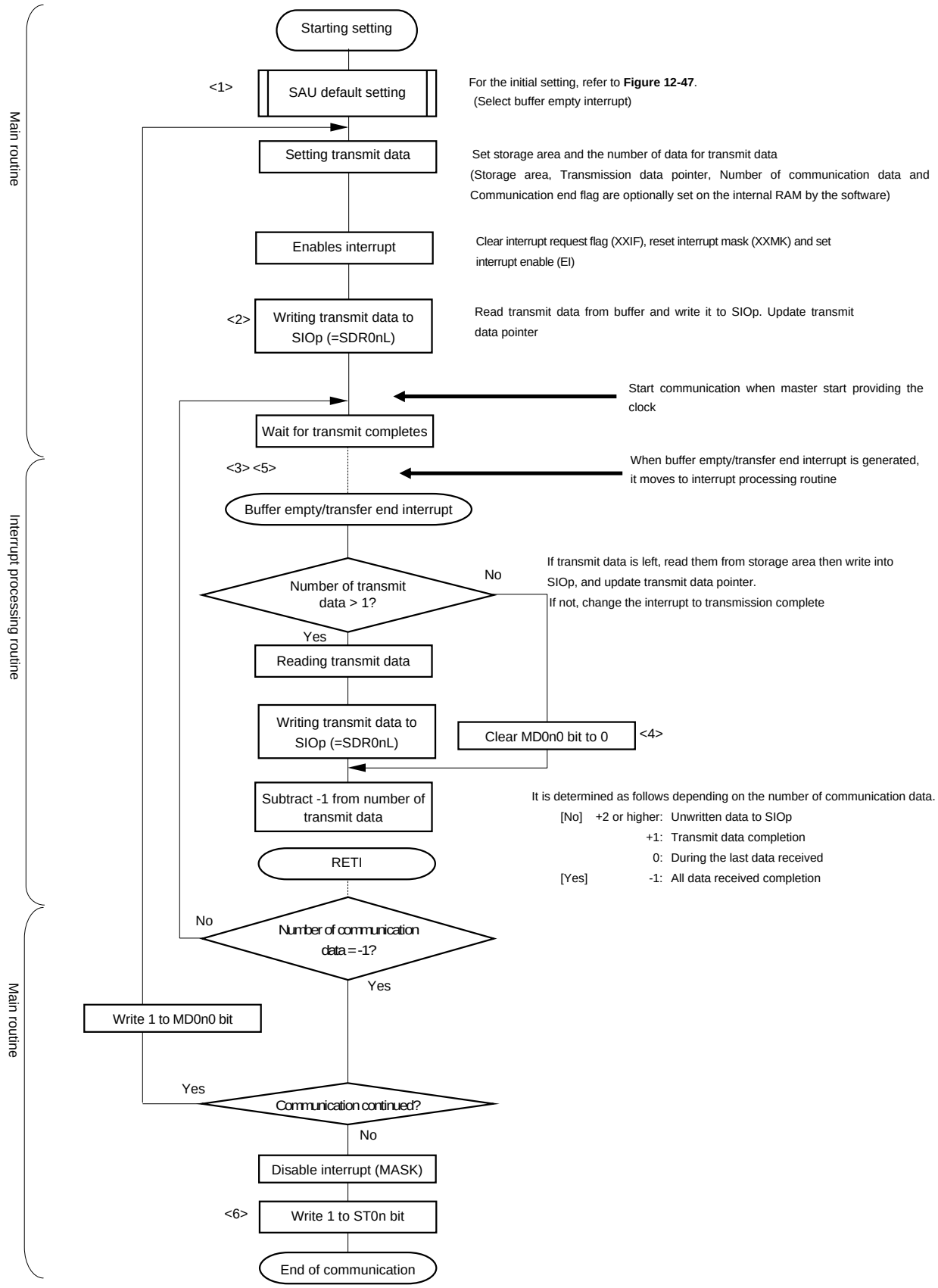


Note If transmit data is written to the SDR0nL register while the BFF0n bit of serial status register 0n (SSR0n) is 1 (valid data is stored in serial data register 0n (SDR0nL)), the transmit data is overwritten.

Caution **The MD0n0 bit of serial mode register 0n (SMR0nL) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.**

Remark n = 0, 1, p: CSI number (p = 00, 01)

Figure 12-53. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-52 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

12.5.5 Slave reception

Slave reception is that the RL78/G10 receives data from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01[Note 1]
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00	SCK01, SI01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVF0n) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. $f_{MCK}/6$ [Hz] [Notes 2, 3]	
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data output starts at the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-inversion • CKP0n = 1: Inverted	
Data direction	MSB or LSB first	

Notes 1. 16-pin products only.

2. Because the external serial clock input to the SCK00 and SCK01 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/6$ [Hz].

3. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

2. n = 0, 1

(1) Register setting

Figure 12-54. Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01 Note) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol:SMR0nH

15	14	13	12	11	10	9	8
CKS0n	CCS0n						STS0n
0/1	0	0	0	0	0	0	0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Symbol:SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0/1

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

15	14	13	12	11	10	9	8
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
0	1	0/1	0/1	0	0	0	0

Selection of the data and clock phase (For details about the setting, see **12.3 Registers Controlling Serial Array Unit**.)

Symbol: SCR0nL

7	6	5	4	3	2	1	
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

15	14	13	12	11	10	9	8
0000000 Baud rate setting							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Receive data							

SIOp

(d) Serial clock output register 0 (CKO0) … The Register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
CKO0							CKO01 Note	CKO00
	0	0	0	0	0	0	×	×

(e) Serial output register 0 (SO0) …The Register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
SO0							SO01 Note	SO00
	0	0	0	0	0	0	×	×

(**Note** and **Remarks** are listed on the next page.)

Figure 12-54. Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01 Note) (2/2)

(f) Serial output enable register 0 (SOE0) …The Register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
SOE0							SOE01 Note	SOE00
	0	0	0	0	0	0	×	×

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0							SS01	SS00
	0	0	0	0	0	0	0/1	0/1

Note 16-pin products only.

Remarks 1. n = 0, 1 p: CSI number (p = 00, 01)

2. □: Setting is fixed in the simplified SPI (CSI) slave reception mode, ■: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-55. Initial Setting Procedure for Slave Reception

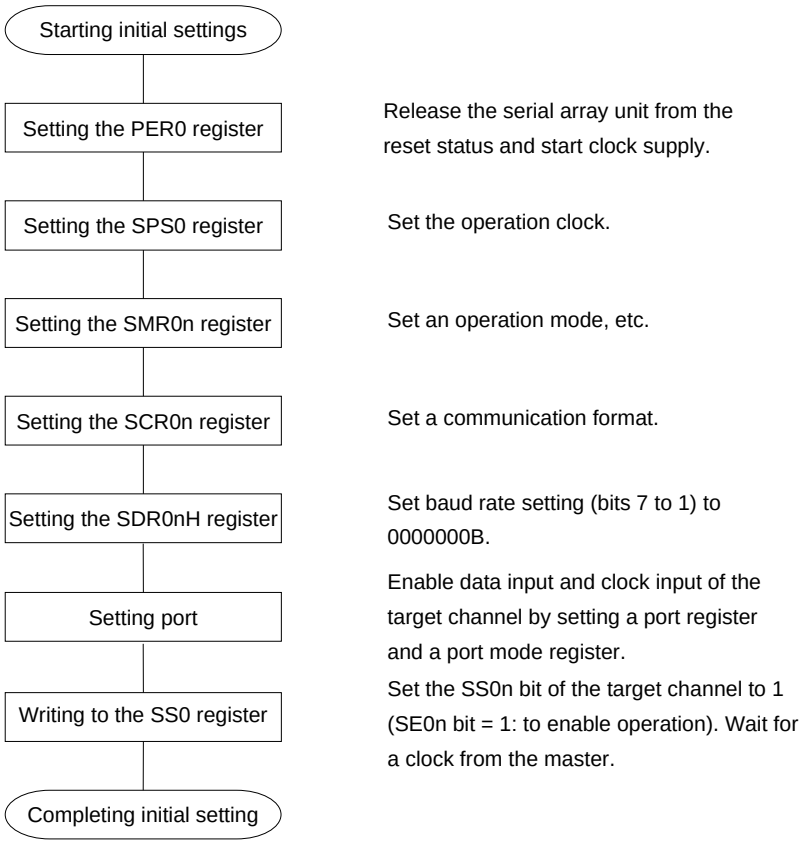


Figure 12-56. Procedure for Stopping Slave Reception

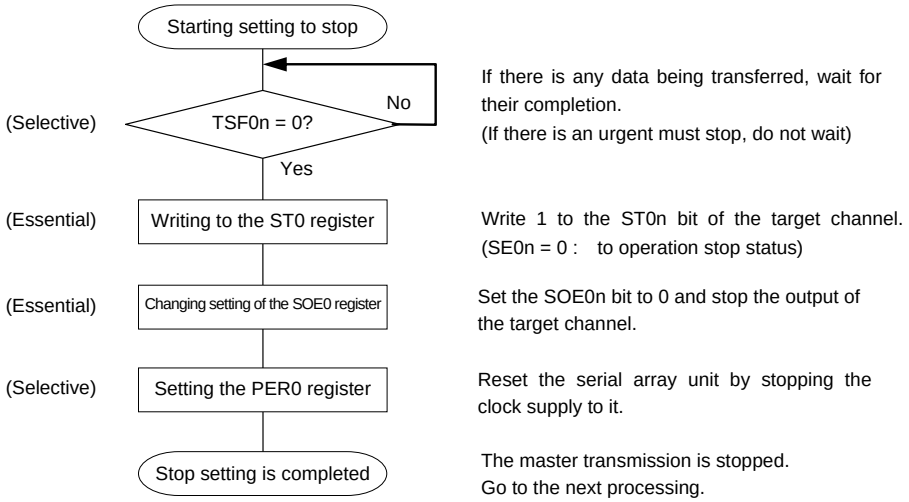
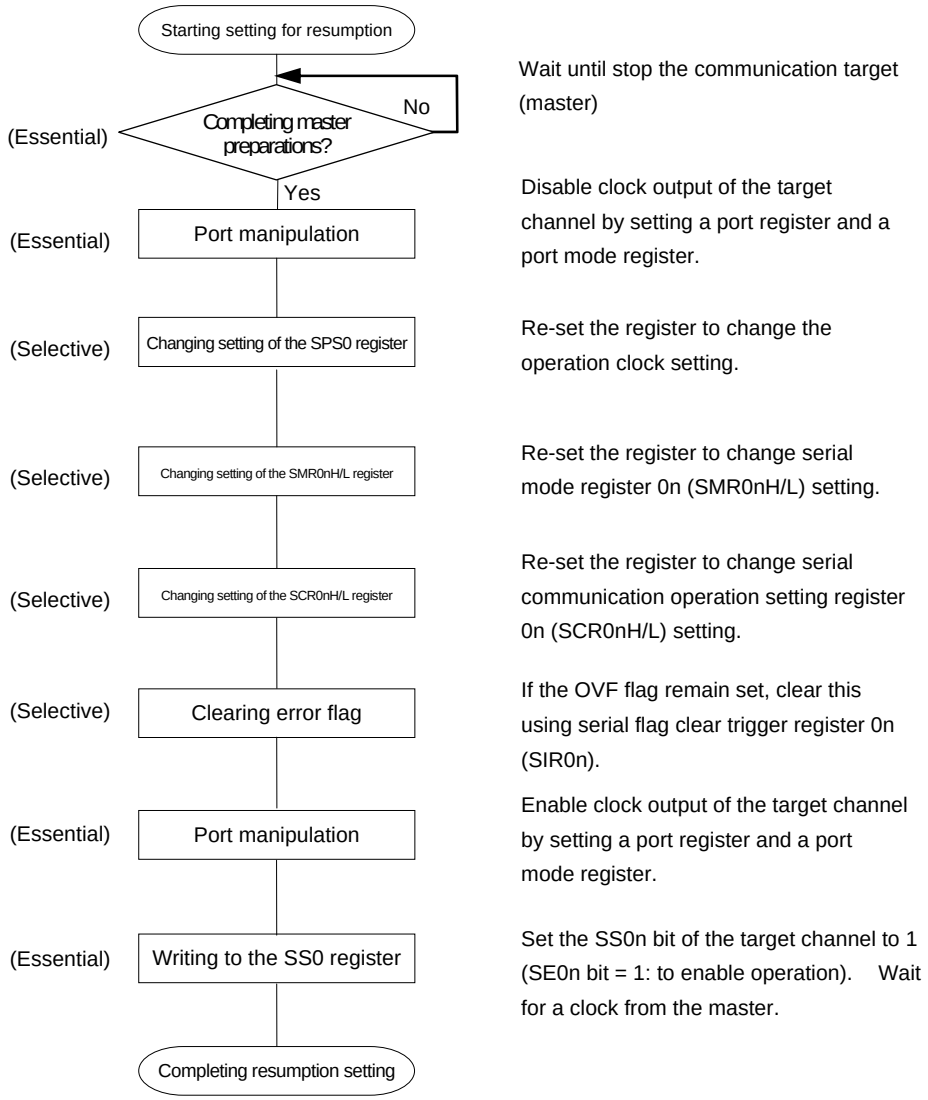


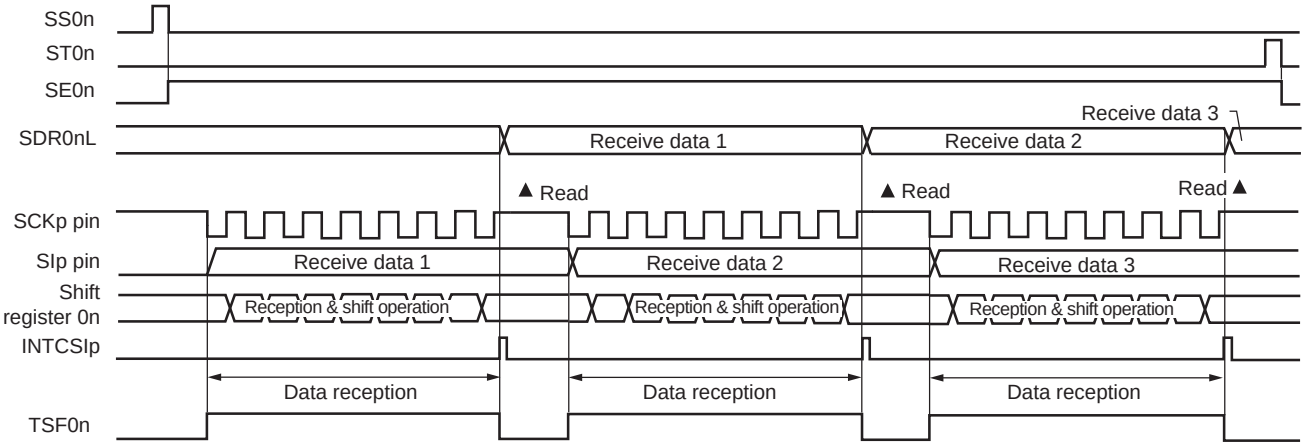
Figure 12-57. Procedure for Resuming Slave Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

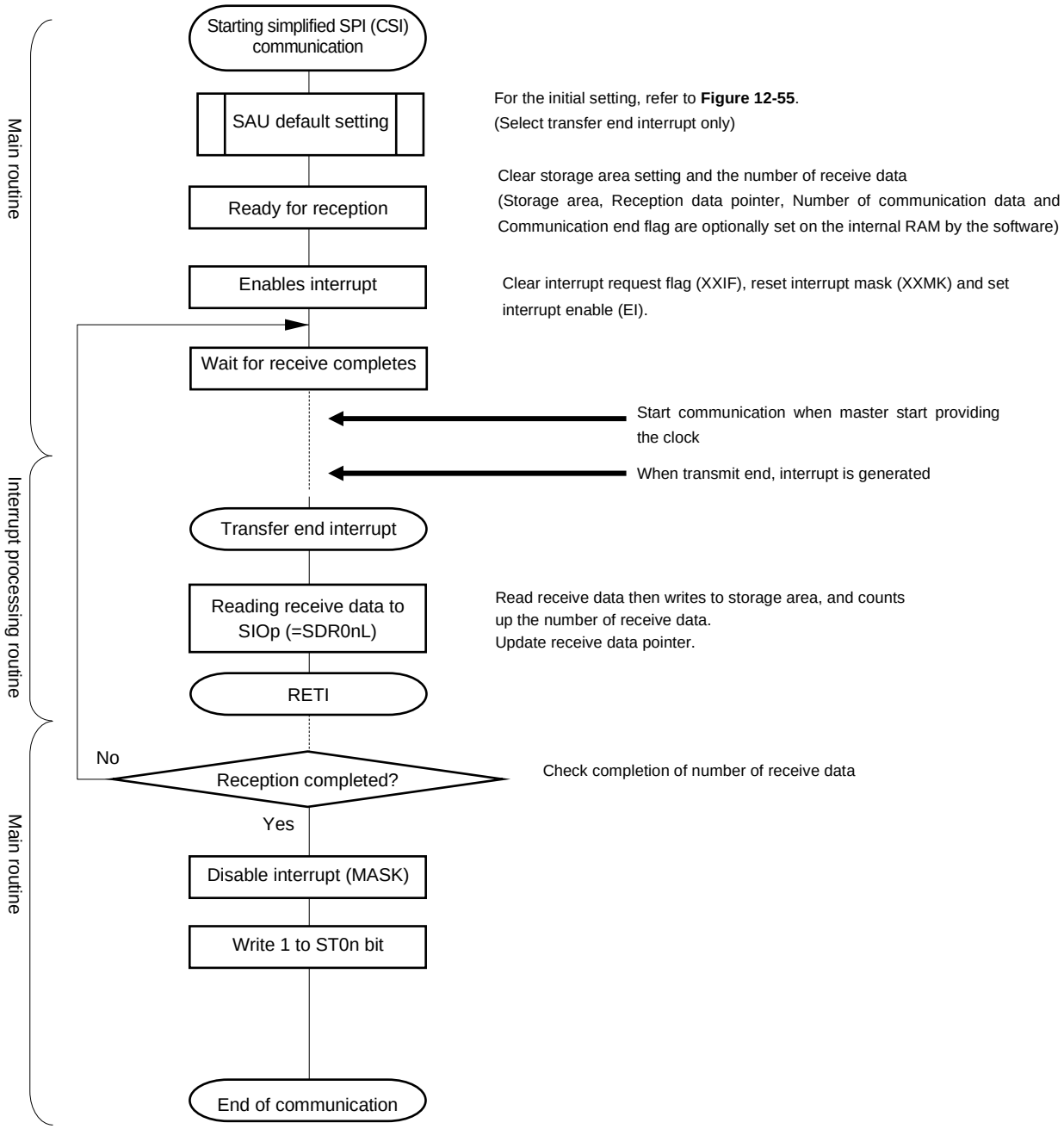
(3) Processing flow (in single-reception mode)

Figure 12-58. Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAP0n = 0, CKP0n = 0)



Remark n = 0, 1, p: CSI number (p = 00, 01)

Figure 12-59. Flowchart of Slave Reception (in Single-Reception Mode)



12.5.6 Slave transmission/reception

Slave transmission/reception is that the RL78/G10 transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01 Note 1
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVF0n) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 2, 3	
Data phase	Selectable by the DAP0n bit of the SCR0nH register • DAP0n = 0: Data output starts at the start of the operation of the serial clock. • DAP0n = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKP0n bit of the SCR0nH register • CKP0n = 0: Non-inversion • CKP0n = 1: Inverted	
Data direction	MSB or LSB first	

Notes 1. 16-pin products only.

2. Because the external serial clock input to the SCK00 and SCK01 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

3. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

2. n = 0, 1

(1) Register setting

Figure 12-60. Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01 Note) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol: SMR0nH

15	14	13	12	11	10	9	8
CKS0n	CCS0n						STS0n
0/1	1	0	0	0	0	0	0

Symbol: SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

15	14	13	12	11	10	9	8
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	1	0/1	0/1	0	0	0	0

Symbol: SCR0nL

7	6	5	4	3	2	1	
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **12.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

15	14	13	12	11	10	9	8
0000000 Baud rate setting							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Transmit data setting/receive data register							

SIOp

(d) Serial clock output register 0 (CKO0) … The register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
CKO0							CKO01 Note	CKO00
	0	0	0	0	0	0	×	×

(e) Serial output register 0 (SO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
SO0							SO01 Note	SO00
	0	0	0	0	0	0	0/1	0/1

(**Note**, **Caution**, and **Remarks** are listed on the next page.)

Figure 12-60. Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01 Note) (2/2)

(f) Serial output enable register 0 (SOE0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SOE0							SOE01 Note	SOE00
	0	0	0	0	0	0	0/1	0/1

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0							SS01	SS00
	0	0	0	0	0	0	0/1	0/1

Note 16-pin products only.

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remarks 1. n = 0, 1, p: CSI number (p = 00, 01)

2. □: Setting is fixed in the simplified SPI (CSI) master transmission/reception mode,

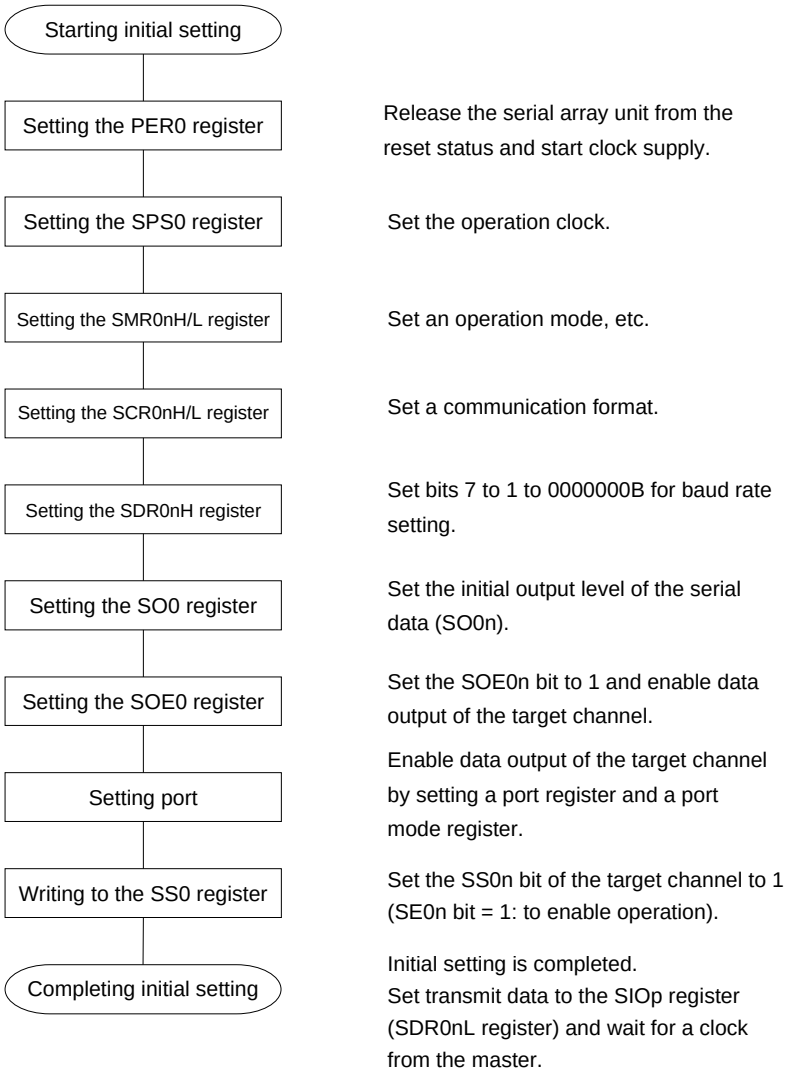
▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-61. Initial Setting Procedure for Slave Transmission/Reception



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Figure 12-62. Procedure for Stopping Slave Transmission/Reception

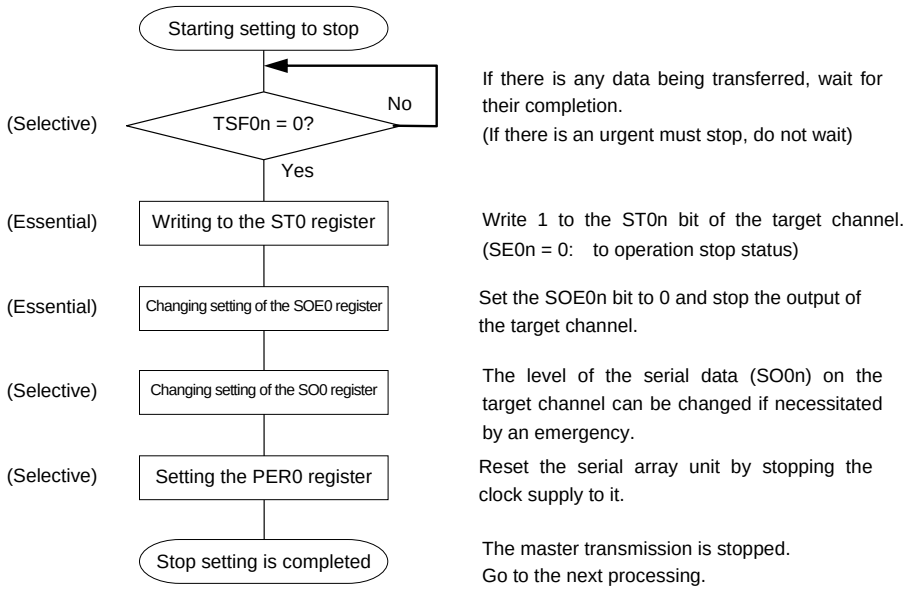
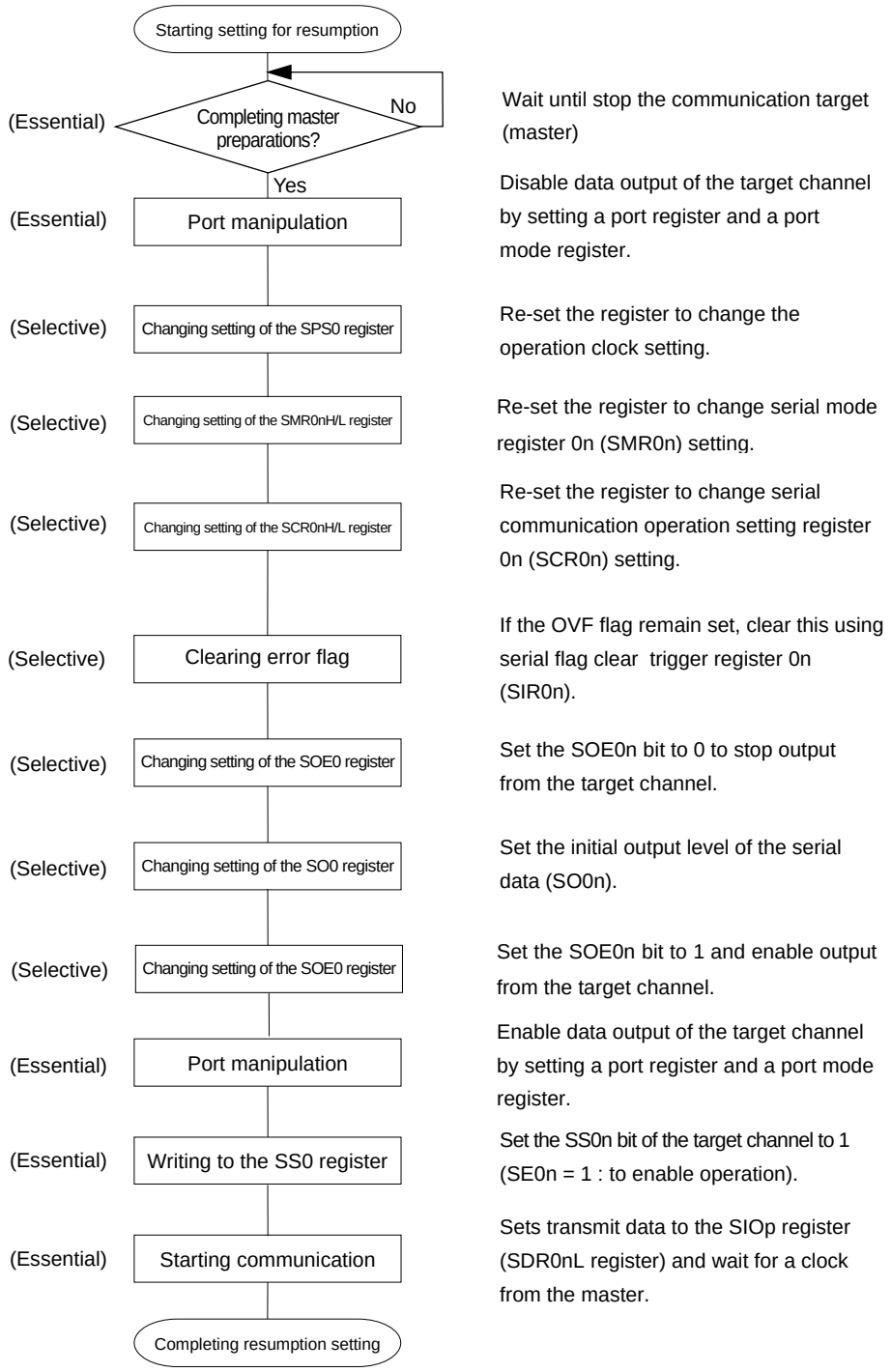


Figure 12-63. Procedure for Resuming Slave Transmission/Reception

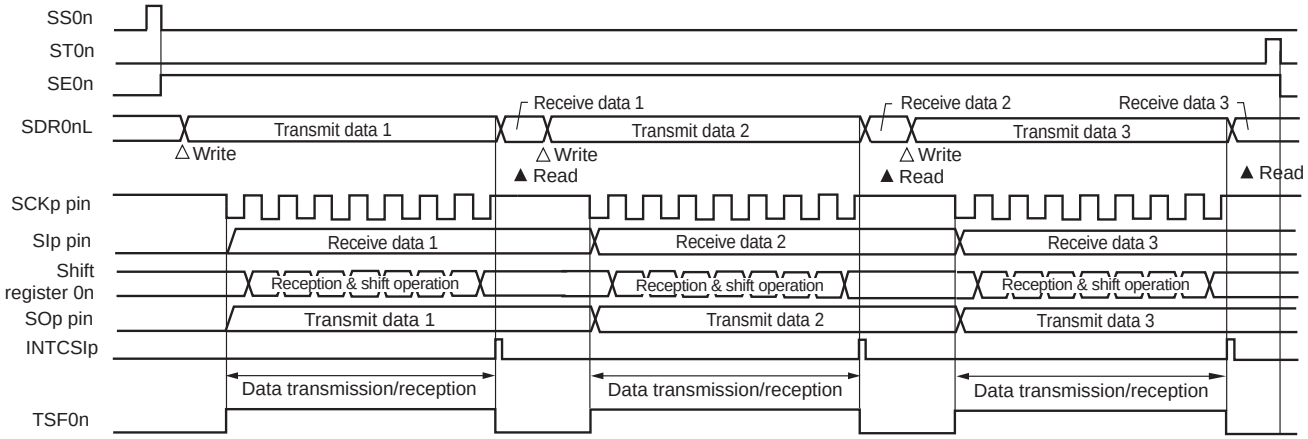


Cautions 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

2. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

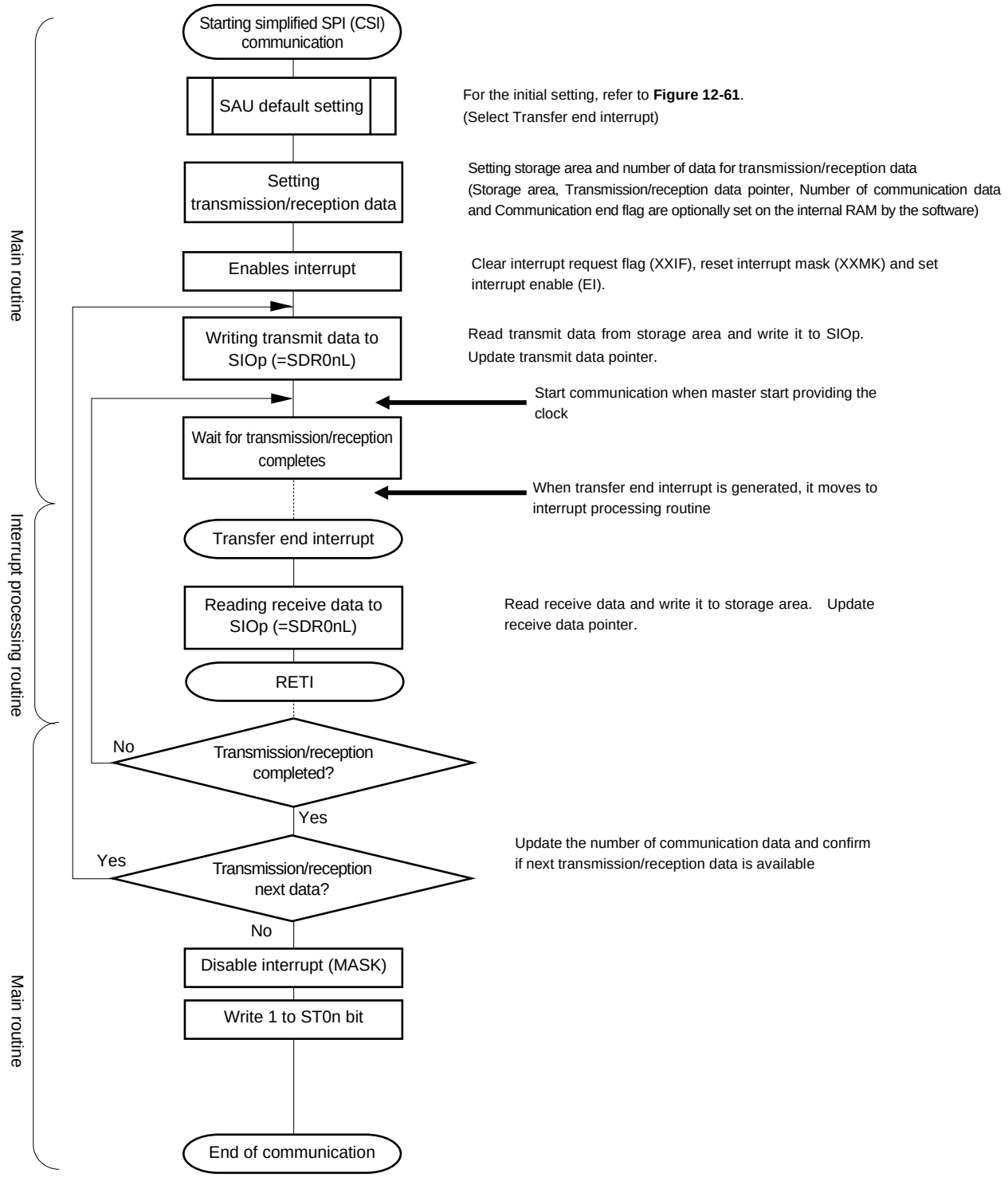
(3) Processing flow (in single-transmission/reception mode)

Figure 12-64. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAP0n = 0, CKP0n = 0)



Remark n = 0, 1, p: CSI number (p = 00, 01)

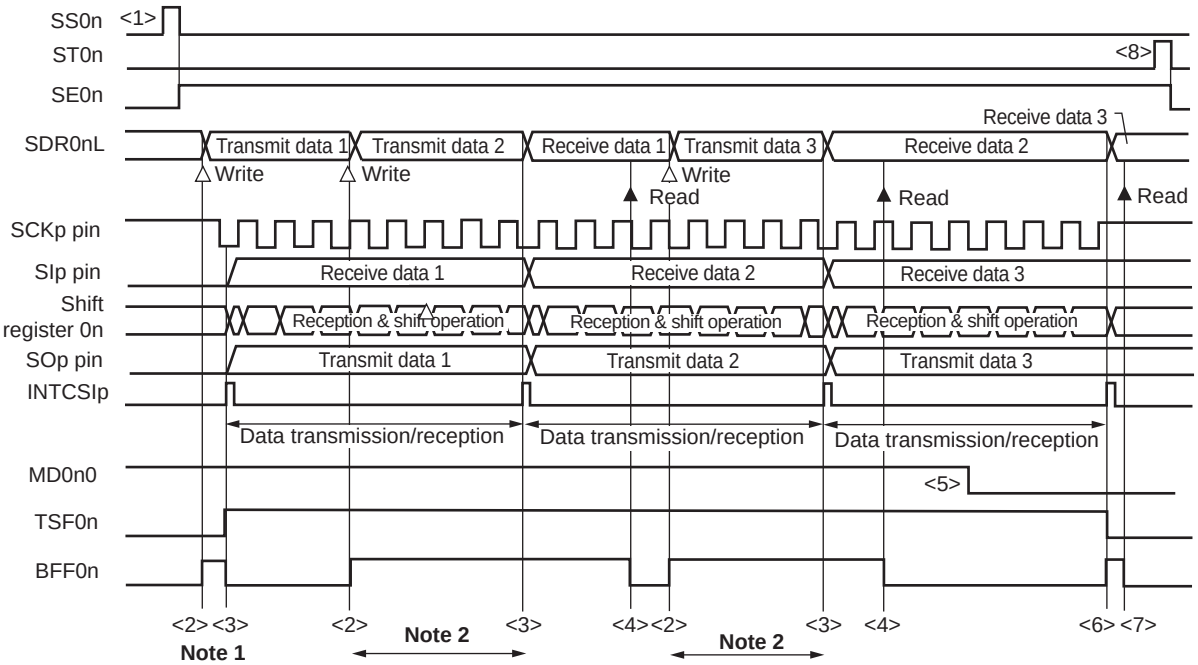
Figure 12-65. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 12-66. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAP0n = 0, CKP0n = 0)



Notes 1. If transmit data is written to the SDR0nL register while the BFF0n bit of serial status register 0n (SSR0n) is 1 (valid data is stored in serial data register 0n (SDR0nL)), the transmit data is overwritten.

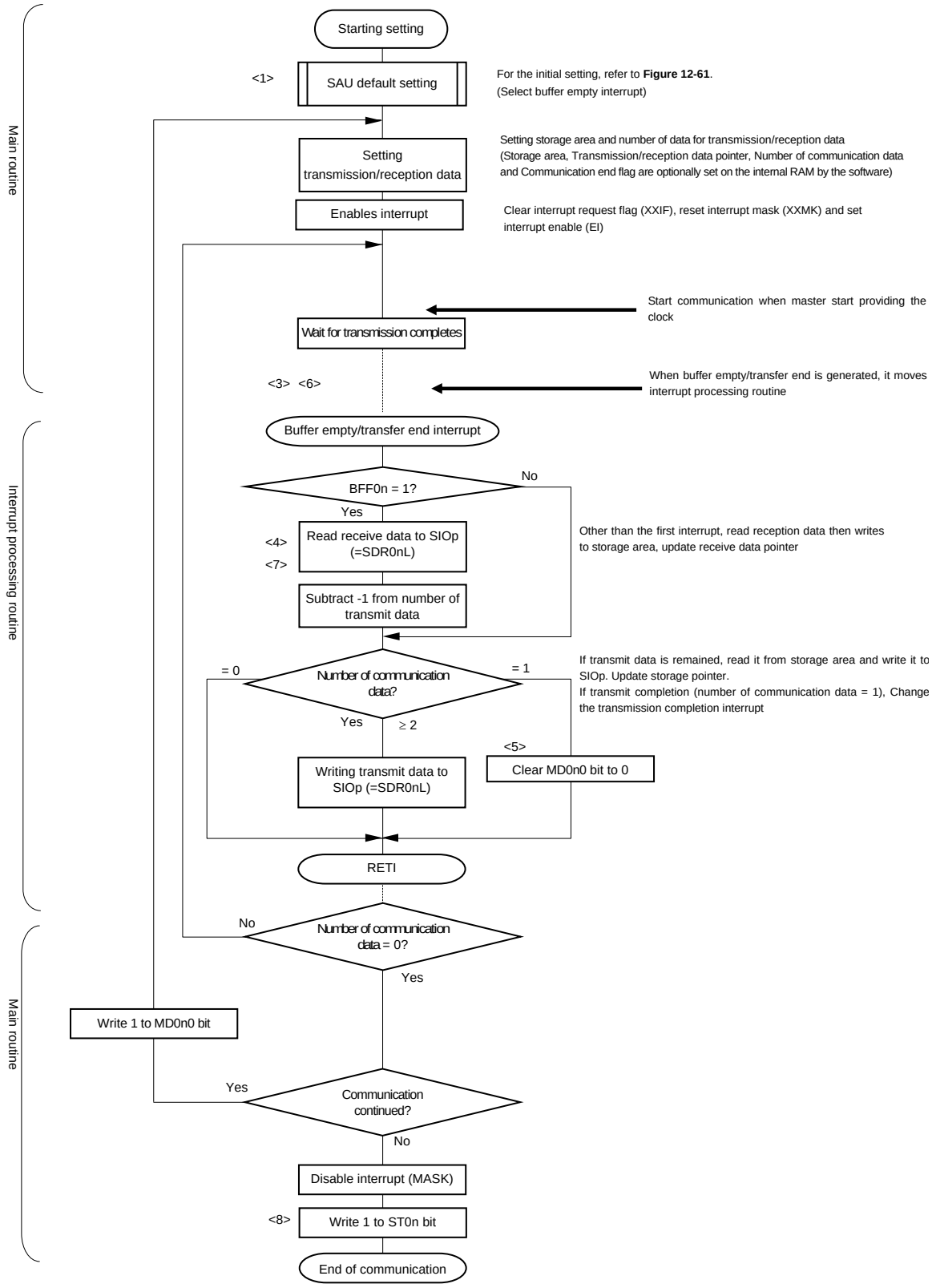
2. The transmit data can be read by reading the SDR0nL register during this period. At this time, the transfer operation is not affected.

Caution The MD0n0 bit of serial mode register 0n (SMR0nL) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-67 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. n = 0, 1, p: CSI number (p = 00, 01)

Figure 12-67. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-66 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

12.5.7 Calculating transfer clock frequency

The transfer clock frequency for simplified SPI (CSI00, CSI01) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDR0nH[7:1] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master}[Note] [Hz]

Note The permissible maximum transfer clock frequency is f_{MCK}/6.

Remark The value of SDR0nH[7:1] is the value of bits 7 to 1 of serial data register 0nH (SDR0nH) (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register 0 (SPS0) and bit 7 (CKS0n) of serial mode register 0n (SMR0nH).

Table 12-2. Selection of Operation Clock For Simplified SPI

SMR0n Register	SPS0 Register								Operation Clock (f_{MCK}) Note	
CKS0n	PRS 13	PRS 12	PRS 11	PRS 10	PRS 03	PRS 02	PRS 01	PRS 00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	312.5 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156.2 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	4.88 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.44 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.22 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	610 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	312.5 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156.2 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	4.88 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.44 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.22 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	610 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register 0 (ST0) = 03H) the operation of the serial array unit (SAU).

Remarks 1. X: don't care

2. n = 0, 1

12.5.8 Procedure for processing errors that occurred during simplified SPI (CSI00, CSI01 Note) communication

The procedure for processing errors that occurred during simplified SPI (CSI00, CSI01 Note) communication is described in Figure 12-68.

Figure 12-68. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register 0n (SDR0nL). →	The BFF0n bit of the SSR0n register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register 0n (SSR0n).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register 0n (SIR0n). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR0n register to the SIR0n register without modification.

Note 16-pin products only.

Remark n = 0, 1

12.6 Operation of UART (UART0) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TXD) and serial data reception (RXD) lines. By using these two communication lines, each data frame, which consists of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (an even-numbered channel) and a channel dedicated to reception (an odd-numbered channel).

[Data transmission/reception]
- Data length of 7 or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending and stop bit check functions

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]
- Framing error, parity error, or overrun error

The ISC register can be used to set up the input signal on the RxD0 pin of UART0 as an external interrupt input or as a timer input for the timer array unit. The input pulse interval measurement mode of the timer array unit can then be used to measure the width at the baud rate of the other party in communications and make the required adjustments in response.

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		–

Caution When UART operation is selected, the even-numbered channel can only be used for transmission and the odd-numbered channel can only be used for reception.

UART performs the following four types of communication operations.
- UART transmission (See **12.6.1**.)
- UART reception (See **12.6.2**.)

12.6.1 UART transmission

UART transmission is an operation to transmit data from the RL78/G10 to another device asynchronously (start-stop synchronization).

Of the two channels used for UART, the even-numbered channel is used for UART transmission.

UART	UART0
Target channel	Channel 0 of SAU0
Pins used	TxD0
Interrupt	INTST0
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	None
Transfer data length	7 or 8 bits (UART0 only)
Transfer rate Note	Max. $f_{MCK}/6$ [bps] (SDR0nH[7:1] = 2 or greater, Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]
Data phase	Non-inverted output (default: high level) Inverted output (default: low level)
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity
Stop bit	The following selectable • Appending 1 bit • Appending 2 bits
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the peripheral function characteristics in the electrical specifications (see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 0)

(1) Register setting

Figure 12-69. Example of Contents of Registers for UART Transmission (UART0) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol: SMR0nH

15	14	13	12	11	10	9	8
CKS0n	CCS0n						
0/1	0	0	0	0	0	0	0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Symbol: SMR0nL

7	6	5	4	3	2	1	0
						MD0n1	MD0n0
0	0	1	0	0	0	1	0/1

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

15	14	13	12	11	10	9	8
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	0	0	0	0	0	0/1	0/1

Setting of parity bit
00B: No parity
01B: Appending 0 parity
10B: Appending Even parity
11B: Appending Odd parity

Symbol: SCR0nL

7	6	5	4	3	2	1	
DIR0n		SLC0n1	SLC0n0				DLS0n0
0/1	0	0/1	0/1	0	1	1	0/1

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of stop bit
01B: Appending 1 bit
10B: Appending 2 bits

(c) Serial data register 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

15	14	13	12	11	10	9	8
Baud rate setting							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Transmit data setting							

TXD0

(d) Serial output level register 0 (SOL0)… Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
SOL0								SOL00
	0	0	0	0	0	0	0	0/1

0: Non-inverted transmission
1: Inverted transmission

Remarks 1. n = 0

2. □ : Setting is fixed in the simplified SPI (CSI) master transmission mode,
▒ : Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-69. Example of Contents of Registers for UART Transmission (UART0) (2/2)

(e) Serial clock output register 0 (CKO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
CKO0	0	0	0	0	0	0	CKO01 ×	CKO00 ×

(f) Serial output register 0 (SO0) … Sets only the bits of the target channel.

Symbol:	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	SO01 ×	SO00 0/1

0: Serial data output value is “0”
1: Serial data output value is “1”

(g) Serial output enable register 0 (SOE0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 ×	SOE00 Note 0/1

(h) Serial channel start register 0 (SS0) … Sets only the bits of the target channel to 1.

Symbol:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 ×	SS00 0/1

Note Before transmission is started, be sure to set to 1 when the SOL00 bit of the target channel is set to 0, and set to 0 when the SOL00 bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remarks 1. n = 0

2. ☐: Setting is fixed in the simplified SPI (CSI) master transmission mode,
▒ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-70. Initial Setting Procedure for UART Transmission

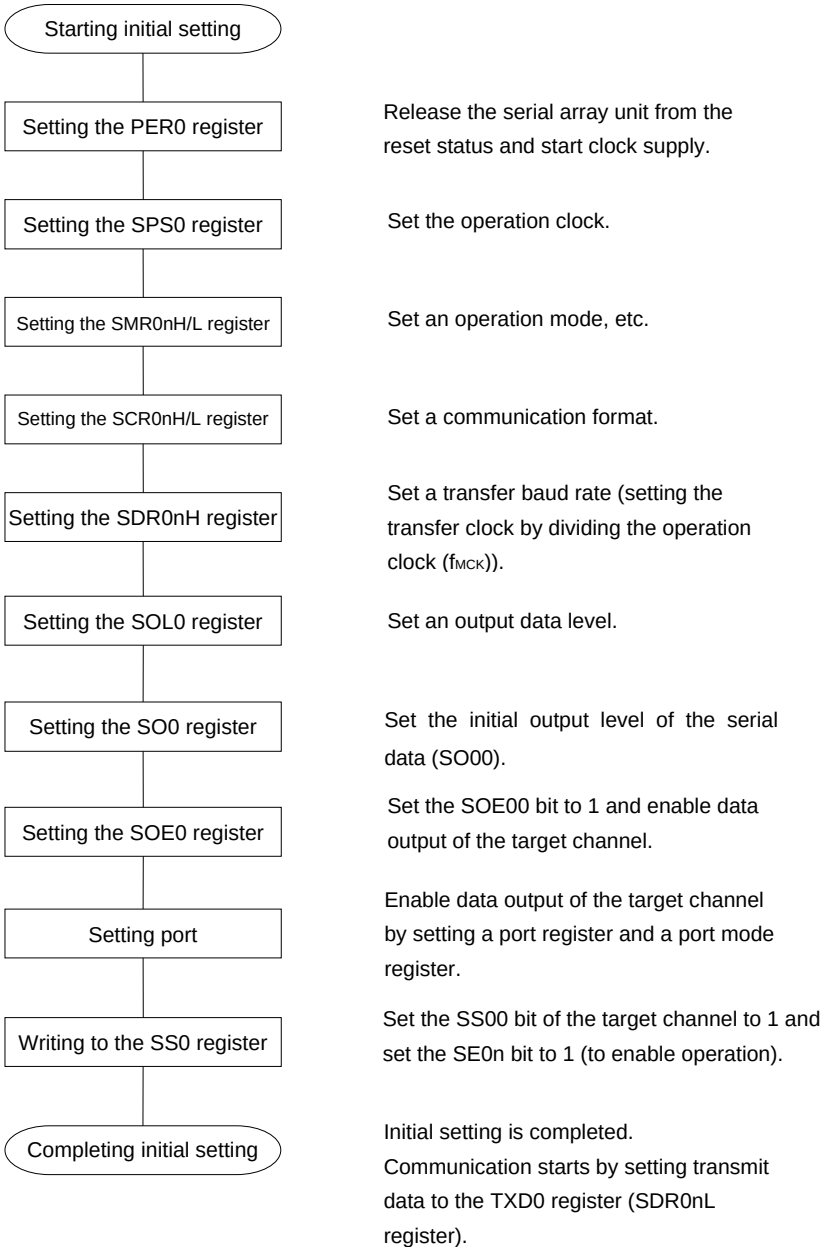


Figure 12-71. Procedure for Stopping UART Transmission

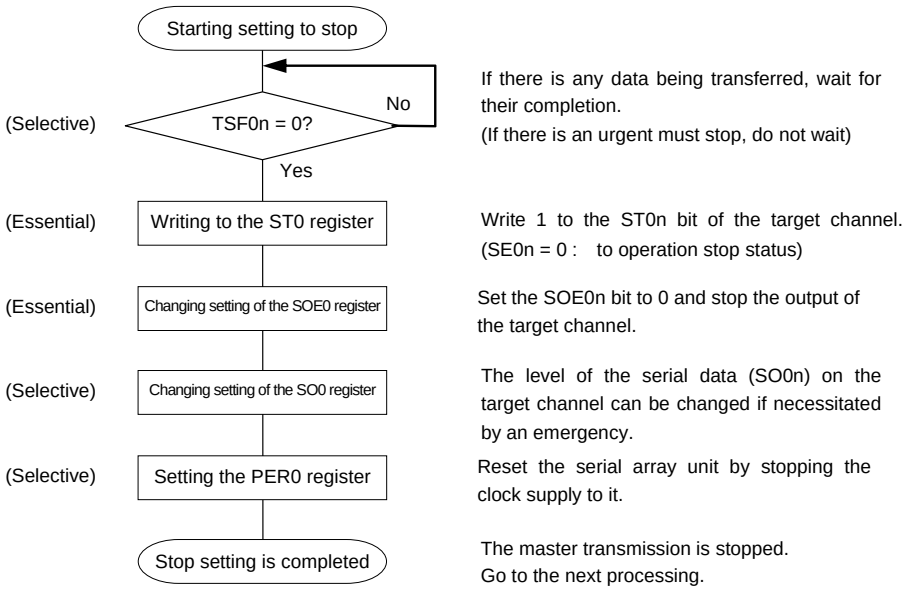
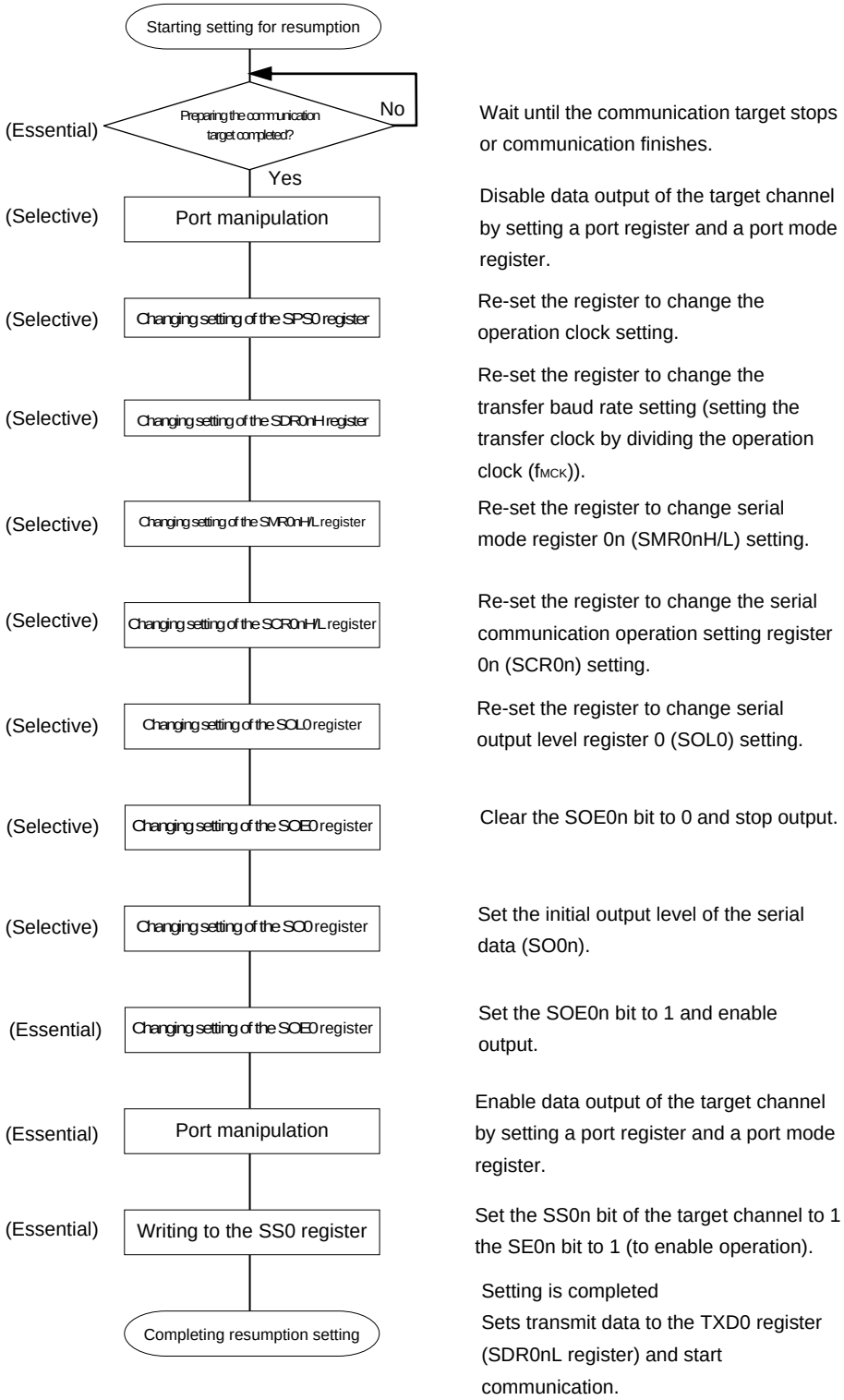


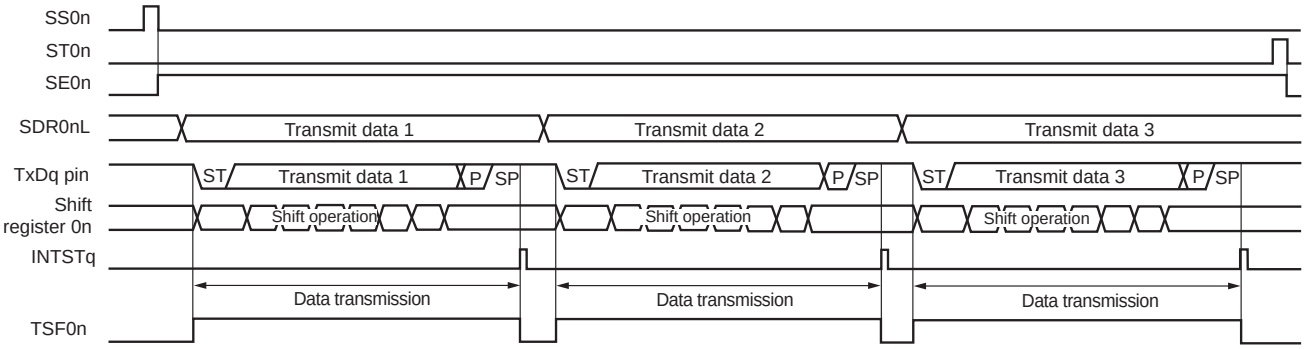
Figure 12-72. Procedure for Resuming UART Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

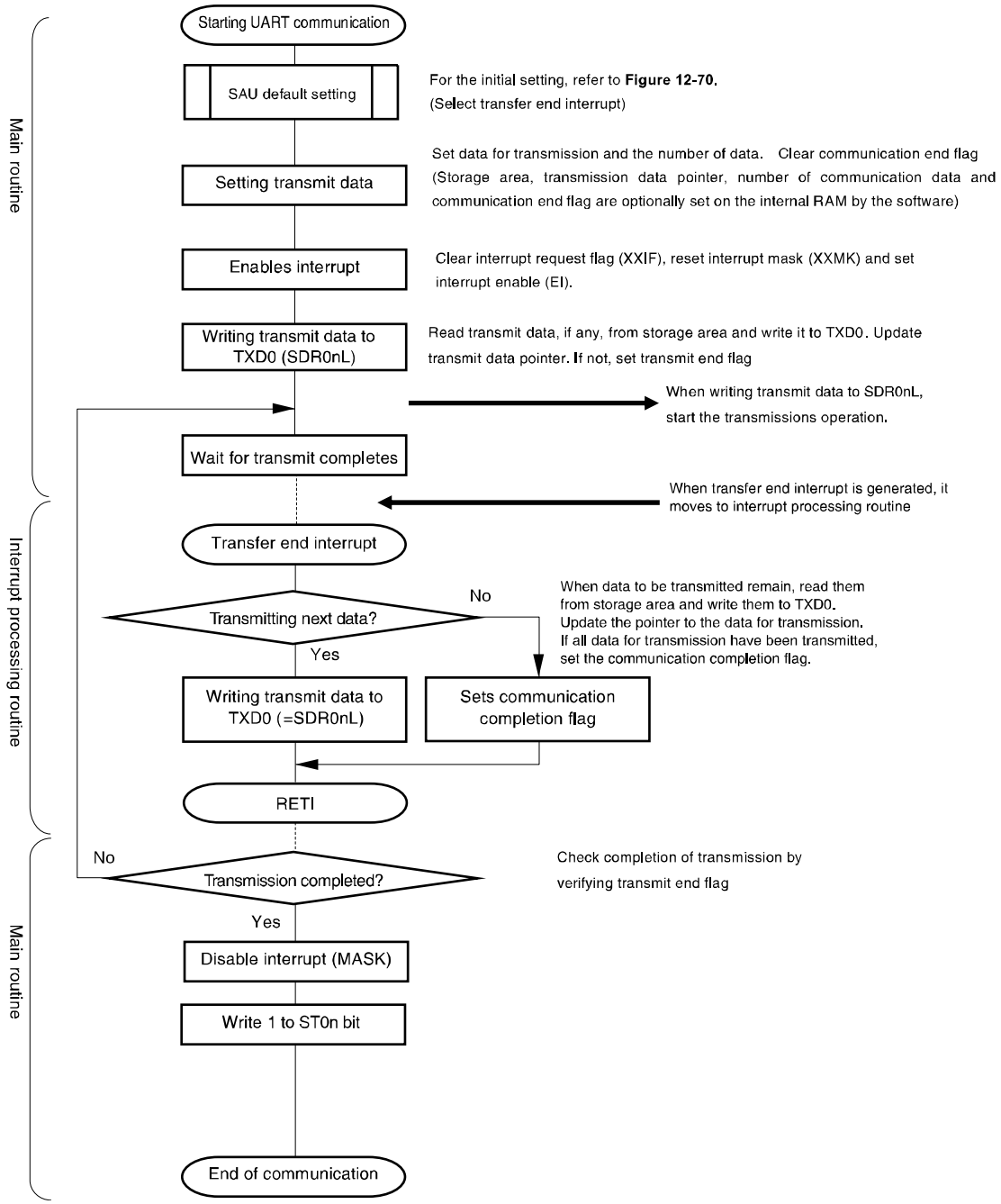
(3) Processing flow (in single-transmission mode)

Figure 12-73. Timing Chart of UART Transmission (in Single-Transmission Mode)



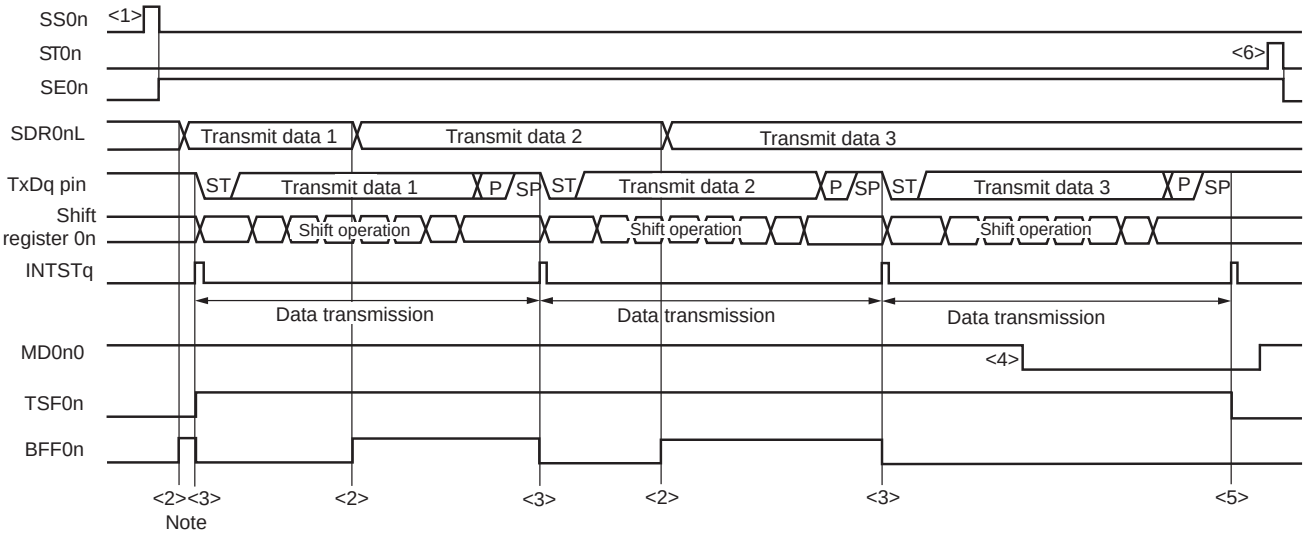
Remark q: UART number (q = 0), n = 0

Figure 12-74. Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 12-75. Timing Chart of UART Transmission (in Continuous Transmission Mode)

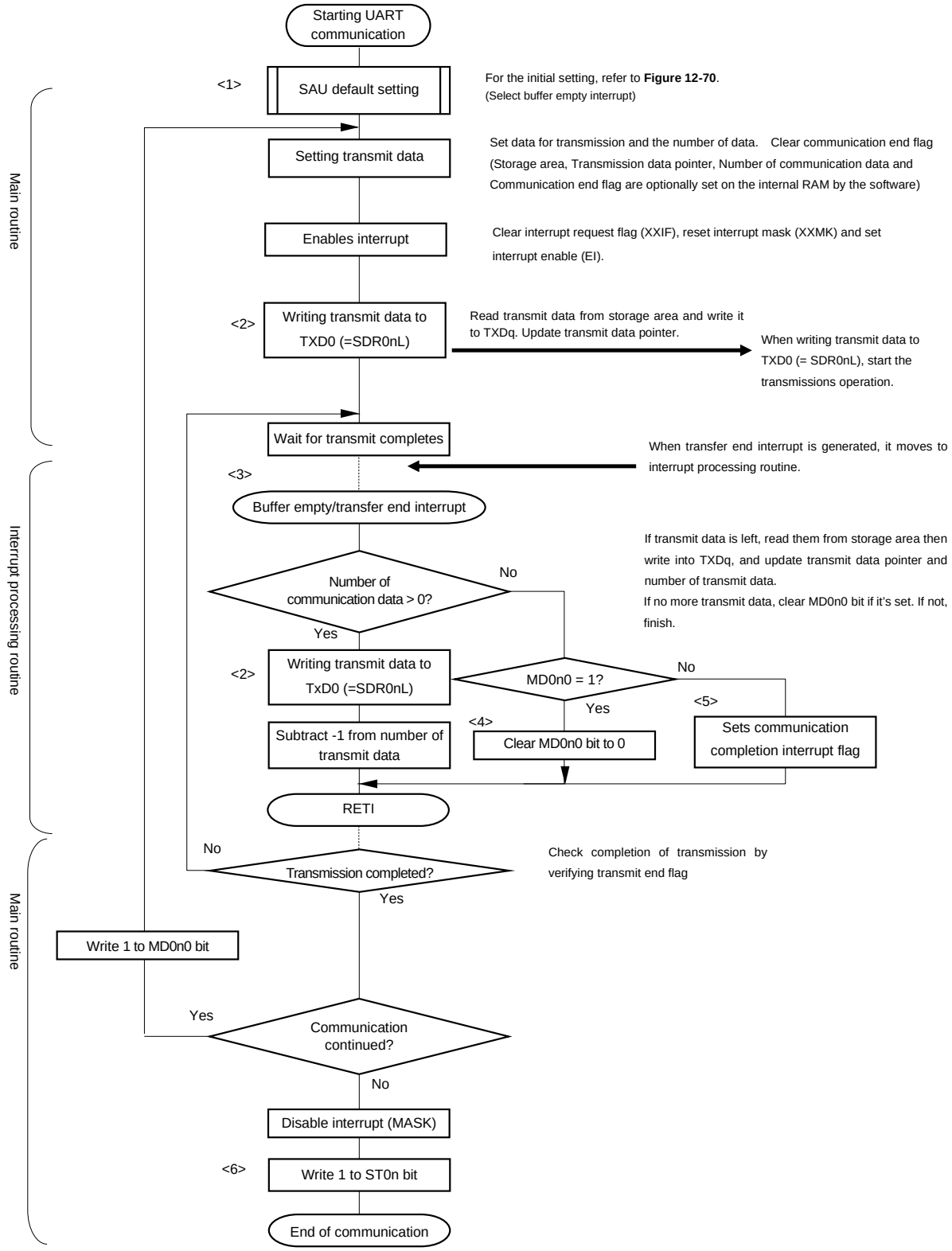


Note If transmit data is written to the SDR0nL register while the BFF0n bit of serial status register 0n (SSR0n) is 1 (valid data is stored in serial data register 0n (SDR0nL)), the transmit data is overwritten.

