User's Manual

16

RL78/G1C

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

Rev.1.40 Apr 2024

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.

© 2024 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/G1C and design and develop application systems and programs for these devices.
The target products are as follows.

- 32-pin: R5F10JBxxxx (xxxx =, CANA, CAFP, CGNA, CGFP)
  R5F10KBxxxx (xxxx =, CANA, CAFP, CGNA, CGFP)
- 48-pin: R5F10JGxxxx (xxxx =, CANA, CAFB, CGNA, CGFB)
  R5F10KGxxxx (xxxx =, CANA, CAFB, CGNA, CGFB)

Purpose

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

Organization

The RL78/G1C manual is separated into two parts: this manual and the software edition (common to the RL78 Family).

RL78/G1C User's Manual (This Manual)	**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 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/G1C User's Manual Hardware	This manual
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	R20UT0008E

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
Quality Grades on NEC Semiconductor Devices	C11531E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E
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.
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) **5**

1.3.1 32-pin products 5

1.3.2 48-pin products 9

1.4 Pin Identification **13**

1.5 Block Diagram **14**

1.5.1 32-pin products 14

1.5.2 48-pin products 15

1.6 Outline of Functions **16**

CHAPTER 2 PIN FUNCTIONS **18**

2.1 Port Function **18**

2.1.1 32-pin products 19

2.1.2 48-pin products 21

2.2 Functions Other than Port Pins **24**

2.3 Connection of Unused Pins **27**

2.4 Block Diagrams of Pins **28**

CHAPTER 3 CPU ARCHITECTURE **40**

3.1 Memory Space **40**

3.1.1 Internal program memory space 44

3.1.2 Mirror area 47

3.1.3 Internal data memory space 49

3.1.4 Special function register (SFR) area 49

3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area 49

3.1.6 Data memory addressing 50

3.2 Processor Registers **51**

3.2.1 Control registers 51

3.2.2 General-purpose registers 53

3.2.3 ES and CS registers 54

3.2.4 Special function registers (SFRs) 55

3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers) 60

3.3 Instruction Address Addressing **69**

3.3.1 Relative addressing 69

3.3.2 Immediate addressing 69

3.3.3 Table indirect addressing 70
3.3.4 Register direct addressing 70
3.4 Addressing for Processing Data Addresses 71
3.4.1 Implied addressing 71
3.4.2 Register addressing 71
3.4.3 Direct addressing 72
3.4.4 Short direct addressing 73
3.4.5 SFR addressing 74
3.4.6 Register indirect addressing 75
3.4.7 Based addressing 76
3.4.8 Based indexed addressing 80
3.4.9 Stack addressing 81

CHAPTER 4 PORT FUNCTIONS 85
4.1 Port Functions 85
4.2 Port Configuration 86
4.2.1 Port 0 87
4.2.2 Port 1 87
4.2.3 Port 2 87
4.2.4 Port 3 88
4.2.5 Port 4 88
4.2.6 Port 5 88
4.2.7 Port 6 88
4.2.8 Port 7 88
4.2.9 Port 12 89
4.2.10 Port 13 89
4.2.11 Port 14 89
4.3 Registers Controlling Port Function 90
4.3.1 Port mode registers (PMxx) 93
4.3.2 Port registers (Pxx) 94
4.3.3 Pull-up resistor option registers (PUxx) 95
4.3.4 Port input mode registers (PIM0, PIM3, PIM5) 96
4.3.5 Port output mode registers (POM0, POM3, POM5, POM7) 97
4.3.6 Port mode control registers (PMC0, PMC12) 98
4.3.7 A/D port configuration register (ADPC) 99
4.3.8 Peripheral I/O redirection register (PIOR) 100
4.4 Port Function Operations 101
4.4.1 Writing to I/O port 101
4.4.2 Reading from I/O port 101
4.4.3 Operations on I/O port 101
4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers 102

4.5 Register Settings When Using Alternate Function 104
4.5.1 Basic concept when using alternate function 104
4.5.2 Register settings for alternate function whose output function is not used 105
4.5.3 Register setting examples for used port and alternate functions 106
4.6 Cautions When Using Port Function 114
4.6.1 Cautions on 1-bit manipulation instruction for port register n (Pn) 114
4.6.2 Cautions on the pin settings on the products other than 48-pin 115

CHAPTER 5 CLOCK GENERATOR 116

5.1 Functions of Clock Generator 116
5.2 Configuration of Clock Generator 118
5.3 Registers Controlling Clock Generator 120
5.3.1 Clock operation mode control register (CMC) 120
5.3.2 System clock control register (CKC) 123
5.3.3 Clock operation status control register (CSC) 124
5.3.4 Oscillation stabilization time counter status register (OSTC) 125
5.3.5 Oscillation stabilization time select register (OSTS) 127
5.3.6 Peripheral enable register 0 (PER0) 129
5.3.7 Subsystem clock supply mode control register (OSMC) 130
5.3.8 High-speed on-chip oscillator frequency select register (HOCODIV) 131
5.3.9 High-speed on-chip oscillator trimming register (HIOTRM) 132
5.3.10 PLL control register (DSCCTL) 133
5.3.11 Main clock control register (MCKC) 134
5.4 System Clock Oscillator 135
5.4.1 X1 oscillator 135
5.4.2 XT1 oscillator 135
5.4.3 High-speed on-chip oscillator 139
5.4.4 Low-speed on-chip oscillator 139
5.4.5 PLL (Phase Locked Loop) 139
5.5 Clock Generator Operation 140
5.6 Controlling Clock 142
5.6.1 Example of setting high-speed on-chip oscillator 142
5.6.2 Example of setting X1 oscillation clock 143
5.6.3 Example of setting XT1 oscillation clock 144
5.6.4 Example of setting PLL circuit 145
5.6.5 CPU clock status transition diagram 146
5.6.6 Condition before changing CPU clock and processing after changing CPU clock 154
5.6.7 Time required for switchover of CPU clock and system clock 157
5.6.8 Conditions before clock oscillation is stopped 158
5.7 Resonator and Oscillator Constants 159

CHAPTER 6 TIMER ARRAY UNIT **163**

6.1 Functions of Timer Array Unit **164**

6.1.1 Independent channel operation function 164

6.1.2 Simultaneous channel operation function 165

6.1.3 8-bit timer operation function (channels 1 and 3 only) 166

6.2 Configuration of Timer Array Unit **167**

6.2.1 Timer count register mn (TCRmn) 170

6.2.2 Timer data register mn (TDRmn) 172

6.3 Registers Controlling Timer Array Unit **173**

6.3.1 Peripheral enable register 0 (PER0) 174

6.3.2 Timer clock select register m (TPSm) 175

6.3.3 Timer mode register mn (TMRmn) 178

6.3.4 Timer status register mn (TSRmn) 183

6.3.5 Timer channel enable status register m (TEm) 184

6.3.6 Timer channel start register m (TSm) 185

6.3.7 Timer channel stop register m (TTm) 186

6.3.8 Timer input select register 0 (TIS0) 187

6.3.9 Timer output enable register m (TOEm) 188

6.3.10 Timer output register m (TOm) 189

6.3.11 Timer output level register m (TOLm) 190

6.3.12 Timer output mode register m (TOMm) 191

6.3.13 Noise filter enable register 1 (NFEN1) 192

6.3.14 Registers controlling port functions of pins to be used for timer I/O 193

6.4 Basic Rules of Timer Array Unit **194**

6.4.1 Basic rules of simultaneous channel operation function 194

6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only) 196

6.5 Operation of Counter **197**

6.5.1 Count clock (fTCLK) 197

6.5.2 Start timing of counter 199

6.5.3 Operation of counter 200

6.6 Channel Output (TOmn pin) Control **205**

6.6.1 TOmn pin output circuit configuration 205

6.6.2 TOmn pin output setting 206

6.6.3 Cautions on channel output operation 207

6.6.4 Collective manipulation of TOmn bit 212

6.6.5 Timer interrupt and TOmn pin output at operation start 213

6.7 Timer Input (TImn) Cntorol **214**

6.7.1 TImn pin input circuit configuration 214

6.7.2 Noise filter 214

6.7.3 Cautions on channel input 215

6.8 Independent Channel Operation Function of Timer Array Unit **216**

6.8.1 Operation as interval timer/square wave output 216
6.8.2 Operation as external event counter 222
6.8.3 Operation as frequency divider (channel 0 only) 227
6.8.4 Operation as input pulse interval measurement 231
6.8.5 Operation as input signal high-/low-level width measurement 235
6.8.6 Operation as delay counter 239
6.9 Simultaneous Channel Operation Function of Timer Array Unit 244
6.9.1 Operation as one-shot pulse output function 244
6.9.2 Operation as PWM function 251
6.9.3 Operation as multiple PWM output function 258
6.10 Cautions When Using Timer Array Unit 266
6.10.1 Cautions when using timer output 266

CHAPTER 7 REAL-TIME CLOCK 267

7.1 Functions of Real-time Clock 267
7.2 Configuration of Real-time Clock 267
7.3 Registers Controlling Real-time Clock 269
7.3.1 Peripheral enable register 0 (PER0) 270
7.3.2 Subsystem clock supply mode control register (OSMC) 271
7.3.3 Real-time clock control register 0 (RTCC0) 272
7.3.4 Real-time clock control register 1 (RTCC1) 273
7.3.5 Second count register (SEC) 275
7.3.6 Minute count register (MIN) 275
7.3.7 Hour count register (HOUR) 276
7.3.8 Day count register (DAY) 278
7.3.9 Week count register (WEEK) 279
7.3.10 Month count register (MONTH) 280
7.3.11 Year count register (YEAR) 280
7.3.12 Watch error correction register (SUBCUD) 281
7.3.13 Alarm minute register (ALARMWM) 282
7.3.14 Alarm hour register (ALARMWH) 282
7.3.15 Alarm week register (ALARMWW) 283
7.3.16 Port mode register 3 (PM3) 284
7.3.17 Port register 3 (P3) 284
7.4 Real-time Clock Operation 285
7.4.1 Starting operation of real-time clock 285
7.4.2 Shifting to HALT/STOP mode after starting operation 286
7.4.3 Reading/writing real-time clock 287
7.4.4 Setting alarm of real-time clock 289
7.4.5 1 Hz output of real-time clock 290
7.4.6 Example of watch error correction of real-time clock 291

CHAPTER 8 12-BIT INTERVAL TIMER **296**

8.1 Functions of 12-bit Interval Timer **296**
8.2 Configuration of 12-bit Interval Timer **296**
8.3 Registers Controlling 12-bit Interval Timer **297**
8.3.1 Peripheral enable register 0 (PER0) 297
8.3.2 Subsytem clock supply mode control register (OSMC) 298
8.3.3 Interval timer control register (ITMC) 299
8.4 12-bit Interval Timer Operation **300**
8.4.1 12-bit interval timer operation timing 300
8.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode 301

CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER **302**

9.1 Functions of Clock Output/Buzzer Output Controller **302**
9.2 Configuration of Clock Output/Buzzer Output Controller **304**
9.3 Registers Controlling Clock Output/Buzzer Output Controller **304**
9.3.1 Clock output select registers n (CKSn) 304
9.3.2 Registers controlling port functions of pins to be used for clock or buzzer output 306
9.4 Operations of Clock Output/Buzzer Output Controller **307**
9.4.1 Operation as output pin 307
9.5 Cautions of Clock Output/Buzzer Output Controller **307**

CHAPTER 10 WATCHDOG TIMER **308**

10.1 Functions of Watchdog Timer **308**
10.2 Configuration of Watchdog Timer **309**
10.3 Register Controlling Watchdog Timer **310**
10.3.1 Watchdog timer enable register (WDTE) 310
10.4 Operation of Watchdog Timer **311**
10.4.1 Controlling operation of watchdog timer 311
10.4.2 Setting overflow time of watchdog timer 313
10.4.3 Setting window open period of watchdog timer 314
10.4.4 Setting watchdog timer interval interrupt 316

CHAPTER 11 A/D CONVERTER **317**

11.1 Function of A/D Converter **317**
11.2 Configuration of A/D Converter **320**
11.3 Registers Controlling A/D Converter **322**
11.3.1 Peripheral enable register 0 (PER0) 323
11.3.2 A/D converter mode register 0 (ADM0) 324
11.3.3 A/D converter mode register 1 (ADM1) 333

11.3.4 A/D converter mode register 2 (ADM2) 334
11.3.5 10-bit A/D conversion result register (ADCR) 336
11.3.6 8-bit A/D conversion result register (ADCRH) 336
11.3.7 Analog input channel specification register (ADS) 337
11.3.8 Conversion result comparison upper limit setting register (ADUL) 339
11.3.9 Conversion result comparison lower limit setting register (ADLL) 339
11.3.10 A/D test register (ADTES) 340
11.3.11 Registers controlling port function of analog input pins 340
11.4 A/D Converter Conversion Operations 341
11.5 Input Voltage and Conversion Results 343
11.6 A/D Converter Operation Modes 344
11.6.1 Software trigger mode (select mode, sequential conversion mode) 344
11.6.2 Software trigger mode (select mode, one-shot conversion mode) 345
11.6.3 Software trigger mode (scan mode, sequential conversion mode) 346
11.6.4 Software trigger mode (scan mode, one-shot conversion mode) 347
11.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode) 348
11.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode) 349
11.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode) 350
11.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode) 351
11.6.9 Hardware trigger wait mode (select mode, sequential conversion mode) 352
11.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode) 353
11.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode) 354
11.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode) 355
11.7 A/D Converter Setup Flowchart 356
11.7.1 Setting up software trigger mode 357
11.7.2 Setting up hardware trigger no-wait mode 358
11.7.3 Setting up hardware trigger wait mode 359
11.7.4 Setup when using temperature sensor (example for software trigger mode and one-shot conversion mode) 360
11.7.5 Setting up test mode 361
11.8 SNOOZE Mode Function 362
11.9 How to Read A/D Converter Characteristics Table 366
11.10 Cautions for A/D Converter 368

CHAPTER 12 SERIAL ARRAY UNIT 372

12.1 Functions of Serial Array Unit 373
12.1.1 Simplified SPI (CSI00, CSI01) 373
12.1.2 UART (UART0) 374
12.1.3 Simplified I^2C (IIC00, IIC01) 375
12.2 Configuration of Serial Array Unit 376
12.2.1 Shift register 378

12.2.2 Lower 8/9 bits of the serial data register mn (SDRmn) ... 378
12.3 Registers Controlling Serial Array Unit ... 380
12.3.1 Peripheral enable register 0 (PER0) ... 381
12.3.2 Serial clock select register m (SPSm) ... 382
12.3.3 Serial mode register mn (SMRmn) ... 383
12.3.4 Serial communication operation setting register mn (SCRmn) ... 384
12.3.5 Serial data register mn (SDRmn) ... 387
12.3.6 Serial flag clear trigger register mn (SIRmn) ... 388
12.3.7 Serial status register mn (SSRmn) ... 389
12.3.8 Serial channel start register m (SSm) ... 391
12.3.9 Serial channel stop register m (STm) ... 392
12.3.10 Serial channel enable status register m (SEm) ... 393
12.3.11 Serial output enable register m (SOEm) ... 394
12.3.12 Serial output register m (SOm) ... 395
12.3.13 Serial output level register m (SOLm) ... 396
12.3.14 Serial standby control register m (SSCm) ... 397
12.3.15 Noise filter enable register 0 (NFEN0) ... 398
12.3.16 Registers controlling port functions of serial input/output pins ... 399
12.4 Operation Stop Mode ... 400
12.4.1 Stopping the operation by units ... 400
12.4.2 Stopping the operation by channels ... 401
12.5 Operation of Simplified SPI (CSI00, CSI01) Communication ... 402
12.5.1 Master transmission ... 403
12.5.2 Master reception ... 413
12.5.3 Master transmission/reception ... 422
12.5.4 Slave transmission ... 431
12.5.5 Slave reception ... 441
12.5.6 Slave transmission/reception ... 448
12.5.7 SNOOZE mode function ... 458
12.5.8 Calculating transfer clock frequency ... 462
12.5.9 Procedure for processing errors that occurred during Simplified SPI (CSI00, CSI01) communication ... 464
12.6 Operation of UART (UART0) Communication ... 465
12.6.1 UART transmission ... 466
12.6.2 UART reception ... 476
12.6.3 Calculating baud rate ... 483
12.6.4 Procedure for processing errors that occurred during UART (UART0) communication ... 487
12.7 Operation of Simplified I²C (IIC00, IIC01) Communication ... 488
12.7.1 Address field transmission ... 489
12.7.2 Data transmission ... 495
12.7.3 Data reception ... 499

12.7.4 Stop condition generation 504
12.7.5 Calculating transfer rate 505
12.7.6 Procedure for processing errors that occurred during simplified I²C (IIC00, IIC01) communication 508

CHAPTER 13 SERIAL INTERFACE IICA 509

13.1 Functions of Serial Interface IICA 509
13.2 Configuration of Serial Interface IICA 512
13.3 Registers Controlling Serial Interface IICA 515
13.3.1 Peripheral enable register 0 (PER0) 515
13.3.2 IICA control register 00 (IICCTL00) 516
13.3.3 IICA status register 0 (IICS0) 521
13.3.4 IICA flag register 0 (IICF0) 523
13.3.5 IICA control register 01 (IICCTL01) 525
13.3.6 IICA low-level width setting register 0 (IICWL0) 527
13.3.7 IICA high-level width setting register 0 (IICWH0) 527
13.3.8 Port mode register 6 (PM6) 528
13.4 I²C Bus Mode Functions 529
13.4.1 Pin configuration 529
13.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers 530
13.5 I²C Bus Definitions and Control Methods 532
13.5.1 Start conditions 532
13.5.2 Addresses 533
13.5.3 Transfer direction specification 533
13.5.4 Acknowledge (ACK) 534
13.5.5 Stop condition 535
13.5.6 Clock stretch 536
13.5.7 Canceling clock stretch 538
13.5.8 Interrupt request (INTIICA0) generation timing and clock stretch control 539
13.5.9 Address match detection method 540
13.5.10 Error detection 540
13.5.11 Extension code 540
13.5.12 Arbitration 541
13.5.13 Wakeup function 543
13.5.14 Communication reservation 546
13.5.15 Cautions 550
13.5.16 Communication operations 551
13.5.17 Timing of I²C interrupt request (INTIICA0) occurrence 558
13.6 Timing Charts 579

CHAPTER 14 USB 2.0 HOST/FUNCTION MODULE (USB) 594

14.1 Functions of USB 2.0 Host/Function Module 594

14.2 Configuration of USB 2.0 Host/Function Module 597

14.3 Registers Used in USB 2.0 Host/Function Module 600

14.3.1 System configuration control register (SYSCFG), system configuration control register 1 (SYSCFG1) 603

14.3.2 System configuration status register n (SYSSTSn) (n = 0, 1) 606

14.3.3 Device state control register n (DVSTCTRn) (n = 0, 1) 607

14.3.4 DMAn-FIFO pin configuration register (DMAnPCFG) (n = 0, 1) 612

14.3.5 CFIFO port register (CFIFOM), DnFIFO port register (DnFIFOM) (n = 0, 1) 613

14.3.6 DMA transfer DnFIFO port register (DnFIFO) (n = 0, 1) 615

14.3.7 CFIFO port select register (CFIFOSEL), DnFIFO port select register (DnFIFOSEL) (n = 0, 1) 616

14.3.8 CFIFO port control register (CFIFOCTR), DnFIFO port control register (DnFIFOCTR) (n = 0, 1) 620

14.3.9 Interrupt enable register 0 (INTENB0) 624

14.3.10 Interrupt enable register n (INTENBn) (n = 1, 2) 626

14.3.11 BRDY interrupt enable register (BRDYENB) 629

14.3.12 NRDY interrupt enable register (NRDYENB) 629

14.3.13 BEMP interrupt enable register (BEMPENB) 630

14.3.14 SOF output configuration register (SOFCFG) 631

14.3.15 Interrupt status register 0 (INTSTS0) 632

14.3.16 Interrupt status register n (INTSTSn) (n = 1, 2) 635

14.3.17 BRDY interrupt status register (BRDYSTS) 640

14.3.18 NRDY interrupt status register (NRDYSTS) 640

14.3.19 BEMP interrupt status register (BEMPSTS) 641

14.3.20 Frame number register (FRMNUM) 641

14.3.21 USB address register (USBADDR) 642

14.3.22 USB request type register (USBREQ) 643

14.3.23 USB request value register (USBVAL) 643

14.3.24 USB request index register (USBINDX) 644

14.3.25 USB request length register (USBLENG) 644

14.3.26 DCP configuration register (DCPCFG) 645

14.3.27 DCP maximum packet size register (DCPMAXP) 646

14.3.28 DCP control register (DCPCTR) 647

14.3.29 Pipe window select register (PIPESEL) 650

14.3.30 Pipe configuration register (PIPECFG) 651

14.3.31 Pipe maximum packet size register (PIPEMAXP) 653

14.3.32 Pipe cycle control register (PIPEPERI) 654

14.3.33 PIPEn control registers (PIPEnCTR) (n = 4 to 7) 655

14.3.34 PIPEn transaction counter enable registers (PIPEnTRE) (n = 4, 5) 662

14.3.35 PIPEn transaction counter registers (PIPEnTRN) (n = 4, 5) 663

14.3.36 BC control register n (USBBCCTRLn) (n = 0, 1) 664

14.3.37 BC option control register n (USBBCOPTn) (n = 0, 1) ... 668
14.3.38 USB module control register (USBMC) ... 672
14.3.39 Device address n configuration registers (DEVADDn) (n = 0 to 5) ... 673
14.4 Operation ... 674
14.4.1 System control ... 674
14.4.1.1 Starting operation ... 674
14.4.1.2 Controller function selection ... 675
14.4.1.3 Controlling USB data bus resistors ... 676
14.4.2 Interrupt sources ... 680
14.4.3 Interrupts ... 684
14.4.3.1 BRDY interrupt ... 684
14.4.3.2 NRDY interrupt ... 687
14.4.3.3 BEMP interrupt ... 689
14.4.3.4 Device state transition interrupt ... 691
14.4.3.5 Control transfer stage transition interrupt ... 692
14.4.3.6 Frame update interrupt ... 693
14.4.3.7 VBUS interrupt ... 693
14.4.3.8 Resume interrupt ... 693
14.4.3.9 Overcurrent interrupt ... 693
14.4.3.10 Bus change interrupt ... 694
14.4.3.11 USB disconnection detection interrupt ... 694
14.4.3.12 Setup transaction normal response interrupt ... 694
14.4.3.13 Setup transaction error interrupt ... 694
14.4.3.14 Connection interrupt ... 694
14.4.3.15 EOF error interrupt ... 694
14.4.3.16 Portable device detection interrupt ... 694
14.4.4 Pipe control ... 695
14.4.4.1 Pipe control register switching procedures ... 696
14.4.4.2 Transfer types ... 696
14.4.4.3 Endpoint number ... 696
14.4.4.4 Maximum packet size setting ... 696
14.4.4.5 Transaction counter (for PIPE4, PIPE5 in reading direction) ... 697
14.4.4.6 Response PID ... 697
14.4.4.7 Data PID sequence bit ... 698
14.4.4.8 Response PID = NAK function ... 698
14.4.4.9 Auto response mode ... 698
14.4.4.10 OUT-NAK mode ... 698
14.4.4.11 Null auto response mode ... 699
14.4.5 FIFO buffer memory ... 700
14.4.5.1 FIFO buffer memory ... 700
14.4.5.2 FIFO port functions ... 702

14.4.5.3 DMA transfers (D0FIFO and D1FIFO ports) 703
14.4.6 Control transfers (DCP) 704
14.4.6.1 Control transfers when host controller function is selected 704
14.4.6.2 Control transfers when function controller function is selected 705
14.4.7 Bulk transfers (PIPE4, PIPE5) 706
14.4.8 Interrupt transfers (PIPE6, PIPE7) 706
14.4.8.1 Interval counter during interrupt transfers when host controller function is selected 706
14.4.9 SOF interpolation function 707
14.4.10 Pipe schedule 708
14.4.10.1 Conditions for generating a transaction 708
14.4.10.2 Transfer schedule 708
14.4.10.3 Enabling USB communication 708
14.4.11 Controlling battery charging detection 709
14.4.12 Battery charging connection detection optional functions 710
14.4.13 Battery charging detection processing 713
14.4.13.1 Processing when function controller is selected 713
14.4.13.2 Processing when host controller is selected 715

CHAPTER 15 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR 718

15.1 Functions of Multiplier and Divider/Multiply-Accumulator 718
15.2 Configuration of Multiplier and Divider/Multiply-Accumulator 718
15.2.1 Multiplication/division data register A (MDAH, MDAL) 720
15.2.2 Multiplication/division data register B (MDBL, MDBH) 721
15.2.3 Multiplication/division data register C (MDCL, MDCH) 722
15.3 Register Controlling Multiplier and Divider/Multiply-Accumulator 724
15.3.1 Multiplication/division control register (MDUC) 724
15.4 Operations of Multiplier and Divider/Multiply-Accumulator 726
15.4.1 Multiplication (unsigned) operation 726
15.4.2 Multiplication (signed) operation 727
15.4.3 Multiply-accumulation (unsigned) operation 728
15.4.4 Multiply-accumulation (signed) operation 730
15.4.5 Division operation 732

CHAPTER 16 DMA CONTROLLER 734

16.1 Functions of DMA Controller 734
16.2 Configuration of DMA Controller 735
16.2.1 DMA SFR address register n (DSAn) 735
16.2.2 DMA RAM address register n (DRAn) 736
16.2.3 DMA byte count register n (DBCn) 737
16.3 Registers Controlling DMA Controller 738
16.3.1 DMA mode control register n (DMCn) 739

16.3.2 DMA operation control register n (DRCn) ... 741
16.4 Operation of DMA Controller ... 742
16.4.1 Operation procedure ... 742
16.4.2 Transfer mode ... 743
16.4.3 Termination of DMA transfer ... 743
16.5 Example of Setting of DMA Controller ... 744
16.5.1 Simplified SPI (CSI) consecutive transmission ... 744
16.5.2 Consecutive capturing of A/D conversion results ... 746
16.5.3 UART consecutive reception + ACK transmission ... 748
16.5.4 Holding DMA transfer pending by DWAITn bit ... 749
16.5.5 Forced termination by software ... 750
16.6 Cautions on Using DMA Controller ... 752

CHAPTER 17 INTERRUPT FUNCTIONS ... 755

17.1 Interrupt Function Types ... 755
17.2 Interrupt Sources and Configuration ... 755
17.3 Registers Controlling Interrupt Functions ... 761
17.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) ... 764
17.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) ... 766
17.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) ... 767
17.3.4 External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1) ... 769
17.3.5 Program status word (PSW) ... 771
17.4 Interrupt Servicing Operations ... 772
17.4.1 Maskable interrupt request acknowledgment ... 772
17.4.2 Software interrupt request acknowledgment ... 775
17.4.3 Multiple interrupt servicing ... 775
17.4.4 Interrupt request hold ... 779

CHAPTER 18 KEY INTERRUPT FUNCTION ... 780

18.1 Functions of Key Interrupt ... 780
18.2 Configuration of Key Interrupt ... 781
18.3 Register Controlling Key Interrupt ... 782
18.3.1 Key return mode register (KRM) ... 782
18.3.2 Port mode register 7 (PM7) ... 783

CHAPTER 19 STANDBY FUNCTION ... 784

19.1 Standby Function ... 784
19.2 Registers Controlling Standby Function ... 785

19.3 Standby Function Operation 786
19.3.1 HALT mode 786
19.3.2 STOP mode 791
19.3.3 SNOOZE mode 796

CHAPTER 20 RESET FUNCTION 799

20.1 Timing of Reset Operation 801
20.2 Operation Statuses During Reset Period 803
20.3 Register for Confirming Reset Source 805
20.3.1 Reset control flag register (RESF) 805

CHAPTER 21 POWER-ON-RESET CIRCUIT 808

21.1 Functions of Power-on-reset Circuit 808
21.2 Configuration of Power-on-reset Circuit 809
21.3 Operation of Power-on-reset Circuit 809

CHAPTER 22 VOLTAGE DETECTOR 813

22.1 Functions of Voltage Detector 813
22.2 Configuration of Voltage Detector 814
22.3 Registers Controlling Voltage Detector 814
22.3.1 Voltage detection register (LVIM) 815
22.3.2 Voltage detection level register (LVIS) 816
22.4 Operation of Voltage Detector 819
22.4.1 When used as reset mode 819
22.4.2 When used as interrupt mode 821
22.4.3 When used as interrupt & reset mode 823
22.5 Cautions for Voltage Detector 828

CHAPTER 23 SAFETY FUNCTIONS 830

23.1 Overview of Safety Functions 830
23.2 Registers Used by Safety Functions 831
23.3 Operation of Safety Functions 831
23.3.1 Flash memory CRC operation function (high-speed CRC) 831
23.3.1.1 Flash memory CRC control register (CRC0CTL) 832
23.3.1.2 Flash memory CRC operation result register (PGCRCL) 833
23.3.2 CRC operation function (general-purpose CRC) 835
23.3.2.1 CRC input register (CRCIN) 835
23.3.2.2 CRC data register (CRCD) 836
23.3.3 RAM parity error detection function 837
23.3.3.1 RAM parity error control register (RPECTL) 837

23.3.4 RAM guard function 839
23.3.4.1 Invalid memory access detection control register (IAWCTL) 839
23.3.5 SFR guard function 840
23.3.5.1 Invalid memory access detection control register (IAWCTL) 840
23.3.6 Invalid memory access detection function 841
23.3.6.1 Invalid memory access detection control register (IAWCTL) 842
23.3.7 Frequency detection function 843
23.3.7.1 Timer input select register 0 (TIS0) 844
23.3.8 A/D test function 845
23.3.8.1 A/D test register (ADTES) 847
23.3.8.2 Analog input channel specification register (ADS) 848
23.3.9 Digital output signal level detection function for I/O ports 850
23.3.9.1 Port mode select register (PMS) 850

CHAPTER 24 REGULATOR 851

24.1 Regulator Overview 851

CHAPTER 25 OPTION BYTE 852

25.1 Functions of Option Bytes 852
25.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H) 852
25.1.2 On-chip debug option byte (000C3H/ 010C3H) 853
25.2 Format of User Option Byte 854
25.3 Format of On-chip Debug Option Byte 858
25.4 Setting of Option Byte 859

CHAPTER 26 FLASH MEMORY 860

26.1 Serial Programming Using Flash Memory Programmer 862
26.1.1 Programming environment 863
26.1.2 Communication mode 863
26.2 Serial Programming Using External Device (that Incorporates UART) 864
26.2.1 Programming environment 864
26.2.2 Communication mode 865
26.3 Connection of Pins on Board 866
26.3.1 P40/TOOL0 pin 866
26.3.2 RESET pin 866
26.3.3 Port pins 867
26.3.4 REGC pin 867
26.3.5 X1 and X2 pins 867
26.3.6 Power supply 867
26.4 Serial Programming Method 868

26.4.1 Serial programming procedure 868
26.4.2 Flash memory programming mode 869
26.4.3 Selecting communication mode 870
26.4.4 Communication commands 871
26.5 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value) 873
26.6 Self-Programming 874
26.6.1 Self-programming procedure 875
26.6.2 Boot swap function 876
26.6.3 Flash shield window function 878
26.7 Security Settings 879
26.8 Data Flash 881
26.8.1 Data flash overview 881
26.8.2 Register controlling data flash memory 881
26.8.2.1 Data flash control register (DFLCTL) 881
26.8.3 Procedure for accessing data flash memory 882

CHAPTER 27 ON-CHIP DEBUG FUNCTION 884

27.1 Connecting E1 On-chip Debugging Emulator 884
27.2 On-Chip Debug Security ID 885
27.3 Securing of User Resources 885

CHAPTER 28 BCD CORRECTION CIRCUIT 887

28.1 BCD Correction Circuit Function 887
28.2 Registers Used by BCD Correction Circuit 887
28.2.1 BCD correction result register (BCDADJ) 887
28.3 BCD Correction Circuit Operation 888

CHAPTER 29 INSTRUCTION SET 890

29.1 Conventions Used in Operation List 891
29.1.1 Operand identifiers and specification methods 891
29.1.2 Description of operation column 892
29.1.3 Description of flag operation column 893
29.1.4 PREFIX instruction 893
29.2 Operation List 894

CHAPTER 30 ELECTRICAL SPECIFICATIONS (A: TA = -40 to +85°C) 911

30.1 Absolute Maximum Ratings 912
30.2 Oscillator Characteristics 914
30.2.1 X1, XT1 oscillator characteristics 914
30.2.2 On-chip oscillator characteristics 914

30.2.3 PLL oscillator characteristics 915
30.3 DC Characteristics 916
30.3.1 Pin characteristics 916
30.3.2 Supply current characteristics 921
30.4 AC Characteristics 928
30.4.1 Basic operation 928
30.5 Peripheral Functions Characteristics 931
30.5.1 Serial array unit 931
30.5.2 Serial interface IICA 954
30.5.3 USB 957
30.6 Analog Characteristics 961
30.6.1 A/D converter characteristics 961
30.6.2 Temperature sensor/internal reference voltage characteristics 966
30.6.3 POR circuit characteristics 966
30.6.4 LVD circuit characteristics 967
30.6.5 Power supply voltage rising slope characteristics 968
30.7 RAM Data Retention Characteristics 969
30.8 Flash Memory Programming Characteristics 969
30.9 Dedicated Flash Memory Programmer Communication (UART) 969
30.10 Timing Specs for Switching Flash Memory Programming Modes 970

CHAPTER 31 ELECTRICAL SPECIFICATIONS (G: TA = -40 to +105°C) 971

31.1 Absolute Maximum Ratings 972
31.2 Oscillator Characteristics 974
31.2.1 X1, XT1 oscillator characteristics 974
31.2.2 On-chip oscillator characteristics 974
31.2.3 PLL oscillator characteristics 975
31.3 DC Characteristics 976
31.3.1 Pin characteristics 976
31.3.2 Supply current characteristics 981
31.4 AC Characteristics 988
31.4.1 Basic operation 988
31.5 Peripheral Functions Characteristics 991
31.5.1 Serial array unit 991
31.5.2 Serial interface IICA 1011
31.5.3 USB 1012
31.6 Analog Characteristics 1016
31.6.1 A/D converter characteristics 1016
31.6.2 Temperature sensor/internal reference voltage characteristics 1021
31.6.3 POR circuit characteristics 1021
31.6.4 LVD circuit characteristics 1022

31.6.5 Power supply voltage rising slope characteristics 1023
31.7 RAM Data Retention Characteristics 1024
31.8 Flash Memory Programming Characteristics 1024
31.9 Dedicated Flash Memory Programmer Communication (UART) 1024
31.10 Timing Specs for Switching Flash Memory Programming Modes 1025

CHAPTER 32 PACKAGE DRAWINGS 1026

32.1 32-pin Products 1026
32.2 48-pin Products 1030

APPENDIX A REVISION HISTORY 1034

A.1 Major Revisions in This Edition 1034
A.2 Revision History of Preceding Editions 1035

CHAPTER 1 OUTLINE

1.1 Features

Ultra-low power consumption technology

- V_{DD} = single power supply voltage of 2.4 to 5.5 V
- HALT mode
- STOP mode
- SNOOZE mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- Minimum instruction execution time: Can be changed from high speed (0.04167 μs: @ 24 MHz operation with high-speed on-chip oscillator) to ultra-low speed (30.5 μs: @ 32.768 kHz operation with subsystem clock)
- Address space: 1 MB
- General-purpose registers: 8-bit register × 8 × 4 banks
- On-chip RAM: 5.5 KB

Code flash memory

- Code flash memory: 32 KB
- Block size: 1 KB
- Prohibition of block erase and rewriting (security function)
- On-chip debug function
- Self-programming (with boot swap function/flash shield window function)

Data flash memory

- Data flash memory: 2 KB
- Back ground operation (BGO): Instructions can be executed from the program memory while rewriting the data flash memory.
- Number of rewrites: 1,000,000 times (TYP.)
- Voltage of rewrites: V_{DD} = 2.4 to 5.5 V

High-speed on-chip oscillator

- Select from 48 MHz, 24 MHz, 12 MHz, and 6 MHz
- High accuracy: ±1.0 % (V_{DD} = 2.4 to 5.5 V, T_A = -20 to +85°C)

Operating ambient temperature

- T_A = -40 to +85°C (A: Consumer applications)
- T_A = -40 to +105°C (G: Industrial applications)

Power management and reset function

- On-chip power-on-reset (POR) circuit
- On-chip voltage detector (LVD) (Select interrupt and reset from 9 levels)

DMA (Direct Memory Access) controller

- 2 channels
- Number of clocks during transfer between 8/16-bit SFR and internal RAM: 2 clocks

Multiplier and divider/multiply-accumulator

- 16 bits × 16 bits = 32 bits (Unsigned or signed)
- 32 bits ÷ 32 bits = 32 bits (Unsigned)
- 16 bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)

USB

- Complying with USB version 2.0, incorporating host/function controller
- Corresponding to full-speed transfer (12 Mbps) and low-speed (1.5 Mbps)
- Complying with Battery Charging Specification Revision 1.2
- Compliant with the 2.1A/1.0A charging mode prescribed in the Apple Inc. MFi specification in the USB power supply component specification[Note1].

Serial interface

- Simplified SPI (CSI[Note2]): 2 channels
- UART: 1 channel
- I^2C/Simplified I^2C communication: 1 channel or 2 channels

Timer

- 16-bit timer: 4 channels
- 12-bit interval timer: 1 channel
- Real-time clock: 1 channel (calendar for 99 years, alarm function, and clock correction function)
- 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: 8 to 9 channels
- Internal reference voltage (1.45 V), and temperature sensor

I/O port

- I/O port: 22 to 38 (N-ch open drain I/O [withstand voltage of 6 V]: 3, 4,
  N-ch open drain I/O [V_{DD} withstand voltage]: 5, 6)
- Can be set to N-ch open drain, TTL input buffer, 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

Note 1. To use the Apple Inc. battery charging mode, you must join in Apple's Made for iPod/iPhone/iPad (MFi) licensing program. Before requesting this specification from Renesas Electronics, please join in the Apple's MFi licensing program.

Note 2. 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.

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

ROM, RAM capacities

Flash ROM	Data flash	RAM	RL78/G1C	
			32-pin	48-pin
32 KB	2 KB	5.5 KB Note	R5F10JBC, R5F10KBC	R5F10JGC, R5F10KGC

Note This is about 4.5 KB when the self-programming function is used. (For details, see **CHAPTER 3 CPU ARCHITECTURE**.)

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

1.2 List of Part Numbers

Figure 1-1. Part Number, Memory Size, and Package of RL78/G1C

<R>

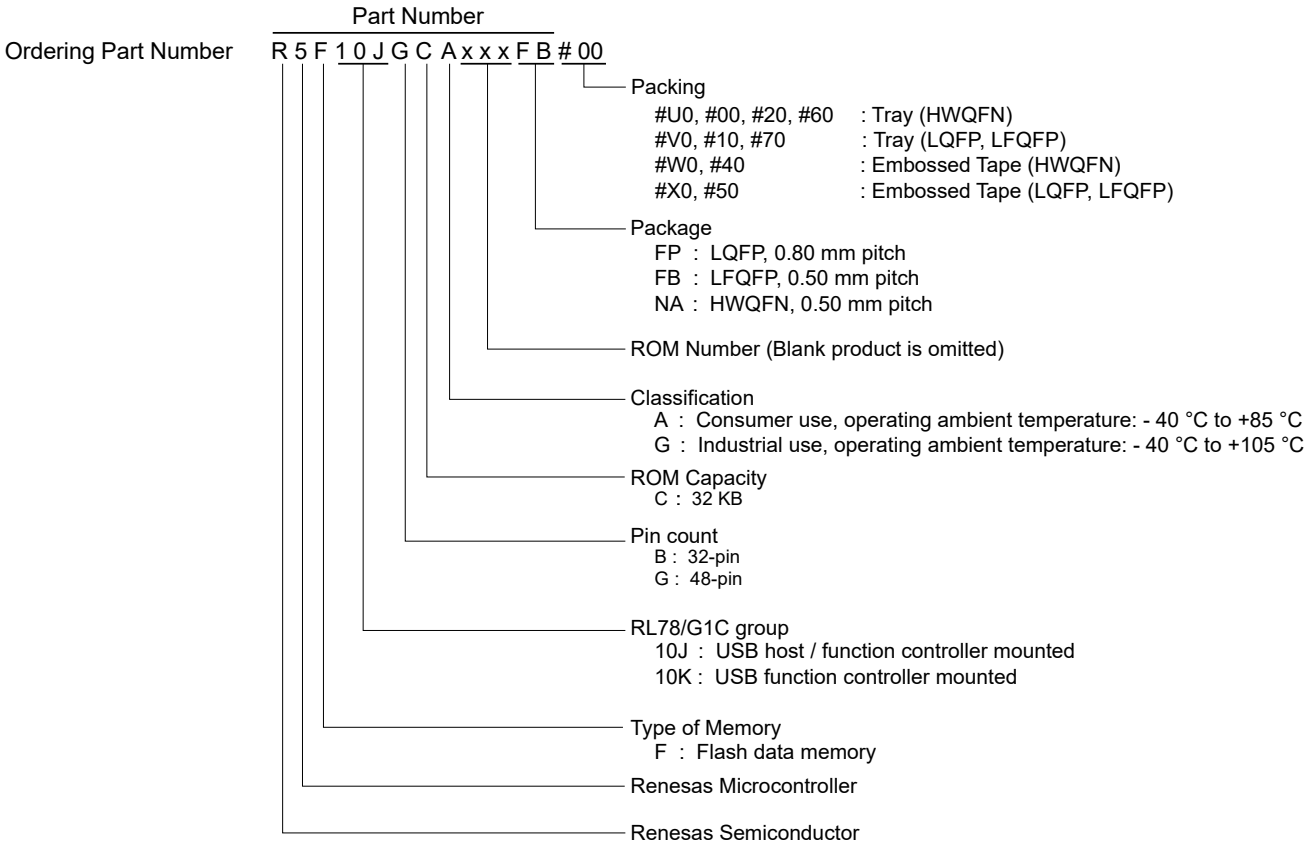


Table 1-1. List of Ordering Part Numbers

Pin count	Package	USB Function	Fields of Application Note	Part Number		RENESAS Code
				Part Number	Packaging specification	
32 pins	32-pin plastic HWQFN (5 × 5 , 0.5 mm pitch)	Host/Function controller	A	R5F10JBCANA	#U0, #W0	PWQN0032KB-A
<R>					#00, #20, #40, #60	PWQN0032KE-A PWQN0032KG-A
			G	R5F10JBCGNA	#U0, #W0	PWQN0032KB-A
<R>					#00, #20, #40, #60	PWQN0032KE-A PWQN0032KG-A
		Function controller only	A	R5F10KBCANA	#U0, #W0	PWQN0032KB-A
<R>					#00, #20, #40, #60	PWQN0032KE-A PWQN0032KG-A
			G	R5F10KBCGNA	#U0, #W0	PWQN0032KB-A
<R>					#00, #20, #40, #60	PWQN0032KE-A PWQN0032KG-A
<R>	32-pin plastic LQFP (7 × 7 , 0.5 mm pitch)	Host/Function controller	A	R5F10JBCAFP	#V0, #X0, #10, #50, #70	PLQP0032GB-A
			G	R5F10JBCGFP		
<R>		Function controller only	A	R5F10KBCAFP		
			G	R5F10KBCGFP		
<R> 48 pins	48-pin plastic LFQFP (7 × 7 , 0.5 mm pitch)	Host/Function controller	A	R5F10JGCAFB	#V0, #X0, #10, #50, #70	PLQP0048KF-A
			G	R5F10JGCGFB		
		Function controller only	A	R5F10KGCAFB		
			G	R5F10KGCGFB		
	48-pin plastic HWQFN (7 × 7 , 0.5 mm pitch)	Host/Function controller	A	R5F10JGCANA	#U0, #W0	PWQN0048KB-A
<R>					#00, #20, #40, #60	PWQN0048KE-A PWQN0048KG-A
<R>			G	R5F10JGCGNA	#U0, #W0	PWQN0048KB-A
<R>					#00, #20, #40, #60	PWQN0048KE-A PWQN0048KG-A
<R>		Function controller only	A	R5F10KGCANA	#U0, #W0	PWQN0048KB-A
<R>					#00, #20, #40, #60	PWQN0048KE-A PWQN0048KG-A
<R>			G	R5F10KGCGNA	#U0, #W0	PWQN0048KB-A
<R>					#00, #20, #40, #60	PWQN0048KE-A PWQN0048KG-A

Note For the fields of application, refer to **Figure 1-1 Part Number, Memory Size, and Package of RL78/G1C**.

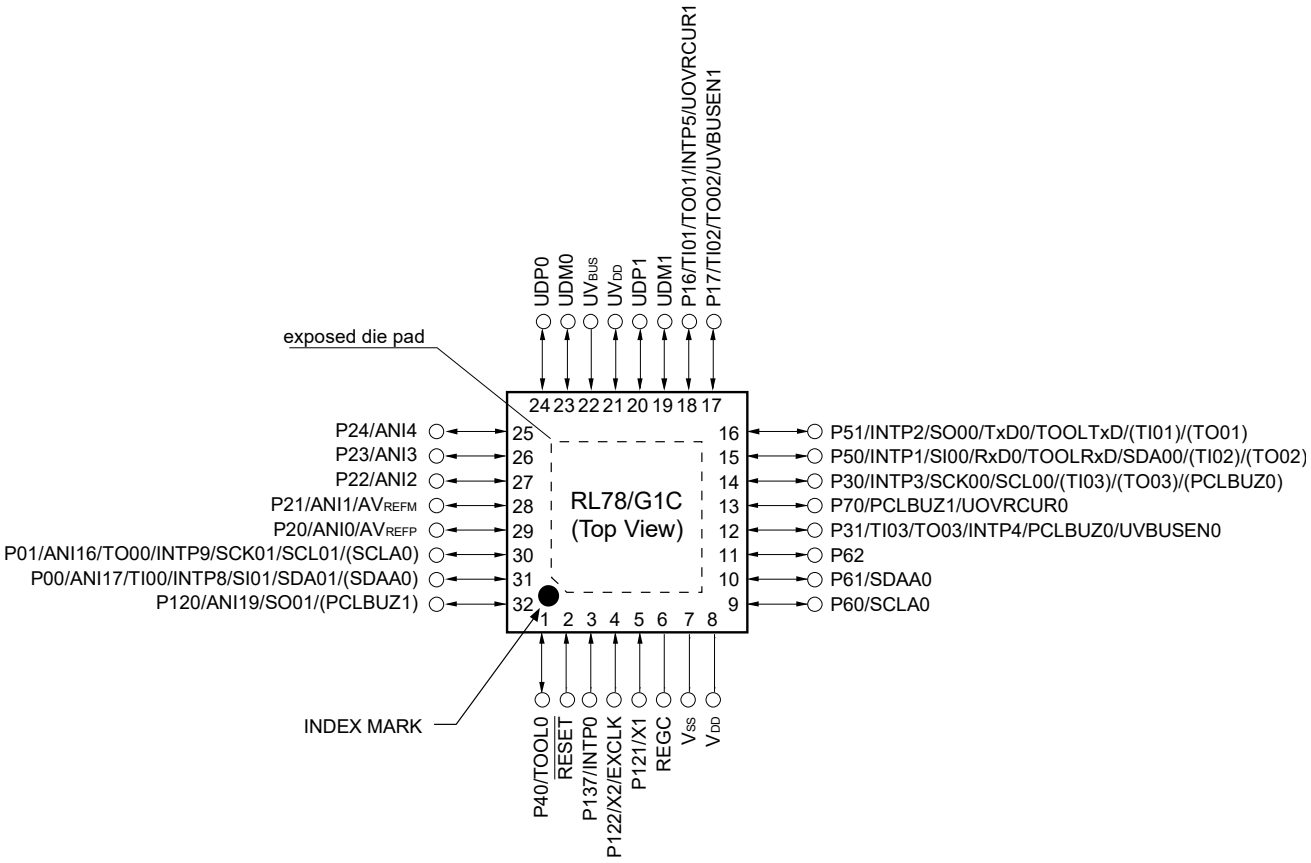
Caution The part number above is valid as of when this manual was issued. For the latest part number, see the web page of the target product on the Renesas Electronics website.

1.3 Pin Configuration (Top View)

1.3.1 32-pin products

- 32-pin plastic HWQFN (5 × 5 mm, 0.5 mm pitch)

(1) USB function: Host/Function controller (R5F10JBC)



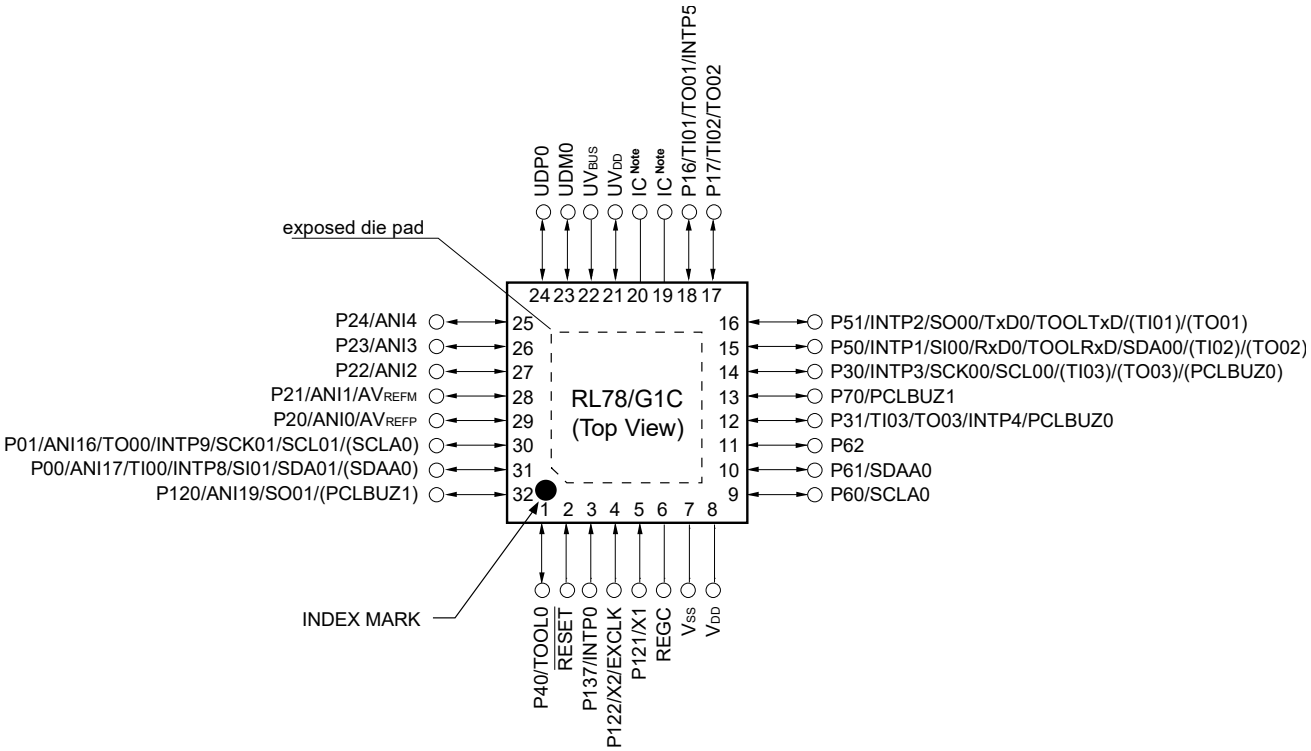
Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

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). Refer to **Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)**.

3. It is recommended to connect an exposed die pad to V_{SS}.

(2) USB function: Function controller only (R5F10KBC)



Note IC: Internal Connection Pin. Leave open.

Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

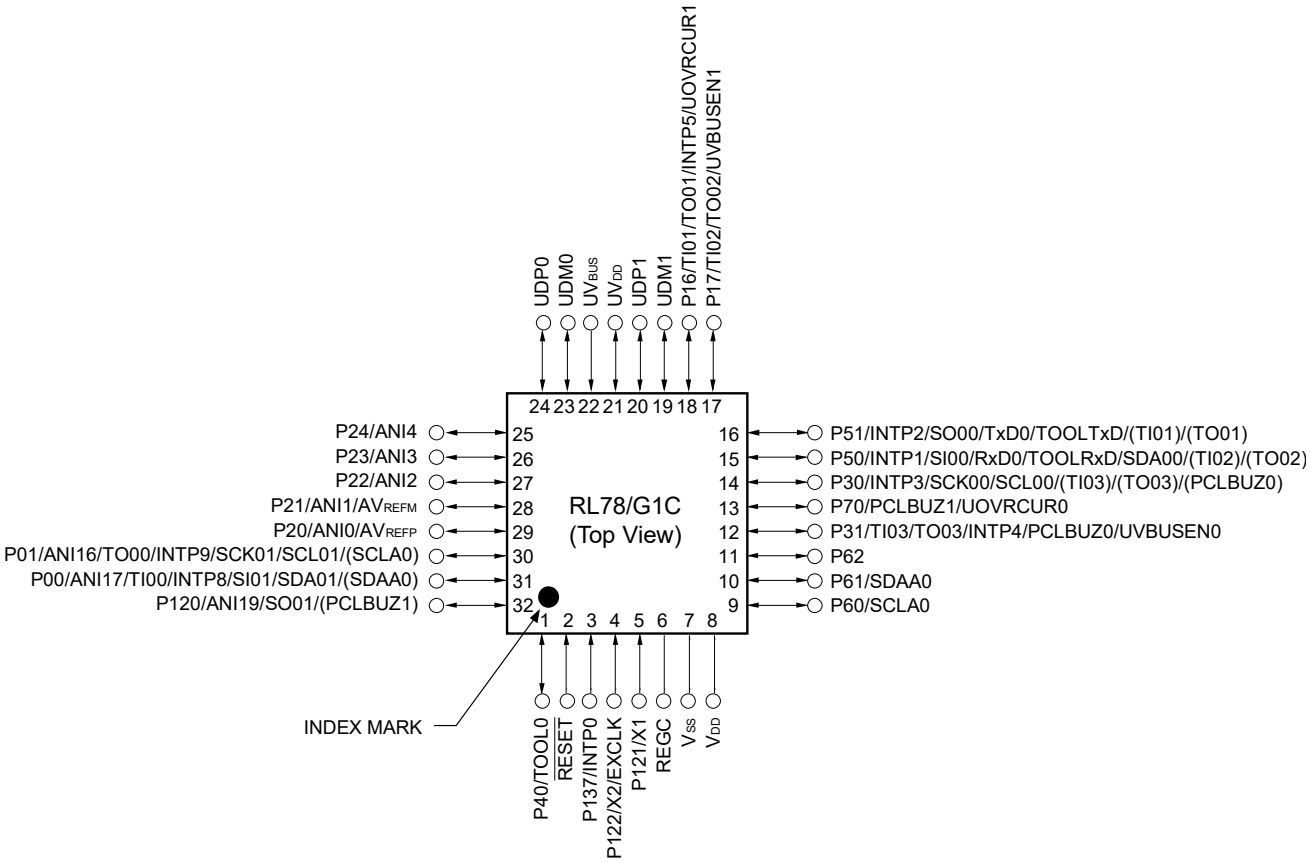
Remarks 1. For pin identification, see **1.4 Pin Identification**.

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)**.

3. It is recommended to connect an exposed die pad to V_{SS}.

- 32-pin plastic LQFP (7 × 7 mm, 0.8 mm pitch)

(1) USB function: Host/Function controller (R5F10JBC)

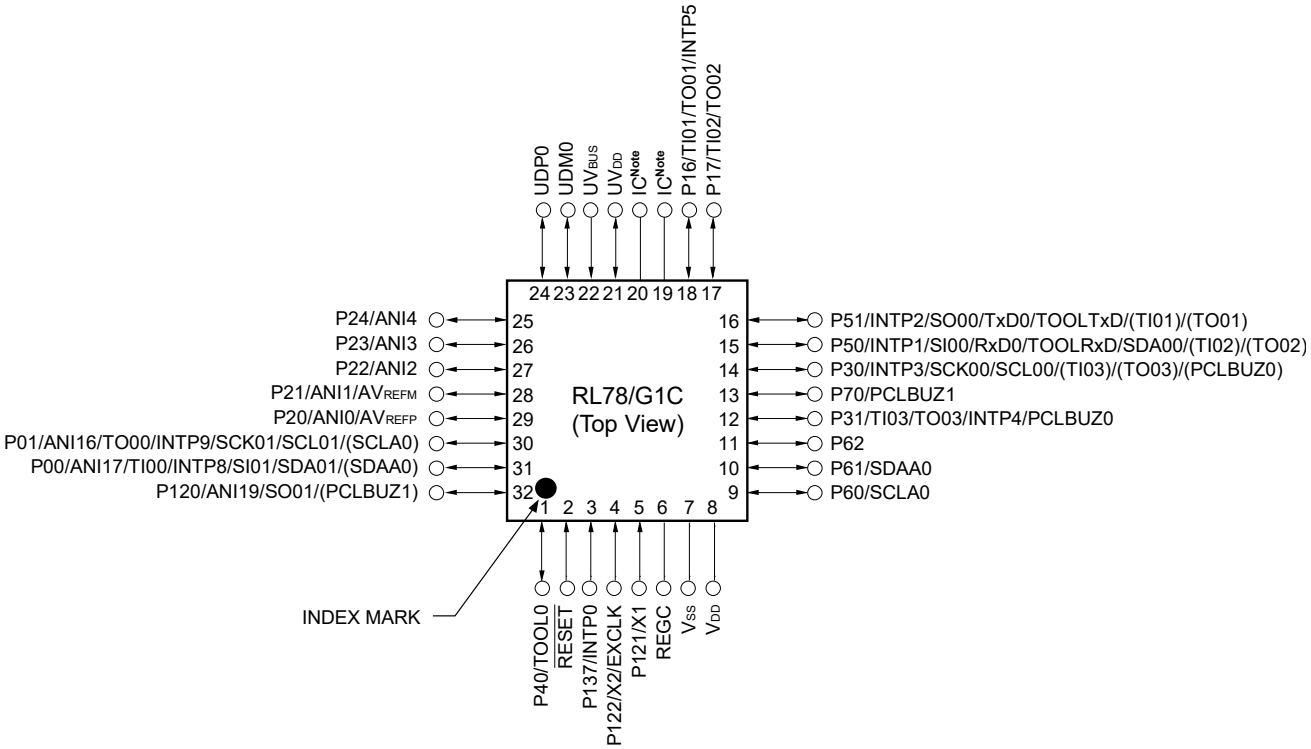


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

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). Refer to **Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)**.

(2) USB function: Function controller only (R5F10KBC)



Note IC: Internal Connection Pin Leave open.

Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

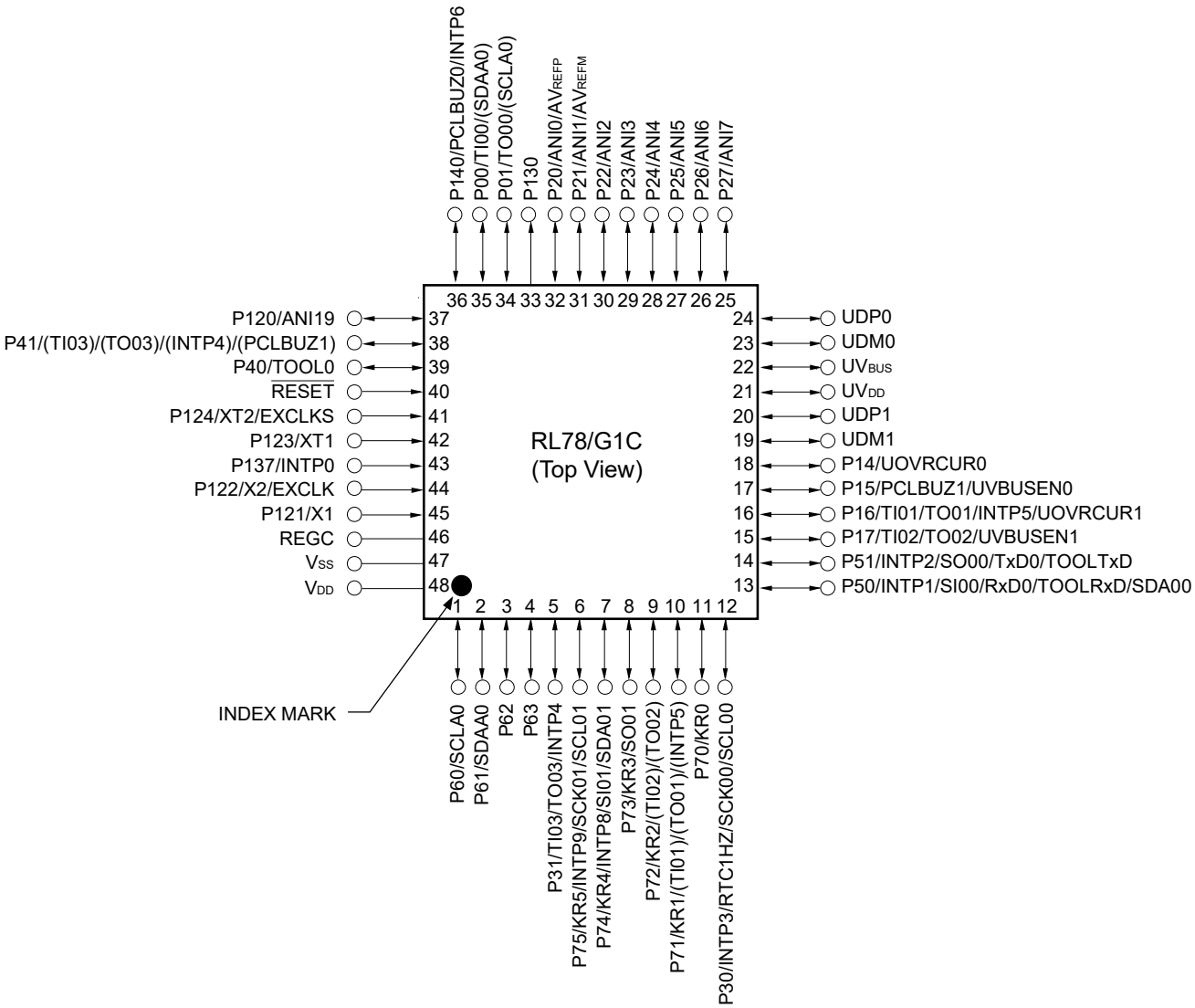
Remarks 1. For pin identification, see **1.4 Pin Identification**.

2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)**.

1.3.2 48-pin products

- 48-pin plastic LFQFP (fine pitch) (7 × 7 , 0.5 mm pitch)

(1) USB function: Host/Function controller (R5F10JGC)

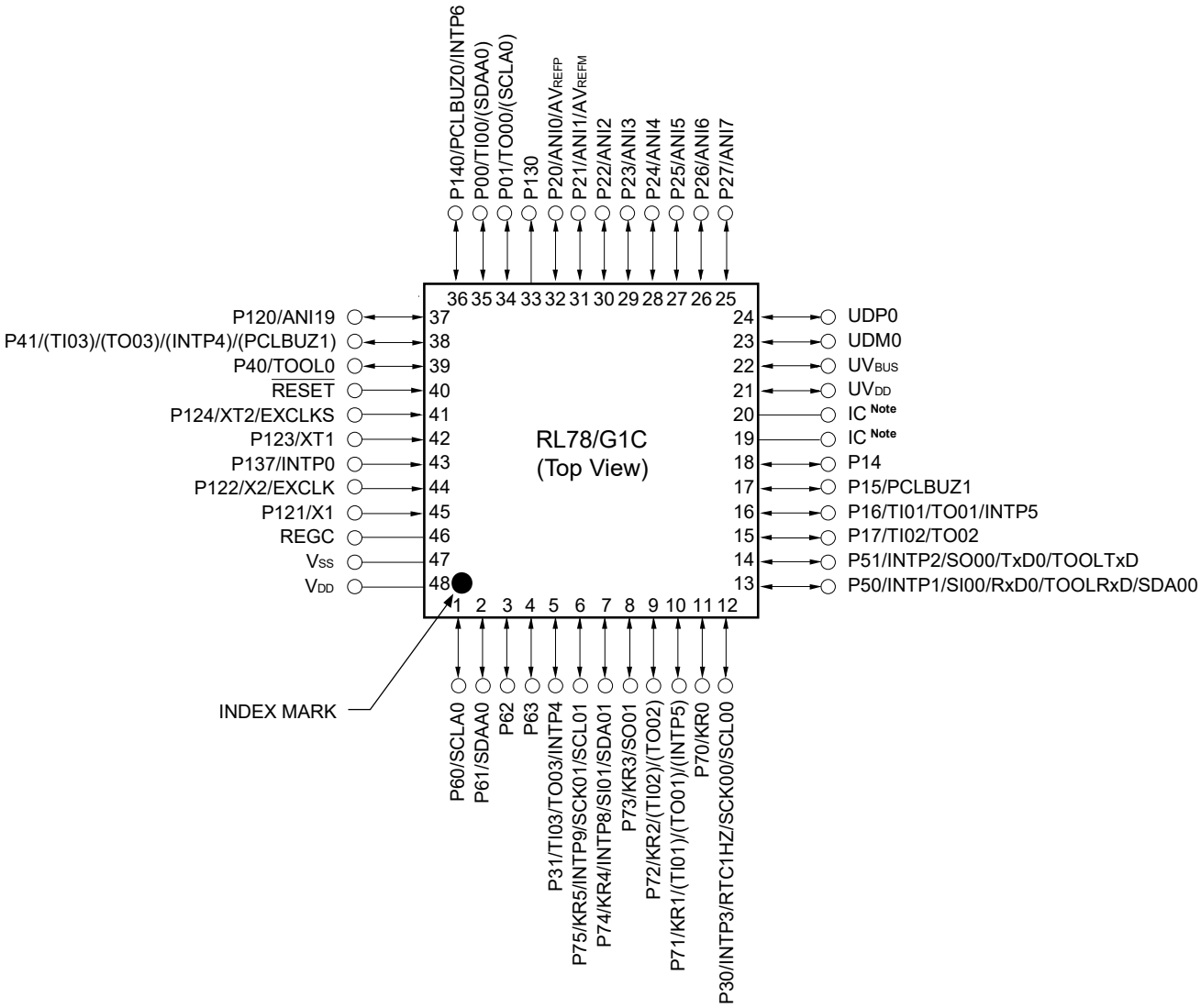


Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

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). Refer to **Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)**.

(2) USB function: Function controller only (R5F10KGC)



Note IC: Internal Connection Pin Leave open.

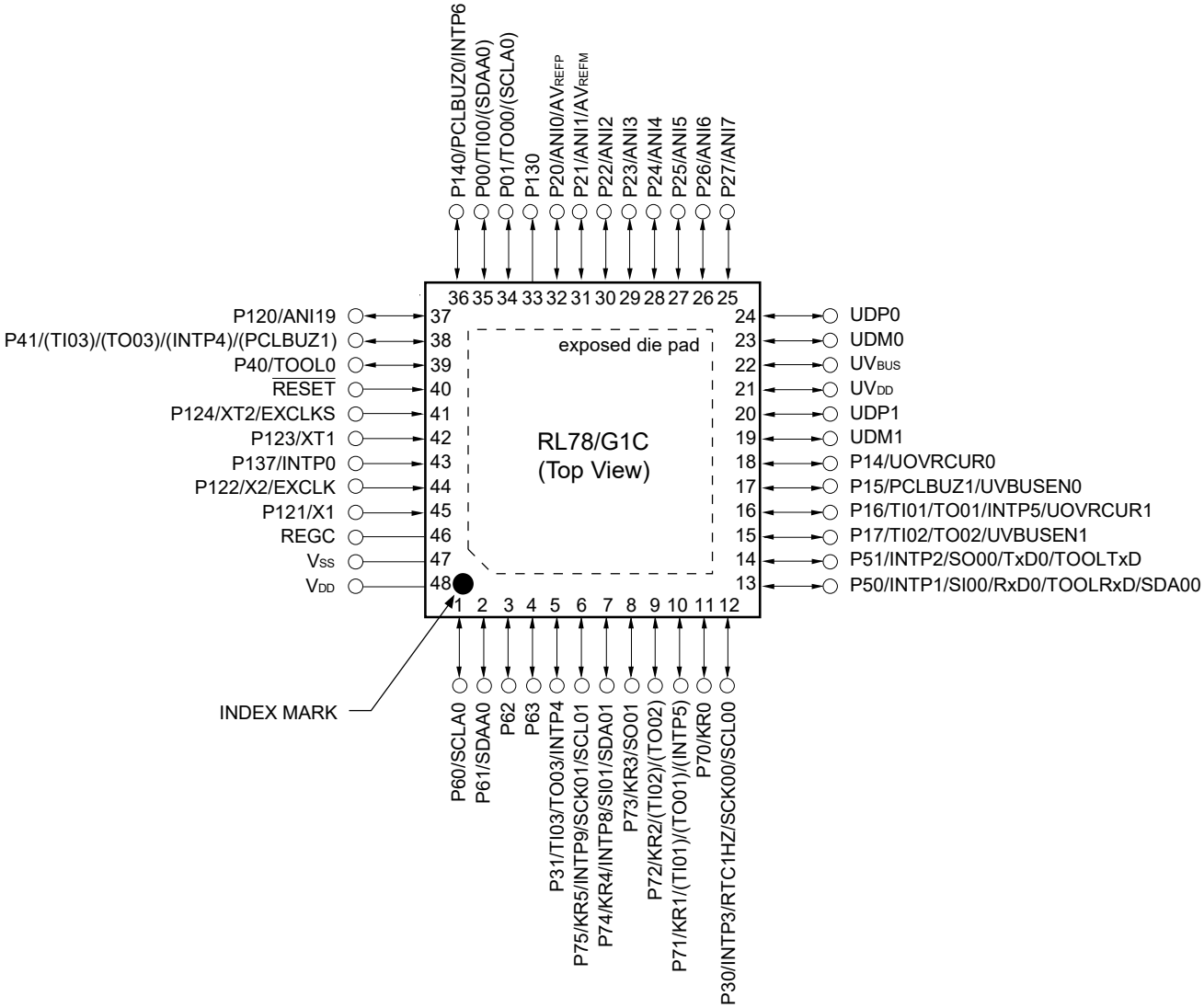
Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

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). Refer to **Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)**.

- 48-pin plastic WHQFN (7 × 7 , 0.5 mm pitch)

(1) USB function: Host/Function controller (R5F10JGC)



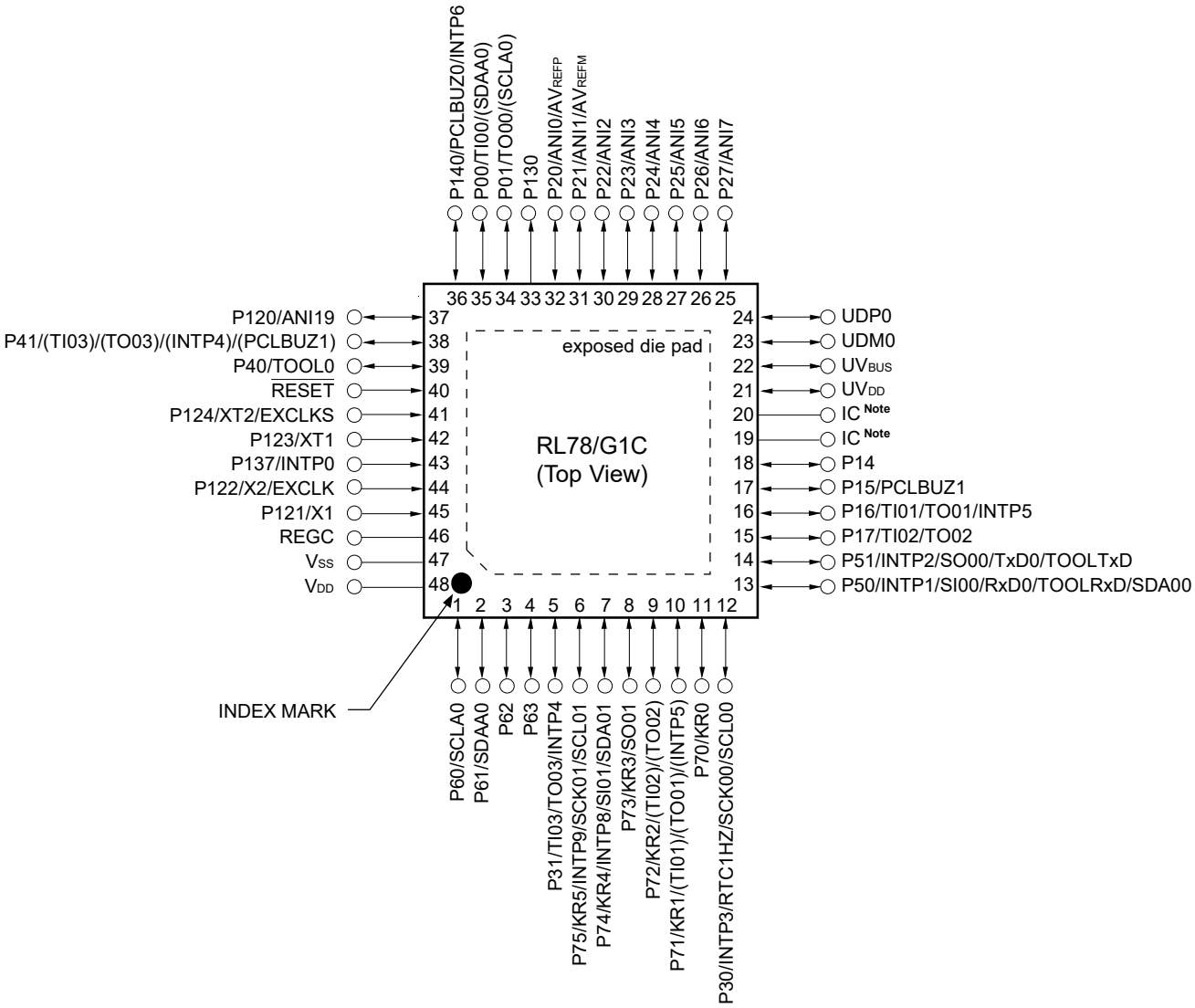
Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

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). Refer to **Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)**.

3. It is recommended to connect an exposed die pad to V_{SS}.

(2) USB function: Function controller only (R5F10KGC)



Note IC: Internal Connection Pin Leave open.

Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

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). Refer to **Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)**.

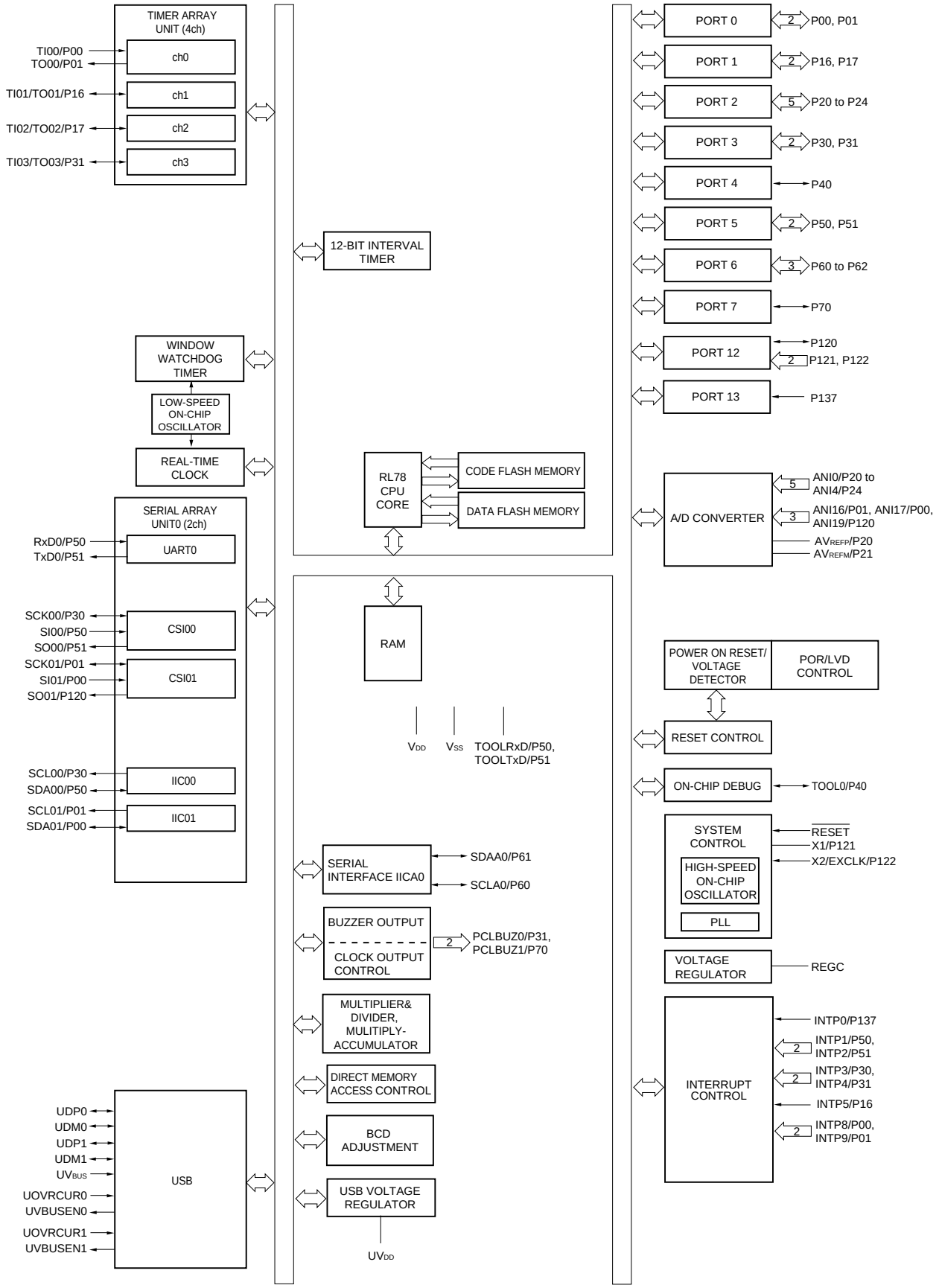
3. It is recommended to connect an exposed die pad to V_{SS}.

1.4 Pin Identification

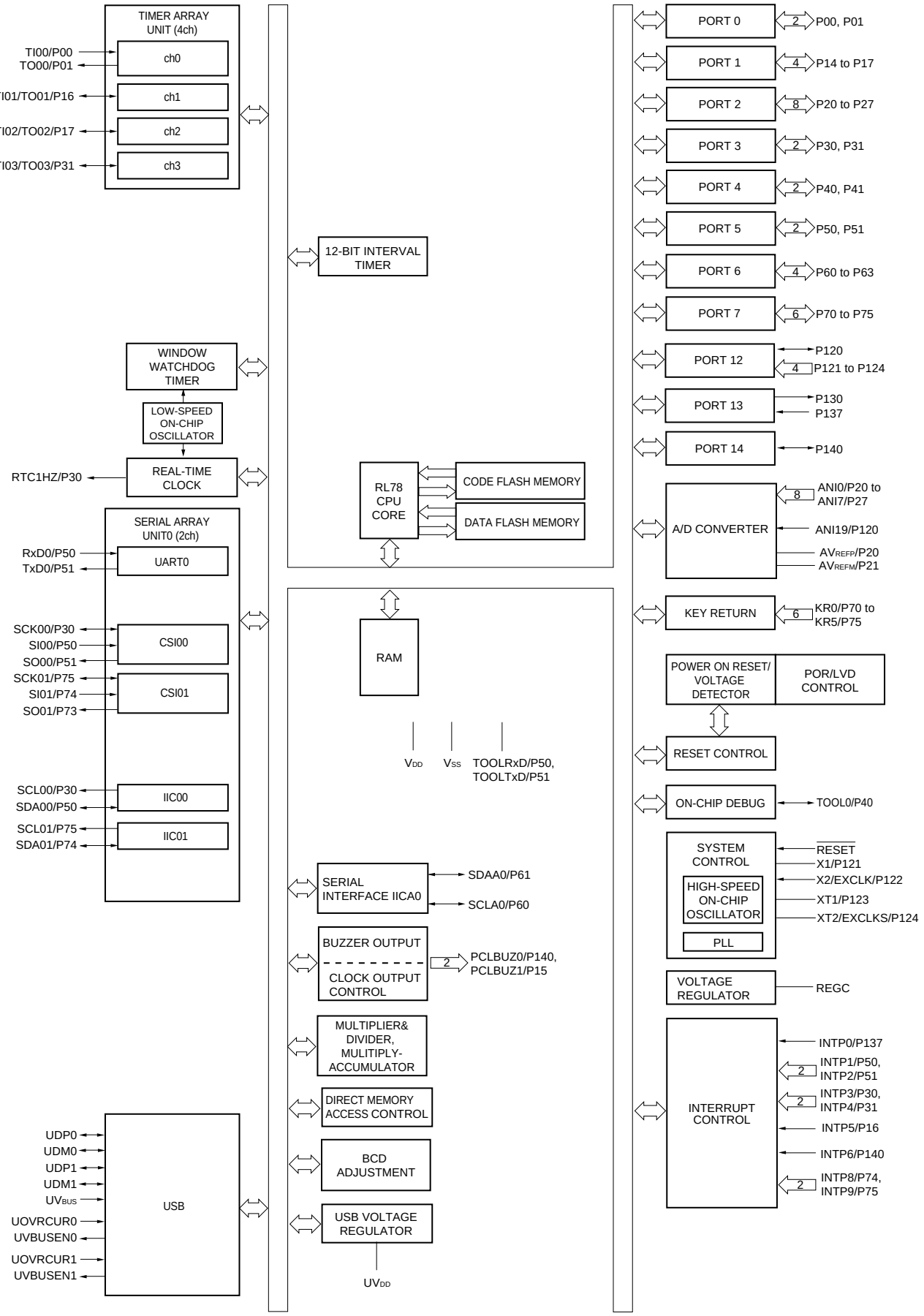
ANI0 to ANI7, ANI16, ANI17, ANI19:	Analog Input
AV_{REFM}:	Analog Reference Voltage Minus
AV_{REFP}:	Analog Reference Voltage Plus
EXCLK:	External Clock Input (Main System Clock)
EXCLKS:	External Clock Input (Sub System Clock)
INTP0 to INTP6, INTP8, INTP9:	External Interrupt Input
KR0 to KR5:	Key Return
P00, P01:	Port 0
P14 to P17:	Port 1
P20 to P27:	Port 2
P30, P31:	Port 3
P40, P41:	Port 4
P50, P51:	Port 5
P60 to P63:	Port 6
P70 to P75:	Port 7
P120 to P124:	Port 12
P130, P137:	Port 13
P140:	Port 14
PCLBUZ0, PCLBUZ1:	Programmable Clock Output/Buzzer Output
REGC:	Regulator Capacitance
$\overline{\text{RESET}}$:	Reset
RTC1HZ:	Real-time Clock Correction Clock (1 Hz) Output
RxD0:	Receive Data
SCK00, SCK01:	Serial Clock Input/Output
SCLA0, SCL00, SCL01:	Serial Clock Input/Output
SDAA0, SDA00, SDA01:	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
TOOLRxD, TOOLTxD:	Data Input/Output for External Device
TxD0:	Transmit Data
UDM0, UDM1, UDP0, UDP1:	USB Input/Output
UOVRCUR0, UOVRCUR1:	USB Input
UVBUSEN0, UVBUSEN1:	USB Output
UV_{DD}:	USB Power Supply/USB Regulator Capacitance
UV_{BUS}:	USB Input/USB Power Supply (USB Optional BC)
V_{DD}:	Power Supply
V_{SS}:	Ground
X1, X2:	Crystal Oscillator (Main System Clock)
XT1, XT2:	Crystal Oscillator (Subsystem Clock)

1.5 Block Diagram

1.5.1 32-pin products



1.5.2 48-pin products



1.6 Outline of Functions

[32-pin, 48-pin products]

(1/2)

Item		32-pin		48-pin	
		R5F10JBC	R5F10KBC	R5F10JGC	R5F10KGC
Code flash memory (KB)		32 KB		32 KB	
Data flash memory (KB)		2 KB		2 KB	
RAM (KB)		5.5 KB Note 1		5.5 KB Note 1	
Memory space		1 MB			
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) HS (High-speed main) mode: 1 to 20 MHz (V_{DD} = 2.7 to 5.5 V), HS (High-speed main) mode: 1 to 16 MHz (V_{DD} = 2.4 to 5.5 V)			
	High-speed on-chip oscillator	1 to 24 MHz (V_{DD} = 2.7 to 5.5 V), 1 to 16 MHz (V_{DD} = 2.4 to 5.5 V)			
	PLL clock	6, 12, 24 MHz Note 2 : V_{DD} = 2.4 to 5.5 V			
Subsystem clock		–		XT1 (crystal) oscillation 32.768 kHz (TYP.): V_{DD} = 2.4 to 5.5 V	
Low-speed on-chip oscillator		On-chip oscillation (Watchdog timer/Real-time clock/12-bit interval timer clock) 15 kHz (TYP.): V_{DD} = 2.4 to 5.5 V			
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)			
Minimum instruction execution time		0.04167 μs (High-speed on-chip oscillator: f_{HOCO} = 48 MHz /f_{IH} = 24 MHz operation)			
		0.04167 μs (PLL clock: f_{PLL} = 48 MHz /f_{IH} = 24 MHz Note 2 operation)			
		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)			
		–		30.5 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation)	
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits) • Rotate, barrel shift, and bit manipulation (Set, reset, test, and Boolean operation), etc.			
I/O port	Total	22		38	
	CMOS I/O	16 (N-ch O.D. I/O [V_{DD} withstand voltage]: 5)		28 (N-ch O.D. I/O [V_{DD} withstand voltage]: 6)	
	CMOS input	3		5	
	CMOS output	–		1	
	N-ch open-drain I/O (6 V tolerance)	3		4	
Timer	16-bit timer	4 channel			
	Watchdog timer	1 channel			
	Real-time clock (RTC)	1 channel Note 3			
	12-bit Interval timer (IT)	1 channel			
	Timer output	4 channels (PWM output: 3) Note 4			
	RTC output	–		1 • 1 Hz (subsystem clock: f_{SUB} = 32.768 kHz)	

Notes 1. In the case of the 5.5 KB, this is about 4.5 KB when the self-programming function is used. (For details, see **CHAPTER 3**)

2. In the PLL clock 48 MHz operation, the system clock is 2/4/8 dividing ratio.

3. In 32-pin products, this channel can only be used for the constant-period interrupt function based on the low-speed on-chip oscillator clock (f_{IL}).

4. The number of PWM outputs varies depending on the setting of channels in use (the number of masters and slaves). (**6.9.3 Operation as multiple PWM output function**)

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

(2/2)

Item		32-pin		48-pin	
		R5F10JBC	R5F10KBC	R5F10JGC	R5F10KGC
Clock output/buzzer output		2		2	
		• 2.93 kHz, 5.86 kHz, 11.7 kHz, 1.5 MHz, 3 MHz, 6 MHz, 12 MHz (Main system clock: f_{MAIN} = 24 MHz operation) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz operation)			
8/10-bit resolution A/D converter		8 channels		9 channels	
Serial interface		Simplified SPI (CSI): 2 channels/UART: 1 channel/simplified I²C: 2 channels			
	I²C bus	1 channel			
USB	Host controller	2 channels	–	2 channels	–
	Function controller	1 channel			
Multiplier and divider/multiply-accumulator		• Multiplier: 16 bits × 16 bits = 32 bits (Unsigned or signed) • Divider: 32 bits ÷ 32 bits = 32 bits (Unsigned) • Multiply-accumulator: 16 bits × 16 bits + 32 bits = 32 bits (Unsigned or signed)			
DMA controller		2 channels			
Vectored interrupt sources	Internal	20		20	
	External	8		10	
Key interrupt		–		6	
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detector • Internal reset by illegal instruction execution [Note] • Internal reset by RAM parity error • Internal reset by illegal-memory access			
Power-on-reset circuit		• Power-on-reset: 1.51 V (TYP.) • Power-down-reset: 1.50 V (TYP.)			
Voltage detector		2.45 V to 4.06 V (9 stages)			
On-chip debug function		Provided			
Power supply voltage		V_{DD} = 2.4 to 5.5 V			
Operating ambient temperature		T_A = –40 to +85 °C (A: Consumer applications), T_A = –40 to +105°C (G: Industrial applications)			

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

CHAPTER 2 PIN FUNCTIONS

2.1 Port Function

Pin I/O buffer power supplies depend on the product. The relationship between these power supplies and the pins is shown below.

Table 2-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
V_{DD}	All pins other than UDP0, UDM0, UDP1, and UDM1
UV_{DD}	UDP0, UDM0, UDP1, UDM1

2.1.1 32-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	8-3-4	I/O	Analog input port	ANI17/TI00/INTP8/ SI01/SDA01/ (SDAA0)	Port 0. 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. Input of P00 and P01 can be set to TTL input buffer. Output of P00 and P01 can be set to N-ch open-drain output (V_{DD} tolerance). P00 and P01 can be set to analog input. Note 1
P01				ANI16/TO00/ INTP9/SCK01/ SCL01/(SCLA0)	
P16	7-1-3	I/O	Input port	TI01/TO01/INTP5/ UOVRCUR1 Note 2	Port 1. 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.
P17				TI02/TO02/ UVBUSEN1 Note 2	
P20	4-3-3	I/O	Analog input port	ANI0/AV_{REFP}	Port 2. 5-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input Note 3.
P21				ANI1/AV_{REFM}	
P22				ANI2	
P23				ANI3	
P24				ANI4	
P30	8-1-4	I/O	Input port	INTP3/SCK00/SCL00/ (TI03)/(TO03)/ (PCLBUZ0)	Port 3. 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. Input of P30 can be set to TTL input buffer. Output of P30 can be set to N-ch open-drain output (V_{DD} tolerance).
P31	7-1-3			TI03/TO03/INTP4/PC LBUZ0/ UVBUSEN0 Note 2	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting.

Notes 1. Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).

2. This function has not been mounted in the R5F10K product.

3. Digital or analog for each pin can be selected with the A/D port configuration register (ADPC).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P50	8-1-4	I/O	Input port	INTP1/SI00/RxD0/ TOOLRxD/SDA00/ (TI02)/(TO02)	Port 5. 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. Input of P50 can be set to TTL input buffer. Output of P50 and P51 can be set to N-ch open-drain output (V_{DD} tolerance).
P51	7-1-4			INTP2/SO00/TxD0/ TOOLTxD/(TI01)/ (TO01)	
P60	12-1-3	I/O	Input port	SCLA0	Port 6. 3-bit I/O port. Input/output can be specified in 1-bit units. Output can be set to N-ch open-drain output (6 V tolerance).
P61				SDAA0	
P62	12-1-2			–	
P70	7-1-3	I/O	Input port	PCLBUZ1/ UOVRCUR0[Note 1]	Port 7. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P120	7-3-3	I/O	Analog input port	ANI19/SO01/ (PCLBUZ1)	Port 12. 1-bit I/O port and 2-bit input port. P120 can be set to analog input.[Note 2] For only P120, input/output can be specified. For only P120, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P121	2-2-1	Input	Input port	X1	
P122				X2/EXCLK	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input port.
$\overline{\text{RESET}}$	2-1-1	Input	–	–	Input only pin for external reset When external reset is not used, connect this pin to V_{DD} directly or via a resistor.
UDP0	18-11-1	I/O	–	–	D+ I/O pin of USB port 0 This pin should be connected to the D+ pin of the USB bus.
UDM0	18-11-1	I/O	–	–	D- I/O pin of USB port 0 This pin should be connected to the D- pin of the USB bus.
UV_{BUS}	17-11-1	Input	–	–	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. Whether VBUS is connected or disconnected can be detected during operation as a function controller.
UDP[Note 1]	18-11-1	I/O	–	–	D+ I/O pin of USB port 1 This pin should be connected to the D+ pin of the USB bus.
UDM1[Note 1]	18-11-1	I/O	–	–	D- I/O pin of USB port 1 This pin should be connected to the D- pin of the USB bus.

Notes 1. This function has not been mounted in the R5F10K product.

2. Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

2.1.2 48-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	8-3-4	I/O	Input port	TI00/(SDAA0)	Port 0. 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. Input of P00 and P01 can be set to TTL input buffer. Output of P00 and P01 can be set to N-ch open-drain output (V_{DD} tolerance). P00 and P01 can be set to analog input. Note 1
P01				TO00/(SCLA0)	
P14	7-1-3	I/O	Input port	UOVRCUR0 Note 2	Port 1. 4-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.
P15				PCLBUZ1/ UVBUSEN0 Note 2	
P16				TI01/TO01/INTP5/ UOVRCUR1 Note 2	
P17				TI02/TO02/ UVBUSEN1 Note 2	
P20	4-3-3	I/O	Analog input port	ANI0/AV_{REFP}	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units. Can be set to analog input Note 3.
P21				ANI1/AV_{REFM}	
P22				ANI2	
P23				ANI3	
P24				ANI4	
P25				ANI5	
P26				ANI6	
P27				ANI7	
P30	8-1-4	I/O	Input port	INTP3/ RTC1HZ/ SCK00/SCL00	Port 3. 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. Input of P30 can be set to TTL input buffer. Output of P30 can be set to N-ch open-drain output (V_{DD} tolerance).
P31	7-1-3			TI03/TO03/INTP4	
P40	7-1-3	I/O	Input port	TOOL0	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)/(TO03)/ (INTP4)/(PCLBUZ1)	

Notes 1. Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).

2. This function has not been mounted in the R5F10K product.

3. Digital or analog for each pin can be selected with the A/D port configuration register (ADPC).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P50	8-1-4	I/O	Input port	INTP1/SI00/RxD0/ TOOLRxD/SDA00	Port 5. 2-bit I/O port. Input of P50 can be set to TTL input buffer. Output of P50 and P51 can be set to N-ch open-drain output (V_{DD} tolerance). 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.
P51	7-1-4			INTP2/SO00/TxD0/ TOOLTxD	
P60	12-1-3	I/O	Input port	SCLA0	Port 6. 4-bit I/O port. Output can be set to N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.
P61				SDAA0	
P62	12-1-2			–	
P63				–	
P70	7-1-3	I/O	Input port	KR0	Port 7. 6-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 P74 can be set to N-ch open-drain output (V_{DD} tolerance).
P71				KR1/(TI01)/(TO01)/ (INTP5)	
P72				KR2/(TI02)/(TO02)	
P73				KR3/SO01	
P74	7-1-4			KR4/INTP8/SI01/ SDA01	
P75	7-1-3			KR5/INTP9/SCK01/ SCL01	
P120	7-3-3	I/O	Analog input port	ANI19	Port 12. 1-bit I/O port and 4-bit input port. P120 can be set to analog input. For only P120, input/output can be specified. For only P120, use of an on-chip pull-up resistor can be specified by a software setting at input port. Note
P121	2-2-1	Input	Input port	X1	
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P130	1-1-1	Output	Output port	–	Port 13. 1-bit input-only port and 1-bit output-only port.
P137	2-1-2	Input	Input port	INTP0	
P140	7-1-3	I/O	Input port	PCLBUZ0/INTP6	Port 14. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
$\overline{\text{RESET}}$	2-1-1	Input	–	–	Input only pin for external reset When external reset is not used, connect this pin to V_{DD} directly or via a resistor.

Note Digital or analog for each pin can be selected with the port mode control register x (PMCx) (can be set in 1-bit unit).

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-8 Format of Peripheral I/O Redirection Register (PIOR)**.

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
UDP0	18-11-1	I/O	–	–	D+ I/O pin of USB port 0 This pin should be connected to the D+ pin of the USB bus.
UDM0	18-11-1	I/O	–	–	D- I/O pin of USB port 0 This pin should be connected to the D- pin of the USB bus.
UV_{BUS}	17-11-1	Input	–	–	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. Whether VBUS is connected or disconnected can be detected during operation as a function controller.
UDP1 Note	18-11-1	I/O	–	–	D+ I/O pin of USB port 1 This pin should be connected to the D+ pin of the USB bus.
UDM1 Note	18-11-1	I/O	–	–	D- I/O pin of USB port 1 This pin should be connected to the D- pin of the USB bus.

Note This function has not been mounted in the R5F10K product.

2.2 Functions Other than Port Pins

(1/3)

Function Name	I/O	Function	48-pin	32-pin
ANI0	Input	A/D converter analog input	√	√
ANI1			√	√
ANI2			√	√
ANI3			√	√
ANI4			√	√
ANI5			√	–
ANI6			√	–
ANI7			√	–
ANI16			–	√
ANI17			–	√
ANI19			√	√
EXCLK	Input	External clock input for main system clock	√	√
EXCLKS	Input	External clock input for subsystem clock	√	–
INTP0	Input	External interrupt request input	√	√
INTP1			√	√
INTP2			√	√
INTP3			√	√
INTP4			√	√
INTP5			√	√
INTP6			√	–
INTP8			√	√
INTP9			√	√
KR0	Input	Key interrupt input	√	–
KR1			√	–
KR2			√	–
KR3			√	–
KR4			√	–
KR5			√	–
PCLBUZ0	Output	Clock output/buzzer output	√	√
PCLBUZ1			√	√
REGC	–	Connecting regulator output stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	√	√
RTC1HZ	Output	Real-time clock correction clock (1 Hz) output	√	–

(2/3)

Function Name	I/O	Function	48-pin	32-pin
$\overline{\text{RESET}}$	Input	System reset input	√	√
RxD0	Input	Serial data input to UART0	√	√
SCK00	I/O	Clock input/output for CSI00	√	√
SCK01			√	√
SCLA0	I/O	Clock input/output for IICA0	√	√
SCL00	Output	Clock output for IIC00, IIC01	√	√
SCL01			√	√
SDAA0	I/O	Serial data I/O for IICA0	√	√
SDA00	I/O	Serial data I/O for IIC00, IIC01	√	√
SDA01			√	√
SI00	Input	Serial data input to CSI00, CSI01	√	√
SI01			√	√
SO00	Output	Serial data output from CSI00, CSI01	√	√
SO01			√	√
TI00	Input	External count clock input to 16-bit timer 00	√	√
TI01		External count clock input to 16-bit timer 01	√	√
TI02		External count clock input to 16-bit timer 02	√	√
TI03		External count clock input to 16-bit timer 03	√	√
TO00	Output	16-bit timer 00 output	√	√
TO01		16-bit timer 01 output	√	√
TO02		16-bit timer 02 output	√	√
TO03		16-bit timer 03 output	√	√
TxD0	Output	Serial data output from UART0	√	√
X1	–	Resonator connection for main system clock	√	√
X2	–		√	√
XT1	–	Resonator connection for subsystem clock	√	–
XT2	–		√	–

(3/3)

Function Name	I/O	Function	48-pin	32-pin
V_{DD}	–	Positive power supply for all pins	√	√
AV_{REFP}	Input	A/D converter reference potential (+ side) input	√	√
AV_{REFM}	Input	A/D converter reference potential (– side) input	√	√
V_{SS}	–	Ground potential for all pins	√	√
TOOLRxD	Input	UART reception pin for the external device connection used during flash memory programming	√	√
TOOLTxD	Output	UART transmission pin for the external device connection used during flash memory programming	√	√
TOOL0	I/O	Data I/O for flash memory programmer/debugger	√	√
UV_{DD}	–	Power supply for USB	√	√
UV_{BUS}	Input	VBUS input	√	√
UDP0	I/O	USB data input/output (+ side) (USB port 0)	√	√
UDM0	I/O	USB data input/output (– side) (USB port 0)	√	√
UDP1	I/O	USB data input/output (+ side) (USB port 1)	√ Note	√ Note
UDM1	I/O	USB data input/output (– side) (USB port 1)	√ Note	√ Note
UVBUSEN0	Output	VBUS supply permit output (for USB port 0)	√ Note	√ Note
UOVRCUR0	Input	Overcurrent detection input (for USB port 0)	√ Note	√ Note
UVBUSEN1	Output	VBUS supply permit output (for USB port 1)	√ Note	√ Note
UOVRCUR1	Input	Overcurrent detection input (for USB port 1)	√ Note	√ Note

Note This function has not been mounted in the R5F10K product.

2.3 Connection of Unused Pins

Table 2-3 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 Function.**

Table 2-3. Connections of Unused Pins

Pin Name	I/O	Recommended Connection of Unused Pins
P00, P01	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P14 to P17	I/O	
P20 to P27	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P30, P31	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P40	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P41	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P50, P51	I/O	
P60 to P63	I/O	
P70 to P75	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P120	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P121 to P124	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P130	Output	Leave open.
P137	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P140	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
$\overline{\text{RESET}}$	Input	Connect directly or via a resistor to V_{DD}.
REGC	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).
UV_{DD}	–	• When the USB power supply is not used, connect directly to V_{DD}, or draw power from an external 3.3 V supply. • When the USB power is generated in the chip, connect to V_{SS} via capacitor (0.33 μF).
UV_{BUS}	Input	Independently connect to V_{SS} via a resistor.
<R> UDM0, UDP0	I/O	Leave open.
UDM1 Note, UDP1 Note	I/O	

Note This function is not mounted in the R5F10K product.

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

2.4 Block Diagrams of Pins

Figures 2-1 to 2-14 show the block diagrams of the pins described in **2.1.1 32-pin products** to **2.1.2 48-pin products**.

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

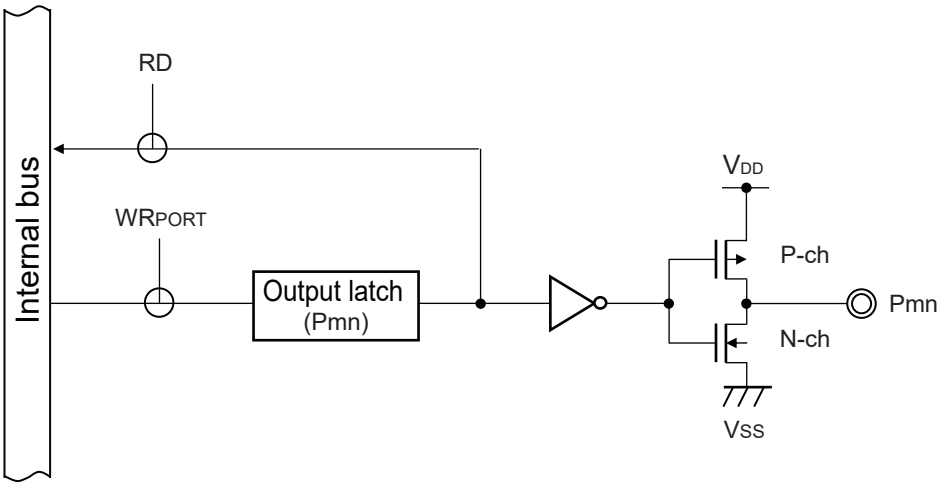


Figure 2-2. Pin Block Diagram for Pin Type 2-1-1

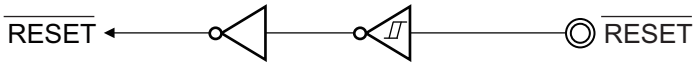
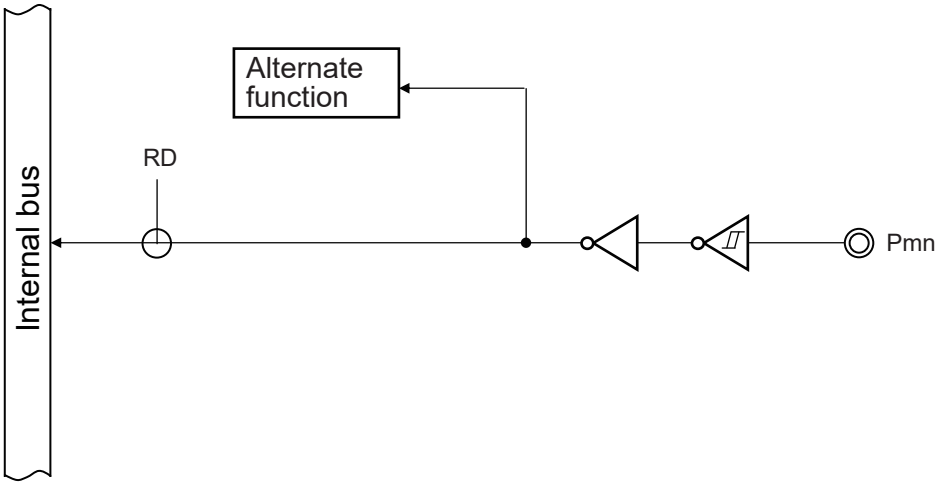
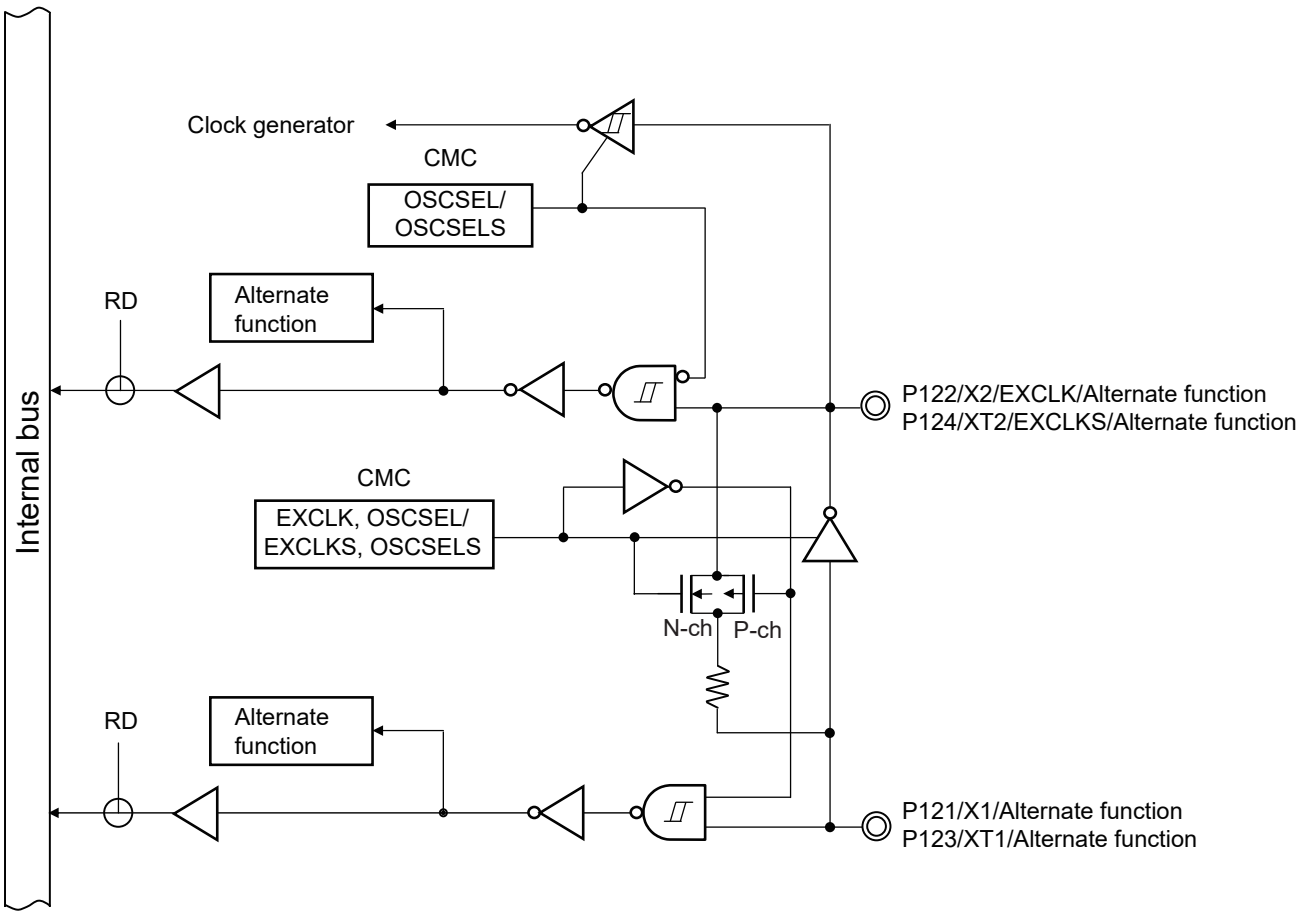


Figure 2-3. Pin Block Diagram for Pin Type 2-1-2



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

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



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

Figure 2-5. Pin Block Diagram for Pin Type 4-3-3

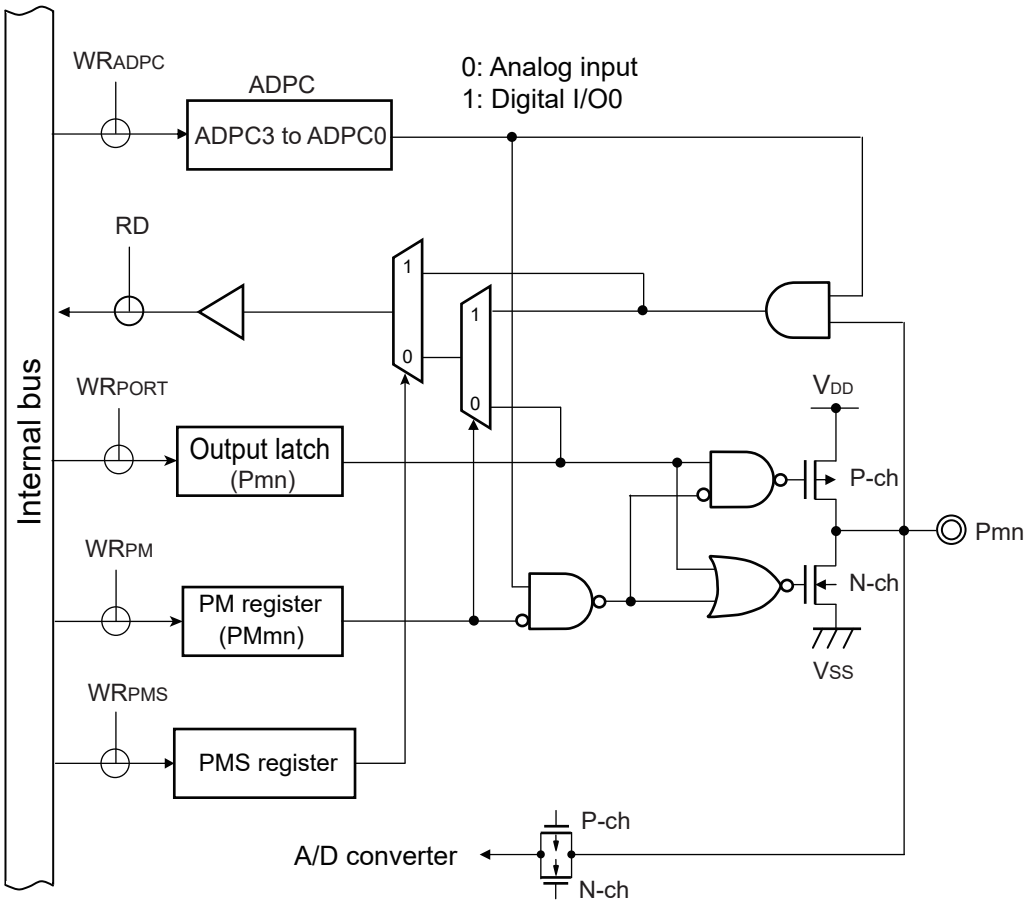
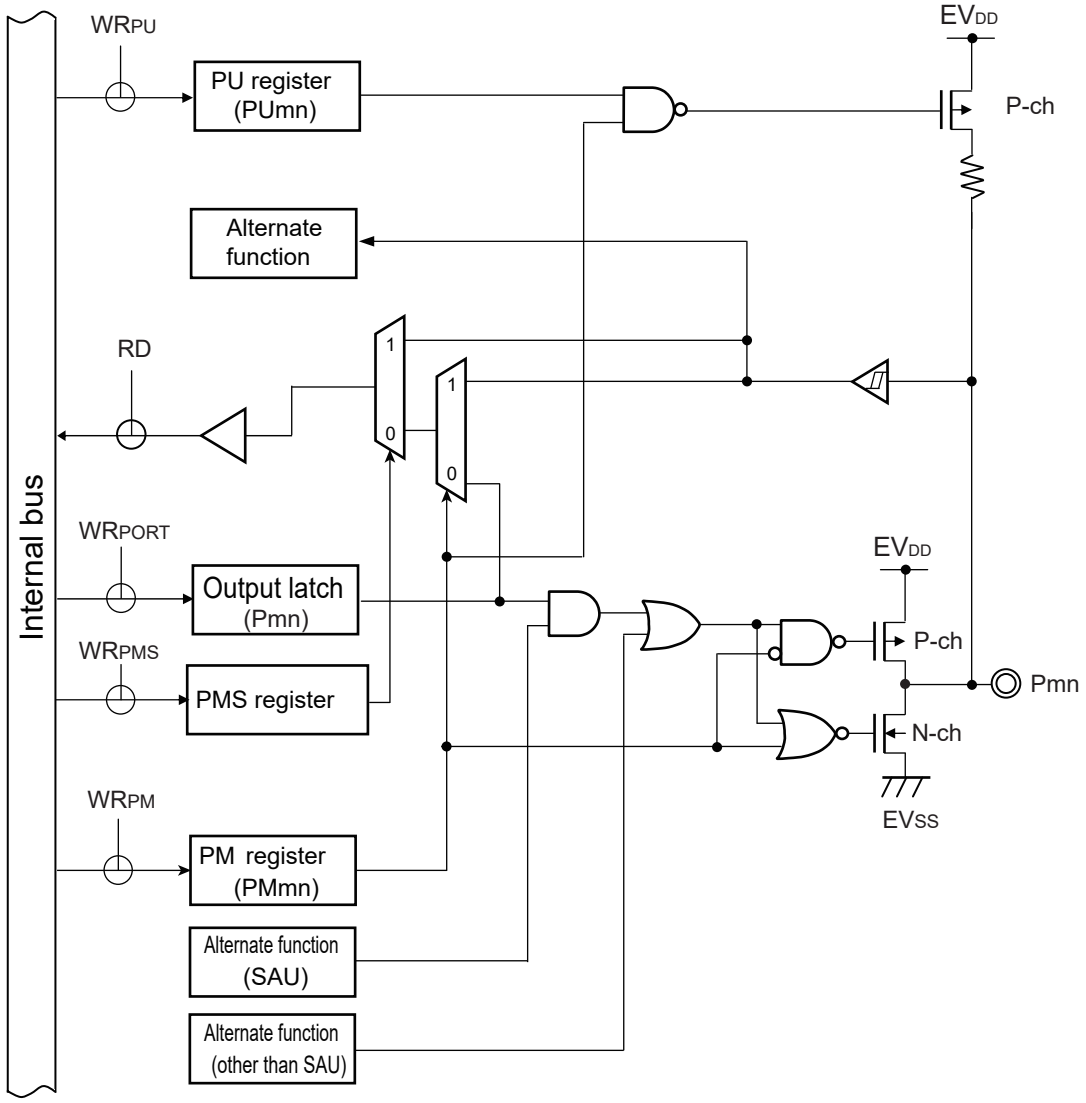


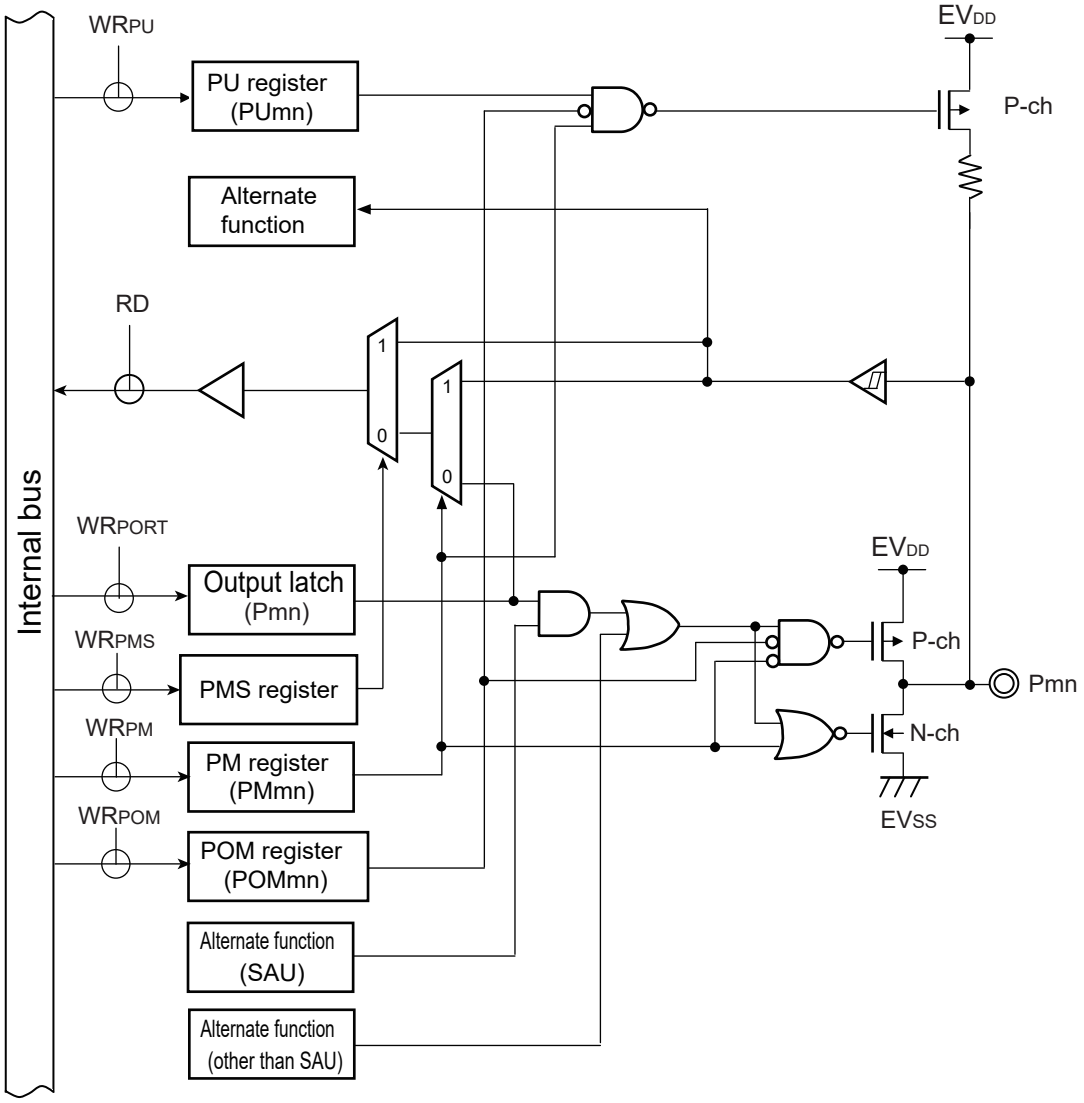
Figure 2-6. Pin Block Diagram for Pin Type 7-1-3



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

2. SAU: Serial array unit

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

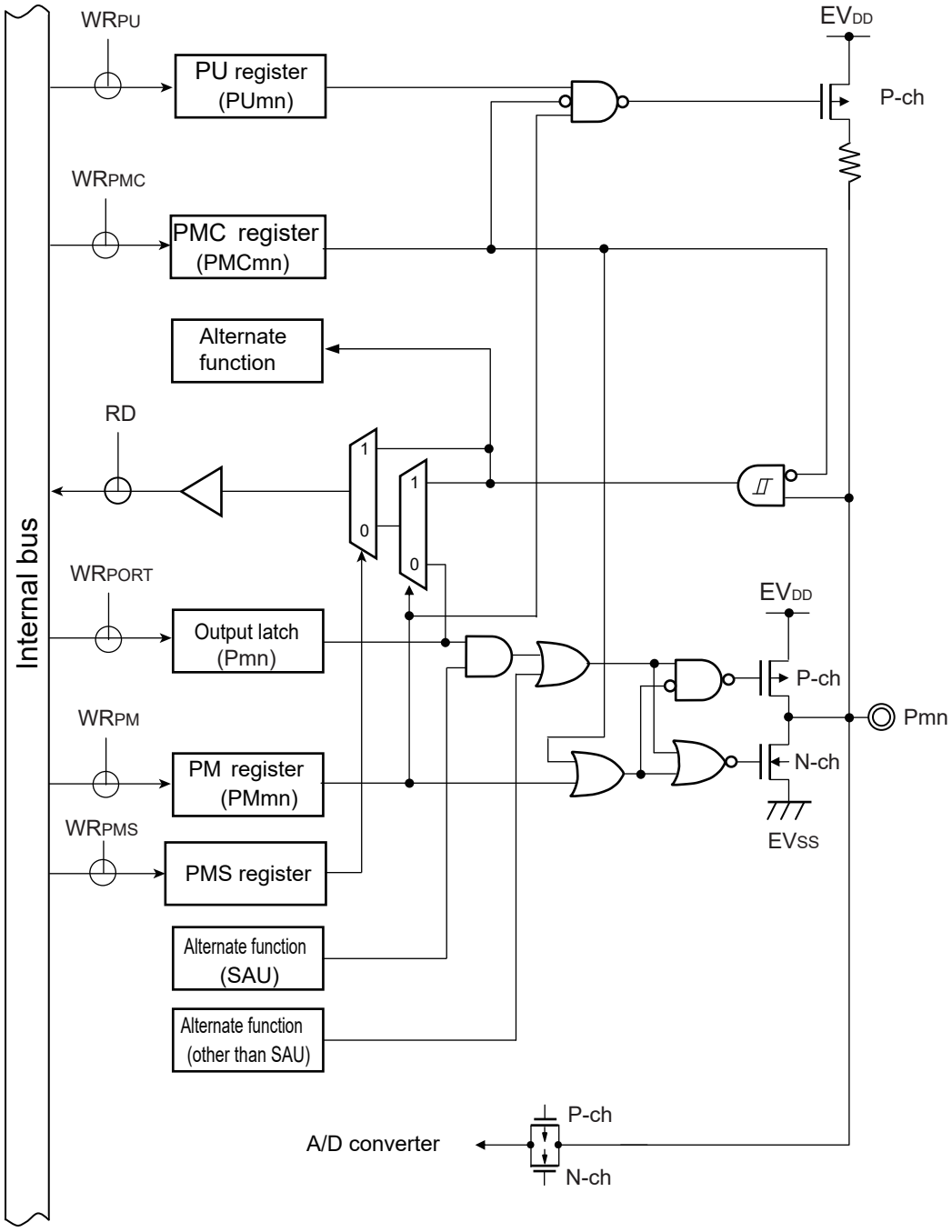


Caution A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

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

2. SAU: Serial array unit

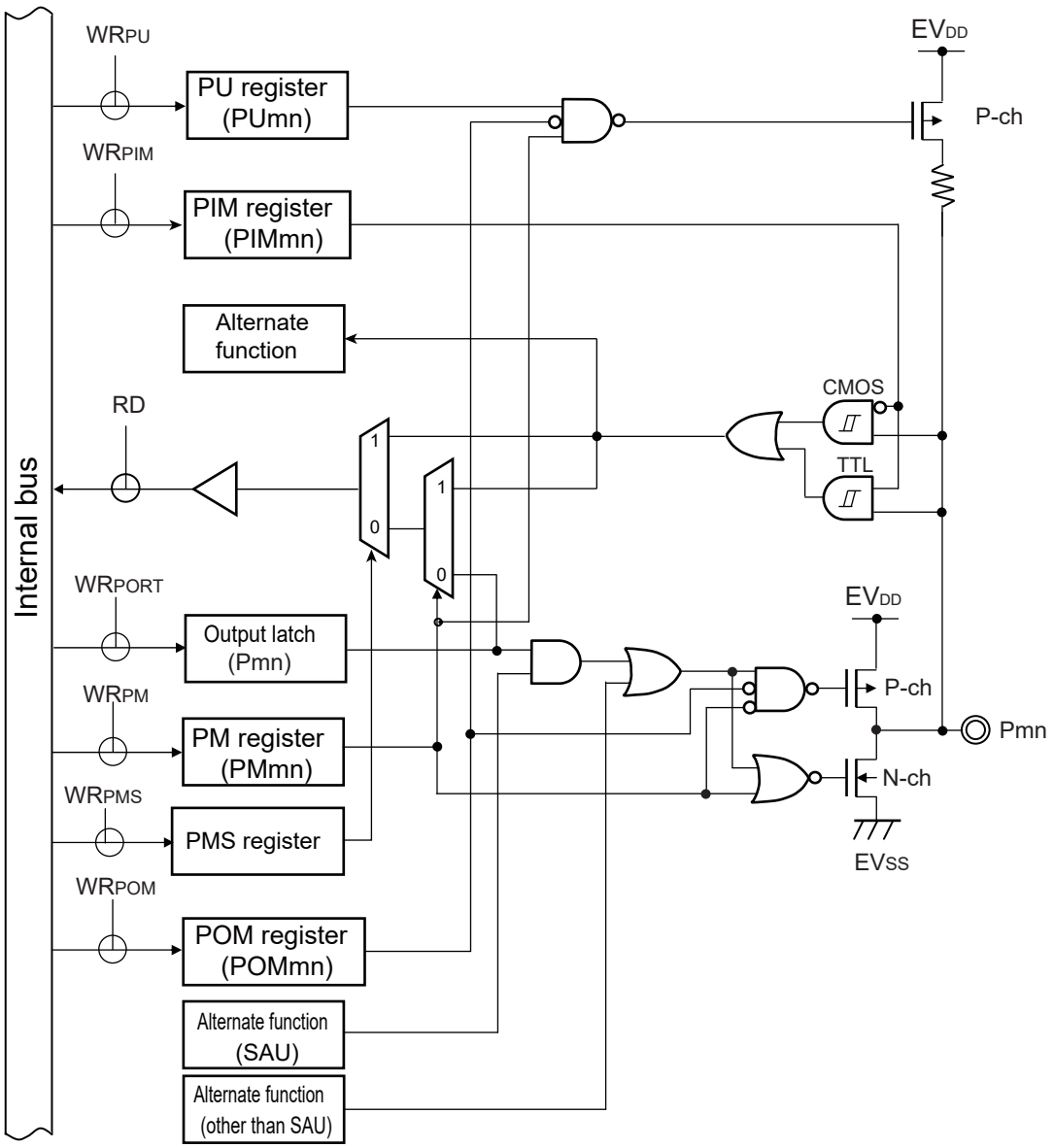
Figure 2-8. Pin Block Diagram for Pin Type 7-3-3



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

2. SAU: Serial array unit

Figure 2-9. Pin Block Diagram for Pin Type 8-1-4



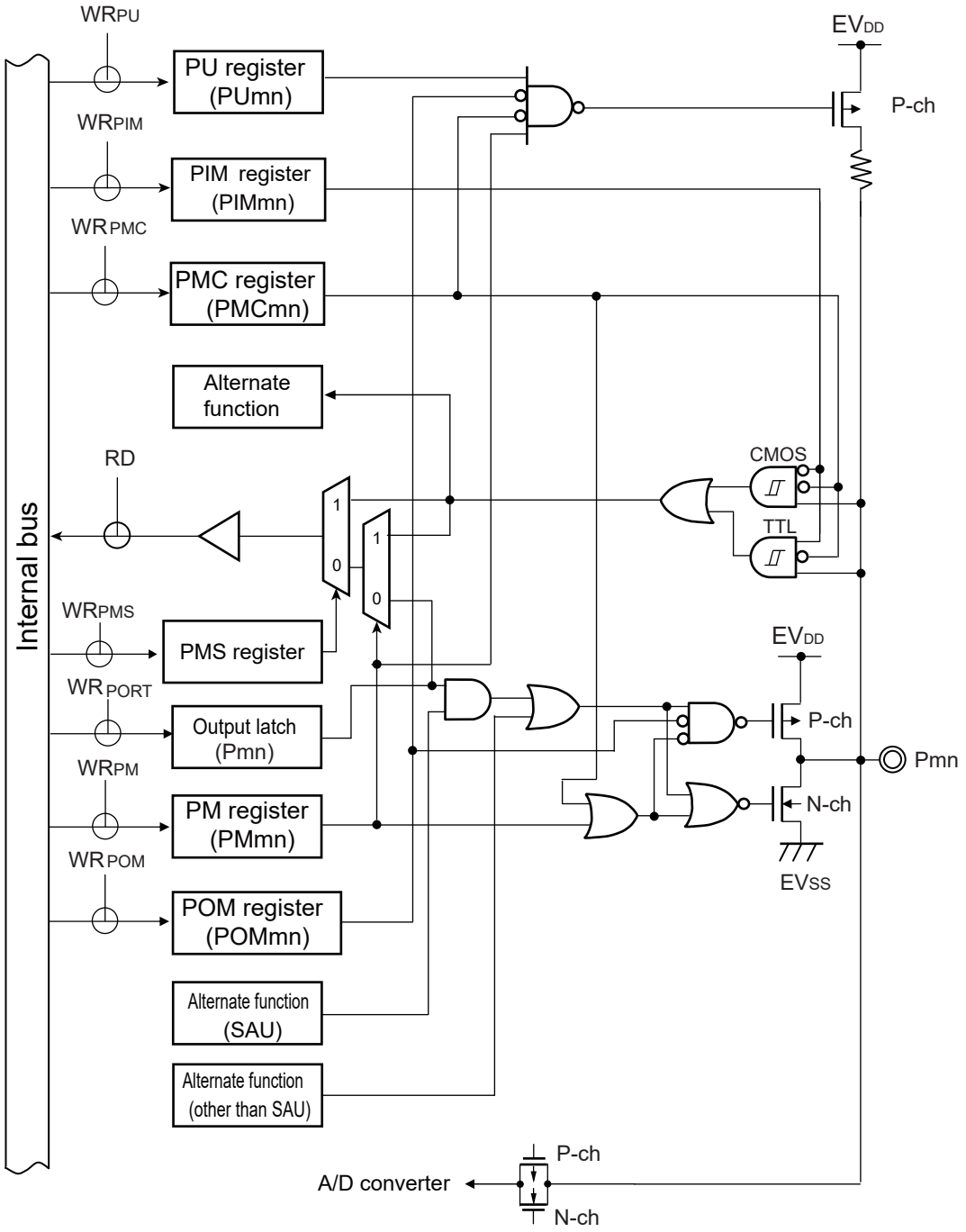
Cautions 1. A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

2. Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, a through current may flow through in the case of high level input. It is recommended to input a low level to prevent a through current.

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

2. SAU: Serial array unit

Figure 2-10. Pin Block Diagram for Pin Type 8-3-4



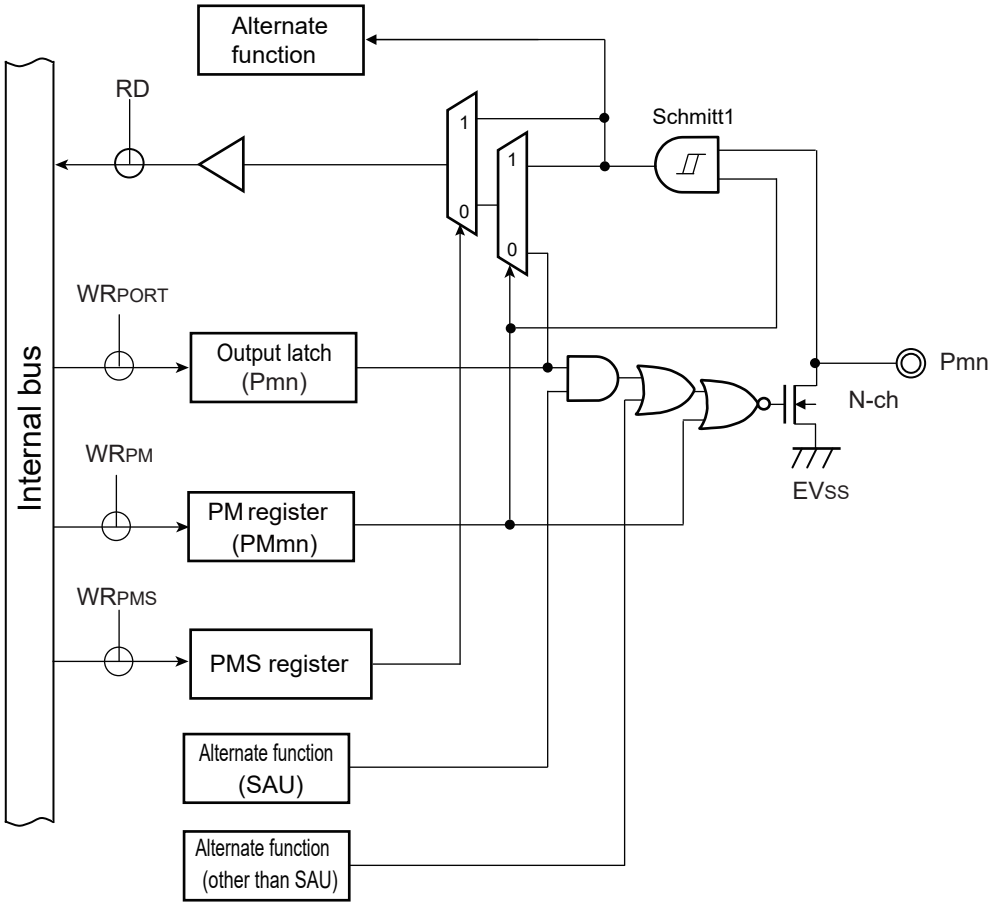
Cautions 1. A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

2. Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, a through current may flow through in the case of high level input. It is recommended to input a low level to prevent a through current.

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

2. SAU: Serial array unit

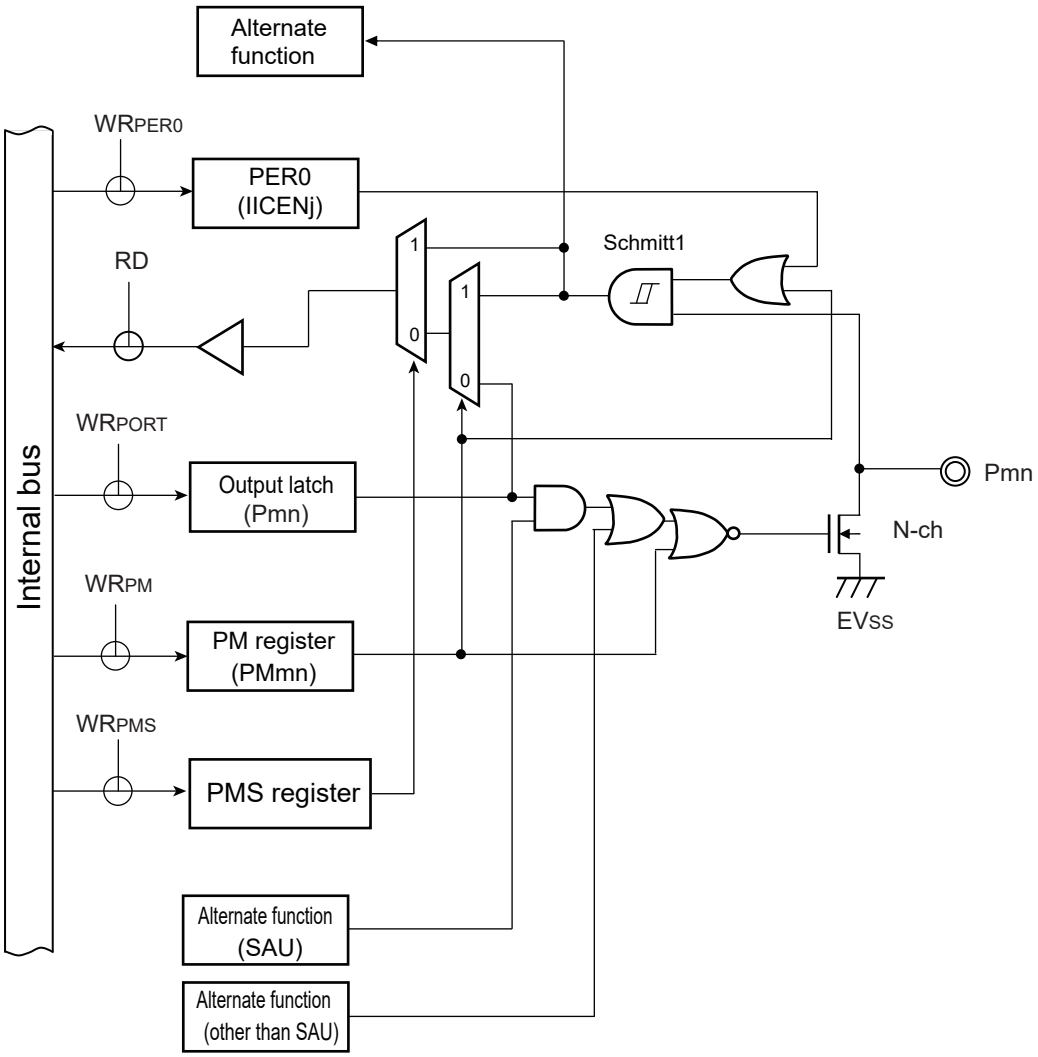
Figure 2-11. Pin Block Diagram for Pin Type 12-1-2



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

2. SAU: Serial array unit

Figure 2-12. Pin Block Diagram for Pin Type 12-1-3



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

2. SAU: Serial array unit

Figure 2-13. Pin Block Diagram for Pin Type 17-11-1

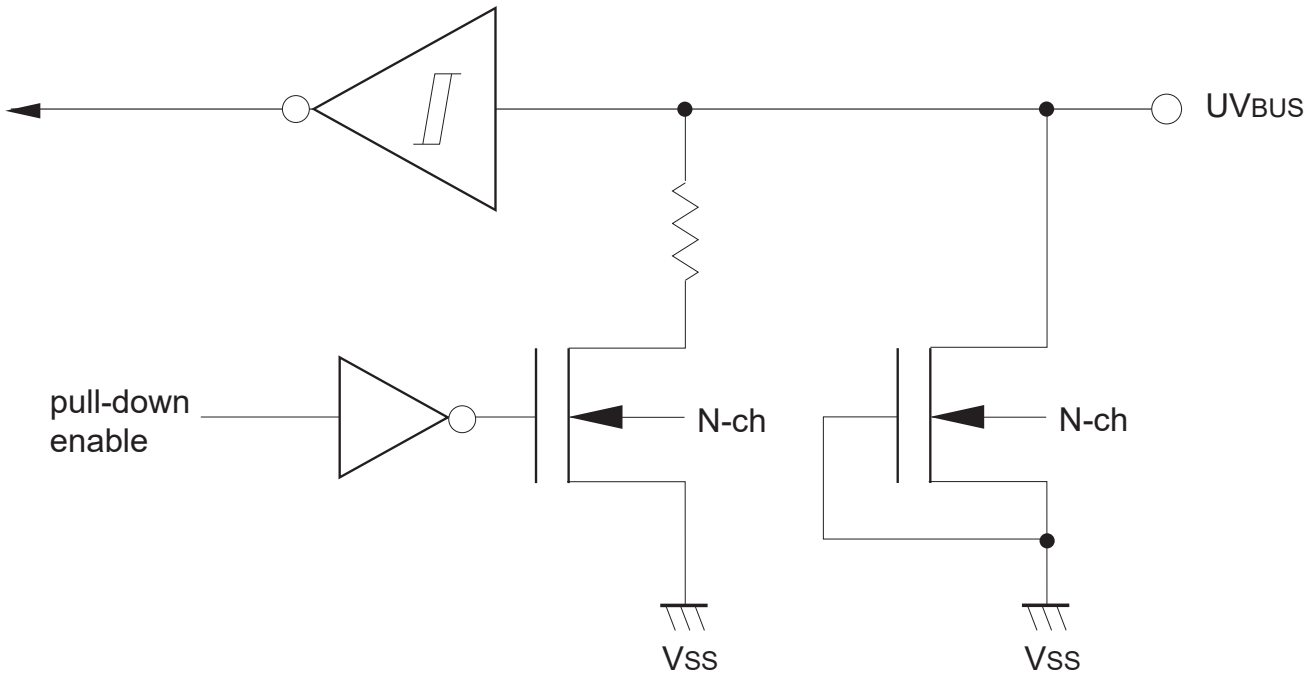
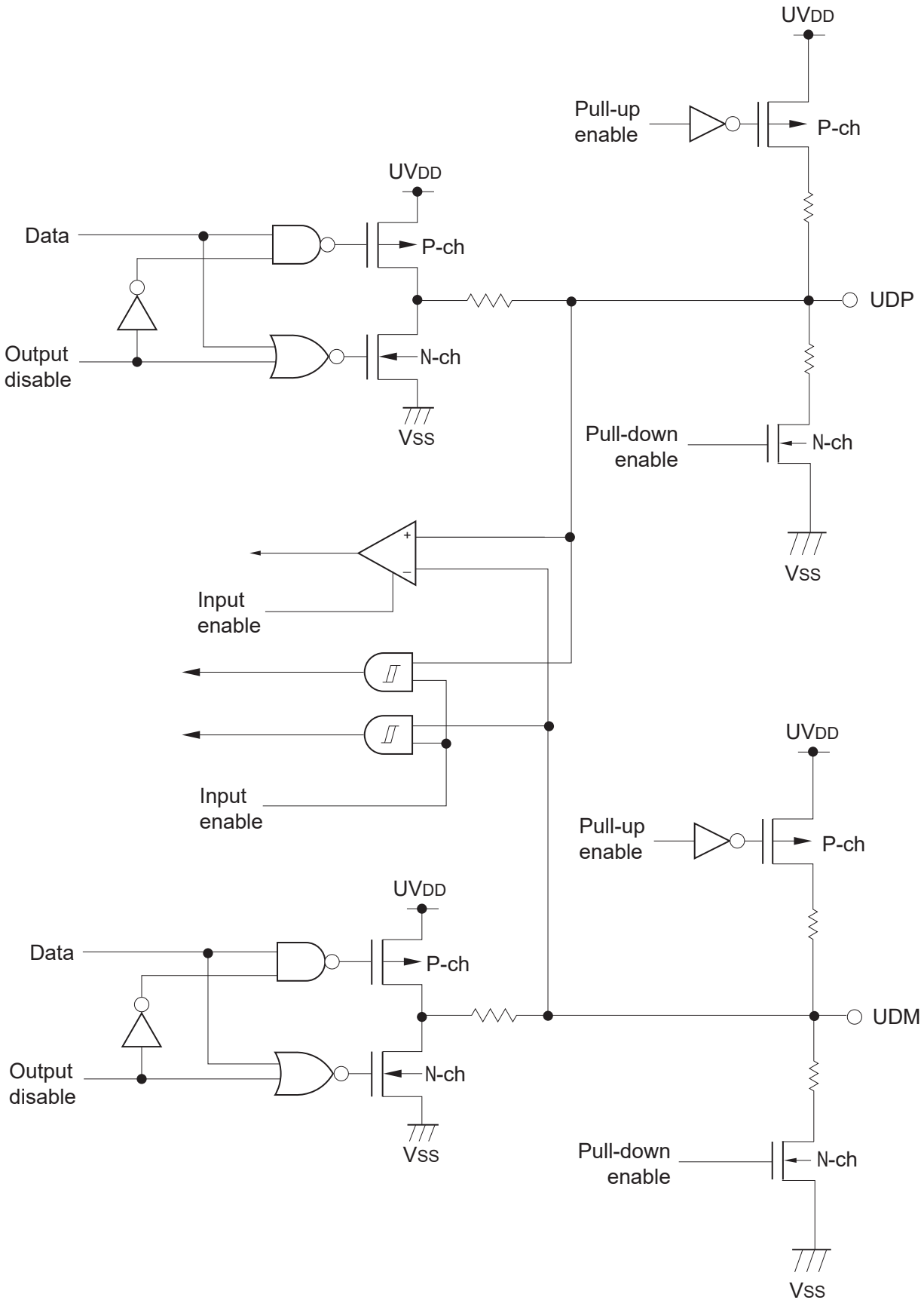


Figure 2-14. Pin Block Diagram for Pin Type 18-11-1

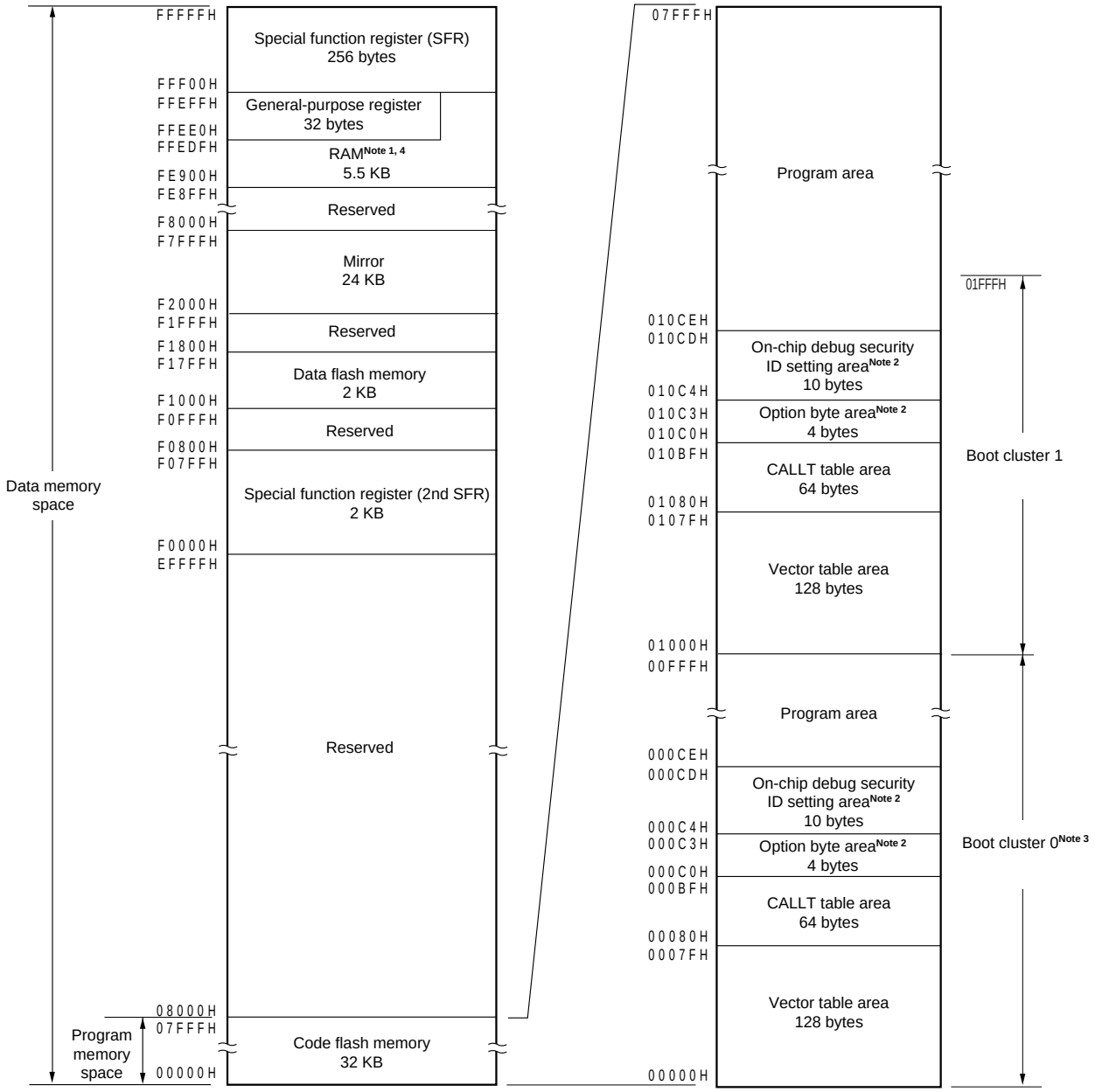


CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the RL78/G1C can access a 1 MB address space. Figure 3-1 shows the memory maps.