Caution **The MD0n0 bit of serial mode register 0n (SMR0nL) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.**

Remark q: UART number (q = 0), n = 0

Figure 12-76. Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-75 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

12.6.2 UART reception

UART reception is an operation wherein the RL78/G10 asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of both the odd- and even-numbered channels must be set.

UART	UART0
Target channel	Channel 1 of SAU0
Pins used	RxD0
Interrupt	INTSR0
	Transfer end interrupt only (setting the buffer empty interrupt is prohibited)
Error interrupt	INTSRE0
Error detection flag	• Framing error detection flag (FEF0n) • Parity error detection flag (PEF0n) • Overrun error detection flag (OVF0n)
Transfer data length	7 or 8 bits (UART0 only)
Transfer rate Note	Max. $f_{MCK}/6$ [bps] (SDR0nH[7:1] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]
Data phase	Non-inverted output (default: high level) Inverted output (default: low level)
Parity bit	The following selectable • No parity check • No parity specified (0 parity) • Appending even parity • Appending odd parity
Stop Bit	1 bit check
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the peripheral characteristics in the electrical specifications (see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**).

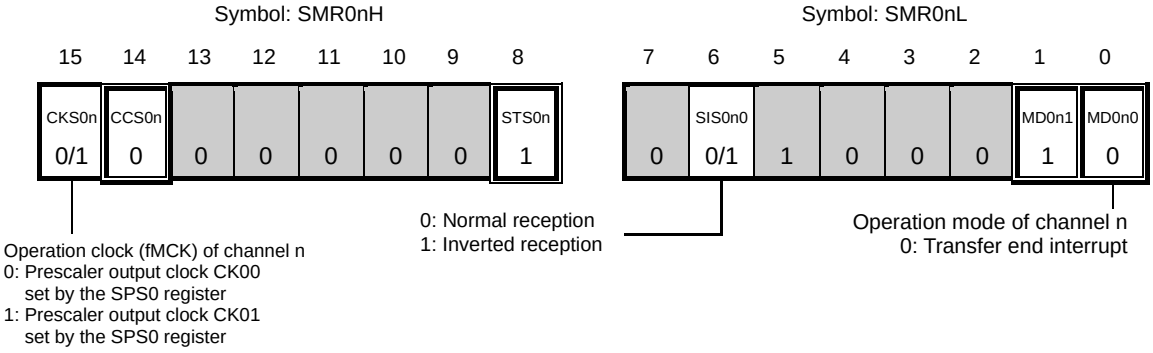
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. n: Channel number (n = 1)

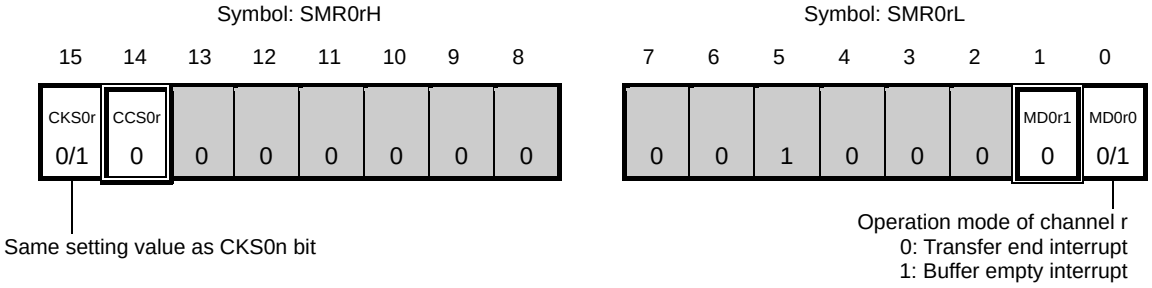
(1) Register setting

Figure 12-77. Example of Contents of Registers for UART Reception (UART0) (1/2)

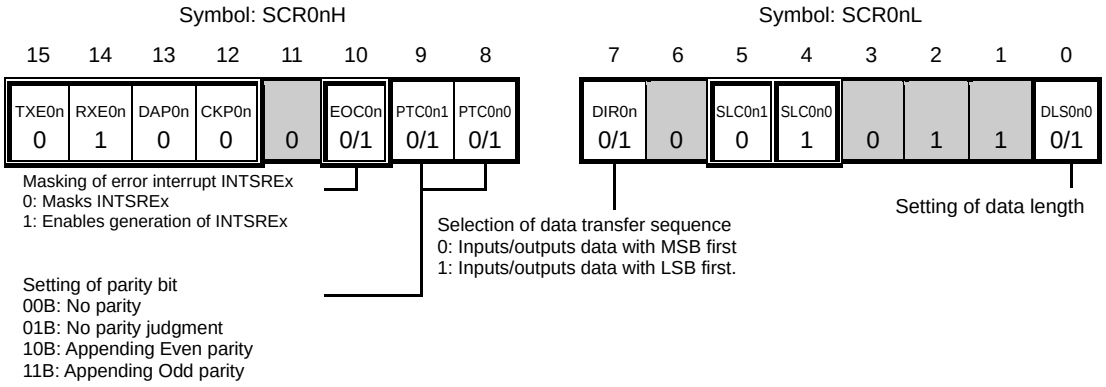
(a) Serial mode register 0n (SMR0nH, SMR0nL)



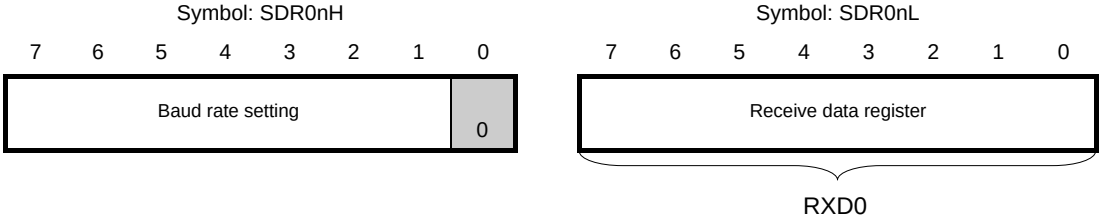
(b) Serial mode register 0r (SMR0rH, SMR0rL)



(c) Serial communication operation setting register 0n (SCR0nH, SCR0nL)



(d) Serial data register 0n (SDR0nH, SDR0nL)



Caution For UART reception, be sure to set the SMR0r register of channel r that is to be paired with channel n.

Remarks 1. n: Channel number (n = 1),
r: Channel number (r = n - 1) q: UART number (q = 0)

2. ☐: Setting is fixed in the UART master transmission mode, ▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-77. Example of Contents of Registers for UART Reception (UART0) (2/2)

(e) Serial clock output register 0 (CKO0) … The register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
CKO0	0	0	0	0	0	0	CKO01 ×	CKO00 ×

(f) Serial output register 0 (SO0) … The register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	SO01 ×	SO00 ×

(g) Serial output enable register 0 (SOE0) …The register that not used in this mode.

Symbol:	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	SOE01 ×	SOE00 ×

(h) Serial channel start register 0 (SS0) … Sets only the bits of the target channel is 1.

Symbol:	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	SS01 0/1	SS00 ×

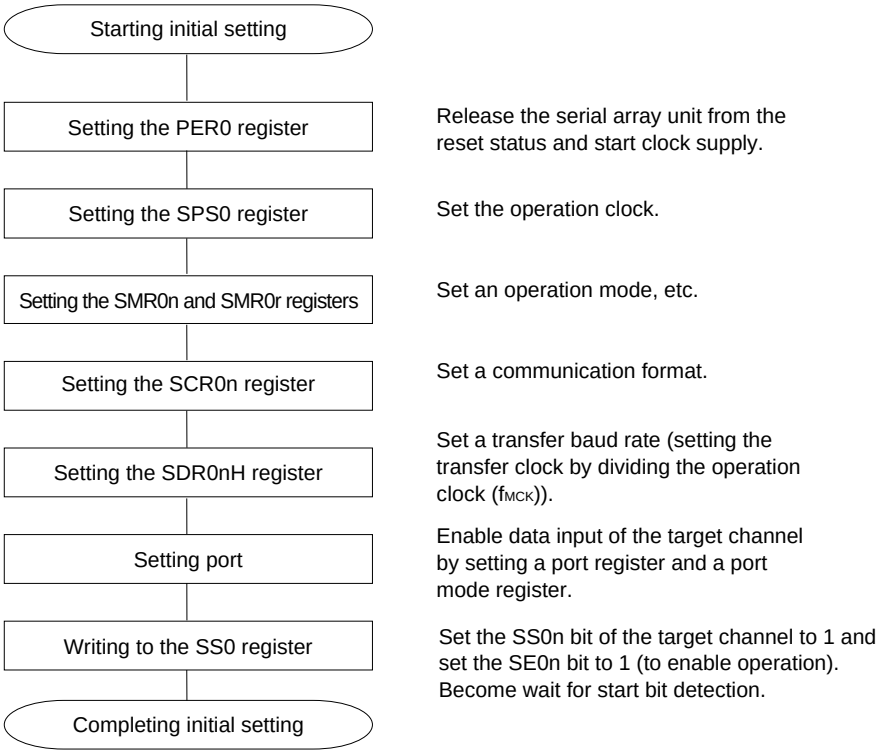
Caution For UART reception, be sure to set the SMR0r register of channel r that is to be paired with channel 0.

Remarks 1. n: Channel number (n = 1)
r: Channel number (r = n - 1) q: UART number (q = 0)

2. □: Setting is fixed in the UART master reception mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 12-78. Initial Setting Procedure for UART Reception



Caution After setting the RXE0n bit of SCR0n register to 1, be sure to set SS0n to 1 after 4 or more f_{MCK} clocks have elapsed.

Figure 12-79. Procedure for Stopping UART Reception

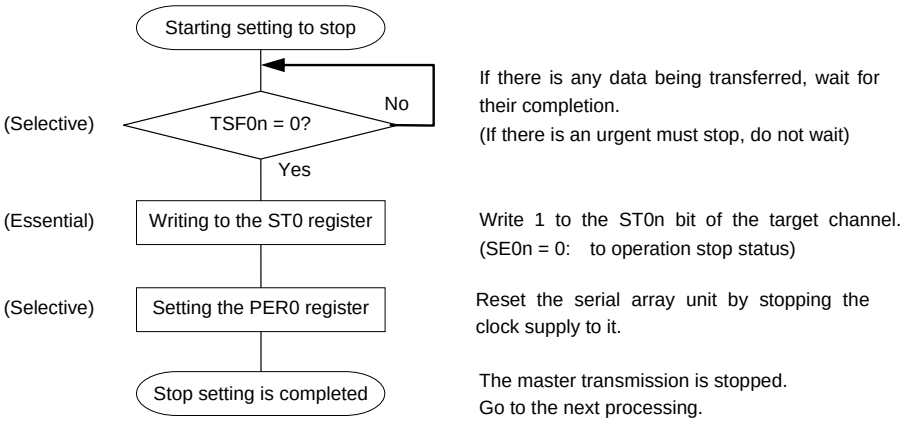
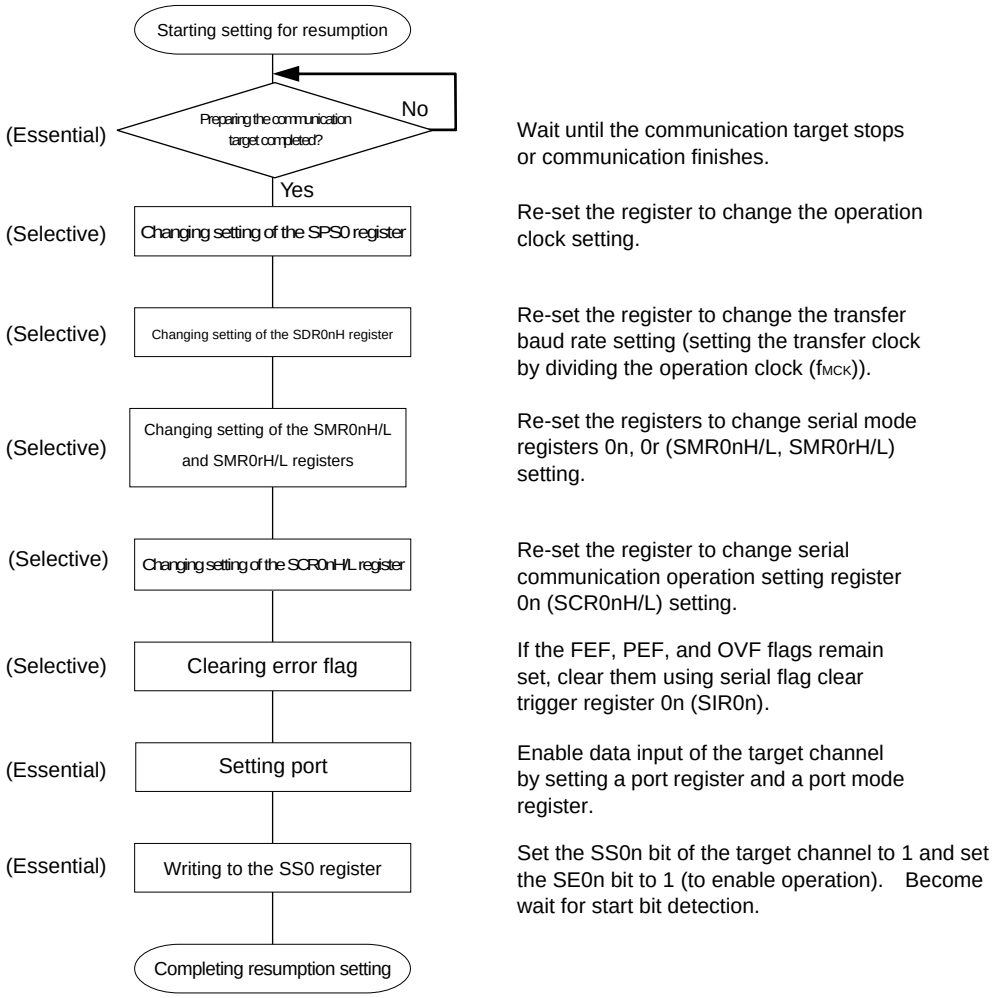


Figure 12-80. Procedure for Resuming UART Reception

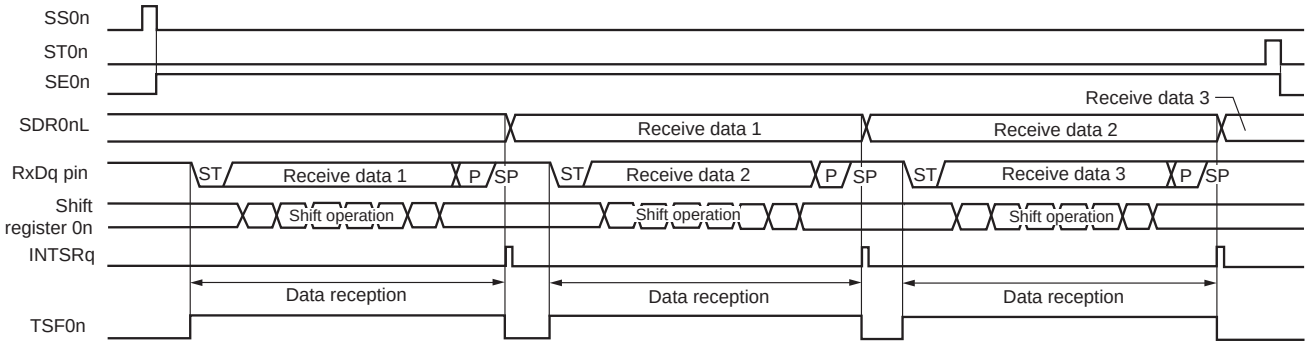


Caution After setting the RXE0n bit of SCR0n register to 1, be sure to set SS0n to 1 after 4 or more f_{CLK} clocks have elapsed.

Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

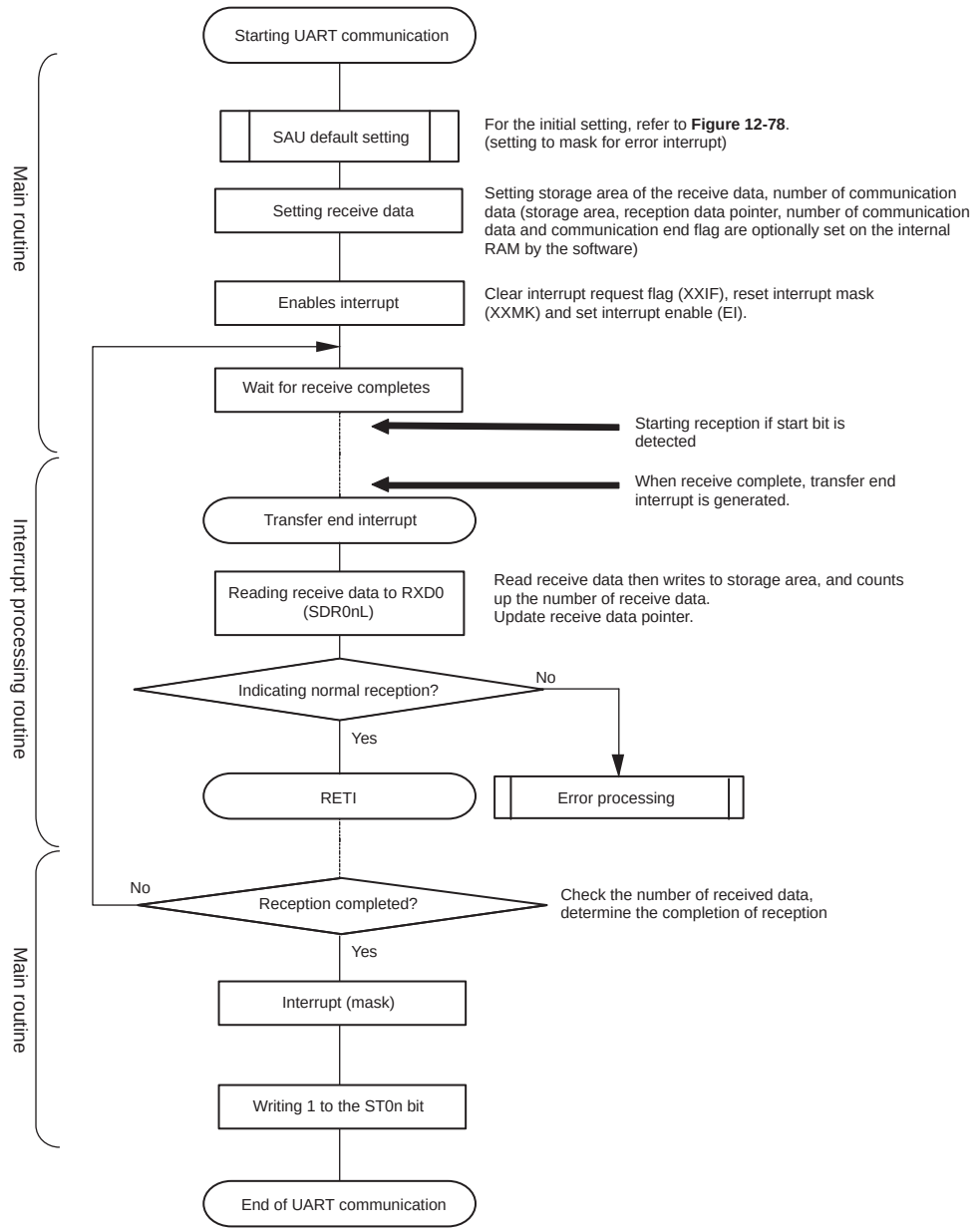
(3) Processing flow

Figure 12-81. UART Reception Timing Chart



Remark n: Channel number (n = 1)
r: Channel number (r = n - 1) q: UART number (q = 0)

Figure 12-82. Flowchart of UART Reception



12.6.3 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDR0nH[7:1] + 1) ÷ 2 [bps]

Caution Setting serial data register 0n (SDR0nH) SDR0nH[7:1] = (0000000B, 0000001B) is prohibited.

Remarks 1. When UART is used, the value of SDR0nH[7:1] is the value of bits 7 to 1 of the SDR0nH register (0000010B to 1111111B) and therefore is 2 to 127.

2. n = 0, 1

The operation clock (f_{MCK}) is determined by serial clock select register 0 (SPS0) and bit 15 (CKS0n) of serial mode register 0n (SMR0n).

Table 12-3. Selection of Operation Clock For UART

SMR0n Register	SPS0 Register								Operation Clock (f_{MCK}) Note	
CKS0n	PRS 13	PRS 12	PRS 11	PRS 10	PRS 03	PRS 02	PRS 01	PRS 00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	312.5 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156.2 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	4.88 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.44 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.22 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	610 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 MHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	312.5 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156.2 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	4.88 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.44 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.22 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	610 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register 0 (ST0) = 03H) the operation of the serial array unit (SAU).

Remarks 1. X: don't care

2. n = 0, 1

(2) Baud rate error during transmission

The baud rate error of UART (UART0) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 – 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 20 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 20 MHz			
	Operation Clock (f_{MCK})	SDR0nH[7:1]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	64	300.48 bps	+0.16%
600 bps	$f_{CLK}/2^8$	64	600.96 bps	+0.16%
1200 bps	$f_{CLK}/2^7$	64	1201.92 bps	+0.16%
2400 bps	$f_{CLK}/2^6$	64	2403.85 bps	+0.16%
4800 bps	$f_{CLK}/2^5$	64	4807.69 bps	+0.16%
9600 bps	$f_{CLK}/2^4$	64	9615.38 bps	+0.16%
19200 bps	$f_{CLK}/2^3$	64	19230.8 bps	+0.16%
31250 bps	$f_{CLK}/2^3$	39	31250.0 bps	±0.0%
38400 bps	$f_{CLK}/2^2$	64	38461.5 bps	+0.16%
76800 bps	$f_{CLK}/2$	64	76923.1 bps	+0.16%
153600 bps	f_{CLK}	64	153846 bps	+0.16%
312500 bps	f_{CLK}	31	312500 bps	±0.0%

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

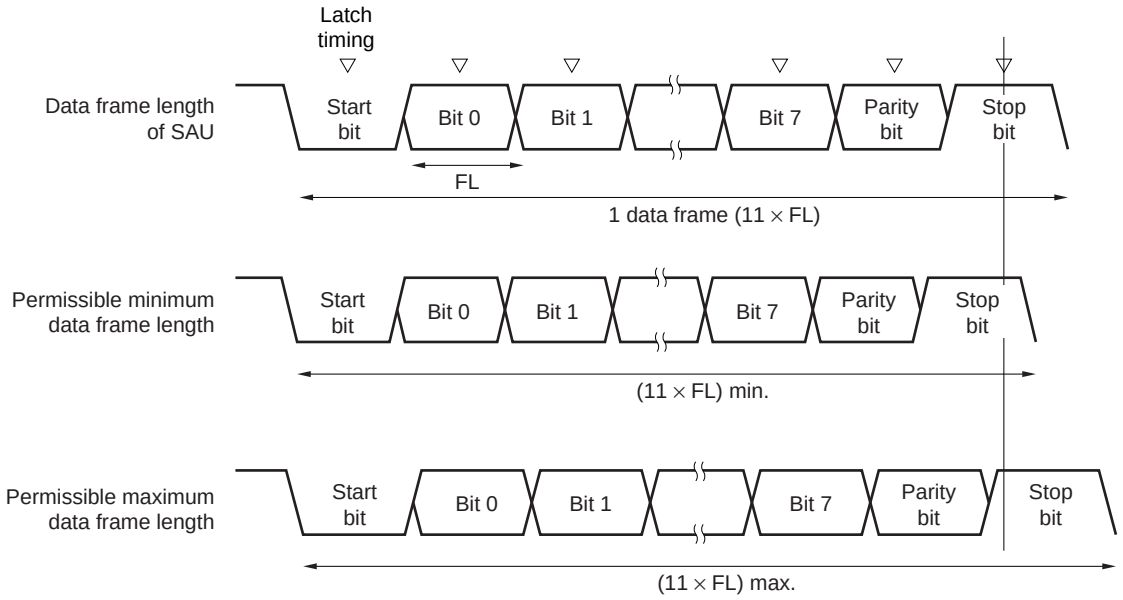
$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **12.6.3 (1) Baud rate calculation expression**.)
k: SDR0nH[7:1] + 1
Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark n = 1

Figure 12-83. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 12-83 the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of serial data register 0nH (SDR0nH) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

12.6.4 Procedure for processing errors that occurred during UART (UART0) communication

The procedure for processing errors that occurred during UART (UART0) communication is described in Figures 12-84 and 12-85.

Figure 12-84. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register 0n (SDR0nL). →	The BFF0n bit of the SSR0n register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register 0n (SSR0n).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register 0n (SIR0n). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR0n register to the SIR0n register without modification.

Figure 12-85. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register 0n (SDR0nL). →	The BFF0n bit of the SSR0n register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register 0n (SSR0n).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register 0n (SIR0n). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSR0n register to the SIR0n register without modification.
Sets the ST0n bit of serial channel stop register 0 (ST0) to 1. →	The SE0n bit of serial channel enable status register 0 (SE0) is set to 0 and channel n stops operating.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SS0n bit of serial channel start register 0 (SS0) to 1. →	The SE0n bit of serial channel enable status register 0 (SE0) is set to 1 and channel n is enabled to operate.	

Remark n = 0, 1

12.7 Operation of Simplified I²C (IIC00) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master.

Operate the control registers by software for setting the start and stop conditions while observing the specifications of the I²C bus line.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note] and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Generation of start condition and stop condition for software

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Overrun error
- ACK error

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Multi master function (Arbitration loss detection function)
- Clock stretch detection function

Note When receiving the last data, ACK will not be output if 0 is written to the SOE00 bit in the SOE0 register and serial communication data output is stopped. See **12.7.3 (2) Processing flow** for details.

Remark Full I²C functions are described in **CHAPTER 13 SERIAL INTERFACE IICA** (16-pin products only).

The channels supporting simplified I²C (IIC00) are channel 0 of SAU0.

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	CSI01		–

Simplified I²C (IIC00) performs the following four types of communication operations.

- Address field transmission (See **12.7.1**.)
- Data transmission (See **12.7.2**.)
- Data reception (See **12.7.3**.)
- Stop condition generation (See **12.7.4**.)

12.7.1 Address field transmission

Address field transmission is a transmission operation that first executes in I²C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I²C	IIC00
Target channel	Channel 0 of SAU0
Pins used	SCL00, SDA00[Note]
Interrupt	INTIIC00
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	ACK error detection flag (PEF0n)
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)
Transfer rate	Max. f_{MCK}/4 [Hz] (SDR0nH[7:1] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I²C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)
Data level	Non-inversion output (default: high level)
Parity bit	No parity bit
Stop bit	Appending 1 bit (for ACK transmission/reception timing)
Data direction	MSB first

Note To perform communication via simplified I²C, set the N-ch open-drain output (V_{DD} tolerance) mode (POM01 = 1) for the port output mode register (POM0) (see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When an Alternate Function Is Used** for details).

Remark n = 0

(1) Register setting

Figure 12-86. Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL)

Symbol: SMR0nH

15	14	13	12	11	10	9	8
CKS0n	CCS0n						STS0n
0/1	0	0	0	0	0	0	0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CK00 set by the SPS0 register
1: Prescaler output clock CK01 set by the SPS0 register

Symbol: SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0

Operation mode of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL)

Symbol: SCR0nH

15	14	13	12	11	10	9	8
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	0	0	0	0	0	0	0

Setting of parity bit
00B: No parity

Symbol: SCR0nL

7	6	5	4	3	2	1	
DIR0n		SLC0n1	SLC0n0				DLS0n0
0	0	0	1	0	1	1	1

Setting of stop bit
01B: Appending 1 bit (ACK)

(c) Serial data register 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

15	14	13	12	11	10	9	8
Baud rate setting							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Transmit data setting (address + R/W)							

SIOr

(d) Serial clock output register 0 (CKO0)

Symbol:	7	6	5	4	3	2	1	0
CKO0							CKO01 Note	CKO00
	0	0	0	0	0	0	×	0/1

(e) Serial output register 0 (SO0)

Symbol:	7	6	5	4	3	2	1	0
SO0							SO01 Note	SO00
	0	0	0	0	0	0	×	0/1

Start condition is generated by manipulating the SO0n bit.

(**Note** and **Remarks** are listed on the next page.)

Figure 12-86. Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00) (2/2)

(f) Serial output enable register 0 (SOE0)

Symbol:	7	6	5	4	3	2	1	0
SOE0							SOE01 Note	SOE00
	0	0	0	0	0	0	×	0/1

SOE0n = 0 until the start condition is generated, and SOE0n = 1 after generation.

(g) Serial channel start register 0 (SS0) … Sets only the bits of the target channel is 1.

Symbol:	7	6	5	4	3	2	1	0
SS0							SS01	SS00
	0	0	0	0	0	0	×	0/1

SS0n = 0 until the start condition is generated, and SS0n = 1 after generation.

Note 16-pin products only.

Remarks 1. n = 0, r: IIC number (r = 00)

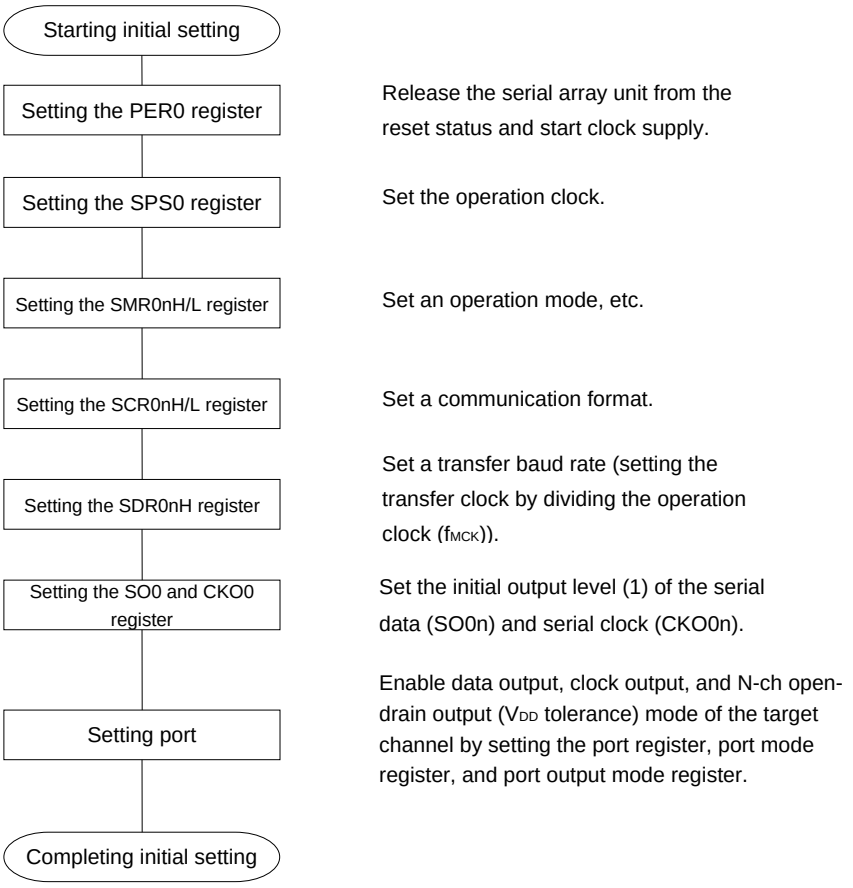
2. □: Setting is fixed in the simplified SPI (CSI) master transmission mode, : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

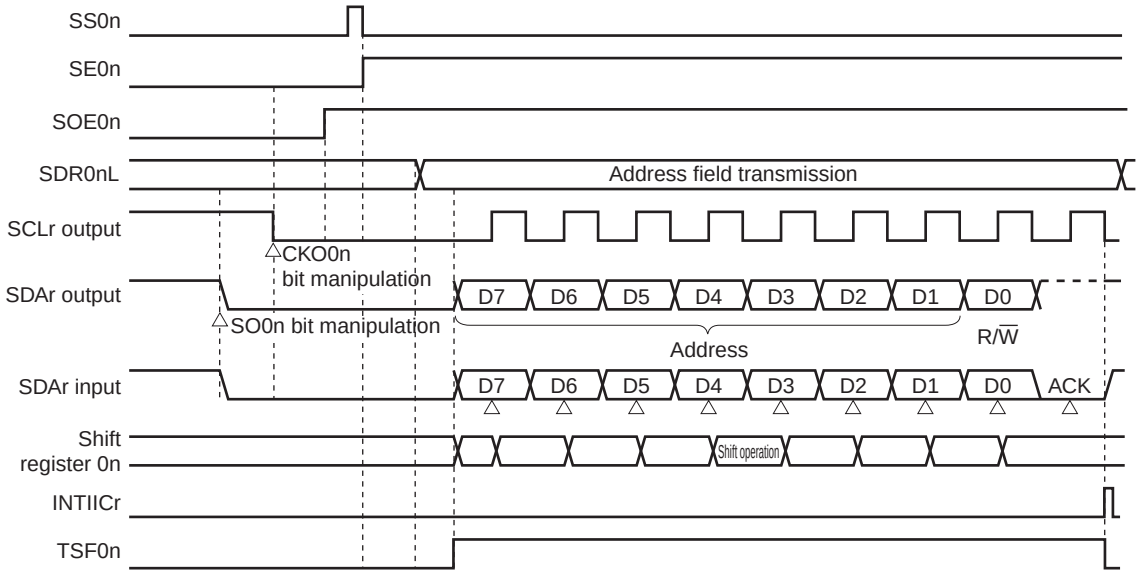
(2) Operation procedure

Figure 12-87. Initial Setting Procedure for simplified I²C Address Field Transmission



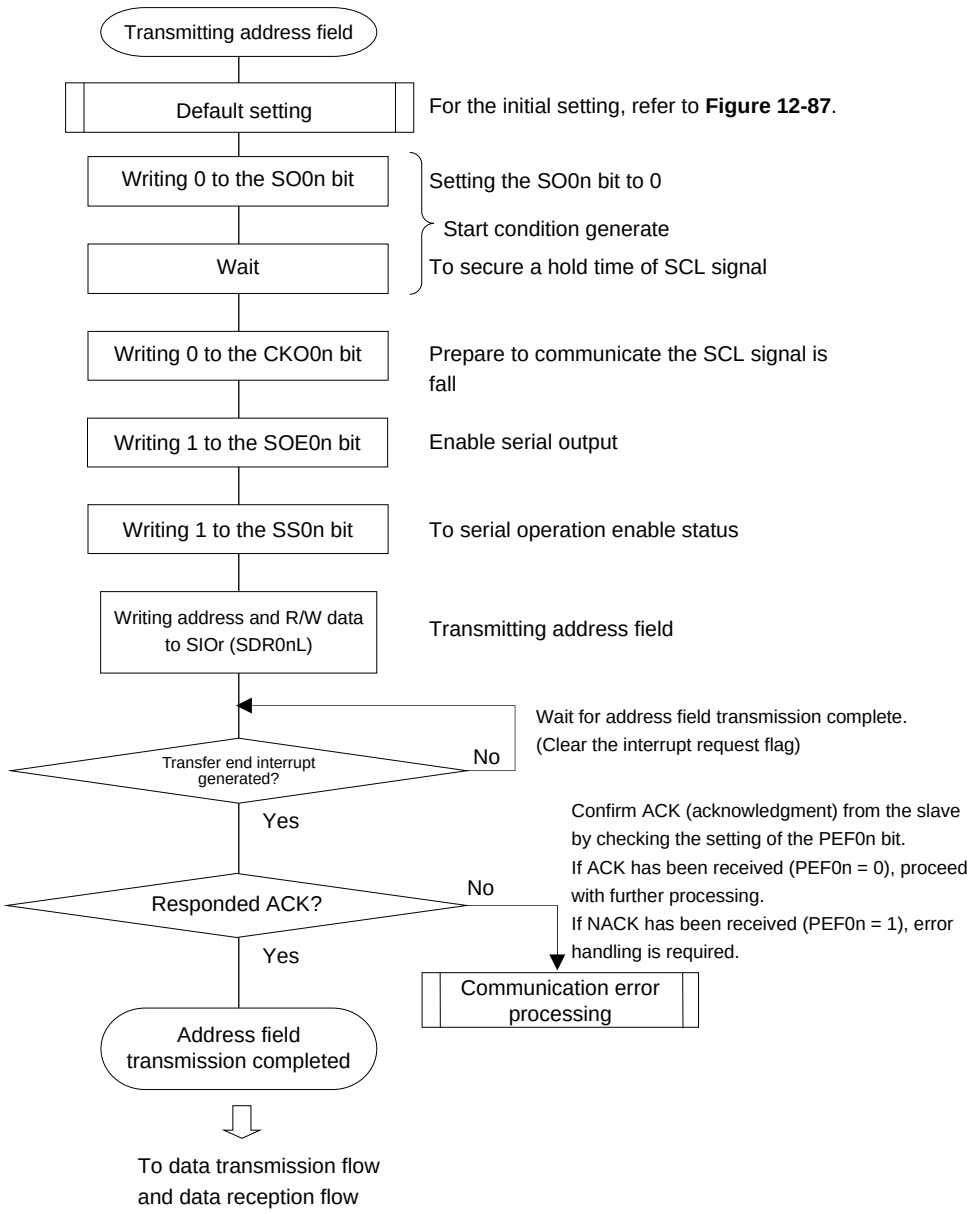
(3) Processing flow

Figure 12-88. Timing Chart of Address Field Transmission



Remark 0n = 00, r: IIC number (r = 00)

Figure 12-89. Flowchart of simplified I²C Address Field Transmission



12.7.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I²C	IIC00
Target channel	Channel 0 of SAU0
Pins used	SCL00, SDA00 Note 1
Interrupt	INTIIC00
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	ACK error detection flag (PEF0n)
Transfer data length	8 bits
Transfer rate Note 2	Max. $f_{MCK}/4$ [Hz] (SDR0nH[7:1] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I²C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)
Data level	Non-inversion output (default: high level)
Parity bit	No parity bit
Stop bit	Appending 1 bit (for ACK reception timing)
Data direction	MSB first

Notes 1. To perform communication via simplified I²C, set the N-ch open-drain output (V_{DD} tolerance) mode (POM01 = 1) for the port output mode register (POM0) (see **4.3 Registers Controlling Port Function** for details).

2. Use this operation within a range that satisfies the conditions above and the peripheral function characteristics in the electrical specifications (see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**).

Remark n = 0

(1) Register setting

Figure 12-90. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL) … Do not manipulate this register during data transmission/reception.

Symbol: SMR0nH

15	14	13	12	11	10	9	8
CKS0n	CCS0n						STS0n
0/1	0	0	0	0	0	0	0

Symbol: SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL) … Do not manipulate the bits of this register, except the TXE0n and RXE0n bits, during data transmission/reception.

Symbol: SCR0nH

15	14	13	12	11	10	9	8
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
1	0	0	0	0	0	0	0

Symbol: SCR0nL

7	6	5	4	3	2	1	
DIR0n		SLC0n1	SLC0n0				DLS0n0
0	0	0	1	0	1	1	1

(c) Serial data register 0n (SDR0nH, SDR0nL) … During data transmission/reception, valid only lower 8-bits (SIOr)

Symbol: SDR0nH

7	6	5	4	3	2	1	0
Baud rate setting							
							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Transmit data setting							

SIOr

(d) Serial clock output register 0 (CKO0) … Do not manipulate this register during data transmission/reception.

Symbol:	7	6	5	4	3	2	1	0
CKO0							CKO01 Note 1	CKO00
	0	0	0	0	0	0	×	0/1

(e) Serial output register 0 (SO0) … Do not manipulate this register during data transmission/reception.

Symbol:	7	6	5	4	3	2	1	0
SO0							SO01 Note 1	SO00
	0	0	0	0	0	0	×	0/1 Note 2

Notes 1. 16-pin products only.

2. The values may change during operation, depending on the communication data.

(**Remarks** are listed on the next page.)

Figure 12-90. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00) (2/2)

(f) Serial output enable register 0 (SOE0) … Do not manipulate this register during data transmission/reception.

Symbol:	7	6	5	4	3	2	1	0
SOE0							SOE01 Note	SOE00
	0	0	0	0	0	0	×	1

(g) Serial channel start register 0 (SS0) … Do not manipulate this register during data transmission/reception.

Symbol:	7	6	5	4	3	2	1	0
SS0							SS01	SS00
	0	0	0	0	0	0	×	0/1

Note 16-pin products only.

Remarks 1. n = 0 r: IIC number (r = 00)

2. □ : Setting is fixed in the IIC master transmission mode, ▒ : Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Processing flow

Figure 12-91. Timing Chart of Data Transmission

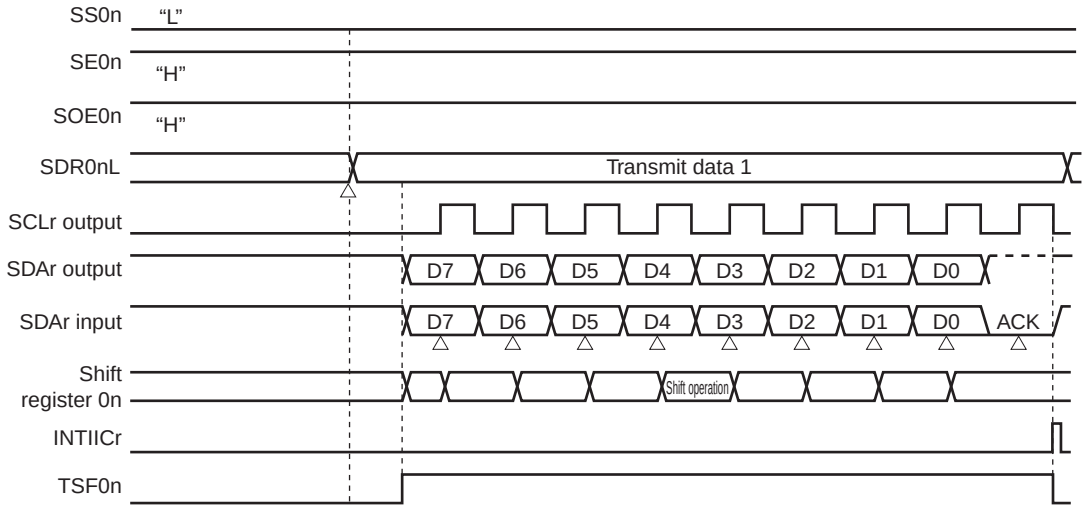
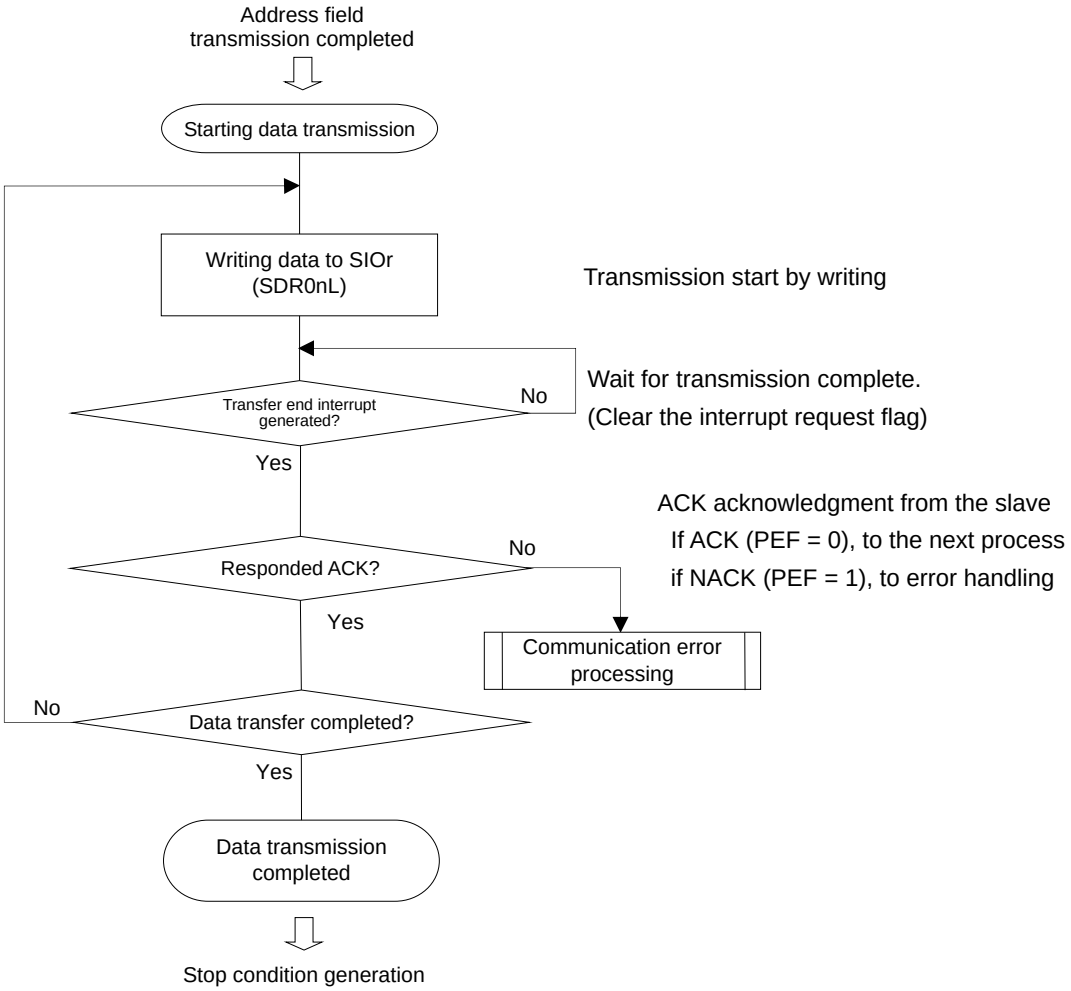


Figure 12-92. Flowchart of Simplified I²C Data Transmission



12.7.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC00
Target channel	Channel 0 of SAU0
Pins used	SCL00, SDA00 Note 1
Interrupt	INTIIC00
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Overrun error detection flag (OVF0n) only
Transfer data length	8 bits
Transfer rate Note 2	Max. $f_{MCK}/4$ [Hz] (SDR0nH[7:1] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)
Data level	Non-inversion output (default: high level)
Parity bit	No parity bit
Stop bit	Appending 1 bit (ACK transmission)
Data direction	MSB first

Notes 1. To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POM01 = 1) for the port output mode register (POM0) (see **4.3 Registers Controlling Port Function** for details).

2. Use this operation within a range that satisfies the conditions above and the peripheral function characteristics in the electrical specifications (see **CHAPTER 24 ELECTRICAL SPECIFICATIONS**).

Remark n = 0

(1) Register setting

Figure 12-93. Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01, IIC11, IIC20) (1/2)

(a) Serial mode register 0n (SMR0nH, SMR0nL) … Do not manipulate this register during data transmission/reception.

Symbol: SMR0nH

15	14	13	12	11	10	9	8
CKS0n	CCS0n						STS0n
0/1	0	0	0	0	0	0	0

Symbol: SMR0nL

7	6	5	4	3	2	1	0
	SIS0n0					MD0n1	MD0n0
0	0	1	0	0	0	0	0

(b) Serial communication operation setting register 0n (SCR0nH, SCR0nL) … Do not manipulate the bits of this register, except the TXE0n and RXE0n bits, during data transmission/reception.

Symbol: SCR0nH

15	14	13	12	11	10	9	8
TXE0n	RXE0n	DAP0n	CKP0n		EOC0n	PTC0n1	PTC0n0
0	1	0	0	0	0	0	0

Symbol: SCR0nL

7	6	5	4	3	2	1	
DIR0n		SLC0n1	SLC0n0				DLS0n0
0	0	0	1	0	1	1	1

(c) Serial data register 0n (SDR0nH, SDR0nL)

Symbol: SDR0nH

7	6	5	4	3	2	1	0
Baud rate setting							0

Symbol: SDR0nL

7	6	5	4	3	2	1	0
Dummy transmit data setting (FFH)							

SIOr

(d) Serial clock output register 0 (CKO0) Do not manipulate this register during data transmission/reception.

Symbol:	7	6	5	4	3	2	1	0
CKO0							CKO01 Note 1	CKO00
	0	0	0	0	0	0	×	0/1

(e) Serial output register 0 (SO0) … Do not manipulate this register during data transmission/reception.

Symbol:	7	6	5	4	3	2	1	0
SO0							SO01 Note 1	SO00
	0	0	0	0	0	0	×	0/1 Note 2

Notes 1. 16-pin products only.

2. The values may change during operation, depending on the communication data.

(**Remarks** are listed on the next page.)

Figure 12-93. Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01, IIC11, IIC20) (2/2)

(f) Serial output enable register 0 (SOE0) … Do not manipulate this register during data transmission/reception.

Symbol:	7	6	5	4	3	2	1	0
SOE0							SOE01 Note	SOE00
	0	0	0	0	0	0	×	0/1

(g) Serial channel start register 0 (SS0) … Do not manipulate this register during data transmission/reception.

Symbol:	7	6	5	4	3	2	1	0
SS0							SS01	SS00
	0	0	0	0	0	0	×	0/1

Note 16-pin products only.

Remarks 1. n = 0 r: IIC number (r = 00)

2. □: Setting is fixed in the IIC master transmission mode, ▒: Setting disabled (set to the initial value)

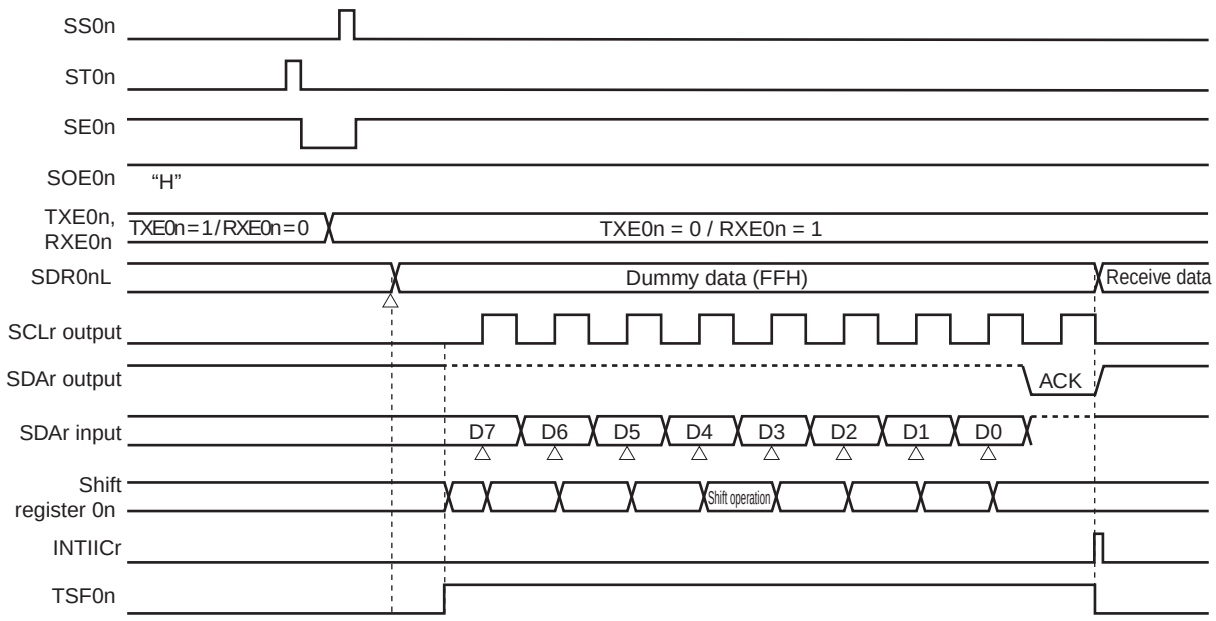
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

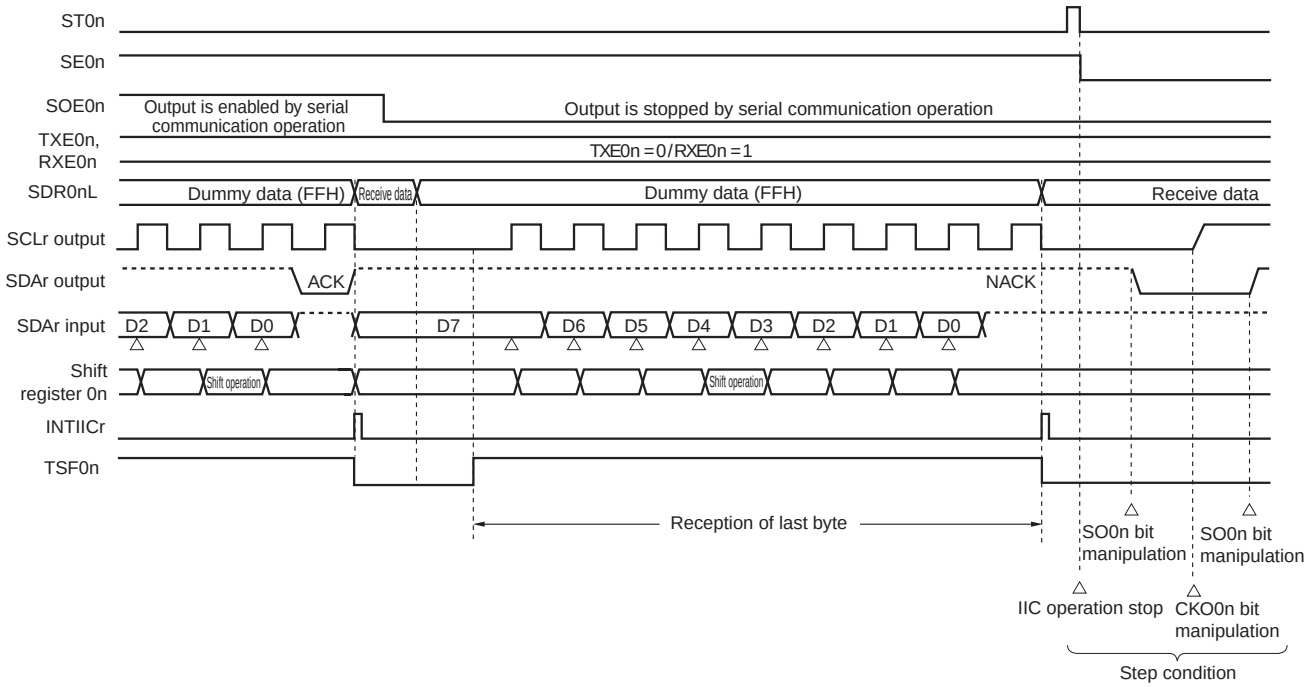
(2) Processing flow

Figure 12-94. Timing Chart of Data Reception

(a) When starting data reception

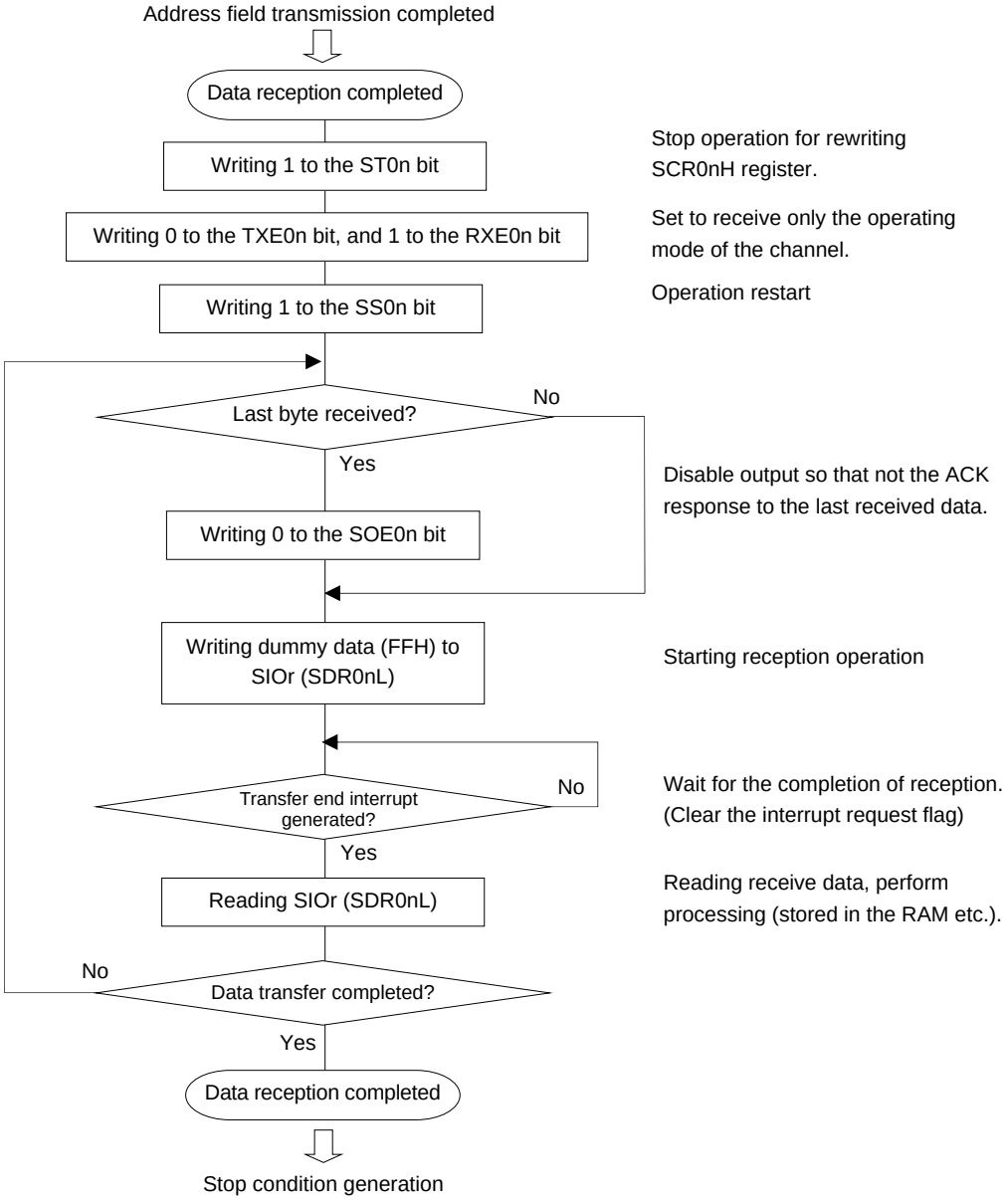


(b) When receiving last data



Remark n = 0 r: IIC number (r = 00)

Figure 12-95. Flowchart of Data Reception



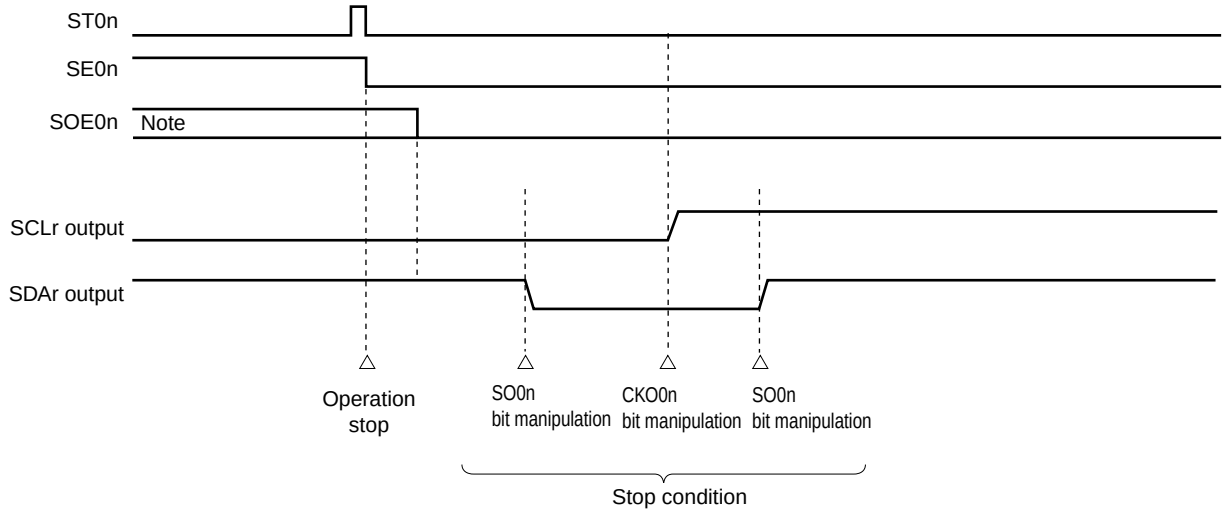
Caution **ACK is not output when the last data is received (NACK). Communication is then completed by setting the ST0n bit of serial channel stop register 0 (ST0) to 1 to stop operation and generating a stop condition.**

12.7.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

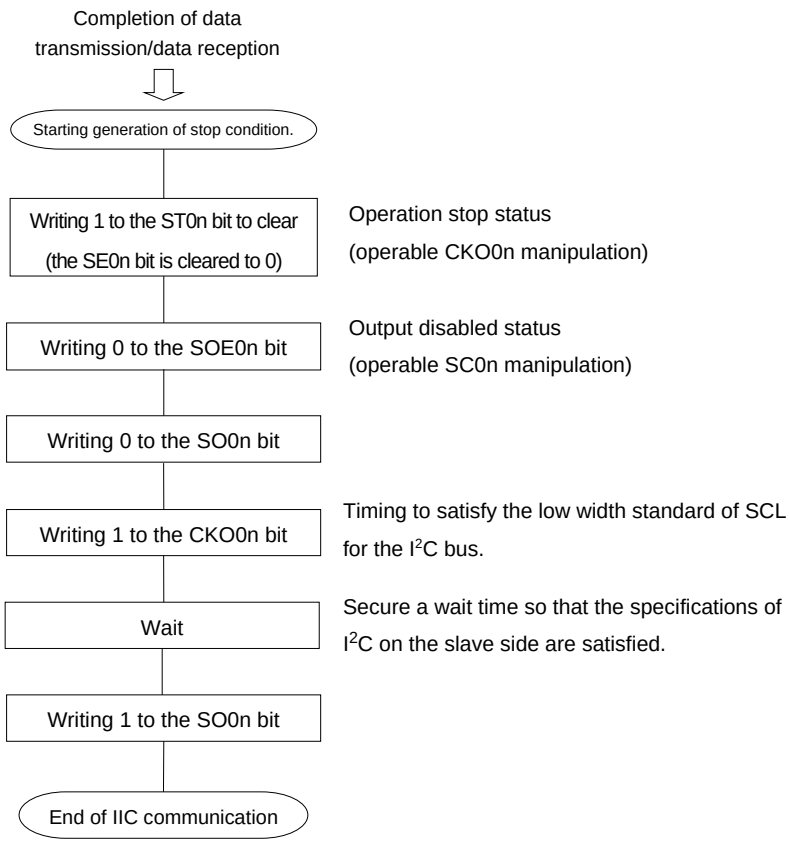
(1) Processing flow

Figure 12-96. Timing Chart of Stop Condition Generation



Note During a receive operation, the SOE0n bit of serial output enable register 0 (SOE0) is cleared to 0 before receiving the last data.

Figure 12-97. Flowchart of Stop Condition Generation



12.7.5 Calculating transfer rate

The transfer rate for simplified I^2C (IIC00) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDR0nH[7:1] + 1) ÷ 2

Caution SDR0nH[7:1] must not be set to 00000000B. Be sure to set a value of 00000001B or greater for SDR0nH[7:1]. The duty ratio of the SCL signal output by the simplified I^2C is 50%. The I^2C bus specifications define that the low-level width of the SCL signal is longer than the highlevel width. If 400 kbps (fast mode) or 1 Mbps (fast mode plus) is specified, therefore, the lowlevel width of the SCL output signal becomes shorter than the value specified in the I^2C bus specifications. Make sure that the SDR0nH[7:1] value satisfies the I^2C bus specifications.

Remarks 1. The value of SDR0nH[7:1] is the value of bits 7 to 1 of the SDR0nH register (0000001B to 1111111B) and therefore is 1 to 127.

2. n = 0

The operation clock (f_{MCK}) is determined by serial clock select register 0 (SPS0) and bit 7 (CKS0n) of serial mode register 0nH (SMR0nH).

Table 12-4. Selection of Operation Clock for Simplified I²C

SMR0n Register	SPS0 Register								Operation Clock (f_{MCK}) Note	
CKS0n	PRS 13	PRS 12	PRS 11	PRS 10	PRS 03	PRS 02	PRS 01	PRS 00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 KHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	312.5 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156.2 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	4.87 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.44 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.22 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	610 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 KHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	312.5 KHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156.2 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.76 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	4.87 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.44 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.22 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	610 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register 0 (ST0) = 000FH) the operation of the serial array unit (SAU).

Remarks 1. X: don't care

2. n = 0

Here is an example of setting an IIC transfer rate where f_{MCK} = f_{CLK} = 20 MHz.

IIC Transfer Mode (Desired Transfer Rate)	f_{CLK} = 20 MHz			
	Operation Clock (f_{MCK})	SDR0nH[7:1]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	49	100 kHz	0.0%
400 kHz	f_{CLK}	25	384.6 kHz	3.8%Note

Note The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

12.7.6 Procedure for processing errors that occurred during simplified I²C (IIC00) communication

The procedure for processing errors that occurred during simplified I²C (IIC00) communication is described in Figures 12-98 and 12-99.

Figure 12-98. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register 0n (SDR0n). →	The BFF0n bit of the SSR0n register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register 0n (SSR0n).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register 0n (SIR0n). →	The error flag is cleared.	The error only during reading can be cleared, by writing the value read from the SSR0n register to the SIR0n register without modification.

Figure 12-99. Processing Procedure in Case of ACK error in Simplified I²C Mode

Software Manipulation	Hardware Status	Remark
Reads serial status register 0n (SSR0n).		The error type is identified and the read value is used to clear the error flag.
Writes serial flag clear trigger register 0n (SIR0n). →	Error flag is cleared.	The error only during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the ST0n bit of serial channel stop register 0 (ST0) to 1. →	The SE0n bit of serial channel enable status register 0 (SE0) is set to 0 and channel n stops operation.	The slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates stop condition.		
Creates start condition.		
Sets the SS0n bit of serial channel start register 0 (SS0) to 1. →	The SE0n bit of serial channel enable status register 0 (SE0) is set to 1 and channel n is enabled to operate.	

Remark n = 0 r: IIC number (r = 00)

CHAPTER 13 SERIAL INTERFACE IICA

Caution 16-pin products have a single serial interface IICA.

13.1 Functions of Serial Interface IICA

Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLA0) line and a serial data bus (SDAA0) line.

This mode complies with the I^2C bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I^2C bus.

Since the SCLA0 and SDAA0 pins are used for open drain outputs, serial interface IICA requires pull-up resistors for the serial clock line and the serial data bus line.

(3) Wakeup mode

The STOP mode can be released by generating an interrupt request signal (INTIICA0) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUP0 bit of IICA control register 01 (IICCTL01).

Figure 13-1 shows a block diagram of serial interface IICA.

Figure 13-1. Block Diagram of Serial Interface IICA

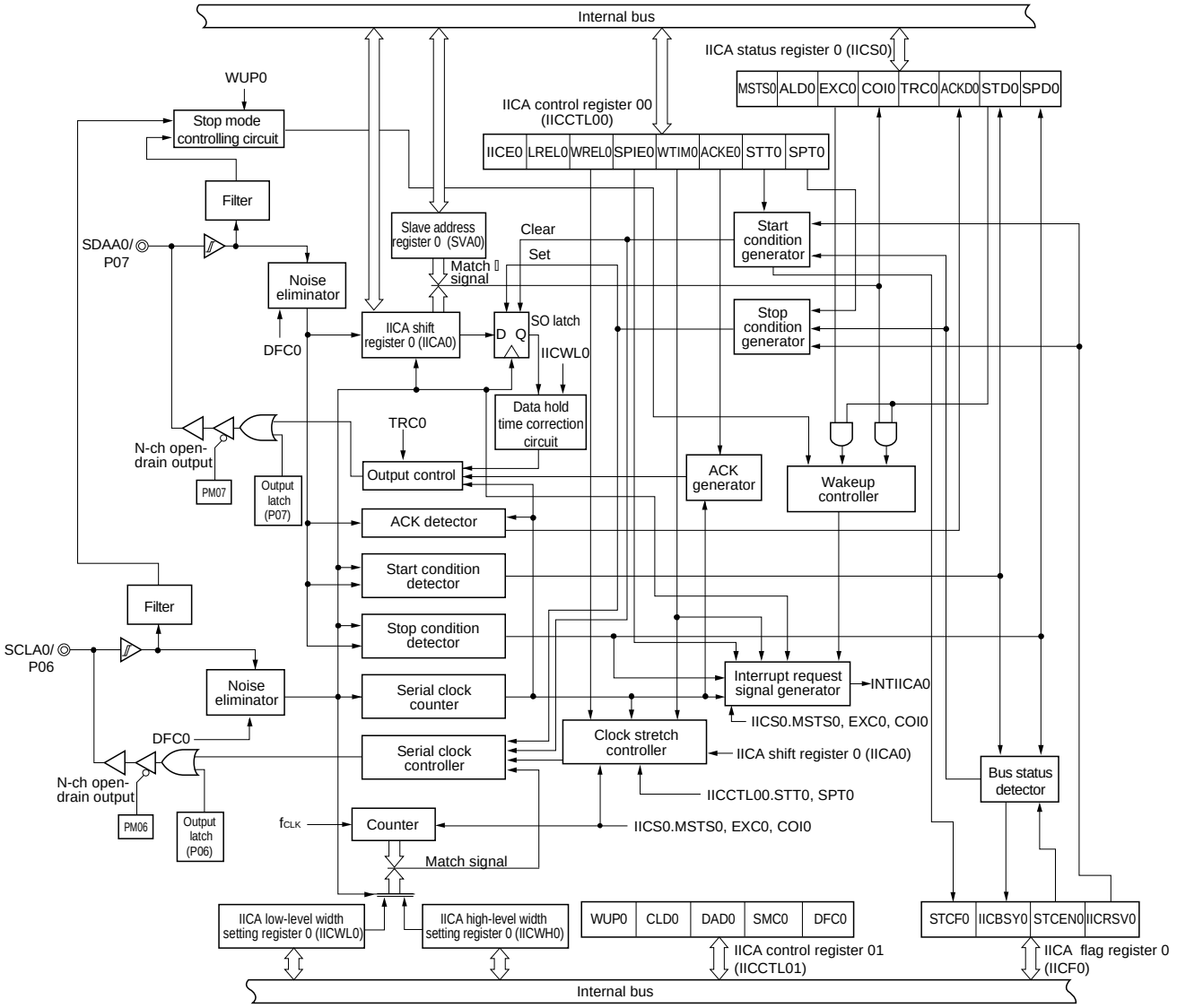
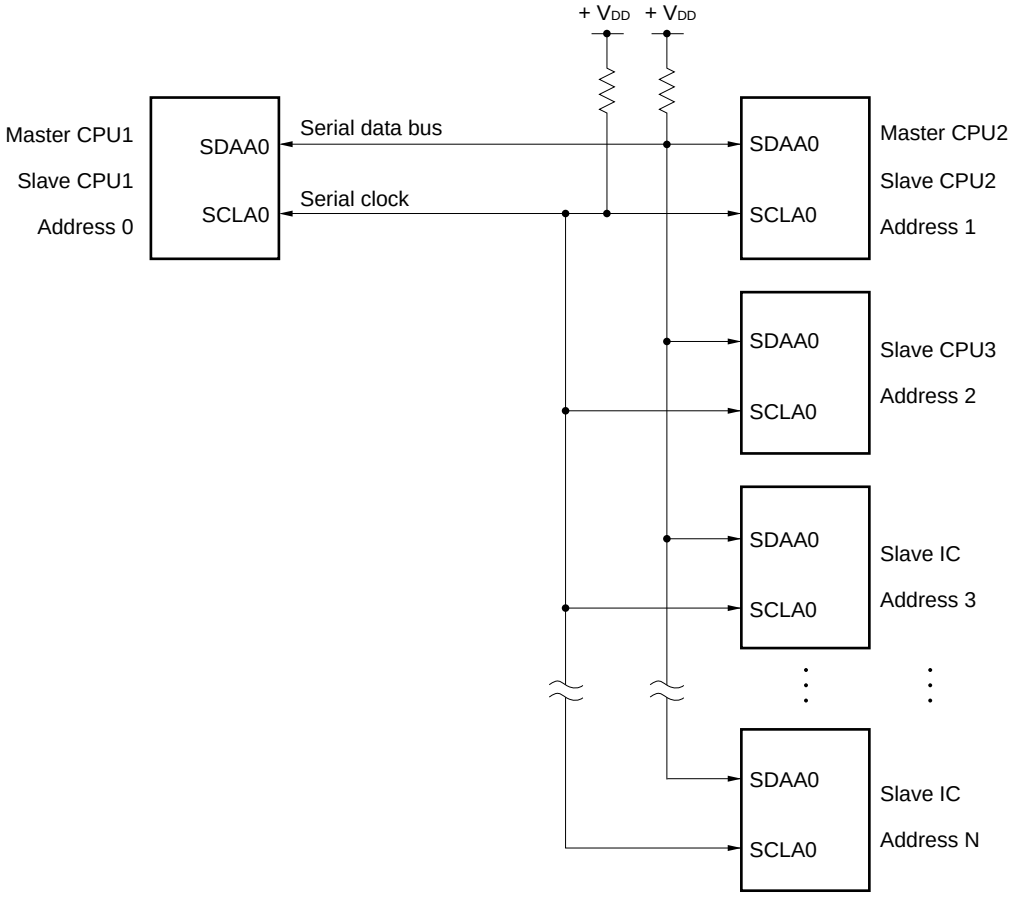


Figure 13-2 shows a serial bus configuration example.

Figure 13-2. Serial Bus Configuration Example Using I²C Bus



13.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 13-1. Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register 0 (IICA0) Slave address register 0 (SVA0)
Control registers	Peripheral enable register 0 (PER0) IICA control register 00 (IICCTL00) IICA status register 0 (IICS0) IICA flag register 0 (IICF0) IICA control register 01 (IICCTL01) IICA low-level width setting register 0 (IICWL0) IICA high-level width setting register 0 (IICWH0) Port mode register 0 (PM0) Port register 0 (P0) Port output mode register 0 (POM0) Port mode control register 0 (PMC0)

(1) IICA shift register 0 (IICA0)

The IICA0 register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICA0 register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to the IICA0 register. Cancel the clock stretch state and start data transfer by writing data to the IICA0 register during the clock stretch period.

The IICA0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears IICA0 to 00H.

Figure 13-3. Format of IICA Shift Register 0 (IICA0)

Address: FFF50H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA0								

Cautions 1. Do not write data to the IICA0 register during data transfer.

2. Write or read the IICA0 register only during the clock stretch period. Accessing the IICA0 register in a communication state other than during the clock stretch period is prohibited. When the device serves as the master, however, the IICA0 register can be written only once after the communication trigger bit (STT0) is set to 1.

3. When communication is reserved, write data to the IICA0 register after the interrupt triggered by a stop condition is detected.

(2) Slave address register 0 (SVA0)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

The SVA0 register can be set by an 8-bit memory manipulation instruction.

However, rewriting to this register is prohibited while STD0 = 1 (while the start condition is detected).

Reset signal generation clears the SVA0 register to 00H.

Figure 13-4. Format of Slave Address Register 0 (SVA0)

Address: F0234H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0	A6	A5	A4	A3	A2	A1	A0	0[Note]

Note Be sure to clear the bit 0 to 0.

(3) SO latch

The SO latch is used to retain the SDAA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICA0) when the address received by this register matches the address value set to the slave address register 0 (SVA0) or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(6) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICA0).

An I^2C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by the WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by the SPIE0 bit)

Remark WTIM0 bit: Bit 3 of IICA control register 00 (IICCTL00)
SPIE0 bit: Bit 4 of IICA control register 00 (IICCTL00)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCLA0 pin from a sampling clock.

(8) Clock stretch controller

This circuit controls the clock stretch timing.

(9) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each status.

(10) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(11) Start condition generator

This circuit generates a start condition when the STT0 bit is set to 1.

However, in the communication reservation disabled status (IICRSV0 bit = 1), when the bus is not released (IICBSY0 bit = 1), start condition requests are ignored and the STCF bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPT0 bit is set to 1.

(13) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN bit.

Remark STT0 bit: Bit 1 of IICA control register 00 (IICCTL00)
SPT0 bit: Bit 0 of IICA control register 00 (IICCTL00)
IICRSV0 bit: Bit 0 of IICA flag register 0 (IICF0)
IICBSY0 bit: Bit 6 of IICA flag register 0 (IICF0)
STCF0 bit: Bit 7 of IICA flag register 0 (IICF0)
STCEN0 bit: Bit 1 of IICA flag register 0 (IICF0)

13.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- IICA control register 00 (IICCTL00)
- IICA flag register 0 (IICF0)
- IICA status register 0 (IICS0)
- IICA control register 01 (IICCTL01)
- IICA low-level width setting register 0 (IICWL0)
- IICA high-level width setting register 0 (IICWH0)
- Port mode register 0 (PM0)
- Port register 0 (P0)
- Port output mode register 0 (POM0)
- Port mode control register 0 (PMC0)

13.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial interface IICA is used, be sure to set bit 4 (IICA0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 13-5. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PER0	TMKAEN Note	CMPEN Note	ADCEN	IICA0EN Note	0	SAU0EN	0	TAU0EN

IICA0EN	Control of serial interface IICA input clock supply
0	Stops input clock supply. • SFR used by serial interface IICA cannot be written. • Serial interface IICA is in the reset status.
1	Enables input clock supply. • SFR used by serial interface IICA can be read/written.

Note 16-pin products only

Cautions 1. When setting serial interface IICA, be sure to set the following registers first while the IICA0EN bit is set to 1. If IICA0EN = 0, the control registers of serial interface IICA are set to their initial values, and writing to them is ignored (except for port mode register 0 (PM0), port register 0 (P0), port output mode register 0 (POM0), and port mode control register 0 (PMC0)).

- **IICA control register 00 (IICCTL00)**
- **IICA flag register 0 (IICF0)**
- **IICA status register 0 (IICS0)**
- **IICA control register 01 (IICCTL01)**
- **IICA low-level width setting register 0 (IICWL0)**
- **IICA high-level width setting register 0 (IICWH0)**
- **IICA shift register 0 (IICA0)**
- **Slave address register 0 (SVA0)**

2. Be sure to clear the following bits to 0.

10-pin products: Bits 1, 3, 4, 6, and 7

16-pin products: Bits 1 and 3

13.3.2 IICA control register 00 (IICCTL00)

This register is used to enable/stop I^2C operations, set clock stretch timing, and set other I^2C operations.

The IICCTL00 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE0, WTIM0, and ACKE0 bits while IICE0 = 0 or during the clock stretch period. These bits can be set at the same time when the IICE0 bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Figure 13-6. Format of IICA Control Register 00 (IICCTL00) (1/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTL00	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable
0	Stop operation. Reset the IICA status register 0 (IICS0)[Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLA0 and SDAA0 lines are at high level.	
Condition for clearing (IICE0 = 0)	Condition for setting (IICE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL0 Notes 2, 3	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLA0 and SDAA0 lines are set to high impedance. The following flags of IICA control register 00 (IICCTL00) and the IICA status register 0 (IICS0) are cleared to 0. • STT0 • SPT0 • MSTS0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LREL0 = 0)	Condition for setting (LREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL0 Notes 2, 3	Clock stretch cancellation
0	Do not cancel clock stretch
1	Cancel clock stretch. This setting is automatically cleared after clock stretch is canceled.
When the WREL0 bit is set (clock stretch canceled) during the clock stretch period at the ninth clock pulse in the transmission status (TRC0 = 1), the SDAA0 line goes into the high impedance state (TRC0 = 0).	
Condition for clearing (WREL0 = 0)	Condition for setting (WREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICA status register 0 (IICS0), the STCF and IICBSY bits of the IICA flag register 0 (IICF0), and the CLD0 and DAD0 bits of IICA control register 01 (IICCTL01) are reset.

2. The signal of this bit is invalid while IICE0 is 0.

3. When the LREL0 and WREL0 bits are read, 0 is always read.

Caution If the operation of I^2C is enabled (IICE0 = 1) when the SCLA0 line is high level, the SDAA0 line is low level, and the digital filter is turned on (DFC0 bit of IICCTL01 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LREL0 bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICE0 = 1).

Figure 13-6. Format of IICA Control Register 00 (IICCTL00) (2/4)

SPIE0[Note 1]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If the WUP0 bit of IICA control register 01 (IICCTL01) is 1, no stop condition interrupt will be generated even if SPIE0 = 1.	
Condition for clearing (SPIE0 = 0)	Condition for setting (SPIE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM0[Note 1]	Control of clock stretch and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and clock stretch is set. Slave mode: After input of eight clocks, the clock is set to low level and clock stretch is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and clock stretch is set. Slave mode: After input of nine clocks, the clock is set to low level and clock stretch is set for master device.
An interrupt is generated at the falling edge of the ninth clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a clock stretch is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a clock stretch is inserted at the falling edge of the ninth clock after an acknowledge (ACK) signal is issued. However, when the slave device has received an extension code, a clock stretch is inserted at the falling edge of the eighth clock.	
Condition for clearing (WTIM0 = 0)	Condition for setting (WTIM0 = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE0 [Notes 1, 2]	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAA0 line is set to low level.
Condition for clearing (ACKE0 = 0)	Condition for setting (ACKE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE0 is 0. Set this bit during that period.

2. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Figure 13-6. Format of IICA Control Register 00 (IICCTL00) (3/4)

STT0 Notes 1, 2	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSY = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICRSV = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV = 1) Even if this bit is set (1), the STT0 bit is cleared and the STT0 clear flag (STCF) is set (1). No start condition is generated. In the clock stretch state (when master device): Generates a restart condition after releasing the clock stretch.

Cautions concerning set timing

- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the clock stretch period when the ACKE0 bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the clock stretch period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as stop condition trigger (SPT0).
- Setting the STT0 bit to 1 and then setting it again before it is cleared condition is prohibited.

Condition for clearing (STT0 = 0)	Condition for setting (STT0 = 1)
• Cleared by setting the STT0 bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Notes **1.** The signal of this bit is invalid while IICE0 is 0.
2. When the STT0 bit is read, 0 is always read.

Remark IICRSV0: Bit 0 of IIC flag register 0 (IICF0)
STCF0: Bit 7 of IIC flag register 0 (IICF0)

Figure 13-6. Format of IICA Control Register 00 (IICCTL00) (4/4)

SPT0 [Note]	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer).

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer.
  Can be set to 1 only in the clock stretch period when the ACKE0 bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period.
  Therefore, set to 1 during the clock stretch period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as start condition trigger (STT0).
- The SPT0 bit can be set to 1 only when in master mode.
- When the WTIM0 bit has been cleared to 0, if the SPT0 bit is set to 1 during the clock stretch period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. The WTIM0 bit should be changed from 0 to 1 during the clock stretch period following the output of eight clocks, and the SPT0 bit should be set to 1 during the clock stretch period that follows the output of the ninth clock.
- Setting the SPT0 bit to 1 and then setting it again before it is cleared condition is prohibited.

Condition for clearing (SPT0 = 0)	Condition for setting (SPT0 = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Note When the SPT0 bit is read, 0 is always read.

Caution When bit 3 (TRC0) of the IICA status register 0 (IICS0) is set to 1 (transmission status), bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 during the ninth clock and clock stretch is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the clock stretch performed while the TRC0 bit is 1 (transmission status) by writing to the IICA shift register 0.

13.3.3 IICA status register 0 (IICS0)

This register indicates the status of I²C.

The IICS0 register is read by a 1-bit or 8-bit memory manipulation instruction only when STT0 = 1 and during the clock stretch period.

Reset signal generation clears this register to 00H.

Caution Reading the IICS0 register while the address match wakeup function is enabled (WUP0 = 1) in STOP mode is prohibited. When the WUP0 bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA0 interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (SPIE0 = 1) the interrupt generated by detecting a stop condition and read the IICS0 register after the interrupt has been detected.

Remark STT0: bit 1 of IICA control register 00 (IICCTL00)
WUP0: bit 7 of IICA control register 01 (IICCTL01)

Figure 13-7. Format of IICA Status Register 0 (IICS0) (1/3)

Address: FFF51H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0 = 0)	Condition for setting (MSTS0 = 1)
• When a stop condition is detected • When ALD0 = 1 (arbitration loss) • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. The MSTS0 bit is cleared.

Condition for clearing (ALD0 = 0)	Condition for setting (ALD0 = 1)
• Automatically cleared after the IICS0 register is read[Note] • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than the IICS0 register. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

Figure 13-7. Format of IICA Status Register 0 (IICS0) (2/3)

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 = 0)	Condition for setting (EXC0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0 = 0)	Condition for setting (COI0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register 0 (SVA0)) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDAA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDAA0 line (valid starting at the falling edge of the first byte’s ninth clock).

Condition for clearing (TRC0 = 0)	Condition for setting (TRC0 = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Cleared by WREL0 = 1[Note] (clock stretch cancel) • When the ALD0 bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS0, EXC0, COI0 = 0) <Master> • When “1” is output to the first byte’s LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When “0” is input to the first byte’s LSB (transfer direction specification bit)	<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

Note When bit 3 (TRC0) of the IICA status register 0 (IICS0) is set to 1 (transmission status), bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 during the ninth clock and clock stretch is canceled, after which the TRC0 bit is cleared (reception status) and the SDAA0 line is set to high impedance. Release the clock stretch performed while the TRC0 bit is 1 (transmission status) by writing to the IICA shift register 0.

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

Figure 13-7. Format of IICA Status Register 0 (IICS0) (3/3)

ACKD0	Detection of acknowledge (ACK)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD0 = 0)	Condition for setting (ACKD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• After the SDAA0 line is set to low level at the rising edge of SCLA0 line's ninth clock

STD0	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect.

Condition for clearing (STD0 = 0)	Condition for setting (STD0 = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.

Condition for clearing (SPD0 = 0)	Condition for setting (SPD0 = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When the WUP0 bit changes from 1 to 0 • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

13.3.4 IICA flag register 0 (IICF0)

This register sets the operation mode of I^{2}C and indicates the status of the I^{2}C bus.

The IICF0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STT0 clear flag (STCF) and I^{2}C bus status flag (IICBSY) bits are read-only.

The IICRSV bit can be used to enable/disable the communication reservation function.

The STCEN bit can be used to set the initial value of the IICBSY bit.

The IICRSV and STCEN bits can be written only when the operation of I^{2}C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) = 0). When operation is enabled, the IICF0 register can be read.

Reset signal generation clears this register to 00H.

Figure 13-8. Format of IICA Flag Register 0 (IICF0)

Address: FFF52H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF0	IICBSY0	0	0	0	0	STCEN0	IICRSV0

STCF0	STT0 clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear the STT0 flag

Condition for clearing (STCF0 = 0)	Condition for setting (STCF0 = 1)
- Cleared by STT0 = 1 - When IICE0 = 0 (operation stop) - Reset	- Generating start condition unsuccessful and the STT0 bit cleared to 0 when communication reservation is disabled (IICRSV0 = 1).

IICBSY0	I^2C bus status flag
0	Bus release status (communication initial status when STCEN0 = 1)
1	Bus communication status (communication initial status when STCEN0 = 0)

Condition for clearing (IICBSY0 = 0)	Condition for setting (IICBSY0 = 1)
- Detection of stop condition - When IICE0 = 0 (operation stop) - Reset	- Detection of start condition - Setting of the IICE0 bit when STCEN0 = 0

STCEN0	Initial start enable trigger
0	After operation is enabled (IICE0 = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE0 = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN0 = 0)	Condition for setting (STCEN0 = 1)
- Cleared by instruction - Detection of start condition - Reset	- Set by instruction

IICRSV0	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV0 = 0)	Condition for setting (IICRSV0 = 1)
- Cleared by instruction - Reset	- Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to the STCEN0 bit only when the operation is stopped (IICE0 = 0).

2. As the bus release status (IICBSY0 = 0) is recognized regardless of the actual bus status when STCEN0 = 1, when generating the first start condition (STT0 = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to IICRSV0 only when the operation is stopped (IICE0 = 0).

Remark STT0: Bit 1 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

13.3.5 IICA control register 01 (IICCTL01)

This register is used to set the operation mode of I²C and detect the statuses of the SCLA0 and SDAA0 pins.

The IICCTL01 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0 and DAD0 bits are read-only.

Set the IICCTL01 register, except the WUP0 bit, while operation of I²C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation clears this register to 00H.

Figure 13-9. Format of IICA Control Register 01 (IICCTL01) (1/2)

Address: F0231H After reset: 00H R/W[Note 1]

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	0
IICCTL01	WUP0	0	CLD0	DAD0	SMC0	DFC0	0	0

WUP0	Control of address match wakeup
0	Stops operation of address match wakeup function in STOP mode.
1	Enables operation of address match wakeup function in STOP mode.

To shift to STOP mode when WUP0 = 1, execute the STOP instruction at least three cycles of the count clock (f_{TCLK}) after setting (1) the WUP0 bit (see **Figure 13-21 Flow When Setting WUP0 = 1**).
Clear (0) the WUP0 bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUP0 bit. (The clock stretch must be released and transmit data must be written after the WUP0 bit has been cleared (0).)
The interrupt timing when the address has matched or when an extension code has been received, while WUP0 = 1, is identical to the interrupt timing when WUP0 = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP0 = 1, a stop condition interrupt is not generated even if the SPIE0 bit is set to 1.
When WUP0 = 0 is set by a source other than an interrupt from serial interface IICA, operation as the master device cannot be performed until the subsequent start condition or stop condition is detected. Do not output a start condition by setting (1) the STT0 bit, without waiting for the detection of the subsequent start condition or stop condition.

Condition for clearing (WUP0 = 0)	Condition for setting (WUP0 = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when the MSTS0, EXC0, and COI0 bits are “0”, and the STD0 bit also “0” (communication not entered))[Note 2]

Notes 1. Bits 4 and 5 are read-only.

2. The status of the IICA status register 0 (IICS0) must be checked and the WUP0 bit must be set during the period shown below.

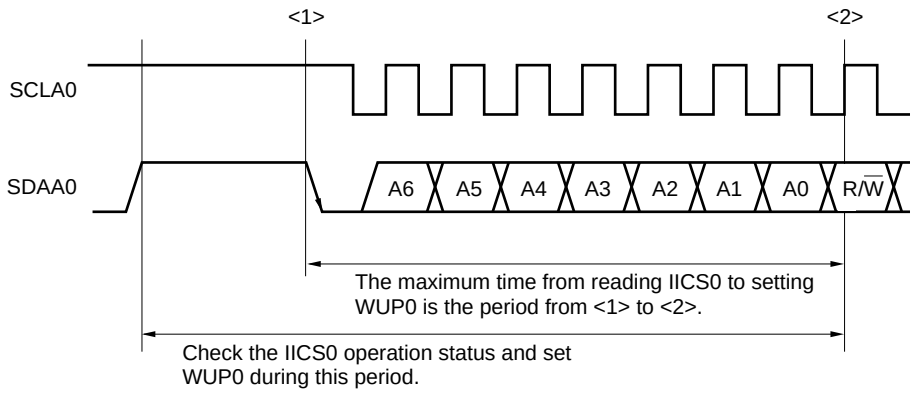


Figure 13-9. Format of IICA Control Register 01 (IICCTL01) (2/2)

CLD0	Detection of SCLA0 pin level (valid only when IICE0 = 1)
0	The SCLA0 pin was detected at low level.
1	The SCLA0 pin was detected at high level.
Condition for clearing (CLD0 = 0)	Condition for setting (CLD0 = 1)
• When the SCLA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SCLA0 pin is at high level

DAD0	Detection of SDAA0 pin level (valid only when IICE0 = 1)
0	The SDAA0 pin was detected at low level.
1	The SDAA0 pin was detected at high level.
Condition for clearing (DAD0 = 0)	Condition for setting (DAD0 = 1)
• When the SDAA0 pin is at low level • When IICE0 = 0 (operation stop) • Reset	• When the SDAA0 pin is at high level

SMC0	Operation mode switching
0	Operates in standard mode (fastest transfer rate: 100 kbps).
1	Operates in fast mode (fastest transfer rate: 400 kbps).

DFC0	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Use the digital filter only in fast mode and fast mode plus. The digital filter is used for noise elimination. The transfer clock does not vary, regardless of the DFC0 bit being set (1) or cleared (0).	

Caution Note the minimum f_{CLK} operation frequency when setting the transfer clock.
The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.
Normal mode: f_{CLK} = 1 MHz (min.)
Fast mode: f_{CLK} = 3.5 MHz (min.)

Remark IICE0: Bit 7 of IICA control register 00 (IICCTL00)

13.3.6 IICA low-level width setting register 0 (IICWL0)

This register is used to set the low-level width (t_{LOW}) of the SCLA0 pin signal that is output by serial interface IICA and to control the SDAA0 pin signal. The IICWL0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWL0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation sets this register to FFH.

For details about setting the IICWL0 register, see **13.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers**.

The data hold time is one-quarter of the time set by the IICWL0 register.

Figure 13-10. Format of IICA Low-Level Width Setting Register 0 (IICWL0)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWL0								

13.3.7 IICA high-level width setting register 0 (IICWH0)

This register is used to set the high-level width of the SCLA0 pin signal that is output by serial interface IICA and to control the SDAA0 pin signal.

The IICWH0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWH0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) is 0).

Reset signal generation sets this register to FFH.

Figure 13-11. Format of IICA High-Level Width Setting Register 0 (IICWH0)

Address: F0233H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWH0								

Remark For setting procedures of the transfer clock on master side and of the IICWL0 and IICWH0 registers on slave side, see **13.4.2 (1)** and **13.4.2 (2)**, respectively.

13.3.8 Registers controlling port functions of IICA serial input/output pins

Using port pins for the IICA requires setting of the corresponding bits in registers that control the port functions multiplexed on the IICA serial I/O pins (SCLA0 and SDAA0 pins): port mode register (PM0), port register (P0), port output mode register (POM0), and port mode control register (PMC0).

For details on the registers that control the port functions, see **4.3.1 Port mode registers 0, 4 (PM0, PM4)**, **4.3.2 Port registers 0, 4, 12, 13 (P0, P4, P12, P13)**, **4.3.4 Port output mode register 0 (POM0)**, and **4.3.5 Port mode control register 0 (PMC0)**.

When you intend to use the clock I/O pin (SCLA0) and serial data I/O pin (SDAA0) of the IICA0, set the corresponding bits in the port mode register (PM0) and port mode control register (PMC0) to 0 and the corresponding bits in the port register (P0) and port output mode register (POM0) to 1.

For details, see **4.5.3 Example of register settings for port and alternate functions used**.

Since these pins are used as n-channel open-drain outputs (with a withstand voltage of V_{DD}), use a resistor to pull them up to the power-supply voltage of the external device.

13.4 I²C Bus Mode Functions

13.4.1 Pin configuration

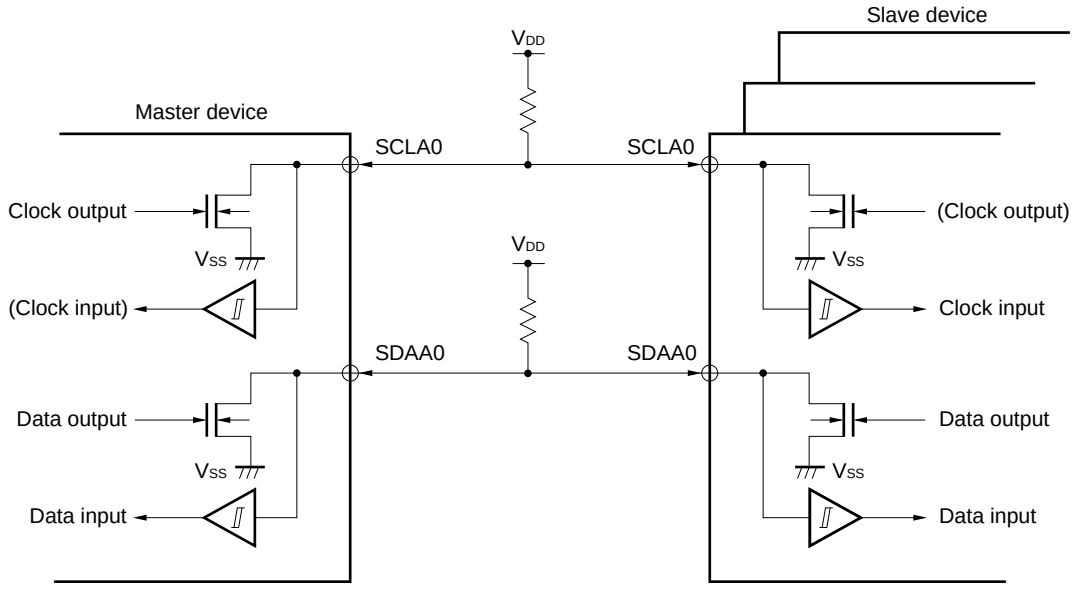
The serial clock pin (SCLA0) and the serial data bus pin (SDAA0) are configured as follows.

(1) SCLA0 .... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDAA0 .... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

Figure 13-12. Pin Configuration Diagram



13.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{CLK}}{\text{IICWL0} + \text{IICWH0} + f_{CLK}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWL0 and IICWH0 registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$\text{IICWL0} = \frac{0.52}{\text{Transfer clock}} \times f_{CLK}$$
$$\text{IICWH0} = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{CLK}$$

- When the normal mode

$$\text{IICWL0} = \frac{0.47}{\text{Transfer clock}} \times f_{CLK}$$
$$\text{IICWH0} = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{CLK}$$

(2) Setting IICWL0 and IICWH0 registers on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$\text{IICWL0} = 1.3\ \mu s \times f_{CLK}$$
$$\text{IICWH0} = (1.2\ \mu s - t_R - t_F) \times f_{CLK}$$

- When the normal mode

$$\text{IICWL0} = 4.7\ \mu s \times f_{CLK}$$
$$\text{IICWH0} = (5.3\ \mu s - t_R - t_F) \times f_{CLK}$$

Caution Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.

Fast mode: f_{CLK} = 3.5 MHz (MIN.)

Normal mode: f_{CLK} = 1 MHz (MIN.)

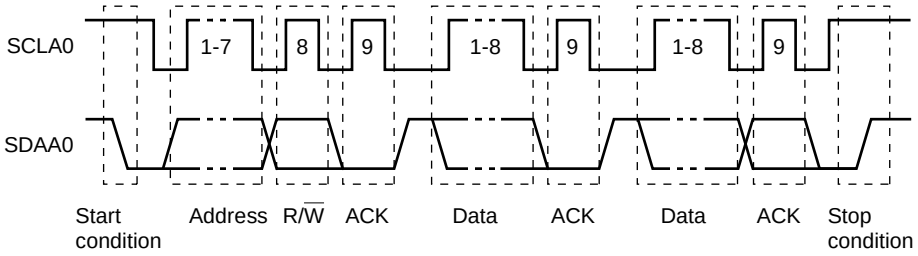
Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAA0 and SCLA0 signals separately, because they differ depending on the pull-up resistance and wire load.

2. IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
t_R: SDAA0 and SCLA0 signal rising times
f_{CLK}: CPU/peripheral hardware clock frequency

13.5 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus. Figure 13-13 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 13-13. I²C Bus Serial Data Transfer Timing



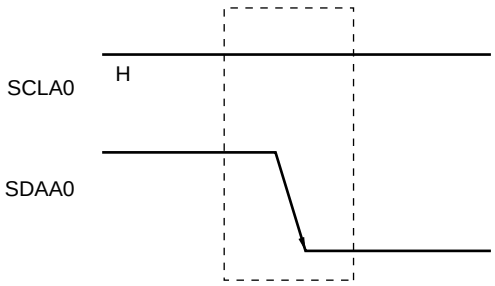
The master device generates the start condition, slave address, and stop condition.
The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLA0) is continuously output by the master device. However, in the slave device, the SCLA0 pin low level period can be extended and a clock stretch can be inserted.

13.5.1 Start conditions

A start condition is met when the SCLA0 pin is at high level and the SDAA0 pin changes from high level to low level. The start conditions for the SCLA0 pin and SDAA0 pin are signals that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 13-14. Start Conditions



A start condition is output when bit 1 (STT0) of IICA control register 00 (IICCTL00) is set (1) after a stop condition has been detected (SPD0: Bit 0 of the IICA status register 0 (IICS0) = 1). When a start condition is detected, bit 1 (STD0) of the IICS0 register is set (1).

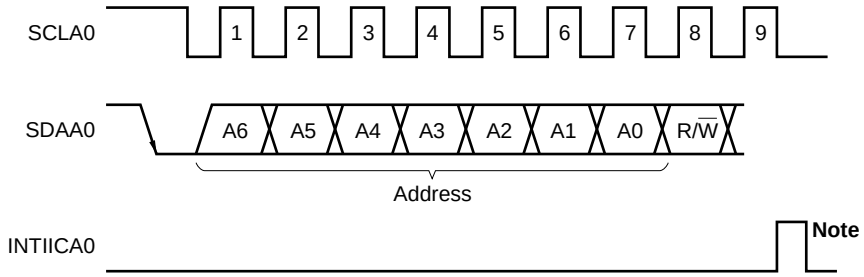
13.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register 0 (SVA0). If the address data matches the SVA0 register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 13-15. Address



Note INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **13.5.3 Transfer direction specification** are written to the IICA shift register 0 (IICA0). The received addresses are written to the IICA0 register.

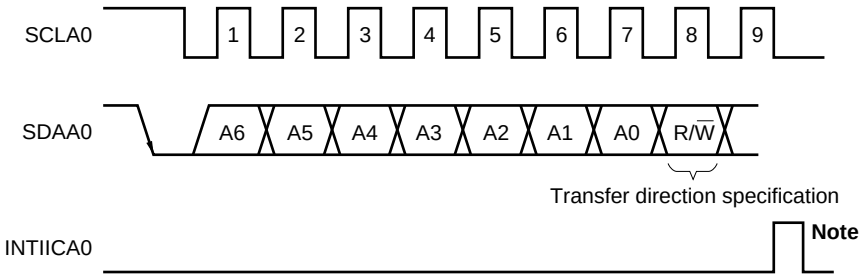
The slave address is assigned to the higher 7 bits of the IICA0 register.

13.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of “1”, it indicates that the master device is receiving data from a slave device.

Figure 13-16. Transfer Direction Specification



Note INTIICA0 is not issued if data other than a local address or extension code is received during slave device operation.

13.5.4 Acknowledge (ACK)

ACK is used to check the status of serial data at the transmission and reception sides.

The reception side returns ACK each time it has received 8-bit data.

The transmission side usually receives ACK after transmitting 8-bit data. When ACK is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether ACK has been detected can be checked by using bit 2 (ACKD0) of the IICA status register 0 (IICS0).

When the master receives the last data item, it does not return ACK and instead generates a stop condition. If a slave does not return ACK after receiving data, the master outputs a stop condition or restart condition and stops transmission. If ACK is not returned, the possible causes are as follows.

<1> Reception was not performed normally.
<2> The final data item was received.
<3> The reception side specified by the address does not exist.

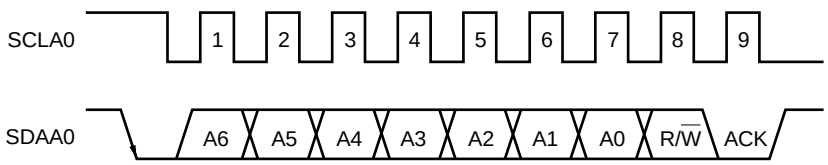
To generate ACK, the reception side makes the SDAA0 line low at the ninth clock (indicating normal reception).

Automatic generation of ACK is enabled by setting bit 2 (ACKE0) of IICA control register 00 (IICCTL00) to 1. Bit 3 (TRC0) of the IICS0 register is set by the data of the eighth bit that follows 7-bit address information. Usually, set the ACKE0 bit to 1 for reception (TRC0 = 0).

If a slave can receive no more data during reception (TRC0 = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKE0 bit to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC0 = 0), it must clear the ACKE0 bit to 0 so that ACK is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 13-17. ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKE0 bit. When an address other than that of the local address is received, ACK is not generated (NACK).

When an extension code is received, ACK is generated if the ACKE0 bit is set to 1 in advance.

How ACK is generated when data is received differs as follows depending on the setting of the clock stretch timing.

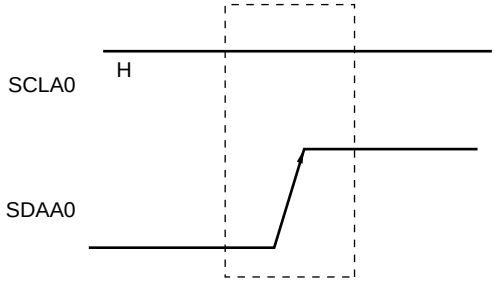
- When the falling edge of the eighth clock is set (bit 3 (WTIM0) of IICCTL00 register = 0):
  By setting the ACKE0 bit to 1 before releasing the clock stretch state, ACK is generated at the falling edge of the eighth clock of the SCLA0 pin.
- When the falling edge of the ninth clock is set (bit 3 (WTIM0) of IICCTL00 register = 1):
  ACK is generated by setting the ACKE0 bit to 1 in advance.

13.5.5 Stop condition

When the SCLA0 pin is at high level, changing the SDAA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 13-18. Stop Condition



A stop condition is generated when bit 0 (SPT0) of IICA control register 00 (IICCTL00) is set to 1. When the stop condition is detected, bit 0 (SPD0) of the IICA status register 0 (IICS0) is set to 1 and INTIICA0 is generated when bit 4 (SPIE0) of the IICCTL00 register is set to 1.

13.5.6 Clock stretch

The clock stretch is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a clock stretch state).

Setting the SCLA0 pin to low level notifies the communication partner of the clock stretch state. When clock stretch state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 13-19. Clock stretch (1/2)

(1) When the master is at the falling edge of the ninth clock and the slave is at the falling edge of the eighth clock and clock stretching occurs
(master transmits, slave receives, and ACKE0 = 1)

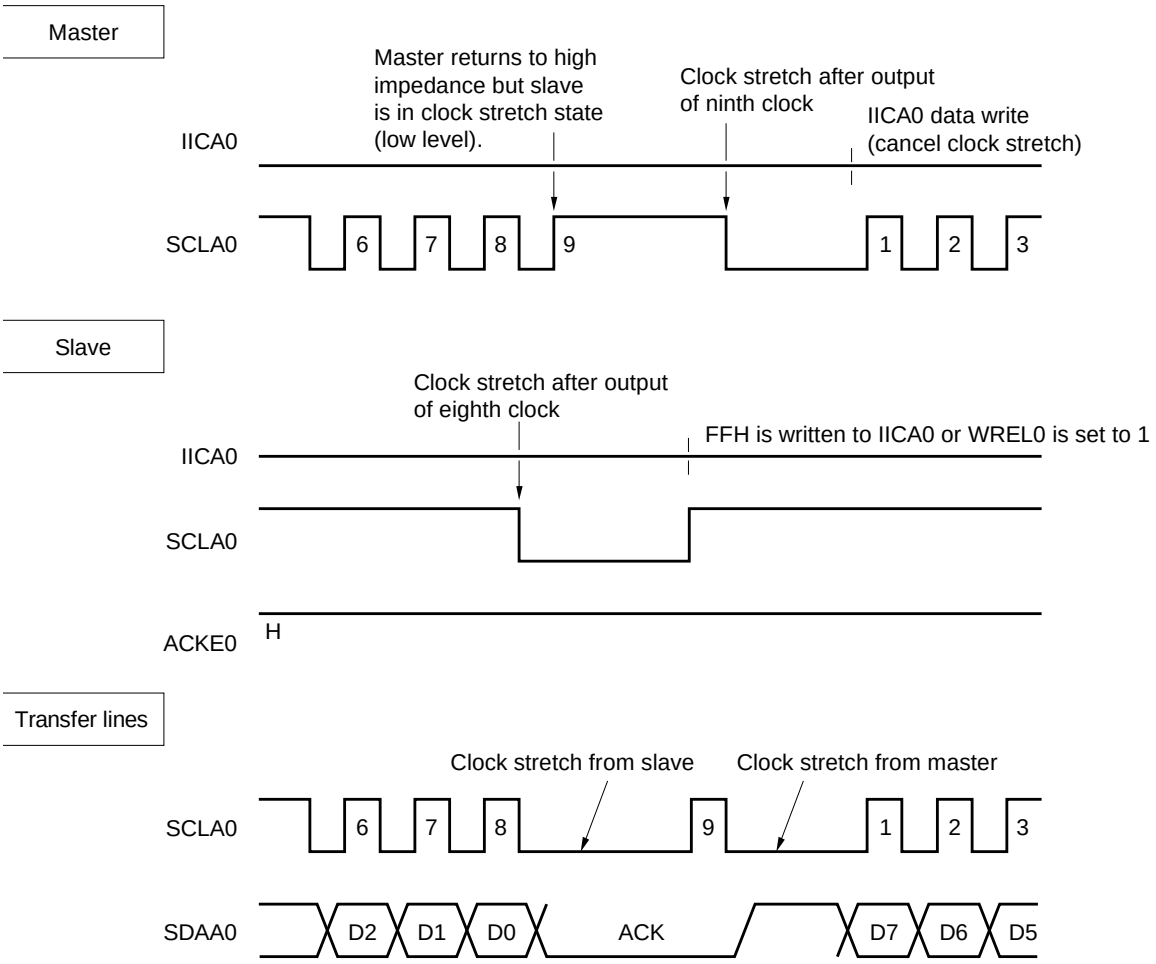
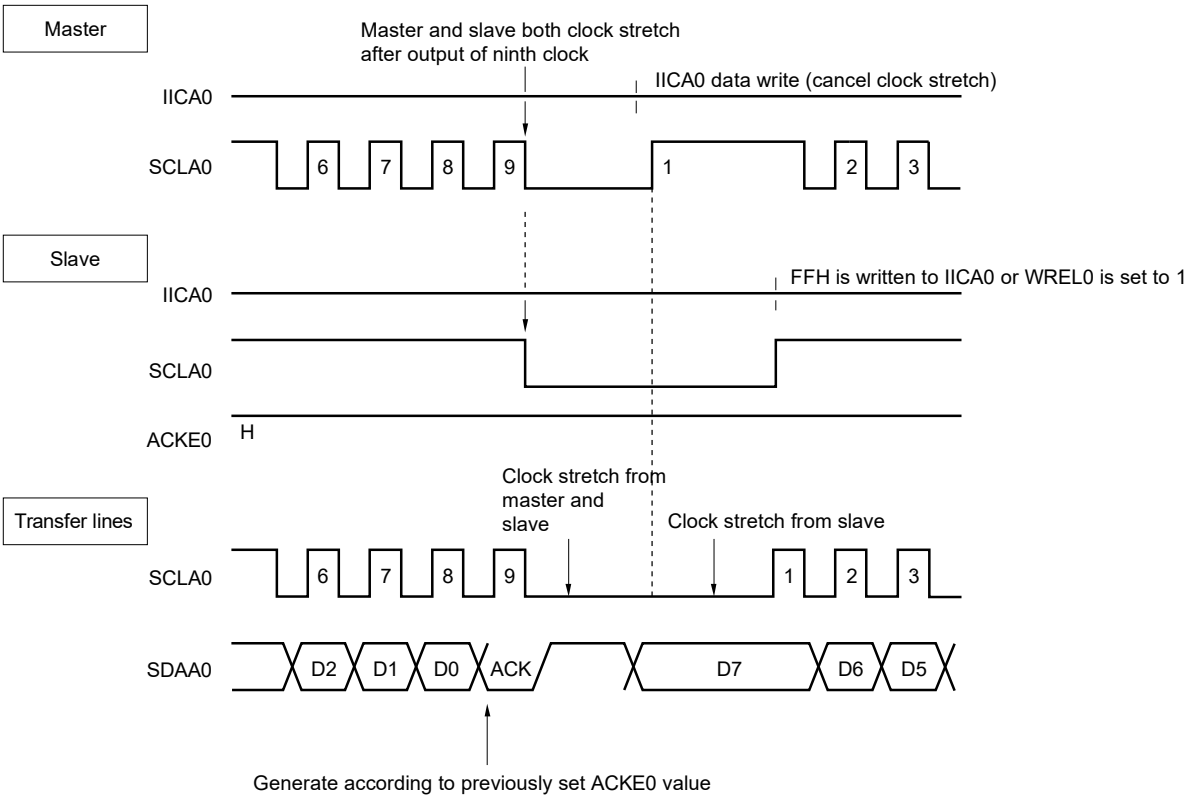


Figure 13-19. Clock stretch (2/2)

(2) When master and slave devices are at the falling edge of the ninth clock and clock stretching occurs (master transmits, slave receives, and ACKE0 = 1)



Remark ACKE0: Bit 2 of IICA control register 00 (IICCTL00)
WREL0: Bit 5 of IICA control register 00 (IICCTL00)

A clock stretch may be automatically generated depending on the setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00).

Normally, the receiving side cancels the clock stretch state when bit 5 (WREL0) of the IICCTL00 register is set to 1 or when FFH is written to the IICA shift register 0 (IICA0), and the transmitting side cancels the clock stretch state when data is written to the IICA0 register.

The master device can also cancel the clock stretch state via either of the following methods.

- By setting bit 1 (STT0) of the IICCTL00 register to 1
- By setting bit 0 (SPT0) of the IICCTL00 register to 1

13.5.7 Canceling clock stretch

The I²C usually cancels a clock stretch state by the following processing.

- Writing data to the IICA shift register 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (canceling clock stretch)
- Setting bit 1 (STT0) of the IICCTL00 register (generating start condition)[Note]
- Setting bit 0 (SPT0) of the IICCTL00 register (generating stop condition)[Note]

Note Master in the I²C communication only

When the above clock stretch canceling processing is executed, the I²C cancels the clock stretch state and communication is resumed.

To cancel a clock stretch state and transmit data (including addresses), write the data to the IICA0 register.

To receive data after canceling a clock stretch state, or to complete data transmission, set bit 5 (WREL0) of the IICCTL00 register to 1.

To generate a restart condition after canceling a clock stretch state, set bit 1 (STT0) of the IICCTL00 register to 1.

To generate a stop condition after canceling a clock stretch state, set bit 0 (SPT0) of the IICCTL00 register to 1.

Execute the canceling processing only once for one clock stretch state.

If, for example, data is written to the IICA0 register after canceling a clock stretch state by setting the WREL0 bit to 1, an incorrect value may be output to SDAA0 line because the timing for changing the SDAA0 line conflicts with the timing for writing the IICA0 register.

In addition to the above, communication is stopped if the IICE0 bit is cleared to 0 when communication has been aborted, so that the clock stretch state can be canceled.

If the I²C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL0) of the IICCTL00 register, so that the clock stretch state can be canceled.

Caution If a processing to cancel a clock stretch state is executed when WUP0 = 1, the clock stretch state will not be canceled.

13.5.8 Interrupt request (INTIICA0) generation timing and clock stretch control

The setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00) determines the timing by which INTIICA0 is generated and the corresponding clock stretch control, as shown in Table 13-2.

Table 13-2. INTIICA0 Generation Timing and Clock Stretch Control

WTIM0	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9 Notes 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Notes 1, 2	9 Note 2	9 Note 2	9	9	9

Notes 1. The slave device's INTIICA0 signal and clock stretch period occurs at the falling edge of the ninth clock only when there is a match with the address set to the slave address register 0 (SVA0).
At this point, ACK is generated regardless of the value set to the IICCTL00 register's bit 2 (ACKE0). For a slave device that has received an extension code, INTIICA0 occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICA0 is generated at the falling edge of the 9th clock, but clock stretch does not occur.

2. If the received address does not match the contents of the slave address register 0 (SVA0) and extension code is not received, neither INTIICA0 nor a clock stretch occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and clock stretch control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and clock stretch timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIM0 bit.
- Master device operation: Interrupt and clock stretch timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: Interrupt and clock stretch timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: Interrupt and clock stretch timing are determined according to the WTIM0 bit.

(4) Clock stretch cancellation method

The four clock stretch cancellation methods are as follows.

- Writing data to the IICA shift register 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (canceling clock stretch)
- Setting bit 1 (STT0) of IICCTL00 register (generating start condition)Note
- Setting bit 0 (SPT0) of IICCTL00 register (generating stop condition)Note

Note Master only.

When clock stretching timing is set to the falling edge of the eighth clock (WTIM0 = 0), the presence/absence of ACK generation must be determined prior to clock stretch cancellation.

(5) Stop condition detection

INTIICA0 is generated when a stop condition is detected (only when SPIE0 = 1).

13.5.9 Address match detection method

In I²C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request (INTIICA0) occurs when the address set to the slave address register 0 (SVA0) matches the slave address sent by the master device, or when an extension code has been received.

13.5.10 Error detection

In I²C bus mode, the status of the serial data bus (SDAA0) during data transmission is captured by the IICA shift register 0 (IICA0) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

13.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC0) is set to 1 for extension code reception and an interrupt request (INTIICA0) is issued at the falling edge of the eighth clock. The local address stored in the slave address register 0 (SVA0) is not affected.

(2) The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer when the SVA0 register is set to 11110xx0. Note that INTIICA0 occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC0 = 1
- Seven bits of data match: COI0 = 1

Remark EXC0: Bit 5 of IICA status register 0 (IICS0)
COI0: Bit 4 of IICA status register 0 (IICS0)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
If the extension code is received while a slave device is operating, then the slave device is participating in communication even if its address does not match.
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set bit 6 (LREL0) of IICA control register 00 (IICCTL00) to 1 to set the standby mode for the next communication operation.

Table 13-3. Bit Definitions of Major Extension Codes

Slave Address	R/W Bit	Description
0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (during address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (after address match, when read command is issued)

Remark See the I²C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

13.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the STT0 bit is set to 1 before the STD0 bit is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (ALD0) in the IICA status register 0 (IICS0) is set (1) via the timing by which the arbitration loss occurred, and the SCLA0 and SDAA0 lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 = 1 setting that has been made by software.

For details of interrupt request timing, see **13.5.8 Interrupt request (INTIICA0) generation timing and clock stretch control**.

Remark STD0: Bit 1 of IICA status register 0 (IICS0)
STT0: Bit 1 of IICA control register 00 (IICCTL00)

Figure 13-20. Arbitration Timing Example

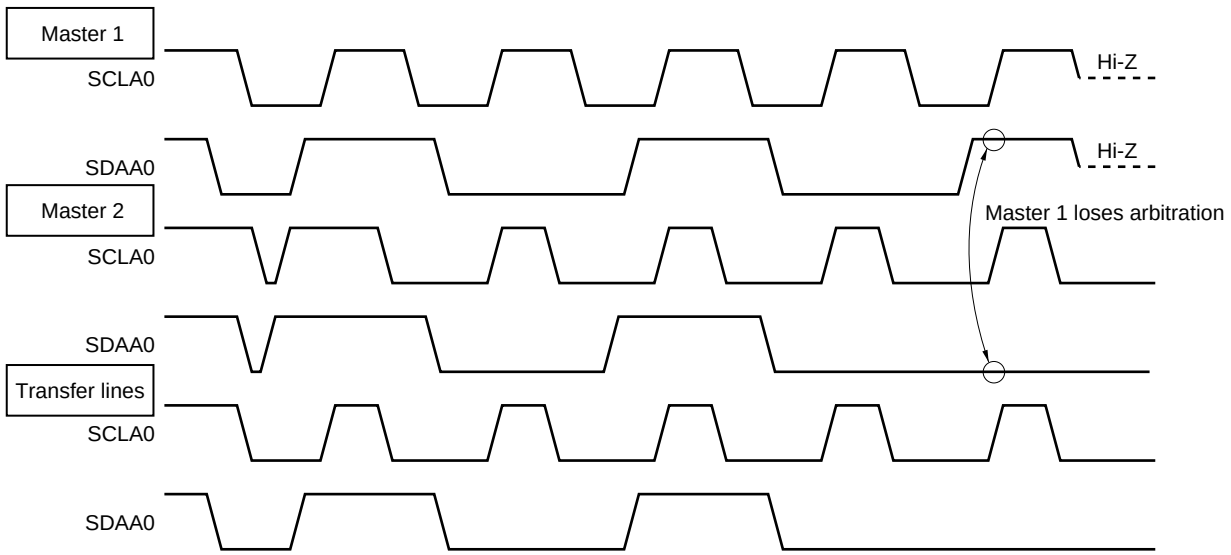


Table 13-4. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During ACK transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when SPIE0 = 1)[Note 2]
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIE0 = 1)[Note 2]
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCLA0 is at low level while attempting to generate a restart condition	

Notes 1. When the WTIM0 bit (bit 3 of IICA control register 00 (IICCTL00)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM0 = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a chance that arbitration will occur, set SPIE0 = 1 for master device operation.

Remark SPIE0: Bit 4 of IICA control register 00 (IICCTL00)

13.5.13 Wakeup function

The I²C bus slave function is a function that generates an interrupt request signal (INTIICA0) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIICA0 signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

To use the wakeup function in the STOP mode, set the WUP0 bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA0) is also generated when a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear (0) the WUP0 bit after this interrupt has been generated.

Figure 13-21 shows the flow for setting WUP0 = 1 and Figure 13-22 shows the flow for setting WUP0 = 0 upon an address match.