Figure 3-1. Memory Map



Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

2. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

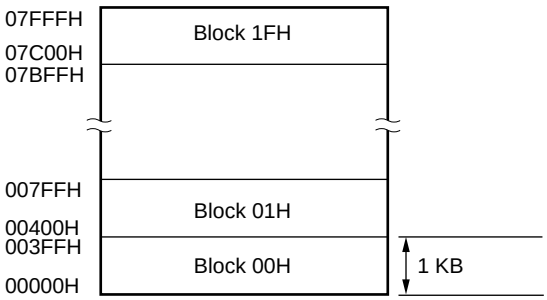
3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **26.6 Security Settings**).

4. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

Caution While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively.

Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 23.3.3 RAM parity error detection function.

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3-1 Correspondence Between Address Values and Block Numbers in Flash Memory**.



Correspondence between the address values and block numbers in the flash memory are shown below.

Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory

Address Value	Block Number
00000H to 003FFH	00H
00400H to 007FFH	01H
00800H to 00BFFH	02H
00C00H to 00FFFH	03H
01000H to 013FFH	04H
01400H to 017FFH	05H
01800H to 01BFFH	06H
01C00H to 01FFFH	07H
02000H to 023FFH	08H
02400H to 027FFH	09H
02800H to 02BFFH	0AH
02C00H to 02FFFH	0BH
03000H to 033FFH	0CH
03400H to 037FFH	0DH
03800H to 03BFFH	0EH
03C00H to 03FFFH	0FH
04000H to 043FFH	10H
04400H to 047FFH	11H
04800H to 04BFFH	12H
04C00H to 04FFFH	13H
05000H to 053FFH	14H
05400H to 057FFH	15H
05800H to 05BFFH	16H
05C00H to 05FFFH	17H
06000H to 063FFH	18H
06400H to 067FFH	19H
06800H to 06BFFH	1AH
06C00H to 06FFFH	1BH
07000H to 073FFH	1CH
07400H to 077FFH	1DH
07800H to 07BFFH	1EH
07C00H to 07FFFH	1FH

3.1.1 Internal program memory space

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

The RL78/G1C products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F10JBxxxx (xxxx = CANA, CAFP, CGNA, CGFP) , R5F10KBxxxx (xxxx = CANA, CAFP, CGNA, CGFP) , R5F10JGxxxx (xxxx = CANA, CAFB, CGNA, CGFB) , R5F10KGxxxx (xxxx = CANA, CAFB, CGNA, CGFB)	Flash memory	32768 × 8 bits (00000H to 07FFFH)

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

(1) Vector table area

The 128-byte area 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 address is a 32 K address of 00000H to 07FFFH, because the vector code is assumed to be 2 bytes and using the area 08000H to 0FFFFH is prohibited.

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

To use the boot swap function, set a vector table also at 01000H to 0107FH.

Table 3-3. Vector Table (1/2)

Vector Table Address	Interrupt Source	48-pin	32-pin
00000H	RESET, POR, LVD, WDT, TRAP, IAW, RAMTOP	√	√
00004H	INTWDTI	√	√
00006H	INTLVI	√	√
00008H	INTP0	√	√
0000AH	INTP1	√	√
0000CH	INTP2	√	√
0000EH	INTP3	√	√
00010H	INTP4	√	√
00012H	INTP5	√	√
0001AH	INTDMA0	√	√
0001CH	INTDMA1	√	√
0001EH	INTST0/INTCSI00/INTIIC00	√	√
00020H	INTTM00	√	√
00022H	INTSR0/INTCSI01/INTIIC01	√	√
00024H	INTSRE0	√	√
	INTTM01H	√	√
0002AH	INTTM03H	√	√
0002CH	INTIICA0	√	√
0002EH	INTTM01	√	√
00030H	INTTM02	√	√
00032H	INTTM03	√	√
00034H	INTAD	√	√
00036H	INTRTC	√	√
00038H	INTIT	√	√
0003AH	INTKR	√	–
0003CH	INTUSB	√	√
0003EH	INTRSUM	√	√

Table 3-3. Vector Table (2/2)

Vector Table Address	Interrupt Source	48-pin	32-pin
0004AH	INTP6	√	–
0004EH	INTP8	√	√
00050H	INTP9	√	√
0005EH	INTMD	√	√
00062H	INTFL	√	√
0007EH	BRK	√	√

(2) CALLT instruction table area

The 64-byte area 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 07FFFH (because an address code is of 2 bytes and using the area 08000H to 0FFFFH is prohibited).

To use the boot swap function, set a CALLT instruction table also at 01080H to 010BFH.

(3) Option byte area

A 4-byte area of 000C0H to 000C3H can be used as an option byte area. Set the option byte at 010C0H to 010C3H when the boot swap is used. For details, see **CHAPTER 25 OPTION BYTE**.

(4) On-chip debug security ID setting area

A 10-byte area of 000C4H to 000CDH and 010C4H to 010CDH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 000C4H to 000CDH when the boot swap is not used and at 000C4H to 000CDH and 010C4H to 010CDH when the boot swap is used. For details, see **CHAPTER 27 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The RL78/G1C mirrors the code flash area of 02000H to 07FFFH, to F2000H to F7FFFH.

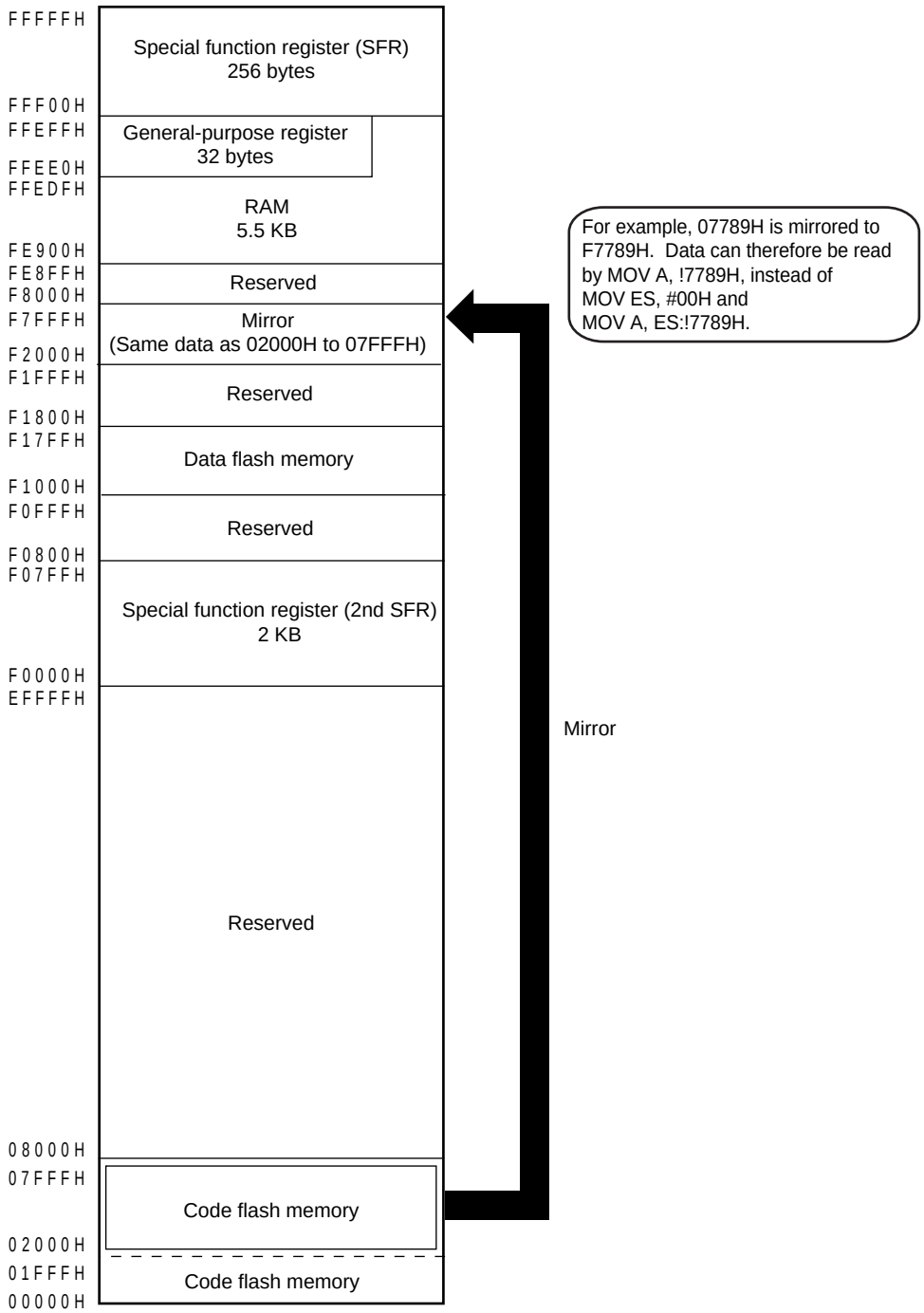
By reading data from F2000H to F7FFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

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 show examples.

Example R5F10JBxxxx (xxxx = CANA, CAFP, CGNA, CGFP) (Flash memory: 32 KB, RAM: 5.5 KB)



The PMC register is described below.

- **Processor mode control register (PMC)**

  This register sets the flash memory space for mirroring to area from F0000H to FFFFFH.

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

  Reset signal generation sets this register to 00H.

Figure 3-2. Format of **Configuration of Processor Mode Control Register (PMC)**

Address: FFFFEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
PMC	0	0	0	0	0	0	0	MAA

MAA	Selection of flash memory space for mirroring to area from F0000H to FFFFFH
0	00000H to 0FFFFH is mirrored to F0000H to FFFFFH
1	Setting prohibited

Caution Be sure to clear bit 0 (MAA) of this register to 0 (initial value).

3.1.3 Internal data memory space

The RL78/G1C products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
R5F10JBxxxx (xxxx = CANA, CAFP, CGNA, CGFP), R5F10KBxxxx (xxxx = CANA, CAFP, CGNA, CGFP), R5F10JGxxxx (xxxx = CANA, CAFB, CGNA, CGFB), R5F10KGxxxx (xxxx = CANA, CAFB, CGNA, CGFB)	5632 × 8bit (FE900H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are fetchod (it is prohibited to use the general-purpose register area for fetching instructions). Four general-purpose register banks consisting of eight 8-bit registers per bank are assigned to the 32-byte area of FFEE0H to FFEFFH of the internal RAM area.

The internal RAM is used as stack memory.

Cautions 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

2. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

3. Use of the RAM areas of the following products is prohibited when performing self-programming and rewriting the data flash memory, because FE900H to FED09H areas are used for each library.

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH (see **Table 3-5** 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 F0000H to F07FFH (see **Table 3-6** 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.

Cautions 1. Do not access addresses to which extended SFRs are not assigned.

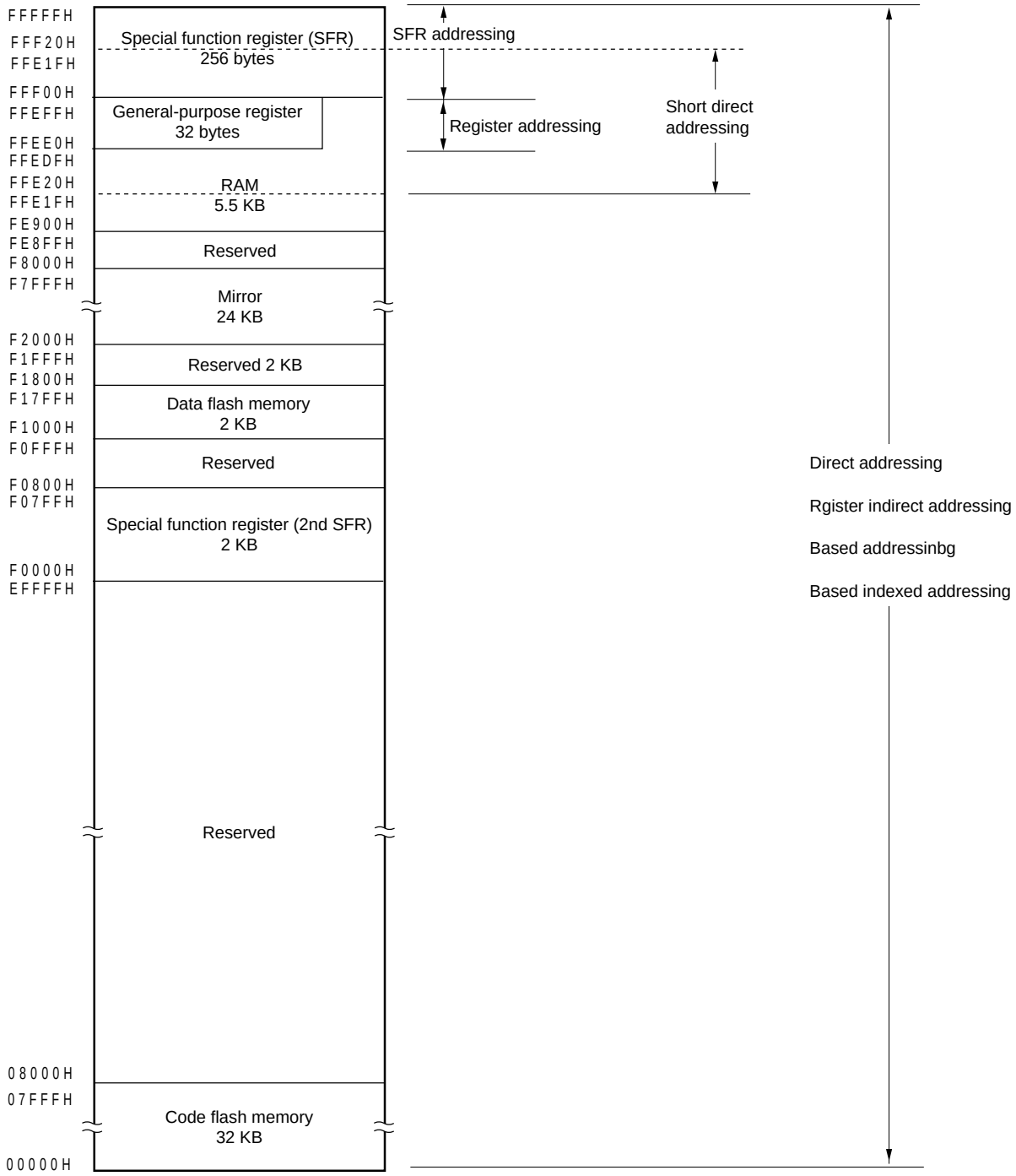
2. When accessing the registers allocated in F0400H to F04FFH, CPU becomes wait mode not executing the next instruction. Therefore, when this wait occurs, the clock counts of the execution time become longer by this wait clock count.

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/G1C, based on operability and other considerations. In particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figure 3-3 shows correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3-3. Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The RL78/G1C 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 00000H and 00001H to the 16 lower-order bits of the program counter. The four higher-order bits of the program counter are cleared to 0000.

Figure 3-4. 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 vectored interrupt request is acknowledged 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 3-5. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	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 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) Register bank select flags (RBS0, RBS1)

These are 2-bit flags to select one of the four register banks.

In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

(d) 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.

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

This flag manages 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 (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **17.3.3**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) 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 3-6. 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 (restore) 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 (FFEE0H to FFEFFH) space for fetching instructions or a stack area.

3. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DMA transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

4. Use of the RAM address area FE900H to FED09H is prohibited when performing self-programming and rewriting the data flash memory, because the area is used for each library.

3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FFEE0H to FFEFFH) of the data memory. The general-purpose registers consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

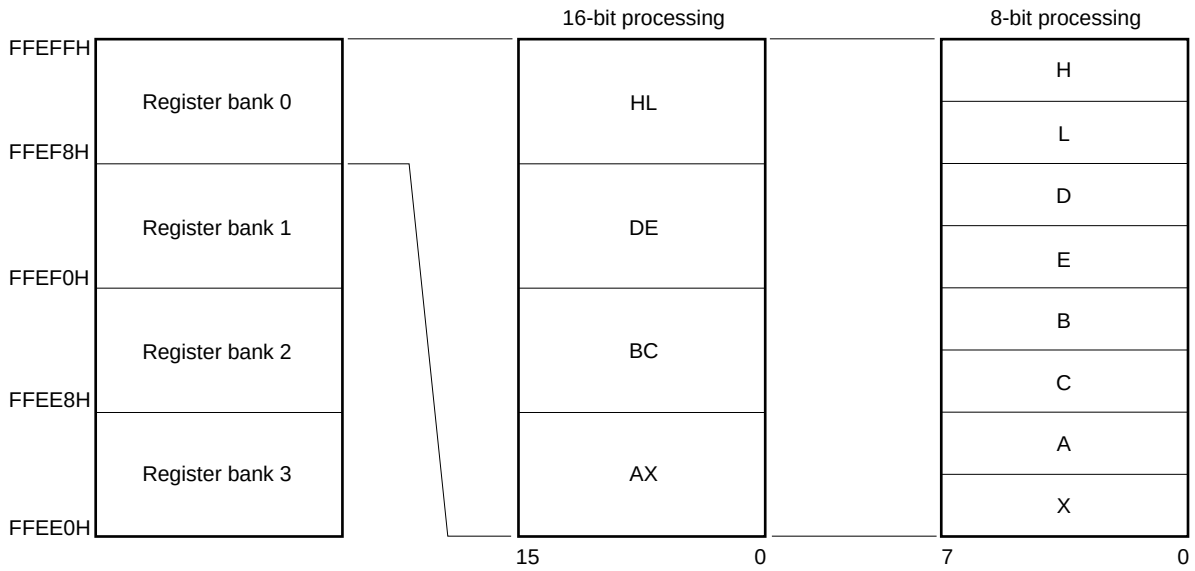
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupt processing for each bank.

Caution It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Figure 3-7. Configuration of General-Purpose Registers

(a) Function name



3.2.3 ES and CS registers

The ES register is used for data access and the CS register is used to specify the higher address when a branch instruction is executed.

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

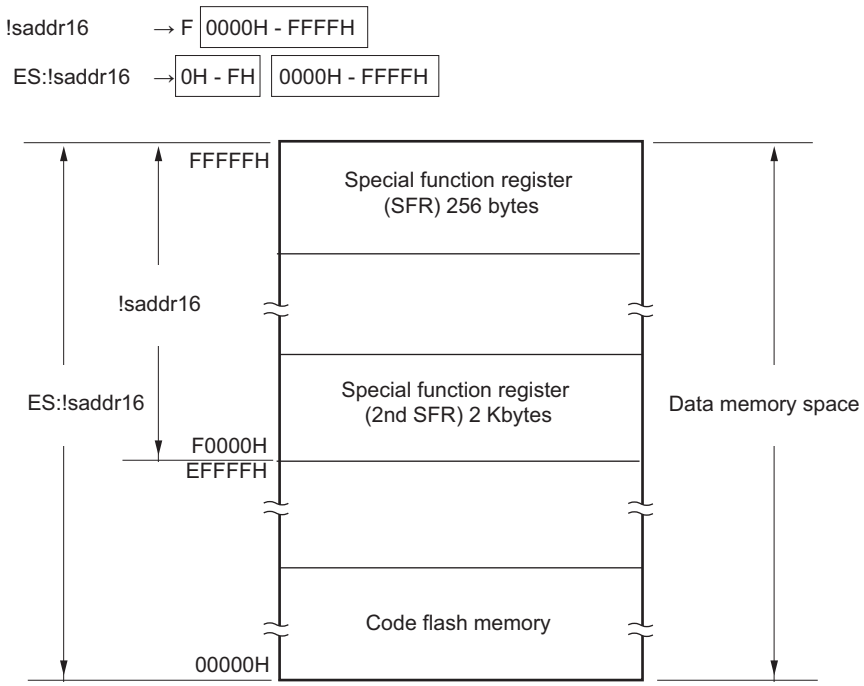
Figure 3-8. Configuration of ES and CS Registers

	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 3-9. 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, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe as follows for the 1-bit manipulation instruction operand (sfr.bit).
  When the bit name is defined: <Bit name>
  When the bit name is not defined: <Register name>, <Bit number> or <Address>, <Bit number>
- 8-bit manipulation
  Describe the symbol defined by the assembler for the 8-bit manipulation instruction operand (sfr). This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol defined by the assembler for the 16-bit manipulation instruction operand (sfrp). When specifying an address, describe an even 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, 8, or 16). "–" 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 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 3-5. SFR List (1/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF00H	Port register 0	P0		R/W	√	√	–	00H
FFF01H	Port register 1	P1		R/W	√	√	–	00H
FFF02H	Port register 2	P2		R/W	√	√	–	00H
FFF03H	Port register 3	P3		R/W	√	√	–	00H
FFF04H	Port register 4	P4		R/W	√	√	–	00H
FFF05H	Port register 5	P5		R/W	√	√	–	00H
FFF06H	Port register 6	P6		R/W	√	√	–	00H
FFF07H	Port register 7	P7		R/W	√	√	–	00H
FFF0CH	Port register 12	P12		R/W	√	√	–	Undefined
FFF0DH	Port register 13	P13		R/W	√	√	–	Undefined
FFF0EH	Port register 14	P14		R/W	√	√	–	00H
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	–	√	√	0000H
FFF11H		–			–	–	√	
FFF12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	–	√	√	0000H
FFF13H		–			–	–	√	
FFF18H	Timer data register 00	TDR00		R/W	–	–	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	–	√	√	00H
FFF1BH		TDR01H			–	√		00H
FFF1EH	10-bit A/D conversion result register	ADCR		R	–	–	√	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	–	√	–	00H
FFF20H	Port mode register 0	PM0		R/W	√	√	–	FFH
FFF21H	Port mode register 1	PM1		R/W	√	√	–	FFH
FFF22H	Port mode register 2	PM2		R/W	√	√	–	FFH
FFF23H	Port mode register 3	PM3		R/W	√	√	–	FFH
FFF24H	Port mode register 4	PM4		R/W	√	√	–	FFH
FFF25H	Port mode register 5	PM5		R/W	√	√	–	FFH
FFF26H	Port mode register 6	PM6		R/W	√	√	–	FFH
FFF27H	Port mode register 7	PM7		R/W	√	√	–	FFH
FFF2CH	Port mode register 12	PM12		R/W	√	√	–	FFH
FFF2EH	Port mode register 14	PM14		R/W	√	√	–	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	√	√	–	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	–	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	√	√	–	00H
FFF37H	Key return mode register	KRM		R/W	√	√	–	00H

Table 3-5. SFR List (2/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	–	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	–	00H
FFF3AH	External interrupt rising edge enable register 1	EGP1		R/W	√	√	–	00H
FFF3BH	External interrupt falling edge enable register 1	EGN1		R/W	√	√	–	00H
FFF50H	IICA shift register 0	IICA0		R/W	–	√	–	00H
FFF51H	IICA status register 0	IICS0		R	√	√	–	00H
FFF52H	IICA flag register 0	IICF0		R/W	√	√	–	00H
FFF58H	D0FIFO port register for DMA transfer	D0FIFOL	D0FIFO	R/W	–	√	√	0000H
FFF59H		–			–	–		
FFF5CH	D1FIFO port register for DMA transfer	D1FIFOL	D1FIFO	R/W	–	√	√	0000H
FFF5DH		–			–	–		
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	–	√	√	00H
FFF67H		TDR03H			–	√		00H
FFF90H	Interval timer control register	ITMC		R/W	–	–	√	0FFFH
FFF91H								
FFF92H	Second count register	SEC		R/W	–	√	–	00H
FFF93H	Minute count register	MIN		R/W	–	√	–	00H
FFF94H	Hour count register	HOUR		R/W	–	√	–	12H[Note]
FFF95H	Week count register	WEEK		R/W	–	√	–	00H
FFF96H	Day count register	DAY		R/W	–	√	–	01H
FFF97H	Month count register	MONTH		R/W	–	√	–	01H
FFF98H	Year count register	YEAR		R/W	–	√	–	00H
FFF99H	Watch error correction register	SUBCUD		R/W	–	√	–	00H
FFF9AH	Alarm minute register	ALARMWM		R/W	–	√	–	00H
FFF9BH	Alarm hour register	ALARMWH		R/W	–	√	–	12H
FFF9CH	Alarm week register	ALARMWW		R/W	–	√	–	00H
FFF9DH	Real-time clock control register 0	RTCC0		R/W	√	√	–	00H
FFF9EH	Real-time clock control register 1	RTCC1		R/W	√	√	–	00H

Note The value of this register is 00H if the AMPM bit (bit 3 of real-time clock control register 0 (RTCC0)) is set to 1 after reset.

Table 3-5. SFR List (3/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFA0H	Clock operation mode control register	CMC		R/W	–	√	–	00H
FFFA1H	Clock operation status control register	CSC		R/W	√	√	–	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	√	√	–	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	–	√	–	07H
FFFA4H	System clock control register	CKC		R/W	√	√	–	00H
FFFA5H	Clock output select register 0	CKS0		R/W	√	√	–	00H
FFFA6H	Clock output select register 1	CKS1		R/W	√	√	–	00H
FFFA8H	Reset control flag register	RESF		R	–	√	–	**Note 1**
FFFA9H	Voltage detection register	LVIM		R/W	√	√	–	00H[Note 1]
FFFAAH	Voltage detection level register	LVIS		R/W	√	√	–	00H/01H/81H[Note 1]
FFFABH	Watchdog timer enable register	WDTE		R/W	–	√	–	1AH/9AH[Note 2]
FFFACH	CRC input register	CRCIN		R/W	–	√	–	00H
FFFB0H	DMA SFR address register 0	DSA0		R/W	–	√	–	00H
FFFB1H	DMA SFR address register 1	DSA1		R/W	–	√	–	00H
FFFB2H	DMA RAM address register 0	DRA0L	DRA0	R/W	–	√	√	00H
FFFB3H		DRA0H		R/W	–	√		00H
FFFB4H	DMA RAM address register 1	DRA1L	DRA1	R/W	–	√	√	00H
FFFB5H		DRA1H		R/W	–	√		00H
FFFB6H	DMA byte count register 0	DBC0L	DBC0	R/W	–	√	√	00H
FFFB7H		DBC0H		R/W	–	√		00H
FFFB8H	DMA byte count register 1	DBC1L	DBC1	R/W	–	√	√	00H
FFFB9H		DBC1H		R/W	–	√		00H
FFFBAH	DMA mode control register 0	DMC0		R/W	√	√	–	00H
FFFBBH	DMA mode control register 1	DMC1		R/W	√	√	–	00H
FFFBCH	DMA operation control register 0	DRC0		R/W	√	√	–	00H
FFFBDH	DMA operation control register 1	DRC1		R/W	√	√	–	00H

Notes 1. The reset values of the registers vary depending on the reset source as shown below.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM parity error	Reset by illegal-memory access	Reset by LVD
RESF	TRAP bit	Cleared (0)		Set (1)	Held			Held
	WDTRF bit			Held	Set (1)	Held		
	RPERF bit			Held		Set (1)	Held	
	IAWRF bit			Held			Set (1)	
	LVIRF bit			Held				Set (1)
LVIM	LVISEN bit	Cleared (0)						Held
	LVIOMSK bit	Held						
	LVIF bit							
LVIS		Cleared (00H/01H/81H)						

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

Table 3-5. SFR List (4/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	Interrupt request flag register 2H	IF2H		R/W	√	√		00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	Interrupt mask flag register 2H	MK2H		R/W	√	√		FFH
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	Priority specification flag register 02H	PR02H		R/W	√	√		FFH
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH	Priority specification flag register 12H	PR12H		R/W	√	√		FFH
FFFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	Interrupt request flag register 1H	IF1H		R/W	√	√		00H
FFFE4H	Interrupt mask flag register 0L	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H	Interrupt mask flag register 0H	MK0H		R/W	√	√		FFH
FFFE6H	Interrupt mask flag register 1L	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H	Interrupt mask flag register 1H	MK1H		R/W	√	√		FFH
FFFE8H	Priority specification flag register 00L	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H	Priority specification flag register 00H	PR00H		R/W	√	√		FFH
FFFEAH	Priority specification flag register 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFEBH	Priority specification flag register 01H	PR01H		R/W	√	√		FFH
FFFECH	Priority specification flag register 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFEDH	Priority specification flag register 10H	PR10H		R/W	√	√		FFH
FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH
FFFF0H FFFF1H	Multiplication/division data register A (L)	MDAL		R/W	–	–	√	0000H
FFFF2H FFFF3H	Multiplication/division data register A (H)	MDAH		R/W	–	–	√	0000H
FFFF4H FFFF5H	Multiplication/division data register B (H)	MDBH		R/W	–	–	√	0000H
FFFF6H FFFF7H	Multiplication/division data register B (L)	MDBL		R/W	–	–	√	0000H
FFFFEH	Processor mode control register	PMC		R/W	√	√	–	00H

Remark For extended SFRs (2nd SFRs), see **Table 3-6 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, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe as follows for the 1-bit manipulation instruction operand (!addr16.bit)
  When the bit name is defined: <Bit name>
  When the bit name is not defined: <Register name>, <Bit number> or <Address>, <Bit number>
- 8-bit manipulation
  Describe the symbol defined by the assembler for the 8-bit manipulation instruction operand (!addr16). This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol defined by the assembler for the 16-bit manipulation instruction operand (!addr16). When specifying an address, describe an even address.

Table 3-6 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, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

Cautions 1. Do not access addresses to which extended SFRs (2nd SFRs) are not assigned.

2. When accessing the registers allocated in F0400H to F04FFH, CPU becomes wait mode not executing the next instruction. Therefore, when this wait occurs, the clock counts of the execution time become longer by this wait clock count.

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

Table 3-6. Extended SFR (2nd SFR) List (1/8)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0010H	A/D converter mode register 2	ADM2	R/W	√	√	–	00H
F0011H	Conversion result comparison upper limit setting register	ADUL	R/W	–	√	–	FFH
F0012H	Conversion result comparison lower limit setting register	ADLL	R/W	–	√	–	00H
F0013H	A/D test register	ADTES	R/W	–	√	–	00H
F0018H	Port mode selection register	PMS	R/W	√	√	–	00H
F001AH	Peripheral I/O redirection register	PIOR	R/W	–	√	–	00H
F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	–	01H
F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H
F0037H	Pull-up resistor option register 7	PU7	R/W	√	√	–	00H
F003CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H
F003EH	Pull-up resistor option register 14	PU14	R/W	√	√	–	00H
F0040H	Port input mode register 0	PIM0	R/W	√	√	–	00H
F0043H	Port input mode register 3	PIM3	R/W	√	√	–	00H
F0045H	Port input mode register 5	PIM5	R/W	√	√	–	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	–	00H
F0053H	Port output mode register 3	POM3	R/W	√	√	–	00H
F0055H	Port output mode register 5	POM5	R/W	√	√	–	00H
F0057H	Port output mode register 7	POM7	R/W	√	√	–	00H
F0060H	Port mode control register 0	PMC0	R/W	√	√	–	FFH
F006CH	Port mode control register 12	PMC12	R/W	√	√	–	FFH
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	–	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	–	00H
F0074H	Timer input select register 0	TIS0	R/W	–	√	–	00H
F0076H	A/D port configuration register	ADPC	R/W	–	√	–	00H
F0077H	Invalid memory access detection control register	IAWCTL	R/W	–	√	–	00H
F0090H	Data flash control register	DFLCTL	R/W	√	√	–	00H

Table 3-6. Extended SFR (2nd SFR) List (2/8)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F00A0H	High-speed on-chip oscillator trimming register	HIOTRM		R/W	–	√	–	Undefined Note 1
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV		R/W	–	√	–	Undefined Note 2
F00E0H	Multiplication/division data register C (L)	MDCL		R/W	–	–	√	0000H
F00E2H	Multiplication/division data register C (H)	MDCH		R/W	–	–	√	0000H
F00E8H	Multiplication/division control register	MDUC		R/W	√	√	–	00H
F00F0H	Peripheral enable register 0	PER0		R/W	√	√	–	00H
F00F3H	Operation speed mode control register	OSMC		R/W	–	√	–	00H
F00F5H	RAM parity error control register	RPECTL		R/W	√	√	–	00H
F00FEH	BCD adjust result register	BCDADJ		R	–	√	–	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	–	√	√	0000H
F0101H		–			–	–		
F0102H	Serial status register 01	SSR01L	SSR01	R	–	√	√	0000H
F0103H		–			–	–		
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	–	√	√	0000H
F0109H		–			–	–		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	–	√	√	0000H
F010BH		–			–	–		
F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F0113H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		–			–	–		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		–			–	–		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		–			–	–		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	–	√	√	0000H
F0127H		–			–	–		

Note 1. The value after a reset is adjusted at the time of shipment.

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

Table 3-6. Extended SFR (2nd SFR) List (3/8)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0128H	Serial output register 0	SO0		R/W	–	–	√	0303H
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		–			–	–		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	–	√	√	0000H
F0135H		–			–	–		
F0138H	Serial standby control register 0	SSC0L	SSC0	R/W	–	√	√	0000H
		–			–	–		
F0180H	Timer counter register 00	TCR00		R	–	–	√	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	–	–	√	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	–	–	√	FFFFH
F0185H								
F0186H	Timer counter register 03	TCR03		R	–	–	√	FFFFH
F0187H								
F0190H	Timer mode register 00	TMR00		R/W	–	–	√	0000H
F0191H								
F0192H	Timer mode register 01	TMR01		R/W	–	–	√	0000H
F0193H								
F0194H	Timer mode register 02	TMR02		R/W	–	–	√	0000H
F0195H								
F0196H	Timer mode register 03	TMR03		R/W	–	–	√	0000H
F0197H								
F01A0H	Timer status register 00	TSR00L	TSR00	R	–	√	√	0000H
F01A1H		–			–	–		
F01A2H	Timer status register 01	TSR01L	TSR01	R	–	√	√	0000H
F01A3H		–			–	–		
F01A4H	Timer status register 02	TSR02L	TSR02	R	–	√	√	0000H
F01A5H		–			–	–		
F01A6H	Timer status register 03	TSR03L	TSR03	R	–	√	√	0000H
F01A7H		–			–	–		
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		–			–	–		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		–			–	–		
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		–			–	–		

Table 3-6. Extended SFR (2nd SFR) List (4/8)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01B6H	Timer clock select register 0	TPS0		R/W	–	–	√	0000H
F01B7H								
F01B8H	Timer output register 0	TO0L	TO0	R/W	–	√	√	0000H
F01B9H		–			–	–		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		–			–	–		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	–	√	√	0000H
F01BDH		–			–	–		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	–	√	√	0000H
F01BFH		–			–	–		
F0230H	IICA control register 00	IICCTL00		R/W	√	√	–	00H
F0231H	IICA control register 01	IICCTL01		R/W	√	√	–	00H
F0232H	IICA low-level width setting register 0	IICWL0		R/W	–	√	–	FFH
F0233H	IICA high-level width setting register 0	IICWH0		R/W	–	√	–	FFH
F0234H	Slave address register 0	SVA0		R/W	–	√	–	00H
F02E5H	PLL control register	DSCCTL		R/W	√	√	–	00H
F02E6H	Main clock control regiter	MCKC		R/W	√	√	–	00H
F02F0H	Flash memory CRC control register	CRC0CTL		R/W	√	√	–	00H
F02F2H	Flash memory CRC operation result register	PGCRCL		R/W	–	–	√	0000H
F02F3H								
F02FAH	CRC data register	CRCD		R/W	–	–	√	0000H
F02FBH								
F0400H	System configuration control register	SYSCFG		R/W	–	–	√	0000H
F0401H								
F0402H	System configuration control register 1	SYSCFG1		R/W	–	–	√	0000H
F0403H								
F0404H	System configuration status register 0	SYSSTS0		R	–	–	√	XX000000
F0405H								00000000B

Table 3-6. Extended SFR (2nd SFR) List (5/8)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0406H	System configuration statusl register 1	SYSSTS1		R	–	–	√	X0000000
F0407H								00000000B
F0408H	Device state control register 0	DVSTCTR0		R/W	–	–	√	0000H
F0409H								
F040AH	Device state control register 1	DVSTCTR1		R/W	–	–	√	0000H
F040BH								
F0410H	DMA0-FIFO pin configuration register	DMA0PCFG		R/W	–	–	√	0000H
F0411H								
F0412H	DMA1-FIFO pin configuration register	DMA1PCFG		R/W	–	–	√	0000H
F0413H								
F0414H	CFIFO port register	CFIFOML	CFIFOM	R/W	–	√	√	00H
F0415H		–			–	–		00H
F0418H	D0FIFO port register	D0FIFOML	D0FIFOM	R/W	–	√	√	00H
F0419H		–			–	–		00H
F041CH	D1FIFO port register	D1FIFOML	D1FIFOM	R/W	–	√	√	00H
F041DH		–			–	–		00H
F0420H	CFIFO port selection register	CFIFOSEL		R/W	–	–	√	0000H
F0421H								
F0422H	CFIFO port control register	CFIFOCTR		R/W	–	–	√	0000H
F0423H								
F0428H	D0FIFO port selection register	D0FIFOSEL		R/W	–	–	√	0000H
F0429H								
F042AH	D0FIFO port control register	D0FIFOCTR		R/W	–	–	√	0000H
F042BH								
F042CH	D1FIFO port selection register	D1FIFOSEL		R/W	–	–	√	0000H
F042DH								
F042EH	D1FIFO port control register	D1FIFOCTR		R/W	–	–	√	0000H
F042FH								

Table 3-6. Extended SFR (2nd SFR) List (6/8)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0430H	Interrupt enable register 0	INTENB0	R/W	–	–	√	0000H
F0431H							
F0432H	Interrupt enable register 1	INTENB1	R/W	–	–	√	0000H
F0433H							
F0434H	Interrupt enable register 2	INTENB2	R/W	–	–	√	0000H
F0435H							
F0436H	BRDY interrupt enable register	BRDYENB	R/W	–	–	√	0000H
F0437H							
F0438H	NRDY interrupt enable register	NRDYENB	R/W	–	–	√	0000H
F0439H							
F043AH	BEMP interrupt register	BEMPENB	R/W	–	–	√	0000H
F043BH							
F043CH	SOF output configuration register	SOFCFG	R/W	–	–	√	0000H
F043DH							
F0440H	Interrupt status register 0	INTSTS0	R/W	–	–	√	00000000
F0441H							X0000000B
F0442H	Interrupt status register 1	INTSTS1	R/W	–	–	√	XX0X0000
F0443H							00000000B
F0444H	Interrupt status register 2	INTSTS2	R/W	–	–	√	X00X0000
F0445H							00000000B
F0446H	BRDY interrupt status register	BRDYSTS	R/W	–	–	√	0000H
F0447H							
F0448H	NRDY Interrupt status register	NRDYSTS	R/W	–	–	√	0000H
F0449H							

Table 3-6. Extended SFR (2nd SFR) List (7/8)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F044AH F044BH	BEMP interrupt status register	BEMPSTS	R/W	–	–	√	0000H
F044CH F044DH	Frame number register	FRMNUM	R/W	–	–	√	0000H
F0450H F0451H	USB address register	USBADDR	R	–	–	√	0000H
F0454H F0455H	USB request type register	USBREQ	**Note**	–	–	√	0000H
F0456H F0457H	USB request value register	USBVAL	**Note**	–	–	√	0000H
F0458H F0459H	USB request index register	USBINDX	**Note**	–	–	√	0000H
F045AH F045BH	USB request length register	USBLENG	**Note**	–	–	√	0000H
F045CH F045DH	DCP configuration register	DCPCFG	R/W	–	–	√	0000H
F045EH F045FH	DCP maximum packet size register	DCPMAXP	R/W	–	–	√	0040H
F0460H F0461H	DCP control register	DCPCTR	R/W	–	–	√	0000H
F0464H F0465H	Pipe window selection register	PIPESEL	R/W	–	–	√	0000H
F0468H F0469H	Pipe configuration register	PIPECFG	R/W	–	–	√	0000H
F046CH F046DH	Pipe maximum packet size register	PIPEMAXP	R/W	–	–	√	0000H/ 0040H
F046EH F046FH	Pipe interval control register	PIPEPERI	R/W	–	–	√	0000H
F0476H F0477H	Pipe 4 control register	PIPE4CTR	R/W	–	–	√	0000H
F0478H F0479H	Pipe 5 control register	PIPE5CTR	R/W	–	–	√	0000H
F047AH F047BH	Pipe 6 control register	PIPE6CTR	R/W	–	–	√	0000H
F047CH F047DH	Pipe 7 control register	PIPE7CTR	R/W	–	–	√	0000H

Note When the function controller is selected, the register can only be read. Note that writing the register is prohibited. On the other hand, when the host controller is selected, the register can be read and write.

Table 3-6. Extended SFR (2nd SFR) List (8/8)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F049CH F049DH	Pipe 4 transaction counter enable register	PIPE4TRE	R/W	–	–	√	0000H
F049EH F049FH	Pipe 4 transaction counter register	PIPE4TRN	R/W	–	–	√	0000H
F04A0H F04A1H	Pipe 5 transaction counter enable register	PIPE5TRE	R/W	–	–	√	0000H
F04A2H F04A3H	Pipe 5 transaction counter register	PIPE5TRN	R/W	–	–	√	0000H
F04B0H F04B1H	BC control register 0	USBBCCTRL0	R/W	–	–	√	0000H
F04B4H F04B5H	BC control register 1	USBBCCTRL1	R/W	–	–	√	0000H
F04B8H F04B9H	BC option control register 0	USBBCOPT0	R/W	–	–	√	0000H
F04BCH F04BDH	BC option control register 1	USBBCOPT1	R/W	–	–	√	0000H
F04CCH F04CDH	USB module control register	USBMC	R/W	–	–	√	0002H
F04D0H F04D1H	Device address 0 configuration register	DEVADD0	R/W	–	–	√	0000H
F04D2H F04D3H	Device address 1 configuration register	DEVADD1	R/W	–	–	√	0000H
F04D4H F04D5H	Device address 2 configuration register	DEVADD2	R/W	–	–	√	0000H
F04D6H F04D7H	Device address 3 configuration register	DEVADD3	R/W	–	–	√	0000H
F04D8H F04D9H	Device address 4 configuration register	DEVADD4	R/W	–	–	√	0000H
F04DAH F04DBH	Device address 5 configuration register	DEVADD5	R/W	–	–	√	0000H

Remark For the SFRs in SFR area, see **Table 3-5. SFR List**.

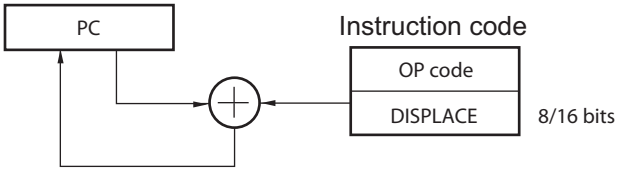
3.3 Instruction Address Addressing

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 3-10. 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 3-11. Example of CALL !!addr20/BR !!addr20

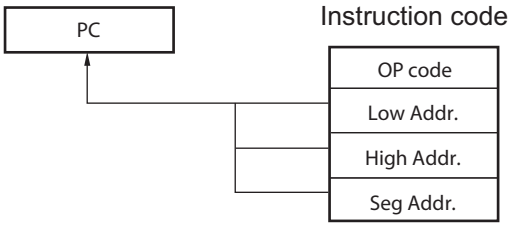
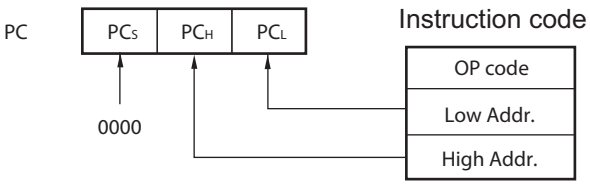


Figure 3-12. Example of CALL !addr16/BR !addr16



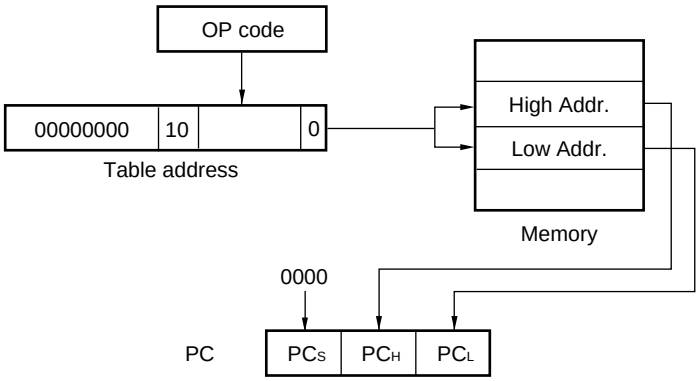
3.3.3 Table indirect addressing

[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 3-13. Outline of Table Indirect Addressing

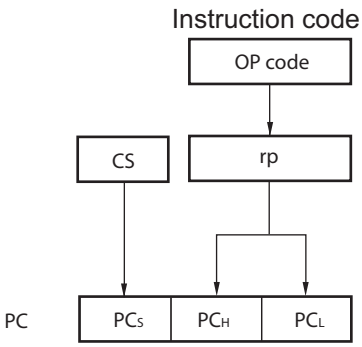


3.3.4 Register direct addressing

[Function]

Register direct 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 direct addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

Figure 3-14. Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

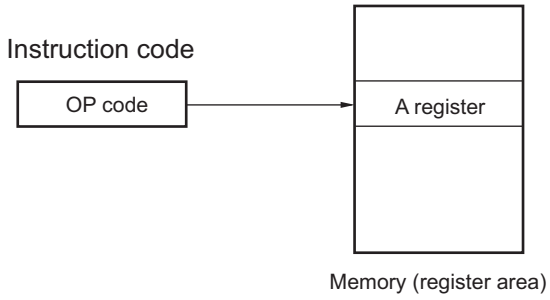
[Function]

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 3-15. Outline of Implied Addressing



3.4.2 Register addressing

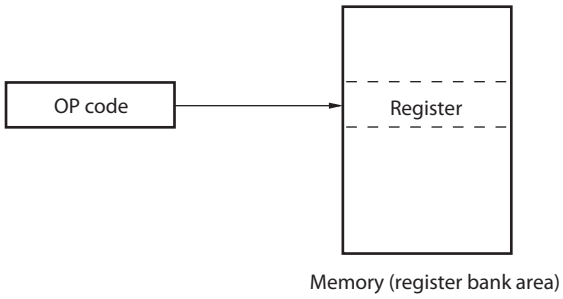
[Function]

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 3-16. Outline of Register Addressing



3.4.3 Direct addressing

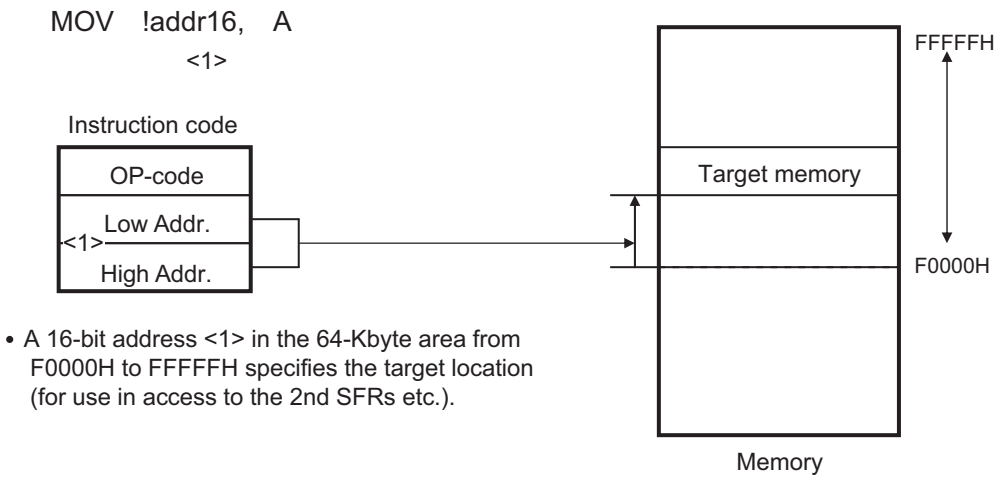
[Function]

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

[Operand format]

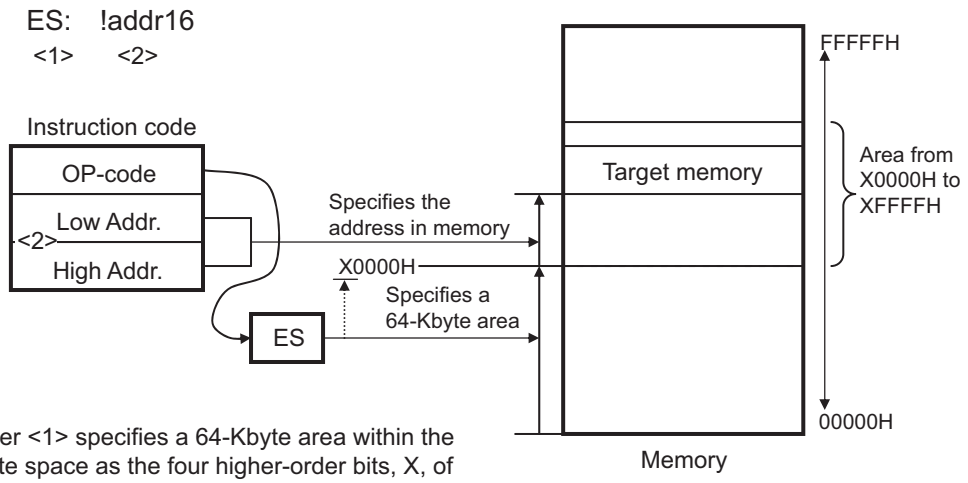
Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3-17. 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 3-18. 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

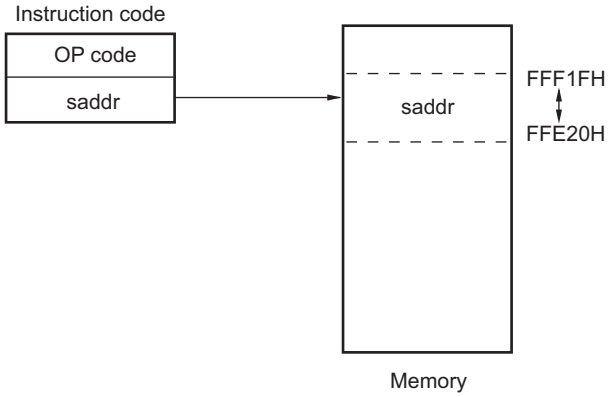
[Function]

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.

[Operand format]

Identifier	Description
SADDR	Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data (only the space from FFE20H to FFF1FH is specifiable)
SADDRP	Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data (even address only) (only the space from FFE20H to FFF1FH is specifiable)

Figure 3-19. 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

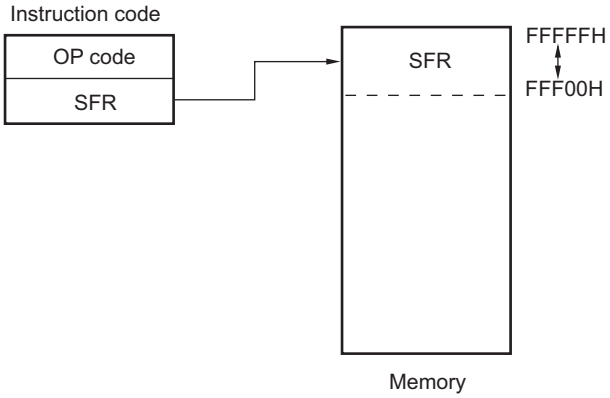
[Function]

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 3-20. Outline of SFR Addressing



3.4.6 Register indirect addressing

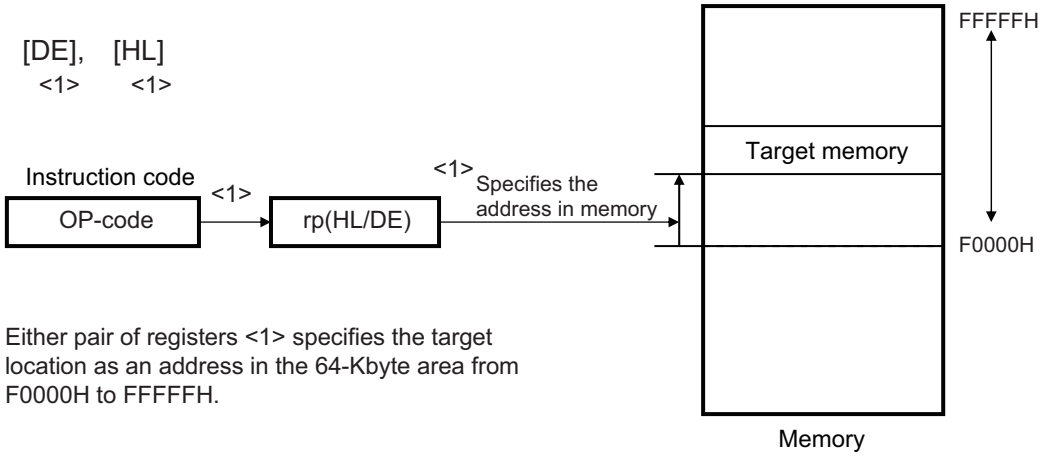
[Function]

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]

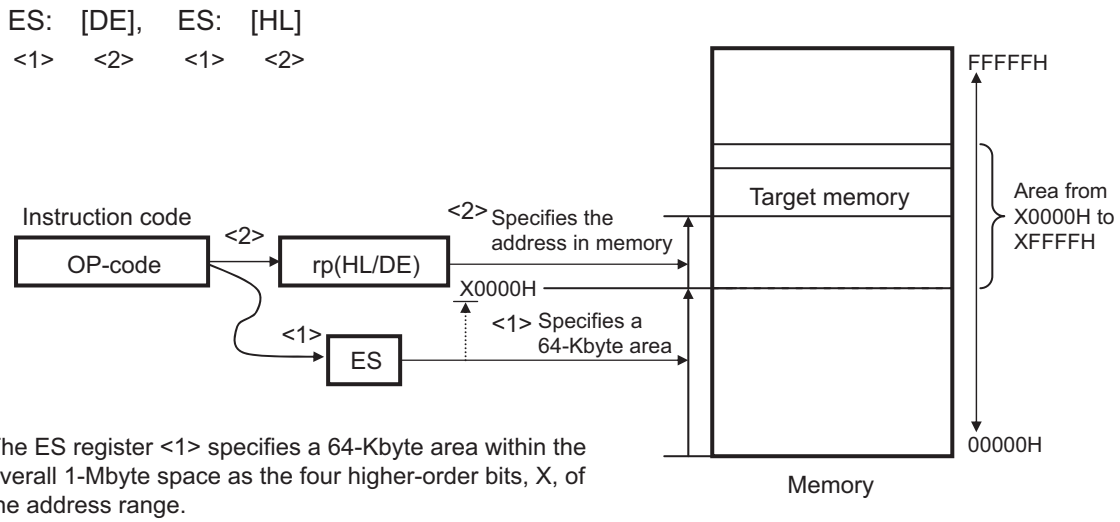
Identifier	Description
–	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

Figure 3-21. 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 3-22. 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

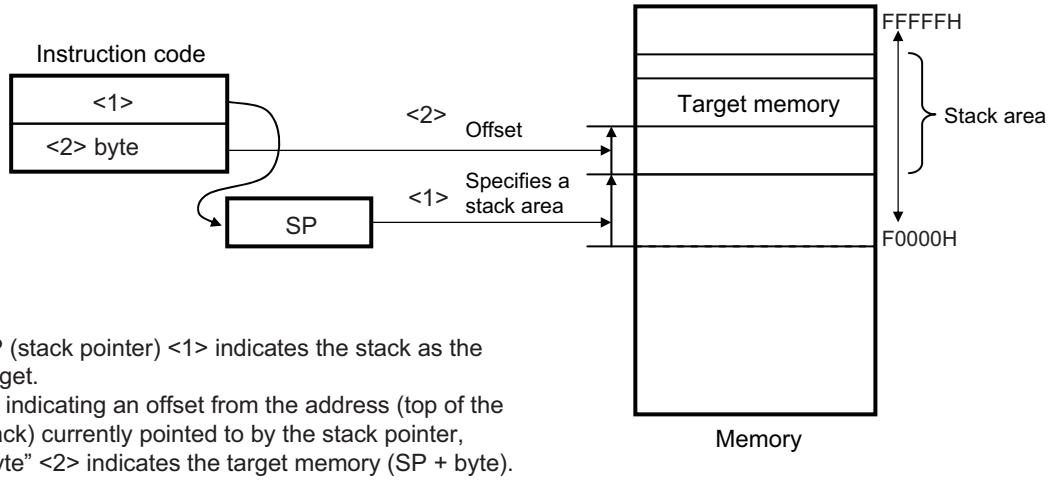
[Function]

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]

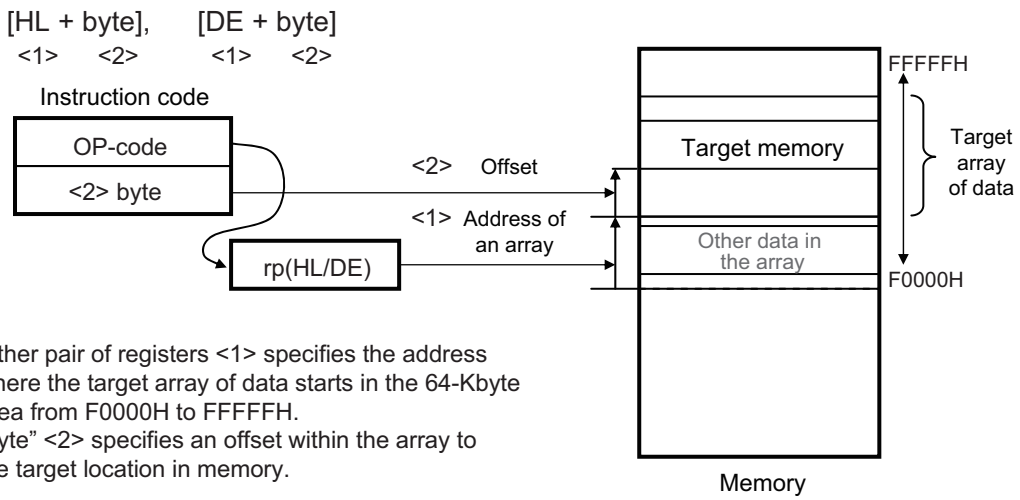
Identifier	Description
–	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
–	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
–	word[BC] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL + byte], ES:[DE + byte] (higher 4-bit 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 3-23. Example of [SP+byte]



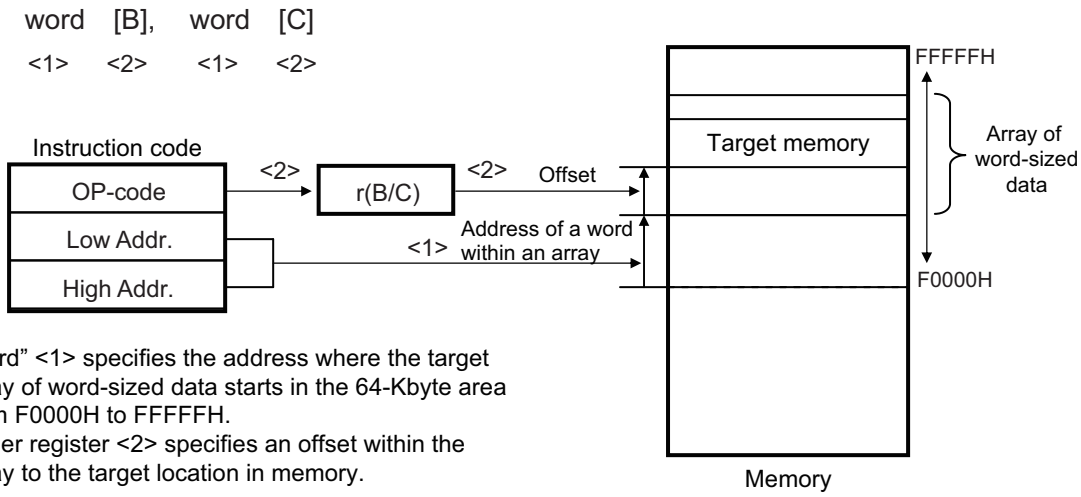
- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, “byte” <2> indicates the target memory (SP + byte).

Figure 3-24. Example of [HL + byte], [DE + byte]



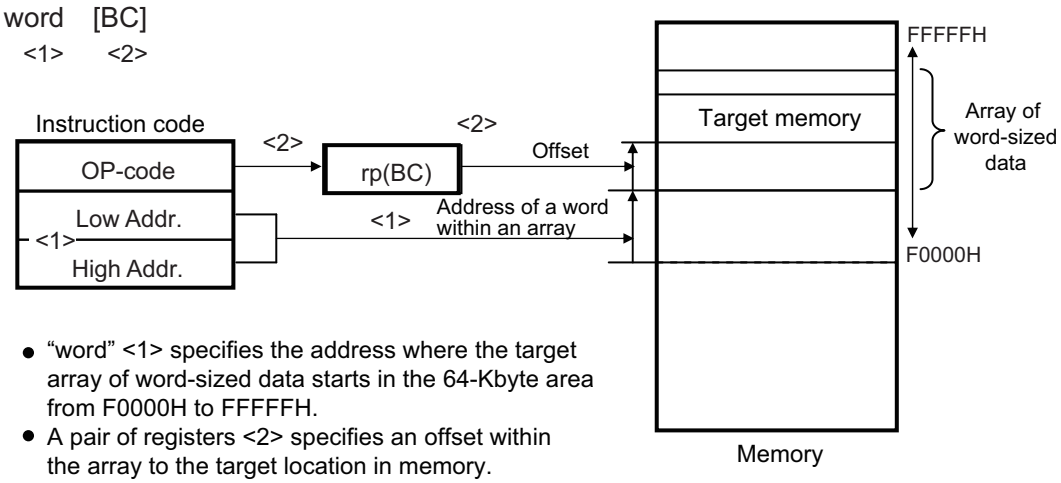
- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- "byte" <2> specifies an offset within the array to the target location in memory.

Figure 3-25. Example of word[B], word[C]



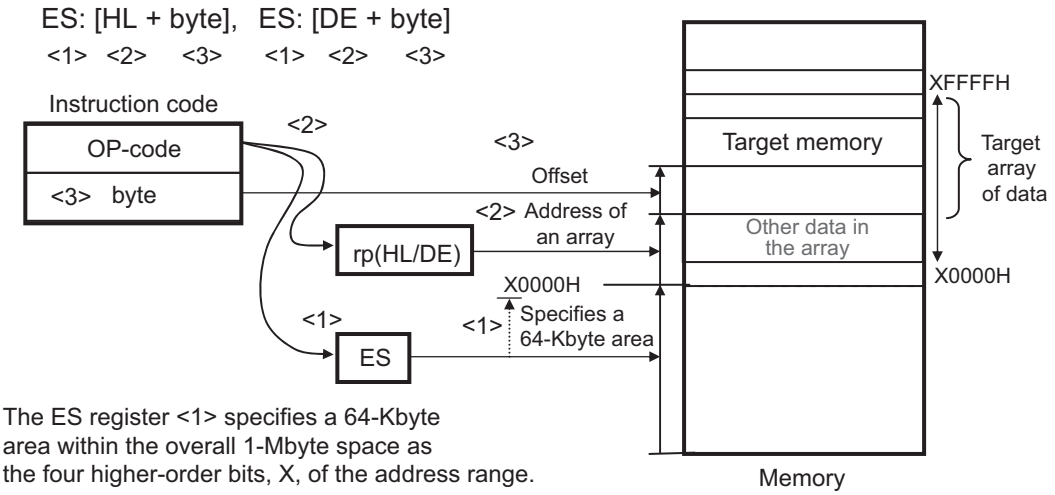
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3-26. Example of word[BC]



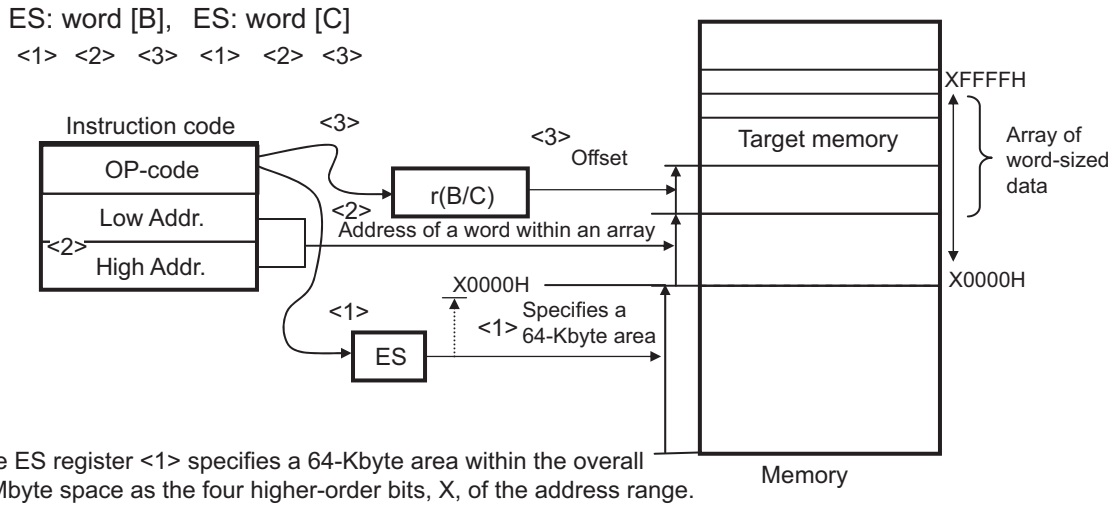
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

Figure 3-27. Example of ES:[HL + byte], ES:[DE + byte]



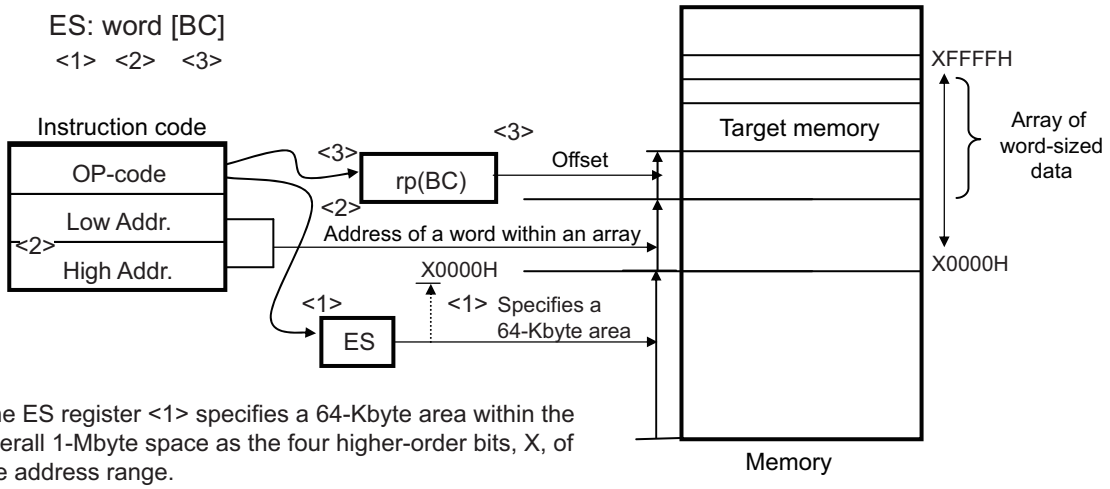
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- “byte” <3> specifies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

Figure 3-28. 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-sizeddata starts in the 64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array tothe target location in memory.

Figure 3-29. 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

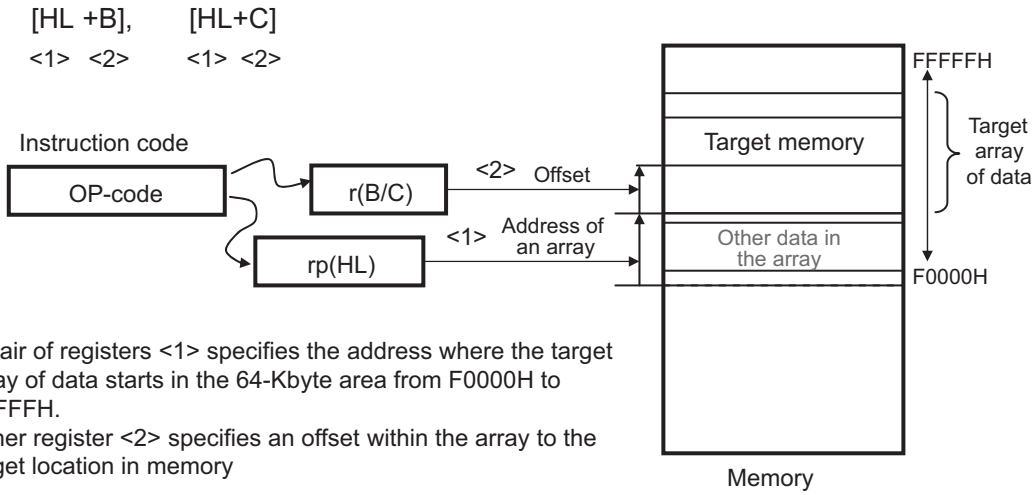
[Function]

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]

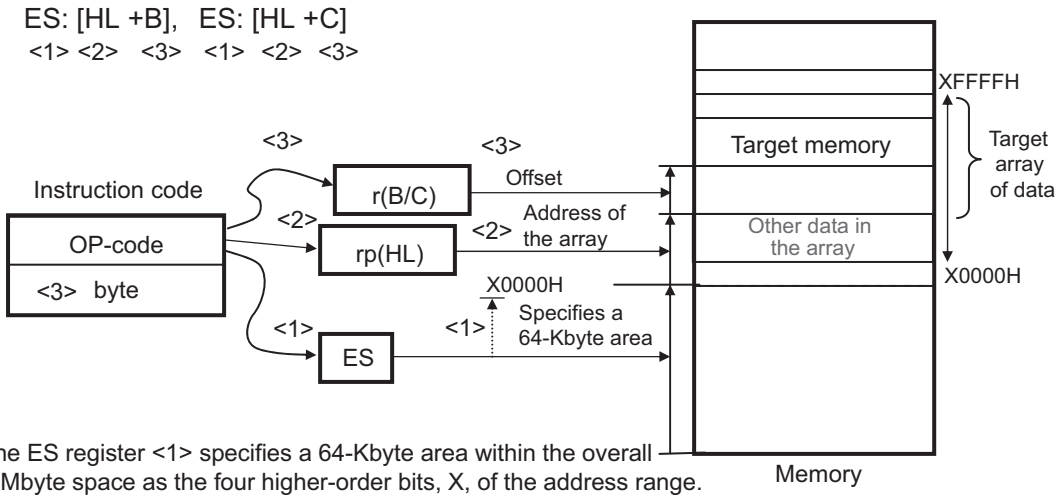
Identifier	Description
–	[HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable)
–	ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register)

Figure 3-30. 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 3-31. 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.

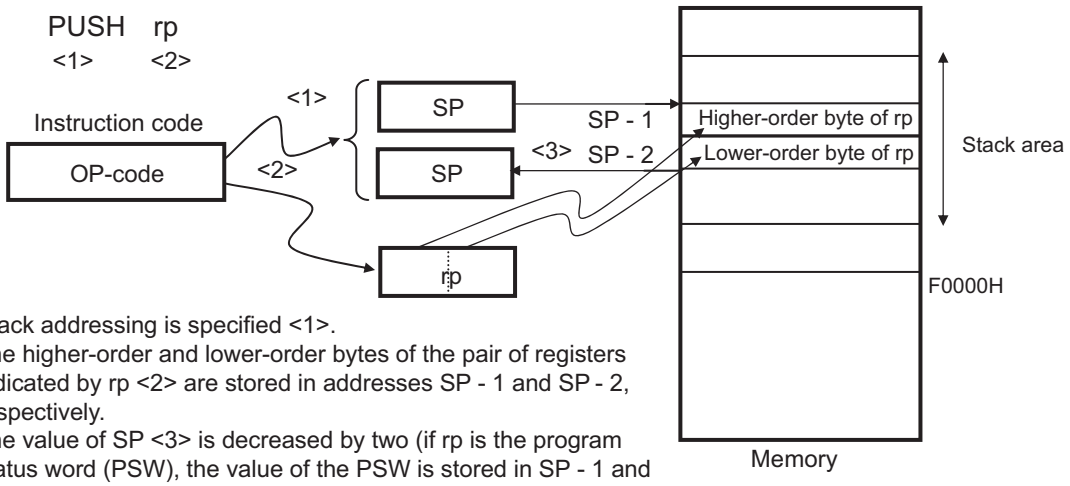
Only the internal RAM area can be set as the stack area.

[Description format]

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

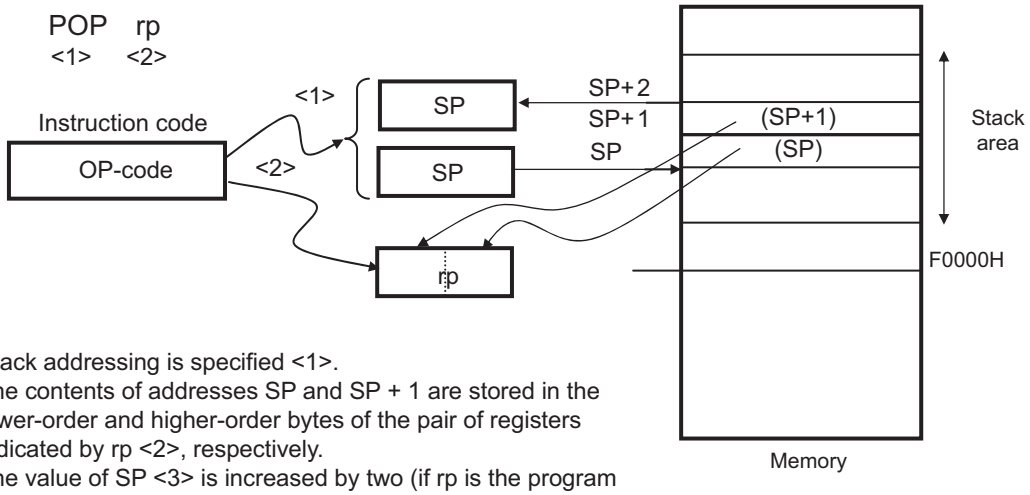
The data that is saved/restored by each stack operation as Figure 3-32 to Figure 3-37.

Figure 3-32. Example of PUSH rp



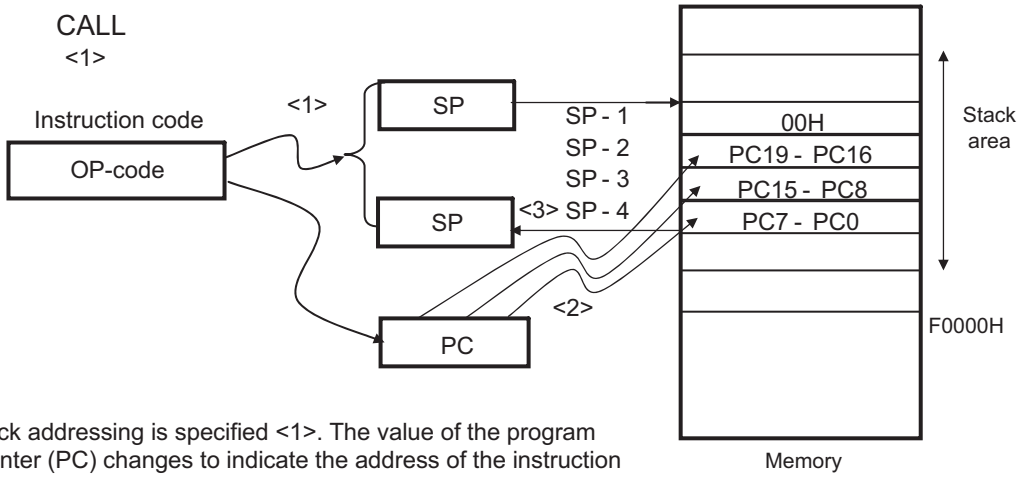
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP - 1 and SP - 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP -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

Figure 3-33. Example of POP



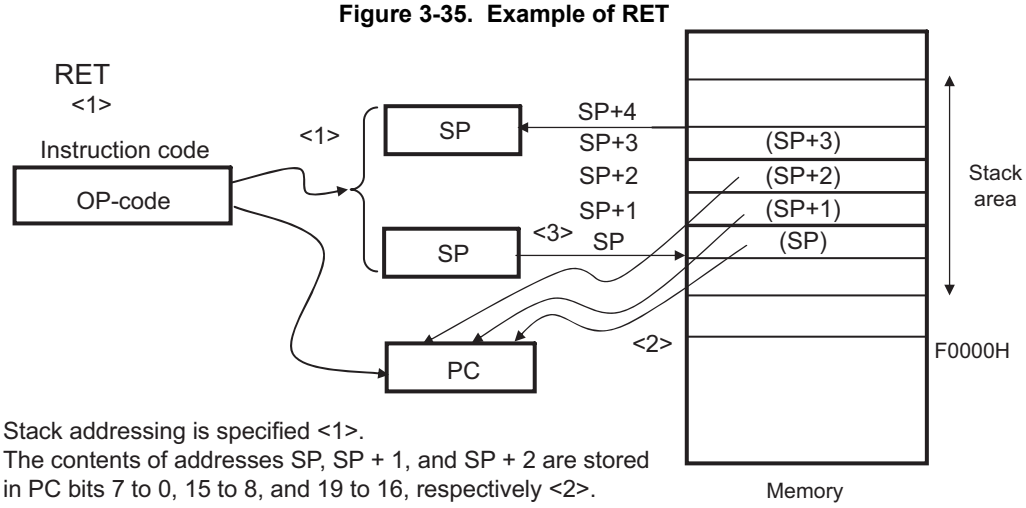
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3-34. Example of CALL, CALLT



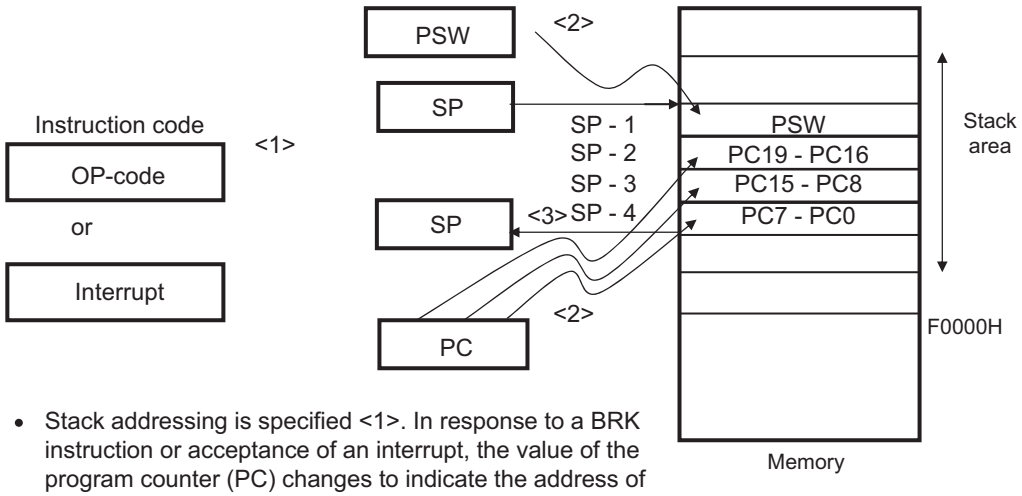
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 1, SP - 2, SP - 3, and SP - 4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3-35. Example of RET



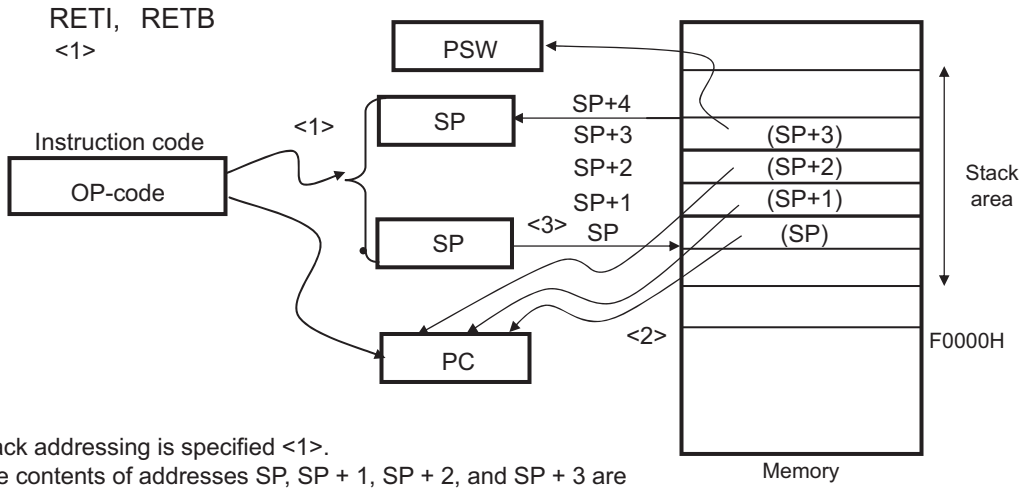
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3-36. 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 3-37. 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/G1C 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 (PM0 to PM7, PM12, PM14) Port registers (P0 to P7, P12 to P14) Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU7, PU12, PU14) Port input mode registers (PIM0, PIM3, PIM5) Port output mode registers (POM0, POM3, POM5, POM7) Port mode control registers (PMC0, PMC12) A/D port configuration register (ADPC) Peripheral I/O redirection register (PIOR)
Port	• 32-pin products : Total: 26 (CMOS I/O: 20, CMOS input: 3, N-ch open drain I/O: 3) • 32-pin products : Total: 24 (CMOS I/O: 18, CMOS input: 3, N-ch open drain I/O: 3) • 48-pin products : Total: 42 (CMOS I/O: 32, CMOS input: 5, CMOS output: 1, N-ch open drain I/O: 4) • 48-pin products : Total: 40 (CMOS I/O: 30, CMOS input: 5, CMOS output: 1, N-ch open drain I/O: 4)
Pull-up resistor	• 32-pin products Total: 11 • 48-pin products Total: 20

Caution Most of the following descriptions in this chapter use the 48-pin products with the 00H setting in peripheral I/O redirection register (PIOR) as an example.

4.2.1 Port 0

Port 0 is an I/O port with an output latch. 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 and P01 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Input to the P00 and P01 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 0 (PIM0).

In the 32-pin products, to use P00 and P01 digital input/output pins, set them in the digital I/O mode by using port mode control register 0 (PMC0) (can be specified in 1-bit units).

This port can also be used for timer I/O.

When reset signal is generated, the following configuration will be set.

- P00 and P01 pins of the 32-pin products … Analog input
- P00 and P01 pins of the 48-pin products … Input mode

4.2.2 Port 1

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P14 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

This port can also be used for timer I/O, clock/buzzer output, external interrupt request input, and interface with USB connector.

Reset signal generation sets port 1 to input mode.

4.2.3 Port 2

Port 2 is an I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units using port mode register 2 (PM2).

This port can also be used for A/D converter analog input and reference voltage input (+ side and - side).

To use P20/ANI0 to P27/ANI7 as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM2 register. Use these pins starting from the upper bit.

To use P20/ANI0 to P27/ANI7 as digital output pins, set them in the digital I/O mode by using the ADPC register and in the output mode by using the PM2 register.

To use P20/ANI0 to P27/ANI7 as analog input pins, set them in the analog input mode by using the A/D port configuration register (ADPC) and in the input mode by using the PM2 register. Use these pins starting from the lower bit.

Table 4-2. Setting Functions of P20/ANI0 to P27/ANI7 Pins

ADPC Register	PM2 Register	ADS Register	P20/ANI0 to P27/ANI7 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	Selects ANI.	Setting prohibited
		Does not select ANI.	

All P20/ANI0 to P27/ANI7 are set in the analog input mode when the reset signal is generated.

4.2.4 Port 3

Port 3 is an I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 and P31 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for external interrupt request input, real-time clock correction clock output, serial interface clock I/O, and timer I/O.

Reset signal generation sets P30 and P31 to input mode.

4.2.5 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 are used as an input port, use of an 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.

Reset signal generation sets port 4 to input mode.

4.2.6 Port 5

Port 5 is an I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 and P51 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

Input to the P50 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 5 (PIM5).

Output from the P50 and P51 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 5 (POM5).

This port can also be used for external interrupt request input, serial interface data I/O, and UART serial data I/O of flash memory programming.

Reset signal generation sets port 5 to input mode.

4.2.7 Port 6

Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6).

The output of the P60 to P63 pins is N-ch open-drain output (6 V tolerance).

This port can also be used for serial interface data I/O and clock I/O.

Reset signal generation sets port 6 to input mode.

4.2.8 Port 7

Port 7 is an I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 7 (PU7).

Output from the P74 pin can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 7 (POM7).

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

Reset signal generation sets port 7 to input mode.

4.2.9 Port 12

P120 is an I/O port with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

P121 to P124 are 4-bit input only ports.

Input to the P120 pin can be specified as analog input or digital input in 1-bit units, using port mode control register 12 (PMC12).

This port can also be used for A/D converter analog input, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, and external clock input for subsystem clock.

Reset signal generation sets P120 to analog input, and sets P121 to P124 to input mode.

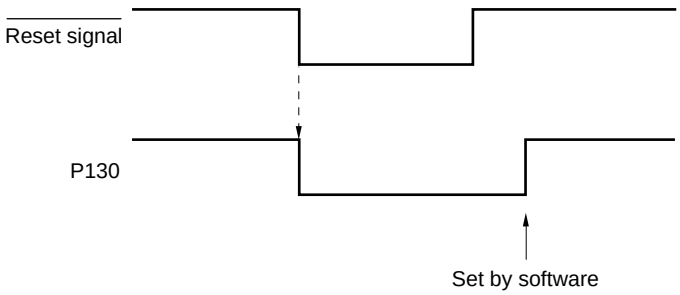
4.2.10 Port 13

P130 is a 1-bit output-only port with an output latch. P137 is a 1-bit input-only port.

P130 is fixed an output port, and P137 is fixed an input ports.

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

Remark When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.



4.2.11 Port 14

Port 14 is an I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

This port can also be used for clock/buzzer output, and external interrupt request input.

Reset signal generation sets P140 to input mode.

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control registers (PMCxx)
- A/D port configuration register (ADPC)
- 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 4-4 and 4-5. Be sure to set bits that are not mounted to their initial values.

Table 4-3. PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (1/3)

Port		Bit name						48 pin	32 pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register		
Port 0	0	PM00	P00	PU00	PIM00	POM00	PMC00 Note	√	√
	1	PM01	P01	PU01	PIM01	POM01	PMC01 Note	√	√
	2	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–
Port 1	0	–	–	–	–	–	–	–	–
	1	–	–	–	–	–	–	–	–
	2	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–
	4	PM14	P14	PU14	–	–	–	√	–
	5	PM15	P15	PU15	–	–	–	√	–
	6	PM16	P16	PU16	–	–	–	√	√
	7	PM17	P17	PU17	–	–	–	√	√
Port 2	0	PM20	P20	–	–	–	–	√	√
	1	PM21	P21	–	–	–	–	√	√
	2	PM22	P22	–	–	–	–	√	√
	3	PM23	P23	–	–	–	–	√	√
	4	PM24	P24	–	–	–	–	√	√
	5	PM25	P25	–	–	–	–	√	–
	6	PM26	P26	–	–	–	–	√	–
	7	PM27	P27	–	–	–	–	√	–

Note 32-pin products only.

Table 4-3. PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (2/3)

Port		Bit name						48 pin	32 pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register		
Port 3	0	PM30	P30	PU30	PIM30	POM30	–	√	√
	1	PM31	P31	PU31	–	–	–	√	√
	2	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–
Port 4	0	PM40	P40	PU40	–	–	–	√	√
	1	PM41	P41	PU41	–	–	–	√	–
	2	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–
Port 5	0	PM50	P50	PU50	PIM50	POM50	–	√	√
	1	PM51	P51	PU51	–	POM51	–	√	√
	2	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–
Port 6	0	PM60	P60	–	–	–	–	√	√
	1	PM61	P61	–	–	–	–	√	√
	2	PM62	P62	–	–	–	–	√	√
	3	PM63	P63	–	–	–	–	√	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–
Port 7	0	PM70	P70	PU70	–	–	–	√	√
	1	PM71	P71	PU71	–	–	–	√	–
	2	PM72	P72	PU72	–	–	–	√	–
	3	PM73	P73	PU73	–	–	–	√	–
	4	PM74	P74	PU74	–	POM74	–	√	–
	5	PM75	P75	PU75	–	–	–	√	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–

Table 4-3. PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (3/3)

Port		Bit name						48 pin	32 pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register		
Port 12	0	PM120	P120	PU120	–	–	PMC120	√	√
	1	–	P121	–	–	–	–	√	√
	2	–	P122	–	–	–	–	√	√
	3	–	P123	–	–	–	–	√	–
	4	–	P124	–	–	–	–	√	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–
Port 13	0	–	P130	–	–	–	–	√	–
	1	–	–	–	–	–	–	–	–
	2	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	P137	–	–	–	–	√	√
Port 14	0	PM140	P140	PU140	–	–	–	√	–
	1	–	–	–	–	–	–	–	–
	2	–	–	–	–	–	–	–	–
	3	–	–	–	–	–	–	–	–
	4	–	–	–	–	–	–	–	–
	5	–	–	–	–	–	–	–	–
	6	–	–	–	–	–	–	–	–
	7	–	–	–	–	–	–	–	–

4.3.1 Port mode registers (PMxx)

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 referencing **4.5 Register Settings When Using Alternate Function**.

Figure 4-1. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	1	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	1	1	1	1	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	1	1	1	1	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	1	1	1	1	1	1	1	PM140	FFF2EH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 7, 12, 14; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

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

4.3.2 Port registers (Pxx)

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 clears these registers to 00H.

Note If P00, P01, P20 to P27, and P120 are set up as analog inputs of the A/D converter, when a port is read while in the input mode, 0 is always returned, not the pin level.

Figure 4-2. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	0	P01	P00	FFF00H	00H (output latch)	R/W
P1	P17	P16	P15	P14	0	0	0	0	FFF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	0	0	0	0	0	0	P31	P30	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H (output latch)	R/W
P5	0	0	0	0	0	0	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	P63	P62	P61	P60	FFF06H	00H (output latch)	R/W
P7	0	0	P75	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W[Note 1]
P13	P137	0	0	0	0	0	0	P130	FFF0DH	**Note 2**	R/W[Note 1]
P14	0	0	0	0	0	0	0	P140	FFF0EH	00H (output latch)	R/W

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

Notes 1. P121 to P124, and P137 are read-only.

2. P137: Undefined

P130: 0 (output latch)

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

Remark m = 0 to 7, 12 to 14; n = 0 to 7

4.3.3 Pull-up resistor option registers (PUxx)

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 set to both normal output mode (POMmn = 0) and input mode (PMmn = 1) for the pins to which the use of an on-chip pull-up resistor has been specified in these registers. 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 clears these registers to 00H (Only PU4 is set to 01H).

Figure 4-3. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	0	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	0	0	0	0	F0031H	00H	R/W
PU3	0	0	0	0	0	0	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	01H	R/W
PU5	0	0	0	0	0	0	PU51	PU50	F0035H	00H	R/W
PU7	0	0	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	0	0	0	0	0	0	0	PU140	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 7, 12, 14; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

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

4.3.4 Port input mode registers (PIM0, PIM3, PIM5)

These registers set the input buffer of P00, P01, P30, and P50 in 1-bit units.

TTL input buffer can be selected during serial communication with an external device of the different potential.

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

Reset signal generation clears these registers to 00H.

Figure 4-4. Format of Port Input Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	0	0	0	PIM01	PIM00	F0040H	00H	R/W
PIM3	0	0	0	0	0	0	0	PIM30	F0043H	00H	R/W
PIM5	0	0	0	0	0	0	0	PIM50	F0045H	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0, 3, 5; n = 0 to 7)
0	Normal input buffer
1	TTL input buffer

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

4.3.5 Port output mode registers (POM0, POM3, POM5, POM7)

These registers set the output mode of P00, P01, P30, P50, P51, and P74 in 1-bit units.

N-ch open drain output (V_{DD} tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA00 and SDA01 pins during simplified I^2C communication with an external device of the same potential.

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

Reset signal generation clears these registers 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 is set.

Figure 4-5. Format of Port Input Mode Register

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
POM3	0	0	0	0	0	0	0	POM30	F0053H	00H	R/W
POM5	0	0	0	0	0	0	POM51	POM50	F0055H	00H	R/W
POM7	0	0	0	POM74	0	0	0	0	F0057H	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 3, 5, 7; n = 0 to 7)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

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

4.3.6 Port mode control registers (PMC0, PMC12)

These registers set the P00, P01, and P120 digital I/O/analog input in 1-bit units.

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

Reset signal generation clears these registers to FFH.

Figure 4-6. Format of Port Mode Control Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	1	1	1	1	1	1	PMC01 **Note 1**	PMC00 **Note 1**	F0060H	FFH	R/W
PMC12	1	1	1	1	1	1	1	PMC120 **Note 2**	F006CH	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 0, 12; n = 0, 1)
0	Digital I/O (alternate function other than analog input)
1	Analog input

Notes 1. 32-pin products only

2. All products mounted

Cautions 1. Select input mode by using port mode registers 0 and 12 (PM0, and PM12) for the ports which are set by the PMCxx register as analog input.

2. Do not set the pin set by the PMCxx register as digital I/O by the analog input channel specification register (ADS).

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

4.3.7 A/D port configuration register (ADPC)

This register switches the P20/ANI0 to P27/ANI7 pins to digital I/O of port or analog input of A/D converter.
The ADPC register can be set by an 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Figure 4-7. Format of A/D Port Configuration Register (ADPC)

Address: F0076H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	0	ADPC3	ADPC2	ADPC1	ADPC0

ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching							
				ANI7/P27	ANI6/P26	ANI5/P25	ANI4/P24	ANI3/P23	ANI2/P22	ANI1/P21	ANI0/P20
0	0	0	0	A	A	A	A	A	A	A	A
0	0	0	1	D	D	D	D	D	D	D	D
0	0	1	0	D	D	D	D	D	D	D	A
0	0	1	1	D	D	D	D	D	D	A	A
0	1	0	0	D	D	D	D	D	A	A	A
0	1	0	1	D	D	D	D	A	A	A	A
0	1	1	0	D	D	D	A	A	A	A	A
0	1	1	1	D	D	A	A	A	A	A	A
1	0	0	0	D	A	A	A	A	A	A	A
Other than above				Setting prohibited							

Cautions 1. Set the port to analog input by ADPC register to the input mode by using port mode registers 2 (PM2).

2. Do not set the pin set by the ADPC register as digital I/O by the analog input channel specification register (ADS).

3. When using AV_{REFP} and AV_{REFM}, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

4.3.8 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, the settings for redirection can be changed only until operation of the function is enabled.
The PIOR register can be set by an 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Figure 4-8. Format of Peripheral I/O Redirection Register (PIOR)

Address: F001AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIOR	0	0	0	0	0	0	PIOR1	PIOR0

Bit	Function	48-pin	
		Setting value	
		0	1
PIOR1	SCLA0	P60	P01
	SDAA0	P61	P00
	TI03/TO03/ INTP4	P31	P41
PIOR0	TI02/TO02	P17	P72
	TI01/TO01/INTP5	P16	P71
	PCLBUZ1	P15	P41

Bit	Function	32-pin	
		Setting value	
		0	1
PIOR1	SCLA0	P60	P01
	SDAA0	P61	P00
PIOR0	TI02/TO02	P17	P50
	TI01/TO01	P16	P51
	PCLBUZ1	P70	P120
	TI03/TO03/PCLBUZ0	P31	P30

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.
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.
The data of the output latch is cleared when a reset signal is generated.

4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers

It is possible to connect an external device operating on a different potential (1.8 V, 2.5 V or 3 V) by switching I/O buffers with the port input mode register (PIMxx) and port output mode register (POMxx).

When receiving input from an external device with a different potential (1.8 V, 2.5 V or 3 V), set the port input mode registers 0, 3, and 5 (PIM0, PIM3, and PIM5) on a bit-by-bit basis to enable normal input (CMOS)/TTL input buffer switching.

When outputting data to an external device with a different potential (1.8 V, 2.5 V or 3 V), set the port output mode registers 0, 3, and 5 (POM0, POM3, and POM5) on a bit-by-bit basis to enable normal output (CMOS)/N-ch open drain (V_{DD} tolerance) switching.

The connection of a serial interface is described in the following.

(1) Setting procedure when using input pins of UART0, and CSI00 functions for the TTL input buffer

In case of UART0: P50 (RxD0)
In case of CSI00: P50 (SI00), P30 (SCK00)

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

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> Set the corresponding bit of the PIM0, PIM3, and PIM5 registers to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.
<3> Enable the operation of the serial array unit and set the mode to the UART/ Simplified SPI (CSI[Note]) mode.

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.

(2) Setting procedure when using output pins of UART0, and CSI00 functions in N-ch open-drain output mode

In case of UART0: P50 (TxD0)
In case of CSI00: P50 (SO00), P30 (SCK00)

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

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM0, POM1, POM4, POM5, and POM14 registers to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.
<5> Enable the operation of the serial array unit and set the mode to the UART/Simplified SPI (CSI) mode.
<6> Set the corresponding bit of the PM0, PM1, PM4, PM5, and PM14 registers to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

(3) Setting procedure when using I/O pin of IIC00 functions with a different potential (1.8 V, 2.5 V, 3 V)

In case of IIC00: P50 (SDA00), P30 (SCL00)

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

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).

<2> After reset release, the port mode is the input mode (Hi-Z).

<3> Set the output latch of the corresponding port to 1.

<4> Set the corresponding bit of the POM3, and POM5 registers to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.

<5> Set the corresponding bit of the PIM3, and PIM5 registers to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.

<6> Enable the operation of the serial array unit and set the mode to the simplified I^2C mode.

<7> Set the corresponding bit of the PM3, and PM5 registers to the output mode (data I/O is possible in the output mode). At this time, the output data is high level, so the pin is in the Hi-Z state.

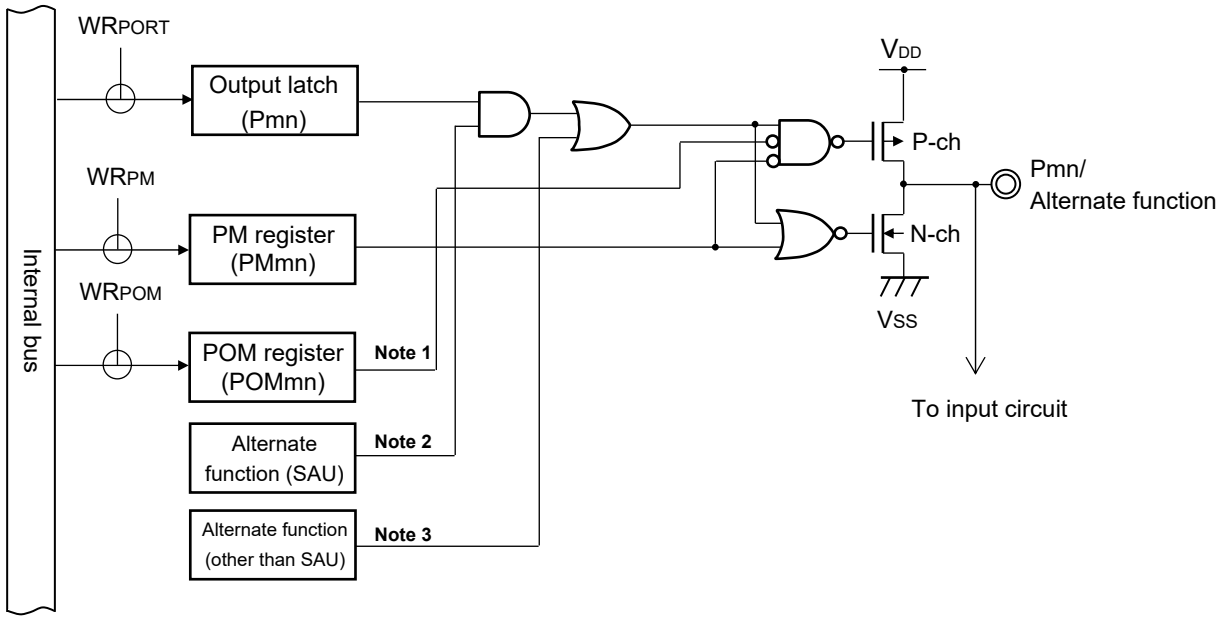
4.5 Register Settings When Using Alternate Function

4.5.1 Basic concept when using alternate function

In the beginning, for a pin also assigned to be used for analog input, use the ADPC register or port mode control register (PMCxx) to specify whether to use the pin for analog input or digital input/output.

Figure 4-10 shows the basic configuration of an output circuit for pins used for digital input/output. The output of the output latch for the port and the output of the alternate SAU function are input to an AND gate. The output of the AND gate is input to an OR gate. The output of an alternate function other than SAU (TAU, RTC, clock/buzzer output, IICA, etc.) is connected to the other input pin of the OR gate. When such kind of pins are used by the port function or an alternate function, the unused alternate function must not hinder the output of the function to be used. An idea of basic settings for this kind of case is shown in Table 4-4.

Figure 4-9. Basic Configuration of Output Circuit for Pins



Notes 1. When there is no POM register, this signal should be considered to be low level (0).

2. When there is no alternate function, this signal should be considered to be high level (1).

3. When there is no alternate function, this signal should be considered to be low level (0).

Remark m: Port number (m = 0 to 7. 12 to 14); n: Bit number (n = 0 to 7)

Table 4-4. Concept of Basic Settings

Output Function of Used Pin	Output Settings of Unused Alternate Function		
	Output Function for Port	Output Function for SAU	Output Function for other than SAU
Output function for port	—	Output is high (1)	Output is low (0)
Output function for SAU	High (1)	—	Output is low (0)
Output function for other than SAU	Low (0)	Output is high (1)	Output is low (0) Note

Caution Since more than one output function other than SAU may be assigned to a single pin, the output of an unused alternate function must be set to low level (0). For details on the setting method, see 4.5.2 Register settings for alternate function whose output function is not used.

4.5.2 Register settings for alternate function whose output function is not used

When the output of an alternate function of the pin is not used, the following settings should be made. Note that when the peripheral I/O redirection function is the target, the output can be switched to another pin by setting the peripheral I/O redirection register (PIOR). This allows usage of the port function or other alternate function assigned to the target pin.

(1) SOp = 1, TxDq = 1 (settings when the serial output (SOp/TxDq) of SAU is not used)
When the serial output (SOp/TxDq) is not used, such as, a case in which only the serial input of SAU is used, set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled) and set the SOmn bit in serial output register m (SOm) to 1 (high). These are the same settings as the initial state.

(2) SCKp = 1, SDAr = 1, SCLr = 1 (settings when channel n in SAU is not used)
When SAU is not used, set bit n (SEmn) in serial channel enable status register m (SEm) to 0 (operation stopped state), set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled), and set the SOmn and CKOmn bits in serial output register m (SOm) to 1 (high). These are the same settings as the initial state.

(3) TOmn = 0 (settings when the output of channel n in TAU is not used)
When the TOmn output of TAU is not used, set the bit in timer output enable register 0 (TOE0) which corresponds to the unused output to 0 (output disabled) and set the bit in timer output register 0 (TO0) to 0 (low). These are the same settings as the initial state.

(4) SDAAn = 0, SCLAn = 0 (setting when IICA is not used)
When IICA is not used, set the IICEn bit in IICA control register n0 (IICCTLn0) to 0 (operation stopped). This is the same setting as the initial state.

(5) PCLBUZn = 0 (setting when clock/buzzer output is not used)
When the clock/buzzer output is not used, set the PCLOEn bit in clock output select register n (CKSn) to 0 (output disabled). This is the same setting as the initial state.

4.5.3 Register setting examples for used port and alternate functions

Register setting examples for used port and alternate functions are shown in Table 4-5. The registers used to control the port functions should be set as shown in Table 4-5. See the following remark for legends used in Table 4-5.

Remark –: Not supported
×: don't care
PIORx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch
Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR).

Table 4-5. Setting Examples of Registers and Output Latches When Using Alternate Function (1/8)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	48-pin
	Function Name	I/O						SAU Output Function	Other than SAU		
P00	P00	Input	–	×	0 Note	1	×	×	×	√	√
		Output	–	0	0 Note	0	0/1	SDA01 = 1	(SDAA0) = 0	√	×
		N-ch open drain output	–	1	0 Note	0	0/1				
	ANI17	Analog input	–	×	1	1	×	×	×	√	×
	TI00	Input	–	×	0 Note	1	×	×	×	√	√
	INTP8	Input	–	×	0	1	×	×	×	√	×
	SI01	Input	–	×	0	1	×	×	×	√	×
	SDA01	I/O	–	1	0	0	1	×	×	√	×
	(SDAA0)	I/O	PIOR1 = 1	1	0	0	1	×	×	×	√
P01	P01	Input	–	×	0 Note	1	×	×	×	√	√
		Output	–	0	0 Note	0	0/1	SCK01 = 1 SCL01 = 1 Note	TO00 = 0 SCLA0 = 0		
		N-ch open drain output	–	1	0	0	0/1				
	ANI16	Analog input	–	×	1	1	×	×	×	√	×
	TO00	Output	–	0	0 Note	0	0	×	×	√	√
	INTP9	Input	–	×	0	1	×	×	×	√	×
	SCK01	Input	–	×	0	1	×	×	×	√	×
		Output	–	0/1	0	0	1	×	×		
	SCL01	Output	–	0/1	0	0	1	×	×	√	×
	(SCLA0)	Output	PIOR1 = 1	1	0	0	1	×	×	√	√

Note 32-pin products only

Table 4-5. Setting Examples of Registers and Output Latches When Using Alternate Function (2/8)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	48-pin
	Function Name	I/O						SAU Output Function	Other than SAU		
P14	P14	Input	–	–	–	1	×	–	×	×	√
		Output	–	–	–	0	0/1	–	×		
	UOVRCUR0 Note	Input	–	–	–	1	×	–	×	×	√
P15	P15	Input	–	–	–	1	×	–	×	×	√
		Output	–	–	–	0	0/1	–	UVBUSEN0 = 0 Note PCLBUZ1 = 0		
	PCLBUZ1	Output	PIOR0 = 0	–	–	0	0	–	UVBUSEN0 = 0 Note	×	√
	UVBUSEN0 Note	Output	–	–	–	0	0	–	PCLBUZ1 = 0	×	√
P16	P16	Input	–	–	–	1	×	–	×	√	√
		Output	–	–	–	0	0/1	–	TO01 = 0		
	TI01	Input	PIOR0 = 0	–	–	1	×	–	×	√	√
	TO01	Output	PIOR0 = 0	–	–	0	0	–	×	√	√
	INTP5	Input	PIOR0 = 0	–	–	1	×	–	×	√	√
	UOVRCUR1 Note	Input	–	–	–	1	×	–	×	√	√
P17	P17	Input	–	–	–	1	×	–	×	√	√
		Output	–	–	–	0	0/1	–	UVBUSEN1 = 0 Note TO02 = 0		
	TI02	Input	PIOR0 = 0	–	–	1	×	–	×	√	√
	TO02	Output	PIOR0 = 0	–	–	0	0	–	UVBUSEN1 = 0 Note	√	√
	UVBUSEN1 Note	Output	–	–	–	0	0	–	TO02 = 0	√	√

Note This function has not been mounted in the R5F10K product.

Table 4-5. Setting Examples of Registers and Output Latches When Using Alternate Function (3/8)

Pin Name	Used Function		ADPC	ADM	PMxx	Pxx	32-pin	48-pin
	Function Name	I/O						
P20	P20	Input	ADPC = 01H	×	1	×	√	√
		Output	ADPC = 01H	×	0	0/1		
	ANI0	Analog input	ADPC = 00H/ 02H to 0FH	00x0xx0x, 10x0xx0x	1	×	√	√
	AV_{REFP}	Reference voltage	ADPC = 00H/ 02H to 0FH	01x0xx0x	1	×	√	√
P21	P21	Input	ADPC = 01H/02H	×	1	×	√	√
		Output	ADPC = 01H/02H	×	0	0/1		
	ANI1	Analog input	ADPC = 00H/ 3 to 0FH	xx00xx0x	1	×	√	√
	AV_{REFM}	Reference voltage	ADPC = 00H/ 3 to 0FH	xx10xx0x	1	×	√	√
P22	P22	Input	ADPC = 01H to 03H	×	1	×	√	√
		Output	ADPC = 01H to 03H	×	0	0/1		
	ANI2	Analog input	ADPC = 00H/ 04H to 0FH	×	1	×	√	√
P23	P23	Input	ADPC = 01 to 04H	×	1	×	√	√
		Output	ADPC = 01 to 04H	×	0	0/1		
	ANI3	Analog input	ADPC = 00H/ 5H to 0FH	×	1	×	√	√
P24	P24	Input	ADPC = 01H to 05H	×	1	×	×	√
		Output	ADPC = 01H to 05H	×	0	0/1		
	ANI4	Analog input	ADPC = 00H/ 06H to 0FH	×	1	×	×	√
P25	P25	Input	ADPC = 01H to 06H	×	1	×	×	√
		Output	ADPC = 01H to 06H	×	0	0/1		
	ANI5	Analog input	ADPC = 00H/ 07H to 0FH	×	1	×	×	√
P26	P26	Input	ADPC = 01H to 07H	×	1	×	×	√
		Output	ADPC = 01H to 07H	×	0	0/1		
	ANI6	Analog input	ADPC = 00H/ 08H to 0FH	×	1	×	×	√
P27	P27	Input	ADPC = 01H to 08H	×	1	×	×	√
		Output	ADPC = 01H to 08H	×	0	0/1		
	ANI7	Analog input	ADPC = 00H/ 09H to 0FH	×	1	×	×	√

Table 4-5. Setting Examples of Registers and Output Latches When Using Alternate Function (4/8)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	48-pin
	Function Name	I/O						SAU Output Function	Other than SAU		
P30	P30	Input	–	×	–	1	×	×	×	√	√
		Output	–	0	–	0	0/1	SCK00/ SCL00 = 1	RTC1HZ = 0 Note 2 (TO03) = 0 Note 1 (PCLBUZ0) = 0 Note 1		
		N-ch open drain output	–	1	–	0	0/1			√	√
	INTP3	Input	–	×	–	1	×	×	×	√	×
	RTC1HZ	Output	–	0	–	0	0	SCK00/ SCL00 = 1	×	√	×
	SCK00	Input	–	×	–	1	×	×	×	√	×
		Output	–	0/1	–	0	1	×	RTC1HZ = 0 Note 2 (TO03) = 0 Note 1 (PCLBUZ0) = 0 Note 1	√	×
	SCL00	Output	–	0/1	–	0	1	×		√	×
	(TI03)	Input	PIOR0 = 1 Note 2	×	–	1	×	×	×	√	×
	(TO03)	Output	PIOR0 = 1 Note 2	0	–	0	0	SCK00/ SCL00 = 1	(PCLBUZ0) = 0 Note 1	√	×
	(PCLBUZ0)	Output	PIOR0 = 1 Note 2	0	–	0	0		(TO03) = 0 Note 1	√	×
P31	P31	Input	–	–	–	1	×	–	×	√	√
		Output	–	–	–	0	0/1	–	TO03 = 0 PLCBUZ0 = 0 Note 1 (PCLBUZ0) = 0 Note 1		
	TI03	Input	PIOR0 = 0 PIOR1 = 0	–	–	1	×	–	×	√	√
	TO03	Output	PIOR0 = 0 PIOR1 = 0	–	–	0	0	–	PLCBUZ0 = 0 Note1 UVBUSEN0 = 0 Note 1	√	√
	INTP4	Input	PIOR1 = 0 Note 2	–	–	1	×	–	×	√	√
	PCLBUZ0	Output	PIOR0 = 0 Note 1	–	–	0	0	–	TO03 = 0 UVBUSEN0 = 0 Note 1	√	×
	UVBUSEN0 Note 3	Output	–	–	–	0	0	–	TO03 = 0 PLCBUZ0 = 0 Note 1	√	×

Notes 1. 32-pin products only

2. 48-pin products only

3. This function has not been mounted in the R5F10K product.

Table 4-5. Setting Examples of Registers and Output Latches When Using Alternate Function (5/8)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	48-pin
	Function Name	I/O						SAU Output Function	Other than SAU		
P40	P40	Input	–	–	–	1	×	–	×	√	√
		Output	–	–	–	0	0/1	–	×		
P41	P41	Input	–	–	–	1	×	–	×	√	√
		Output	–	–	–	0	0/1	–	(TO03) = 0 (PCLBUZ1) = 0 Note 2		
	(TI03)	Input	PIOR1 = 1 Note 2	–	–	1	×	–	×	×	√
	(TO03)	Output	PIOR1 = 1 Note 2	–	–	0	0	–	(PCLBUZ1) = 0 Note 2	×	√
	(INTP4)	Input	PIOR1 = 1 Note 2	–	–	1	×	–	×	×	√
	(PCLBUZ1)	Output	PIOR1 = 1 Note 2	–	–	0	0	–	(TO03) = 0 Note 2	×	√
P50	P50	Input	–	×	–	1	×	×	×	√	√
		Output	–	0	–	0	0/1	SDA00 = 1	(TO02) = 0 Note 2		
		N-ch open drain output	–	1	–	0	0/1		×		
	INTP1	Input	–	×	–	1	×	×	×	√	√
	SI00	Input	–	×	–	1	×	×	×	√	√
	RxD0	Input	–	×	–	1	×	×	×	√	√
	SDA00	I/O	–	1	–	0	1	×	(TO02) = 0 Note 2	√	√
	(TI02)	Input	PIOR0 = 1 Note 1	×	–	1	×	×	×	√	×
	(TO02)	Output	PIOR0 = 1 Note 1	0	–	0	1	SDA00 = 1	×	√	×
P51	P51	Input	–	×	–	1	×	×	×	√	√
		Output	–	0	–	0	0/1	SO00/ TxD00 = 1	(TO01) = 0 Note 1		
		N-ch open drain output	–	1	–	0	0/1				
	INTP2	Input	–	×	–	1	×	×	×	√	√
	SO00	Output	–	0/1	–	0	1	×	(TO01) = 0 Note 1	√	√
	TxD0	Output	–	0/1	–	0	1	×		√	√
	(TI01)	Input	PIOR0 = 1 Note 1	×	–	1	×	×	×	√	×
	(TO01)	Output	PIOR0 = 1 Note 1	0	–	0	0	SO00/ TxD00 = 1	×	√	×
P60	P60	Input	–	–	–	1	×	–	×	√	√
		N-ch open drain output (6-V tolerance))	–	–	–	0	0/1	–	SCLA0 = 0		
	SCLA0	I/O	PIOR1 = 0	–	–	0	0	–	×	√	√
P61	P61	Input	–	–	–	1	×	–	×	√	√
		N-ch open drain output (6-V tolerance)	–	–	–	0	0/1	–	SDAA0 = 0		
	SDAA0	I/O	PIOR1 = 0	–	–	0	0	–	×	√	√
P62	P62	Input	–	–	–	1	×	–	×	√	√
		Output	–	–	–	0	0/1	–	×		
P63	P63	Input	–	–	–	1	×	–	×	×	√
		Output	–	–	–	0	0/1	–	×		

Notes 1. 32-pin products only

2. 48-pin products only

Table 4-5. Setting Examples of Registers and Output Latches When Using Alternate Function (6/8)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	48-pin
	Function Name	I/O						SAU Output Function	Other than SAU		
P70	P70	Input	–	–	–	1	×	–	×	√	√
		Output	–	–	–	0	0/1	–	PCLBUZ1 = 0		
	KR0	Input	–	–	–	1	×	–	×	×	√
	PCLBUZ1	Output	PIOR0 = 0 Note 1	–	–	0	0	–	×	√	×
	UOVRCUR0 Note 3	Input	–	–	–	1	×	–	×	√	×
P71	P71	Input	–	–	–	1	×	–	×	×	√
		Output	–	–	–	0	0/1	–	TO01 = 0		
	KR1	Input	–	–	–	1	×	–	×	×	√
	(TI01)	Input	PIOR0 = 1	–	–	1	×	–	×	×	√
	(TO01)	Output	PIOR0 = 1	–	–	0	0	–	×	×	√
	(INTP5)	Input	PIOR0 = 1	–	–	1	×	–	×	×	√
P72	P72	Input	–	–	–	1	×	–	×	×	√
		Output	–	–	–	0	0/1	–	TO02 = 0		
	KR2	Input	–	–	–	1	×	–	×	×	√
	(TI02)	Input	PIOR0 = 1	–	–	1	×	–	×	×	√
	(TO02)	Output	PIOR0 = 1	–	–	0	0	–	×	×	√
P73	P73	Input	–	–	–	1	×	–	–	×	√
		Output	–	–	–	0	0/1	SO01 = 1 Note 2	–		
	KR3	Input	–	–	–	1	×	×	–	×	√
	SO01	Output	–	–	–	0	1	–	–	×	√
P74	P74	Input	–	×	–	1	×	×	–	×	√
		Output	–	0	–	0	0/1	SDA01 = 1 Note 2	–		
		N-ch open drain output	–	1	–	0	0/1				
	KR4	Input	–	×	–	1	×	×	–	×	√
	INTP8	Input	–	×	–	1	×	×	–	×	√
	SI01	Input	–	×	–	1	×	×	–	×	√
	SDA01	I/O	–	1	–	0	1	×	–	×	√
P75	P75	Input	–	–	–	1	×	×	–	×	√
		Output	–	–	–	0	0/1	SCK01/ SCL01 = 1 Note 2	–		
	KR5	Input	–	–	–	1	×	×	–	×	√
	INTP9	Input	–	–	–	1	×	×	–	×	√
	SCK01	Input	–	–	–	1	×	×	–	×	√
		Output	–	–	–	0	1	×	–	×	√
	SCL01	Output	–	–	–	0	1	×	–	×	√
P120	P120	Input	–	–	0	1	×	×	×	√	√
		Output	–	–	0	0	0/1	SO01 = 1 Note 1	(PCLBUZ1) = 0 Note 1		
	ANI19	Analog input	–	–	1	1	×	×	×	√	√
	SO01	Output	–	–	0	0	1	×	(PCLBUZ1) = 0 Note 1	√	×
	(PCLBUZ1)	Output	PIOR0 = 1 Note 1	–	0	0	0	SO01 = 1 Note 1	×	√	×

Notes 1. 32-pin products only
2. 48-pin products only
3. This function has not been mounted in the R5F10K product.

Table 4-5. Setting Examples of Registers and Output Latches When Using Alternate Function (7/8)

Pin Name	Used Function		CMC	Pxx	32-pin	48-pin
	Function Name	I/O	(EXCLK,OSCSEL, EXCLKS, OSCSELS)			
P121	P121	Input	00xx/10 xx/11 xx	×	√	√
	X1	–	01 xx	–	√	√
P122	P122	Input	00 xx/10 xx	×	√	√
	X2	–	01 xx	–	√	√
	EXCLK	Input	11 xx	–	√	√
P123	P123	Input	xx 00/xx 10/xx11	×	×	√
	XT1	–	xx 01	–	×	√
P124	P123	Input	xx 00/xx 10	×	×	√
	XT2	–	xx 01	–	×	√
	EXCLKS	Input	xx 11	–	×	√

Table 4-5. Setting Examples of Registers and Output Latches When Using Alternate Function (8/8)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	48-pin
	Function Name	I/O						SAU Output Function	Other than SAU		
P130	P130	Output	–	–	–	–	0/1	–	–	×	√
P137	P137	Input	–	–	–	–	×	–	–	√	√
	INTP0	Input	–	–	–	–	×	–	–	√	√
P140	P140	Input	–	–	–	1	×	–	×	×	√
		Output	–	–	–	0	0/1	–	PCLBUZ0 = 0 Note		
	PCLBUZ0	Output	–	–	–	0	0	–	×	×	√
	INTP6	Input	–	–	–	1	×	–	×	×	√

Note 48-pin products only

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 P20 is an output port, P21 to P27 are input ports (all pin statuses are high level), and the port latch value of port 2 is 00H, if the output of output port P20 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 2 is FFH.

Explanation: The targets of writing to and reading from the Pn register of a port whose PMnm 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/G1C.

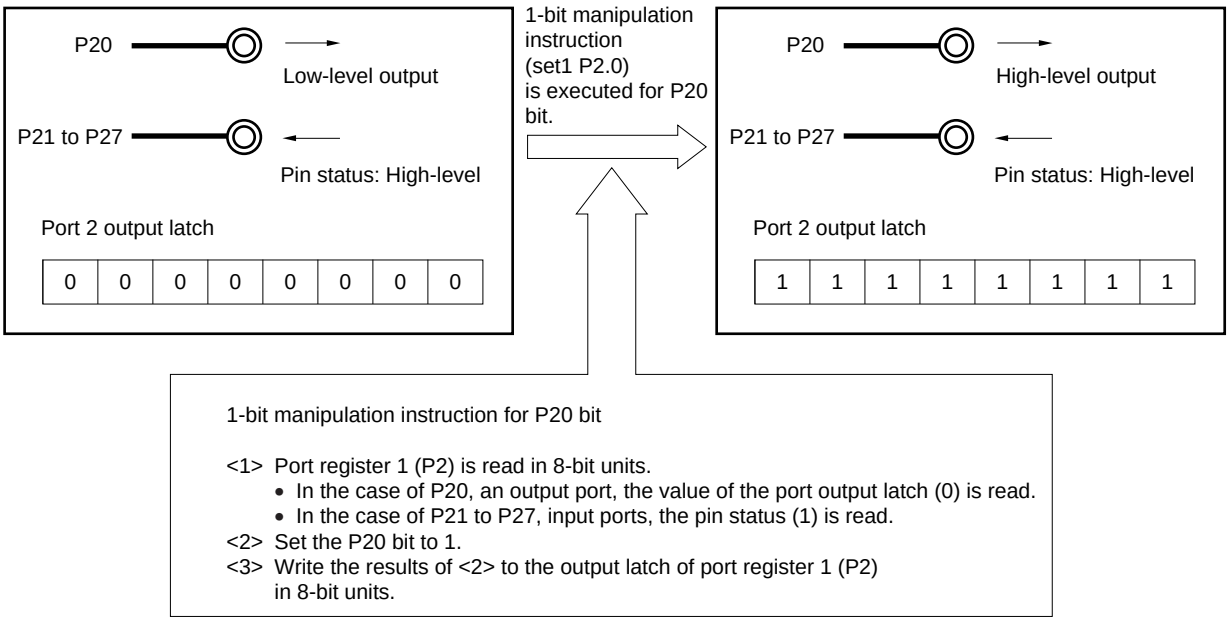
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P20, which is an output port, is read, while the pin statuses of P21 to P27, which are input ports, are read. If the pin statuses of P21 to P27 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 4-10. Bit Manipulation Instruction (P20)



4.6.2 Cautions on the pin settings on the products other than 48-pin

For an output pin to which multiple 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 function output, see **4.5 Register Settings When Using Alternate Function**.

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

The presence or absence of connecting resonator pin for main system clock, connecting resonator pin for subsystem clock, external clock input pin for main system clock, and external clock input pin for subsystem clock, depends on the product.

Output pin	32-pin	48-pin
X1, X2 pins	√	√
EXCLK pin	√	√
XT1, XT2 pins	–	√
EXCLKS pin	–	√

Caution The 32-pin products don't have the subsystem clock.

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

This circuit oscillates a clock of f_X = 1 to 20 MHz by connecting a resonator to X1 and X2 pins.
Oscillation can be stopped 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 f_{HOCO} = 48 MHz (typ.) or 24 MHz (typ.) by using the option byte (000C2H). When the CKSELR bit (bit 0 of the MCKC register) is set to 0, a clock generated by dividing the f_{HOCO} frequency by 2, 4, or 8 (divided by 2 is the default) by setting the RDIV0 and RDIV1 bits (bits 1 and 2 of the MCKC register) is selected as the main system clock source (f_{IH}). After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock.
When selecting the high-speed on-chip oscillator as the USB clock (this setting is only available when using the USB function controller in low-speed mode), be sure to select 48 MHz as the f_{HOCO} frequency and then select the main system clock source (f_{IH}) with f_{HOCO} divided as the main system clock (f_{MAIN}).
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 Select Register (HOCODIV)**.

<3> High-speed system clock multiplication function using PLL (phase locked loop)

This clock function is mainly used for clock supply to the USB host/function controller. Set the DSCCTL register so that the PLL oscillation frequency (f_{PLL}) is 48 MHz. When the CKSELR bit is set to 1, a clock generated by dividing the f_{PLL} frequency by 2, 4, or 8 by setting the RDIV0 and RDIV1 bits is selected as the main system clock source (f_{IH}). When selecting the PLL clock as the USB source clock, be sure to select the main system clock source (f_{IH}) with f_{PLL} divided as the main system clock (f_{MAIN})..
The PLL can be operated or stopped by setting the DSCON bit (bit 0 in the DSCCTL register). For details on PLL settings and the connection with the USB clock, see **Figure 5-12** and **Table 5-3**.

An external main system clock (f_{EX} = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of the MSTOP bit.
As the main system clock, a high-speed system clock (X1 clock or external main system clock) or high-speed on-chip oscillator clock or PLL clock can be selected by setting of the MCM0 bit (bit 4 of the system clock control register (CKC)).
However, note that the usable frequency range of the main system clock differs depending on the setting of the power supply voltage (V_{DD}). The operating voltage of the flash memory must be set by using the CMODE0 and CMODE1 bits of the option byte (000C2H) (see **CHAPTER 25 OPTION BYTE**).

(2) Subsystem clock

- **XT1 clock oscillator**

  This circuit oscillates a clock of f_{XT} = 32.768 kHz by connecting a 32.768 kHz resonator to XT1 and XT2 pins. Oscillation can be stopped by setting the XTSTOP bit (bit 6 of the clock operation status control register (CSC)).

An external subsystem clock (f_{EXS} = 32.768 KHz) can also be supplied from the EXCLKS/XT2/P124 pin. An external subsystem clock input can be disabled by the setting of the XTSTOP bit.

(3) Low-speed on-chip oscillator clock (Low-speed on-chip oscillator)

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
- Real-time clock
- 12-bit interval timer

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC), 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 low-speed on-chip oscillator stops if the HALT or STOP instruction is executed.

Caution The low-speed on-chip oscillator clock (f_{IL}) can only be selected as the real-time clock count clock when the fixed-cycle interrupt function is used.

Remark f_X: X1 clock oscillation frequency
f_{IH}: Main system clock source frequency divided by 2, 4, or 8 when the high-speed on-chip oscillator clock or the PLL clock is selected (24 MHz max.)
f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)
f_{EX}: External main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{PLL}: PLL oscillation frequency

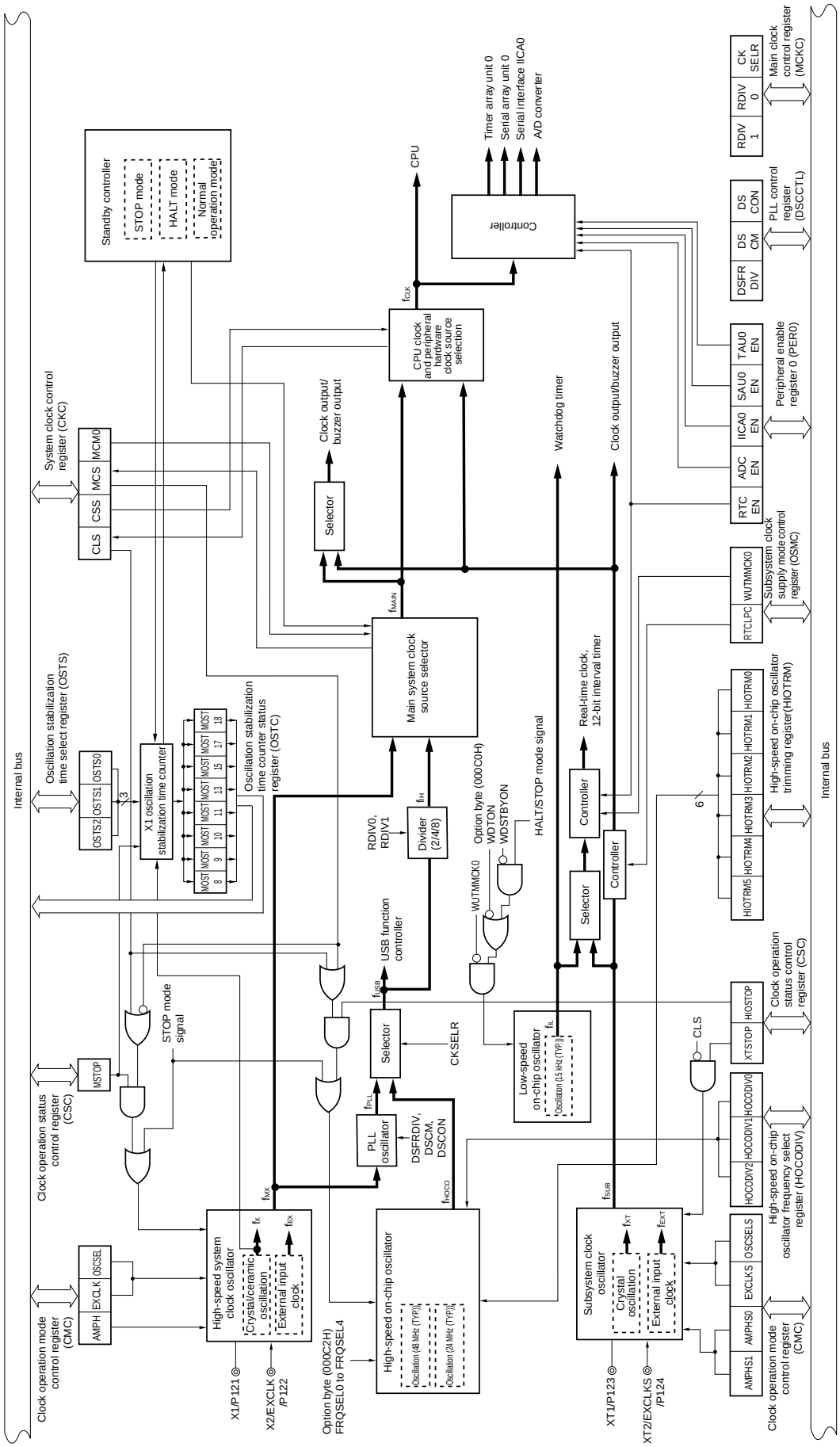
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
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) Subsystem clock supply mode control register (OSMC) High-speed on-chip oscillator frequency select register (HOCODIV) High-speed on-chip oscillator trimming register (HIOTRM) PLL control register (DSCCTL) Main clock control register (MCKC)
Oscillators	X1 oscillator XT1 oscillator High-speed on-chip oscillator Low-speed on-chip oscillator PLL oscillator

Figure 5-1. Block Diagram of Clock Generator



Note Be sure to select f_{IH} when using the USB host/function controller.
(**Remark** is listed on the next page after next.)

Remark f_X: X1 clock oscillation frequency
f_{IH}: Main system clock source frequency divided by 2, 4, or 8 when the high-speed on-chip oscillator clock or the PLL clock is selected (24 MHz max.)
f_{HOCO}: High-speed on-chip oscillator clock frequency (48 MHz max.)
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency
f_{PLL}: PLL oscillation frequency
f_{USB}: USB clock frequency

5.3 Registers Controlling Clock Generator

The following 11 registers are used to control the clock generator.

- 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)
- Subsystem clock supply mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- High-speed on-chip oscillator trimming register (HIOTRM)
- PLL control register (DSCCTL)
- Main clock control register (MCKC)

5.3.1 Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/P121, X2/EXCLK/P122, XT1/P123, and XT2/EXCLKS/P124 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	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	AMPH

EXCLK	OSCSEL	High-speed system clock pin operation mode	X1/P121 pin	X2/EXCLK/P122 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

EXCLKS	OSCSELS	Subsystem clock pin operation mode	XT1/P123 pin	XT2/EXCLKS/P124 pin
0	0	Input port mode	Input port	
0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

AMPHS1	AMPHS0	XT1 oscillator oscillation mode selection
0	0	Low power consumption oscillation (default)
0	1	Normal oscillation
1	0	Ultra-low power consumption oscillation
1	1	Setting prohibited

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 or XT1 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.

4. Specify the settings for the AMPH, AMPHS1, and AMPHS0 bits while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX} or f_{sub}).

5. Oscillation stabilization time of f_{XT}, counting on the software.

6. Although the maximum system clock frequency is 24 MHz, the maximum frequency of the X1 oscillator is 20 MHz.

(**Cautions** and **Remark** are given on the next page.)

Cautions 7. The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **Before using the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators described in 5.7 Resonator and Oscillator Constants.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as V_{SS} as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

Remark fx: X1 clock frequency

5.3.2 System clock control register (CKC)

This register is used to select a CPU/peripheral hardware clock and 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 1]

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

CLS	Status of CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

CSS	Selection of CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1 [Note 2]	Subsystem clock (f_{SUB})

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

MCM0 [Note 2]	Main system clock (f_{MAIN}) operation control
0	Selects the high-speed on-chip oscillator clock/PLL clock (f_{IH}) as the main system clock (f_{MAIN}) [Note 3]
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

Notes 1. Bits 7 and 5 are read-only.

2. Changing the value of the MCM0 bit is prohibited while the CSS bit is set to 1.

3. High-speed on-chip oscillator clock or PLL clock is selected by the CKSELR bit of the MCKC register.

Remark f_{IH}: Main system clock source frequency divided by 2, 4, or 8 when the high-speed on-chip oscillator clock or the PLL clock is selected (24 MHz max.)

f_{MX}: High-speed system clock frequency

f_{MAIN}: Main system clock frequency

f_{SUB}: Subsystem clock frequency

Cautions 1. Be sure to clear bit 3 to 0 to 0.

2. The clock set by the CSS bit is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the real-time clock, 12-bit interval timer, clock output/buzzer output, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.

3. If the subsystem clock is used as the peripheral hardware clock, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 30 and CHAPTER 31 ELECTRICAL SPECIFICATIONS.

5.3.3 Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock, high-speed on-chip oscillator clock, and subsystem 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 C0H.

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

Address: FFFA1H After reset: C0H R/W

Symbol	<7>	<6>	5	4	3	2	1	<0>
CSC	MSTOP	XTSTOP	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	External clock from EXCLK pin is valid	Input port
1	X1 oscillator stopped	External clock from EXCLK pin is invalid	

XTSTOP	Subsystem clock operation control		
	XT1 oscillation mode	External clock input mode	Input port mode
0	XT1 oscillator operating	External clock from EXCLKS pin is valid	Input port
1	XT1 oscillator stopped	External clock from EXCLKS pin is invalid	

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

Cautions 1. After reset release, set the clock operation mode control register (CMC) before setting the CSC register.

2. 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.

3. 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).

4. When starting XT1 oscillation by setting the XTSTOP bit, wait for oscillation of the subsystem clock to stabilize by setting a wait time using software.

5. Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the OSC register.

6. The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) 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. Stopping Clock Method

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock External main system clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock. (CLS = 0 and MCS = 0, or CLS = 1)	MSTOP = 1
XT1 clock External subsystem clock	CPU and peripheral hardware clocks operate with a clock other than the subsystem clock. (CLS = 0)	XTSTOP = 1
High-speed on-chip oscillator clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed on-chip oscillator clock. (CLS = 0 and MCS = 1, or CLS = 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 or subsystem 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

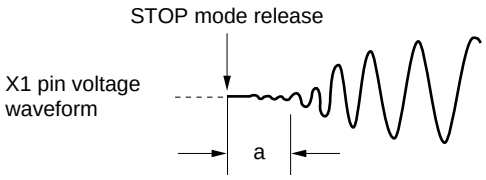
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	102 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204 μs min.	102 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819 μs min.	409 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.63 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.1 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx 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.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem 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 X1 clock is made to oscillate by clearing the MSTOP bit to start the X1 oscillation circuit operating, actual operation is automatically delayed for the time set in the OSTS register.

When switching the CPU clock from the high-speed on-chip oscillator clock or the subsystem clock to the X1 clock, and when using the high-speed on-chip oscillator clock for switching the X1 clock from the oscillating state to STOP mode, use the oscillation stabilization time counter status register (OSTC) to confirm that the desired oscillation stabilization time has elapsed after release from the STOP mode. That is, use the OSTC register to check that the oscillation stabilization time corresponding to its setting has been reached.

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

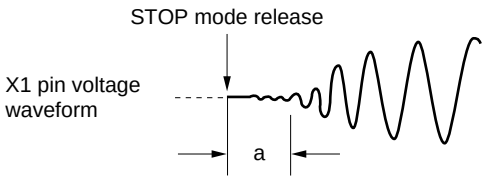
OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	12.8 μs
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204 μs	102 μs
1	0	0	$2^{13}/f_X$	819 μs	409 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.63 ms
1	1	0	$2^{17}/f_X$	13.6 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.2 ms	13.1 ms

Cautions 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

2. 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 or subsystem 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

5.3.6 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.

- Real-time clock, 12-bit interval timer
- A/D converter
- Serial interface IICA0
- Serial array unit 0
- Timer array unit 0

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	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

RTCEN	Control of real-time clock (RTC) and 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. • The real-time clock (RTC) and 12-bit interval timer are in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer 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.

IICA0EN	Control of serial interface IICA0 input clock supply
0	Stops input clock supply. • SFR used by the serial interface IICA0 cannot be written. • The serial interface IICA0 is in the reset status.
1	Enables input clock supply. • SFR used by the serial interface IICA0 can be read and written.

Caution Be sure to clear bits 1, 3, 6 to 0.