Figure 13-21. Flow When Setting WUP0 = 1

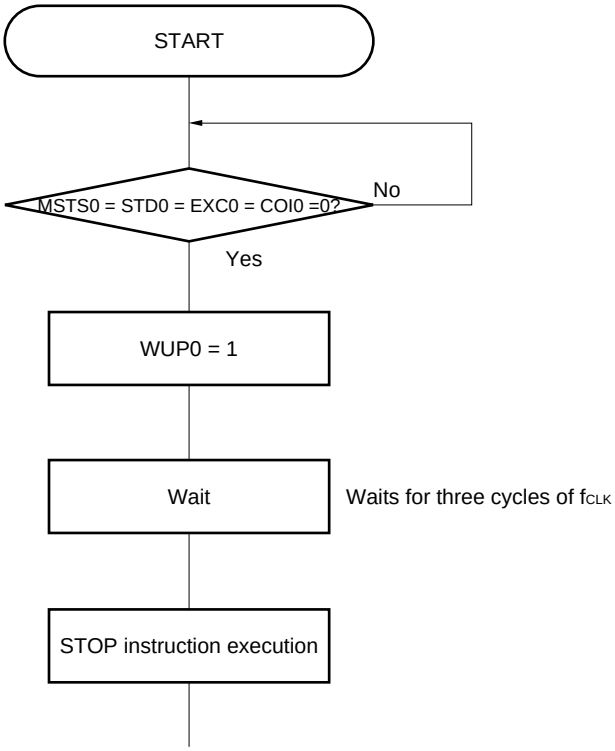
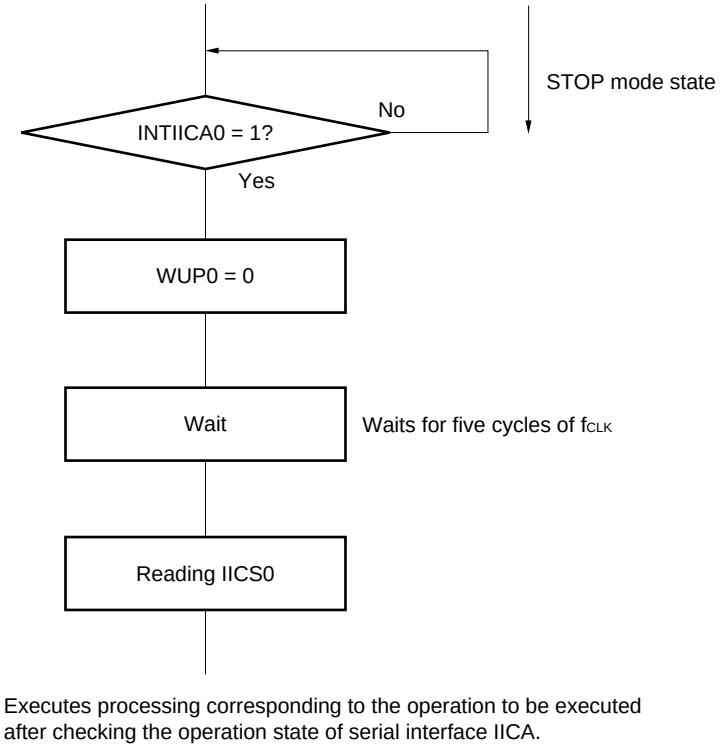


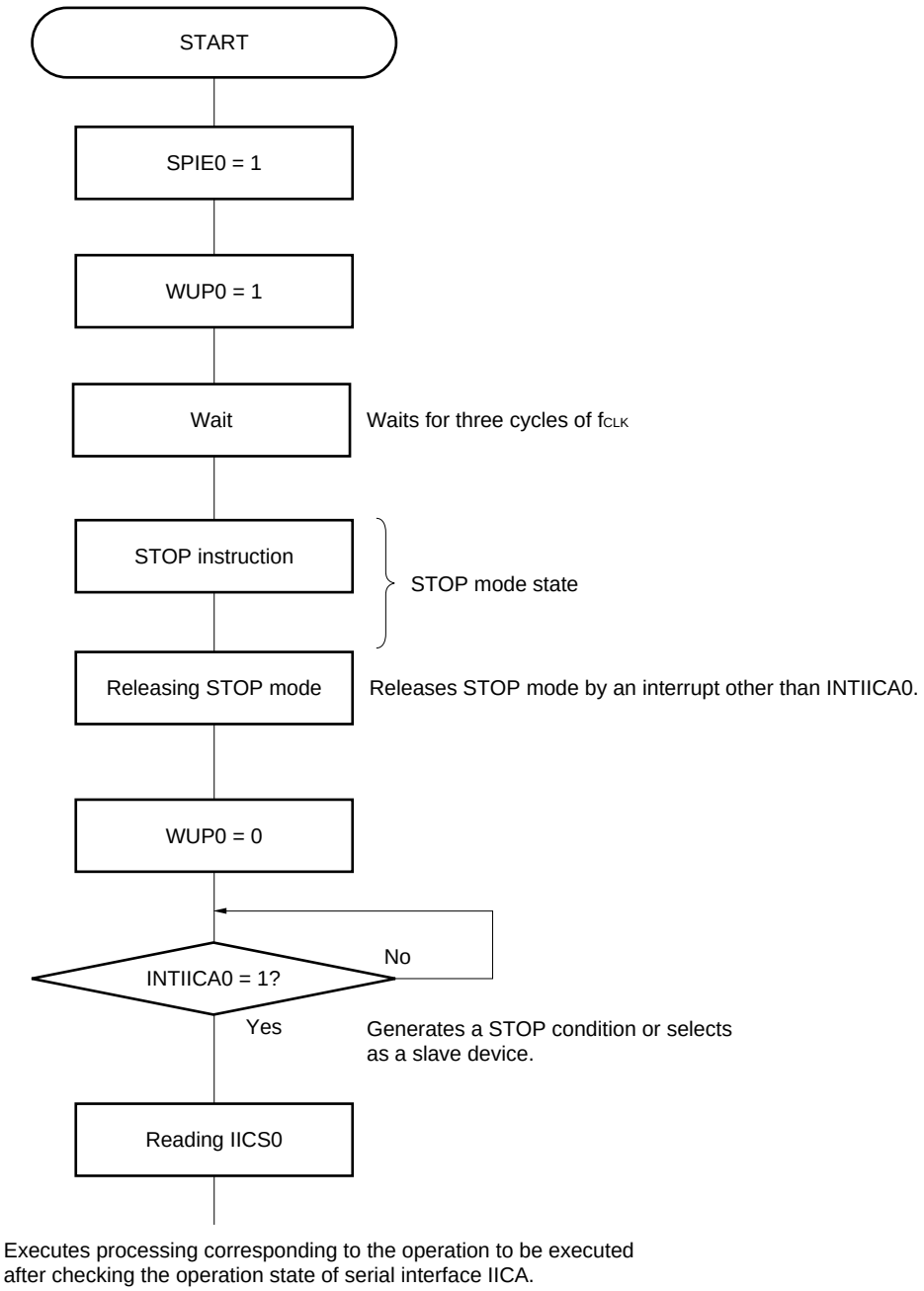
Figure 13-22. Flow When Setting WUP0 = 0 upon Address Match (Including Extension Code Reception)



Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICA0) generated from serial interface IICA.

- Master device operation: Flow shown in Figure 13-23
- Slave device operation:
  Cases where the trigger for recovery from the STOP mode is the INTIICA0 interrupt:
  Same as the flow in Figure 13-22
  Cases where the trigger for recovery from the STOP mode is other than the INTIICA0 interrupt:
  Continues the operation with WUP0 = 1 until the INTIICA0 interrupt is generated.

Figure 13-23. When Operating as Master Device after Releasing STOP Mode other than by INTIICA0



13.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IICA flag register 0 (IICF0) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LREL0) of IICA control register 00 (IICCTL00) to 1 and saving communication).

If bit 1 (STT0) of the IICCTL00 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.
If an address is written to the IICA shift register 0 (IICA0) after bit 4 (SPIE0) of the IICCTL00 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICA0) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICA0 register before the stop condition is detected is invalid.
When the STT0 bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released ..... a start condition is generated
- If the bus has not been released (standby mode) ..... communication reservation

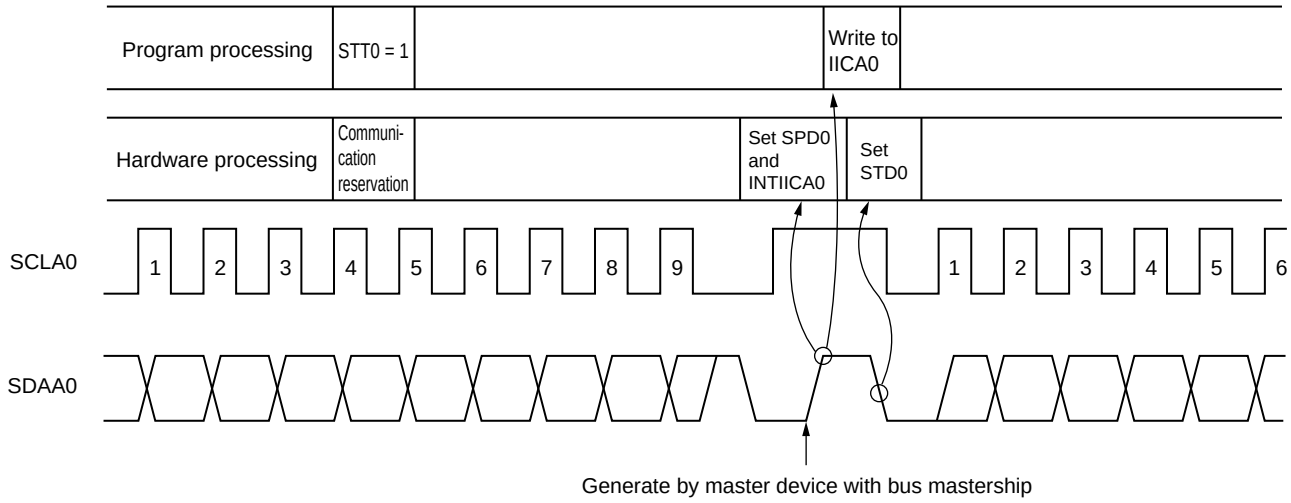
Check whether the communication reservation operates or not by using the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) after the STT0 bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT0 = 1 to checking the MSTS0 flag (the number of cycles of f_{CLK}): (IICWL0 setting value + IICWH0 setting value + 4) + $t_F \times 2 \times f_{CLK}$ [clocks]

Remark IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 13-24 shows the communication reservation timing.

Figure 13-24. Communication Reservation Timing



Remark IICA0: IICA shift register 0
STT0: Bit 1 of IICA control register 00 (IICCTL00)
STD0: Bit 1 of IICA status register 0 (IICS0)
SPD0: Bit 0 of IICA status register 0 (IICS0)

Communication reservations are accepted via the timing shown in Figure 13-25. After bit 1 (STD0) of the IICA status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IICA control register 00 (IICCTL00) to 1 before a stop condition is detected.

Figure 13-25. Timing for Accepting Communication Reservations

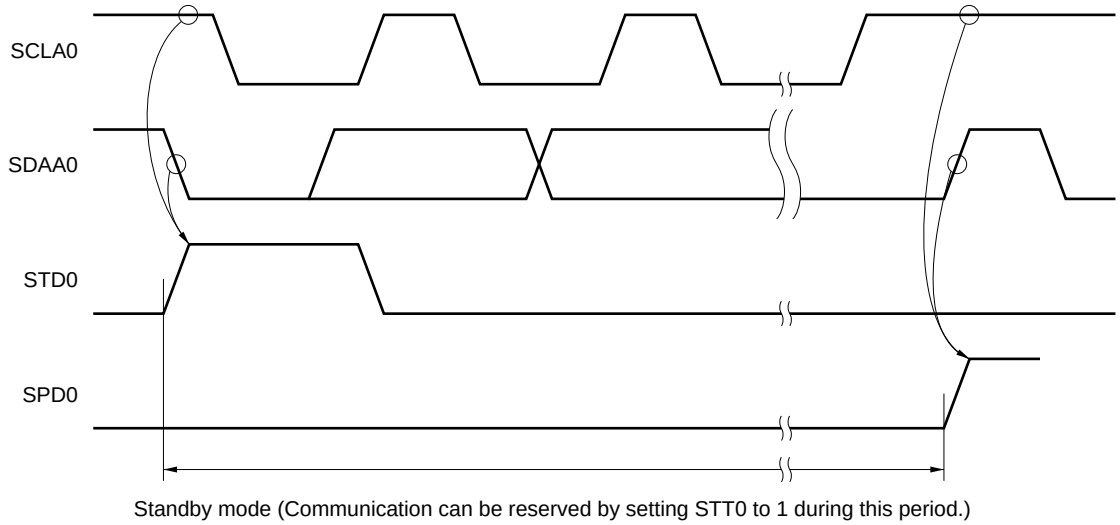
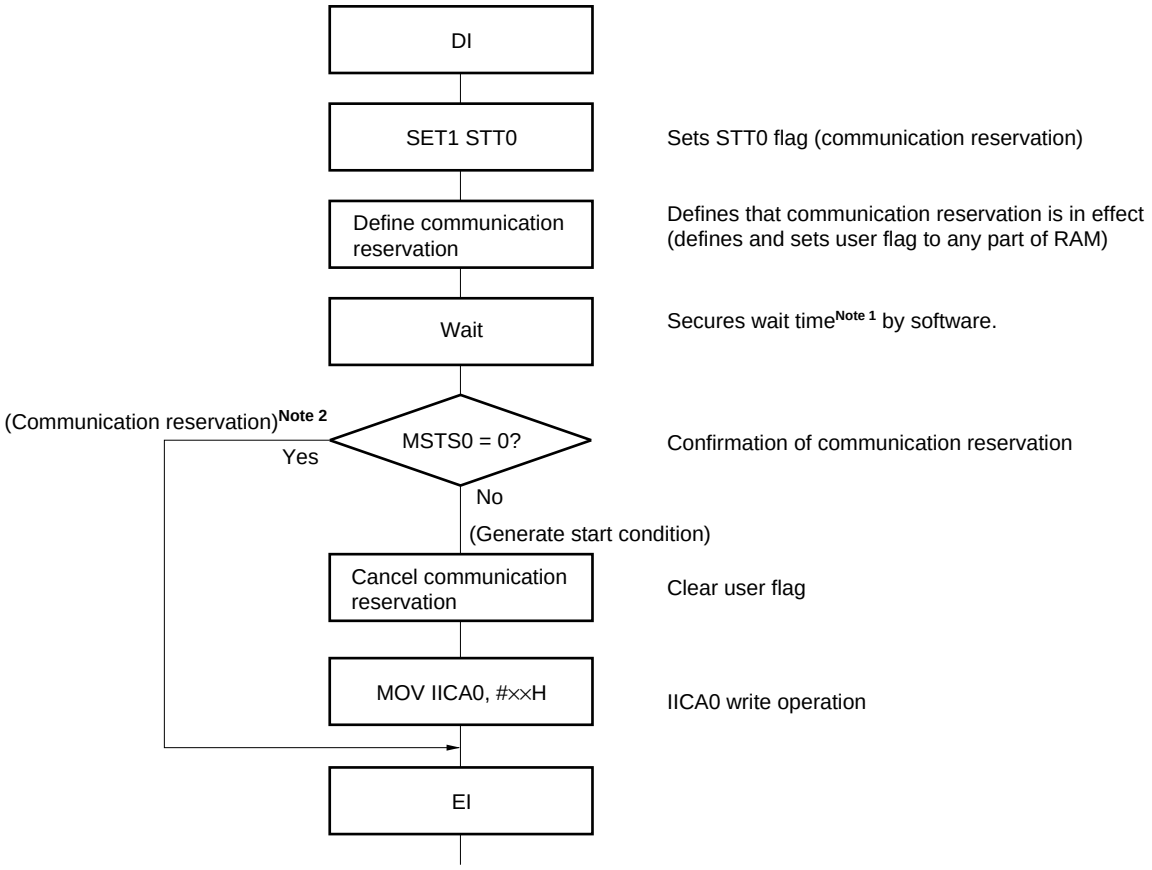


Figure 13-26 shows the communication reservation protocol.

Figure 13-26. Communication Reservation Protocol



Notes 1. The wait time (the number of cycles of f_{CLK}) is calculated as follows.
(IICWL0 setting value + IICWH0 setting value + 4) + $t_F \times 2 \times f_{CLK}$ [clocks]

2. The communication reservation operation executes a write to the IICA shift register 0 (IICA0) when a stop condition interrupt request occurs.

Remark STT0: Bit 1 of IICA control register 00 (IICCTL00)
MSTS0: Bit 7 of IICA status register 0 (IICS0)
IICA0: IICA shift register 0
IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IICA flag register 0 (IICF0) = 1)

When bit 1 (STT0) of IICA control register 00 (IICCTL00) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LREL0) of the IICCTL00 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCF (bit 7 of the IICF0 register). It takes up to 5 clocks until the STCF bit is set to 1 after setting STT0 = 1. Therefore, secure the time by software.

13.5.15 Cautions

(1) When STCEN0 = 0

Immediately after I²C operation is enabled (IICE0 = 1), the bus communication status (IICBSY0 = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IICA control register 01 (IICCTL01).
<2> Set bit 7 (IICE0) of IICA control register 00 (IICCTL00) to 1.
<3> Set bit 0 (SPT0) of the IICCTL00 register to 1.

(2) When STCEN0 = 1

Immediately after I²C operation is enabled (IICE0 = 1), the bus released status (IICBSY0 = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT0 = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I²C communications are already in progress

If I²C operation is enabled and the device participates in communication already in progress when the SDAA0 pin is low and the SCLA0 pin is high, the macro of I²C recognizes that the SDAA0 pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, ACK is returned, but this interferes with other I²C communications. To avoid this, start I²C in the following sequence.

<1> Clear bit 4 (SPIE0) of the IICCTL00 register to 0 to disable generation of an interrupt request signal (INTIICA0) when the stop condition is detected.
<2> Set bit 7 (IICE0) of the IICCTL00 register to 1 to enable the operation of I²C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL0) of the IICCTL00 register to 1 before ACK is returned (4 to 72 clocks after setting the IICE0 bit to 1), to forcibly disable detection.

(4) Setting the STT0 and SPT0 bits (bits 1 and 0 of the IICCTL00 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the SPIE0 bit (bit 4 of the IICTL0 register) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to the IICA shift register 0 (IICA0) after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the SPIE0 bit to 1 when the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) is detected by software.

13.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the RL78/G10 as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

In the I^2C bus multimaster system, whether the bus is released or used cannot be judged by the I^2C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the RL78/G10 takes part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the RL78/G10 looses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

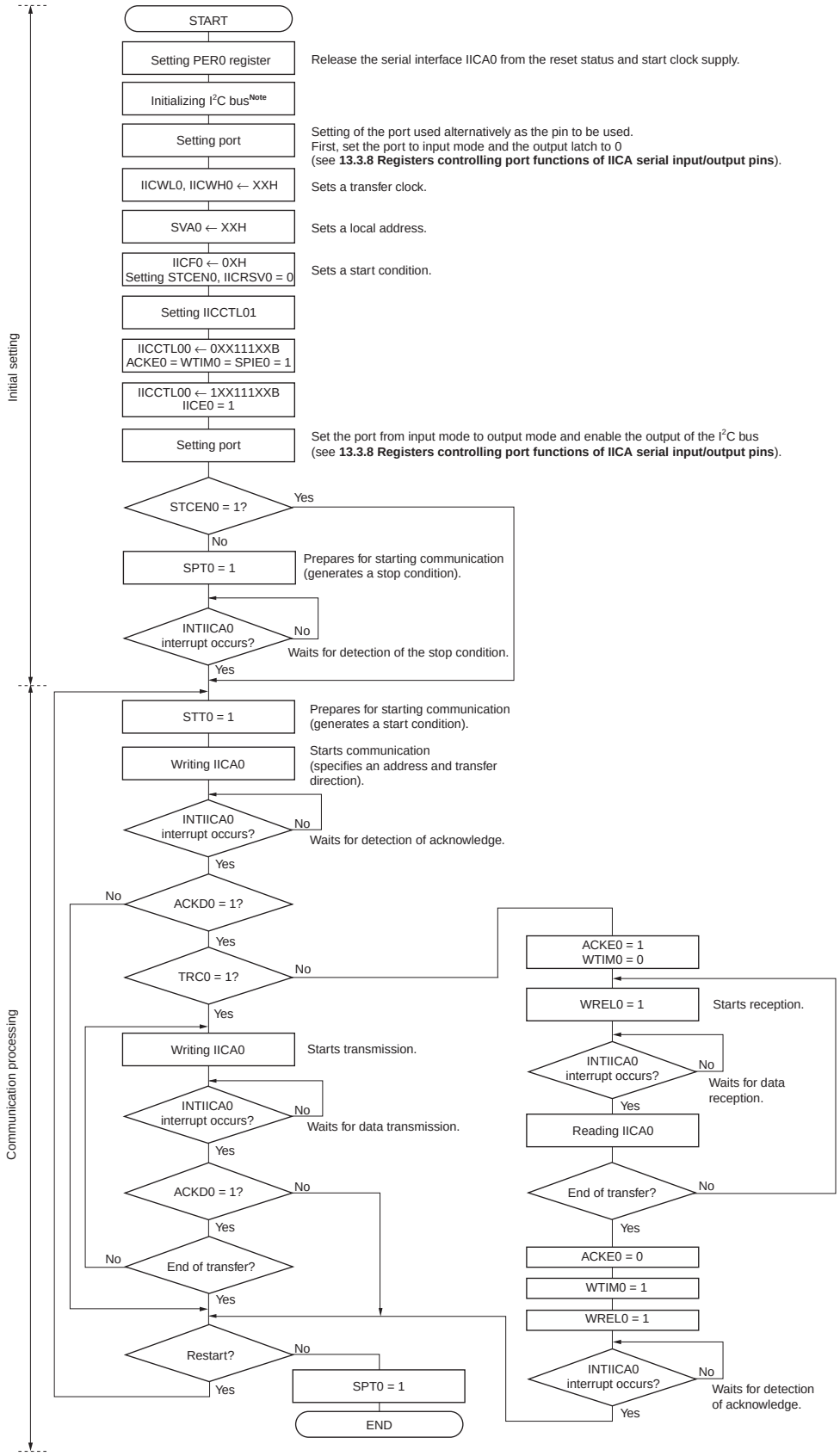
An example of when the RL78/G10 is used as the I^2C bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA0 interrupt occurrence (communication waiting). When an INTIICA0 interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

(1) Master operation in single-master system

Figure 13-27. Master Operation in Single-Master System



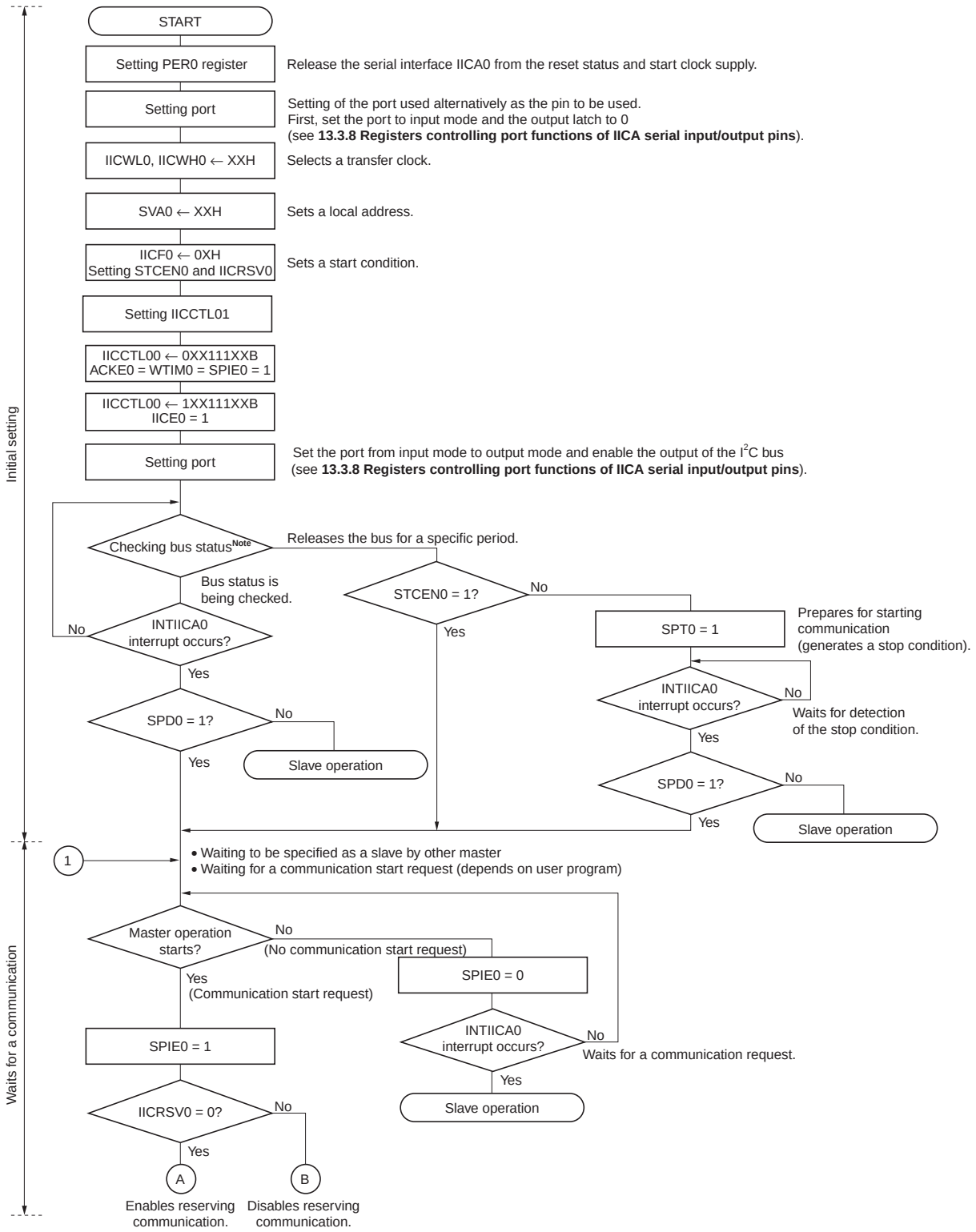
(Note and **Remark** are listed on the next page.)

Note Release (SCLA0 and SDAA0 pins = high level) the I^2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAA0 pin, for example, set the SCLA0 pin in the output port mode, and output a clock pulse from the output port until the SDAA0 pin is constantly at high level.

Remark Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

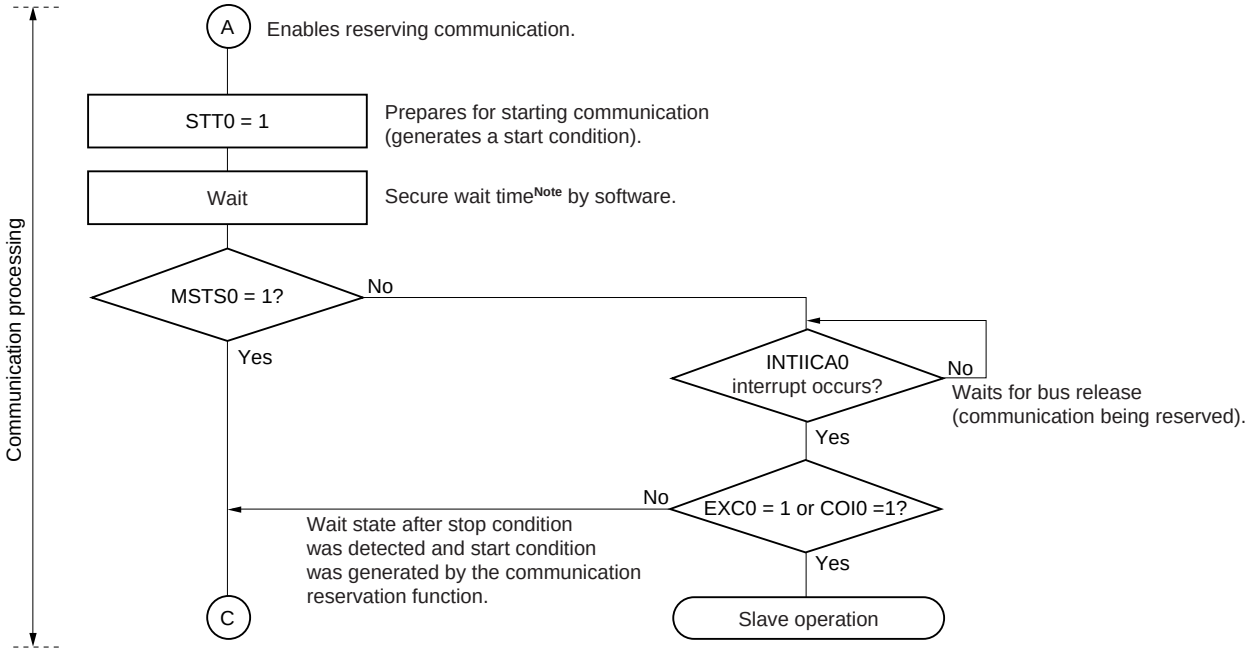
(2) Master operation in multi-master system

Figure 13-28. Master Operation in Multi-Master System (1/3)

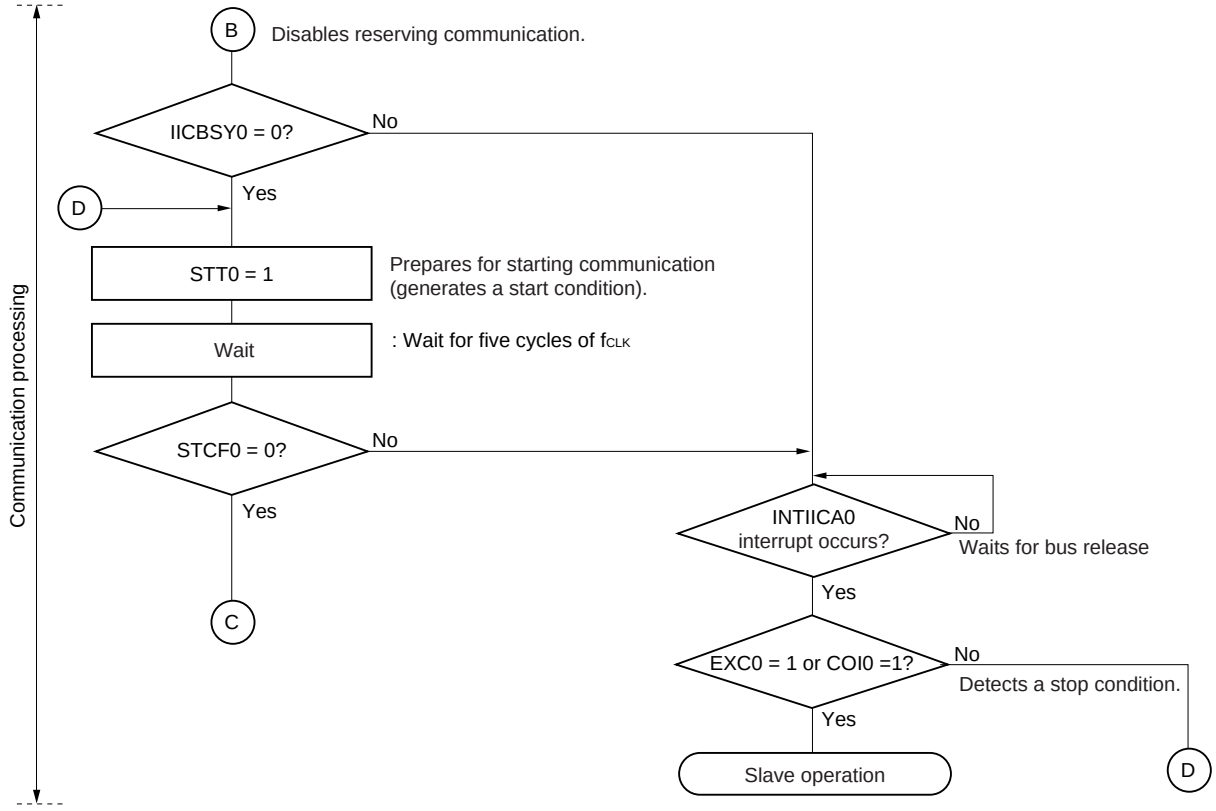


Note Confirm that the bus is released (CLD0 bit = 1, DAD0 bit = 1) for a specific period (for example, for a period of one frame). If the SDAA0 pin is constantly at low level, decide whether to release the I²C bus (SCLA0 and SDAA0 pins = high level) in conformance with the specifications of the product that is communicating.

Figure 13-28. Master Operation in Multi-Master System (2/3)

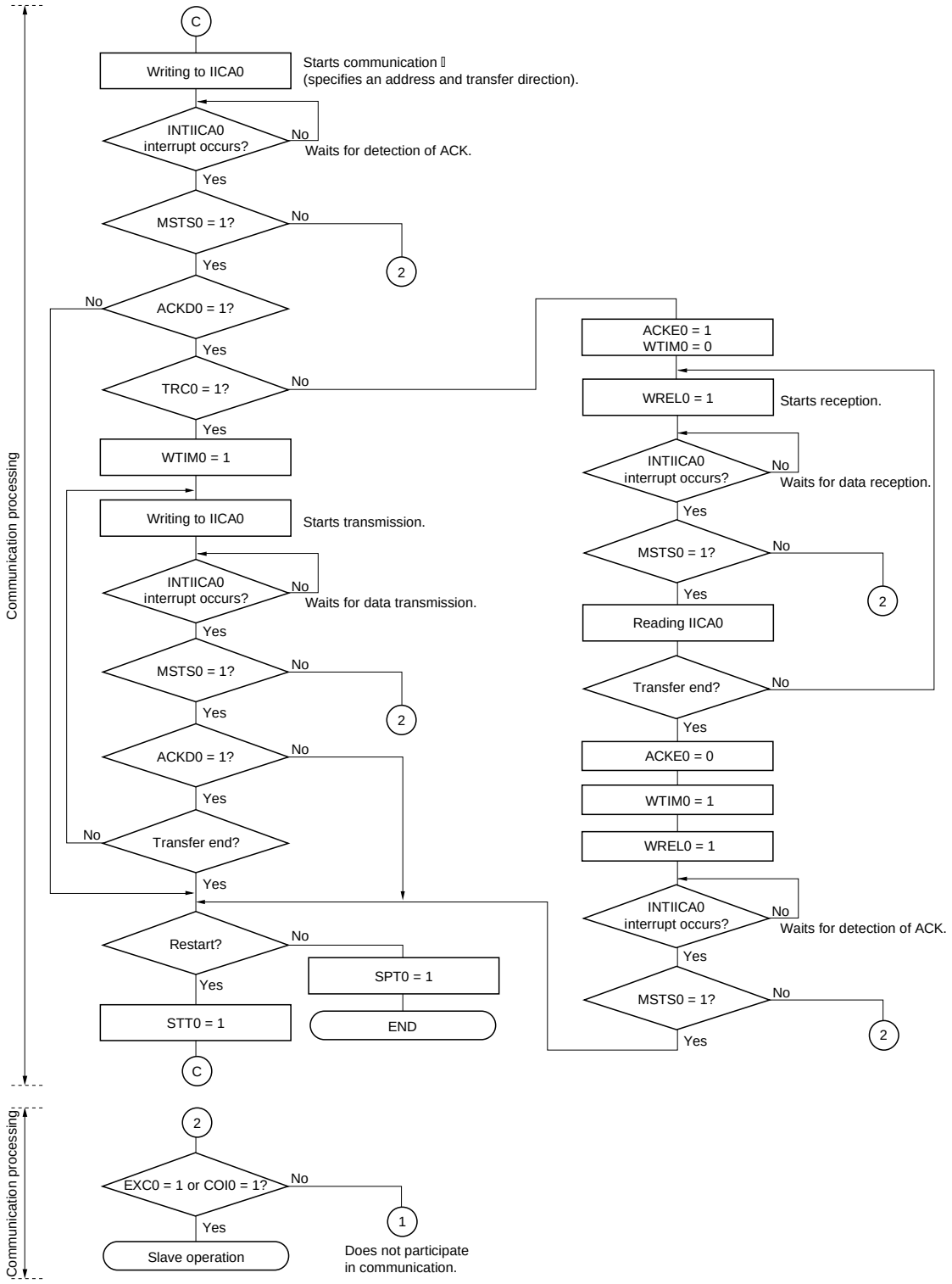


Note The wait time (the number of cycles of f_{CLK}) is calculated as follows.
(IICWL0 setting value + IICWH0 setting value + 4)/f_{CLK} + $t_F \times 2$ [clocks]



Remark IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 13-28. Master Operation in Multi-Master System (3/3)



Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. To use the device as a master in a multi-master system, read the MSTS0 bit each time interrupt INTIICA0 has occurred to check the arbitration result.

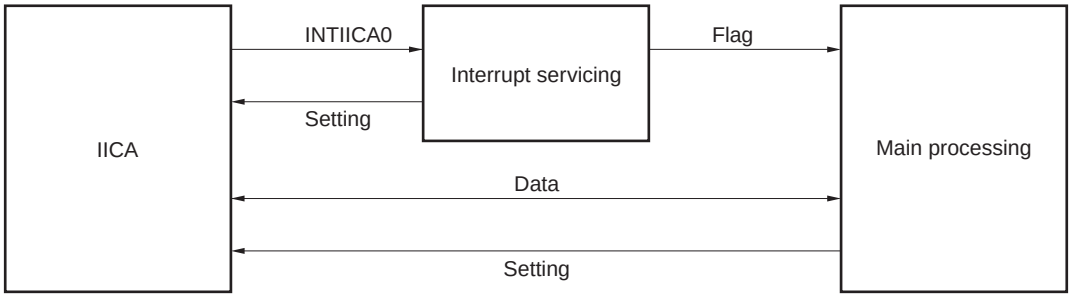
3. To use the device as a slave in a multi-master system, check the status by using the IICA status register 0 (IICS0) and IICA flag register 0 (IICF0) each time interrupt INTIICA0 has occurred, and determine the processing to be performed next.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICA0 interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIICA0 interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICA0.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICA0 interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as the TRC0 bit.

The main processing of the slave operation is explained next.

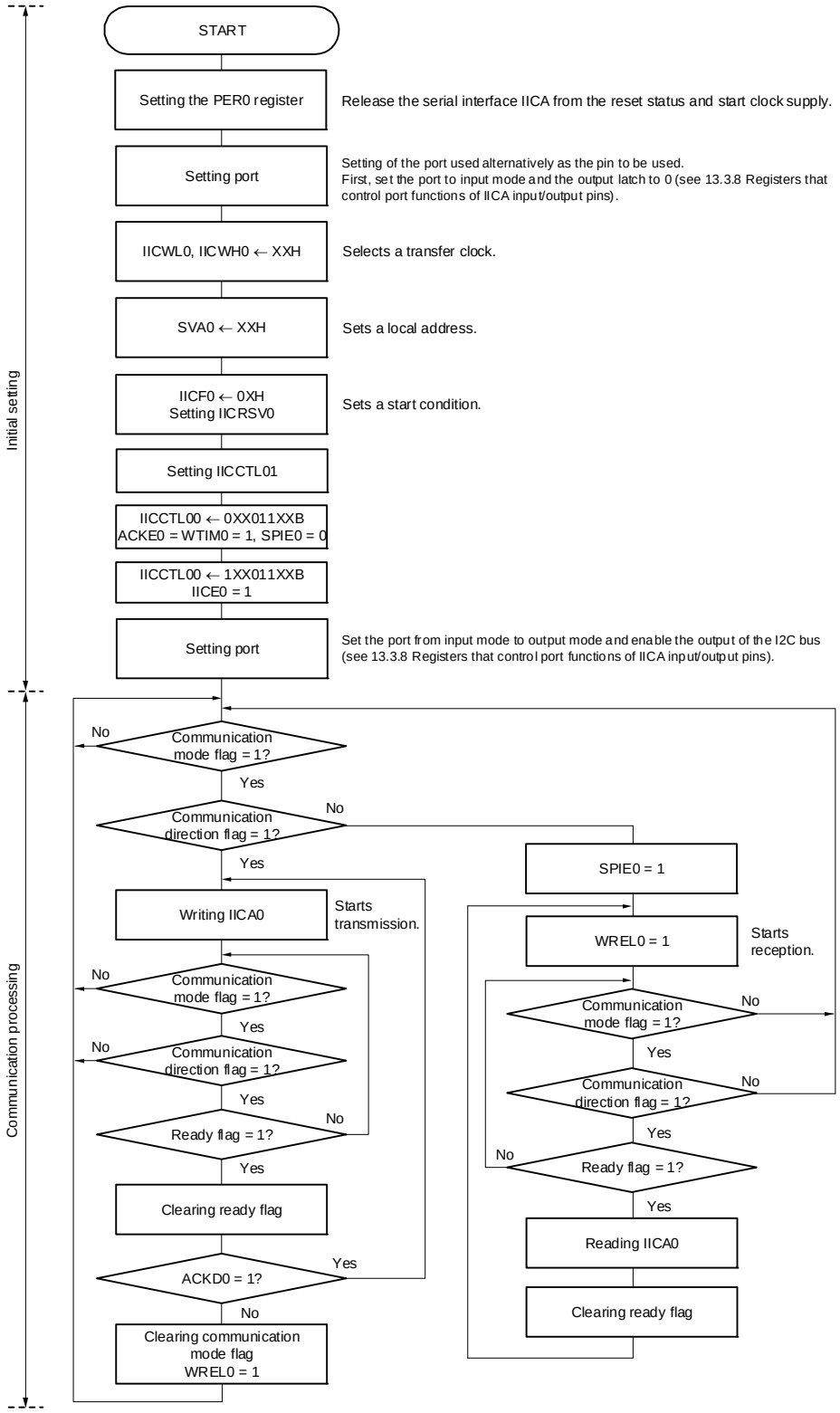
Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns ACK. If ACK is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, ACK is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

<R>

Figure 13-29. Slave Operation Flowchart (1)



Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIICA0 interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA0 interrupt checks the status, and the following operations are performed.

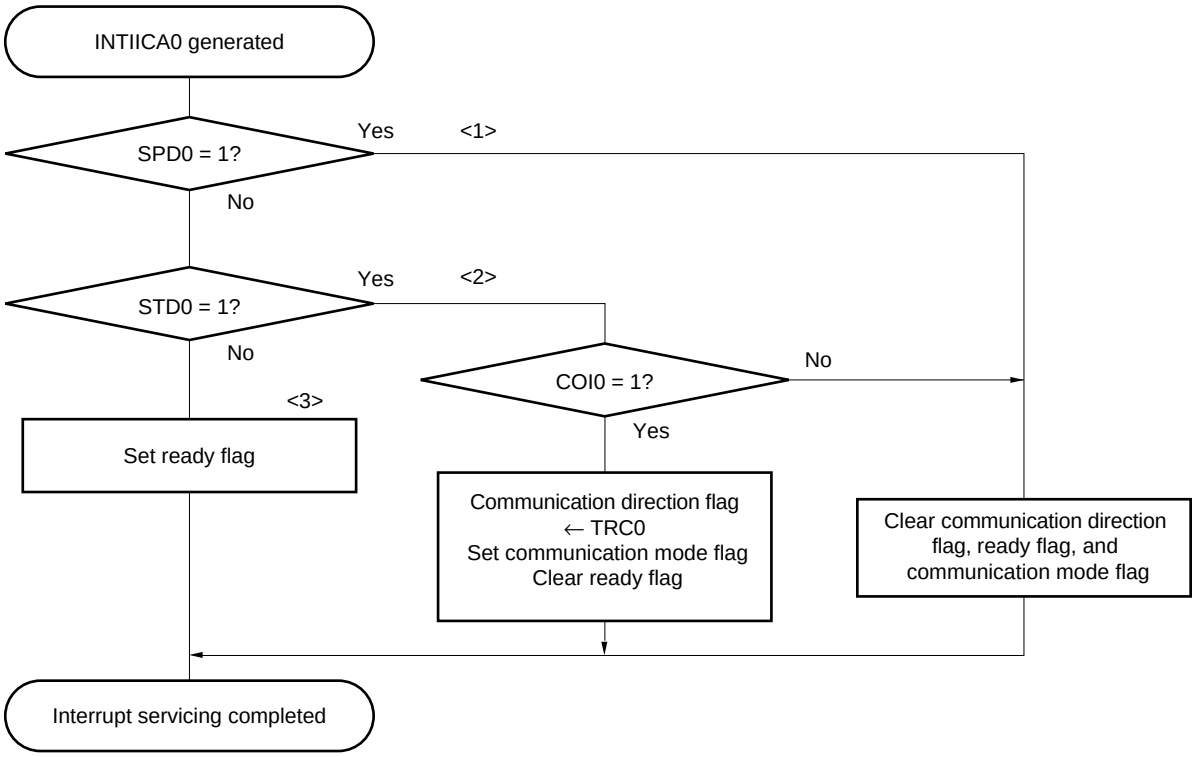
<1> Communication is stopped if the stop condition is issued.

<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).

<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in Figure 13-30 Slave Operation Flowchart (2).

Figure 13-30. Slave Operation Flowchart (2)



13.5.17 Timing of I²C interrupt request (INTIICA0) occurrence

The timing of transmitting or receiving data and generation of interrupt request signal INTIICA0, and the value of the IICA status register 0 (IICS0) when the INTIICA0 signal is generated are shown below.

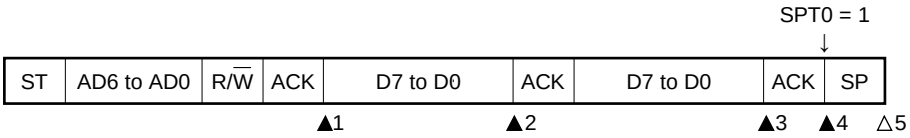
Remark

ST:	Start condition
AD6 to AD0:	Address
R/$\overline{W}$:	Transfer direction specification
ACK:	Acknowledge
D7 to D0:	Data
SP:	Stop condition

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM0 = 0

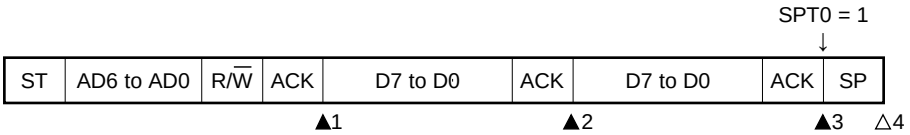


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B
▲3: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

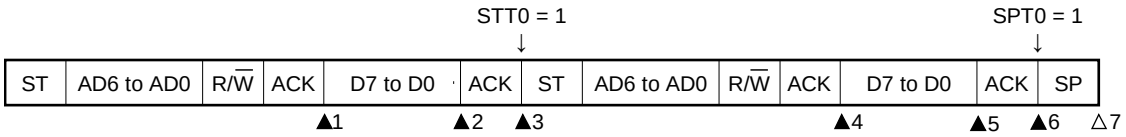


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B
▲3: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0

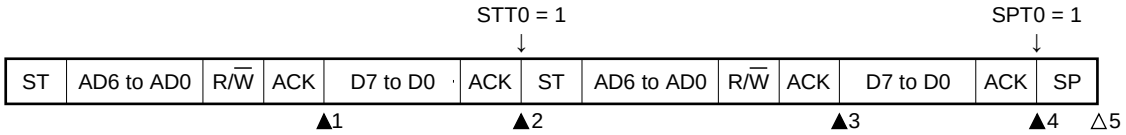


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note 1]
▲3: IICS0 = 1000××00B (Clears the WTIM0 bit to 0[Note 2], sets the STT0 bit to 1)
▲4: IICS0 = 1000×110B
▲5: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note 3]
▲6: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△7: IICS0 = 00000001B

Notes 1. To generate a start condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.
2. Clear the WTIM0 bit to 0 to restore the original setting.
3. To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

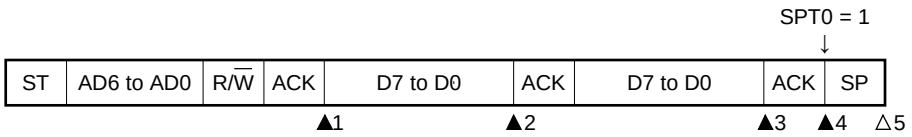


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
▲3: IICS0 = 1000×110B
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0

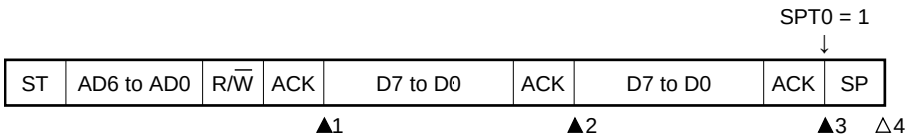


▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×000B
▲3: IICS0 = 1010×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×100B
▲3: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) When WTIM0 = 0

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP

▲1 ▲2 ▲3 △4

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP

▲1 ▲2 ▲3 △4

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, matches with SVA0)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
				▲1	▲2					▲3	▲4	△5

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches with SVA0)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
				▲1		▲2				▲3	▲4	△5

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= extension code))

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
				▲1	▲2				▲3		▲4	△5

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= extension code))

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
				▲1		▲2			▲3	▲4	▲5	△6

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×110B
▲5: IICS0 = 0010××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
				▲1	▲2					▲3		△4

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
				▲1		▲2				▲3		△4

▲1: IICS0 = 0001×110B
▲2: IICS0 = 0001××00B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(3) Slave device operation (when receiving extension code)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

(i) When WTIM0 = 0

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1		▲2		▲3	△4

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1	▲2		▲3		▲4 △5

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, matches SVA0)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1		▲2					▲3	▲4	△5

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0001×110B
▲4: IICS0 = 0001×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, matches SVA0)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1	▲2		▲3				▲4	▲5	△6

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0001×110B
▲5: IICS0 = 0001××00B
△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, extension code reception)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1		▲2				▲3		▲4	△5

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×010B
▲4: IICS0 = 0010×000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (after restart, extension code reception)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1	▲2		▲3			▲4	▲5	▲6	△7

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 0010×010B
▲5: IICS0 = 0010×110B
▲6: IICS0 = 0010××00B
△7: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1		▲2					▲3		△4

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 00000×10B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1 (after restart, does not match address (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1	▲2		▲3				▲4		△5

▲1: IICS0 = 0010×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010××00B
▲4: IICS0 = 00000×10B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
								△1

△1: IICS0 = 00000001B

Remark △: Generated only when SPIE0 = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

(i) When WTIM0 = 0

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1		▲2		▲3	△4

▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×000B
▲3: IICS0 = 0001×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP

▲1 ▲2 ▲3 △4

▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

(i) When WTIM0 = 0

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP

▲1 ▲2 ▲3 △4

▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×000B
▲3: IICS0 = 0010×000B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1	▲2		▲3		▲4 △5

▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS0 bit each time interrupt request signal INTIICA0 has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
				▲1				△2

▲1: IICS0 = 01000110B
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When arbitration loss occurs during transmission of extension code

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1					△2

▲1: IICS0 = 0110×010B
Sets LREL0 = 1 by software
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

(i) When WTIM0 = 0

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
				▲1	▲2			△3

▲1: IICS0 = 10001110B
▲2: IICS0 = 01000000B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
				▲1		▲2		△3

▲1: IICS0 = 10001110B
▲2: IICS0 = 01000100B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatches with SVA0)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to Dn	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
				▲1					▲2		△3

▲1: IICS0 = 1000×110B
▲2: IICS0 = 01000110B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care
n = 6 to 0

(ii) Extension code

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to Dn	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
				▲1				▲2			△3

▲1: IICS0 = 1000×110B
▲2: IICS0 = 01100010B
Sets LREL0 = 1 by software
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care
n = 6 to 0

(e) When loss occurs due to stop condition during data transfer

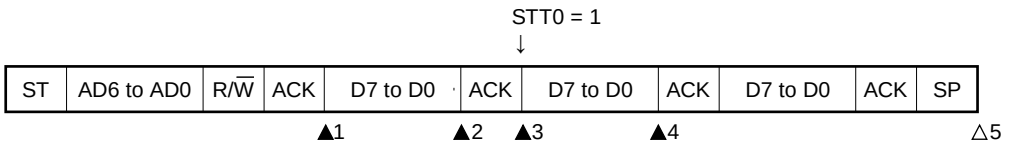
ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to Dn	SP
				▲1	△2

▲1: IICS0 = 10000110B
△2: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care
n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

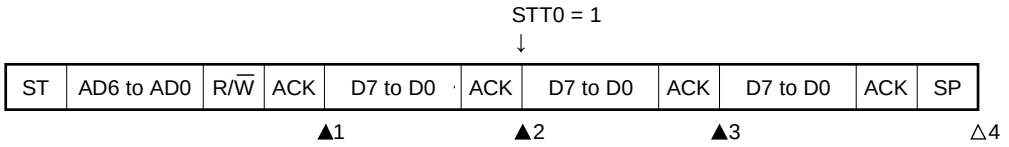
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000×100B (Clears the WTIM0 bit to 0)
▲4: IICS0 = 01000000B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

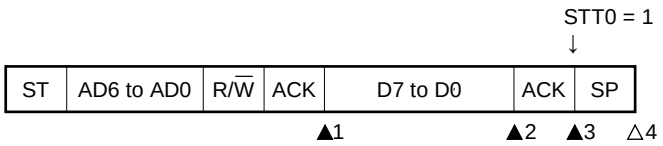


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the STT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

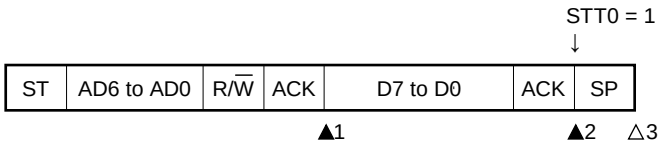
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△4: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1

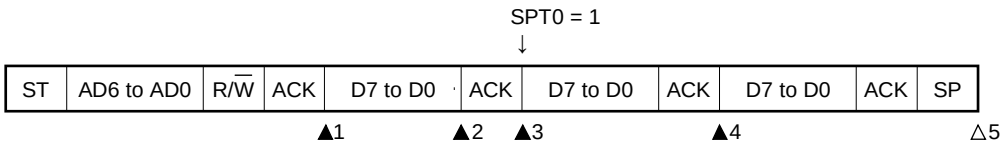


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△3: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

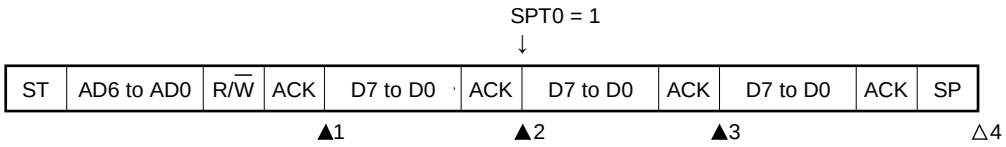
(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)
▲3: IICS0 = 1000×100B (Clears the WTIM0 bit to 0)
▲4: IICS0 = 01000100B
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the SPT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

13.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC0 bit (bit 3 of the IICA status register 0 (IICS0)), which specifies the data transfer direction, and then starts serial communication with the slave device.

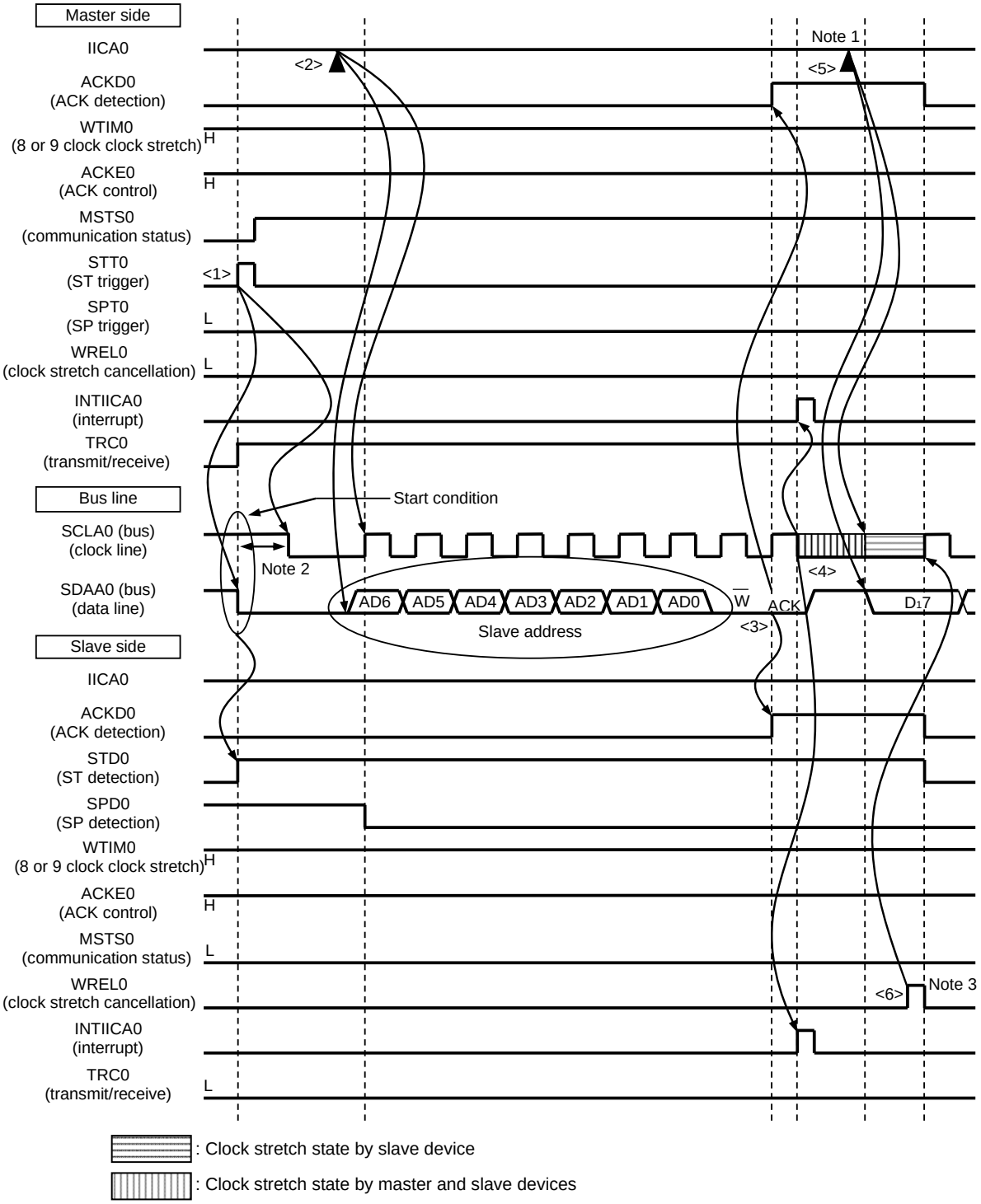
Figures 13-31 and 13-32 show timing charts of the data communication.

The IICA shift register 0 (IICA0)'s shift operation is synchronized with the falling edge of the serial clock (SCLA0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAA0 pin.

Data input via the SDAA0 pin is captured into IICA0 at the rising edge of SCLA0.

Figure 13-31. Example of Master to Slave Communication
(9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (1/4)

(1) Start condition ~ address ~ data



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during transmission by a master device.

2. Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. For releasing clock stretch state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 13-31 are explained below.

<1> The start condition trigger is set by the master device (STT0 = 1) and a start condition (SDAA0 = 0 and SCLA0 = 1) is generated once the bus data line (SDAA0) goes low. When the start condition is subsequently detected, the master device enters the master device communication status (MSTS0 = 1). The master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register 0 (IICA0) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device**Note**, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match)**Note**.

<5> The master device writes the data to transmit to the IICA0 register and releases the clock stretch status that it set by the master device.

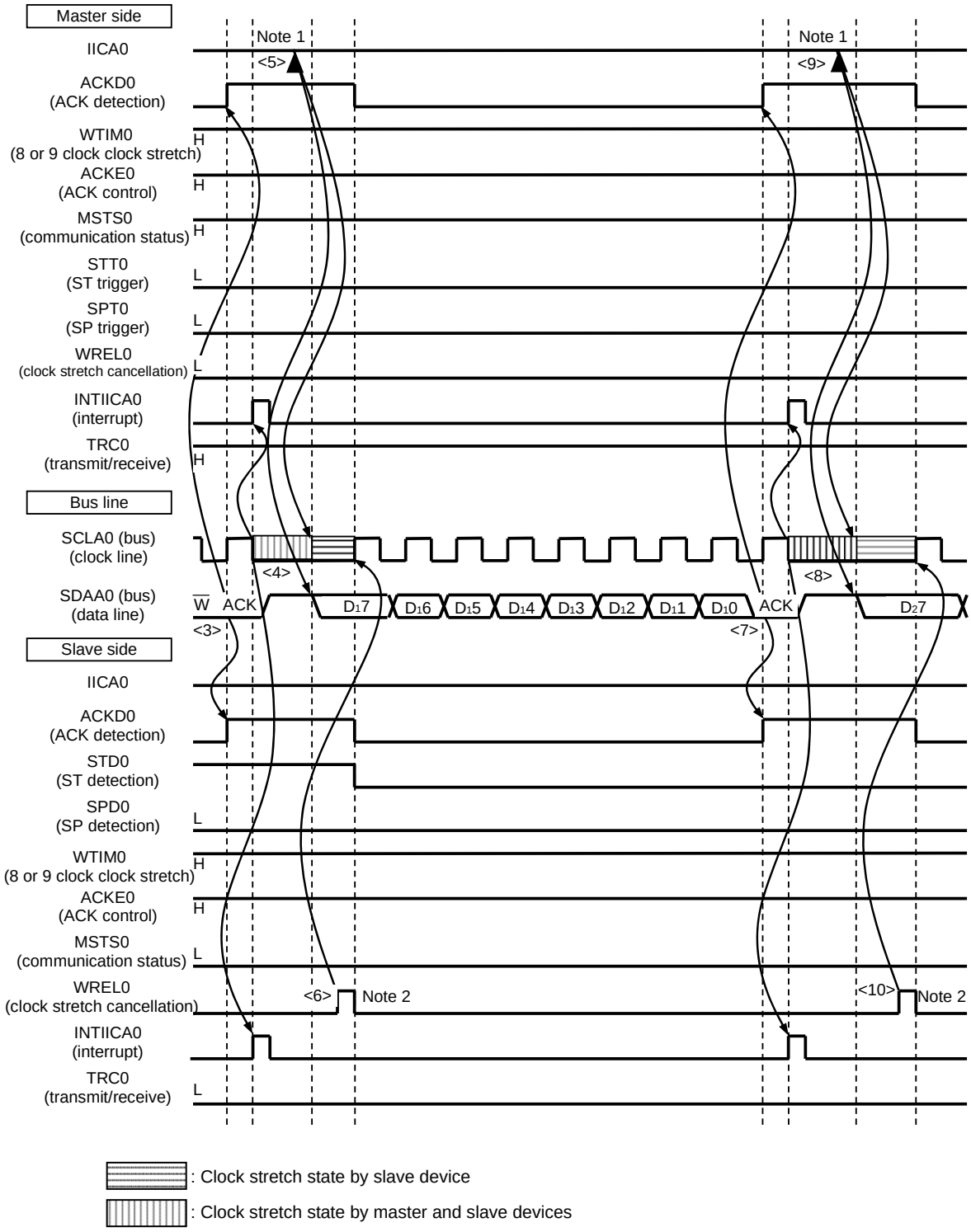
<6> If the slave device releases the clock stretch status (WREL0 = 1), the master device starts transferring data to the slave device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 13-31 following descriptions the entire procedure for communicating data using the I^2C bus.
Figure 13-31 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 13-31 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 13-31 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 13-31. Example of Master to Slave Communication
(9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (2/4)

(2) Address ~ data ~ data



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during transmission by a master device.

2. For releasing clock stretch state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 13-31 are explained below.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match)[Note].

<5> The master device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the clock stretch status that it set by the master device.

<6> If the slave device releases the clock stretch status (WREL0 = 1), the master device starts transferring data to the slave device.

<7> After data transfer is completed, because of ACKE0 = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<9> The master device writes the data to transmit to the IICA0 register and releases the clock stretch status that it set by the master device.

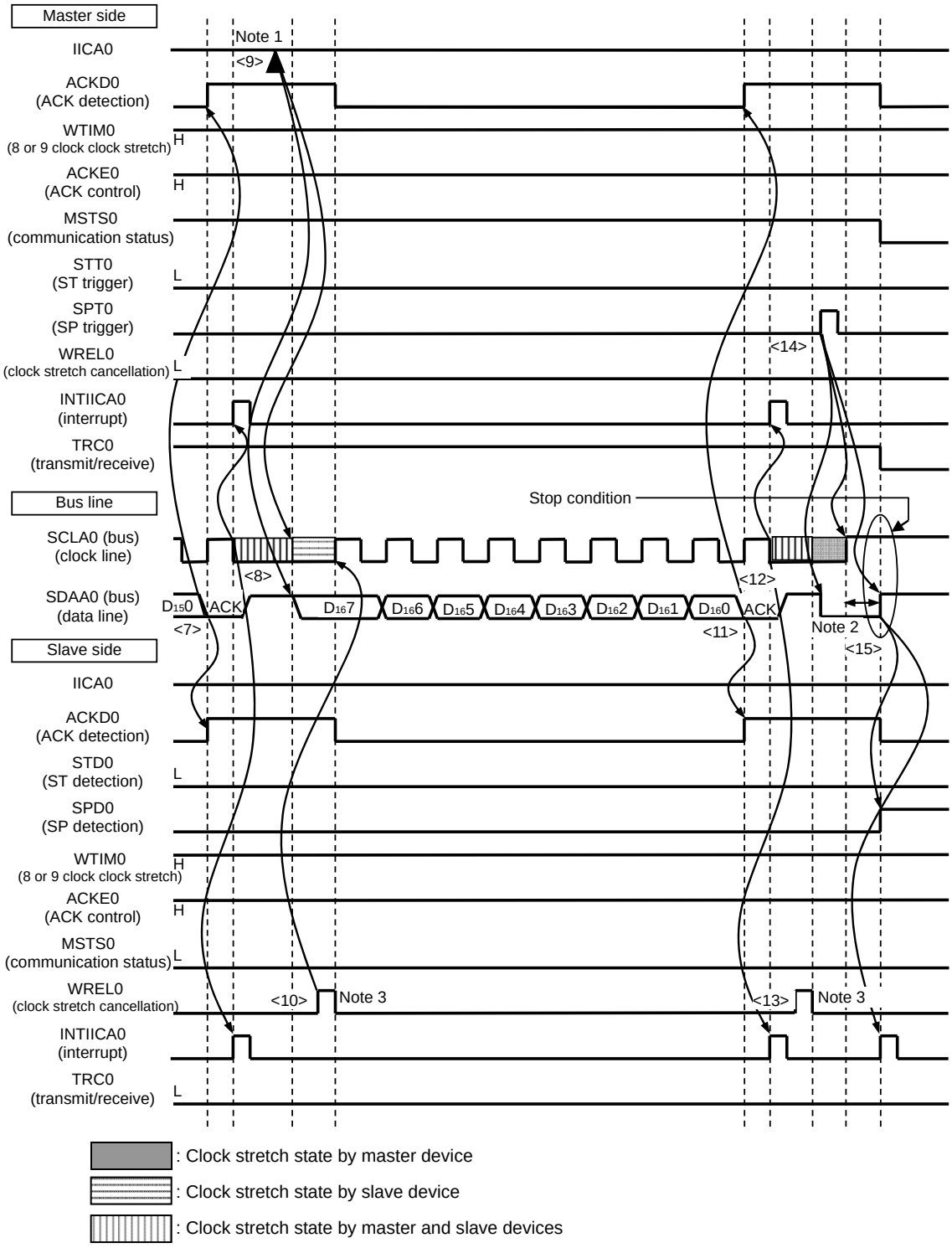
<10> The slave device reads the received data and releases the clock stretch status (WREL0 = 1). The master device then starts transferring data to the slave device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 13-31 following descriptions the entire procedure for communicating data using the I^2C bus.
Figure 13-31 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 13-31 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 13-31 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 13-31. Example of Master to Slave Communication
(9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (3/4)

(3) Data ~ data ~ Stop condition



Notes 1. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during transmission by a master device.

2. Make sure that the time between the rise of the SCLA0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. For releasing clock stretch state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 13-31 are explained below.

<7> After data transfer is completed, because of ACKE0 = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the clock stretch status that it set by the master device.

<10> The slave device reads the received data and releases the clock stretch status (WREL0 = 1). The master device then starts transferring data to the slave device.

<11> When data transfer is complete, the slave device (ACKE0 =1) sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<12> The master device and slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<13> The slave device reads the received data and releases the clock stretch status (WREL0 = 1).

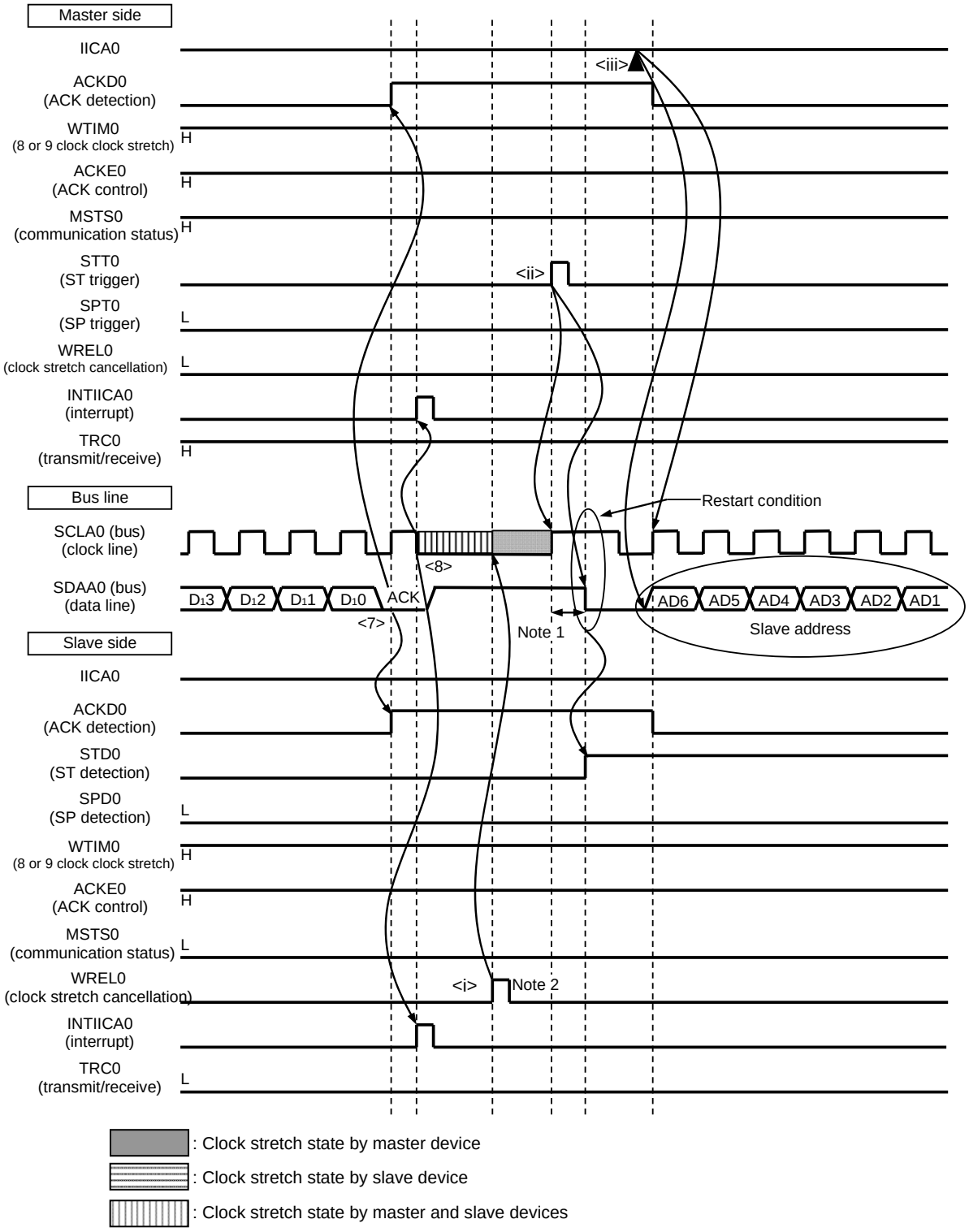
<14> By the master device setting a stop condition trigger (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the bus clock line is set (SCLA0 = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAA0 = 1), the stop condition is then generated (i.e. SCLA0 =1 changes SDAA0 from 0 to 1).

<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICA0: stop condition).

Remark <1> to <15> in Figure 13-31 represent the entire procedure for communicating data using the I^2C bus. Figure 13-31 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 13-31 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 13-31 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 13-31. Example of Master to Slave Communication
(9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (4/4)

(4) Data ~ restart condition ~ address



Notes 1. Make sure that the time between the rise of the SCLA0 pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

2. For releasing clock stretch state during reception of a slave device, write “FFH” to IICA0 or set the WREL0 bit.

The following describes the operations in Figure 13-31 (4) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <1> to <3> are performed. These steps return the processing to step <3>, the data transmission step.

<7> After data transfer is completed, because of ACKE0 = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

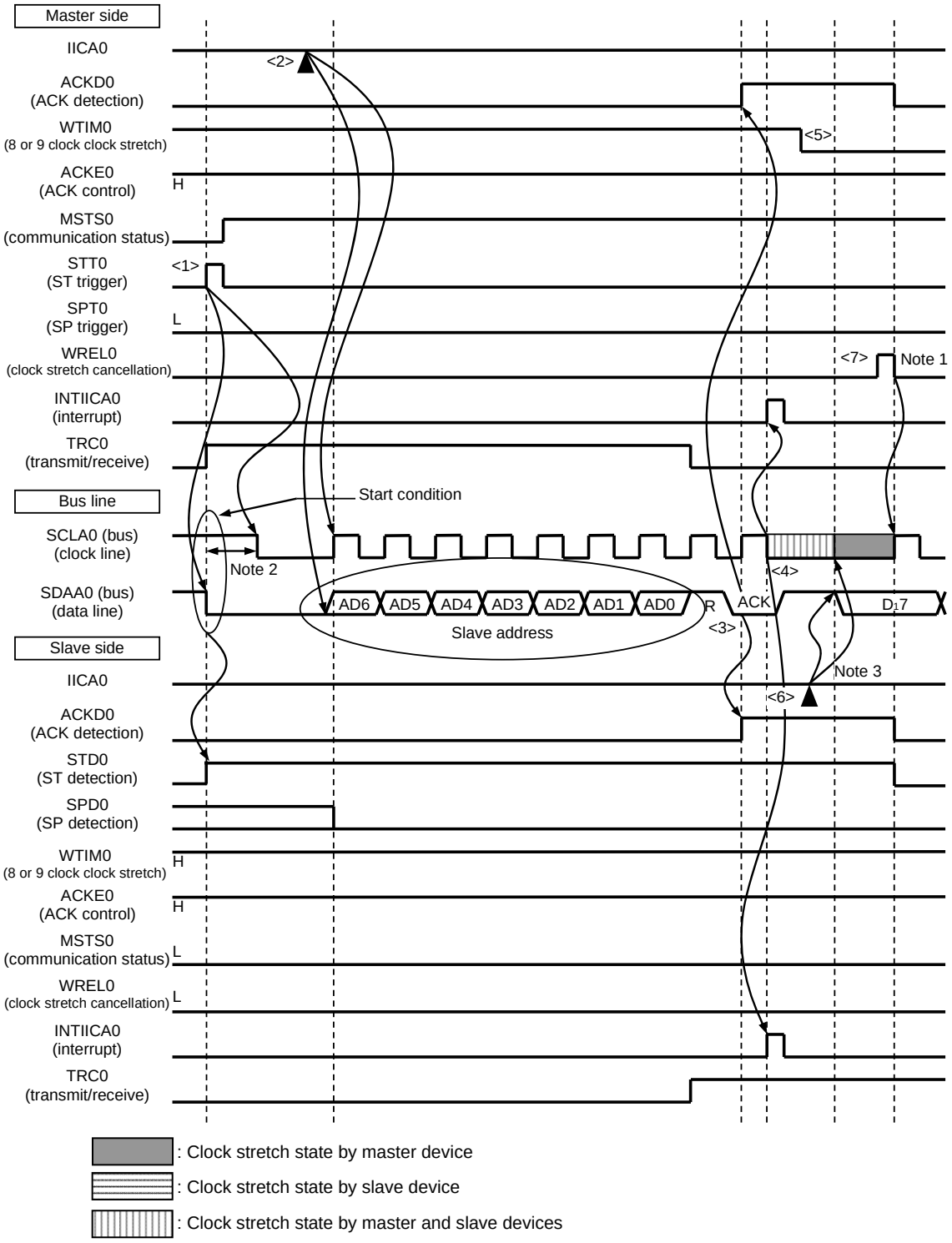
<i> The slave device reads the received data and releases the clock stretch status (WREL0 = 1).

<ii> The start condition trigger is set again by the master device (STT0 = 1) and a start condition (i.e. SCLA0 =1 changes SDAA0 from 1 to 0) is generated once the bus clock line goes high (SCLA0 = 1) and the bus data line goes low (SDAA0 = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<iii> The master device writing the address + R/W (transmission) to the IICA shift register (IICA0) enables the slave address to be transmitted.

Figure 13-32. Example of Slave to Master Communication
(8-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (1/3)

(1) Start condition ~ address ~ data



Notes 1. For releasing clock stretch state during reception of a master device, write “FFH” to IICA0 or set the WREL0 bit.

2. Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during transmission by a slave device.

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 13-32 are explained below.

<1> The start condition trigger is set by the master device (STT0 = 1) and a start condition (i.e. SCLA0 =1 changes SDAA0 from 1 to 0) is generated once the bus data line goes low (SDAA0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS0 = 1). The master device is ready to communicate once the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register 0 (IICA0) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device **Note**, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match) **Note**.

<5> The timing at which the master device sets the clock stretch status changes to the 8th clock (WTIM0 = 0).

<6> The slave device writes the data to transmit to the IICA0 register and releases the clock stretch status that it set by the slave device.

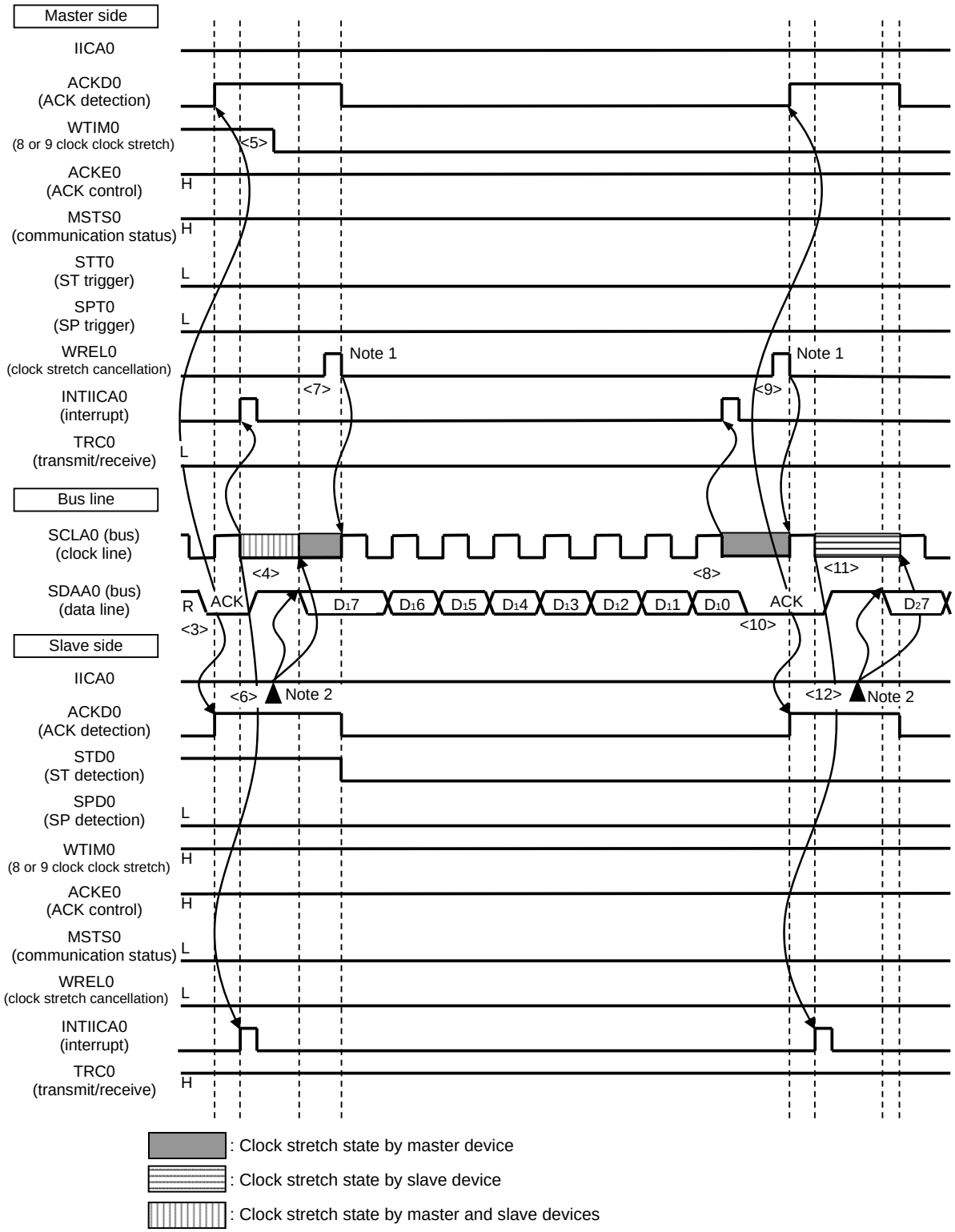
<7> The master device releases the clock stretch status (WREL0 = 1) and starts transferring data from the slave device to the master device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 13-32 following descriptions the entire procedure for communicating data using the I^2C bus.
Figure 13-32 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 13-32 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 13-32 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 13-32. Example of Slave to Master Communication
(8-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (2/3)

(2) Address ~ data ~ data



Notes 1. For releasing clock stretch state during reception of a master device, write “FFH” to IICA0 or set the WREL0 bit.

2. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during transmission by a slave device.

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 13-32 are explained below.

<3> In the slave device if the address received matches the address (SVA0 value) of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA0: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLA0 = 0) and issues an interrupt (INTIICA0: address match)[Note].

<5> The master device changes the timing of the clock stretch status to the 8th clock (WTIM0 = 0).

<6> The slave device writes the data to transmit to the IICA shift register 0 (IICA0) and releases the clock stretch status that it set by the slave device.

<7> The master device releases the clock stretch status (WREL0 = 1) and starts transferring data from the slave device to the master device.

<8> The master device sets a clock stretch status (SCLA0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA0: end of transfer). Because of ACKE0 = 1 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the clock stretch status (WREL0 = 1).

<10> The ACK is detected by the slave device (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA0: end of transfer).

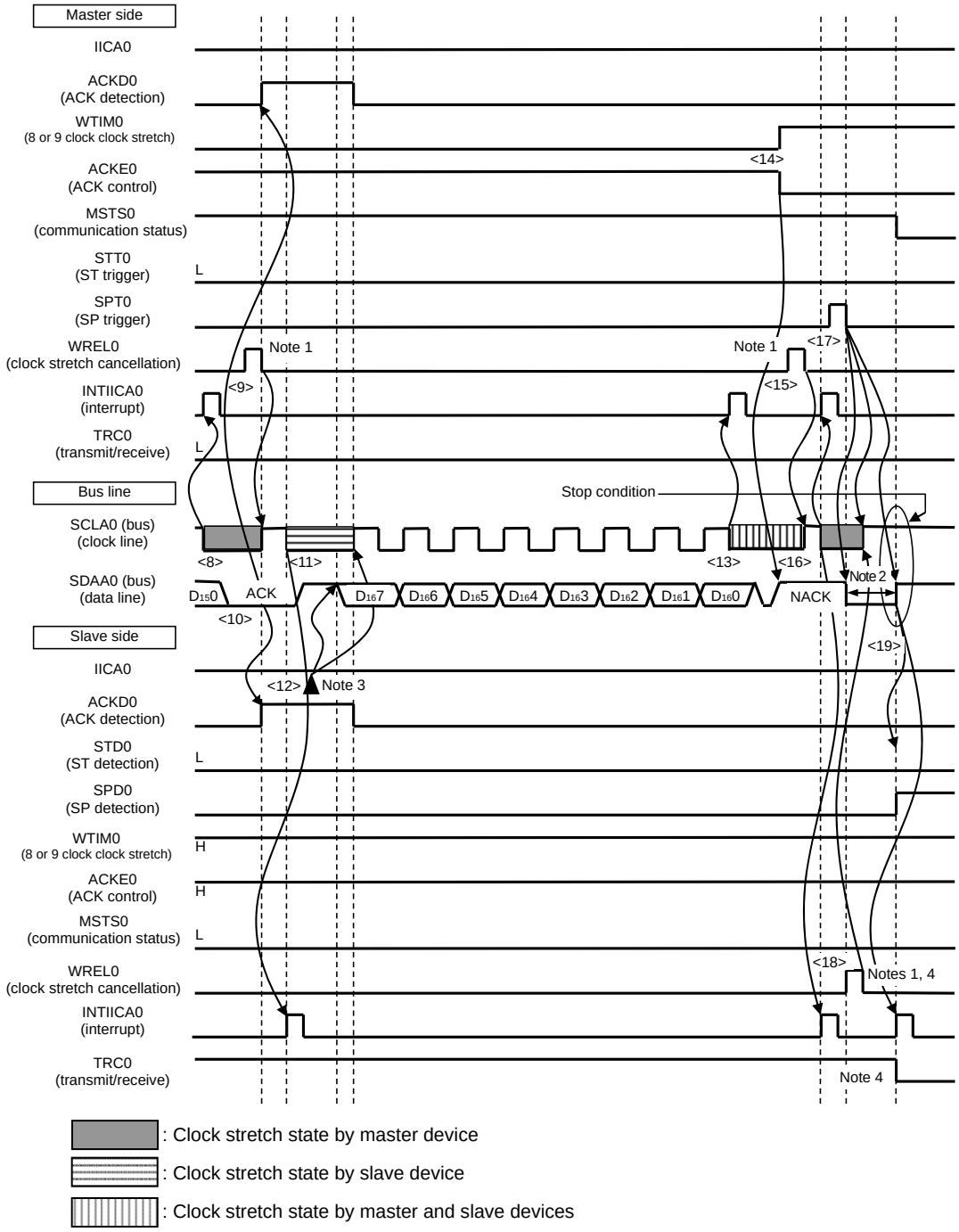
<12> By the slave device writing the data to transmit to the IICA0 register, the clock stretch status set by the slave device is released. The slave device then starts transferring data to the master device.

Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAA0 = 1). The slave device also does not issue the INTIICA0 interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICA0 interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 13-32 following descriptions the entire procedure for communicating data using the I^2C bus.
Figure 13-32 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 13-32 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 13-32 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 13-32. Example of Slave to Master Communication
(8-Clock and 9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (3/3)

(3) Data ~ data ~ stop condition



Notes 1. To cancel a clock stretch state, write “FFH” to IICA0 or set the WREL0 bit.

2. Make sure that the time between the rise of the SCLA0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA0, not setting the WREL0 bit, in order to cancel a clock stretch state during slave transmission.

4. If a clock stretch state during transmission by a slave device is canceled by setting the WREL0 bit, the TRC0 bit will be cleared.

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 13-32 are explained below.

<8> The master device sets a clock stretch status (SCLA0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA0: end of transfer). Because of ACKE0 = 0 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the clock stretch status (WREL0 = 1).

<10> The ACK is detected by the slave device (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA0: end of transfer).

<12> By the slave device writing the data to transmit to the IICA register, the clock stretch status set by the slave device is released. The slave device then starts transferring data to the master device.

<13> The master device issues an interrupt (INTIICA0: end of transfer) at the falling edge of the 8th clock, and sets a clock stretch status (SCLA0 = 0). Because ACK control (ACKE0 = 1) is performed, the bus data line is at the low level (SDAA0 = 0) at this stage.

<14> The master device sets NACK as the response (ACKE0 = 0) and changes the timing at which it sets the clock stretch status to the 9th clock (WTIM0 = 1).

<15> If the master device releases the clock stretch status (WREL0 = 1), the slave device detects the NACK (ACK = 0) at the rising edge of the 9th clock.

<16> The master device and slave device set a clock stretch status (SCLA0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA0: end of transfer).

<17> When the master device issues a stop condition (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the master device releases the clock stretch status. The master device then waits until the bus clock line is set (SCLA0 = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the clock stretch status (WREL0 = 1) to end communication. Once the slave device releases the clock stretch status, the bus clock line is set (SCLA0 = 1).

<19> Once the master device recognizes that the bus clock line is set (SCLA0 = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAA0 = 1) and issues a stop condition (i.e. SCLA0 =1 changes SDAA0 from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (INTIICA0: stop condition).

Remark <1> to <19> in Figure 13-32 following descriptions the entire procedure for communicating data using the I^2C bus.
Figure 13-32 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 13-32 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 13-32 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

CHAPTER 14 INTERRUPT FUNCTIONS

The interrupt function switches the program execution to other processing. When the branch processing is finished, the program returns to the interrupted processing.

The number of interrupt sources differs, depending on the product.

		10-pin products	16-pin products
Maskable interrupts	External	3	5
	Internal	8	14

14.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR10L, PR10H, PR01L, PR11L).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the priority of vectored interrupt servicing. For the priority order, see **Table 14-1**.

A standby release signal is generated and STOP and HALT modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

14.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to four reset sources (see **Table 14-1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 14-1. Interrupt Source List (10-pin Products)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal /External	Vector Table Address	Basic Configuration Type Note 2
		Name	Trigger			
Maskable	0	INTWDTI	Watchdog timer interval (75% of overflow time +3/(4 × f_{IL}))	Internal	0004H	(a)
	1	INTP0	Pin input edge detection	External	0006H	(b)
	2	INTP1			0008H	
	3	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end	Internal	000AH	(a)
	4	INTSR0	UART0 reception transfer end		000CH	
	5	INTSRE0	UART0 reception communication error occurrence		000EH	
	6	INTTM01H	End of counting or start of operations by timer channel 1 (at higher 8-bit timer operation)		0010H	
	7	INTTM00	End of counting, completion of capture, or start of operations by timer channel 0		0012H	
	8	INTTM01	End of counting, completion of capture, or start of operations by timer channel 1 (at 16-bit or lower 8-bit timer operation)		0014H	
	9	INTAD	End of A/D conversion		0016H	
	10	INTKR	Key return signal detection	External	0018H	(c)
Software	–	BRK	Execution of BRK instruction	–	007EH	(d)
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	0000H	–
		SPOR	Selectable power-on-reset			
		WDT	Overflow of watchdog timer			
		TRAP	Execution of illegal instruction Note 3			

Notes

1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 10 indicates the lowest priority.
2. Basic configuration types (a) to (d) correspond to (a) to (d) in Figure 14-1.
3. When the instruction code in FFH is executed.
   No reset is issued even if an illegal instruction is executed during emulation with the on-chip debug emulator.

Table 14-2. Interrupt Source List (16-pin products)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal /External	Vector Table Address	Basic Configuration Type Note 2
		Name	Trigger			
Maskable	0	INTWDTI	Watchdog timer interval (75% of overflow time +3/(4 × f_{IL}))	Internal	0004H	(a)
	1	INTP0	Pin input edge detection	External	0006H	(b)
	2	INTP1			0008H	
	3	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end	Internal	000AH	(a)
	4	INTSR0 INTCSI01	UART0 reception transfer end End of CSI01 communication		000CH	
	5	INTSRE0	UART0 reception communication error occurrence		000EH	
	6	INTTM01H	End of counting or start of operations by timer channel 1 (at higher 8-bit timer operation)		0010H	
	7	INTTM00	End of counting, completion of capture, or start of operations by timer channel 0		0012H	
	8	INTTM01	End of counting, completion of capture, or start of operations by timer channel 1 (at 16-bit or lower 8-bit timer operation)		0014H	
	9	INTAD	End of A/D conversion		0016H	
	10	INTKR	Key return signal detection	External	0018H	(c)
	11	INTP2	Pin input edge detection		001AH	(b)
	12	INTP3			001CH	
	13	INTTM03H	End of counting or start of operations by timer channel 3 (at higher 8-bit timer operation)	Internal	001EH	(a)
	14	INTIICA0	End of IICA communication		0020H	
	15	INTTM02	End of counting, completion of capture, or start of operations by timer channel 2		0022H	
	16	INTTM03	End of counting, completion of capture, or start of operations by timer channel 3 (at 16-bit or lower 8-bit timer operation)		0024H	
	17	INTIT	Signal detection by the interval timer		0026H	
	18	INTCMP0	Valid edge detection by the comparator		0028H	
Software	—	BRK	Execution of BRK instruction	—	007EH	(d)
Reset	—	RESET	$\overline{\text{RESET}}$ pin input	—	0000H	—
		SPOR	Selectable power-on-reset			
		WDT	Overflow of watchdog timer			
		TRAP	Execution of illegal instruction Note 3			

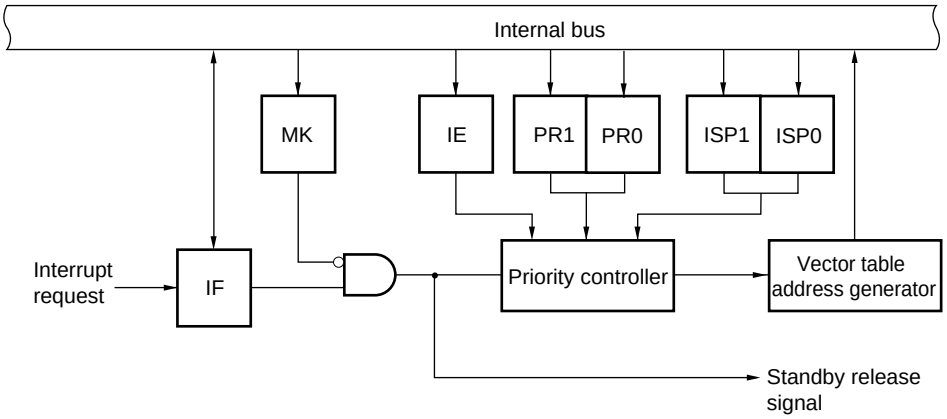
Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 10 indicates the lowest priority.

2. Basic configuration types (a) to (d) correspond to (a) to (d) in Figure 14-1.

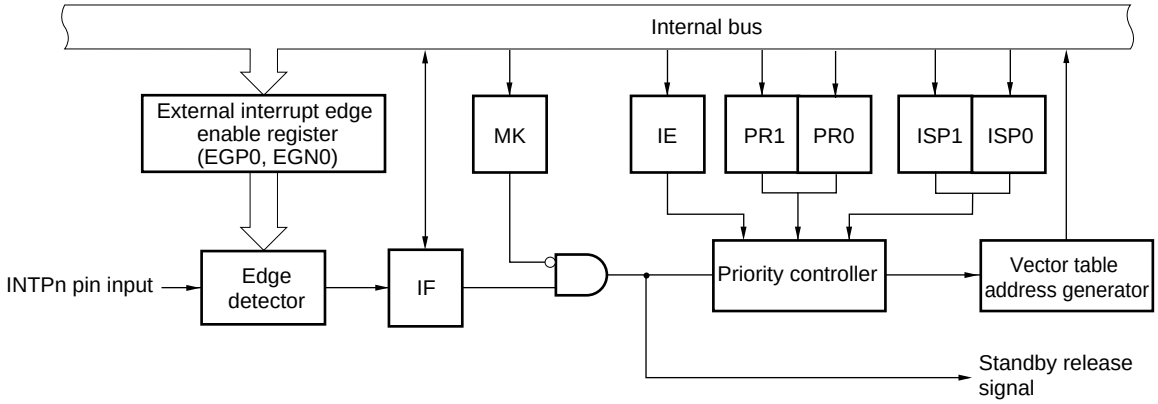
3. When the instruction code in FFH is executed.
No reset is issued even if an illegal instruction is executed during emulation with the on-chip debug emulator.

Figure 14-1. Basic Configuration of Interrupt Function (1/2)

(a) Internal maskable interrupt



(b) External maskable interrupt (INTPn)

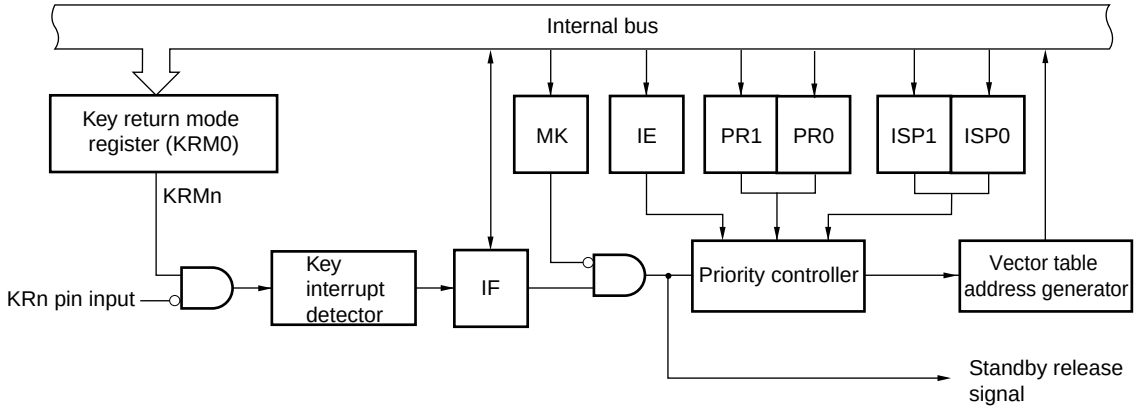


IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Remark 10-pin product : n = 0, 1
16-pin product : n = 0 to 3

Figure 14-1. Basic Configuration of Interrupt Function (2/2)

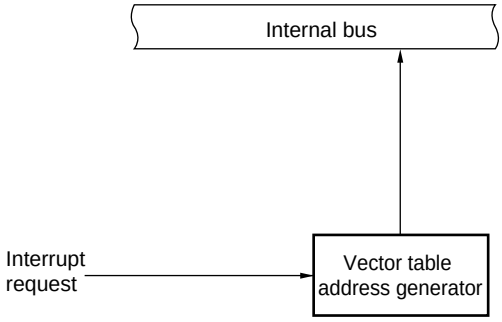
(c) External maskable interrupt (INTKR)



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Remark n = 0 to 5

(d) Software interrupt



14.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L)
- Interrupt mask flag registers (MK0L, MK0H, MK1L)
- Priority specification flag registers (PR00L, PR00H, PR10L, PR10H, PR01L, PR11L)
- External interrupt rising edge enable register 0 (EGP0)
- External interrupt falling edge enable register 0 (EGN0)
- Program status word (PSW)

Tables 14-3 and 14-4 show a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 14-3. Flags Corresponding to Interrupt Request Sources (10-pin products)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L
INTP0	PIF0		PMK0		PPR00, PPR10	
INTP1	PIF1		PMK1		PPR01, PPR11	
INTST0[Note] INTCSI00[Note] INTIIC00[Note]	STIF0[Note] CSIIF00[Note] IICIF00[Note]		STMK0[Note] CSIMK00[Note] IICMK00[Note]		STPR00, STPR10[Note] CSIPR000, CSIPR100[Note] IICPR000, IICPR100[Note]	
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	
INTTM01H	TMIF01H		TMMK01H		TMPR001H, TMPR101H	
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100	
INTTM01	TMIF01	IF0H	TMMK01	MK0H	TMPR001, TMPR101	PR00H, PR10H
INTAD	ADIF		ADMK		ADPR0, ADPR1	
INTKR	KRIF		KRMK		KRPR0, KRPR1	

Note If interrupt source INTST0, INTCSI00, or INTIIC00 occurs, bit 3 of the IF0L register is set to 1.
Bit 3 of the MK0L, PR00L, and PR10L registers supports these three interrupt sources.

Table 14-4. Flags Corresponding to Interrupt Request Sources (16-pin products)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0 WDTIPR1	PR00L, PR10L
INTP0	PIF0		PMK0		PPR00 PPR10	
INTP1	PIF1		PMK1		PPR01 PPR11	
INTST0 Note 1 INTCSI00 Note 1 INTIIC00 Note 1	STIF0 Note 1 CSIIF00 Note 1 IICIF00 Note 1		STMK0 Note 1 CSIMK00 Note 1 IICMK00 Note 1		STPR00 STPR10 Note 1 CSIPR000 CSIPR100 Note 1 IICPR000 IICPR100 Note 1	
INTSR0 Note 2 INTCSI01 Note 2	SRIF0 Note 2 CSIIF01 Note 2		SRMK0 Note 2 CSIMK01 Note 2		SRPR00 SRPR10 Note 2 CSIPR001 CSIPR101 Note 2	
INTSRE0	SREIF0		SREMK0		SREPR00 SREPR10	
INTTM01H	TMIF01H		TMMK01H		TMPR001H TMPR101H	
INTTM00	TMIF00		TMMK00		TMPR000 TMPR100	
INTTM01	TMIF01	IF0H	TMMK01	MK0H	TMPR001 TMPR101	PR00H, PR10H
INTAD	ADIF		ADMK		ADPR0 ADPR1	
INTKR	KRIF		KRMK		KRPR0 KRPR1	
INTP2	PIF2		PMK2		PPR02 PPR12	
INTP3	PIF3		PMK3		PPR03 PPR13	
INTTM03H	TMIF03H		TMMK03H		TMPR003H TMPR103H	
INTIICA0	IICAIF0		IICAMK0		IICAPR00 IICAPR10	
INTTM02	TMIF02		TMMK02		TMPR002 TMPR102	
INTTM03	TMIF03	IF1L	TMMK03	MK1L	TMPR003 TMPR103	PR01L, PR11L
INTIT	ITIF		ITMK		ITPR0 ITPR1	
INTCMP0	CMPIF0		CMPMK0		CMPPR00 CMPPR10	

Notes 1. If interrupt source INTST0, INTCSI00, or INTIIC00 occurs, bit 3 of the IF0L register is set to 1.
Bit 3 of the MK0L, PR00L, and PR10L registers supports these three interrupt sources.

2. If interrupt source INTSR0 or INTCSI01 occurs, bit 4 of the IF0L register is set to 1.
Bit 4 of the MK0L, PR00L, and PR10L registers supports these two interrupt sources.

14.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when the interrupt request is acknowledged, a reset signal is generated, or an instruction is executed.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

IF0L, IF0H, and IF1L registers can be set by a 1-bit or 8-bit memory manipulation instruction. Reset signal generation clears these registers to 00H.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 14-2. Format of Interrupt Request Flag Registers (IF0L, IF0H) (10-pin product)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	TMIF00	TMIF01H	SREIF0	SRIF0	STIF0 CSIIF00 IICIF00	PIF1	PIF0	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
IF0H	0	0	0	0	0	KRIF	ADIF	TMIF01

XXIFXX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Figure 14-3. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L) (16-pin product)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	TMIF00	TMIF01H	SREIF0	SRIF0 CSIIF01	STIF0 CSIIF00 IICIF00	PIF1	PIF0	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	TMIF02	IICAIF0	TMIF03H	PIF3	PIF2	KRIF	ADIF	TMIF01

Address: FFFE2H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
IF1L	0	0	0	0	0	CMPIF0	ITIF	TMIF03

XXIFXX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

(**Cautions** are listed on the next page.)

Cautions 1. Do not change undefined bit data.

2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as IF0L.0 = 0; or _asm("clr1 IF0L.0"); because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1). If a program is described in C language such as IF0L & = 0xfe; and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of the another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between mov a, IF0L and mov IF0L, a, the flag is cleared to 0 at mov IF0L, a. Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

14.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

The MK0L, MK0H, and MK1L registers can be set by a 1-bit or 8-bit memory manipulation instruction. Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 14-4. Format of Interrupt Mask Flag Registers (MK0L, MK0H) (10-pin product)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	TMMK00	TMMK01H	SREMK0	SRMK0	STMK0 CSIMK00 IICMK00	PMK1	PMK0	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
MK0H	1	1	1	1	1	KRMK	ADMK	TMMK01

XXMKXX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Figure 14-5. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L) (16-pin product)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	TMMK00	TMMK01H	SREMK0	SRMK0 CSIMK01	STMK0 CSIMK00 IICMK00	PMK1	PMK0	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	TMMK02	IICAMK0	TMMK03H	PMK3	PMK2	KRMK	ADMK	TMMK01

Address: FFFE6H After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
MK1L	1	1	1	1	1	CMPMK0	ITMK	TMMK03

XXMKXX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Do not change undefined bit data.

14.3.3 Priority specification flag registers (PR00L, PR00H, PR10L, PR10H, PR01L, PR11L)

The priority specification flag registers are used to set the priority level of the corresponding maskable interrupt.

A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L).

The PR00L, PR00H, PR10L, PR10H, PR01L, and PR11L registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 14-6. Format of Priority Specification Flag Registers (PR00L, PR00H, PR10L, PR10H) (10-pin product)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	TMPR000	TMPR001H	SREPR00	SRPR00	STPR00 CSIPR000 IICPR000	PPR01	PPR00	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	TMPR100	TMPR101H	SREPR10	SRPR10	STPR10 CSIPR100 IICPR100	PPR11	PPR10	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR00H	1	1	1	1	1	KRPR0	ADPR0	TMPR001

Address: FFFEDH After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR10H	1	1	1	1	1	KRPR1	ADPR1	TMPR101

XXPR1X	XXPR0X	Priority Level Selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	1	Specifying level 3 (low priority)

Caution Do not change undefined bit data.

Figure 14-7. Format of Priority Specification Flag Registers (PR00L, PR00H, PR10L, PR10H, PR01L, PR11L) (16-pin product)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	TMPR000	TMPR001H	SREPR00	SRPR00 CSIPR001	STPR00 CSIPR000 IICPR000	PPR01	PPR00	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	TMPR100	TMPR101H	SREPR10	SRPR10 CSIPR101	STPR10 CSIPR100 IICPR100	PPR11	PPR10	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	TMPR002	IICAPR00	TMPR003H	PPR03	PPR02	KRPR0	ADPR0	TMPR001

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	TMPR102	IICAPR10	TMPR103H	PPR13	PPR12	KRPR1	ADPR1	TMPR101

Address: FFFEAH After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR01L	1	1	1	1	1	CMPPR00	ITPR0	TMPR003

Address: FFFEEH After reset: FFH R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR11L	1	1	1	1	1	CMPPR10	ITPR1	TMPR103

XXPR1X	XXPR0X	Priority Level Selection
0	0	Specifying level 0 (high priority)
0	1	Specifying level 1
1	0	Specifying level 2
1	1	Specifying level 3 (low priority)

Caution Do not change undefined bit data.

14.3.4 External interrupt rising edge enable register 0 (EGP0), external interrupt falling edge enable register 0 (EGN0)

These registers specify the valid edge for INTP0, INTP1, INTP2, and INTP3.

The EGP0 and EGN0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 14-8. Format of External Interrupt Rising Edge Enable Register 0 (EGP0) and External Interrupt Falling Edge Enable Register 0 (EGN0)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	0	0	0	EGP3 Note	EGP2 Note	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	0	0	0	EGN3 Note	EGN2 Note	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 3)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Note 16-pin products only.

Caution **When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge. When switching the input port pins to the output mode, set the port mode register (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).**

Remarks **1.** For the edge detection port, see **2.1 Port Functions**.

2. n = 0 to 3

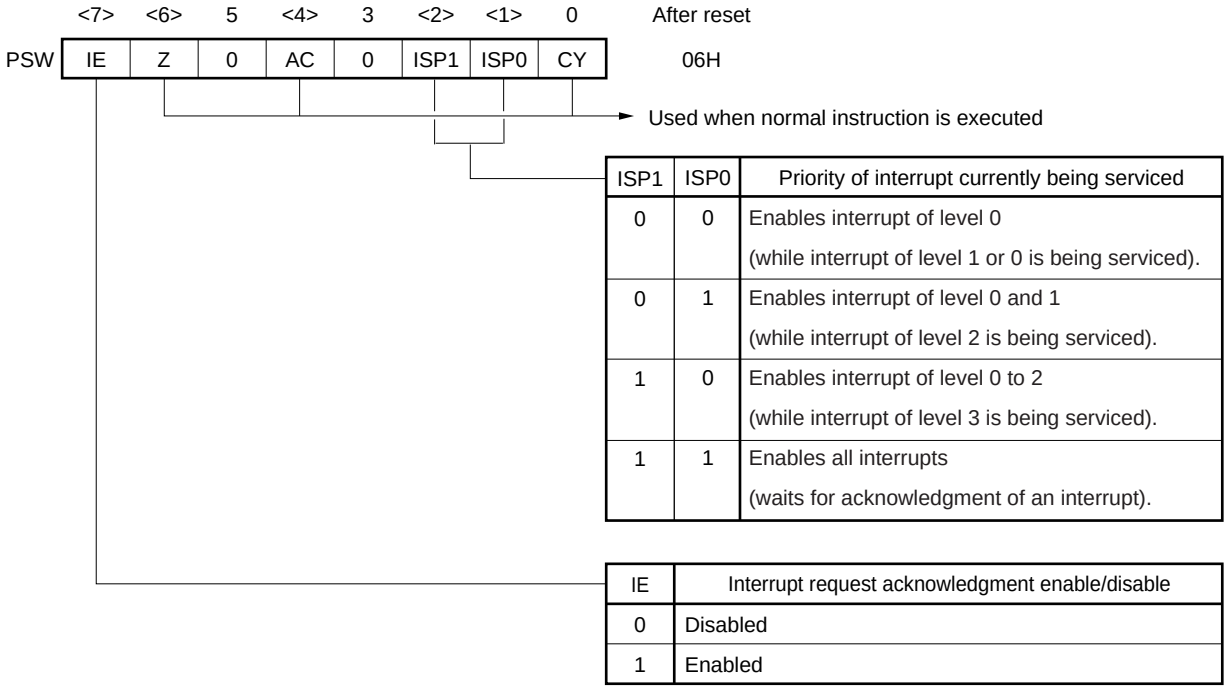
14.3.5 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. Upon acknowledgment of a maskable interrupt request, if the value of the priority specification flag register of the acknowledged interrupt is not 00, its value minus 1 is transferred to the ISP0 and ISP1 flags. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 14-9. Configuration of Program Status Word



14.4 Interrupt Servicing Operations

14.4.1 Maskable interrupt request acknowledgment

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt servicing is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority vectored interrupt request is not acknowledged during servicing of a higher priority interrupt request.

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 14-5 below.

For the interrupt request acknowledgment timing, see **Figures 14-11** and **14-12**.

Table 14-5. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	11 clocks	18 clocks

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

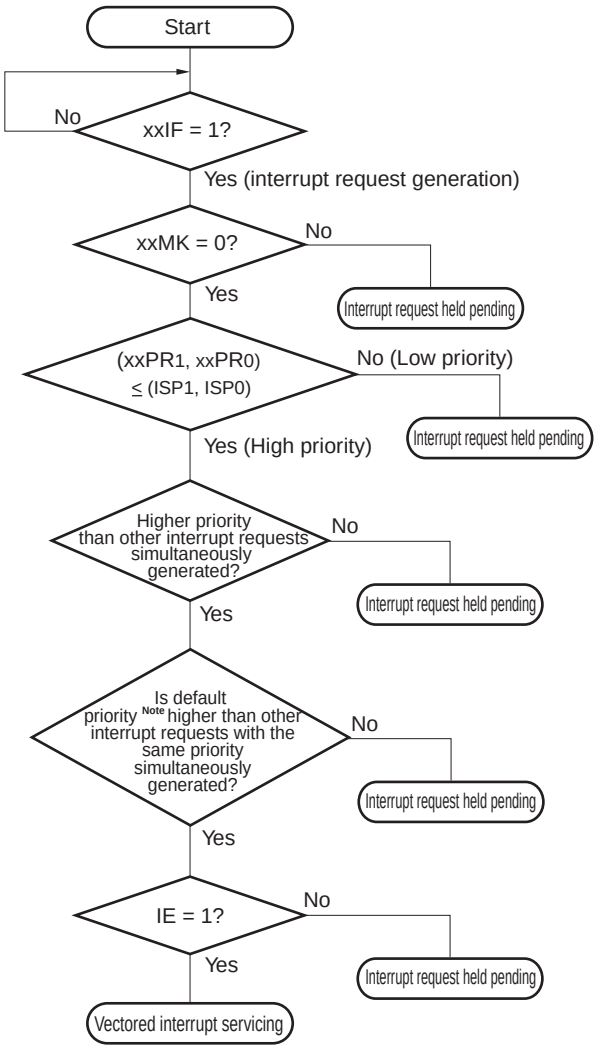
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 14-10 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

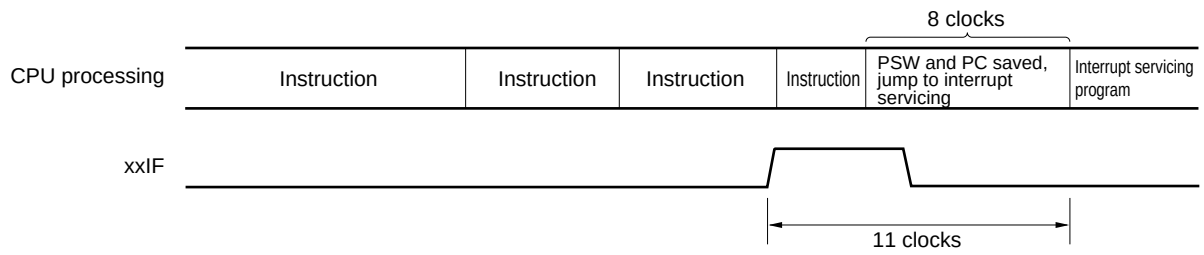
Figure 14-10. Interrupt Request Acknowledgment Processing Algorithm



××IF:	Interrupt request flag
××MK:	Interrupt mask flag
××PR0:	Priority specification flag 0
××PR1:	Priority specification flag 1
IE:	Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1:	Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 14-9**)

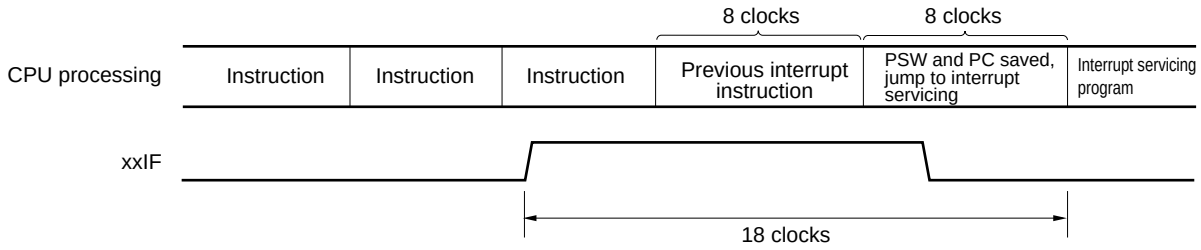
Note For the default priority, refer to **Table 14-1 Interrupt Source List (10-pin Products)**.

Figure 14-11. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 14-12. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

14.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution Can not use the RETI instruction for restoring from the software interrupt.

14.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently
<R> being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority equal to or lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. However, when setting the IE flag to 1 during the interruption at level 0, other level 0 interruptions can be allowed. Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 14-6 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 14-13 shows multiple interrupt servicing examples.

Table 14-6. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Interrupt Being Serviced \ Multiple Interrupt Request		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0 ISP0 = 0	○	×	×	×	×	×	×	×	○
	ISP1 = 0 ISP0 = 1	○	×	○	×	×	×	×	×	○
	ISP1 = 1 ISP0 = 0	○	×	○	×	○	×	×	×	○
	ISP1 = 1 ISP0 = 1	○	×	○	×	○	×	○	×	○
Software interrupt		○	×	○	×	○	×	○	×	○

Remarks 1. ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP0, ISP1, and IE are flags contained in the PSW.

ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.

ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.

ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.

ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment (enable all interrupts).

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in the PR00L, PR00H, PR10L, PR10H, PR01L, and PR11L registers.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)

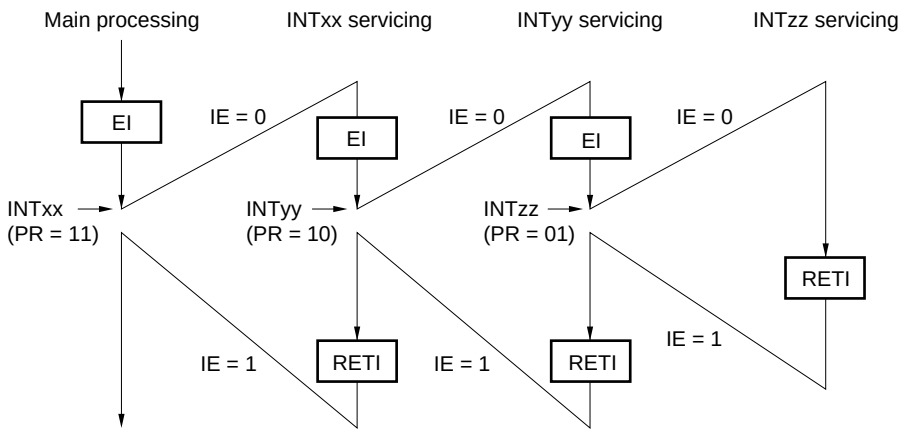
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

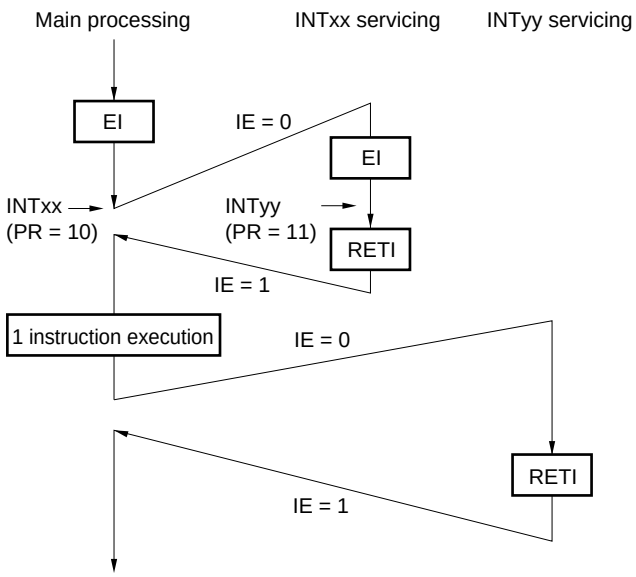
Figure 14-13. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

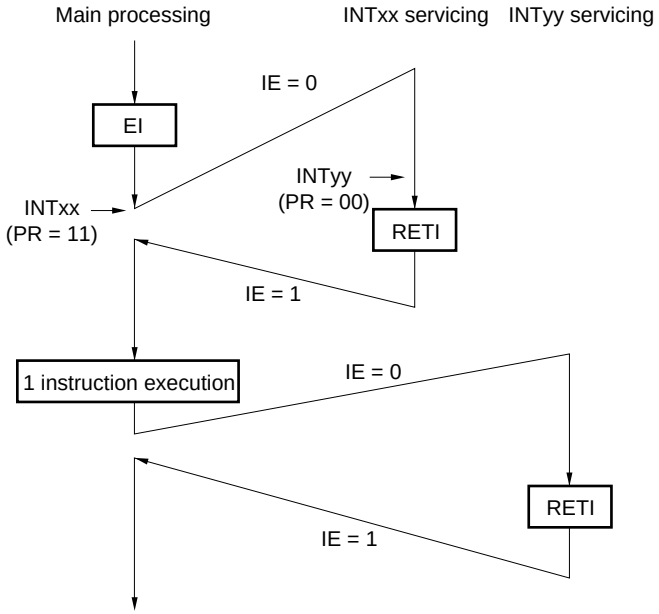


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 14-13. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

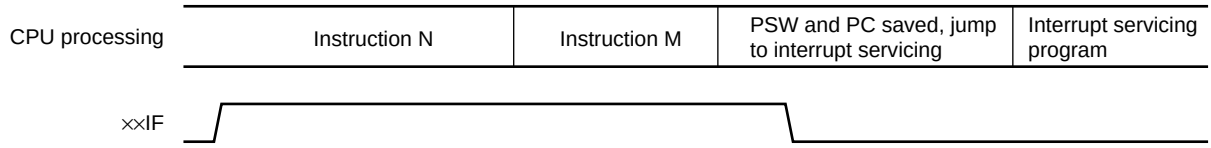
14.4.4 Interrupt request hold

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- Instructions that write data for the IF0L, IF0H, IF1L, MK0L, MK0H, MK1L, PR00L, PR00H, PR10L, PR10H, PR01L, and PR11L registers

Figure 14-14 shows the timing at which interrupt requests are held pending.

Figure 14-14. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 15 KEY INTERRUPT FUNCTION

15.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by setting the key return mode register (KRM) and inputting a rising edge/falling edge to the key interrupt input pins (KR0 to KR5).

Table 15-1. Assignment of Key Interrupt Detection Pins

Key Interrupt Pins	Key return mode registers (KRM0)	Key return flag register (KRF)
KR0	KRM00	KRF0
KR1	KRM01	KRF1
KR2	KRM02	KRF2
KR3	KRM03	KRF3
KR4	KRM04	KRF4
KR5	KRM05	KRF5

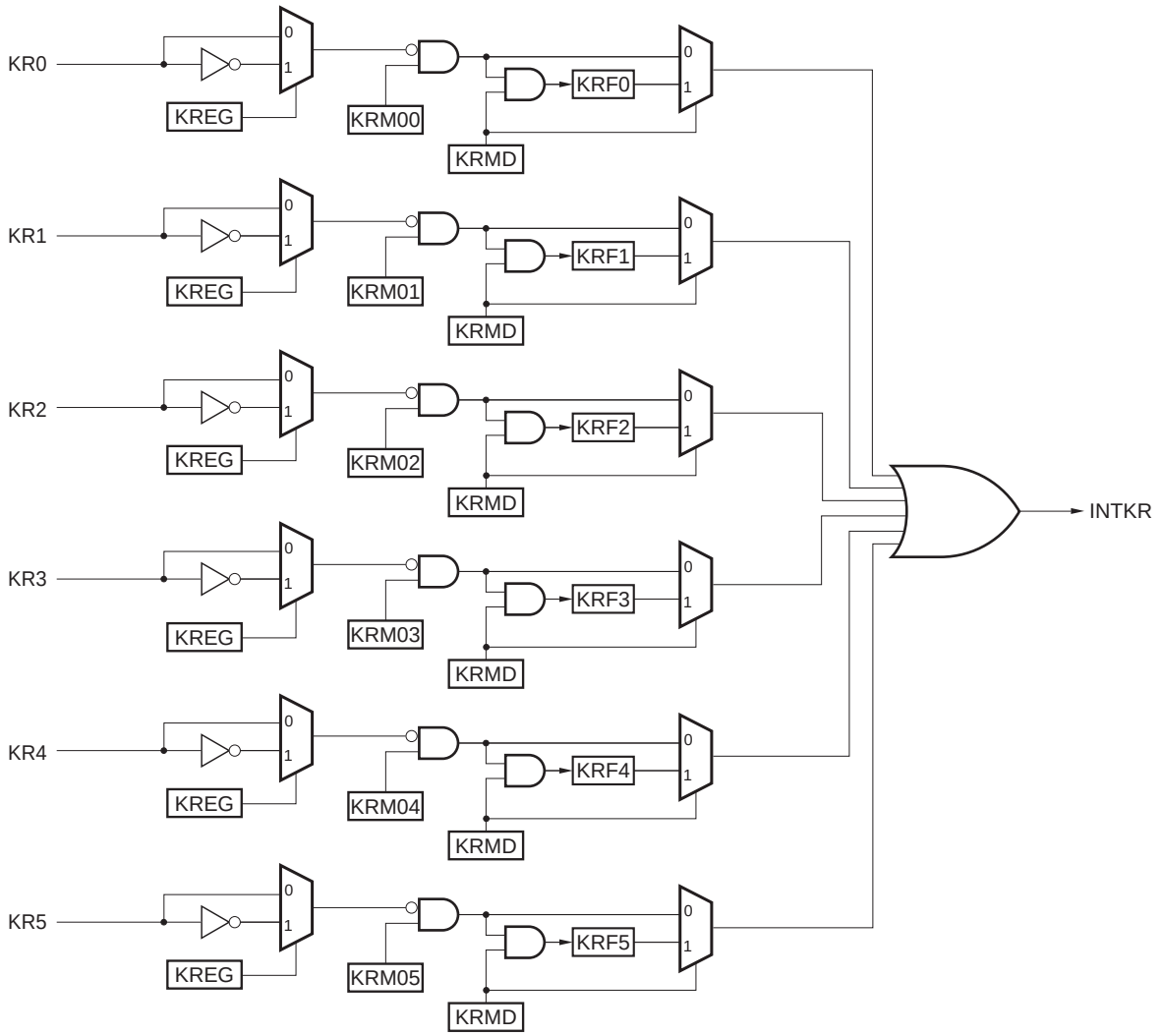
15.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 15-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return control register (KRCTL) Key return mode register (KRM0) Key return flag register (KRF) Port mode control register 0 (PMC0) Port mode registers 0, 4 (PM0, PM4)

Figure 15-1. Block Diagram of Key Interrupt



15.3 Register Controlling Key Interrupt

The key interrupt function is controlled by the following five registers:

- Key return control register (KRCTL)
- Key return mode register (KRM0)
- Key return flag register (KRF)
- Port mode control register 0 (PMC0)
- Port mode registers 0, 4 (PM0, PM4)

15.3.1 Key interrupt control register (KRCTL)

This register controls the usage of the key interrupt flags (KRF0 to KRF5) and sets the detection edge.
The KRCTL register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 15-2. Format of Key Return Control Register (KRCTL)

Address: FFF34H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRCTL	KRMD	0	0	0	0	0	0	KREG

KRMD	Usage of Key Interrupt Flags (KRF0 to KRF5)
0	Does not use key interrupt flags
1	Uses key interrupt flags

KREG	Selection of Detection Edge (KR0 to KR5)
0	Falling edge
1	Rising edge

KRMD	KREG	Interrupt function
0	0	Key interrupt, external interrupt (specified by the port level) Note
0	1	External interrupt (specified by the port level)
1	0	External interrupt (specified by the flag)
1	1	

Note When the falling edge is detected, the function and operation of the external interrupt are the same as those of the key interrupt.

15.3.2 Key return mode register (KRM0)

This register sets the key interrupt mode.

The KRM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 15-3. Format of Key Return Mode Register (KRM0)

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM0	0	0	KRM05	KRM04	KRM03	KRM02	KRM01	KRM00

Cautions 1. When a key interrupt signal is detected (KRM0n = 1) by selecting the falling edge (KRMD = 0), pull up the relevant input pins to V_{DD} by an external resistor. For the KR1, KR6 to KR9 pins, the internal pull-up resistor can be used by setting the relevant bits to 1 in the key interrupt input pins PU01 to PU04, PU40, and PU125 (Pull-up resistor option registers 0, 4, 12 (PU0, PU4, PU12)).

2. An interrupt will be generated if the target bit of the KRM0 register is set while a low level (when KREG = 0)/high level (when KREG = 1) is being input to the key interrupt input pin. To ignore this interrupt, set the KRM0 register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing.

3. The bits not used in the key interrupt mode can be used as normal ports.

15.3.3 Key return flag register (KRF)

This register controls the key interrupt flags (KRF0 to KRF5).

The KRF register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-4. Format of Key Return Flag Register (KRF)

Address: FFF35H After reset: 00H R/W [Note]

Symbol	7	6	5	4	3	2	1	0
KRF	0	0	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0

KRFn	Key interrupt flag
0	No key interrupt signal has been detected.
1	A key interrupt signal has been detected.

Note Writing to 1 is invalid. To clear KRFn, write “0” to the target bits and write “1” to other bits, with the 8-bit memory manipulation instruction.

Caution When the key interrupt flag is not used (KRMD = 0), accessing the KRF register is prohibited.

15.3.4 Registers controlling port functions of key interrupt input pins

Using a port pin for key interrupt input requires setting of the registers that control the port functions multiplexed on the target pin (port mode registers 0, 4 (PM0, PM4) and port mode control register 0 (PMC0)).

For details, see **4.3.1 Port mode registers 0, 4 (PM0, PM4)** and **4.3.5 Port mode control register 0 (PMC0)**.

For an example of settings when using a port for key interrupt input, see **4.5.3 Example of register settings for port and alternate functions used**.

Using a port pin with a multiplexed key interrupt input function (e.g. P01/ANI0/SI00/RXD0/SDA00/KR2)

for key interrupt input requires setting the corresponding bit in the port mode register 0 (PM0) to 1, and the corresponding bit in the pot mode control register 0 (PMC0) to 0. In this case, the corresponding bit in the port register 0 (P0) can be set to 0 or 1.

Example: When P01/ANI0/SI00/RXD0/SDA00/KR2 is to be used for key interrupt input

Set the PMC01 bit of port mode control register 0 to 0.