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	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

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

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

Caution Be sure to clear bits 1, 3, 6 to 0.

5.3.7 Subsystem clock supply mode control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except the real-time clock and 12-bit interval timer, is stopped in STOP mode or HALT mode while subsystem clock is selected as CPU clock.

In addition, the OSMC register can be used to select the count clock of the real-time clock and 12-bit interval timer.

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 Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

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

RTCLPC	Setting in STOP mode or HALT mode while subsystem clock is selected as CPU clock
0	Enables supply of subsystem clock to peripheral functions (See Table 19-1, Table 19-2, and Table 19-3 for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than real-time clock and 12-bit interval timer.

WUTMMCK0	Selection of count clock for real-time clock and 12-bit interval timer.
0	Subsystem clock (f_{SUB})
1	Low-speed on-chip oscillator clock (f_{IL})

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

The frequency of the high-speed on-chip oscillator which is set by an option byte (000C2H) can be changed by using high-speed on-chip oscillator frequency select register (HOCODIV).

The HOCODIV register 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 Select Register (HOCODIV)

Address: F00A8H After reset: the 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	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	High-Speed On-Chip Oscillator Clock Frequency[Note]
0	0	0	f_{HOCO} = 48 MHz, f_{IH} = 24/12/6 MHz
0	0	1	f_{HOCO} = 24 MHz, f_{IH} = 12/6/3 MHz
0	1	0	f_{HOCO} = 12 MHz, f_{IH} = 6/3 MHz
0	1	1	f_{HOCO} = 6 MHz, f_{IH} = 3 MHz
Other than aboves			Setting prohibited

Note A clock devided by 2/4/8 set by the RDIV1 bit and RDIV0 bit of the MCKC register is supplied for the system clock. However, set the frequency $f_{CLK} \geq 1$ MHz.

Cautions 1. Set the HOCODIV register within the operable voltage range of the flash operation mode set in the option byte (000C2H) before and after the frequency change.

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 5.5 V
			1 to 24 MHz	2.7 to 5.5 V

2. Set the HOCODIV register with the high-speed on-chip oscillator clock (f_{HOCO}) 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 three clocks at the pre-change frequency**
- **CPU/peripheral hardware clock wait at the post-change frequency for up to three clocks**

5.3.9 High-speed on-chip oscillator trimming register (HIOTRM)

This register is used to adjust the accuracy of the high-speed on-chip oscillator.

With self-measurement of the high-speed on-chip oscillator frequency via a timer using high-accuracy external clock input (timer array unit), and so on, the accuracy can be adjusted.

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

Caution The frequency will vary if the temperature and V_{DD} pin voltage change after accuracy adjustment. When the temperature and V_{DD} voltage change, accuracy adjustment must be executed regularly or before the frequency accuracy is required.

Figure 5-10. Format of High-Speed On-Chip Oscillator Trimming Register (HIOTRM)

Address: F00A0H After reset: **Note** R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	0	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0

HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0	High-speed on-chip oscillator
0	0	0	0	0	0	Minimum speed
0	0	0	0	0	1	↑
0	0	0	0	1	0	
0	0	0	0	1	1	
0	0	0	1	0	0	
• • •						
1	1	1	1	1	0	↓
1	1	1	1	1	1	Maximum speed

Note The value after a reset is adjusted at the time of shipment.

Cautions 1. The HIOTRM register holds a six-bit value used to adjust the high-speed on-chip oscillator with an increment of 1 corresponding to an increase of frequency by about 0.05%.

2. For the usage example of the HIOTRM register, see the application note for RL78 MCU series High-speed On-chip Oscillator (HOCO) Clock Frequency Correction (R01AN0464).

5.3.10 PLL control register (DSCCTL)

This register is used to control the operations of the PLL oscillator.

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

Reset signal generation clears this register to 00H.

Figure 5-11. Format of PLL Control Register (DSCCTL)

Address: F02E5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DSCCTL	0	0	0	0	0	DSFRDIV	DSCM	DSCON

DSFRDIV	PLL reference clock divider control
0	No division
1	Divided by 2

Remark PLL reference clock is the high-speed system clock (f_{MX}).

DSCM	PLL multiplication selection
0	12 times (6 times)
1	16 times (8 times)

Remark The frequency is devided by 2 in the last stage of the PLL oscillator, therefore the multiplication ratio becomes the value in parentheses.

DSCON	PLL oscillation and output control
0	Stop
1	Ocsillation, output

Cautions 1. Be sure to clear bits 3 to 7 to 0.

2. Be sure to set the DSCON bit to 0 before changing DSFRDIV and DSCM.

3. Do not set the DSCON bit to 0 while the PLL clock is selected as the system clock.

The combination which user can select as the USB clock when the PLL is used is shown below.

Figure 5-12. Relationship between the PLL and the USB Clock

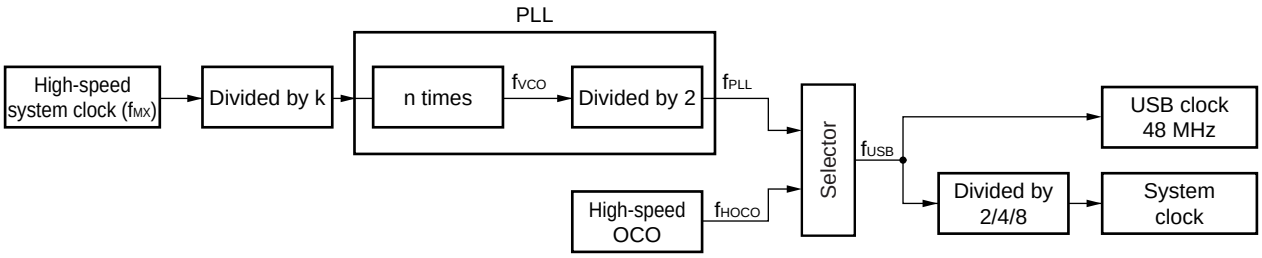


Table 5-3. USB Clock Frequency Setting Example

High-speed system clock (f_{MX})	Divided by k DSFRDIV	n times DSCM		Frequency after n times (f_{VCO})	USB clock (f_{USB})
16 MHz	Divided by 2	0	12 times	96 MHz	48 MHz
12 MHz	Divided by 2	1	16 times	96 MHz	48 MHz
8 MHz	No division	0	12 times	96 MHz	48 MHz
6 MHz	No division	1	16 times	96 MHz	48 MHz

5.3.11 Main clock control register (MCKC)

This register is used to control the operations of the main clock.

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

Reset signal generation clears this register to 00H.

Figure 5-13. Format of Main Clock Control Register (MCKC)

Address: F02E6H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MCKC	0	0	0	0	0	RDIV1	RDIV0	CKSELR

RDIV1	RDIV0	High-speed on-chip oscillator clock/PLL clock division ratio (divided by 2/4/8) selection
0	0	Divided by 2
0	1	Divided by 4
1	0	Divided by 8
1	1	

CKSELR	High-speed on-chip oscillator clock/PLL clock selection
0	High-speed on-chip oscillator clock (f_{HOCO})
1	PLL clock (f_{PLL})

Cautions 1. When using the USB, use the clock specified by using this bit as the USB clock (The high-speed on-chip oscillator can be used only when using the USB function controller in low-speed mode). When the USB is used, set this bit to 1 to select the PLL clock. In that case, be sure to set the MCM0 bit to 0.

2. When the clock is switched between the PLL clock (f_{PLL}) and the high-speed on-chip oscillator clock (f_{HOCO}), both clocks must be oscillating.

3. Be sure to clear bits 3 to 7 to 0.

Remark A clock selected by this bit when the MCM0 bit is 0 becomes the main clock.

5.4 System Clock Oscillator

5.4.1 X1 oscillator

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 also be input. In this 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

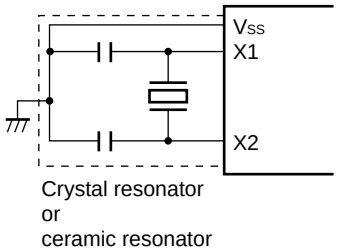
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not used as input port pins, either, see **Table 2-3 Connection of Unused Pins**.

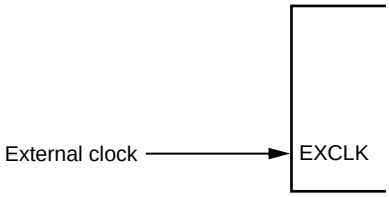
Figure 5-14 shows an example of the external circuit of the X1 oscillator.

Figure 5-14. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Cautions are listed on the next page.

5.4.2 XT1 oscillator

The XT1 oscillator oscillates with a crystal resonator (32.768 kHz (TYP.)) connected to the XT1 and XT2 pins.

To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

An external clock can also be input. In this case, input the clock signal to the EXCLKS pin.

To use the XT1 oscillator, set bits 5 and 4 (EXCLKS, OSCSELS) of the clock operation mode control register (CMC) as follows.

- Crystal or ceramic oscillation: EXCLKS, OSCSELS = 0, 1
- External clock input: EXCLKS, OSCSELS = 1, 1

When the XT1 oscillator is not used, set the input port mode (EXCLKS, OSCSELS = 0, 0).

When the pins are not used as input port pins, either, see **Table 2-3 Connection of Unused Pins**.

Figure 5-15 shows an example of the external circuit of the XT1 oscillator.

Figure 5-15. Example of External Circuit of XT1 Oscillator

(a) Crystal oscillation **(b) External clock**

Caution **When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-14 and 5-15 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.**

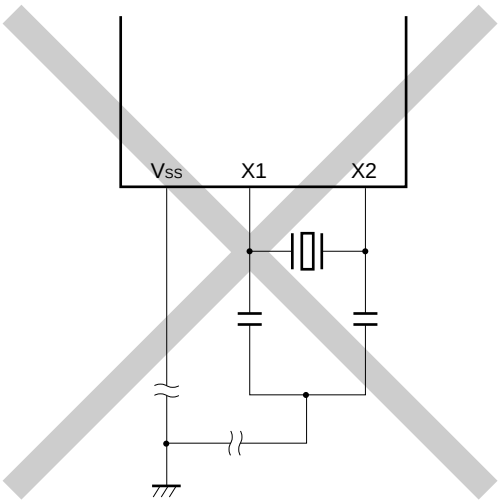
The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **Before using the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators described in 5.7 Resonator and Oscillator Constants.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as V_{SS} as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

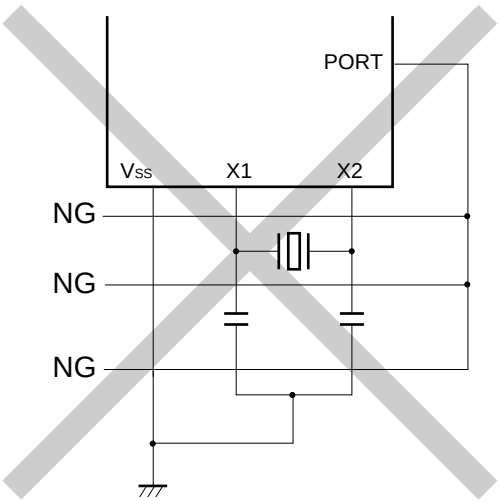
Figure 5-16 shows examples of incorrect resonator connection.

Figure 5-16. Examples of Incorrect Resonator Connection (1/2)

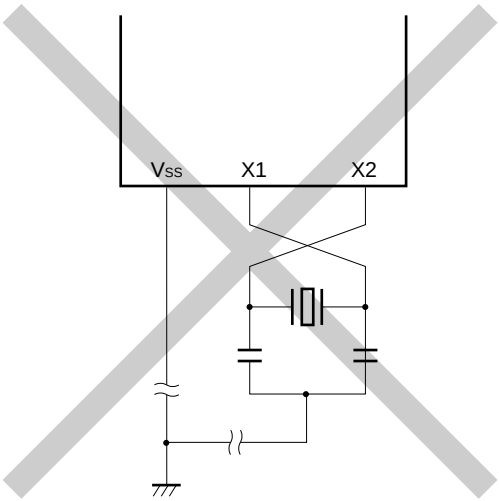
(a) Too long wiring



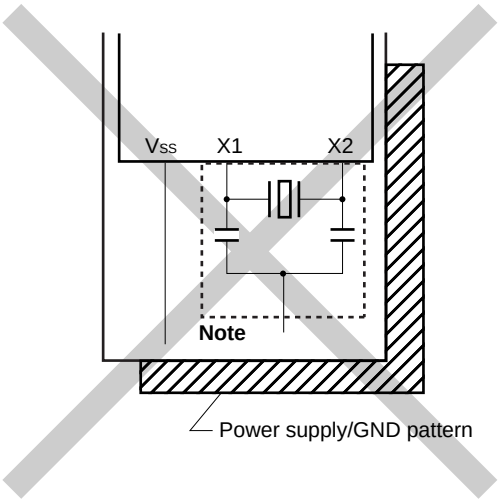
(b) Crossed signal line



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



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

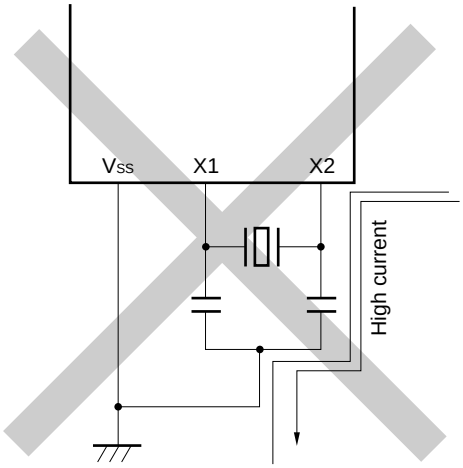


Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

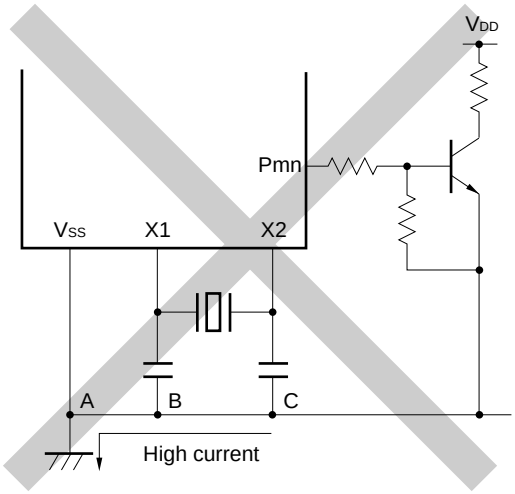
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5-16. Examples of Incorrect Resonator Connection (2/2)

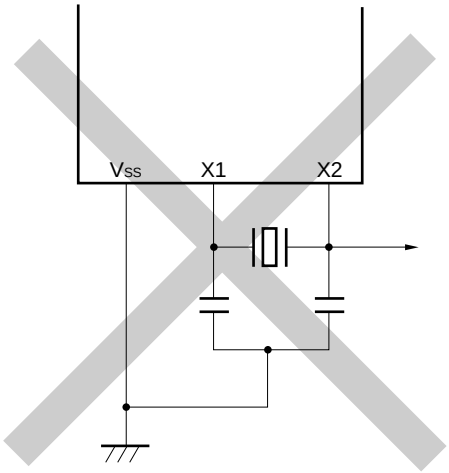
(e) Wiring near high alternating current



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



(g) Signals are fetched



Caution When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.

Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

5.4.3 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/G1C. Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC). The high-speed on-chip oscillator automatically starts oscillating after reset release.

5.4.4 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/G1C.

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

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC), or both are set to 1.

Unless the watchdog timer is stopped and WUTMMCK0 is a value other than zero, oscillation of the low-speed on-chip oscillator continues. Note that only when the watchdog timer is operating and the WUTMMCK0 bit is 0, oscillation of the low-speed on-chip oscillator will stop while the WDSTBYON bit is 0 and operation is in the HALT, STOP, or SNOOZE mode. While the watchdog timer operates, the low-speed on-chip oscillator clock does not stop even if the program freezes.

5.4.5 PLL (Phase Locked Loop)

The PLL circuit is incorporated in the RL78/G1C.

The PLL can be used to multiply the high-speed system clock.

Operation of the PLL circuit can be controlled by using bit 0 (DSCON) of the PLL control register (DSCCTL).

Cautions 1. When switching from PLL mode to the internal high-speed oscillation clock and the high-speed system clock, stop the function (USB function controller) that provides the PLL output clock (f_{PLL}).

2. Do not set the DSCON bit to 1 to start the PLL operating while the subsystem clock is the operating clock for the CPU.

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**).

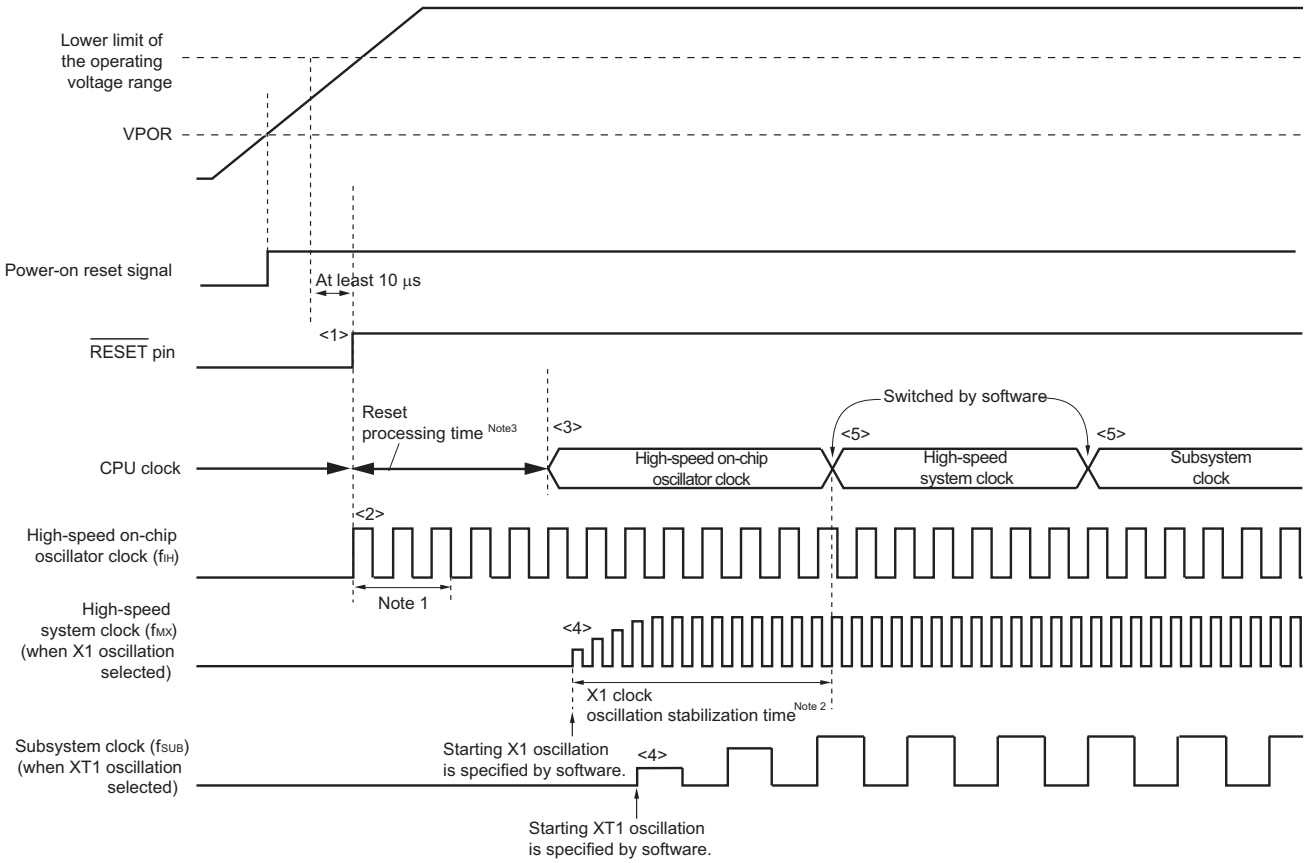
- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - High-speed on-chip oscillator clock f_{IH} **Note** (= f_{HOCO}/n ; n = 2, 4, 8)
  - PLL clock f_{IH} **Note** (= f_{PLL}/n ; n = 2, 4, 8)
- Subsystem clock f_{SUB}
  - XT1 clock f_{XT}
  - External subsystem clock f_{EXS}
- Low-speed on-chip oscillator clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

Note f_{IH} is the main system clock source frequency divided by 2, 4, or 8 when the high-speed on-chip oscillator clock or the PLL clock is selected (24 MHz max.).

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/G1C.

When the power supply voltage is turned on, the clock generator operation is shown in **Figure 5-17**.

Figure 5-17. 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 power-on-reset (POR) circuit. Note that the reset state is maintained after a reset by the voltage detector or an external reset until the voltage reaches the range of operating voltage described in **30.4 AC Characteristics** and **31.4 AC Characteristics** (the above figure is an example when the external reset is in use).

<2> When the reset is released, 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 a reset processing have been performed after reset release.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **5.6.2 Example of setting X1 oscillation clock** and **5.6.3 Example of setting XT1 oscillation clock**).

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **5.6.2 Example of setting X1 oscillation clock** and **5.6.3 Example of setting XT1 oscillation clock**).

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 the reset processing time, see **CHAPTER 21 POWER-ON-RESET CIRCUIT.**

Caution It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.

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 from 48 and 24 MHz by using FRQSEL0 to FRQSEL4 of the option byte (000C2H).

[Option byte setting]

Address: 000C2H

Option byte (000C2H)	7	6	5	4	3	2	1	0
	CMODE1	CMODE0		FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0
	1	1	1	1	0	0	0	0/1

CMODE1	CMODE0	Setting of flash operation mode
1	1	HS (high speed main) mode V_{DD} = 2.4 V to 5.5 V @ 1 MHz to 16 MHz V_{DD} = 2.7 V to 5.5 V @ 1 MHz to 24 MHz
Other than above		Setting prohibited

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator	
					f_{HOCO}	f_{IH}
1	0	0	0	0	48 MHz	24 MHz
1	0	0	0	1	24 MHz	12 MHz
Other than above					Setting prohibited	

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

Address: F00A8H

	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of high-speed on-chip oscillator clock frequency
0	0	0	f_{HOCO} = 48 MHz, f_{IH} = 24/12/6 MHz
0	0	1	f_{HOCO} = 24 MHz, f_{IH} = 12/6/3 MHz
0	1	0	f_{HOCO} = 12 MHz, f_{IH} = 6/3 MHz
0	1	1	f_{HOCO} = 6 MHz, f_{IH} = 3 MHz
Other than aboves			Setting prohibited

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 oscillation 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 counter status register (OSTC). After the oscillation stabilizes, set the X1 oscillation 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 where 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	OSCSEL	EXCLKS	OSCSELS		AMPHS1	AMPHS0	AMPH
	0	1	0	0	0	0	0	0/1

AMPH bit: Set this bit to 0 if the X1 oscillation 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 102 μs is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTS						OSTS2	OSTS1	OSTS0
	0	0	0	0	0	0	1	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	XTSTOP						HIOSTOP
	0	1	0	0	0	0	0	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 102.4 μs is set based on a 10 MHz resonator.

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

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

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

Caution Set the HOCODIV register within the operable voltage range of the flash operation mode set in the option byte (000C2H) before and after the frequency change.

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 5.5 V
			1 to 24 MHz	2.7 to 5.5 V

5.6.3 Example of setting XT1 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 XT1 oscillation clock, set the oscillator and start oscillation by using the subsystem clock supply mode control register (OSMC), clock operation mode control register (CMC), and clock operation status control register (CSC), set the XT1 oscillation 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> To run only the real-time clock and 12-bit interval timer on the subsystem clock (ultra-low current consumption) when in the STOP mode or HALT mode during CPU operation on the subsystem clock, set the RTCLPC bit to 1.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC 0/1	 0	 0	WUTMMCK0 0	 0	 0	 0	 0

<2> Set (1) the OSCSELS bit of the CMC register to operate the XT1 oscillator.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	 0	AMPHS1 0/1	AMPHS0 0/1	AMPH 0

AMPHS0 and AMPHS1 bits: These bits are used to specify the oscillation mode of the XT1 oscillator.

<3> Clear (0) the XTSTOP bit of the CSC register to start oscillating the XT1 oscillator.

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

<4> Use the timer function or another function to wait for oscillation of the subsystem clock to stabilize by using software.

<5> Use the CSS bit of the CKC register to specify the XT1 oscillation clock as the CPU/peripheral hardware clock.

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

5.6.4 Example of setting PLL circuit

After setting the high-speed system clock (see **5.6.2 Example of setting X1 oscillation clock**), use the PLL control register (DSCCTL) to control the PLL circuit.

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

<1> Set the HIOSTOP bit in the CSC register to make the high-speed on-chip oscillator run.

	7	6	5	4	3	2	1	0
CSC	0/1	0/1	0	0	0	0	0	HIOSTOP 0 Note 1

<2> Set the DSFRDIV bit and DSCM bit in the DSCCTL register to set the PLL multiplication and division.

	7	6	5	4	3	2	1	0
DSCCTL	0	0	0	0	0	DSFRDIV 0/1	DSCM 0/1	DSCON 0

<3> Set the RDIV1, RDIV0 bits of the MCKC register to set the division of the system clock.

	7	6	5	4	3	2	1	0
MCKC	0	0	0	0	0	RDIV1 0/1	RDIV0 0/1	CKSELR 0

<4> After wait more than 1 μs, set (1) the DSCON bit of the DSCCTL register to operate the PLL circuit Note 2.

	7	6	5	4	3	2	1	0
DSCCTL	0	0	0	0	0	DSFRDIV 0/1	DSCM 0/1	DSCON 1

<5> Set (1) the CKSELR bit of the MCKC register to select PLL output for the system clock.

	7	6	5	4	3	2	1	0
MCKC	0	0	0	0	0	RDIV1 0/1	RDIV0 0/1	CKSELR 1

<6> Use software to set up a wait of 135 μs Note 3.

<7> Set the HIOSTOP bit in the CSC register to stop the high-speed on-chip oscillator. Note2

	7	6	5	4	3	2	1	0
CSC	0/1	0/1	0	0	0	0	0	HIOSTOP 0 Note 1

<8> When the PLL clock frequency divided by 2, 4, or 8 is selected as the main system clock (f_{MAIN}), set the MCM0 bit in the CKC register to select the source for deriving the main system clock as a signal with a frequency (f_{IH}) of up to 24 MHz.

	7	6	5	4	3	2	1	0
CKC	CLS 0/1	CSS 0/1	MCS 0	MCM0 0	0	0	0	0

Notes 1. No setting is required to change to the PLL while the CKSELR bit is 1.
When setting the CKSELR bit to 1, ensure that the high-speed on-chip oscillator is running.

2. After oscillation by the X1 oscillator clock has become stable, allow at least 1 μs to elapse before starting the PLL. When restarting the PLL after it has been stopped, wait for at least 4 μs before using it in operations.

3. Wait for 40 μs for oscillation by the oscillator clock to become stabled if the HIOSTOP bit is not set to 0.

5.6.5 CPU clock status transition diagram

Figure 5-18 shows the CPU clock status transition diagram of this product.

Figure 5-18. CPU Clock Status Transition Diagram

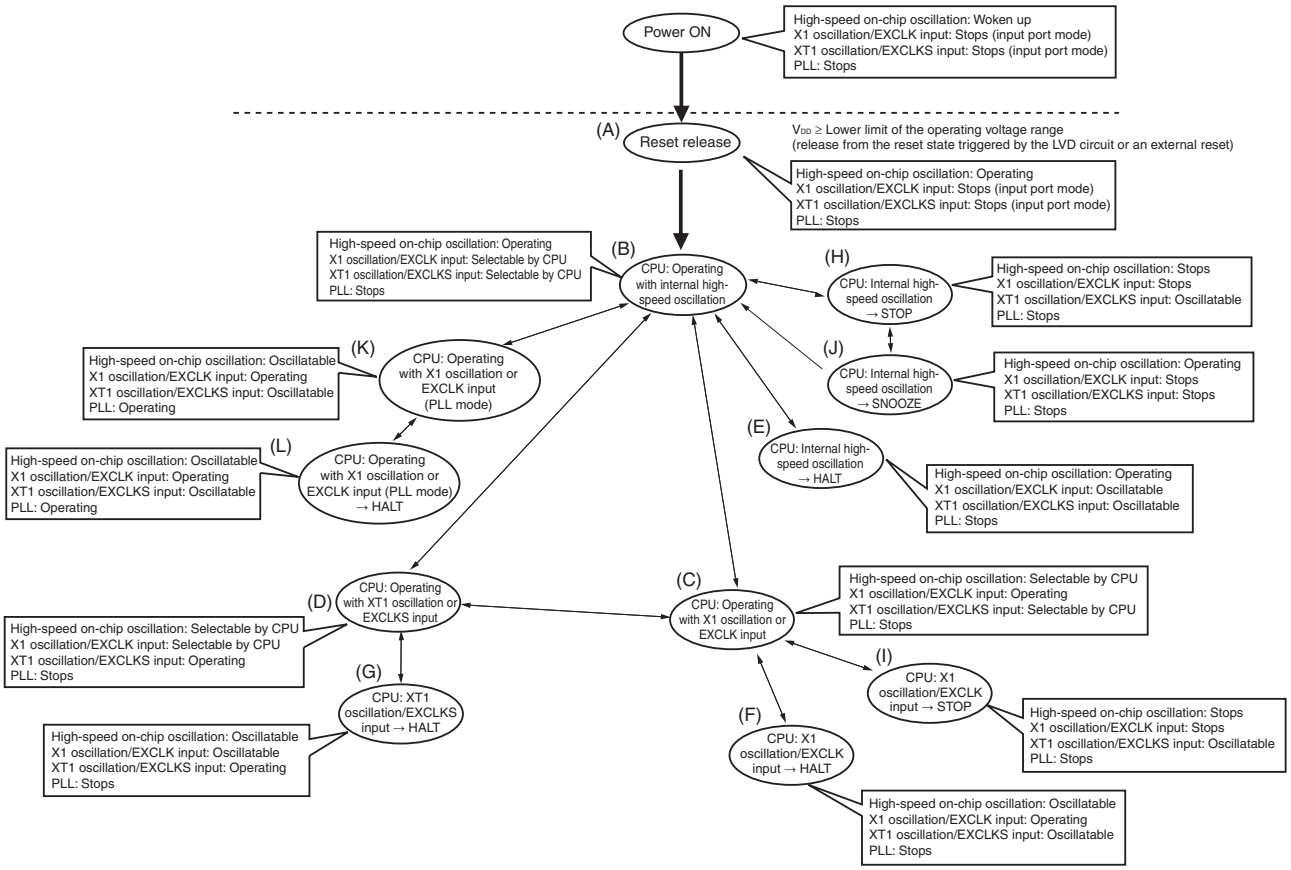


Table 5-4 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (1/7)

(1) CPU operating with high-speed on-chip oscillator clock (B) after reset release (A)

Status Transition	SFR Register Setting
(A) → (B)	SFR registers do not have to be set (default status after reset release).

(2) CPU operating with high-speed system clock (C) after reset release (A)

(The CPU operates with the high-speed on-chip oscillator clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note 1			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A) → (B) → (C) (X1 clock: 1 MHz ≤ fx ≤ 10 MHz)	0	1	0	**Note 2**	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	**Note 2**	0	Must be checked	1
(A) → (B) → (C) (external main clock)	1	1	×	**Note 2**	0	Must not be checked	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 30 or CHAPTER 31 ELECTRICAL SPECIFICATIONS).

(3) CPU operating with subsystem clock (D) after reset release (A)

(The CPU operates with the high-speed on-chip oscillator clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note				CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS	OSCSELS	AMPHS1	AMPHS0	XTSTOP		CSS
(A) → (B) → (D) (XT1 clock)	0	1	0/1	0/1	0	Necessary	1
(A) → (B) → (D) (external sub clock)	1	1	×	×	0	Necessary	1

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

Remarks 1. ×: don't care

2. (A) to (L) in Table 5-4 correspond to (A) to (L) in Figure 5-18.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (2/7)

(4) CPU clock changing from high-speed on-chip oscillator clock (B) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note 1]			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(B) → (C) (X1 clock: 1 MHz ≤ fx ≤ 10 MHz)	0	1	0	**Note 2**	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	**Note 2**	0	Must be checked	1
(B) → (C) (external main clock)	1	1	×	**Note 2**	0	Must not be checked	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

Notes 1. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release. This setting is not necessary if it has already been set.

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 30 or CHAPTER 31 ELECTRICAL SPECIFICATIONS).

(5) CPU clock changing from high-speed on-chip oscillator clock (B) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note]			CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS	OSCSELS	AMPHS1,0	XTSTOP		CSS
(B) → (D) (XT1 clock)	0	1	00: Low power consumption oscillation 01: Normal oscillation 10: Ultra-low power consumption oscillation	0	Necessary	1
(B) → (D) (external sub clock)	1	1	×	0	Necessary	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the subsystem clock

Note The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release. This setting is not necessary if it has already been set.

Remarks 1. ×: don't care

2. (A) to (L) in Table 5-4 correspond to (A) to (L) in Figure 5-18.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (3/7)

(6) CPU clock changing from high-speed system clock (C) to high-speed on-chip oscillator clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		MCM0
(C) → (B)	0	18 μs to 135 μs	0

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

Remark The oscillation accuracy stabilization time changes according to the temperature conditions and the period over which the HIOSTOP bit is 0.

(7) CPU clock changing from high-speed system clock (C) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Waiting for Oscillation Stabilization	CKC Register
	XTSTOP		CSS
(C) → (D)	0	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

(8) CPU clock changing from subsystem clock (D) to high-speed on-chip oscillator clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		CSS
(D) → (B)	0	18 μs to 135 μs	0

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

Remarks 1. (A) to (L) in Table 5-4 correspond to (A) to (L) in Figure 5-18.

2. The oscillation accuracy stabilization time changes according to the temperature conditions and the period over which the HIOSTOP bit is 0.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (4/7)

(9) CPU clock changing from subsystem clock (D) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	OSTS Register	CSC Register	OSTC Register	CKC Register
		MSTOP		CSS
(D) → (C) (X1 clock: 1 MHz ≤ f_X ≤ 10 MHz)	**Note**	0	Must be checked	0
(D) → (C) (X1 clock: 10 MHz < f_X ≤ 20 MHz)	**Note**	0	Must be checked	0
(D) → (C) (external main clock)	**Note**	0	Must not be checked	0

Unnecessary if the CPU is operating with the high-speed system clock

Note 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 30 or CHAPTER 31 ELECTRICAL SPECIFICATIONS).

Remark (A) to (L) in Table 5-4 correspond to (A) to (L) in Figure 5-18.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (5/7)

(10) • CPU clock changing from high-speed on-chip oscillator clock (B) to high-speed system clock (PLL mode) (K)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note 1]			OSTS Register	CSC Register	OSTC Register	DSCCTL Register		MCKC Register		Waiting for Oscillation Stabilization	DSCCTL Register	Waiting for Oscillation Stabilization	MCKC Register
	EXCLKS	OSCSELS	AMPH		MSTOP		DSFRDIV	DSCM	RDIV1	RDIV0		DSCON		CKSELR
(B) → (K) (divided by 2)	0/1	1	0/1	**Note 2**	0	Must be checked	0/1	0/1	0	0	1 μs	1	40 μs	1
(B) → (K) (divided by 4)	0/1	1	0/1	**Note 2**	0	Must be checked	0/1	0/1	0	1		1		1
(B) → (K) (divided by 8)	0/1	1	0/1	**Note 2**	0	Must be checked	0/1	0/1	1	0		1		1

Notes 1. Writing to the clock operating mode control register (CMC) can only proceed once and must be by an 8-bit memory manipulation instruction after release from the reset state.

2. Set the oscillation stabilization time in the oscillation stabilization time select register (OSTS) as follows.

- Desired oscillation stabilization time setting of the oscillation stabilization time counter status register (OSTC) ≤ Oscillation stabilization time set in the OSTS register

Caution Completion of clock switching after the CKSELR bit has been set to 1 requires up to 2 clock cycles when the FRQSEL4 bit is 1, and up to 10 clock cycles when the FRQSEL4 bit is 0. Until the clock switching is completed, do not stop the high-speed on-chip oscillator.

Remark (A) to (L) in Table 5-4 correspond to (A) to (L) in Figure 5-18.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (6/7)

(11) • CPU clock changing from high-speed system clock (PLL mode) (K) to high-speed on-chip oscillator clock (B)

(Setting sequence of SFR registers) →

Status Transition \ Setting Flag of SFR Register	CSC Register	Waiting for Oscillation Stabilization	MCKC Register	Waiting for Oscillation Stabilization	DSCCTL Register
	HIOSTOP		CKSELR		DSCON
(K) → (B) FRQSEL4 = 0	0	18 to 65 μs	0	256 clocks	0
(K) → (B) FRQSEL4 = 1		18 to 135 μs		16 clocks	

(12) • HALT mode (E) set while CPU is operating with high-speed on-chip oscillator clock (B)
• HALT mode (F) set while CPU is operating with high-speed system clock (C)
• HALT mode (G) set while CPU is operating with subsystem clock (D)
• HALT mode (L) set while CPU is operating with high-speed system clock (PLL mode) (K)

Status Transition	Setting
(B) → (E) (C) → (F) (D) → (G) (K) → (L)	Executing HALT instruction

(13) • STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

Status Transition		Setting		
(B) → (H)		Stopping peripheral functions that cannot operate in STOP mode	–	Executing STOP instruction
(C) → (I)	In X1 oscillation		Sets the OSTS register	
	External main system clock		–	

Remark (A) to (L) in Table 5-4 correspond to (A) to (L) in Figure 5-18.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (7/7)

(14) CPU changing from STOP mode (H) to SNOOZE mode (J)

For details about the setting for switching from the STOP mode to the SNOOZE mode, see **11.8 SNOOZE Mode Function**, and **12.5.7 SNOOZE mode function**.

(15) • STOP mode (I) set while CPU is operating with high-speed system clock (PLL mode) (K)

Switch to high-speed on-chip oscillator clock operation from PLL mode, stop the PLL (DSCON = 0), and then execute the STOP instruction.

Remark (A) to (L) in Table 5-4 correspond to (A) to (L) in Figure 5-18.

5.6.6 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 5-5. Changing CPU Clock (1/3)

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	After checking that the CPU clock is switched to the clock after change, operating current can be reduced by stopping high-speed on-chip oscillator (HIOSTOP = 1).
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	PLL clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time Or enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0 Oscillation of PLL • DSCON = 1	
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	After checking that the CPU clock is switched to the clock after change, X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	After checking that the CPU clock is switched to the clock after change, X1 oscillation can be stopped (MSTOP = 1).
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	After checking that the CPU clock is switched to the clock after change, X1 oscillation can be stopped (MSTOP = 1).
	PLL clock	Oscillation of PLL • DSCON = 1 Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	–

Table 5-5. Changing CPU Clock (2/3)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
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).
	X1 clock	Transition not possible	–
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1).
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	External main system clock input can be disabled (MSTOP = 1).
	PLL clock	Oscillation of PLL • DSCON = 1 Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • The oscillation accuracy stabilization time has elapsed	–
XT1 clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock • HIOSTOP = 0, MCS = 0	XT1 oscillation can be stopped (XTSTOP = 1)
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	External subsystem clock	Transition not possible	–
	PLL clock	Transition not possible	–

Table 5-5. Changing CPU Clock (3/3)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
External subsystem clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as main system clock • HIOSTOP = 0, MCS = 0	External subsystem clock input can be disabled (XTSTOP = 1).
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	XT1 clock	Transition not possible	–
	PLL clock	Transition not possible	–
PLL clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator • HIOSTOP = 0	Operating current can be reduced by stopping PLL (DSCON = 0).
	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	XT1 clock	Transition not possible	–
	External subsystem clock	Transition not possible	–

5.6.7 Time required for switchover of CPU clock and system clock

By setting bits 4 and 6 (MCM0, CSS) of the system clock control register (CKC), the CPU clock can be switched (between the main system clock and the subsystem clock), and main system clock can be switched (between the high-speed on-chip oscillator clock and the high-speed system clock).

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 5-6 to Table 5-8).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 7 (CLS) of the CKC register. 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 clock is also switched.

Table 5-6. Maximum Time Required for System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{IH}	←→	f_{MX}	See Table 5-7.
f_{MAIN}	←→	f_{SUB}	See Table 5-8.

Table 5-7. 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 ($f_{MAIN} = f_{IH}$)	$f_{MX} \geq f_{IH}$		2 clocks
	$f_{MX} < f_{IH}$		$2f_{IH}/f_{MX}$ clocks
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{IH}$	$2f_{MX}/f_{IH}$ clocks	
	$f_{MX} < f_{IH}$	2 clocks	

Table 5-8. Maximum Number of Clocks Required for $f_{MAIN} \leftrightarrow f_{SUB}$

Set Value Before Switchover	Set Value After Switchover	
CSS	CSS	
	0 ($f_{CLK} = f_{MAIN}$)	1 ($f_{CLK} = f_{SUB}$)
0 ($f_{CLK} = f_{MAIN}$)		$1 + 2f_{MAIN}/f_{SUB}$ clocks
1 ($f_{CLK} = f_{SUB}$)	3 clocks	

Remarks 1. The number of clocks listed in Table 5-7 and Table 5-8 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in Table 5-7 and Table 5-8 by removing the decimal portion.

Example When switching the main system clock from the high-speed on-chip oscillator clock (when 12 MHz selected) to the high-speed system clock (@ oscillation with f_{IH} = 12 MHz, f_{MX} = 10 MHz)

$1 + f_{IH}/f_{MX} = 1 + 12/10 = 1 + 1.2 = 2.2 \rightarrow 2$ clocks

5.6.8 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 5-9. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
High-speed on-chip oscillator clock	MCS = 1 or CLS = 1 (The CPU is operating on a clock other than the high-speed on-chip oscillator clock.)	HIOSTOP = 1
X1 clock External main system clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
XT1 clock External subsystem clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock.)	XTSTOP = 1

5.7 Resonator and Oscillator Constants

For the resonators for which operation has been verified and their oscillator constants, see the target product page on the Renesas Web site.

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 5-19. External Oscillation Circuit Example

(1) X1 oscillation:

As of March, 2013 (1/2)

Manufacturer	Resonator	Part Number[Note 3]	SMD/ Lead	Frequency (MHz)	Flash operation mode[Note 1]	Recommended Circuit Constants[Note 2] (reference)			Oscillation Voltage Range (V)	
						C1 (pF)	C2 (pF)	Rd (kΩ)	MIN.	MAX.
Murata Manufacturing Co., Ltd.	Ceramic resonator	CSTCC2M00G56-R0	SMD	2	LV	(47)	(47)	0	1.6	5.5
		CSTCR4M00G55-R0	SMD	4		(39)	(39)	0		
		CSTLS4M00G53-B0	Lead			(15)	(15)	0		
		CSTCC2M00G56-R0	SMD	2	LS	(47)	(47)	0	1.8	5.5
		CSTCR4M00G55-R0	SMD	4		(39)	(39)	0		
		CSTLS4M00G53-B0	Lead			(15)	(15)	0		
		CSTCR4M19G55-R0	SMD	4.194		(39)	(39)	0		
		CSTLS4M19G53-B0	Lead			(15)	(15)	0		
		CSTCR4M91G53-R0	SMD	4.915		(15)	(15)	0		
		CSTLS4M91G53-B0	Lead			(15)	(15)	0		
		CSTCR5M00G53-R0	SMD	5		(15)	(15)	0		
		CSTLS5M00G53-B0	Lead			(15)	(15)	0		
		CSTCR6M00G53-R0	SMD	6		(15)	(15)	0		
		CSTLS6M00G53-B0	Lead			(15)	(15)	0		
		CSTCE8M00G52-R0	SMD	8		(10)	(10)	0		
		CSTLS8M00G53-B0	Lead			(15)	(15)	0		
		CSTCE8M38G52-R0	SMD	8.388	HS	(10)	(10)	0	2.4	5.5
		CSTLS8M38G53-B0	Lead			(15)	(15)	0		
		CSTCE10M0G52-R0	SMD	10		(10)	(10)	0		
		CSTLS10M0G53-B0	Lead			(15)	(15)	0		
		CSTCE12M0G52-R0	SMD	12		(10)	(10)	0		
		CSTCE16M0V53-R0	SMD	16		(15)	(15)	0		
		CSTLS16M0X51-B0	Lead			(5)	(5)	0		
		CSTCE20M0V51-R0	SMD	20		(5)	(5)	0	2.7	5.5
		CSTLS20M0X51-B0	Lead			(5)	(5)	0		

Notes 1. Set the flash operation mode by using CMODE1 and CMODE0 bits of the option byte (000C2H).

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

3. Products supporting 105°C operation have different part numbers. For details, contact Murata Co., Ltd (http://www.murata.co.jp).

Remark Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: $2.7\ V \le V_{DD} \le 5.5\ V$@1 MHz to 24 MHz

$2.4\ V \le V_{DD} \le 5.5\ V$@1 MHz to 16 MHz

As of March, 2013 (2/2)

Manufacturer	Resonator	Part Number	SMD/ Lead	Frequency (MHz)	Flash operation mode[Note 1]	Recommended Circuit Constants[Note 2] (reference)			Oscillation Voltage Range (V)	
						C1 (pF)	C2 (pF)	Rd (kΩ)	MIN.	MAX.
Nihon Dempa Kogyo Co., Ltd.	Crystal resonator	NX8045GB[Note 3]	SMD	8	**Note 3**					
		NX5032GA[Note 3]	SMD	16						
		NX3225HA[Note 3]	SMD	20						
Kyocera Crystal Device Co., Ltd.	Crystal resonator	CX8045GB04000D0PPTZ1[Note 4]	SMD	4	LV	12	12	0	1.6	5.5
					LS				1.6	5.5
		CX8045GB04915D0PPTZ1[Note 4]	SMD	4.915	LS	12	12	0	1.8	5.5
		CX8045GB08000D0PPTZ1[Note4]	SMD	8	LS	12	12	0	1.8	5.5
		CX8045GB10000D0PPTZ1[Note 4]	SMD	10	HS	12	12	0	2.4	5.5
		CX3225GB12000B0PPTZ1[Note 4]	SMD	12	HS	5	5	0	2.4	5.5
		CX3225GB16000B0PPTZ1[Note 4]	SMD	16	HS	5	5	0	2.4	5.5
		CX3225SB20000B0PPTZ1[Note 4]	SMD	20	HS	5	5	0	2.7	5.5
RIVER ELETEC CORPORATION	Crystal resonator	FCX-03-8.000MHZ-J21140[Note 5]	SMD	8	HS	3	3	0	2.4	5.5
		FCX-04C-10.000MHZ-J21139[Note 5]	SMD	10	HS	4	4	0	2.4	5.5
		FCX-05-12.000MHZ-J21138[Note 5]	SMD	12	HS	6	6	0	2.4	5.5
		FCX-06-16.000MHZ-J21137[Note 5]	SMD	16	HS	4	4	0	2.4	5.5

Notes 1. Set the flash operation mode by using CMODE1 and CMODE0 bits of the option byte (000C2H).

2. This resonator supports operation at up to 85°C. Contact crystal oscillator manufacturers with regard to products supporting operation at up to 105°C.

3. When using this resonator, for details about the matching, contact Nihon Dempa Kogyo Co., Ltd (http://www.ndk.com/en).

4. When using this resonator, for details about the matching, contact Kyocera Crystal Device Co., Ltd. (http://www.kyocera-crystal.jp, http://www.kyocera.co.jp).

5. When using this resonator, for details about the matching, contact RIVER ELETEC CORPORATION (http://www.river-ele.co.jp).

Remark Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: $2.7\ V \le V_{DD} \le 5.5\ V$@1 MHz to 24 MHz

$2.4\ V \le V_{DD} \le 5.5\ V$@1 MHz to 16 MHz

(2) XT1 oscillation: Crystal resonator

As of March, 2013

Manufacturer	Part Number[Note 2]	SMD/ Lead	Frequency (KHz)	Load Capacitance CL (pF)	XT1 oscillation mode[Note 1]	Recommended Circuit Constants			Oscillation Voltage Range	
						C3 (pF)	C4 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
Seiko Instruments Inc.	SSP-T7-F Note 3	SMD	32.768	7	Normal oscillation	11	11	0	1.6	5.5
	SSP-T7-FL Note 3			6		9	9	0		
				6	Low power consumption oscillation	9	9	0		
				4.4		6	5	0		
				4.4	Ultra-low power consumption oscillation	6	5	0		
				3.7		4	4	0		
	VT-200-FL Note 3	Lead		6	Normal oscillation	9	9	0		
				6	Low power consumption oscillation	9	9	0		
				4.4		6	5	0		
				4.4	Ultra-low power consumption oscillation	6	5	0		
				3.7		4	4	0		
Nihon Dempa Kogyo Co., Ltd.	NX3215SA Note 4	SMD	32.768	6	Normal oscillation	7	7	0	1.6	5.5
					Low power consumption oscillation					
					Ultra-low power consumption oscillation					
Kyocera Crystal Device Co., Ltd.	ST3215SB Note 5	SMD	32.768	7	Normal oscillation	10	10	0	1.6	5.5
					Low power consumption oscillation					
					Ultra-low power consumption oscillation					
RIVER ELETEC CORPORATION	TFX-02-32.768KHZ-J20986[Note 6]	SMD	32.768	9	Normal oscillation	12	10	0	1.6	5.5
					Low power consumption oscillation					
	TFX-03-32.768KHZ-J13375[Note 6]	SMD	32.768	7	Normal oscillation	12	10	0	1.6	5.5

Notes 1. Set the XT1 oscillation mode by using AMPHS0 and AMPHS1 bits of the clock operation mode control register (CMC).

2. This resonator supports operation at up to 85°C. Contact crystal oscillator manufacturers with regard to products supporting operation at up to 105°C.

3. When using this resonator, for details about the matching, contact Seiko Instruments Inc., Ltd (http://www.sii-crystal.com).

4. When using this resonator, for details about the matching, contact Nihon Dempa Kogyo Co., Ltd (http://www.ndk.com/en).

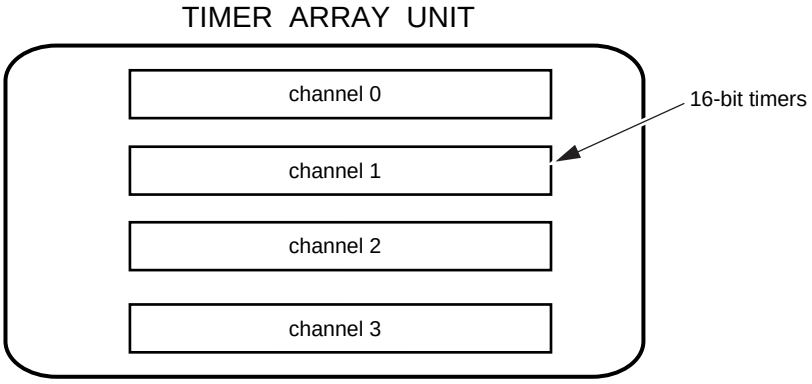
5. When using this resonator, for details about the matching, contact Kyocera Crystal Device Co., Ltd. (http://www.kyocera-crystal.jp, http://www.kyocera.co.jp).

6. When using this resonator, for details about the matching, contact RIVER ELETEC CORPORATION (http://www.river-ele.co.jp).

CHAPTER 6 TIMER ARRAY UNIT

The timer array unit has 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 to create a high-accuracy timer.



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 Note (→ 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) • PWM output(→ refer to 6.9.2) • Multiple PWM output(→ refer to 6.9.3)

Note Only channel 0.

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 (upper or lower 8-bit timer)/square wave output (lower 8-bit timer only)
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)

6.1 Functions of Timer Array Unit

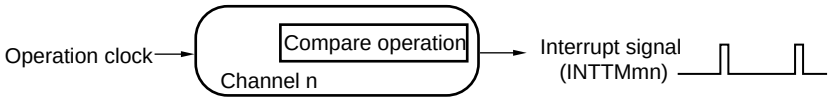
Timer array unit has the following functions.

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.

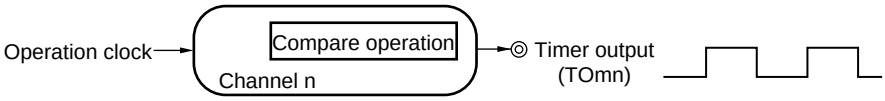
(1) Interval timer

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



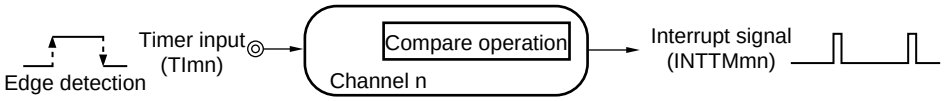
(2) Square wave output

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



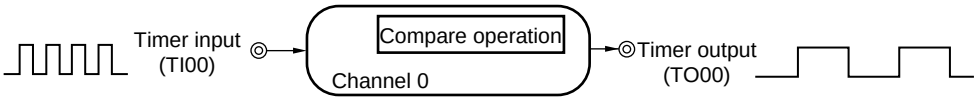
(3) External event counter

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 (TImn) has reached a specific value.



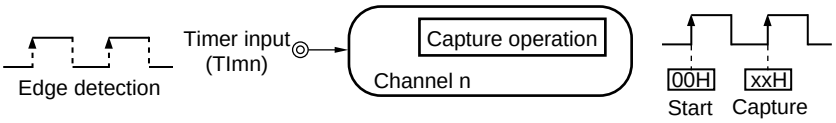
(4) Divider function (channel 0 only)

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



(5) Input pulse interval measurement

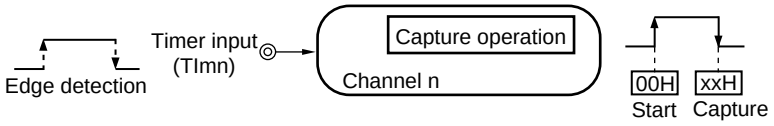
Counting is started by the valid edge of a pulse signal input to a timer input pin (TImn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



(**Remark** is listed on the next page.)

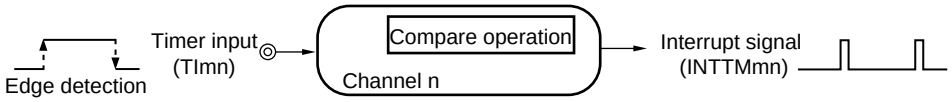
(6) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TImn), 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 (TImn), and an interrupt is generated after any delay period.



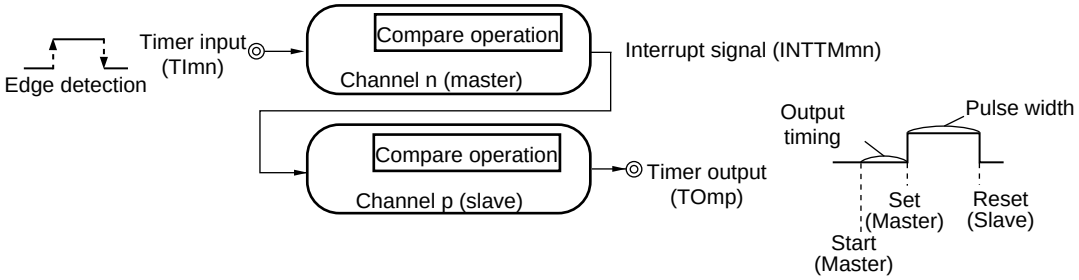
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.1.2 Simultaneous channel operation function

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

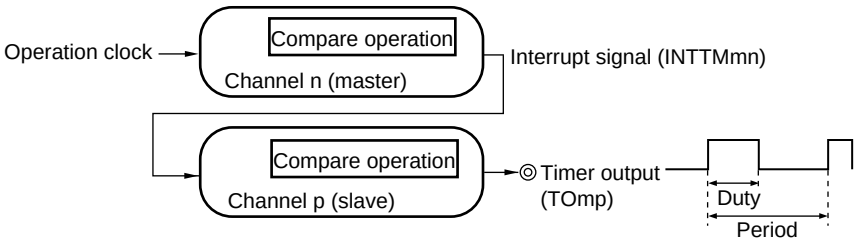
(1) One-shot pulse output

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



(2) PWM (Pulse Width Modulation) output

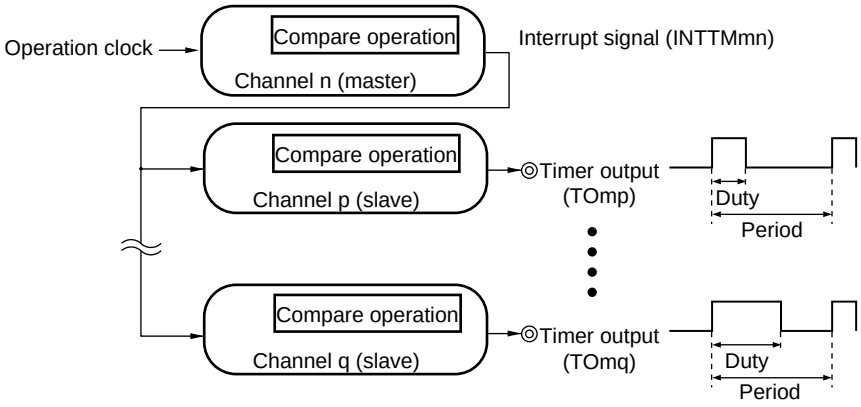
Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



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

(3) Multiple PWM (Pulse Width Modulation) output

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.



Caution For details about the rules of simultaneous channel operation function, see 6.4.1 Basic rules of simultaneous channel operation function.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3),
p, q: Slave channel number (n < p < q ≤ 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 channel in a configuration consisting of two 8-bit timer channels. This function can only be used for channels 1 and 3.

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 (channels 1 and 3 only).

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 count register mn (TCRmn)
Register	Timer data register mn (TDRmn)
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 m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSm) • Timer channel stop register m (TTm) • Timer input select register 0 (TIS0) • Timer output enable register m (TOEm) • Timer output register m (TOm) • Timer output level register m (TOLm) • Timer output mode register m (TOMm)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Noise filter enable register 1 (NFEN1) • Port mode control register (PMC0) • Port mode registers (PM0, PM1, PM3) • Port registers (P0, P1, P3)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Table 6-2. Timer I/O Pins provided in Each Product

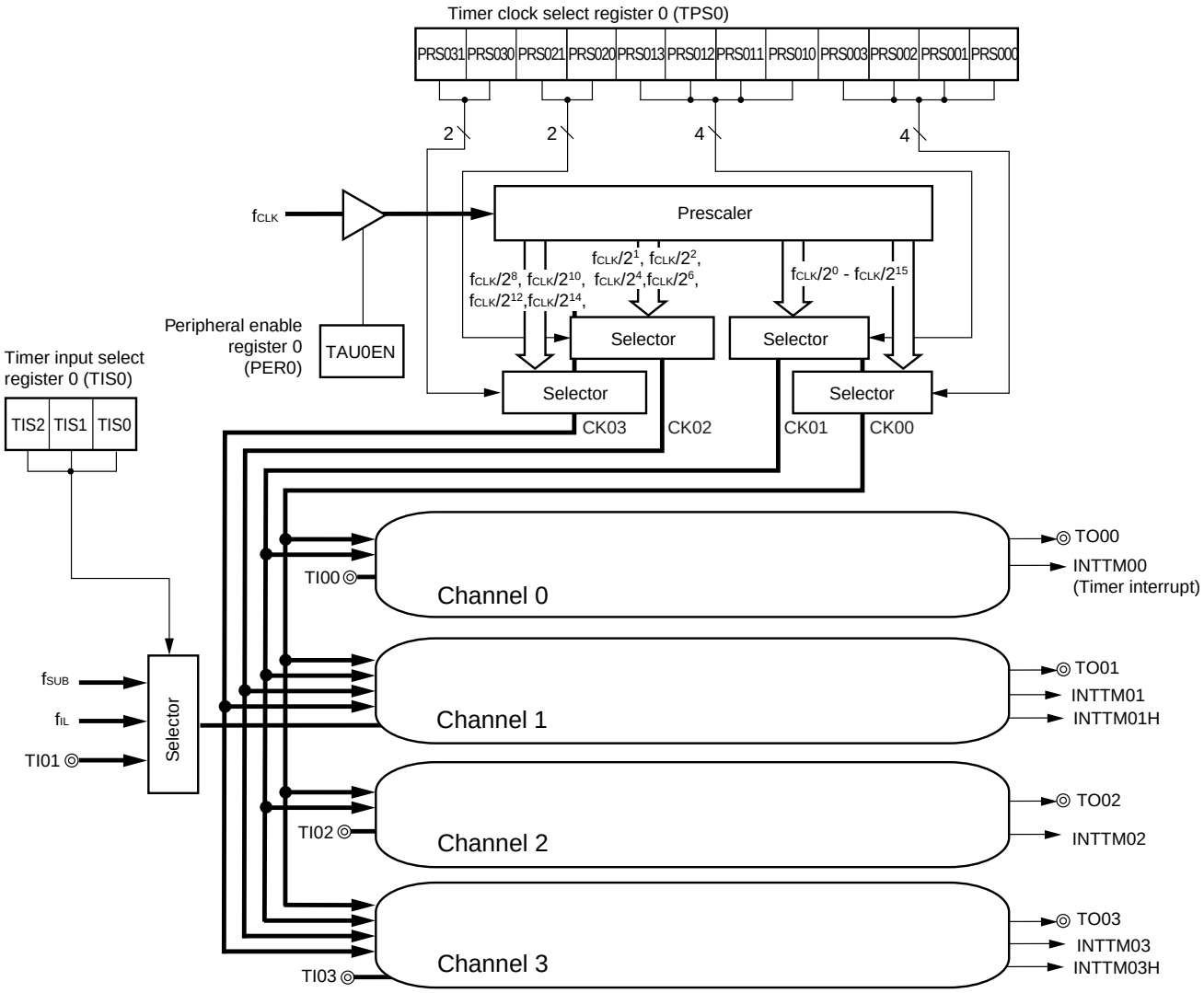
Timer array unit channels	I/O Pins of Each Product	
	48-pin	32-pin
Channel 0	P00/TI00, P01/TO00	
Channel 1	P16/TI01/TO01(P71)	P16/TI01/TO01(P51)
Channel 2	P17/TI02/TO02(P72)	P17/TI02/TO02(P50)
Channel 3	P31/TI03/TO03(P41)	P31/TI03/TO03(P30)

Remarks 1. When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

2. “(Pxx)” indicates an alternate port when the bit 0 of the peripheral I/O redirection register (PIOR) is set to “1”.

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

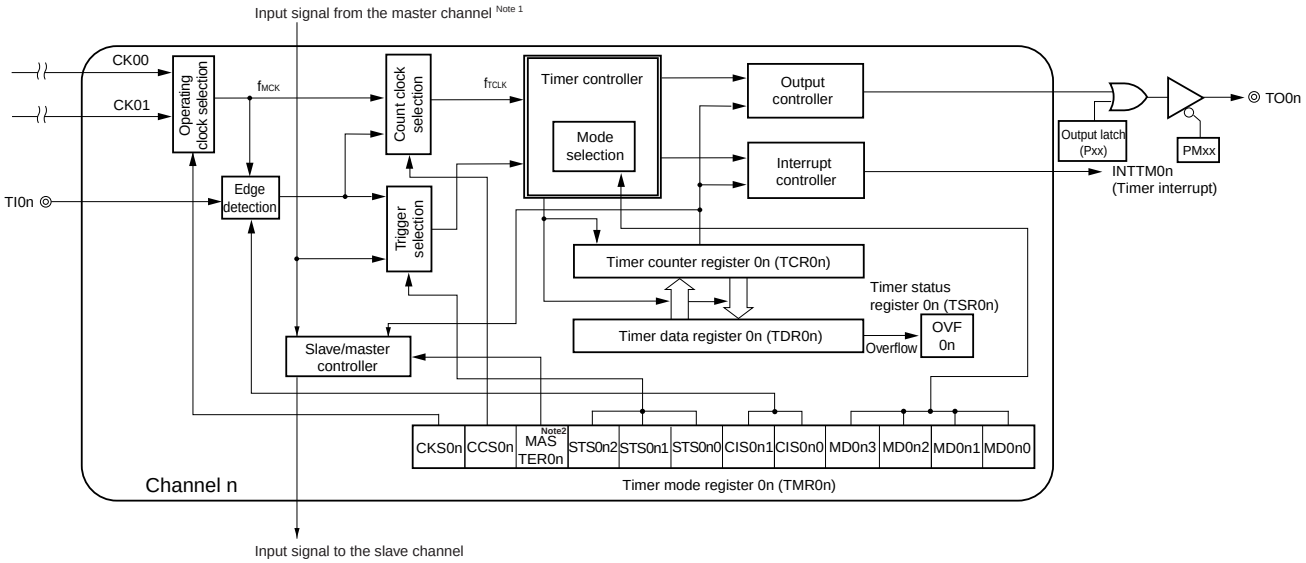
Figure 6-1. Entire Configuration of Timer Array Unit



Remark f_{SUB}: Subsystem clock frequency

f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 6-2. Internal Block Diagram of Channels 0 and 2 of Timer Array Unit



Note n = 2 only

Remark n = 2

Figure 6-3. Internal Block Diagram of Channel 1 of Timer Array Unit

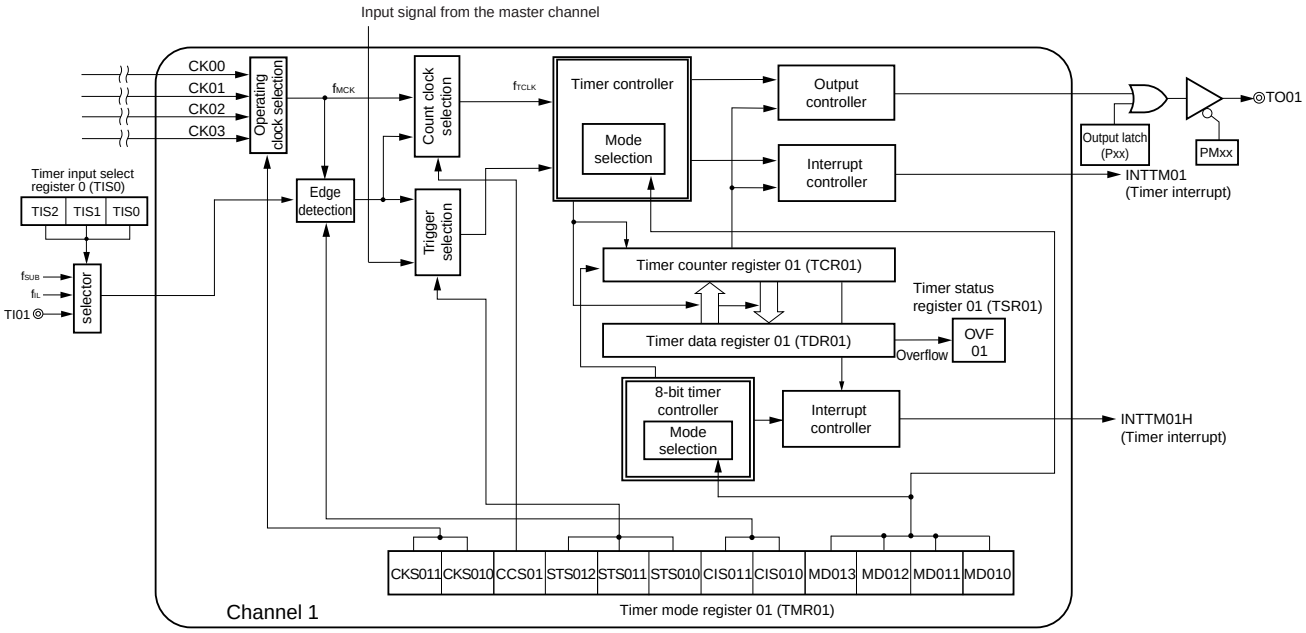
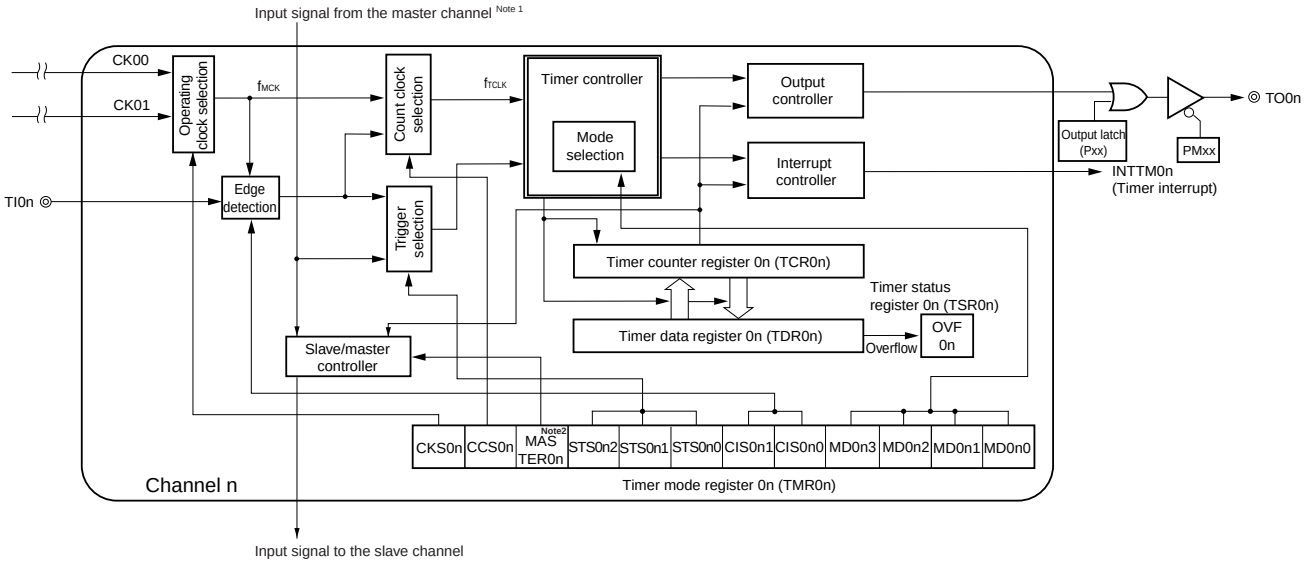


Figure 6-4. Internal Block Diagram of Channel 3 of Timer Array Unit



Remark n = 3

6.2.1 Timer count register mn (TCRmn)

The TCRmn register is a 16-bit read-only register and is used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock.

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn) (refer to **6.3.3 Timer mode register mn (TMRmn)**).

Figure 6-5. Format of Timer Count Register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F0186H, F0187H (TCR03) After reset: FFFFH R

	F0181H (TCR00)								F0180H (TCR00)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRmn																

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

The count value can be read by reading timer count register mn (TCRmn).

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 of the slave channel 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

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

Caution The count value is not captured to timer data register mn (TDRmn) even when the TCRmn register is read.

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

Table 6-3. Timer Count Register mn (TCRmn) Read Value in Various Operation Modes

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

Note This indicates the value read from the TCRmn register when channel n has stopped operating as a timer (TEmn = 0) and has been enabled to operate as a counter (TSmn = 1). The read value is held in the TCRmn register until the count operation starts.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.2.2 Timer data register mn (TDRmn)

This is a 16-bit register from which a capture function and a compare function can be selected.

The capture or compare function can be switched by selecting an operation mode by using the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn).

The value of the TDRmn register can be changed at any time.

This register can be read or written in 16-bit units.

In addition, for the TDRm1 and TDRm3 registers, while in the 8-bit timer mode (when the SPLITm1 and SPLITm3 bits of timer mode registers m1 and m3 (TMRm1, TMRm3) are 1), it is possible to read and write the data in 8-bit units, with TDRm1H and TDRm3H used as the higher 8 bits, and TDRm1L and TDRm3L used as the lower 8 bits.

Reset signal generation clears this register to 0000H.

Figure 6-6. Format of Timer Data Register mn (TDRmn) (n = 0, 2)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02) After reset: 0000H R/W

	FFF19H (TDR00)								FFF18H (TDR00)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDRmn																

Figure 6-7. Format of Timer Data Register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03) After reset: 0000H R/W

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

(i) When timer data register mn (TDRmn) is used as compare register

Counting down is started from the value set to the TDRmn register. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. The TDRmn register holds its value until it is rewritten.

Caution The TDRmn 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 mn (TDRmn) is used as capture register

The count value of timer count register mn (TCRmn) is captured to the TDRmn register when the capture trigger is input.

A valid edge of the TImn pin can be selected as the capture trigger. This selection is made by timer mode register mn (TMRmn).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer input select register 0 (TIS0)
- Timer output enable register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Noise filter enable register 1 (NFEN1)
- Port mode registers (PM0, PM1, PM3)
- Port registers (P0, P1, P3)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 6-8. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	<2>	1	<0>
PER0	RTCEN	0	ADCEN	IICA0EN	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.

Caution **When setting the timer array unit, be sure to set the following registers first while the TAUmEN bit is set to 1. If TAUmEN = 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 timer input select register 0 (TIS0), noise filter enable register 1 (NFEN1), port mode control register 0 (PMC0), port mode register 0, 1, 3 (PM0, PM1, PM3), port register 0, 1,3 (P0, P1, P3)).**

- **Timer clock select register m (TPSm)**
- **Timer mode register mn (TMRmn)**
- **Timer status register mn (TSRmn)**
- **Timer channel enable status register m (TEm)**
- **Timer channel start register m (TSm)**
- **Timer channel stop register m (TTm)**
- **Timer output enable register m (TOEm)**
- **Timer output register m (TOm)**
- **Timer output level register m (TOLm)**
- **Timer output mode register m (TOMm)**

6.3.2 Timer clock select register m (TPSm)

The TPSm register is a 16-bit register that is used to select two types or four types of operation clocks (CKm0, CKm1, CKm2, CKm3) that are commonly supplied to each channel. CKm0 is selected by using bits 3 to 0 of the TPSm register, and CKm1 is selected by using bits 7 to 4 of the TPSm register. In addition, only for channels 1 and 3, CKm2 and CKm3 can be also selected. CKm2 is selected by using bits 9 and 8 of the TPSm register, and CKm3 is selected by using bits 13 and 12 of the TPSm register.

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

If the PRSm00 to PRSm03 bits can be rewritten (n = 0 to 3):

All channels for which CKm0 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 0) are stopped (TEmn = 0).

If the PRSm10 to PRSm13 bits can be rewritten (n = 0 to 3):

All channels for which CKm1 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 1) are stopped (TEmn = 0).

If the PRSm20 and PRSm21 bits can be rewritten (n = 1, 3):

All channels for which CKm2 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 0) are stopped (TEmn = 0).

If the PRSm30 and PRSm31 bits can be rewritten (n = 1, 3):

All channels for which CKm3 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6-9. Format of Timer Clock Select register m (TPSm) (1/2)

Address: F01B6H, F01B7H (TPS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRS m31	PRS m30	0	0	PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operation clock (CKmk) Note (k = 0, 1)					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.62 kHz	39.1 kHz	78.1 kHz	156.2 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.76 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).

Cautions 1. Be sure to clear bits 15, 14, 11, 10 to "0".

2. If f_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (n = 0 or 1, m = 0 to 7), interrupt requests output from timer array units cannot be used.

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

2. Waveform of the clock to be selected in the TPSm register which becomes high level for one period of f_{CLK} from its rising edge. For details, see **6.5.1 Count clock (f_{TCLK}).**

Figure 6-9. Format of Timer Clock Select register m (TPSm) (2/2)

Address: F01B6H, F01B7H (TPS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRS m31	PRS m30	0	0	PRS m21	PRS m20	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS m21	PRS m20	Selection of operation clock (CKm2) Note					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	1	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
1	0	$f_{CLK}/2^4$	125 kHz	312.5 kHz	625 MHz	1.25 MHz	1.5 MHz
1	1	$f_{CLK}/2^6$	31.25 kHZ	78.1 kHz	156.2 kHz	312.5 kHz	375 kHZ

PRS m31	PRS m30	Selection of operation clock (CKm3) Note					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
0	1	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz	23.4 kHz
1	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.85 kHz
1	1	$f_{CLK}/2^{14}$	122 HZ	305 Hz	610 Hz	1.22 kHz	1.46 kHZ

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).
The timer array unit must also be stopped if the operating clock (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected.

Caution Be sure to clear bits 15, 14, 11, 10 to "0".

By using channels 1 and 3 in the 8-bit timer mode and specifying CKm2 or CKm3 as the operation clock, the interval times shown in Table 6-4 can be achieved by using the interval timer function.

Table 6-4. Interval Times Available for Operation Clock CKSm2 or CKSm3

Clock		Interval time (f_{CLK} = 24 MHz)			
		10 μs Note	100 μs Note	1 ms Note	10 ms Note
CKm2	$f_{CLK}/2$	√	–	–	–
	$f_{CLK}/2^2$	√	–	–	–
	$f_{CLK}/2^4$	√	√	–	–
	$f_{CLK}/2^6$	√	√	–	–
CKm3	$f_{CLK}/2^8$	–	√	√	–
	$f_{CLK}/2^{10}$	–	√	√	–
	$f_{CLK}/2^{12}$	–	–	√	√
	$f_{CLK}/2^{14}$	–	–	√	√

Note The margin is within 5 %.

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

2. For details of asignal of $f_{CLK}/2^r$ selected with the TPSm register, see **6.5.1 Count clock (f_{TCLK}).**

6.3.3 Timer mode register mn (TMRmn)

The TMRmn register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, 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 TMRmn register is prohibited when the register is in operation (when TEmn = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEmn = 1) (for details, see **6.7 Independent Channel Operation Function of Timer Array Unit** and **6.8 Simultaneous Channel Operation Function of Timer Array Unit**.

The TMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Caution The bits mounted depend on the channels in the bit 11 of TMRmn register.
TMRm2: MASTERmn bit (n = 2)
TMRm1, TMRm3: SPLITmn bit (n = 1, 3)
TMRm0: Fixed to 0

Figure 6-10. Format of Timer Mode Register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKS mn1	CKS mn0	0	CCS mn	0 Note	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CKS mn1	CKS mn0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CKm0 set by timer clock select register m (TPSm)
0	1	Operation clock CKm2 set by timer clock select register m (TPSm)
1	0	Operation clock CKm1 set by timer clock select register m (TPSm)
1	1	Operation clock CKm3 set by timer clock select register m (TPSm)
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 CCSmn bit. The operation clocks CKm2 and CKm3 can only be selected for channels 1 and 3.		

CCS mn	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
1	Valid edge of input signal input from the TImn pin In channel 1, valid edge of input signal selected by TIS0
Count clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Note Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Cautions 1. Be sure to clear bits 13, 5, and 4 to "0".

2. The timer array unit must be stopped (TTm = 000FH) 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 specified by using the CKSmn0 and CKSmn1 bits (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-10. Format of Timer Mode Register mn (TMRmn) (2/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

(Bit 11 of TMRmn (n = 2))

MASTERmn	Selection between using channel n independently or simultaneously with another channel(as a slave or master)
0	Operates in independent channel operation function or as slave channel in simultaneous channel operation function.
1	Operates as master channel in simultaneous channel operation function.
Only the channel 2 can be set as a master channel (MASTERmn = 1). Be sure to use channel 0 is fixed to 0 (Regardless of the bit setting, channel 0 operates as master, because it is the highest channel). Clear the MASTERmn bit to 0 for a channel that is used with the independent channel operation function.	

(Bit 11 of TMRmn (n = 1, 3))

SPLITmn	Selection of 8 or 16-bit timer operation for channels 1 and 3
0	Operates as 16-bit timer. (Operates in independent channel operation function or as slave channel in simultaneous channel operation function.)
1	Operates as 8-bit timer.

STSmn2	STSmn1	STSmn0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of the TImn pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of the TImn pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function).
Other than above			Setting prohibited

Note Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-10. Format of Timer Mode Register mn (TMRmn) (3/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKS mn1	CKS mn0	0	CCS mn	MASTERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKS mn1	CKS mn0	0	CCS mn	0 Note	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

CIS mn1	CIS mn0	Selection of TImn 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 STSmn2 to STSmn0 bits is other than 010B, set the CISmn1 to CISmn0 bits to 10B.		

MD mn3	MD mn2	MD mn1	MD mn0	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	1/0	Interval timer mode	Interval timer / Square wave output / Divider function / PWM output (master)	Counting down
0	1	0	1/0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	0	Event counter mode	External event counter	Counting down
1	0	0	1/0	One-count mode	Delay counter / One-shot pulse output / PWM output (slave)	Counting down
1	1	0	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Counting up
Other than above				Setting prohibited		
The operation of each mode varies depending on MDmn0 bit (see next table).						

Note Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-10. Format of Timer Mode Register mn (TMRmn) (4/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKS mn1	CKS mn0	0	CCS mn	MAST ERmn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKS mn1	CKS mn0	0	CCS mn	SPLIT mn	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKS mn1	CKS mn0	0	CCS mn	0 Note 1	STS mn2	STS mn1	STS mn0	CIS mn1	CIS mn0	0	0	MD mn3	MD mn2	MD mn1	MD mn0

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MD mn0	Setting of starting counting and interrupt
• Interval timer mode (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 (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode Note 2 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation Note 3. At that time, 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 interrupt is not generated.
Other than above		Setting prohibited

Notes 1. Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

2. In one-count mode, interrupt output (INTTMmn) when starting a count operation and TOmn output are not controlled.

3. If the start trigger (TSmn = 1) is issued during operation, the counter is initialaized, an interrupt is generated, and recounting is started (does not occur the interrupt request).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.4 Timer status register mn (TSRmn)

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

The TSRmn register is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). See Table 6-5 for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSRmn register can be set with an 8-bit memory manipulation instruction with TSRmnL.

Reset signal generation clears this register to 0000H.

Figure 6-11. Format of Timer Status Register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01A6H, F01A7H (TSR03) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRmn	0	0	0	0	0	0	0	0	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 m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Table 6-5. 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	–
	set	(Use prohibited)

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 m (TEm)

The TEm register is used to enable or stop the timer operation of each channel.

Each bit of the TEm register corresponds to each bit of the timer channel start register m (TSm) and the timer channel stop register m (TTm). When a bit of the TSm register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TTm register is set to 1, the corresponding bit of this register is cleared to 0.

The TEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TEmL.

Reset signal generation clears this register to 0000H.

Figure 6-12. Format of Timer Channel Enable Status register m (TEm)

Address: F01B0H, F01B1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	TEHm3	0	TEHm1	0	0	0	0	0	TEm3	TEm2	TEm1	TEm0

TEHm3	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEHm1	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEmn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TEm1 and TEm3 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.6 Timer channel start register m (TSm)

The TSm register is a trigger register that is used to initialize timer count register mn (TCRmn) and start the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is set to 1. The TSmn, TSHm1, TSHm3 bits are immediately cleared when operation is enabled (TEmn, TEHm1, TEHm3 = 1), because they are trigger bits.

The TSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSm register can be set with a 1-bit or 8-bit memory manipulation instruction with TSmL.

Reset signal generation clears this register to 0000H.

Figure 6-13. Format of Timer Channel Start register m (TSm)

Address: F01B2H, F01B3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	TSHm3	0	TSHm1	0	0	0	0	0	TSm3	TSm2	TSm1	TSm0

TSHm3	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm3 bit is set to 1 and the count operation becomes enabled. The TCRm3 register count operation start in the interval timer mode in the count operation enabled state (see Table 6-6 in **6.5.2 Start timing of counter**).

TSHm1	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm1 bit is set to 1 and the count operation becomes enabled. The TCRm1 register count operation start in the interval timer mode in the count operation enabled state (see Table 6-6 in **6.5.2 Start timing of counter**).

TSmn	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TEmn bit is set to 1 and the count operation becomes enabled. The TCRmn register count operation start in the count operation enabled state varies depending on each operation mode (see Table 6-6 in **6.5.2 Start timing of counter**). This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TSm1 and TSm3 when channel 1 or 3 is in the 8-bit timer mode.

Cautions 1. Be sure to clear bits 15 to 12, 10, 8 to 4 to "0"

2. When switching from a function that does not use TImn pin input to one that does, the following wait period is required from when timer mode register mn (TMRmn) is set until the TSmn (TSHm1, TSHm3) bit is set to 1.

When the TImn pin noise filter is enabled (TNFENmn = 1): Four cycles of the operation clock (f_{MCK})

When the TImn pin noise filter is disabled (TNFENmn = 0): Two cycles of the operation clock (f_{MCK})

Remarks 1. When the TSm register is read, 0 is always read.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.7 Timer channel stop register m (TTm)

The TTm register is a trigger register that is used to stop the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is cleared to 0. The TTmn, TTHm1, TTHm3 bits are immediately cleared when operation is stopped (TEmn, TTHm1, TTHm3 = 0), because they are trigger bits.

The TTm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TTm register can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

Figure 6-14. Format of Timer Channel Stop register m (TTm)

Address: F01B4H, F01B5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	TTHm3	0	TTHm1	0	0	0	0	0	TTm3	TTm2	TTm1	TTm0

TTHm3	Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	TEHm3 bit is cleared to 0 and the count operation is stopped.

TTHm1	Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	TEHm1 bit is cleared to 0 and the count operation is stopped.

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	TEmn bit is cleared to 0 and the count operation is stopped. This bit is the trigger to stop operation of the lower 8-bit timer for TTm1 and TTm3 when channel 1 or 3 is in the 8-bit timer mode.

Caution Be sure to clear bits 15 to 12, 10, 8 to 4 of the TTm register to "0".

Remarks 1. When the TTm register is read, 0 is always read.

2. m: Unit number (m = 0),n: Channel number (n = 0 to 3)

6.3.8 Timer input select register 0 (TIS0)

The TIS0 register is used to select the channel 1 timer input..

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-15. Format of Timer Input Select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	
0	1	0	
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

Caution High-level width, low-level width of timer input is selected, will require more than 1/f_{MCK} +10 ns. Therefore, when selecting f_{SUB} to f_{CLK} (CSS bit of CKC register = 1), can not TIS02 bit set to 1.

6.3.9 Timer output enable register m (TOEm)

The TOEm 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 TOmn bit of timer output register m (TOm) 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 (TOmn).

The TOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TOEmL.

Reset signal generation clears this register to 0000H.

Figure 6-16. Format of Timer Output Enable register m (TOEm)

Address: F01BAH, F01BBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEm	0	0	0	0	0	0	0	0	0	0	0	0	TOEm3	TOEm2	TOEm1	TOEm0

TOEmn	Timer output enable/disable of channel n
0	The TOmn operation stopped by count operation (timer channel output bit). Writing to the TOmn bit is enabled. The TOmn pin functions as data output, and it outputs the level set to the TOmn bit. The output level of the TOmn pin can be manipulated by software.
1	Enable output of timer. Reflected in the TOmn bit timer operation, to generate the output waveform. Writing to the TOmn bit is disabled (writing is ignored).

Caution Be sure to clear bits 15 to 4 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.10 Timer output register m (TOm)

The TOm 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 (TOmn) of each channel.

The TOmn bit oh this register can be rewritten by software only when timer output is disabled (TOEmn = 0). When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P00/TI00, P01/TO00, P16/TI01/TO01, P17/TI02/TO02, or P31/TI03/TO03 pin as a port function pin, set the corresponding TOmn bit to “0”.

The TOm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOm register can be set with an 8-bit memory manipulation instruction with TOmL.

Reset signal generation clears this register to 0000H.

Figure 6-17. Format of Timer Output register m (TOm)

Address: F01B8H, F01B9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOm	0	0	0	0	0	0	0	0	0	0	0	0	TOm3	TOm2	TOm1	TOm0

TOmn	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Caution Be sure to clear bits 15 to 4 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.11 Timer output level register m (TOLm)

The TOLm 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 (TOEmn = 1) in the Slave channel output mode (TOMmn = 1). In the master channel output mode (TOMmn = 0), this register setting is invalid.

The TOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOLm register can be set with an 8-bit memory manipulation instruction with TOLmL.

Reset signal generation clears this register to 0000H.

Figure 6-18. Format of Timer Output Level register m (TOLm)

Address: F01BCH, F01BDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	0	0	0	0	TOLm3	TOLm2	TOLm1	0

TOLmn	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Negative logic output (active-low)

Caution Be sure to clear bits 15 to 4, and 0 to “0”.

Remarks 1. If the value of this register is rewritten during timer operation, the timer output logic is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.3.12 Timer output mode register m (TOMm)

The TOMm 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 (PWM output, one-shot pulse output, or multiple PWM output), 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 (TOEmn = 1).

The TOMm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOMm register can be set with an 8-bit memory manipulation instruction with TOMmL.

Reset signal generation clears this register to 0000H.

Figure 6-19. Format of Timer Output Mode register m (TOMm)

Address: F01BEH, F01BFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOMm	0	0	0	0	0	0	0	0	0	0	0	0	TOMm3	TOMm2	TOMm1	0

TOMmn	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTMmn))
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel)

Caution Be sure to clear bits 15 to 4, and 0 to "0".

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 3 (n = 0, 2 for master channel)
p: Slave channel number
$n < p \leq 3$
(For details of the relation between the master channel and slave channel, refer to **6.4.1 Basic rules of simultaneous channel operation function**.)

6.3.13 Noise filter enable register 1 (NFEN1)

The NFEN1 register is used to set whether the noise filter can be used for the timer input 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 ON, match detection and synchronization of the 2 clocks is performed with the operation clock (f_{MCK}) of the target channel. When the noise filter is OFF, only synchronization is performed with the operation clock (f_{MCK}) of the target channel [Note].

The NFEN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note For details, see **6.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1), 6.5.2 Start timing of counter, and 6.7 Channel Input (TImn pin) Control**.

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	TNFEN02	TNFEN01	TNFEN00

TNFEN03	Enable/disable using noise filter of TI03 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00 pin input signal
0	Noise filter OFF
1	Noise filter ON

6.3.14 Registers controlling port functions of pins to be used for timer I/O

When using the timer array unit, set the register related to the port to be shared with the target channel (port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx)). For details, **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, and **4.3.6 Port mode control registers (PMCxx)**.

When using the ports (such as P00/TI00 and P01/TO00) to be shared with the timer output pin for timer output, set the port mode control register (PMCxx) bit, port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example: When using P01/TO00 for timer output
Set the PMC01 bit of port mode control register 0 to 0.
Set the PM01 bit of port mode register 0 to 0.
Set the P01 bit of port register 0 to 0.

When using the ports (such as P00/TI00) to be shared with the timer input pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. And set the port mode control register (PMCxx) bit corresponding to each port to 0. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P00/TI00 for timer input
Set the PMC00 bit of port mode control register 0 to 0.
Set the PM00 bit of port mode register 0 to 1.
Set the P00 bit of port register 0 to 0 or 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.
(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 master channels, channel 1 can be set as the slave channel of master channel 0. Channel 3 cannot be set as the slave channels of master channel 0.

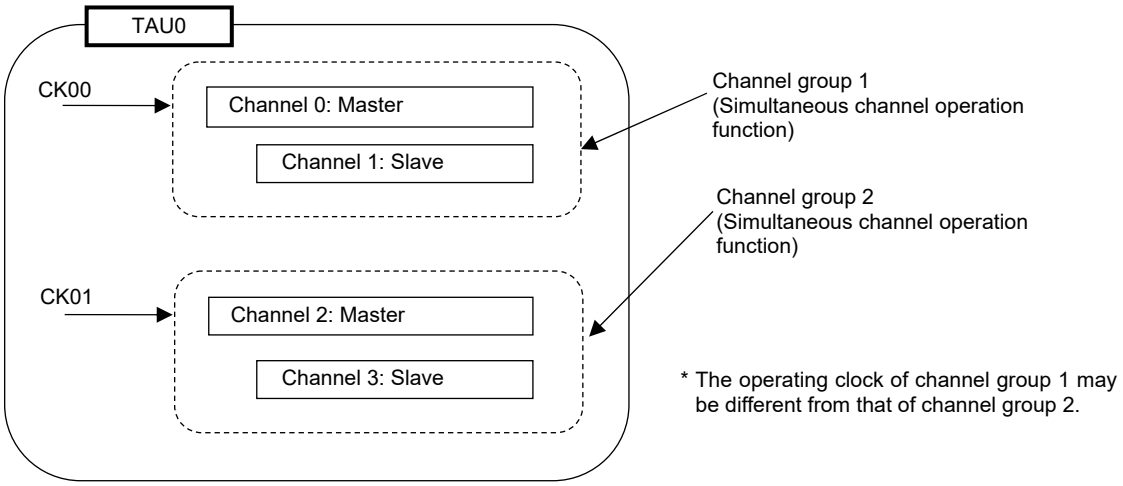
(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKSmn0, CKSmn1 bits (bit 15, 14 of timer mode register mn (TMRmn)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
(7) A master channel can transmit INTTMmn (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use INTTMmn (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or count clock to channels with lower channel numbers.
(9) A master channel cannot use INTTMmn (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TSmn) of the channels in combination must be set at the same time.
(11) During the counting operation, a TSmn bit of a master channel or TSmn bits of all channels which are operating simultaneously can be set. It cannot be applied to TSmn bits of slave channels alone.
(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TTmn) of the channels in combination must be set at the same time.
(13) CKm2/CKm3 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.
(14) Timer mode register m0 (TMRm0) has no master bit (it is fixed as “0”). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

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.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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.

6.4.2 Basic rules of 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 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 SPLITmn bit of timer mode register mn (TMRmn) 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 INTTMm1H/INTTMm3H (an interrupt) (which is the same operation performed when MDmn0 is set to 1).
(5) The operation clock of the higher 8 bits is selected according to the CKSmn1 and CKSmn0 bits of the lower-bit TMRmn register.
(6) For the higher 8 bits, the TSHm1/TSHm3 bit is manipulated to start channel operation and the TTHm1/TTHm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEHm1/TEHm3 bit.
(7) The lower 8 bits operate according to the TMRmn register settings. The following three functions support operation of the lower 8 bits:
   - Interval timer function/square waveform function
   - External event counter function
   - Delay count function

(8) For the lower 8 bits, the TSm1/TSm3 bit is manipulated to start channel operation and the TTm1/TTm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEm1/TEm3 bit.
(9) During 16-bit operation, manipulating the TSHm1, TSHm3, TTHm1, and TTHm3 bits is invalid. The TSm1, TSm3, TTm1, and TTm3 bits are manipulated to operate channels 1 and 3. The TEHm3 and TEHm1 bits are not changed.
(10) For the 8-bit timer function, the simultaneous operation functions (one-shot pulse, PWM, and multiple PWM) cannot be used.

Remark m: Unit number (m = 0), n: Channel number (n = 1, 3)

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 CCSmn bit of timer mode register mn (TMRmn) .

- Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
- Valid edge of input signal input from the TImn pin

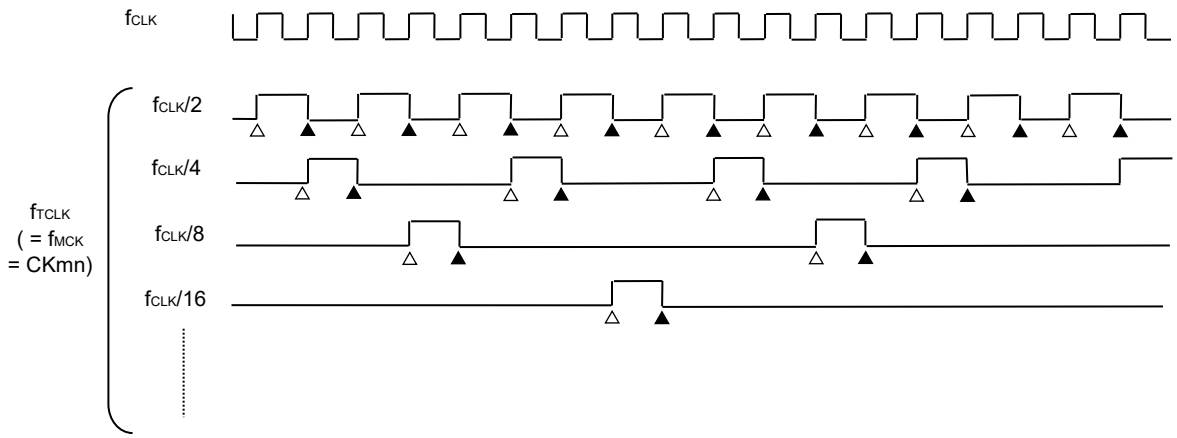
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)

The count clock (f_{TCLK}) is between f_{CLK} to f_{CLK} /2^{15} by setting of timer clock select register m (TPSm). When a divided f_{CLK} is selected, however, the count clock is not a signal which is simply divided f_{CLK} by 2^r, but a signal which becomes high level for one period of f_{CLK} from its rising edge (r = 1 to 15).

Counting of timer count register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as “counting at rising edge of the count clock”, as a matter of convenience.

Figure 6-21. Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 0)



Remarks 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter

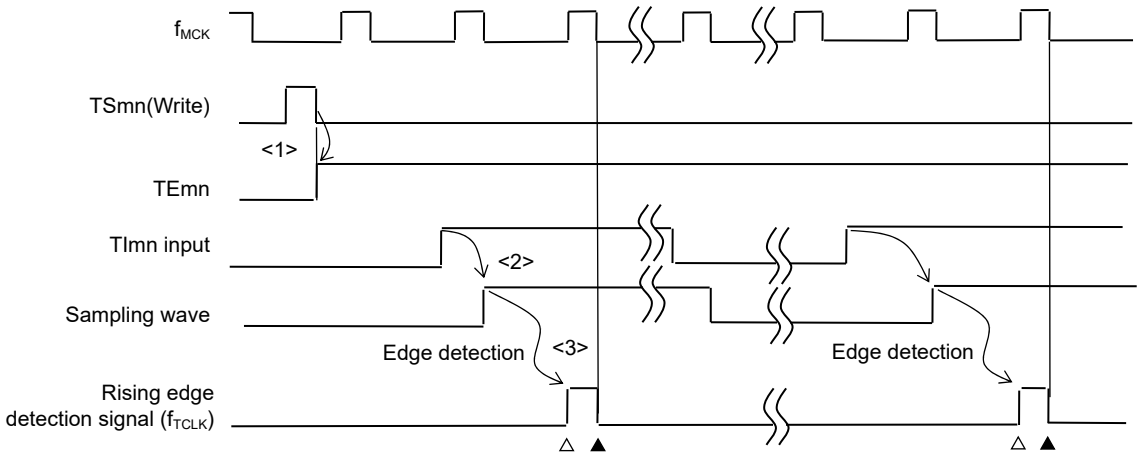
2. f_{CLK}: CPU/peripheral hardware clock

(2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TImn 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 TImn pin (when a noise filter is used, the delay becomes 3 to 4 clock).

Counting of timer count register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as "counting at valid edge of input signal via the TImn pin", as a matter of convenience.

Figure 6-22. Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 1, noise filter unused)



<1> Setting TSmn bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TImn pin.

<2> The rise of input signal via the TImn pin is sampled by f_{MCK}.

<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remarks 1. △ : Rising edge of the count clock

▲ : 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 TImn 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 6-21.

6.5.2 Start timing of counter

Timer count register mn (TCRmn) becomes enabled to operation by setting of TSmn bit of timer channel start register m (TSm).

Operations from count operation enabled state to timer count Register mn (TCRmn) count start is shown in Table 6-6.

Table 6-6. Operations from Count Operation Enabled State to Timer count Register mn (TCRmn) Count Start

Timer operation mode	Operation when TSmn = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TSmn=1) until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3 (1) Operation of interval timer mode**).
• Event counter mode	Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register. If detect edge of TImn input. The subsequent count clock performs count down operation (see **6.5.3 (2) Operation of event counter mode**).
• Capture mode	No operation is carried out from start trigger detection (TSmn = 1) until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (3) Operation of capture mode (input pulse interval measurement)**).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3 (4) Operation of one-count mode**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (5) Operation of capture & one-count mode (high-level interval measurement)**).

6.5.3 Operation of counter

Here, the counter operation in each mode is explained.

(1) Operation of interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer count register mn (TCRmn) holds the initial value until count clock generation.

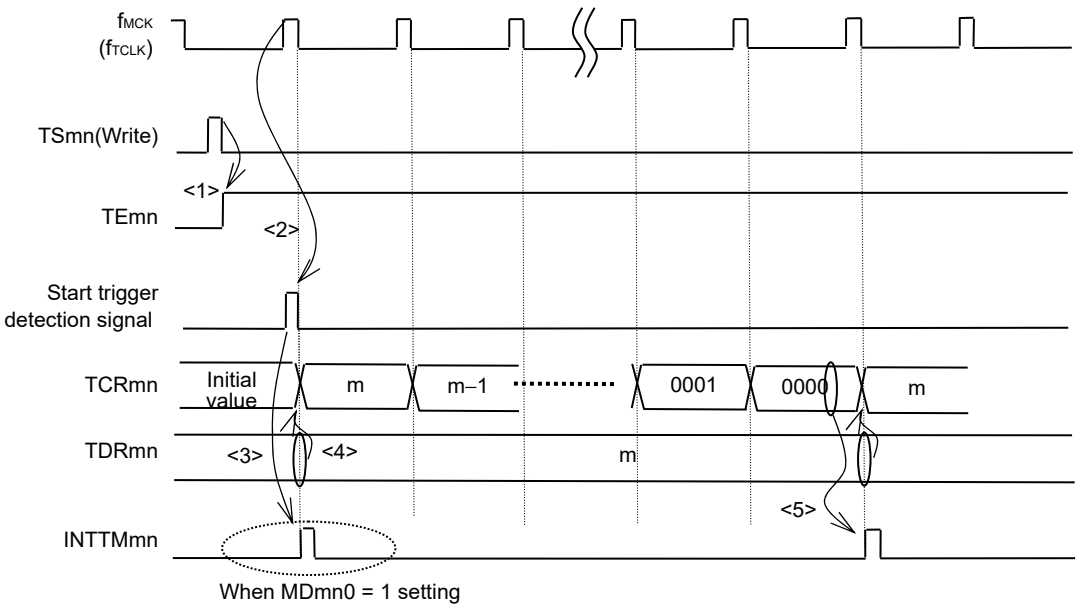
<2> A start trigger is generated at the first count clock after operation is enabled.

<3> When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.

<4> By the first count clock after the operation enable, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting starts in the interval timer mode.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting keeps on.

Figure 6-23. Operation Timing (In Interval Timer Mode)



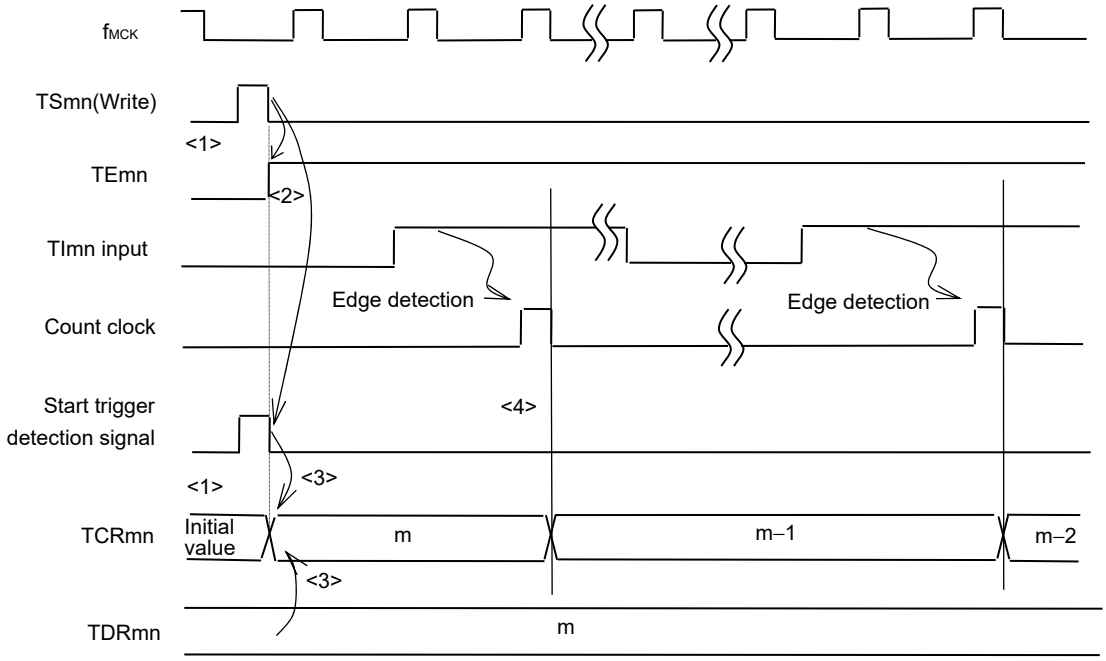
Caution In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

Remark f_{MCK}, the start trigger detection signal, and INTTMmn become active between one clock in synchronization with f_{CLK}.

(2) Operation of event counter mode

<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).
<2> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<3> As soon as 1 has been written to the TSmn bit and 1 has been set to the TEmn bit, the value of timer data register mn (TDRmn) is loaded to the TCRmn register to start counting.
<4> After that, the TCRmn register value is counted down according to the count clock of the valid edge of the TImn input .

Figure 6-24. Operation Timing (In Event Counter Mode)

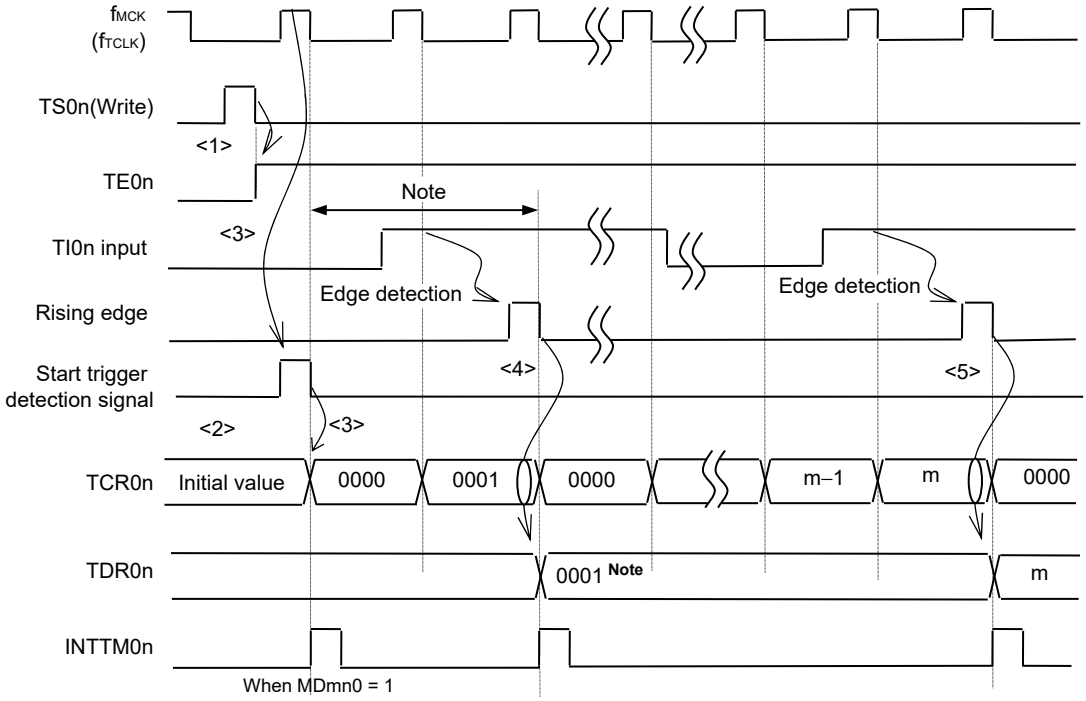


Remark The timing is shown in Figure 6-24 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

(3) Operation of capture mode (input pulse interval measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<2> Timer count register mn (TCRmn) holds the initial value until count clock 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 TCRmn register and counting starts in the capture mode. (When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.)
<4> On detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated. However, this capture value is nomeaning. The TCRmn register keeps on counting from 0000H.
<5> On next detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 6-25. Operation Timing (In Capture Mode : Input Pulse Interval Measurement)



Note If a clock has been input to TImn (the trigger exists) when capturing starts, counting starts when a trigger is detected, 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 after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

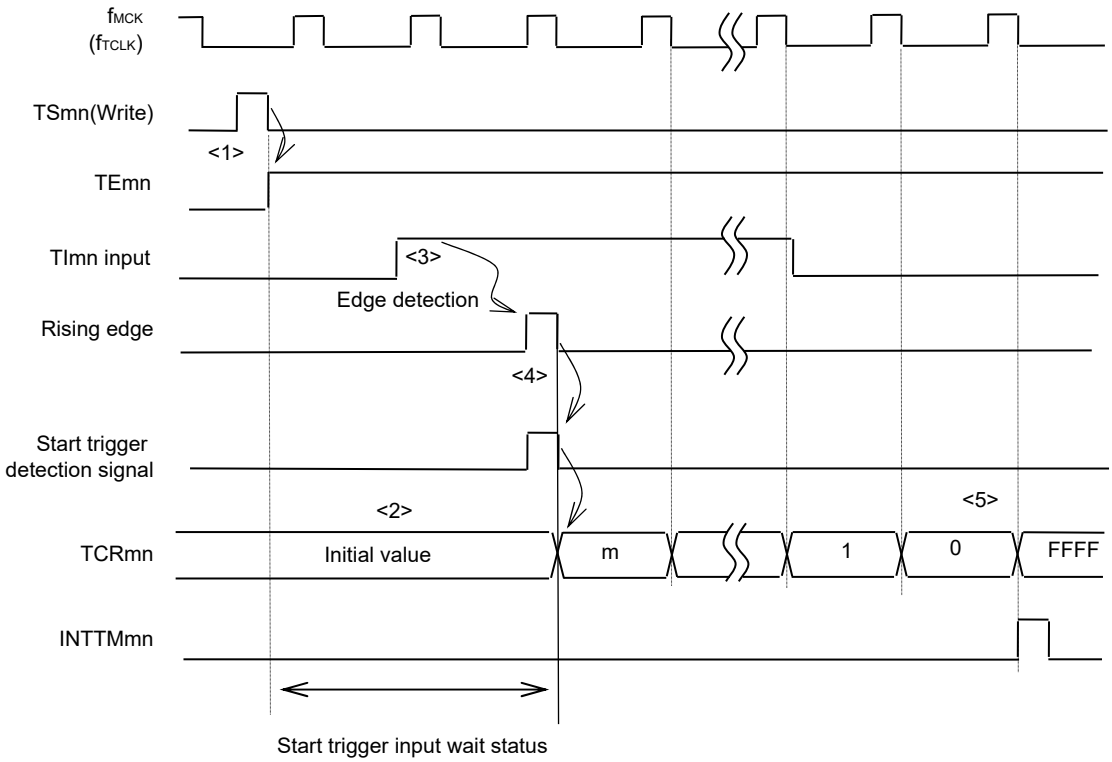
Remark The timing is shown in Figure 6-25 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

(4) Operation of one-count mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.
<3> Rising edge of the TImn input is detected.
<4> On start trigger detection, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and count starts.
<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of the TCRmn register becomes FFFFH and counting stops

.

Figure 6-26. Operation Timing (In One-count Mode)

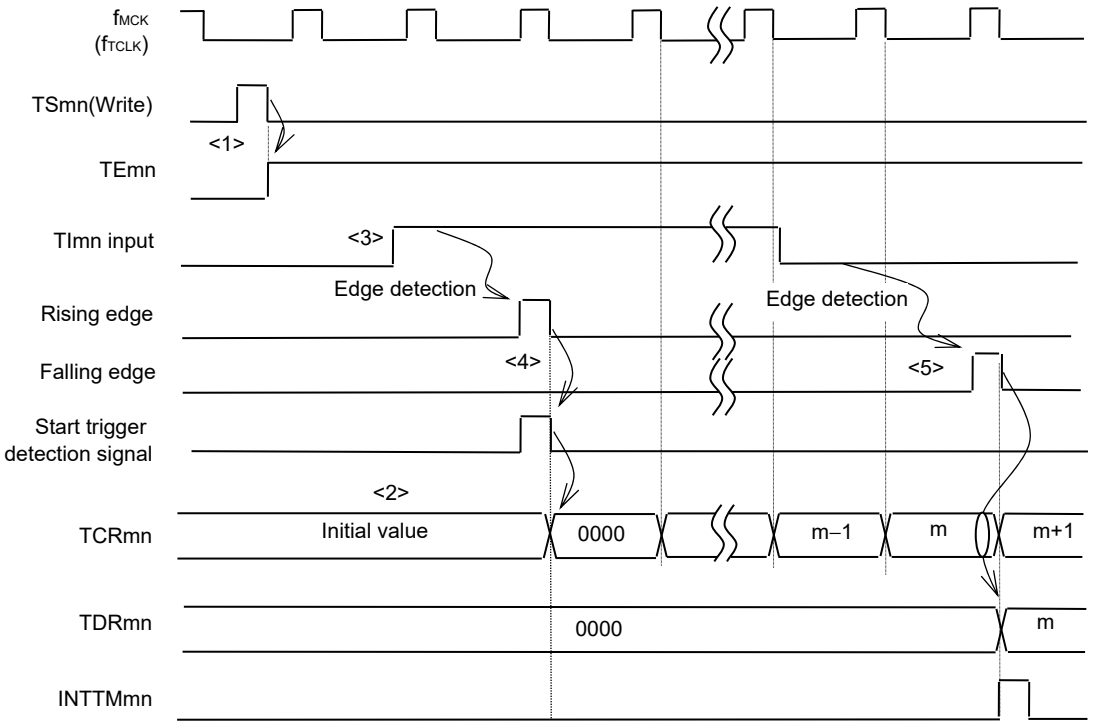


Remark The timing is shown in Figure 6-26 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

(5) Operation of capture & one-count mode (high-level width measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit of timer channel start register m (TSm).
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.
<3> Rising edge of the TImn input is detected.
<4> On start trigger detection, the value of 0000H is loaded to the TCRmn register and count starts.
<5> On detection of the falling edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 6-27. Operation Timing (In Capture & One-count Mode : High-level Width Measurement)

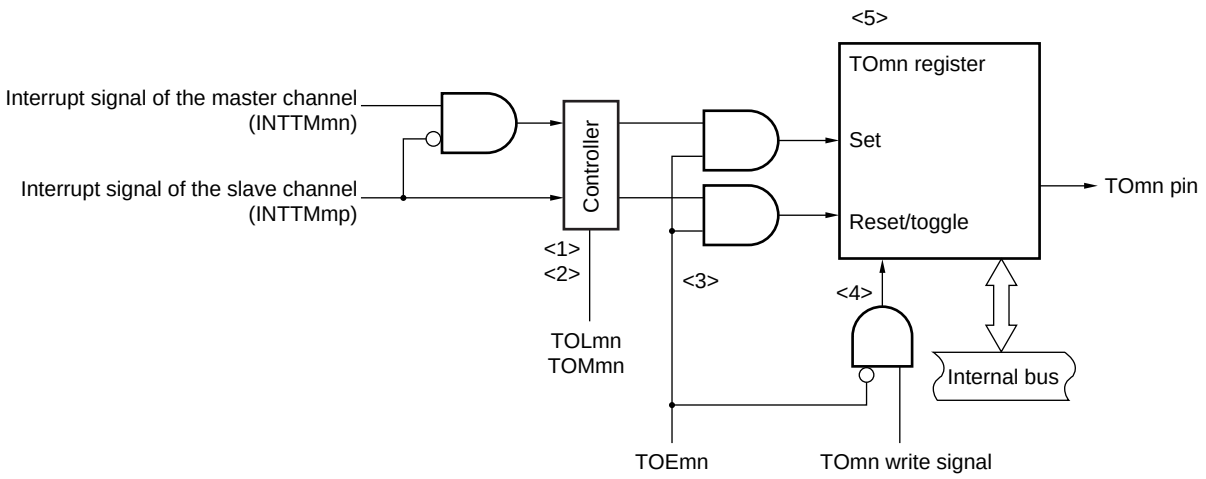


Remark The timing is shown in Figure 6-27 indicates while the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

6.6 Channel Output (TOmn pin) Control

6.6.1 TOmn pin output circuit configuration

Figure 6-28. Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTMmp (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are transmitted to the TOm register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:

When TOLmn = 0: Forward operation (INTTMmn → set, INTTMmp → reset)
When TOLmn = 1: Reverse operation (INTTMmn → reset, INTTMmp → set)

When INTTMmn and INTTMmp are simultaneously generated, (0% output of PWM), INTTMmp (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid.
When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set timer operation is stopeed (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabeled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabeled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TOm register.

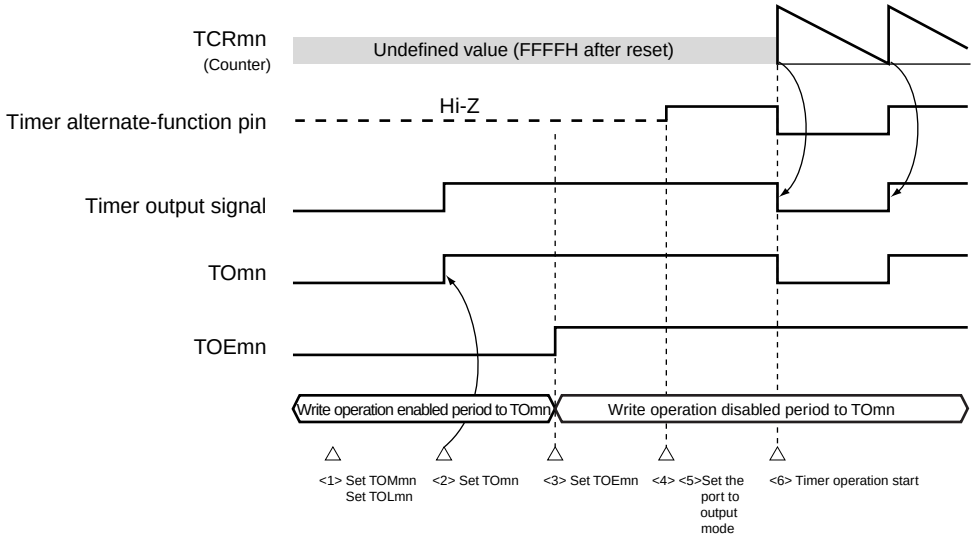
<5> The TOm register can always be read, and the TOmn pin output level can be checked.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 3 (n = 0, 2 for master channel)
p: Slave channel number
$n < p \leq 3$

6.6.2 TOmn pin output setting

The following figure shows the procedure and status transition of the TOmn output pin from initial setting to timer operation start.

Figure 6-29. Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOMmn bit (0: Master channel output mode, 1: Slave channel output mode)
- TOLmn 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 m (TOm).

<3> The timer output operation is enabled by writing 1 to the TOEmn bit (writing to the TOm register is disabled).

<4> The port is set to digital I/O by port mode control register (PMCxx) (see **6.3.14 Registers controlling port functions of pins to be used for timer I/O**).

<5> The port I/O setting is set to output (see **6.3.14 Registers controlling port functions of pins to be used for timer I/O**).

<6> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.6.3 Cautions on channel output operation

(1) Changing values set in the registers TOm, TOEm, and TOLm during timer operation

Since the timer operations (operations of timer count register mn (TCRmn) and timer data register mn (TDRmn)) are independent of the TOmn output circuit and changing the values set in timer output register m (TOm), timer output enable register m (TOEm), and timer output level register m (TOLm) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOmn pin by timer operation, however, set the TOm, TOEm, TOLm, and TOMm registers to the values stated in the register setting example of each operation shown by 6.7 and 6.8.

When the values set to the TOEm, and TOLm registers (but not the TOm register) are changed close to the occurrence of the timer interrupt (INTTMmn) of each channel, the waveform output to the TOmn pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) occurs.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

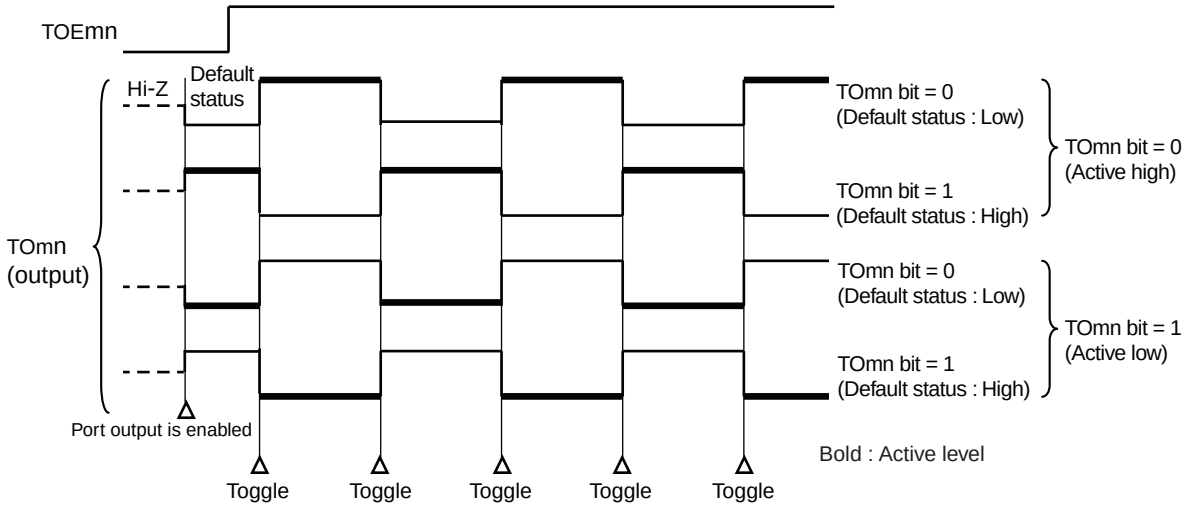
(2) Default level of TOmn pin and output level after timer operation start

The change in the output level of the TOmn pin when timer output register m (TOm) is written while timer output is disabled (TOEmn = 0), the initial level is changed, and then timer output is enabled (TOEmn = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOMmn = 0) setting

The setting of timer output level register m (TOLm) is invalid when master channel output mode (TOMmn = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TOmn pin is reversed.

Figure 6-30. TOmn Pin Output Status at Toggle Output (TOMmn = 0)



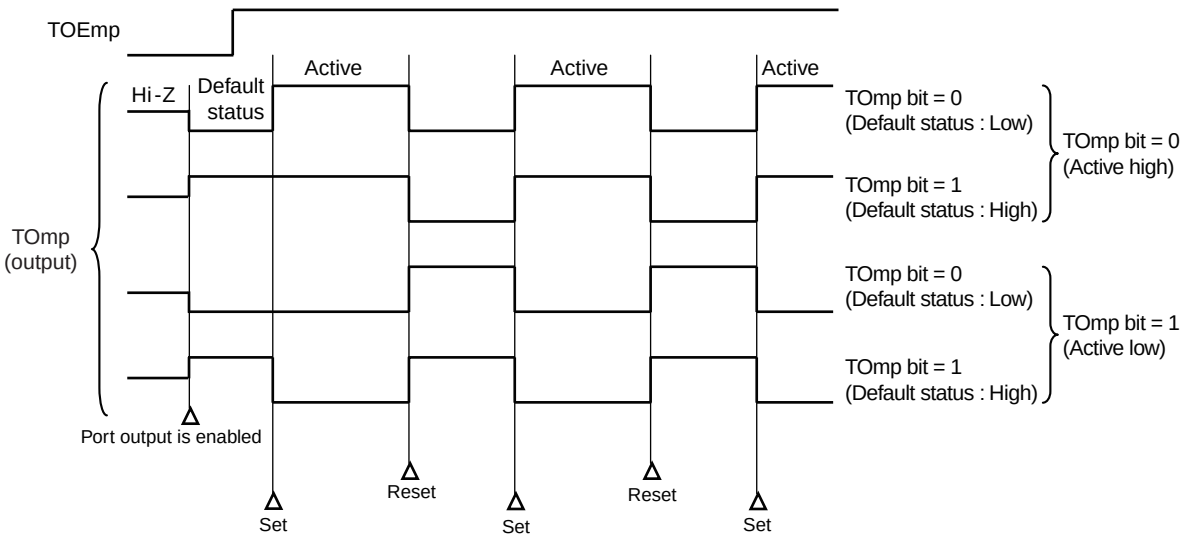
Remarks 1. Toggle: Reverse TOmn pin output status

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

(b) When operation starts with slave channel output mode (TOMmn = 1) setting (PWM output))

When slave channel output mode (TOMmn = 1), the active level is determined by timer output level register m (TOLm) setting.

Figure 6-31. TOmn Pin Output Status at PWM Output (TOMmn = 1)



Remarks 1. Set: The output signal of the TOmn pin changes from inactive level to active level.
Reset: The output signal of the TOmn pin changes from active level to inactive level.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

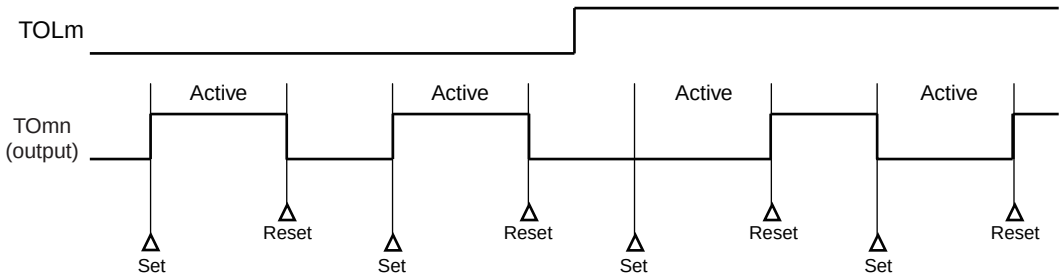
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(a) When timer output level register m (TOLm) setting has been changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOmn pin change condition. Rewriting the TOLm register does not change the output level of the TOmn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEmn = 1) is shown below.

Figure 6-32. Operation when TOLm Register Has Been Changed Contents during Timer Operation



Remarks 1. Set: The output signal of the TOmn pin changes from inactive level to active level.
Reset: The output signal of the TOmn pin changes from active level to inactive level.

2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TOmn pin/TOmn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

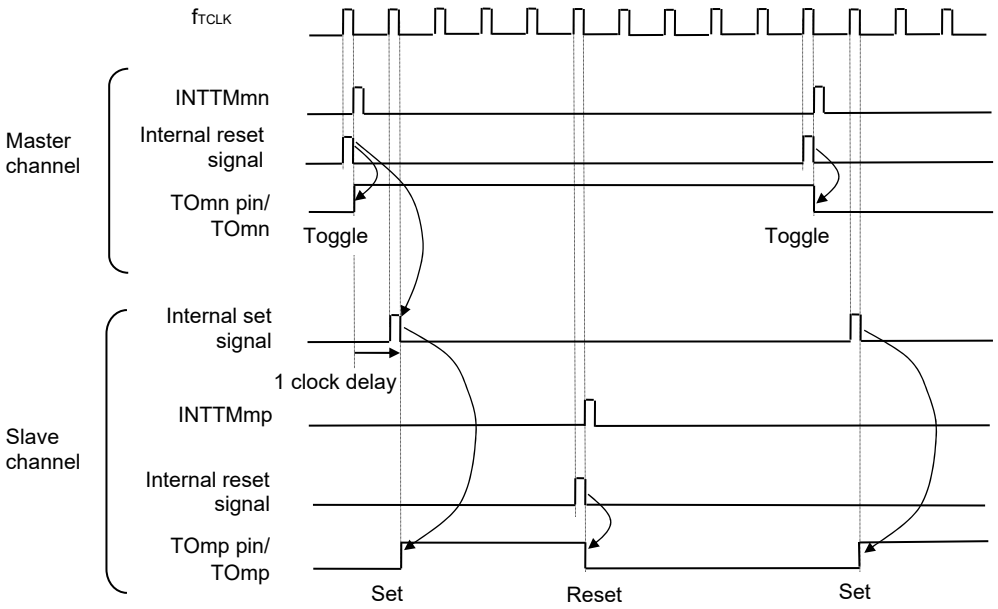
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-33 shows the set/reset operating statuses where the master/slave channels are set as follows.

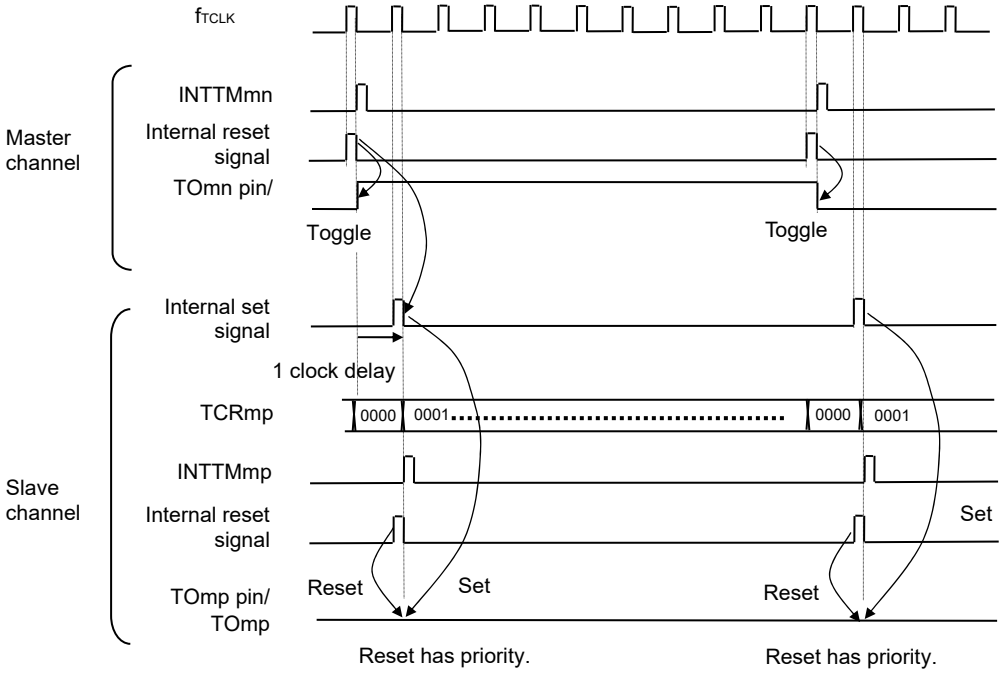
Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 6-33. Set/Reset Timing Operating Statuses

(1) Basic operation timing



(2) Operation timing when 0 % duty



Remarks 1. Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

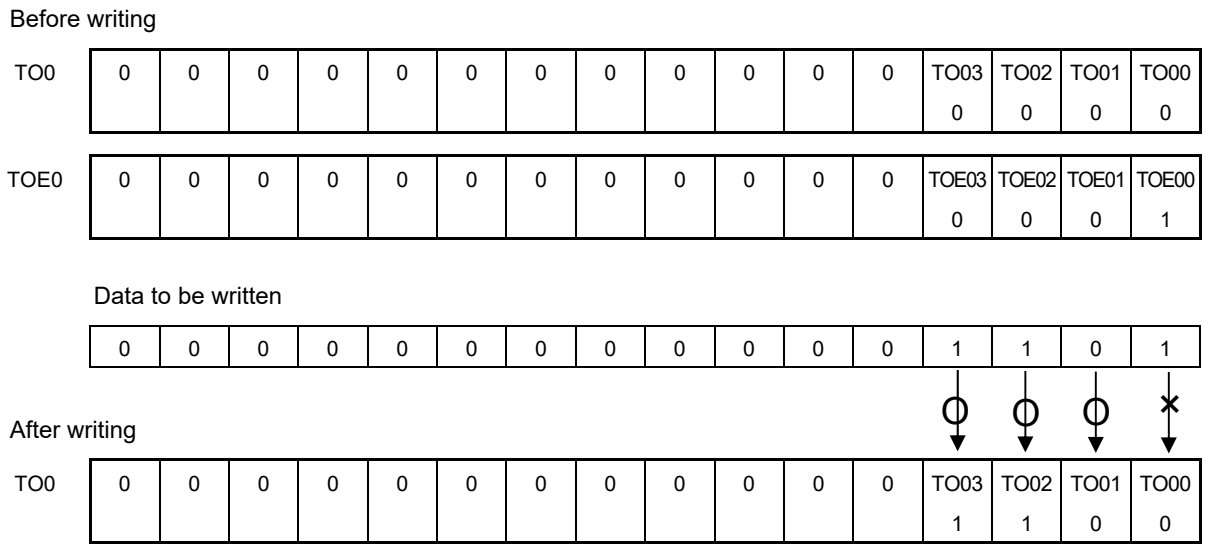
2. m: Unit number (m = 0)
n: Channel number
n = 0 to 3 (n = 0, 2 for master channel)
p: Slave channel number
$n < p \leq 3$

6.6.4 Collective manipulation of TOmn bit

In timer output register m (TOm), the setting bits for all the channels are located in one register in the same way as timer channel start register m (TSm). Therefore, the TOmn bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TOmn bits (TOEmn = 0) that correspond to the relevant bits of the channel used to perform output (TOmn).

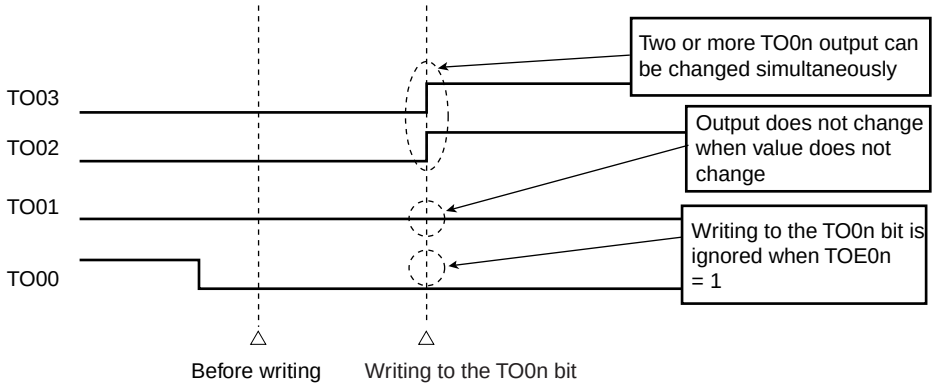
Figure 6-34 Example of TO0n Bit Collective Manipulation



Writing is done only to the TOmn bit with TOEmn = 0, and writing to the TOmn bit with TOEmn = 1 is ignored.

TOmn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to the TOmn bit, it is ignored and the output change by timer operation is normally done.

Figure 6-35. TO0n Pin Statuses by Collective Manipulation of TO0n Bit



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.6.5 Timer interrupt and TOmn pin output at operation start

In the interval timer mode or capture mode, the MDmn0 bit in timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start.

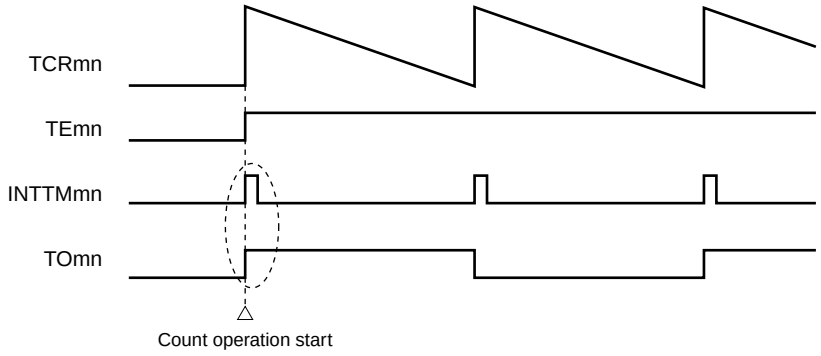
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

In the other modes, neither timer interrupt at count operation start nor TOmn output is controlled.

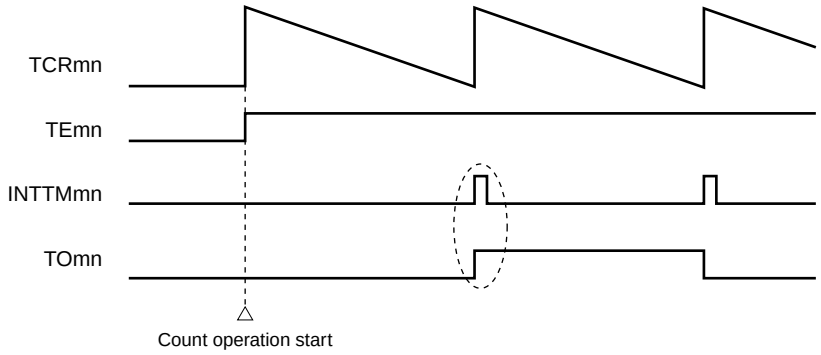
Figure 6-36 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 6-36. Operation examples of timer interrupt at count operation start and TOmn output

(a) When MDmn0 is set to 1



(b) When MDmn0 is set to 0



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOmn does not change either. After counting one cycle, INTTMmn is output and TOmn performs a toggle operation.

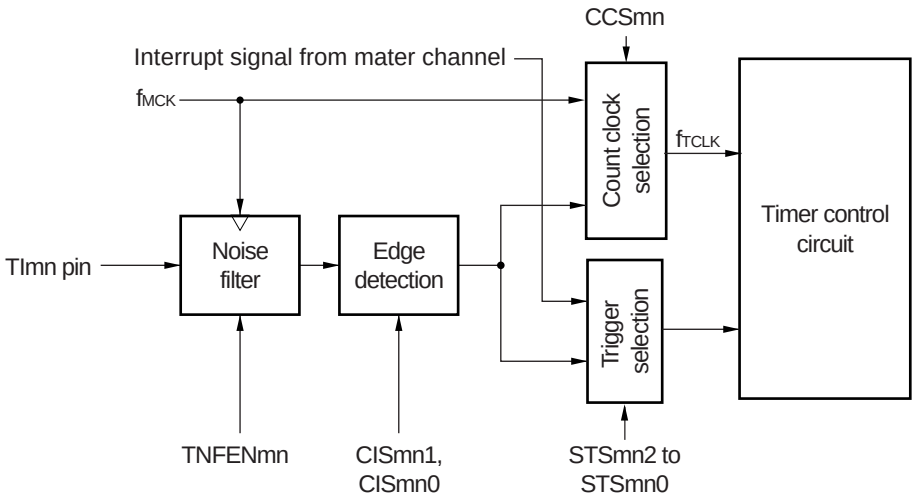
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.7 Timer Input (TImn) Cntorol

6.7.1 TImn pin input circuit configuration

The signal input from a timer input pin passes through a noise filter and edge detector, and is then input to the timer controller. If it is necessary to eliminate noise at the pin in question, enable its noise filter. The configuration of the input circuit is shown below.

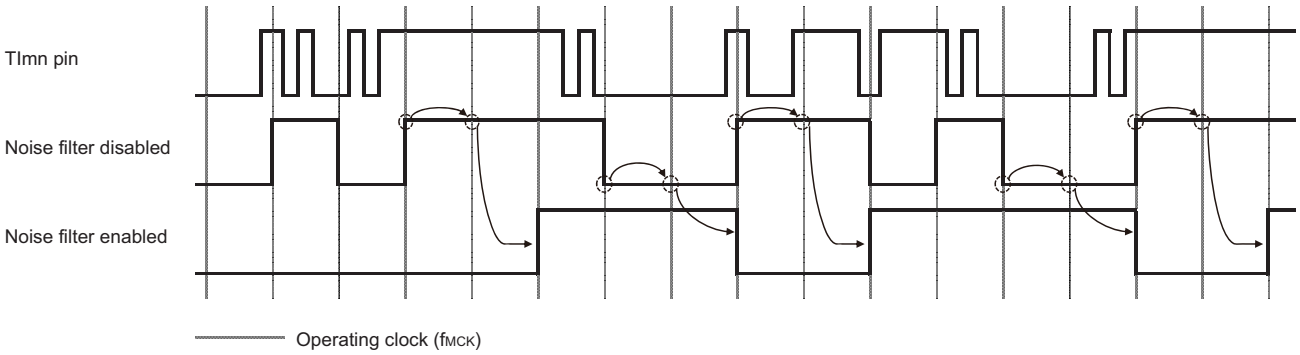
Figure 6-37. Iutput Circuit Configuration



6.7.2 Noise filter

If the noise filter is disabled, only synchronization is performed, based on the operating clock of channel n (f_{MCK}). If the noise filter is enabled, synchronization is performed based on the operating clock of channel n (f_{MCK}), and then two-clock match detection is performed. The following shows the waveforms of a signal after passing through a noise filter when the noise filter is enabled and disabled.

Figure 6-38. Waveform of Sampling Signal Input to TImn Pin When Noise Filter Is Enabled and Disabled



6.7.3 Cautions on channel input

When timer input pins are not used, the operating clock is not supplied to the noise filters. When use of a timer input pin is specified, therefore, the system must wait for the time shown below until the channel operation enable trigger flag for the channel corresponding to the timer input pin is set.

(1) When the noise filter is disabled

If bit 12 (CCSmn), bit 9 (STSmn1), or bit 8 (STSmn0) of timer mode register mn (TMRmn) is set when all of these bits are 0, wait for at least two cycles of the operating clock (f_{MCK}), and then set the corresponding timer operation enable trigger flag of the timer channel start register (TSm).

(2) When the noise filter is enabled

If bit 12 (CCSmn), bit 9 (STSmn1), or bit 8 (STSmn0) of timer mode register mn (TMRmn) is set when all of these bits are 0, wait for at least four cycles of the operating clock (f_{MCK}), and then set the corresponding timer operation enable trigger flag of the timer channel start register (TSm).

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 INTTMmn (timer interrupt) at fixed intervals.
The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

(2) Operation as square wave output

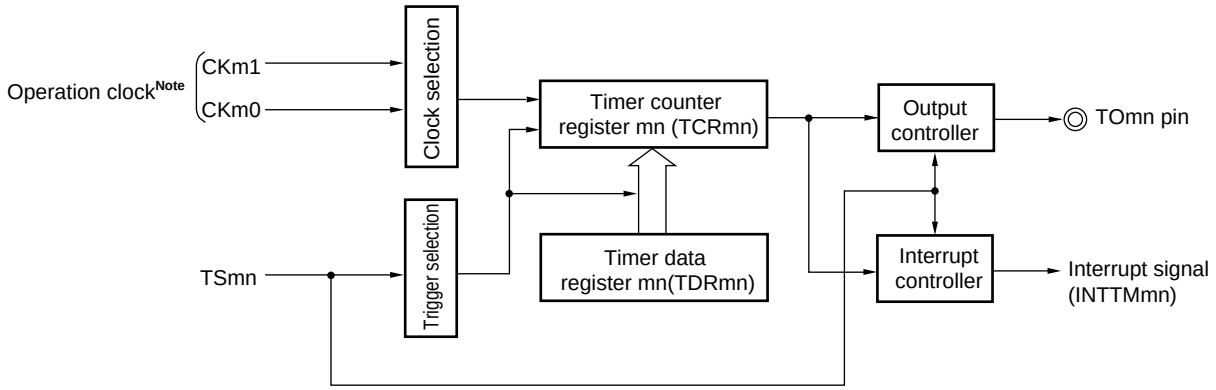
TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.
The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

• Period of square wave output from TOmn = Period of count clock × (Set value of TDRmn + 1) × 2
• Frequency of square wave output from TOmn = Frequency of count clock/{(Set value of TDRmn + 1) × 2}

Timer count register mn (TCRmn) operates as a down counter in the interval timer mode.
The TCRmn register loads the value of timer data register mn (TDRmn) at the first count clock after the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1. If the MDmn0 bit of timer mode register mn (TMRmn) is 0 at this time, INTTMmn is not output and TOmn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and TOmn is toggled.
After that, the TCRmn register count down in synchronization with the count clock.
When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

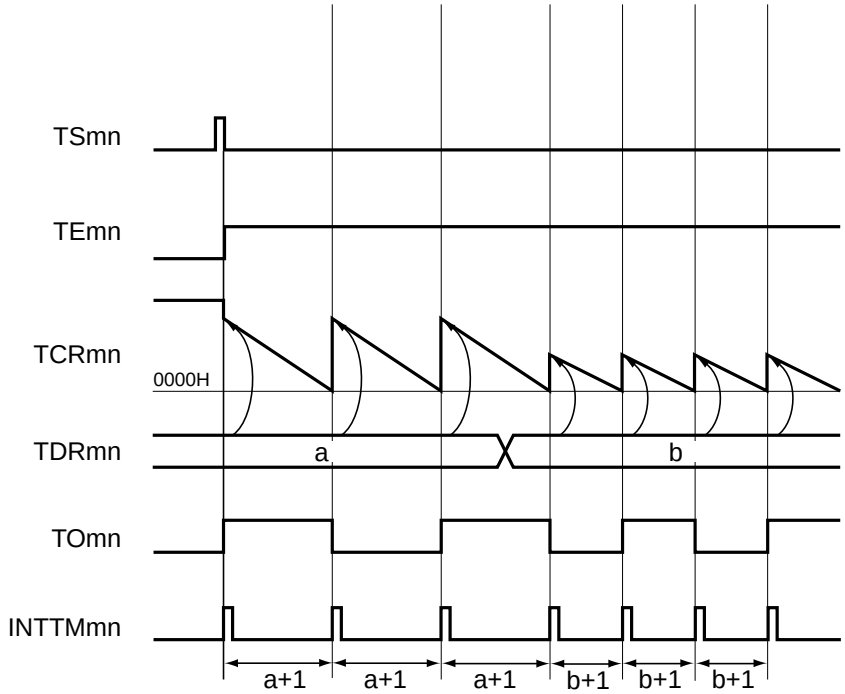
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-39. Block Diagram of Operation as Interval Timer/Square Wave Output



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6-40. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

2. TSmn: Bit n of timer channel start register m (TSm)

TEmn: Bit n of timer channel enable status register m (TEm)

TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

TOmn: TOmn pin output signal

Figure 6-41. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0/1	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.
1: Generates INTTMmn and inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn or SPLITmn bit
0: Independent channel operation function.
(This is set to 1 when using channels 1 and 3 (TMRm1 and TMRm3) in the 8-bit timer mode.)

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	1/0

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	1/0

0: Stops the TOmn output operation by counting operation.
1: Enables the TOmn output operation by counting operation.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-41. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-42. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUnEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOmn output Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output. →	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets the TOEmn bit to 1 and enables operation of TOmn. →	TOmn does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmn pin outputs the TOmn set level.
Operation start	(Sets the TOEmn bit to 1 only if using TOmn output and resuming operation.). Sets the TSmn (TSHm1, TSHm3) bit to 1. → The TSmn (TSHm1, TSHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn). INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed. Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOm and TOEm registers can be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn (TTHm1, TTHm3) bit is set to 1. → The TTmn (TTHm1, TTHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3), and count operation stops. The TCRmn register holds count value and stops. The TOmn output is not initialized but holds current status.
	The TOEmn bit is cleared to 0 and value is set to the TOmn bit. →	The TOmn pin outputs the TOmn bit set level.

Operation is resumed. → Operation start

(**Remark** is listed on the next page.)

Figure 6-42. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOmn pin output level Clears the TOmn bit to 0 after the value to be held is set to the port register. →	The TOmn pin output level is held by port function.
	When holding the TOmn pin output level is not necessary Setting not required.	
	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 TImn pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDRmn + 1

Timer count register mn (TCRmn) operates as a down counter in the event counter mode.

The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit (TSmn) of timer channel start register m (TSm) to 1.

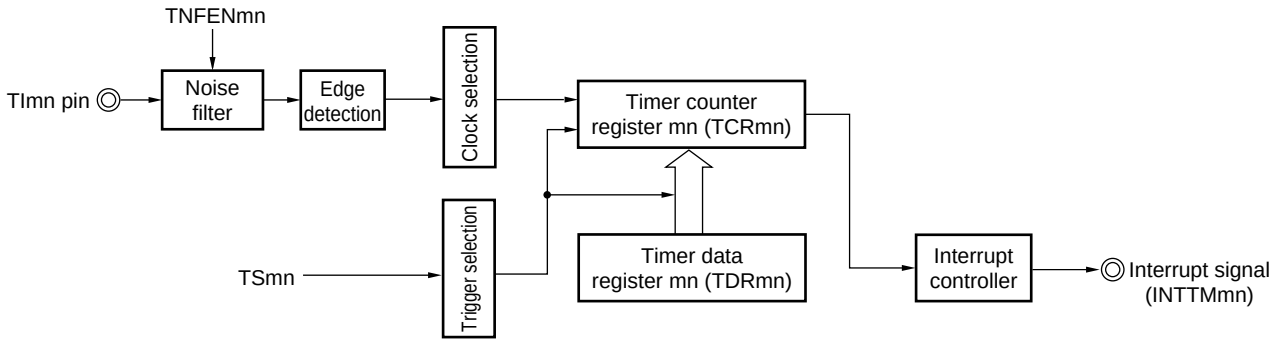
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TOmn pin. Stop the output by setting the TOEmn bit of timer output enable register m (TOEm) to 0.

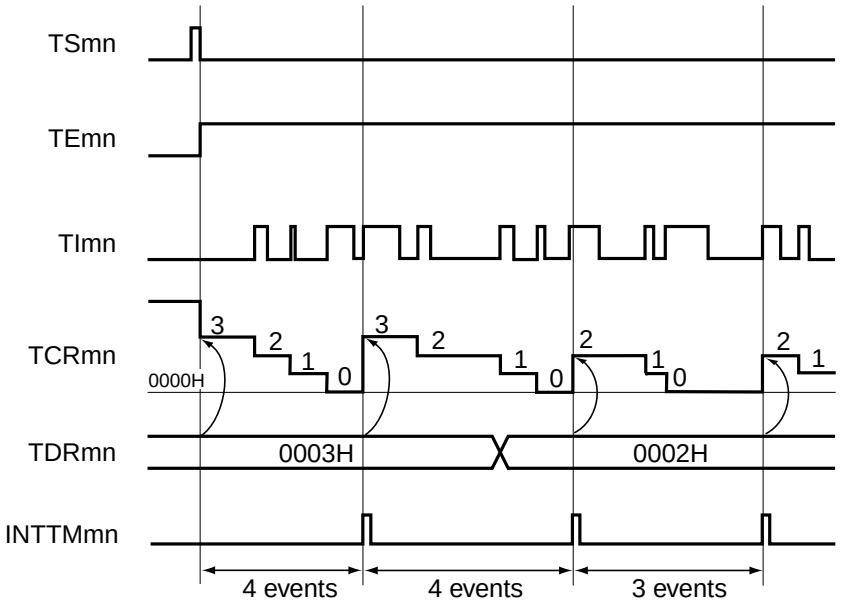
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 6-43. Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-44. Example of Basic Timing of Operation as External Event Counter



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

2. TSmn: Bit n of timer channel start register m (TSm)

TEmn: Bit n of timer channel enable status register m (TEm)

TImn: TImn pin input signal

TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

Figure 6-45. Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	1	0/1	0	0	0	1/0	1/0	0	0	0	1	1	0

Operation mode of channel n
011B: Event count mode

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
1: Selects the TImn pin input valid edge.

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-45. Example of Set Contents of Registers in External Event Counter Mode (2/2)

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-46. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets number of counts to timer data register mn (TDRmn). Clears the TOEmn bit of timer output enable register m (TOEm) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. →	TEmn = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn) and detection of the TImn pin input edge is awaited.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts down each time input edge of the TImn pin has been detected. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again, and the count operation is continued. By detecting TCRmn = 0000H, the INTTMmn output is generated. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. →	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.8.3 Operation as frequency divider (channel 0 only)

The timer array unit can be used as a frequency divider that divides a clock input to the TI00 pin and outputs the result from the TO00 pin.

The divided clock frequency output from TO00 can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDR00 + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDR00 + 1)

Timer count register 00 (TCR00) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS00) of timer channel start register 0 (TS0) is set to 1, the TCR00 register loads the value of timer data register 00 (TDR00) when the TI00 valid edge is detected.

If the MD000 bit of timer mode register 00 (TMR00) is 0 at this time, INTTM00 is not output and TO00 is not toggled. If the MD000 bit of timer mode register 00 (TMR00) is 1, INTTM00 is output and TO00 is toggled.

After that, the TCR00 register counts down at the valid edge of the TI00 pin. When TCR00 = 0000H, it toggles TO00. At the same time, the TCR00 register loads the value of the TDR00 register again, and continues counting.

If detection of both the edges of the TI00 pin is selected, the duty factor error of the input clock affects the divided clock period of the TO00 output.

The period of the TO00 output clock includes a sampling error of one period of the operation clock.

Clock period of TO00 output = Ideal TO00 output clock period ± Operation clock period (error)

The TDR00 register can be rewritten at any time. The new value of the TDR00 register becomes valid during the next count period.

Figure 6-47. Block Diagram of Operation as Frequency Divider

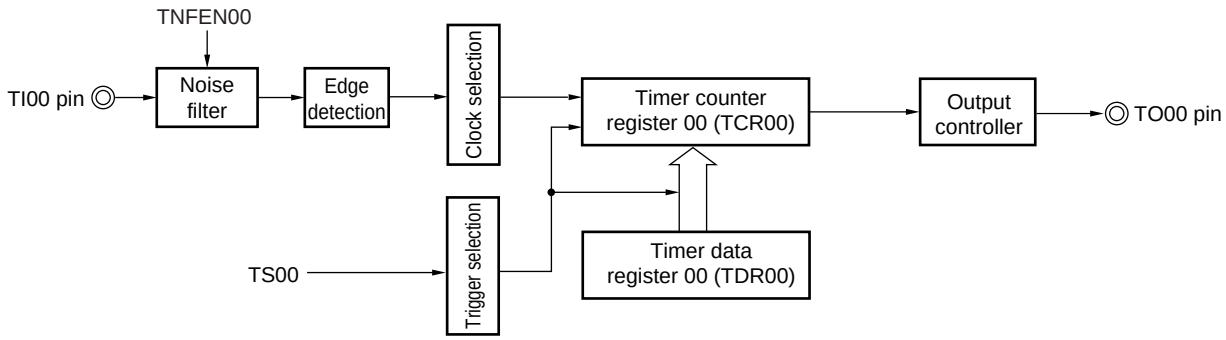
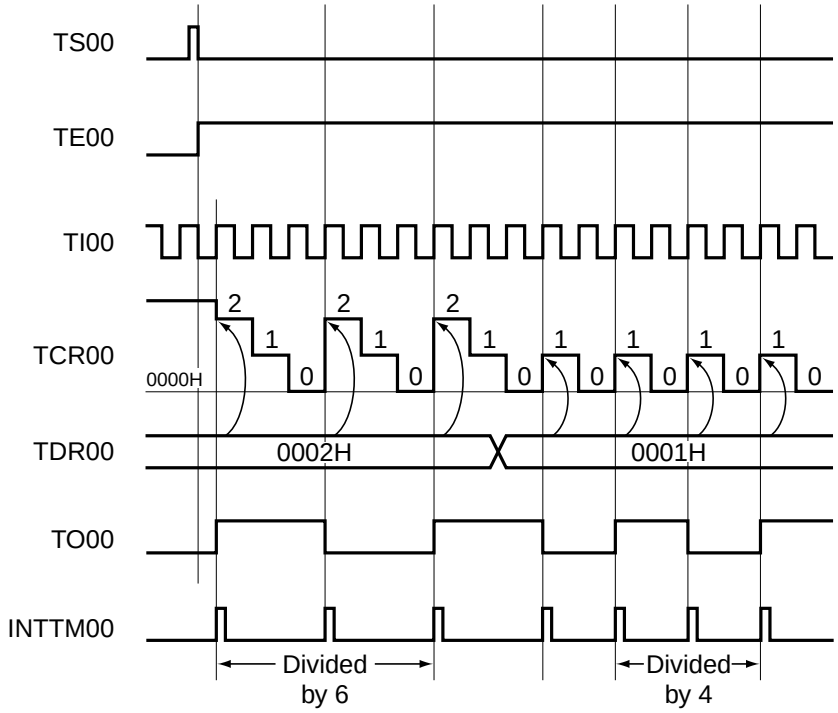


Figure 6-48. Example of Basic Timing of Operation as Frequency Divider (MD000 = 1)



Remark TS00: Bit n of timer channel start register 0 (TS0)
TE00: Bit n of timer channel enable status register 0 (TE0)
TI00: TI00 pin input signal
TCR00: Timer count register 00 (TCR00)
TDR00: Timer data register 00 (TDR00)
TO00: TO00 pin output signal

Figure 6-49. Example of Set Contents of Registers During Operation as Frequency Divider

(a) Timer mode register 00 (TMR00)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR00	CKS0n1	CKS0n0		CCS00		STS002	STS001	STS000	CIS001	CIS000			MD003	MD002	MD001	MD000
	1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	0	0	1/0

Operation mode of channel 0
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTM00 nor inverts timer output when counting is started.
1: Generates INTTM00 and inverts timer output when counting is started.

Selection of TI00 pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Count clock selection
1: Selects the TI00 pin input valid edge.

Operation clock (f_{MCK}) selection
00B: Selects CK00 as operation clock of channel 0.
10B: Selects CK01 as operation clock of channel 0.

(b) Timer output register 0 (TO0)

	Bit 0	
TO0	TO00	0: Outputs 0 from TO00.
	1/0	1: Outputs 1 from TO00.

(c) Timer output enable register 0 (TOE0)

	Bit 0	
TOE0	TOE00	0: Stops the TO00 output operation by counting operation.
	1/0	1: Enables the TO00 output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit 0	
TOL0	TOL00	0: Cleared to 0 when TOM00 = 0 (master channel output mode)
	0	

(e) Timer output mode register 0 (TOM0)

	Bit 0	
TOM0	TOM00	0: Sets master channel output mode.
	0	

Figure 6-50. Operation Procedure When Frequency Divider Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets timer mode register 0n (TMR0n) (determines operation mode of channel and selects the detection edge). Sets interval (period) value to timer data register 00 (TDR00).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM00 bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL00 bit to 0. Sets the TO00 bit and determines default level of the TO00 output.	The TO00 pin goes into Hi-Z output state. The TO00 default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE00 bit to 1 and enables operation of TO00.	TO00 does not change because channel stops operating.
	Clears the port register and port mode register to 0.	The TO00 pin outputs the TO00 set level.
Operation start	Sets the TOE00 bit to 1 (only when operation is resumed). Sets the TS00 bit to 1. The TS00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 1, and count operation starts. Value of the TDR00 register is loaded to timer count register 00 (TCR00) at the count clock input. INTTM00 is generated and TO00 performs toggle operation if the MD000 bit of the TMR00 register is 1.
During operation	Set value of the TDR00 register can be changed. The TCR00 register can always be read. The TSR00 register is not used. Set values of the TO0 and TOE0 registers can be changed. Set values of the TMR00 register, TOM00, and TOL00 bits cannot be changed.	Counter (TCR00) counts down. When count value reaches 0000H, the value of the TDR00 register is loaded to the TCR00 register again, and the count operation is continued. By detecting TCR00 = 0000H, INTTM00 is generated and TO00 performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT00 bit is set to 1. The TT00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 0, and count operation stops. The TCR00 register holds count value and stops. The TO00 output is not initialized but holds current status.
	The TOE00 bit is cleared to 0 and value is set to the TO00 bit.	The TO00 pin outputs the TO00 set level.
TAU stop	To hold the TO00 pin output level Clears the TO00 bit to 0 after the value to be held is set to the port register. When holding the TO00 pin output level is not necessary Setting not required.	The TO00 pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO00 bit is cleared to 0 and the TO00 pin is set to port mode).

Operation is resumed.

6.8.4 Operation as input pulse interval measurement

The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured. In addition, the count value can be captured by using software operation (TSmn = 1) as a capture trigger while the TEmn bit is set to 1.

The pulse interval can be calculated by the following expression.

TImn input pulse interval = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of up to one operating clock cycle occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TCRmn register counts up from 0000H in synchronization with the count clock.

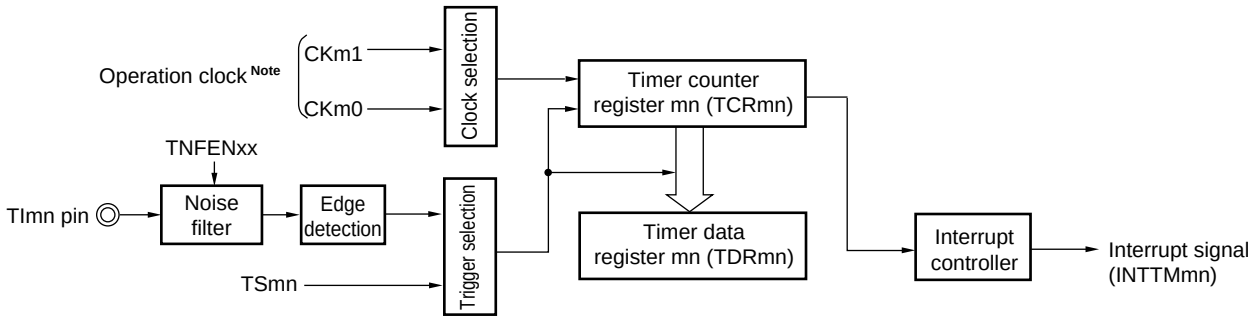
When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) 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 TDRmn register, the OVF bit of the TSRmn 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 TSRmn 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 STSmn2 to STSmn0 bits of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

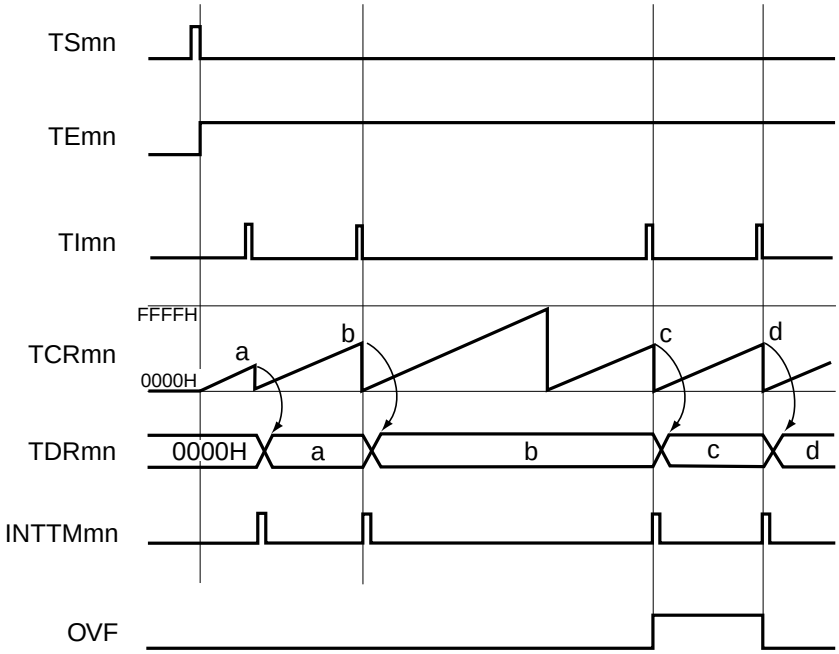
Figure 6-51. Block Diagram of Operation as Input Pulse Interval Measurement



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-52. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

2. TSmn: Bit n of timer channel start register m (TSm)

TEmn: Bit n of timer channel enable status register m (TEm)

TImn: TImn pin input signal

TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

OVF: Bit 0 of timer status register mn (TSRmn)

Figure 6-53. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	0	0	0	1	1/0	1/0	0	0	0	1	0	1/0

Operation mode of channel n
010B: Capture mode

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Capture trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn or SPLITmn bit
0: Independent channel operation function.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode.

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-54. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSmn bit to 1. →	TEmn = 1, and count operation starts. Timer count register mn (TCRmn) is cleared to 0000H at the count clock input. When the MDmn0 bit of the TMRmn register is 1, INTTMmn is generated.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts up from 0000H. When the TImn pin input valid edge is detected, the count value is transferred (captured) to timer data register mn (TDRmn). At the same time, the TCRmn register is cleared to 0000H, and the INTTMmn signal is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. →	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAU0EN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.8.5 Operation as input signal high-/low-level width measurement

By starting counting at one edge of the TImn pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

Signal width of TImn input = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error equivalent to one operation clock occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait status is set.

When the TImn pin input start edge (rising edge of the TImn 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 TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value “value transferred to the TDRmn register + 1”, and the TImn 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 TDRmn register, the OVF bit of the TSRmn 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 TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

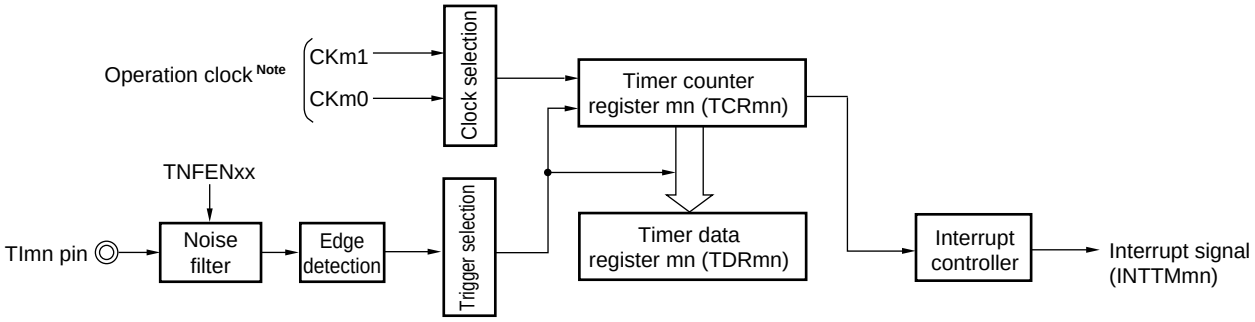
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

Because this function is used to measure the signal width of the TImn pin input, the TSmn bit cannot be set to 1 while the TEmn bit is 1.

CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.

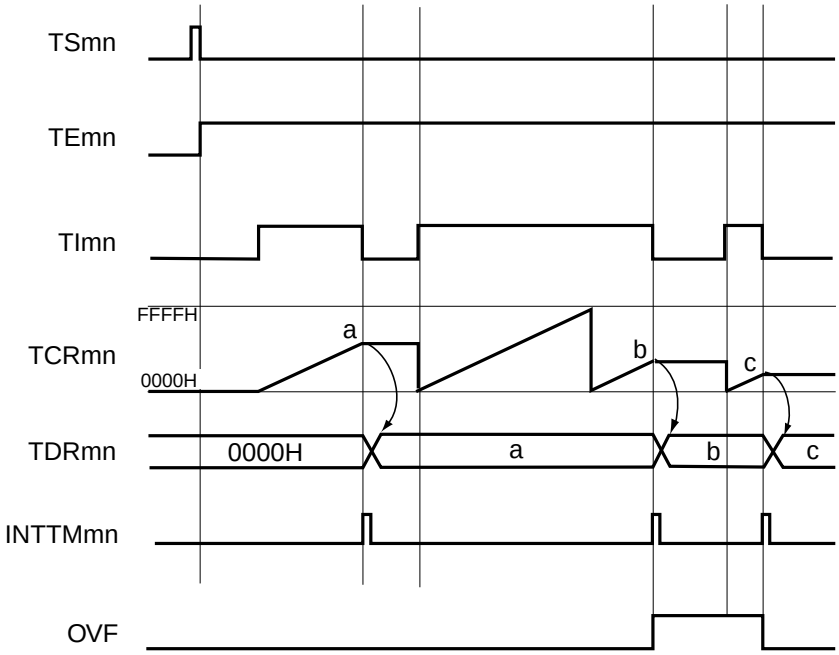
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 6-55. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Note For channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6-56. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 6-57. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	0	0	1	0	1	1/0	0	0	1	1	0	0

Operation mode of channel n
110B: Capture & one-count

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.

Selection of TImn pin input edge
10B: Both edges (to measure low-level width)
11B: Both edges (to measure high-level width)

Start trigger selection
010B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channel)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
1: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-58. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1.	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait status is set.
	Detects the TImn pin input count start valid edge.	Clears timer count register mn (TCRmn) to 0000H and starts counting up.
During operation	The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to timer data register mn (TDRmn) and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. The TCRmn register stops the count operation until the next TImn pin start edge is detected.
Operation stop	The TTmn bit is set to 1. The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.8.6 Operation as delay counter

It is possible to start counting down when the valid edge of the TImn pin input is detected (an external event), and then generate INTTMmn (a timer interrupt) after any specified interval.

It can also generate INTTMmn (timer interrupt) at any interval by making a software set TSmn = 1 and the count down start during the period of TEmn = 1.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

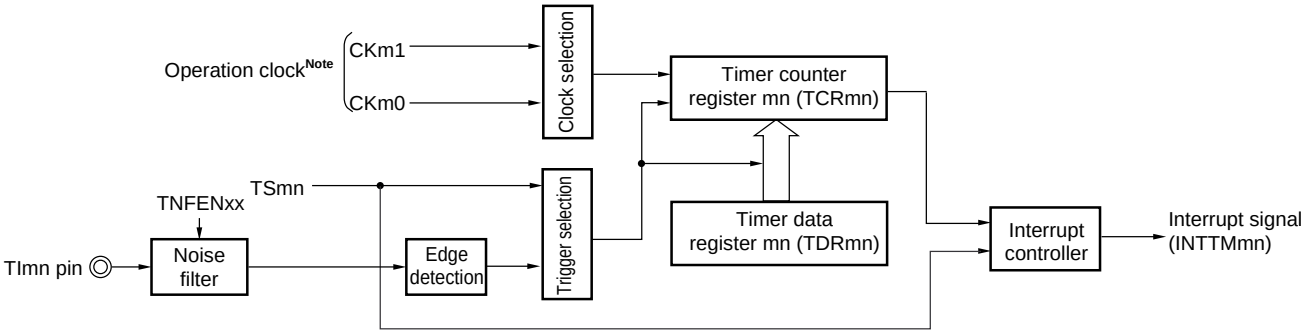
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

When the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1, the TEmn, TEHm1, TEHm3 bits are set to 1 and the TImn pin input valid edge detection wait status is set.

Timer count register mn (TCRmn) starts operating upon TImn pin input valid edge detection and loads the value of timer data register mn (TDRmn). The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next TImn pin input valid edge is detected.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

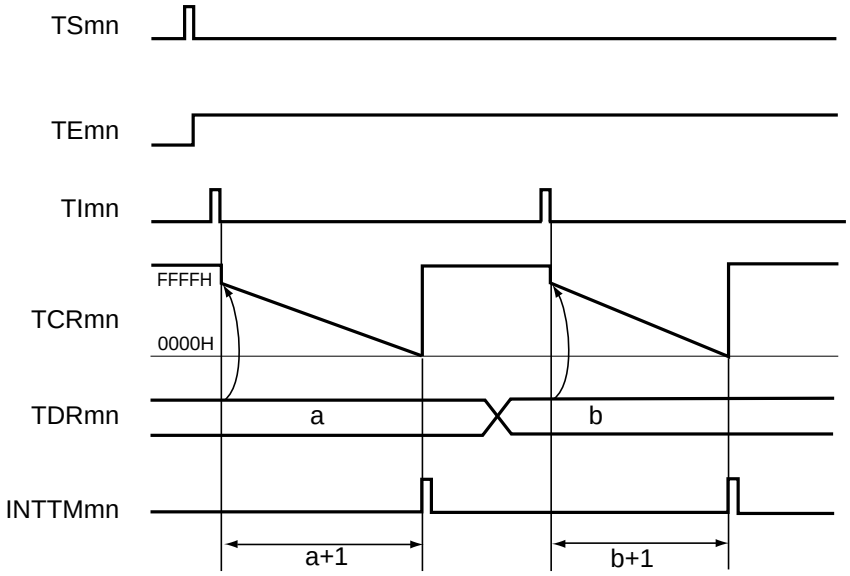
Figure 6-59. Block Diagram of Operation as Delay Counter



Note For using channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-60. Example of Basic Timing of Operation as Delay Counter



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

2. TSmn: Bit n of timer channel start register m (TSm)

TEmn: Bit n of timer channel enable status register m (TEm)

TImn: TImn pin input signal

TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

Figure 6-61. Example of Set Contents of Registers to Delay Counter (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0/1	0	0	1	1/0	1/0	0	0	1	0	0	0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.
1: Trigger input is valid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.

Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-61. Example of Set Contents of Registers to Delay Counter (2/2)

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 6-62. Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). INTTMmn output delay is set to timer data register mn (TDRmn). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1) wait status is set.
	The counter starts counting down by the next start trigger detection. → • Detects the TImn pin input valid edge. • Sets the TSmn bit to 1 by the software.	Value of the TDRmn register is loaded to the timer count register mn (TCRmn).
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used.	The counter (TCRmn) counts down. When the count value of TCRmn reaches 0000H, the INTTMmn output is generated, and the count operation stops until the next start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1).
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

6.9 Simultaneous Channel Operation Function of Timer Array Unit

6.9.1 Operation as one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the TImn pin.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDRmn (master) + 2} × Count clock period
Pulse width = {Set value of TDRmp (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer count register mn (TCRmn) of the master channel starts operating upon start trigger detection and loads the value of timer data register mn (TDRmn).

The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next start trigger is detected.

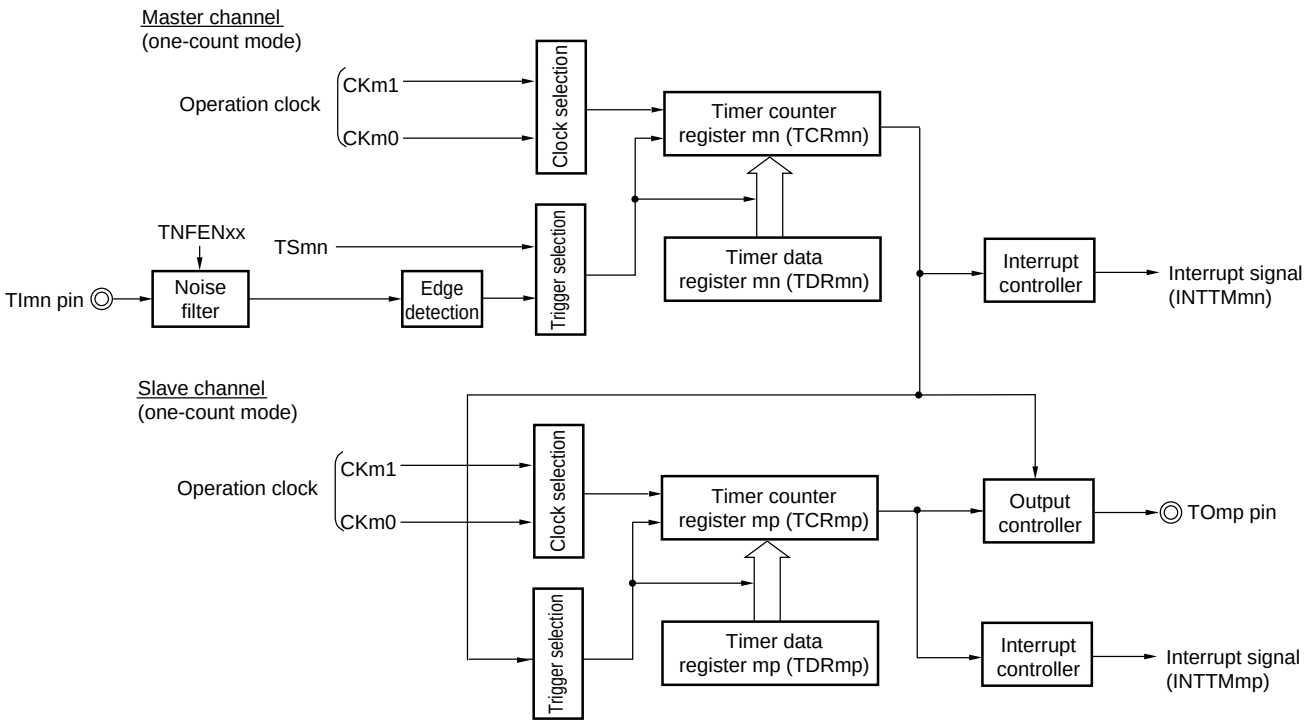
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

Caution The timing of loading of timer data register mn (TDRmn) of the master channel is different from that of the TDRmp register of the slave channel. If the TDRmn and TDRmp registers are rewritten during operation, therefore, an illegal waveform may be output in conflict with the timing of loading. Rewrite the TDRmn register after INTTMmn is generated and the TDRmp register after INTTMmp is generated.

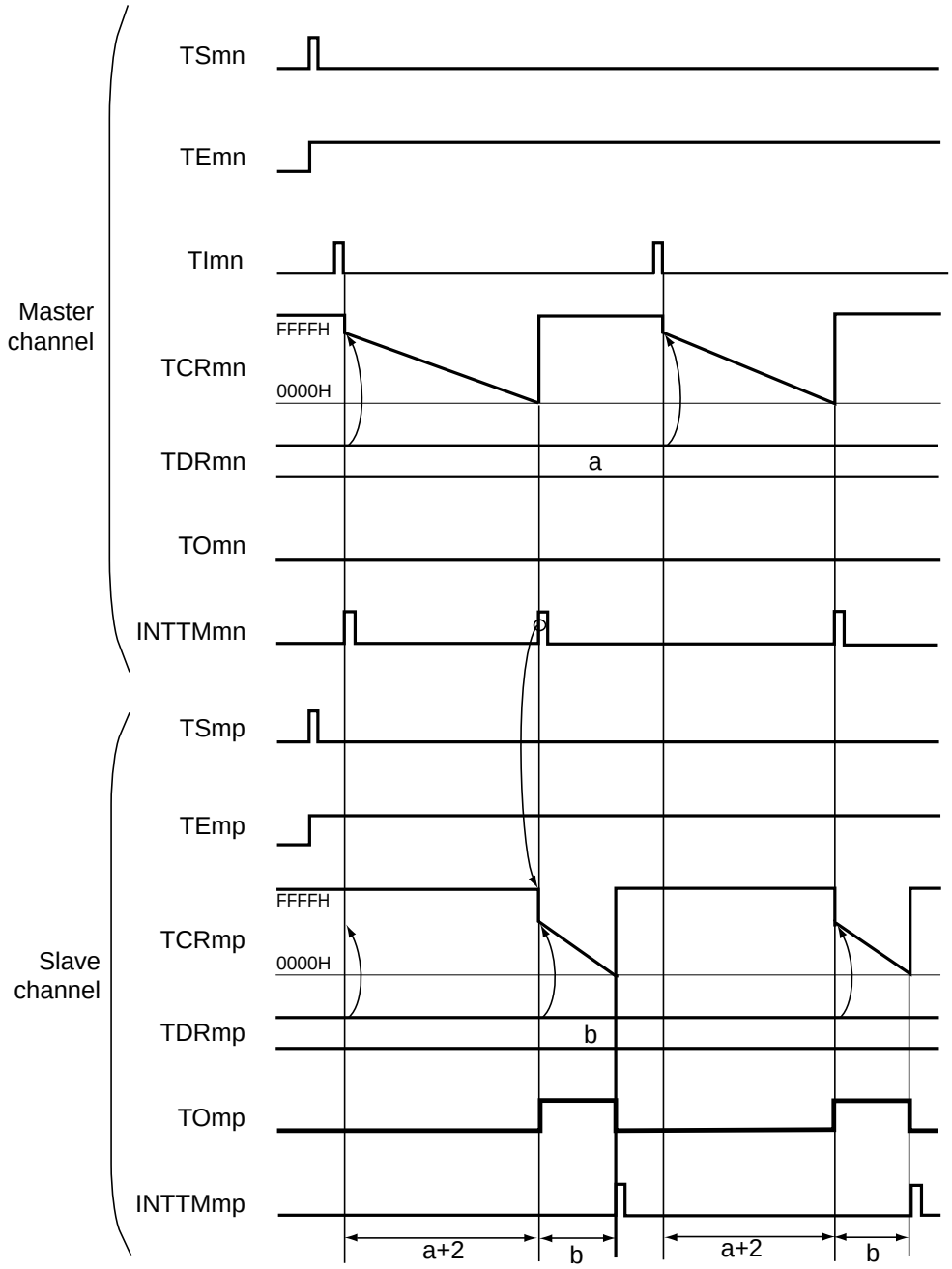
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-63. Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-64. Example of Basic Timing of Operation as One-Shot Pulse Output Function



Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TImn, TImp: TImn and TImp pins input signal
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 6-65. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MASTER mn[Note]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	1	1/0	1/0	0	0	1	0	0	0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channels n.
10B: Selects CKm1 as operation clock of channels n.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

Figure 6-66. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register mp (TMRmp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

Operation mode of channel p
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmn bit (channel 2)
0: Slave channel
Setting of SPLITmp bit (channels 1, 3)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p
TOm	TOmp 1/0

0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p
TOEm	TOEmp 1/0

0: Stops the TOmp output operation by counting operation.
1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p
TOLm	TOLmp 1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p
TOMm	TOMmp 1

1: Sets the slave channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-67. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable registers 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 1. Sets timer mode register mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register mn (TDRmn) of the master channel, and a pulse width is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

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

Figure 6-67. Operation Procedure of One-Shot Pulse Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	The TEmn and TEmp bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1) wait status. Counter stops operating.
	Count operation of the master channel is started by start trigger detection of the master channel. • Detects the TImn pin input valid edge. • Sets the TSmn bit of the master channel to 1 by software Note. **Note** Do not set the TSmn bit of the slave channel to 1.	Master channel starts counting.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers by slave channel can be changed.	Master channel loads the value of the TDRmn register to timer count register mn (TCRmn) by the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1), and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and the counter stops until the next start trigger detection. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit.	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

6.9.2 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRmp (slave)}/{Set value of TDRmn (master) + 1} × 100
0% output: Set value of TDRmp (slave) = 0000H
100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, an interrupt (INTTMmn) is output, the value set to timer data register mn (TDRmn) is loaded to timer count register mn (TCRmn), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTMmn is output, the value of the TDRmn register is loaded again to the TCRmn register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TTmn) of timer channel stop register m (TTm) is set to 1.

If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TOmp) cycle.

The slave channel operates in one-count mode. By using INTTMmn from the master channel as a start trigger, the TCRmp register loads the value of the TDRmp register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTMmp and waits until the next start trigger (INTTMmn from the master channel) is generated.

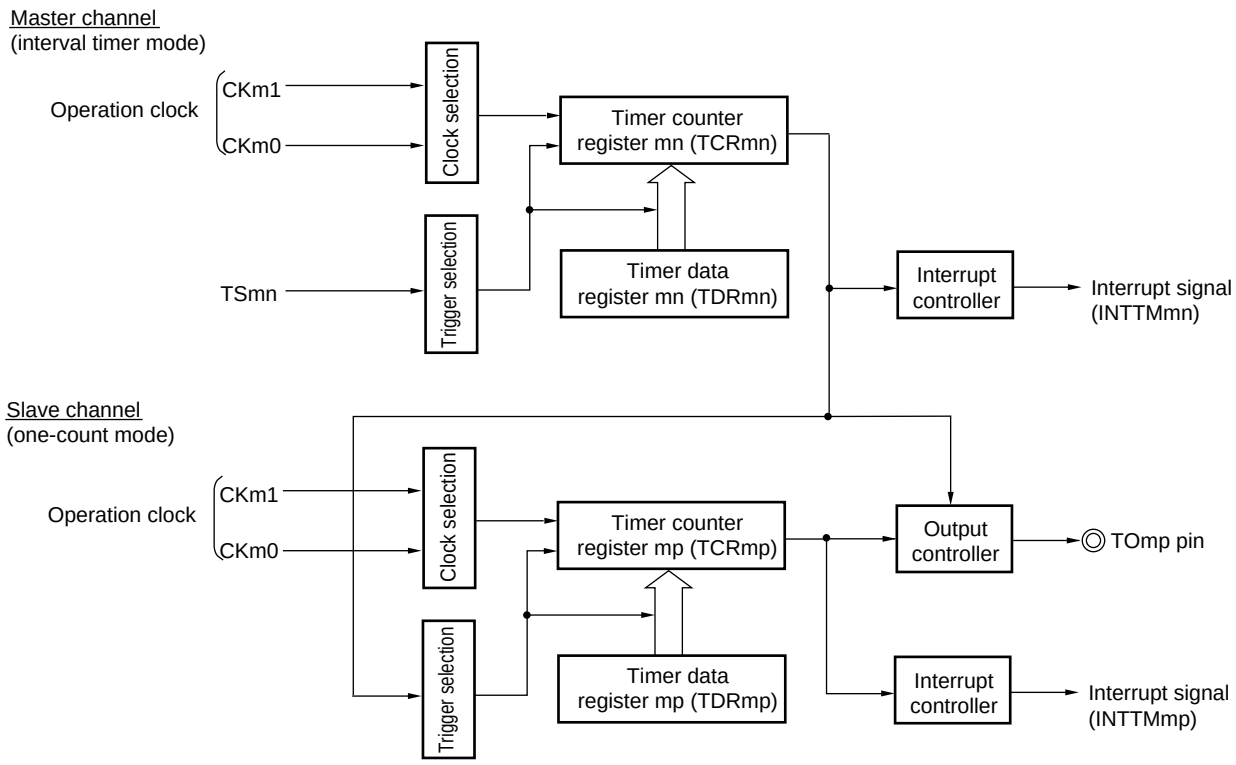
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.

PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

Caution **To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.**

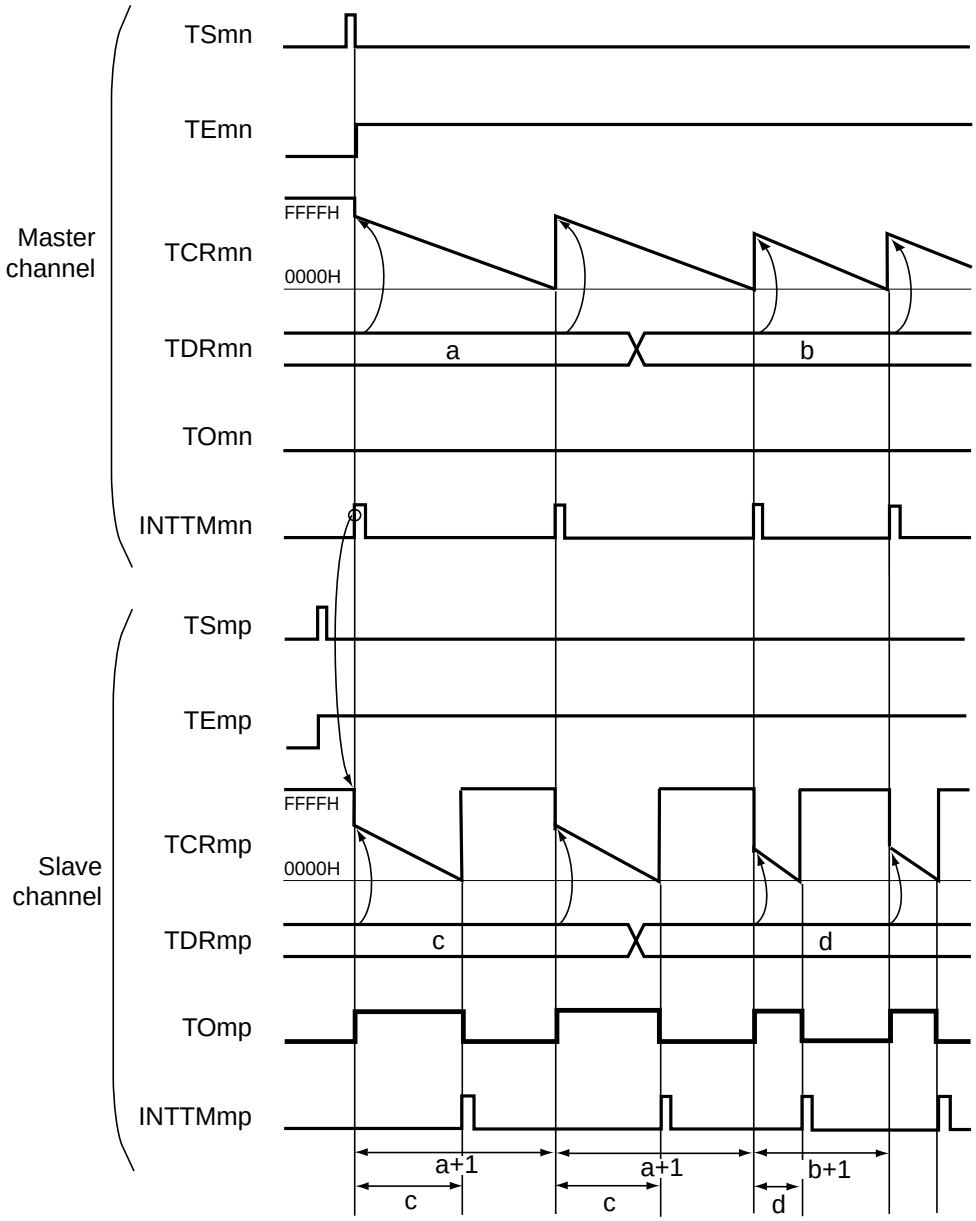
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-68. Block Diagram of Operation as PWM Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

Figure 6-69. Example of Basic Timing of Operation as PWM Function



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 6-70. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MASTER mn[Note]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of the MASTERmn bit (channel 2)
1: Master channel.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

Figure 6-71. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMRmp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmn bit (channel 2)
0: Slave channel
Setting of SPLITmp bit (channels 1, 3)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p
TOm	TOmp 1/0

0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p
TOEm	TOEmp 1/0

0: Stops the TOmp output operation by counting operation.
1: Enables the TOmp output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit p
TOLm	TOLmp 1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit p
TOMm	TOMmp 1

1: Sets the slave channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number ($n < p \leq 3$)

Figure 6-72. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

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

Figure 6-72. Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn), and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel, the value of the TDRmp register is loaded to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit.	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n < p ≤ 3)

6.9.3 Operation as multiple PWM output function

By extending the PWM function and using multiple slave channels, many PWM waveforms with different duty values can be output.

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 TDRmn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

Timer count register mn (TCRmn) of the master channel operates in the interval timer mode and counts the periods.

The TCRmp register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

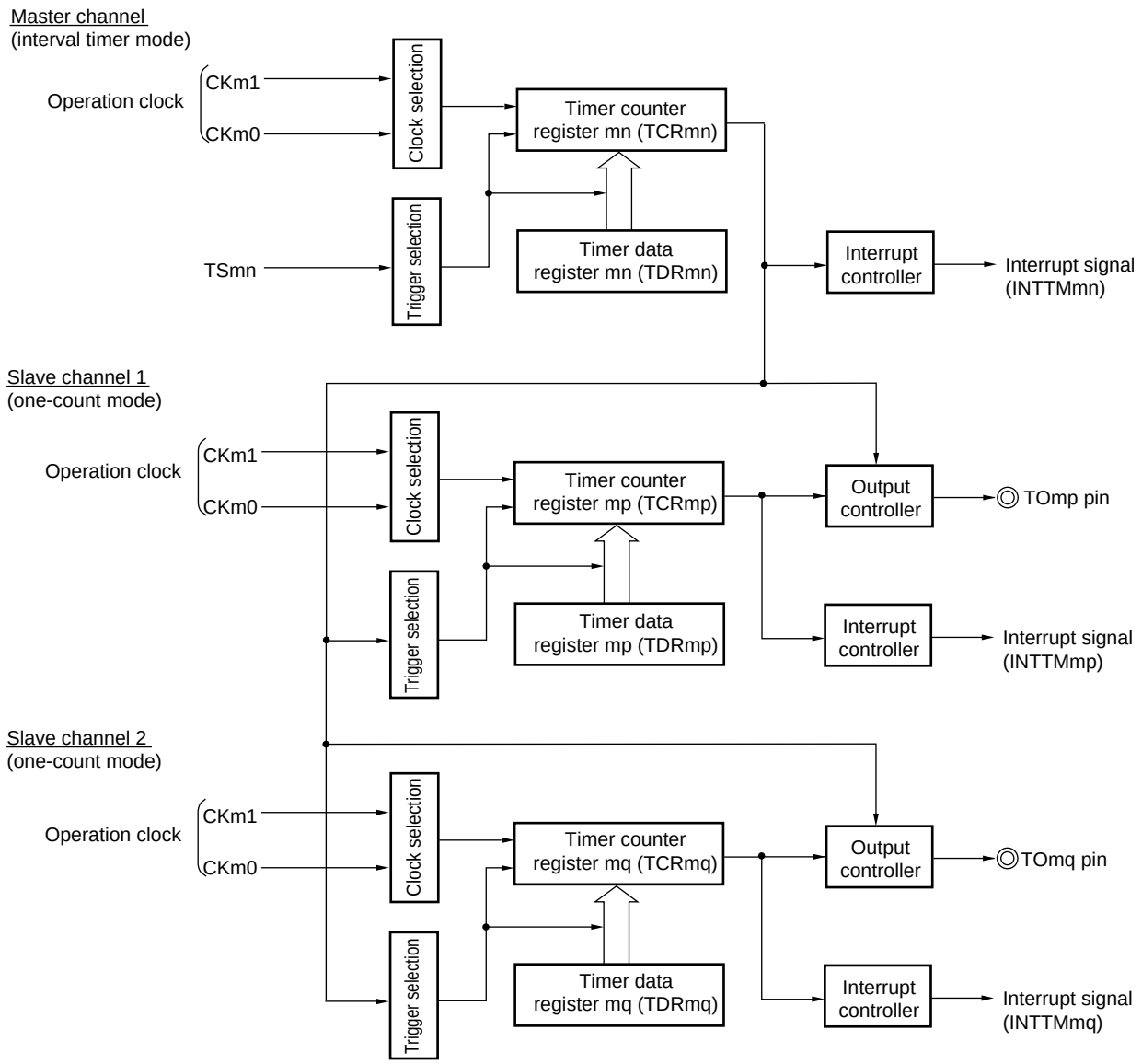
In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 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.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1, write access is necessary at least twice. Since the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to the TDRmq register of the slave channel 2).

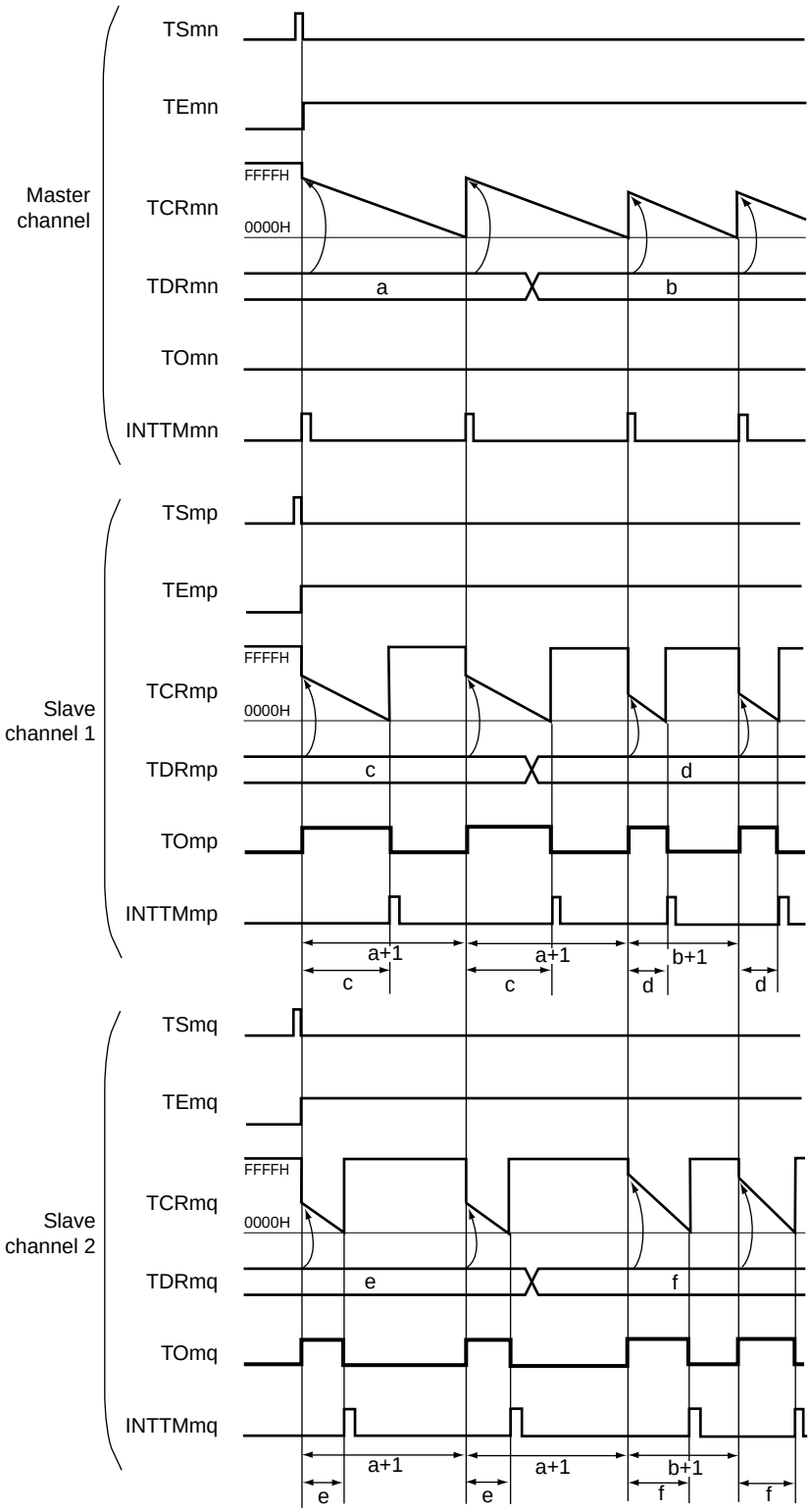
Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number , q: Slave channel number
n < p < q ≤ 3 (Where p and q are consecutive integers greater than n)

Figure 6-73. Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)



Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number , q: Slave channel number
n < p < q ≤ 3 (Where p and q are consecutive integers greater than n)

Figure 6-74. Example of Basic Timing of Operation as Multiple PWM Output Function (Output two types of PWMs) (1/2)



(**Remark** is listed on the next page.)

Figure 6-74. Example of Basic Timing of Operation as Multiple PWM Output Function (Output two types of PWMs) (2/2)

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 3 (Where p and q are consecutive integers greater than n)

2. TSmn, TSmp, TSmq: Bit n, p, q of timer channel start register m (TSm)
TEmn, TEmp, TEmq: Bit n, p, q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq: Timer count registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq: TOmn, TOmp, and TOmq pins output signal

Figure 6-75. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn		STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn 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
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Remark m: Unit number (m = 0), n: Channel number (n = 0)

Figure 6-76. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)

(a) Timer mode register mp, mq (TMRmp, TMRmq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1
TMRmq	CKSmq1	CKSmq0		CCSmq	M/S Note	STSmq2	STSmq1	STSmq0	CISmq1	CISmq0			MDmq3	MDmq2	MDmq1	MDmq0
	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 TImp and TImq pins input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp and MASTERmq bits (channel 2)
0: Independent channel operation function.

Setting of SPLITmp and SPLITmq bits (channels 1, 3)
1: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p, q.
10B: Selects CKm1 as operation clock of channel p, q.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit q	Bit p
TOm	TOmq	TOmp
	1/0	1/0

0: Outputs 0 from TOmp or TOmq.
1: Outputs 1 from TOmp or TOmq.

(c) Timer output enable register m (TOEm)

	Bit q	Bit p
TOEm	TOEmq	TOEmp
	1/0	1/0

0: Stops the TOmp or TOmq output operation by counting operation.
1: Enables the TOmp or TOmq output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit q	Bit p
TOLm	TOLmq	TOLmp
	1/0	1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit q	Bit p
TOMm	TOMmq	TOMmp
	1	1

1: Sets the slave channel output mode.

Note TMRm1, TMRm3: SPLITmp, SPLIT0q bit
TMRm2: MASTERmp, MASTERmq bit

Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number, q: Slave channel number
$n < p < q \leq 3$ (Where p and q are consecutive integers greater than n)

Figure 6-77. Operation Procedure When Multiple PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp, 0q (TMRmn, TMRmp, TMRmq) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp and TDRmq registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode). Clears the TOLmp and TOLmq bits to 0. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. →	The TOmp and TOmq pins go into Hi-Z output state. The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq. →	TOmp and TOmq do not change because channels stop operating.
	Clears the port register and port mode register to 0. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.

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

Figure 6-77. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	(Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.) The TSmn bit (master), and TSmp and TSmq (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp, TEmq = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. The TSRmn, TSRmp, and TSR0q registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn) and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel 1, the values of the TDRmp register are transferred to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDRmq register are transferred to TCRmq regster, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	TEmn, TEmp, TEmq = 0, and count operation stops. The TCRmn, TCRmp, and TCRmq registers hold count value and stop. The TOmp and TOmq output are not initialized but hold current status.
	The TOEmp and TOEmq bits of slave channels are cleared to 0 and value is set to the TOmp and TOmq bits.	The TOmp and TOmq pins output the TOmp and TOmq set levels.
TAU stop	To hold the TOmp and TOmq pin output levels Clears the TOmp and TOmq bits to 0 after the value to be held is set to the port register. When holding the TOmp and TOmq pin output levels are not necessary Setting not required	The TOmp and TOmq pin output levels are held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number, q: Slave channel number
$n < p < q \leq 3$ (Where p and q are consecutive integers greater than n)

6.10 Cautions When Using Timer Array Unit

6.10.1 Cautions when using timer output

Pins may be assigned multiplexed timer output and other alternate functions. The assignment depends on the product. If you intend to use a timer output, set the outputs from all other multiplexed pin functions to their initial values.

For details, see **4.5 Settings of Port Related Register When Using Alternate Function**.

CHAPTER 7 REAL-TIME CLOCK

7.1 Functions of Real-time Clock

The real-time clock has the following features.

- Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
- Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: week, hour, minute)
- Pin output function of 1 Hz (48-pin product only)

The real-time clock interrupt signal (INTRTC) can be utilized for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

Caution The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock. When the low-speed oscillation clock (f_{IL}) is selected, only the constant-period interrupt function is available. The 32-pin product has the constant-period interrupt function only, because this product has no subsystem clock.
However, the constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.

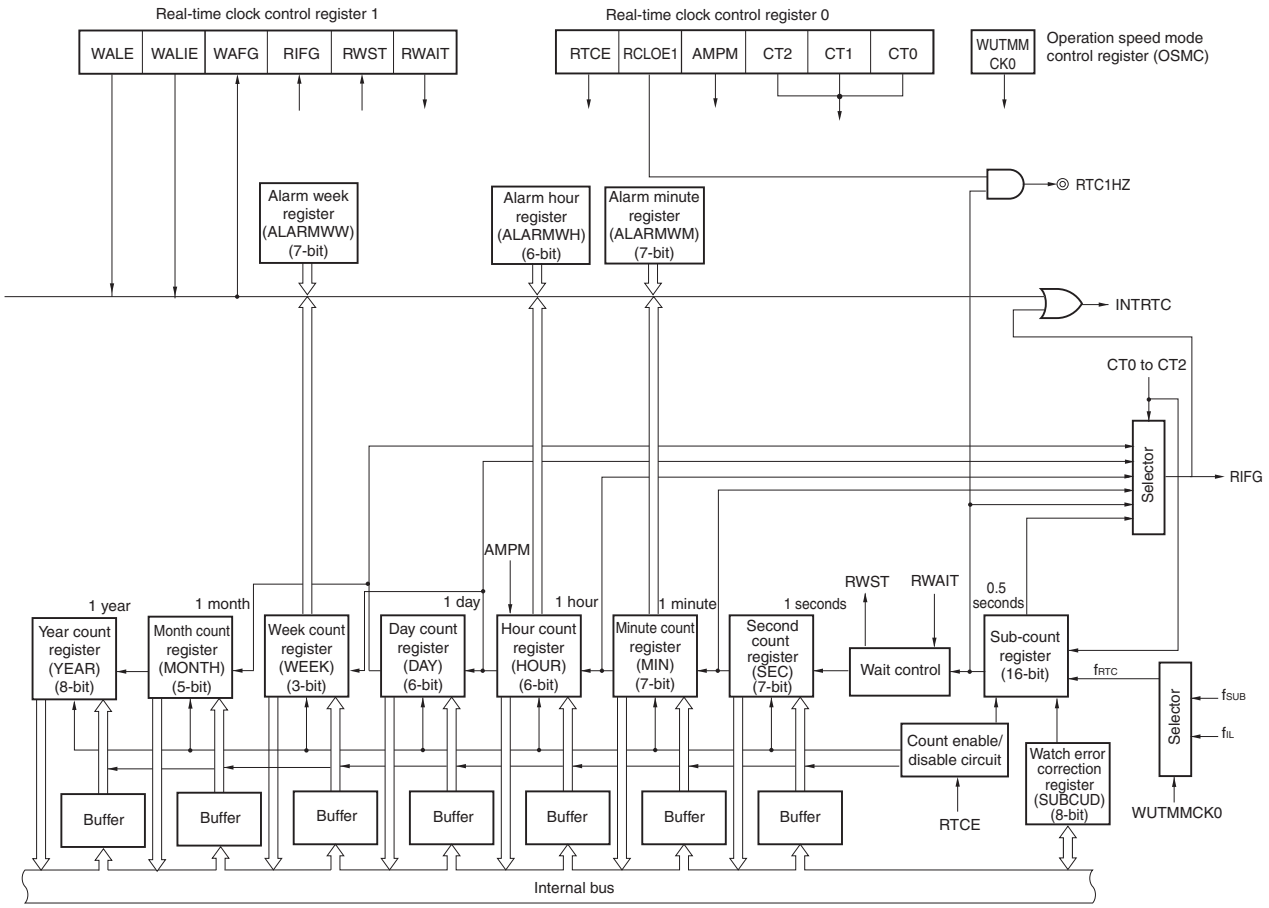
7.2 Configuration of Real-time Clock

The real-time clock includes the following hardware.

Table 7-1. Configuration of Real-time Clock

Item	Configuration
Counter	Internal counter (16-bit)
Control registers	Peripheral enable register 0 (PER0)
	Subsystem clock supply mode control register (OSMC)
	Real-time clock control register 0 (RTCC0)
	Real-time clock control register 1 (RTCC1)
	Second count register (SEC)
	Minute count register (MIN)
	Hour count register (HOUR)
	Day count register (DAY)
	Week count register (WEEK)
	Month count register (MONTH)
	Year count register (YEAR)
	Watch error correction register (SUBCUD)
	Alarm minute register (ALARMWM)
	Alarm hour register (ALARMWH)
	Alarm week register (ALARMWW)

Figure 7-1. Block Diagram of Real-time Clock



Caution The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock. When the low-speed oscillation clock (f_{IL}) is selected, only the constant-period interrupt function is available. The 32-pin product has the constant-period interrupt function only, because this product has no subsystem clock.

However, the constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.

7.3 Registers Controlling Real-time Clock

The real-time clock is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- Real-time clock control register 0 (RTCC0)
- Real-time clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Watch error correction register (SUBCUD)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm week register (ALARMWW)
- Port mode register 3 (PM3)
- Port register 3 (P3)

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 a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the real-time clock is used, be sure to set bit 7 (RTCEN) 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 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	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

RTCEN	Control of real-time clock (RTC) and 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. • The real-time clock (RTC) and 12-bit interval timer are in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer can be read and written.

Cautions 1. When using the real-time clock, first set the RTCEN bit to 1 and then set the following registers, while oscillation of the count clock (f_{RTC}) is stable. If RTCEN = 0, writing to the control registers of the real-time clock is ignored, and, even if the registers are read, only the default values are read (except for the subsystem clock supply mode control register (OSMC), port mode register 3 (PM3), port register 3 (P3)).

- **Real-time clock control register 0 (RTCC0)**
- **Real-time clock control register 1 (RTCC1)**
- **Second count register (SEC)**
- **Minute count register (MIN)**
- **Hour count register (HOUR)**
- **Day count register (DAY)**
- **Week count register (WEEK)**
- **Month count register (MONTH)**
- **Year count register (YEAR)**
- **Watch error correction register (SUBCUD)**
- **Alarm minute register (ALARMWM)**
- **Alarm hour register (ALARMWH)**
- **Alarm week register (ALARMWW)**

2. The subsystem clock supply to peripheral functions other than the real-time clock and 12-bit interval timer can be stopped in STOP mode or HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to 1. In this case, set the RTCEN bit of the PER0 register to 1 and the other bits (bits 0 to 6) to 0.

3. Be sure to clear the bits 1, 3, 6 to 0.

7.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the real-time clock operation clock (f_{RTC}).

In addition, by stopping clock functions that are unnecessary, the RTCLPC bit can be used to reduce power consumption. For details about setting the RTCLPC bit, see **CHAPTER 5 CLOCK GENERATOR**.

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 Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

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

WUTMMCK0	Selection of operation clock (f_{RTC}) for real-time clock and 12-bit interval timer.
0	Subsystem clock (f_{SUB})
1	Low-speed on-chip oscillator clock (f_{IL})

Caution The count of year, month, week, day, hour, minutes and second can only be performed when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of the real-time clock. When the low-speed oscillation clock (f_{IL}) is selected, only the constant-period interrupt function is available. The 32-pin product has the constant-period interrupt function only, because this product has no subsystem clock.
However, the constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.

7.3.3 Real-time clock control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time clock operation, control the RTC1HZ pin, and set a 12- or 24-hour system and the constant-period interrupt function.

The RTCC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-4. Format of Real-time Clock Control Register 0 (RTCC0)

Address: FFF9DH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1 Note	0	AMPM	CT2	CT1	CT0

RTCE	Real-time clock operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz).
1	Enables output of the RTC1HZ pin (1 Hz).

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system

- Rewrite the AMPM bit value after setting the RWAIT bit (bit 0 of real-time clock control register 1 (RTCC1)) to 1. If the AMPM bit value is changed, the values of the hour count register (HOUR) change according to the specified time system.
- Table 7-2 shows the displayed time digits that are displayed.

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)

When changing the values of the CT2 to CT0 bits while the counter operates (RTCE = 1), rewrite the values of the CT2 to CT0 bits after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of the CT2 to CT0 bits, enable interrupt servicing after clearing the RIFG and RTCIF flags.

Note Set the RCLOE1 bit to 0 in the 32-pin products.

Cautions 1. Do not change the value of the RCLOE1 bit when RTCE = 1.

2. 1 Hz is not output even if RCLOE1 is set to 1 when RTCE = 0.

Remark ×: don't care

7.3.4 Real-time clock control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

The RTCC1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-5. Format of Real-time Clock Control Register 1 (RTCC1) (1/2)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of real-time clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm week register (ALARMWW)), set match operation to be invalid (“0”) for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1” one cycle of f_{RTC} after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

Figure 7-5. Format of Real-time Clock Control Register 1 (RTCC1) (2/2)

RIFG	Constant-period interrupt status flag
0	Constant-period interrupt is not generated.
1	Constant-period interrupt is generated.
This flag indicates the status of generation of the constant-period interrupt. When the constant-period interrupt is generated, it is set to “1”. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

RWST	Wait status flag of real-time clock
0	Counter is operating.
1	Mode to read or write counter value
This status flag indicates whether the setting of the RWAIT bit is valid. Before reading or writing the counter value, confirm that the value of this flag is 1.	

RWAIT	Wait control of real-time clock
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value
This bit controls the operation of the counter. Be sure to write “1” to it to read or write the counter value. As the internal counter (16-bit) is continuing to run, complete reading or writing within one second and turn back to 0. When reading or writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt. When RWAIT = 1, it takes up to one cycle of f_{RTC} until the counter value can be read or written (RWST = 1).[Notes 1, 2] When the internal counter (16-bit) overflowed while RWAIT = 1, it keeps the event of overflow until RWAIT = 0, then counts up. However, when it wrote a value to second count register, it will not keep the overflow event.	

Notes 1. When setting RWAIT=1 during 1 operating clock (f_{RTC}), after setting RTCE=1, it may take two clock time of the operation clock (f_{RTC}), until RWST bit is set to “1”.

2. When setting RWAIT=1 during 1 operating clock (f_{RTC}), after returning from a stand-by (HALT mode, STOP mode and SNOOZE mode), it may take two clock time of the operation clock (f_{RTC}), until RWST bit is set to “1”.

Caution If writing is performed to the RTCC1 register with a 1-bit manipulation instruction, the RIFG flag and WAFG flag may be cleared. Therefore, to perform writing to the RTCC1 register, be sure to use an 8-bit manipulation instruction. To prevent the RIFG flag and WAFG flag from being cleared during writing, disable writing by setting 1 to the corresponding bit. If the RIFG flag and WAFG flag are not used and the value may be changed, the RTCC1 register may be written by using a 1-bit manipulation instruction.

Remarks 1. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

2. The internal counter is cleared when the second count register (SEC) is written.

7.3.5 Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the sub-count register overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-6. Format of Second Count Register (SEC)

Address: FFF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Remark The internal counter is cleared when the second count register (SEC) is written.

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 7.4.3 Reading/writing real-time clock.

7.3.6 Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Even if the second count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-7. Format of Minute Count Register (MIN)

Address: FFF93H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 7.4.3 Reading/writing real-time clock.

7.3.7 Hour count register (HOUR)

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of real-time clock control register 0 (RTCC0).

If the AMPM bit value is changed, the values of the HOUR register change according to the specified time system.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Figure 7-8. Format of Hour Count Register (HOUR)

Address: FFF94H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Cautions 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 7.4.3 Reading/writing real-time clock.

Table 7-2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 7-2. Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 1)	
Time	HOUR Register	Time	HOUR Register
0	00H	12 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	12 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

The HOUR register value is set to 12-hour display when the AMPM bit is “0” and to 24-hour display when the AMPM bit is “1”.

In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

7.3.8 Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.

It counts up when the hour counter overflows.

This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code.

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

Reset signal generation clears this register to 01H.

Figure 7-9. Format of Day Count Register (DAY)

Address: FFF96H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 7.4.3 Reading/writing real-time clock.

7.3.9 Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays. It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Set a decimal value of 00 to 06 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-10. Format of Week Count Register (WEEK)

Address: FFF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Cautions 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 7.4.3 Reading/writing real-time clock.

7.3.10 Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code.

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

Reset signal generation clears this register to 01H.

Figure 7-11. Format of Month Count Register (MONTH)

Address: FFF97H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 7.4.3 Reading/writing real-time clock.

7.3.11 Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month count register (MONTH) overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTC} later. Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-12. Format of Year Count Register (YEAR)

Address: FFF98H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 7.4.3 Reading/writing real-time clock.

7.3.12 Watch error correction register (SUBCUD)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the internal counter (16-bit) to the second count register (SEC) (reference value: 7FFFH).

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

Reset signal generation clears this register to 00H.

Figure 7-13. Format of Watch Error Correction Register (SUBCUD)

Address: FFF99H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).
Writing to the SUBCUD register at the following timing is prohibited. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • When DEV = 1 is set: For a period of SEC = 00H	

F6	Setting of watch error correction value
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.
When (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. * is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100). Range of correction value: (when F6 = 0) 2, 4, 6, 8, … , 120, 122, 124 (when F6 = 1) –2, –4, –6, –8, … , –120, –122, –124	

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	–189.2 ppm to 189.2 ppm	–63.1 ppm to 63.1 ppm
Maximum excludes quantization error	± 1.53 ppm	± 0.51 ppm
Minimum resolution	± 3.05 ppm	± 1.02 ppm

Remark If a correctable range is –63.1 ppm or lower and 63.1 ppm or higher, set 0 to DEV.

7.3.13 Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.

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

Reset signal generation clears this register to 00H.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 7-14. Format of Alarm Minute Register (ALARMWM)

Address: FFF9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

7.3.14 Alarm hour register (ALARMWH)

This register is used to set hours of alarm.

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

Reset signal generation clears this register to 12H.

However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 7-15. Format of Alarm Hour Register (ALARMWH)

Address: FFF9BH After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

7.3.15 Alarm week register (ALARMWW)

This register is used to set date of alarm.

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

Reset signal generation clears this register to 00H.

Figure 7-16. Format of Alarm Week Register (ALARMWW)

Address: FFF9CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

7.3.16 Port mode register 3 (PM3)

The PM3 register can be set by a 1-bit or 8-bit manipulation instruction.

Reset signal generation sets this register to FFH.

When using the port 3 as the RTC1HZ pin for output of 1 Hz, set the PM30 bit to 0.

Figure 7-17. Format of Port Mode Register 3 (PM3)

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	1	1	PM31	PM30

7.3.17 Port register 3 (P3)

The P3 register can be set by a 1-bit or 8-bit manipulation instruction.

Reset signal generation sets this register to 00H.

When using the port 3 as 1-Hz output to the RTC1HZ pin, set the P30 bit to 0.

Figure 7-18. Format of Port Register 3 (P3)

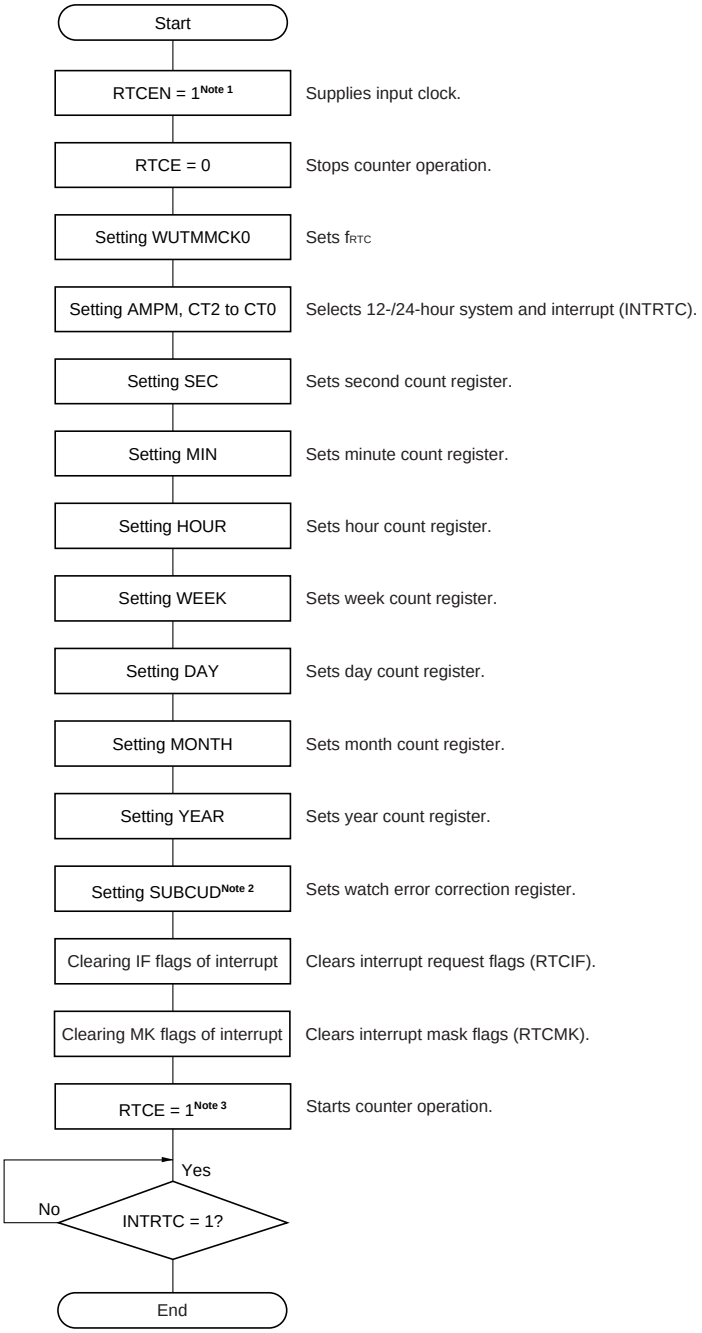
Address: FFF03H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
P3	0	0	0	0	0	0	P31	P30

7.4 Real-time Clock Operation

7.4.1 Starting operation of real-time clock

Figure 7-19. Procedure for Starting Operation of Real-time Clock



Notes 1. First set the RTCEN bit to 1, while oscillation of the count clock (f_{RTC}) is stable.

2. Set up the SUBCUD register only if the watch error must be corrected. For details about how to calculate the correction value, see **7.4.6 Example of watch error correction of real-time clock**.

3. Confirm the procedure described in **7.4.2 Shifting to HALT/STOP mode after starting operation** when shifting to HALT/STOP mode without waiting for INTRTC = 1 after RTCE = 1.

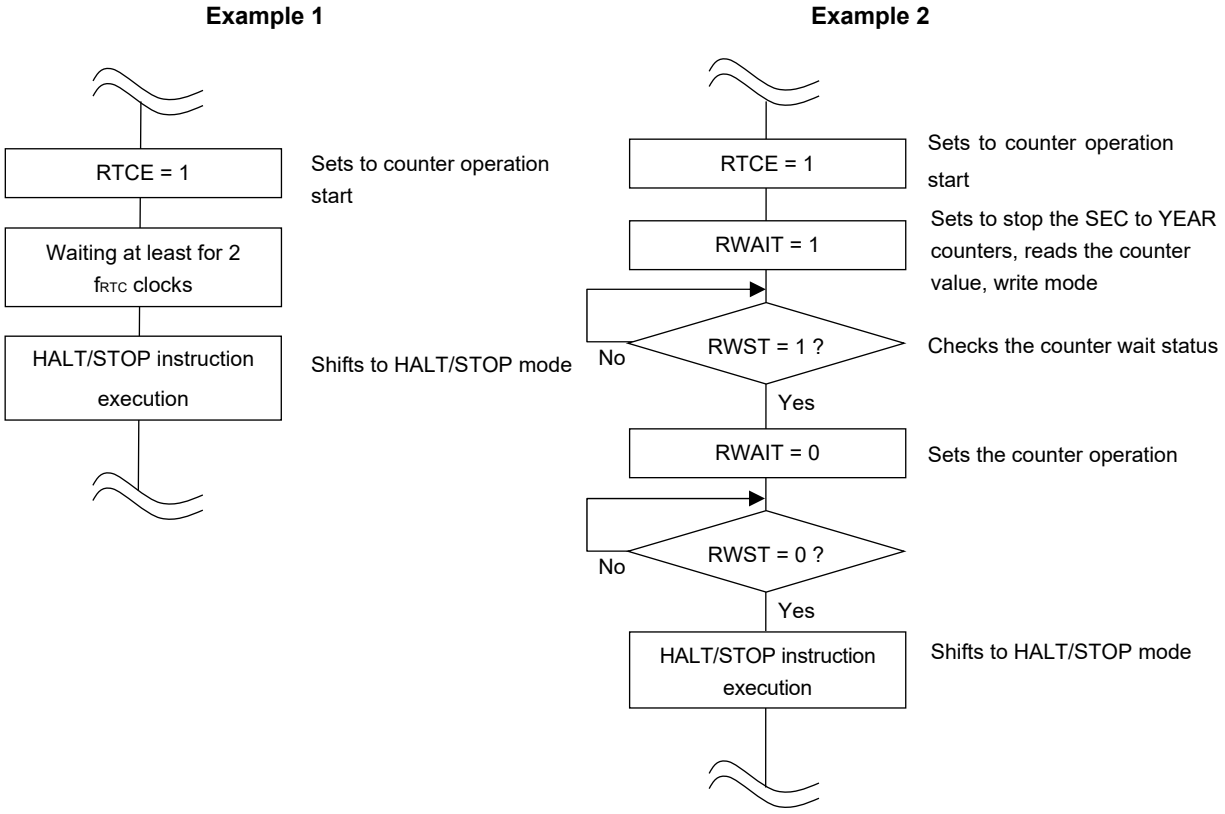
7.4.2 Shifting to HALT/STOP mode after starting operation

Perform one of the following processing when shifting to HALT/STOP mode immediately after setting the RTCE bit to 1.

However, after setting the RTCE bit to 1, this processing is not required when shifting to HALT/STOP mode after the INTRTC interrupt has occurred.

- Shifting to HALT/STOP mode when at least two count clocks (f_{RTC}) have elapsed after setting the RTCE bit to 1 (see **Figure 7-20**, **Example 1**).
- Checking by polling the RWST bit to become 1, after setting the RTCE bit to 1 and then setting the RWAIT bit to 1. Afterward, setting the RWAIT bit to 0 and shifting to HALT/STOP mode after checking again by polling that the RWST bit has become 0 (see **Figure 7-20**, **Example 2**).

Figure 7-20. Procedure for Shifting to HALT/STOP Mode After Setting RTCE bit to 1

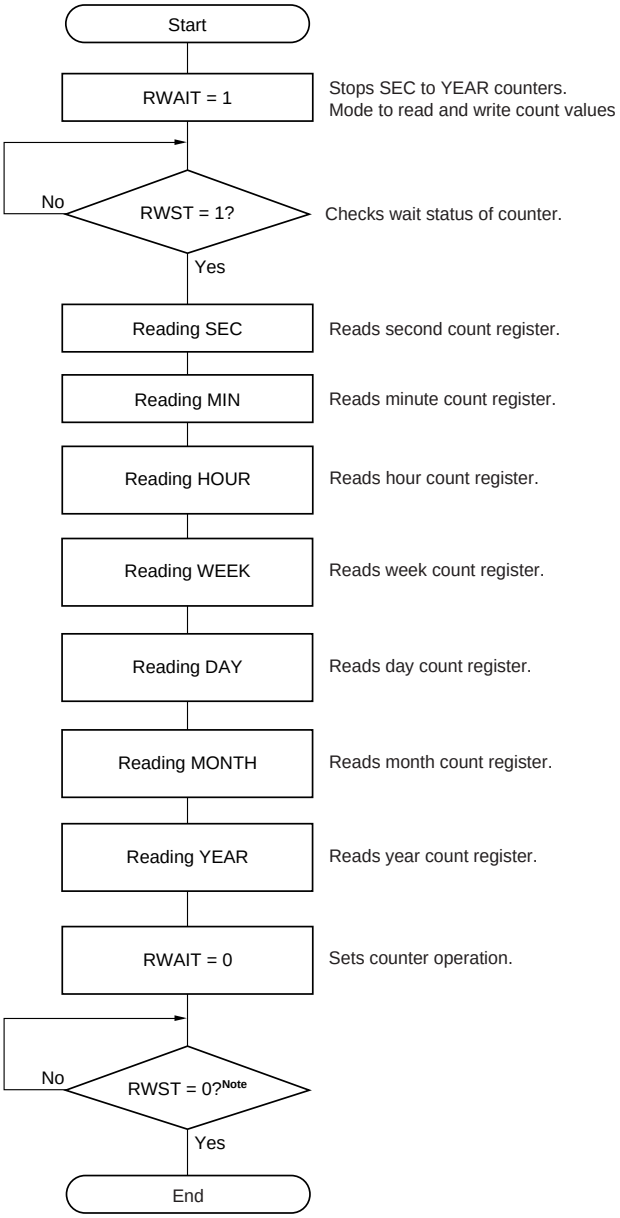


7.4.3 Reading/writing real-time clock

Read or write the counter after setting 1 to RWAIT first.

Set RWAIT to 0 after completion of reading or writing the counter.

Figure 7-21. Procedure for Reading Real-time Clock



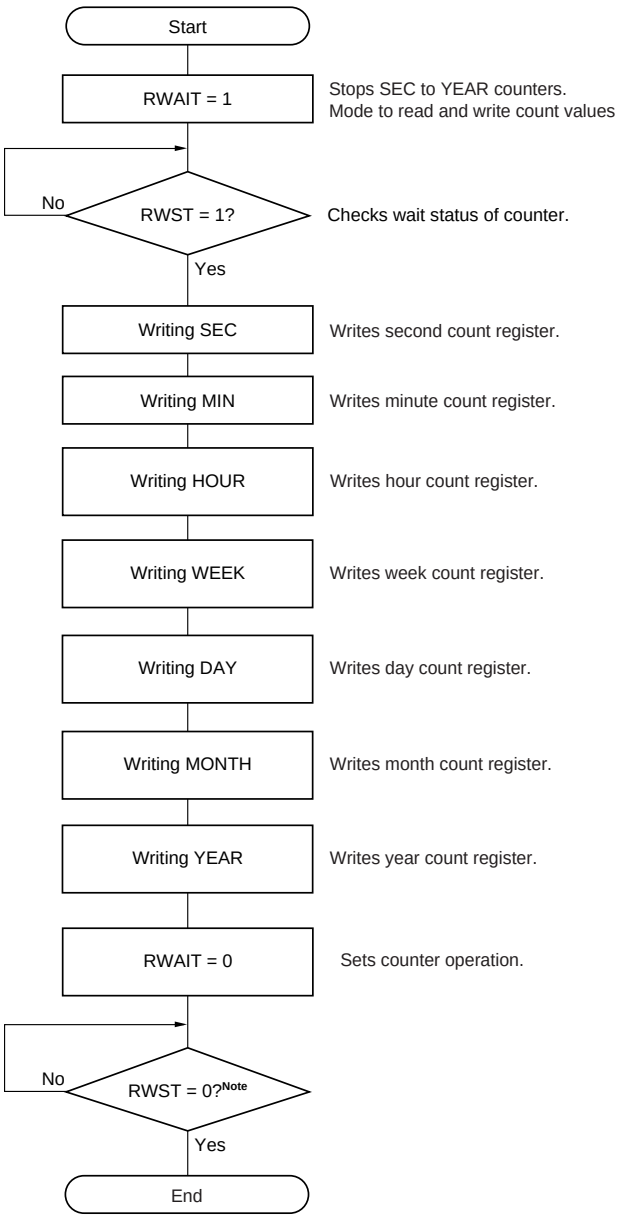
Note Be sure to confirm that RWST = 0 before setting HALT/STOP mode.

Caution **Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second. When reading to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.**

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence.

All the registers do not have to read and only some registers may be read.

Figure 7-22. Procedure for Writing Real-time Clock



Note Be sure to confirm that RWST = 0 before setting HALT/STOP mode.

Cautions 1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second. When writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.

2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR register while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the MIN register.

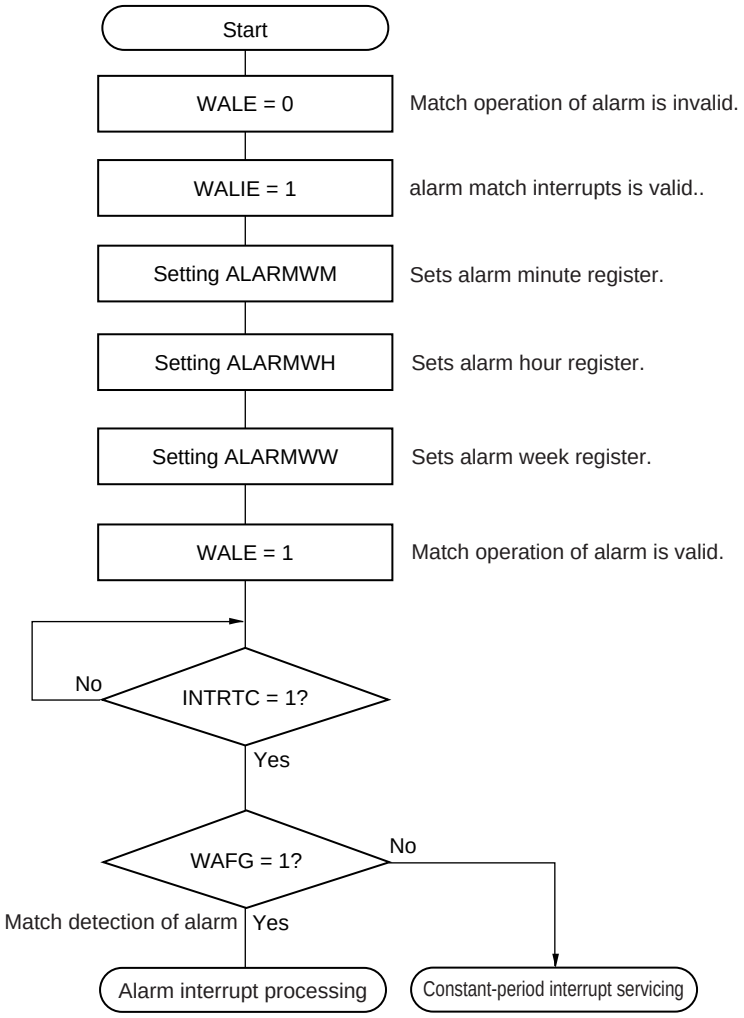
Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence.

All the registers do not have to be set and only some registers may be written.

7.4.4 Setting alarm of real-time clock

Set time of alarm after setting 0 to WALE (alarm operation invalid.) first.

Figure 7-23. Alarm processing Procedure

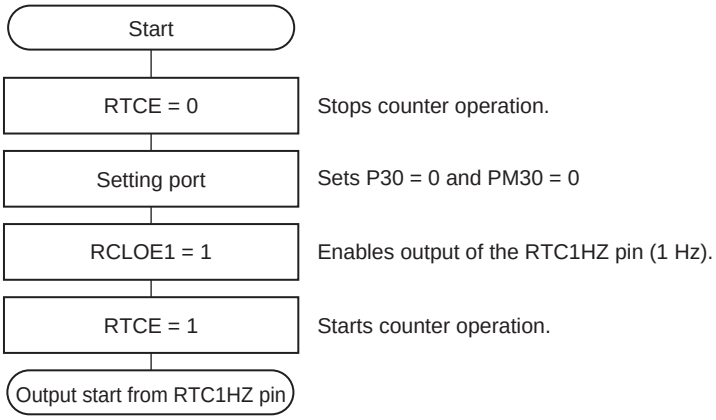


Remarks 1. The alarm week register (ALARMWW), alarm hour register (ALARMWH), and alarm week register (ALARMWW) may be written in any sequence.

2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

7.4.5 1 Hz output of real-time clock

Figure 7-24. 1 Hz Output Setting Procedure



Cautions 1. First set the RTCEN bit to 1, while oscillation of the count clock (f_{SUB}) is stable.

2. The 32-pin product does not have the 1Hz output function of the real-time clock.

7.4.6 Example of watch error correction of real-time clock

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the watch error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the internal counter (16-bit) is calculated by using the following expression.

Set the DEV bit to 0 when the correction range is –63.1 ppm or less, or 63.1 ppm or more.

(When DEV = 0)

Correction value[Note] = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3

(When DEV = 1)

Correction value[Note] = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60

Note The correction value is the watch error correction value calculated by using bits 6 to 0 of the watch error correction register (SUBCUD).

(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) – 1} × 2

(When F6 = 1) Correction value = – {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), watch error correction is not performed. "*" is 0 or 1. /F5 to /F0 are bit-inverted values (000011 when 111100).

Remarks 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

2. The oscillation frequency is the count clock (f_{RTC}).
It can be calculated from the output frequency of the RTC1HZ pin × 32768 when the watch error correction register is set to its initial value (00H).

3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

Correction example 1

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32.768 kHz from the PCLBUZ0 pin, or by outputting about 1 Hz from the RTC1HZ pin when the watch error correction register (SUBCUD) is set to its initial value (00H).

Note See 7.4.5 for the setting procedure of the RTC1Hz output, and see 9.4 for the setting procedure of outputting about 32 kHz from the PCLBUZ0 pin.

[Calculating the correction value]

(When the output frequency from the PCLBUZ0 pin is 32772.3 Hz)

Assume the target frequency to be 32768 Hz (32772.3 Hz–131.2 ppm) and DEV to be 0, because the correctable range of –131.2 ppm is –63.1 ppm or lower.

The expression for calculating the correction value when DEV is 0 is applied.

Correction value = Number of correction counts in 1 minute ÷ 3
= (Oscillation frequency ÷ target frequency – 1) × 32768 × 60 ÷ 3
= (32772.3 ÷ 32768 – 1) × 32768 × 60 ÷ 3
= 86

[Calculating the values to be set to (F6 to F0)]

(When the correction value is 86)

If the correction value is 0 or larger (when slowing), assume F6 to be 0.

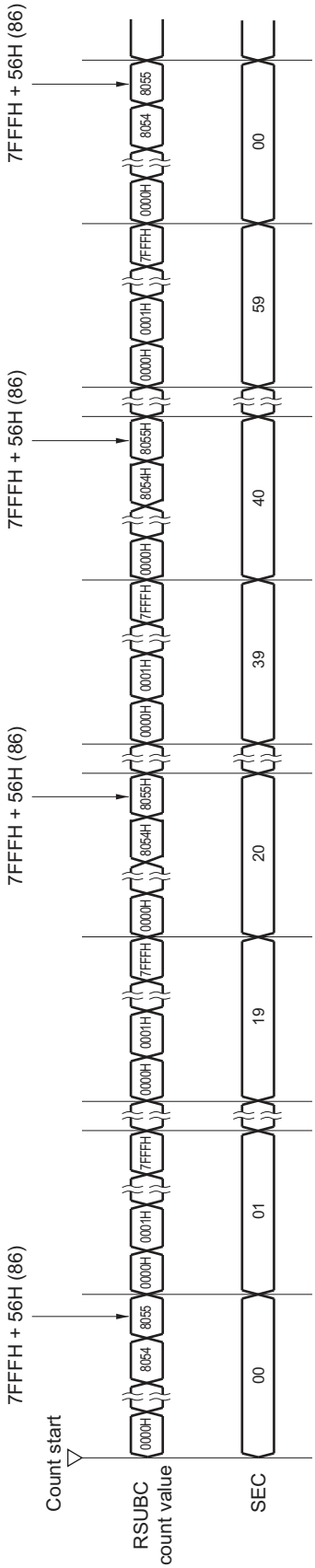
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

{ (F5, F4, F3, F2, F1, F0) – 1} × 2 = 86
(F5, F4, F3, F2, F1, F0) = 44
(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 0, 0)

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of the SUBCUD register: 0101100) results in 32768 Hz (0 ppm).

Figure 7-25 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 7-25. Correction Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)



Correction example 2

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 1 Hz from the RTC1HZ pin when the watch error correction register (SUBCUD) is set to its initial value (00H).

Note See **7.4.5 1 Hz output of real-time clock** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin.

[Calculating the correction value]

(When the output frequency from the RTC1HZ pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

Correction value = Number of correction counts in 1 minute
= (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60
= (32767.4 ÷ 32768 – 1) × 32768 × 60
= –36

[Calculating the values to be set to (F6 to F0)]

(When the correction value is –36)

If the correction value is 0 or less (when quickening), assume F6 to be 1.

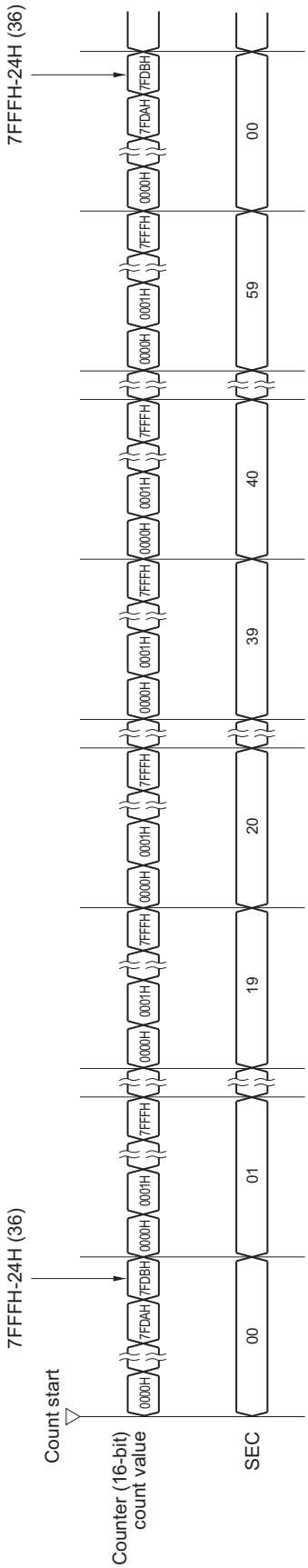
Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

– {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2	= –36
(/F5, /F4, /F3, /F2, /F1, /F0)	= 17
(/F5, /F4, /F3, /F2, /F1, /F0)	= (0, 1, 0, 0, 0, 1)
(F5, F4, F3, F2, F1, F0)	= (1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is –36 (bits 6 to 0 of the SUBCUD register: 1101110) results in 32768 Hz (0 ppm).

Figure 7-26 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 7-26. Correction Operation when (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



CHAPTER 8 12-BIT INTERVAL TIMER

8.1 Functions of 12-bit Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be utilized for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

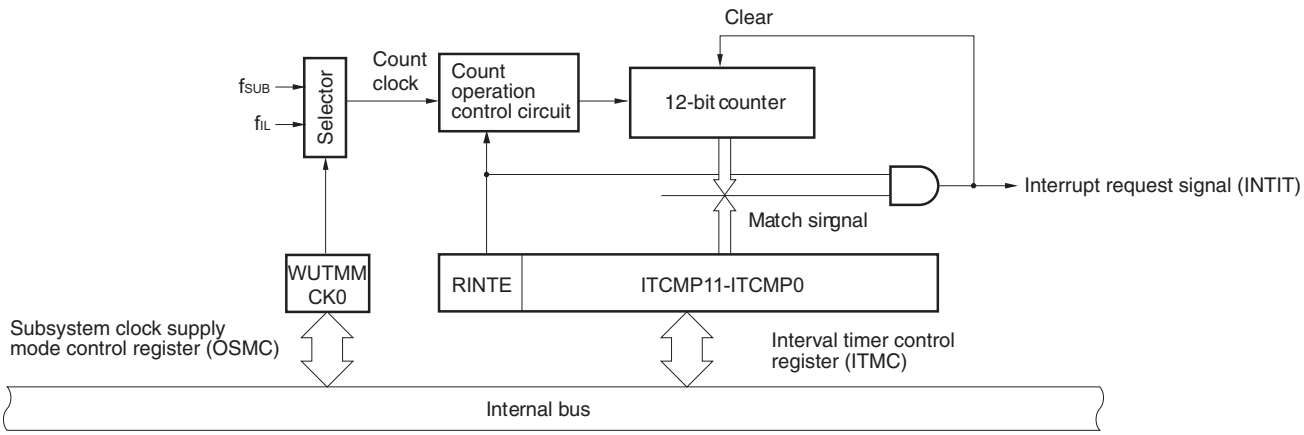
8.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 8-1. Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 0 (PER0)
	Subsystem clock supply mode control register (OSMC)
	Interval timer control register (ITMC)

Figure 8-1. Block Diagram of 12-bit Interval Timer



Caution The subsystem clock (f_{SUB}) is selectable as a count clock in the 48-pin products.

8.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- Interval timer control register (ITMC)

8.3.1 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 12-bit interval timer is used, be sure to set bit 7 (RTCEN) 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 8-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

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

RTCEN	Control of real-time clock (RTC) and 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer cannot be written. • The real-time clock (RTC) and 12-bit interval timer are in the reset status.
1	Enables input clock supply. • SFR used by the real-time clock (RTC) and 12-bit interval timer can be read and written.

Cautions 1. When using the 12-bit interval timer, be sure to first set the RTCEN bit to 1 and then set the interval timer control register (ITMC), while oscillation of the count clock is stable. If RTCEN = 0, writing to the registers controlling the 12-bit interval timer is ignored, and, even if the register is read, only the default value is read (except the subsystem clock supply mode control register (OSMC)).

2. Clock supply to peripheral functions other than the real-time clock and 12-bit interval timer can be stopped in STOP mode or HALT mode when the subsystem clock is used, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to 1.

3. Be sure to clear the bits 1, 3, 6 to 0.

8.3.2 Subsytem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the 12-bit interval timer operation clock or real-time clock.

In addition, by stopping clock functions that are unnecessary, the RTCLPC bit can be used to reduce power consumption. For details about setting the RTCLPC bit, see **CHAPTER 5 CLOCK GENERATOR**.

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

Reset signal generation clears this register to 00H.

Figure 8-3. Format of Subsytem Clock Supply Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

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

WUTMMCK0	Selection of operation clock for real-time clock and 12-bit interval timer.
0	Subsystem clock (f_{SUB})
1	Low-speed on-chip oscillator clock (f_{IL})

8.3.3 Interval timer control register (ITMC)

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.

The ITMC register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0FFFH.

Figure 8-4. Format of Interval Timer Control Register (ITMC)

Address: FFF90H After reset: 0FFFH R/W

Symbol	15	14	13	12	11 to 0
ITMC	RINTE	0	0	0	ITCMP11 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 prohibit

Example interrupt cycles when 001H or FFFH is specified for ITCMP11 to ITCMP0

- ITCMP11 to ITCMP0 = 001H, count clock: when f_{SUB} = 32.768 kHz
  1/32.768 [kHz] × (1 + 1) = 0.06103515625 [ms] ≅ 61.03 [μs]
- ITCMP11 to ITCMP0 = FFFH, count clock: when f_{SUB} = 32.768 kHz
  1/32.768 [kHz] × (4095 + 1) = 125 [ms]

Cautions 1. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from 0 to 1) again, clear the ITIF flag, and then enable the interrupt servicing.

2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.

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 RINTE = 0.
However, it is possible to change the settings of the ITCMP11 to ITCMP0 bits at the same time as when changing RINTE from 0 to 1 or 1 to 0.

8.4 12-bit Interval Timer Operation

8.4.1 12-bit interval timer operation timing

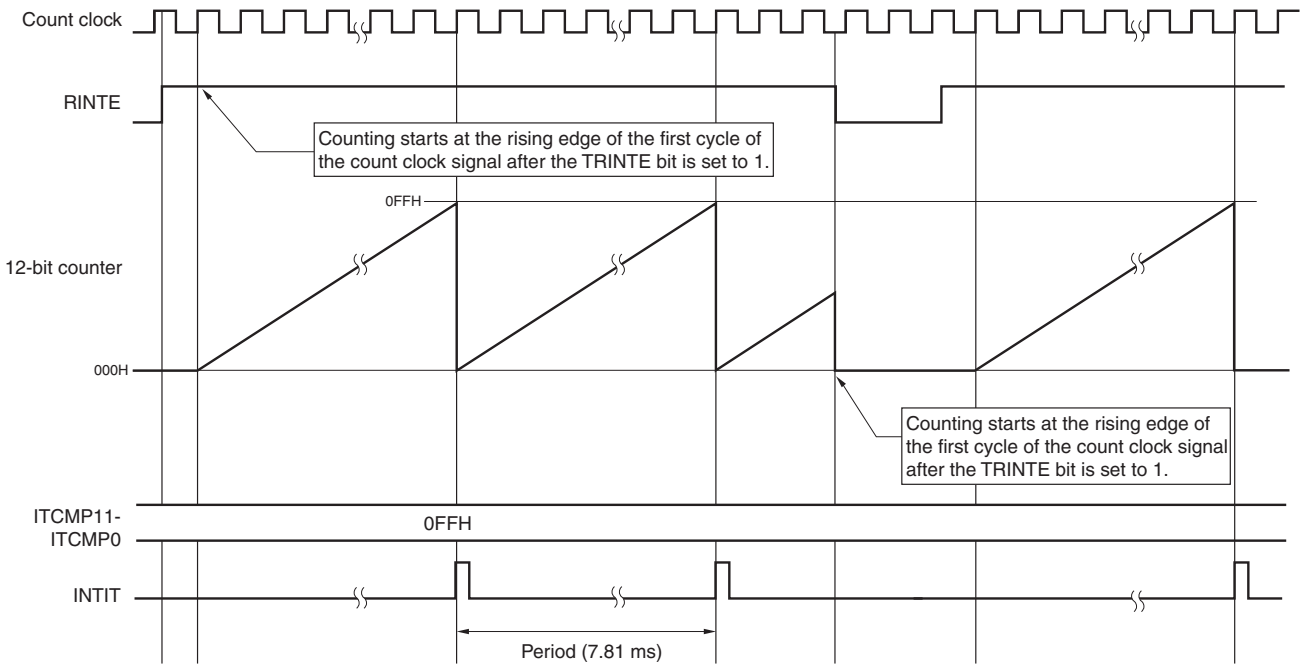
The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate an 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

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 as follows.

Figure 8-5. 12-bit Interval Timer Operation Timing (ITCMP11 to ITCMP0 = 0FFH, count clock: f_{SUB} = 32.768 kHz)

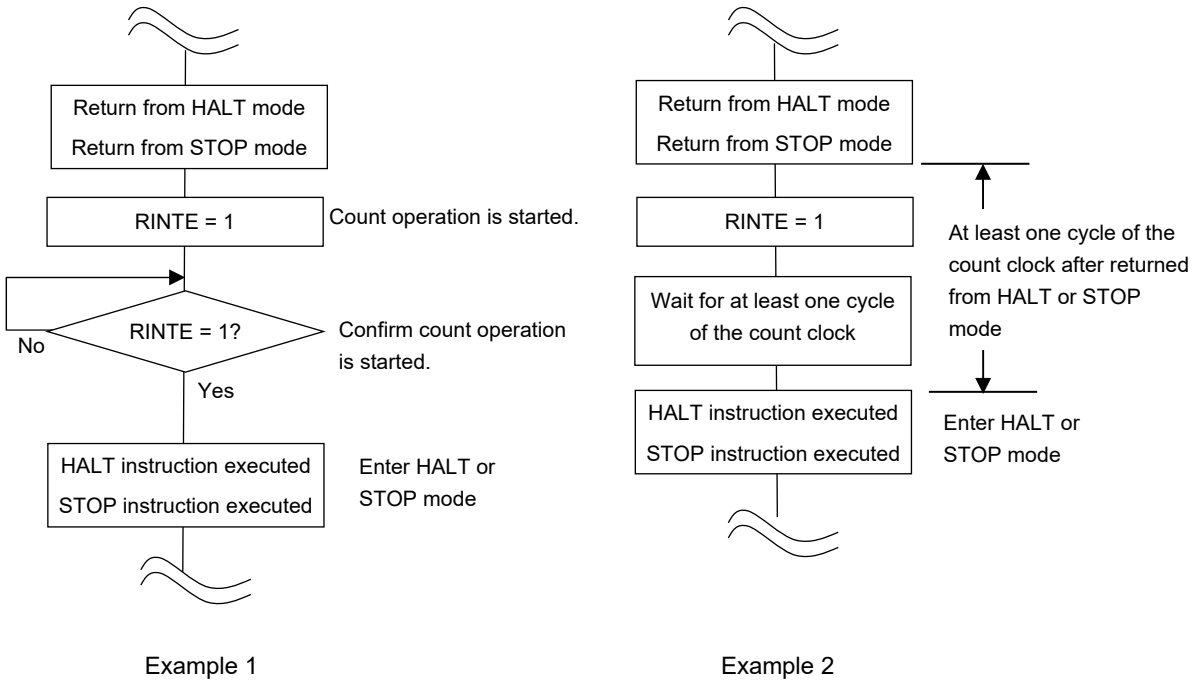


8.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

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 8-6).
- 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 8-6).

Figure 8-6. Procedure of entering to HALT or STOP mode after setting RINTE to 1



Example 1 Example 2

CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

Caution Most of the following descriptions in this chapter use the 48-pin product as an example.

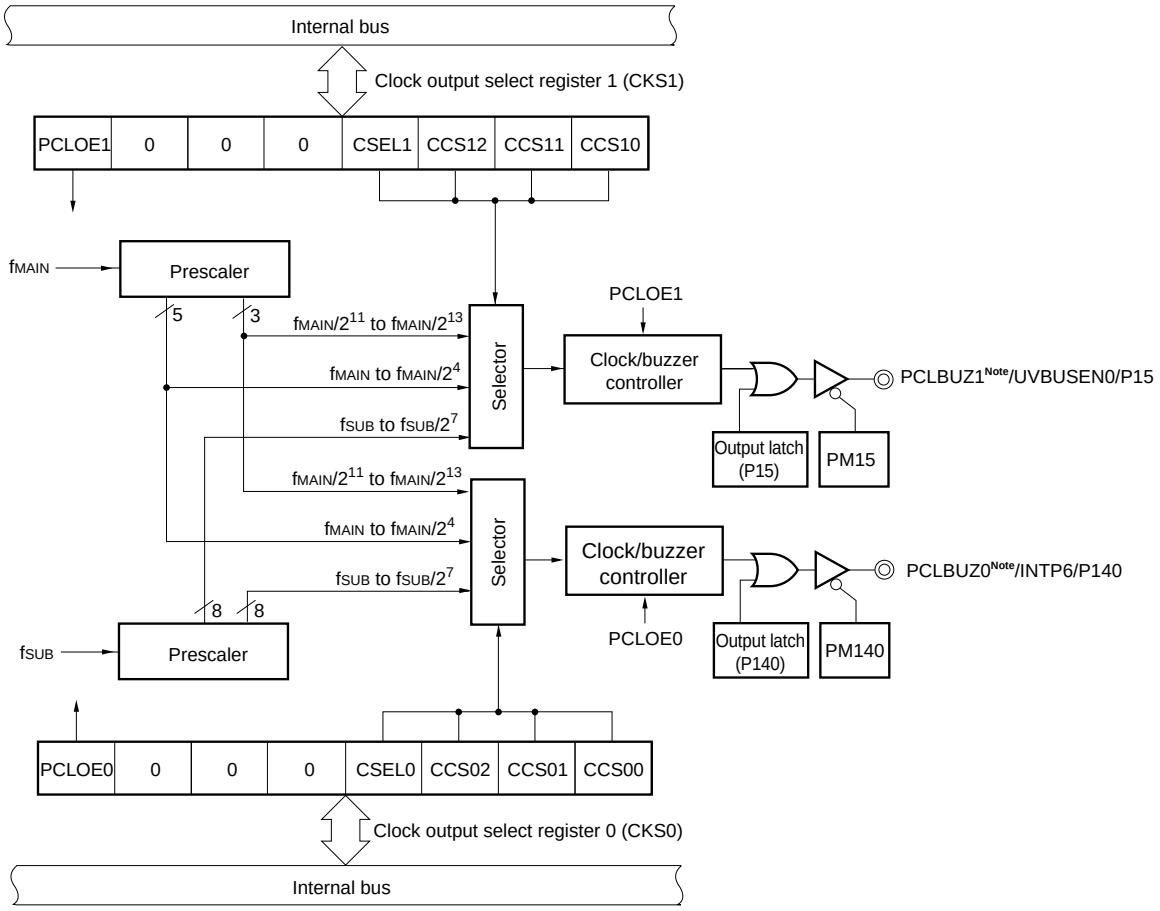
9.1 Functions of Clock Output/Buzzer Output Controller

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 PCLBUZn pin outputs a clock selected by clock output select register n (CKSn).
Figure 9-1 shows the block diagram of clock output/buzzer output controller.

Caution It is not possible to output the subsystem clock (f_{SUB}) from the PCLBUZn pin while the RTCLPC bit of the operation speed mode control register (OSMC), which controls the supply of the subsystem clock, is set to 1 and moreover while HALT mode is set with the subsystem clock (f_{SUB}) selected as CPU clock.

Remark n = 0, 1

Figure 9-1. Block Diagram of Clock Output/Buzzer Output Controller



Note For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to **30.4** and **31.4 AC Characteristics**.

Remark The clock output/buzzer output pins in above diagram shows the information of 48-pin product with PIOR0 = 0.
In other cases, the name of pins, output latches (Pxx) and PMxx should be read differently (xx = 15, 30, 31, 41, 70, 120 or 140).

9.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 9-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers n (CKSn) Port mode registers 1, 3, 4, 7, 12, 14 (PM1, PM3, PM4, PM7, PM12, PM14) Port registers 1, 3, 4, 7, 12, 14 (P1, P3, P4, P7, P12, P14)

9.3 Registers Controlling Clock Output/Buzzer Output Controller

The following registers are used to control the clock output/buzzer output controller.

- Clock output select registers n (CKSn)
- Port mode registers 1, 3, 4, 7, 12, 14 (PM1, PM3, PM4, PM7, PM12, PM14)
- Port register 1, 3, 4, 7, 12, 14 (P1, P3, P4, P7, P12, P14)

9.3.1 Clock output select registers n (CKSn)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZn), and set the output clock.

Select the clock to be output from the PCLBUZn pin by using the CKSn register.

The CKSn register are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 9-2. Format of Clock Output Select Register n (CKSn)

Address: FFFA5H (CKS0), FFFA6H (CKS1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn pin output enable/disable specification
0	Output disable (default)
1	Output enable

CSELn	CCSn2	CCSn1	CCSn0	PCLBUZn pin output clock selection				
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz	f_{MAIN} = 24 MHz
0	0	0	0	f_{MAIN}	5 MHz	10 MHz[Note]	Setting prohibited[Note]	Setting prohibited[Note]
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz[Note]	12 MHz[Note]
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz	6 MHz[Note]
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	0	0	0	f_{SUB}	32.768 kHz			
1	0	0	1	$f_{SUB}/2$	16.384 kHz			
1	0	1	0	$f_{SUB}/2^2$	8.192 kHz			
1	0	1	1	$f_{SUB}/2^3$	4.096 kHz			
1	1	0	0	$f_{SUB}/2^4$	2.048 kHz			
1	1	0	1	$f_{SUB}/2^5$	1.024 kHz			
1	1	1	0	$f_{SUB}/2^6$	512 Hz			
1	1	1	1	$f_{SUB}/2^7$	256 Hz			

Note Use the output clock within a range of 16 MHz. See **30.4** and **31.4 AC Characteristics** for details.

Cautions 1. Change the output clock after disabling clock output (PCLOEn = 0).

2. To shift to STOP mode when the main system clock is selected (CSELn = 0), set PCLOEn = 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn = 1 can be set because the clock can be output while the RTCLPC bit of the operation speed mode control (OSMC) register (the bit which controls the supply of the subsystem clock) is set to 0 and moreover while STOP mode is set.

3. It is not possible to output the subsystem clock (f_{SUB}) from the PCLBUZn pin while the RTCLPC bit of the operation speed mode control register (OSMC), which controls the supply of the subsystem clock, is set to 1 and moreover while HALT mode is set with the subsystem clock (f_{SUB}) selected as CPU clock.

Remarks 1. n = 0, 1

2. f_{MAIN}: Main system clock frequency

f_{SUB}: Subsystem clock frequency

9.3.2 Registers controlling port functions of pins to be used for clock or buzzer output

Using a port pin for clock or buzzer output requires setting of the registers that control the port functions multiplexed on the target pin (port mode register (PMxx), port register (Pxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

Specifically, using a port pin with a multiplexed clock or buzzer output function (e.g. P140/INTP6/PCLBUZ0, P15/PCLBUZ1/UVBUSEN0) for clock or buzzer output, requires setting the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

Example: When P140/INTP6/PCLBUZ0 is to be used for clock or buzzer output
Set the PM140 bit of port mode register 14 to 0.
Set the P140 bit of port register 14 to 0.

9.4 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).
The PCLBUZ1 pin outputs a clock/buzzer selected by the clock output select register 1 (CKS1).

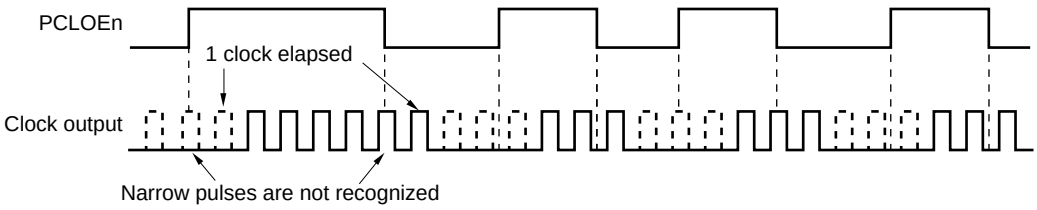
9.4.1 Operation as output pin

The PCLBUZn pin is output as the following procedure.

<1> Set 0 in the bit of the port mode register (PMxx) and port register (Px) which correspond to the port which has a pin used as the PCLBUZn pin.
<2> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled status).
<3> Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

Remarks 1. The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOEn bit) is switched. At this time, pulses with a narrow width are not output. Figure 9-4 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.
2. n = 0, 1

Figure 9-4. Timing of Outputting Clock from PCLBUZn Pin



9.5 Cautions of Clock Output/Buzzer Output Controller

When the main system clock is selected for the PCLBUZn output (CSEL = 0), if STOP mode is entered within 1.5 clock cycles output from the PCLBUZn pin after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.

CHAPTER 10 WATCHDOG TIMER

10.1 Functions of Watchdog Timer

The counting operation of the watchdog timer is set by the option byte (000C0H).

The watchdog timer operates on the low-speed on-chip oscillator clock (f_{IL}).

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
- If data is written to the WDTE register during a window close period

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 20 RESET FUNCTION**.

When 75% + 1/2/f_{IL} of the overflow time is reached, an interval interrupt can be generated.

10.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 10-1. Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

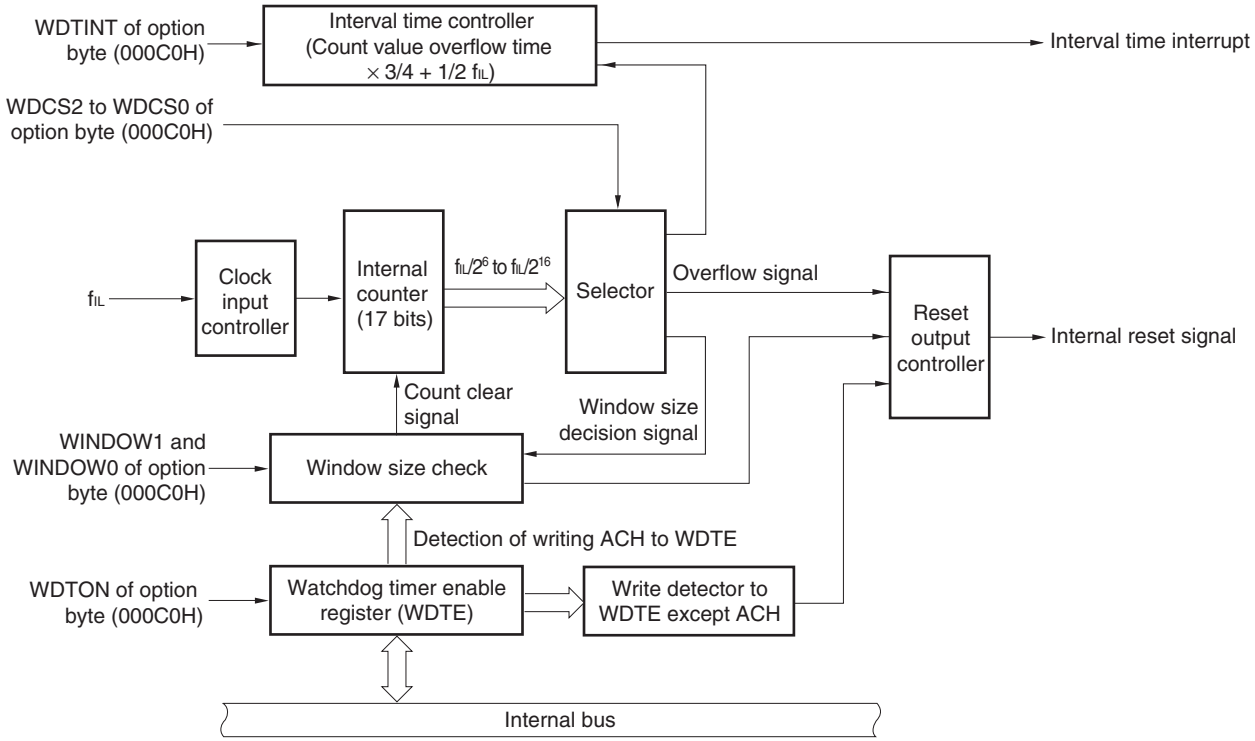
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 10-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
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 25 OPTION BYTE**.

Figure 10-1. Block Diagram of Watchdog Timer



Remark f_{IL}: Low-speed on-chip oscillator clock

10.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

10.3.1 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 9AH or 1AH[Note].

Figure 10-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH After reset: 9AH/1AH[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. After "ACH" is written to the WDTE register, an error of up to 2 clocks (f_{IL}) may occur before the watchdog timer is cleared.

3. The value read from the WDTE register is 9AH/1AH (this differs from the written value (ACH)).

10.4 Operation of Watchdog Timer

10.4.1 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 25**).

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 **10.4.2** and **CHAPTER 25**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **10.4.3** and **CHAPTER 25**).

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. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. 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. When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

2. If the watchdog timer is cleared by writing “ACH” to the WDTE register, the actual overflow time may be different from the overflow time set by the option byte by up to 2/f_{IL} seconds.

3. The watchdog timer can be cleared immediately before the count value overflows.

Cautions 4. The operation of the watchdog timer in the HALT and STOP and SNOOZE modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		
In SNOOZE mode		

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 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 and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

10.4.2 Setting overflow time of watchdog timer

Set the overflow 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) during the window open period before the overflow time.

The following overflow times can be set.

Table 10-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.89 ms) Note
1	1	0	$2^{14}/f_{IL}$ (949.79 ms) Note
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms) Note

Note Using the watchdog timer under the following conditions may lead to the generation of an interval interrupt (INTWDTI) after one cycle of the watchdog timer clock once the watchdog timer counter has been cleared.

Usage conditions that may lead to the generation of an interval interrupt:
- The overflow time of the watchdog timer is set to 213/fIL, 214/fIL, or 216/fIL.
- The interval interrupt is in use (the setting of the WDTINT bit of the relevant option byte is 1).
- ACH is written to the WDTE register (FFFABH) when the watchdog timer counter has reached or exceeded 75% of the overflow time.

This interrupt can be masked by clearing the watchdog timer counter through steps 1 to 5 below.
1. Set the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 1 before clearing the watchdog timer counter.
2. Clear the watchdog timer counter.
3. Wait for at least 80 μs.
4. Clear the WDTIIF bit of the interrupt request flag register (IF0L) to 0.
5. Clear the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 0.

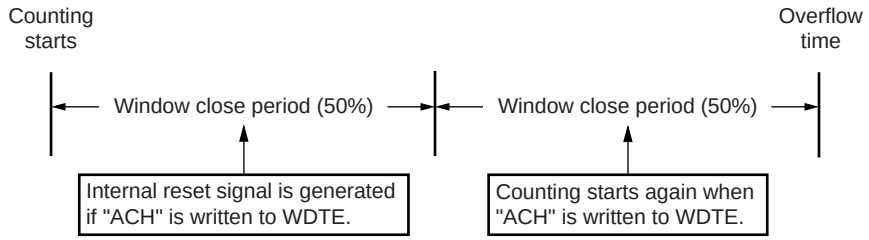
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

10.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 50%



Caution When data is written to the WDTE register for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period can be set is as follows.

Table 10-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	Setting prohibited
0	1	50%
1	0	75%
1	1	100%

<R> **Note** When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{IL}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)	1899.59 ms to 2570.04 ms

Caution When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.

Remark If the overflow time is set to $2^9/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 20.08 ms	0 to 10.04 ms	None
Window open time	20.08 to 29.68 ms	10.04 to 29.68 ms	0 to 29.68 ms

<When window open period is 50%>

- Overflow time:
  $2^9/f_{IL}$ (MAX.) = 2^9/17.25 kHz = 29.68 ms
- Window close time:
  0 to $2^9/f_{IL}$ (MIN.) × (1 – 0.5) = 0 to 2^9/12.75 kHz × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^9/f_{IL}$ (MIN.) × (1 – 0.5) to $2^9/f_{IL}$ (MAX.) = 2^9/12.75 kHz × 0.5 to 2^9/17.25 kHz = 20.08 to 29.68 ms

10.4.4 Setting watchdog timer interval interrupt

Depending on the setting of bit 7 (WDTINT) of an option byte (000C0H), an interval interrupt (INTWDTI) can be generated when 75% + 1/2f_{IL} of the overflow time is reached.

Table 10-5. Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is used.
1	Interval interrupt is generated when 75% + 1/2f_{IL} of overflow time is reached.

Caution When operating with the X1 oscillation clock 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 and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

Remark 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.

CHAPTER 11 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

	32-pin	48-pin
Analog input channels	8 ch (ANI0 to ANI4, ANI16, ANI17, ANI19)	9 ch (ANI0 to ANI7, ANI19)

Caution Most of the following descriptions in this chapter use the 48-pin product as an example.

11.1 Function of A/D Converter

The A/D converter is used to convert analog input signals into digital values, and is configured to control analog inputs, including up to 9 channels of A/D converter analog inputs (ANI0 to ANI7 and ANI19). 10-bit or 8-bit resolution can be selected by the ADTYP bit of the A/D converter mode register 2 (ADM2).

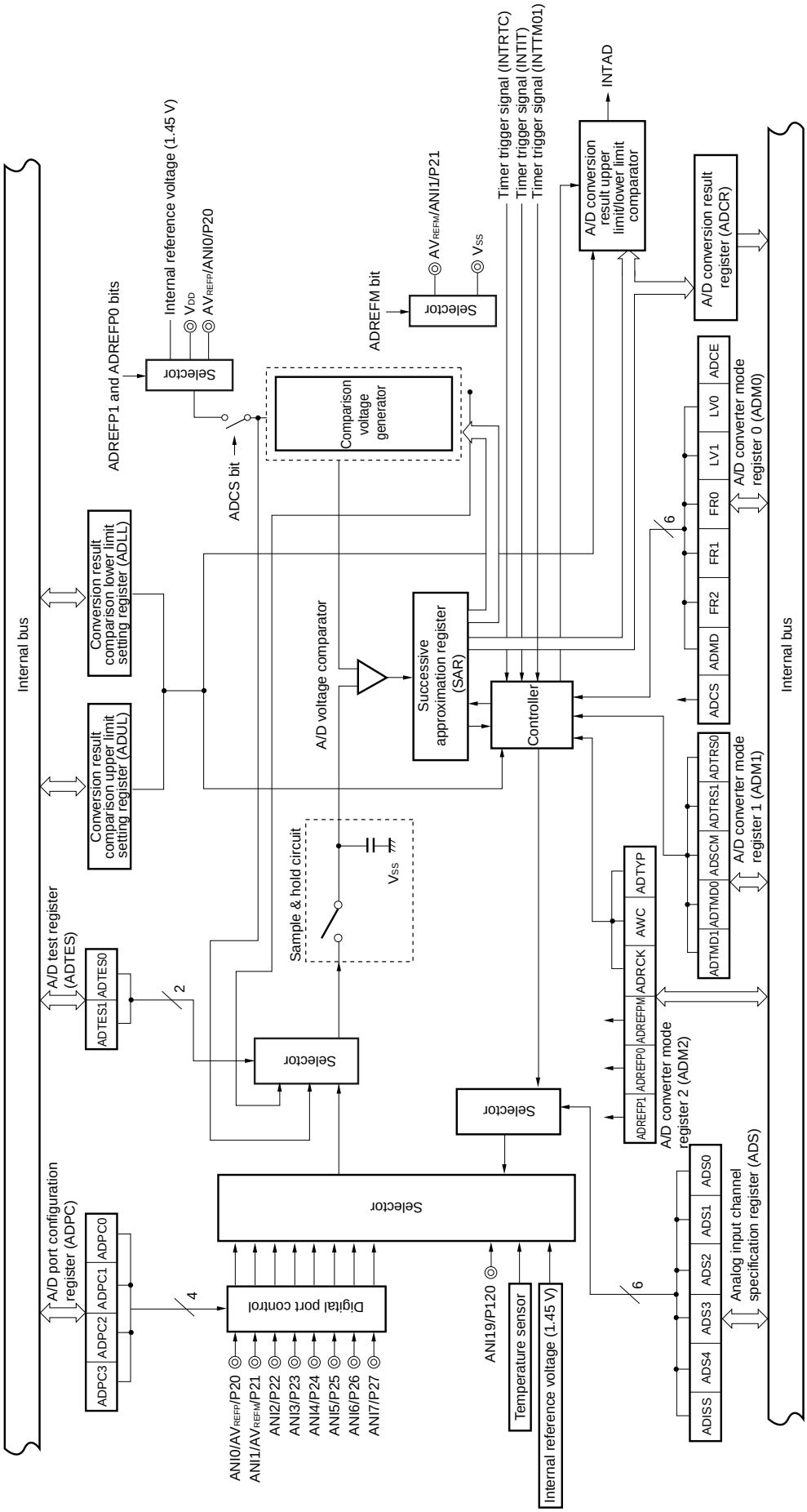
The A/D converter has the following function.

- **10-bit/8-bit resolution A/D conversion**
  10-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI7 and ANI19. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger	Conversion is started by software.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	The power is turned on by detecting a hardware trigger while the system is off and in the conversion standby state, and conversion is then started automatically after the stabilization wait time passes. When using the SNOOZE mode function, specify the hardware trigger wait mode.
Channel selection mode	Select mode	A/D conversion is performed on the analog input of one selected channel.
	Scan mode	A/D conversion is performed on the analog input of four channels in order. Four consecutive channels can be selected from ANI0 to ANI7 as analog input channels.
Conversion operation mode	One-shot conversion mode	A/D conversion is performed on the selected channel once.
	Sequential conversion mode	A/D conversion is sequentially performed on the selected channels until it is stopped by software.
Operation voltage mode	Standard 1 or standard 2 mode	Conversion is done in the operation voltage range of 2.7 V ≤ V_{DD} ≤ 5.5 V.
	Low voltage 1 or low voltage 2 mode	Conversion is done in the operation voltage range of 2.4 V ≤ V_{DD} ≤ 5.5 V. Select this mode for conversion at a low voltage. Because the operation voltage is low, it is internally boosted during conversion.
Sampling time selection	Sampling clock cycles: 7 f_{AD}	The sampling time in standard 1 or low voltage 1 mode is seven cycles of the conversion clock (f_{AD}). Select this mode when the output impedance of the analog input source is high and the sampling time should be long.
	Sampling clock cycles: 5 f_{AD}	The sampling time in standard 2 or low voltage 2 mode is five cycles of the conversion clock (f_{AD}). Select this mode when enough sampling time is ensured (for example, when the output impedance of the analog input source is low).

Figure 11-1. Block Diagram of A/D Converter



Remark Analog input pin for figure 11-1 when a 48-pin product is used.

11.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI7 and ANI19 pins

These are the analog input pins of the 9 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.

(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 AV_{REF}) 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 AV_{REF}), 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 AV_{REF})
Bit 9 = 1: (3/4 AV_{REF})

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.

Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

(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 register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(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 INTAD through the A/D conversion result upper limit/lower limit comparator.

(9) AV_{REFP} pin

This pin inputs an external reference voltage (AV_{REFP}).

If using AV_{REFP} as the + side reference voltage of the A/D converter, set the ADREFP1 and ADREFP0 bits of A/D converter mode register 2 (ADM2) to 0 and 1, respectively.

The analog signals input to ANI0 to ANI7 and ANI19 are converted to digital signals based on the voltage applied between AV_{REFP} and the – side reference voltage (AV_{REFM}/V_{SS}).

In addition to AV_{REFP}, it is possible to select V_{DD} or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

(10) AV_{REFM} pin

This pin inputs an external reference voltage (AV_{REFM}). If using AV_{REFM} as the – side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to AV_{REFM}, it is possible to select V_{SS} as the – side reference voltage of the A/D converter.

11.3 Registers Controlling 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 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- A/D port configuration register (ADPC)
- Port mode control registers 0, 12 (PMC0, PMC12)
- Port mode registers 0, 2, 12 (PM0, PM2, PM12)

11.3.1 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 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	RTCEN	0	ADCEN	IICA0EN	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.

Cautions 1. When setting the A/D converter, be sure to set the following registers first while the ADCEN bit is set to 1. 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 port mode registers 0, 2, and 12 (PM0, PM2, and PM12), port mode control registers 0, and 12 (PMC0, and PMC12), and A/D port configuration register (ADPC)).

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 1 (ADM1)**
- **A/D converter mode register 2 (ADM2)**
- **10-bit A/D conversion result register (ADCR)**
- **8-bit A/D conversion result register (ADCRH)**
- **Analog input channel specification register (ADS)**
- **Conversion result comparison upper limit setting register (ADUL)**
- **Conversion result comparison lower limit setting register (ADLL)**
- **A/D test register (ADTES).**

2. Be sure to clear the bits 1, 3, 6 to 0.

11.3.2 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 11-3. 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	ADMD	FR2[Note 1]	FR1[Note 1]	FR0[Note 1]	LV1[Note 1]	LV0[Note 1]	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation [When read] Conversion stopped/standby status
1	Enables conversion operation [When read[Note 2]] While in the software trigger mode: Conversion operation status While in the hardware trigger wait mode: A/D power supply stabilization wait status + conversion operation status

ADMD	Specification of the A/D conversion channel selection mode
0	Select mode
1	Scan mode

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 FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Table 11-3 A/D Conversion Time Selection**.

2. While in the software trigger mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 1 μs from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 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. Otherwise, ignore data of the first conversion.

Cautions 1. Change the ADMD, FR2 to FR0, LV1, and LV0 bits while conversion is stopped (ADCS = 0, ADCE = 0).

2. Do not set the ADCS bit to 1 and the ADCE bit to 0 at the same time.

3. Do not change the ADCS and ADCE bits from 0 to 1 at the same time by using an 8-bit manipulation instruction. Be sure to set these bits in the order described in 11.7 A/D Converter Setup Flowchart.

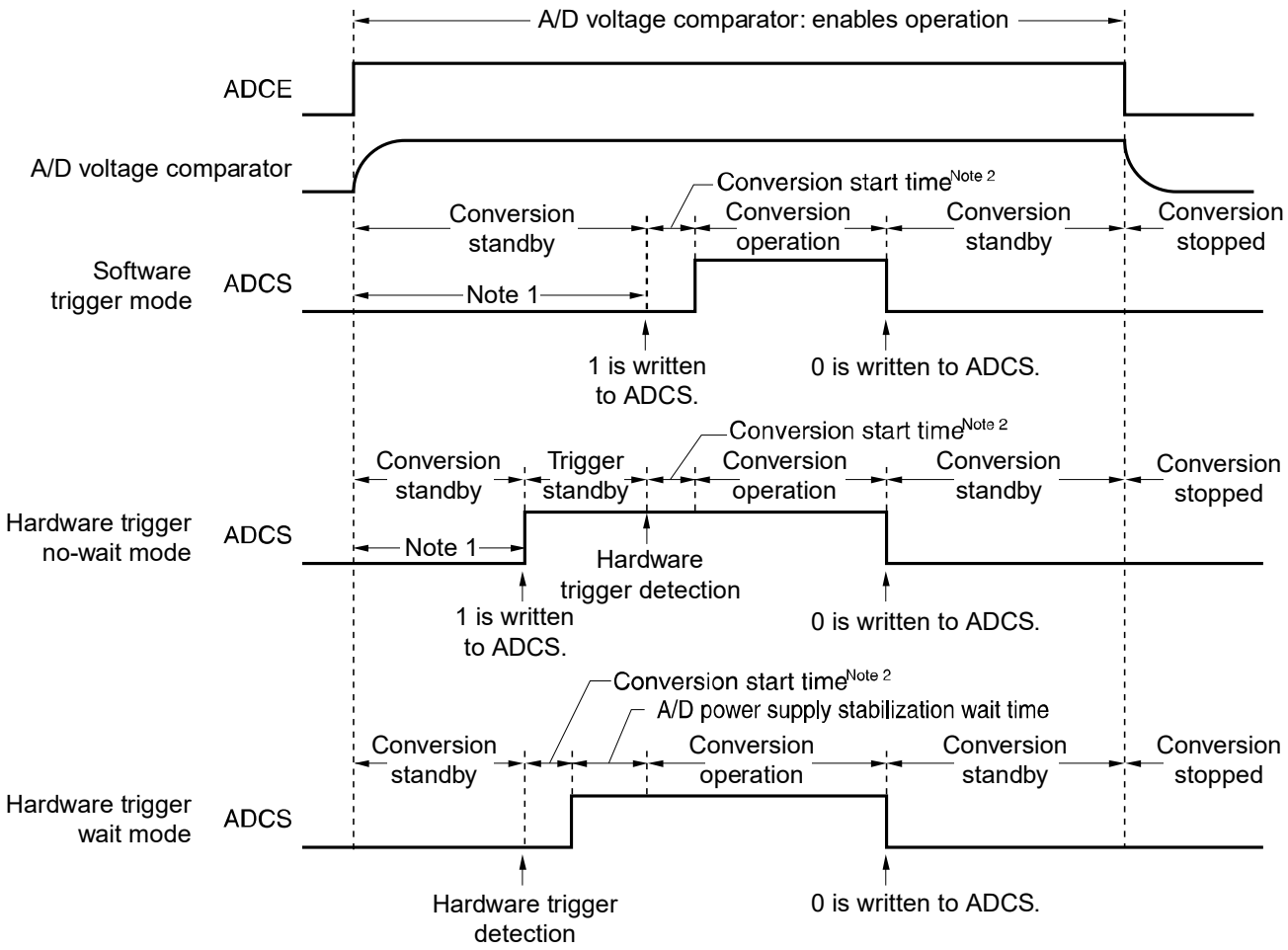
Table 11-1. Settings of ADCS and ADCE Bits

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

Table 11-2. Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Select mode	Sequential conversion mode	When a hardware trigger is input	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.

Figure 11-4. Timing Chart When A/D Voltage Comparator Is Used



Notes 1. While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer to stabilize the internal circuit.

2. The following time is the maximum amount of time necessary to start conversion.

ADM0			Conversion Clock (f_{AD})	Conversion Start Time (Number of f_{CLK} Clocks)	
FR2	FR1	FR0		Software trigger mode/ Hardware trigger no wait mode	Hardware trigger wait mode
0	0	0	$f_{CLK}/64$	63	1
0	0	1	$f_{CLK}/32$	31	
0	1	0	$f_{CLK}/16$	15	
0	1	1	$f_{CLK}/8$	7	
1	0	0	$f_{CLK}/6$	5	
1	0	1	$f_{CLK}/5$	4	
1	1	0	$f_{CLK}/4$	3	
1	1	1	$f_{CLK}/2$	1	

However, for the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected.

Cautions 1. If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby status.

2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS flag is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).

4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:

Hardware trigger no wait mode: 2 f_{CLK} clock + A/D conversion time

Hardware trigger wait mode: 2 f_{CLK} clock + stabilization wait time + A/D conversion time

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (1/4)

(1) When there is no A/D power supply stabilization wait time
Normal mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock Note	Conversion Time	Conversion Time at 10-Bit Resolution 2.7 V ≤ V_{DD} ≤ 5.5 V				
FR2	FR1	FR0	LV1	LV0					f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	1	0	Normal 1	$f_{CLK}/64$	19 f_{AD} (number of sampling clock: 7 f_{AD})	$1216/f_{CLK}$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				$f_{CLK}/32$		$608/f_{CLK}$				38 μs	25.3333 μs
0	1	0				$f_{CLK}/16$		$304/f_{CLK}$			38 μs	19 μs	12.6667 μs
0	1	1				$f_{CLK}/8$		$152/f_{CLK}$		38 μs	19 μs	9.5 μs	6.3333 μs
1	0	0				$f_{CLK}/6$		$114/f_{CLK}$		28.5 μs	14.25 μs	7.125 μs	4.7500 μs
1	0	1				$f_{CLK}/5$		$95/f_{CLK}$		23.75 μs	11.875 μs	5.9375 μs	3.9583 μs
1	1	0				$f_{CLK}/4$		$76/f_{CLK}$		19 μs	9.5 μs	4.75 μs	3.1667 μs Notes 2, 3
1	1	1				$f_{CLK}/2$		$38/f_{CLK}$	38 μs	9.5 μs	4.75 μs	2.375 μs Notes 2, 3	Setting prohibited
0	0	0	1	1	Normal 2	$f_{CLK}/64$	17 f_{AD} (number of sampling clock: 5 f_{AD})	$1088/f_{CLK}$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				$f_{CLK}/32$		$544/f_{CLK}$				34 μs	22.6667 μs
0	1	0				$f_{CLK}/16$		$272/f_{CLK}$			34 μs	17 μs	11.3333 μs
0	1	1				$f_{CLK}/8$		$136/f_{CLK}$		34 μs	17 μs	8.5 μs	5.6667 μs
1	0	0				$f_{CLK}/6$		$102/f_{CLK}$		25.5 μs	12.75 μs	6.375 μs	4.2500 μs
1	0	1				$f_{CLK}/5$		$85/f_{CLK}$		21.25 μs	10.625 μs	5.3125 μs	3.5417 μs Note 2
1	1	0				$f_{CLK}/4$		$68/f_{CLK}$		17 μs	8.5 μs	4.25 μs	2.8333 μs Notes 2, 3
1	1	1				$f_{CLK}/2$		$34/f_{CLK}$	34 μs	8.5 μs	4.25 μs	2.125 μs Notes 2, 3	Setting prohibited

Notes 1. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

2. This value is prohibited when using the temperature sensor

3. Setting prohibited in the 3.6 V

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 30.6.1 A/D converter characteristics or 31.6.1 A/D converter characteristics.