Set the PM01 bit of port mode register 0 to 1.

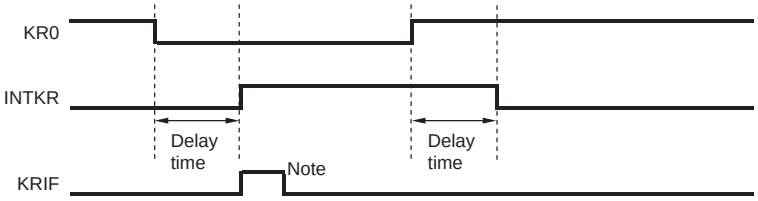
15.4 Key Interrupt Operation

15.4.1 When not using the key interrupt flag (KRMD = 0)

A key interrupt (INTKR) is generated when the valid edge specified by the setting of the KREG bit is input to a key interrupt pin (KR0 to KR5). The channel to which the valid edge was input can be identified by reading the port register and checking the port level after the key interrupt (INTKR) is generated.

The INTKR signal changes according to the input level of the key interrupt input pin (KR0 to KR5).

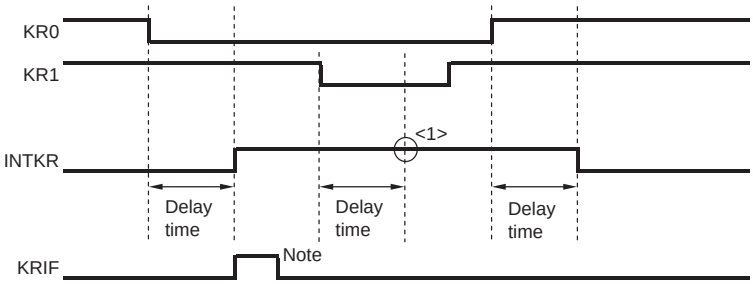
Figure 15-5. Operation of INTKR Signal When a Key Interrupt is Input to a Single Channel (When KRMD = 0 and KREG = 0)



Note Acknowledgment of vectored interrupt request or bit cleared by software

The operation when a valid edge is input to multiple key interrupt input pins is shown in Figure 15-6 below. The INTKR signal is set while a low level is being input to one pin (when KREG is set to 0). Therefore, even if a falling edge is input to another pin in this period, a key interrupt (INTKR) will not be generated again (<1> in the figure).

Figure 15-6. Operation of INTKR Signal When Key Interrupts Are Input to Multiple Channels (When KRMD = 0 and KREG = 0)



Note Acknowledgment of vectored interrupt request or bit cleared by software

15.4.2 When using the key interrupt flag (KRMD = 1)

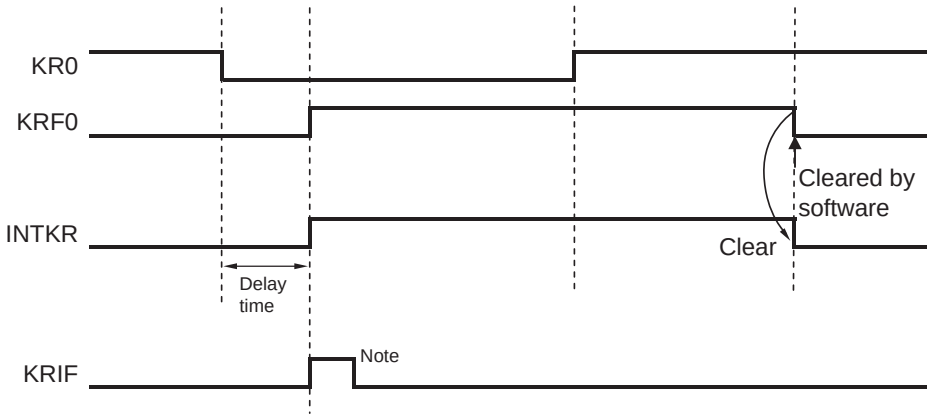
A key interrupt (INTKR) is generated when the valid edge specified by the setting of the KREG bit is input to a key interrupt pin (KR0 to KR5). The channels to which the valid edge was input can be identified by reading the key return flag register (KRF) after the key interrupt (INTKR) is generated.

If the KRMD bit is set to 1, the INTKR signal is cleared by clearing the corresponding bit in the KRF register.

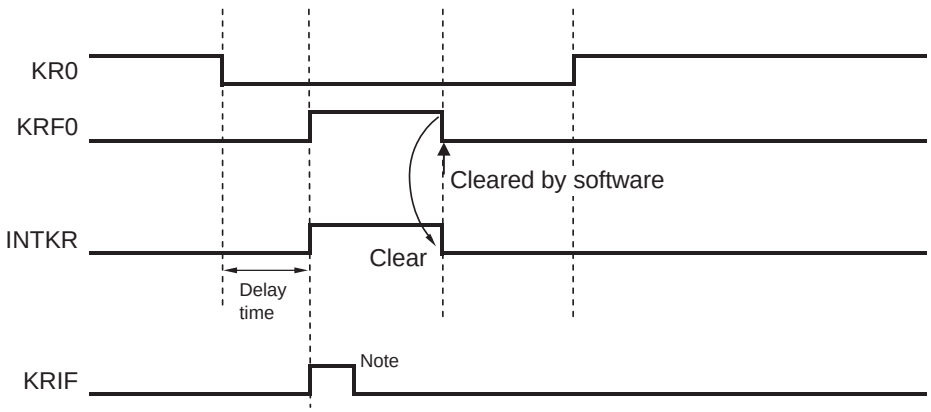
As shown in Figure 15-7, only one interrupt is generated each time a falling edge is input to one channel (when KREG = 0), regardless of whether the KRFn bit is cleared before or after a rising edge is input.

Figure 15-7. Basic Operation of the INTKR Signal When the Key Interrupt Flag Is Used (When KRMD = 1 and KREG = 0)

(a) When KRF0 is cleared after a rising edge is input to the KR0 pin



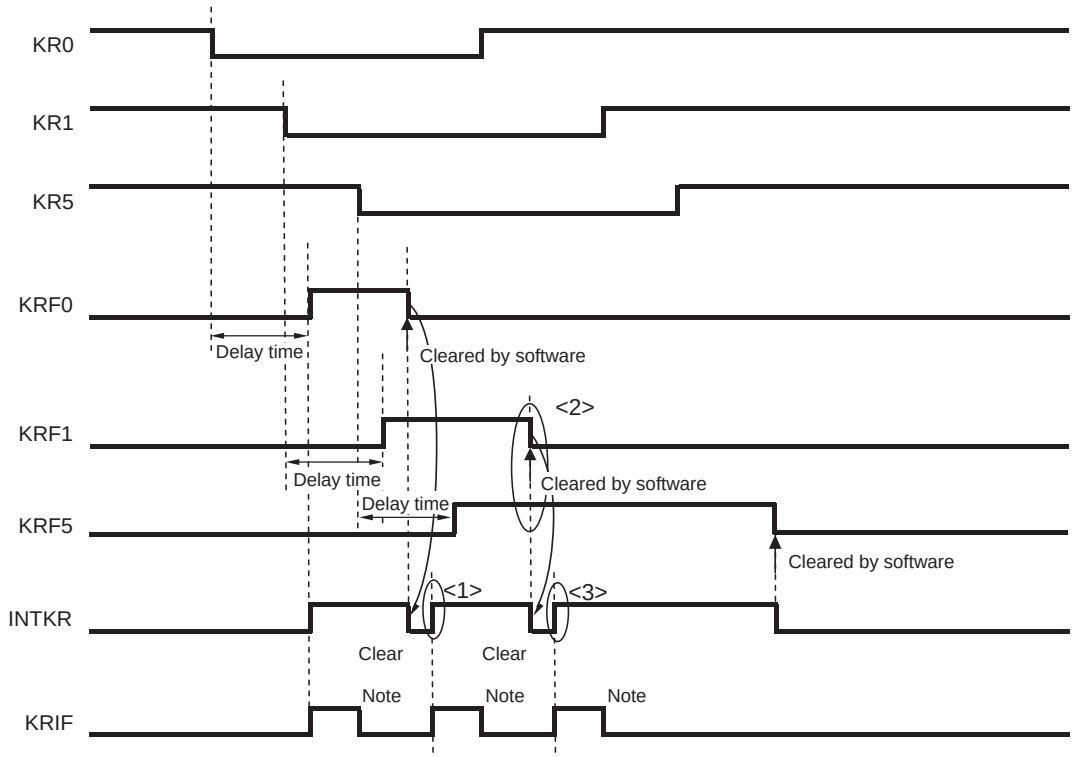
(b) When KRF0 is cleared before a rising edge is input to the KR0 pin



Note Acknowledgment of vectored interrupt request or bit cleared by software

The operation when a valid edge is input to multiple key interrupt input pins is shown in Figure 15-8 below. A falling edge is also input to the KR1 and KR5 pins after a falling edge was input to the KR0 pin (when KREG = 0). The KRF1 bit is set when the KRF0 bit is cleared. A key interrupt (INTKR) is therefore generated one clock (f_{CLK}) after the KRF0 bit is cleared (<1> in the figure). Also, after a falling edge has been input to the KR5 pin, the KRF5 bit is set (<2> in the figure) when the KRF1 bit is cleared. A key interrupt (INTKR) is therefore generated one clock (f_{CLK}) after the KRF1 bit is cleared (<3> in the figure). It is thus possible to generate a key interrupt (INTKR) when a valid edge is input to multiple channels.

Figure 15-8. Operation of INTKR Signal When Key Interrupts Are Input to Multiple Channels (When KRMD = 1 and KREG = 0)



Note Acknowledgment of vectored interrupt request or bit cleared by software

Remark f_{CLK}: CPU/peripheral hardware clock frequency

CHAPTER 16 STANDBY FUNCTION

16.1 Overview

The standby function reduces the operating current of the system, and the following three modes are available.
For details of each register, see **CHAPTER 5 CLOCK GENERATOR**.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator or high-speed on-chip oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and high-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.
Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with X1 oscillation or EXCLK input before executing STOP instruction Note.

2. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

3. It can be selected by the option byte whether the low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 19 OPTION BYTE.

Note 16-pin products only.

16.2 Registers controlling standby function

The standby function is controlled by the following registers.

For details of each register, see **CHAPTER 5 CLOCK GENERATOR**.

Register which enables or stops the operation of the low-speed on-chip oscillator in the HALT or STOP mode.

- Operation speed mode control register (OSMC)

Registers which controls oscillation stabilization time of the X1 clock when the STOP mode is released.

- Oscillation stabilization time counter status register (OSTC) **Note**
- Oscillation stabilization time select register (OSTS) **Note**

Note 16-pin products only.

16.3 Standby Function Operation

16.3.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock (16-pin products only) or the high-speed on-chip oscillator clock.

The operating statuses in the HALT mode are shown below.

Caution Because the interrupt request signal is used to clear the HALT mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the HALT mode is not entered even if the HALT instruction is executed in such a situation.

Table 16-1. Operating Statuses in HALT Mode

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Operation disabled	
		f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and bit 4 (WUTMMCK0) of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory					
RAM					
Port (latch)			Status before HALT mode was set is retained		
Timer array unit			Operable		
12-bit interval timer Note					
Watchdog timer			Set by bit 0 (WDSTBYON) of option byte (000C0H) • WDSTBYON = 0: Operation stopped • WDSTBYON = 1: Operation continues		
Clock output/buzzer output			Operable		
A/D converter					
Comparator Note					
Serial array unit (SAU)					
Serial interface IICA Note					
Selectable power-on-reset function					
External interrupt					
Key interrupt function					

Note 16-pin products only.

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
f_{IH}: High-speed on-chip oscillator clock
f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock Note
f_{EX}: External main system clock Note

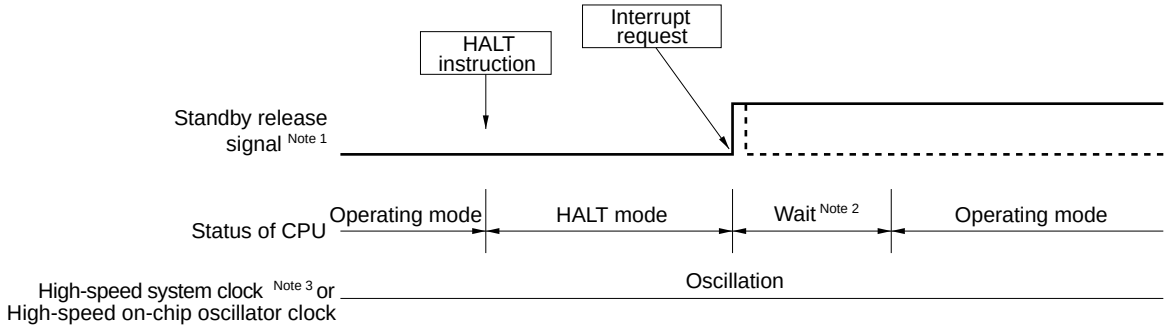
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 16-1. HALT Mode Release by Interrupt Request Generation



Notes 1. For details of the standby release signal, see **Figure 14-1 Basic Configuration of Interrupt Function**.

2. Wait time for HALT mode release: When vectored interrupt servicing is carried out: 28 to 29 clocks
When vectored interrupt servicing is not carried out: 20 to 21 clocks

3. 16-pin products only.

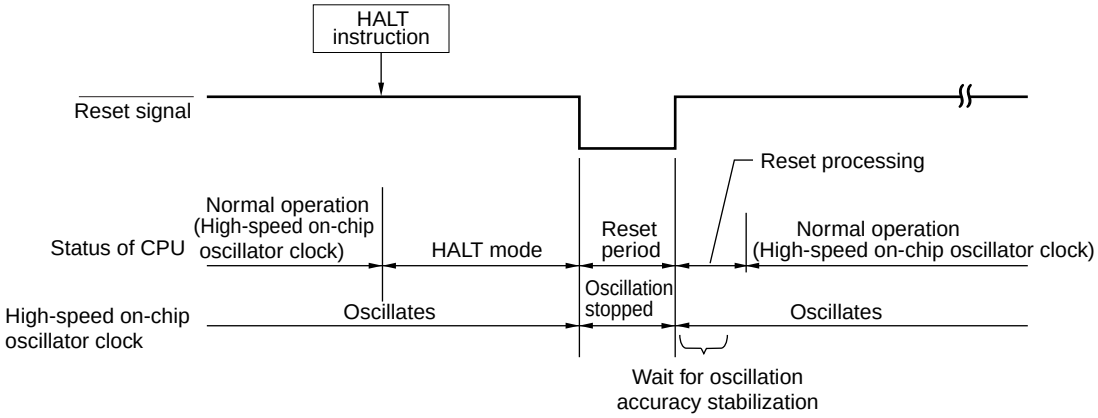
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) HALT mode release by reset signal generation

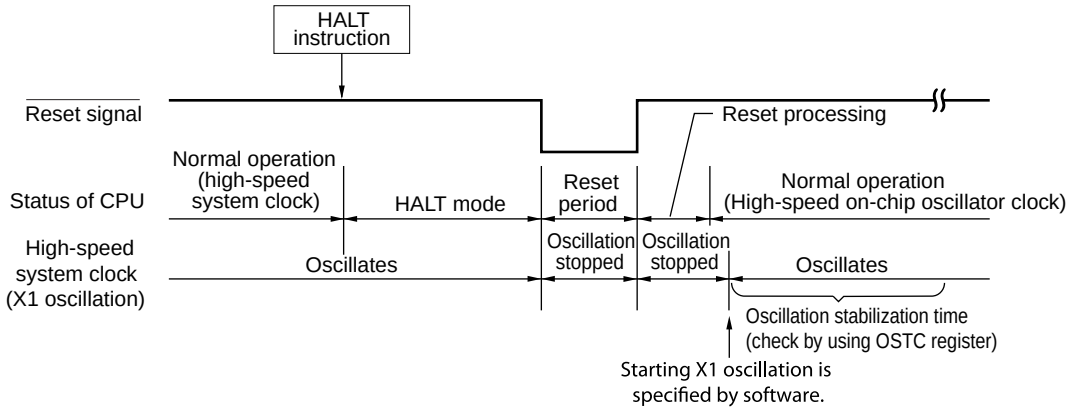
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 16-2. HALT Mode Release by Reset Signal Generation

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock (16-pin products only)



Note For the reset processing time, see **CHAPTER 17 RESET FUNCTION**.

For the reset processing time of the SPOR circuit, see **CHAPTER 18 SELECTABLE POWER-ON-RESET CIRCUIT**.

16.3.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction.

Caution Because the interrupt request signal is used to clear the STOP mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the STOP mode is immediately cleared if set when the STOP instruction is executed in such a situation. Accordingly, once the STOP instruction is executed, the system returns to its normal operating mode after the elapse of release time from the STOP mode.

The operating statuses in the STOP mode are shown below.

Table 16-2. Operating Statuses in STOP Mode

Item \ STOP Mode Setting			When STOP Instruction Is Executed While CPU Is Operating		
			When CPU Is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Stopped		
		f_X			
		f_{EX}			
	Low Speed On-chip Oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and bit 4 (WUTMMCK0) of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory					
RAM			Operation stopped		
Port (latch)			Status before STOP mode was set is retained		
Timer array unit			Operation disabled		
12-bit interval timer Note			Operable		
Watchdog timer			Set by bit 0 (WDSTBYON) of option byte (000C0H) • WDSTBYON = 0: Operation stopped • WDSTBYON = 1: Operation continues		
Clock output/buzzer output			Operation disabled		
A/D converter					
Comparator Note			Operable (only when the digital filter is not in use)		
Serial array unit (SAU)			Operation disabled		
Serial interface (IICA) Note			Wakeup by address match operable		
Selectable power-on-reset function			Operable		
External interrupt					
Key interrupt function					

Note 16-pin products only.

Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.

Operation disabled: Operation is stopped before switching to the STOP mode.

f_{IH}: High-speed on-chip oscillator clock

f_{IL}: Low-speed on-chip oscillator clock

f_X: X1 clock Note

f_{EX}: External main system clock Note

(2) STOP mode release

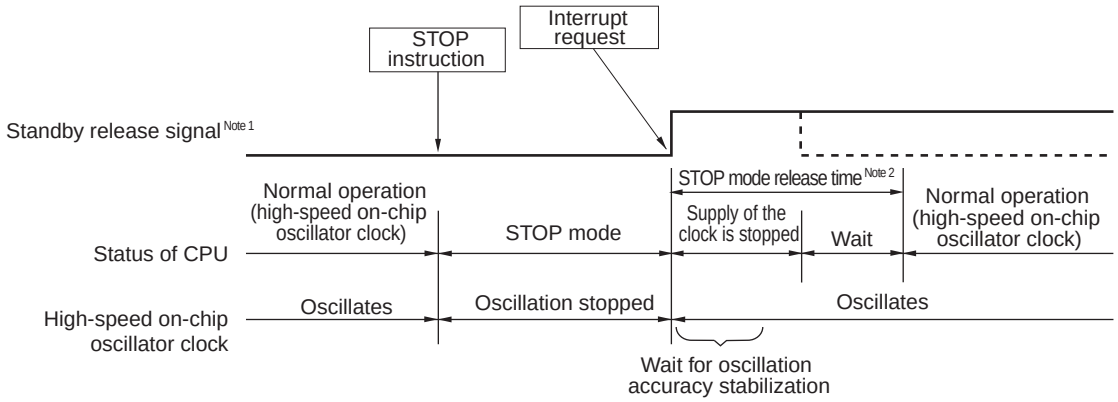
The STOP mode can be released by the following two sources.

(a) STOP mode release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 16-3. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



Notes 1. For details of the standby release signal, see **Figure 14-1 Basic Configuration of Interrupt Function**.

2. STOP mode release time: Supply of the clock is stopped: 27 μs (typ.)

Wait

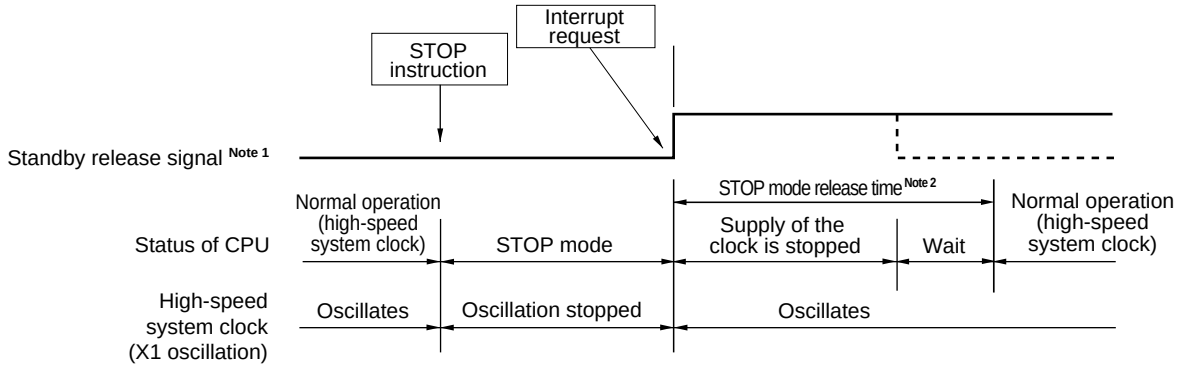
- When vectored interrupt servicing is carried out: 11 clocks
- When vectored interrupt servicing is not carried out: 3 clocks

Remarks 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 16-3. STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock (16-pin products only)



Notes 1. For details of the standby release signal, see **Figure 14-1 Basic Configuration of Interrupt Function**.

2. STOP mode release time: Whichever is longer 27 μs (typ.) or the oscillation stabilization time (set by OSTS)

Wait

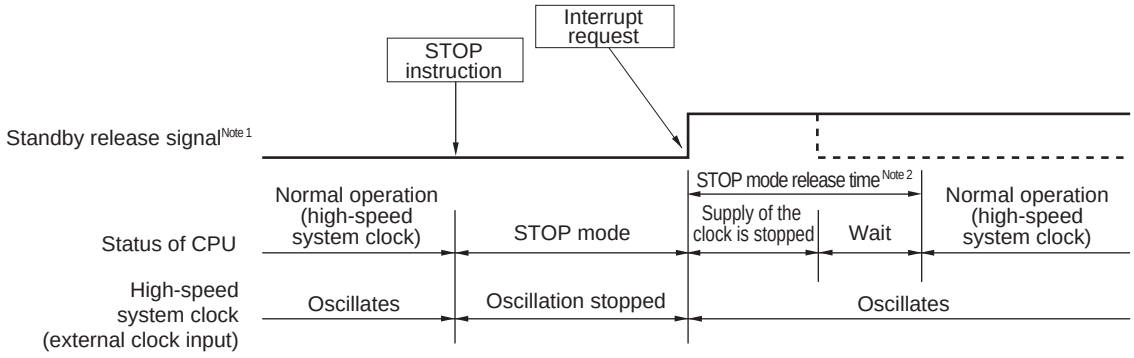
- When vectored interrupt servicing is carried out: 14 to 15 clocks
- When vectored interrupt servicing is not carried out: 6 to 7 clocks

Caution **To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.**

Remarks 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(3) When high-speed system clock (external clock input) is used as CPU clock (16-pin products only)



Notes 1. For details of the standby release signal, see **Figure 14-1 Basic Configuration of Interrupt Function**.

2. STOP mode release time: $2^4/f_{EX}$

Wait

- When vectored interrupt servicing is carried out: 11 clocks
- When vectored interrupt servicing is not carried out: 3 clocks

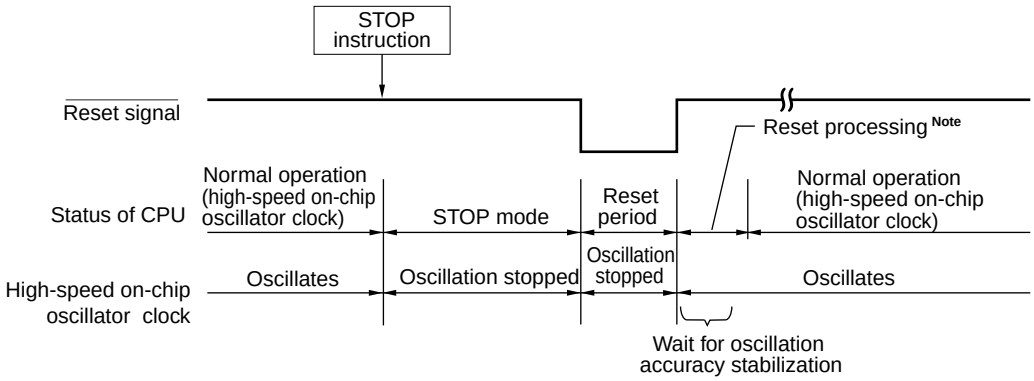
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) STOP mode release by reset signal generation

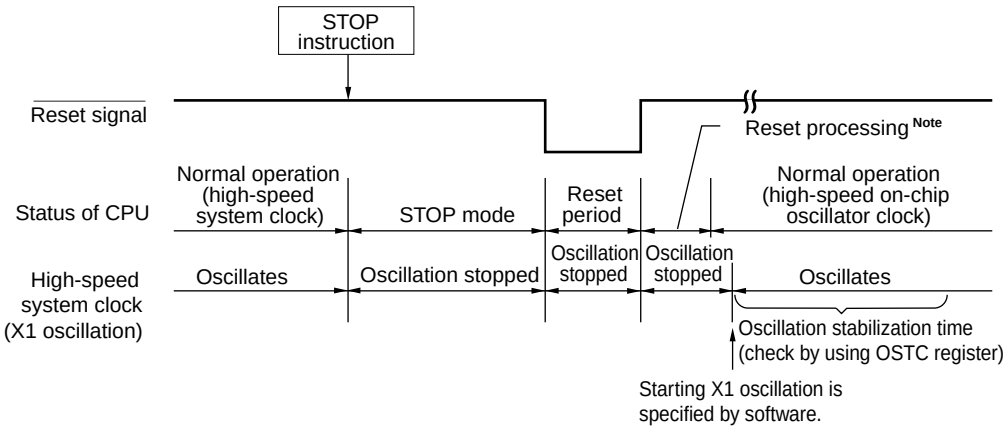
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 16-4. STOP Mode Release by Reset Signal Generation

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock (16-pin products only)



Note For the reset processing time, see **CHAPTER 17 RESET FUNCTION**.

For the reset processing time of the SPOR circuit, see **CHAPTER 18 SELECTABLE POWER-ON-RESET CIRCUIT**.

CHAPTER 17 RESET FUNCTION

The following five operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of selectable power-on-reset (SPOR) circuit
(4) Internal reset by execution of illegal instruction[Note]
(5) Internal reset by data retention power supply voltage

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.

Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the on-chip debug emulator.

Cautions 1. For an external reset, set the PORTSELB bit of the user option byte (000C1H) to 1 so that the P125 pin operates as $\overline{\text{RESET}}$, and input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
(To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, and then apply power supply. The $\overline{\text{RESET}}$ pin must be kept low for at least 10 μs during the period in which the supply voltage is within the operating range shown in 24.4 AC Characteristics before inputting a high level to the $\overline{\text{RESET}}$ pin.

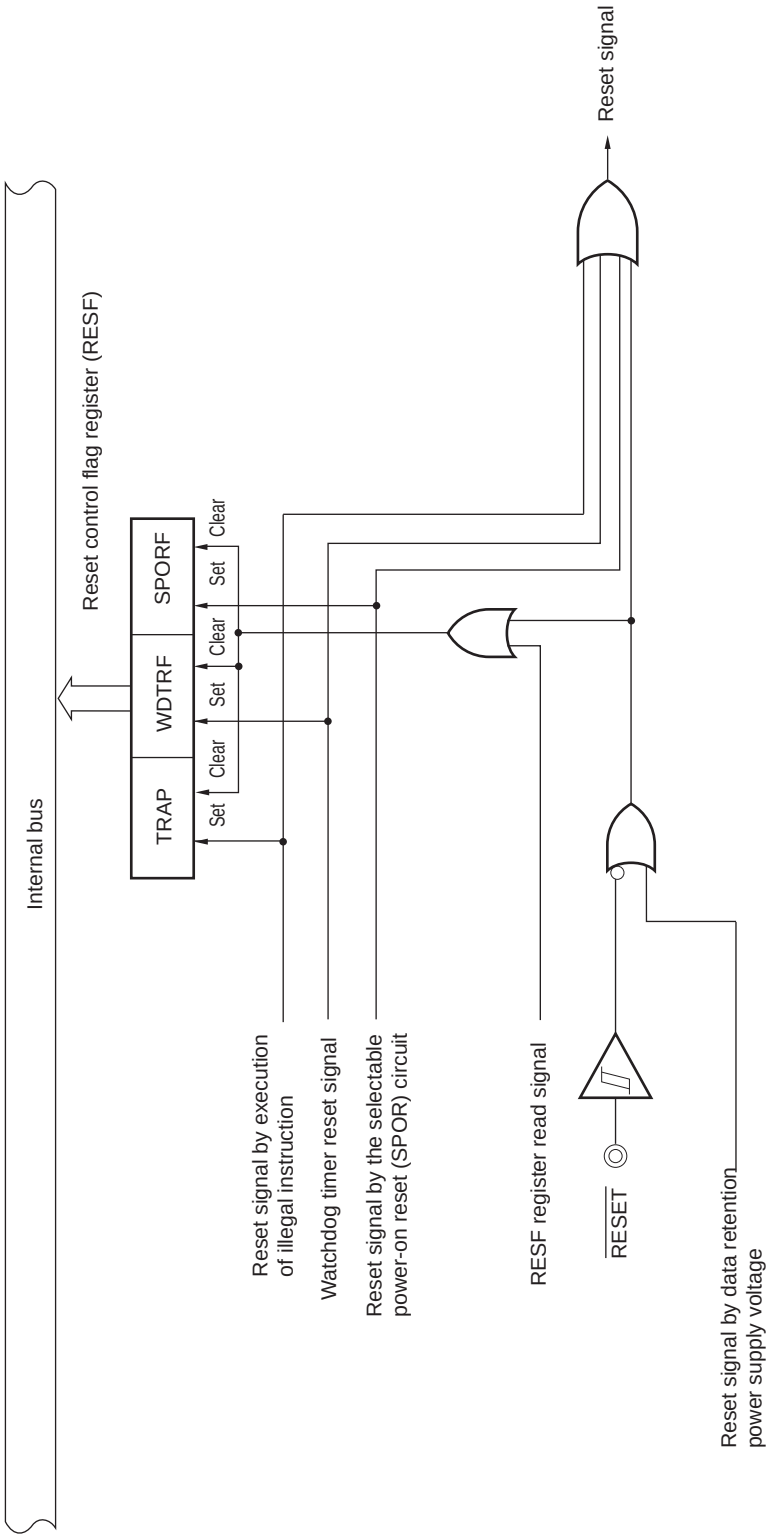
2. During reset input, the X1 clock [Note], high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating, and external main system clock [Note] input is invalid.

3. The port pin becomes the following status because each SFR and 2nd SFR are initialized after reset.

- **P40: High-impedance during external reset period or reset period by the data retention power supply voltage. High level during other types of reset or after receiving a reset (connected to the internal pull-up resistor).**
- **P125: Low level during external reset period (low level input to $\overline{\text{RESET}}$ pin). High level during other types of reset period or after receiving a reset (connected to the internal pull-up resistor).**
- **Ports other than P40 and P125: High-impedance during reset period or after receiving a reset.**

Note 16-pin products only.

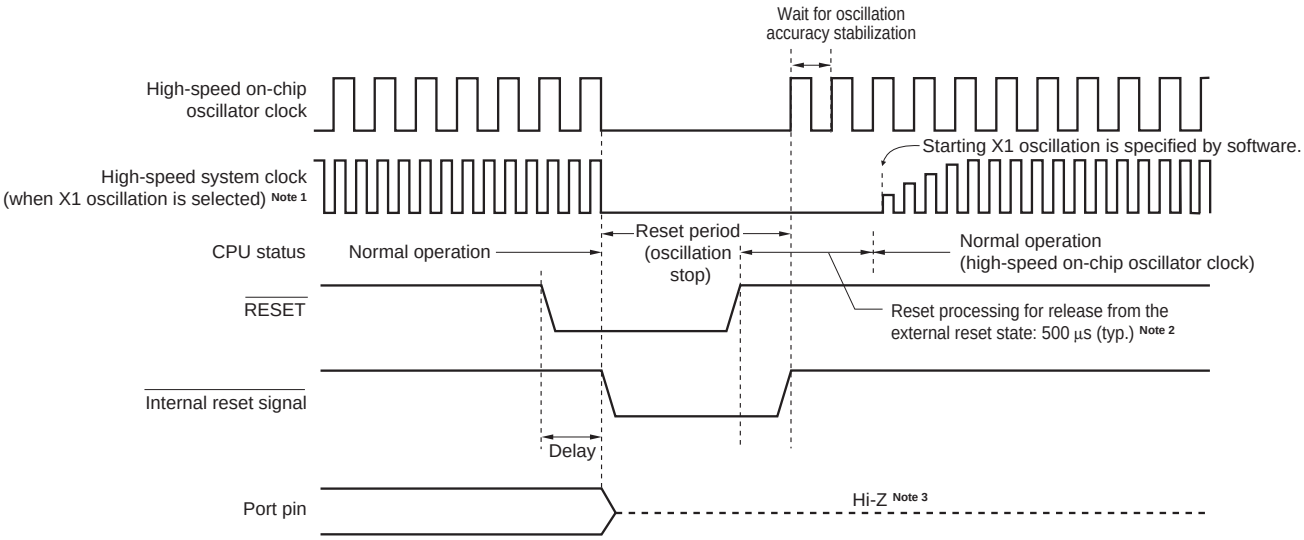
Figure 17-1. Block Diagram of Reset Function



17.1 Timing of Reset Operation

This LSI is reset by input of the low level on the $\overline{\text{RESET}}$ pin and released from the reset state by input of the high level on the $\overline{\text{RESET}}$ pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 17-2. Timing of Reset by $\overline{\text{RESET}}$ Input



Notes 1. 16-pin products only.

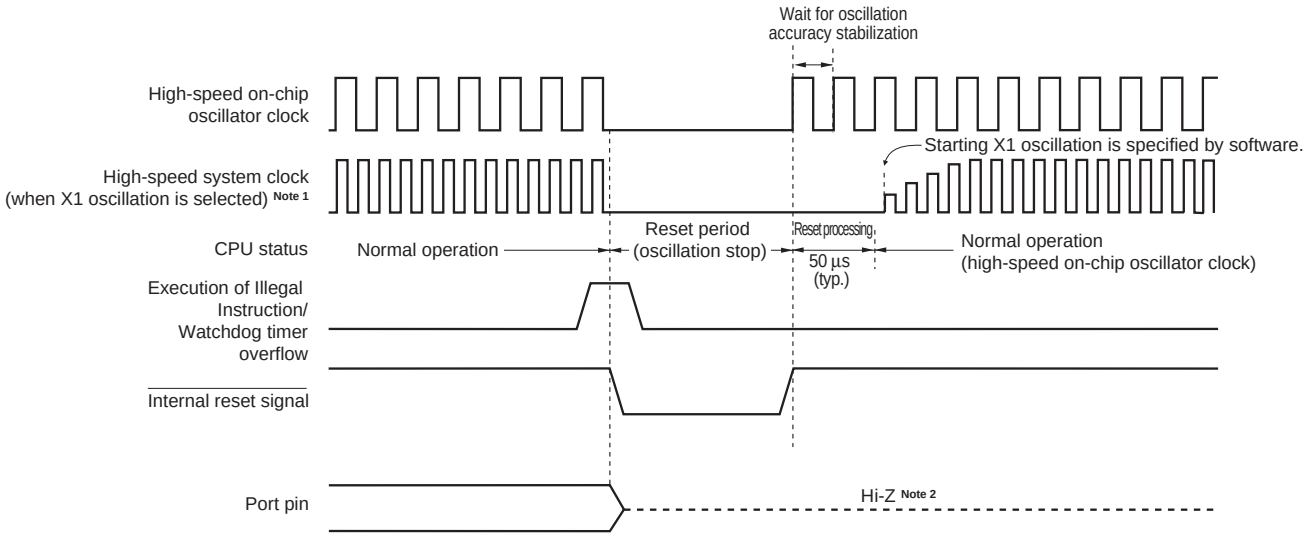
2. After power is supplied, an SPOR reset processing time of (MAX. 3.01 ms) is required before reset processing starts after release of the external reset.

3. Status of port pin P40 is as follows.

- High-impedance during external reset period or reset period by the data retention power supply voltage
- High level after receiving a reset (connected to the internal pull-up resistor)

Release from the reset state is automatic in the cases of a reset due to the watchdog timer overflow or execution of illegal instruction. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 17-3. Timing of Reset Due to Watchdog Timer Overflow or Execution of Illegal Instruction



Notes 1. 16-pin products only.

2. Statuses of port pins P40 and P125 pins are as follows.

- High level during reset period or after receiving a reset (connected to the internal pull-up resistor).

Remark For the reset timing due to the voltage detection by the selectable power-on-reset (SPOR) circuit, see **CHAPTER 18 SELECTABLE POWER-ON-RESET CIRCUIT**.

17.2 States of Operation During Reset Periods

Table 17-1 shows the states of operation during reset periods. Table 17-2 shows the state of the hardware after acceptance of a reset.

Table 17-1. States of Operation During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_X	Operation stopped (the X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (the pin is input port mode)
	f_{IL}		Operation stopped
CPU			
Code flash memory			Operation stopped
RAM			Operation stopped
Port (latch)			High impedance Note 2
Timer array unit			Operation stopped
12-bit Interval timer			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Comparator Note 1			
Serial array unit (SAU)			
Serial interface (IICA)			
Selectable power-on-reset function			Detection operation possible
External interrupt			Operation stopped
Key interrupt function			

Notes **1.** 16-pin products only.

2. Statuses of P40 and P125 pins are as follows

- P40: High-impedance during external reset period or reset period by the data retention power supply voltage. High level during other types of reset period or after receiving a reset (connected to the internal pull-up resistor).
- P125: Low level during external reset period (low level input to $\overline{\text{RESET}}$ pin). High level during other types of reset period or after receiving a reset (connected to the internal pull-up resistor).

Remark f_{IH}: High-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock
f_{IL}: Low-speed on-chip oscillator clock

Table 17-2. State of Hardware After Acceptance of Reset

Hardware		After Acceptance of Reset Note 1
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined
	General-purpose registers	Undefined

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Remark For the state of the special function register (SFR: Special Function Register) after receiving a reset, see **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

17.3 Register for Confirming Reset Source

17.3.1 Reset Control Flag Register (RESF)

Many internal reset generation sources exist in the RL78 microcontroller. The reset control flag register (RESF) is used to store which source has generated the reset request.

The RESF register can be read by an 8-bit memory manipulation instruction.

The external reset, a reset by the data retention lower limit voltage, and reading the RESF register clear TRAP, WDTRF, and SPORF flags.

Figure 17-4. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: Undefined [Note1] R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	0	0	SPORF

TRAP	Internal reset request by execution of illegal instruction[Note 2]
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

SPORF	Internal reset request by selectable power-on reset (SPOR) circuit
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

Notes 1. The value after reset varies depending on the reset source.

2. The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the on-chip debug emulator.

Caution Do not read data by a 1-bit memory manipulation instruction.

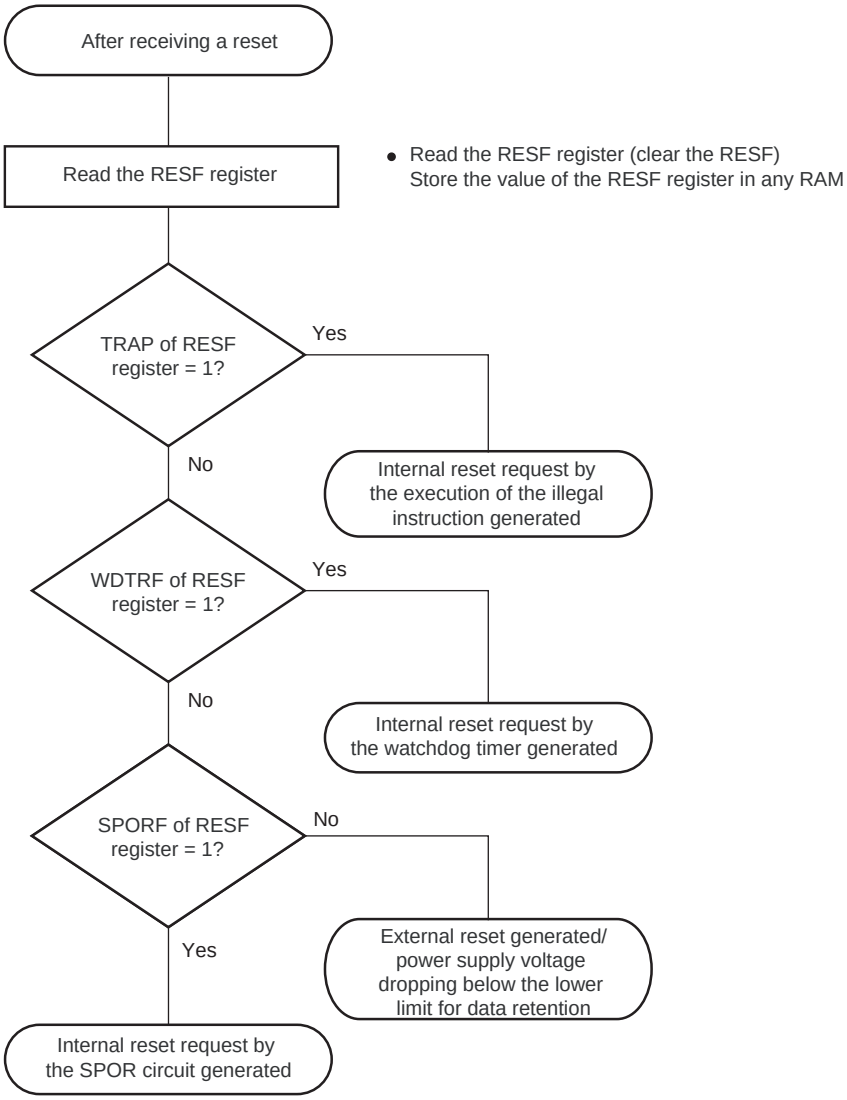
The status of the RESF register when a reset request is generated is shown in Table 17-3.

Table 17-3. RESF Register Status When Reset Request Is Generated

Flag \ Reset Source	$\overline{\text{RESET}}$ Input	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by SPOR	Reset by data retention lower limit voltage
TRAP bit	Cleared (0)	Set (1)	Held	Held	Cleared (0)
WDTRF bit		Held	Set (1)	Held	
SPORF bit		Held	Held	Set (1)	

The RESF register is automatically cleared when it is read by an 8-bit memory manipulation instruction. Figure 17-5 shows the example of the procedure for checking the reset source.

Figure 17-5. Example of Procedure for Checking Reset Source



The flow described above is an example of the procedure for checking.

CHAPTER 18 SELECTABLE POWER-ON-RESET CIRCUIT

18.1 Functions of Selectable Power-on-reset Circuit

The selectable power-on-reset (SPOR) circuit has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage exceeds the detection voltage ($V_{DD} \geq V_{SPOR}$).
- The SPOR circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{SPDR}), and generates an internal reset signal when $V_{DD} < V_{SPDR}$.
- The detection level for the power supply detection voltage (V_{SPOR}, V_{SPDR}) can be selected by using the option byte (000C1H) as one of 4 levels (for details, see **19.2 Format of User Option Byte**).

Bit 0 (SPORF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 17 RESET FUNCTION**.

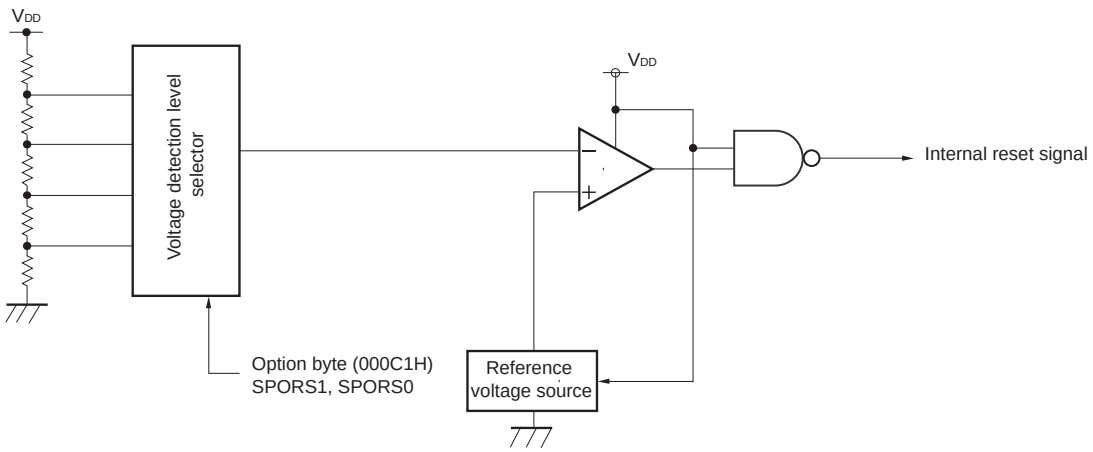
Note The values of all flags in the reset control flag register (RESF) are retained until V_{DD} reaches data retention lower limit voltage.

Remark V_{SPOR}: SPOR power supply rise detection voltage
V_{SPDR}: SPOR power supply fall detection voltage
For details, see **24.6.4 SPOR circuit characteristics**.

18.2 Configuration of Selectable Power-on-reset Circuit

The block diagram of the selectable power-on-reset circuit is shown in Figure 18-1.

Figure 18-1. Block Diagram of Selectable Power-on-reset Circuit



18.3 Operation of Selectable Power-on-reset Circuit

Specify the voltage detection level by using the option byte 000C1H.

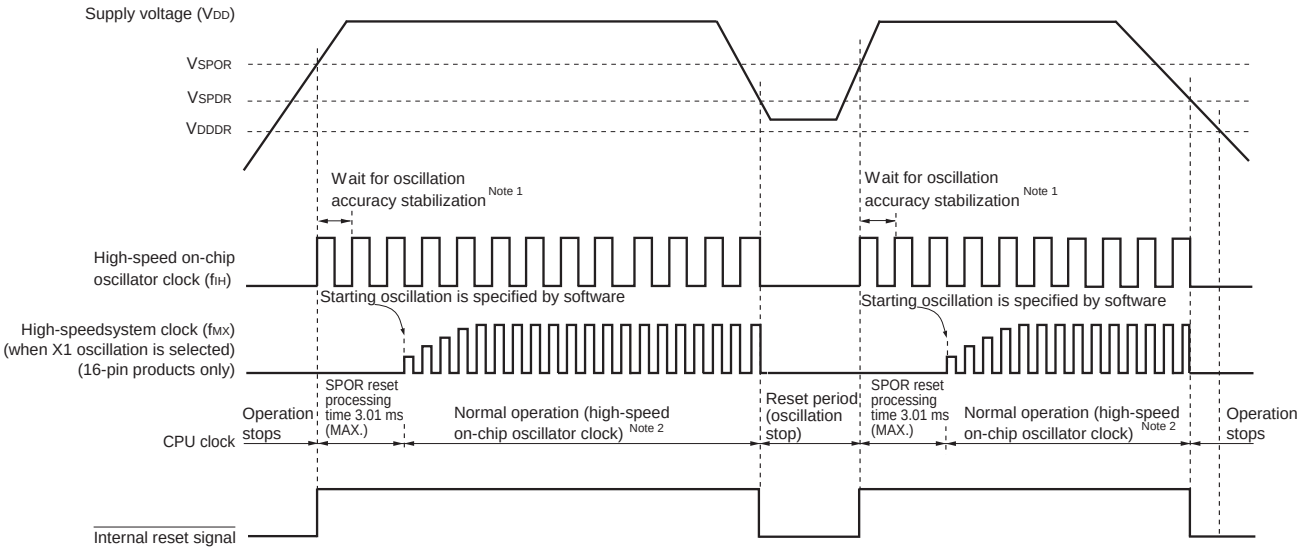
The internal reset signal is generated at power on.

The internal reset status is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{SPOR}). The internal reset is cleared when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{SPOR}).

The internal reset is generated when the supply voltage (V_{DD}) drops lower than the voltage detection level (V_{SPDR}).

Figure 18-2 shows the timing of generation of the internal reset signal by the selectable power-on-reset circuit.

Figure 18-2. Timing of Internal Reset Signal Generation



Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

2. The high-speed on-chip oscillator clock and a high-speed system clock can be selected as the CPU clock (16-pin products only). To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time.

Remark V_{SPOR}: SPOR power supply rise detection voltage

V_{SPDR}: SPOR power supply fall detection voltage

V_{DDDR}: Data retention lower limit voltage

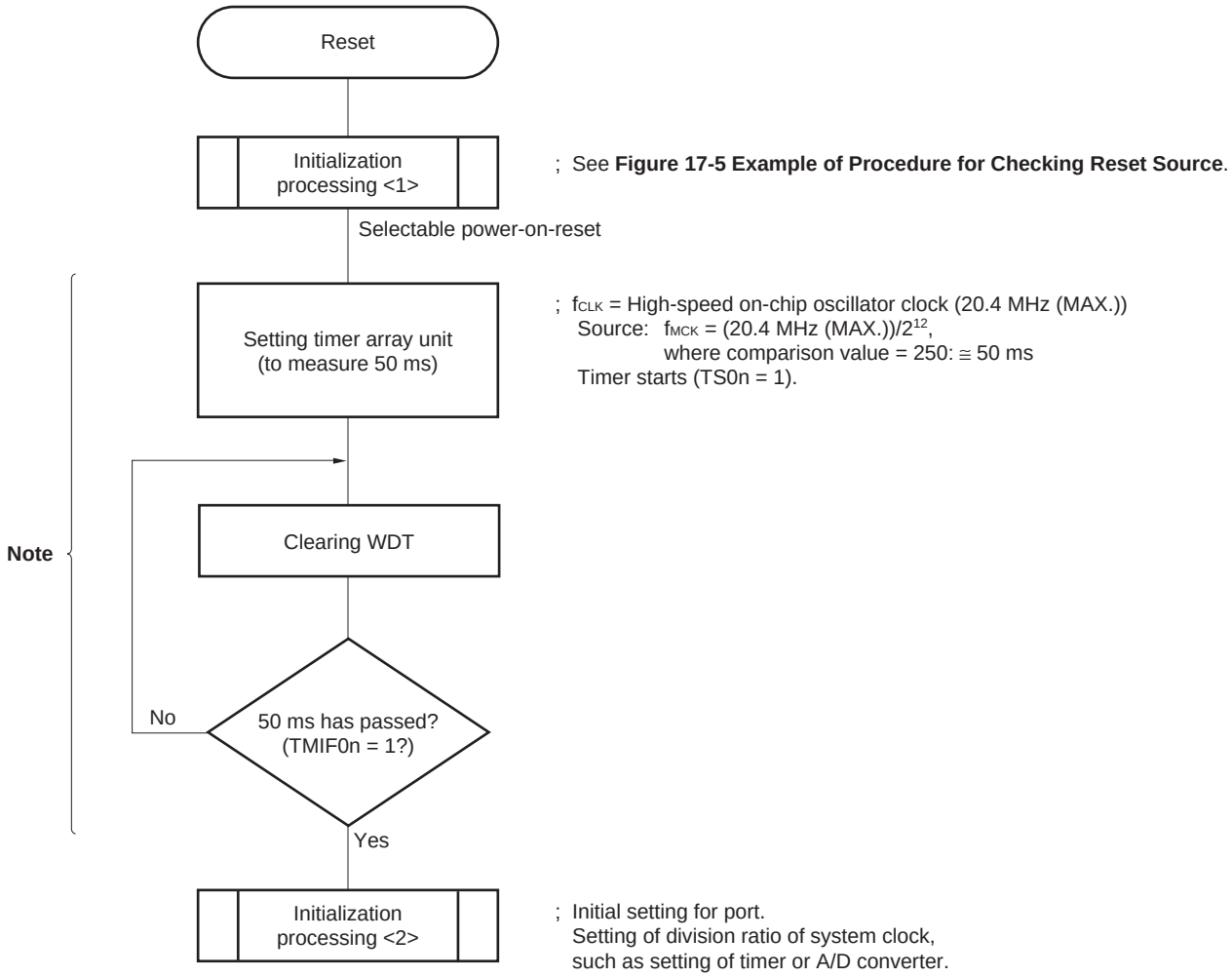
18.4 Cautions for Selectable Power-on-reset Circuit

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the SPOR detection voltage (V_{SPOR}, V_{SPDR}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a timer and etc., and then initialize the ports.

Figure 18-3. Example of Software Processing When Supply Voltage Fluctuation is 50 ms or Less in Vicinity of the Voltage Detection Level



Note If reset is generated again during this period, initialization processing <2> is not started.

Remark n: Channel number
For 10-pin products, n = 0, 1; for 16-pin products, n = 0 to 3.

CHAPTER 19 OPTION BYTE

19.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/G10 form an option byte area.

Option bytes consist of user option byte (000C0H to 000C2H) and on-chip debug option byte (000C3H).

Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set. When using the product, be sure to set the following functions by using the option bytes.

The bits to which no function is allocated must be used with their initial value.

Caution The option bytes should always be set regardless of whether each function is used.

19.1.1 User option byte (000C0H to 000C2H)

(1) 000C0H

- Operation of watchdog timer
  - Counter operation is enabled or disabled.
  - Operation is stopped or enabled in the HALT or STOP mode.
- Time setting of watchdog timer
  - Setting of overflow time of watchdog timer
  - Setting of interval interrupt time of watchdog timer

(2) 000C1H

- Setting of SPOR detection level (V_{SPOR})
- Controlling of P125/KR1/$\overline{\text{RESET}}$ pin
  - P125/KR1 or $\overline{\text{RESET}}$

(3) 000C2H

- Setting of the frequency of the high-speed on-chip oscillator
  - Select from 1.25 to 20 MHz.

19.1.2 On-chip debug option byte (000C3H)

- Control of on-chip debug operation
  - On-chip debug operation is disabled or enabled.

19.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 19-1. Format of User Option Byte (000C0H)

Address: 000C0H

7	6	5	4	3	2	1	0
1	1	1	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Overflow Time (when f_{IL} = 17.25 kHz (MAX.))	Interval Interrupt Time (when f_{IL} = 17.25 kHz (MAX.))
0	0	0	$(2^6 -1)/f_{IL}$ (3.65 ms)	$2^6/f_{IL} \times 0.75$ (2.78 ms)
0	0	1	$(2^7 -1)/f_{IL}$ (7.36 ms)	$2^7/f_{IL} \times 0.75$ (5.56 ms)
0	1	0	$(2^8 -1)/f_{IL}$ (14.7 ms)	$2^8/f_{IL} \times 0.75$ (11.1 ms)
0	1	1	$(2^9 -1)/f_{IL}$ (29.6 ms)	$2^9/f_{IL} \times 0.75$ (22.2 ms)
1	0	0	$(2^{11} -1)/f_{IL}$ (118 ms)	$2^{11}/f_{IL} \times 0.75$ (89.0 ms)
1	0	1	$(2^{13} -1)/f_{IL}$ (474 ms)	$2^{13}/f_{IL} \times 0.75$ (356 ms)
1	1	0	$(2^{14} -1)/f_{IL}$ (949 ms)	$2^{14}/f_{IL} \times 0.75$ (712 ms)
1	1	1	$(2^{16} -1)/f_{IL}$ (3799 ms)	$2^{16}/f_{IL} \times 0.75$ (2849 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode
1	Counter operation enabled in HALT/STOP mode

Cautions 1. Be sure to write 1 to bits 7 to 5.

2. Setting WDTON = 0 and WDSTBYON = 1 is prohibited.

3. The watchdog timer always generates an interval interrupt when the specified time is reached unless this is specifically disabled. If the interval interrupt from the watchdog timer is not to be used, be sure to disable the interrupt by setting the WDTIMK bit to 1.

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 19-2. Format of User Option Byte (000C1H)

Address: 000C1H

7	6	5	4	3	2	1	0
1	1	1	PORTSELB	SPORS1	SPORS0	1	1

- Setting of SPOR detection voltage

Detection voltage (V_{SPOR})		Option byte setting value	
Rising edge	Falling edge	SPORS1	SPORS0
4.28 V	4.20 V	0	0
2.90 V	2.84 V	0	1
2.57 V	2.52 V	1	0
2.16 V	2.11 V	1	1

- P125/KR1/$\overline{\text{RESET}}$ pin control

PORTSELB	P125/KR1/$\overline{\text{RESET}}$ pin control
0	Port function (P125/KR1)
1	$\overline{\text{RESET}}$ input (internal pull-up resistor can be always connected.)

Cautions 1. Be sure to write 1 to bits 7 to 5, 1, and 0.

2. Set the detection voltage (V_{SPOR}) to be within the operating voltage range. The operating voltage range depends on the setting of the user option byte (000C2H).
The operating voltage ranges are as follows.
For CPU operating frequencies from 1 MHz to 20 MHz: V_{DD} = 2.7 to 5.5 V
For CPU operating frequencies from 1 MHz to 5 MHz: V_{DD} = 2.0 to 5.5 V

Remarks 1. For details on the SPOR circuit, see **CHAPTER 18 SELECTABLE POWER-ON-RESET CIRCUIT**.

2. The detection voltage is a typical value. For details, see **24.6.4 SPOR circuit characteristics**.

Figure 19-3. Format of User Option Byte (000C2H)

Address: 000C2H

7	6	5	4	3	2	1	0
1	1	1	1	1	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator	
			Operating frequency (f_{MAIN})	Operating voltage range (V_{DD})
0	0	1	20 MHz	2.7 V to 5.5 V
0	1	0	10 MHz	
0	1	1	5 MHz	2.0 V to 5.5 V Note
1	0	0	2.5 MHz	
1	0	1	1.25 MHz	
Other than above			Setting prohibited	

Note Use this product within the voltage range from 2.25 to 5.5 V because the detection voltage (V_{SPOR}) of the selectable power-on-reset (SPOR) circuit should also be considered.

Caution Be sure to write 1 to bits 7 to 3.

19.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 19-4. Format of On-chip Debug Option Byte (000C3H)

Address: 000C3H

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

OCDENSET	Control of on-chip debug operation
0	Disables on-chip debug operation.
1	Enables on-chip debugging. Note

Note Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Caution **Bit 7 (OCDENSET) can only be specified a value.**
Be sure to set 0000101B to bits 6 to 0.

Remark The value on bits 3 and 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

19.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the link option in addition to describing to the source.

When doing so, the contents set by using the link option take precedence, even if descriptions exist in the source, as mentioned below.

A software description example of the option byte setting is shown below.