2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (2/4)

(2) When there is no A/D power supply stabilization wait time

Low-voltage mode 1, 2 [Note 1] (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock [Note 2]	Conversion Time	Conversion Time at 10-Bit Resolution				
									2.4 V ≤ V_{DD} ≤ 5.5 V				**Note 3**
FR2	FR1	FR0	LV1	LV0					f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	1	0	Low-voltage 1	f_{CLK}/64	19 f_{AD} (number of sampling clock: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32		608/f_{CLK}				38 μs	25.3333 μs
0	1	0				f_{CLK}/16		304/f_{CLK}			38 μs	19 μs	12.6667 μs
0	1	1				f_{CLK}/8		152/f_{CLK}		38 μs	19 μs	9.5 μs [Note 4]	6.3333 μs
1	0	0				f_{CLK}/6		114/f_{CLK}		28.5 μs	14.25 μs [Note 4]	7.125 μs [Note 4]	4.7500 μs
1	0	1				f_{CLK}/5		95/f_{CLK}		23.75 μs	11.875 μs [Note 4]	5.9375 μs [Note 4]	3.9583 μs
1	1	0				f_{CLK}/4		76/f_{CLK}		19 μs	9.5 μs [Note 4]	4.75 μs [Note 4]	3.1667 μs [Note 5]
1	1	1				f_{CLK}/2		38/f_{CLK}	38 μs	9.5 μs [Note 4]	4.75 μs [Note 4]	2.375 μs [Note 5]	Setting prohibited
0	0	0	1	1	Low-voltage 2	f_{CLK}/64	17 f_{AD} (number of sampling clock: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32		544/f_{CLK}				34 μs	22.6667 μs
0	1	0				f_{CLK}/16		272/f_{CLK}			34 μs	17 μs	11.3333 μs
0	1	1				f_{CLK}/8		136/f_{CLK}		34 μs	17 μs	8.5 μs [Note 4]	5.6667 μs
1	0	0				f_{CLK}/6		102/f_{CLK}		25.5 μs	12.75 μs [Note 4]	6.375 μs [Note 4]	4.2500 μs
1	0	1				f_{CLK}/5		85/f_{CLK}		21.25 μs	10.625 μs [Note 4]	5.3125 μs [Note 4]	3.5417 μs
1	1	0				f_{CLK}/4		68/f_{CLK}		17 μs	8.5 μs [Note 4]	4.25 μs [Note 4]	2.8333 μs [Note 5]
1	1	1				f_{CLK}/2		34/f_{CLK}	34 μs	8.5 μs [Note 4]	4.25 μs [Note 4]	2.125 μs [Note 5]	Setting prohibited

Notes 1. This value is prohibited when using the temperature sensor

2. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

3. 2.7 V ≤ V_{DD} ≤ 5.5 V

4. Setting prohibited in the 2.7 V

5. Setting prohibited in the 3.6 V

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 30.6.1 A/D converter characteristics or 31.6.1 A/D converter characteristics.

2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (3/4)

(3) When there is A/D power supply stabilization wait time
Normal mode 1, 2 (hardware trigger wait mode Note 1)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Stabilization Wait Clock	Number of Conversion Clock Note 2	Stabilization Wait Time+ Conversion Time	Stabilization Wait Time + Conversion Time at 10-Bit Resolution				
										2.7 V ≤ V_{DD} ≤ 5.5 V				
FR2	FR1	FR0	LV1	LV0						f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	8 f_{AD}	19 f_{AD} (number of sampling clock: 7 f_{AD})	1728/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32			864/f_{CLK}				54 μs	36.0 μs
0	1	0				f_{CLK}/16			432/f_{CLK}			54 μs	27 μs	18.0 μs
0	1	1				f_{CLK}/8			216/f_{CLK}		54 μs	27 μs	13.5 μs	9.0 μs
1	0	0				f_{CLK}/6			162/f_{CLK}		40.5 μs	20.25 μs	10.125 μs	6.8 μs
1	0	1				f_{CLK}/5			135/f_{CLK}		33.75 μs	16.875 μs	8.4375 μs	5.6 μs
1	1	0				f_{CLK}/4			108/f_{CLK}		27 μs	13.5 μs	6.75 μs	4.5 μs Notes 3, 4
1	1	1				f_{CLK}/2			54/f_{CLK}	54 μs	13.5 μs	6.75 μs	3.375 μs Notes 3, 4	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	8 f_{AD}	17 f_{AD} (number of sampling clock: 5 f_{AD})	1600/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32			800/f_{CLK}				50 μs	33.3333 μs
0	1	0				f_{CLK}/16			400/f_{CLK}			50 μs	25 μs	16.6667 μs
0	1	1				f_{CLK}/8			200/f_{CLK}		50 μs	25 μs	12.5 μs	8.3333 μs
1	0	0				f_{CLK}/6			150/f_{CLK}		37.5 μs	18.75 μs	9.375 μs	6.2500 μs
1	0	1				f_{CLK}/5			125/f_{CLK}		31.25 μs	15.625 μs	7.8125 μs	5.2083 μs Note 3
1	1	0				f_{CLK}/4			100/f_{CLK}		25 μs	12.5 μs	6.25 μs	4.1667 μs Notes 3, 4
1	1	1				f_{CLK}/2			50/f_{CLK}	50 μs	12.5 μs	6.25 μs	3.125 μs Notes 3, 4	Setting prohibited

Notes 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **table 11-3 (1/4)**).

2. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

3. This value is prohibited when using the temperature sensor

4. Setting prohibited in the 3.6 V

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 30.6.1 A/D converter characteristics or 31.6.1 A/D converter characteristics. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.

2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

4. When hardware trigger wait mode, specify the conversion time, including the A/D power supply stabilization wait time from the hardware trigger detection.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 11-3. A/D Conversion Time Selection (4/4)

(4) When there is A/D power supply stabilization wait time
Low-voltage mode 1, 2 Note 1 (hardware trigger wait mode Note 2)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Stabilization Wait Clock	Number of Conversion Clock Note 3	Stabilization Wait Time + Conversion Time	Stabilization Wait Time + Conversion Time at 10-Bit Resolution				
										2.4 V ≤ V_{DD} ≤ 5.5 V				**Note 4**
FR2	FR1	FR0	LV1	LV0						f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	1	1	Low-voltage 1	f_{CLK}/64	2 f_{AD}	19 f_{AD} (number of sampling clock: 7 f_{AD})	1344/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32			672/f_{CLK}				42 μs	28 μs
0	1	0				f_{CLK}/16			336/f_{CLK}			42 μs	21 μs	14 μs
0	1	1				f_{CLK}/8			168/f_{CLK}		42 μs	21 μs	10.5 μs Note 5	7 μs
1	0	0				f_{CLK}/6			126/f_{CLK}		31.5 μs	15.75 μs Note 5	7.875 μs Note 5	5.25 μs
1	0	1				f_{CLK}/5			105/f_{CLK}		26.25 μs	13.125 μs Note 5	6.5625 μs Note 5	4.375 μs
1	1	0				f_{CLK}/4			84/f_{CLK}		21 μs	10.5 μs Note 5	5.25 μs Note 5	3.5 μs Note 6
1	1	1				f_{CLK}/2			42/f_{CLK}	42 μs	10.5 μs Note 5	5.25 μs Note 5	2.625 μs Note 6	Setting prohibited
0	0	0	0	1	Low-voltage 2	f_{CLK}/64	2 f_{AD}	17 f_{AD} (number of sampling clock: 5 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	1				f_{CLK}/32			608/f_{CLK}				38 μs	25.3333 μs
0	1	0				f_{CLK}/16			304/f_{CLK}			38 μs	19 μs	12.6667 μs
0	1	1				f_{CLK}/8			152/f_{CLK}		38 μs	19 μs	9.5 μs Note 5	6.3333 μs
1	0	0				f_{CLK}/6			114/f_{CLK}		28.5 μs	14.25 μs Note 5	7.125 μs Note 5	4.7500 μs
1	0	1				f_{CLK}/5			96/f_{CLK}		23.75 μs	11.875 μs Note 5	5.9375 μs Note 5	3.9583 μs
1	1	0				f_{CLK}/4			76/f_{CLK}		19 μs	9.5 μs Note 5	4.75 μs Note 5	3.1667 μs Note 6
1	1	1				f_{CLK}/2			38/f_{CLK}	38 μs	9.5 μs Note 5	4.75 μs Note 5	2.375 μs Note 6	Setting prohibited

Notes 1. This value is prohibited when using the temperature sensor

2. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 11-3 (2/4)**).

3. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

4. 2.7 V ≤ V_{DD} ≤ 5.5 V

5. Setting prohibited in the 2.7 V

6. Setting prohibited in the 3.6 V

Cautions 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 30.6.1 A/D converter characteristics or 31.6.1 A/D converter characteristics. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.

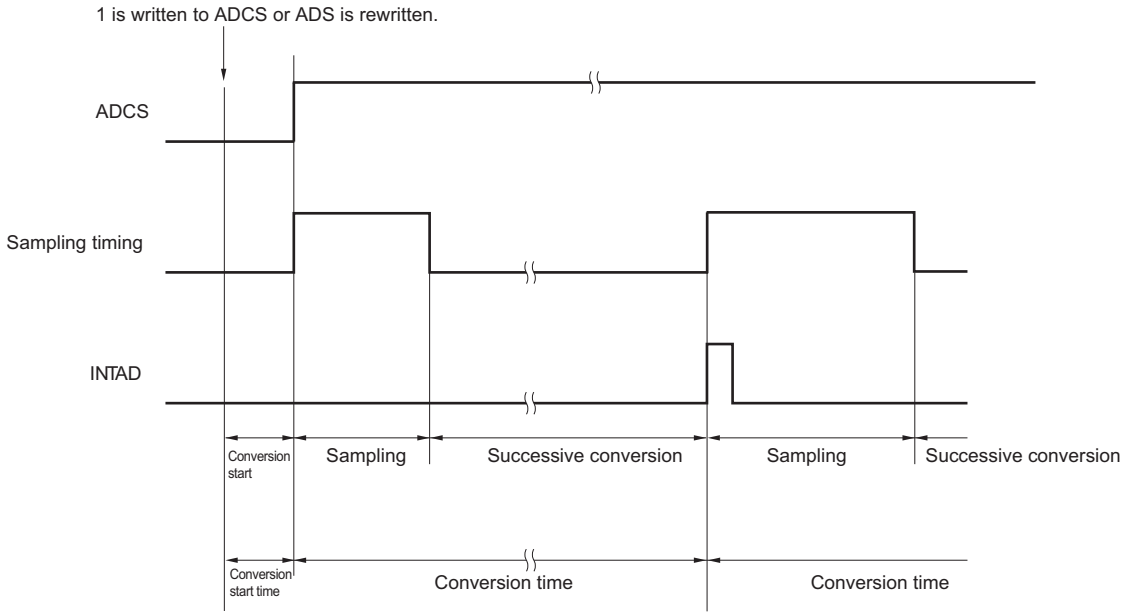
2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

4. When hardware trigger wait mode, specify the conversion time, including the A/D power supply stabilization wait time from the hardware trigger detection.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 11-5. A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



11.3.3 A/D converter mode register 1 (ADM1)

This register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal.

The ADM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-6. Format of A/D Converter Mode Register 1 (ADM1)

Address: FFF32H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	–	Software trigger mode
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	End of timer channel 1 count or capture interrupt signal (INTTM01)
0	1	Setting prohibited
1	0	Real-time clock interrupt signal (INTRTC)
1	1	12-bit interval timer interrupt signal (INTIT)

Cautions 1. Rewrite the value of the ADM1 register while conversion is stopped (ADCS = 0, ADCE = 0).

2. To complete A/D conversion, specify at least the following time as the hardware trigger interval:

Hardware trigger no wait mode: 2 f_{CLK} clock + conversion start time + A/D conversion time

Hardware trigger wait mode: 2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

3. In modes other than SNOOZE mode, input of the next INTRTC or INTIT will not be recognized as a valid hardware trigger for up to four f_{CLK} cycles after the first INTRTC or INTIT is input.

11.3.4 A/D converter mode register 2 (ADM2)

This register is used to select the + side or - side reference voltage of the A/D converter, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use the SNOOZE mode.

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 11-7. Format of A/D Converter Mode Register 2 (ADM2) (1/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the + side reference voltage source of the A/D converter
0	0	Supplied from V_{DD}
0	1	Supplied from P20/AV_{REFP}/ANI0
1	0	Supplied from the internal reference voltage (1.45 V)**Note**
1	1	Setting prohibited

- When ADREFP1 or ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures.
  (1) Set ADCE = 0
  (2) Change the values of ADREFP1 and ADREFP0
  (3) Reference voltage stabilization wait time (A)
  (4) Set ADCE = 1
  (5) Reference voltage stabilization wait time (B)
  When ADREFP1 and ADREFP0 are set to 1 and 0, the setting is changed to A = 5 μs, B = 1 μs.
  When ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1, A needs no wait and B = 1 μs.
- When ADREFP1 and ADREFP0 are set to 1 and 0, respectively, A/D conversion cannot be performed on the temperature sensor output and internal reference voltage output.
  Be sure to perform A/D conversion while ADISS = 0.

ADREFM	Selection of the – side reference voltage source of the A/D converter
0	Supplied from V_{SS}
1	Supplied from P21/AV_{REFM}/ANI1

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (<1>).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (<2>) or the ADUL register < the ADCR register (<3>).
Figure 11-8 shows the generation range of the interrupt signal (INTAD) for <1> to <3>.	

Note This setting can be used only in HS (high-speed main) mode.

Cautions 1. Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

2. Do not set the ADREFP1 bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. Also, if the ADREFP1 bit is set to 1, the temperature sensor operating current indicated in 30.4.2 and 31.4.2 Supply current characteristics (I_{TMPS}) will be added to the current consumption when shifting to HALT mode while the CPU is operating on the main system clock.

3. When using AV_{REFP} and AV_{REFM}, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

Figure 11-7. Format of A/D Converter Mode Register 2 (ADM2) (2/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.

When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).

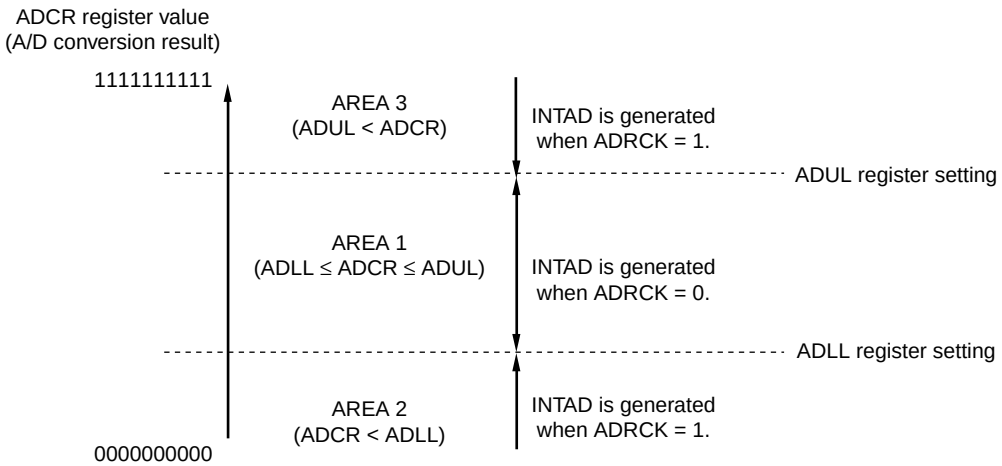
- The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited.
- Using the SNOOZE mode function in the sequential conversion mode is prohibited.
- When using the SNOOZE mode function, specify a hardware trigger interval of at least "shift time to SNOOZE mode **Note** + conversion start time + A/D power supply stabilization wait time + A/D conversion time +2 f_{CLK} clock"
- Even when using SNOOZE mode, be sure to set the AWC bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode.
  Also, be sure to change the AWC bit to 0 after returning from STOP mode to normal operation mode.
  If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode.

ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Note Refer to "From STOP to SNOOZE" in **19.3.3 SNOOZE mode**

Caution Only rewrite the value of the ADM2 register while conversion operation is stopped (which is indicated by the ADCS bit of A/D converter mode register 0 (ADM0) being 0).

Figure 11-8. ADRCK Bit Interrupt Signal Generation Range



Remark If INTAD does not occur, the A/D conversion result is not stored in the ADCR or ADCRH register.

11.3.5 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH [Note].

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 11-8**), the result is not stored.

Figure 11-9. Format of 10-bit A/D Conversion Result Register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR											0	0	0	0	0	0

Cautions 1. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the lower two bits (ADCR1 and ADCR0).

2. When the ADCR register is accessed in 16-bit units, the higher 10 bits of the conversion result are read in order starting at bit 15.

11.3.6 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored [Note].

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

Reset signal generation clears this register to 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 11-8**), the result is not stored.

Figure 11-10. Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to the A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, and ADPC registers. Using timing other than the above may cause an incorrect conversion result to be read.

11.3.7 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 11-11. Format of Analog Input Channel Specification Register (ADS) (1/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AV_{REFP} pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV_{REFM} pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin Note 1
0	0	0	1	1	0	ANI6	P26/ANI6 pin Note 1
0	0	0	1	1	1	ANI7	P27/ANI7 pin Note 1
0	1	0	0	0	0	ANI16	P01/ANI16 pin Note 2
0	1	0	0	0	1	ANI17	P00/ANI17 pin Note 2
0	1	0	0	1	1	ANI19	P120/ANI19 pin
1	0	0	0	0	0	–	Temperature sensor output Note 3
1	0	0	0	0	1	–	Internal reference voltage output (1.45 V) Note 3
Other than the above						Setting prohibited	

Notes 1. 48-pin product only

2. 32-pin product only

3. Operation is possible only in HS (high-speed main) mode.

Figure 11-11. Format of Analog Input Channel Specification Register (ADS) (2/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○ Scan mode (ADMD = 1)

	ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
							Scan 0	Scan 1	Scan 2	Scan 3
	0	0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
	0	0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
Note	0	0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
Note	0	0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
Note	0	0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
	Other than the above						Setting prohibited			

Note 48-pin product only

Cautions 1. Be sure to clear bits 5 and 6 to 0.

2 Set a channel to be used for A/D conversion in the input mode by using port mode registers 0, 2, 3, 10 to 12, 14, or 15 (PM0, PM2, PM3, PM10 t o PM12, PM14, PM15).

3. Do not set the pin that is set by the A/D port configuration register (ADPC) as digital I/O by the ADS register.

4. Do not set the pin that is set by port mode control register 0, 3, 10 to 12, or 14 (PMC0, PMC3, PMC10 to PMC12, PMC14) as digital I/O by the ADS register.

5. Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

6. If using AV_{REFP} as the + side reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.

7. If using AV_{REFM} as the – side reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.

8. If the ADISS bit is set to 1, the internal reference voltage output (1.45 V) cannot be used for the + side reference voltage. After the ADISS bit is set to 1, the initial conversion result cannot be used. For the setting flow, see 11.7.4 Setup when temperature sensor output/internal reference voltage output is selected (example for software trigger mode and one-shot conversion mode).

9. Do not set the ADISS bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. Also, if the ADREFP1 bit is set to 1, the A/D converter reference voltage current (I_{ADREF}) indicated in 30.3.2 and 31.3.2 Supply current characteristics will be added to the current consumption when shifting to HALT mode while the CPU is operating on the main system clock.

11.3.8 Conversion result comparison upper limit setting register (ADUL)

This register is used to specify the setting for checking the upper limit of the A/D conversion results.

The A/D conversion results and ADUL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 11-8**).

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

Reset signal generation sets this register to FFH.

Caution When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADUL register.

Figure 11-12. Format of Conversion Result Comparison Upper Limit Setting Register (ADUL)

Address: F0011H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

11.3.9 Conversion result comparison lower limit setting register (ADLL)

This register is used to specify the setting for checking the lower limit of the A/D conversion results.

The A/D conversion results and ADLL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 11-8**).

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

Reset signal generation clears this register to 00H.

Figure 11-13. Format of Conversion Result Comparison Lower Limit Setting Register (ADLL)

Address: F0012H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Cautions
1. **When A/D conversion with 10-bit resolution is selected, the eight higher-order bits of the 10-bit A/D conversion result register (ADCR) are compared with the values in the ADUL and ADLL registers.**
2. **Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).**
3. **The setting of the ADUL registers must be greater than that of the ADLL register.**

11.3.10 A/D test register (ADTES)

This register is used to select the + side reference voltage or - side reference voltage for the converter, an analog input channel (ANIxx), the temperature sensor output, or the internal reference voltage output (1.45 V) as the target for A/D conversion.

When using this register to test the converter, set as follows.

- For zero-scale measurement, select the - side reference voltage as the target for conversion.
- For full-scale measurement, select the + side reference voltage as the target for conversion.

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

Reset signal generation clears this register to 00H.

Figure 11-14. 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	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output Note/internal reference voltage output (1.45 V)Note (This is specified using the analog input channel specification register (ADS).)
1	0	The - side reference voltage (selected by the ADREFM bit of the ADM2 register)
1	1	The + side reference voltage (selected by the ADREFP1 or ADREFP0 bit of the ADM2 register)
Other than the above		Setting prohibited

Note The temperature sensor output and internal reference voltage output (1.45 V) can be selected only in the HS (high-speed main) mode.

11.3.11 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 registers (PMxx), port mode control registers (PMCxx), and A/D port configuration register (ADPC)). For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.6 Port mode control registers (PMCxx)**, and **4.3.7 A/D port configuration register (ADPC)**.

When using the ANI0 to ANI7 pins for analog input of the A/D converter, set the port mode register (PMxx) bit corresponding to each port to 1 and select analog input through the A/D port configuration register (ADPC).

When using the ANI19 pins for analog input of the A/D converter, set the port mode register (PMxx) bit and port mode control register (PMCxx) bit corresponding to each port to 1.

11.4 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) AV_{REF} 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) AV_{REF}, the MSB bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, 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) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

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 (ADCR, ADCRH) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

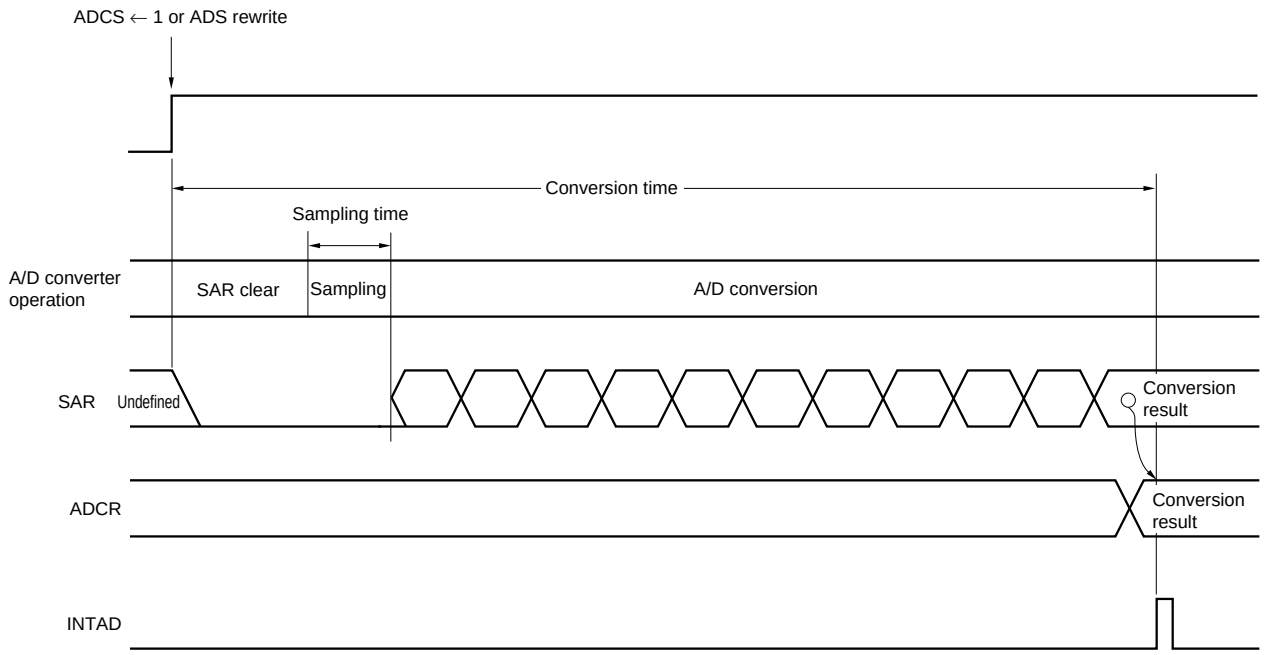
<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0[Note].
To stop the A/D converter, clear the ADCS bit to 0.

Note While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Remarks 1. Two types of the A/D conversion result registers are available.
- ADCR register (16 bits): Store 10-bit A/D conversion value
- ADCRH register (8 bits): Store 8-bit A/D conversion value

2. AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

Figure 11-15. Conversion Operation of A/D Converter (Software Trigger Mode)



In one-shot conversion mode, the ADCS bit is automatically cleared to 0 after completion of A/D conversion.

In sequential conversion mode, A/D conversion operations proceed continuously until the software clears bit 7 (ADCS) of the A/D converter mode register 0 (ADM0) to 0.

Writing to the analog input channel specification register (ADS) during A/D conversion interrupts the current conversion after which A/D conversion of the analog input specified by the ADS register proceeds. Data from the A/D conversion that was in progress are discarded.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

11.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7 and ANI19) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$SAR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$ADCR = SAR \times 64$$

or

$$\left(\frac{ADCR}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{ADCR}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

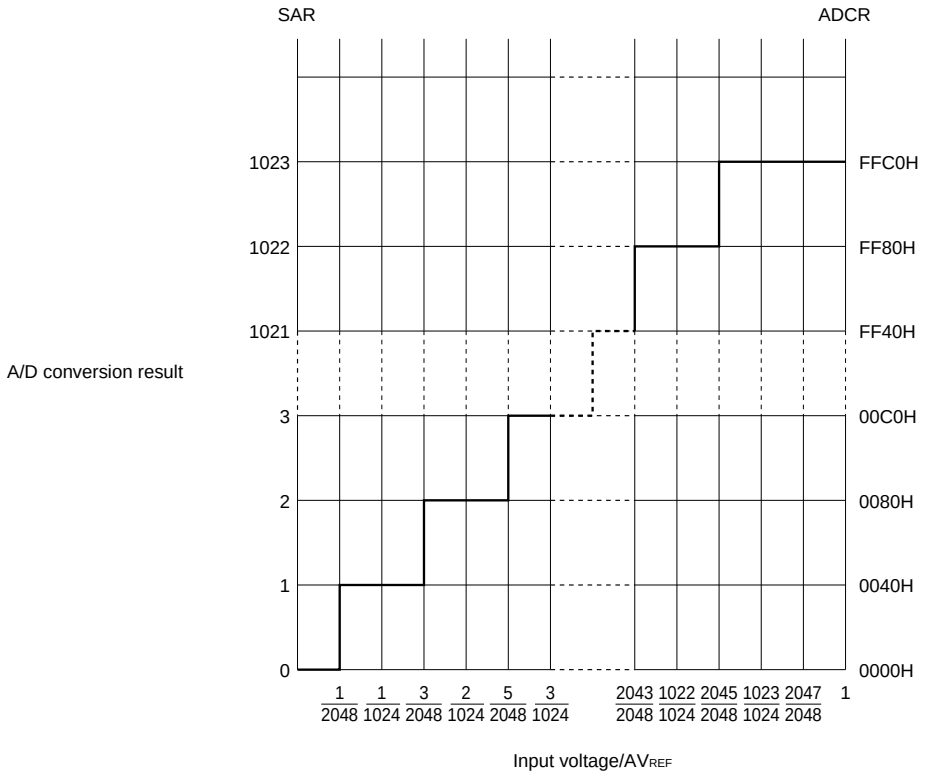
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 11-16 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 11-16. Relationship Between Analog Input Voltage and A/D Conversion Result



Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

11.6 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in **11.7 A/D Converter Setup Flowchart**.

11.6.1 Software trigger mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform 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 (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

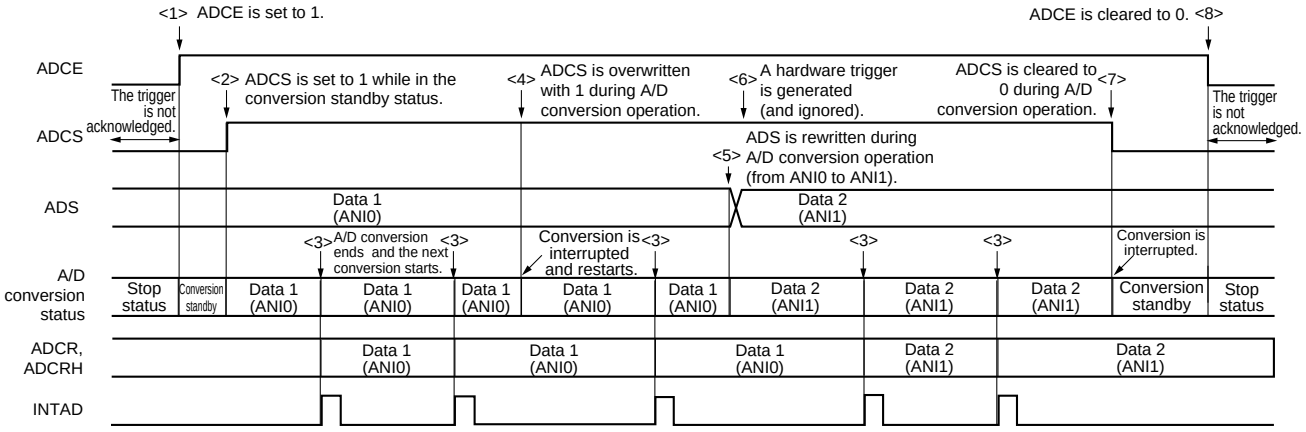
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 11-17. Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



11.6.2 Software trigger mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform 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 (ADCR, ADCRH), 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 A/D conversion standby status.

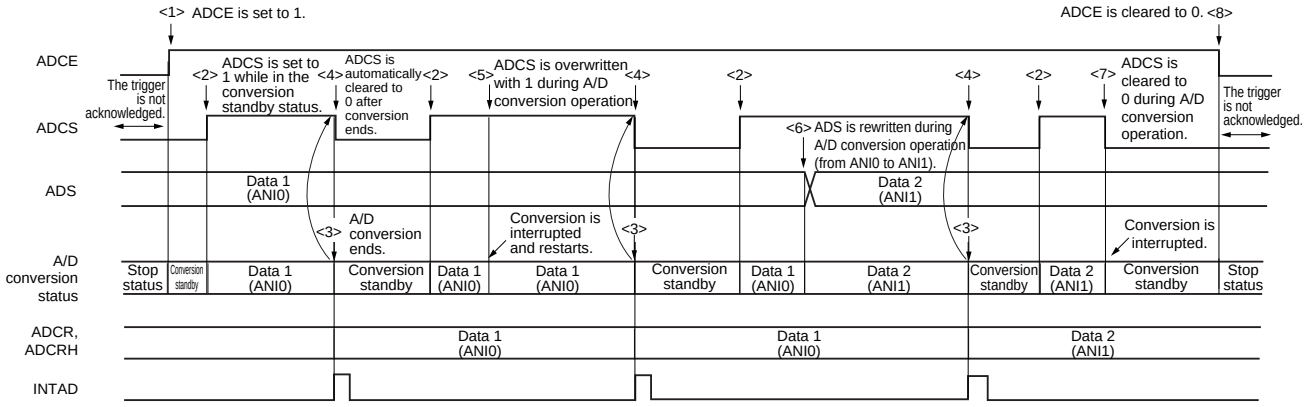
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 11-18. Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



11.6.3 Software trigger mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts (until all four channels are finished).

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

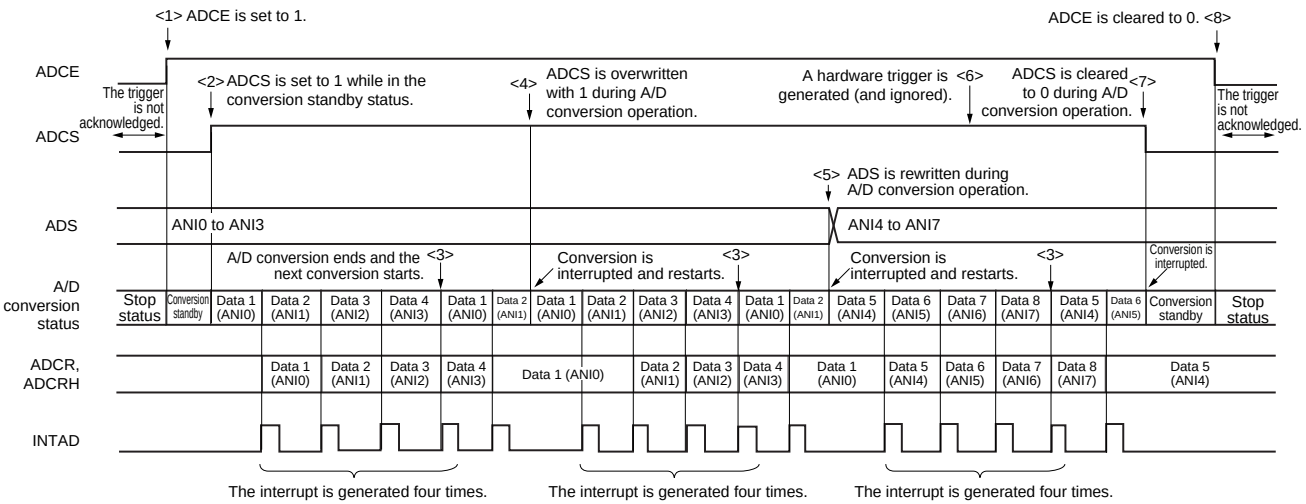
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 11-19. Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



11.6.4 Software trigger mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

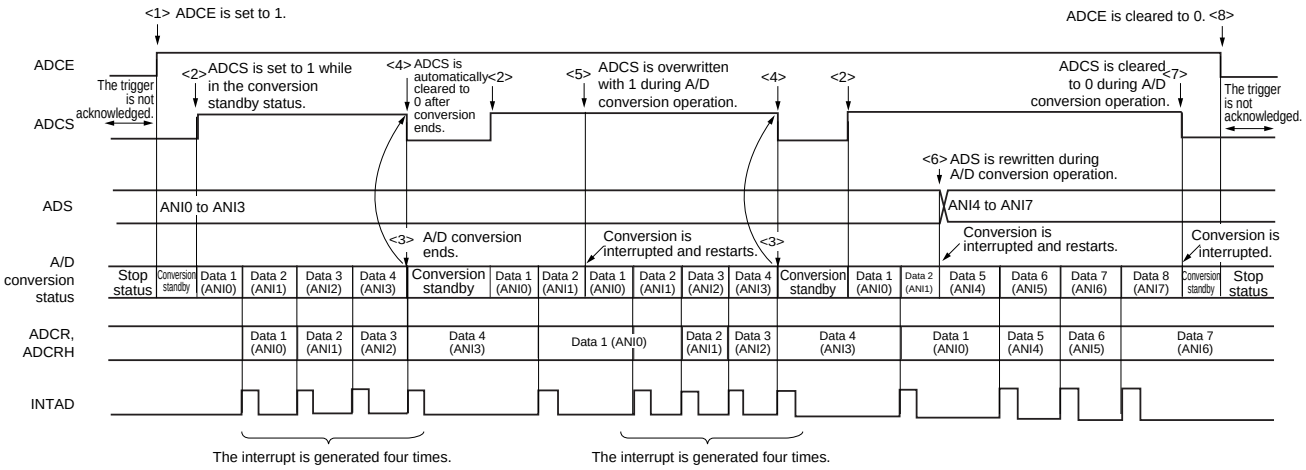
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 11-20. Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



11.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

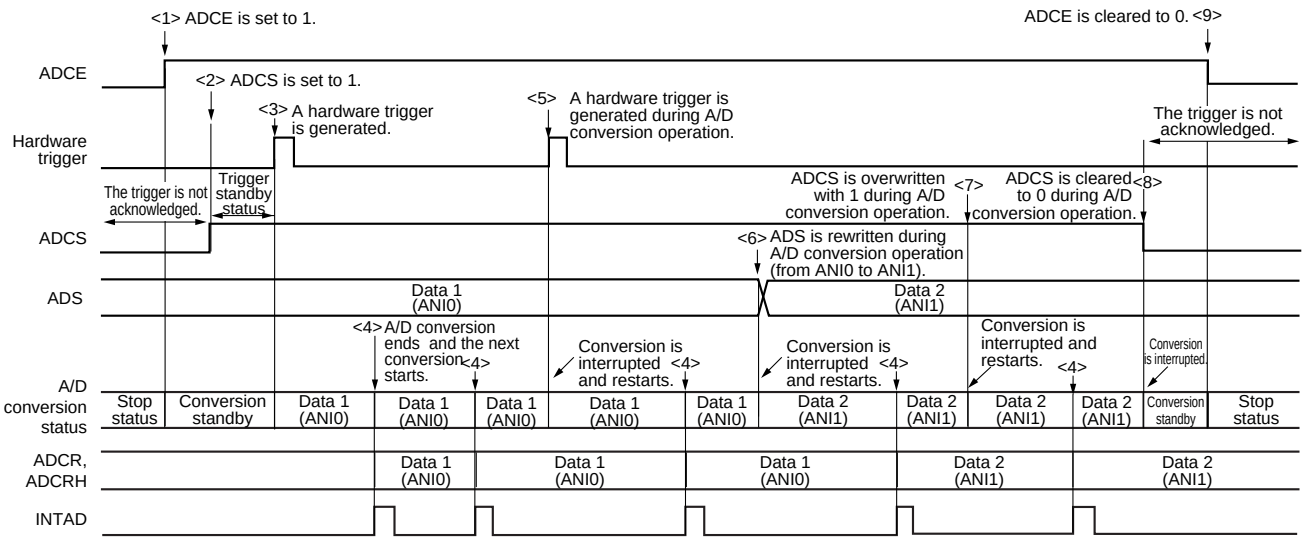
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-21. Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



11.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

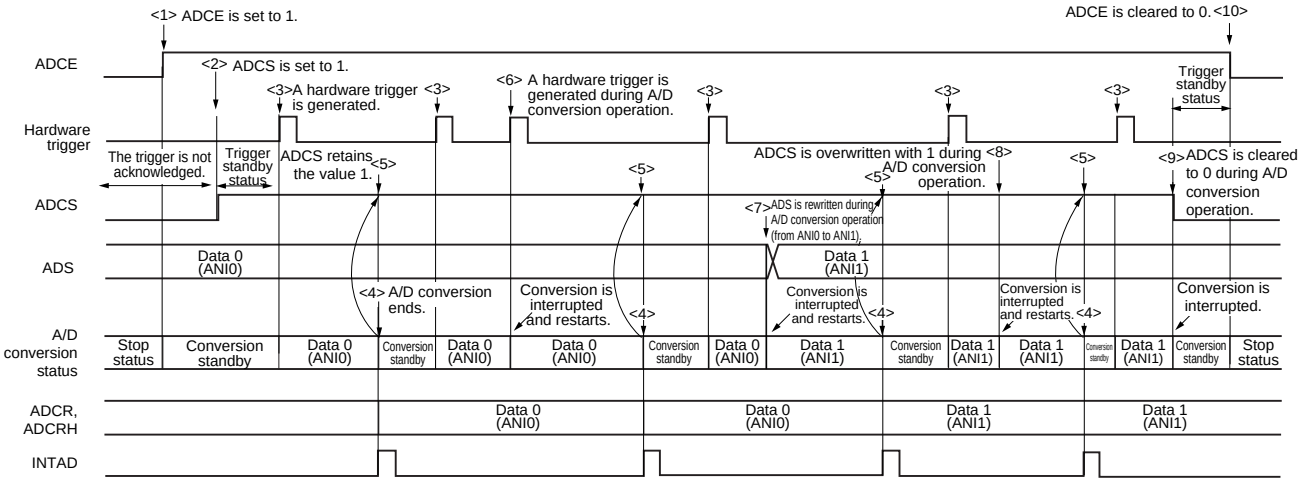
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-22. Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



11.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

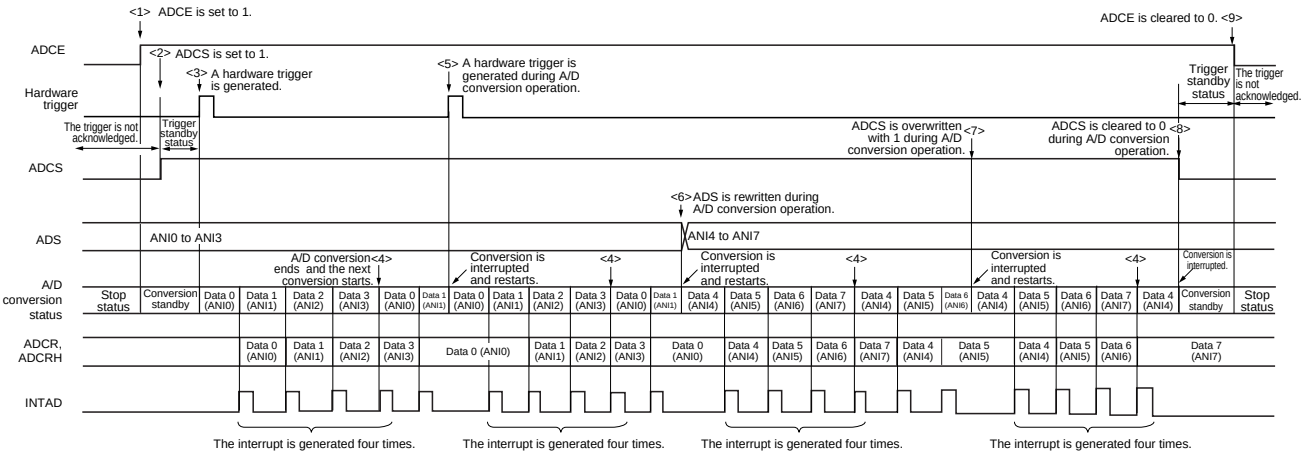
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 11-23. Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



11.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

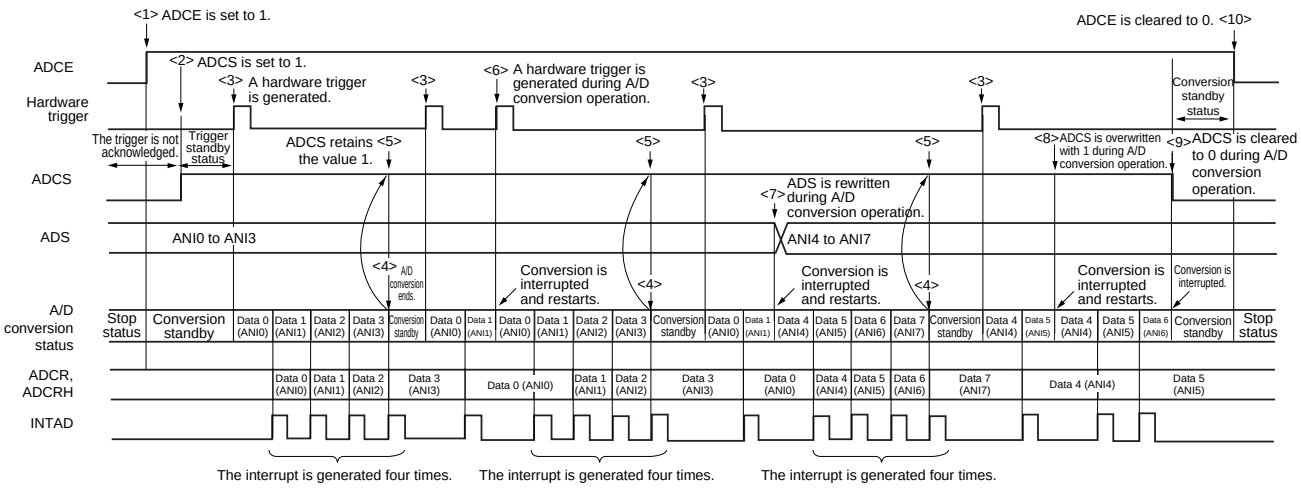
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-24. Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



11.6.9 Hardware trigger wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)

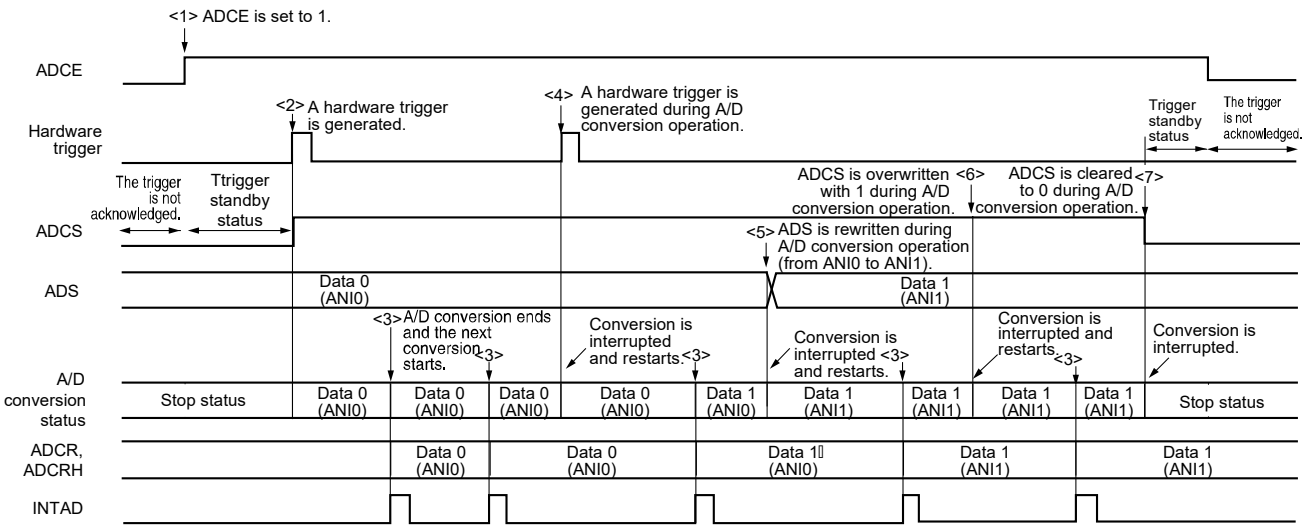
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-25. Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



11.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), 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 A/D converter enters the stop status.

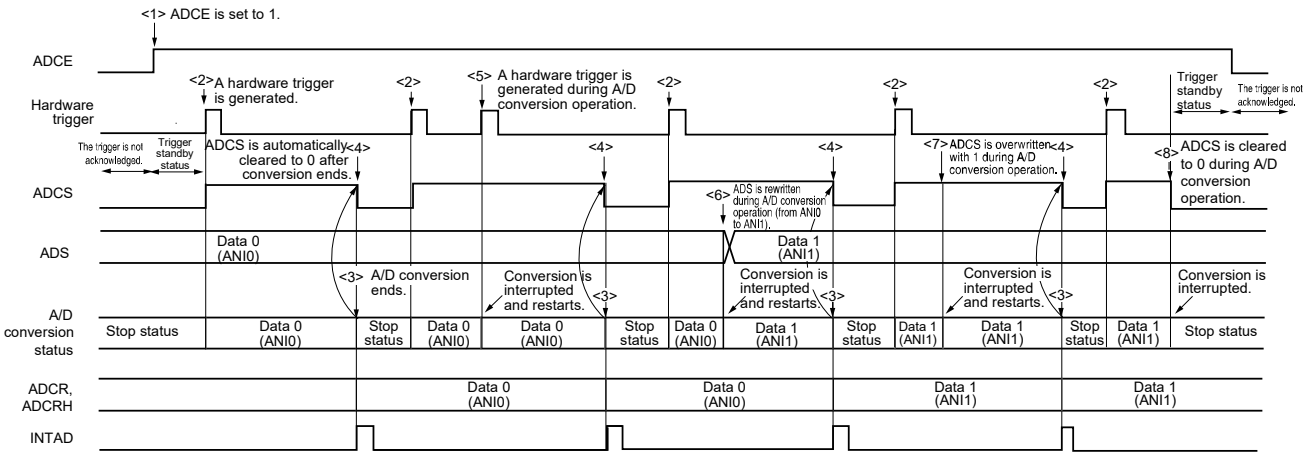
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-26. Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



11.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

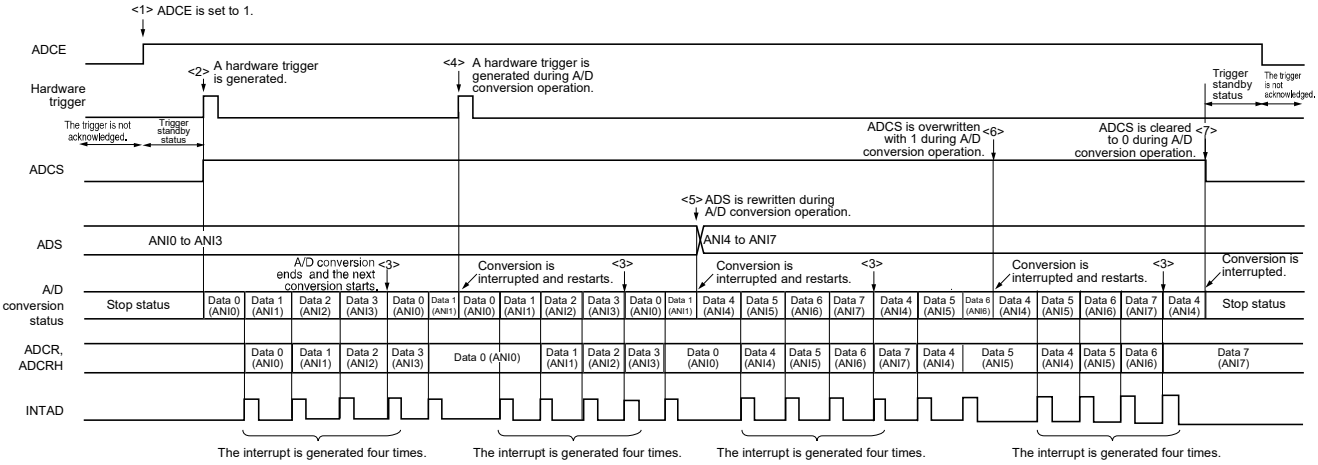
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-27. Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



11.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, 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 A/D converter enters the stop status.

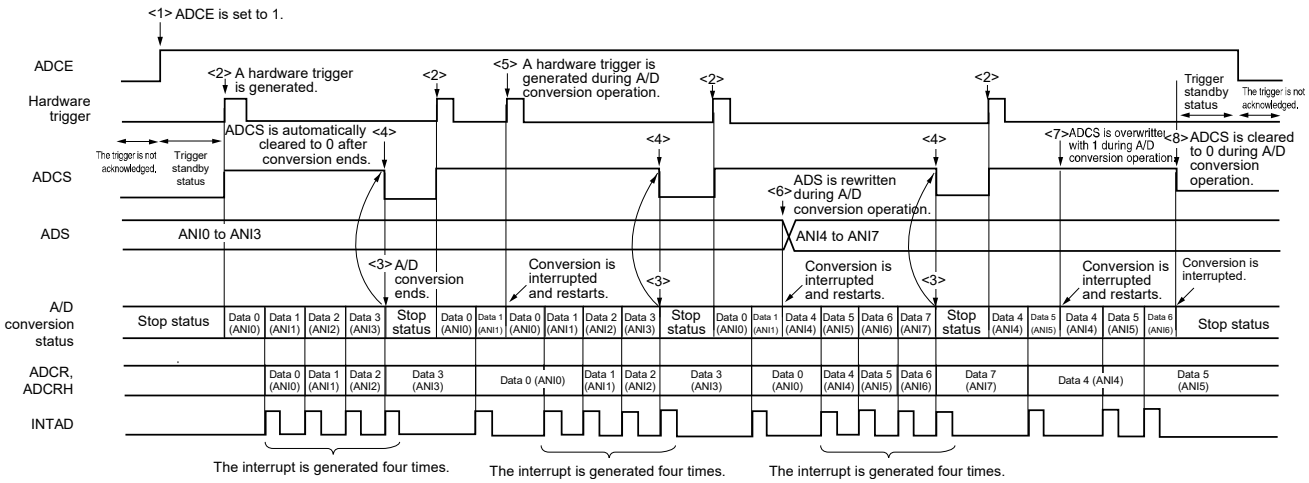
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 11-28. Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing

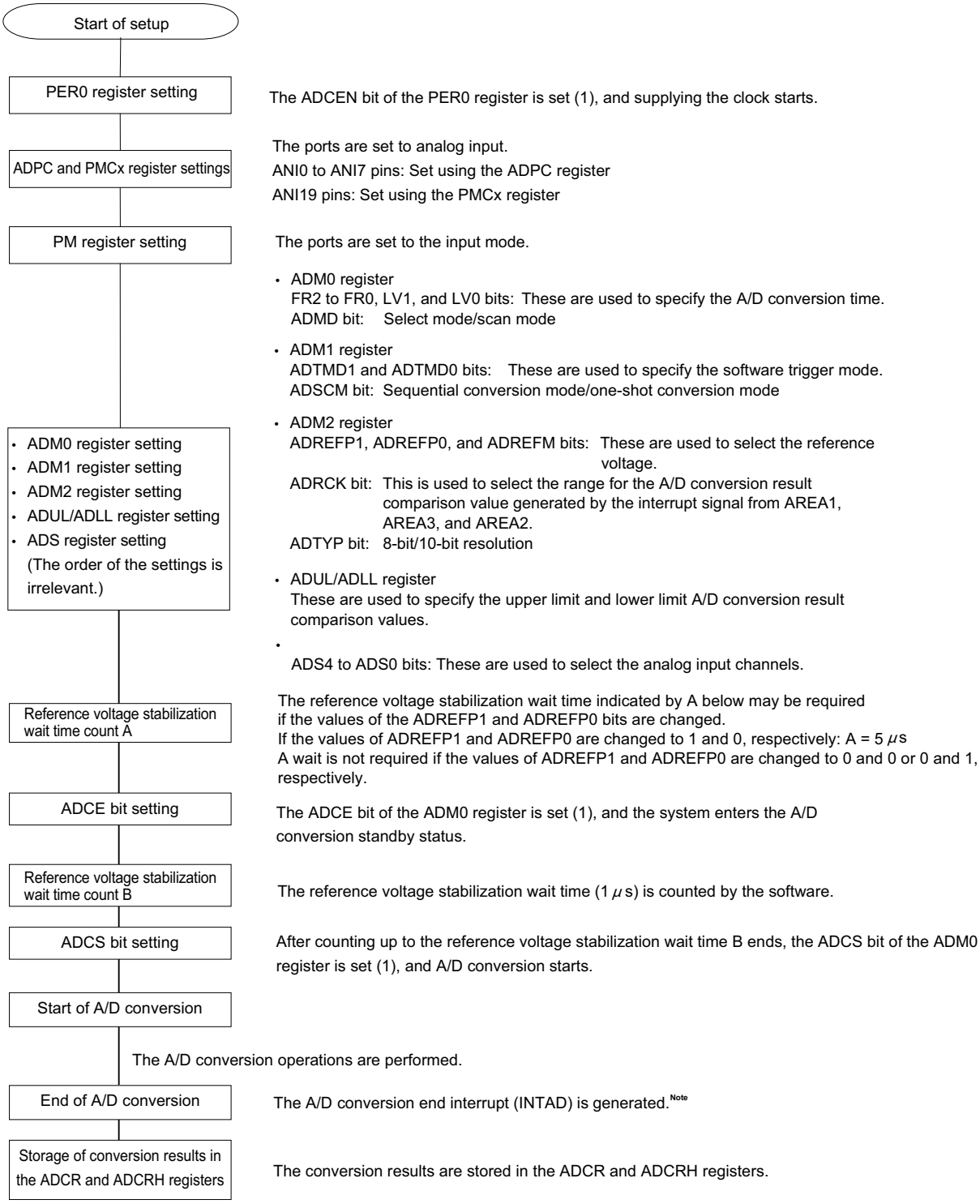


11.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

11.7.1 Setting up software trigger mode

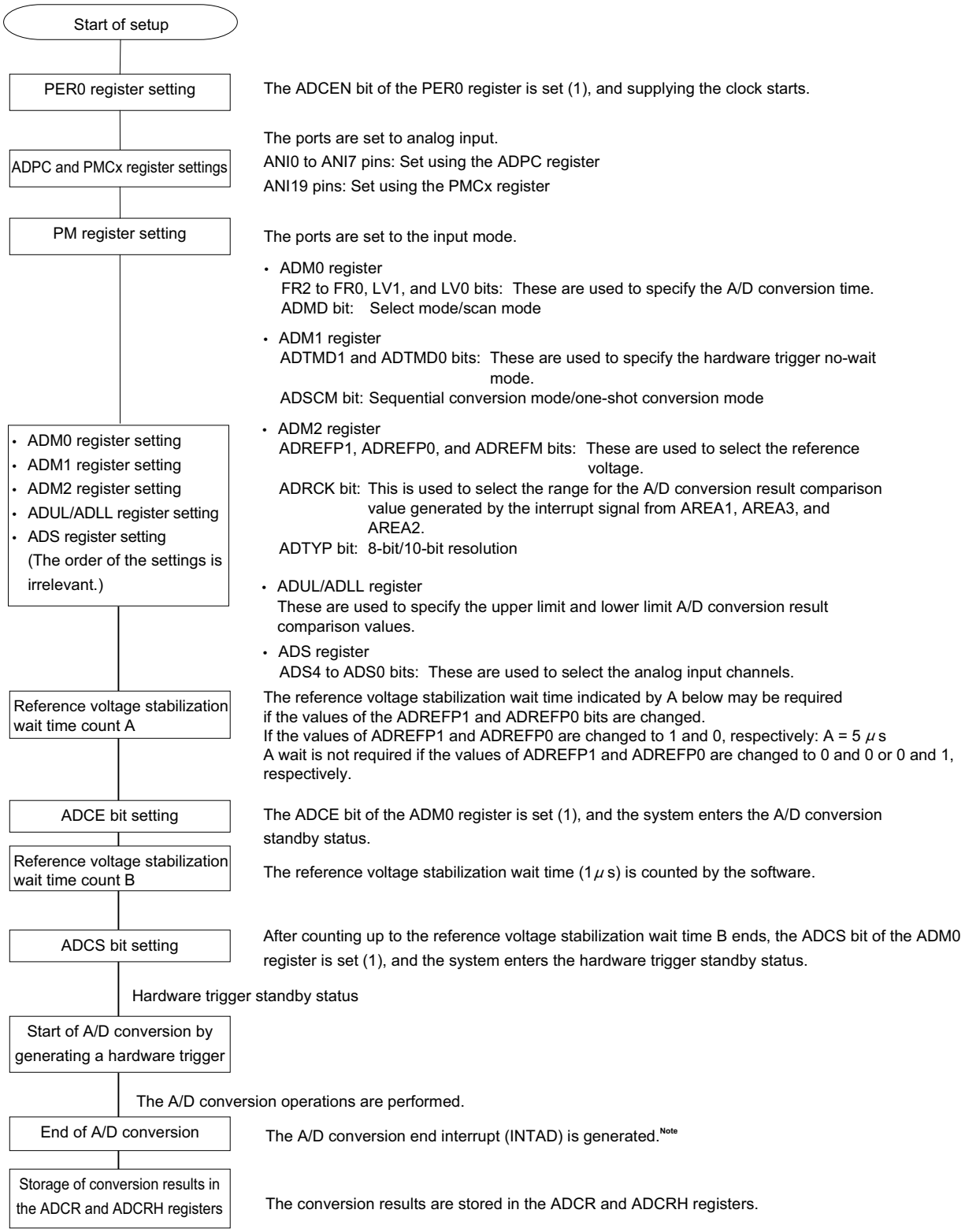
Figure 11-29. Setting up Software Trigger Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR and ADCRH registers.

11.7.2 Setting up hardware trigger no-wait mode

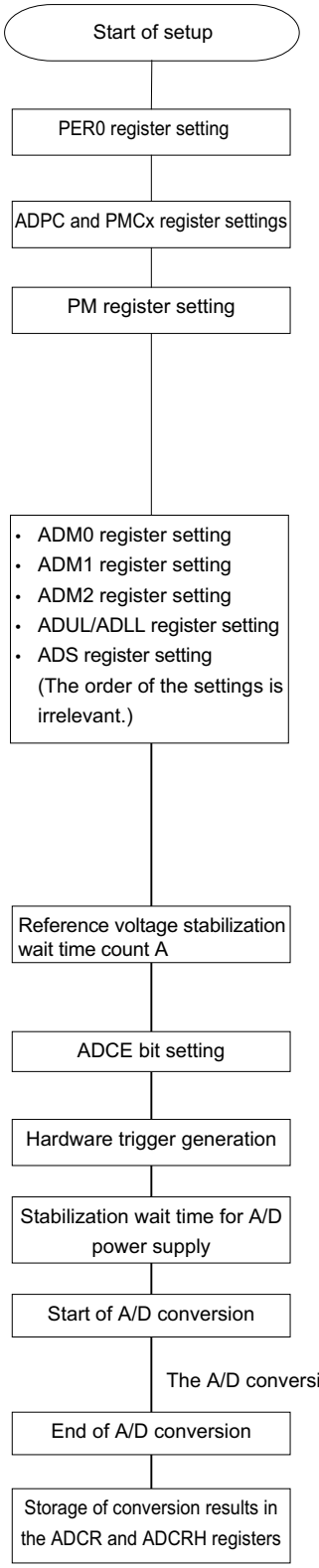
Figure 11-30. Setting up Hardware Trigger No-Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

11.7.3 Setting up hardware trigger wait mode

Figure 11-31. Setting up Hardware Trigger Wait Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 to ANI7 pins: Set using the ADPC register
ANI19 pins: Set using the PMCx register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: Select mode/scan mode
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the hardware trigger wait mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
  ADTRS1 and ADTRS0 bits: These are used to select the hardware trigger signal.
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage.
  ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2.
  ADTYP bit: 8-bit/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADS4 to ADS0 bits: These are used to select the analog input channels.

The reference voltage stabilization wait time indicated by A below may be required if the values of the ADREFP1 and ADREFP0 bits are changed.
If the values of ADREFP1 and ADREFP0 are changed to 1 and 0, respectively: A = 5 μs
A wait is not required if the values of ADREFP1 and ADREFP0 are changed to 0 and 0 or 0 and 1, respectively.

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The system automatically counts up to the stabilization wait time for A/D power supply.

After counting up to the reference voltage stabilization wait time ends, A/D conversion starts

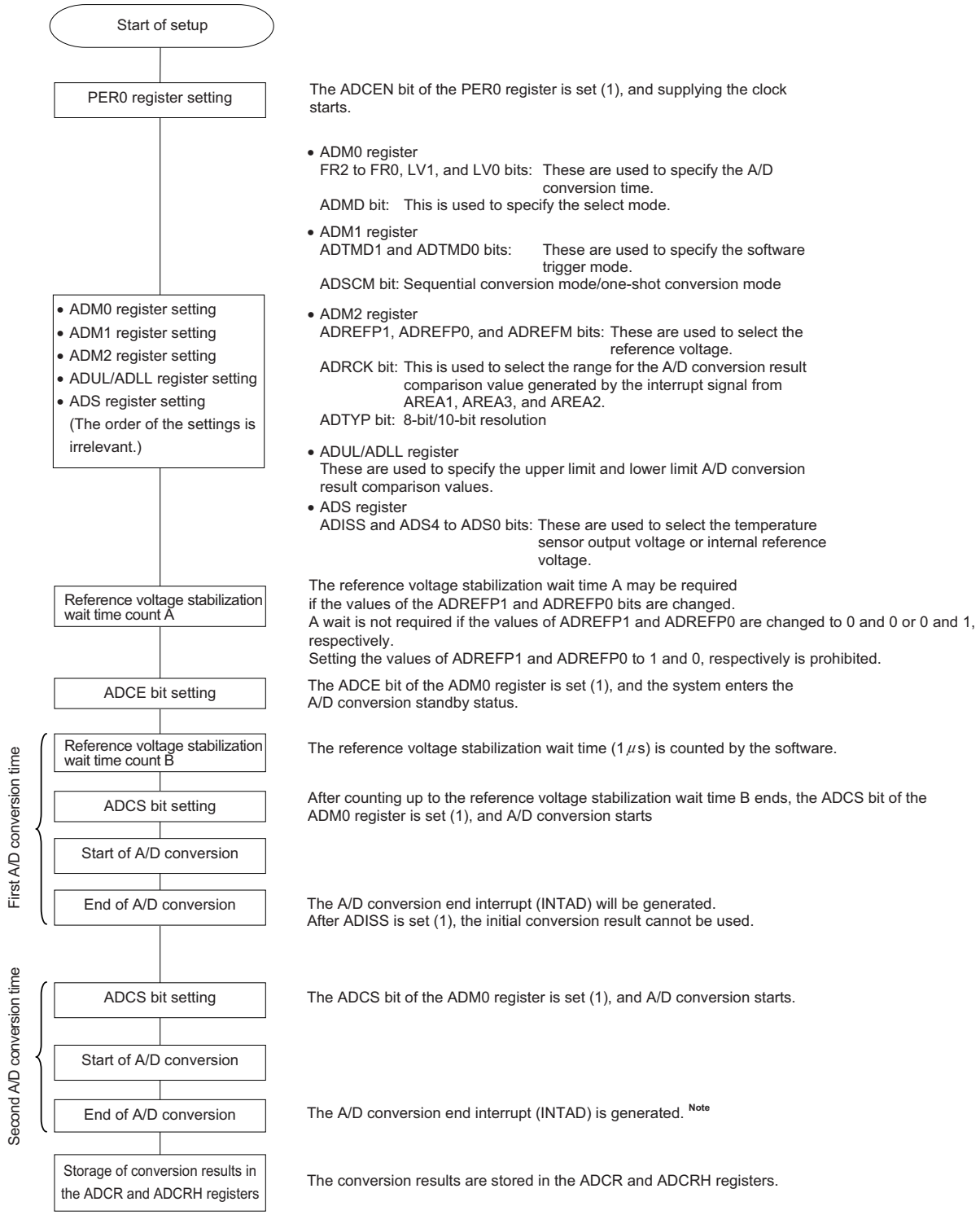
The A/D conversion end interrupt (INTAD) is generated.[Note]

The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

11.7.4 Setup when using temperature sensor (example for software trigger mode and one-shot conversion mode)

Figure 11-32. Setup when temperature sensor output/internal reference voltage output is selected

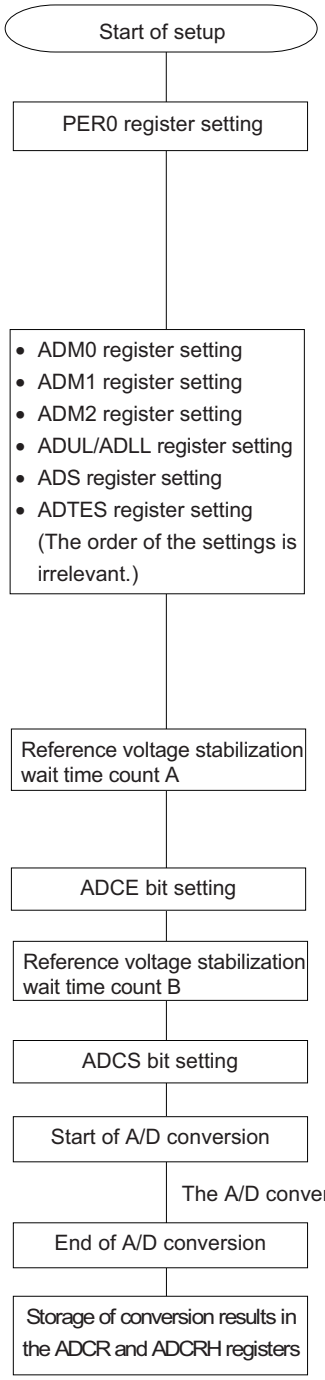


Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

Caution Operation is possible only in HS (high-speed main) mode.

11.7.5 Setting up test mode

Figure 11-33. Setting up Test Trigger Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: This is used to specify the select mode.
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode.
  ADSCM bit: This is used to specify the one-shot conversion mode.
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select for the reference voltage.
  ADRCK bit: This is used to set the range for the A/D conversion result comparison value generated by the interrupt signal to AREA2.
  ADTYP bit: This is used to specify 10-bit resolution.
- ADUL/ADLL register
  These set ADUL to FFH and ADLL to 00H (initial values).
- ADS register
  ADS4 to ADS0 bits: These are used to set to ANI0.
- ADTES register
  ADTES1, ADTES0 bits: AV_{REFM}/AV_{REFP}

The reference voltage stabilization wait time indicated by A below may be required if the values of the ADREFP1 and ADREFP0 bits are changed.
If the values of ADREFP1 and ADREFP0 are changed to 1 and 0, respectively: A = 5 μs
A wait is not required if the values of ADREFP1 and ADREFP0 are changed to 0 and 0 or 0 and 1, respectively.

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

The A/D conversion end interrupt (INTAD) is generated.[Note]

The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

Caution For the procedure for testing the A/D converter, see 23.3.8 A/D test function.

11.8 SNOOZE Mode Function

In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode, A/D conversion can be performed without operating the CPU by inputting a hardware trigger. This is effective for reducing the operation current.

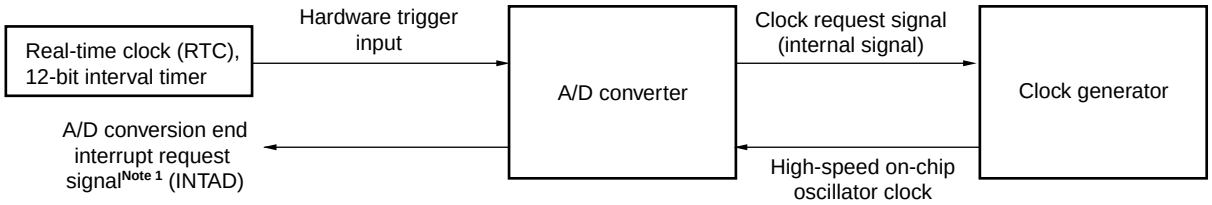
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

In the SNOOZE mode, only the following two conversion modes can be used:

- Hardware trigger wait mode (select mode, one-shot conversion mode)
- Hardware trigger wait mode (scan mode, one-shot conversion mode)

Caution That the SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 11-34. Block Diagram When Using SNOOZE Mode Function



When using the SNOOZE mode function, the initial setting of each register is specified before switching to the STOP mode (for details about these settings, see **11.7.3 Setting up hardware trigger wait mode**Note 2). Just before move to STOP mode, bit 2 (AWC) of A/D converter mode register 2 (ADM2) is set to 1. After the initial settings are specified, bit 0 (ADCE) of A/D converter mode register 0 (ADM0) is set to 1.

If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the system automatically counts up to the A/D power supply stabilization wait time, and then A/D conversion starts.

The SNOOZE mode operation after A/D conversion ends differs depending on whether an interrupt signal is generatedNote 1.

Notes 1. Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL/ADLL register), there is a possibility of no interrupt signal being generated.

2. Be sure to set the ADM1 register to E2H or E3H.

Remark The hardware trigger is INTRTC or INTIT.
Specify the hardware trigger by using the A/D Converter Mode Register 1 (ADM1).

(1) If an interrupt is generated after A/D conversion ends

If the A/D conversion result value is inside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is generated.

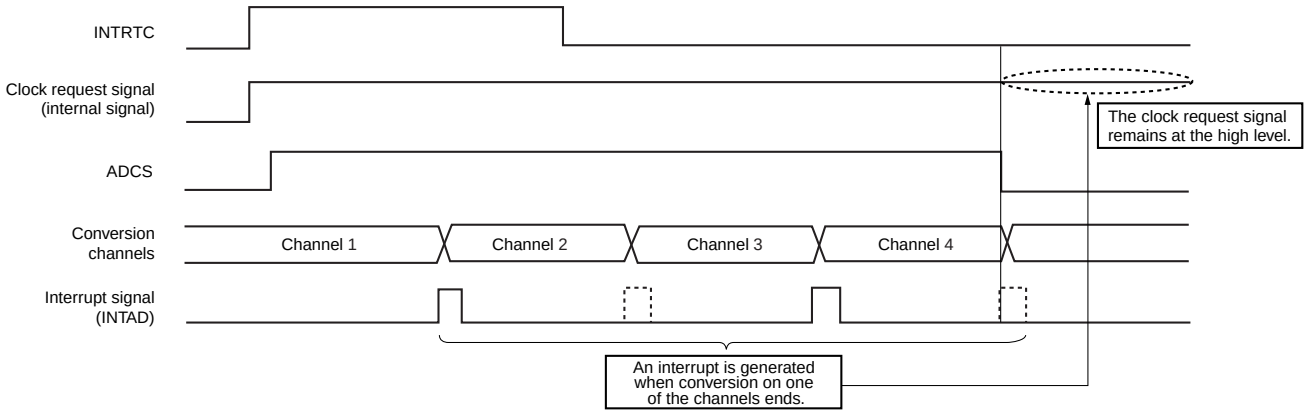
- While in the select mode

  When A/D conversion ends and an A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter returns to normal operation mode from SNOOZE mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of the A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

- While in the scan mode

  If even one A/D conversion end interrupt request signal (INTAD) is generated during A/D conversion of the four channels, the clock request signal remains at the high level, and the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of A/D converter mode register 2 (ADM2) to 0. If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

Figure 11-35. Operation Example When Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)



(2) If no interrupt is generated after A/D conversion ends

If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated.

- While in the select mode
  If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

- While in the scan mode
  If the A/D conversion end interrupt request signal (INTAD) is not generated even once during A/D conversion of the four channels, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

Figure 11-36. Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)

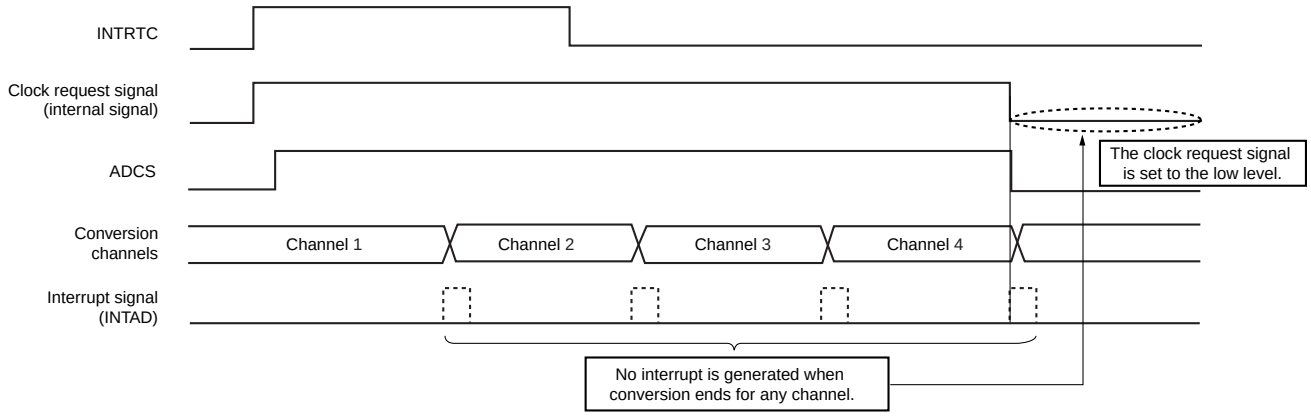
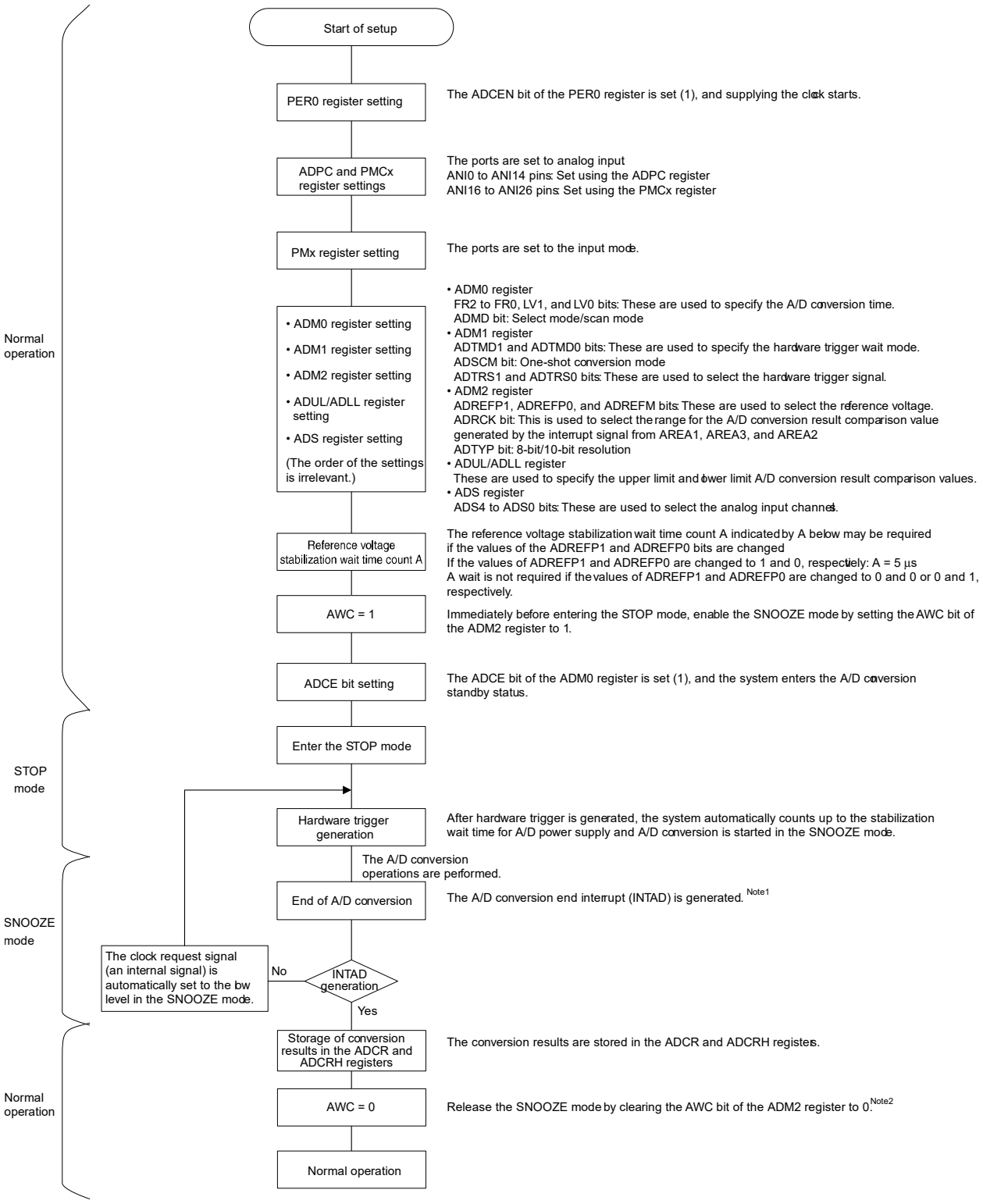


Figure 11-37. Flowchart for Setting up SNOOZE Mode



Notes 1. If the A/D conversion end interrupt request signal (INTAD) is not generated by setting ADRCK bit and ADUL/ADLL register, the result is not stored in the ADCR and ADCRH registers.
The system enters the STOP mode again. If a hardware trigger is input later, A/D conversion operation is again performed in the SNOOZE mode.

2. If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode. Be sure to clear the AWC bit to 0.

11.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(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.

(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.

(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 11-38. Overall Error

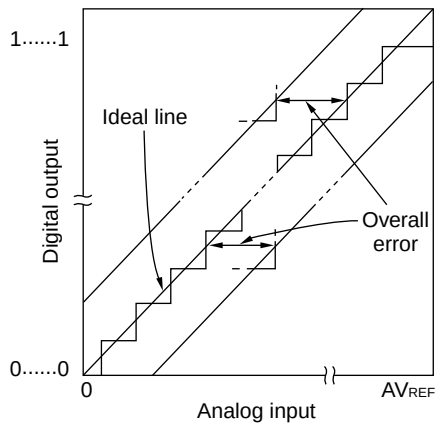
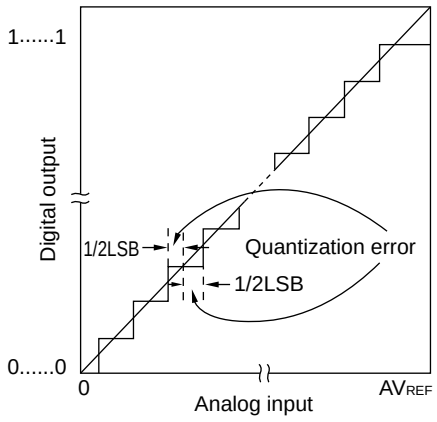


Figure 11-39. Quantization Error



(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.

(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.

(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.

(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 11-40. Zero-Scale Error

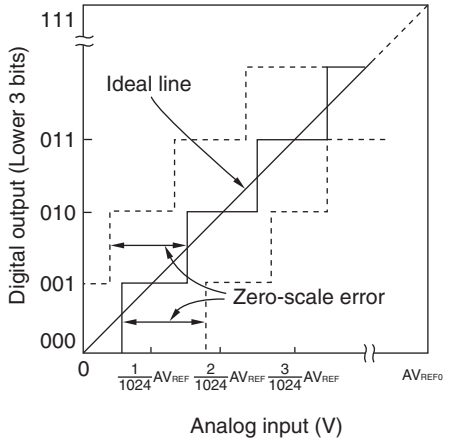


Figure 11-41. Full-Scale Error

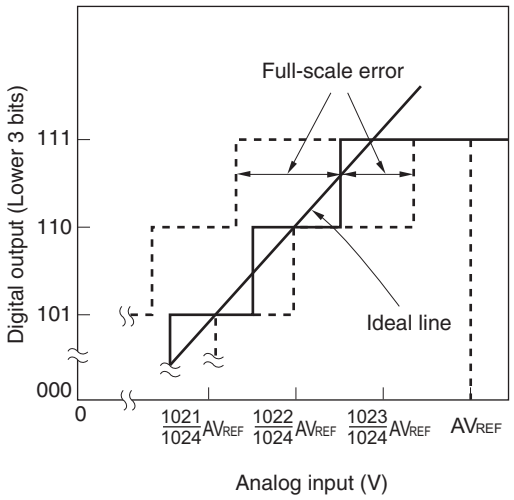


Figure 11-42. Integral Linearity Error

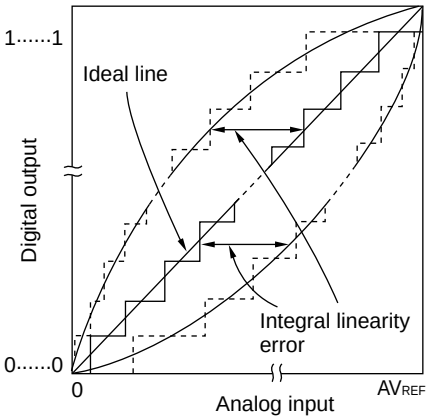
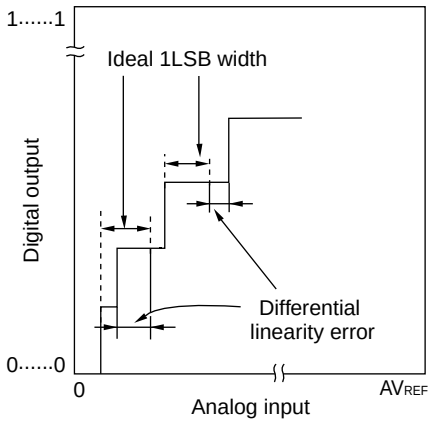


Figure 11-43. Differential Linearity Error



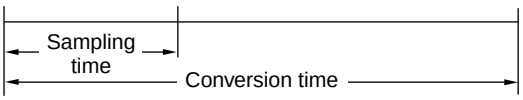
(8) Conversion time

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.

(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.



11.10 Cautions for A/D Converter

(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.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

(2) Input range of ANI0 to ANI7 and ANI19 pins

Observe the rated range of the ANI0 to ANI7 and ANI19 pins input voltage. If a voltage exceeding V_{DD} and AV_{REFP} or below V_{SS} and AV_{REFM} or lower (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.

When internal reference voltage (1.45 V) is selected as the reference voltage for the + side of the A/D converter, do not input voltage exceeding internal reference voltage (1.45 V) to a pin selected by the ADS register. However, it is no problem that a pin not selected by the ADS register is input voltage exceeding the internal reference voltage (1.45 V).

Caution The internal reference voltage (1.45 V) can be selected only in HS (high-speed main) mode.

(3) Conflicting operations

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion

The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

<2> Conflict between the ADCR or ADCRH register write and the A/D converter mode register 0 (ADM0) write, the analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion

The ADM0, ADS, or ADPC registers write has priority. The ADCR or ADCRH register write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REFP}, V_{DD}, ANI0 to ANI7 and ANI19 pins.

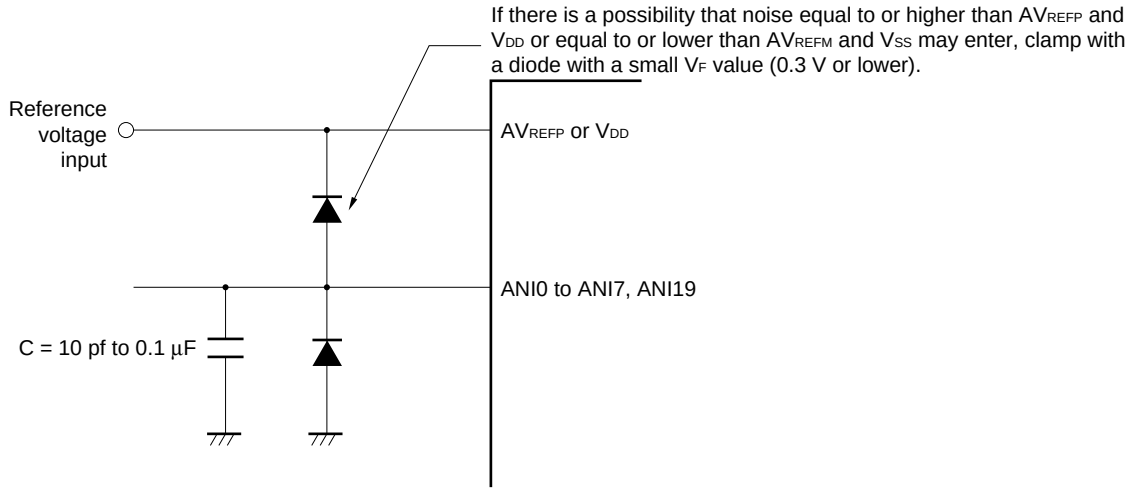
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external capacitor as shown in Figure 11-44 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.

Figure 11-44. Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> The analog input pins (ANI0 to ANI7 and ANI19) are also used as input port pins (P20 to P27 and P120). When A/D conversion is performed with any of the ANI0 to ANI17 and ANI19 pins selected, do not access P20 to P27 and P120 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

(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 100 pF to the ANI0 to ANI7 and ANI19 pins (see **Figure 11-45**).

The sampling capacitor may be being charged while the setting of the ADCS bit is 0 and immediately after sampling is restarted and so is not defined at these times. Accordingly, the state of conversion is undefined after charging starts in the next round of conversion after the value of the ADCS bit has been 1 or when conversion is repeated. Thus, to secure full charging regardless of the size of fluctuations in the analog signal, ensure that the output impedances of the sources of analog inputs are low or secure sufficient time for the completion of conversion.

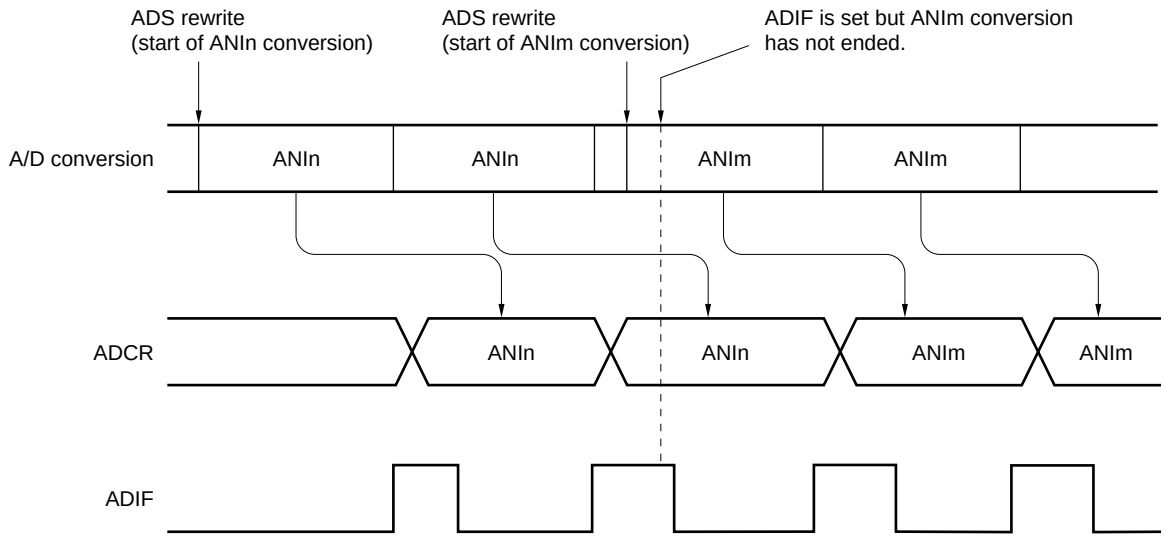
(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF flag for the pre-change analog input may be set just before the ADS register rewrite. Caution is therefore required since, at this time, when ADIF flag is read immediately after the ADS register rewrite, ADIF flag is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

Figure 11-45. Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

While in the software trigger mode or hardware trigger no-wait mode, 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 1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), A/D port configuration register (ADPC), and port mode control register (PMC), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, ADPC, or PMC register. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 11-46. Internal Equivalent Circuit of ANIn Pin

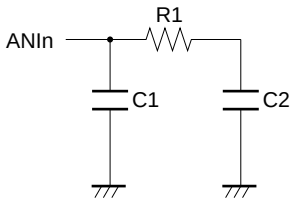


Table 11-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REFP}, V_{DD}	ANIn Pins	R1 [kΩ]	C1 [pF]	C2 [pF]
3.6 V ≤ V_{DD} ≤ 5.5 V	ANI0 to ANI7	14	8	2.5
	ANI19	18	8	7.0
2.4 V ≤ V_{DD} < 3.6 V	ANI0 to ANI7	632	8	2.5
	ANI19	902	8	7.0

Remark The resistance and capacitance values shown in Table 11-6 are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the AV_{REFP} and V_{DD} voltages stabilize.

CHAPTER 12 SERIAL ARRAY UNIT

Serial array unit has two serial channels. Each channel can achieve Simplified SPI (CSI[Note]), UART, and simplified I^{2}C communication.

Function assignment of each channel supported by the RL78/G1C is as shown below.

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.

Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	CSI00	UART0	IIC00
1	CSI01		IIC01

12.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/G1C 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. $f_{CLK}/2$ (CSI00 only)
  Max. $f_{CLK}/4$
  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only CSI00 can be specified for asynchronous reception.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 30, 31 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).

[Data transmission/reception]

- Data length of 7, 8, or 9 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

12.1.3 Simplified I²C (IIC00, IIC01)

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, IIC01) Communication**.

[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.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Parity error (ACK error), or overrun error

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Clock stretch detection functions

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See the processing flow in **12.7.3 (2)** for details.

Remark To use an I²C bus of full function, see **CHAPTER 13 SERIAL INTERFACE IICA**.

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	9 bits
Buffer register	Lower 9 bits of serial data register mn (SDRmn)[Note]
Serial clock I/O	SCK00, SCK01 pins (for simplified SPI), SCL00, SCL01 pins (for simplified I^2C)
Serial data input	SI00, SI01 pins (for simplified SPI), RxD0 pin (for UART)
Serial data output	SO00, SO01 pins (for simplified SPI), TxD0 pin (for UART)
Serial data I/O	SDA00, SDA01 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Serial standby control register m (SSCm) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn)
	• Port input mode registers 3, 5 (PIM3, PIM5) • Port output mode registers 0, 3, 5, 7 (POM0, POM3, POM5, POM7) • Port mode registers 0, 3, 5, 7 (PM0, PM3, PM5, PM7) • Port registers 0, 3, 5, 7 (P0, P3, P5, P7)

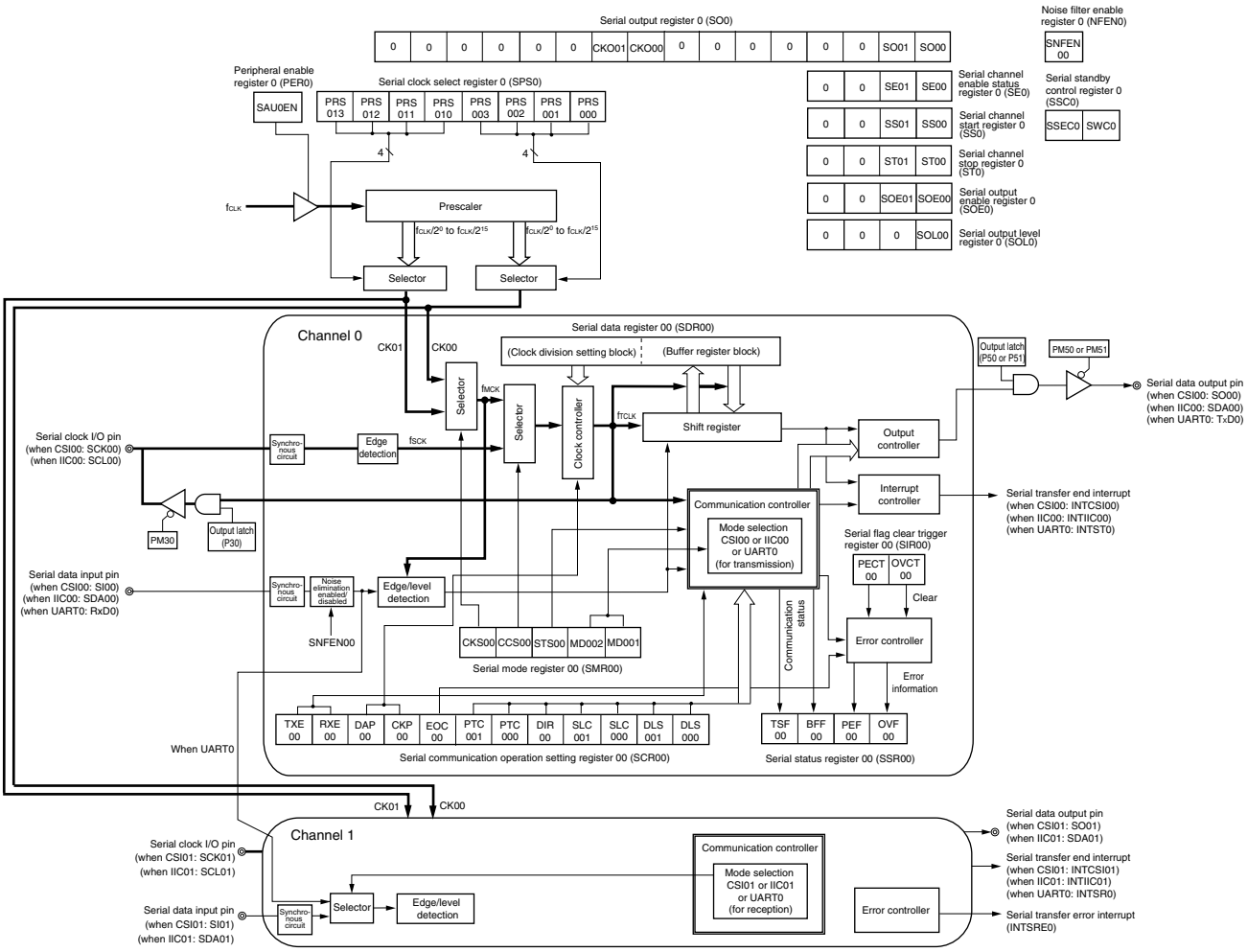
Note The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

- CSIp communication … SIOp (CSIp data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)
- IICr communication … SIOr (IICr data register)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), q: UART number (q = 0), r: IIC number (r = 00, 01)

Figure 12-1 shows the block diagram of the serial array unit.

Figure 12-1. Block Diagram of Serial Array Unit



12.2.1 Shift register

This is a 9-bit register that converts parallel data into serial data or vice versa.

In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used.

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 the shift register, use the lower 9 bits of serial data register mn (SDRmn).

	8	7	6	5	4	3	2	1	0
Shift register									

12.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits)[Note 1] or bits 7 to 0 (lower 8 bits) function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

When data is received, parallel data converted by the shift register is stored in the lower 9 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 9 bits.

The data stored in the lower 9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLSmn1) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of SDRmn register)

The SDRmn register can be read or written in 16-bit units.

The lower 8 bits of the SDRmn register can be read or written[Note 2] as the following SFR, depending on the communication mode.

- CSIp communication … SIOp (CSIp data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)
- IICr communication … SIOr (IICr data register)

Reset signal generation clears the SDRmn register to 0000H.

Notes 1. Only the UART0 can be specified for the 9-bit data length.

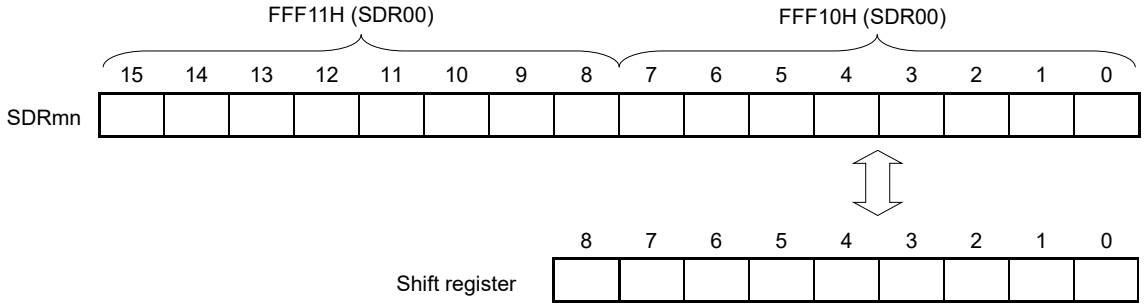
2. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remarks 1. After data is received, “0” is stored in bits 0 to 8 in bit portions that exceed the data length.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), q: UART number (q = 0), r: IIC number (r = 00, 01)

Figure 12-2. Format of Serial Data Register mn (SDRmn) (mn = 00, 01)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W



Remark For the function of the higher 7 bits of the SDRmn 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 m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Serial standby control register m (SSCm)
- Noise filter enable register 0 (NFEN0)
- Port input mode registers 3, 5 (PIM3, PIM5)
- Port output mode registers 0, 3, 5, 7 (POM0, POM3, POM5, POM7)
- Port mode registers 0, 3, 5, 7 (PM0, PM3, PM5, PM7)
- Port registers 0, 3, 5, 7 (P0, P3, P5, P7)

Remark m: Unit number (m = 0), 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 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	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

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

Cautions 1. When setting serial array unit m, be sure to first set the following registers with the SAUmEN bit set to 1. If SAUmEN = 0, control registers of serial array unit m become default values and writing to them is ignored (except for the noise filter enable register 0 (NFEN0), port input mode registers 3, 5 (PIM3, PIM5), port output mode registers 0, 3, 5, 7 (POM0, POM3, POM5, POM7), port mode registers 0, 3, 5, 7 (PM0, PM3, PM5, PM7), and port registers port registers 0, 3, 5, 7 (P0, P3, P5, P7)).

- **Serial clock select register m (SPSm)**
- **Serial mode register mn (SMRmn)**
- **Serial communication operation setting register mn (SCRmn)**
- **Serial data register mn (SDRmn)**
- **Serial flag clear trigger register mn (SIRmn)**
- **Serial status register mn (SSRmn)**
- **Serial channel start register m (SSm)**
- **Serial channel stop register m (STm)**
- **Serial channel enable status register m (SEm)**
- **Serial output enable register m (SOEm)**
- **Serial output level register m (SOLm)**
- **Serial output register m (SOm)**
- **Serial standby control register m (SSCm)**

2. Be sure to clear bits 1, 3, 6 to 0.

12.3.2 Serial clock select register m (SPSm)

The SPSm register is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of the SPSm register , and CKm0 is selected by bits 3 to 0.

Rewriting the SPSm register is prohibited when the register is in operation (when SEmn = 1).

The SPSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SPSm register can be set with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears the SPSm register to 0000H.

Figure 12-4. Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Section of operation clock (CKmk) Note					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 kHz	305 Hz	610 Hz	732 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 m (STm) = 0003H) the operation of the serial array unit (SAU).

Caution Be sure to clear bits 15 to 8 to "0".

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

2. m: Unit number (m = 0)

3. k = 0, 1

12.3.3 Serial mode register mn (SMRmn)

The SMRmn register is a register that sets 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 simplified I²C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

The SMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SMRmn register to 0020H.

Figure 12-5. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00), F0112H, F0113H (SMR01) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn Note	0	SISmn0 Note	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCSmn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn 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 CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STSmn	Selection of start trigger source
0	Only software trigger is valid (selected for simplified SPI (CSI), UART transmission, and simplified I²C).
1	Valid edge of the RxDq pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Note The SMR01 register only.

Caution **Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00 register) to "0". Be sure to set bit 5 to "1".**

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), q: UART number (q = 0), r: IIC number (r = 00, 01)

Figure 12-5. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00), F0112H, F0113H (SMR01) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn[Note]	0	SIS mn0[Note]	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0 Note	Controls inversion of level of receive data of channel n in UART mode
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.

MD mn2	MD mn1	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

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out.	

Note The SMR01 register only.

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00 register) to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01),
q: UART number (q = 0), r: IIC number (r = 00, 01)

12.3.4 Serial communication operation setting register mn (SCRmn)

The SCRmn register is a communication operation setting register 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 SCRmn register is prohibited when the register is in operation (when SEmn = 1).

The SCRmn register can be set by a 16-bit memory manipulation instruction.

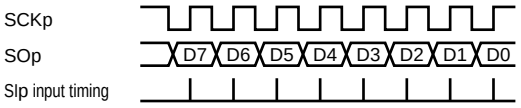
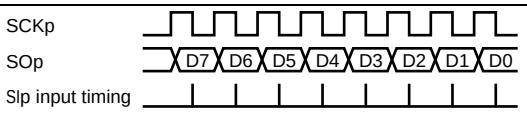
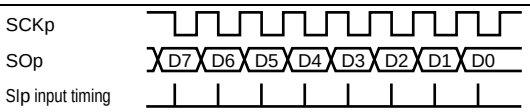
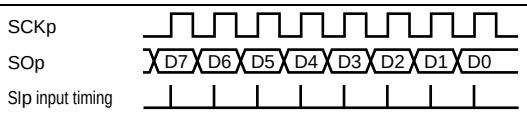
Reset signal generation sets the SCRmn register to 0087H.

Figure 12-6. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) , F011AH, F011BH (SCR01) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 Note 1	SLC mn0	0	1	DLSm n1	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	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 DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

EOC mn	Selection of masking of error interrupt signal (INTSRE0)
0	Masks error interrupt INTSRE0 (INTSR0 is not masked).
1	Enables generation of error interrupt INTSRE0 (INTSR0 is masked if an error occurs).
Set EOCmn = 0 in the simplified SPI (CSI) mode, simplified I^2C mode, and during UART transmission Note 2. Set EOCmn = 1 during UART reception.	

Notes 1. The SCR00 register only.

2. When using CSI01 not with EOC01 = 0, error interrupt INTSRE0 may be generated.

Caution Be sure to clear bits 3, 6, and 11 to "0" (Also clear bit 5 of the SCR01 register to 0.). Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-6. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00) , F011AH, F011BH (SCR01) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 Note 1	SLC mn0	0	1	DLSm n1	DLS mn0

PTC mn1	PTC mn0	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 PTCmn1, PTCmn0 = 0, 0 in the simplified SPI (CSI) mode and simplified I²C mode.			

DIR mn	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 DIRmn = 0 in the simplified I²C mode.	

SLCm n1 Note 1	SLC mn0	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 (mn = 00, 02, 10, 12 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 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I²C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the simplified SPI (CSI) mode. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) or 2 bits (SLCmn1, SLCmn0 = 1, 0) during UART transmission.		

DLSm n1	DLS mn0	Setting of data length in simplified SPI (CSI) and UART modes
0	1	9-bit data length (stored in bits 0 to 8 of the SDRmn register) (settable in UART mode only)
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)
Other than above		Setting prohibited
Be sure to set DLSmn1, DLSmn0 = 1, 1 in the simplified I²C mode.		

Notes 1. The SCR00 register only.

2. 0 is always added regardless of the data contents.

Caution Be sure to clear bits 3, 6, and 11 to "0". (Also clear bit 5 of the SCR01 register to 0.). Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

12.3.5 Serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) of SDR00, SDR01 function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of the SDRmn register is used as the transfer clock.

If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of SDR00, SDR01 to 0000000B. The input clock f_{SCK} (slave transfer in simplified SPI (CSI) mode) from the SCKp pin is used as the transfer clock.

The lower 9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 9 bits.

The SDRmn register can be read or written in 16-bit units.

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 9 bits of the SDRmn register. When the SDRmn register is read during operation, 0 is always read.

Reset signal generation clears the SDRmn register to 0000H.

Figure 12-7. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W

	FFF11H (SDR00)								FFF10H (SDR00)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

SDRmn[15:9]							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 SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

2. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I²C is used. Set SDRmn[15:9] to 0000001B or greater.

3. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remarks 1. For the function of the lower 9 bits of the SDRmn register, see **12.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.3.6 Serial flag clear trigger register mn (SIRmn)

The SIRmn register is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.

The SIRmn register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SIRmn register can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears the SIRmn register to 0000H.

Figure 12-8. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00), F010AH, F010BH (SIR01) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn[Note]	PECTmn	OVCTmn

FECTmn	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PECTmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Note The SIR01 register only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00 register) to “0”.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

2. When the SIRmn register is read, 0000H is always read.

12.3.7 Serial status register mn (SSRmn)

The SSRmn register is a register that indicates the communication status and error occurrence status of channel n. The errors indicated by this register are a framing error, parity error, and overrun error.

The SSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSRmn register can be set with an 8-bit memory manipulation instruction with SSRmnL.

Reset signal generation clears the SSRmn register to 0000H.

Figure 12-9. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEFm n[Note]	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.

<Clear conditions>

- The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended).
- Communication ends.

<Set condition>

- Communication starts.

BFF mn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.

<Clear conditions>

- Transferring transmit data from the SDRmn register to the shift register ends during transmission.
- Reading receive data from the SDRmn register ends during reception.
- The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled).

<Set conditions>

- Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode).
- Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode).
- A reception error occurs.

Note The SSR01 register only.

Caution When the simplified SPI (CSI) is performing reception operations in the SNOOZE mode (SWCm = 1), the BFFmn flag will not change.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

Figure 12-9. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00), F0102H, F0103H (SSR01) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSFmn	BFFmn	0	0	FEFmn[Note]	PEFmn	OVFmn

FEFmn[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 FECTmn bit of the SIRmn register.

<Set condition>
- A stop bit is not detected when UART reception ends.

PEFmn	Parity error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception) or ACK is not detected (during I^2C transmission).

<Clear condition>
- 1 is written to the PECTmn bit of the SIRmn 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).

OVFmn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs

<Clear condition>
- 1 is written to the OVCTmn bit of the SIRmn register.

<Set condition>
- Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn 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 The SSR01 register only.

Cautions 1. If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVEmn = 1) is detected.

2. When the simplified SPI (CSI) is performing reception operations in the SNOOZE mode (SWCm = 1), the OVFmn flag will not change.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.3.8 Serial channel start register m (SSm)

The SSm register is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (Operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn = 1.

The SSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSm register can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

Figure 12-10. Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS01	SS00

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status Note.

Note If set the SSmn = 1 to during a communication operation, will wait status to stop the communication. At this time, holding status value of control register and shift register, SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Cautions 1. Be sure to clear bits 15 to 2 to "0".

2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after 4 or more f_{CLK} clocks have elapsed.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

2. When the SSm register is read, 0000H is always read.

12.3.9 Serial channel stop register m (STm)

The STm register is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

The STm register can set written by a 16-bit memory manipulation instruction.

The lower 8 bits of the STm register can be set with a 1-bit or 8-bit memory manipulation instruction with STmL.

Reset signal generation clears the STm register to 0000H.

Figure 12-11. Format of Serial Channel Stop Register m (STm)

Address: F0124H, F0125H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST01	ST00

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operation[Note].

Note Communication stops while holding the value of the control register and shift register, and the status of the serial clock I/O pin, serial data output pin, and each error flag (FEFmn: framing error flag, PEFmn: parity error flag, OVFmn: overrun error flag).

Caution Be sure to clear bits 15 to 2 to "0".

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

2. When the STm register is read, 0000H is always read.

12.3.10 Serial channel enable status register m (SEm)

The SEm register indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of the CKOmn bit (serial clock output of channel n) of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial clock pin.

Channel n that stops operation can set the value of the CKOmn bit of the SOm register by software and output its value 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 SEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SEmL.

Reset signal generation clears the SEm register to 0000H.

Figure 12-12. Format of Serial Channel Enable Status Register m (SEm)

Address: F0120H, F0121H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE01	SE00

SEm n	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.3.11 Serial output enable register m (SOEm)

The SOEm register is a register that is used to enable or stop output of the serial communication operation of each channel.

Channel n that enables serial output cannot rewrite by software the value of the SOmn bit of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform of the start condition and stop condition can be created by software.

The SOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears the SOEm register to 0000H.

Figure 12-13. Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE01	SOE00

SOEmn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear bits 15 to 2 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.3.12 Serial output register m (SOm)

The SOm register is a buffer register for serial output of each channel.

The value of the SOmn bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

The SOmn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKOmn bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding CKOmn and SOmn bits to “1”.

The SOm register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears the SOm register to 0303H.

Figure 12-14. Format of Serial Output Register m (SOm)

Address: F0128H, F0129H After reset: 0303H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	CKO 01	CKO 00	0	0	0	0	0	0	SO 01	SO 00

CKO mn	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SO mn	Serial data output of channel n
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Caution Be sure to clear bits 15 to 10 and 7 to 2 to “0”.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.3.13 Serial output level register m (SOLm)

The SOLm register is a register that is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 0 for corresponding bit in the simplified SPI (CSI) mode and simplifies I^2C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting the SOLm register is prohibited when the register is in operation (when SEmn = 1).

The SOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOLm register can be set with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears the SOLm register to 0000H.

Figure 12-15. Format of Serial Output Level Register m (SOLm)

Address: F0134H, F0135H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL00

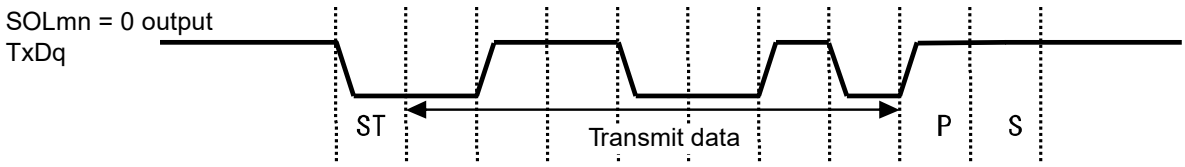
SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Be sure to clear bits 15 to 1 to "0".

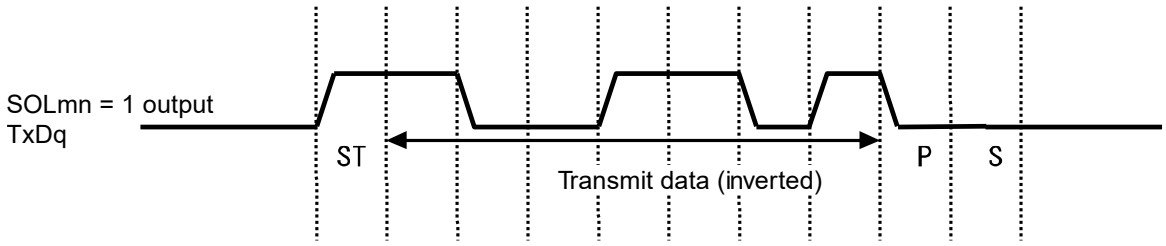
Figure 12-16 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 12-16. Examples of Reverse Transmit Data

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0), n: Channel number (n = 0)

12.3.14 Serial standby control register m (SSCm)

The SSCm register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 serial data.

The SSCm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSCm register can be set with an 8-bit memory manipulation instruction with SSCmL.

Reset signal generation clears the SSCm register to 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.

- **When using CSI00 : 1 Mbps**

Figure 12-17. Format of Serial Standby Control Register m (SSCm)

Address: F0138H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSECm	SWCm

SSECm	Selection of whether to enable or stop the generation of communication error interrupts in the SNOOZE mode
0	Enable the generation of error interrupts (INTSRE0).
1	Stop the generation of error interrupts (INTSRE0).
Be sure to clear the SSECm bit to 0 in the RL78/G1C.	

SWCm	Selection of whether to enable or stop the startup of CSI00 while in the STOP mode
0	Stop the startup of reception while in the STOP mode.
1	Enable the startup of reception while in the STOP mode. (During asynchronous CSI00 reception is enabled.)

Caution Setting SSECm, SWCm = 1, 0 is prohibited.

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 to each channel.

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 is performed with the operation clock (f_{MCK}) of the target channel, 2-clock match detection is performed. When the noise filter is disabled, only synchronization is performed with the operation clock (f_{MCK}) of 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.

Note For details, see **6.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)** and **6.5.2 Start timing of counter**.

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 the other than RxD0 pin.	

Caution Be sure to clear bits 7 to 1 to "0".

12.3.16 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 (PMxx), port register (Pxx), port input mode register (PIMxx), port output mode register (POMxx)).

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.4 Port input mode registers (PIMxx)**, **4.3.5 Port output mode registers (POMxx),** and **4.3.6 Port mode control registers (PMCxx)**.

Specifically, using a port pin with a multiplexed serial data or serial clock output function (e.g. P51/INTP2/SO00/TXD0/TOOLTxD/TI01/TO01) for serial data or serial clock output, requires setting the corresponding bits in the port output mode register (POMxx) and port mode register (PMxx) to 0, and the corresponding bit in the port register (Pxx) 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 (POMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Example: P51/INTP2/SO00/TXD0/TOOLTxD/TI01/TO01 is to be used for serial data output
Set the POM51 bit of port output mode register 5 to 0.
Set the PM51 bit of port mode register 5 to 0.
Set the P51 bit of port register 5 to 1.

Specifically, using a port pin with a multiplexed serial data or serial clock input function (e.g. P50/INTP1/SI00/RXD0/TOOLRxD/SDA00/(TI02)/(TO02)) for serial data or serial clock input, requires setting the corresponding bit in the port mode register (PMxx) to 1, and the corresponding bit in the port mode control register (PMCxx) to 0.. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.

When the TTL input buffer is selected, set the corresponding bit in the port input mode register (PIMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Example: When P50/INTP1/SI00/RXD0/TOOLRxD/SDA00/(TI02)/(TO02) is to be used for serial data input
Set the PM50 bit of port mode register 5 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 pin for serial interface 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.

To stop the operation of serial array unit, set bit 2 (SAU0EN) to 0.

Figure 12-19. Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0) … Set only the bit of SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	RTCEN ×	0	ADCEN ×	IICA0EN ×	0	SAU0EN 0/1	0	TAU0EN ×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Cautions 1. If SAU0EN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read
Note that this does not apply to the following registers.
- **Noise filter enable register 0 (NFEN0)**
- **Serial standby control register m (SSCm)**
- **Port input mode registers 0, 3, 5, 7 (PIM0, PIM3, PIM5, PIM7)**
- **Port output mode registers 0, 3, 5, 7 (POM0, POM3, POM5, POM7)**
- **Port mode registers 0, 3, 5, 7 (PM0, PM3, PM5, PM7)**
- **Port registers 0, 3, 5, 7 (P0, P3, P5, P7)**

2. Be sure to clear the bits 1, 3, 6 to 0.

Remark ×: 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 12-20. Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm) … This register is a trigger register that is used to enable stopping communication/count by each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	STm1 0/1	STm0 0/1

1: Clears the SEmn bit to 0 and stops the communication operation

* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

(b) Serial Channel Enable Status Register m (SEm) … This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SEm1 0/1	SEm0 0/1

0: Operation stops

* The SEm register is a read-only status register, whose operation is stopped by using the STm register. With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm) … This register is a register that is used to enable or stop output of the serial communication operation of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

(d) Serial output register m (SOm) …This register is a buffer register for serial output of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

1: Serial clock output value is "1" 1: Serial data output value is "1"

* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to "1".

Remarks 1. m: Unit number (m = 0), 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) 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]

- 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. $f_{CLK}/2$ (CSI00 only)
  Max. $f_{CLK}/4$
  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

In addition, CSI00 supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only CSI00 can be specified for asynchronous reception.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 30 or 31 ELECTRICAL SPECIFICATIONS**.

The channels supporting simplified SPI (CSI00, CSI01) are channels 0 and 1.

Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	CSI00	UART0	IIC00
1	CSI01		IIC01

Simplified SPI (CSI00, CSI01) 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**.)
- SNOOZE mode function (See **12.5.7**.)

12.5.1 Master transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00	CSI01
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	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 30 or 31 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

(1) Register setting

Figure 12-21. Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00, CSI01) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0/1	0/1	0	0	0	0	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 mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data (Transmit data setting)							

SIOp

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	0/1	0/1	0	0	0	0	0	0	0/1	0/1

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

Remarks 1. m: Unit number (m = 0), n: Channel number (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)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 12-21. Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00, CSI01) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (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 12-22. Initial Setting Procedure for Master Transmission

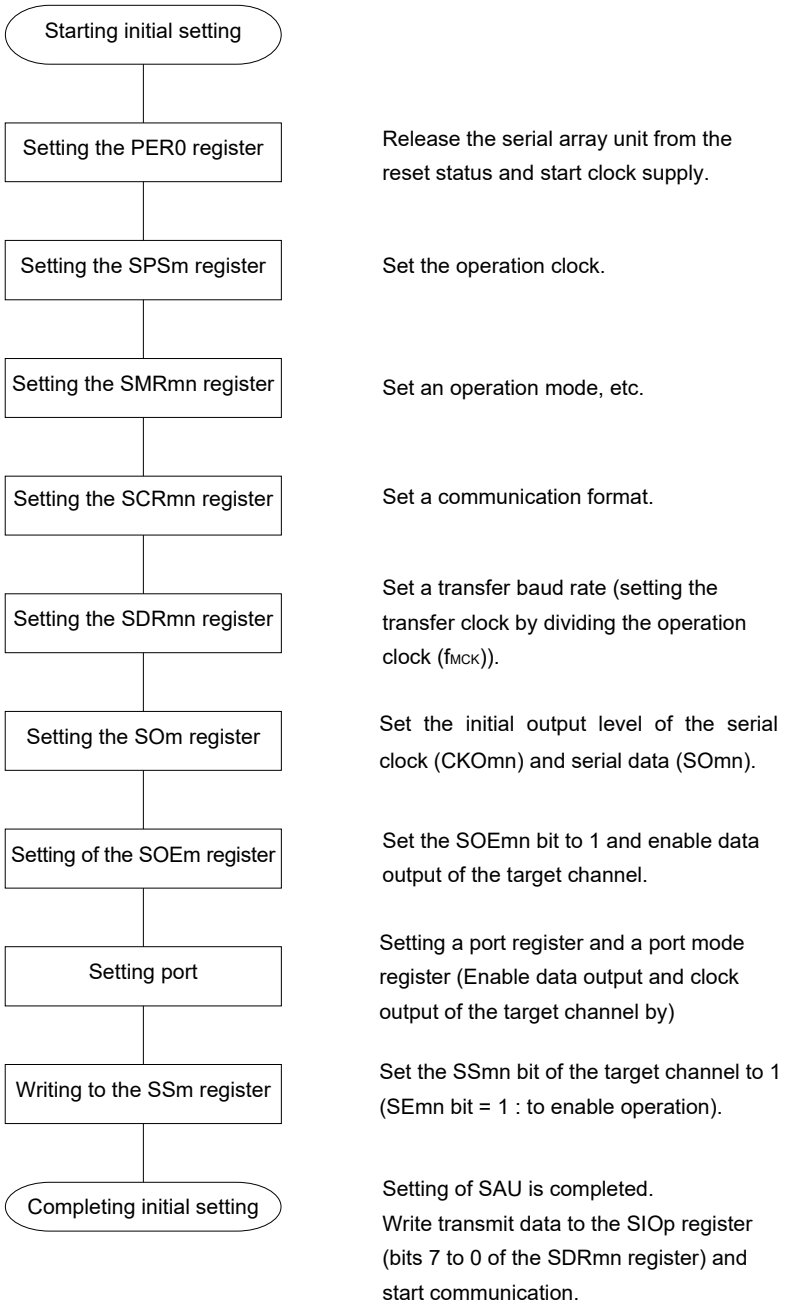


Figure 12-23. Procedure for Stopping Master Transmission

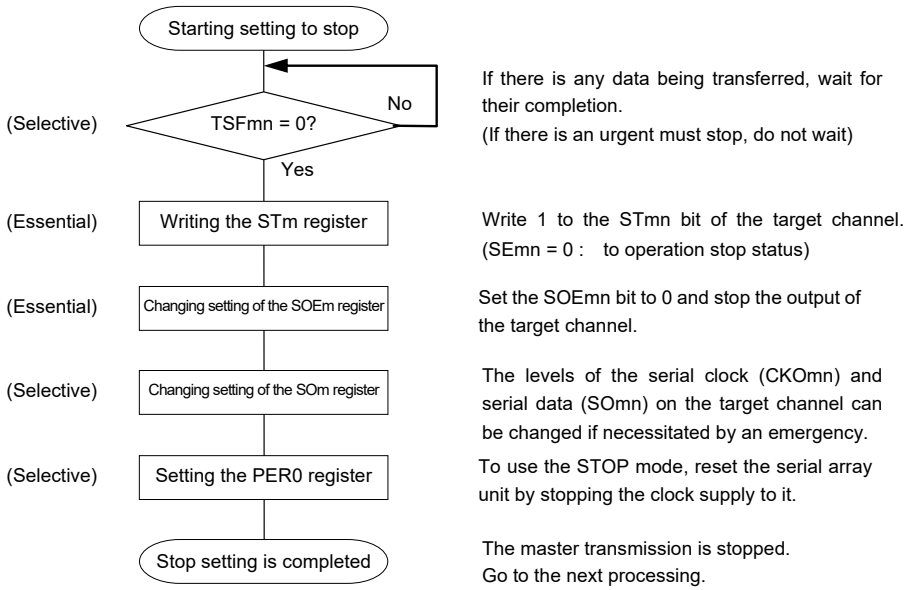
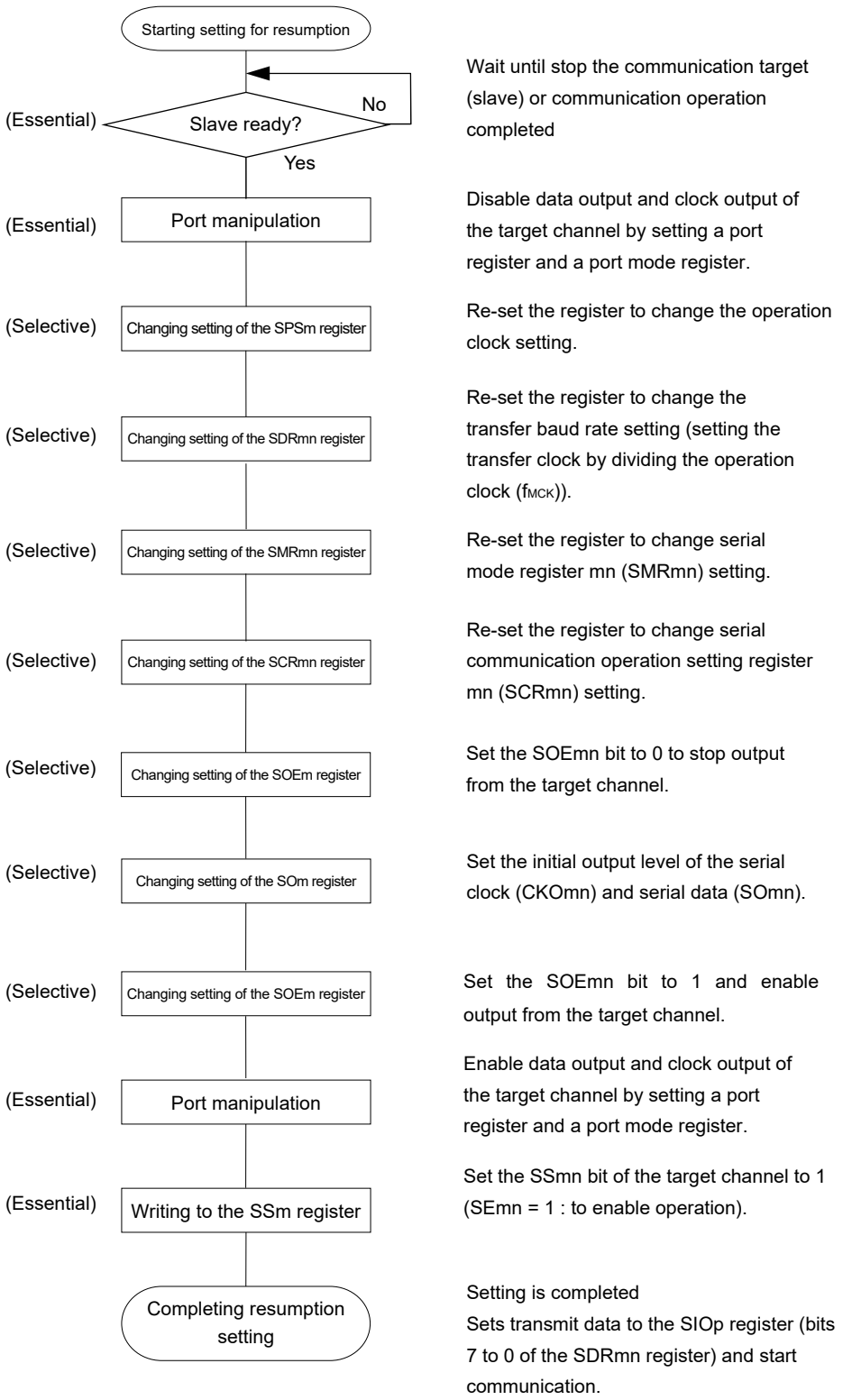


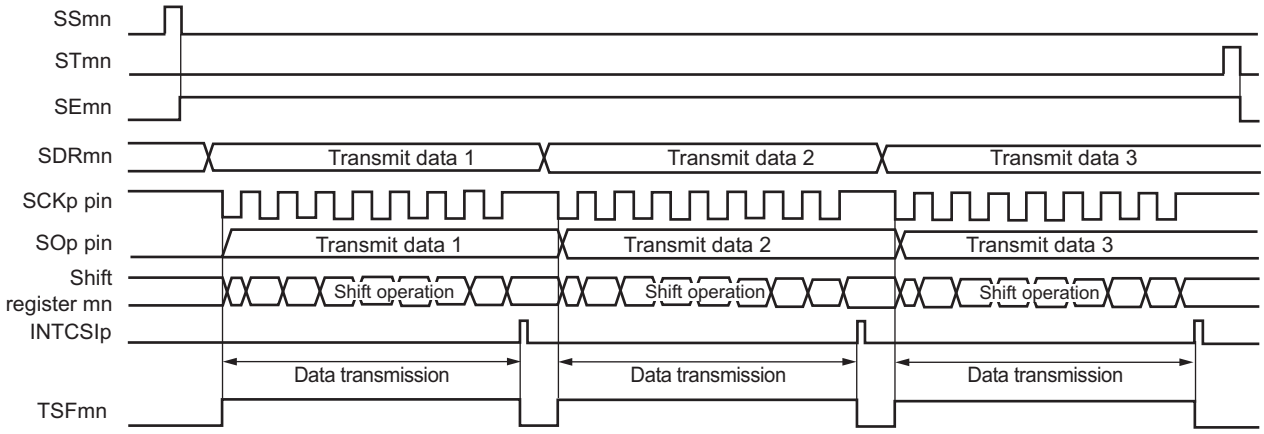
Figure 12-24. Procedure for Resuming Master Transmission



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.

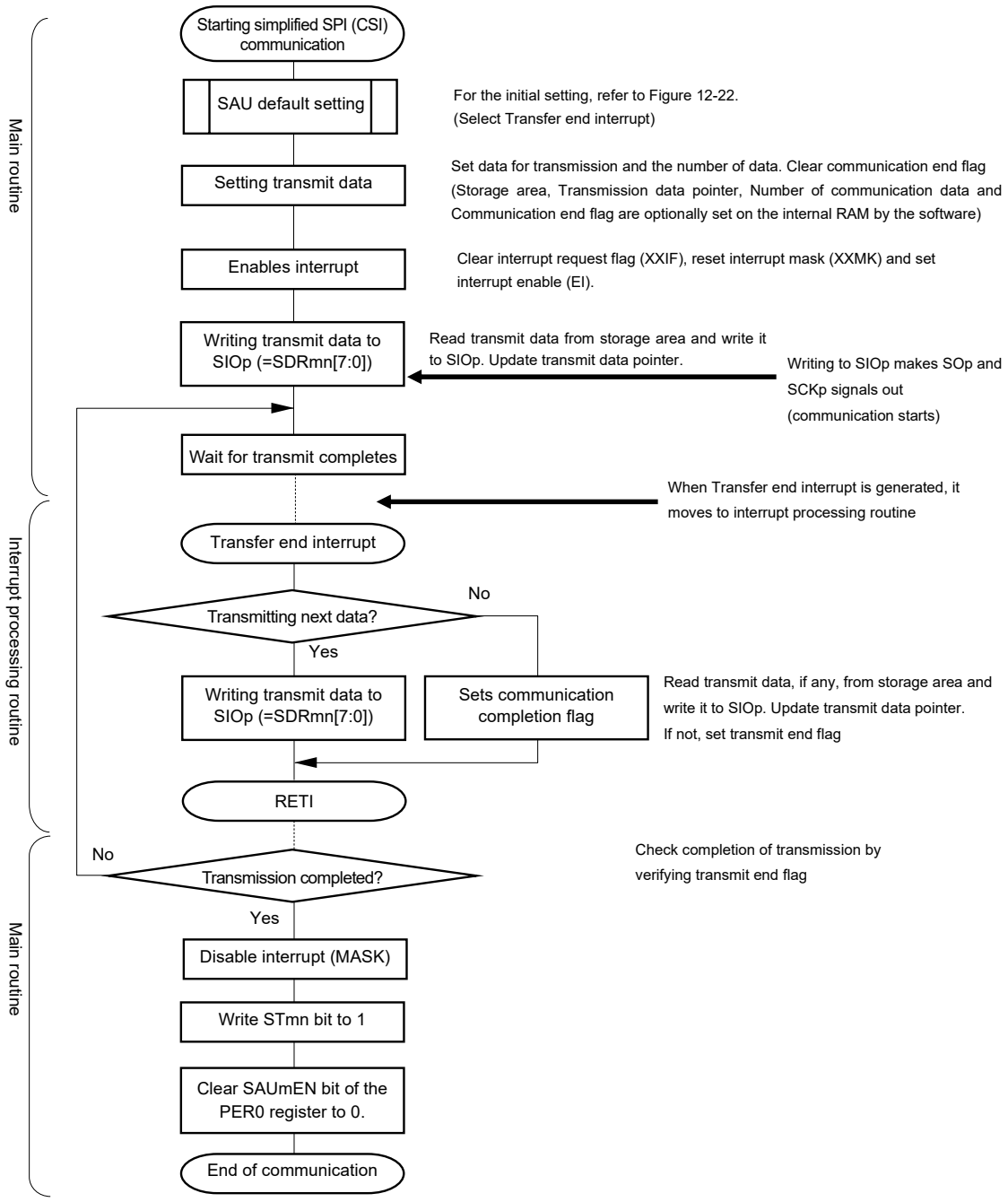
(3) Processing flow (in single-transmission mode)

Figure 12-25. Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



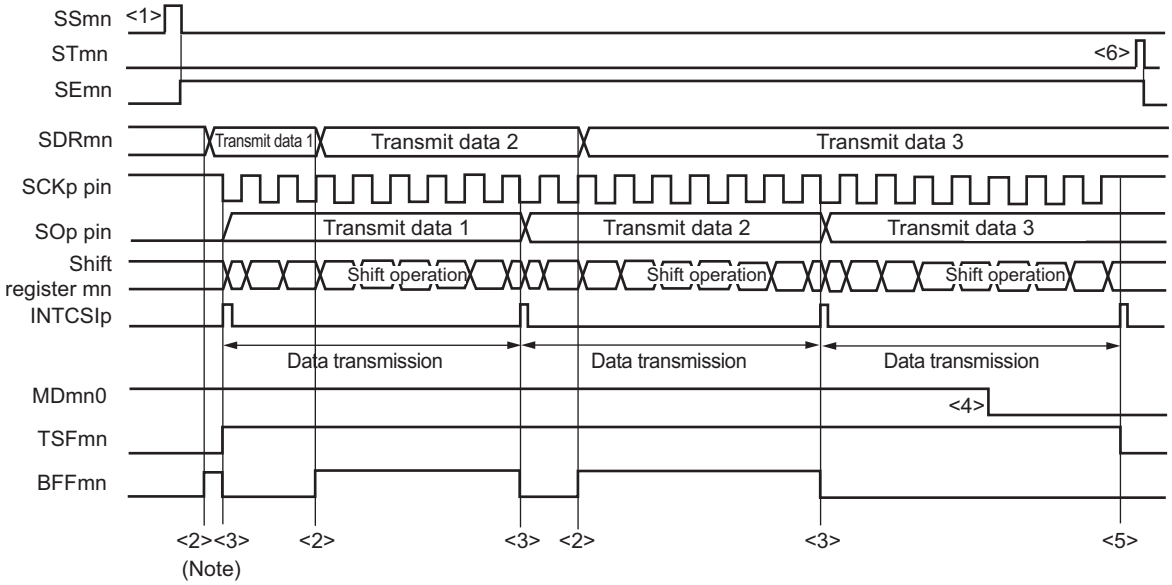
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-26. Flowchart of Master Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 12-27. Timing Chart of Master Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

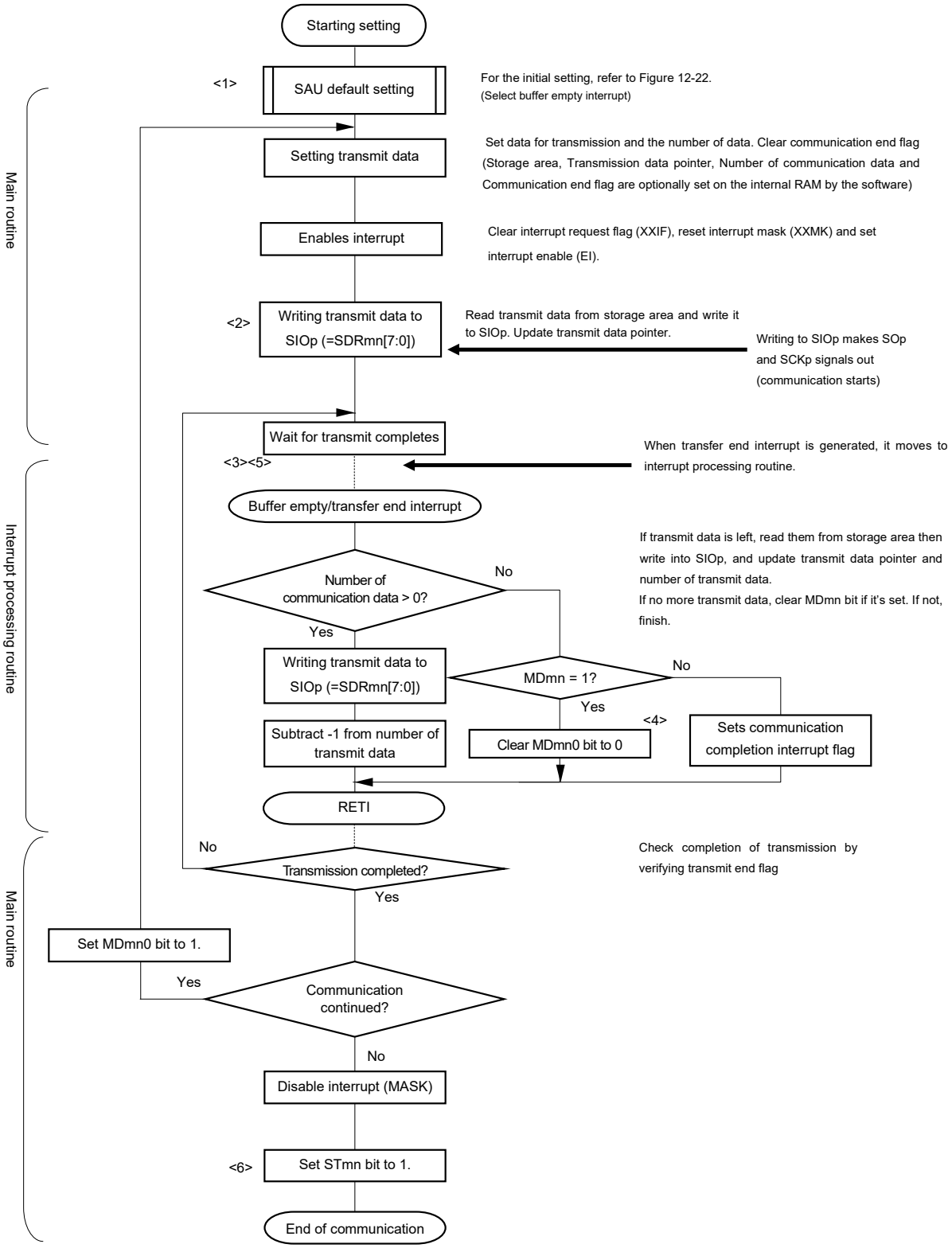


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) 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 m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-28. Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-29 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

12.5.2 Master reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from other device.

Simplified SPI	CSI00	CSI01
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 (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate Note	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 30 or 31 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

(1) Register setting

Figure 12-29. Example of Contents of Registers for Master Reception of Simplified SPI (CSI00, CSI01) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	0	1	0/1	0/1	0	0	0	0	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 mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Receive data (Write FFH as dummy data.)							

SIOp

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	0/1	0/1	0	0	0	0	0	0	×	×

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

2. ☐: Setting is fixed in the simplified SPI (CSI) 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

Figure 12-29. Example of Contents of Registers for Master Reception of Simplified SPI (CSI00, CSI01) (2/2)

(e) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 ×

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

2. □: Setting is fixed in the simplified SPI (CSI) 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 12-30. Initial Setting Procedure for Master Reception

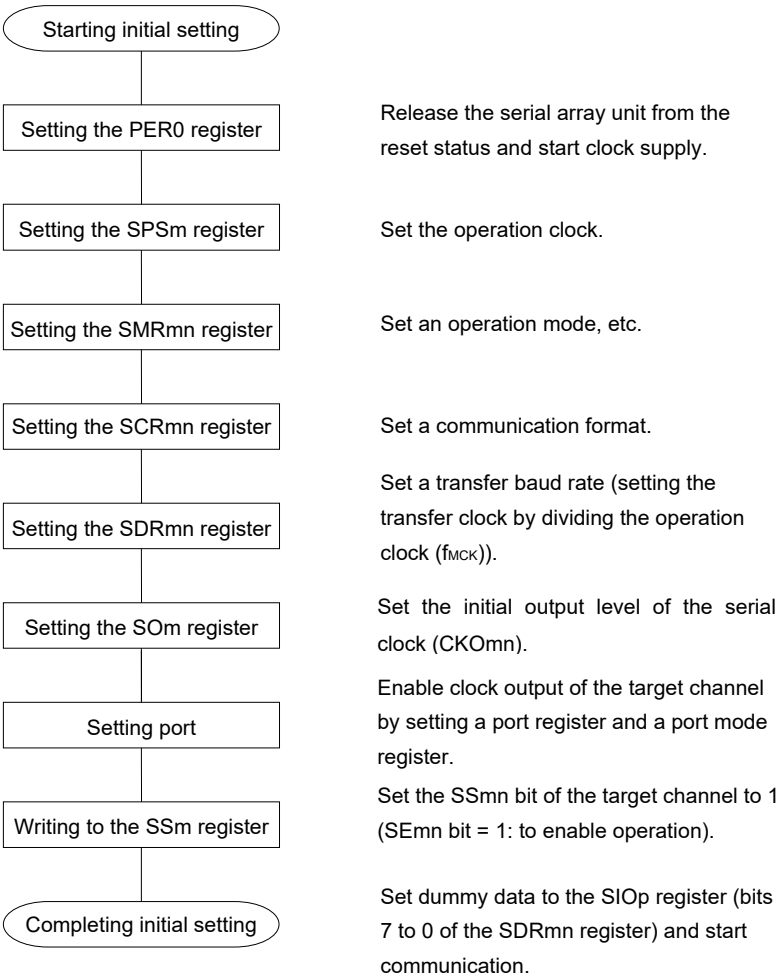


Figure 12-31. Procedure for Stopping Master Reception

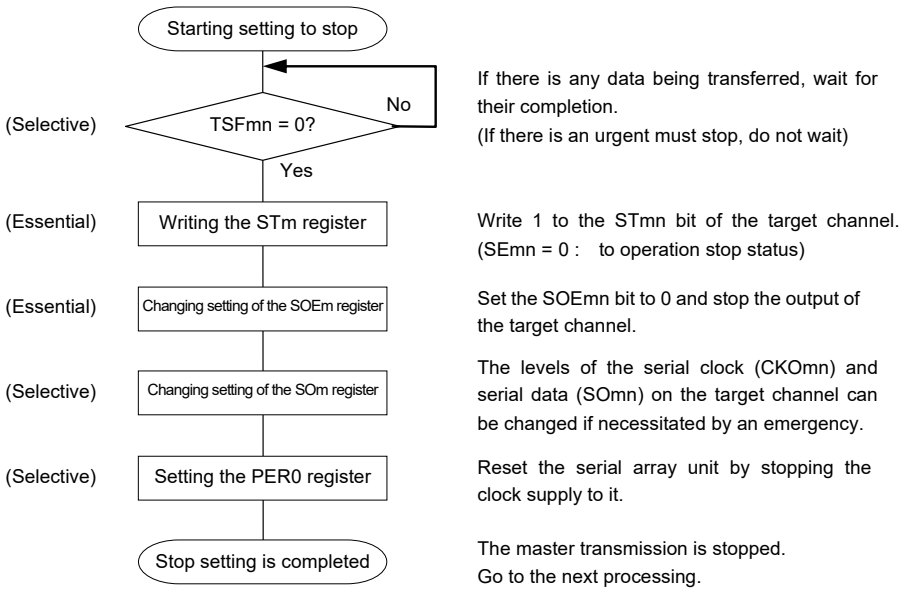
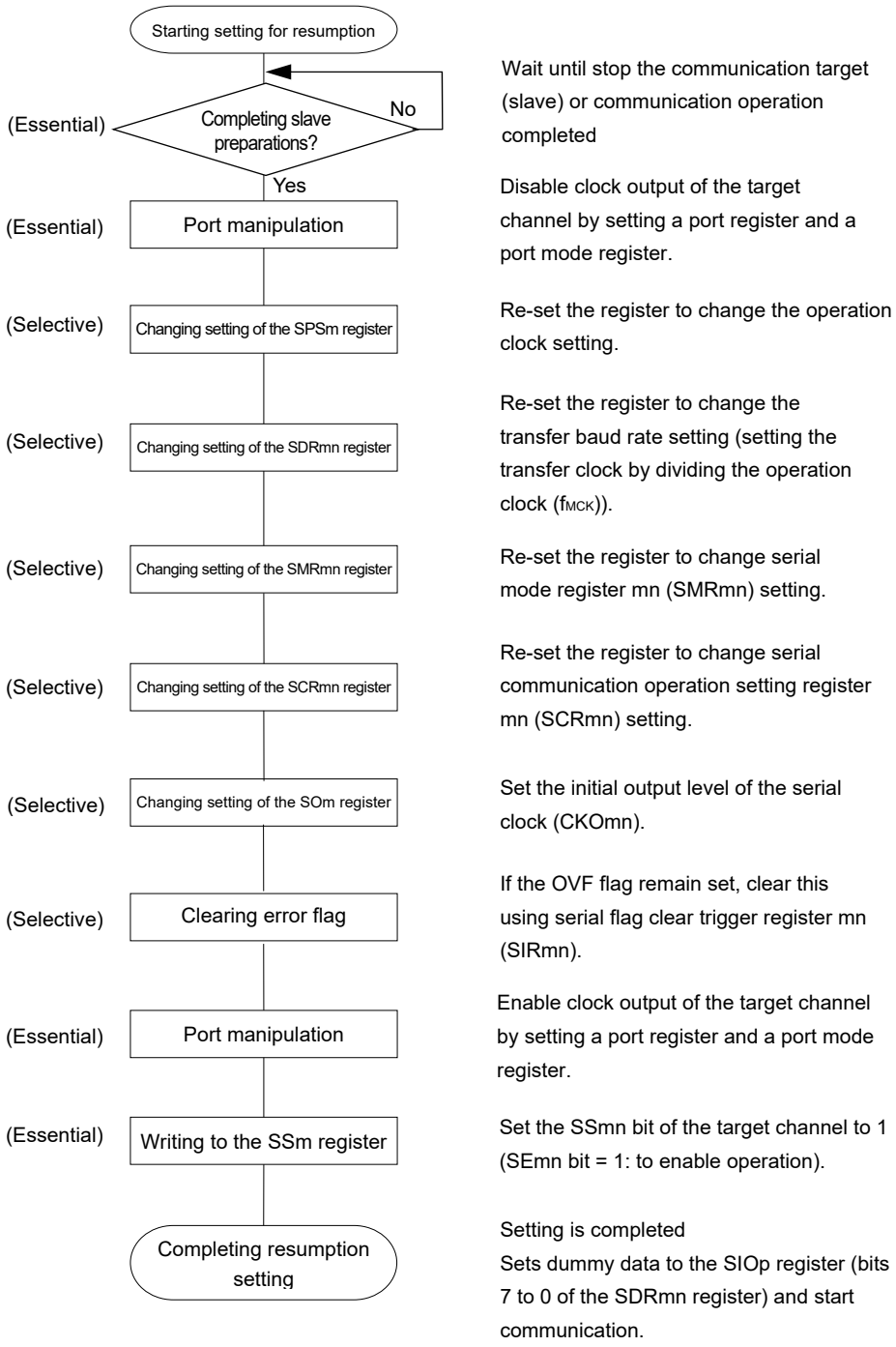


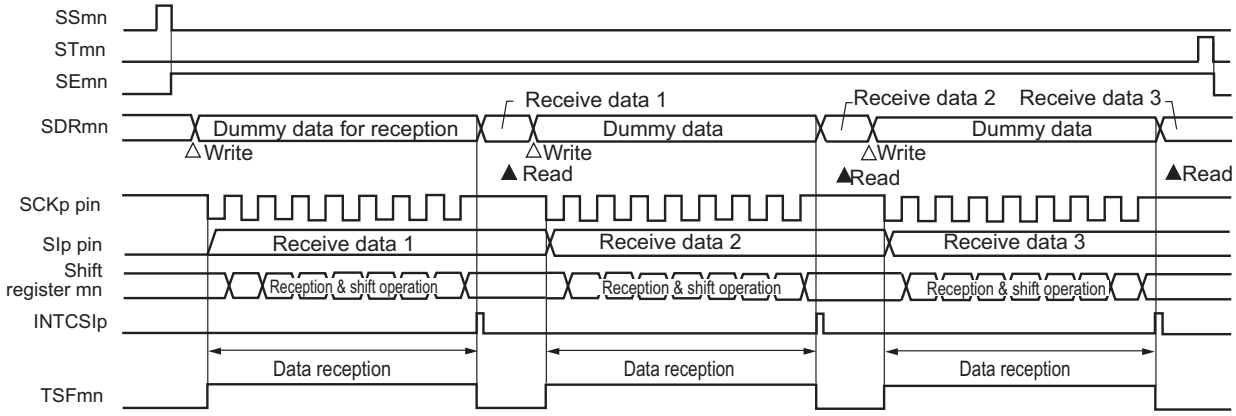
Figure 12-32. Procedure for Resuming Master Reception



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.

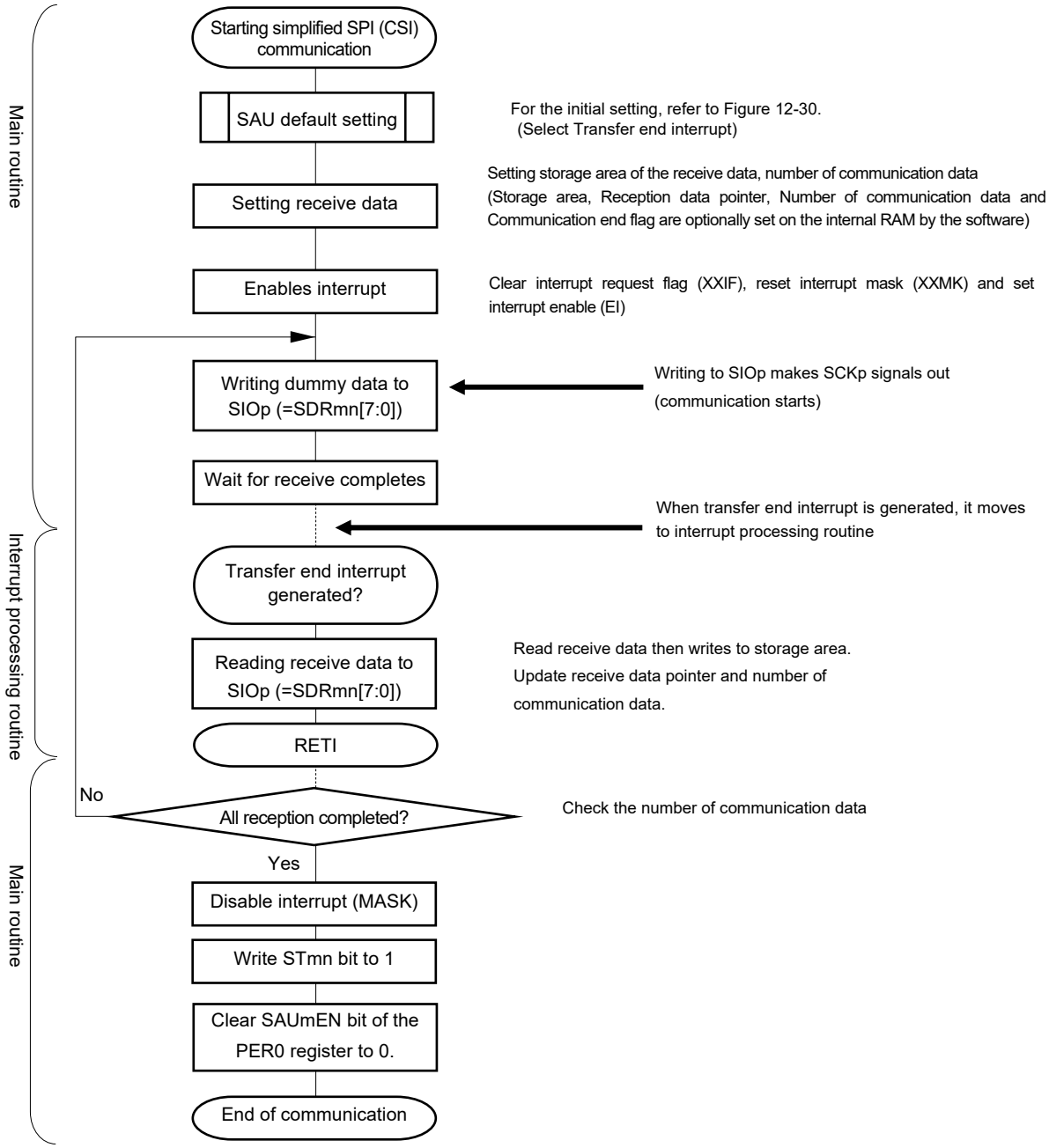
(3) Processing flow (in single-reception mode)

Figure 12-33. Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



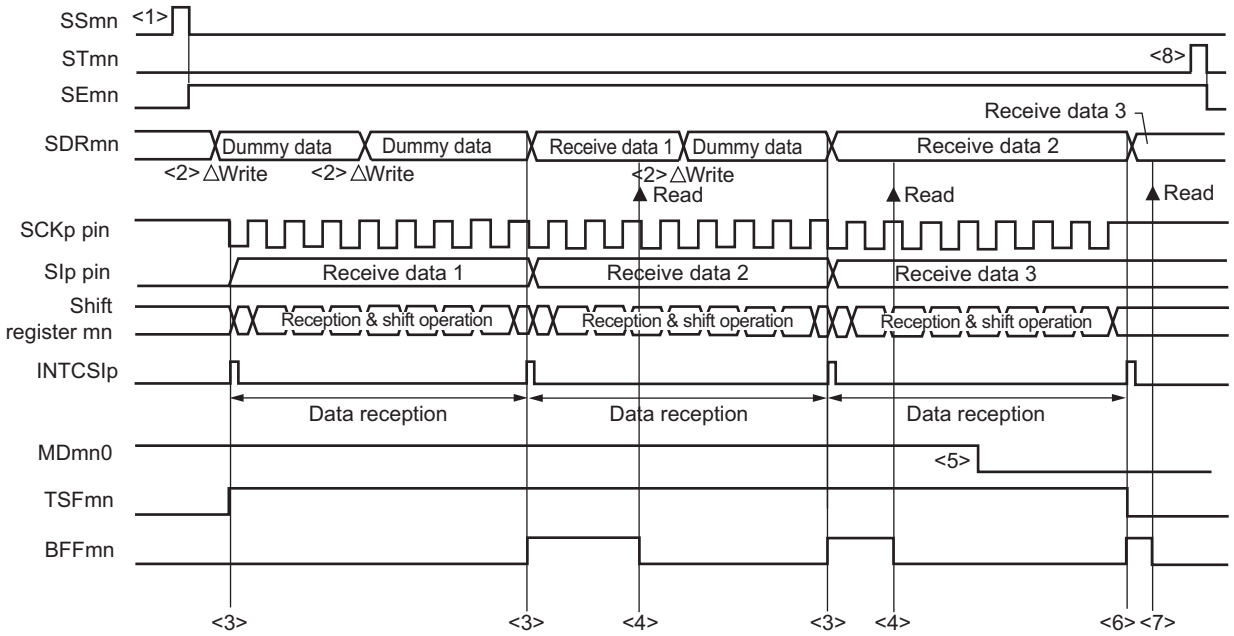
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-34. Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing flow (in continuous reception mode)

Figure 12-35. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

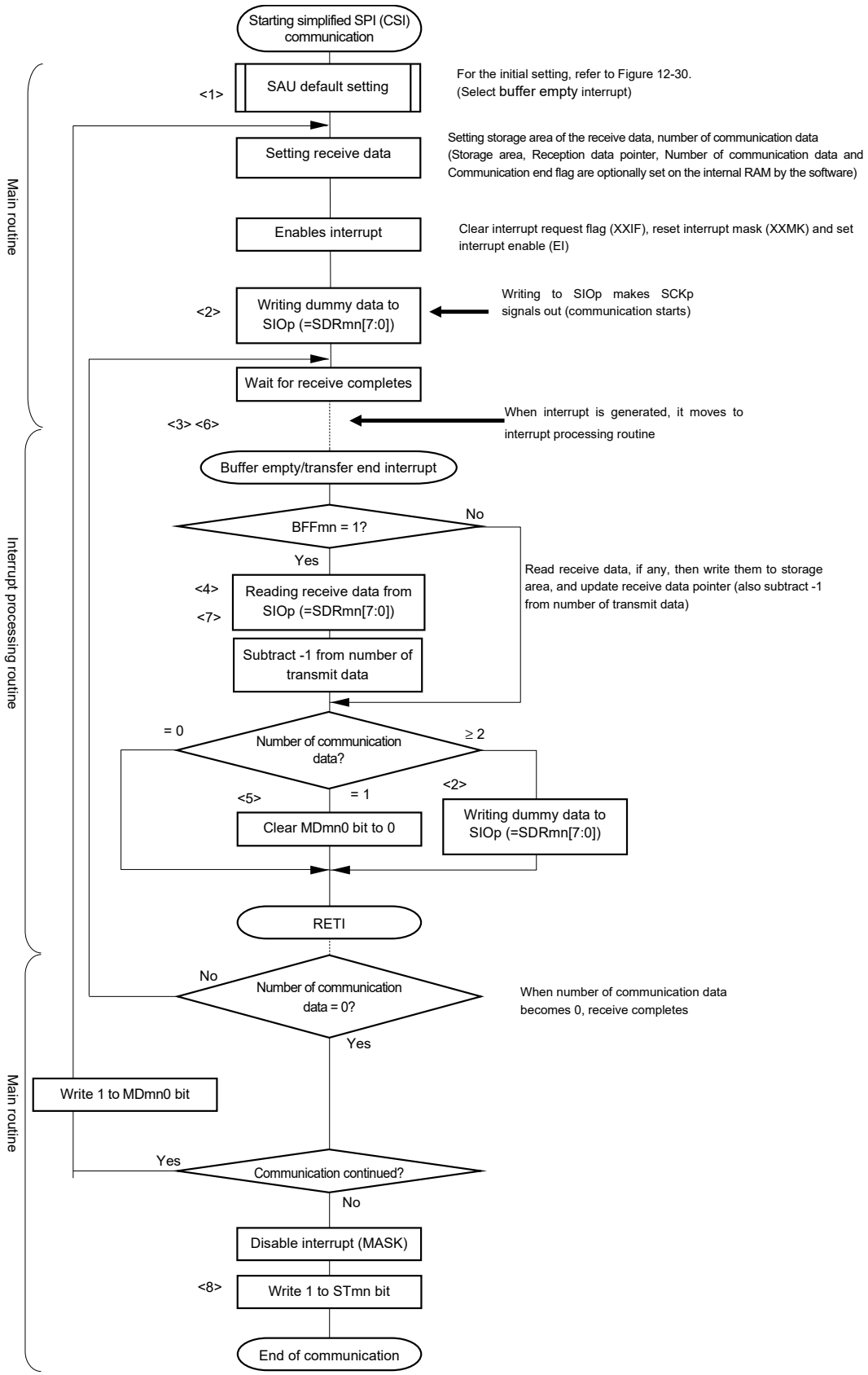


Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-36 Flowchart of Master Reception (in Continuous Reception Mode)**.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-36. Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-35 Timing Chart of Master Reception (in Continuous Reception Mode)**.

12.5.3 Master transmission/reception

Master transmission/reception is that the RL78 microcontroller outputs a transfer clock and transmits/receives data to/from other device.

Simplified SPI	CSI00	CSI01
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 (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate Note	Max. $f_{CLK}/2$ [Hz] (CSI00), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 30 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

(1) Register setting

Figure 12-37. Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 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 mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data setting/receive data register							

SIOp

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

Remarks 1. m: Unit number (m = 0), n: Channel number (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

Figure 12-37. Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (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 12-38. Initial Setting Procedure for Master Transmission/Reception

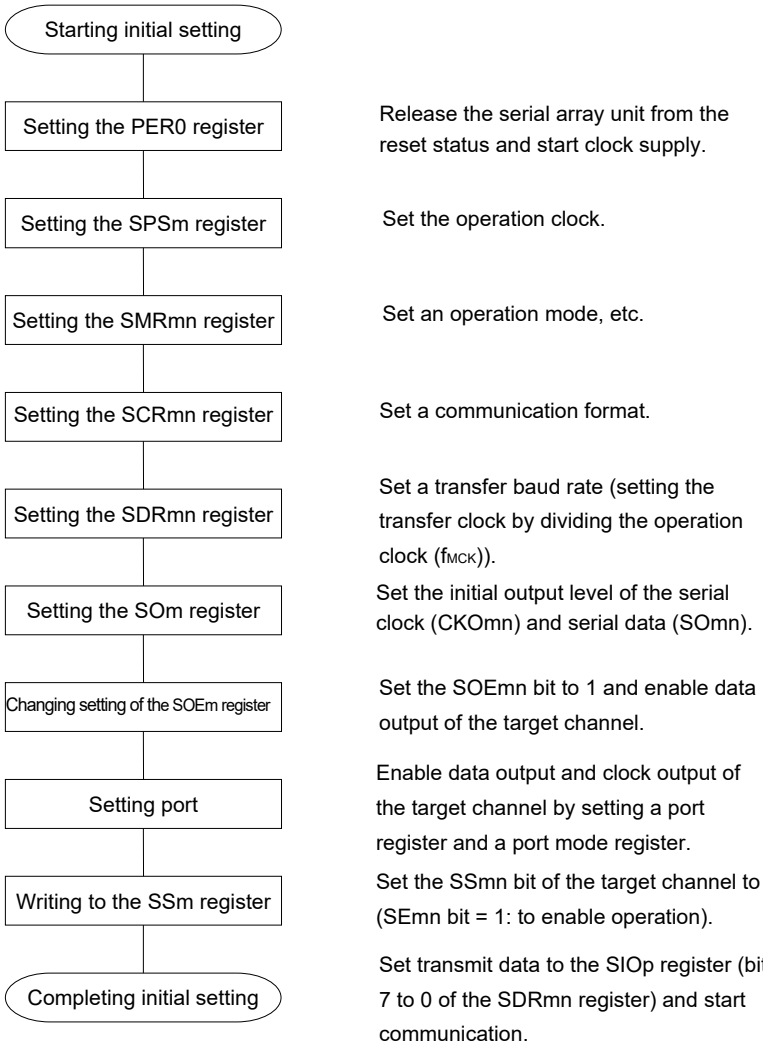


Figure 12-39. Procedure for Stopping Master Transmission/Reception

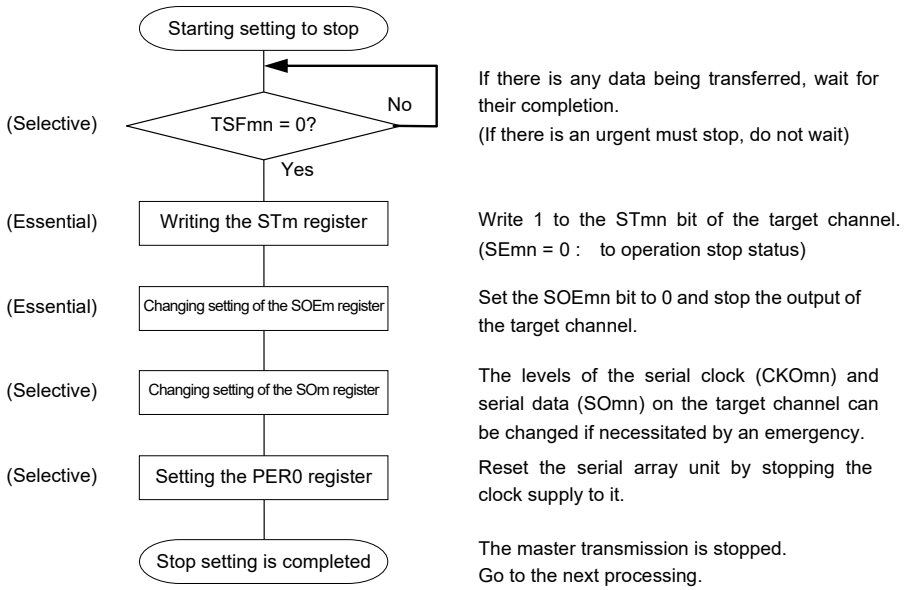
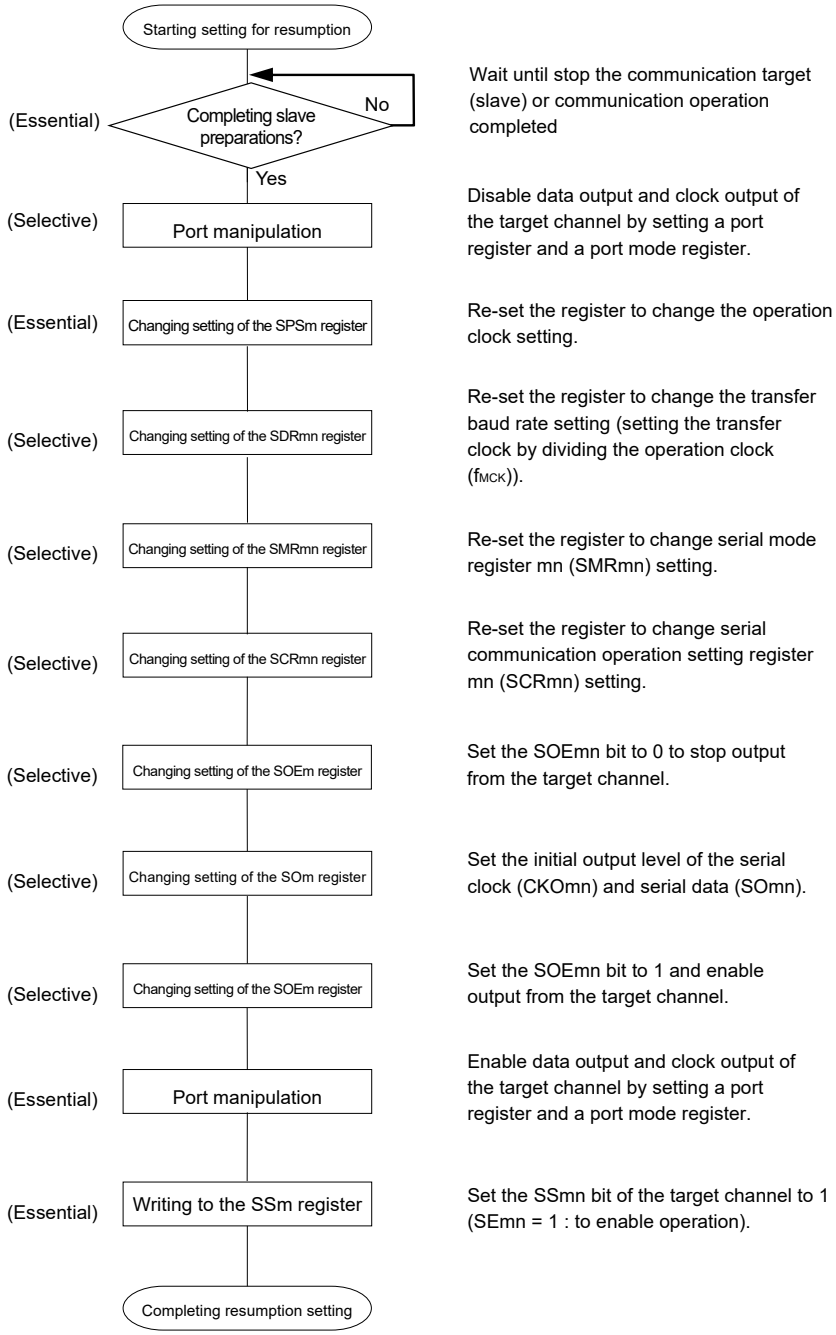
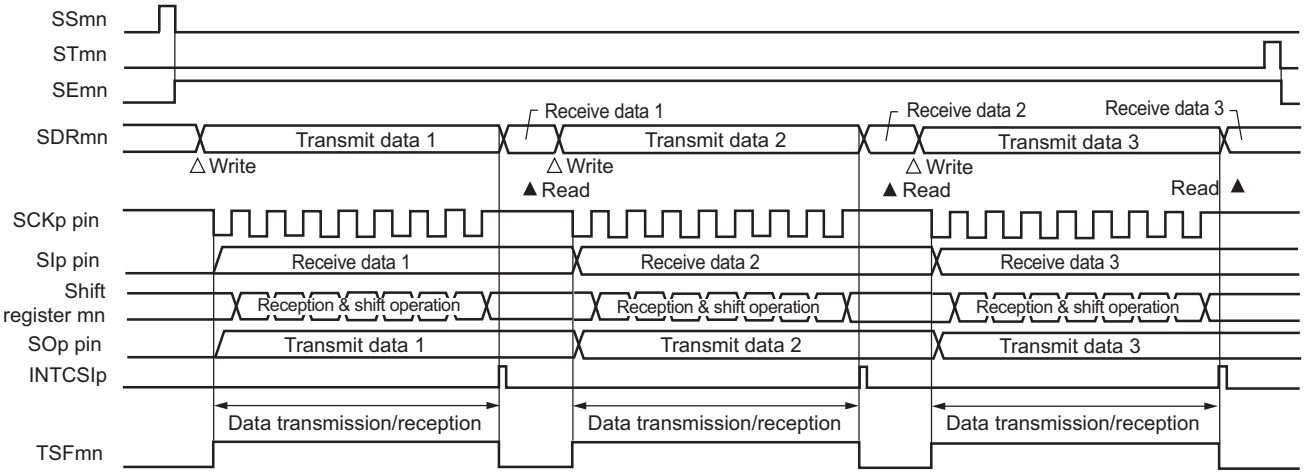


Figure 12-40. Procedure for Resuming Master Transmission/Reception



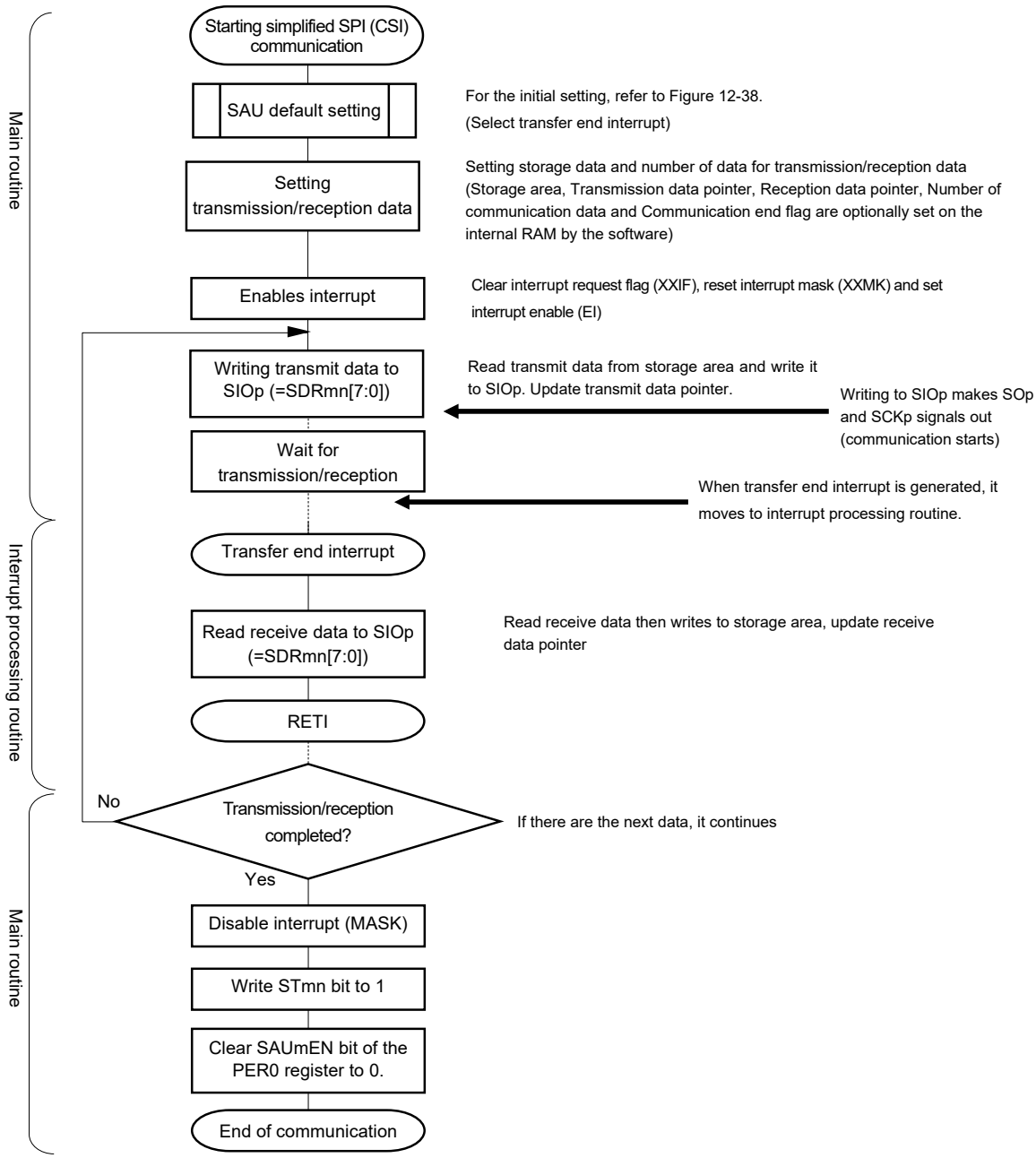
(3) Processing flow (in single-transmission/reception mode)

Figure 12-41. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



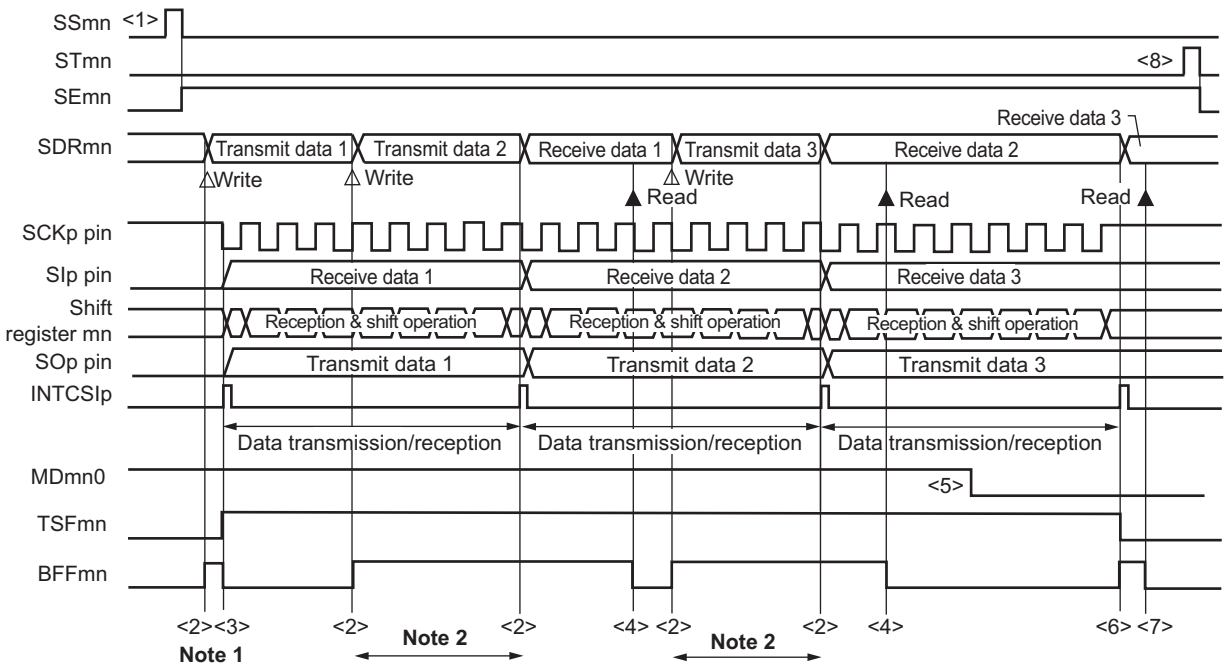
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-42. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)



(4) Processing flow (in continuous transmission/reception mode)

Figure 12-43. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

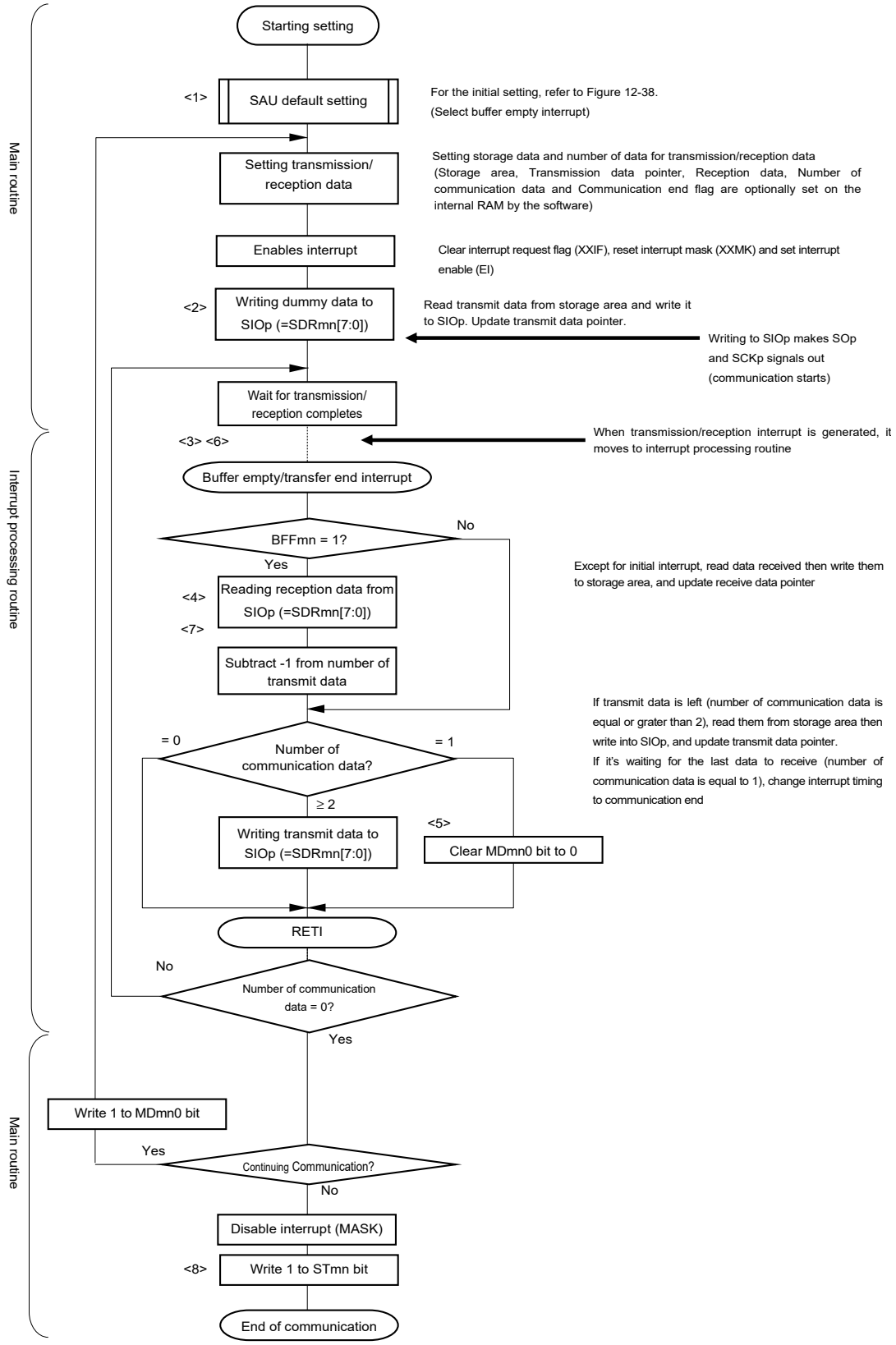
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit of serial mode register mn (SMRmn) 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 12-44 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-44. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 12-43 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

12.5.4 Slave transmission

Slave transmission is that the RL78 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
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 (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. f_{MCK}/6 [Hz][Notes 1, 2].	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Notes 1. 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].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 30 or 31 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

(1) Register setting

Figure 12-45. Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0/1	0/1	0	0	0	0	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 mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting							

SIOp

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	×	×	0	0	0	0	0	0	0/1	0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

2. □: Setting is fixed in the simplified SPI (CSI) slave 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

Figure 12-45. Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

2. □ : Setting is fixed in the simplified SPI (CSI) slave 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 12-46. Initial Setting Procedure for Slave Transmission

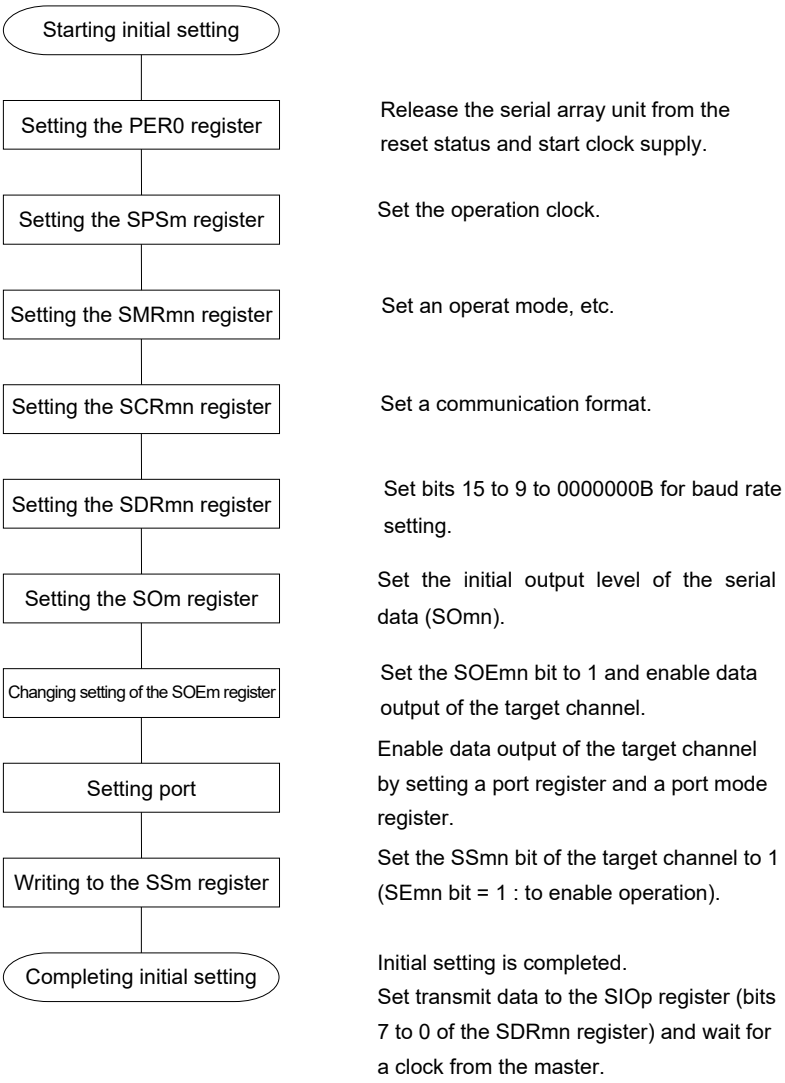


Figure 12-47. Procedure for Stopping Slave Transmission

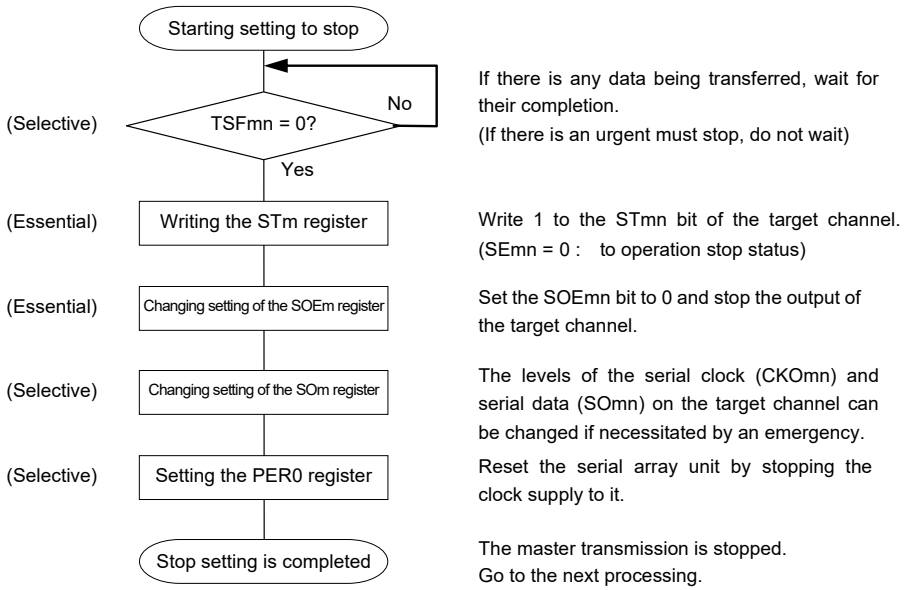
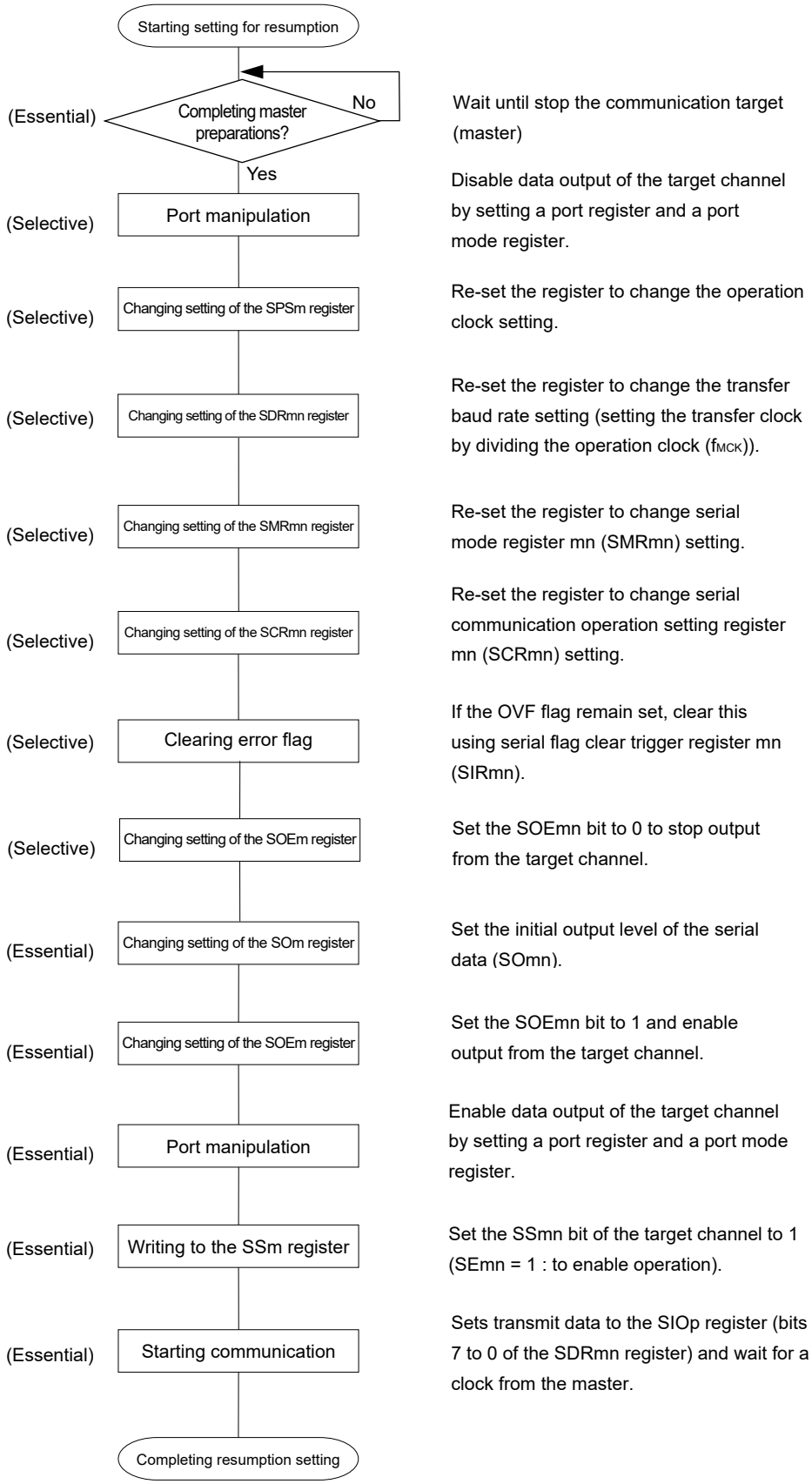


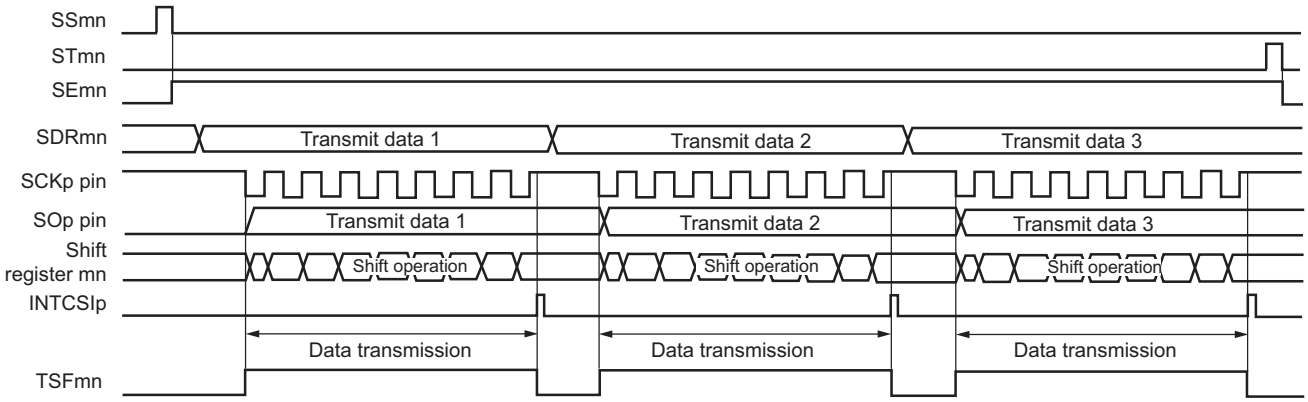
Figure 12-48. Procedure for Resuming Slave Transmission



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.

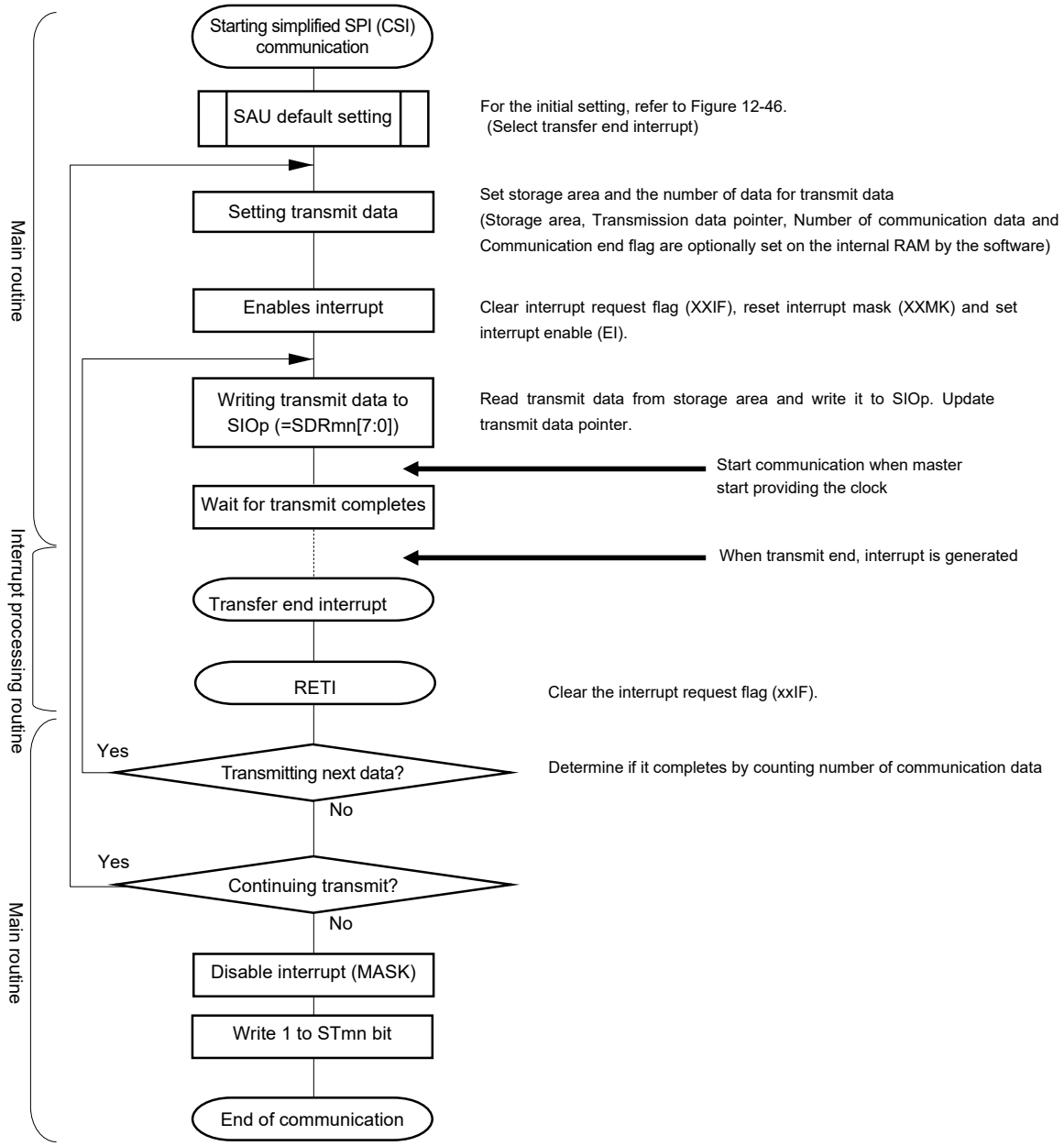
(3) Processing flow (in single-transmission mode)

Figure 12-49. Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



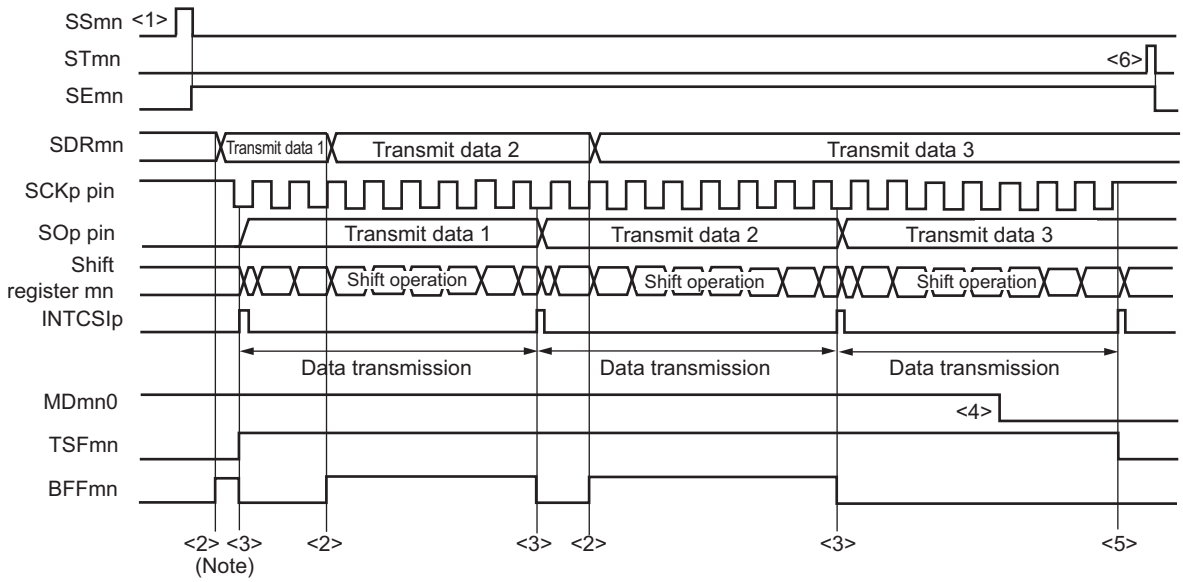
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-50. Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 12-51. Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

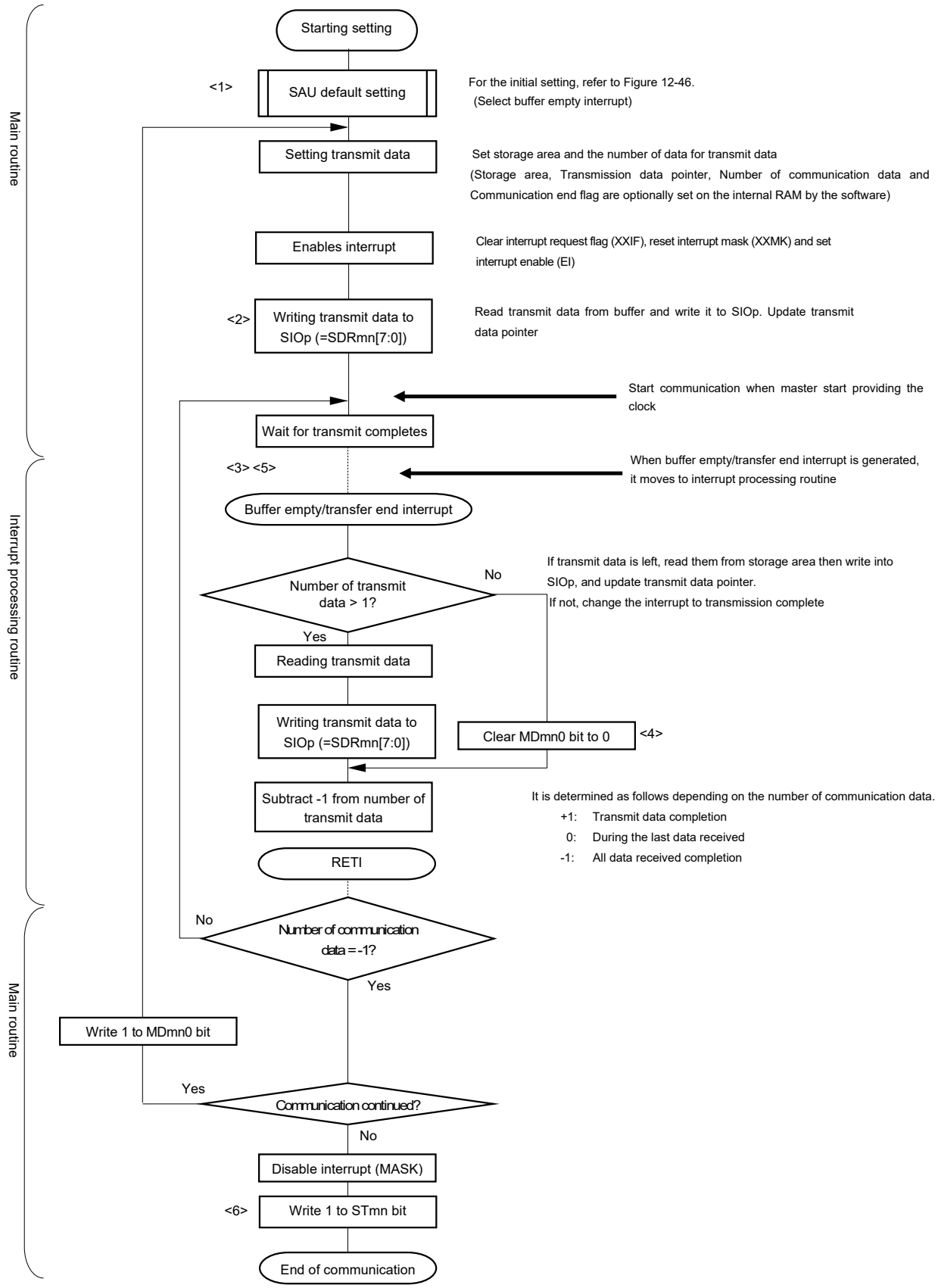


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-52. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-51 Timing Chart of Slave Transmission (in Continuous Transmission Mode)**.

12.5.5 Slave reception

Slave reception is that the RL78 microcontroller receives data from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00	SCK01, SI01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. f_{MCK}/6 [Hz] Notes 1, 2	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Notes 1. 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].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 30 or 31 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

(1) Register setting

Figure 12-53. Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	0	1	0/1	0/1	0	0	0	0	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 mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Receive data							

SIOp

(d) Serial output register m (SOm) …The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	×	×	0	0	0	0	0	0	×	×

Remarks 1. m: Unit number (m = 0), n: Channel number (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

Figure 12-53. Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01) (2/2)

(e) Serial output enable register m (SOEm) …The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 ×

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (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 12-54. Initial Setting Procedure for Slave Reception

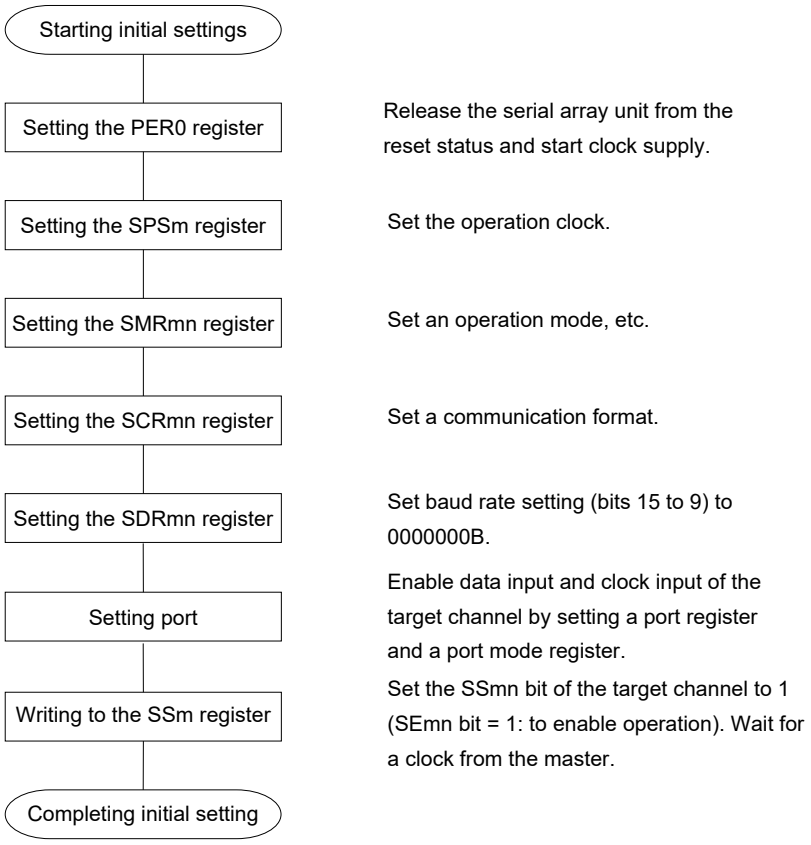


Figure 12-55. Procedure for Stopping Slave Reception

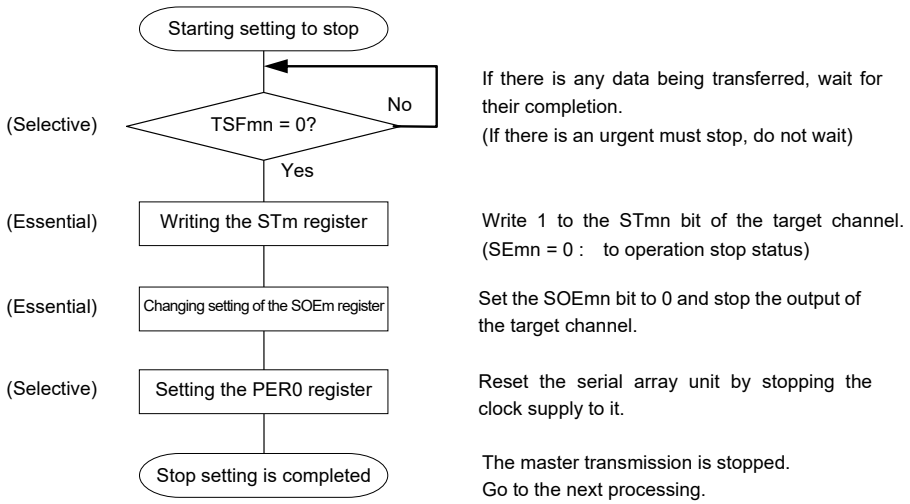
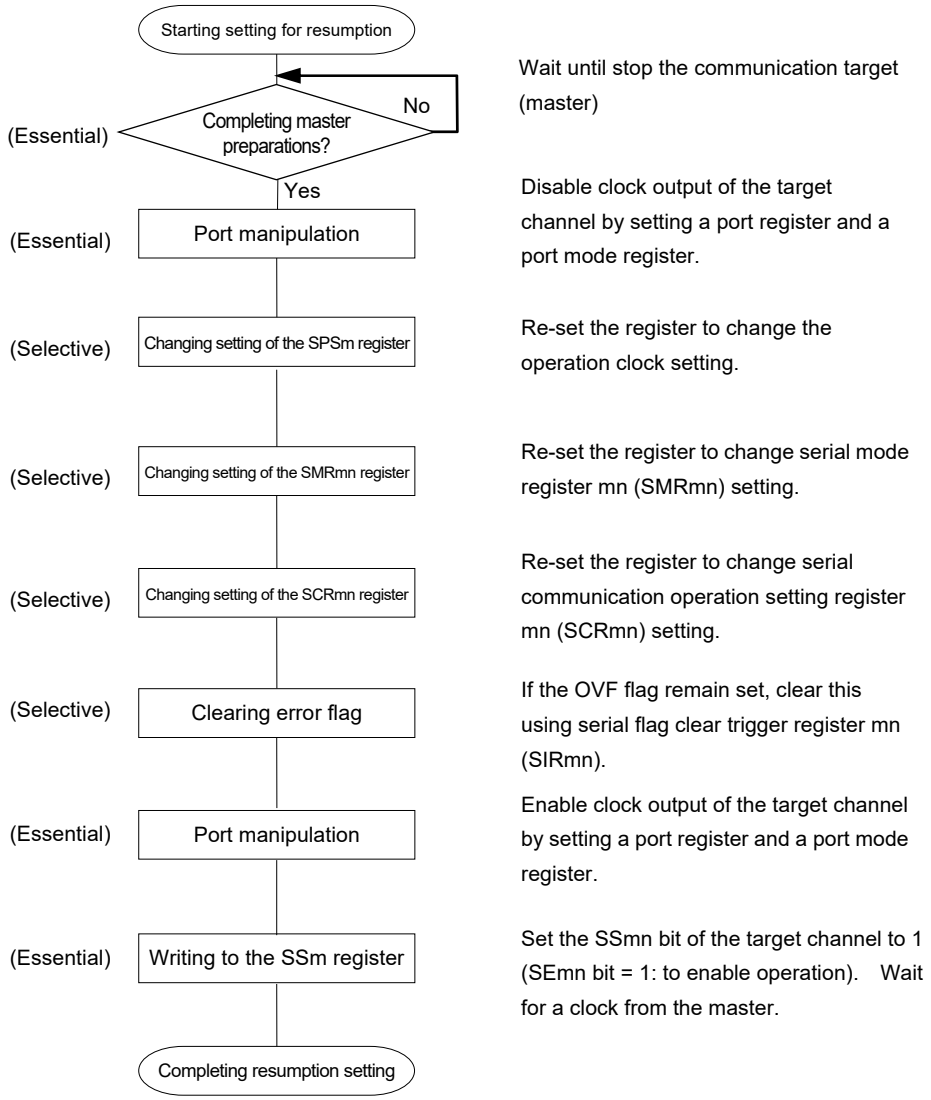


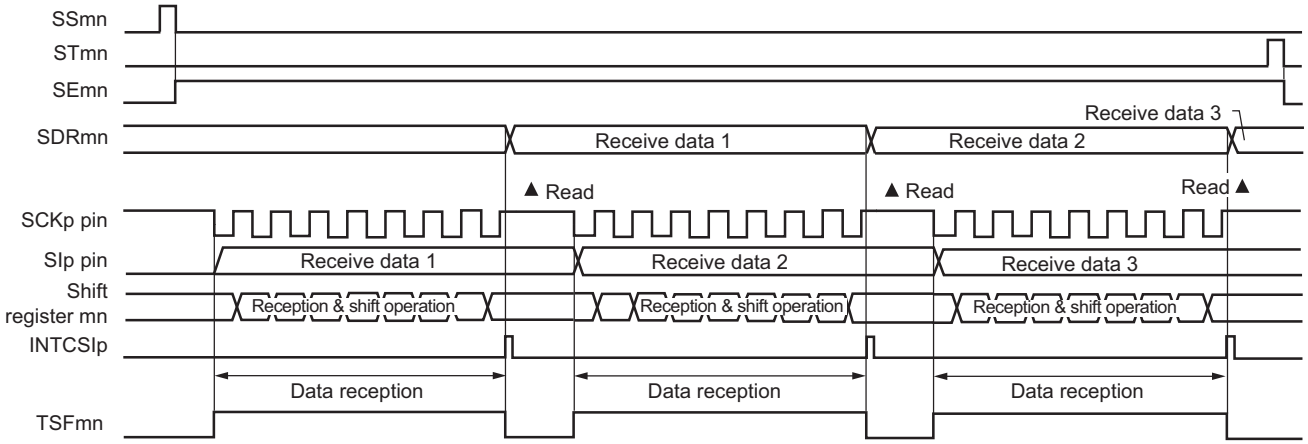
Figure 12-56. Procedure for Resuming Slave Reception



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.

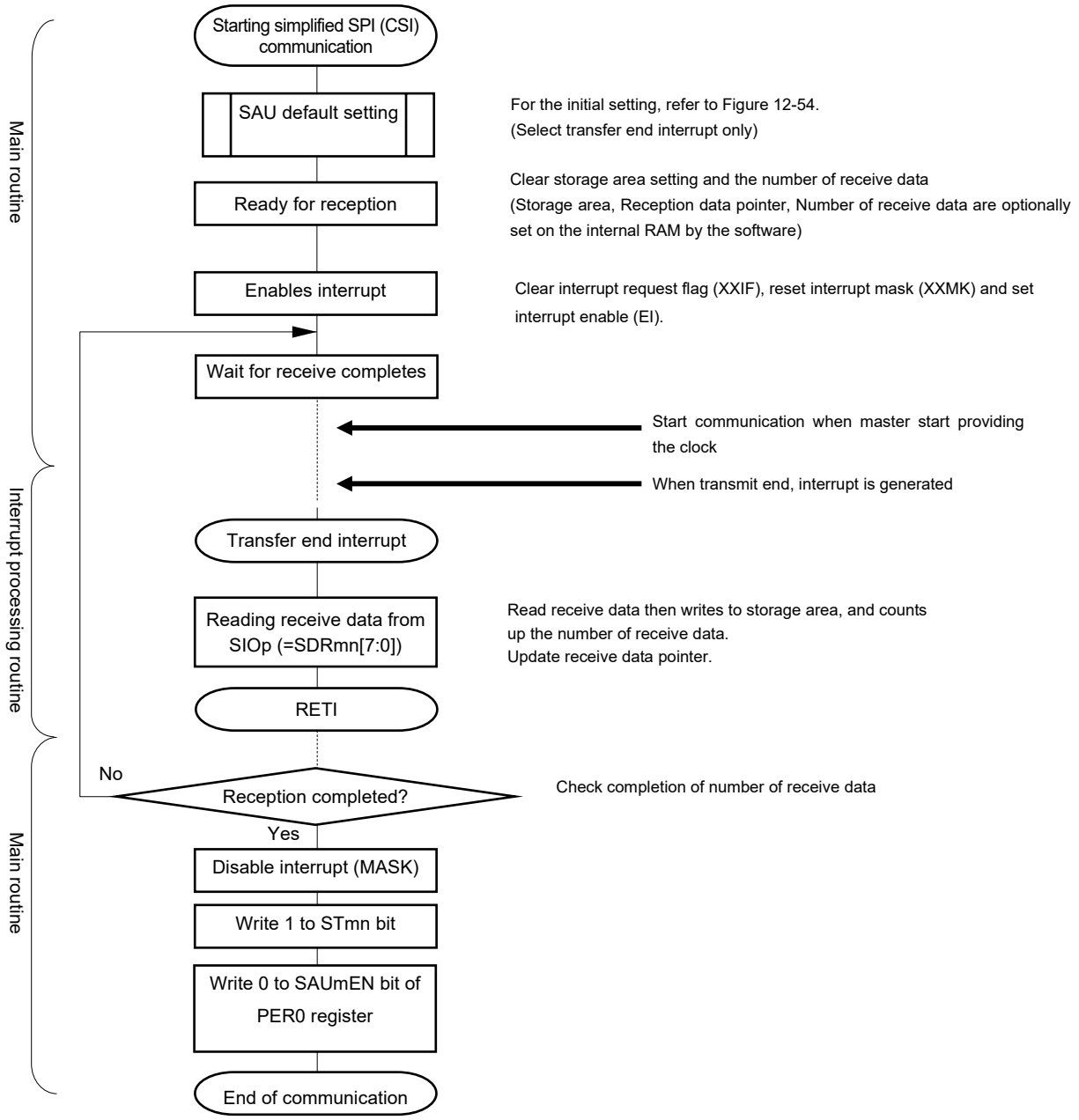
(3) Processing flow (in single-reception mode)

Figure 12-57. Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-58. Flowchart of Slave Reception (in Single-Reception Mode)



12.5.6 Slave transmission/reception

Slave transmission/reception is that the RL78 microcontroller transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
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 (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. f_{MCK}/6 [Hz][Notes 1, 2].	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Notes 1. 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].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 30 or 31 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel

2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

(1) Register setting

Figure 12-59. Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 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 mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting/receive data register							

SIOp

(d) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

2. ☐: Setting is fixed in the simplified SPI (CSI) slave 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

Figure 12-59. Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01) (2/2)

(e) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

2. □: Setting is fixed in the simplified SPI (CSI) slave transmission/reception mode,

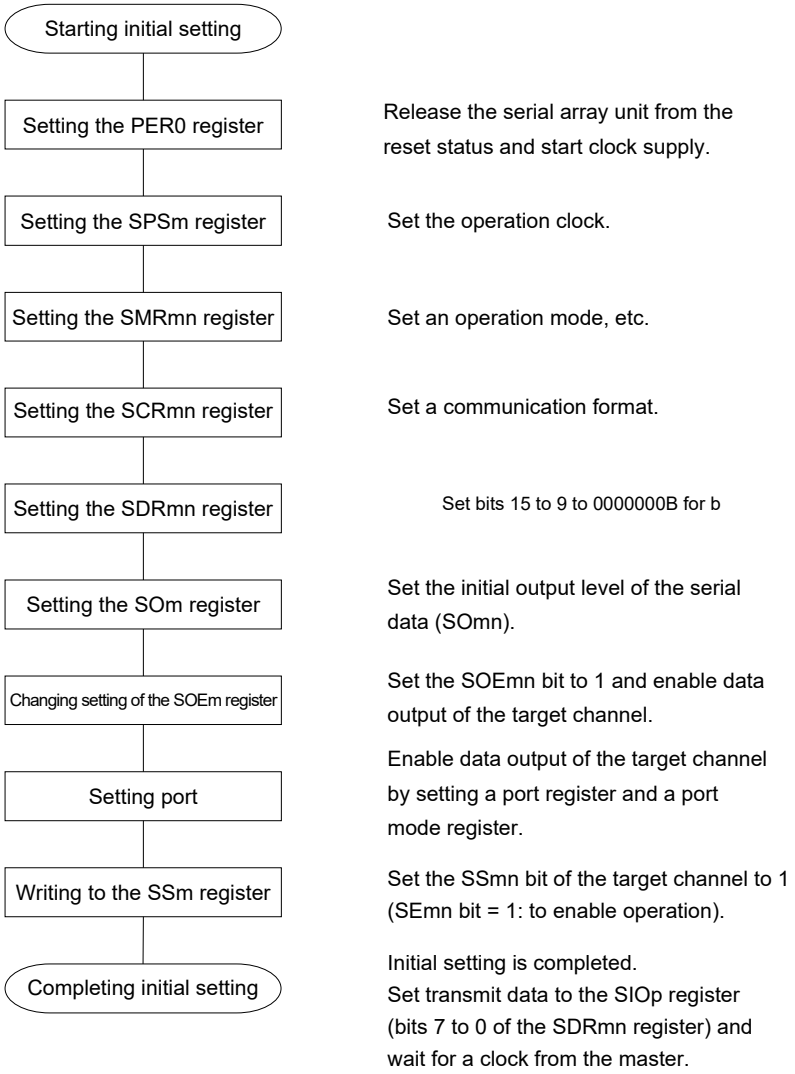
■: Setting disabled (set to the initial value)

×: 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 12-60. 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 12-61. Procedure for Stopping Slave Transmission/Reception

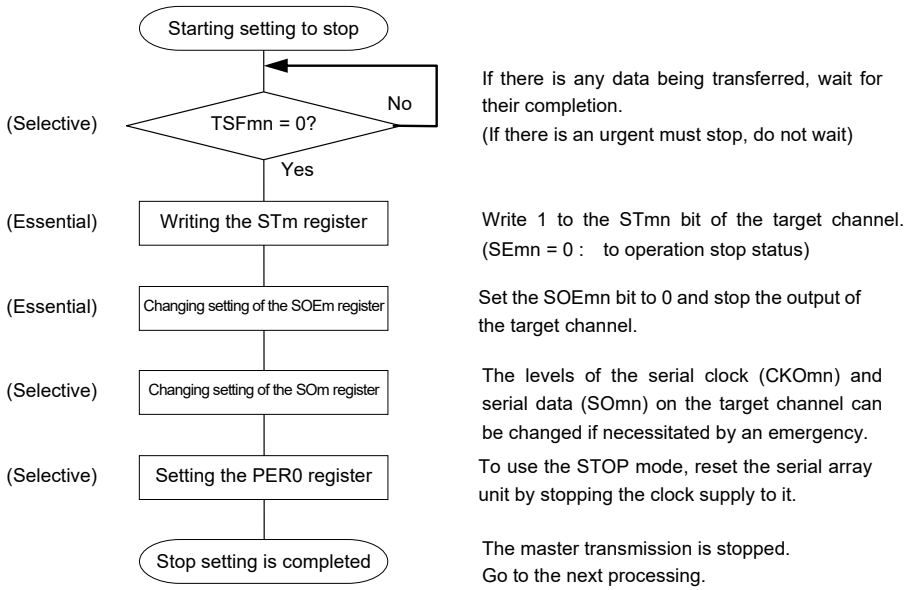
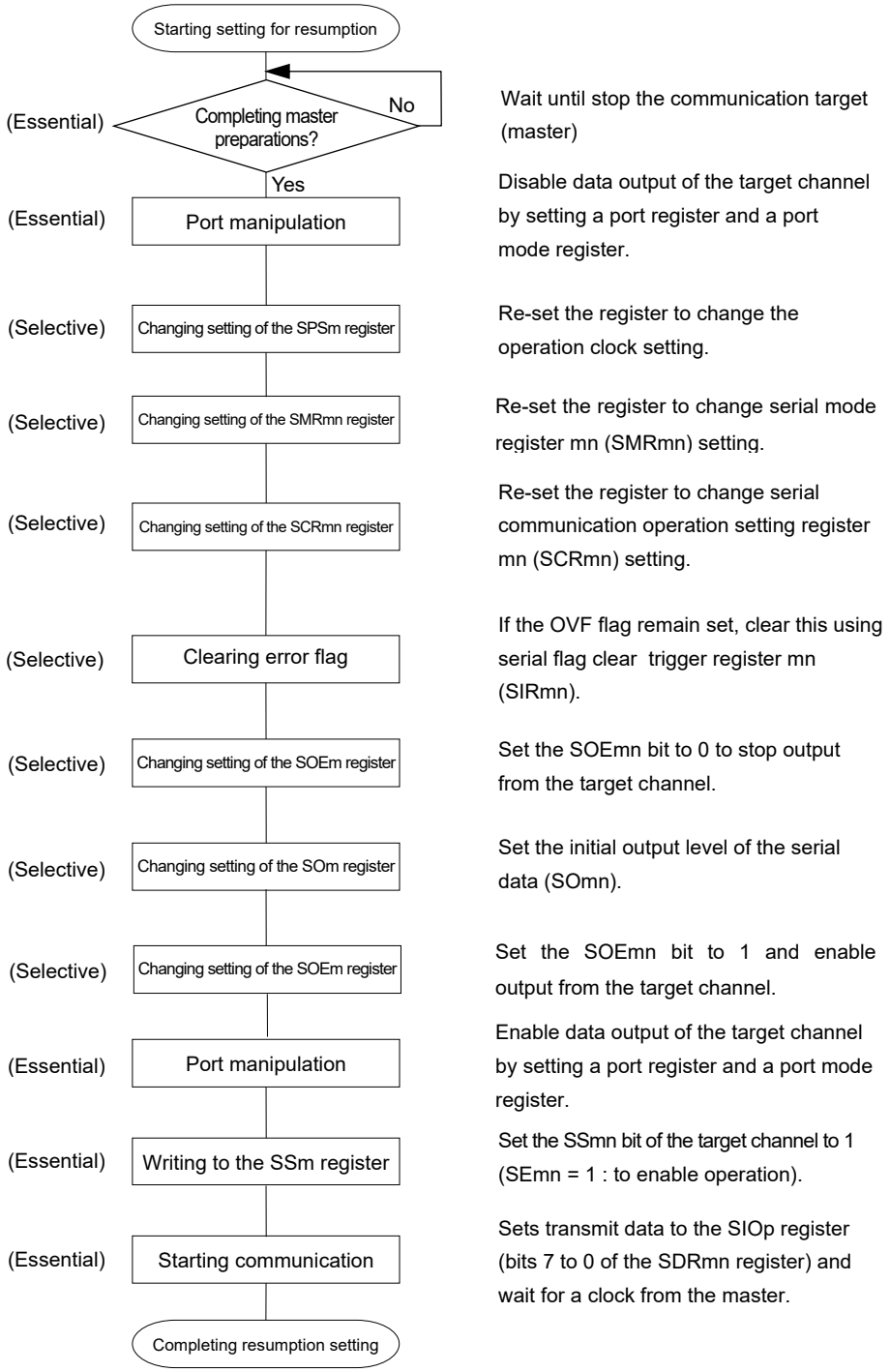


Figure 12-62. Procedure for Resuming Slave Transmission/Reception

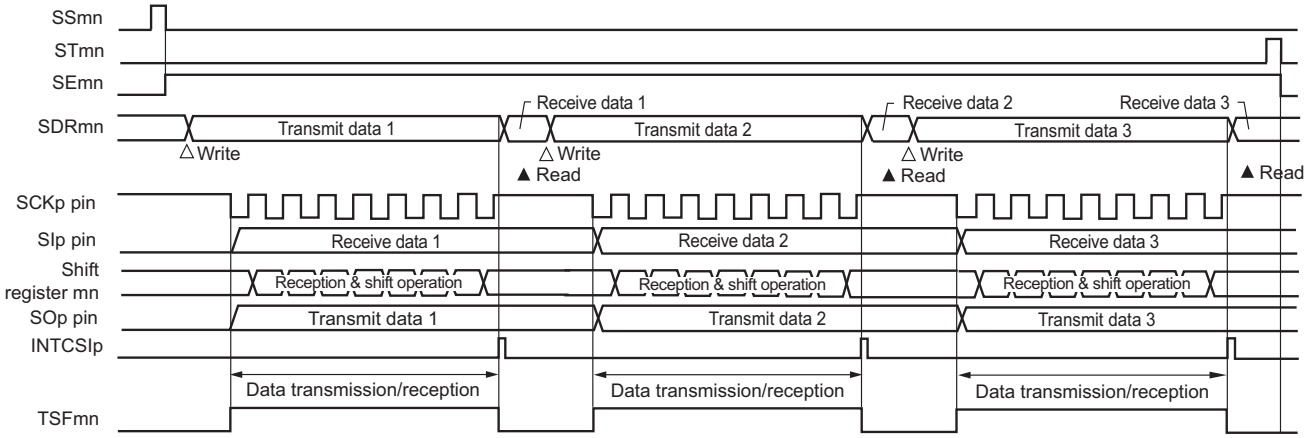


Cautions 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

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.

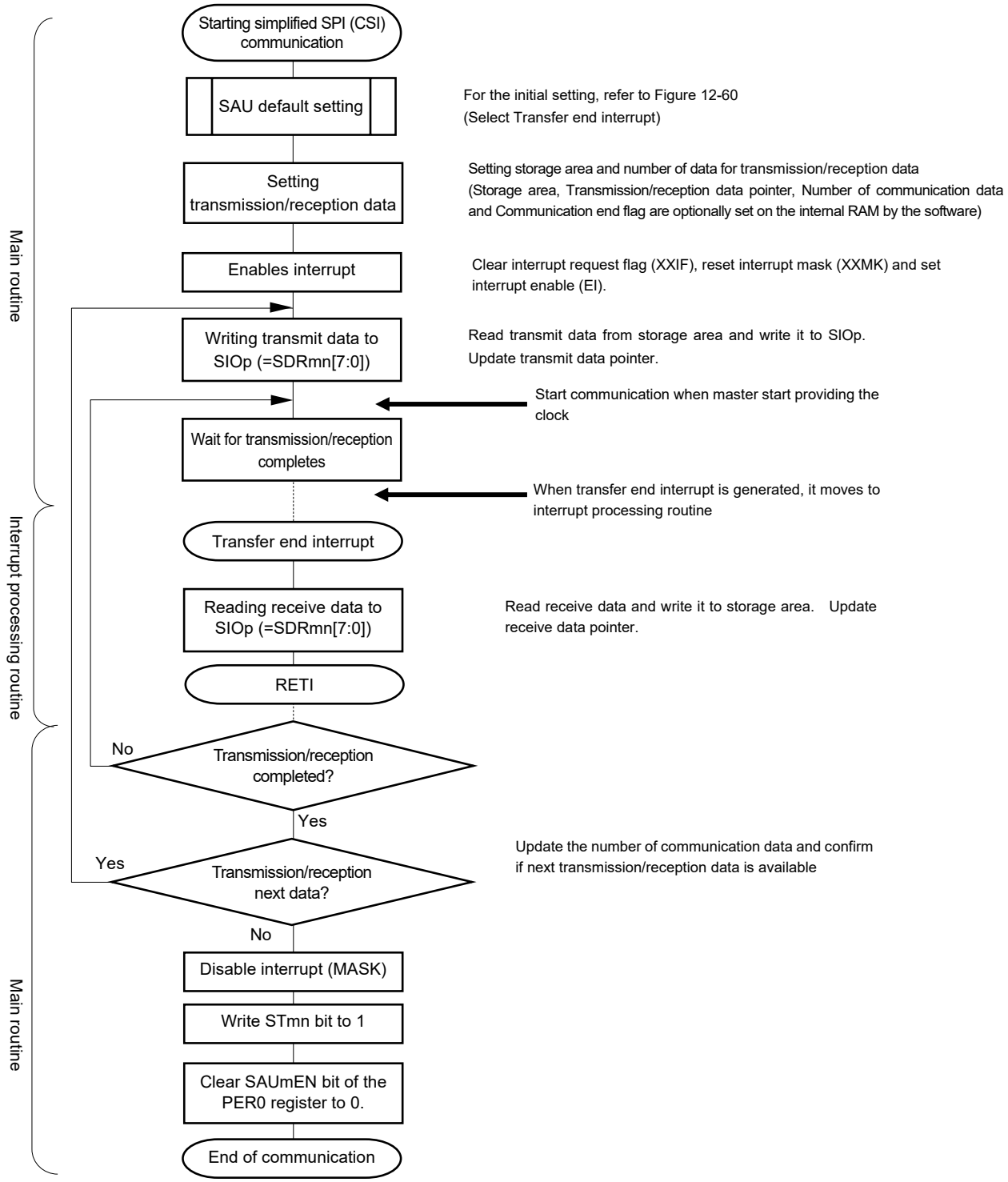
(3) Processing flow (in single-transmission/reception mode)

Figure 12-63. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

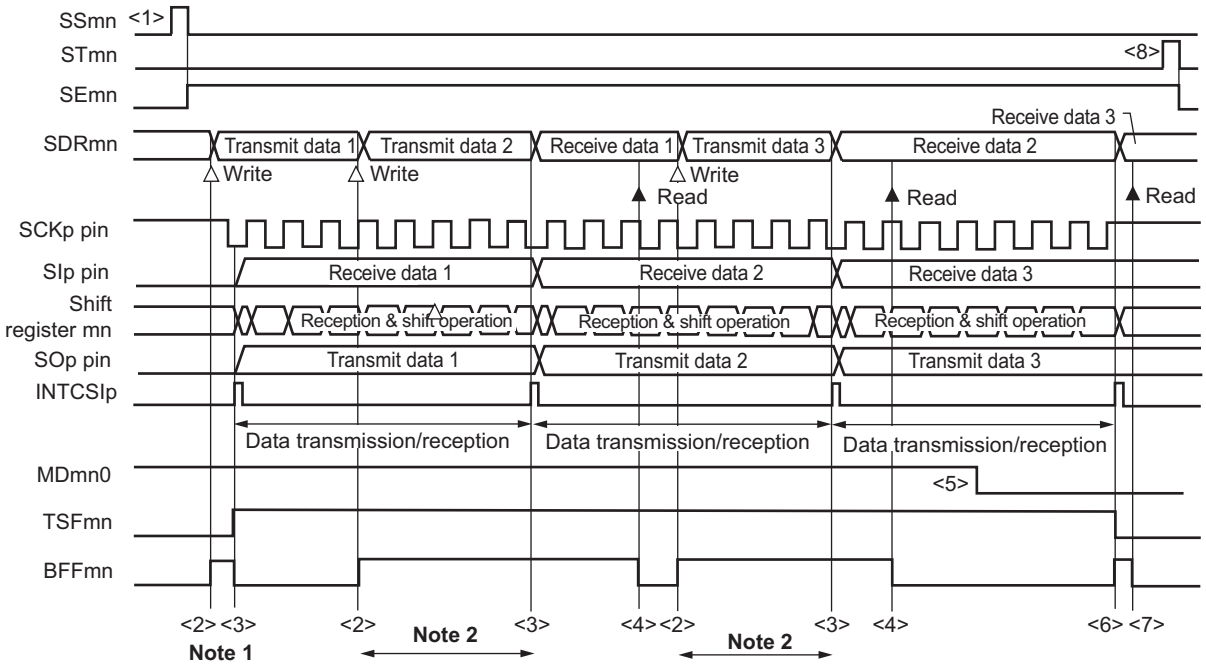
Figure 12-64. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)



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 12-65. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

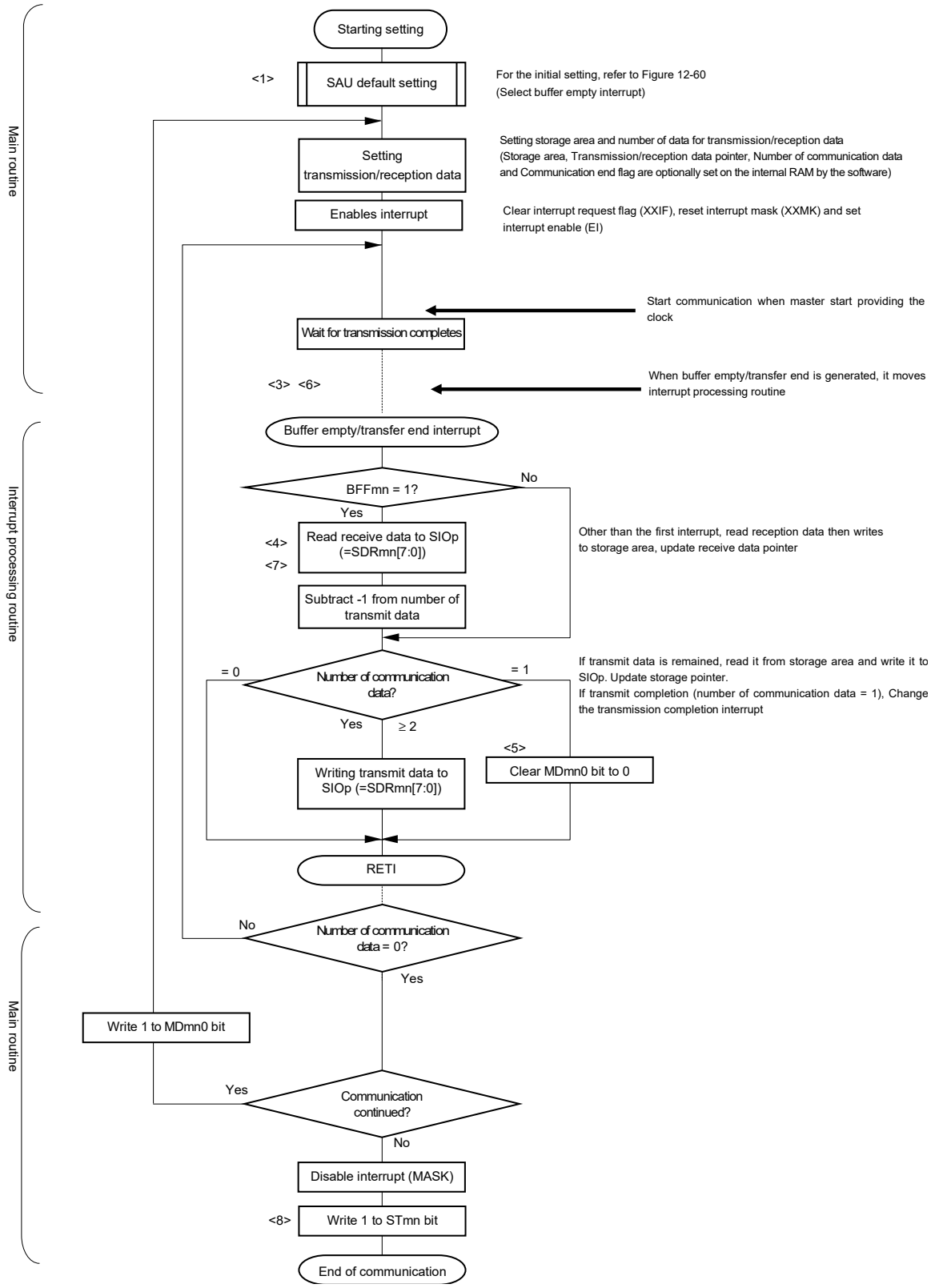
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit of serial mode register mn (SMRmn) 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 12-66 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 12-66. 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 12-65 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

12.5.7 SNOOZE mode function

When SCKp pin input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only CSI00 can be specified for asynchronous reception.

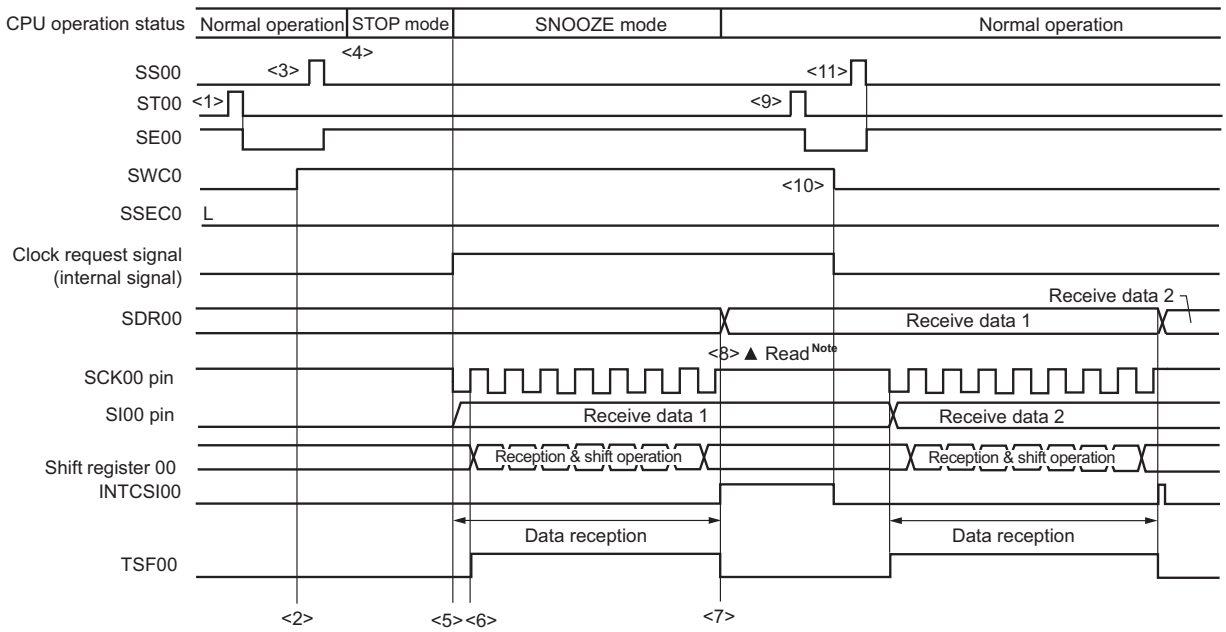
When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 before switching to the STOP mode.

Cautions 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

2. The maximum transfer rate when using CSI00 in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 12-67. Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

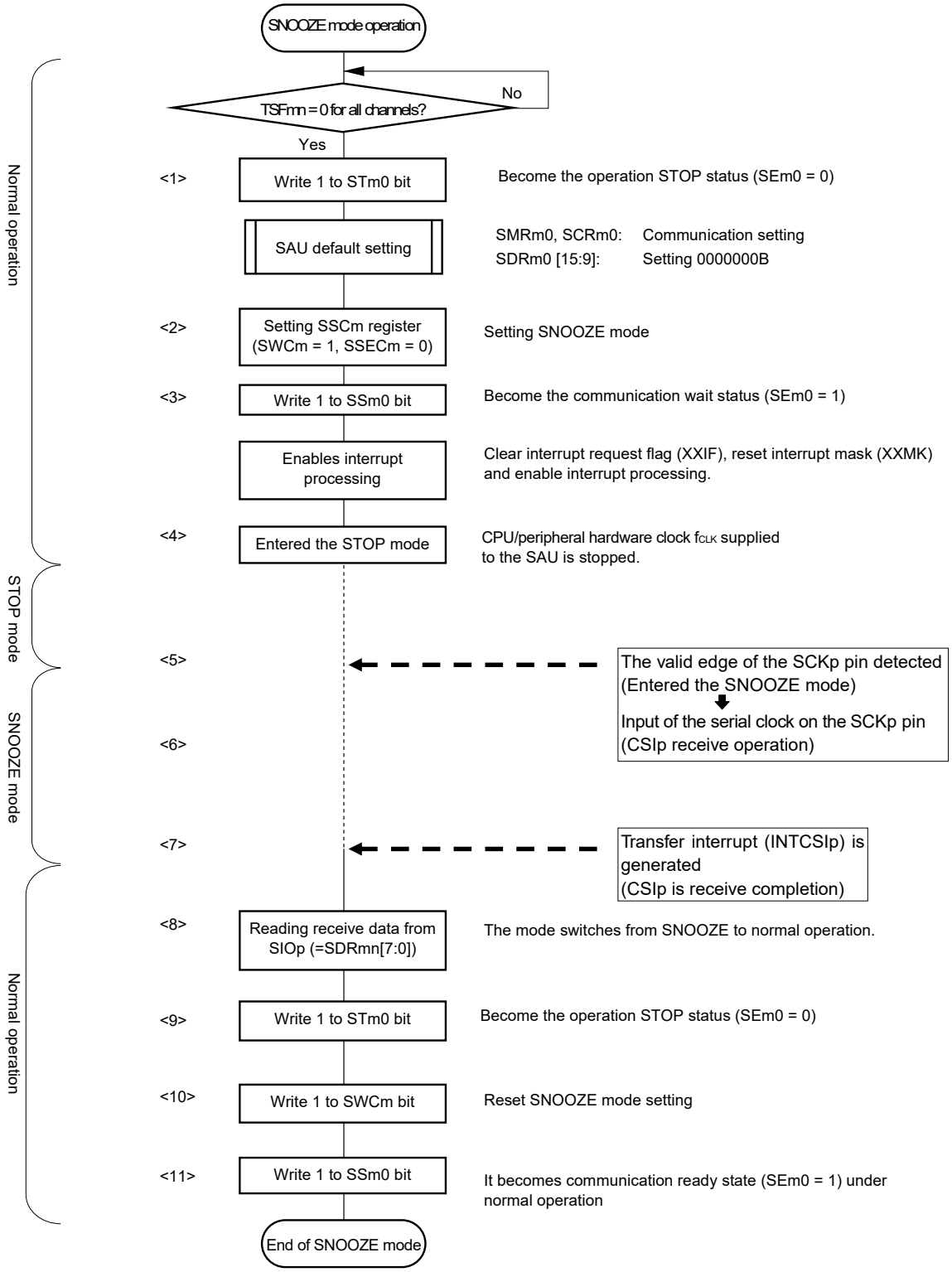
Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit and stop the operation).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 12-68 Flowchart of SNOOZE Mode Operation (once startup)**.

2. m = 0; p = 00

Figure 12-68. Flowchart of SNOOZE Mode Operation (once startup)

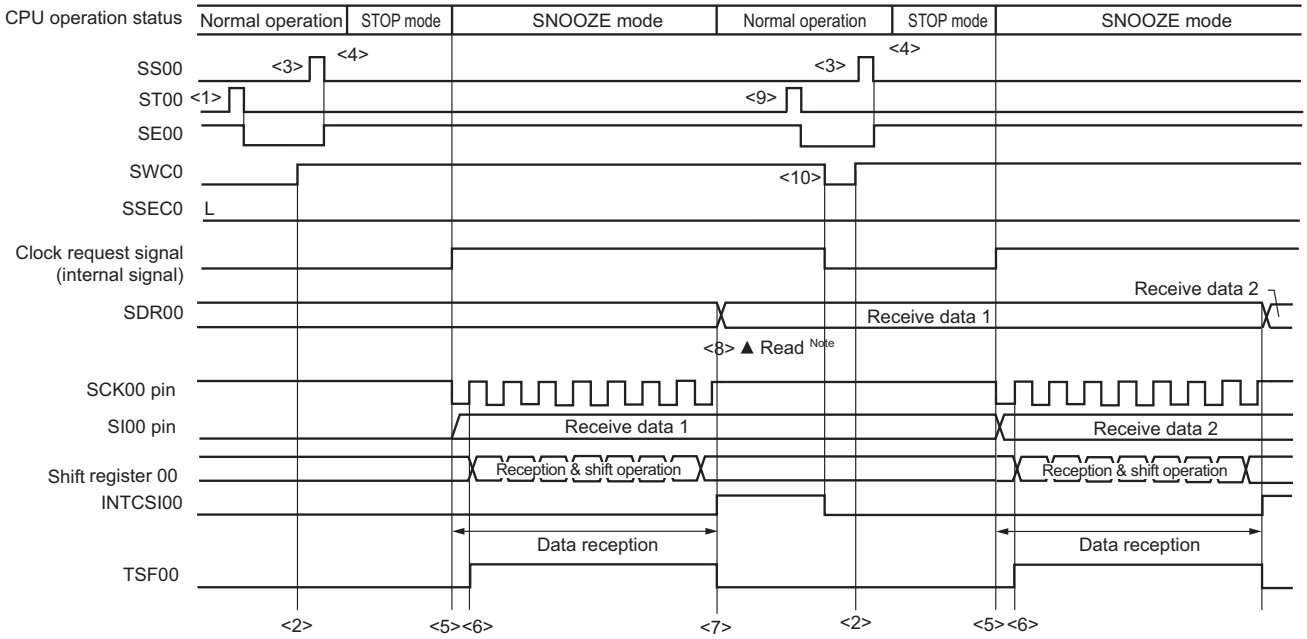


Remarks 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 12-67. Timing Chart of SNOOZE Mode Operation (once startup)**.

2. m = 0; p = 00

(2) SNOOZE mode operation (continuous startup)

Figure 12-69. Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)

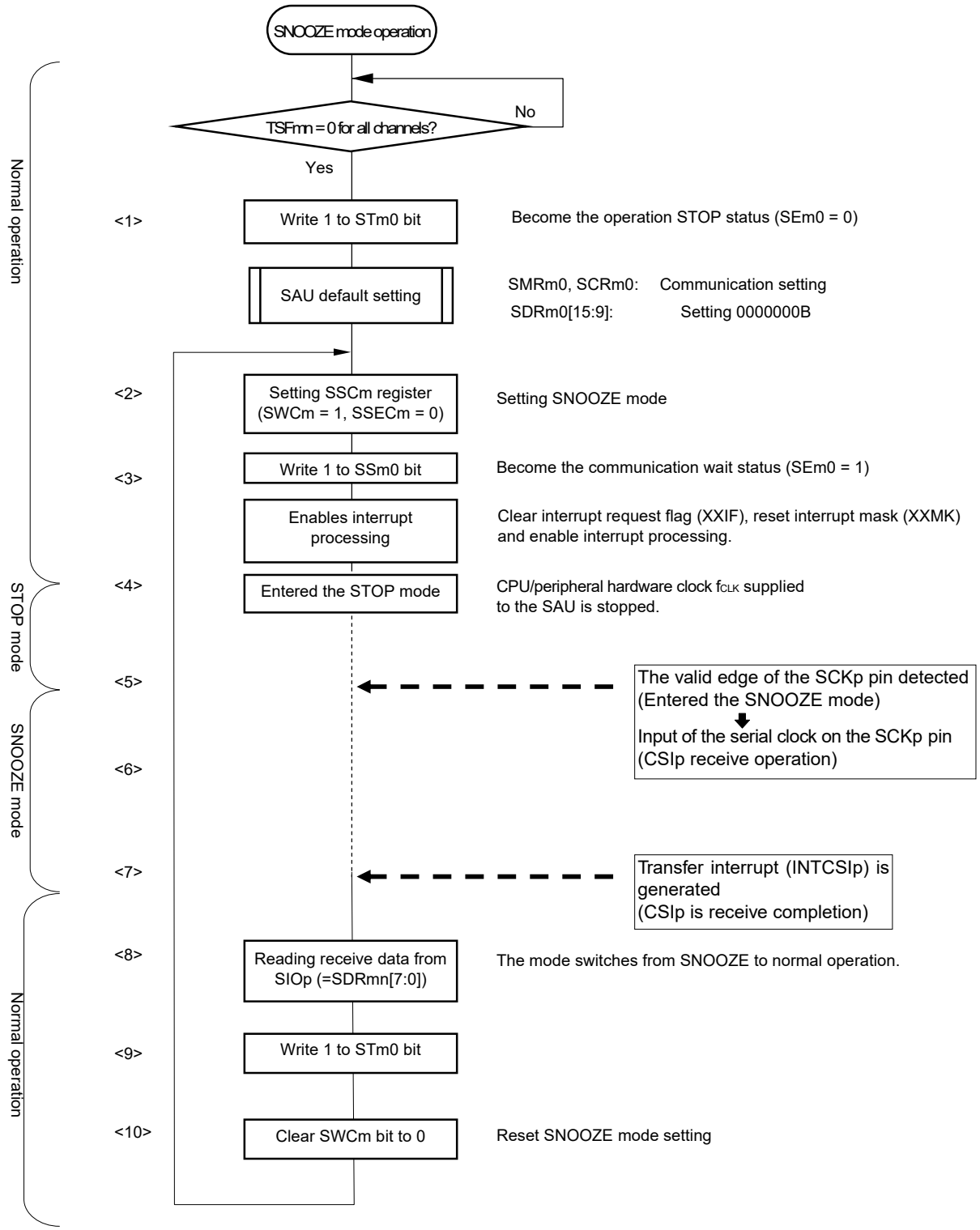


Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

Cautions 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit and stop the operation).
After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).
2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 12-68. Flowchart of SNOOZE Mode Operation (continuous startup)**.
2. m = 0; p = 00

Figure 12-70. Flowchart of SNOOZE Mode Operation (continuous startup)



Remarks 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 12-69. Timing Chart of SNOOZE Mode Operation (continuous startup)**.

2. m = 0; p = 00

12.5.8 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} ÷ (SDRmn[15:9] + 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 SDRmn[15:9] is the value of bits 15 to 9 of serial data register mn (SDRmn) (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 12-2. Selection of Operation Clock For Simplified SPI

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	732 Hz
Other than above									Setting prohibited	

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 m (STm) = 0003H) the operation of the serial array unit (SAU).

Remarks 1. X: Don't care

2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.5.9 Procedure for processing errors that occurred during Simplified SPI (CSI00, CSI01) communication

The procedure for processing errors that occurred during simplified SPI (CSI00, CSI01) communication is described in Figure 12-71.

Figure 12-71. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn 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 mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0), n: Channel number (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 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).

[Data transmission/reception]
- Data length of 7, 8, or 9 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

UART0 uses channels 0 and 1.

Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	CSI00	UART0	IIC00
1	CSI01		IIC01

Caution When using serial array unit as UART, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UART.

UART performs the following two 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 microcontroller to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even 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, 8, or 9 bits
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 3 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] Note
Data phase	Forward output (default: high level) Reverse 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 AC characteristics in the electrical specifications (see **CHAPTER 30 or 31 ELECTRICAL SPECIFICATIONS**).

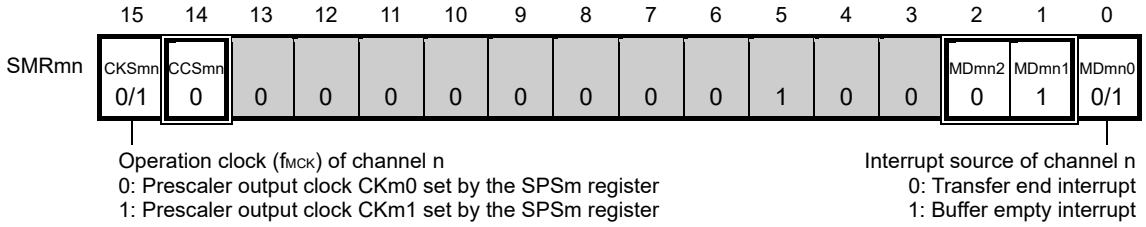
Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 0)

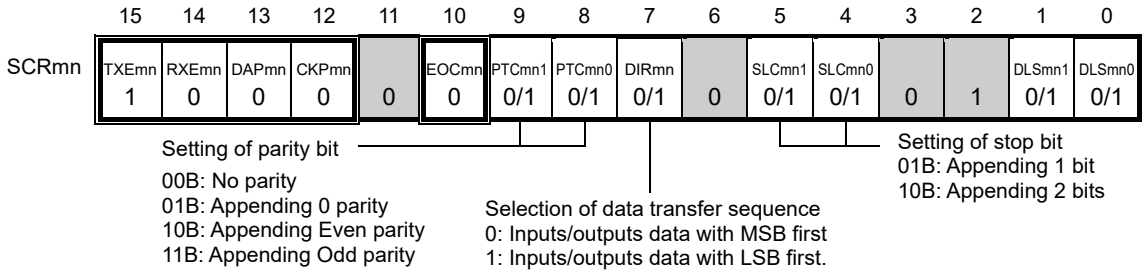
(1) Register setting

Figure 12-72. Example of Contents of Registers for UART Transmission of UART (UART0) (1/2)

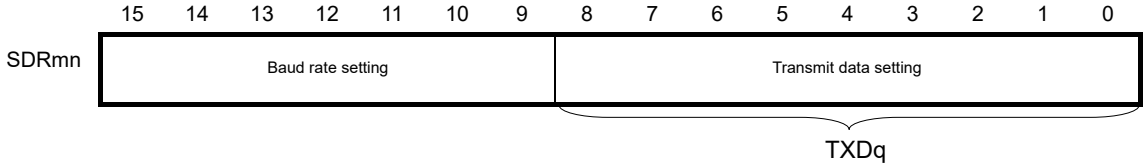
(a) Serial mode register mn (SMRmn)



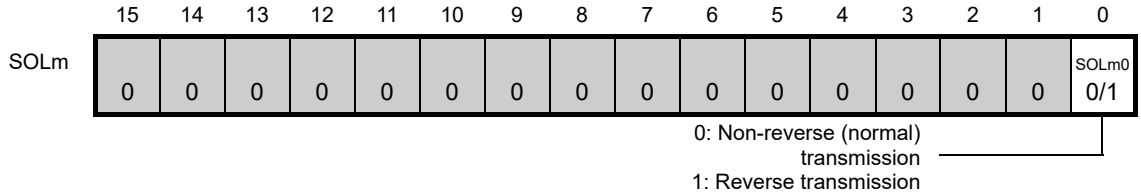
(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



(d) Serial output level register m (SOLm) … Sets only the bits of the target channel.



Note When UART0 performs 9-bit communication, bits 0 to 8 of the SDRm0 register are used as the transmission data specification area.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0)

2. □: Setting is fixed in the UART 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

Figure 12-72. Example of Contents of Registers for UART Transmission of UART (UART0) (2/2)

(e) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 0/1[Note]

0: Serial data output value is “0”
1: Serial data output value is “1”

(f) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 0/1

(g) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

Note Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0)

2. ☐ : 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 12-73. Initial Setting Procedure for UART Transmission

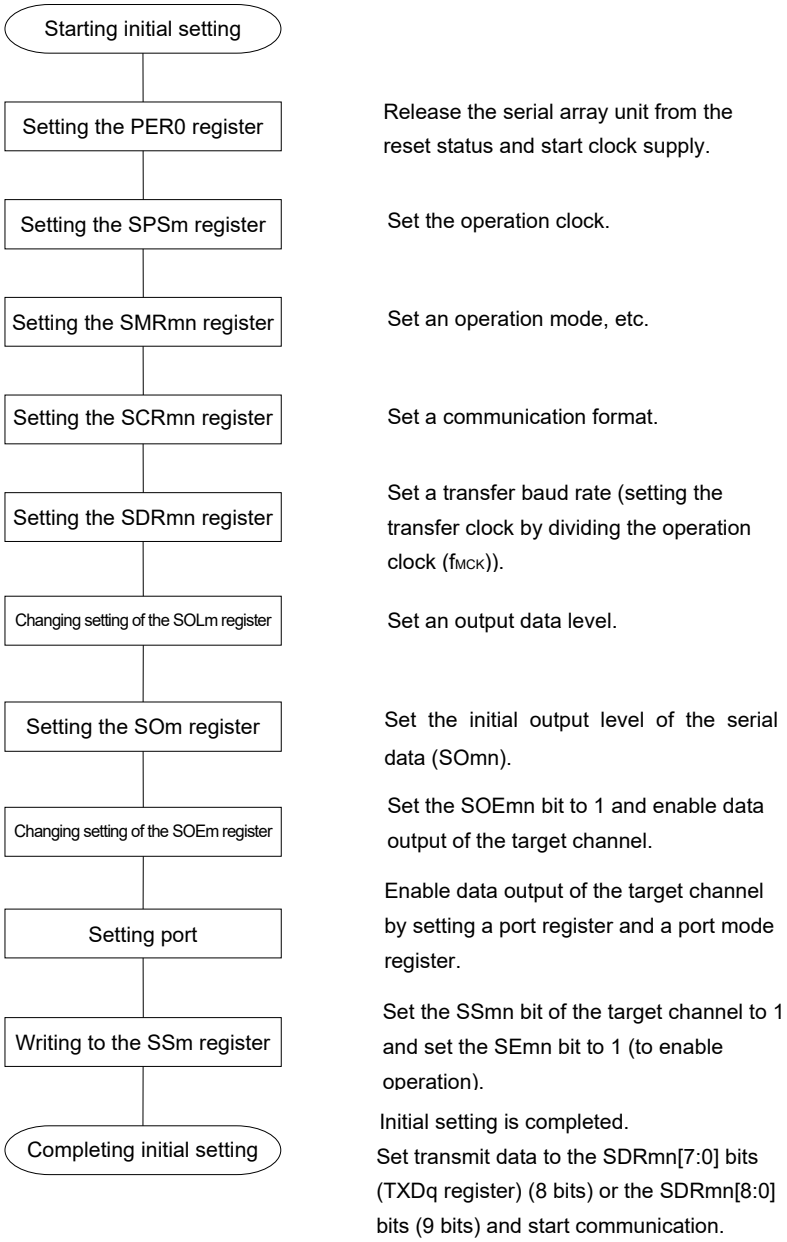


Figure 12-74. Procedure for Stopping UART Transmission

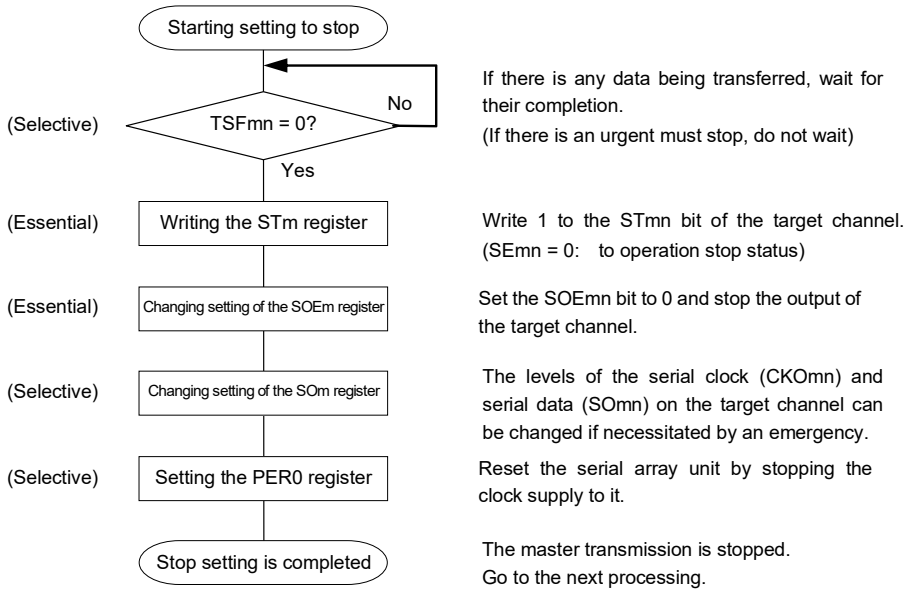
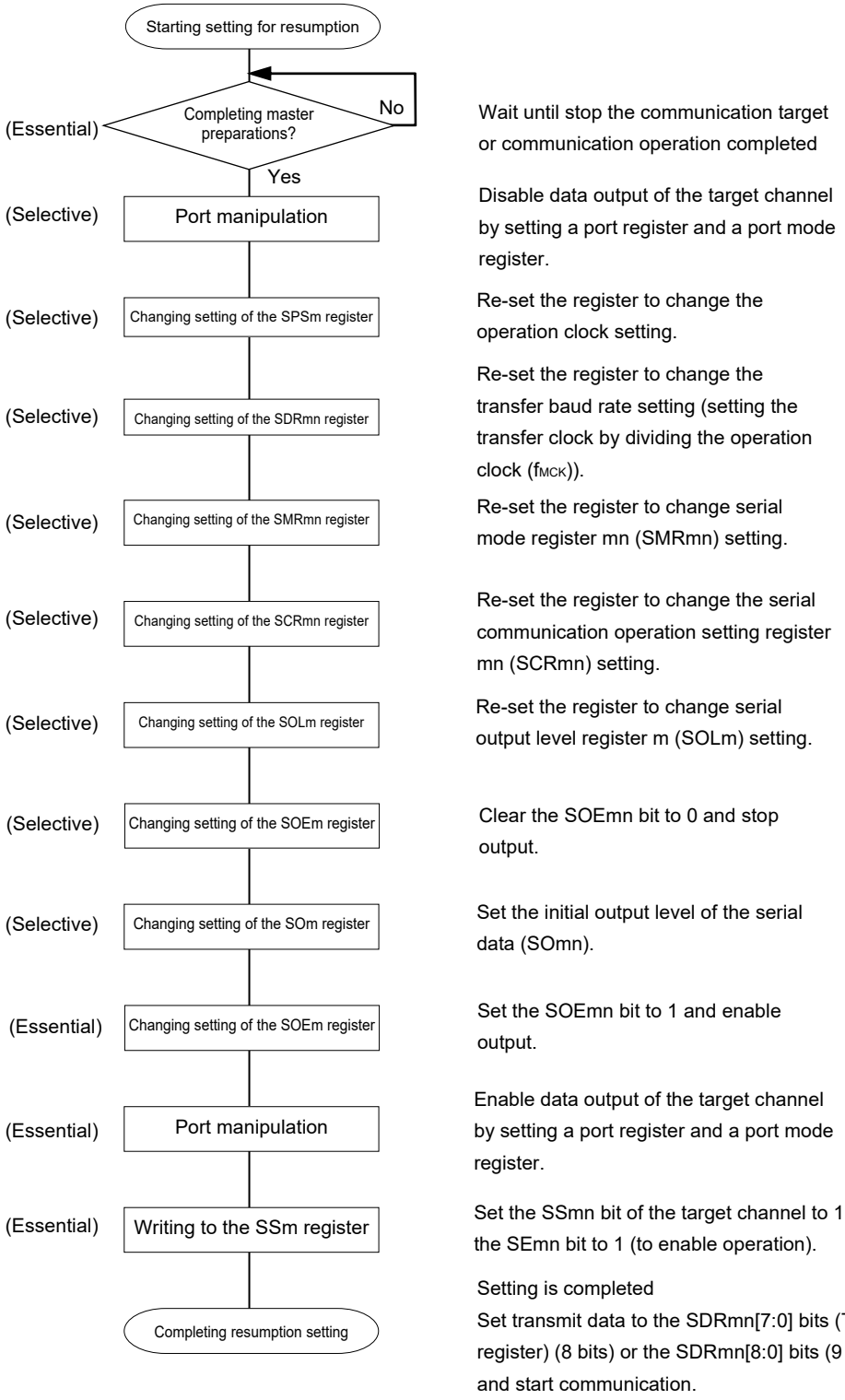


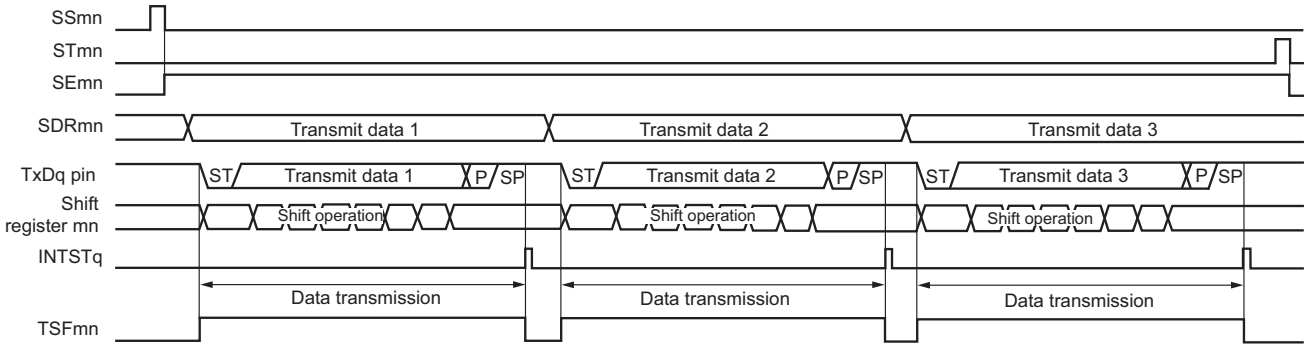
Figure 12-75. Procedure for Resuming UART Transmission



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.

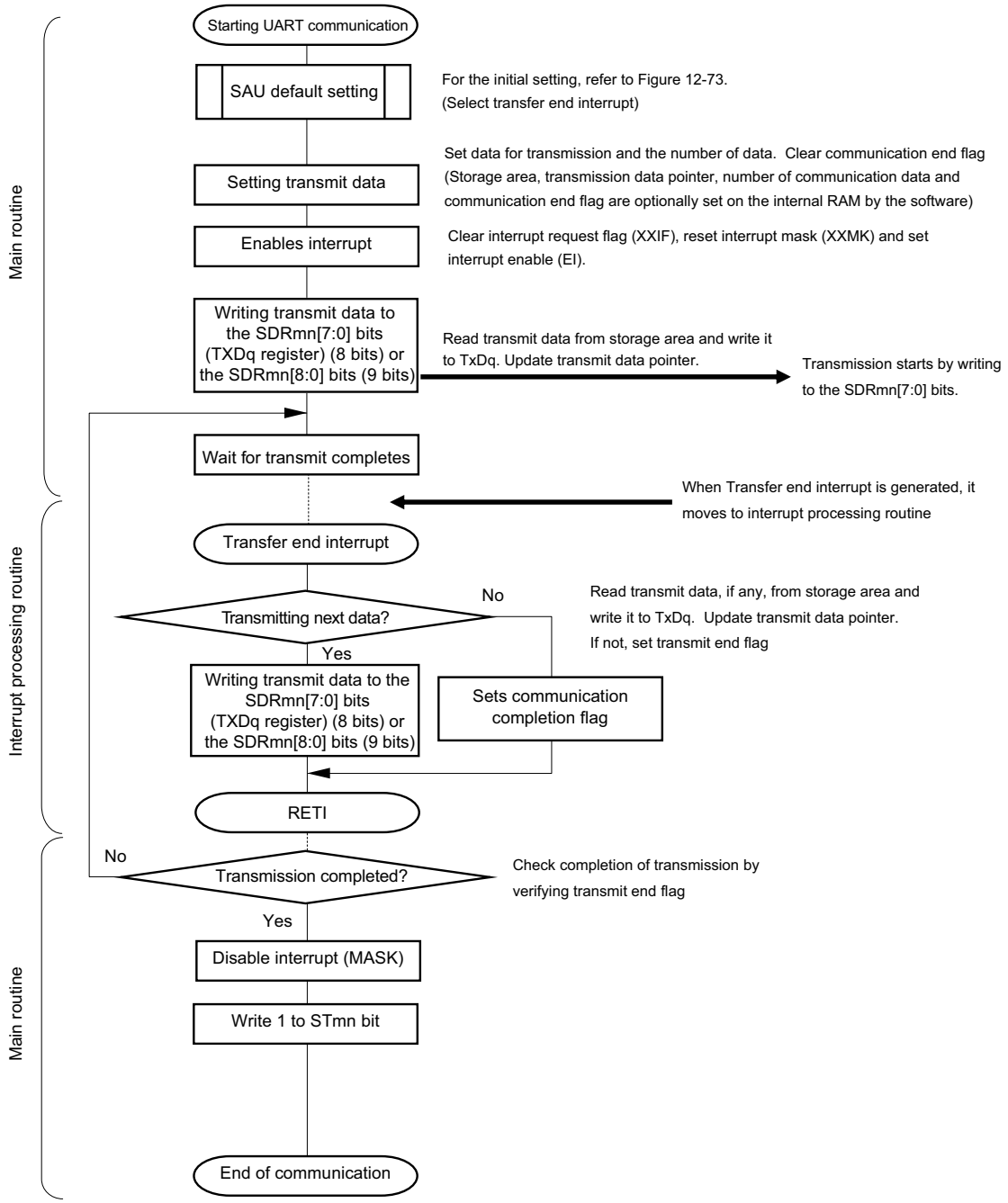
(3) Processing flow (in single-transmission mode)

Figure 12-76. Timing Chart of UART Transmission (in Single-Transmission Mode)



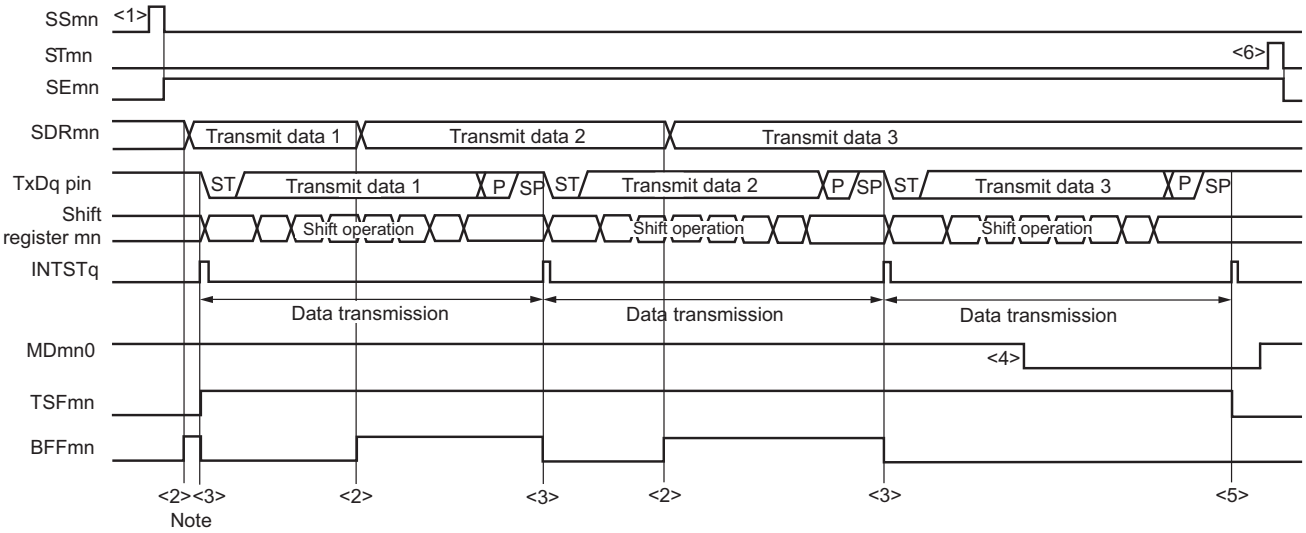
Remark m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0)

Figure 12-77. Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 12-78. Timing Chart of UART Transmission (in Continuous Transmission Mode)

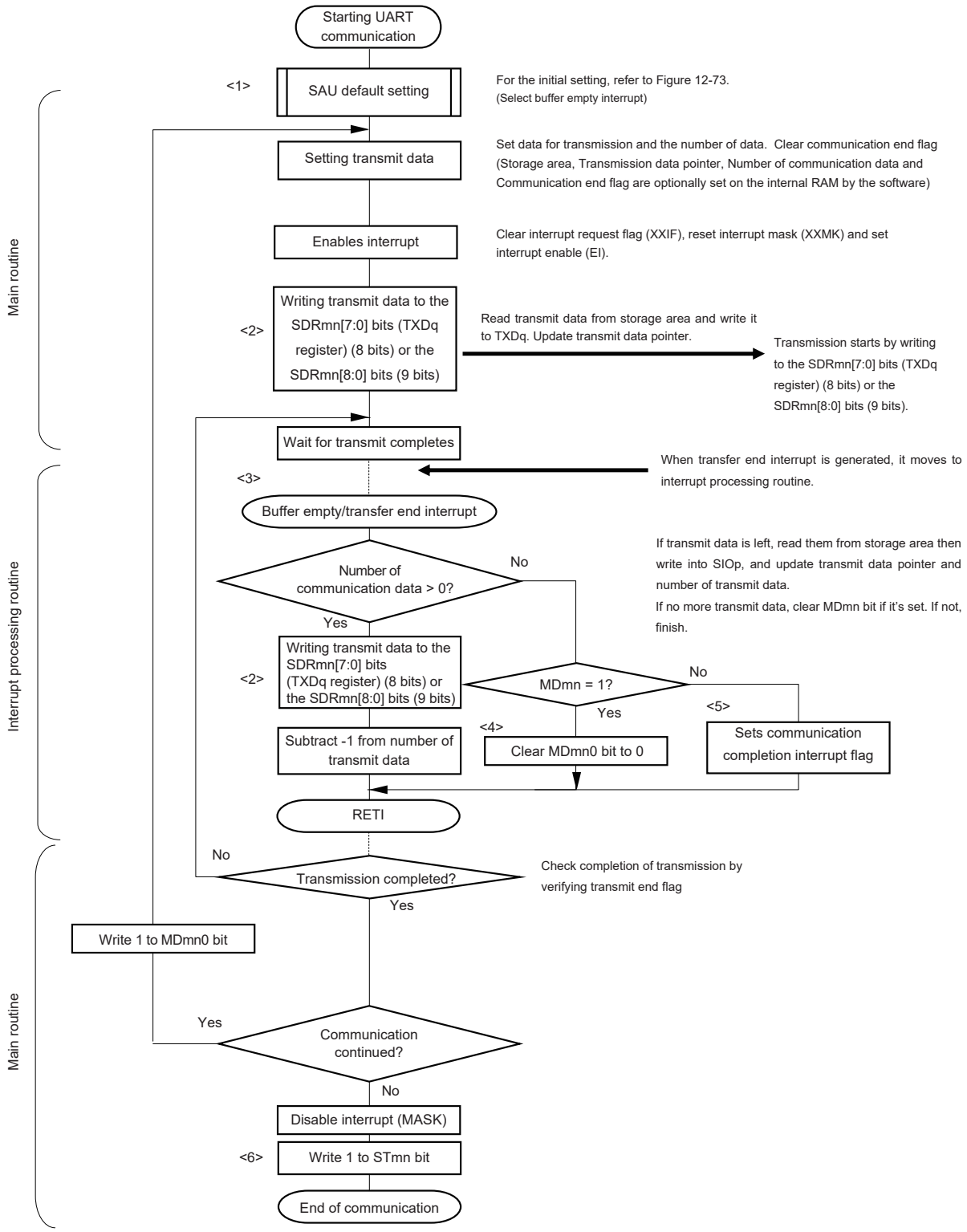


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 bit of serial mode register mn (SSRmn) 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 m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0)

Figure 12-79. Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 12-78 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

12.6.2 UART reception

UART reception is an operation wherein the RL78 microcontroller 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 (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)
Transfer data length	7, 8 or 9 bits
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 3 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] Note
Data phase	Forward output (default: high level) Reverse output (default: low level)
Parity bit	The following selectable • No parity bit (no parity check) • Appending 0 parity (no parity check) • Appending even parity • Appending odd parity
Stop bit	Appending 1 bit
Data direction	MSB or LSB first

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 30 or 31 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

2. m: Unit number (m = 0), n: Channel number (n = 1)

(1) Register setting

Figure 12-80. Example of Contents of Registers for UART Reception of UART (UART0) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 1	0	SISmn0 0/1	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

0: Forward (normal) reception
1: Reverse reception

Operation mode of channel n
0: Transfer end interrupt

(b) Serial mode register mr (SMRmr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmr	CKSmr 0/1	CCSmr 0	0	0	0	0	0	0	0	0	1	0	0	MDmr2 0	MDmr1 1	MDmr0 0/1

Same setting value as CKSmn bit

Operation mode of channel r
0: Transfer end interrupt
1: Buffer empty interrupt

(c) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	EOCmn 0/1	PTCmn1 0/1	PTCmn0 0/1	DIRmn 0/1	0	0	0	0	1	DLSmn1 0/1	DLSmn0 0/1

Setting of parity bit
00B: No parity
01B: No parity judgment
10B: Appending Even parity
11B: Appending Odd parity

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first.

Setting of data length

(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							Receive data register								

RXDq

Caution For the UART reception, be sure to set the SMRmr register of channel r to UART transmission mode that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 1),
r: Channel number (r = 0), q: UART number (q = 0)

2. □: Setting is fixed in the UART 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

Figure 12-80. Example of Contents of Registers for UART Reception of UART (UART0) (2/2)

(e) Serial output register m (SOm) … The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 ×

(f) Serial output enable register m (SOEm) …The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 ×

(g) Serial channel start register m (SSm) … Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 ×

Caution For the UART reception, be sure to set the SMRmr register of channel r to UART Transmission mode that is to be paired with channel n.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 1),
r: Channel number (r = 0), q: UART number (q = 0)

2. □: Setting is fixed in the UART 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 12-81. Initial Setting Procedure for UART Reception

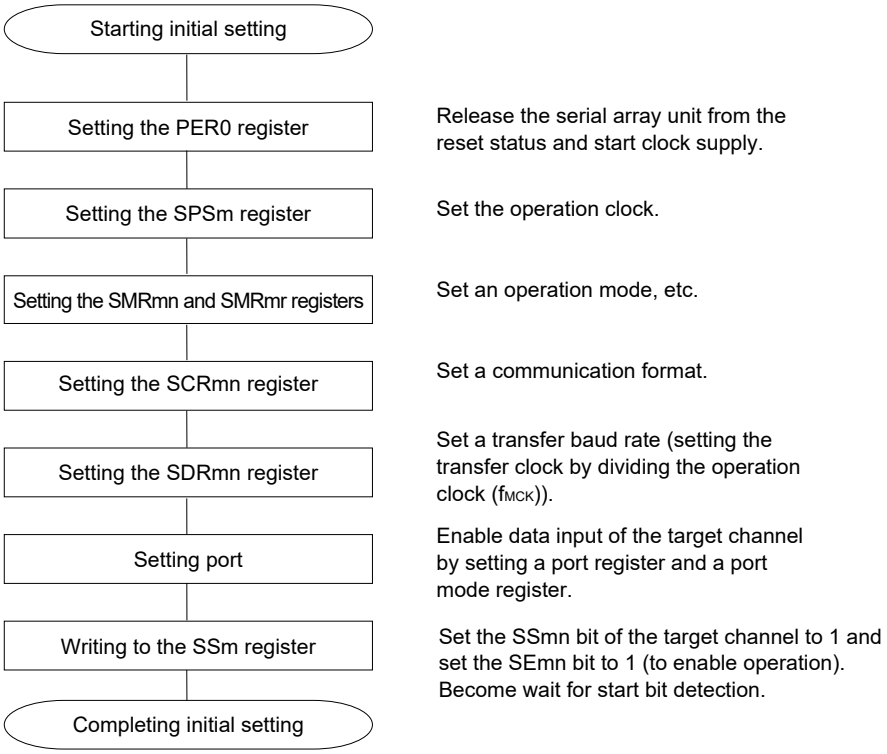


Figure 12-82. Procedure for Stopping UART Reception

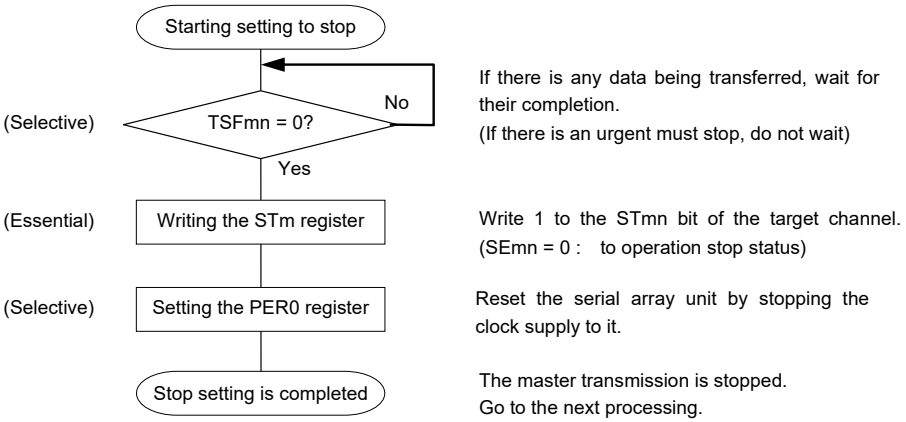
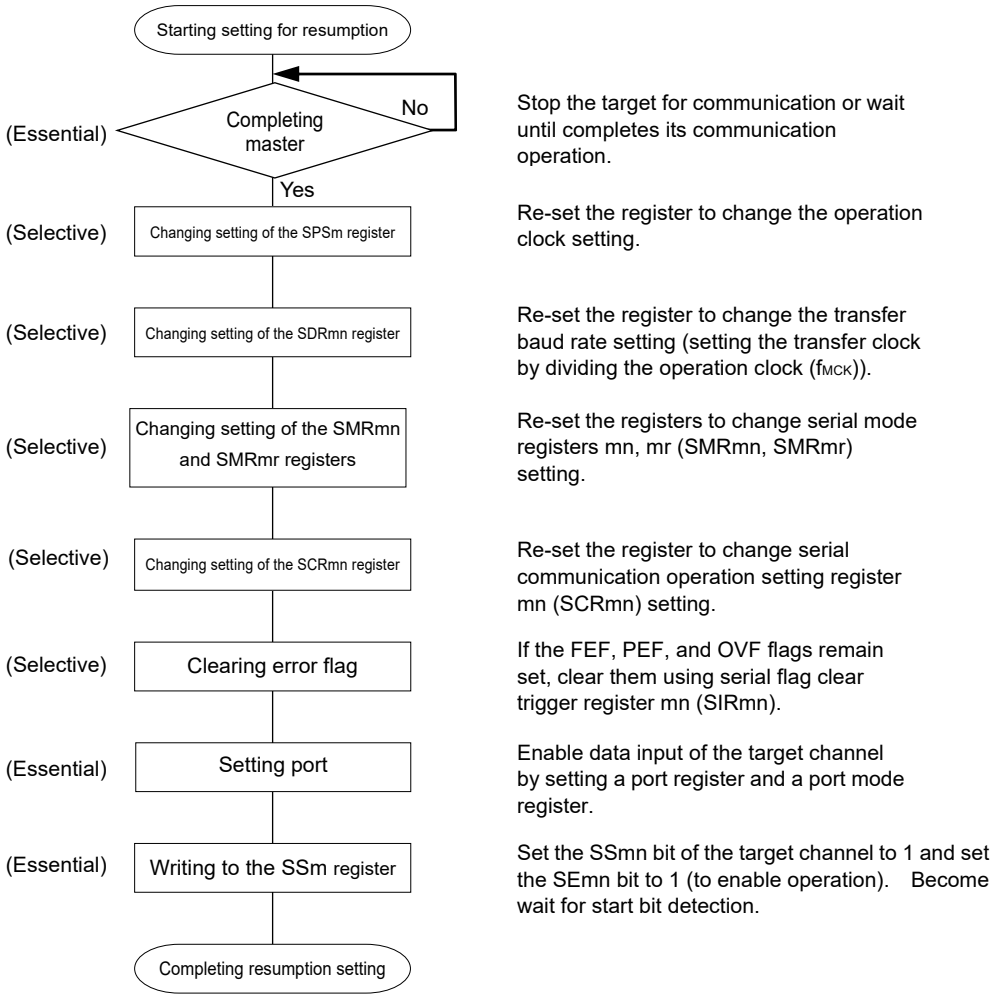


Figure 12-83. Procedure for Resuming UART Reception

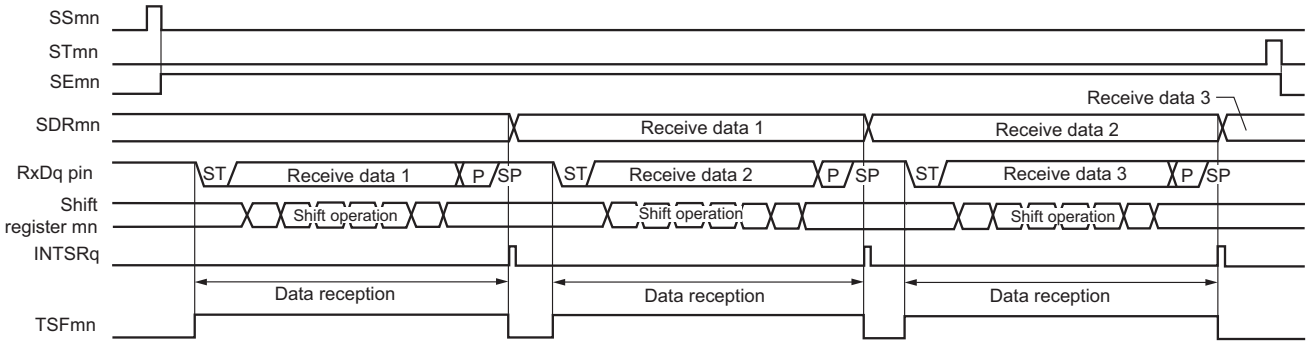


Caution After is set RXEmn bit to 1 of SCRmn register, set the SSmn = 1 from an interval of at least four clocks of f_{MCK}.

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.

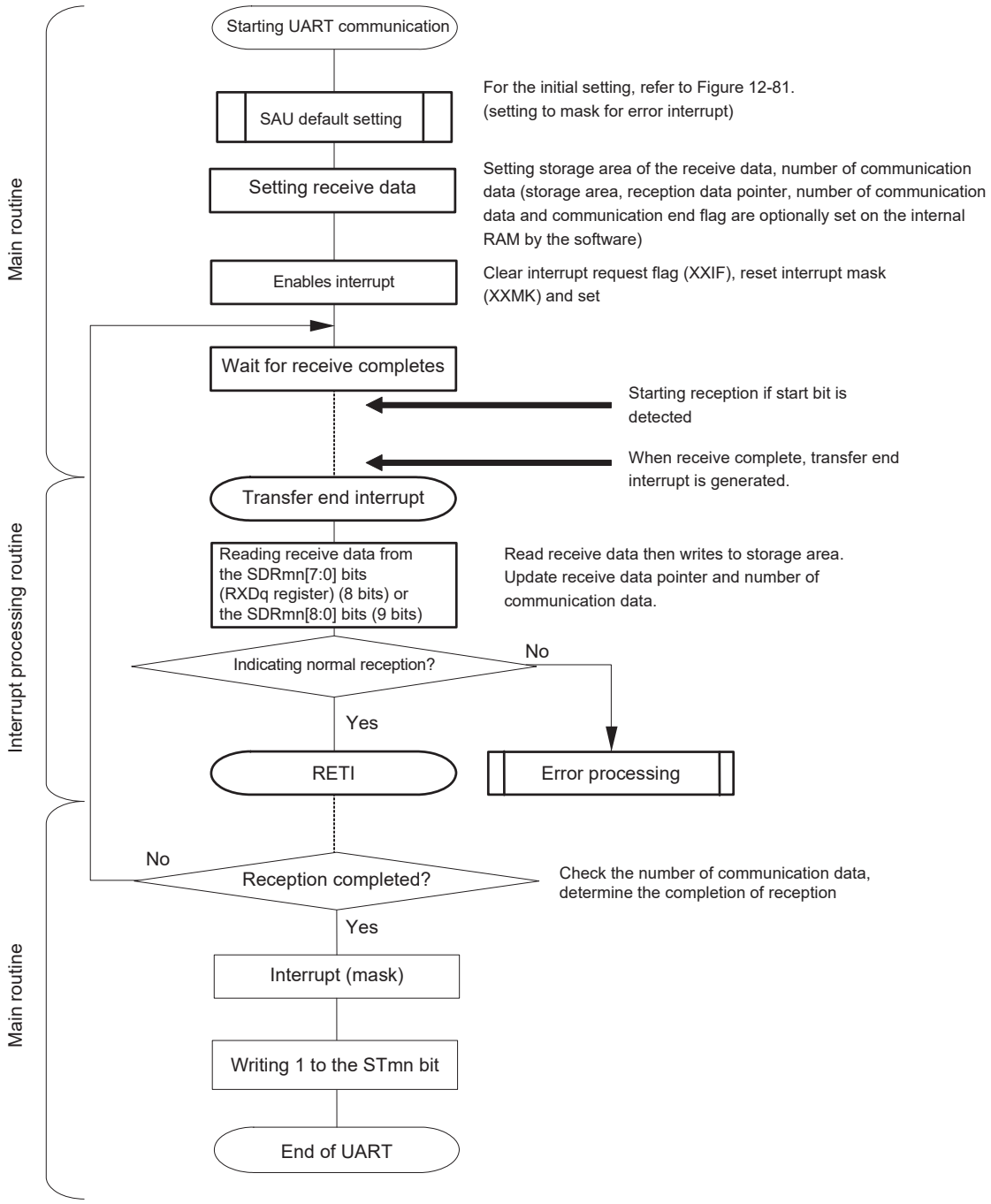
(3) Processing flow

Figure 12-84. Timing Chart of UART Reception



Remark m: Unit number (m = 0), n: Channel number (n = 1),
r: Channel number (r = 0), q: UART number (q = 0)

Figure 12-85. 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} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

Remarks 1. When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 12-3. Selection of Operation Clock For UART

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	732 Hz
Other than above									Setting prohibited	

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 m (STm) = 0003H) the operation of the serial array unit (SAU).