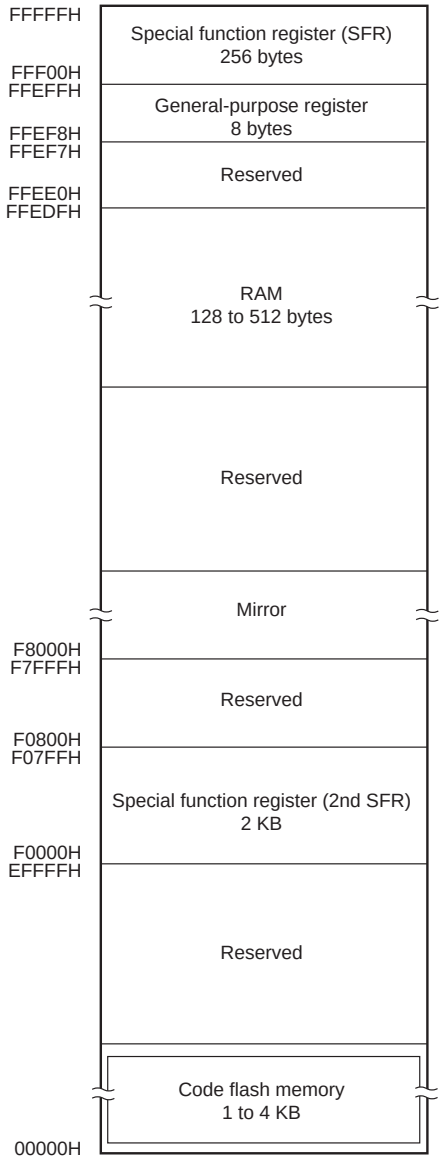
```
OPT   CSEG      OPT_BYTE
      DB        F7H     ; Enables watchdog timer operation,
                        ; Overflow time of watchdog timer is 2^9/fIL,
                        ; Stops watchdog timer operation during HALT/STOP mode
      DB        E7H     ; Select 2.7 V for VSPDR
                        ; Use the port function (P125/KR1)
      DB        FDH     ; Select 1.25 MHz as the frequency of the high-speed on-chip oscillator clock
      DB        85H     ; Enables on-chip debug operation
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction.

CHAPTER 20 FLASH MEMORY

The RL78 microcontroller incorporates the flash memory to which a program can be written, erased, and overwritten.

Caution The operating voltage during flash memory programming must be in the range from 4.5 V to 5.5 V.



The methods for programming the flash memory are as follows.

The contents of the code flash memory can be rewritten by serial programming using a flash memory programmer or an external device (UART communication).

- Serial programming by using a flash memory programmer (see **20.1**)
  Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

- Serial programming by using an external device (UART communication) (see **20.2**)
  Data can be written to the flash memory on-board, by using UART communication with an external device (a microcontroller or ASIC).

<R> ## 20.1 Serial Programming by Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78 microcontroller.

- PG-FP5, PG-FP6
- E1, E2, E2 Lite, E20 on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the RL78 microcontroller has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the RL78 microcontroller is mounted on the target system.

<R> **Table 20-1. Wiring Between RL78/G10 and Dedicated Flash Memory Programmer**

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name of RL78/G10	Pin No. of 10-pin products	Pin No. of 16-pin products
Signal Name		I/O	Pin Function			
PG-FP5, PG-FP6	E1, E2, E2 Lite, E20 on-chip debugging emulator					
–	TOOL0	I/O	Transmit/receive signal	TOOL0/P40	1	2
SI/RxD	–	I/O	Transmit/receive signal			
–	RESET_OUT	Output	Reset signal	$\overline{\text{RESET}}$	2	3
$\overline{\text{RESET}}$	–	Output				
V_{DD}[Note]		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	5	8
GND		–	Ground	V_{SS}	4	7
EMV_{DD}		–	Driving power for TOOL pin	V_{DD}	5	8

Note The name of the signal for connection in the case of the PG-FP6 is V_{CC}.

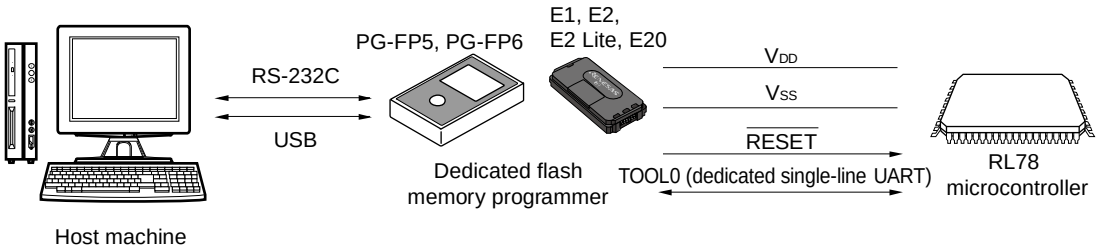
Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

About a connection between RL78 microcontroller and a connector, refer to the user's manual of each programmer. About a connection with E1, refer to **21.1 Connecting E1 On-chip Debugging Emulator**.

20.1.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

<R> **Figure 20-1. Environment for Writing Program to Flash Memory**



A host machine that controls the dedicated flash memory programmer is necessary.

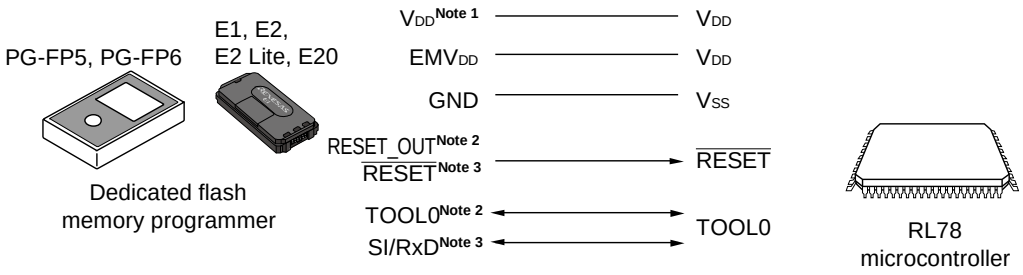
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

20.1.2 Communication mode

Communication between the dedicated flash memory programmer and the RL78 microcontroller is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78 microcontroller.

Transfer rate: Fixed to 115200 bps

<R> **Figure 20-2. Communication with Dedicated Flash Memory Programmer**



Notes 1. The name of the signal for connection in the case of the PG-FP6 is V_{CC}.

2. When using E1, E2, E2 Lite, E20 on-chip debugging emulator.

3. When using PG-FP5 or PG-FP6.

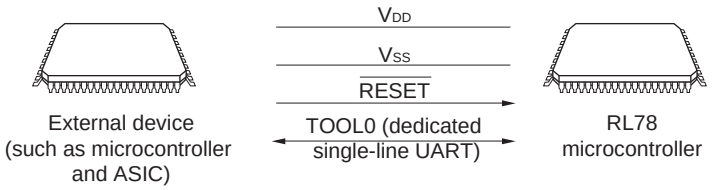
20.2 Writing to Flash Memory by Using External Device (that Incorporates UART)

On-board data writing to the internal flash memory is possible by using the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART.

20.2.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 20-3. Environment for Writing Program to Flash Memory



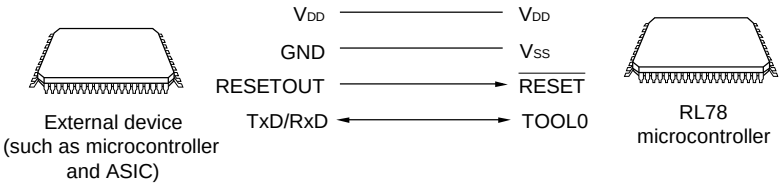
Processing to write data to or delete data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

20.2.2 Communication Mode

Communication between the external device and the RL78 microcontroller is established by serial communication using the TOOL0 pin via the dedicated UART of the RL78 microcontroller.

Transfer rate: Fixed to 115200 bps

Figure 20-4. Communication with External Device



The external device generates the following signals for the RL78 microcontroller.

Table 20-2. Pin Connection

External Device			RL78 microcontroller
Signal Name	I/O	Pin Function	Pin Name
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND	–	Ground	V_{SS}
RESETOUT	Output	Reset signal output	$\overline{RESET}$
RxD	Input	Receive signal	TOOL0
TxD	Output	Transmit signal	

20.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

Remark For the flash memory programming mode, see **20.4.2 Flash memory programming mode**.

20.3.1 P40/TOOL0 pin

In the flash memory programming mode, pull up externally with a 1 kΩ resistor, and connect it to the dedicated flash memory programmer.

When this pin is used as the port pin, use that by the following method.

When used as an input pin: Input of low-level is prohibited for t_{HD} period after external pin reset release. However, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

When used as an output pin: When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

Remarks 1. t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end for setting of the flash memory programming mode (see **24.10 Timing of Entry to Flash Memory Programming Modes**)

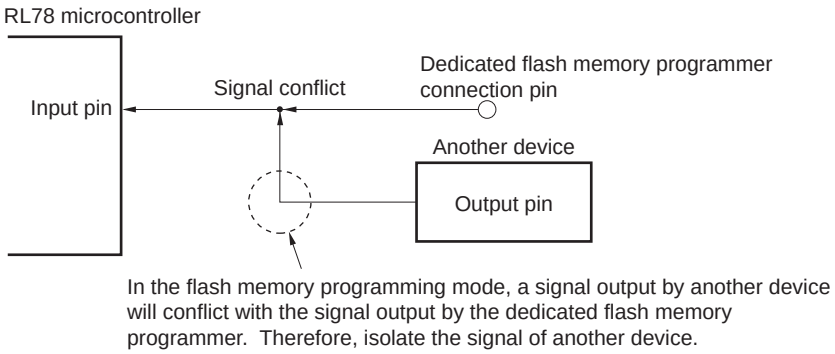
2. The SAU and IICA pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

20.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.

The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer and external device.

Figure 20-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)



20.3.3 Port pins

In the flash memory programming mode, all the pins not used for flash memory programming enter the same status as that immediately after reset. If an external device connected to the ports does not recognize the port status immediately after reset, the port pin must be connected to either to V_{DD} or V_{SS} via a resistor.

20.3.4 X1 and X2 pins (16-pin products only)

Connect X1 and X2 pins in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillator clock (f_{IH}) is used.

20.3.5 Power supply

To perform serial programming by using the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

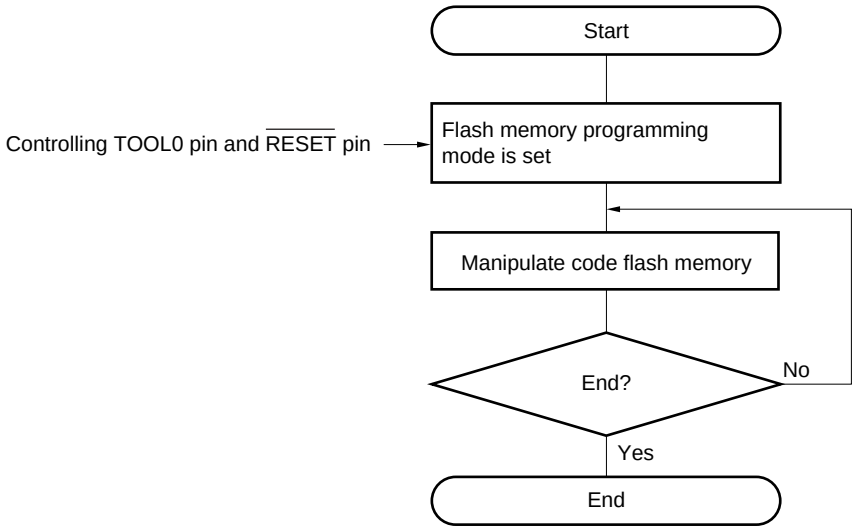
Note that the operating voltage during flash memory programming must be in the range from 4.5 V to 5.5 V. If the on-board supply voltage is less than 4.5 V, satisfy the requirement for operating voltage (4.5 V to 5.5 V) by, for example, switching to the voltage from a dedicated flash memory programmer, and isolate the on-board supply voltage.

20.4 Serial Programming Method

20.4.1 Serial programming procedure

The following figure illustrates the procedure to rewrite the contents of the code flash memory by serial programming.

Figure 20-6. Code Flash Memory Manipulation Procedure



For the flash memory programming mode, see **20.4.2 Flash memory programming mode**.

20.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory by serial programming, the flash memory programming mode must be entered.

<When performing serial programming by using the dedicated flash memory programmer>

Connect the RL78 microcontroller to the dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode. The operating voltage during the flash memory programming mode is 4.5 V to 5.5 V.

<When performing serial programming by using an external device (UART communication)>

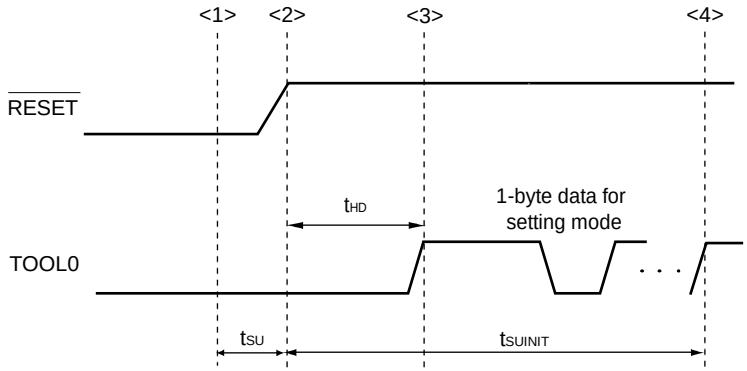
Set the TOOL0 pin to the low level, and then cancel the reset (refer to **Table 20-3**). The flash memory programming mode is then entered by following the procedures <1> to <4> shown in **Figure 20-7**.

The operating voltage during the flash memory programming mode is 4.5 V to 5.5 V.

Table 20-3. Relationship Between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

Figure 20-7. Entry to Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (SPOR reset must end before the external reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Setting of entry to the flash memory programming mode by UART reception.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until an external reset ends
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external reset ends

For details, refer to **24.10 Timing of Entry to Flash Memory Programming Modes**.

20.4.3 Communication mode

Communication mode of the RL78 microcontroller is as follows.

Table 20-4. Communication Mode

Communication Mode	Standard Setting Note 1				Pin Used
	Port	Speed Note 2	Frequency	Multiply Rate	
1-line mode	UART	115200 bps	–	–	TOOL0

Notes 1. Selection items for standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

20.4.4 Communication commands

The RL78 microcontroller performs serial programming by using commands described in Table 20-5.

The signals sent from the flash memory programmer or external device to the RL78 microcontroller are called commands, and the RL78 microcontroller performs processing corresponding to the respective commands.

Table 20-5. Flash Memory Control Commands

Classification	Command Name	Function
CRC check	CRC check	Calculate checksums
Write after erase	Write after erase	Write data after erasing data in the flash memory

<R>

20.5 Processing Time for Each Command When Dedicated Flash Memory Programmer is in Use (Reference Value)

The processing time of each command (reference values) when using PG-FP5、PG-FP6 as the dedicated flash memory programmer is shown below.

Table 20-6. Processing Time of Each Command When Using PG-FP5 (Reference Values)

Command of PG-FP5	Code Flash		
	1 KB	2 KB	4 KB
	R5F10Y14 R5F10Y44	R5F10Y16 R5F10Y46	R5F10Y17 R5F10Y47
Write after erase	1.0 s	1.0 s	1.3 s
CRC check	0.5 s	0.5 s	0.5 s

Remark The command processing times (reference values) shown in the table are typical values under the following conditions.

Port: TOOL0 (single-line UART)

Speed: 115,200 bps

<R> **Table 20-7. Processing Time for Each Command When PG-FP6 is in Use (Reference Value)**

Command of PG-FP5	Code Flash		
	1 KB	2 KB	4 KB
	R5F10Y14 R5F10Y44	R5F10Y16 R5F10Y46	R5F10Y17 R5F10Y47
Write after erase	0.6 s	0.8 s	1.1 s
CRC check	0.1 s	0.1 s	0.1 s

Remark The command processing times (reference values) shown in the table are typical values under the following conditions.

Port: TOOL0 (single-line UART)

Speed: 1,000,000 bps

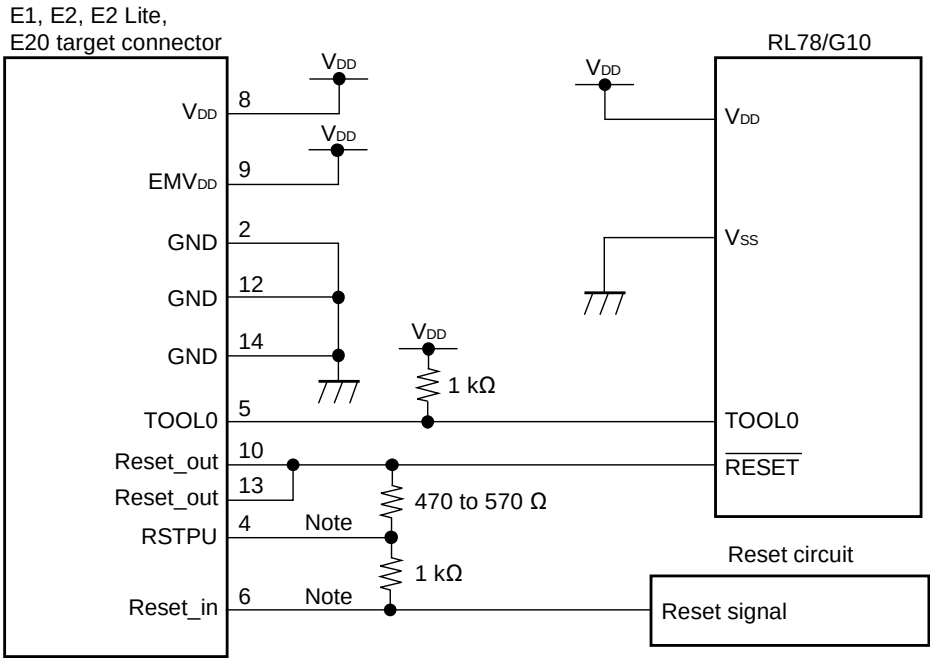
CHAPTER 21 ON-CHIP DEBUG FUNCTION

<R> 21.1 Connecting E1, E2, E2 Lite, E20 On-chip Debugging Emulator

The RL78 microcontroller uses the V_{DD}, $\overline{RESET}$, TOOL0, and V_{SS} pins to communicate with the host machine via an E1, E2, E2 Lite, E20 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

Caution The RL78 microcontroller has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

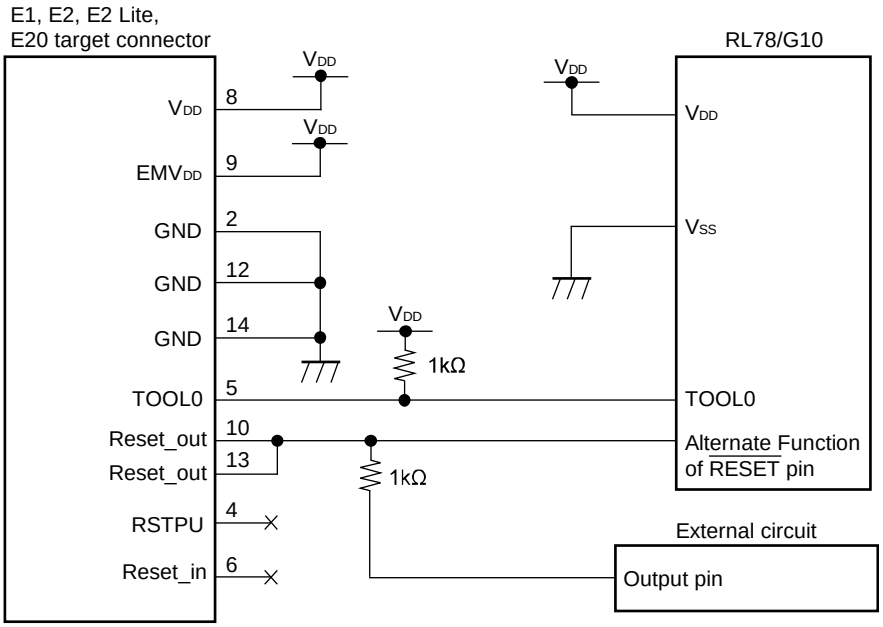
<R> **Figure 21-1. Connection Example of E1, E2, E2 Lite, E20 On-chip Debugging Emulator and RL78 microcontroller**



Note Connecting is not necessary during flash programming.

For the target system which uses the multi-use feature of $\overline{\text{RESET}}$ pin, its connection to an external circuit should be isolated.

<R> **Figure 21-2. Connection Example of E1, E2, E2 Lite, E20 On-chip Debugging Emulator and RL78 microcontroller (When using to the alternative function of $\overline{\text{RESET}}$ pin)**



21.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 19 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

Table 21-1. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes

21.3 Securing of User Resources

To perform communication between the RL78 microcontroller and E1, E2, E2 Lite, E20 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

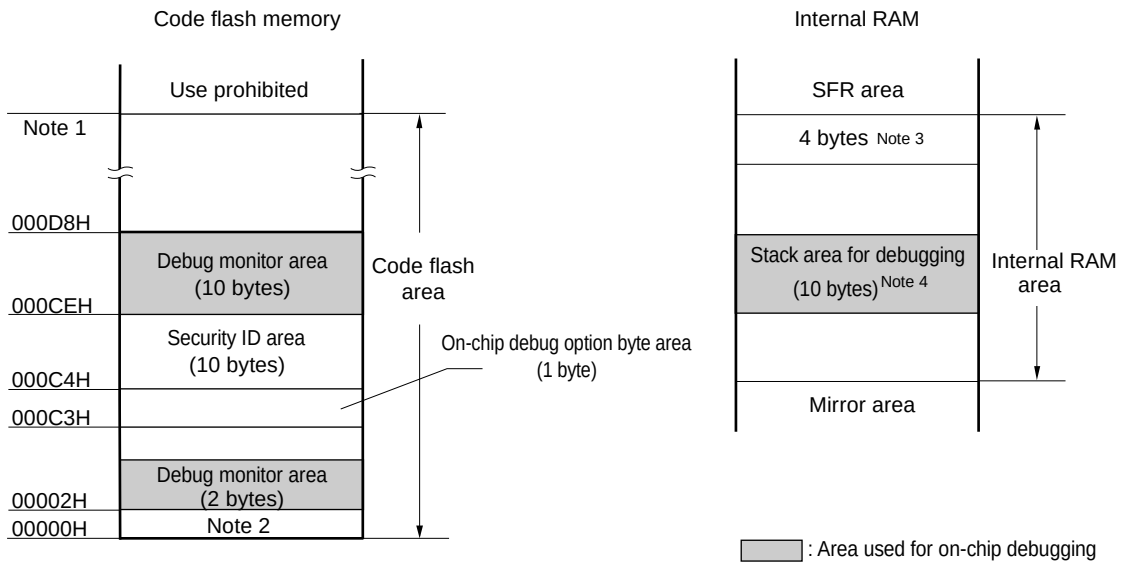
If Renesas Electronics assembler or compiler is used, the items can be set by using linker options.

(1) Securement of memory space

The shaded portions in Figure 21-3 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 21-3. Memory Spaces Where Debug Monitor Programs Are Allocated

<R>



Notes 1. Address differs depending on products as follows.

Products	Address
R5F10Y14, R5F10Y44	003FFH
R5F10Y16, R5F10Y46	007FFH
R5F10Y17, R5F10Y47	00FFFH

2. In debugging, reset vector is rewritten to address allocated to a monitor program.

<R>

3. When using the pseudo real-time RAM monitor (RRM) function and pseudo dynamic memory modification (DMM) function, four bytes of RAM area are used.
The RAM area used for the pseudo-RRM and pseudo-DMM functions is set with the build tool.
If the RAM area is not set, the first four bytes of RAM are used.
(For details on how to set the RAM area, refer to the user's manual for the build tool.)
When the pseudo-RRM and pseudo-DMM functions are not used, the RAM area can be used as internal RAM.

4. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 10 extra bytes are consumed for the stack area used.

CHAPTER 22 BCD CORRECTION CIRCUIT

22.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.

The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCD correction result register (BCDADJ).

22.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

22.2.1 BCD correction result register (BCDADJ)

The BCDADJ register stores correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand.

The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags.

The BCDADJ register is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

Figure 22-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

22.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by adding in binary the value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Examples 1: 99 + 89 = 188

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #99H ; <1>`	99H	–	–	–
`ADD A, #89H ; <2>`	22H	1	1	66H
`ADD A, !BCDADJ ; <3>`	88H	1	0	–

Examples 2: 85 + 15 = 100

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #85H ; <1>`	85H	–	–	–
`ADD A, #15H ; <2>`	9AH	0	0	66H
`ADD A, !BCDADJ ; <3>`	00H	1	1	–

Examples 3: 80 + 80 = 160

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #80H ; <1>`	80H	–	–	–
`ADD A, #80H ; <2>`	00H	1	0	60H
`ADD A, !BCDADJ ; <3>`	60H	1	0	–

(2) Subtraction: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 – 52 = 39

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #91H ; <1>`	91H	–	–	–
`SUB A, #52H ; <2>`	3FH	0	1	06H
`SUB A, !BCDADJ ; <3>`	39H	0	0	–

CHAPTER 23 INSTRUCTION SET

This chapter lists the instructions for the RL78-S1 core of the RL78 microcontroller. For details of each operation and operation code, refer to the separate document **RL78 Microcontrollers User's Manual: Software (R01US0015)**.

Remark The RL78-S2 core shares all instructions with the RL78-S1 core. Note, however, that the cores take different numbers of clock cycles to execute some instructions. The instructions which require different numbers of clock cycles are indicated by shading in the table under **23.2 Operation List**.

23.1 Conventions Used in Operation List

23.1.1 Operand identifiers and specification methods

Operands are described in the “Operand” column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Alphabetic letters in capitals and the symbols, #, !, !!, $, $!, [ ], and ES: are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: 16-bit absolute address specification
- !!: 20-bit absolute address specification
- $: 8-bit relative address specification
- $!: 16-bit relative address specification
- [ ]: Indirect address specification
- ES:: Extension address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, !!, $, $!, [ ], and ES: symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in Table 23-1 below, R0, R1, R2, etc.) can be used for description.

Table 23-1. Operand Identifiers and Specification Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol (SFR symbol) FFF00H to FFFFFH
sfrp	Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only[Note]) FFF00H to FFFFFH
saddr	FFE20H to FFF1FH Immediate data or labels
saddrp	FFE20H to FF1FH Immediate data or labels (even addresses only[Note])
addr20	00000H to FFFFFH Immediate data or labels
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions[Note])
addr5	0080H to 00BFH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label

Note Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See **Table 3-4 SFR List** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Table 3-5 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

23.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 23-2. Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
∨̄	Exclusive logical sum (exclusive OR)
¯	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

23.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

Table 23-3. Symbols in “Flag” Column

Symbol	Change of Flag Value
(Blank)	Unchanged
0	Cleared to 0
1	Set to 1
×	Set/cleared according to the result
R	Previously saved value is restored

23.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

A interrupt and DMA transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 23-4. Use Example of PREFIX Operation Code

Instruction	Opcode				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	–
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	–	–	–	–
MOV A, ES:[HL]	11H	8BH	–	–	–

Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.

23.2 Operation List

Table 23-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	–	r ← byte			
		PSW, #byte	3	3	–	PSW ← byte	×	×	×
		CS, #byte	3	1	–	CS ← byte			
		ES, #byte	2	1	–	ES ← byte			
		!addr16, #byte	4	1	–	(addr16) ← byte			
		ES:!addr16, #byte	5	2	–	(ES, addr16) ← byte			
		saddr, #byte	3	1	–	(saddr) ← byte			
		sfr, #byte	3	1	–	sfr ← byte			
		[DE+byte], #byte	3	1	–	(DE+byte) ← byte			
		ES:[DE+byte],#byte	4	2	–	((ES, DE)+byte) ← byte			
		[HL+byte], #byte	3	1	–	(HL+byte) ← byte			
		ES:[HL+byte],#byte	4	2	–	((ES, HL)+byte) ← byte			
		[SP+byte], #byte	3	1	–	(SP+byte) ← byte			
		word[B], #byte	4	1	–	(B+word) ← byte			
		ES:word[B], #byte	5	2	–	((ES, B)+word) ← byte			
		word[C], #byte	4	1	–	(C+word) ← byte			
		ES:word[C], #byte	5	2	–	((ES, C)+word) ← byte			
		word[BC], #byte	4	1	–	(BC+word) ← byte			
		ES:word[BC], #byte	5	2	–	((ES, BC)+word) ← byte			
		A, r Note 3	1	1	–	A ← r			
		r, A Note 3	1	1	–	r ← A			
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	×	×	×
		A, CS	2	1	–	A ← CS			
		CS, A	2	1	–	CS ← A			
		A, ES	2	1	–	A ← ES			
		ES, A	2	1	–	ES ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		A, ES:!addr16	4	2	5	A ← (ES, addr16)			
		!addr16, A	3	1	–	(addr16) ← A			
		ES:!addr16, A	4	2	–	(ES, addr16) ← A			
		A, saddr	2	1	–	A ← (saddr)			
		saddr, A	2	1	–	(saddr) ← A			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

3. Except r = A

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	**Note 2**		Z AC CY
8-bit data transfer	MOV	A, sfr	2	1	–	A ← sfr	
		sfr, A	2	1	–	sfr ← A	
		A, [DE]	1	1	4	A ← (DE)	
		[DE], A	1	1	–	(DE) ← A	
		A, ES:[DE]	2	2	5	A ← (ES, DE)	
		ES:[DE], A	2	2	–	(ES, DE) ← A	
		A, [HL]	1	1	4	A ← (HL)	
		[HL], A	1	1	–	(HL) ← A	
		A, ES:[HL]	2	2	5	A ← (ES, HL)	
		ES:[HL], A	2	2	–	(ES, HL) ← A	
		A, [DE+byte]	2	1	4	A ← (DE + byte)	
		[DE+byte], A	2	1	–	(DE + byte) ← A	
		A, ES:[DE+byte]	3	2	5	A ← ((ES, DE) + byte)	
		ES:[DE+byte], A	3	2	–	((ES, DE) + byte) ← A	
		A, [HL+byte]	2	1	4	A ← (HL + byte)	
		[HL+byte], A	2	1	–	(HL + byte) ← A	
		A, ES:[HL+byte]	3	2	5	A ← ((ES, HL) + byte)	
		ES:[HL+byte], A	3	2	–	((ES, HL) + byte) ← A	
		A, [SP+byte]	2	1	–	A ← (SP + byte)	
		[SP+byte], A	2	1	–	(SP + byte) ← A	
		A, word[B]	3	1	4	A ← (B + word)	
		word[B], A	3	1	–	(B + word) ← A	
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)	
		ES:word[B], A	4	2	–	((ES, B) + word) ← A	
		A, word[C]	3	1	4	A ← (C + word)	
		word[C], A	3	1	–	(C + word) ← A	
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)	
		ES:word[C], A	4	2	–	((ES, C) + word) ← A	
		A, word[BC]	3	1	4	A ← (BC + word)	
		word[BC], A	3	1	–	(BC + word) ← A	
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)	
		ES:word[BC], A	4	2	–	((ES, BC) + word) ← A	

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (3/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	**Note 2**		Z AC CY
8-bit data transfer	MOV	A, [HL+B]	2	1	4	A ← (HL + B)	
		[HL+B], A	2	1	–	(HL + B) ← A	
		A, ES:[HL+B]	3	2	5	A ← ((ES, HL) + B)	
		ES:[HL+B], A	3	2	–	((ES, HL) + B) ← A	
		A, [HL+C]	2	1	4	A ← (HL + C)	
		[HL+C], A	2	1	–	(HL + C) ← A	
		A, ES:[HL+C]	3	2	5	A ← ((ES, HL) + C)	
		ES:[HL+C], A	3	2	–	((ES, HL) + C) ← A	
		X, !addr16	3	1	4	X ← (addr16)	
		X, ES:!addr16	4	2	5	X ← (ES, addr16)	
		X, saddr	2	1	–	X ← (saddr)	
		B, !addr16	3	1	4	B ← (addr16)	
		B, ES:!addr16	4	2	5	B ← (ES, addr16)	
		B, saddr	2	1	–	B ← (saddr)	
		C, !addr16	3	1	4	C ← (addr16)	
		C, ES:!addr16	4	2	5	C ← (ES, addr16)	
		C, saddr	2	1	–	C ← (saddr)	
		ES, saddr	3	1	–	ES ← (saddr)	
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	–	A ←→ r	
		A, !addr16	4	2	–	A ←→ (addr16)	
		A, ES:!addr16	5	3	–	A ←→ (ES, addr16)	
		A, saddr	3	2	–	A ←→ (saddr)	
		A, sfr	3	2	–	A ←→ sfr	
		A, [DE]	2	2	–	A ←→ (DE)	
		A, ES:[DE]	3	3	–	A ←→ (ES, DE)	
		A, [HL]	2	2	–	A ←→ (HL)	
		A, ES:[HL]	3	3	–	A ←→ (ES, HL)	
		A, [DE+byte]	3	2	–	A ←→ (DE + byte)	
		A, ES:[DE+byte]	4	3	–	A ←→ ((ES, DE) + byte)	
		A, [HL+byte]	3	2	–	A ←→ (HL + byte)	
		A, ES:[HL+byte]	4	3	–	A ←→ ((ES, HL) + byte)	

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

3. Except r = A

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (4/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit data transfer	XCH	A, [HL+B]	2	2	–	A ←→ (HL+B)			
		A, ES:[HL+B]	3	3	–	A ←→ ((ES, HL)+B)			
		A, [HL+C]	2	2	–	A ←→ (HL+C)			
		A, ES:[HL+C]	3	3	–	A ←→ ((ES, HL)+C)			
	ONEB	A	1	1	–	A ← 01H			
		X	1	1	–	X ← 01H			
		B	1	1	–	B ← 01H			
		C	1	1	–	C ← 01H			
		!addr16	3	1	–	(addr16) ← 01H			
		ES:!addr16	4	2	–	(ES, addr16) ← 01H			
		saddr	2	1	–	(saddr) ← 01H			
	CLRB	A	1	1	–	A ← 00H			
		X	1	1	–	X ← 00H			
		B	1	1	–	B ← 00H			
		C	1	1	–	C ← 00H			
		!addr16	3	1	–	(addr16) ← 00H			
		ES:!addr16	4	2	–	(ES,addr16) ← 00H			
		saddr	2	1	–	(saddr) ← 00H			
	MOVS	[HL+byte], X	3	1	–	(HL+byte) ← X	×		×
		ES:[HL+byte], X	4	2	–	(ES, HL+byte) ← X	×		×
16-bit data transfer	MOVW	rp, #word	3	2	–	rp ← word			
		saddrp, #word	4	2	–	(saddrp) ← word			
		sfrp, #word	4	2	–	sfrp ← word			
		AX, rp Note 3	1	2	–	AX ← rp			
		rp, AX Note 3	1	2	–	rp ← AX			
		AX, !addr16	3	2	5	AX ← (addr16)			
		!addr16, AX	3	2	–	(addr16) ← AX			
		AX, ES:!addr16	4	3	6	AX ← (ES, addr16)			
		ES:!addr16, AX	4	3	–	(ES, addr16) ← AX			
		AX, saddrp	2	2	–	AX ← (saddrp)			
		saddrp, AX	2	2	–	(saddrp) ← AX			
		AX, sfrp	2	2	–	AX ← sfrp			
		sfrp, AX	2	2	–	sfrp ← AX			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

3. Except rp = AX

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (5/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	**Note 2**		Z AC CY
16-bit data transfer	MOVW	AX, [DE]	1	2	5	AX ← (DE)	
		[DE], AX	1	2	–	(DE) ← AX	
		AX, ES:[DE]	2	3	6	AX ← (ES, DE)	
		ES:[DE], AX	2	3	–	(ES, DE) ← AX	
		AX, [HL]	1	2	5	AX ← (HL)	
		[HL], AX	1	2	–	(HL) ← AX	
		AX, ES:[HL]	2	3	6	AX ← (ES, HL)	
		ES:[HL], AX	2	3	–	(ES, HL) ← AX	
		AX, [DE+byte]	2	2	5	AX ← (DE+byte)	
		[DE+byte], AX	2	2	–	(DE+byte) ← AX	
		AX, ES:[DE+byte]	3	3	6	AX ← ((ES, DE) + byte)	
		ES:[DE+byte], AX	3	3	–	((ES, DE) + byte) ← AX	
		AX, [HL+byte]	2	2	5	AX ← (HL + byte)	
		[HL+byte], AX	2	2	–	(HL + byte) ← AX	
		AX, ES:[HL+byte]	3	3	6	AX ← ((ES, HL) + byte)	
		ES:[HL+byte], AX	3	3	–	((ES, HL) + byte) ← AX	
		AX, [SP+byte]	2	2	–	AX ← (SP + byte)	
		[SP+byte], AX	2	2	–	(SP + byte) ← AX	
		AX, word[B]	3	2	5	AX ← (B + word)	
		word[B], AX	3	2	–	(B+ word) ← AX	
		AX, ES:word[B]	4	3	6	AX ← ((ES, B) + word)	
		ES:word[B], AX	4	3	–	((ES, B) + word) ← AX	
		AX, word[C]	3	2	5	AX ← (C + word)	
		word[C], AX	3	2	–	(C + word) ← AX	
		AX, ES:word[C]	4	3	6	AX ← ((ES, C) + word)	
		ES:word[C], AX	4	3	–	((ES, C) + word) ← AX	
		AX, word[BC]	3	2	5	AX ← (BC + word)	
		word[BC], AX	3	2	–	(BC + word) ← AX	
		AX, ES:word[BC]	4	3	6	AX ← ((ES, BC) + word)	
		ES:word[BC], AX	4	3	–	((ES, BC) + word) ← AX	

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
16-bit data transfer	MOVW	BC, !addr16	3	2	5	BC ← (addr16)			
		BC, ES:!addr16	4	3	6	BC ← (ES, addr16)			
		DE, !addr16	3	2	5	DE ← (addr16)			
		DE, ES:!addr16	4	3	6	DE ← (ES, addr16)			
		HL, !addr16	3	2	5	HL ← (addr16)			
		HL, ES:!addr16	4	3	6	HL ← (ES, addr16)			
		BC, saddrp	2	2	–	BC ← (saddrp)			
		DE, saddrp	2	2	–	DE ← (saddrp)			
		HL, saddrp	2	2	–	HL ← (saddrp)			
	XCHW	AX, rp Note 3	1	2	–	AX ←→ rp			
	ONEW	AX	1	2	–	AX ← 0001H			
		BC	1	2	–	BC ← 0001H			
	CLRW	AX	1	2	–	AX ← 0000H			
		BC	1	2	–	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr)+byte	×	×	×
		A, r Note 4	2	1	–	A, CY ← A + r	×	×	×
		r, A	2	1	–	r, CY ← r + A	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A+ (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A+((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

3. Except rp = AX

4. Except r = A

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (7/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	–	A, CY ← A+byte+CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) +byte+CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A + r + CY	×	×	×
		r, A	2	1	–	r, CY ← r + A + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)+CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)+CY	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)+CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A+ (HL) + CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A+ (ES, HL) + CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A+ (HL+byte) + CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A+ ((ES, HL)+byte) + CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A+ (HL+B) +CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A+((ES, HL)+B)+CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A+ (HL+C)+CY	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A+ ((ES, HL)+C)+CY	×	×	×
	SUB	A, #byte	2	1	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r	×	×	×
		r, A	2	1	–	r, CY ← r – A	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16)	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A – (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A – ((ES, HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A – ((ES, HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A – (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A – ((ES, HL)+C)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

3. Except r = A

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r – CY	×	×	×
		r, A	2	1	–	r, CY ← r – A – CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16) – CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES, addr16) – CY	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr) – CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A – (ES, HL) – CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A – (HL+byte) – CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A – ((ES, HL)+byte) – CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A – (HL+B) – CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A – ((ES, HL)+B) – CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A – (HL+C) – CY	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY ← A – ((ES:HL)+C) – CY	×	×	×
	AND	A, #byte	2	1	–	A ← A ∧ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	–	A ← A ∧ r	×		
		r, A	2	1	–	r ← r ∧ A	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, saddr	2	1	–	A ← A ∧ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∧ (HL+byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∧ ((ES:HL)+byte)	×		
		A, [HL+B]	2	1	4	A ← A ∧ (HL+B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∧ ((ES:HL)+B)	×		
		A, [HL+C]	2	1	4	A ← A ∧ (HL+C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∧ ((ES:HL)+C)	×		

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

3. Except r = A

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (9/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	**Note 2**		Z AC CY
8-bit operation	OR	A, #byte	2	1	–	A ← A∨byte	×
		saddr, #byte	3	2	–	(saddr) ← (saddr)∨byte	×
		A, r Note 3	2	1	–	A ← A∨r	×
		r, A	2	1	–	r ← r∨A	×
		A, !addr16	3	1	4	A ← A∨(addr16)	×
		A, ES:!addr16	4	2	5	A ← A∨(ES:addr16)	×
		A, saddr	2	1	–	A ← A∨(saddr)	×
		A, [HL]	1	1	4	A ← A∨(H)	×
		A, ES:[HL]	2	2	5	A ← A∨(ES:HL)	×
		A, [HL+byte]	2	1	4	A ← A∨(HL+byte)	×
		A, ES:[HL+byte]	3	2	5	A ← A∨((ES:HL)+byte)	×
		A, [HL+B]	2	1	4	A ← A∨(HL+B)	×
		A, ES:[HL+B]	3	2	5	A ← A∨((ES:HL)+B)	×
		A, [HL+C]	2	1	4	A ← A∨(HL+C)	×
		A, ES:[HL+C]	3	2	5	A ← A∨((ES:HL)+C)	×
	XOR	A, #byte	2	1	–	A ← A∨̄byte	×
		saddr, #byte	3	2	–	(saddr) ← (saddr)∨̄byte	×
		A, r Note 3	2	1	–	A ← A∨̄r	×
		r, A	2	1	–	r ← r∨̄A	×
		A, !addr16	3	1	4	A ← A∨̄(addr16)	×
		A, ES:!addr16	4	2	5	A ← A∨̄(ES:addr16)	×
		A, saddr	2	1	–	A ← A∨̄(saddr)	×
		A, [HL]	1	1	4	A ← A∨̄(HL)	×
		A, ES:[HL]	2	2	5	A ← A∨̄(ES:HL)	×
		A, [HL+byte]	2	1	4	A ← A∨̄(HL+byte)	×
		A, ES:[HL+byte]	3	2	5	A ← A∨̄((ES:HL)+byte)	×
		A, [HL+B]	2	1	4	A ← A∨̄(HL+B)	×
		A, ES:[HL+B]	3	2	5	A ← A∨̄((ES:HL)+B)	×
		A, [HL+C]	2	1	4	A ← A∨̄(HL+C)	×
		A, ES:[HL+C]	3	2	5	A ← A∨̄((ES:HL)+C)	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

3. Except r = A

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (10/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	–	A – byte	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r Note3	2	1	–	A – r	×	×	×
		r, A	2	1	–	r – A	×	×	×
		A, !addr16	3	1	4	A – (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A – (ES:addr16)	×	×	×
		A, saddr	2	1	–	A – (saddr)	×	×	×
		A, [HL]	1	1	4	A – (HL)	×	×	×
		A, ES:[HL]	2	2	5	A – (ES:HL)	×	×	×
		A, [HL+byte]	2	1	4	A – (HL+byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A – ((ES:HL)+byte)	×	×	×
		A, [HL+B]	2	1	4	A – (HL+B)	×	×	×
		A, ES:[HL+B]	3	2	5	A – ((ES:HL)+B)	×	×	×
		A, [HL+C]	2	1	4	A – (HL+C)	×	×	×
		A, ES:[HL+C]	3	2	5	A – ((ES:HL)+C)	×	×	×
	CMP0	A	1	1	–	A – 00H	×	0	0
		X	1	1	–	X – 00H	×	0	0
		B	1	1	–	B – 00H	×	0	0
		C	1	1	–	C – 00H	×	0	0
		!addr16	3	1	4	(addr16) – 00H	×	0	0
		ES:!addr16	4	2	5	(ES:addr16) – 00H	×	0	0
		saddr	2	1	–	(saddr) – 00H	×	0	0
	CMPS	X, [HL+byte]	3	1	4	X – (HL+byte)	×	×	×
		X, ES:[HL+byte]	4	2	5	X – ((ES:HL)+byte)	×	×	×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

3. Except r = A

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (11/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	2	–	AX, CY ← AX+word	×	×	×
		AX, AX	1	2	–	AX, CY ← AX+AX	×	×	×
		AX, BC	1	2	–	AX, CY ← AX+BC	×	×	×
		AX, DE	1	2	–	AX, CY ← AX+DE	×	×	×
		AX, HL	1	2	–	AX, CY ← AX+HL	×	×	×
		AX, !addr16	3	2	5	AX, CY ← AX+(addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX, CY ← AX+(ES:addr16)	×	×	×
		AX, saddrp	2	2	–	AX, CY ← AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX, CY ← AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	3	6	AX, CY ← AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	2	–	AX, CY ← AX – word	×	×	×
		AX, BC	1	2	–	AX, CY ← AX – BC	×	×	×
		AX, DE	1	2	–	AX, CY ← AX – DE	×	×	×
		AX, HL	1	2	–	AX, CY ← AX – HL	×	×	×
		AX, !addr16	3	2	5	AX, CY ← AX – (addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, saddrp	2	2	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX, CY ← AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	3	6	AX, CY ← AX – ((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	2	–	AX – word	×	×	×
		AX, BC	1	2	–	AX – BC	×	×	×
		AX, DE	1	2	–	AX – DE	×	×	×
		AX, HL	1	2	–	AX – HL	×	×	×
		AX, !addr16	3	2	5	AX – (addr16)	×	×	×
		AX, ES:!addr16	4	3	6	AX – (ES:addr16)	×	×	×
		AX, saddrp	2	2	–	AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	2	5	AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	3	6	AX – ((ES:HL)+byte)	×	×	×
Multiply	MULU	X	1	2	–	AX ← A×X			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (12/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Increment/ decrement	INC	r	1	1	–	r ← r+1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16)+1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16)+1	×	×	
		saddr	2	2	–	(saddr) ← (saddr)+1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte)+1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1	×	×	
	DEC	r	1	1	–	r ← r – 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) – 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) – 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte) – 1	×	×	
	INCW	rp	1	2	–	rp ← rp+1			
		!addr16	3	4	–	(addr16) ← (addr16)+1			
		ES:!addr16	4	5	–	(ES, addr16) ← (ES, addr16)+1			
		saddrp	2	4	–	(saddrp) ← (saddrp)+1			
		[HL+byte]	3	4	–	(HL+byte) ← (HL+byte)+1			
		ES: [HL+byte]	4	5	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1			
	DECW	rp	1	2	–	rp ← rp – 1			
		!addr16	3	4	–	(addr16) ← (addr16) – 1			
		ES:!addr16	4	5	–	(ES, addr16) ← (ES, addr16) – 1			
		saddrp	2	4	–	(saddrp) ← (saddrp) – 1			
		[HL+byte]	3	4	–	(HL+byte) ← (HL+byte) – 1			
		ES: [HL+byte]	4	5	–	((ES:HL)+byte) ← ((ES:HL)+byte) – 1			
Shift	SHR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) ×cnt			×
	SHRW	AX, cnt	2	2	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) ×cnt			×
	SHL	A, cnt	2	1	–	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) ×cnt			×
		B, cnt	2	1	–	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) ×cnt			×
		C, cnt	2	1	–	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) ×cnt			×
	SHLW	AX, cnt	2	2	–	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) ×cnt			×
		BC, cnt	2	2	–	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) ×cnt			×
	SAR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) ×cnt			×
	SARW	AX, cnt	2	2	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) ×cnt			×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

Remarks 1. These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

2. cnt indicates the bit shift count.

Table 23-5. Operation List (13/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Rotate	ROR	A, 1	2	1	–	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$)×1			×
	ROL	A, 1	2	1	–	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$)×1			×
	RORC	A, 1	2	1	–	(CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$)×1			×
	ROLC	A, 1	2	1	–	(CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$)×1			×
	ROLWC	AX,1	2	2	–	(CY $\leftarrow AX_{15}$, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) ×1			×
		BC,1	2	2	–	(CY $\leftarrow BC_{15}$, $BC_0 \leftarrow$ CY, $BC_{m+1} \leftarrow BC_m$) ×1			×
Bit manipulate	MOV1	CY, A.bit	2	1	–	CY ← A.bit			×
		A.bit, CY	2	1	–	A.bit ← CY			
		CY, PSW.bit	3	1	–	CY ← PSW.bit			×
		PSW.bit, CY	3	4	–	PSW.bit ← CY	×	×	
		CY, saddr.bit	3	1	–	CY ← (saddr).bit			×
		saddr.bit, CY	3	2	–	(saddr).bit ← CY			
		CY, sfr.bit	3	1	–	CY ← sfr.bit			×
		sfr.bit, CY	3	2	–	sfr.bit ← CY			
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		[HL].bit, CY	2	2	–	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	–	(ES, HL).bit ← CY			
	AND1	CY, A.bit	2	1	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∧ PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∧ sfr.bit			×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, A.bit	2	1	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∨ PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∨ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (14/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Bit manipulate	XOR1	CY, A.bit	2	1	–	CY ← CY ⊻ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ⊻ PSW.bit			×
		CY, saddr.bit	3	1	–	CY ← CY ⊻ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ⊻ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⊻ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⊻ (ES, HL).bit			×
	SET1	A.bit	2	1	–	A.bit ← 1			
		PSW.bit	3	4	–	PSW.bit ← 1	×	×	×
		!addr16.bit	4	2	–	(addr16).bit ← 1			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 1			
		saddr.bit	3	2	–	(saddr).bit ← 1			
		sfr.bit	3	2	–	sfr.bit ← 1			
		[HL].bit	2	2	–	(HL).bit ← 1			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 1			
	CLR1	A.bit	2	1	–	A.bit ← 0			
		PSW.bit	3	4	–	PSW.bit ← 0	×	×	×
		!addr16.bit	4	2	–	(addr16).bit ← 0			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 0			
<R>		saddr.bit	3	2	–	(saddr).bit ← 0			
		sfr.bit	3	2	–	sfr.bit ← 0			
		[HL].bit	2	2	–	(HL).bit ← 0			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 0			
	SET1	CY	2	1	–	CY ← 1			1
	CLR1	CY	2	1	–	CY ← 0			0
	NOT1	CY	2	1	–	CY ← $\overline{CY}$			×

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Call/ return	CALL	rp	2	4	–	(SP – 2) ← $(PC+2)_S$, (SP – 3) ← $(PC+2)_H$, (SP – 4) ← $(PC+2)_L$, PC ← CS, rp, SP ← SP – 4			
		$!addr20	3	4	–	(SP – 2) ← $(PC+3)_S$, (SP – 3) ← $(PC+3)_H$, (SP – 4) ← $(PC+3)_L$, PC ← PC+3+jdisp16, SP ← SP – 4			
		!addr16	3	4	–	(SP – 2) ← $(PC+3)_S$, (SP – 3) ← $(PC+3)_H$, (SP – 4) ← $(PC+3)_L$, PC ← 0000, addr16, SP ← SP – 4			
		!!addr20	4	4	–	(SP – 2) ← $(PC+4)_S$, (SP – 3) ← $(PC+4)_H$, (SP – 4) ← $(PC+4)_L$, PC ← addr20, SP ← SP – 4			
	CALLT	[addr5]	2	6	–	(SP – 2) ← $(PC+2)_S$, (SP – 3) ← $(PC+2)_H$, (SP – 4) ← $(PC+2)_L$, PC_S ← 0000, PC_H ← (0000, addr5+1), PC_L ← (0000, addr5), SP ← SP – 4			
	BRK	-	2	7	–	(SP – 1) ← PSW, (SP – 2) ← $(PC+2)_S$, (SP – 3) ← $(PC+2)_H$, (SP – 4) ← $(PC+2)_L$, PC_S ← 0000, PC_H ← (0007FH), PC_L ← (0007EH), SP ← SP – 4, IE ← 0			
	RET	-	1	7	–	PC_L ← (SP), PC_H ← (SP+1), PC_S ← (SP+2), SP ← SP+4			
	RETI	-	2	8	–	PC_L ← (SP), PC_H ← (SP+1), PC_S ← (SP+2), PSW← (SP+3), SP ← SP+4	R	R	R
	RETB	-	2	8	–	PC_L ← (SP), PC_H ← (SP+1), PC_S ← (SP+2), PSW ← (SP+3), SP ← SP+4	R	R	R

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	**Note 2**		Z AC CY
Stack manipulate	PUSH	PSW	2	2	–	(SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP–2	
		rp	1	2	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2	
	POP	PSW	2	4	–	PSW ← (SP+1), SP ← SP + 2	R R R
		rp	1	2	–	rp_L ←(SP), rp_H ← (SP+1), SP ← SP + 2	
	MOVW	SP, #word	4	2	–	SP ← word	
		SP, AX	2	2	–	SP ← AX	
		AX, SP	2	2	–	AX ← SP	
		HL, SP	3	2	–	HL ← SP	
		BC, SP	3	2	–	BC ← SP	
		DE, SP	3	2	–	DE ← SP	
	ADDW	SP, #byte	2	2	–	SP ← SP + byte	
	SUBW	SP, #byte	2	2	–	SP ← SP – byte	
Unconditional branch	BR	AX	2	3	–	PC ← CS, AX	
		$addr20	2	3	–	PC ← PC + 2 + jdisp8	
		$!addr20	3	3	–	PC ← PC + 3 + jdisp16	
		!addr16	3	3	–	PC ← 0000, addr16	
		!!addr20	4	3	–	PC ← addr20	
Conditional branch	BC	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if CY = 1	
	BNC	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if CY = 0	
	BZ	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if Z = 1	
	BNZ	$addr20	2	2/4 Note3	–	PC ← PC + 2 + jdisp8 if Z = 0	
	BH	$addr20	3	2/4 Note3	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=0	
	BNH	$addr20	3	2/4 Note3	–	PC ← PC + 3 + jdisp8 if (Z∨CY)=1	
	BT	saddr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1	
		sfr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1	
		A.bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if A.bit = 1	
		PSW.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1	
		[HL].bit, $addr20	3	3/5 Note3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1	
		ES:[HL].bit, $addr20	4	4/6 Note3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1	

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

Table 23-5. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	**Note 2**		Z	AC	CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5 Note3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6 Note3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5 Note3	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5 Note3	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6 Note3	–	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit			
Conditional skip	SKC	–	2	1	–	Next instruction skip if CY = 1			
	SKNC	–	2	1	–	Next instruction skip if CY = 0			
	SKZ	–	2	1	–	Next instruction skip if Z = 1			
	SKNZ	–	2	1	–	Next instruction skip if Z = 0			
	SKH	–	2	1	–	Next instruction skip if (Z∨CY)=0			
	SKNH	–	2	1	–	Next instruction skip if (Z∨CY)=1			
CPU control	NOP	–	1	1	–	No Operation			
	EI	–	3	4	–	IE ← 1 (Enable Interrupt)			
	DI	–	3	4	–	IE ← 0 (Disable Interrupt)			
	HALT	–	2	3	–	Set HALT Mode			
	STOP	–	2	3	–	Set STOP Mode			

Notes 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the quadruple of the number given here plus 6 further clock cycles.

CHAPTER 24 ELECTRICAL SPECIFICATIONS

Cautions 1. The RL78 microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

2. The pins mounted depend on the product. Refer to 2.1 Port Functions and 2.2.1 Functions for each product.

3. Use this product within the voltage range from 2.25 to 5.5 V because the detection voltage (V_{SPOR}) of the selectable power-on-reset (SPOR) circuit should also be considered.

24.1 Absolute Maximum Ratings

(T_A = 25°C)

Parameter	Symbols	Conditions		Ratings	Unit
Supply Voltage	V_{DD}			−0.5 to +6.5	V
Input Voltage	V_{I1}			−0.3 to V_{DD} + 0.3 Note	V
Output Voltage	V_{O1}			−0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH1}	Per pin		−40	mA
		Total of all pins	P40, P41	−70	mA
			P00 to P07	−100	mA
Output current, low	I_{OL1}	Per pin		40	mA
		Total of all pins	P40, P41	70	mA
			P00 to P07	100	mA
Operating ambient temperature	T_A			−40 to +85	°C
Storage temperature	T_{stg}			−65 to +150	°C

Note Must be 6.5 V or lower.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

2. The reference voltage is V_{SS}.

24.2 Oscillator Characteristics

24.2.1 X1 oscillator characteristics

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X)Note	Ceramic resonator/ crystal resonator	2.7 V ≤ V_{DD} ≤ 5.5 V	1		20	MHz
		2.0 V ≤ V_{DD} < 2.7 V	1		5	MHz

Note Indicates only permissible oscillator frequency ranges. Refer to AC Characteristics for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark When using the X1 oscillator, refer to **5.4 System Clock Oscillator.**

24.2.2 On-chip oscillator characteristics

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator oscillation clock frequency Notes 1, 2	f_{IH}		1.25		20	MHz
High-speed on-chip oscillator oscillation clock frequency accuracy		T_A = -20 to +85°C	-2.0		+2.0	%
		T_A = -40 to -20°C	-3.0		+3.0	%
Low-speed on-chip oscillator oscillation clock frequency	f_{IL}			15		kHz
Low-speed on-chip oscillator oscillation clock frequency accuracy			-15		+15	%

Notes 1. High-speed on-chip oscillator frequency is selected by bits 0 to 2 of option byte (000C2H).

2. This only indicates the oscillator characteristics. Refer to AC Characteristics for instruction execution time.

24.3 DC Characteristics

24.3.1 Pin characteristics

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for 10-pin products: P00 to P04, P40 16-pin products: P00 to P07, P40, P41				-10.0 Note 2	mA
		Total of 10-pin products: P40 16-pin products: P40, P41 (When duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			-20.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			-4.0	mA
			2.0 V ≤ V_{DD} < 2.7 V			-3.0	mA
		Total of 10-pin products: P00 to P04 16-pin products: P00 to P07 (When duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			-60.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			-12.0	mA
			2.0 V ≤ V_{DD} < 2.7 V			-9.0	mA
		Total of all pins (When duty ≤ 70% Note 3)				-80.0	mA
Output current, low Note 4	I_{OL1}	Per pin for 10-pin products: P00 to P04, P40 16-pin products: P00 to P07, P40, P41				20.0 Note 2	mA
		Total of 10-pin products: P40 16-pin products: P40, P41 (When duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			40.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			6.0	mA
			2.0 V ≤ V_{DD} < 2.7 V			1.2	mA
		Total of 10-pin products: P00 to P04 16-pin products: P00 to P07 (When duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			80.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			12.0	mA
			2.0 V ≤ V_{DD} < 2.7 V			2.4	mA
		Total of all pins (When duty ≤ 70% Note 3)				120.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from the V_{DD} pin to an output pin.

2. Do not exceed the total current value.

3. This is the output current value under conditions where the duty factor ≤ 70%.
The output current value when the duty factor > 70% can be calculated with the following expression (when changing the duty factor to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)
  <Example> Where n = 80 % and I_{OH} = - 10.0 mA
  Total output current of pins = (- 10.0 × 0.7)/(80 × 0.01) ≅ - 8.7 mA
- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)
  <Example> Where n = 80 % and I_{OL} = 10.0 mA
  Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≅ 8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

4. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

Caution P00, P01, P06, and P07 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port.

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}			0.8 V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}			0		0.2 V_{DD}	V
Output voltage, high Note 1	V_{OH1}	4.0 V ≤ V_{DD} ≤ 5.5 V	I_{OH} = -10 mA	V_{DD} - 1.5			V
			I_{OH} = -3.0 mA	V_{DD} - 0.7			V
		2.7 V ≤ V_{DD} ≤ 5.5 V	I_{OH} = -2.0 mA	V_{DD} - 0.6			V
		2.0 V ≤ V_{DD} ≤ 5.5 V	I_{OH} = -1.5 mA	V_{DD} - 0.5			V
Output voltage, low Note 2	V_{OL1}	4.0 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 20 mA			1.3	V
			I_{OL} = 8.5 mA			0.7	V
		2.7 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 3.0 mA			0.6	V
			I_{OL} = 1.5 mA			0.4	V
		2.0 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 0.6 mA			0.4	V
Input leakage current, high	I_{LIH1}	P00 to P07, P40, P41, P125, P137 V_I = V_{DD}				1	μA
	I_{LIH2}	P121, P122 (X1, X2, EXCLK) V_I = V_{DD}	In input port or external clock input			1	
			In resonator connection			10	
Input leakage current, low	I_{LIL1}	P00 to P07, P40, P41, P125, P137 V_I = V_{SS}				-1	μA
	I_{LIL2}	P121, P122 (X1, X2, EXCLK) V_I = V_{SS}	In input port or external clock input			-1	
			In resonator connection			-10	
On-chip pull-up resistance	R_U	V_I = V_{SS}		10	20	100	kΩ

Notes 1. The value under the condition which satisfies the high-level output current (I_{OH1}).

2. The value under the condition which satisfies the low-level output current (I_{OL1}).

Caution The maximum value of V_{IH} of P00, P01, P06, and P07 is V_{DD} even in N-ch open-drain mode. P00, P01, P06, and P07 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port.

24.3.2 Supply current characteristics

(1) Flash ROM: 1 and 2 KB of 10-pin products

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	Basic operation	f_{IH} = 20 MHz	V_{DD} = 3.0 V, 5.0 V		0.91		mA
			Normal operation	f_{IH} = 20 MHz	V_{DD} = 3.0 V, 5.0 V		1.57	2.04	
				f_{IH} = 5 MHz	V_{DD} = 3.0 V, 5.0 V		0.85	1.15	
	I_{DD2} Note 2	HALT mode		f_{IH} = 20 MHz	V_{DD} = 3.0 V, 5.0 V		350	820	μA
				f_{IH} = 5 MHz	V_{DD} = 3.0 V, 5.0 V		290	600	
	I_{DD3} Note 3	STOP mode		V_{DD} = 3.0 V			0.56	2.00	μA

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, I/O port, and on-chip pull-up/pull-down resistors.

2. During HALT instruction execution by flash memory.

3. Not including the current flowing into the watchdog timer.

Remarks 1. f_{IH}: High-speed on-chip oscillator clock frequency

2. Temperature condition of the typical value is T_A = 25°C

(2) Flash ROM: 4 KB of 10-pin products, and 16-pin products

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	Basic operation	f_{IH} = 20 MHz Note 4	V_{DD} = 3.0 V, 5.0 V		0.92		mA
			Normal operation	f_{IH} = 20 MHz Note 4	V_{DD} = 3.0 V, 5.0 V		1.59	2.14	