Remarks 1. X: Don't care

2. m: Unit number (m = 0), n: Channel number (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} = 24 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	77	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^8$	77	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^7$	77	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^6$	77	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^5$	77	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^4$	77	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^3$	77	19230.8 bps	+0.16 %
31250 bps	$f_{CLK}/2^3$	47	31250.0 bps	±0.0 %
38400 bps	$f_{CLK}/2^2$	77	38461.5 bps	+0.16 %
76800 bps	$f_{CLK}/2$	77	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	77	153846 bps	+0.16 %
312500 bps	f_{CLK}	37	315789 bps	+1.05 %

Remark m: Unit number (m = 0), n: Channel number (n = 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.

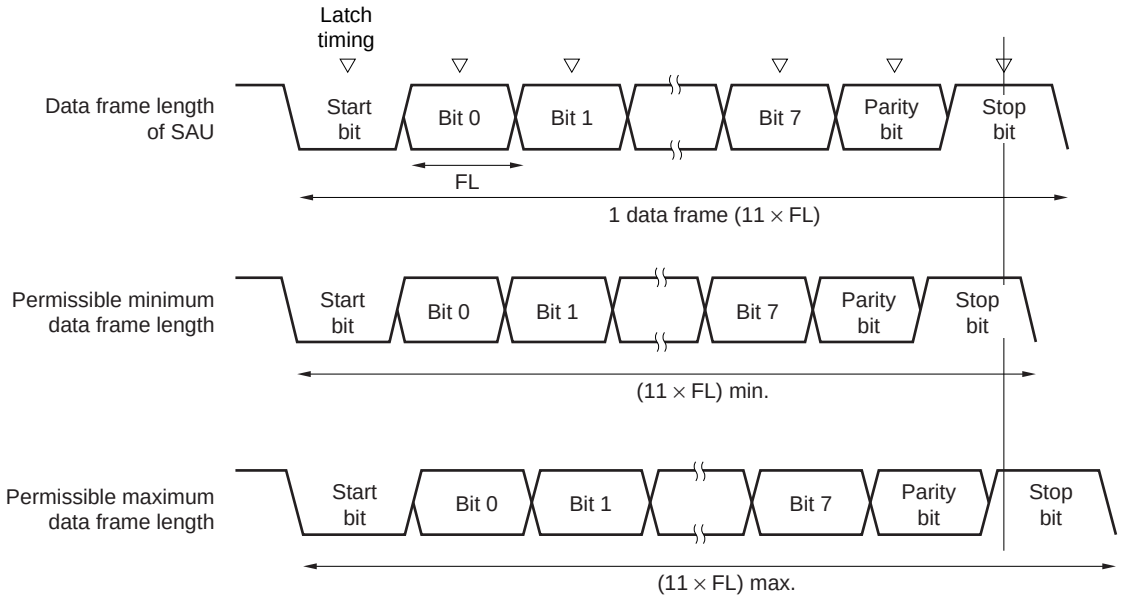
$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **12.6.3 (1) Baud rate calculation expression**.)
k: SDRmn[15:9] + 1
Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0), n: Channel number (n = 1)

Figure 12-86 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 12-86, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of serial data register mn (SDRmn) 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 12-87 and 12-88.

Figure 12-87. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn 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 mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 12-88. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn 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 mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) 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 SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

12.7 Operation of Simplified I²C (IIC00, IIC01) 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.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[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.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Overrun error
- Parity error (ACK error)

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- 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 SOEmn (SOEm register) bit and serial communication data output is stopped. See the processing flow in **12.7.3 (2)** for details.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

The channel supporting simplified I²C (IIC00, IIC01) is channels 0 and 1.

Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	CSI00	UART0	IIC00
1	CSI01		IIC01

Simplified I²C (IIC00, IIC01) 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	IIC01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1
Interrupt	INTIIC00	INTIIC01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Parity error detection flag (PEFmn)	
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 Note 2	Max. $f_{MCK}/2$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I²C. • Max. 1 MHz (first mode plus) • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)	
Data level	Forward 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 (POMxx = 1) with the port output mode register (POMxx).
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.3 Registers Controlling Port Function** and **4.4.4 Connecting to external device with different potential (1.8 V, 2.5 V, 3 V)**.

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 30 or 31 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

(1) Register setting

Figure 12-89. Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC01) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn Note 1 0	0	SISmn0 Note 1 0	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Operation mode of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 Note 2 0	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit (ACK)

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting (address + R/W)							

SIOr

(d) Serial output register m (SOm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

Start condition is generated by manipulating the SOmn bit.

(e) Serial output enable register m (SOEm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

SOEmn = 0 until the start condition is generated, and SOEmn = 1 after generation.

Notes 1. Only provided for the SMR01 register.
2. Only provided for the SCR00 register.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01)
2. □: Setting is fixed in the IIC 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

Figure 12-89. Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC01) (2/2)

(f) Serial channel start register m (SSm) … Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01)

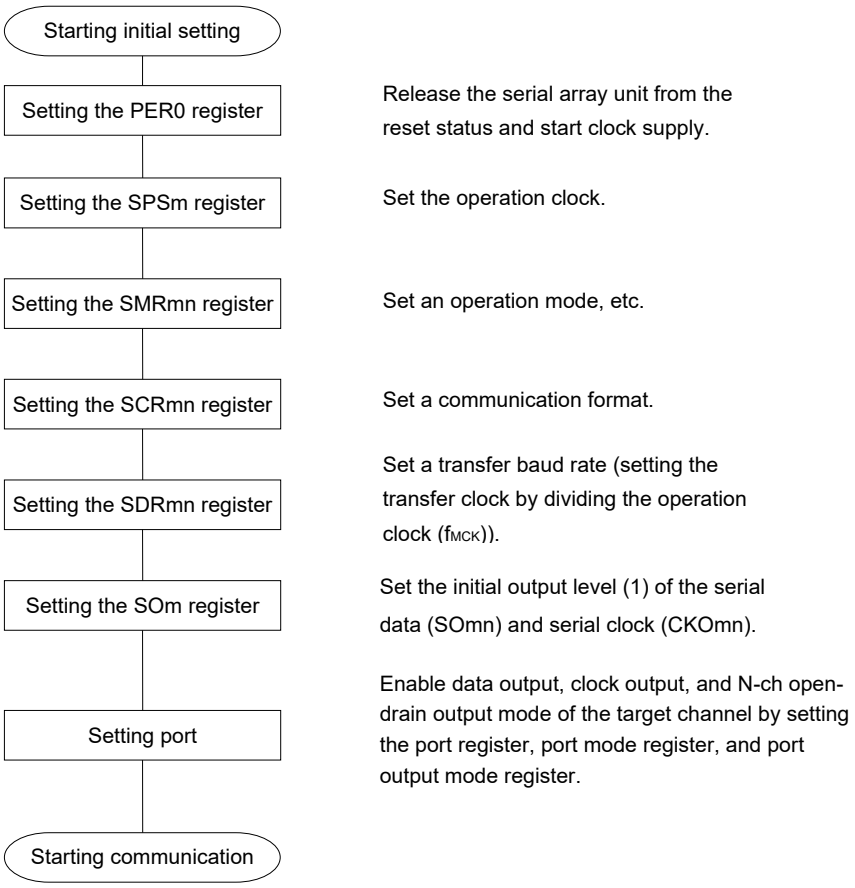
2. ☐: Setting is fixed in the IIC mode, ▒ : Setting disabled (set to the initial value)

×: 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

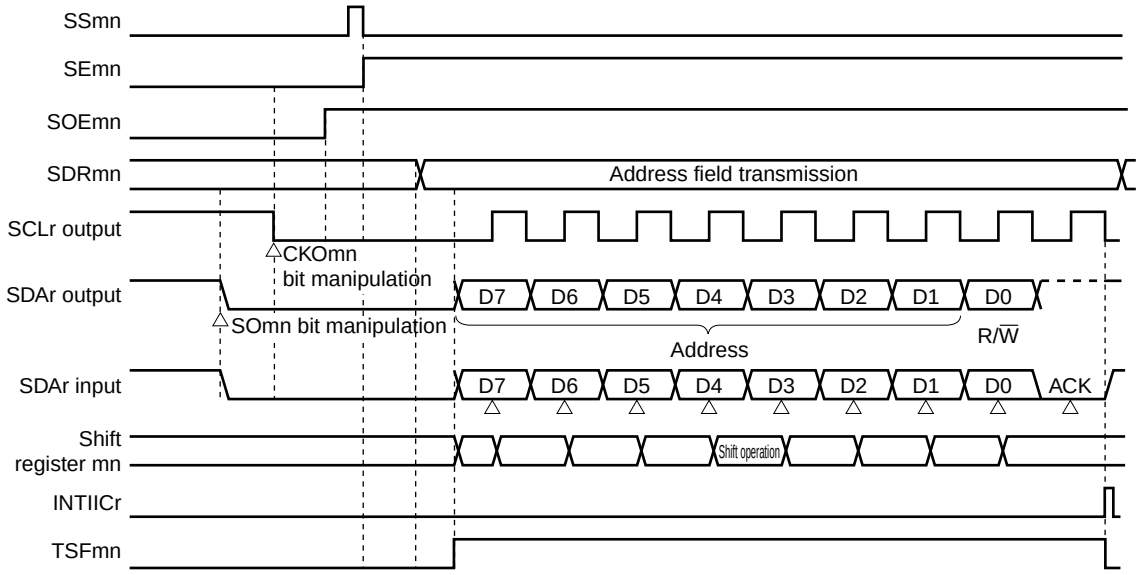
Figure 12-90. Initial Setting Procedure for Simplified I²C



Remark At the end of the initial setting, the simplified I²C (IIC00, IIC01) must be set so that output is disabled and operations are stopped.

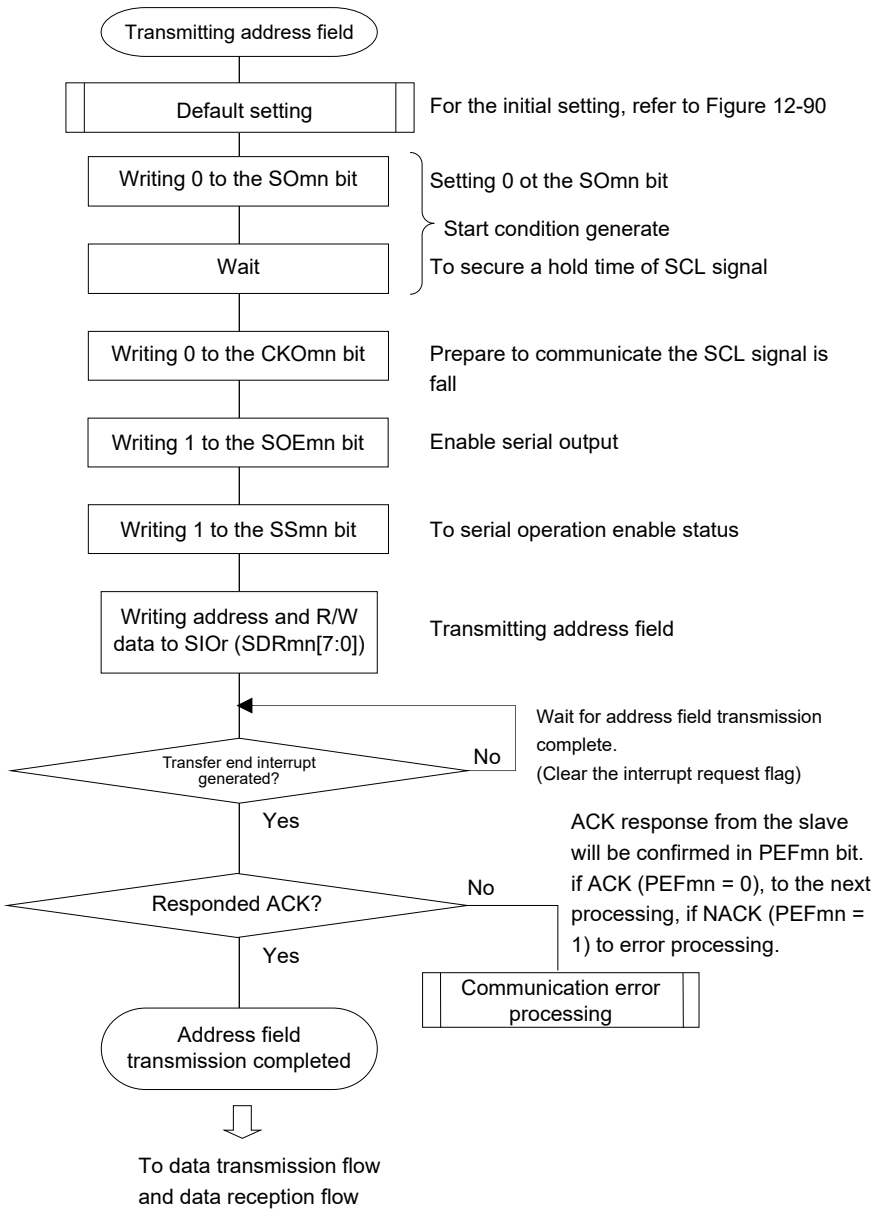
(3) Processing flow

Figure 12-91. Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01)

Figure 12-92. 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 I^{2}C	IIC00	IIC01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1
Interrupt	INTIIC00	INTIIC01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Parity error detection flag (PEFmn)	
Transfer data length	8 bits	
Transfer rate Note 2	Max. f_{MCK}/2 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^{2}C. • Max. 1 MHz (first mode plus) • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)	
Data level	Forward 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 I^{2}C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx).
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.3 Registers Controlling Port Function** and **4.4.4 Connecting to external device with different potential (1.8 V, 2.5 V, 3 V)**.

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 30 and 31 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

(1) Register setting

Figure 12-93. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00, IIC01) (1/2)

(a) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 Note 1	0	SISmn0 0 Note 1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(b) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 Note 2	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting							

SIOr

(d) Serial output register m (SOm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1 Note 3	CKOm0 0/1 Note 3	0	0	0	0	0	0	SOm1 0/1 Note 3	SOm0 0/1 Note 3

(e) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 1	SOEm0 1

Notes 1. Only provided for the SMR01 register.
2. Only provided for the SCR00 register.
3. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01)
2. ☐: Setting is fixed in the IIC 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

Figure 12-93. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00, IIC01) (2/2)

(f) Serial channel start register m (SSm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01)

2. ▒ : 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 Transmission

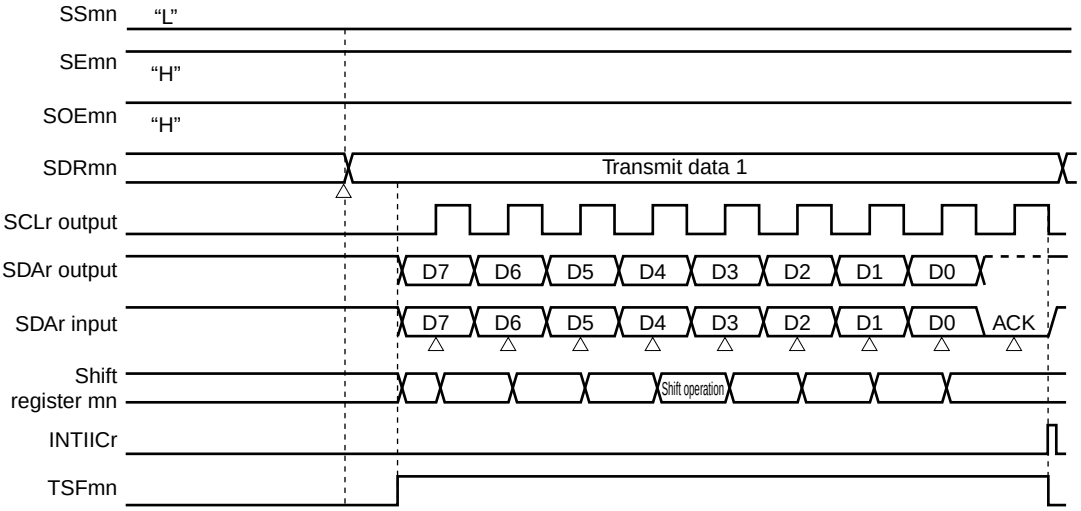
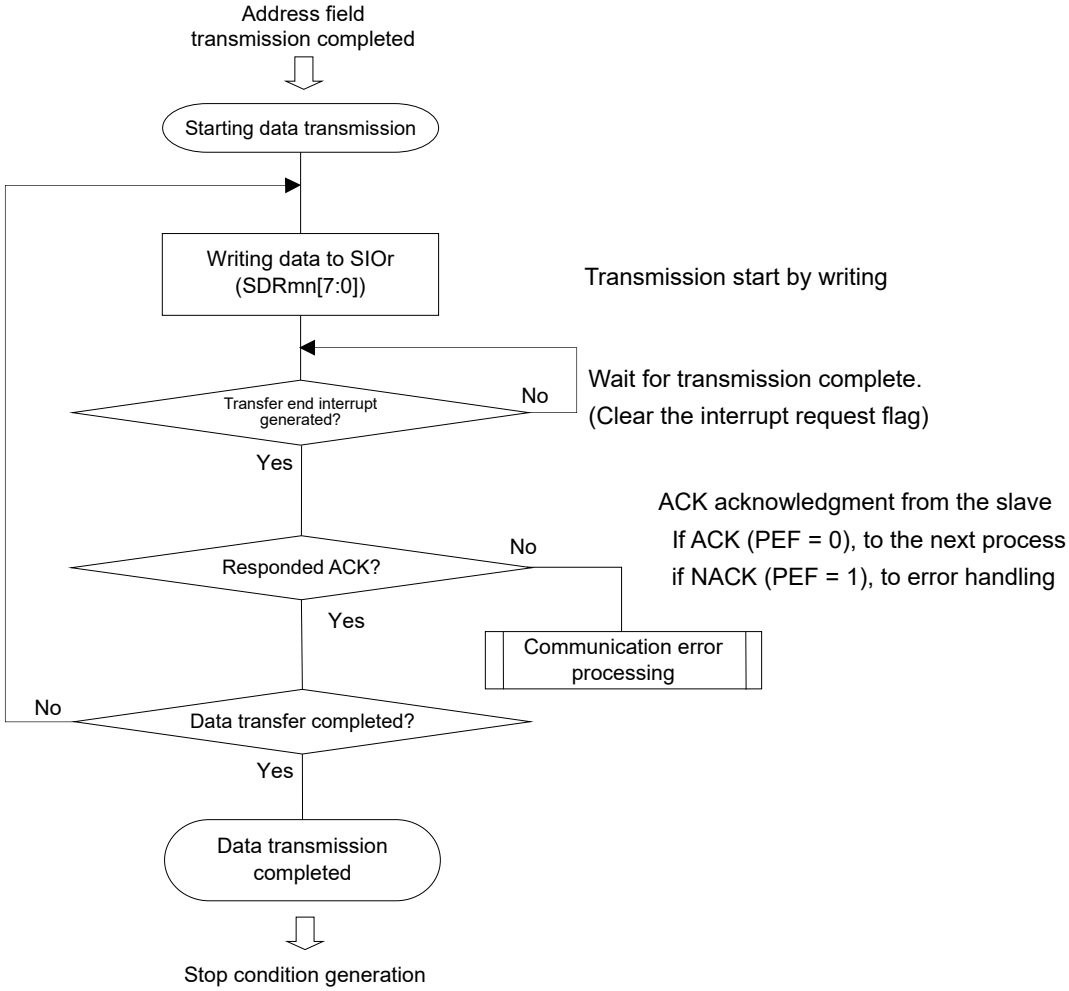


Figure 12-95. Flowchart of Simplified I²C Data Transmission



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	IIC01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1
Interrupt	INTIIC00	INTIIC01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	8 bits	
Transfer rate Note 2	Max. $f_{MCK}/2$ [Hz] (SDRmn[15:9] = 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. 1 MHz (first mode plus) • Max. 400 kHz (first mode) • Max. 100 kHz (standard mode)	
Data level	Forward 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 (POMxx = 1) with the port output mode register (POMxx).
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.3 Registers Controlling Port Function** and **4.4.4 Connecting to external device with different potential (1.8 V, 2.5 V, 3 V)**.

2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 30 and 31 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

(1) Register setting

Figure 12-96. Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01) (1/2)

(a) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 Note 1	0	SISmn0 0 Note 1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(b) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 Note 2	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Dummy transmit data setting (FFH)							

SIOr

(d) Serial output register m (SOm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1 Note 3	CKOm0 0/1 Note 3	0	0	0	0	0	0	SOm1 0/1 Note 3	SOm0 0/1 Note 3

(e) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

Notes 1. Not provided for the SMR00 register.

2. Not provided for the SCR00 register.

3. The value varies depending on the communication data during communication operation.

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01)

2. □: Setting is fixed in the IIC 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

Figure 12-96. Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01) (2/2)

(f) Serial channel start register m (SSm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remarks 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01)

2. ☐: Setting is fixed in the IIC mode, ▒: Setting disabled (set to the initial value)

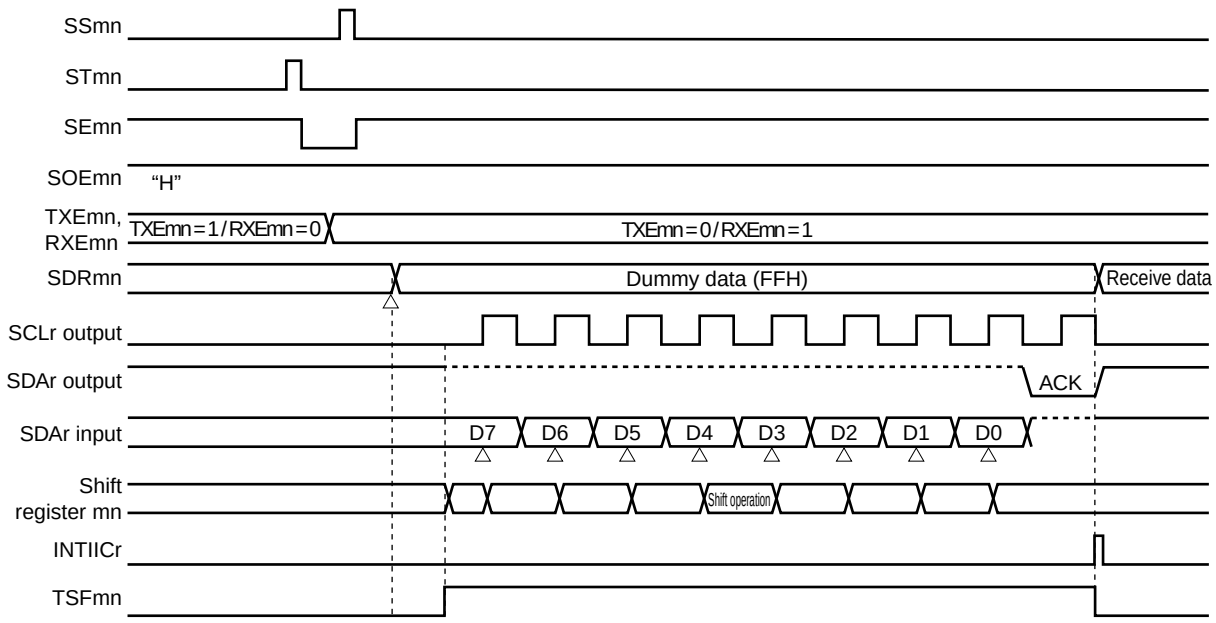
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

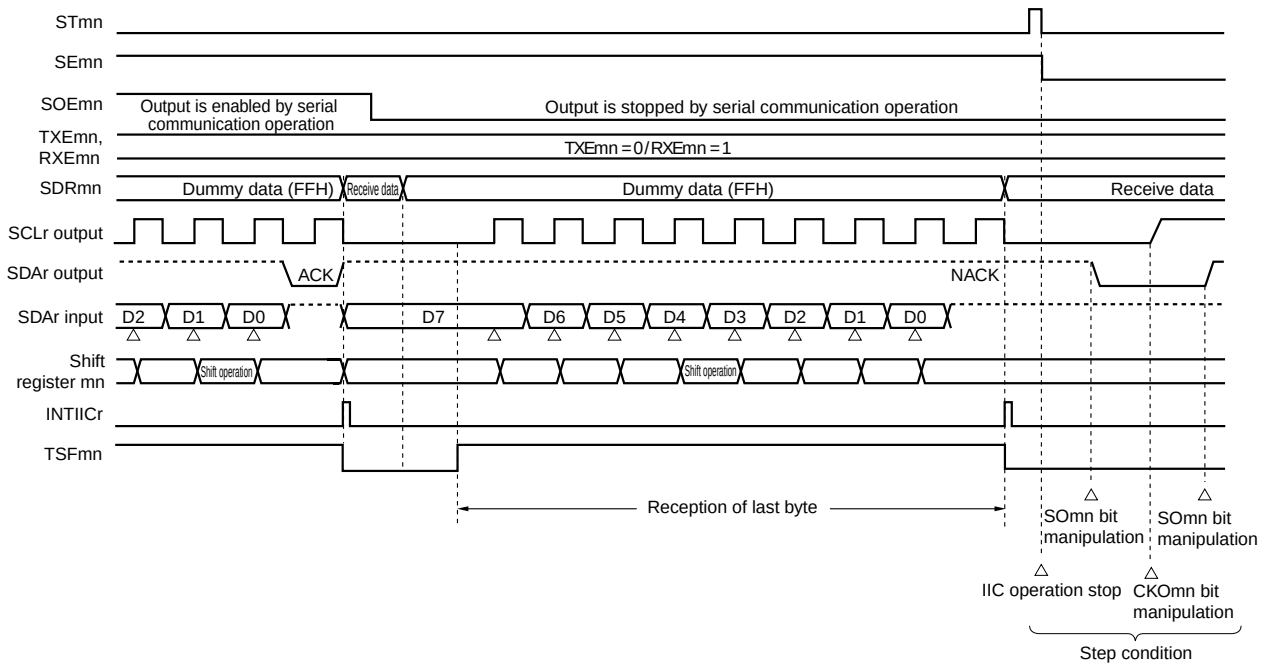
(2) Processing flow

Figure 12-97. Timing Chart of Data Reception

(a) When starting data reception

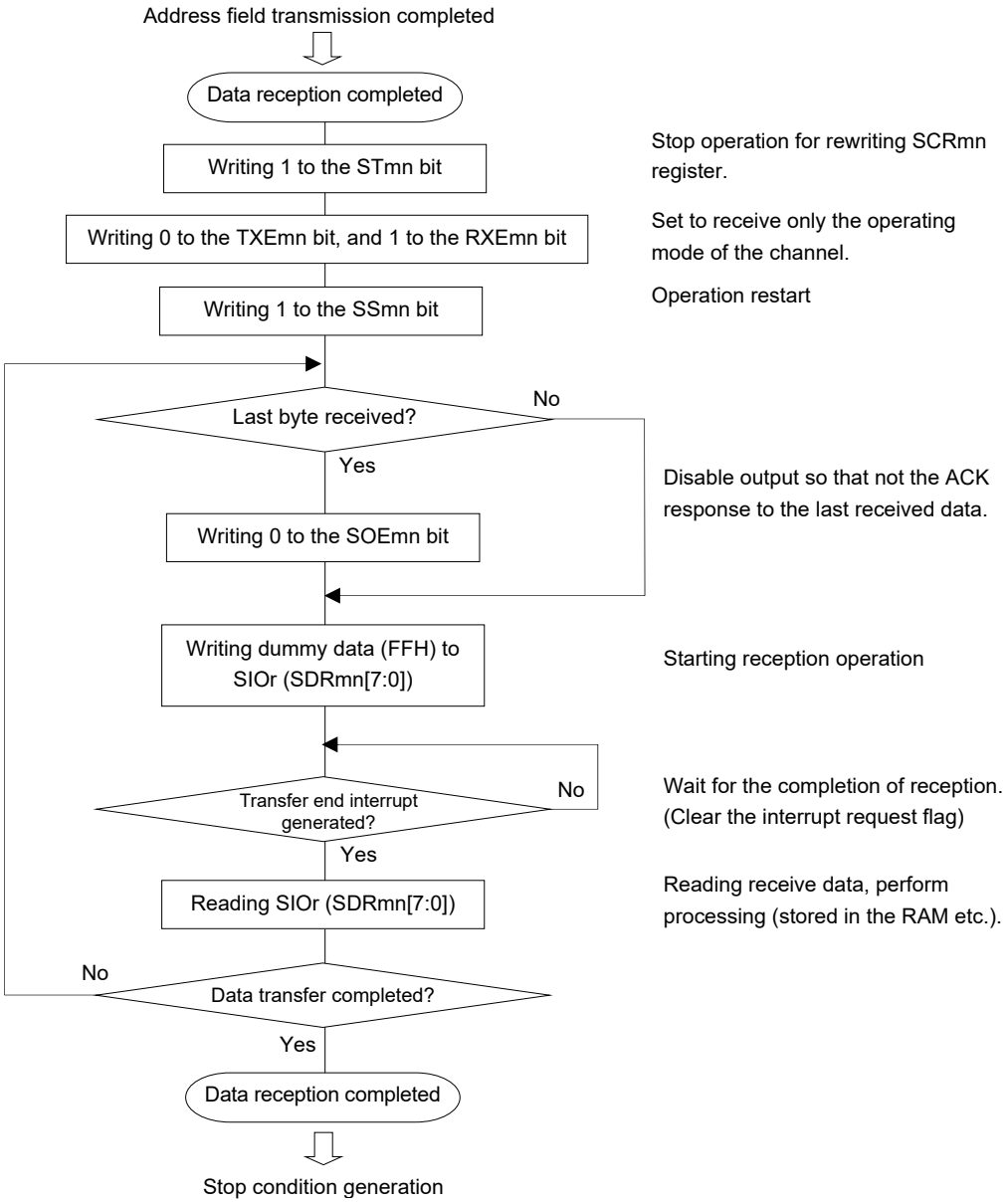


(b) When receiving last data



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01)

Figure 12-98. Flowchart of Data Reception



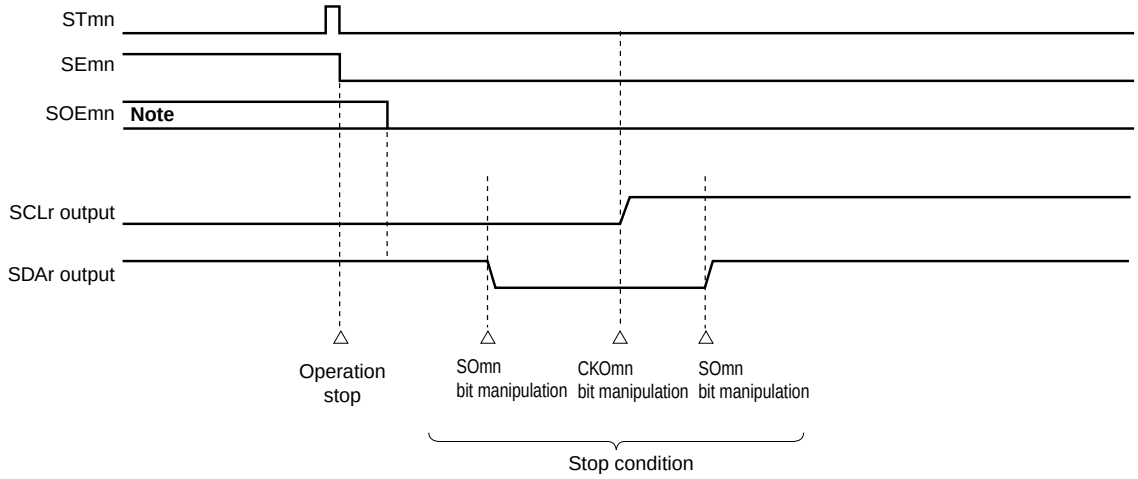
Caution **ACK is not output when the last data is received (NACK). Communication is then completed by setting "1" to the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.**

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.

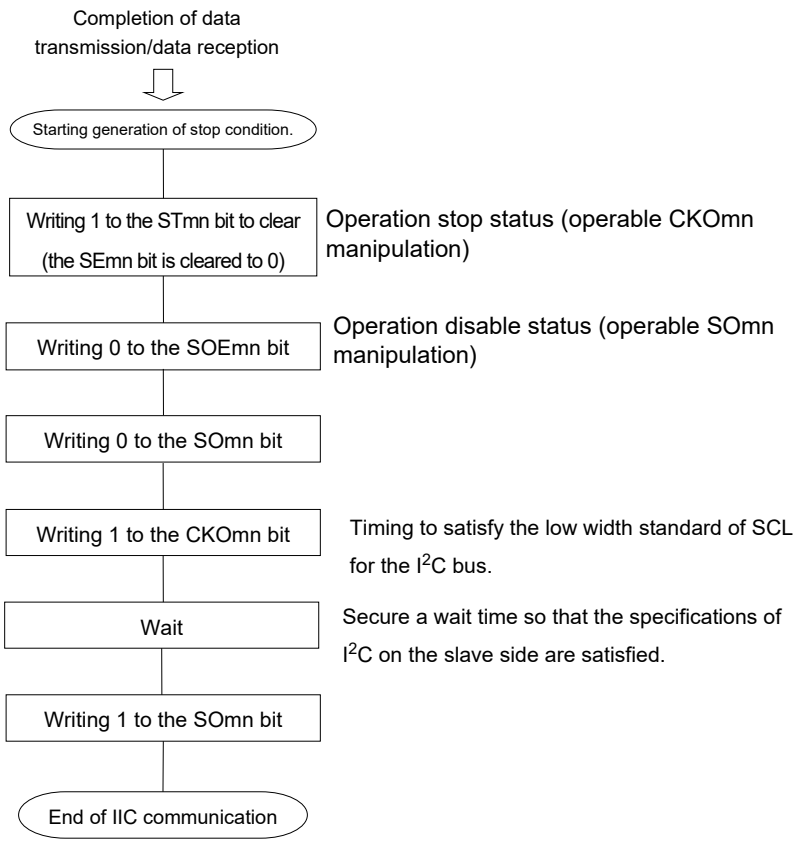
(1) Processing flow

Figure 12-99. Timing Chart of Stop Condition Generation



Note During a receive operation, the SOEmn bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 12-100. Flowchart of Stop Condition Generation



12.7.5 Calculating transfer rate

The transfer rate for simplified I²C (IIC00, IIC01) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution Setting SDRmn[15:9] = 0000000B is prohibited. Setting SDRmn[15:9] = 0000001B or more.

Remarks 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 12-4. Selection of Operation Clock For Simplified I²C

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	24 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	12 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	24 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	12 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	732 Hz
Other than above									Setting prohibited	

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 m (STm) = 0003H) the operation of the serial array unit (SAU).

Remarks 1. X: Don't care

2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

Here is an example of setting an I^2C transfer rate where $f_{MCK} = f_{CLK} = 24$ MHz.

I^2C Transfer Mode (Desired Transfer Rate)	$f_{CLK} = 24$ MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	59	100 kHz	0.0%
400 kHz	f_{CLK}	31	375 kHz	6.25% Note
1 MHz	f_{CLK}	14	0.80 MHz	20.0% 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, IIC01) communication

The procedure for processing errors that occurred during simplified I²C (IIC00, IIC01) communication is described in Figure 12-101 and 12-102.

Figure 12-101. Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn 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 mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	The 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.

Figure 12-102. Processing Procedure in Case of Parity Error (ACK error) in Simplified I²C Mode

Software Manipulation	Hardware Status	Remark
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	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 SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01)

CHAPTER 13 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

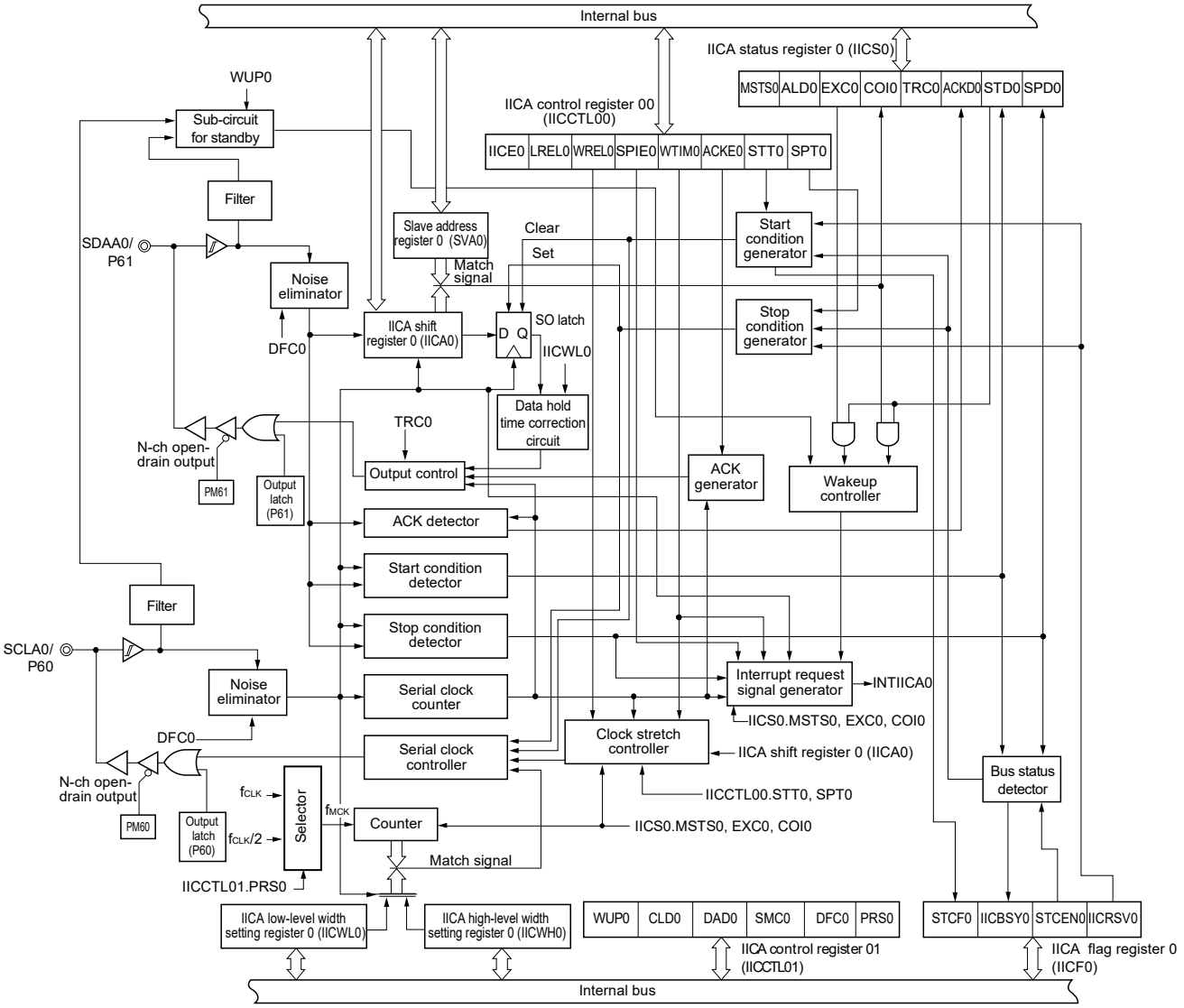
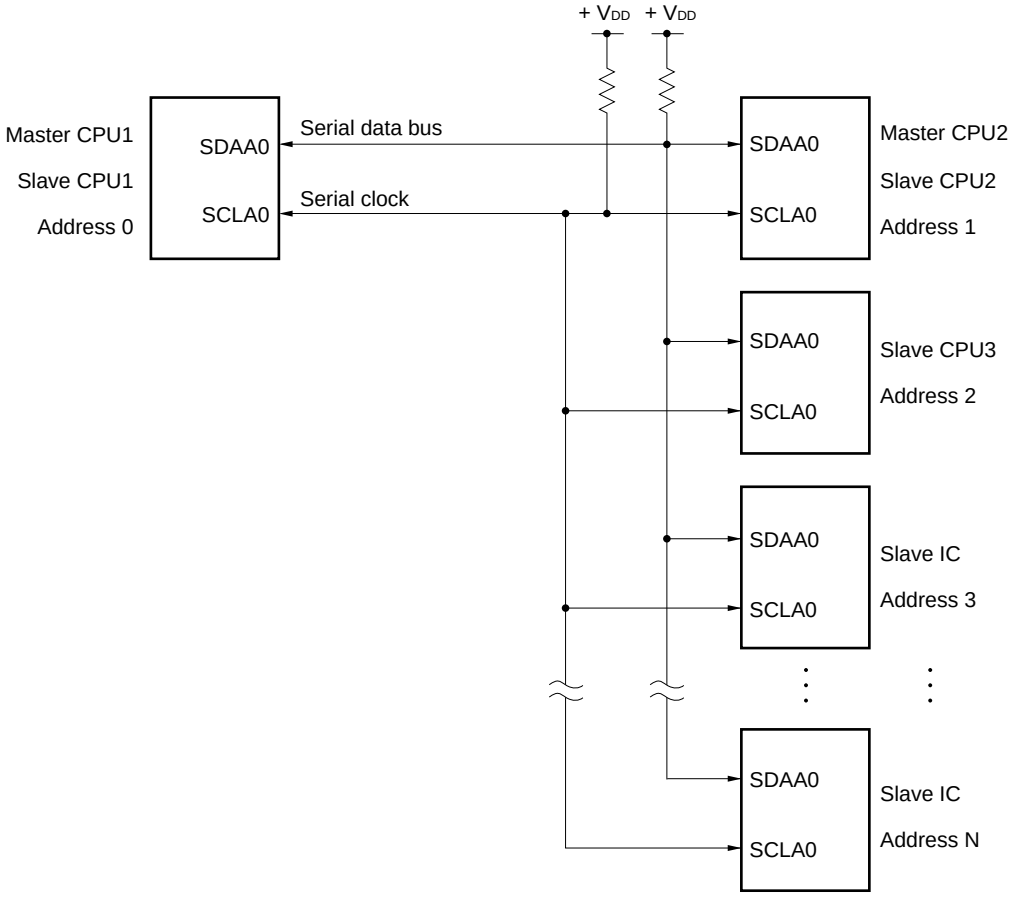


Figure 13-2 shows a serial bus configuration example.

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 6 (PM6) Port register 6 (P6)

(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 Bit 0 is fixed 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 STCF0 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 STCEN0 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 nine 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 6 (PM6)
- Port register 6 (P6)

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	RTCEN	0	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

IICAmEN	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.

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 6 (PM6) and port register 6 (P6)).

- **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)**

2. Be sure to clear the bits 1, 3, 6 to 0.

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 ($\overline{ACK}$) 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 IICBSY0 = 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 (IICRSV0 = 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 (IICRSV0 = 1) Even if this bit is set (1), the STT0 bit is cleared and the STT0 clear flag (STCF0) 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).
- Once STT0 is set (1), setting it again (1) before the clear condition is met is not allowed.

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. The STT0 bit is always read as 0.

Remarks 1. Bit 1 (STT0) becomes 0 when it is read after data setting.
2. 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 it 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.
- Once STT0 is set (1), setting it again (1) before the clear condition is met is not allowed.

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

Notes 1. The signal of this bit is invalid while IICE0 is 0.
2. When the SPT0 register 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.

Remark Bit 0 (SPT0) becomes 0 when it is read after data setting.

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 ($\overline{ACK}$)
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 (STCF0) and I^{2}C bus status flag (IICBSY0) bits are read-only.

The IICRSV0 bit can be used to enable/disable the communication reservation function.

The STCEN0 bit can be used to set the initial value of the IICBSY0 bit.

The IICRSV0 and STCEN0 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	PRS0

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 f_{MCK} clocks after setting (1) the WUP0 bit (see **Figure 13-22 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.	
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.

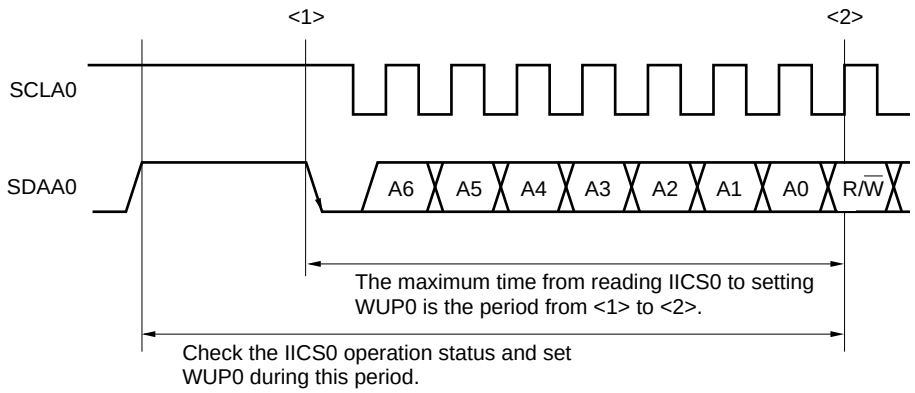


Figure 13-9. Format of IICA Control Register 01 (IICCTL01) (2/2)

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) or fast mode plus (fastest transfer rate: 1 Mbps).

DFC0	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in fast mode and fast mode plus. In fast mode and fast mode plus, the transfer clock does not vary, regardless of the DFCn bit being set (1) or cleared (0).	

PRS0	IICA operation clock (f_{MCK}) control
0	Selects f_{CLK} (1 MHz ≤ f_{CLK} ≤ 20 MHz)
1	Selects $f_{CLK}/2$ (20 MHz < f_{CLK})

Cautions 1. The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (max.). Set bit 0 (PRS0) of the IICA control register 01 (IICCTL01) to "1" only when the f_{CLK} exceeds 20 MHz.

2. 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.)
Fast mode plus: f_{CLK} = 10 MHz (MIN.)
Normal mode: f_{CLK} = 1 MHz (MIN.)

3. The fast mode plus is only available in the products for "A: Consumer applications (T_A = -40 °C to +85°C)".

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. The data hold time is decided by value the higher 6 bits of IICWL register.

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								

Setting value of data hold time

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.

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 Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCLA0 pin as clock I/O and the P61/SDAA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set the IICE0 bit (bit 7 of IICA control register 00 (IICCTL00)) to 1 before setting the output mode because the P60/SCLA0 and P61/SDAA0 pins output a low level (fixed) when the IICE0 bit is 0.

The PM6 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 13-12. Format of Port Mode Register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

13.4 I²C Bus Mode Functions

13.4.1 Pin configuration

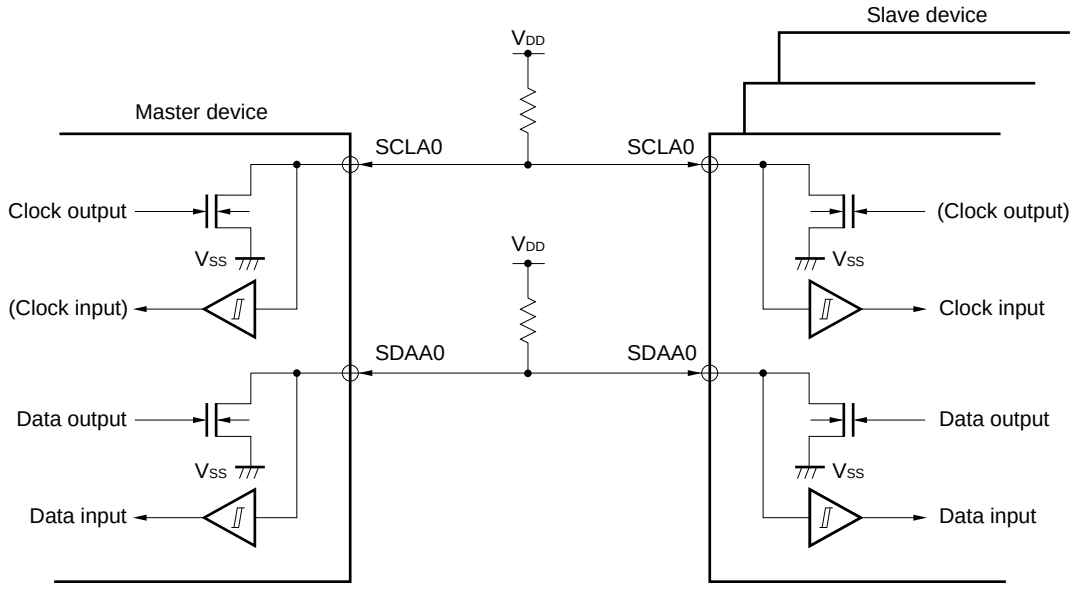
The serial clock pin (SCLA0) and the serial data bus pin (SDAA0) are configured as follows.

(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 13-13. Pin Configuration Diagram



13.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{MCK}}{\text{IICWL0} + \text{IICWH0} + f_{MCK}(t_R + t_F)}$$

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

$$\text{IICWL0} = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$$
$$\text{IICWH0} = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the normal mode

$$\text{IICWL0} = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$$
$$\text{IICWH0} = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the fast mode plus

$$\text{IICWL0} = \frac{0.50}{\text{Transfer clock}} \times f_{MCK}$$
$$\text{IICWH0} = \left(\frac{0.50}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

(2) Setting IICWL0 and IICWH0 registers on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$\text{IICWL0} = 1.3\ \mu s \times f_{MCK}$$
$$\text{IICWH0} = (1.2\ \mu s - t_R - t_F) \times f_{MCK}$$

- When the normal mode

$$\text{IICWL0} = 4.7\ \mu s \times f_{MCK}$$
$$\text{IICWH0} = (5.3\ \mu s - t_R - t_F) \times f_{MCK}$$

- When the fast mode plus

$$\text{IICWL0} = 0.50\ \mu s \times f_{MCK}$$
$$\text{IICWH0} = (0.50\ \mu s - t_R - t_F) \times f_{MCK}$$

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

Cautions 1. The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (max.). Set bit 0 (PRS0) of the IICA control register 01 (IICCTL01) to "1" only when the f_{CLK} exceeds 20 MHz.

2. 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.)
Fast mode plus: f_{CLK} = 10 MHz (MIN.)
Normal mode: f_{CLK} = 1 MHz (MIN.)

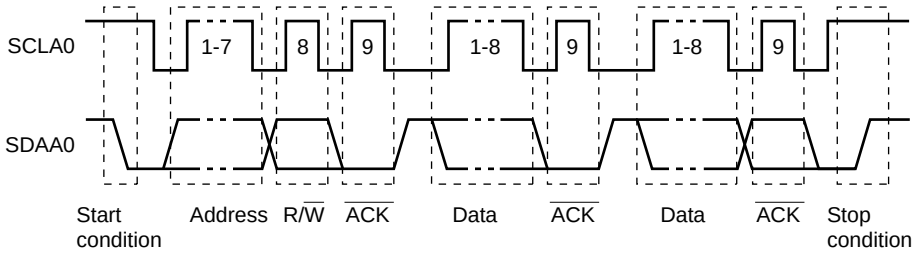
Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAA0 and SCLA0 signals separately, because they differ depending on the pull-up resistance and wire load.

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_{MCK}: IICA operation 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 13-14 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I²C bus's serial data bus.

Figure 13-14. I²C Bus Serial Data Transfer Timing



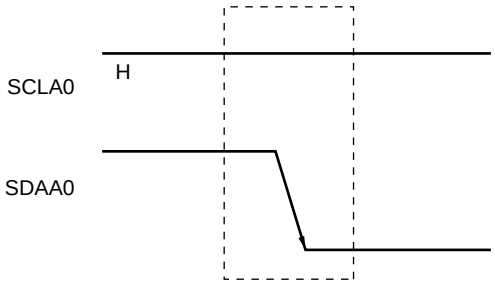
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

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 13-15. 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).

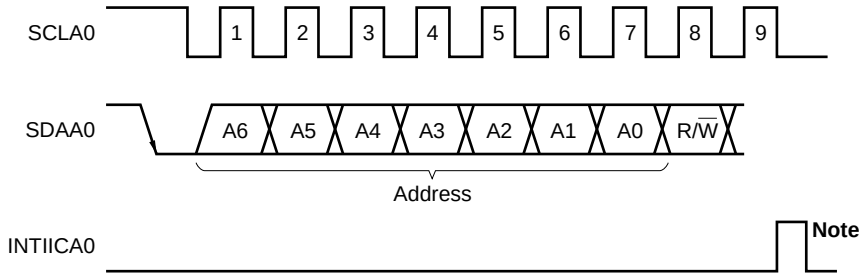
13.5.2 Addresses

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 13-16. 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.

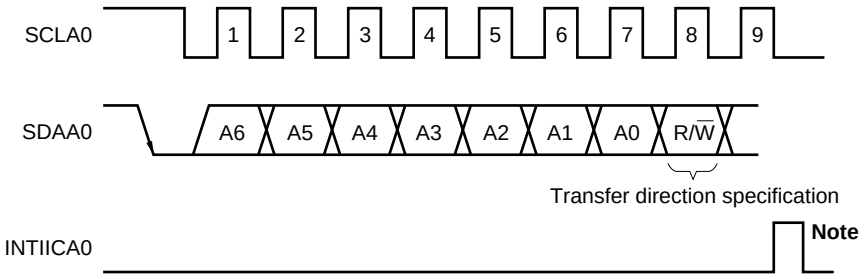
The slave address is assigned to the higher 7 bits of the IICA0 register.

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 13-17. 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 ($\overline{\text{ACK}}$)

$\overline{\text{ACK}}$ is used to check the status of serial data at the transmission and reception sides.

The reception side returns $\overline{\text{ACK}}$ each time it has received 8-bit data.

The transmission side usually receives $\overline{\text{ACK}}$ after transmitting 8-bit data. When $\overline{\text{ACK}}$ is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether $\overline{\text{ACK}}$ 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 $\overline{\text{ACK}}$ and instead generates a stop condition. If a slave does not return $\overline{\text{ACK}}$ after receiving data, the master outputs a stop condition or restart condition and stops transmission. If $\overline{\text{ACK}}$ 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.

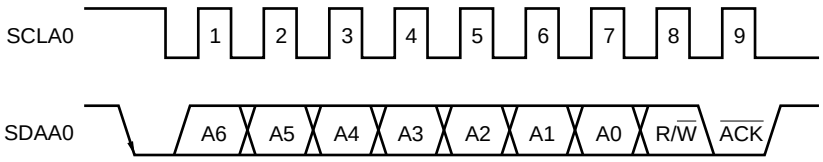
To generate $\overline{\text{ACK}}$, the reception side makes the SDAA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{\text{ACK}}$ 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 $\overline{\text{ACK}}$ 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 13-18. $\overline{\text{ACK}}$



When the local address is received, $\overline{\text{ACK}}$ is automatically generated, regardless of the value of the ACKE0 bit. When an address other than that of the local address is received, $\overline{\text{ACK}}$ is not generated (NACK).

When an extension code is received, $\overline{\text{ACK}}$ is generated if the ACKE0 bit is set to 1 in advance.

How $\overline{\text{ACK}}$ is generated when data is received differs as follows depending on the setting of the clock stretch timing.

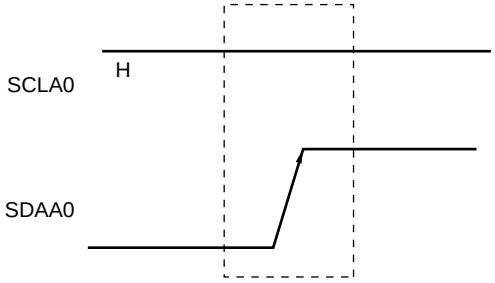
- When 8-clock clock stretch state is selected (bit 3 (WTIM0) of IICCTL00 register = 0):
  By setting the ACKE0 bit to 1 before releasing the clock stretch state, $\overline{\text{ACK}}$ is generated at the falling edge of the eighth clock of the SCLA0 pin.
- When 9-clock clock stretch state is selected (bit 3 (WTIM0) of IICCTL00 register = 1):
  $\overline{\text{ACK}}$ 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 13-19. 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 13-20. Clock stretch (1/2)

(1) When master device has a nine-clock clock stretch and slave device has an eight-clock clock stretch (master transmits, slave receives, and ACKE0 = 1)

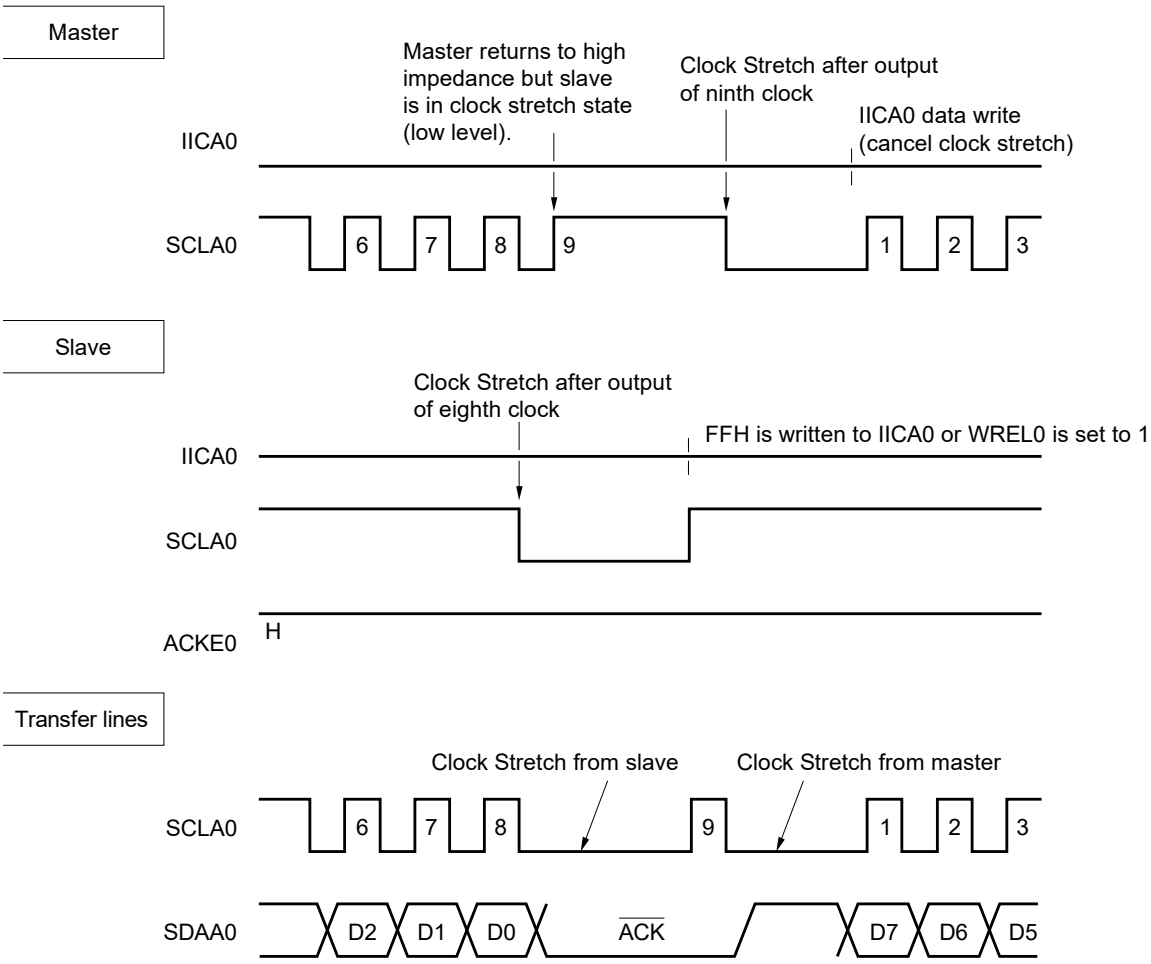
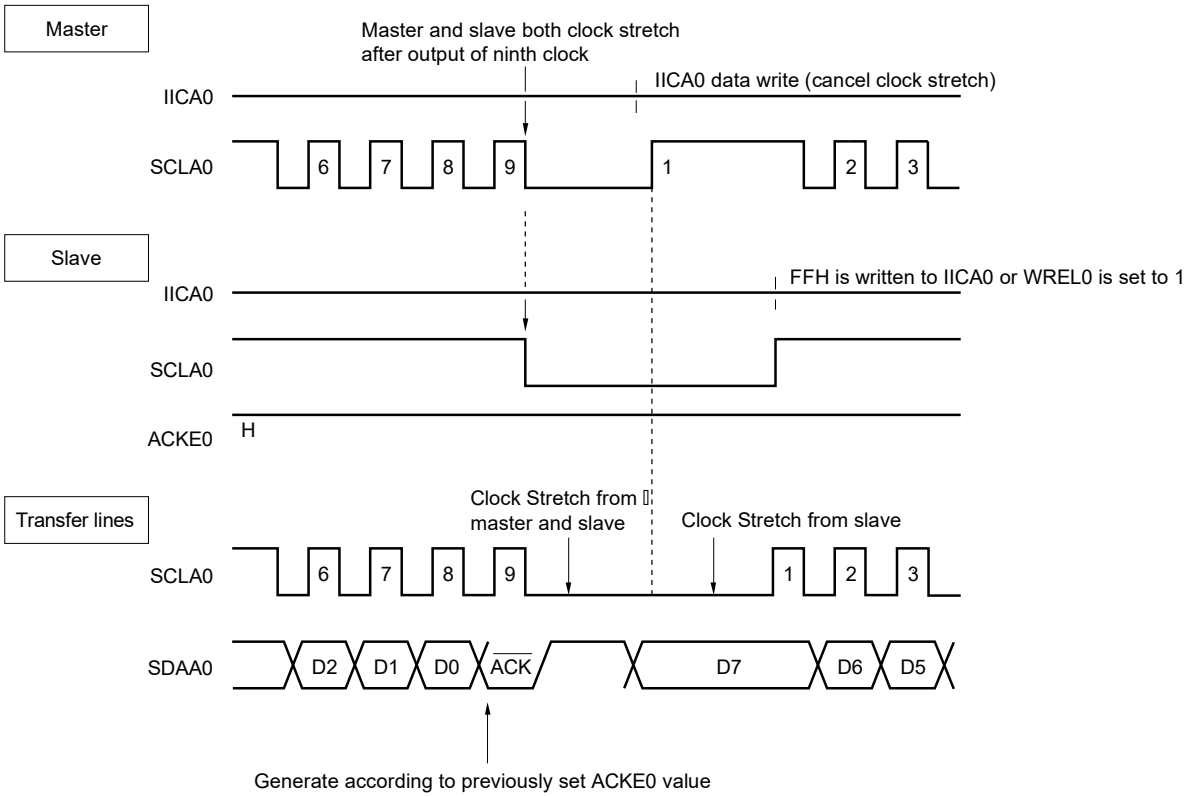


Figure 13-20. Clock stretch (2/2)

(2) When master and slave devices both have a nine-clock clock stretch (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 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, $\overline{ACK}$ 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 an 8-clock clock stretch has been selected (WTIM0 = 0), the presence/absence of $\overline{ACK}$ 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 13-21. Arbitration Timing Example

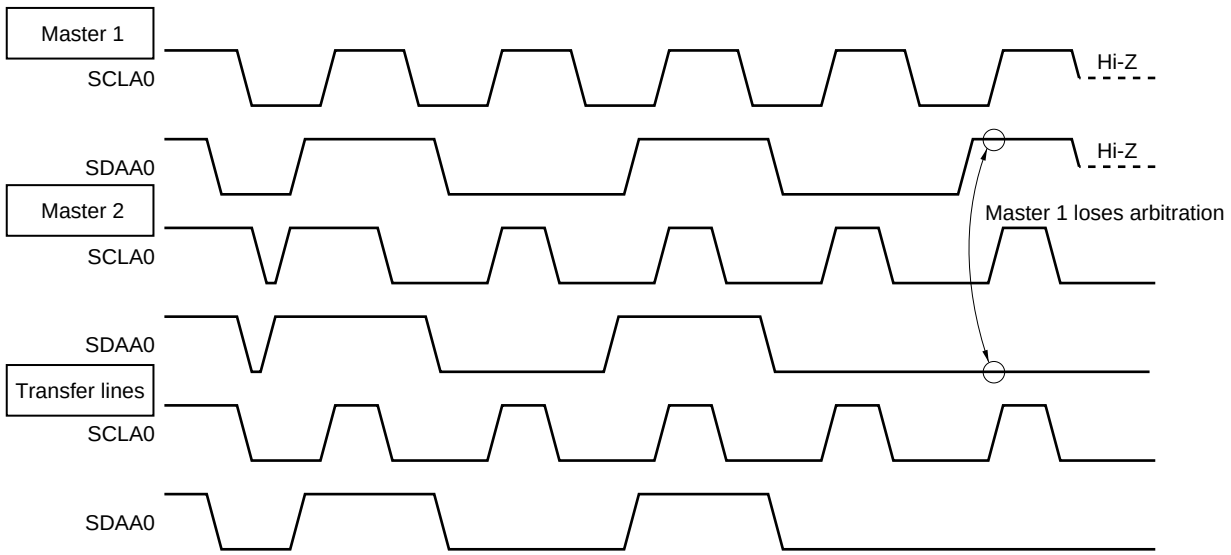


Table 13-4. Status During Arbitration and Interrupt Request Generation Timing

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 $\overline{\text{ACK}}$ 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 13-22 shows the flow for setting WUP0 = 1 and Figure 13-23 shows the flow for setting WUP0 = 0 upon an address match.

Figure 13-22. Flow When Setting WUP0 = 1

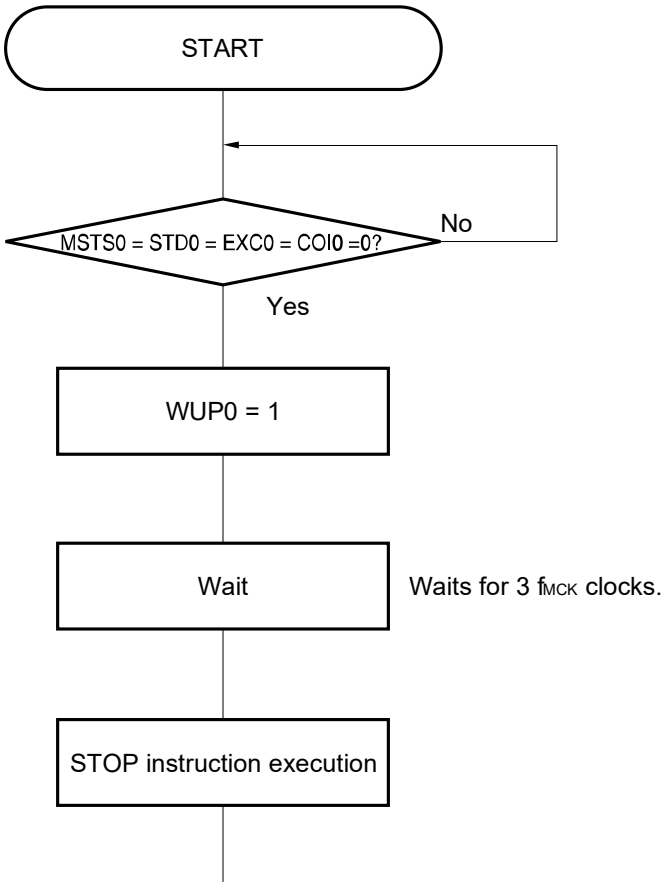
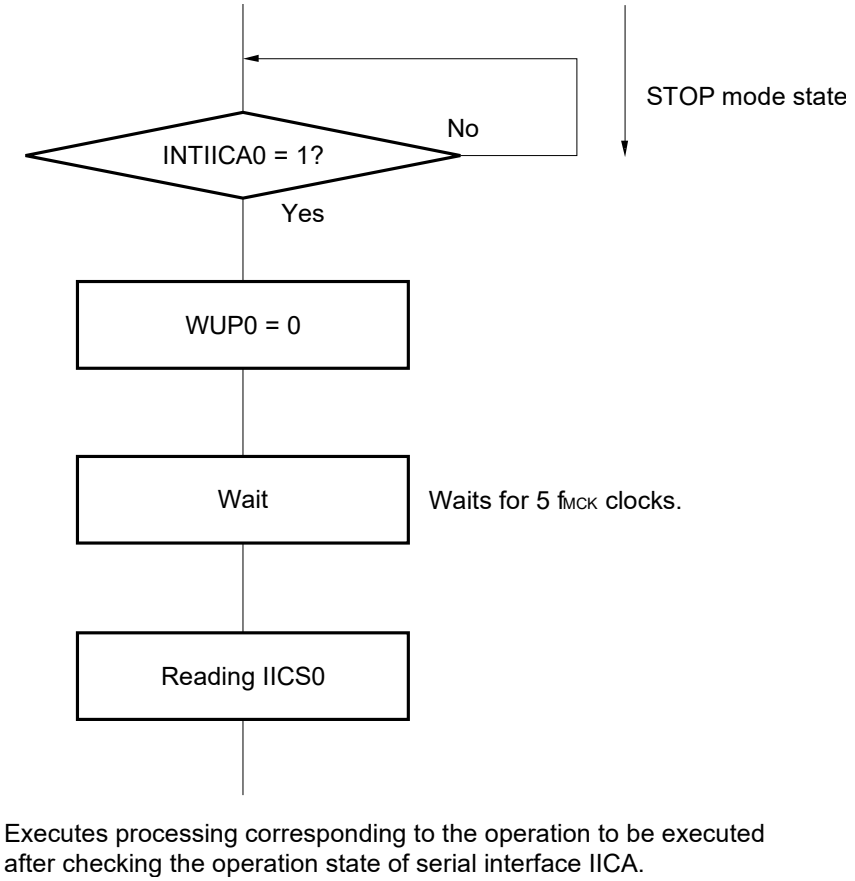


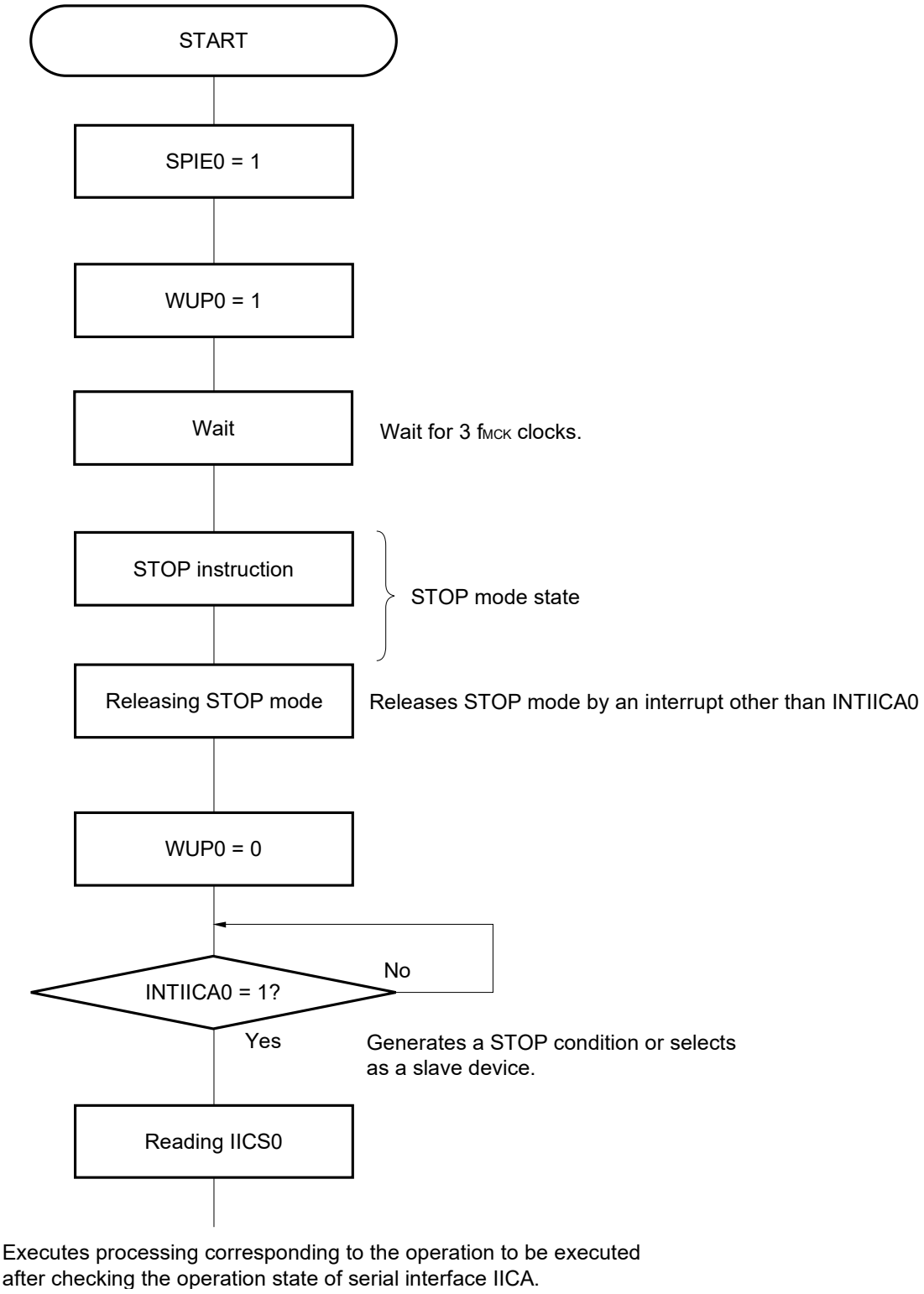
Figure 13-23. Flow When Setting WUP0 = 0 upon Address Match (Including Extension Code Reception)



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.

- When operating next IIC communication as master: Flow shown in Figure 13-24
- When operating next IIC communication as slave:
  When restored by INTIICA0 interrupt: Same as the flow in Figure 13-23
  When restored by other than INTIICA0 interrupt: Wait for INTIICA0 interrupt with WUP0 left set to 1.

Figure 13-24. 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 (IICRSV0) 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 ($\overline{ACK}$ 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

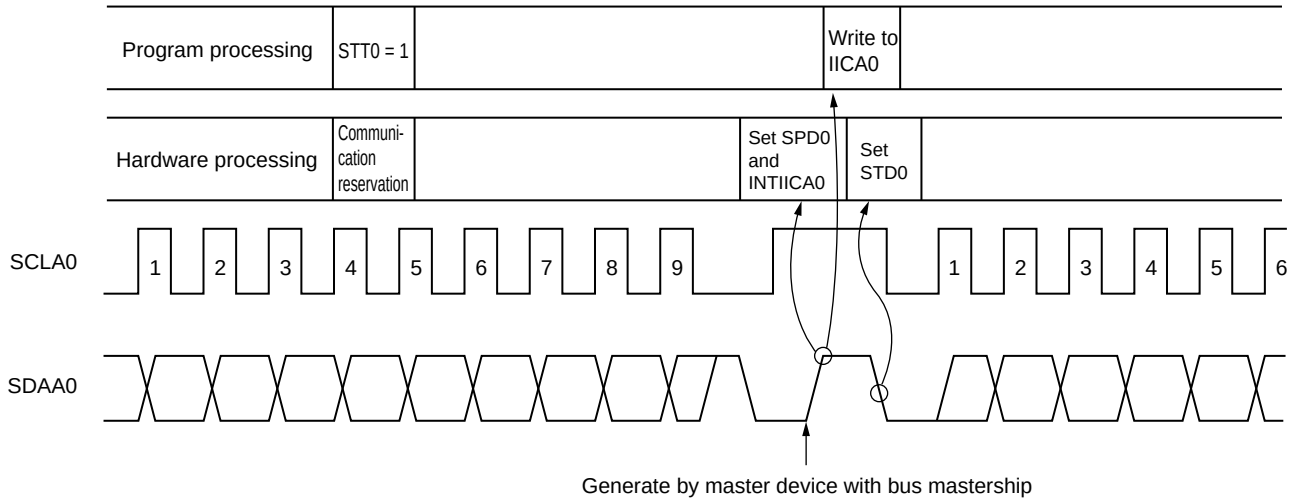
Check whether the communication reservation operates or not by using the MSTS0 bit (bit 7 of the IICA status register 0 (IICS0)) after the STT0 bit is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT0 = 1 to checking the MSTS0 flag: (IICWL0 setting value + IICWH0 setting value + 4)/f_{MCK} + $t_F \times 2$

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_{MCK}: IICA operation clock frequency

Figure 13-25 shows the communication reservation timing.

Figure 13-25. 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 13-26. 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 13-26. Timing for Accepting Communication Reservations

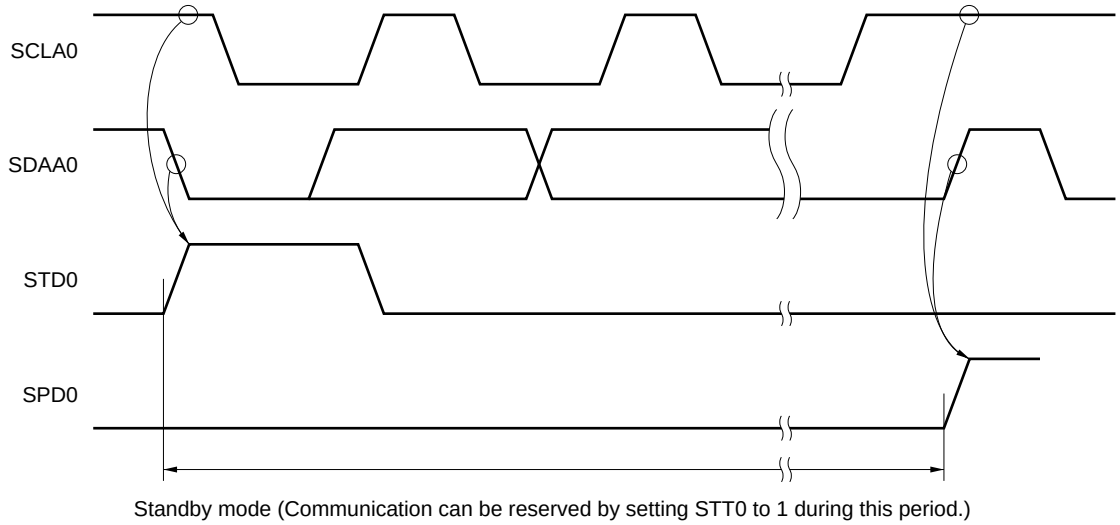
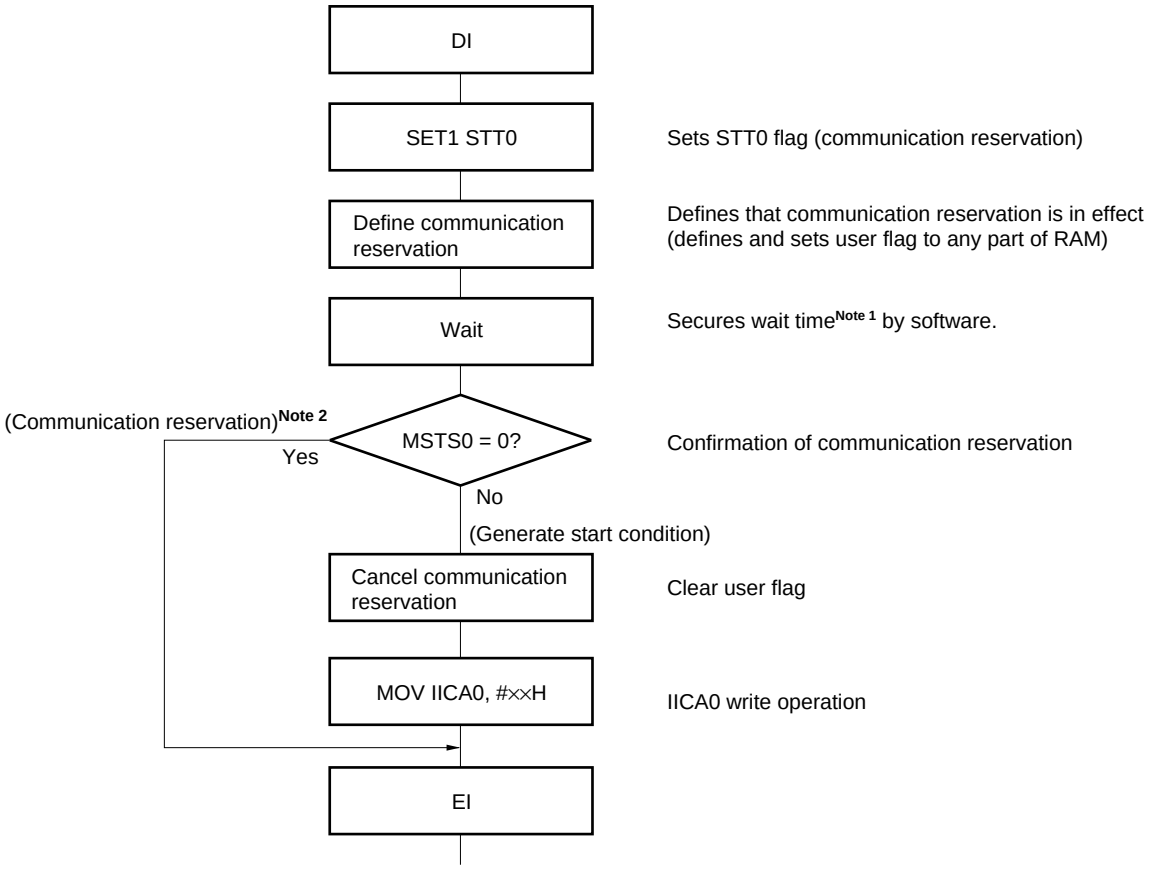


Figure 13-27 shows the communication reservation protocol.

Figure 13-27. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.

(IICWL0 setting value + IICWH0 setting value + 4)/f_{MCK} + $t_F \times 2$

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_{MCK}: IICA operation clock frequency

(2) When communication reservation function is disabled (bit 0 (IICRSV0) 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 ($\overline{ACK}$ 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 STCF0 (bit 7 of the IICF0 register). It takes up to 5 f_{MCK} clocks until the STCF0 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, $\overline{\text{ACK}}$ 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 $\overline{\text{ACK}}$ is returned (4 to 72 f_{MCK} 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 IICTL00 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/G1C 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/G1C 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/G1C 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

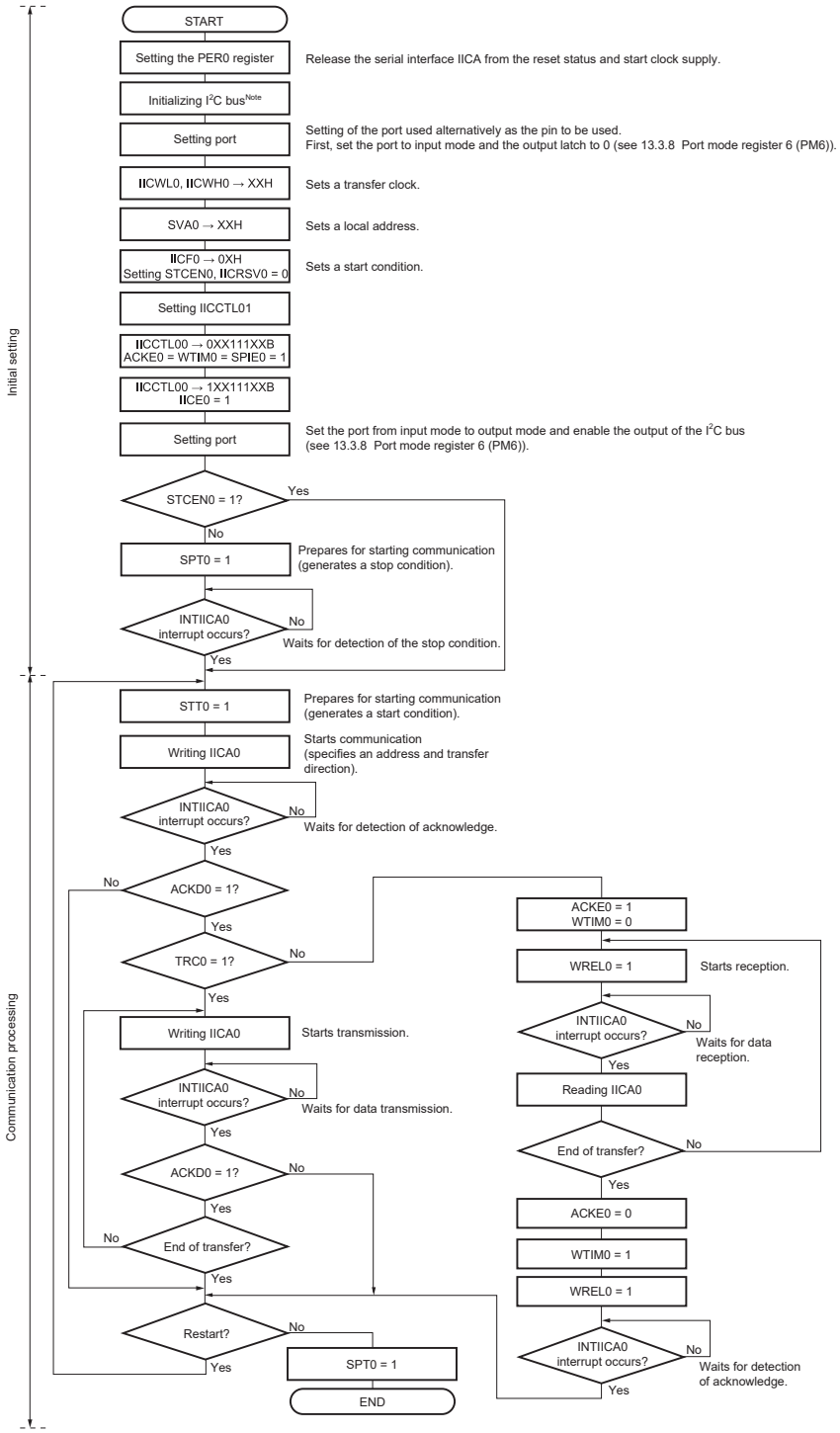
An example of when the RL78/G1C is used as the I^2C bus slave is shown below.

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 13-28. Master Operation in Single-Master System

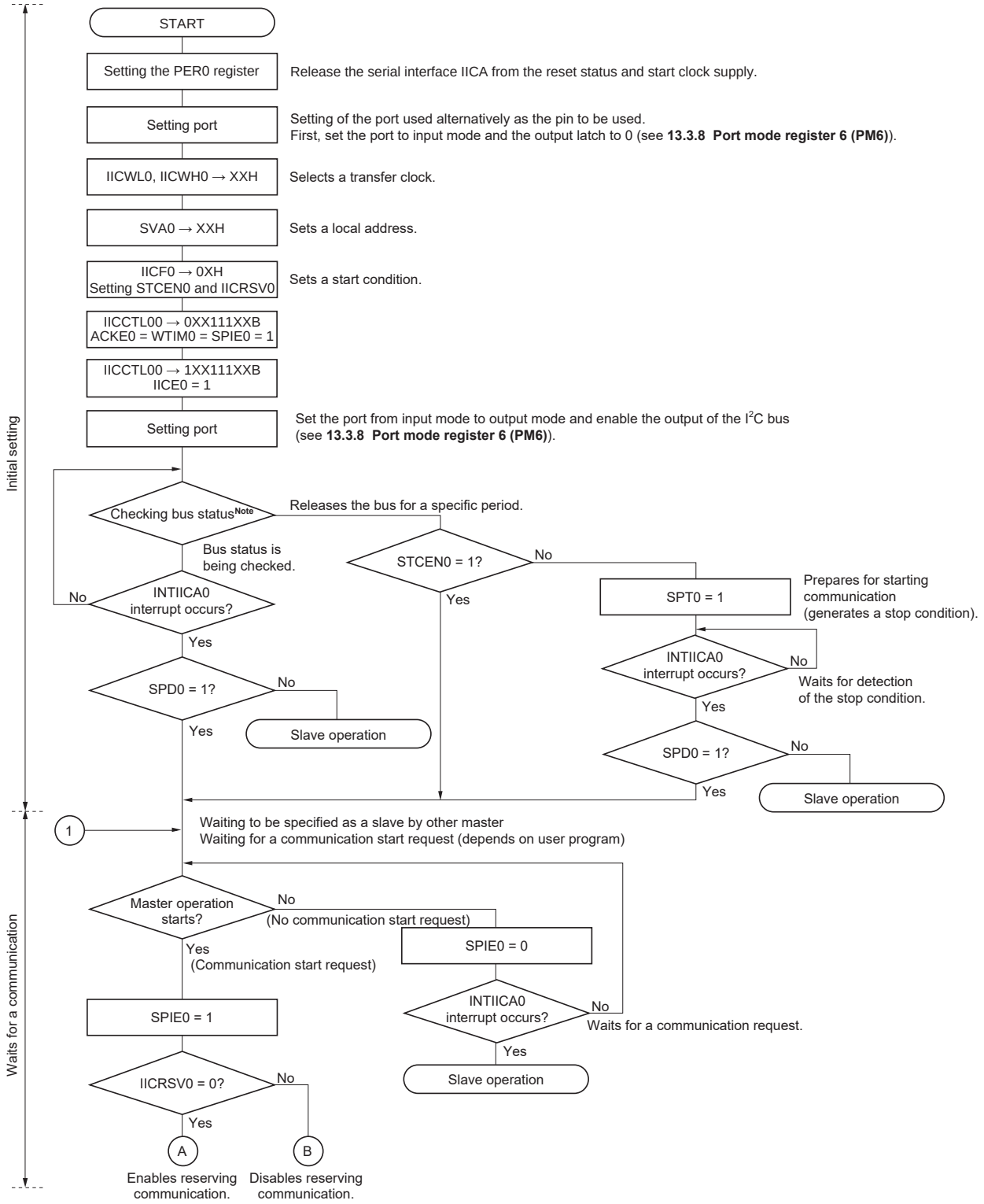


Note Release (SCLA0 and SDAA0 pins = high level) the I²C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAA0 pin, for example, set the SCLA0 pin in the output port mode, and output a clock pulse from the output port until the 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.

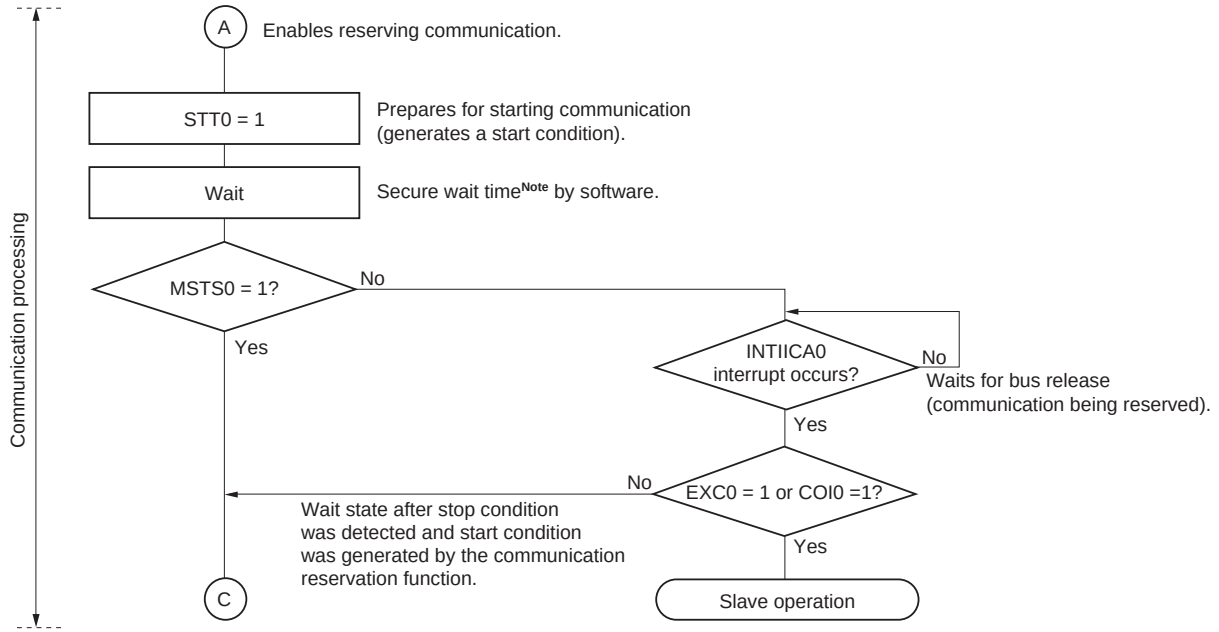
(2) Master operation in multi-master system

Figure 13-29. Master Operation in Multi-Master System (1/3)



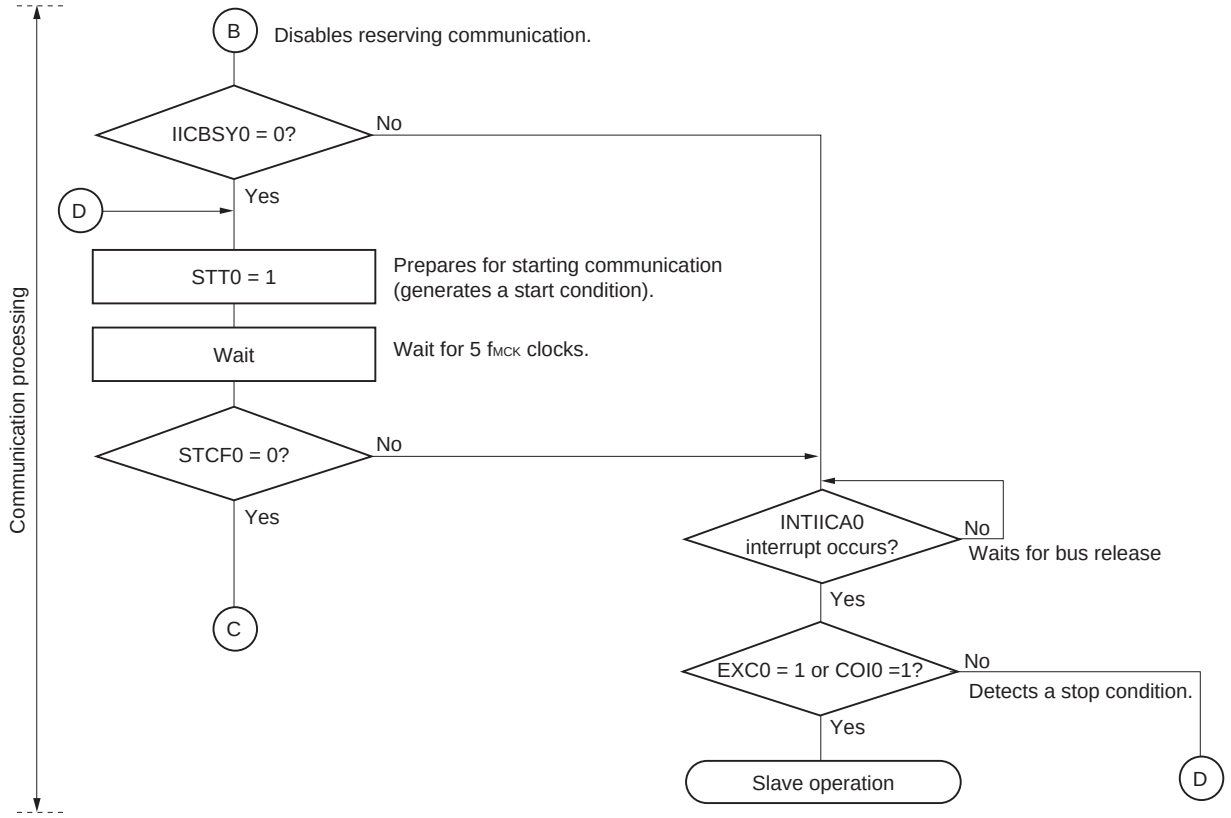
Note Confirm that the bus is released (CLD0 bit = 1, DAD0 bit = 1) for a specific period (for example, for a period of one frame). If the SDAA0 pin is constantly at low level, decide whether to release the I²C bus (SCLA0 and SDAA0 pins = high level) in conformance with the specifications of the product that is communicating.

Figure 13-29. Master Operation in Multi-Master System (2/3)



Note The wait time is calculated as follows.

(IICWL0 setting value + IICWH0 setting value + 4)/f_{MCK} + $t_F \times 2$



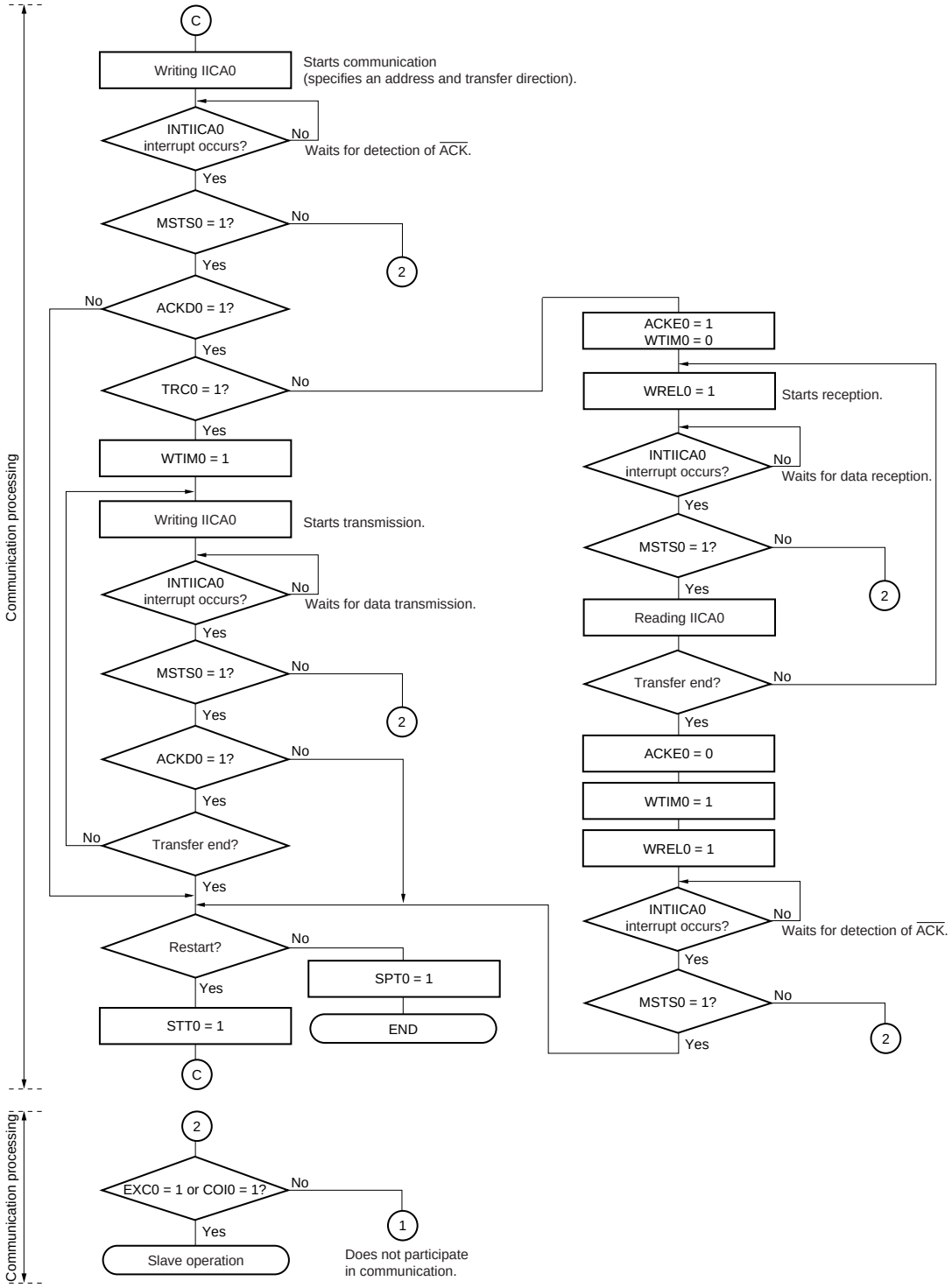
Remark IICWL0: IICA low-level width setting register 0

IICWH0: IICA high-level width setting register 0

t_F: SDAA0 and SCLA0 signal falling times

f_{MCK}: IICA operation clock frequency

Figure 13-29. 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.

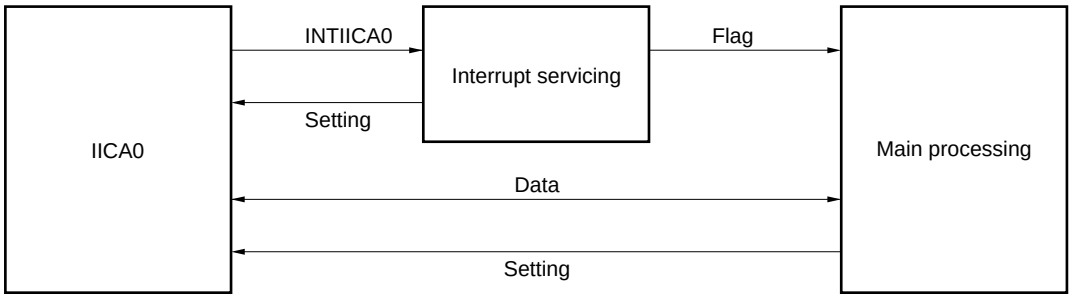
3. To use the device as a slave in a multi-master system, check the status by using the IICA status register 0 (IICS0) and IICA flag register 0 (IICF0) each time interrupt INTIICA0 has occurred, and determine the processing to be performed next.

(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 $\overline{\text{ACK}}$ 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.

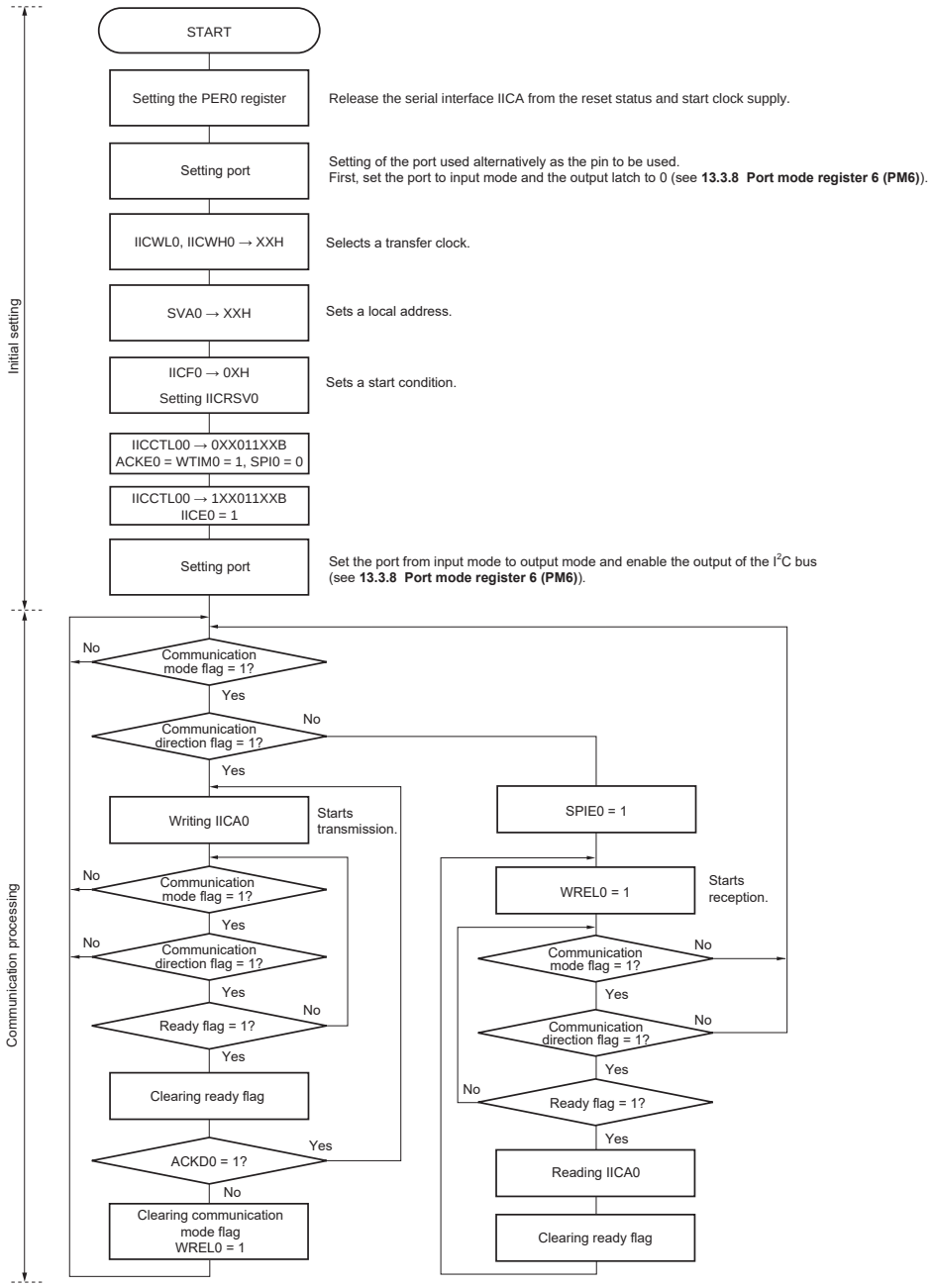
The main processing of the slave operation is explained next.

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 $\overline{\text{ACK}}$. If $\overline{\text{ACK}}$ is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, $\overline{\text{ACK}}$ 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.

Figure 13-30. Slave Operation Flowchart (1)



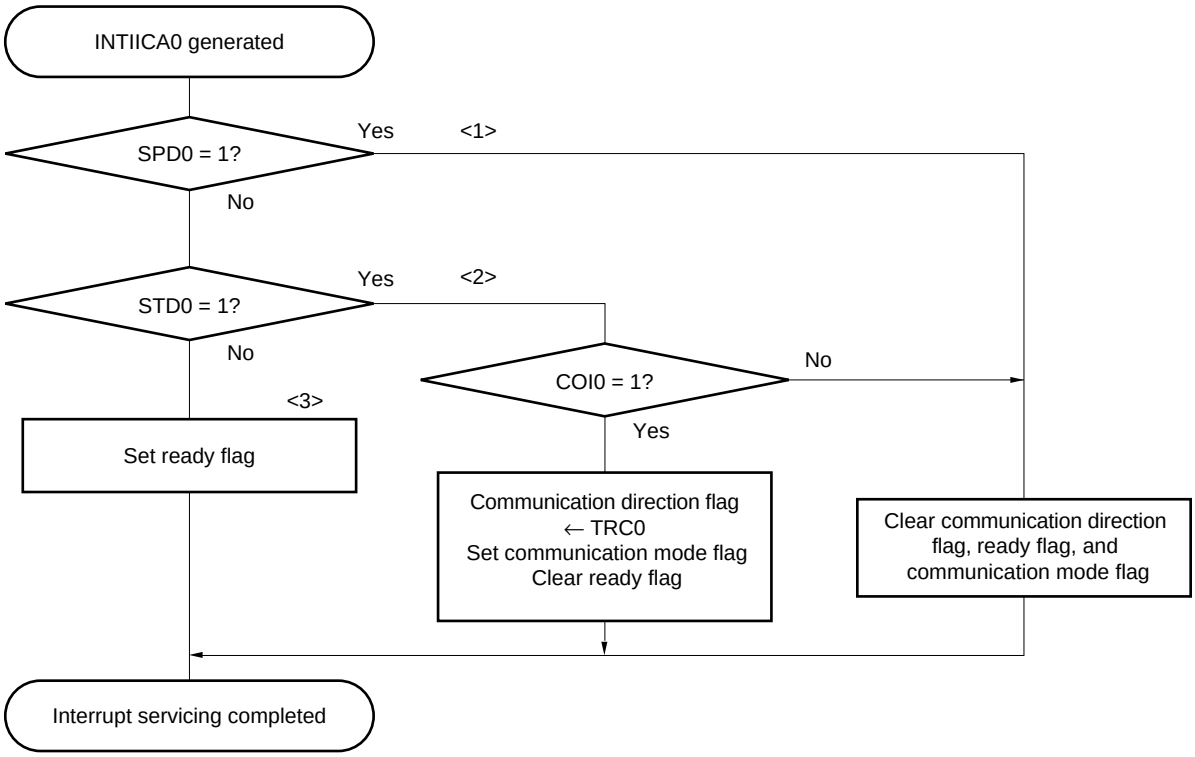
Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

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 I²C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in **Figure 13-31 Slave Operation Flowchart (2)**.

Figure 13-31. 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.

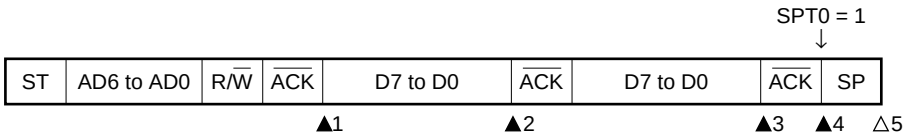
Remark

ST:	Start condition
AD6 to AD0:	Address
R/$\overline{W}$:	Transfer direction specification
$\overline{ACK}$:	Acknowledge
D7 to D0:	Data
SP:	Stop condition

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM0 = 0

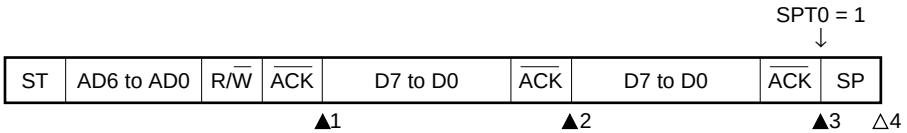


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B
▲3: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

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

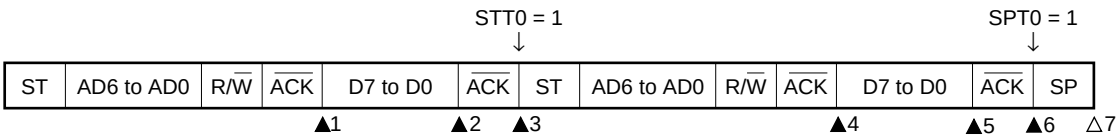


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B
▲3: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0

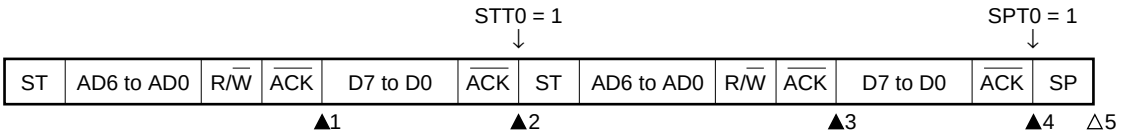


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note 1]
▲3: IICS0 = 1000××00B (Clears the WTIM0 bit to 0[Note 2], sets the STT0 bit to 1)
▲4: IICS0 = 1000×110B
▲5: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note 3]
▲6: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△7: IICS0 = 00000001B

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

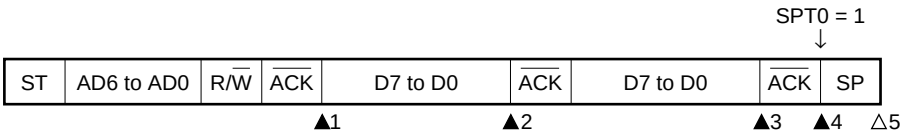


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
▲3: IICS0 = 1000×110B
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0

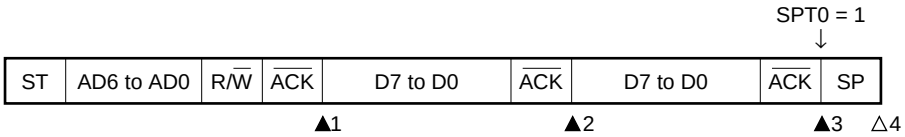


▲1: IICS0 = 1010×110B
▲2: IICS0 = 1010×000B
▲3: IICS0 = 1010×000B (Sets the WTIM0 bit to 1)[Note]
▲4: IICS0 = 1010××00B (Sets the SPT0 bit to 1)
△5: IICS0 = 00000001B

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 = 00001001B

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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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)

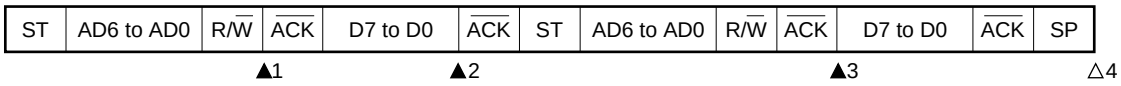
ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1	▲2		▲3			▲4	▲5	▲6	△7

▲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

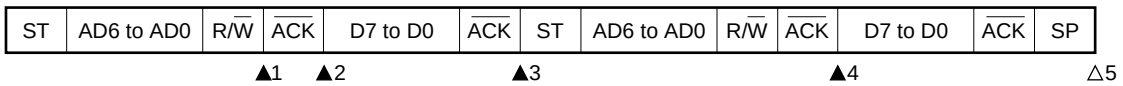
(i) When WTIM0 = 0 (after restart, does not match address (= not extension code))



▲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))



▲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/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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

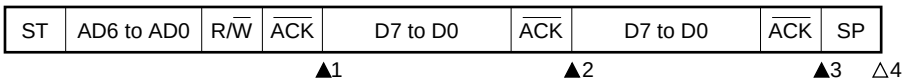
(i) When WTIM0 = 0

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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

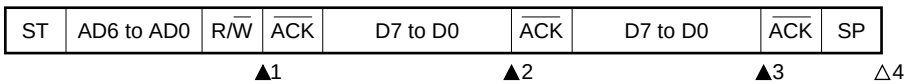


▲1: IICS0 = 0101×110B
▲2: IICS0 = 0001×100B
▲3: IICS0 = 0001××00B
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

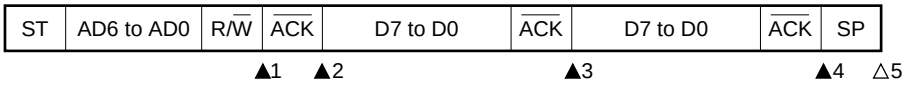
(i) When WTIM0 = 0



▲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



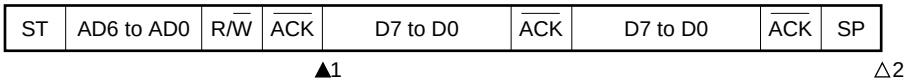
▲1: IICS0 = 0110×010B
▲2: IICS0 = 0010×110B
▲3: IICS0 = 0010×100B
▲4: IICS0 = 0010××00B
△5: IICS0 = 00000001B

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.

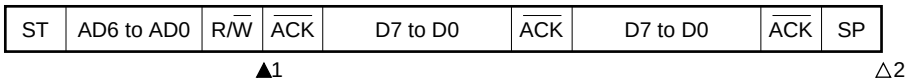
(a) When arbitration loss occurs during transmission of slave address data (when WTIM0 = 1)



▲1: IICS0 = 01000110B
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When arbitration loss occurs during transmission of extension code

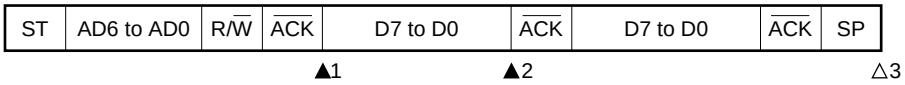


▲1: IICS0 = 0110×010B
Sets LREL0 = 1 by software
△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

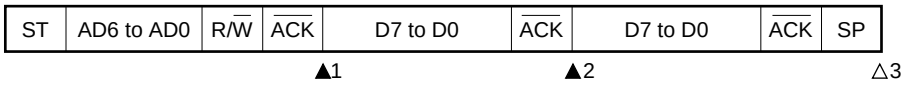
(i) When WTIM0 = 0



▲1: IICS0 = 10001110B
▲2: IICS0 = 01000000B
△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(ii) When WTIM0 = 1

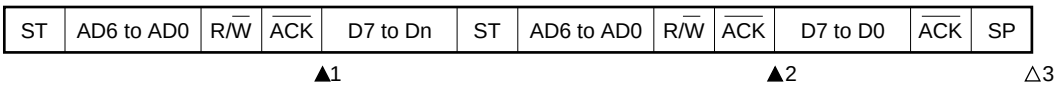


▲1: IICS0 = 10001110B
▲2: IICS0 = 01000100B
△3: IICS0 = 00000001B

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)



▲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/$\overline{W}$	$\overline{ACK}$	D7 to Dn	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	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

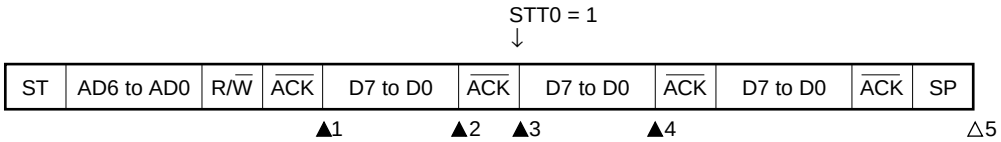
ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to Dn	SP
				▲1	△2

▲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

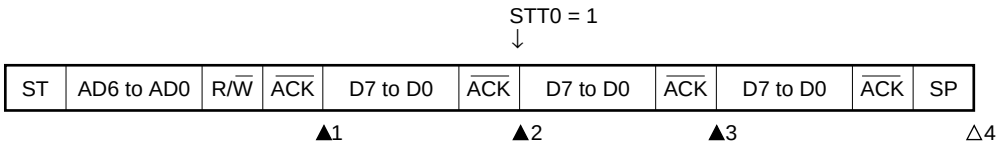
(i) When WTIM0 = 0



▲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

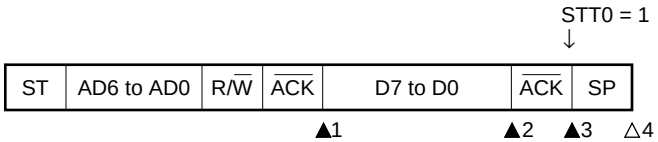


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000×100B (Sets the STT0 bit to 1)
▲3: IICS0 = 01000100B
△4: IICS0 = 00000001B

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

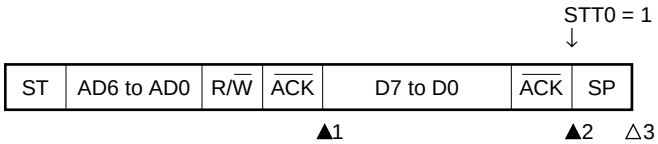
(i) When WTIM0 = 0



▲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

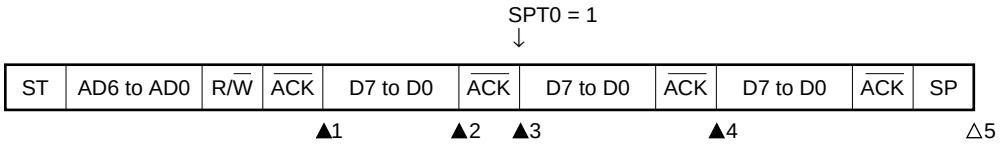


▲1: IICS0 = 1000×110B
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)
△3: IICS0 = 01000001B

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

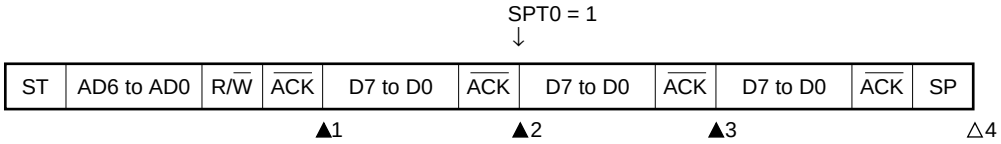
(i) When WTIM0 = 0



▲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.

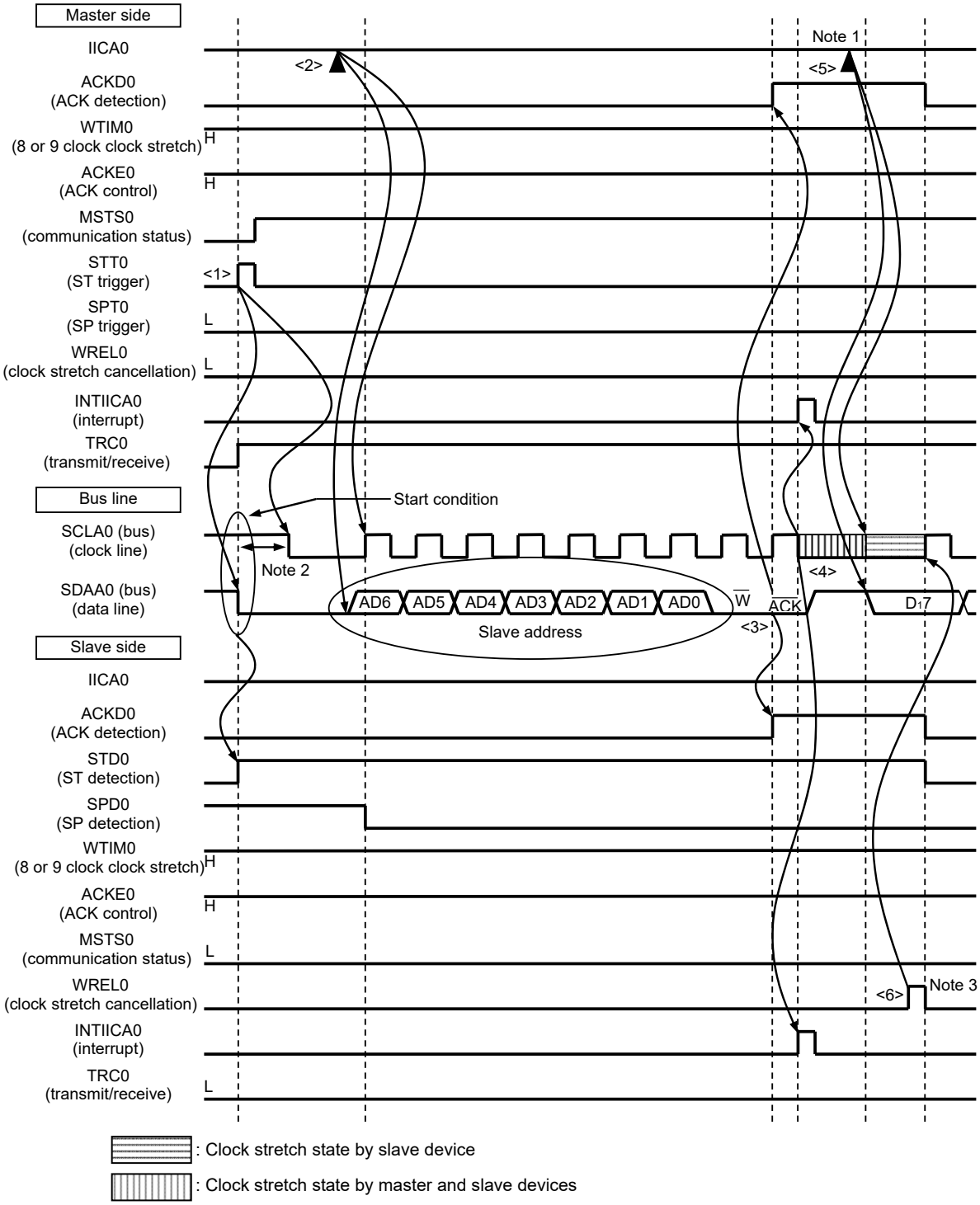
Figures 13-32 and 13-33 show timing charts of the data communication.

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 13-32. 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 13-32 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 + 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.

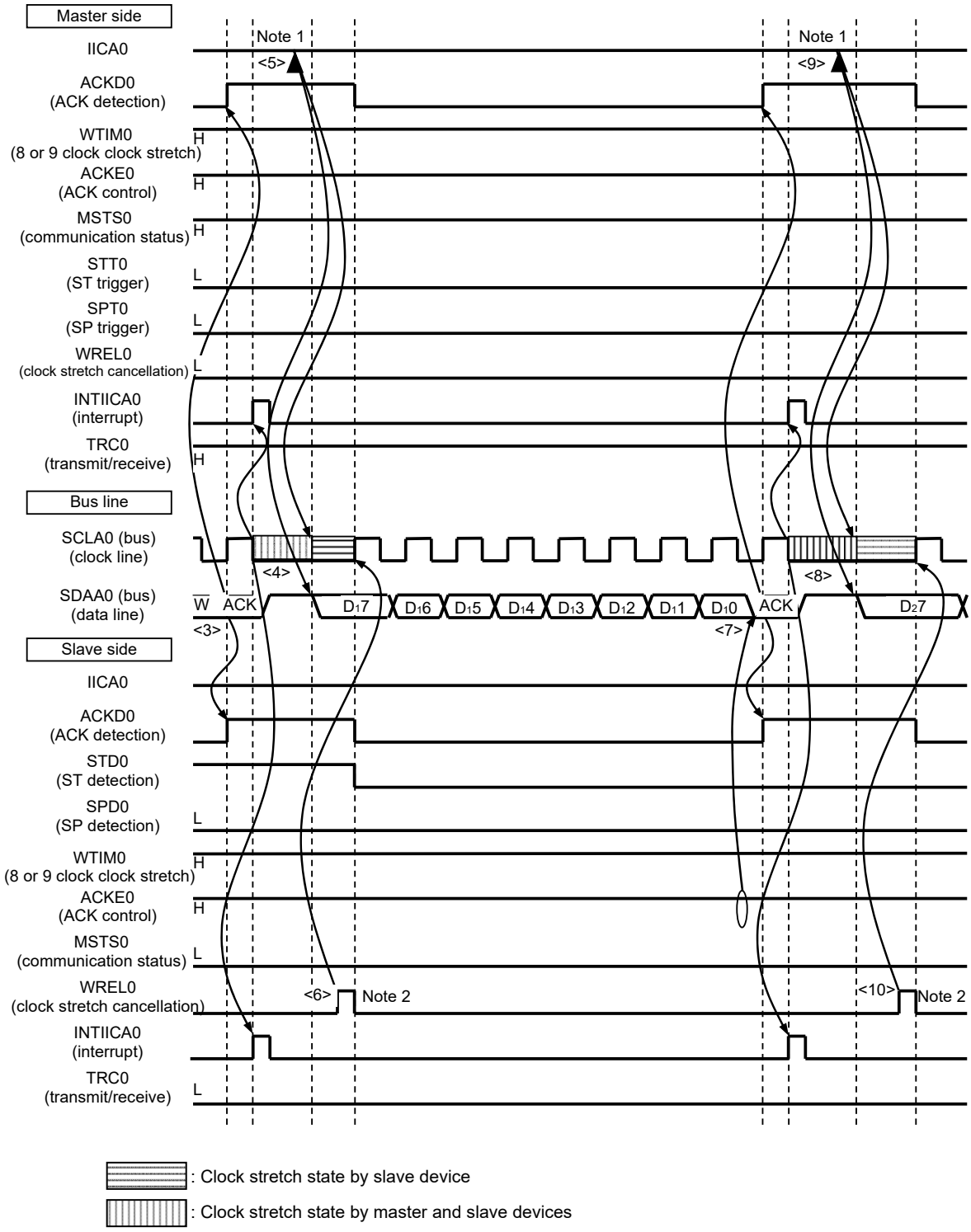
<6> If the slave device releases the clock stretch status (WREL0 = 1), the master device starts transferring data to the slave device.

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 13-32 represent 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 <6>, Figure 13-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 13-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 13-32. 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 13-32 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 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.

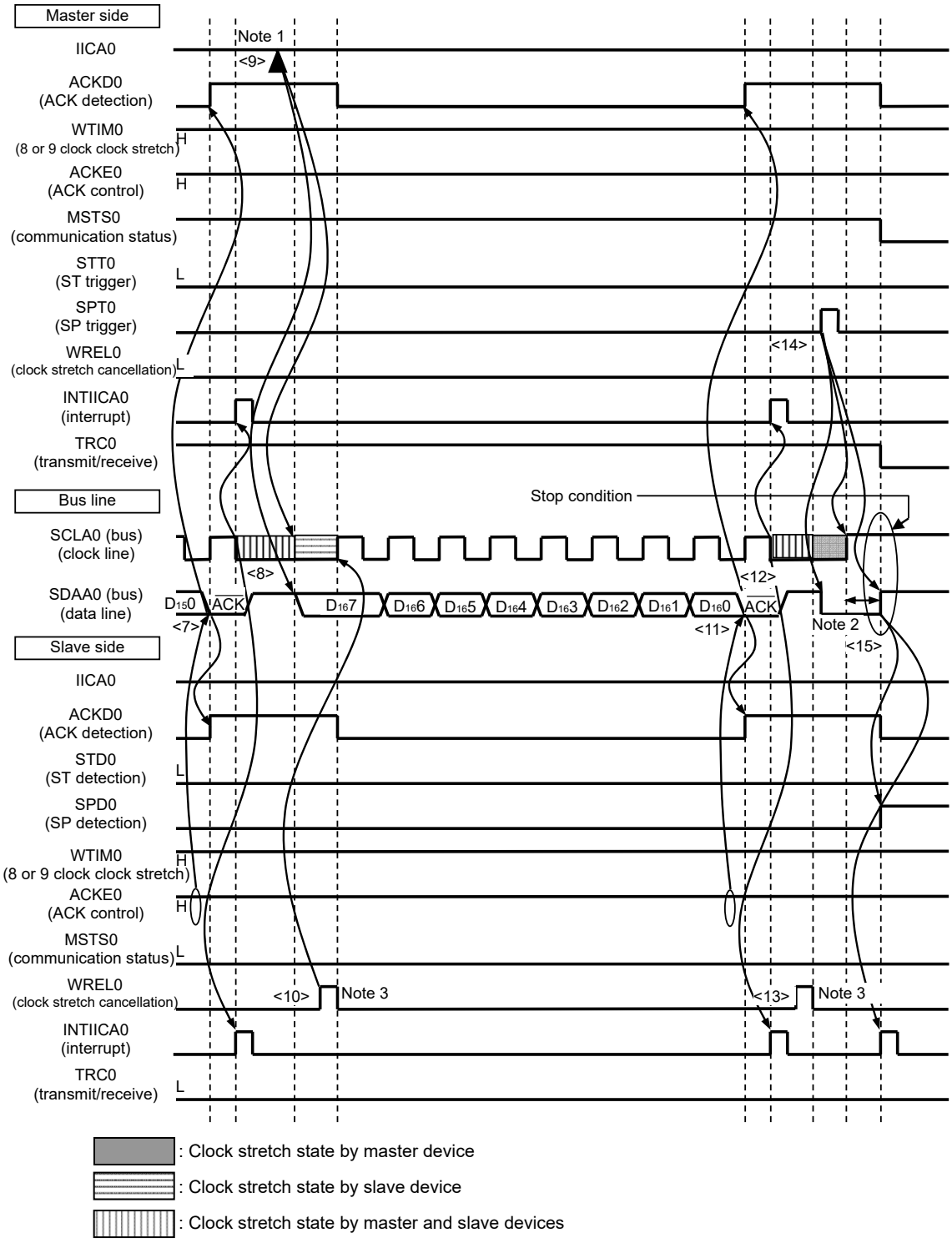
<10> The slave device reads the received data and releases the clock stretch status (WREL0 = 1). The master device then starts transferring data to the slave device.

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 13-32 represent 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 <6>, Figure 13-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 13-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 13-32. 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 13-32 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).

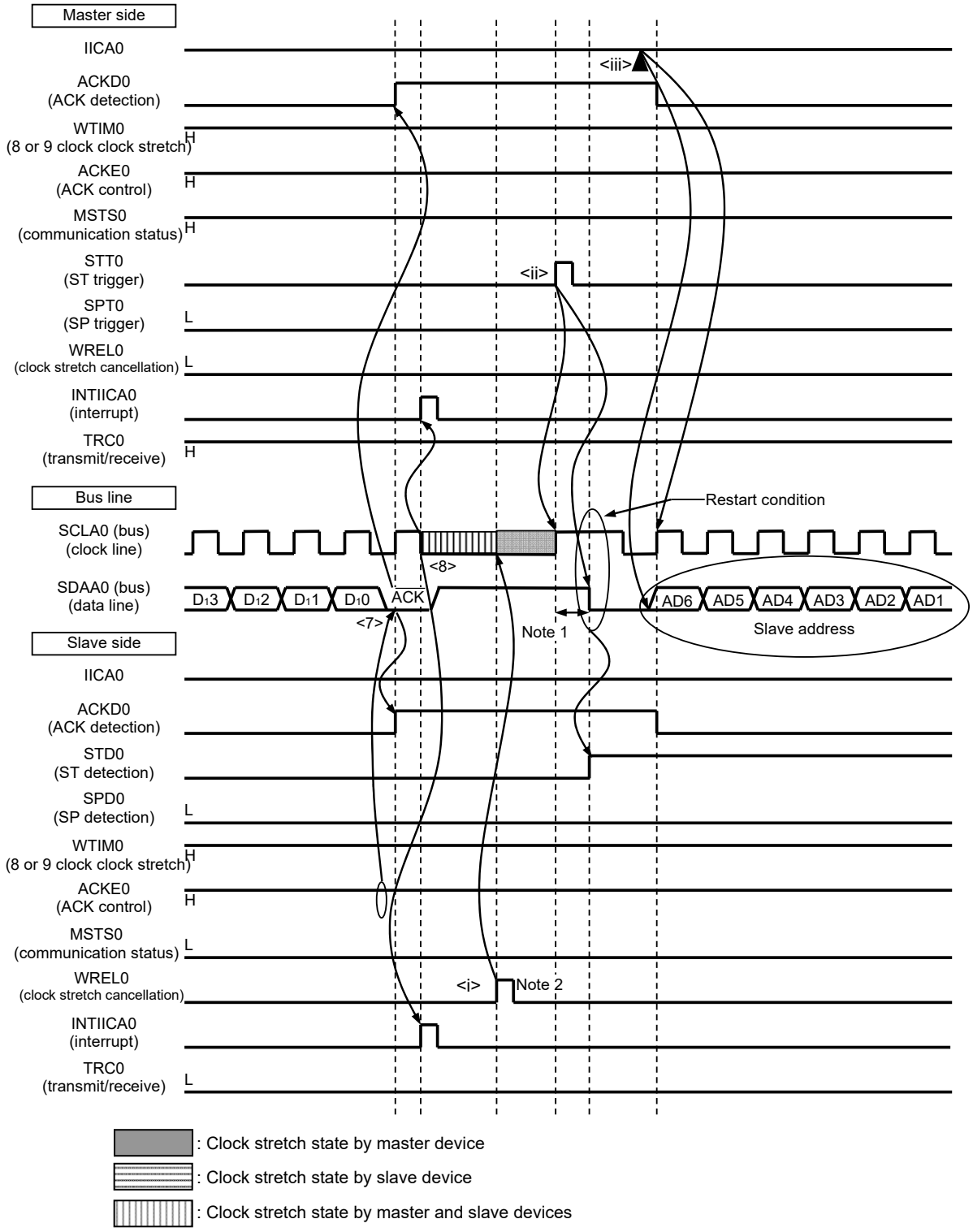
<14> By the master device setting a stop condition trigger (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the bus clock line is set (SCLA0 = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAA0 = 1), the stop condition is then generated (i.e. SCLA0 =1 changes 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 13-32 represent 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 <6>, Figure 13-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 13-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 13-32. 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 13-32 (4) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <iii>, 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).

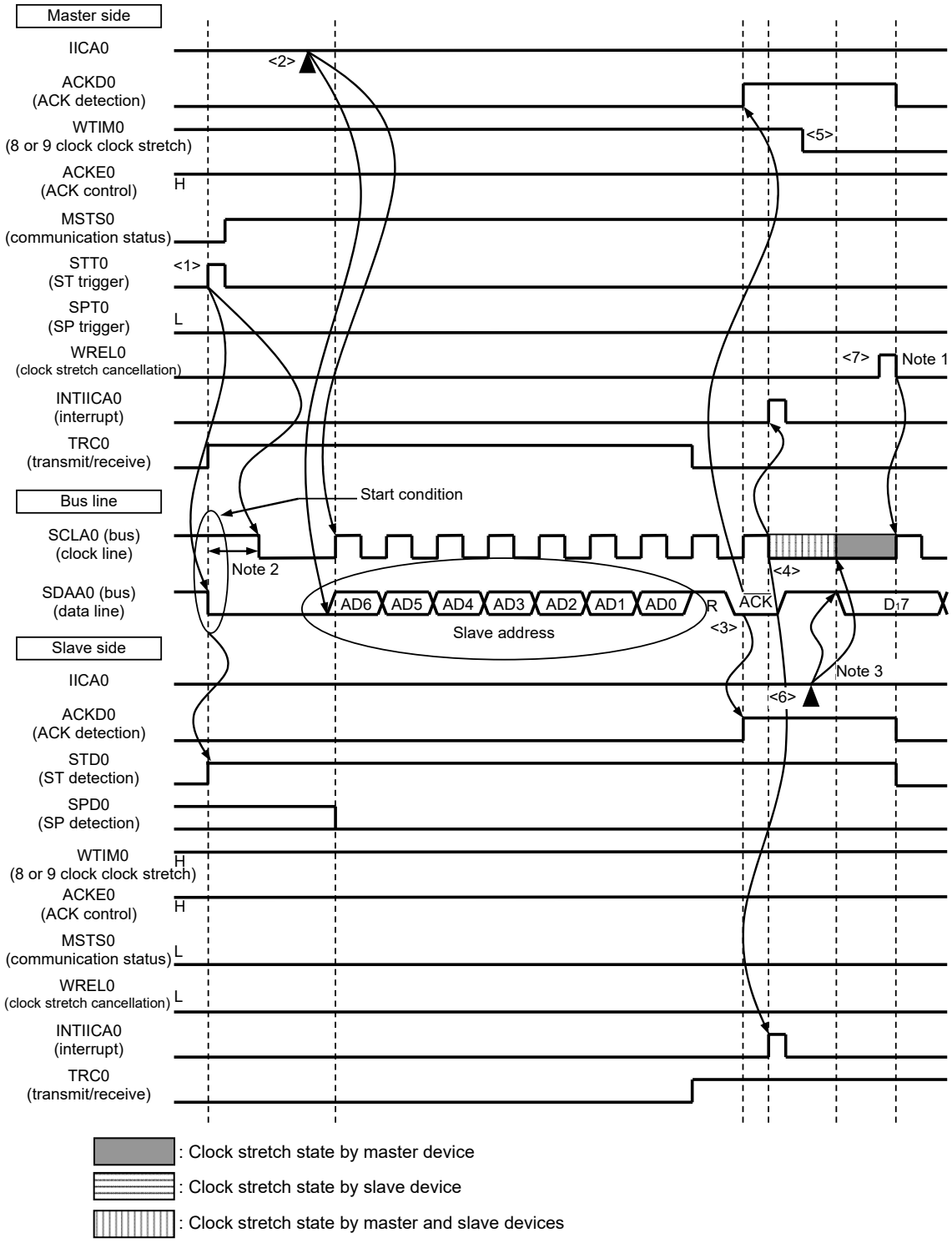
<i> The slave device reads the received data and releases the clock stretch status (WREL0 = 1).

<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 13-33. 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 13-33 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.

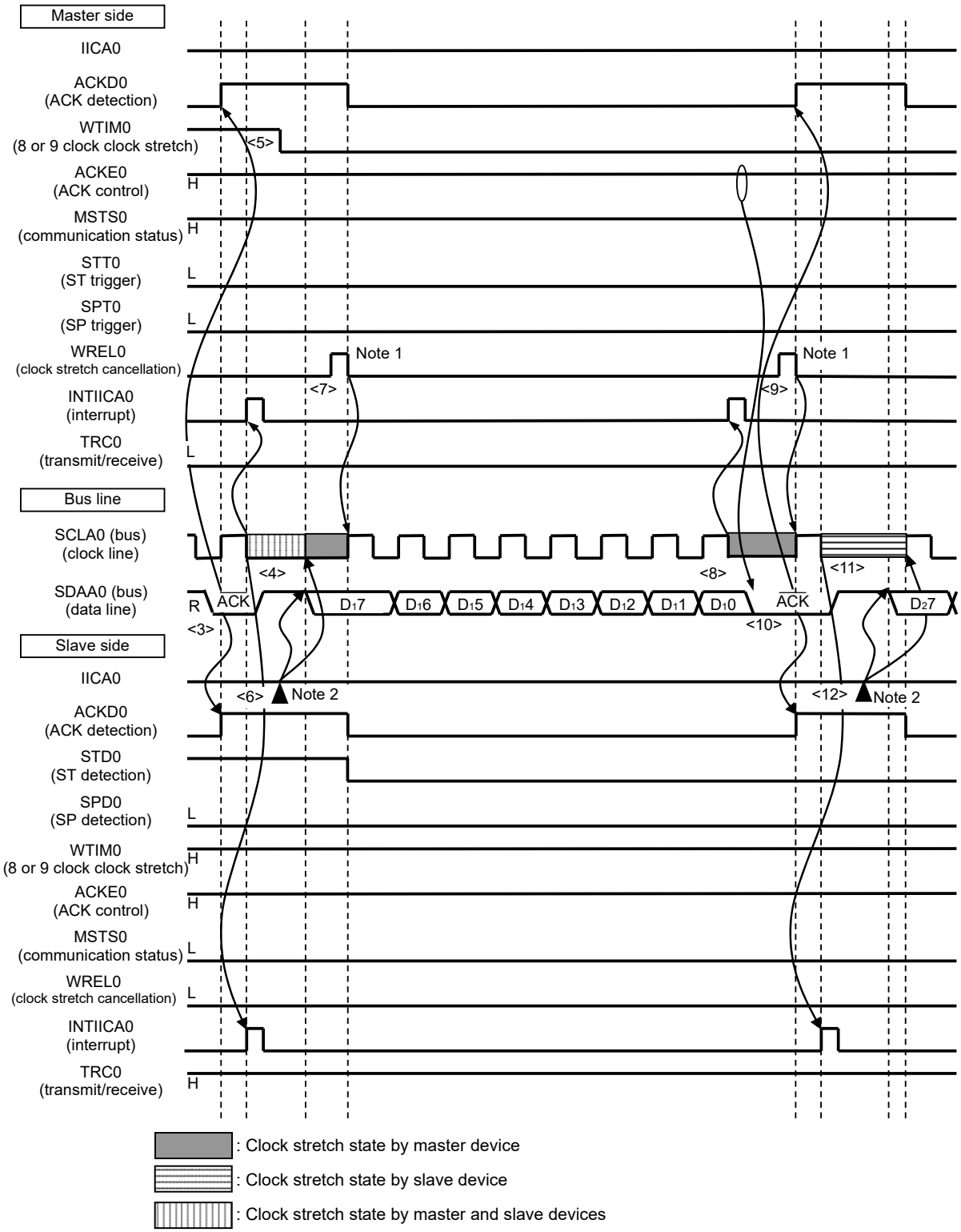
<7> The master device releases the clock stretch status (WREL0 = 1) and starts transferring data from the slave device to the master device.