				f_{IH} = 5 MHz Note 4	V_{DD} = 3.0 V, 5.0 V		0.87	1.20	
				f_{MX} = 20 MHz Notes 5, 6 V_{DD} = 3.0 V, 5.0 V	Square wave input		1.43	1.93	
					Resonator connection		1.54	2.13	
				f_{MX} = 5 MHz Notes 5, 6 V_{DD} = 3.0 V, 5.0 V	Square wave input		0.67	1.02	
					Resonator connection		0.72	1.12	
	I_{DD2} Note 2	HALT mode		f_{IH} = 20 MHz Note 4	V_{DD} = 3.0 V, 5.0 V		360	900	μA
				f_{IH} = 5 MHz Note 4	V_{DD} = 3.0 V, 5.0 V		310	660	
				f_{MX} = 20 MHz Notes 5, 6 V_{DD} = 3.0 V, 5.0 V	Square wave input		200	700	
					Resonator connection		300	900	
				f_{MX} = 5 MHz Notes 5, 6 V_{DD} = 3.0 V, 5.0 V	Square wave input		100	440	
					Resonator connection		150	540	
	I_{DD3} Note 3	STOP mode		V_{DD} = 3.0 V			0.61	2.25	μA

Notes 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the A/D converter, comparator (16-pin products only), I/O port, and on-chip pull-up/pull-down resistors.

2. During HALT instruction execution by flash memory.

3. Not including the current flowing into the 12-bit interval timer and watchdog timer.

4. When the high-speed system clock is stopped.

5. When the high-speed on-chip oscillator is stopped.

6. 16-pin products only

Remarks 1. f_{IH}: High-speed on-chip oscillator clock frequency

2. f_{MX}: High-speed system clock frequency (X1 clock oscillator frequency or external main system clock frequency)

3. Temperature condition of the typical value is T_A = 25°C

(3) Peripheral Functions (Common to all products)

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.30		μA
12-bit interval timer operating current	I_{TMKA} Notes 1, 2, 3				0.01		μA
Watchdog timer operating current	I_{WDT} Notes 1, 4				0.01		μA
A/D converter operating current	I_{ADC} Notes 1, 5	When conversion at maximum speed	V_{DD} = 5.0 V		1.30	1.90	mA
			V_{DD} = 3.0 V		0.50		mA
Comparator operating current	I_{CMP} Notes 1, 6	In high-speed mode	V_{DD} = 5.0 V		6.50		μA
		In low-speed mode	V_{DD} = 5.0 V		1.70		μA
Internal reference voltage operating current	I_{VREG} Note 1				10		μA

Notes 1. Current flowing to V_{DD}.

2. When high speed on-chip oscillator and high-speed system clock are stopped.

3. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1}, I_{DD2} or I_{DD3} and I_{FIL} and I_{TMKA}, when the 12-bit interval timer is in operation.

4. Current flowing only to the watchdog timer (excluding the operating current of the low-speed on-chip oscillator). The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{FIL} and I_{WDT} when the watchdog timer is in operation.

5. Current flowing only to the A/D converter. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

6. Current flowing only to the comparator. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{CMP} when the comparator is in operation.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. Temperature condition of the typical value is T_A = 25°C

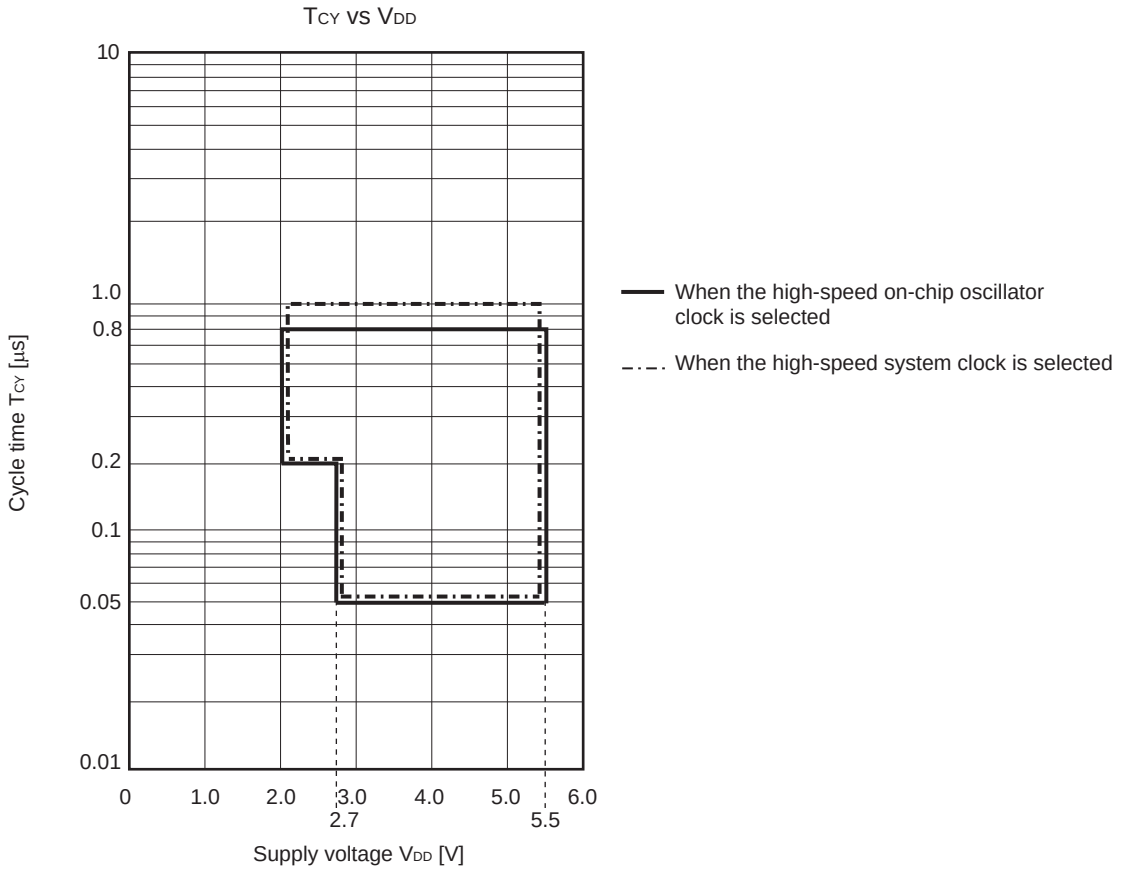
24.4 AC Characteristics

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

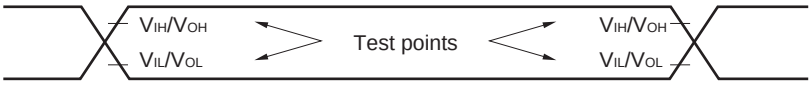
Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	When high-speed on-chip oscillator clock (f_{IH}) is selected	2.7 V ≤ V_{DD} ≤ 5.5 V	0.05		0.8	μs
			2.0 V ≤ V_{DD} < 2.7 V	0.2		0.8	μs
		When high-speed system clock (f_{MX}) is selected	2.7 V ≤ V_{DD} ≤ 5.5 V	0.05		1.0	μs
			2.0 V ≤ V_{DD} < 2.7 V	0.2		1.0	μs
External system clock frequency	T_{EX}		2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20	MHz
			2.0 V ≤ V_{DD} < 2.7 V	1.0		5	MHz
External system clock input high-level width, low-level width	T_{EXH}, T_{EXL}		2.7 V ≤ V_{DD} ≤ 5.5 V	24			ns
			2.0 V ≤ V_{DD} < 2.7 V	95			ns
TI00 to TI03 input high-level width, low-level width	t_{TIH}, t_{TIL}	Noise filter is not used		1/f_{MCK} + 10			ns
TO00 to TO03 output frequency	f_{TO}	4.0 V ≤ V_{DD} ≤ 5.5 V				10	MHz
		2.7 V ≤ V_{DD} < 4.0 V				5	MHz
		2.0 V ≤ V_{DD} < 2.7 V				2.5	MHz
PCLBUZ0 output frequency	f_{PCL}	4.0 V ≤ V_{DD} ≤ 5.5 V				10	MHz
		2.7 V ≤ V_{DD} < 4.0 V				5	MHz
		2.0 V ≤ V_{DD} < 2.7 V				2.5	MHz
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			μs

Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the timer clock select register 0 (TPS0) and the CKS0n1 bit of timer mode register 0nH (TMR0nH). n: Channel number (n = 0 to 3))

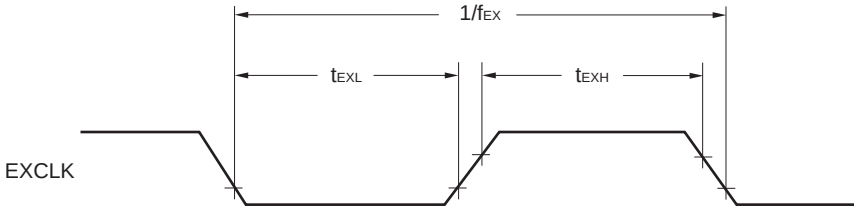
Minimum Instruction Execution Time during Main System Clock Operation



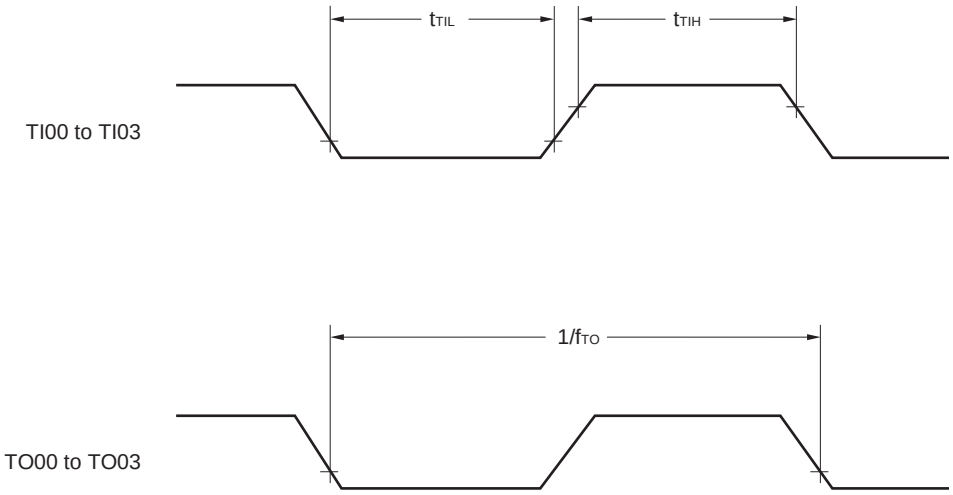
AC Timing Test Points



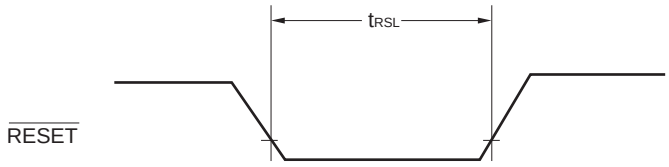
External System Clock Timing



TI/TO Timing

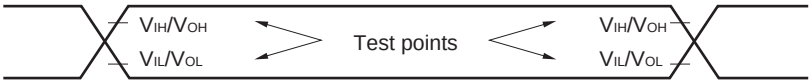


$\overline{\text{RESET}}$ Input Timing



24.5 Serial Interface Characteristics

AC Timing Test Points



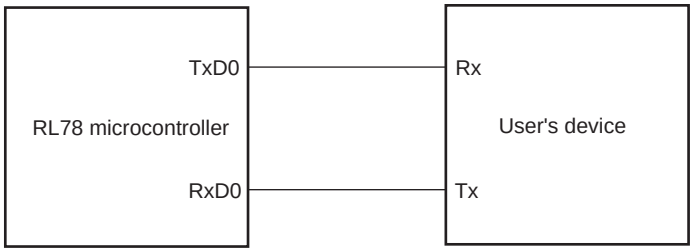
24.5.1 Serial array unit

(1) UART mode

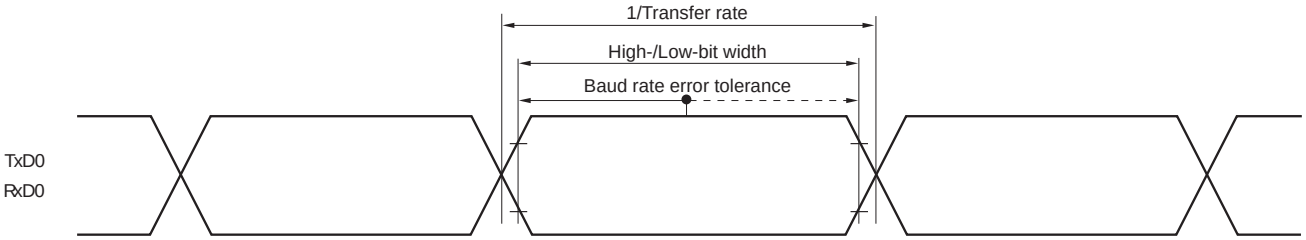
(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{CLK} = f_{MCK} = 20 MHz			3.3	Mbps

UART mode connection diagram



UART mode bit width (reference)



Remark f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register 0 (SPS0) and the CKS0n bit of the serial mode register 0nH (SMR0nH). n: Channel number (n = 0, 1))

(2) Simplified SPI (CSI) mode (master mode, SCKp... internal clock output)

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V	200			ns
			2.0 V ≤ V_{DD} ≤ 5.5 V	800			ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 18			ns
		2.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 50			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 5.5 V		47			ns
		2.0 V ≤ V_{DD} ≤ 5.5 V		110			ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}			19			ns
Delay time from SCKp↓ to SOp output Note 2	t_{KSO1}	C = 30 pF Note 3				25	ns

Notes 1. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp setup time becomes "to SCKp↓" and SIp hold time becomes "from SCKp↓" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The delay time to SOp output becomes "from SCKp↑" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

3. C is the load capacitance of the SCKp and SOp output lines.

(3) Simplified SPI (CSI) mode (slave mode, SCKp... external clock input)

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 5.5 V	f_{MCK} > 16 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}			ns
		2.0 V ≤ V_{DD} ≤ 5.5 V		6/f_{MCK}			ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	2.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2 - 18			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 20			ns
		2.0 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 30			ns
SIp hold time (from SCKp↑) Note 1	t_{KSI2}	2.0 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 31			ns
Delay time from SCKp↓ to SOp output Note 2	t_{KSO2}	C = 30 pF Note 3	2.7 V ≤ V_{DD} ≤ 5.5 V			2/f_{MCK} + 50	ns
			2.0 V ≤ V_{DD} ≤ 5.5 V			2/f_{MCK} + 110	ns

Notes 1. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp setup time becomes "to SCKp↓" and the SIp hold time becomes "from SCKp↓" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

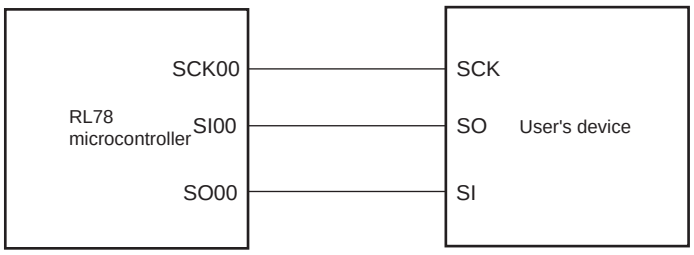
2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The delay time to SOp output becomes "from SCKp↑" when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

3. C is the load capacitance of the SOp output lines.

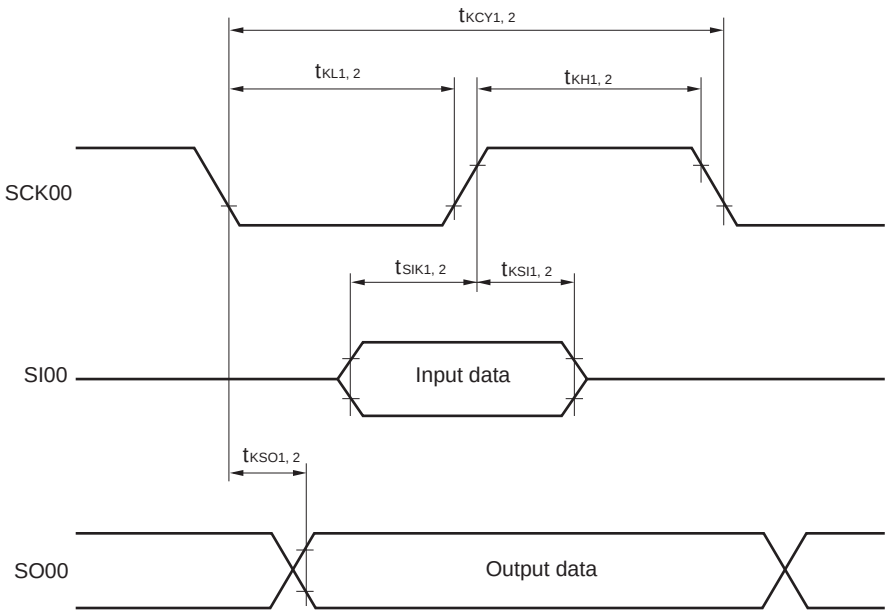
Remarks 1. p: CSI number (p = 00, 01), n: Channel number (n = 0, 1)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register 0 (SPS0) and the CKS0n bit of the serial mode register 0nH (SMR0nH). n: Channel number (n = 0, 1))

Simplified SPI (CSI) mode connection diagram



Simplified SPI (CSI) mode serial transfer timing
(When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1.)



Remark p: CSI number (p = 00, 01), n: Channel number (n = 0, 1)

(4) Simplified I²C mode

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

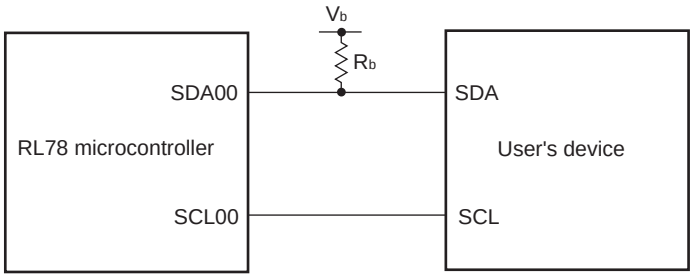
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	C_b = 100 pF, R_b = 3 kΩ		400 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	C_b = 100 pF, R_b = 3 kΩ	1150		ns
Hold time when SCLr = "H"	t_{HIGH}	C_b = 100 pF, R_b = 3 kΩ	1150		ns
Data setup time (reception)	$t_{SU: DAT}$	C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK} + 145 Note 2		ns
Data hold time (transmission)	$t_{HD: DAT}$	C_b = 100 pF, R_b = 3 kΩ	0	355	ns

Notes 1. The value must also be equal to or less than f_{MCK}/4.

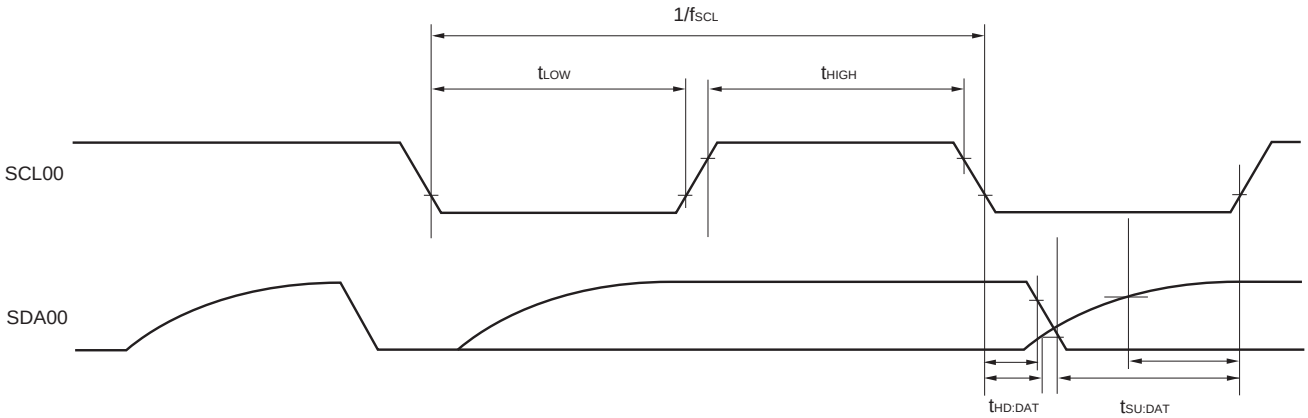
2. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

Caution Select the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin by using the port output mode register 0 (POM0).

Simplified I²C mode connection diagram



Simplified I²C mode serial transfer timing



Remarks 1. R_b [Ω]: Communication line (SDAr) pull-up resistance,
C_b [F]: Communication line (SCLr, SDAr) load capacitance

2. r: IIC number (r = 00)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the serial clock select register 0 (SPS0) and the CKS0n bit of the serial mode register 0nH (SMR0nH). n: Channel number (n = 0))

24.5.2 Serial interface IICA

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

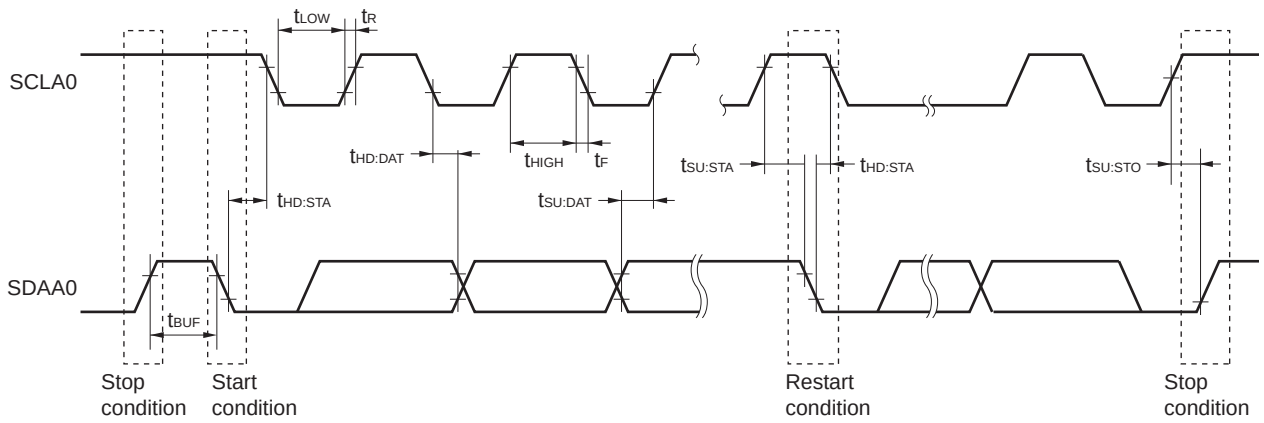
Parameter	Symbol	Conditions	Standard Mode		Fast Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz			0	400	kHz
		Standard mode: f_{CLK} ≥ 1 MHz	0	100			kHz
Setup time of restart condition	$t_{SU:STA}$		4.7		0.6		μs
Hold time[Note 1]	$t_{HD:STA}$		4.0		0.6		μs
Hold time when SCLA0 = "L"	t_{LOW}		4.7		1.3		μs
Hold time when SCLA0 = "H"	t_{HIGH}		4.0		0.6		μs
Data setup time (reception)	$t_{SU:DAT}$		250		100		ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0	3.45	0	0.9	μs
Setup time of stop condition	$t_{SU:STO}$		4.0		0.6		μs
Bus-free time	t_{BUF}		4.7		1.3		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.
2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

Remark The maximum value of Cb (communication line capacitance) and the value of Rb (communication line pull-up resistor) at that time in each mode are as follows.

Standard mode: C_b = 400 pF, R_b = 2.7 kΩ
Fast mode: C_b = 200 pF, R_b = 1.7 kΩ

IICA serial transfer timing



24.6 Analog Characteristics

24.6.1 A/D converter characteristics

(Target pin: ANI0 to ANI6, internal reference voltage)
(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Notes 1, 2, 3	AINL	10-bit resolution	V_{DD} = 5 V		±1.7	±3.1	LSB
			V_{DD} = 3 V		±2.3	±4.5	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI0 to ANI6	2.7 V ≤ V_{DD} ≤ 5.5 V	3.4		18.4	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V Note 5	4.6		18.4	μs
		10-bit resolution Target pin: internal reference voltage Note 6	2.4 V ≤ V_{DD} ≤ 5.5 V	4.6		18.4	μs
Zero-scale error Notes 1, 2, 3, 4	E_{ZS}	10-bit resolution	V_{DD} = 5 V			±0.19	%FSR
			V_{DD} = 3 V			±0.39	%FSR
Full-scale error Notes 1, 2, 3, 4	E_{FS}	10-bit resolution	V_{DD} = 5 V			±0.29	%FSR
			V_{DD} = 3 V			±0.42	%FSR
Integral linearity error Notes 1, 2, 3	ILE	10-bit resolution	V_{DD} = 5 V			±1.8	LSB
			V_{DD} = 3 V			±1.7	LSB
Differential linearity error Notes 1, 2, 3	DLE	10-bit resolution	V_{DD} = 5 V			±1.4	LSB
			V_{DD} = 3 V			±1.5	LSB
Analog input voltage	V_{AIN}	Target pin: ANI0 to ANI6		0		V_{DD}	V
		Target pin: internal reference voltage Note 6		V_{REG} Note 7			V

Notes 1. TYP. Value is the average value at T_A = 25°C. MAX. value is the average value ±3σ at normal distribution.
2. These values are the results of characteristic evaluation and are not checked for shipment.
3. Excludes quantization error (±1/2 LSB).
4. This value is indicated as a ratio (%FSR) to the full-scale value.
5. Set the LV0 bit in the A/D converter mode register 0 (ADM0) to 0 when conversion is done in the operating voltage range of 2.4 V ≤ V_{DD} < 2.7 V.
6. Set the LV0 bit in the A/D converter mode register 0 (ADM0) to 0 when the internal reference voltage is selected as the target for conversion.
7. Refer to **24.6.3 Internal reference voltage characteristics**.

Cautions 1. Arrange wiring and insert the capacitor so that no noise appears on the power supply/ground line.
2. Do not allow any pulses that rapidly change such as digital signals to be input/output to/from the pins adjacent to the conversion pin during A/D conversion.
3. Note that the internal reference voltage cannot be used as the reference voltage of the comparator when the internal reference voltage is selected as the target for A/D conversion.

24.6.2 Comparator characteristics

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage range	I_{VREF}	IVREF0 pin input (when C0VFR bit = 0)		0		V_{DD} - 1.4	V
		Internal reference voltage (when C0VRF bit = 1) Note 1		V_{REG} Note 2			V
	I_{VCMP}	IVCMP0 pin input		-0.3		V_{DD} + 0.3	V
Output delay	t_d	V_{DD} = 3.0 V, input slew rate > 50 mV/µs	High-speed mode			0.5	µs
			Low-speed mode		2.0		µs
Operation stabilization wait time	t_{CMP}			100			µs

Notes 1. When the internal reference voltage is selected as the reference voltage of the comparator, the internal reference voltage cannot be used as the target for A/D conversion.

2. Refer to **24.6.3 Internal reference voltage characteristics**.

24.6.3 Internal reference voltage characteristics

(T_A = –40 to +85°C, 2.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Internal reference voltage	V_{REG}		0.74	0.815	0.89	V
Operation stabilization wait time	t_{AMP}	When A/D converter is used (ADS register = 07H)	5			µs

Note The internal reference voltage cannot be simultaneously used by the A/D converter and the comparator; only one of them must be selected.

<R> **24.6.4 SPOR circuit characteristics**

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Power supply voltage level	V_{SPOR0}	The power supply voltage is rising.	4.08	4.28	4.45	V
			The power supply voltage is falling.	4.00	4.20	4.37	V
		V_{SPOR1}	The power supply voltage is rising.	2.76	2.90	3.02	V
			The power supply voltage is falling.	2.70	2.84	2.96	V
		V_{SPOR2}	The power supply voltage is rising.	2.44	2.57	2.68	V
			The power supply voltage is falling.	2.40	2.52	2.62	V
		V_{SPOR3}	The power supply voltage is rising.	2.05	2.16	2.25	V
			The power supply voltage is falling.	2.00	2.11	2.20	V
Minimum pulse width Note		T_{LSPW}		300			μs

Note Time required for the reset operation by the SPOR when V_{DD} becomes under V_{SPOR}.

Caution Set the detection voltage (V_{SPOR}) in the operating voltage range. The operating voltage range depends on the setting of the user option byte (000C2H). The operating voltage range is as follows:
When the CPU operating frequency is from 1 MHz to 20 MHz: V_{DD} = 2.7 to 5.5 V
When the CPU operating frequency is from 1 MHz to 5 MHz: V_{DD} = 2.0 to 5.5 V

24.6.5 Power supply voltage rising slope characteristics

(T_A = –40 to +85°C, V_{SS} = 0 V)

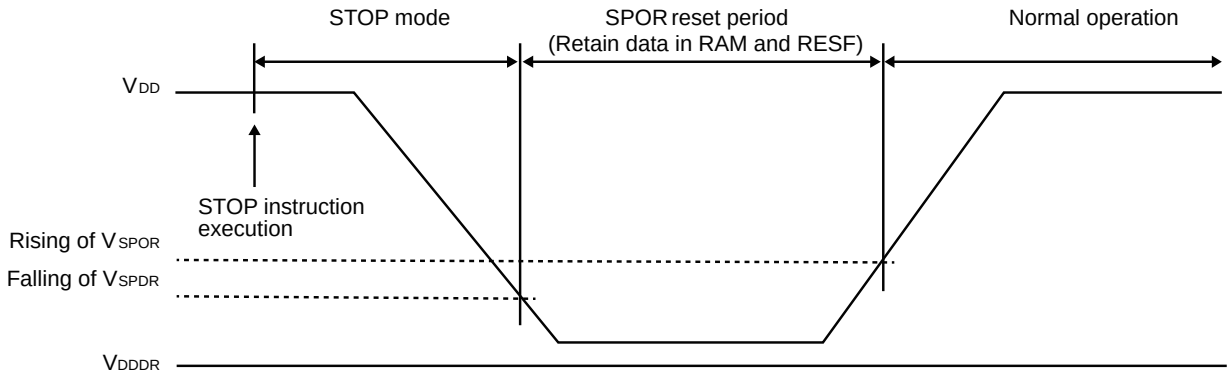
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

24.7 RAM Data Retention Characteristics

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.9		5.5	V

Caution Data in RAM is retained until the power supply voltage becomes under the minimum value of the data retention power supply voltage (V_{DDDR}). Note that data in the RESF register might not be cleared even if the power supply voltage becomes under the minimum value of the data retention power supply voltage (V_{DDDR}).



24.8 Flash Memory Programming Characteristics

(T_A = 0 to + 40°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Code flash memory rewritable times Notes 1, 2, 3	C_{erwr}	Retained for 20 years.	T_A = +85°C	1000			Times

Notes 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

2. When using flash memory programmer.

3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

24.9 Dedicated Flash Memory Programmer Communication (UART)

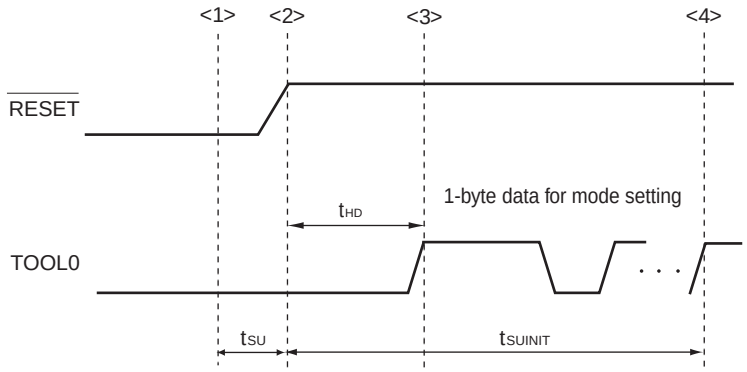
(T_A = 0 to + 40°C, 4.5 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate				115,200		bps

Remark The transfer rate during flash memory programming is fixed to 115,200 bps.

24.10 Timing of Entry to Flash Memory Programming Modes

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	SPOR reset must be released before the external reset is released.			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t_{SU}	SPOR reset must be released before the external reset is released.	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released	t_{HD}	SPOR reset must be released before the external reset is released.	1			ms



<1> The low level is input to the TOOL0 pin.

<2> The external reset is released (SPOR reset must be released before the external reset is released.).

<3> The TOOL0 pin is set to the high level.

<4> Setting of entry to the flash memory programming mode by UART reception is completed.

Remark t_{SUINIT}: Communication for the initial setting must be completed within 100 ms after the external reset is released during this period.

t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level

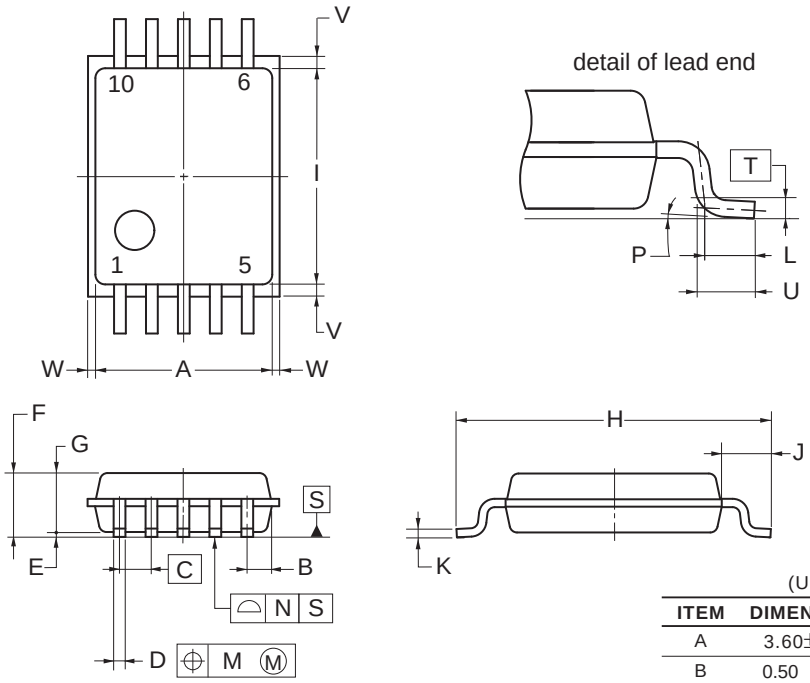
t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released

CHAPTER 25 PACKAGE DRAWINGS

25.1 10-pin products

R5F10Y17ASP, R5F10Y16ASP, R5F10Y14ASP
R5F10Y17DSP , R5F10Y16DSP, R5F10Y14DSP

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP10-4.4x3.6-0.65	PLSP0010JA-A	P10MA-65-CAC-2	0.05



(UNIT:mm)

ITEM	DIMENSIONS
A	3.60±0.10
B	0.50
C	0.65 (T.P.)
D	0.24±0.08
E	0.10±0.05
F	1.45 MAX.
G	1.20±0.10
H	6.40±0.20
I	4.40±0.10
J	1.00±0.20
K	$0.17^{+0.08}_{-0.07}$
L	0.50
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25 (T.P.)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

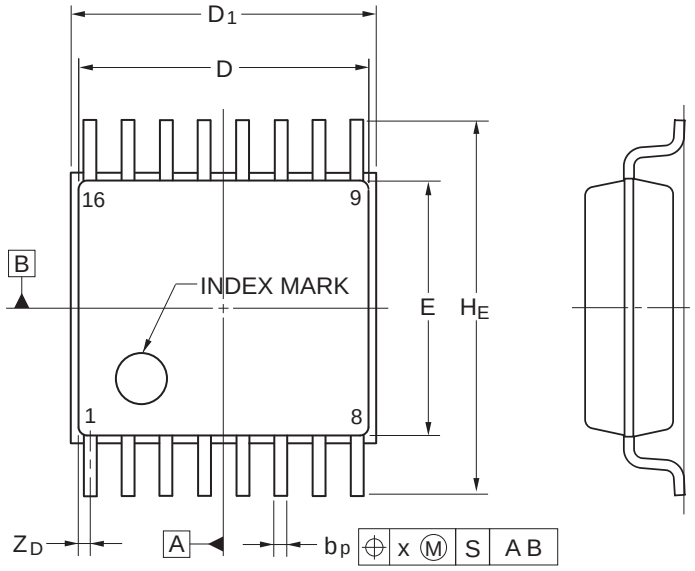
NOTE
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

©2012 Renesas Electronics Corporation. All rights reserved.

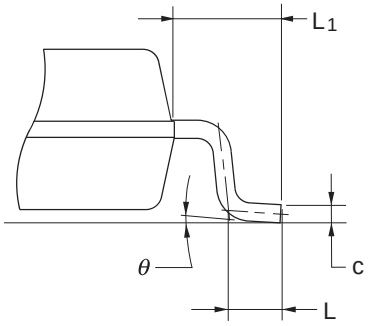
25.2 16-pin products

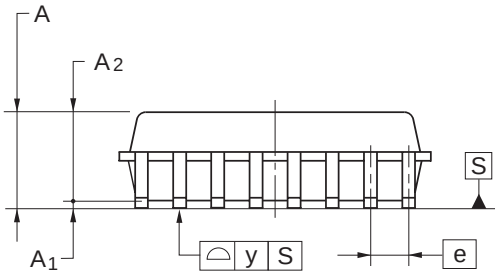
R5F10Y47ASP, R5F10Y46ASP, R5F10Y44ASP
R5F10Y47DSP, R5F10Y46DSP, R5F10Y44DSP

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-SSOP16-4.4x5-0.65	PRSP0016JC-B	P16MA-65-FAB-1	0.08

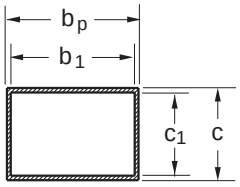


detail of lead end





Terminal cross section



Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	4.85	5.00	5.15
D_1	5.05	5.20	5.35
E	4.20	4.40	4.60
A_2	—	1.50	—
A_1	0.075	0.125	0.175
A	—	—	1.725
b_p	0.17	0.24	0.32
b_1	—	0.22	—
c	0.14	0.17	0.20
c_1	—	0.15	—
θ	0°	—	8°
H_E	6.20	6.40	6.60
e	—	0.65	—
x	—	—	0.13
y	—	—	0.10
Z_D	—	0.225	—
L	0.35	0.50	0.65
L_1	—	1.00	—

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

(1/2)

Page	Description	Classification
All		
-	"3-Wire Serial I/O" and "3-wire serial" were modified to " Simplified SPI"	(c)
-	The module name for CSI was changed to Simplified SPI	(c)
-	"wait" for IIC was modified to "clock stretch"	(c)
CHAPTER 1 OUTLINE		
p.2	Addition of Note 1 in **1 Features**	(c)
p.3	Modification of description in **Figure 1-1. Part Number, Memory Size, and Package of RL78/G10**	(b)
p.3	Modification of description in **Table 1-1. List of Ordering Part Numbers**	(b)
p.4	Addition of **Table 1-2. Alternate Function of 10-pin products**	(c)
p.5	Addition of **Table 1-3. Alternate Function of 16-pin products**	(c)
CHAPTER 3 CPU ARCHITECTURE		
p.44	Modification of title and description in **3.3.4 Register indirect addressing**	(a)
CHAPTER 4 PORT FUNCTIONS		
p.60	Modification of description in **4.2.1 Port 0**	(a)
p.64	Modification of description in **4.3.3 Pull-up resistor option registers 0, 4, 12 (PU0, PU4, PU12)**	(a)
p.70	Addition of Remark in **4.5.2 Register settings for alternate functions that do not use an output function**	(c)
p.72	Modification of description in **Table 4-5. Examples of Register And Output Latch Settings With Pin Functions (2/4)**	(a)
CHAPTER 5 CLOCK GENERATOR		
p.88	Modification of Caution 3 in **Figure 5-9. Format of High-Speed On-Chip Oscillator Frequency Selection Register (HOCODIV)**	(c)
CHAPTER 6 TIMER ARRAY UNIT		
p.175	Modification of **Figure 6-53. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MD0n0 = 0)**	(a)
CHAPTER 12 SERIAL ARRAY UNIT		
p.276	Addition of Note	(c)
CHAPTER 13 SERIAL INTERFACE IICA		
p.458	Modification of **Figure 13-29. Slave Operation Flowchart (1)**	(c)
CHAPTER 14 INTERRUPT FUNCTIONS		
p.512	Modification of description in **14.4.3 Multiple interrupt servicing**	(c)
CHAPTER 20 FLASH MEMORY		
p.551	Modification of description and deletion of **Remark** in **20.1 Serial Programming by Using Flash Memory Programmer**	(b)
p.551	Modification of description and addition of **Note** in **Table 20-1. Wiring Between RL78/G10 and Dedicated Flash Memory Programmer**	(b)
p.552	Modification of description in **Figure 20-1. Environment for Writing Program to Flash Memory**	(b)

Remark "Classification" in the above table classifies revisions as follows.

(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(2/2)

Page	Description	Classification
p.552	Modification of description and Notes in **Figure 20-2. Communication with Dedicated Flash Memory Programmer**	(b)
p.558	Modification of title and description in **20.5 Processing Time for Each Command When Dedicated Flash Memory Programmer is in Use (Reference Value)**	(b)
p.559	Addition of table and Remark in **Table 20-7. Processing Time for Each Command When PG-FP6 is in Use (Reference Value)**	(b)
CHAPTER 21 ON-CHIP DEBUG FUNCTION		
p.560	Modification of title and description in **21.1 Connecting E1, E2, E2 Lite, E20 On-chip Debugging Emulator**	(b)
p.560	Modification of title and description in **Figure 21-1. Connection Example of E1, E2, E2 Lite, E20 On-chip Debugging Emulator and RL78 microcontroller**	(b)
p.561	Modification of title and description in **Figure 21-2. Connection Example of E1, E2, E2 Lite, E20 On-chip Debugging Emulator and RL78 microcontroller (When using to the alternative function of $\overline{\text{RESET}}$ pin)**	(b)
p.562	Modification of description in **21.3 Securing of User Resources**	(b)
p.562	Modification of **Figure 21-3. Memory Spaces Where Debug Monitor Programs Are Allocated** and Note3	(c)
CHAPTER 23 INSTRUCTION SET		
p.583	Modification of description in **Table 23-5. Operation List**	(a)
CHAPTER 24 ELECTRICAL SPECIFICATIONS		
p.605	Modification of description in **24.6.4 SPOR circuit characteristics**	(c)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

A.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/10)

Edition	Description	Chapter
Rev.3.11	Correction of **3.1.2 Mirror area**	**CHAPTER 3 CPU ARCHITECTURE**
	Addition of Note and correction of **Table 3-4. SFR List (1/2)**	
	Deletion of Processor mode control register in **Table 3-4. SFR List (2/2)**	
	Addition of Note 1 to **Table 3-5. Extended SFR (2nd SFR) List (1/2)**	
	Correction of **Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (1/2)**	**CHAPTER 5 CLOCK GENERATOR**
	Correction of **5.4.3 Low-speed on-chip oscillator**	
	Addition of description to **11.3.3 Comparator Filter Control Register (COMPFIR)**	**CHAPTER 11 COMPARATOR**
	Correction of **Figure 13-9. Format of IICA Control Register 01 (IICCTL01) (1/2)**	**CHAPTER 13 SERIAL INTERFACE IICA**
	Correction of **Table 23-5. Operation List (13/17)**	**CHAPTER 23 INSTRUCTION SET**

(2/10)

Edition	Description	Chapter
Rev.3.10	Deletion of under development	Throughout
	Corrected **Table 1-1. List of Ordering Part Numbers**	CHAPTER 1 OUTLINE
	Modification of **1.3 Pin Configuration (Top View)**	
	Addition of Caution to **Figure 2-5. Pin Block Diagram for Pin Type 7-1-2**	CHAPTER 2 PIN FUNCTIONS
	Addition of Caution to **Figure 2-7. Pin Block Diagram for Pin Type 7-3-2**	
	Addition of description to **Figure 5-4. Format of Clock Operation Status Control Register (CSC)**	CHAPTER 5 CLOCK GENERATOR
	Modification of **Table 5-4. Changing CPU Clock**	
	Addition of description to **5.6.6 Conditions before clock oscillation is stopped**	
	Addition of description to **6.4.2 Basic rules of 8-bit timer operation function (only channels 1 and 3)**	CHAPTER 6 TIMER ARRAY UNIT
	Corrected Note of **Figure 6-77. Example of Set Contents of Registers for PWM Output Function (Slave Channel) (1/2)**	
	Corrected **Figure 10-17. Overall Error** to **Figure 10-22. Differential Linearity Error**	CHAPTER 10 A/D CONVERTER
	Addition of description to **13.3.6 IICA low-level width setting register 0 (IICWL0)**	CHAPTER 13 SERIAL INTERFACE IICA
	Corrected **13. 5. 17 (1) (c) (ii) When WTIM0 = 1**	
	Corrected Caution 2 of **Figure 14-3. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L) (16-pin product)**	CHAPTER 14 INTERRUPT FUNCTIONS
Rev.3.00	Addition of industrial applications in **Figure 1-1 Part Number, Memory Size, and Package of RL78/G10**	CHAPTER 1 OUTLINE
	Addition of industrial applications in **Table 1-1 List of Ordering Part Numbers**	
	Addition of description to pin configuration in **1.3.1 10-pin products** and **1.3.2 16-pin products**	
	Addition of **5.7 Resonator and Oscillator Constants**	CHAPTER 5 CLOCK GENERATOR
	Addition of figure in **12.2.1 Shift register**	CHAPTER 12 SERIAL ARRAY UNIT
	Modification of caution in **Figure 12-14 Format of Serial Output Register 0 (SO0)**	
	Modification of caution in **Figure 12-15 Format of Serial Clock Output Register 0 (CKO0)**	
	Addition of description in **Figure 13-27 Master Operation in Single-Master System**	CHAPTER 13 SERIAL INTERFACE IICA
	Addition of description in **Figure 13-28 Master Operation in Multi-Master System (1/3)**	
	Modification of description in **Figure 13-28 Master Operation in Multi-Master System (2/3)**	
	Addition of description in **Figure 13-29 Slave Operation Flowchart (1)**	
	Addition of description and remark 1 in **20.3.1 P40/TOOL0 pin**	CHAPTER 20 FLASH MEMORY
	Correction of error in **24.5.1 (3) CSI mode (slave mode, SCKp... external clock input)**	CHAPTER 24 ELECTRICAL SPECIFICATIONS
	Renamed to **24.7 RAM Data Retention Characteristics** and modification of figure	
	Addition of industrial application in **25.1 10-pin products**	CHAPTER 25 PACKAGE DRAWINGS

(3/10)

Edition	Description	Chapter
Rev.3.00	Addition of industrial application in **25.2 16-pin products** and modification of package drawing	CHAPTER 25 PACKAGE DRAWINGS
Rev.2.00	Modification of descriptions in 1.1 Features	CHAPTER 1 OUTLINE
	Modification of description in 1.2 List of Part Numbers	
	Modification of remark 2 in 1.3.1 10-pin products and 1.3.2 16-pin products	
	Addition of description of R5F10Y17ASP in 1.6 Outline of Functions	
	Modification of description in 2.1.1 10-pin products	CHAPTER 2 PIN FUNCTIONS
	Modification of description in 2.1.2 16-pin products	
	Modification of description in 2.2.1 Functions for each product	
	Modification of description in 2.2.2 Description of functions	
	Addition of Figure 2-8 in 2.4 Block Diagrams of Pins	
	Modification of description in Figure 3-1 Memory Map for the R5F10Y14ASP and R5F10Y44ASP to Figure 3-3 Memory Map for the R5F10Y17ASP and R5F10Y47ASP	CHAPTER 3 CPU ARCHITECTURE
	Addition of R5F10Y17ASP in Table 3-1 Internal ROM Capacity	
	Addition of specification of 16-pin products in Table 3-2	
	Modification of figure in 3.1.2	
	Modification of description and addition of caution in 3.1.3 Internal data memory space	
	Addition of description of R5F10Y17ASP in Table 3-3 Internal RAM Capacity	
	Modification of description in Figure 3-4 Correspondence Between Data Memory and Addressing	CHAPTER 3 CPU ARCHITECTURE
	Modification of description in (3) Stack pointer (SP) and addition of caution 2	
	Addition of registers related to the comparator in Table 3-4 SFR List	
	Modification of note 2 in Table 3-4 SFR List	
	Addition of A/D test register in Table 3-5 Extended SFR (2nd SFR) List	
	Addition of note 2 in Table 3-5 Extended SFR (2nd SFR) List	
	Modification of description in 4.2.1 Port 0 to 4.2.4 Port 13	CHAPTER 4 PORT FUNCTIONS
	Modification of caution in 4.3 Registers Controlling Port Function	
	Addition of caution in Figure 4-1 Format of Port Mode Registers 0, 4 (PM0, PM4)	
	Modification of note in 4.3.2 Port registers 0, 4, 12, 13 (P0, P4, P12, P13)	
	Addition of caution in Figure 4-2 Format of Port Registers 0, 4, 12, 13 (P0, P4, P12, P13)	
	Modification of description in 4.3.3 Pull-up resistor option registers 0, 4, 12 (PU0, PU4, PU12)	
	Modification of note and addition of caution in Figure 4-3 Format of Pull-up Resistor Option Registers 0, 4, 12 (PU0, PU4, PU12)	
	Addition of caution in 4.3.4 Port output mode register 0 (POM0)	
	Addition of caution in Figure 4-4 Format of Port Output Mode Register 0 (POM0)	
	Modification of caution 1 and addition of caution 2 in Figure 4-5 Format of Port Mode Control Register 0 (PMC0)	
	Modification of description and addition of caution 2 in Figure 4-6. Format of Peripheral I/O Redirection Register (PIOR)	
	Modification of description in 4.6.2 Notes on specifying the pin settings	

(4/10)

Edition	Description	Chapter
Rev.2.00	Modification of description in (1) Main system clock	CHAPTER 5 CLOCK GENERATOR
	Addition of cautions 1 to 3 in Figure 5-3 Format of System Clock Control Register (CKC)	
	Addition of caution in Figure 5-7 Format of Peripheral Enable Register 0 (PER0)	
	Addition of specification in (2) CPU clock changing from high-speed system clock (B) to high-speed on-chip oscillator clock (A)	
	Addition of description	CHAPTER 6 TIMER ARRAY UNIT
	Addition of description in (2) Two-channel input with one-shot pulse output function (16-pin products only)	
	Modification of Figure 6-1 Entire Configuration of Timer Array Unit	
	Block diagram of (b) Channels 1 and 3 was divided into (b) Channel 1 and (c) Channel 3 in Figure 6-2 Internal Block Diagram of Channel of Timer Array Unit	
	Modification of caution 2 in Figure 6-6 Format of Peripheral Enable Register 0 (PER0)	
	Modification of description and addition of caution in Figure 6-8 Format of Timer Mode Register 0n (TMR0n) (2/3)	
	Modification of description in 6.3.5 Timer channel enable status register 0 (TE0, TEH0 (8-bit mode))	
	Modification of description in 6.3.8 Timer output enable register 0 (TOE0)	
	Modification of description and remark in 6.3.11 Timer output mode register 0 (TOM0)	
	Modification of description and addition of caution in 6.3.12 Noise filter enable register 1 (NFEN1)	
	Addition of 6.3.13 Input switch control register (ISC)	
	Modification of description in 6.3.14 Registers controlling port functions of pins to be used for timer I/O	
	Modification of description in 6.4.2 Basic rules of 8-bit timer operation function (only channels 1 and 3)	
	Modification of description in 6.6.1 TO0n pin output circuit configuration	
	Addition of description in 6.7 Timer Input (TI0n) Control	
	Modification of description in 6.8.1 Operation as interval timer/square wave output	
	Modification of description in Figure 6-43 Procedure for Operating Interval Timer/Outputting Square Wave	
	Modification of description in 6.8.2 Operation as external event counter	
	Modification of description in Figure 6-47 Procedure for Operating External Event Counter	
	Modification of description in Figure 6-51 Procedure for Operating Frequency Divider	
	Modification of description in 6.8.4 Operation as input pulse interval measurement	
	Modification of description in Figure 6-55 Procedure for Measuring Input Pulse Interval	
	Modification of description in Figure 6-59 Procedure for Measuring Input Signal High-/Low-Level Width	
	Addition of caution in 6.8.6 Operation as delay counter	
	Modification of description in Figure 6-63 Procedure for Operating Delay Counter	
	Addition of caution in 6.9.1 Operation as one-shot pulse output	

(5/10)

Edition	Description	Chapter
Rev.2.00	Modification of description in Figure 6-68 Procedure for Outputting One-Shot Pulse	CHAPTER 6 TIMER ARRAY UNIT
	Addition of description in 6.9.2 Two-channel input with one-shot pulse output function	
	Modification of description in Figure 6-78 Procedure for Using PWM Output Function	
	Modification of description in 6.9.4 Operation as multiple PWM output function	
	Modification of description in Figure 6-83 Procedure for Using Multiple PWM Output Function (Output Two Types of PWMs)	
	Modification of description in Figure 7-1 Block Diagram of 12-bit Interval Timer	CHAPTER 7 12-BIT INTERVAL TIMER
	Modification of cautions 1 and 3 in Figure 7-2 Format of Peripheral Enable Register 0 (PER0)	
	Modification of description in 7.3.3 Interval timer control register (ITMCH, ITMCL)	
	Modification of description and cautions 1 to 4 in Figure 7-4 Format of Interval Timer Control Register (ITMCH, ITMCL)	
	Modification of description in Figure 7-5 12-bit Interval Timer Operation Timing	
	Modification of description in 7.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode	
	Addition of 16-pin products in Figure 8-1 Block Diagram of Clock Output/Buzzer Output Controller	CHAPTER 8 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER
	Modification of description in 8.3.2 Registers controlling port functions of clock output/buzzer output pin	
	Modification of description in Figure 9-1 Block Diagram of Watchdog Timer	CHAPTER 9 WATCHDOG TIMER
	9.4.2 Setting overflow time of watchdog timer was modified to 9.4.2 Setting time of watchdog timer	
	Addition of description and note	CHAPTER 10 A/D CONVERTER
	Modification of description in Figure 10-1 Block Diagram of A/D Converter	
	Addition of description in (1) ANI0 to ANI6 pins	
	Modification of description in Figure 10-5 A/D Converter Sampling and A/D Conversion Timing	
	Modification of caution in Figure 10-7 Format of A/D Conversion Result Higher-Order Bit Storage Register (ADCRH)	
	Addition of note 2 and modification of cautions 1 to 4 in Figure 10-10 Format of Analog Input Channel Specification Register (ADS)	
	Addition of description in 10.3.7 A/D test register (ADTES)	
	Addition of description in 10.3.8 Registers controlling port function of analog input pins	
	Modification of description in Figure 10-12 Conversion Operation of A/D Converter	
	Addition of description and figure in 10.7.2 Setting up A/D conversion of the internal reference voltage (16-pin products only)	
	Modification of description in 10.9.3 Conflicting operations	
	Addition of description	CHAPTER 11 COMPARATOR

(6/10)

Edition	Description	Chapter
Rev.2.00	Addition of description of 16-pin products	CHAPTER 12 SERIAL ARRAY UNIT
	Modification of description in Figure 12-1 Block Diagram of Serial Array Unit 0	
	Modification of description in (2) Serial data register 0nL (SDR0nL)	
	Modification of caution in Figure 12-5 Format of Serial Mode Register 0n (SMR0nH, SMR0nL) (1/2)	
	Modification of caution in Figure 12-5 Format of Serial Mode Register 0n (SMR0nH, SMR0nL) (2/2)	
	Modification of caution in Figure 12-6 Format of Serial Mode Register 0n (SCR0nH, SCR0nL) (1/2)	
	Modification of description in 12.3.5 Serial data register 0n (SDR0nH, SDR0nL)	
	Modification of caution in Figure 12-8 Format of Serial Flag Clear Trigger Register 0n (SIR0n)	
	Modification of caution in Figure 12-13 Format of Serial Output Enable Register 0 (SOE0)	
	Modification of caution in Figure 12-14 Format of Serial Output Register 0 (SO0)	
	Modification of caution in Figure 12-15 Format of Serial Clock Output Register (CKO0)	
	Modification of description in 12.3.16 Input switch control register (ISC)	
	Modification of description in 12.3.17 Registers controlling port functions of serial input/output pins	
	Modification of cautions 1 and 2 in Figure 12-20 Peripheral Enable Register 0 (PER0) Setting When Stopping Operation by Units	
	Addition of description in 12.5 Operation of 3-Wire Serial I/O (CSI00, CSI01) Communication	
	Modification of notes 1 to 3 in 12.5.4 Slave transmission	
	Modification of note in Table 12-3 Selection of Operation Clock For UART	
	Addition of description in 12.7 Operation of Simplified I2C (IIC00) Communication	
	Addition of Figure 12-98 Processing Procedure in Case of Overrun Error	
	Modification of description in Figure 13-1 Block Diagram of Serial Interface IICA	CHAPTER 13 SERIAL INTERFACE IICA
	Modification of note in Figure 13-4 Format of Slave Address Register 0 (SVA0)	
	Modification of cautions 1 and 2 in Figure 13-5 Format of Peripheral Enable Register 0 (PER0)	
	Addition of note 2 in Figure 13-6 Format of IICA Control Register 00 (IICCTL00) (3/4)	
	Addition of note in Figure 13-6 Format of IICA Control Register 00 (IICCTL00) (4/4)	
	Modification of description in Figure 13-9 Format of IICA Control Register 01 (IICCTL01) (2/2)	
	Modification of description in 13.3.8 Registers controlling port functions of IICA serial input/output pins	
	Modification of description in Figure 13-22 Flow When Setting WUP0 = 0 upon Address Match (Including Extension Code Reception)	
	Modification of description in (3) If other I2C communications are already in progress	
	Modification of description in Figure 13-27 Master Operation in Single-Master System	
	Modification of description in Figure 13-28 Master Operation in Multi-Master System (2/3) and (3/3)	
	Modification of description in Figure 13-29 Slave Operation Flowchart (1)	

(7/10)

Edition	Description	Chapter
Rev.2.00	Addition of specification and notes 1 to 3 in Table 14-2 Interrupt Source List (16-pin products)	CHAPTER 14 INTERRUPT FUNCTIONS
	Addition of specification, notes 1 and 2 in Table 14-4 Flags Corresponding to Interrupt Request Sources (16-pin products)	
	Addition of Figure 14-3 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L) (16-pin product)	
	Addition of Figure 14-5 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L) (16-pin product)	
	Addition of Figure 14-5 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L) (16-pin product)	
	Addition of description in 14.3.3 Priority specification flag registers (PR00L, PR00H, PR10L, PR10H, PR01L, PR11L)	
	Addition of Figure 14-7 Format of Priority Specification Flag Registers (PR00L, PR00H, PR10L, PR10H, PR01L, PR11L) (16-pin product)	
	Addition of remarks 1 and 2 in Figure 14-8 Format of External Interrupt Rising Edge Enable Register 0 (EGP0) and External Interrupt Falling Edge Enable Register 0 (EGN0)	
	Modification of description in 15.3.4 Registers controlling port functions of key interrupt input pins	CHAPTER 15 KEY INTERRUPT FUNCTION
	Modification of note 2 in Figure 16-3 STOP Mode Release by Interrupt Request Generation (2/2)	CHAPTER 16 STANDBY FUNCTION
	Modification of Figure 18-2 Timing of Internal Reset Signal Generation	CHAPTER 18 SELECTABLE POWER-ON-RESET CIRCUIT
	Addition of caution in 19.1 Functions of Option Bytes	CHAPTER 19 OPTION BYTE
	Addition of caution 2 in Figure 19-2 Format of User Option Byte (000C1H)	
	Modification of description in Figure 19-3 Format of User Option Byte (000C2H)	
	Addition of R5F10Y17 in Table 20-6 Processing Time of Each Command When Using PG-FP5 (Reference Values)	CHAPTER 20 FLASH MEMORY
	Addition of R5F10Y17 to note 1 in Figure 21-3 Memory Spaces Where Debug Monitor Programs Are Allocated	CHAPTER 21 ON-CHIP DEBUG FUNCTION
	Addition of addr5 in Table 23-2 Symbols in "Operation" Column	CHAPTER 23 INSTRUCTION SET
	Modification of note 2 in Table 23-5 Operation List (1/17) to (17/17)	
	Modification of description in 24.1 Absolute Maximum Ratings	CHAPTER 24 ELECTRICAL SPECIFICATIONS
	Modification of description in 24.2 Oscillator Characteristics	
	Modification of description, notes 1 to 4, and caution in 24.3.1 Pin characteristics	
	Addition of description, notes 1 to 6, and remarks 1 and 2 in (2) Flash ROM: 4 KB of 10-pin products, and 16-pin products	
	Addition of description, notes 1 to 6, and remarks 1 to 3 in (3) Peripheral Functions (Common to all products)	
	Modification of description in 24.4 AC Characteristics	
	Addition of figure of Minimum Instruction Execution Time during Main System Clock Operation	
	Addition of figure of External System Clock Timing	
	Modification of TI/TO Timing	
	Addition of description in 24.5.2 Serial interface IICA	

(8/10)

Edition	Description	Chapter
Rev.2.00	Modification of description and notes 1 to 6 in 24.6.1 A/D converter characteristics	CHAPTER 24 ELECTRICAL SPECIFICATIONS
	Addition of description, notes 1 and 2 in 24.6.2 Comparator characteristics	
	Addition of description and note in 24.6.3 Internal reference voltage characteristics	
	Addition of caution in 24.6.4 SPOR Circuit characteristics	
	Addition of figure in 24.6.6 Data retention power supply voltage characteristics	
	Addition of R5F10Y17ASP in 25.1 10-pin products	CHAPTER 25 PACKAGE DRAWINGS
	Modification of package drawing in 25.2 16-pin products	
Rev.1.00	Modification of description of the power supply voltage in 1.1 Features	CHAPTER 1 OUTLINE
	Modification of Figure 1-1 Classification of Part Number	
	Modification of description in 1.6 Outline of Functions and addition of note	
	Modification of error in 2.1.2 16-pin products	CHAPTER 2 PIN FUNCTIONS
	Modification of 2.2.2 Description of functions	
	Addition of caution in Figure 3-1 Memory Map for the R5F10Y14ASP and R5F10Y44ASP to Figure 3-3 Memory Map for the R5F10Y47ASP	CHAPTER 3 CPU ARCHITECTURE
	Modification of error in Figure 3-7 Format of Stack Pointer	
	Modification of error in 4.2.1 Port 0	CHAPTER 4 PORT FUNCTIONS
	Modification of error in Figure 4-4 Format of Port Output Mode Register 0 (POM0)	
	Addition of caution in Figure 4-6 Format of Peripheral I/O Redirection Register (PIOR)	
	Modification of error in Table 4-5 Examples of Register And Output Latch Settings With Pin Functions (1/4)	
	Modification of error in Table 4-5 Examples of Register And Output Latch Settings With Pin Functions (2/4)	
	Addition of note in 5.1 Functions of Clock Generator	CHAPTER 5 CLOCK GENERATOR
	Modification of error in 5.3.1 Clock operation mode control register (CMC)	
	Modification of error in 5.3.3 Clock operation status control register (CSC)	
	Modification of description in Figure 5-5 Format of Oscillation Stabilization Time Counter Status Register (OSTC)	
	Modification of description in Figure 5-6 Format of Oscillation Stabilization Time Select Register (OSTS)	
	Modification of value after reset in Figure 5-9 Format of High-Speed On-Chip Oscillator Frequency Selection Register (HOCODIV)	
	Addition of note in 5.5 Clock Generator Operation	
	Modification of Figure 5-12 Clock Generator Operation When Power Supply Voltage Is Turned On	
	Modification of 5.6.2 Example of setting X1 oscillation clock	
	Modification of Figure 5-13 CPU Clock Status Transition Diagram	
	Modification of remark in Table 5-5 Maximum Number of Clocks Required for $f_{IH} \leftrightarrow f_{MX}$	
	Modification of description of the timer array unit function in the beginning of the chapter	CHAPTER 6 TIMER ARRAY UNIT
	Addition of caution in Figure 6-3 Format of Timer/Counter Register 0n (TCR0n) (n = 0 to 3)	
	Addition of caution in Table 6-3 Timer/counter Register 0n (TCR0n) Read Value in Various Operation Modes	
	Modification of description in 6.2.2 Timer data register 0n (TDR0n) and addition of caution	
	Modification of description in Figure 6-7 Format of Timer Clock Select Register 0 (TPS0)	

(9/10)

Edition	Description	Chapter
Rev.1.00	Modification of error in Figure 6-16 Format of Timer Output Enable Register 0 (TOE0)	CHAPTER 6 TIMER ARRAY UNIT
	Addition of caution in 6.4.2 Basic rules of 8-bit timer operation function (only channels 1 and 3)	
	Modification of error in Figure 6-26 Operation Timing (In Capture Mode: Input Pulse Interval Measurement)	
	Modification of error in Figure 6-28 Operation Timing (In Capture & One-count Mode: High-level Width Measurement)	
	Addition of caution in Figure 6-41 Operation Procedure of Interval Timer/Square Wave Output Function	
	Modification of error in Figure 6-61 Operation Procedure When Delay Counter Function Is Used	
	Addition of caution in 6.8.2 Operation as PWM function	
	Addition of caution in 6.8.3 Operation as multiple PWM output function	
	Modification of Figure 6-74 Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used	
	Modification of Figure 6-75 Example of Set Contents of Registers When Multiple PWM Output Function (Slave Channel) Is Used (Output Two Types of PWMs)	
	Modification of error in Figure 7-2 Format of Peripheral Enable Register 0 (PER0)	CHAPTER 7 12-BIT INTERVAL TIMER
	Modification of cautions in Figure 8-2 Format of Clock Output Select Register 0 (CKS0)	CHAPTER 8 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER
	Addition of note in Figure 8-3 Format of Port Mode Registers 0, 4 (PM0, PM4)	
	Modification of description in 8.4.1 Operation as output pin	
	Modification of Figure 10-13 Conversion Operation of A/D Converter	CHAPTER 10 A/D CONVERTER
	Modification of Table 10-5 Resistance and Capacitance Values of Equivalent Circuit	
	Modification of description in Figure 12-21 Peripheral Enable Register 0 (PER0) Setting When Stopping Operation by Units	CHAPTER 12 SERIAL ARRAY UNIT
	Addition of note in Figure 12-22 Each Register Setting When Stopping Operation by Channels	
	Modification of error in 12.7 Operation of Simplified I^2C (IIC00) Communication	
	Modification of caution in Figure 14-5 Format of External Interrupt Rising Edge Enable Register (EGP0) and External Interrupt Falling Edge Enable Register (EGN0)	CHAPTER 14 INTERRUPT FUNCTIONS
	Modification of error in Table 15-1 Assignment of Key Interrupt Detection Pins	CHAPTER 15 KEY INTERRUPT FUNCTION
	Modification of caution in Figure 15-3 Format of Key Return Mode Register (KRM0)	
	Addition of caution in Figure 15-4 Format of Key Return Flag Register (KRF)	
	Addition of note in Figure 15-5 Format of Port Mode Registers 0, 4 (PM0, PM4)	
	Modification of note in Figure 15-6 Operation of INTKR Signal When a Key Interrupt is Input to a Single Channel (When KRMD = 0 and KREG = 0)	
	Modification of note in Figure 15-7 Operation of INTKR Signal When Key Interrupts Are Input to Multiple Channels (When KRMD = 0 and KREG = 0)	
	Modification of note in Figure 15-8 Basic Operation of the INTKR Signal When the Key Interrupt Flag Is Used (When KRMD = 1 and KREG = 0)	
	Modification of Figure 15-9 Operation of INTKR Signal When Key Interrupts Are Input to Multiple Channels (When KRMD = 1 and KREG = 0)	

(10/10)

Edition	Description	Chapter
Rev.1.00	Modification of notes in Figure 16-1 HALT Mode Release by Interrupt Request Generation	CHAPTER 16 STANDBY FUNCTION
	Modification of note in Figure 16-2 HALT Mode Release by Reset Signal Generation	
	Modification of error in Table 16-2 Operating Statuses in STOP Mode	
	Modification of notes and remarks in Figure 16-3 STOP Mode Release by Interrupt Request Generation (1/2)	
	Modification of notes in Figure 16-3 STOP Mode Release by Interrupt Request Generation (2/2)	
	Modification of note in Figure 16-4 STOP Mode Release by Reset Signal Generation	
	Modification of caution 1	CHAPTER 17 RESET FUNCTION
	Modification of Figure 17-2 Timing of Reset by $\overline{\text{RESET}}$ Input	
	Modification of Figure 17-3 Timing of Reset Due to Watchdog Timer Overflow or Execution of Illegal Instruction	
	Modification of Figure 18-2 Timing of Internal Reset Signal Generation	CHAPTER 18 SELECTABLE POWER-ON-RESET CIRCUIT
	Modification of Figure 18-3 Example of Software Processing When Supply Voltage Fluctuation is 50 ms or Less in Vicinity of the Voltage Detection Level	
	Addition of remark in Figure 19-2 Format of User Option Byte (000C1H)	CHAPTER 19 OPTION BYTE
	Modification of description in 20.4.2 Flash memory programming mode	CHAPTER 20 FLASH MEMORY
	Modification of Table 20-7 Processing Time of Each Command When Using PG-FP5 (Reference Values)	
	Addition of caution 5	CHAPTER 24 ELECTRICAL SPECIFICATIONS (10-PIN PRODUCTS)
	Modification of error in 24.6.2 SPOR circuit characteristics	

RL78/G10 User’s Manual: Hardware

Publication Date: Rev.3.20 Mar 20, 2023

Published by: Renesas Electronics Corporation

RL78/G10

Renesas Electronics Corporation

R01UH0384EJ0320