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 13-33 represent the entire procedure for communicating data using the I^2C bus. Figure 13-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 13-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 13-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 13-33. 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 13-33 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).

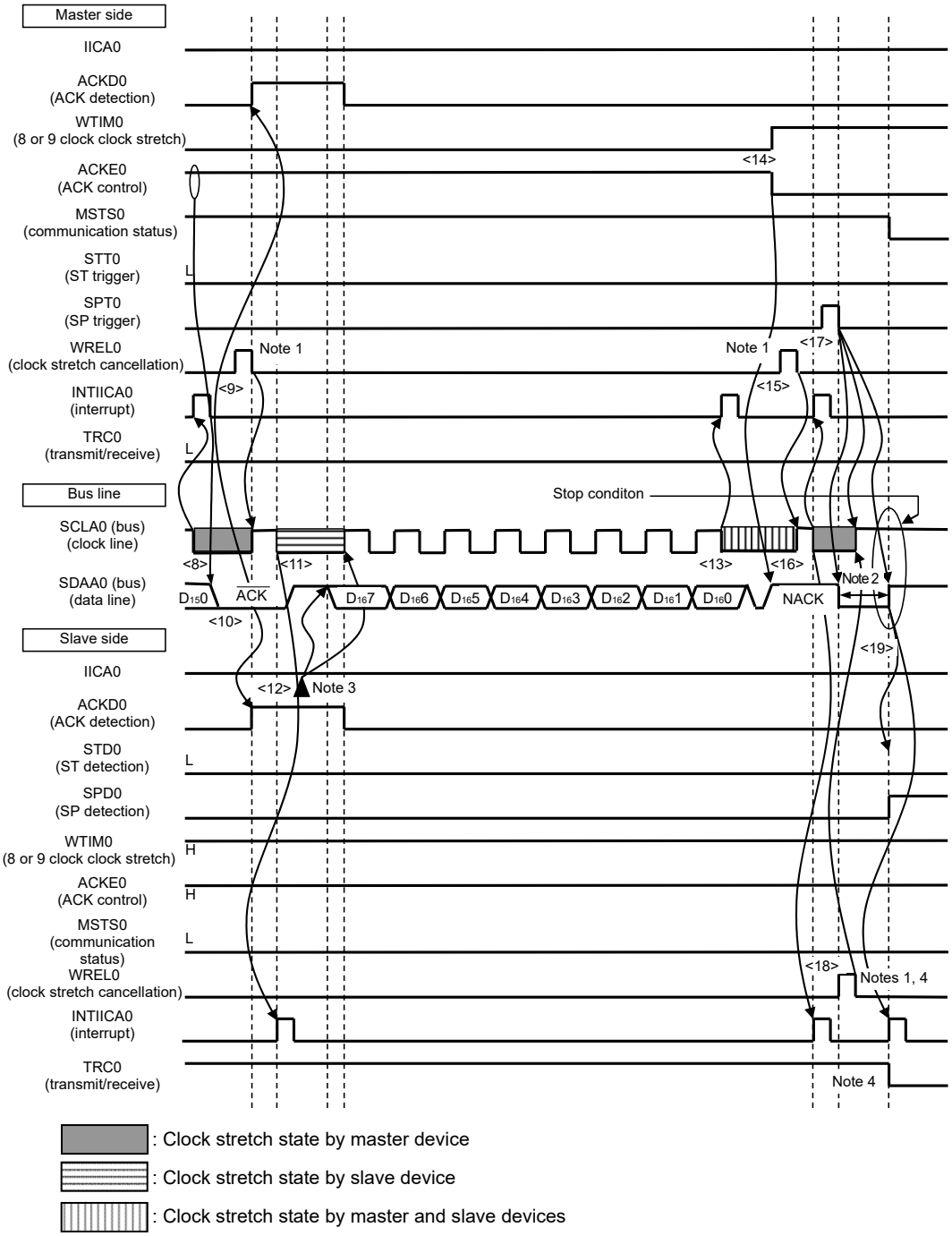
<12> By the slave device writing the data to transmit to the IICA0 register, the clock stretch status set by the slave device is released. The slave device then starts transferring data to the master device.

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 13-33 represent the entire procedure for communicating data using the I^2C bus. Figure 13-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 13-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 13-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 13-33. 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 transmission by a slave device.

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 13-33 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 13-33 represent the entire procedure for communicating data using the I^2C bus. Figure 13-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 13-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 13-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

CHAPTER 14 USB 2.0 HOST/FUNCTION MODULE (USB)

The availability of each pin of USB port, VBUS supply enable output, and overcurrent detection input differs, depending on the product.

	R5F10J products	R5F10K products
UDP0 pin	√	√
UDM0 pin	√	√
UV_{BUS} pin	√	√
UDP1 pin	√	–
UDM1 pin	√	–
UVBUSEN0 pin	√	–
UVBUSEN1 pin	√	–
UOVRCUR0 pin	√	–
UOVRCUR1 pin	√	–
UV_{DD} pin	√	√

14.1 Functions of USB 2.0 Host/Function Module

The RL78/G1C incorporates a USB 2.0 host/function module (USB module) compliant to USB (Universal Serial Bus) Specification 2.0. The USB module provides capabilities as a host**Note** /function controller which supports full-speed (12 Mbps) and low-speed (1.5 Mbps) transfer.

The USB module can also detect battery charging (hereafter BC) connection during host**Note** /function controller operation compliant to USB Battery Charging Specification Revision 1.2.

The USB host/function module can detect connection of USB host**Note**/function devices compliant with the 2.1A/1.0A charging mode prescribed in USB power supply component specification, which is defined in the Apple Inc. MFi specifications.

Table 14-1 lists the USB Specifications.

Note The host function is not available in the R5F10K products.

Table 14-1. USB module Specifications (1/2)

Item	Specifications
Features	• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated. • The USB host controller and USB function controller are incorporated (can be switched by software). • Operable with self-power mode and bus-power mode.
	Features of the USB host controller **Note** - Two ports are provided. - Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported - Automatic scheduling for SOF and packet transmissions - Programmable intervals for interrupt transfers - On-chip D+/D- pin pull-down resistor
	Features of the USB function controller - One port is provided. - Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported - Control transfer stage control function - Device state control function - Auto response function for SET_ADDRESS request - SOF interpolation function - On-chip D+/D- pin pull-up resistor
Communication data transfer type	• Control transfer • Bulk transfer • Interrupt transfer
Pipe configuration	• Buffer memory for USB communications is provided for 448 byte. • Up to five pipes can be selected (including the default control pipe). • Usable pipe numbers are 0, 4 to 7. • Endpoint numbers can be assigned flexibly to PIPE4 to PIPE7.
	Transfer conditions that can be set for each pipe: • PIPE0: Control transfer only (default control pipe: DCP) Buffer size: 64 byte (single buffer) • PIPE4, PIPE5: Bulk transfer only Buffer size: 64 byte (double buffer can be specified) • PIPE6, PIPE7: Interrupt transfer only Buffer size: 64 byte (single buffer)

Note The host function is not available in the R5F10K products.

Table 14-1. USB module Specifications (2/2)

Item	Specifications
Others	• Reception ending function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • NAK setting function for response PID generated by end of transfer (SHTNAK) • Automatic buffer memory clearing function after reading data in the pipe specified by the DnFIFO (n = 0, 1) port (DCLRM) • USB Battery Charging is supported (USB Battery Charging Specification Revision 1.2) - Host (charging downstream port, dedicated charging port) BC connection detection function (two ports)**Note** is supported - Function (portable device) BC connection detection function (one port) is supported • Apple Inc. MFi specification supported - USB host controller supports 2.1A/1.0A charging mode prescribed in USB power supply component specification (2 ports)**Note** - USB function controller supports 2.1A/1.0A charging mode prescribed in USB power supply component specification (1 port) • Optional functions for battery charging connection detection are provided - USB port voltage output function (four patterns) - USB port voltage detection function (16 stages)

Note The host function is not available in the R5F10K products.

14.2 Configuration of USB 2.0 Host/Function Module

The USB module consists of the following hardware.

- USB 2.0 Host/Function Controller Note 1
  This controls the host/function supporting full-speed (12 Mbps) and low-speed (1.5 Mbps) transfer.

- USB transceiver
  This is a USB transceiver of one alternate port (USB port 0) Note 1 for the host/function, and of one dedicated port (USB port 1) Note 2 for the host.
  This transceiver contains an on-chip pull-down resistor (for the host)/pull-up resistor (for the function) for transfer rate detection.
  An external power supply (via the UV_{DD} pin) or the internal power supply for the USB can be selected as the power supply for the USB transceiver (hereafter USB power supply).

- Buffer memory for USB communications and FIFO/memory controller
  Up to five pipes can be used. End point numbers can be assigned flexibly to PIPE4 to PIPE7 according to peripheral devices and a user system for communication.

- Battery charging detection/controller
  This processes BC connection detection during host Note 1/function controller operation compliant to Battery Charging Specification Revision 1.2.

- Various registers
  There are various registers for control, monitoring, and transmitting/receiving data.
  See **Table 14-3**.

- Various pins
  USB port I/O pins (UDP0, UDM0, UDP1 Note 2, UDM1 Note 2)
  VBUS input pin (UV_{BUS})
  USB transceiver power supply pin (UV_{DD})
  VBUS supply control output pins (UVBUSEN0, UVBUSEN1) Note 2
  Overcurrent detection input pins (UOVRCUR0, UOVRCUR1) Note 2
  See **Table 14-2** for details.

- Others
  The clock controller operates or stops various clocks to be used by the USB module and divides their frequencies.
  The bus interface controller controls access between the CPU, DMA, and each register of the USB module.

Table 14-2 lists the USB Module I/O Pins and **Figure 14-1** the USB Module Block Diagram.

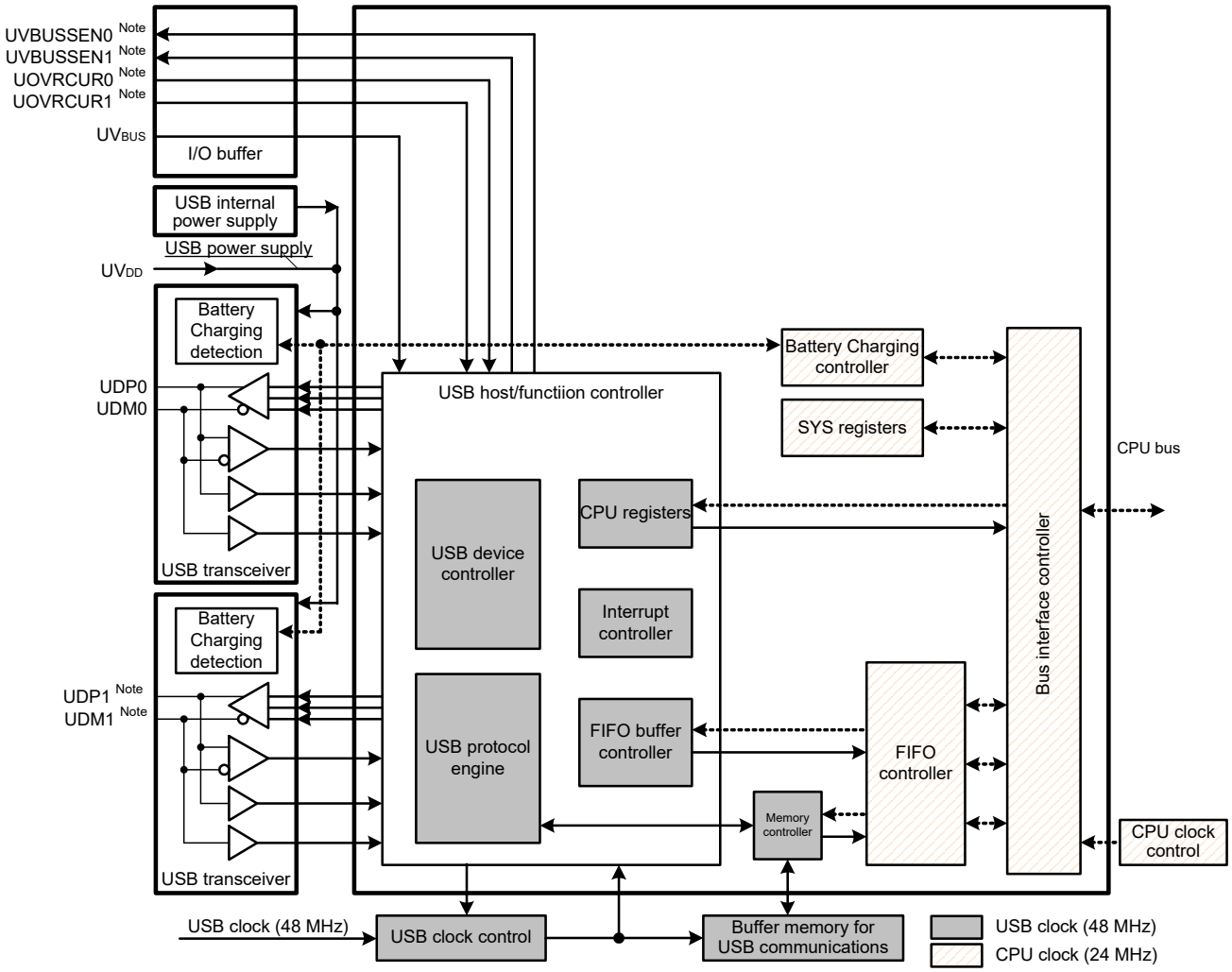
Notes 1. The host function is not available in the R5F10K products.
2. Not available in the R5F10K products.

Table 14-2. USB module I/O Pins

Pin Name	I/O	Function
UDP0	I/O	D+ I/O pin of USB port 0 This pin should be connected to the D+ pin of the USB bus.
UDM0	I/O	D- I/O pin of USB port 0 This pin should be connected to the D- pin of the USB bus.
UV_{BUS}	Input	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. Whether VBUS is connected or disconnected can be detected during operation as a function controller.
UDP1 **Note**	I/O	D+ I/O pin of USB port 1 This pin should be connected to the D+ pin of the USB bus.
UDM1 **Note**	I/O	D- I/O pin of USB port 1 This pin should be connected to the D- pin of the USB bus.
UVBUSEN0 **Note**	Output	VBUS (5 V) supply enable signal to the external power supply IC for USB port 0
UVBUSEN1 **Note**	Output	VBUS (5 V) supply enable signal to the external power supply IC for USB port 1
UOVRCUR0 **Note**	Input	An external overcurrent detection signal for USB port 0 should be connected to this pin.
UOVRCUR1 **Note**	Input	An external overcurrent detection signal for USB port 1 should be connected to this pin.
UV_{DD}	I/O	Power supply (USB power supply) pin for the USB transceiver. This pin should be connected to an external power supply. When using the internal power supply for the USB, a capacitor of 0.33 μF should be connected between this pin and V_{SS}.

Note This function is not mounted in the R5F10K products.

Figure 14-1. Block diagram of USB module



Note This function is not mounted in the R5F10K products.

14.3 Registers Used in USB 2.0 Host/Function Module

Table 14-3 lists the USB Registers.

Table 14-3. List of USB Registers (1/2)

Register Name	Symbol	After Reset	Address	Access Size
System Configuration Control Register	SYSCFG	0000H	F0400H, F0401H	16
System Configuration Control Register 1	SYSCFG1 Note	0000H	F0402H, F0403H	16
System Configuration Status Register 0	SYSSTS0	X00X0000 00X00000B	F0404H, F0405H	16
System Configuration Status Register 1	SYSSTS1 Note	X00X0000 00X00000B	F0406H, F0407H	16
Device State Control Register 0	DVSTCTR0	0000H	F0408H, F0409H	16
Device State Control Register 1	DVSTCTR1 Note	0000H	F040AH, F040BH	16
DMA0-FIFO Pin Configuration Register	DMA0PCFG	0000H	F0410H, F0411H	16
DMA1-FIFO Pin Configuration Register	DMA1PCFG	0000H	F0412H, F0413H	16
CFIFO Port Register	CFIFOM	0000H	F0414H, F0415H	8, 16
D0FIFO Port Register	D0FIFOM	0000H	F0418H, F0419H	8, 16
D1FIFO Port Register	D1FIFOM	0000H	F041CH, F041DH	8, 16
CFIFO Port Select Register	CFIFOSEL	0000H	F0420H, F0421H	16
CFIFO Port Control Register	CFIFOCTR	0000H	F0422H, F0423H	16
D0FIFO Port Select Register	D0FIFOSEL	0000H	F0428H, F0429H	16
D0FIFO Port Control Register	D0FIFOCTR	0000H	F042AH, F042BH	16
D1FIFO Port Select Register	D1FIFOSEL	0000H	F042CH, F042DH	16
D1FIFO Port Control Register	D1FIFOCTR	0000H	F042EH, F042FH	16
Interrupt Enable Register 0	INTENB0	0000H	F0430H, F0431H	16
Interrupt Enable Register 1	INTENB1 Note	0000H	F0432H, F0433H	16
Interrupt Enable Register 2	INTENB2 Note	0000H	F0434H, F0435H	16
BRDY Interrupt Enable Register	BRDYENB	0000H	F0436H, F0437H	16
NRDY Interrupt Enable Register	NRDYENB	0000H	F0438H, F0439H	16
BEMP Interrupt Enable Register	BEMPENB	0000H	F043AH, F043BH	16
SOF Output Configuration Register	SOFCFG	0000H	F043CH, F043DH	16
Interrupt Status Register 0	INTSTS0	00000000 X0000000B	F0440H, F0441H	16
Interrupt Status Register 1	INTSTS1 Note	XX0X0000 00000000B	F0442H, F0443H	16
Interrupt Status Register 2	INTSTS2 Note	X00X0000 00000000B	F0444H, F0445H	16
BRDY Interrupt Status Register	BRDYSTS	0000H	F0446H, F0447H	16
NRDY Interrupt Status Register	NRDYSTS	0000H	F0448H, F0449H	16
BEMP Interrupt Status Register	BEMPSTS	0000H	F044AH, F044BH	16
Frame Number Register	FRMNUM	0000H	F044CH, F044DH	16

Note The SYSCFG1, SYSSTS1, DVSTCTR1, INTENB1, INTENB2, INTSTS1, and INTSTS2 registers cannot be used in the following products.
Part number :
R5F10KBCGNA/R5F10KBCGXXXNA, R5F10KBCANA/R5F10KBCAXXXNA
R5F10KGCGNA/R5F10KGCGXXXNA, R5F10KGCANA/R5F10KGCAXXXNA
R5F10KBCGFP/R5F10KBCGXXXFP, R5F10KBCAFP/R5F10KBCAXXXFP
R5F10KGCGFB/R5F10KGCGXXXFB, R5F10KGCAFB/R5F10KGCAXXXFB XXX: ROM number

Table 14-3. List of USB Registers (2/2)

Register Name	Symbol	After Reset	Address	Access Size
USB Address Register	USBADDR	0000H	F0450H, F0451H	16
USB Request Type Register	USBREQ	0000H	F0454H, F0455H	16
USB Request Value Register	USBVAL	0000H	F0456H, F0457H	16
DMA Transfer D0FIFO Port Register	D0FIFO	0000H	FFF58H, FFFF59H **Note 1**	8, 16
DMA Transfer D1FIFO Port Register	D1FIFO	0000H	FFF5CH, FFFF5DH **Note 1**	8, 16
USB Request Index Register	USBINDX	0000H	F0458H, F0459H	16
USB Request Length Register	USBLENG	0000H	F045AH, F045BH	16
DCP Configuration Register	DCPCFG	0000H	F045CH, F045DH	16
DCP Maximum Packet Size Register	DCPMAXP	0040H	F045EH, F045FH	16
DCP Control Register	DCPCTR	0040H	F0460H, F0461H	16
Pipe Window Select Register	PIPESEL	0000H	F0464H, F0465H	16
Pipe Configuration Register	PIPECFG	0000H	F0468H, F0469H	16
Pipe Maximum Packet Size Register	PIPEMAXP	0000H/0040H **Note 2**	F046CH, F046DH	16
Pipe Cycle Control Register	PIPEPERI	0000H	F046EH, F046FH	16
Pipe 4 Control Register	PIPE4CTR	0000H	F0476H, F0477H	16
Pipe 5 Control Register	PIPE5CTR	0000H	F0478H, F0479H	16
Pipe 6 Control Register	PIPE6CTR	0000H	F047AH, F047BH	16
Pipe 7 Control Register	PIPE7CTR	0000H	F047CH, F047DH	16
Pipe 4 Transaction Counter Enable Register	PIPE4TRE	0000H	F049CH, F049DH	16
Pipe 4 Transaction Counter Register	PIPE4TRN	0000H	F049EH, F049FH	16
Pipe 5 Transaction Counter Enable Register	PIPE5TRE	0000H	F04A0H, F04A1H	16
Pipe 5 Transaction Counter Register	PIPE5TRN	0000H	F04A2H, F04A3H	16
BC Control Register 0	USBBCCTRL0	0000H	F04B0H, F04B1H	16
BC Control Register 1	USBBCCTRL1 **Note 3**	0000H	F04B4H, F04B5H	16
BC Option Control Register 0	USBBCOPT0	0000H	F04B8H, F04B9H	16
BC Option Control Register 1	USBBCOPT1 **Note 3**	0000H	F04BCH, F04BDH	16
USB Module Control Register	USBMC	0002H	F04CCH, F04CDH	16
Device Address 0 Configuration Register	DEVADD0	0000H	F04D0H, F04D1H	16
Device Address 1 Configuration Register	DEVADD1	0000H	F04D2H, F04D3H	16
Device Address 2 Configuration Register	DEVADD2	0000H	F04D4H, F04D5H	16
Device Address 3 Configuration Register	DEVADD3	0000H	F04D6H, F04D7H	16
Device Address 4 Configuration Register	DEVADD4	0000H	F04D8H, F04D9H	16
Device Address 5 Configuration Register	DEVADD5	0000H	F04DAH, F04DBH	16

Notes 1. The addresses for DMA transfers are allocated in the special function register (SFR) area.

2. The initial value of this register differs according to the setting of the PIPESEL3 to PIPESEL0 bits. The initial value is 0000H when the pipe is not selected and 0040H when selected.

3. The USBBCCTRL1, USBBCOPT1 registers cannot be used in the following products.
Part number :
R5F10KBCGNA/R5F10KBCGXXXNA, R5F10KBCANA/R5F10KBCAXXXNA
R5F10KGCGNA/R5F10KGCGXXXNA, R5F10KGCANA/R5F10KGCAXXXNA
R5F10KBCGFP/R5F10KBCGXXXFP, R5F10KBCAFP/R5F10KBCAXXXFP
R5F10KGCGFB/R5F10KGCGXXXFB, R5F10KGCAFB/R5F10KGCAXXXFB XXX: ROM number

Table 14-4. Registers Initialized by Writing USBE = 0 (When Function Controller Function is Selected)

Register	Symbol	Remarks
SYSSTS0	LNST1, LNST0	The value is retained when the host controller function is selected.
SYSSTS1	LNST1, LNST0	The value is retained when the host controller function is selected.
DVSTCTR0	RHST2 to RHST0	
DVSTCTR1	RHST2 to RHST0	
INTSTS0	DVSQ2 to DVSQ0	The value is retained when the host controller function is selected.
USBADDR	USBADDR	The value is retained when the host controller function is selected.
USBREQ	BREQUEST, BMREQUESTTYPE	The value is retained when the host controller function is selected.
USBVAL	WVALUE	The value is retained when the host controller function is selected.
USBINDX	WINDEX	The value is retained when the host controller function is selected.
USBLENG	WLENGTH	The value is retained when the host controller function is selected.

Table 14-5. Registers Initialized by Writing USBE = 0 (When Host Controller Function is Selected)

Register	Symbol	Remarks
DVSTCTR0	RHST2 to RHST0	
DVSTCTR1	RHST2 to RHST0	
FRMNUM	FRNM	The value is retained when the function controller function is selected.

14.3.1 System configuration control register (SYSCFG), system configuration control register 1 (SYSCFG1)

Figure 14-2. Format of System Configuration Control Register (SYSCFG)

Address: F0400H, F0401H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSCFG	–	–	–	–	–	SCKE	–	CNEN	–	DCFM **Note 1**	DRPD	DRPDU	DMRPU	–	–	USBE

Bits 15 to 11	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

SCKE	USB module clock enable	R/W
0	Stops supplying the clock signal to the USB module.	R/W
1	Enables supplying the clock signal to the USB module.	
Stops or enables supplying 48-MHz clock signals to the USB module. After setting the SCKE bit to 1, be sure to read this bit to confirm that it is actually set to 1. When the SCKE bit is 0, the USB module registers that can be read and written are SYSCFG, SYSCFG1, DMA0PCFG, DMA1PCFG, USBBCCTRL0, USBBCCTRL1, USBBCOPT0, USBBCOPT1, and USBMC. When this bit is 0, only the SYSCFG, SYSCFG1, DMA0PCFG, and DMA1PCFG registers can be read from and written to.		

Bit 9	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

CNEN	USB port 0 single end receiver enable	R/W
0	Single end receiver operation is disabled.	R/W
1	Single end receiver operation is enabled.	
Enables or disables the single end receiver. Setting the CNEN bit is set to 1 allows the USB module to enable the single end receiver of USB port 0 and set the LNST bit to monitor the status of the D+/D- lines. The CNEN bit is used to monitor LNST when the USB module operates as a Portable Device for Battery Charging.		

Bit 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DCFM**Note 1**	Controller function select	R/W
0	Function controller function is selected.	R/W
1	Host controller function is selected.	
Selects the function of the USB module. The DCFM bit should be modified while DPRPU is 0 and DRPD is 0.		

DRPD	USB port 0 D+/D- line resistor control	R/W
0	Pulling down the lines is disabled.	R/W
1	Pulling down the lines is enabled.	
Enables or disables pulling down the D+/D- lines of USB port 0 when the host controller function is selected. The DRPD bit should be set to 1 if the host controller function is selected, and should be set to 0 if the function controller function, host BC connection detection function, or function BC connection detection function is selected.		

DPRPU	USB port 0 D+ line resistor control Note 2	R/W
0	Pulling up the line is disabled.	R/W
1	Pulling up the line is enabled.	
Enables or disables pulling up the D+ line when the function controller function is selected. Setting the DPRPU bit to 1 when the function controller function is selected allows the USB module to enable pulling up the D+ line of USB port 0, thus notifying the USB host of connection as a full-speed device. Modifying the DPRPU bit from 1 to 0 allows the USB module to disable pulling up the D+ line of USB port 0, thus notifying the USB host of disconnection. The DPRPU bit should be set to 1 if the function controller function is selected, and should be set to 0 if the function controller function, host BC connection detection function, or function BC connection detection function is selected.		

DMRPU	USB port 0 D- line resistor control Note 2	R/W
0	Pulling up the line is disabled.	R/W
1	Pulling up the line is enabled.	
Enables or disables pulling up the D- line when the function controller function is selected. Setting the DMRPU bit to 1 when the function controller function is selected allows the USB module to enable pulling up the D- line of USB port 0, thus notifying the USB host of connection as a low-speed device. Modifying the DMRPU bit from 1 to 0 allows the USB module to disable pulling up the D- line of USB port 0, thus notifying the USB host of disconnection. The DMRPU bit should be set to 1 if the function controller function is selected, and should be set to 0 if the function controller function, host BC connection detection function, or function BC connection detection function is selected.		

Bits 2, 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

USBE	USB module operation enable	R/W
0	USB module operation is disabled.	R/W
1	USB module operation is enabled.	
Enables or disables operation of the USB module. Modifying the USBE bit from 1 to 0 initializes some register bits as listed in Tables 14-4 and 14-5. This bit should be modified while SCKE is 1. When the host controller function is selected, USBE should be set to 1 after setting DRPD to 1, eliminating LNST bit chattering, and checking that the USB bus state has been settled.		

Notes 1. When using any of the following products, use the DCFM bit setting as is (0). If this bit needs to be written, write 0.
Part number :
R5F10KBCGNA/R5F10KBCGXXXNA, R5F10KBCANA/R5F10KBCAXXXNA
R5F10KGCGNA/R5F10KGCGXXXNA, R5F10KGCANA/R5F10KGCAXXXNA
R5F10KBCGFP/R5F10KBCGXXXFP, R5F10KBCAFP/R5F10KBCAXXXFP
R5F10KGCGFB/R5F10KGCGXXXFB, R5F10KGCAFB/R5F10KGCAXXXFB XXX: ROM number

2. Setting the DMRPU and DPRPU bits simultaneously to 1 (to enable pulling up the line) is prohibited.

Figure 14-3. Format of System Configuration Control Register 1 (SYSCFG1)

Address: F0402H, F0403H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSCFG1	–	–	–	–	–	–	–	CNEN	–	–	DRPD	–	–	–	–	–

Bit 15	Reserved	R/W
–	The write value must be 0. The read value is 0.	–

Bits 14 to 9	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

CNEN	USB port 1 single end receiver enable	R/W
0	Single end receiver operation is disabled.	R/W
1	Single end receiver operation is enabled.	
Enables or disables the single end receiver. Setting the CNEN bit is set to 1 allows the USB module to enable the single end receiver of USB port 1 and set the LINST bit to monitor the status of the D+/D- lines. The CNEN bit is used to monitor LNST when the USB module operates as a Portable Device for Battery Charging.		

Bits 7, 6	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DRPD	USB port 1 D+/D- line resistor control	R/W
0	Pulling down the lines is disabled.	R/W
1	Pulling down the lines is enabled.	
Enables or disables pulling down the D+/D- lines of USB port 1 when the host controller function is selected. This bit should be set to 1 if the function controller function is selected, and should be set to 0 if the function controller function, host BC connection detection function, or function BC connection detection function is selected.		

Bits 4 to 0	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

14.3.2 System configuration status register n (SYSSTSn) (n = 0, 1)

Figure 14-4. Format of System Configuration Status Register n (SYSSTSn) (n = 0, 1)

Address: F0404H, F0405H (SYSSTS0), F0406H, F0407H (SYSSTS1) After reset: X00X0000 00X00000B

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYSSTSn	OVCMON1	–	–	–	–	–	–	–	–	HTACT	–	–	–	–	LNST1	LNST0

OVCMON1	External UOVRCURn input pin monitor Note	R/W
These bits indicate the status of overcurrent from an external power-supply IC. The OVCMON1 bit indicates the status of the UOVRCURn pin.		R

Bit 13	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

Bits 12	Reserved	R/W
–	The write value must be 0. The read value is undefined.	–

Bits 11 to 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

HTACT	USB port n USB host sequencer status monitor	R/W
0	Host sequencer of the USB module is completely stopped.	R
1	Host sequencer of the USB module is not completely stopped.	
This bit is 0 when the host sequencer of the USB module is completely stopped. Make sure the HTACT bit is 0 when stopping the clock supply to the USB module.		

Bit 5	Reserved	R/W
–	The write value must be 0. The read value is undefined.	–

Bits 4 to 2	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

LNST1	LNST0	USB port n USB data line status monitor	R/W
0	0	SE0	R
0	1	J-State (full speed)/K-State (low speed)	
1	0	K-State (full speed)/J-State (low speed)	
1	1	SE1	
These bits indicate the status of the USB data bus lines (D+ and D-). The LNST1 and LNST0 bits should be read after enabling pull-down of the line (DRPD = 1).			

Note The read value depends on the status of the UOVRCURn pin.

14.3.3 Device state control register n (DVSTCTRn) (n = 0, 1)

Figure 14-5. Format of Device State Control Register 0 (DVSTCTR0)

Address: F0408H, F0409H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVSTCTR0	–	–	–	–	–	–	VBUSEN Note 1	WKUP	RWUPE	USBRST	RESUME	UACT	–	RHST2	RHST1	RHST0

Bit 15	Reserved	R/W
–	The write value must be 0. The read value is 0.	–

Bits 14 to 12	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

Bits 11, 10	Reserved	R/W
–	The write value must be 0. The read value is 0.	–

VBUSEN Note 1	USB port 0 UVBUSEN0 output pin control	R/W
The VBUSEN bit value is output as the status of the external UVBUSEN0 pin without change.		R/W

WKUP	USB port 0 wakeup output	R/W
0	Remote wakeup signal is not output.	R/W Note 2
1	Remote wakeup signal is output.	
Enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller function is selected. The USB module controls the output time of a remote wakeup signal. When this bit is set to 1, the USB module clears this bit to 0 after outputting the 10-ms K-state. According to the USB Specifications, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. Even if the USB module writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms. Do not write 1 to this bit, unless the device state is in the suspended state (DVSQ2 to DVSQ0 in the INTSTS0 register = 1xxB) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to WKUP while SCKE in the SYSCFG register = 1). This bit should be set to 0 if the host controller function is selected.		

RWUPE	USB port 0 wakeup detection enable	R/W
0	Downstream port wakeup is disabled.	R/W
1	Downstream port wakeup is enabled.	
Enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller function is selected. With this bit set to 1, on detecting the remote wakeup signal, the USB module detects the resume signal (K-state for 2.5 μs) from the downstream port device and performs the resume processing (drives the port to the K-state). With this bit set to 0, the USB module ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port. While the PWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SCKE should be set to 1). This bit should be set to 0 if the function controller function is selected.		

USBRST	USB port 0 USB bus reset output	R/W
0	USB bus reset signal is not output.	R/W
1	USB bus reset signal is output.	
Controls the USB bus reset signal output when the host controller function is selected. When the host controller function is selected, setting this bit to 1 allows the USB module to drive SE0 of the USB port to reset the USB bus. The USB module continues outputting SE0 while USBRST is 1 (until software sets USBRST to 0). The USBRST bit should be 1 (= USB bus reset period) for the time defined by the USB Specifications 2.0. Writing 1 to this bit during communication (UACT = 1) or during the resume processing (RESUME = 1) prevents the USB module from starting the USB bus reset processing until UACT and RESUME become 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to USBRST). This bit should be set to 0 if the function controller function is selected.		

RESUME	USB port 0 resume output	R/W
0	Resume signal is not output.	R/W
1	Resume signal is output.	
Controls the resume signal output when the host controller function is selected. Setting the RESUME bit to 1 allows the USB module to drive the port to the K-state and output the resume signal. The USB module continues outputting K-state while RESUME is 1 (until software sets RESUME to 0). The RESUME bit should be 1 (= resume period) for the time defined by the USB Specifications 2.0. This bit should be set to 1 in the suspended state. Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to RESUME). This bit should be set to 0 if the function controller function is selected.		

UACT	USB port 0 USB bus enable	R/W
0	Downstream port is disabled (SOF transmission is disabled).	R/W
1	Downstream port is enabled (SOF transmission is enabled).	
Enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller function is selected. With this bit set to 1, the USB module puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception. This module starts outputting SOF packets within one frame after software has written 1 to UACT. With this bit set to 0, the USB module enters the idle state after outputting SOF packets. The USB module sets the UACT bit to 0 on any of the following conditions. • A DTCH interrupt is detected during communication (while UACT = 1). • An EOFERR interrupt is detected during communication (while UACT = 1). Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to USBRST) or at the end of the resume processing from the suspended state (writing 0 to RESUME). This bit should be set to 0 if the function controller function is selected.		

Bit 3	Reserved	R/W
–	Nothing is assigned. The write value must be 0. The read value is 0.	–

RHST2	RHST1	RHST0	USB port 0 USB bus reset status	R/W
• When the host controller function is selected				R
0	0	0	Communication speed not determined (powered state or no connection)	
1	X	X	USB bus reset in progress	
0	0	1	Low-speed connection	
0	1	0	Full-speed connection	
• When the function controller function is selected				
0	0	0	Communication speed not determined	
1	X	X	USB bus reset in progress	
0	0	1	Low-speed connection	
0	1	0	Full-speed connection	
The RHST[2:0] bits indicate the status of the USB bus reset. When the host controller function is selected, these bits indicate 100B after software has written 1 to USBRST. The USB module fixes the value of these bits when software writes 0 to USBRST and the USB module completes SE0 driving. When the function controller function is selected, a DVST interrupt is generated as soon as the USB module detects the USB bus reset and then these bits are fixed to 010B.				

Notes 1. When using any of the following products, use the VBUSEN bit setting as is (0). If this bit needs to be written, write 0.
R5F10KBCGNA/R5F10KBCGXXXNA, R5F10KBCANA/R5F10KBCAXXXNA
R5F10KGCGNA/R5F10KGCGXXXNA, R5F10KGCANA/R5F10KGCAXXXNA
R5F10KBCGFP/R5F10KBCGXXXFP, R5F10KBCAFP/R5F10KBCAXXXFP
R5F10KGCGFB/R5F10KGCGXXXFB, R5F10KGCAFB/R5F10KGCAXXXFB XXX: ROM number

2. Only 1 can be written.

Remark x = Don't care

Figure 14-6. Format of Device State Control Register 1 (DVSTCTR1)

Address: F040AH, F040BH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVSTCTR1	–	–	–	–	–	–	VBUSEN	–	RWUPE	USBRST	RESUME	UACT	–	RHST2	RHST1	RHST0

Bits 15, 14	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

Bit 13	Reserved	R/W
–	The write value must be 0. The read value is 0.	–

Bits 12 to 10	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

VBUSEN	USB port 1 UVBUSEN1 output pin control	R/W
The VBUSEN bit value is output as the status of the external UVBUSEN1 pin without change.		R/W

Bit 8	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

RWUPE	USB port 1 wakeup detection enable	R/W
0	Downstream port wakeup output is disabled.	R/W
1	Downstream port wakeup is enabled.	
Enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller function is selected. With this bit set to 1, on detecting the remote wakeup signal, the USB module detects the resume signal (K-state for 2.5 μs) from the downstream port device and performs the resume processing (drives the port to the K-state). With this bit set to 0, the USB module ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port. While the PWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SCKE should be set to 1). This bit should be set to 0 if the function controller function is selected.		

USBRST	USB port 1 USB bus reset output	R/W
0	USB bus reset signal is not output.	R/W
1	USB bus reset signal is not output.	
Controls the USB bus reset signal output when the host controller function is selected. When the host controller function is selected, setting this bit to 1 allows the USB module to drive SE0 of the USB port to reset the USB bus. The USB module continues outputting SE0 while USBRST is 1 (until software sets USBRST to 0). The USBRST bit should be 1 (= USB bus reset period) for the time defined by the USB Specifications 2.0. Writing 1 to this bit during communication (UACT = 1) or during the resume processing (RESUME = 1) prevents the USB module from starting the USB bus reset processing until both UACT and RESUME become 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to USBRST). This bit should be set to 0 if the function controller function is selected.		

RESUME	USB port 1 resume output	R/W
0	Resume signal is not output.	R/W
1	Resume signal is output.	
Controls the resume signal output when the host controller function is selected. Setting the RESUME bit to 1 allows the USB module to drive the port to the K-state and output the resume signal. The USB module continues outputting K-state while RESUME is 1 (until software sets RESUME to 0). The RESUME bit should be 1 (= resume period) for the time defined by the USB Specifications 2.0. This bit should be set to 1 in the suspended state. Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to RESUME). This bit should be set to 0 if the function controller function is selected.		

UACT	USB port 1 USB bus enable	R/W
0	Downstream port is disabled (SOF transmission is disabled).	R/W
1	Downstream port is enabled (SOF transmission is enabled).	
Enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller function is selected. With this bit set to 1, the USB module puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception. This module starts outputting SOF packets within one frame after software has written 1 to the UACT bit. With this bit set to 0, the USB module enters the idle state after outputting SOF packets. The USB module sets the UACT bit to 0 on any of the following conditions. A DTCH interrupt is detected during communication (while UACT = 1). An EOFERR interrupt is detected during communication (while UACT = 1). Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to USBRST) or at the end of the resume processing from the suspended state (writing 0 to RESUME). This bit should be set to 0 if the function controller function is selected.		

Bit 3	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

RHST2	RHST1	RHST0	USB port 1 USB bus reset status	R/W
• When the host controller function is selected				R
0	0	0	Communication speed not determined (powered state or no connection)	
1	X	X	USB bus reset in progress	
0	0	1	Low-speed connection	
0	1	0	Full-speed connection	
The RHST[2:0] bits indicate 000B when the function controller function is selected. These bits indicate the status of the USB bus reset when the host controller function is selected. When the host controller function is selected, these bits indicate 100B after software has written 1 to USBRST. The USB module fixes the value of these bits when software writes 0 to USBRST and the USB module completes SE0 driving.				

Remark x = Don't care

14.3.4 DMAn-FIFO pin configuration register (DMAnPCFG) (n = 0, 1)

Figure 14-7. Format of DMAn-FIFO Pin Configuration Register (DMAnPCFG) (n = 0, 1)

Address: F0410H, F0411H (DMA0PCFG), F0412H, F0413H (DMA1PCFG) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAnPCFG	–	–	–	–	–	–	–	DnFW RENDE	–	–	–	–	–	–	–	–

Bits 15 to 9	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DnFWRENDE	DMAn-FIFO port write end enable bit	R/W
0	Disabled	R/W
1	Enabled	
Enables or disables the end signal from the DMAC.		

Bits 7 to 0	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

14.3.5 CFIFO port register (CFIFOM), DnFIFO port register (DnFIFOM) (n = 0, 1)

Figure 14-8. Format of CFIFO Port Register (CFIFOM)

Address: F0414H, F0415H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIFOM	CFIFO[15:0]															

CFIFO[15:0]	CFIFO port	R/W
Accessing the CFIFO bits allow reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer. The CFIFO port register can be accessed only while the FRDY bit in the CFIFO port control register (CFIFOCTR) is 1. The valid bits in the CFIFO port register depend on the settings of the corresponding MBW and BIGEND bits as shown in Tables 14-6 and 14-7.		R/W

Cautions 1. The FIFO buffer for DCP (control transfer) cannot be accessed using a DMA transfer.

2. When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed.

3. The same pipe should not be assigned to different FIFO ports.

4. There are two FIFO buffer states: the access right is on the CPU side and it is on the SIE side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

Table 14-6. Endian Operation in 16-Bit Access

BIGEND Bit in CFIFO Port Register, DnFIFO Port Register	Bits 15 to 8	Bits 7 to 0
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

Table 14-7. Endian Operation in 8-Bit Access

BIGEND Bit in CFIFO Port Register, DnFIFO Port Register	Bits 15 to 8	Bits 7 to 0
0	Access prohibited	N + 0 data
1	Access prohibited	N + 0 data

Figure 14-9. Format of DnFIFO Port Register (DnFIFOM) (n = 0, 1)

Address: F0418H, F0419H (D0FIFOM), F041CH, F041DH (D1FIFOM) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DnFIFOM	DnFIFO[15:0]															

DnFIFO[15:0]	FIFO port	R/W
Addresses for CPU transfers using the DnFIFO port. Accessing the DnFIFO bits allow reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer. The DnFIFO port register can be accessed only while the FRDY bit in the DnFIFO port control register (DnFIFOCTR) is 1. The valid bits in the DnFIFO port register depend on the settings of the corresponding MBW and BIGEND bits as shown in Tables 14-6 and 14-7.		R/W

Cautions 1. The FIFO buffer for DCP (control transfer) cannot be accessed. This register cannot be used for the addresses for DMA transfers.

2. When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed.

3. The same pipe should not be assigned to different FIFO ports.

4. There are two FIFO buffer states: the access right is on the CPU side and it is on the SIE side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

14.3.6 DMA transfer DnFIFO port register (DnFIFO) (n = 0, 1)

Figure 14-10. Format of DMA Transfer DnFIFO Port Register (DnFIFO) (n = 0, 1)

Address: FFF58H, FFF59H (D0FIFO), FFF5CH, FFF5DH (D1FIFO) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DnFIFO	DnFIFO[15:0]															

DnFIFO[15:0]	DMA transfer DnFIFO port	R/W
Addresses for DMA transfers using the DnFIFO port. The valid bits in the DMA transfer DnFIFO port register depend on the settings of the corresponding MBW and BIGEND bits as shown in Tables 14-6 and 14-7.		R/W

Cautions 1. Accessing this register from the CPU is prohibited.

2. When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed.

3. The same pipe should not be assigned to different FIFO ports.

14.3.7 CFIFO port select register (CFIFOSEL), DnFIFO port select register (DnFIFOSEL) (n = 0, 1)

Figure 14-11. Format of CFIFO Port Select Register (CFIFOSEL)

Address: F0420H, F0421H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIFOSEL	RCNT	REW	–	–	–	MBW	–	BIGEND	–	–	ISEL	–	CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0

RCNT	Read count mode	R/W
0	When all of the receive data has been read from the CFIFO, 0 is written to the DTLN bit. (In double buffer mode, the DTLN bit value is cleared when all the data has been read from only a single plane.)	R/W
1	The DTLN bit is decremented each time the receive data is read from the CFIFO.	
Specifies the read mode for the value in the DTLN[8:0] bits in the CFIFOCTR register.		

REW	Buffer pointer rewind	R/W
0	Disable (the buffer pointer is not rewound.)	R/W **Note 1**
1	The buffer pointer is rewound.	
Specifies whether or not to rewind the buffer pointer. When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed). Do not set the REW bit to 1 simultaneously with modifying the CURPIPE3 to CURPIPE0 bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1. To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.		

Bits 13 to 11	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

MBW	CFIFO port access bit width	R/W
0	8-bit width	R/W
1	16-bit width	
Specifies the bit width for accessing the CFIFO port. When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read. When the selected pipe is in the receiving direction, set the CURPIPE3 to CURPIPE0 and MBW bits simultaneously. If the size of data to be read is an odd number of bytes when 16-bit width is selected, delete unnecessary bytes after reading the data in words. When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory. An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.		

Bit 9	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

BIGEND	CFIFO port endian control	R/W
0	Little endian	R/W
1	Big endian	
Specifies the byte endian for the CFIFO port.		

Bits 7, 6	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

ISEL	CFIFO port access direction when DCP is selected	R/W
0	Reading from the buffer memory is selected	R/W
1	Writing to the buffer memory is selected	
After writing to the ISEL bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process. Set this bit and the CURPIPE3 to CURPIPE0 bits simultaneously.		

Bit 4	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0	CFIFO port access pipe specification Note 2	R/W
0	0	0	0	DCP (Default control pipe)	R/W
0	1	0	0	Pipe 4	
0	1	0	1	Pipe 5	
0	1	1	0	Pipe 6	
0	1	1	1	Pipe 7	
Other than above				Do not set.	
The CURPIPE3 to CURPIPE0 bits specify the pipe number using which data is read or written through the CFIFO port. After writing to the CURPIPE3 to CURPIPE0 bits, read these bits to check that the written value agrees with the read value before proceeding to the next process. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.					

Notes 1. Only 0 can be read.

2. The same pipe number should not be set by the CURPIPE3 to CURPIPE0 bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

Figure 14-12. Format of DnFIFO port select register (DnFIFOSEL) (n = 0, 1)

Address: F0428H, F0429H (D0FIFOSEL), F042CH, F042DH (D1FIFOSEL) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DnFIFOSEL	RCNT	REW	DCLRM	DREQE	–	MBW	–	BIGEND	–	–	–	–	CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0

RCNT	Read count mode	R/W
0	When all of the receive data has been read from the DnFIFO, 0 is written to the DTLN[8:0] bits. (In double buffer mode, the DTLN[8:0] bit value is cleared when all the data has been read from only a single plane.)	R/W
1	The DTLN[8:0] bits are decremented each time the receive data is read from the DnFIFO.	
Specifies the read mode for the value in the DTLN[8:0] bits in the DnFIFOCTR register. When accessing DnFIFO with the BFRE bit set to 1, set the RCNT bit to 0.		

REW	Buffer pointer rewind	R/W
0	Disable (the buffer pointer is not rewound.)	R/W **Note 1**
1	The buffer pointer is rewound.	
Specifies whether or not to rewind the buffer pointer. When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed). Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, be sure to check that FRDY is 1. When accessing the DnFIFO with the BFRE bit set to 1, do not set the REW bit to 1 while reading of the short packet data is completed. To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.		

DCLRM	Auto buffer memory clear mode accessed after specified pipe data is read	R/W
0	Auto buffer clear mode is disabled.	R/W
1	Auto buffer clear mode is enabled.	
Enables or disables the buffer memory to be cleared automatically after data has been read out using the selected pipe. With the DCLRM bit set to 1, the USB module sets BCLR to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while BFRE in the PIPECFG register is 1. When using the USB module with BRDYM in the SOFCFG register set to 1, set the DCLRM bit to 0.		

DREQE	DMA transfer request enable	R/W
0	DMA transfer request disabled	R/W
1	DMA transfer request enabled	
Enables or disables the DMA transfer request to be issued. Before setting the DREQE bit to 1 to enable the DMA transfer request to be issued, set the CURPIPE bits. When modifying the setting of the CURPIPE bits, set the DREQE bit to 0 first.		

Bit 11	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

MBW	DnFIFO port access bit width	R/W
0	8-bit width	R/W
1	16-bit width	
Specifies the bit width for accessing the DnFIFO port. When the selected pipe is in the receiving direction, once reading data is started after setting the MBW bit, this bit should not be modified until all the data has been read. When the selected pipe is in the receiving direction, set the CURPIPE and MBW bits simultaneously. If the size of data to be read is an odd number of bytes when 16-bit width is selected, delete unnecessary bytes after reading the data in words. When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory. An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.		

Bit 9	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

BIGEND	DnFIFO port endian control	R/W
0	Little endian	R/W
1	Big endian	
Specifies the byte endian for the DnFIFO port.		

Bit 7	Reserved	R/W
–	The write value must be 0. The read value is 0.	–

Bits 6 to 4	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

CURPIPE3	CURPIPE2	CURPIPE1	CURPIPE0	DnFIFO port access pipe specification Note 2	R/W
0	0	0	0	No pipe specified	R/W
0	1	0	0	Pipe 4	
0	1	0	1	Pipe 5	
0	1	1	0	Pipe 6	
0	1	1	1	Pipe 7	
Other than above				Do not set.	
The CURPIPE3 to CURPIPE0 bits specify the pipe number using which data is read or written through the DnFIFO port. After writing to CURPIPE3 to CURPIPE0 bits, then read these bits to check that the written value agrees with the read value before proceeding to the next process. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.					

Notes 1. Only 0 can be read.

2. The same pipe number should not be set by the CURPIPE3 to CURPIPE0 bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

14.3.8 CFIFO port control register (CFIFOCTR), DnFIFO port control register (DnFIFOCTR) (n = 0, 1)

Figure 14-13. Format of CFIFO Port Control Register (CFIFOCTR)

Address: F0422H, F0423H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIFOCTR	BVAL	BCLR	FRDY	–	–	–	–	DTLN[8:0]								

BVAL	Buffer memory valid flag	R/W
0	Invalid	R/W Note 1
1	Writing ended	
This bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE3 to CURPIPE0 bits (selected pipe). When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB module switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission. • To transmit a short packet, set the BVAL bit to 1 after data has been written. • To transmit a zero-length packet, set the BVAL bit to 1 while the FIFO buffer is empty. When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB module sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission. Writing 1 to the BVAL bit should be done while FRDY is 1 (set by the USB module). When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.		

BCLR	CPU buffer clear	R/W
0	Invalid	R/W Note 2
1	Clears the buffer memory on the CPU side.	
This bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe. When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB module clears only one plane of the FIFO buffer even when both planes are read-enabled. When the selected pipe is the DCP, setting BCLR to 1 allows the USB module to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the PID bits for the DCP control register to NAK before setting BCLR to 1. When the selected pipe is in the transmitting direction, if 1 is written to the BVAL and BCLR bits simultaneously, the USB module clears the data that has been written before it, enabling transmission of a zero-length packet. When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while FRDY in the FIFO port control register is 1 (set by the USB module).		

FRDY	FIFO port ready	R/W
0	FIFO port access is disabled.	R
1	FIFO port access is enabled.	
Indicates whether the FIFO port can be accessed by the CPU. In the following cases, the USB module sets FRDY to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set BCLR to 1 to clear the FIFO buffer, and enable transmission and reception of the next data. • A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty. • A short packet is received and the data is completely read while BFRE in the PIPECFG register is 1.		

Bits 12 to 9	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DTLN[8:0]	Receive data length	R/W
The DTLN[8:0] bits indicate the length of the receive data. While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the RCNT bit value as described below. • RCNT = 0 The USB module sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU has read all the received data from a single FIFO buffer plane. While BFRE in the PIPECFG register is 1, these bits retain the length of the receive data until BCLR is set to 1 even after all the data has been read. • RCNT = 1 The USB module decrements the value indicated by these bits each time data is read from the FIFO buffer. (The value is decremented by one when MBW is 0, and by two when MBW is 1.) The USB module sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB module sets the DTLN[8:0] bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.		R

Notes 1. Only 1 can be written.

2. Only 0 can be read and 1 can be written.

Figure 14-14. Format of DnFIFO Port Control Register (DnFIFOCTR) (n = 0, 1)

Address: F042AH, F042BH (D0FIFOCTR), F042EH, F042FH (D1FIFOCTR) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DnFIFOCTR	BVAL	BCLR	FRDY	–	–	–	–	DTLN[8:0]								

BVAL	Buffer memory valid flag	R/W
0	Invalid	R/W Note 1
1	Writing ended	
This bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE3 to CURPIPE0 bits (selected pipe). When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB module switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission. • To transmit a short packet, set the BVAL bit to 1 after data has been written. • To transmit a zero-length packet, set the BVAL bit to 1 while the FIFO buffer is empty. When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB module sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission. Writing 1 to the BVAL bit should be done while FRDY is 1 (set by the USB module). When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.		

BCLR	CPU buffer clear	R/W
0	Invalid	R/W Note 2
1	Clears the buffer memory on the CPU side.	
This bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe. When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB module clears only one plane of the FIFO buffer even when both planes are read-enabled. When the selected pipe is in the transmitting direction, if 1 is written to the BVAL and BCLR bits simultaneously, the USB module clears the data that has been written before it, enabling transmission of a zero-length packet. When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while FRDY is 1 (set by the USB module).		

FRDY	FIFO port ready	R/W
0	FIFO port access is disabled.	R
1	FIFO port access is enabled.	
Indicates whether the FIFO port can be accessed by the CPU or DMAC. In the following cases, the USB module sets FRDY to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set BCLR to 1 to clear the FIFO buffer, and enable transmission and reception of the next data. • A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty. • A short packet is received and the data is completely read when BFRE is set to 1.		

Bits 12 to 9	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DTLN[8:0]	Receive data length	R/W
The DTLN[8:0] bits indicate the length of the receive data. While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the RCNT bit value as described below. • RCNT = 0 The USB module sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU has read all the received data from a single FIFO buffer plane. While the BFRE bit in the PIPECFG register is 1, these bits retain the length of the receive data until BCLR is set to 1 even after all the data has been read. • RCNT = 1 The USB module decrements the value indicated by these bits each time data is read from the FIFO buffer. (The value is decremented by one when MBW is 0, and by two when MBW is 1.) The USB module sets DTLN to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB module sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.		R/W

Notes 1. Only 1 can be written

2. Only 0 can be read and 1 can be written.

14.3.9 Interrupt enable register 0 (INTENB0)

Figure 14-15. Format of Interrupt Enable Register 0 (INTENB0)

Address: F0430H, F0431H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENB0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	–	–	–	–	–	–	–	–

VBSE	VBUS interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the VBINT interrupt is detected.		

RSME	Resume interrupt enable Note	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the RESM interrupt is detected.		

SOFE	Frame number update interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the SOFR interrupt is detected.		

DVSE	Device state transition interrupt enable Note	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the DVST interrupt is detected.		

CTRE	Control transfer stage transition interrupt enable Note	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the CTRT interrupt is detected.		

BEMPE	Buffer empty interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the BEMP interrupt is detected.		

NRDYE	Buffer Not Ready Response Interrupt Enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the NRDY interrupt is detected.		

BRDYE	Buffer Ready Interrupt Enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the BRDY interrupt is detected.		

Bits 7 to 0	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

Note The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller function is selected; do not set these bits to 1 (interrupt output enabled) when the host controller function is selected.

14.3.10 Interrupt enable register n (INTENBn) (n = 1, 2)

Figure 14-16. Format of Interrupt Enable Register 1 (INTENB1)

Address: F0432H, F0433H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENB1	OVRCRE	BCHGE	–	DTCHE	ATTCHE	–	–	–	–	EOFERRE	SIGNE	SACKE	–	–	–	PDDETINTE

OVRCRE	USB port 0 overcurrent input change interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the OVRCR interrupt is detected.		

BCHGE	USB port 0 USB bus change interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the BCHG interrupt is detected.		

Bit 13	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DTCHE	USB port 0 disconnection detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the DTCH interrupt is detected.		

ATTCHE	USB port 0 connection detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the ATTCH interrupt is detected.		

Bits 10 to 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

EOFERRE	USB port 0 EOF error detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the EOFERR interrupt is detected.		

SIGNE	Setup transaction error interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the SIGN interrupt is detected.		

SACKE	Setup transaction normal response interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the SACK interrupt is detected.		

Bits 3 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PDDETINTE	USB port 0 Portable Device detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the PDDETINT interrupt is detected.		

Caution The bits in INTENB1 can be set to 1 only when the host controller function is selected; do not set these bits to 1 (interrupt output enabled) when the function controller function is selected.

Figure 14-17. Format of Interrupt Enable Register 2 (INTENB2)

Address: F0434H, F0435H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENB2	OVRCRE	BCHGE	–	DTCHE	ATTCHE	–	–	–	–	EOFERRE	–	–	–	–	–	PDDETINTE

OVRCRE	USB port 1 Overcurrent input change interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the OVRCR interrupt is detected.		

BCHGE	USB port 1 USB bus change interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the BCHG interrupt is detected.		

Bit 13	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DTCHE	USB port 1 disconnection detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the DTCH interrupt is detected.		

ATTCHE	USB port 1 connection detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the ATTCH interrupt is detected.		

Bits 10 to 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

EOFERRE	USB port 1 EOF error detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the EOFERR interrupt is detected.		

Bits 5 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PDDETINTE	USB port 1 Portable Device detection interrupt enable	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	
Enables or disables the USB interrupt output when the PDDETINT interrupt is detected.		

Caution The bits in INTENB2 can be set to 1 only when the host controller function is selected; do not set these bits to 1 (interrupt output enabled) when the function controller function is selected.

14.3.11 BRDY interrupt enable register (BRDYENB)

Figure 14-18. Format of BRDY Interrupt Enable Register (BRDYENB)

Address: F0436H, F0437H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRDYENB	–	–	–	–	–	–	–	–	PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	–	–	–	PIPE0BRDYE

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PIPEnBRDYE	BRDY interrupt enable for PIPEn	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	

Remark n = 7 to 4, 0

14.3.12 NRDY interrupt enable register (NRDYENB)

Figure 14-19. Format of NRDY Interrupt Enable Register (NRDYENB)

Address: F0438H, F0439H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDYENB	–	–	–	–	–	–	–	–	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	–	–	–	PIPE0NRDYE

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PIPEnNRDYE	NRDY interrupt enable for PIPEn	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	

Remark n = 7 to 4, 0

14.3.13 BEMP interrupt enable register (BEMPENB)

Figure 14-20. Format of BEMP Interrupt Enable Register (BEMPENB)

Address: F043AH, F043BH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BEMPENB	–	–	–	–	–	–	–	–	PIPE7 BEMPE	PIPE6 BEMPE	PIPE5 BEMPE	PIPE4 BEMPE	–	–	–	PIPE0 BEMPE

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PIPEnBEMPE	BEMP interrupt enable for PIPEn	R/W
0	Interrupt output disabled	R/W
1	Interrupt output enabled	

Remark n = 7 to 4, 0

14.3.14 SOF output configuration register (SOFCFG)

Figure 14-21. Format of SOF Output Configuration Register (SOFCFG)

Address: F043CH, F043DH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOFCFG	–	–	–	–	–	–	–	TRNENSEL	–	BRDYM	–	EDGESTS	–	–	–	–

Bits 15 to 9	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

TRNENSEL	Transaction-enabled time select	R/W
0	For non-low-speed communication	R/W
1	For low-speed communication	
This bit selects, for full-speed or low-speed communication, the transaction-enabled time in which the USB module issues tokens in a frame via the port. The TRNENSEL bit is valid only when the host controller function is selected. This bit should be set to 0 if the function controller function is selected.		

Bit 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

BRDYM	BRDY interrupt status clear timing for each pipe	R/W
0	Software clears the status.	R/W
1	The USB module clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.	
Specifies the timing for clearing the BRDY interrupt status for each pipe.		

Bit 5	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

EDGESTS	Edge interrupt output status monitor **Note**	R/W
0	The edge interrupt output signal is not in the middle of the edge processing.	R
1	The edge interrupt output signal is in the middle of the edge processing.	
Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.		

Bits 3 to 0	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

Note Make sure the EDGESTS bit is 0 when stopping the clock supply to the USB module.

14.3.15 Interrupt status register 0 (INTSTS0)

Figure 14-22. Format of Interrupt Status Register 0 (INTSTS0)

Address: F0440H, F0441H After reset: 00000000 X0000000B

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTSTS0	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ2	DVSQ1	DVSQ0	VALID	CTSQ2	CTSQ1	CTSQ0

VBINT	VBUS interrupt status Note 1	R/W
0	VBUS interrupts are not generated.	R/W Note 2
1	VBUS interrupts are generated.	
The USB module sets the VBINT bit to 1 on detecting a level change (high to low or low to high) in the UV_{BUS} pin input value. The USB module sets the VBSTS bit to indicate the VBUS pin input value. When the UV_{BUS} interrupt is generated, use software to repeat reading the VBSTS bit until the same value is read three or more times, and eliminate chattering.		

RESM	Resume interrupt status Notes 1, 3	R/W
0	Resume interrupts are not generated.	R/W Note 2
1	Resume interrupts are generated.	
When the function controller function is selected, the USB module sets the RESM bit to 1 on detecting the falling edge of the signal on the USB_DP pin in the suspended state (DVSQ2 to DVSQ2 = 1xxB). When the host controller function is selected, the read value is invalid.		

SOFR	Frame number refresh interrupt status	R/W
0	SOF interrupts are not generated.	R/W Note 2
1	SOF interrupts are generated.	
(1) When the host controller function is selected The USB module sets the SOFR bit to 1 on updating the frame number when software has set the UACT bit in DVSTCTR0 to 1. (A frame number refresh interrupt is detected every 1 ms.) (2) When the function controller function is selected The USB module sets the SOFR bit to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.) The USB module can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.		

DVST	Device state transition interrupt status Note 3	R/W
0	Device state transition interrupts are not generated.	R/W Note 2
1	Device state transition interrupts are generated.	
When the function controller function is selected, the USB module updates the DVSQ2 to DVSQ2 value and sets the DVST bit to 1 on detecting a change in the device state. When a device state transition interrupt is generated, clear the status before the USB module detects the next device state transition. When the host controller function is selected, the read value is invalid.		

CTRT	Control transfer stage transition interrupt status Note 3	R/W
0	Control transfer stage transition interrupts are not generated	R/W Note 2
1	Control transfer stage transition interrupts are generated.	
When the function controller function is selected, the USB module updates the CTSQ2 to CTSQ0 value and sets the CTRT bit to 1 on detecting a change in the control transfer stage. When a control transfer stage transition interrupt is generated, clear the status before the USB module detects the next control transfer stage transition. When the host controller function is selected, the read value is invalid.		

BEMP	Buffer empty interrupt status	R/W
0	BEMP interrupts are not generated.	R
1	BEMP interrupts are generated.	
Indicates the BEMP interrupt status. The USB module sets the BEMP bit to 1 when at least one PIPEnBEMP bit in the BEMPSTS register is set to 1 among the PIPEnBEMP bits corresponding to the PIPEnBEMPE bits in the BEMPENB register to which 1 has been set (when the USB module detects the BEMP interrupt status in at least one pipe among the pipes for which software enables the BEMP interrupt output). For the conditions for PIPEnBEMP status assertion, refer to **14.4.3.3 BEMP interrupt**. The USB module clears the BEMP bit to 0 when software writes 0 to all the PIPEnBEMP bits corresponding to the PIPEnBEMPE bits to which 1 has been set. The BEMP bit cannot be cleared to 0 even if software writes 0 to this bit.		

NRDY	Buffer not ready interrupt status	R/W
0	NRDY interrupts are not generated.	R
1	NRDY interrupts are generated.	
Indicates the NRDY interrupt status. The USB module sets the NRDY bit to 1 when at least one PIPEnNRDY bit in NRDYSTS is set to 1 among the PIPEnNRDY bits corresponding to the PIPEnNRDYE bits in NRDYENB to which 1 has been set (when the USB module detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output). For the conditions for PIPEnNRDY status assertion, refer to **14.4.3.2 NRDY interrupt**. The USB module clears the NRDY bit to 0 when software writes 0 to all the PIPEnNRDY bits corresponding to the PIPEnNRDYE bits to which 1 has been set. The NRDY bit cannot be cleared to 0 even if software writes 0 to this bit.		

BRDY	Buffer ready interrupt status	R/W
0	BRDY interrupts are not generated.	R
1	BRDY interrupts are generated.	
Indicates the BRDY interrupt status. The USB module sets the BRDY bit to 1 when at least one PIPEnBRDY bit in the BRDYSTS register is set to 1 among the PIPEnBRDY bits corresponding to the PIPEnBRDYE bits in the BRDYENB register to which 1 has been set (when the USB module detects the BRDY interrupt status in at least one pipe among the pipes for which software enables the BRDY interrupt output). For the conditions for PIPEnBRDY status assertion, refer to **14.4.3.1 BRDY interrupt**. The USB module clears the BRDY bit to 0 when software writes 0 to all the PIPEnBRDY bits corresponding to the PIPEnBRDYE bits to which 1 has been set. The BRDY bit cannot be cleared to 0 even if software writes 0 to this bit.		

VBSTS	VBUS interrupt status Note 4	R/W
0	UV_{BUS} pin is low.	R
1	UV_{BUS} pin is high.	

DVSQ2	DVSQ1	DVSQ0	Device state	R/W
0	0	0	Powered state	R
0	0	1	Default state	
0	1	0	Address state	
0	1	1	Configured state	
1	X	X	Suspended state	
These bits indicate the device status. When the host controller function is selected, the read value is invalid.				

VALID	USB request reception	R/W
0	Not detected	R/W **Note 2**
1	Setup packet reception	
Indicates the USB request reception status. When the host controller function is selected, the read value is invalid.		

CTSQ2	CTSQ1	CTSQ0	Control transfer stage	R/W
0	0	0	Idle or setup stage	R
0	0	1	Control read data stage	
0	1	0	Control read status stage	
0	1	1	Control write data stage	
1	0	0	Control write status stage	
1	0	1	Control write (no data) status stage	
1	1	0	Control transfer sequence error	
1	1	1	Do not set.	
These bits indicate the control transfer stage status. When the host controller function is selected, the read value is invalid.				

Notes 1. A change in the status indicated by the VBINT and RESM bits can be detected even while the clock supply is stopped (SCKE = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.

2. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bit, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

3. A change in the status of the RESM, DVST, and CTRT bits occur only when the function controller function is selected; set the corresponding interrupt enable bits to 0 (disabled) when the host controller function is selected.

4. The value after reset depends on the value of the UV_{BUS} pin. This bit is 1 when the UV_{BUS} pin input is high level and 0 when the input is low level.

14.3.16 Interrupt status register n (INTSTSn) (n = 1, 2)

Figure 14-23. Format of Interrupt Status Register 1 (INTSTS1)

Address: F0442H, F0443H After reset: XX0X0000 00000000B

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTSTS1	OVRCR	BCHG	–	DTCH	ATTCH	–	–	–	–	EOFERR	SIGN	SACK	–	–	–	PDDETINT

OVRCR	USB port 0 overcurrent input change interrupt status Note 1	R/W
0	OVRCR interrupts are not generated.	R/W Note 2
1	OVRCR interrupts are not generated.	
Indicates the status of the UOVRCUR0 input pin change interrupt. The USB module detects the OVRCR interrupt when a change (high to low or low to high) occurs in at least one of the input values to the UOVRCUR0 pin, and sets the OVRCR bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt.		

BCHG	USB port 0 USB bus change interrupt status Note 1	R/W
0	BCHG interrupts are not generated.	R/W Note 2
1	BCHG interrupts are generated.	
Indicates the status of the USB bus change interrupt. The USB module detects the BCHG interrupt when a change in the full-speed/low-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. The USB module sets the LNST bits in the SYSSTS0 register to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST bits until the same value is read three or more times, and eliminate chattering. A change in the USB bus state can be detected even while the internal clock supply is stopped. When the function controller function is selected, the read value is invalid.		

Bit 13	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DTCH	USB port 0 USB disconnection detection interrupt status	R/W
0	DTCH interrupts are not generated.	R/W Note 2
1	DTCH interrupts are generated.	
Indicates the status of the USB disconnection detection interrupt when the host controller function is selected. The USB module detects the disconnection detection interrupt on detecting USB bus disconnection, and sets the DTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. The USB module detects bus disconnection based on the USB Specification 2.0. After detecting the disconnection detection interrupt, the USB module controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried out for the USB port and make a transition to the wait state for bus connection to the USB port (wait state for connection detection interrupt generation). • Modifies the UACT bit for the port in which a disconnection detection interrupt has been detected to 0. • Puts the port in which a disconnection detection interrupt has been generated into the idle state. When the function controller function is selected, the read value is invalid.		

ATTCH	USB port 0 connection detection interrupt status	R/W
0	Connection detection interrupts are not generated.	R/W Note 2
1	Connection detection interrupts are generated.	
Indicates the status of the ATTCH interrupt when the host controller function is selected. The USB module detects the ATTCH interrupt on detecting J-state or K-state of the full-speed signal level for 2.5 μs, and sets the ATTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. Specifically, the USB module detects the connection detection interrupt on any of the following conditions. • K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 μs. • J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 μs. When the function controller function is selected, the read value is invalid.		

Bits 10 to 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

EOFERR	USB port 0 EOF error detection interrupt status	R/W
0	EOFERR interrupts are not generated.	R/W Note 2
1	EOFERR interrupts are generated.	
Indicates the status of the EOFERR interrupt when the host controller function is selected. The USB module detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by the USB Specifications 2.0, and sets the EOFERR bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the EOFERR interrupt. After detecting the EOFERR interrupt, the USB module controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried for the USB port and perform re-enumeration of the USB port. • Modifies the UACT bit for the port in which an EOFERR interrupt has been detected to 0. • Puts the port in which an EOFERR interrupt has been generated into the idle state. When the function controller function is selected, the read value is invalid.		

SIGN	Setup transaction error interrupt status	R/W
0	SIGN interrupts are not generated.	R/W Note 2
1	SIGN interrupts are generated.	
Indicates the status of the setup transaction error interrupt when the host controller function is selected. The USB module detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets the SIGN bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the SIGN interrupt. Specifically, the USB module detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions. • Timeout is detected by the USB module when the peripheral device has returned no response. • A damaged ACK packet is received. • A handshake other than ACK (NAK, NYET, or STALL) is received. When the function controller function is selected, the read value is invalid.		

SACK	Setup transaction normal response interrupt status	R/W
0	SACK interrupts are not generated.	R/W Note 2
1	SACK interrupts are generated.	
Indicates the status of the setup transaction normal response interrupt when the host controller function is selected. The USB module detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB module, and sets the SACK bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the SACK interrupt. When the function controller function is selected, the read value is invalid.		

Bits 3 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PDDETINT	USB port 0 Portable Device detection interrupt status	R/W
0	PDDETINT interrupts are not generated.	R/W **Note 2**
1	PDDETINT interrupts are generated.	
Indicates the status of the Portable Device detection interrupt when the host controller function is selected. The USB module detects when a change (high to low or low to high) occurs in the input value to the USB transceiver VDPDET pin, and sets this bit to 1. The USB module indicates the input value to the USB transceiver VDPDET pin to the PDDETSTS bit. When the PDDETINT interrupt is generated, use software to repeat reading the PDDETSTS bit until the same value is read three or more times, and eliminate chattering.		

Notes 1. A change in the status indicated by the OVRCR or BCHG bit can be detected even while the clock supply is stopped (SCKE = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after enabling the clock supply. No interrupts other than those indicated by the BCHG and OVRCR bits can be detected while the clock supply is stopped (SCKE = 0).

2. To clear the status indicated by the bits in the INTSTS1 register, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Figure 14-24. Format of Interrupt Status Register 2 (INTSTS2)

Address: F0444H, F0445H After reset: X00X0000 00000000B

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTSTS2	OVRCR	BCHG	–	DTCH	ATTCH	–	–	–	–	EOFERR	–	–	–	–	–	PDDETINT

OVRCR	USB port 1 overcurrent input change interrupt status Note 1	R/W
0	OVRCR interrupts are not generated.	R/W Note 2
1	OVRCR interrupts are not generated.	
Indicates the status of the UOVRCUR1 input pin change interrupt. The USB module detects the overcurrent interrupt when a change (high to low or low to high) occurs in at least one of the input values to the UOVRCUR1 pin, and sets the OVRCR bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt.		

BCHG	USB port 1 USB bus change interrupt status Note 1	R/W
0	BCHG interrupts are not generated.	R/W Note 2
1	BCHG interrupts are generated.	
Indicates the status of the USB bus change interrupt. The USB module detects the BCHG interrupt when a change in the full-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. The USB module sets the LNST bits in SYSSTS1 to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST bits until the same value is read three or more times, and eliminate chattering. A change in the USB bus state can be detected even while the internal clock supply is stopped. When the function controller function is selected, the read value is invalid.		

Bit 13	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DTCH	USB port 1 disconnection detection interrupt status	R/W
0	DTCH interrupts are not generated.	R/W Note 2
1	DTCH interrupts are generated.	
Indicates the status of the USB disconnection detection interrupt when the host controller function is selected. The USB module detects the disconnection detection interrupt on detecting USB bus disconnection, and sets the DTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. The USB module detects bus disconnection based on the USB Specifications 2.0. After detecting the disconnection detection interrupt, the USB module controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried out for the USB port and make a transition to the wait state for bus connection to the USB port (wait state for connection detection interrupt generation). • Modifies the UACT bit for the port in which a disconnection detection interrupt has been detected to 0. • Puts the port in which a disconnection detection interrupt has been generated into the idle state. When the function controller function is selected, the read value is invalid.		

ATTCH	USB port 1 connection detection interrupt status	R/W
0	Connection detection interrupts are not generated.	R/W Note 2
1	Connection detection interrupts are generated.	
Indicates the status of the ATTCH interrupt when the host controller function is selected. The USB module detects the ATTCH interrupt on detecting J-state or K-state of the full-speed signal level for 2.5 μs, and sets the ATTCH bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the interrupt. Specifically, the USB module detects the connection detection interrupt on any of the following conditions. • K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 μs. • J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 μs. When the function controller function is selected, the read value is invalid.		

Bits 10 to 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

EOFERR	USB port 1 EOF error detection interrupt status	R/W
0	EOFERR interrupts are not generated.	R/W Note 2
1	EOFERR interrupts are generated.	
Indicates the status of the EOFERR interrupt when the host controller function is selected. The USB module detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by the USB Specifications 2.0, and sets the EOFERR bit to 1. Here, if software has set the corresponding interrupt enable bit to 1, the USB module generates the EOFERR interrupt. After detecting the EOFERR interrupt, the USB module controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). Software should terminate all the pipes in which communications are currently carried for the USB port and perform re-enumeration of the USB port. • Modifies the UACT bit for the port in which an EOFERR interrupt has been detected to 0. • Puts the port in which an EOFERR interrupt has been generated into the idle state. When the function controller function is selected, the read value is invalid.		

Bits 5 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PDDETINT	USB port 1 portable Device detection interrupt status	R/W
0	PDDETINT interrupts are not generated.	R/W Note 2
1	PDDETINT interrupts are generated.	
Indicates the status of the Portable Device detection interrupt when the host controller function is selected. The USB module detects when a change (high to low or low to high) occurs in the input value to the USB transceiver VDPDET pin, and sets this bit to 1. The USB module indicates the input value to the USB transceiver VDPDET pin to the PDDETSTS bit. When the PDDETINT interrupt is generated, use software to repeat reading the PDDETSTS bit until the same value is read three or more times, and eliminate chattering.		

Notes 1. A change in the status indicated by the OVRCR or BCHG bit can be detected even while the clock supply is stopped (SCKE = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after enabling the clock supply. No interrupts other than those indicated by the BCHG and OVRCR bits can be detected while the clock supply is stopped (SCKE = 0).

2. To clear the status indicated by the bits in the INTSTS2 register, write 0 only to the bits to be cleared; write 1 to the other bits.

14.3.17 BRDY interrupt status register (BRDYSTS)

Figure 14-25. Format of BRDY Interrupt Status Register (BRDYSTS)

Address: F0446H, F0447H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRDYSTS	–	–	–	–	–	–	–	–	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	–	–	–	PIPE0 BRDY

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PIPEnBRDY	BRDY interrupt status for PIPEn **Note 1**	R/W
0	Interrupts are not generated.	R/W **Note 2**
1	Interrupts are generated.	

Notes 1. When BRDYM in SOFCGFG is 0, clearing BRDY interrupts should be done before accessing the FIFO.

2. When BRDYM in SOFCGFG is 0, to clear the status indicated by the bits in BRDYSTS, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Remark n = 7 to 4, 0

14.3.18 NRDY interrupt status register (NRDYSTS)

Figure 14-26. Format of NRDY Interrupt Status Register (NRDYSTS)

Address: F0448H, F0449H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRDYSTS	–	–	–	–	–	–	–	–	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	–	–	–	PIPE0 NRDY

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PIPEnNRDY	NRDY interrupt status for PIPEn	R/W
0	Interrupts are not generated.	R/W **Note**
1	Interrupts are generated.	

Note To clear the status indicated by the bits in the NRDYSTS register, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Remark n = 7 to 4, 0

14.3.19 BEMP interrupt status register (BEMPSTS)

Figure 14-27. Format of BEMP Interrupt Status Register (BEMPSTS)

Address: F044AH, F044BH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BEMPSTS	–	–	–	–	–	–	–	–	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	–	–	–	PIPE0 BEMP

Bits 15 to 8, 3 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PIPEnBEMP	BEMP interrupt status for PIPEn	R/W
0	Interrupts are not generated.	R/W **Note**
1	Interrupts are generated.	

Note To clear the status indicated by the bits in the BEMPSTS register, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Remark n = 7 to 4, 0

14.3.20 Frame number register (FRMNUM)

Figure 14-28. Format of Frame Number Register (FRMNUM)

Address: F044CH, F044DH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRMNUM	–	–	–	–	–	FRNM[10:0]										

Bits 15 to 11	Reserved	R/W
–	The write value must be 0. The read value is 0.	–

FRNM[10:0]	Frame number	R/W
The USB module sets the FRNM[10:0] bits to indicate the latest frame number, which is updated every time an SOF packet is issued or received (every 1 ms). Repeat reading the FRNM[10:0] bits until the same value is read twice.		R

14.3.21 USB address register (USBADDR)

Figure 14-29. Format of USB Address Register (USBADDR)

Address: F0450H, F0451H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBADDR	–	–	–	–	–	–	–	–	–	USBADDR[6:0]						

Bits 15 to 12	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

Bits 11 to 8	Reserved	R/W
–	The write value must be 0. The read value is 0.	–

Bit 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

USBADDR[6:0]	USB address	R/W
The current USB address value can be read. This register is not used while in the host mode.		R

14.3.22 USB request type register (USBREQ)

Figure 14-30. Format of USB Request Type Register (USBREQ)

Address: F0454H, F0455H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBREQ	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							

BREQUEST[7:0]	Request	R/W
These bits store the USB request bRequest value. [When the host controller function is selected] The USB request data value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ is 1. [When the function controller function is selected] These bits indicate the USB request data value received during the setup transaction. Writing to these bits is invalid.		R/W **Note**

BMREQUESTTYPE[7:0]	Request type	R/W
These bits store the USB request bmRequestType value. [When the host controller function is selected] The USB request data value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ is 1. [When the function controller function is selected] These bits indicate the USB request data value received during the setup transaction. Writing to these bits is invalid.		R/W **Note**

Note When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

14.3.23 USB request value register (USBVAL)

Figure 14-31. Format of USB Request Value Register (USBVAL)

Address: F0456H, F0457H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBVAL	WVALUE[15:0]															

WVALUE[15:0]	Value	R/W
These bits store the USB request wValue value. [When the host controller function is selected] The USB request wValue value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ is 1. [When the function controller function is selected] These bits indicate the USB request wValue value received during the setup transaction. Writing to these bits is invalid.		R/W **Note**

Note When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

14.3.24 USB request index register (USBINDX)

Figure 14-32. Format of USB Request Index Register (USBINDX)

Address: F0458H, F0459H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBINDX	WINDEX[15:0]															

WINDEX[15:0]	Index	R/W
	These bits store the USB request wIndex value. [When the host controller function is selected] The USB request wIndex value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ in the DCPCTR register is 1. [When the function controller function is selected] These bits indicate the USB request wIndex value received during the setup transaction. Writing to these bits is invalid.	R/W **Note**

Note When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

14.3.25 USB request length register (USBLENG)

Figure 14-33. Format of USB Request Length Register (USBLENG)

Address: F045AH, F045BH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBLENG	WLENGTH[15:0]															

WLENGTH[15:0]	Length	R/W
	These bits store the USB request wLength value. [When the host controller function is selected] The USB request wLength value for the setup transaction to be transmitted should be set in these bits. Do not modify these bits while SUREQ in the DCPCTR register is 1. [When the function controller function is selected] These bits indicate the USB request wLength value received during the setup transaction. Writing to these bits is invalid.	R/W **Note**

Note When the function controller function is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller function is selected, these bits can be read from and written to.

14.3.26 DCP configuration register (DCPCFG)

Figure 14-34. Format of DCP Configuration Register (DCPCFG)

Address: F045CH, F045DH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCPCFG	–	–	–	–	–	–	–	–	SHTNAK	–	–	DIR	–	–	–	–

Bits 15 to 8	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

SHTNAK	Pipe disabled at end of transfer Note	R/W
0	Pipe continued at the end of transfer	R/W
1	Pipe disabled at the end of transfer	
Specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction. The SHTNAK bit is valid when the selected pipe in the receiving direction. When the SHTNAK bit is set to 1, the USB module modifies the PID bits for the DCP to NAK on determining the end of the transfer. The USB module determines that the transfer has ended on the following condition. • A short packet (including a zero-length packet) is successfully received.		

Bits 6, 5	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DIR	Transfer direction Note	R/W
0	Data receiving direction	R/W
1	Data transmitting direction	
When the host controller function is selected, the DIR bit sets the transfer direction of the data stage and status stage. When the function controller function is selected, the DIR bit should be set to 0.		

Bits 3 to 0	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

Note Modify these bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

14.3.27 DCP maximum packet size register (DCPMAXP)

Figure 14-35. Format of DCP Maximum Packet Size Register (DCPMAXP)

Address: F045EH, F045FH After reset: 0040H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCPMAXP	–	DEVSEL2	DEVSEL1	DEVSEL0	–	–	–	–	–	MXPS[6:0]						

DEVSEL2	DEVSEL1	DEVSEL0	Device select Note 1	R/W
0	0	0	USB address 000	R/W
0	0	1	USB address 001	
0	1	0	USB address 010	
0	1	1	USB address 011	
1	0	0	USB address 100	
1	0	1	USB address 101	
Other than above			Do not set.	
When the host controller function is selected, the DEVSEL2 to DEVSEL0 bits specify the address of the peripheral device which is the communication target during control transfer. The DEVSEL2 to DEVSEL0 bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in these bits. For example, before setting DEVSEL2 to DEVSEL0 = 010B, the address should be set to DEVADD2. When the function controller function is selected, these bits should be set to 000B.				

Bits 11 to 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

MXPS[6:0]	Maximum packet size Note 2	R/W
These bits specify the maximum data payload (maximum packet size) for the DCP. The initial value is 40H (64 bytes). These bits should be set to the value based on the USB Specifications. While the MXPS bits are 0, do not write to the FIFO buffer or do not set PID to BUF.		R/W

Notes 1. Modify the DEVSEL bits while PID is NAK and the SUREQ bit is 0. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

2. Modify the MXPS[6:0] bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary. After modifying the MXPS[6:0] bits and the DCP has been set to the CURPIPE bits, clear the buffer by setting BCLR to 1.

14.3.28 DCP control register (DCPCTR)

Figure 14-36. Format of DCP Control Register (DCPCTR)

Address: F0460H, F0461H After reset: 0040H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCPCTR	BSTS	SUREQ	–	–	SUREQCLR	–	–	SQCLR	SQSET	SQMON	PBUSY	–	–	CCPL	PID1	PID0

BSTS	Buffer status	R/W
0	Buffer access is enabled.	R
1	Buffer access is disabled.	
Indicates whether DCP FIFO buffer access is enabled or disabled. The meaning of the BSTS bit depends on the ISEL bit setting as follows. • When ISEL is 0, the BSTS bit indicates whether the received data can be read from the buffer. • When ISEL is 1, the BSTS bit indicates whether the data to be transmitted can be written to the buffer.		

SUREQ	Setup token transmission	R/W
0	Invalid	R/W **Note 1**
1	Transmits the setup packet.	
The USB module transmits the setup packet by setting the SUREQ bit to 1 when the host controller function is selected. After completing the setup transaction process, the USB module generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0. The USB module also clears the SUREQ bit to 0 when software sets the SUREQCLR bit to 1. Before setting the SUREQ bit to 1, set the DEVSEL bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the desired USB request in the setup transaction. Before setting SUREQ to 1, check that the PID bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not modify the DEVSEL bits, the USBREQ, USBVAL, USBINDX, or USBLENG register until the setup transaction is completed (SUREQ = 1). Write 1 to the SUREQ bit only when transmitting the setup token; for other purposes, write 0. When the function controller function is selected, be sure to write 0 to the SUREQ bit.		

Bits 13, 12	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

SUREQCLR	SUREQ bit clear	R/W
0	Invalid	R/W **Note 3**
1	0 is written to the SUREQ bit.	
When the host controller function is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit always indicates 0. Set the SUREQCLR bit to 1 through software when communication has stopped with SUREQ being 1 during the setup transaction. However, for normal setup transactions, the USB module automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit through software is not necessary. Controlling the SUREQ bit through the SUREQCLR bit must be done while UACT is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected. When the function controller function is selected, be sure to write 0 to the SUREQCLR bit.		

Bits 10, 9	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

SQCLR	Toggle bit clear Note 2	R/W
0	Invalid	R/W Note 3
1	Specifies DATA0.	
Specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit always indicates 0. Do not set the SQCLR and SQSET bits to 1 simultaneously.		

SQSET	Toggle bit set Note 2	R/W
0	Invalid	R/W Note 3
1	Specifies DATA1.	
Specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. Do not set the SQCLR and SQSET bits to 1 simultaneously.		

SQMON	Sequence toggle bit monitor	R/W
0	DATA0	R
1	DATA1	
Indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The USB module allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction. When the function controller function is selected, the USB module sets the SQMON bit to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet. When the function controller function is selected, the USB module does not reference the SQMON bit during the IN/OUT transaction of the status stage, and does not allow the SQMON bit to toggle upon normal completion.		

PBUSY	Pipe busy	R/W
0	DCP is not used for the transaction.	R
1	DCP is used for the transaction.	
Indicates whether DCP is used or not for the transaction when USB changes the PID bits from BUF to NAK. The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction. Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible. For details, refer to **14.4.4.1 Pipe control register switching procedures**.		

Bits 4, 3	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

CCPL	Control transfer end enable	R/W
0	Completion of control transfer is disabled.	R/W
1	Completion of control transfer is enabled.	
When the function controller function is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed. When software sets the CCPL bit to 1 while the corresponding PID bits are set to BUF, the USB module completes the control transfer stage. Specifically, during control read transfer, the USB module transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, the USB module operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit. The USB module modifies the CCPL bit from 1 to 0 on receiving a new setup packet. Software cannot write 1 to the CCPL bit while VALID is 1. When the host controller function is selected, be sure to write 0 to the CCPL bit.		

PID1	PID0	Response PID	R/W
0	0	NAK response	R/W
0	1	BUF response (depending on the buffer state)	
1	0	STALL response	
1	1	STALL response	
The PID1 and PID0 bits control the response type of the USB module during control transfer. [When the host controller function is selected] Modify the setting of the PID1 and PID0 bits from NAK to BUF using the following procedure. • When the transmitting direction is set Write all the transmit data to the FIFO buffer while UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, the USB module executes the OUT transaction. • When the receiving direction is set Check that the FIFO buffer is empty (or empty the buffer) while UACT is 1 and PID is NAK, and then set PID to BUF. After PID has been set to BUF, the USB module executes the IN transaction. The USB module modifies the setting of the PID1 and PID0 bits as follows. • The USB module sets PID to STALL (11B) on receiving the data of a size exceeding the maximum packet size when software has set the PID1 and PID0 bits to BUF. • The USB module sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times. • The USB module also sets PID to STALL (11B) on receiving the STALL handshake. [When the function controller function is selected] The USB module modifies the setting of the PID1 and PID0 bits as follows. • The USB module modifies the PID1 and PID0 bits to NAK on receiving the setup packet. Here, the USB module sets VALID to 1. Software cannot modify the setting of the PID1 and PID0 bits until software sets VALID to 0. • The USB module sets PID to STALL (11B) on receiving the data of a size exceeding the maximum packet size when software has set the PID1 and PID0 bits to BUF. • The USB module sets PID to STALL (1xB) on detecting the control transfer sequence error. • The USB module sets PID to NAK on detecting the USB bus reset. The USB module does not reference to the setting of the PID1 and PID0 bits while the SET_ADDRESS request is processed (auto processing).			

Notes 1. Only 1 can be written.

2. Write 1 to the SQSET and SQCLR bits while PID is NAK. Before modifying these bits after modifying the PID bits for the DCP from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

3. This bit is always read as 0. Only 1 can be written.

14.3.29 Pipe window select register (PIPESEL)

Figure 14-37. Format of Pipe Window Select Register (PIPESEL)

Address: F0464H, F0465H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPESEL	–	–	–	–	–	–	–	–	–	–	–	–	PIPESEL3	PIPESEL2	PIPESEL1	PIPESEL0

Bits 15 to 4	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PIPESEL3	PIPESEL2	PIPESEL1	PIPESEL0	Pipe window select	R/W
0	0	0	0	No pipe selected	R/W
0	1	0	0	PIPE4	
0	1	0	1	PIPE5	
0	1	1	0	PIPE6	
0	1	1	1	PIPE7	
Other than above				Do not set.	
The PIPESEL3 to PIPESEL0 bits select the pipe number corresponding to the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers which data are written to or read from. Selecting a pipe number through these bits allows writing to and reading from the PIPECFG, PIPEMAXP, and PIPEPERI registers which correspond to the selected pipe number. When the PIPESEL3 to PIPESEL0 bits are set to 0000B, 0 is read from all of the bits in the PIPECFG, PIPEMAXP, and PIPEPERI registers. Writing to these bits is invalid.					

Caution After selecting the pipe using the PIPESEL register, functions of the pipe should be set using the PIPECFG, PIPEMAXP, and PIPEPERI registers. The PIPEnCTR, PIPEnTRE, and PIPEnTRN registers can be set regardless of the pipe selection in the PIPESEL register.

14.3.30 Pipe configuration register (PIPECFG)

Figure 14-38. Format of Pipe Configuration Register (PIPECFG)

Address: F0468H, F0469H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPECFG	TYPE1	TYPE0	–	–	–	BFRE	DBLB	–	SHTNAK	–	–	–	DIR	EPNUM[3:0]		

TYPE1	TYPE0	Transfer type Note 1	R/W
PIPE4, PIPE5			R/W
0	0	Pipe not used	
0	1	Bulk transfer	
1	0	Do not set.	
1	1	Do not set.	
PIPE6, PIPE7			
0	0	Pipe not used	
0	1	Do not set.	
1	0	Interrupt transfer	
1	1	Do not set.	
These bits select the transfer type for the pipe selected by the PIPESEL3 to PIPESEL0 bits (selected pipe). Before setting PID to BUF for the selected pipe (before starting USB communication using the selected pipe), be sure to set the TYPE1 and TYPE0 bits to a value other than 00B.			

Bits 13 to 11	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

BFRE	BRDY interrupt operation specification Notes 2, 3	R/W
0	BRDY interrupt upon transmitting or receiving data	R/W
1	BRDY interrupt upon completion of reading data	
Specifies the BRDY interrupt generation timing from the USB module to the CPU with respect to the selected pipe. When software has set the BFRE bit to 1 and the selected pipe is in the receiving direction, the USB module detects the transfer completion and generates the BRDY interrupt on having read the relevant packet. When the BRDY interrupt is generated with the above conditions, software needs to write 1 to the BCLR bit. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit. When software has set the BFRE bit to 1 and the selected pipe is in the transmitting direction, the USB module does not generate the BRDY interrupt. For details, refer to **14.4.3.1 BRDY interrupt**.		

DBLB	Double buffer mode Notes 2, 3	R/W
0	Single buffer	R/W
1	Double buffer	
Selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The DBLB bit is valid when PIPE4 and PIPE5 are selected.		

Bit 8	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

SHTNAK	Pipe disabled at end of transfer Note 1	R/W
0	Pipe continued at the end of transfer	R/W
1	Pipe disabled at the end of transfer	
Specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction. The SHTNAK bit is valid when the selected pipe is PIPE4 and PIPE5 in the receiving direction. When software has set the SHTNAK bit to 1 for the selected pipe in the receiving direction, the USB module modifies the PID bits corresponding to the selected pipe to NAK on determining the end of the transfer. The USB module determines that the transfer has ended on any of the following conditions. • A short packet (including a zero-length packet) is successfully received. • The transaction counter is used and the number of packets specified by the counter are successfully received.		

Bits 6, 5	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DIR	Transfer direction Notes 2, 3	R/W
0	Receiving direction	R/W
1	Transmitting direction	
Specifies the transfer direction for the selected pipe. When software has set the DIR bit to 0, the USB module uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB module uses the selected pipe in the transmitting direction.		

EPNUM[3:0]	Endpoint number Note 1	R/W
These bits specify the endpoint number for the selected pipe. Setting 0000B means unused pipe. During function controller operation, do not allow the combination of the settings of the DIR and EPNUM bits to be the same for two or more pipes (EPNUM = 0000B can be set for all of the pipes).		R/W

Notes 1. Modify the TYPE1 and TYPE0, SHTNAK, and EPNUM bits while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

2. Modify the BFRE, DBLB, and DIR bits while PID is NAK and before the pipe is selected by the CURPIPE3 to CURPIPE0 bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

3. To modify the BFRE, DBLB, and DIR bits after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE bits are in the state described in the above note 2.

14.3.31 Pipe maximum packet size register (PIPEMAXP)

Figure 14-39. Format of Pipe Maximum Packet Size Register (PIPEMAXP)

Address: F046CH, F046DH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEMAXP	–	DEVSEL2	DEVSEL1	DEVSEL0	–	–	–	MXPS[8:0]								

Bit 15	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DEVSEL2	DEVSEL1	DEVSEL0	Device select Note 1	R/W
0	0	0	USB address 000	R/W
0	0	1	USB address 001	
0	1	0	USB address 010	
0	1	1	USB address 011	
1	0	0	USB address 100	
1	0	1	USB address 101	
Other than above			Do not set.	
When the host controller function is selected, the DEVSEL2 to DEVSEL0 bits specify the USB device address of the peripheral device which is the communication target. The DEVSEL2 to DEVSEL0 bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in these bits. For example, before setting the DEVSEL bits to 010B, the address should be set to DEVADD2. When the function controller function is selected, these bits should be set to 000B.				

Bits 11 to 9	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

MXPS[8:0]	Maximum packet size Note 2	R/W
These bits specify the maximum data payload (maximum packet size). The range of values can be set for each pipe is shown as follows: PIPE4, PIPE5: 8 bytes (008H), 16 bytes (010H), 32 bytes (020H), 64 bytes (040H) (Bits [8:7] and [2:0] are not provided.) PIPE6, PIPE7: 1 byte (001H) to 64 bytes (040H) (Bits [8:7] are not provided.) The MXPS bits should be set to the appropriate value for each transfer type based on the USB Specifications. While the MXPS bits are 0, do not write to the FIFO buffer or set PID to BUF.		R/W

Notes 1. Modify the DEVSEL bits while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

2. Modify the MXPS bits while PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Caution The value of the PIPEMAXP register after reset differs depending on when no pipe is selected and when a pipe is selected with the PIPESEL3 to PIPESEL0 bits in the PIPESEL register. The value after reset is 0000H when no pipe is selected and 0040H when a pipe is selected.

14.3.32 Pipe cycle control register (PIPEPERI)

Figure 14-40. Format of Pipe Cycle Control Register (PIPEPERI)

Address: F046EH, F046FH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEPERI	–	–	–	–	–	–	–	–	–	–	–	–	–	IITV[2:0]		

Bits 15 to 13	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

Bit 12	Reserved	R/W
–	The write value must be 0. The read value is 0.	–

Bits 11 to 3	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

IITV[2:0]	Interval error detection interval **Note**	R/W
Specifies the interval error detection timing for the selected pipe in terms of frames, which is expressed as n-th power of 2. For details on the functions, refer to **14.4.8 Interrupt transfers (PIPE6, PIPE7)**. Before modifying the IITV[2:0] bits after USB communication has been completed with the IITV[2:0] bits set to a certain value, set PID to NAK and then set ACLRM to 1 to initialize the interval timer. The IITV[2:0] bits are invalid for PIPE4 and PIPE5. Set these bits to 000B for PIPE4 and PIPE5.		R/W

Note Modify the IITV bits while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

14.3.33 PIPEn control registers (PIPEnCTR) (n = 4 to 7)

Figure 14-41. Format of PIPEn Control Registers (PIPEnCTR) (n = 4, 5)

Address: F0476H, F0477H (PIPE4CTR), F0478H, F0479H (PIPE5CTR) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEnCTR	BSTS	INBUFM	–	–	–	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	–	–	–	PID1	PID0

BSTS	Buffer status	R/W
0	Buffer access by the CPU is disabled.	R
1	Buffer access by the CPU is enabled.	
Indicates the FIFO buffer status for the relevant pipe. The meaning of the BSTS bit depends on the settings of the DIR, BFRE, and DCLRM bits as shown in Table 14-8.		

INBUFM	Transmit buffer monitor	R/W
0	There is no data to be transmitted in the buffer memory.	R
1	There is data to be transmitted in the buffer memory.	
Indicates the relevant FIFO buffer status when the relevant pipe is in the transmitting direction. When the relevant pipe is in the transmitting direction (DIR = 1), the USB module sets the INBUFM bit to 1 when software (or DMA) completes writing data to at least one FIFO buffer plane. The USB module sets the INBUFM bit to 0 when the USB module completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (DBLB = 1), the USB module sets the INBUFM bit to 0 when the USB module completes transmitting the data from the two FIFO buffer planes before software (or DMAC) completes writing data to one FIFO buffer plane. The INBUFM bit indicates the same value as the BSTS bit when the relevant pipe is in the receiving direction (DIR = 0).		

Bits 13 to 11	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

ATREPM	Auto response mode Note 1	R/W
0	Auto response is disabled.	R/W
1	Auto response is enabled.	
Enables or disables auto response mode for the relevant pipe. When the function controller function is selected and the relevant pipe is for bulk transfer, the ATREPM bit can be set to 1. When the ATREPM bit is set to 1, the USB module responds to the token from the USB host as described below. [When the relevant pipe is for bulk IN transfer (TYPE bits = 01B and DIR = 1)] When ATREPM is 1 and PID is BUF, the USB module transmits a zero-length packet in response to the IN token. The USB module updates (allows toggling of) the sequence toggle bit (DATA-PID) each time the USB module receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.). In this case, the USB module does not generate the BRDY or BEMP interrupt. [When the relevant pipe is for bulk OUT transfer (TYPE bits = 01B and DIR = 0)] When ATREPM is 1 and PID is BUF, the USB module returns NAK in response to the OUT token and generates the NRDY interrupt. For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. When the host controller function is selected, be sure to set the ATREPM bit to 0.		

ACLRM	Auto buffer clear mode Note 2	R/W
0	Disabled	R/W
1	Enabled (all buffers are initialized)	
Enables or disables auto buffer clear mode for the relevant pipe. To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously. Table 14-9 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.		

SQCLR	Toggle bit clear Note 1	R/W
0	Invalid	R/W Note 3
1	Specifies DATA0.	
The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQCLR bit to 1 through software allows the USB module to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQCLR bit to 0.		

SQSET	Toggle bit set Note 1	R/W
0	Invalid	R/W Note 3
1	Specifies DATA1.	
The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQSET bit to 1 through software allows the USB module to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQSET bit to 0.		

SQMON	Toggle bit monitor	R/W
0	DATA0	R
1	DATA1	
Indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe. The USB module allows the SQMON bit to toggle upon normal completion of the transaction of the relevant pipe. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.		

PBUSY	Pipe busy	R/W
0	The relevant pipe is not used for the transaction.	R
1	The relevant pipe is used for the transaction.	
Indicates whether the relevant pipe is being currently used or not for the transaction. The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction. Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible. For details, refer to **14.4.4.1 Pipe control register switching procedures**.		

Bits 4 to 2	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PID1	PID0	Response PID	R/W
0	0	NAK response	R/W
0	1	BUF response (depending on the buffer state)	
1	0	STALL response	
1	1	STALL response	

The PID1 and PID0 bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID1 and PID0 bits is NAK. Modify the setting of these bits to BUF to use the relevant pipe for USB transfer. Tables 14-10 and 14-11 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB module depending on the PID bit setting.
After modifying the setting of the PID1 and PID0 bits through software from BUF to NAK during USB communication using the relevant pipe, check that PBUSY is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state.
The USB module modifies the setting of the PID1 and PID0 bits in the following cases.

- The USB module sets PID to NAK on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and software has set the SHTNAK bit for the selected pipe to 1.
- The USB module sets PID to STALL (11B) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB module sets PID to NAK on detecting a USB bus reset when the function controller function is selected.
- The USB module sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller function is selected.
- The USB module sets PID to STALL (11B) on receiving the STALL handshake when the host controller function is selected.

To specify each response type, set the PID1 and PID0 bits as follows.

- To make a transition from NAK (00B) to STALL, write 10B.
- To make a transition from BUF (01B) to STALL, write 11B.
- To make a transition from STALL (11B) to NAK, write 10B and then 00B.
- To make a transition from STALL to BUF, write 00B (NAK) and then 01B (BUF).

Notes 1. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

2. Modify the ATREPM bit while PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

3. Only 0 can be read and 1 can be written.

Table 14-8. Operation of BSTS Bit

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Bit Function
0	0	0	1 when the received data can be read from the FIFO buffer; 0 when the received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	1 when the received data can be read from the FIFO buffer; 0 when software has set BCLR to 1 after the received data has been completely read from the FIFO buffer.
		1	1 when the received data can be read from the FIFO buffer; 0 when the received data has been completely read from the FIFO buffer.
1	0	0	1 when the transmit data can be written to the FIFO buffer; 0 when the transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

Table 14-9. Information Cleared by USB Module by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the relevant pipe (both FIFO buffer planes are cleared when double buffer mode is selected)	
2	Internal flags concerning the BFRE bit	When the BFRE setting is modified
3	FIFO buffer toggle control	When the DBLB setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

Table 14-10. Operation of USB Module depending on PID Bit Setting (When Host Controller Function is Selected)

PID Bits (PID1 and PID0)	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB Module
00 (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01 (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while UACT is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception. Does not issue tokens while UACT is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception.
10 (STALL) or 11 (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

Table 14-11. Operation of USB Module depending on PID Bit Setting (When Function Controller Function is Selected)

PID Bits (PID1 and PID0)	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB Module
00 (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host For the operation when ATREPM is 1, refer to the description of the ATREPM bit.
01 (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
10 (STALL) or 11 (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.

Figure 14-42. Format of PIPEn Control Registers (PIPEnCTR) (n = 6, 7)

Address: F047AH, F047BH (PIPE6CTR), F047CH, F047DH (PIPE7CTR) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEnCTR	BSTS	–	–	–	–	–	ACLRM	SQCLR	SQSET	SQMON	PBUSY	–	–	–	PID1	PID0

BSTS	Buffer status	R/W
0	Buffer access by the CPU is disabled.	R
1	Buffer access by the CPU is enabled.	
Indicates the FIFO buffer status for the relevant pipe. The meaning of the BSTS bit depends on the settings of the DIR, BFRE, and DCLR bits as shown in Table 14-8.		

Bits 14 to 10	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

ACLRM	Auto buffer clear mode Note 1	R/W
0	Auto buffer clear mode is disabled.	R/W
1	Auto buffer clear mode is enabled (all buffers are initialized)	
Enables or disables auto buffer clear mode for the relevant pipe. To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously. Table 14-12 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.		

SQCLR	Toggle bit clear Note 2	R/W
0	Invalid	R/W Note 3
1	Specifies DATA0.	
The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQCLR bit to 1 through software allows the USB module to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQCLR bit to 0.		

SQSET	Toggle bit set Note 2	R/W
0	Invalid	R/W Note 3
1	Specifies DATA1.	
The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe. Setting the SQSET bit through software allows the USB module to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB module always sets the SQSET bit to 0.		

SQMON	Toggle bit monitor	R/W
0	DATA0	R
1	DATA1	
Indicates the value of the sequence toggle bit for the next transaction of the relevant pipe. When the relevant pipe is not for the isochronous transfer, the USB module allows the SQMON bit to toggle upon normal completion of the transaction of the relevant pipe. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.		

PBUSY	Pipe busy	R/W
0	The relevant pipe is not used for the transaction.	R
1	The relevant pipe is used for the transaction.	
Indicates whether the relevant pipe is being currently used or not for the transaction. The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction. Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible. For details, refer to **14.4.4.1 Pipe control register switching procedures**.		

Bits 4 to 2	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PID1	PID0	Response PID	R/W
0	0	NAK response	R/W
0	1	BUF response (depending on the buffer state)	
1	0	STALL response	
1	1	STALL response	
The PID1 and PID0 bits specify the response type for the next transaction of the relevant pipe. The default setting of the PID1 and PID0 bits is NAK. Modify the setting of these bits to BUF to use the relevant pipe for USB transfer. Tables 14-10 and 14-11 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB module depending on the PID bit setting. After modifying the setting of the PID1 and PID0 bits through software from BUF to NAK during USB communication using the relevant pipe, check that PBUSY is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. The USB module modifies the setting of the PID1 and PID0 bits in the following cases. • The USB module sets PID to NAK on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and software has set the SHTNAK bit for the selected pipe to 1. • The USB module sets PID to STALL (11B) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe. • The USB module sets PID to NAK on detecting a USB bus reset when the function controller function is selected. • The USB module sets PID to NAK on detecting a receive error, such as a CRC error, three consecutive times when the host controller function is selected. • The USB module sets PID to STALL (11B) on receiving the STALL handshake when the host controller function is selected. To specify each response type, set the PID1 and PID0 bits as follows. • To make a transition from NAK (00B) to STALL, write 10B. • To make a transition from BUF (01B) to STALL, write 11B. • To make a transition from STALL (11B) to NAK, write 10 and then 00B. • To make a transition from STALL to BUF, write 00B (NAK) and then 01B (BUF).			

Notes 1. Modify the ACLRM bit while PID is NAK and before the pipe is selected by the CURPIPE bits. Before modifying this bit after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

2. Write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

3. Only 0 can be read and 1 can be written.

Table 14-12. Information Cleared by USB Module by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the selected pipe	
2	The interval count value when the selected pipe is for interrupt transfer and the host controller function is selected	When the interval count value is to be reset
3	Internal flags concerning the BFRE bit	When the BFRE setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

14.3.34 PIPEn transaction counter enable registers (PIPEnTRE) (n = 4, 5)

Figure 14-43. Format of PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 4, 5)

Address: F049CH, F049DH (PIPE4TRE), F04A0H, F04A1H (PIPE5TRE) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEnTRE	–	–	–	–	–	–	TRENB	TRCLR	–	–	–	–	–	–	–	–

Bits 15 to 10	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

TRENB	Transaction counter enable	R/W
0	Transaction counter is disabled.	R/W
1	Transaction counter is enabled.	
Enables or disables the transaction counter. For the pipe in the receiving direction, setting the TRENB bit to 1 after setting the total number of the packets to be received in the TRNCNT bits through software allows the USB module to control hardware as described below on having received the number of packets equal to the setting of the TRNCNT bits. • While SHTNAK is 1, the USB module modifies the PID bits to NAK for the corresponding pipe on having received the number of packets equal to the setting of the TRNCNT bits. • While BFRE is 1, the USB module asserts the BRDY interrupt on having received the number of packets equal to the setting of the TRNCNT bits and then reading out the last received data. For the pipe in the transmitting direction, set the TRENB bit to 0. When the transaction counter is not used, set the TRENB bit to 0. When the transaction counter is used, set the TRNCNT bits before setting the TRENB bit to 1. Set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.		

TRCLR	Transaction counter clear	R/W
0	Invalid	R/W
1	The current counter value is cleared.	
Clears the current value of the transaction counter corresponding to the relevant pipe and then sets the TRCLR bit to 0.		

Bits 7 to 0	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

Caution Modify each bit in the PIPEnTRE register while CSST is 1 and PID is NAK.
Before modifying these bits after modifying the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, if the PID bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

14.3.35 PIPEn transaction counter registers (PIPEnTRN) (n = 4, 5)

Figure 14-44. Format of PIPEn Transaction Counter Registers (PIPEnTRN) (4, 5)

Address: F049EH, F049FH (PIPE4TRN), F04A2H, F04A3H (PIPE5TRN) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPEnTRN	TRNCNT[15:0]															

TRNCNT[15:0]	Transaction counter	R/W
	[When written to] Specifies the number of transactions during a DMA transfer. [When read from] Indicates the specified number of transactions if TRENB is 0. Indicates the number of currently counted transactions if TRENB is 1. The USB module increments the value of the TRNCNT bits by one when all of the following conditions are satisfied on receiving the packet. • TRENB = 1 • (TRNCNT set value ? current counter value + 1) on receiving the packet. • The payload of the received packet agrees with the setting of the MXPS bits. The USB module clears the value of the TRNCNT bits to 0 when any of the following conditions are satisfied. (1) All of the following conditions are satisfied. • TRENB = 1 • (TRNCNT set value = current counter value + 1) on receiving the packet. • The payload of the received packet agrees with the setting of the MXPS bits. (2) All of the following conditions are satisfied. • TRENB = 1 • The USB module has received a short packet. (3) All of the following conditions are satisfied. • TRENB = 1 • Software has set the TRCLR bit to 1. For the pipe in the transmitting direction, set the TRNCNT bits to 0. When the transaction counter is not used, set the TRNCNT bits to 0. Setting the number of transactions to be transferred to the TRNCNT bits is only enabled when the TRENB bit in the PIPEnTRE register is 0. To modify the number of transactions to be transferred, set the TRCLR bit in the PIPEnTRE register to 1 (to clear the current counter value) before setting TRENB to 1.	R/W

14.3.36 BC control register n (USBBCCTRLn) (n = 0, 1)

Figure 14-45. Format of BC Control Register 0 (USBBCCTRL0)

Address: F04B0H, F04B1H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBBCCTRL0	–	–	–	–	–	–	PDDETSTS0	CHGDETSTS0	BATCHGE0	DCPMODE0	VDMSRCE0	IDPSINKE0	VDPSRCE0	IDMSINKE0	IDPSRCE0	RPDME0

Bits 15 to 10	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PDDETSTS0	V_{DP_SRC} (0.6 V) input detection flag for UDP0 pin	R/W
0	Not detected	R
1	Detected	
Detects V_{DP_SRC} (0.6 V) is applied to the UDP0 pin from the connected device. (Detects that the voltage applied to UDP0 is in the range of V_{DAT_REF} to V_{IH} (UDP0).) To use this bit for detection, set the CNEN bit (bit 8) in the SYSCFG register to 1, and enable the single end receiver of USB port 0. Valid when IDPSINKE0 is 1.		

CHGDETSTS0	V_{DM_SRC} (0.6 V) input detection flag for UDM0 pin	R/W
0	Not detected	R
1	Detected	
Detects V_{DM_SRC} (0.6 V) is applied to the UDM0 pin from the connected device. (Detects that the voltage applied to UDM0 is in the range of V_{DAT_REF} to V_{IH} (UDM0).) To use this bit for detection, set the CNEN bit (bit 8) in the SYSCFG register to 1, and enable the single end receiver of USB port 0. Valid when IDMSINKE0 is 1.		

BATCHGE0	USB port 0 BC connection detection operation enable	R/W
0	Operation disabled	R/W
1	Operation enabled	
When this bit is set to enabled, each bit setting of VDPSRCE0, VDMSRCE0, IDPSINKE0, IDMSINKE0, and IDPSRCE0 is valid BC connection detection can be operated via USB port 0.		

DCPMODE0	Dedicated charging port resistor connection control for UDP0/UDM0 pins	R/W
0	Resistor disabled	R/W
1	Resistor enabled	
Connects the resistor (R_{DCP_DAT}) between the UDP0 and UDM0 pins used for the host (dedicated charging port) BC connection detection function.		

VDMSRCE0	UDM0 pin V_{DM_SRC} (0.6 V) output control	R/W
0	V_{DM_SRC} output disabled	R/W
1	V_{DM_SRC} output enabled (0.6 V output)	
Controls the V_{DM_SRC} output.		

IDPSINKE0	UDP0 pin V_{DP_SRC} (0.6 V) input detection (comparator and sink) control	R/W
0	UDP0 pin (0.6 V) input detection disabled	R/W
1	UDP0 pin (0.6 V) input detection enabled	
Controls the I_{DP_SINK} (sink current) used for the 0.6 V input detection circuit (comparator) and detection for the UDP0 pin.		

VDPSRCE0	UDP0 pin V_{DP_SRC} (0.6 V) output control	R/W
0	V_{DP_SRC} output disabled	R/W
1	V_{DP_SRC} output enabled (0.6 V output)	
Controls the V_{DP_SRC} output.		

IDMSINKE0	UDM0 pin V_{DM_SRC} (0.6 V) input detection (comparator and sink) control	R/W
0	UDM0 pin (0.6 V) input detection disabled	R/W
1	UDM0 pin (0.6 V) input detection enabled	
Controls the I_{DM_SINK} (sink current) used for the 0.6 V input detection circuit (comparator) and detection for the UDM0 pin.		

IDPSRCE0	UDP0 pin I_{DP_SRC} (10 μA) output control	R/W
0	I_{DP_SRC} output disabled	R/W
1	I_{DP_SRC} output enabled (10 μA output)	
Controls the I_{DP_SRC} output.		

RPDME0	UDM0 pull-down control	R/W
0	Pulling down disabled	R/W
1	Pulling down enabled	
Only the UDM0 pin can be pulled down (R_{PD}) by setting this bit.		

Figure 14-46. Format of BC Control Register 1 (USBBCCTRL1)

Address: F04B4H, F04B5H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBBCCTRL1	–	–	–	–	–	–	PDDETSTS1	–	BATCHGE1	DCPMODE1	VDMSRCE1	IDPSINKE1	–	–	–	–

Bits 15 to 10	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PDDETSTS1	V_{DP_SRC} (0.6 V) input detection flag for UDP1 pin	R/W
0	Not detected	R
1	Detected	
Detects V_{DP_SRC} (0.6 V) is applied to the UDP1 pin from the connected device. (Detects that the voltage applied to UDP1 is in the range of V_{DAT_REF} to V_{IH} (UDP1).) To use this bit for detection, set the CNEN bit (bit 8) in the SYSCFG1 register to 1, and enable the single end receiver of USB port 1. Valid when IDPSINKE1 is 1.		

Bit 8	Reserved	R/W
–	The write value must be 0.	R/W

BATCHGE1	USB port 1 BC connection detection operation enable	R/W
0	Operation disabled	R/W
1	Operation enabled	
When this bit is set to enabled, each bit setting of VDMSRCE1, IDPSINKE1 is valid BC connection detection can be operated via USB port 1.		

DCPMODE1	Dedicated charging port resistor connection control for UDP1/UDM1 pins	R/W
0	Resistor disabled	R/W
1	Resistor enabled	
Connects the resistor (R_{DCP_DAT}) between the UDP1 and UDM1 pins used for the host (dedicated charging port) BC connection detection function.		

VDMSRCE1	UDM1 pin V_{DM_SRC} (0.6 V) output control	R/W
0	V_{DM_SRC} output disabled	R/W
1	V_{DM_SRC} output enabled (0.6 V output)	
Controls the V_{DM_SRC} output.		

IDPSINKE1	UDP1 pin V_{DP_SRC} (0.6 V) input detection (comparator and sink) control	R/W
0	UDP1 pin (0.6 V) input detection disabled	R/W
1	UDP1 pin (0.6 V) input detection enabled	
Controls the I_{DP_SINK} (sink current) used for the 0.6 V input detection circuit (comparator) and detection for the UDP1 pin.		

Bit 3	Reserved	R/W
–	The write value must be 0.	R/W

Bit 2	Reserved	R/W
–	The write value must be 0.	R/W

Bit 1	Reserved	R/W
–	The write value must be 0.	R/W

Bit 0	Reserved	R/W
–	The write value must be 0.	R/W

14.3.37 BC option control register n (USBBCOPTn) (n = 0, 1)

Figure 14-47. Format of BC Option Control Register 0 (USBBCOPT0)

Address: F04B8H, F04B9H After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBBCOPT0	–	–	–	–	–	–	DMCUSDET0	DPCUSDET0	–	CUSDETE0	VDOUTE0	–	VDSEL03	VDSEL02	VDSEL01	VDSEL00

Bit 15	Reserved	R/W
–	The write value must be 0.	R/W

Bit 14	Reserved	R/W
–	The write value must be 0.	R/W

Bits 13 to 10	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DMCUSDET0 **Note**	UDM0 voltage detectiion (Option BC: Host/functiion)	R/W
0	• Host UDM0 pin output voltage is maintained (Within the comparison voltage range selected by the VDSEL0x bit). • Function UDM0 pin voltage is lower than the comparison voltage selected by the VDSEL0x bit.	R
1	• Host UDM0 pin output voltage change is detected (When the comparison voltage range selected by the VDSEL0x bit is exceeded). • Function UDM0 pin voltage exceeds the comparison voltage selected by the VDSEL0x bit.	

DPCUSDET0 **Note**	UDP0 voltage detectiion (Option BC: Host/functiion)	R/W
0	• Host UDP0 pin output voltage is maintained (Within the comparison voltage range selected by the VDSEL0x bit). • Function UDP0 pin voltage is lower than the comparison voltage selected by the VDSEL0x bit.	R
1	• Host UDP0 pin output voltage change is detected (When the comparison voltage range selected by the VDSEL0x bit is exceeded). • Function UDP0 pin voltage exceeds the comparison voltage selected by the VDSEL0x bit.	

Bit 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

CUSDETE0	Option voltage detectiion circuit control (Option BC: Host/functiion)	R/W
0	Disabled	R/W
1	Enabled	

VDOUTE0	Option voltage output control (Option BC: Host)	R/W
0	Disabled	R/W
1	Enabled	

Bit 4	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

VDSEL03 to VDSEL00	UDP0/UDM0 pin option output voltage and comparison voltage select (Option BC: Host/functiion)	R/W
Select UDP0/UDM0 pin output voltage and comparison voltage value (Refer to **Remark**)		R/W

Note Valid when CUSDETE0 = 1

Remark The option output voltage values of the UDP0 and UDM0 pins and the comparison voltage values of the option BC detection circuit when UV_{BUS} = 5.0 V are shown below. The following voltages vary in proportion to the UV_{BUS} input voltage.

VDOUTE0	VDSEL03	VDSEL02	VDSEL01	VDSEL00	Output voltage (V)		Comparison voltage (V) (Valid when CUSDETE0 = 1)	
					UDP0	UDM0	UDP0	UDM0
1 (Host)	1	0	0	0	2.00	2.00	1.60 to 2.60	1.60 to 2.60
	1	0	0	1	2.68	2.00	2.45 to 2.80	1.60 to 2.60
	1	0	1	0	2.00	2.68	1.60 to 2.60	2.45 to 2.80
	1	1	0	0	3.30	3.30	–	–
0 (Function)	0	0	0	0	–		1.60	
	0	0	0	1	–		1.70	
	0	0	1	0	–		1.85	
	0	0	1	1	–		2.00	
	0	1	0	0	–		2.15	
	0	1	0	1	–		2.30	
	0	1	1	0	–		2.45	
	0	1	1	1	–		2.60	
	1	0	0	0	–		2.80	
	1	0	0	1	–		3.00	
	1	0	1	0	–		3.20	
	1	0	1	1	–		3.40	
	1	1	0	0	–		3.60	
	1	1	0	1	–		3.80	
	1	1	1	0	–		4.00	
	1	1	1	1	–		4.20	
Other than above					Setting prohibited			

Figure 14-48. Format of BC Option Control Register 1 (USBBCOPT1)

Address: F04BCH, F04BDH After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBBCOPT1	–	–	–	–	–	–	DMCUSDET1	DPCUSDET1	–	CUSDETE1	VDOUTE1	–	VDSEL13	VDSEL12	VDSEL11	VDSEL10

Bit 15	Reserved	R/W
–	The write value must be 0.	R/W

Bit 14	Reserved	R/W
–	The write value must be 0.	R/W

Bits 13 to 10	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

DMCUSDET1 **Note**	UDM1 voltage detectiion (Option BC: Host)	R/W
0	• Host UDM1 pin output voltage is maintained (Within the comparison voltage range selected by the VDSEL1x bit).	R
1	• Host UDM1 pin output voltage change is detected (When the comparison voltage range selected by the VDSEL1x bit is exceeded).	

DPCUSDET1 **Note**	UDP1 voltage detectiion (Option BC: Host)	R/W
0	• Host UDP1 pin output voltage is maintained (Within the comparison voltage range selected by the VDSEL1x bit).	R
1	• Host UDP1 pin output voltage change is detected (When the comparison voltage range selected by the VDSEL1x bit is exceeded).	

Bit 7	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

CUSDETE1	Option voltage detectiion circuit control (Option BC: Host)	R/W
0	Disabled	R/W
1	Enabled	

VDOUTE1	Option voltage output control (Option BC: Host)	R/W
0	Disabled	R/W
1	Enabled	

Bit 4	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

VDSEL13 to VDSEL10	UDP1/UDM1 pin option output voltage and comparison voltage select (Option BC: Host)	R/W
Select UDP1/UDM1 pin output voltage and comparison voltage value (Refer to **Remark**)		R/W

Note Valid when CUSDETE1 = 1

Remark The option output voltage values of the UDP1 and UDM1 pins and the comparison voltage values of the option BC detection circuit when UV_{BUS} = 5.0 V are shown below. The following voltages vary in proportion to the UV_{BUS} input voltage.

VDOUTE1	VDSEL13	VDSEL12	VDSEL11	VDSEL10	Output voltage (V)		Comparison voltage (V) (Valid when CUSDETE1 = 1)	
					UDP1	UDM1	UDP1	UDM1
1 (Host)	1	0	0	0	2.00	2.00	1.60 to 2.60	1.60 to 2.60
	1	0	0	1	2.68	2.00	2.45 to 2.80	1.60 to 2.60
	1	0	1	0	2.00	2.68	1.60 to 2.60	2.45 to 2.80
	1	1	0	0	3.30	3.30	–	–
Other than above					Setting prohibited			

14.3.38 USB module control register (USBMC)

Figure 14-49. Format of USB Module Control Register (USBMC)

Address: F04CCH, F04CDH After reset: 0002H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBMC	–	–	–	–	–	–	–	–	VBRPDCUT	–	–	–	–	–	PXXCON	VDDUSBE

Bits 15 to 8	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

VBRPDCUT **Note 1**	UV_{BUS} pin pull-down resistor control	R/W
0	Pull-down resistor is valid.	R/W
1	Pull-down resistor is invalid.	

Bits 6 to 2	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

PXXCON	VDDUSBE	USB internal power supply control	R/W
0	0	The internal power supply for USB is stopped UV_{DD} pin input is pull-down setting **Note 2**	R/W
0	1	The internal power supply for USB is stopped **Note 3** Setting for when power (3.3 V) is applied externally via the UV_{DD} pin **Note 3** Be sure to specify this setting when using the BC connection detection function (except the optional BC connection detection functions) as well as the above settings **Note 4.**	
1	0	The internal power supply for USB is stopped Setting for when power (3.3 V) is applied externally via the UV_{DD} pin **Note 3**	
1	1	The internal power supply for USB is turned on. Power (3.3 V) is generated internal USB power supply **Notes 3, 4**	
Controlling to the USB internal power supply.			

Notes 1. Be sure to set VBRPDCUT = 1 (pull-down resistor is invalid) when this pin is used as VBUS power supply in the BC option (host), etc. Set VBRPDCUT = 0 (pull-down resistor is valid) when the VBUS input is used, or this pin is not used.

2. To prevent malfunction caused by a floating USB power supply, be sure to specify this setting when power is not being applied externally from the UV_{DD} pin or when the internal USB power supply might stop.

3. When PXXCON = 0 and VDDUSBE = 1 or PXXCON = 1 and VDDUSBE = 0, the high level of the UDP and UDM pins is based on the level of the external power supply input from the UV_{DD} pin. When PXXCON = 1 and VDDUSBE = 1, the high level of the UDP and UDM pins is based on the voltage generated by the internal USB power supply (3.3 V), and 3.3 V is output from the UV_{DD} pin.

4. Specify PXXCON = 0 and VDDUSBE = 1 when using the BC connection detection function (except the optional BC connection detection functions) with external power being applied to the UV_{DD} pin. Specify PXXCON = 1 and VDDUSBE = 1 when using the BC connection detection function (except the optional BC connection detection functions) with the internal USB power supply. Note that the temperature sensor cannot be used and A/D conversion that uses the internal reference voltage cannot be performed when using the internal USB power supply or using the BC connection detection function (except the optional BC connection detection functions) (VDDUSBE = 1).

14.3.39 Device address n configuration registers (DEVADDn) (n = 0 to 5)

Figure 14-50. Format of Device Address n Configuration Registers (DEVADDn) (n = 0 to 5)

Address: F04D0H, F04D1H (DEVADD0), F04D2H, F04D3H (DEVADD1), F04D4H, F04D5H (DEVADD2), F04D6H, F04D7H (DEVADD3), F04D8H, F04D9H (DEVADD4), F04DAH, F04DBH (DEVADD5) After reset: 0000H

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEVADDn	–	–	–	–	–	–	–	–	USBSPD1	USBSPD0	–	–	–	–	–	RTPORT

Bits 15 to 8	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

USBSPD1	USBSPD0	Transfer speed of communication target device	R/W
0	0	The DEVADDn register is not used	R/W
0	1	Low speed	
1	0	Full speed	
1	1	Do not set.	
These bits specify the USB transfer speed of the communication target peripheral device. When the host controller function is selected, the USB module refers to the setting of the USBSPD[1:0] bits to generate packets. When the function controller function is selected, set these bits to 00B.			

Bits 5 to 1	Nothing is assigned	R/W
–	The write value must be 0. The read value is 0.	–

RTPORT	Route hub port number	R/W
0	USB port 0	R/W
1	USB port 1	
Specifies the corresponding device is connected to which port. When the host controller function is selected, the USB module references to the setting of the RTPORT bit and generates a packet. When the function controller function is selected, this setting is ignored.		

14.4 Operation

14.4.1 System control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

14.4.1.1 Starting operation

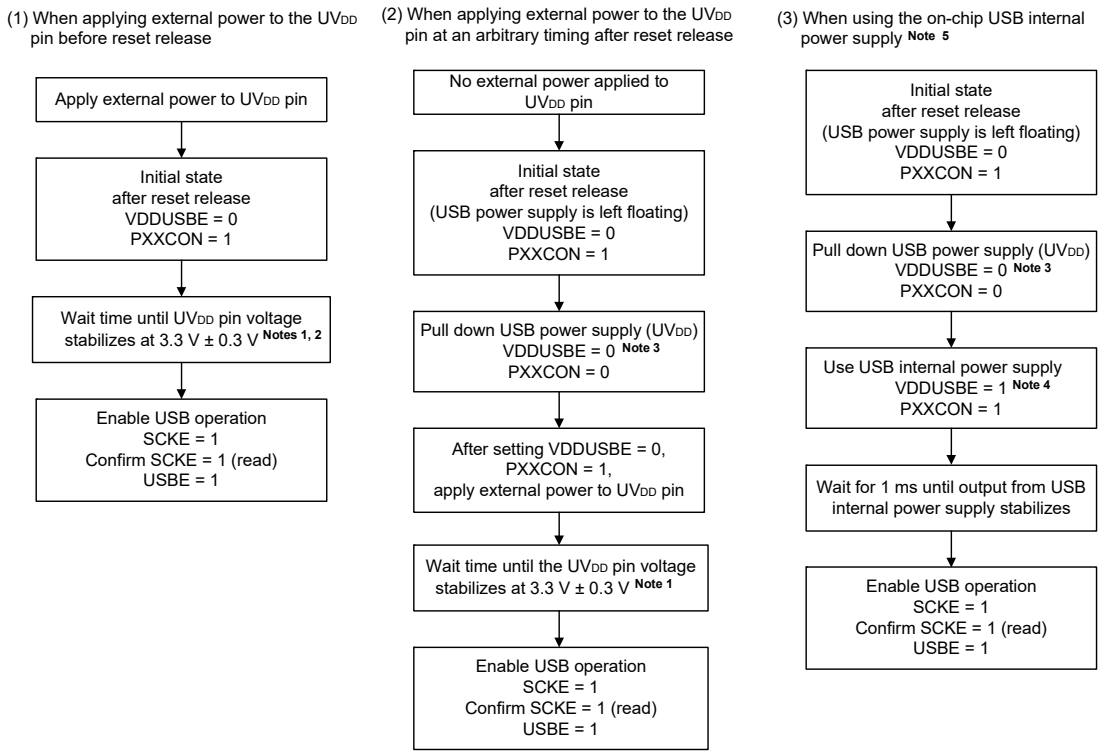
The source for the USB power supply (UV_{DD}) can be applied either externally from the UV_{DD} pin or from the on-chip USB internal power supply.

When no external power is applied to the UV_{DD} pin, the initial state after releasing the USB power supply reset is a floating state. To prevent erroneous operation caused by the floating state, pull down the USB power supply (UV_{DD}) by setting the register (VDDUSBE bit = 0, PXXCON bit = 0) until applying external power to the UV_{DD} pin or using the USB internal power supply.

When using the on-chip USB internal power supply, it is necessary to connect an external 0.33 μF stabilization capacitance (for V_{SS}) to the UV_{DD} pin. While the USB internal power supply is used, A/D conversion cannot be performed using the temperature sensor or internal reference voltage at the same time.

Figure 14-51 shows the flow for turning on the USB power when using the USB host/function controller. For details about the flow for turning on the USB power when using the USB host/function controller and the BC connection detection function (except the optional BC connection detection functions), see the application note.

Figure 14-51. Flow for Turning on USB Power



Notes 1. The wait time depends on the characteristics of an external power supply to be used.

2. No wait time is required if the UV_{DD} pin voltage stabilizes during the reset release sequence.

3. To prevent malfunction caused by a floating USB power supply, be sure not to change this setting when power is not being applied externally from the UV_{DD} pin or when the internal USB power supply might be stop.

4. When the USB internal power supply is used, A/D conversion cannot be performed using the temperature sensor or internal reference voltage.

5. When using the USB internal power supply, connect an external 0.33 μF stabilization capacitance (for V_{SS}) to the UV_{DD} pin.

14.4.1.2 Controller function selection

For the USB module, the host controller function or function controller function can be selected using the DCFM bit in the SYSCFG register. The DCFM bit should be modified in the initial settings immediately after a power-on reset or in the D+ pull-up-disabled (DPRPU = 0) and D+ and D- pull-down-disabled (DRPD = 0) state.

Table 14-13 shows USB Port Function Selection.

Table 14-13. USB Port Function Selection

When the host controller function is selected		
USB port 0	USB port 1	Remark
Full or low	Full or low	The transfer scheduling is common to USB port 0 and USB port 1. The output is driven to both USB port 0 and USB port 1.
When the function controller function is selected		
USB port 0	USB port 1	Remark
Full or low	Not used	USB port 1 is invalid.

14.4.1.3 Controlling USB data bus resistors

The USB module contains pull-up and pull-down resistors of the D+ and D- lines. Pull up or pull down these lines by setting the DPRPU and DRPD bits in the SYSCFG register.

When the function controller function is selected, confirm that connection to the USB host is made, then set the DPRPU bit in the SYSCFG register to 1 and pull up the D+ (full speed)/D- (low speed) line.

When the DPRPU bit in the SYSCFG register is set to 1 during communication with the PC, the USB module disables the pull-up resistor of the USB data line, thus notifying the USB host of connection.

When the host controller function is selected, set the DRPD bit (SYSCFG or SYSCFG1 register) for the used port to 1 and pull down the D+/D- lines.

Table 14-14 lists the settings for Controlling USB Data Bus Resistors of USB port 0 and Table 14-15 lists the settings for Controlling USB Data Bus Resistors of USB port 1.

Table 14-14. Controlling USB Data Bus Resistors of USB port 0

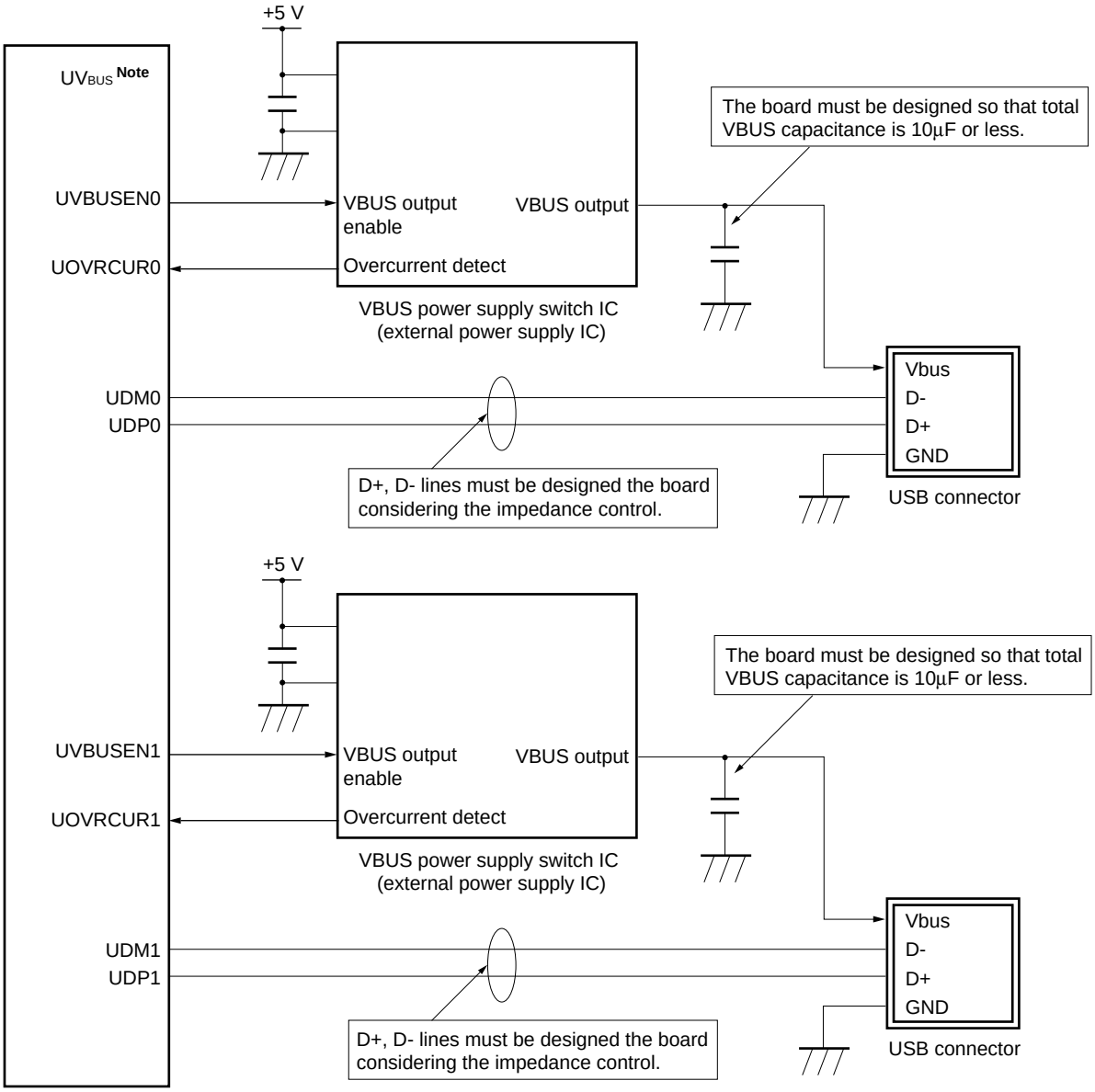
Settings			USB Data Bus Resistor Control		
DRPD	DPRPU	DMRPU	D- Line	D+ Line	Remarks
0	0	0	Open	Open	When USB port 0 is not used
0	1	0	Open	Pull-up	Set to this state when operating as the function controller (full speed).
0	0	1	Pull-up	Open	Set to this state when operating as the function controller (low speed).
1	0	0	Pull-down	Pull-down	Set to this state when operating as the host controller.
Other than above			–	–	Setting prohibited

Table 14-15. Controlling USB Data Bus Resistors of USB port 1

Settings	USB Data Bus Resistor Control		
DRPD	D- Line	D+ Line	Remarks
0	Open	Open	When USB port 1 is not used
1	Pull-Down	Pull-Down	Set to this state when operating as the host controller.

Figures 14-52 to 14-56 show the examples of USB external connection circuit.

Figure 14-52. USB Connector Host Connection Example



Note The UV_{BUS} input pin is not used when the host controller function is selected.
The UV_{BUS} pin is connected to an on-chip pulled-down resistor while the VBRPDCUT bit (bit 7 in the USBMC register) is in the default state (0), in order to prevent the pin level from becoming unstable when the pin is left open.
To fix the voltage of the UV_{BUS} pin by applying an external voltage, set the VBRPDCUT bit to 1 and disconnect this on-chip resistor to prevent unnecessary current consumption.
When using the battery charging connection detection optional functions, the VBUS voltage to be connected to the USB connector must be applied to the UV_{BUS} pin. In this case, set the VBRPDCUT bit to 1 and cut off the internal pull-down resistor of the UV_{BUS} pin.

Figure 14-53. USB Connector Function Connection Example in Self-powered Mode (5 V)

<R>

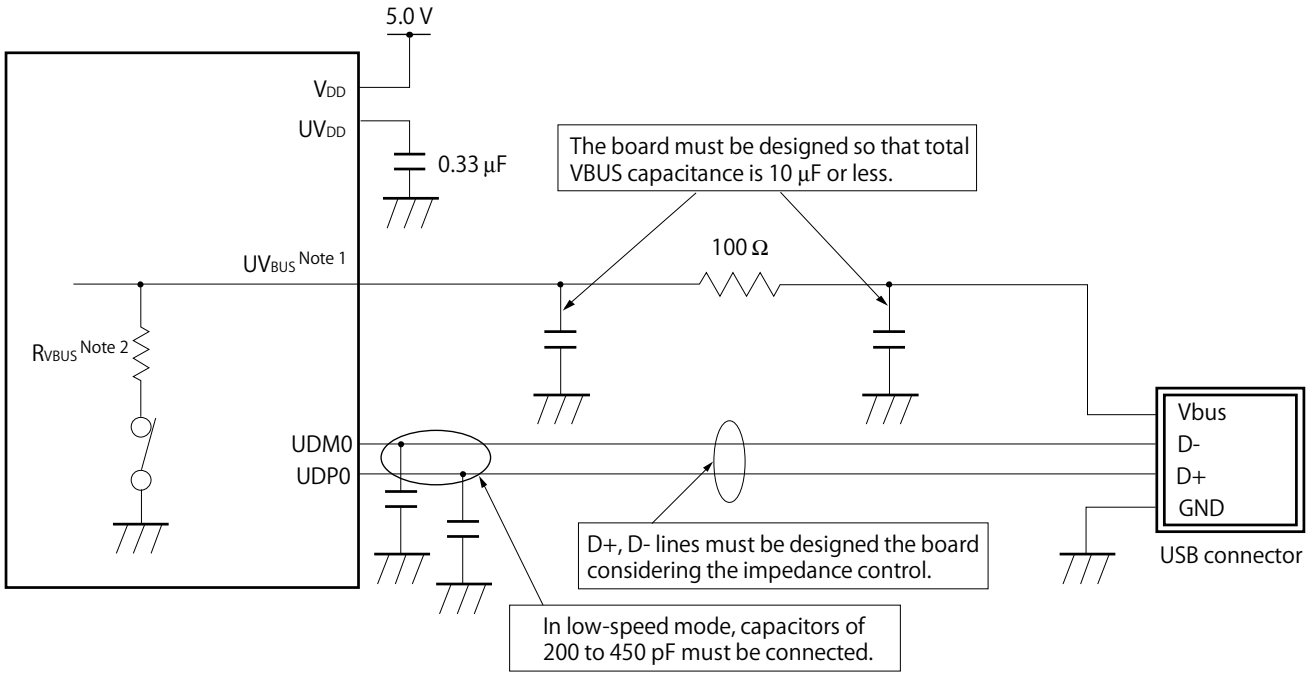
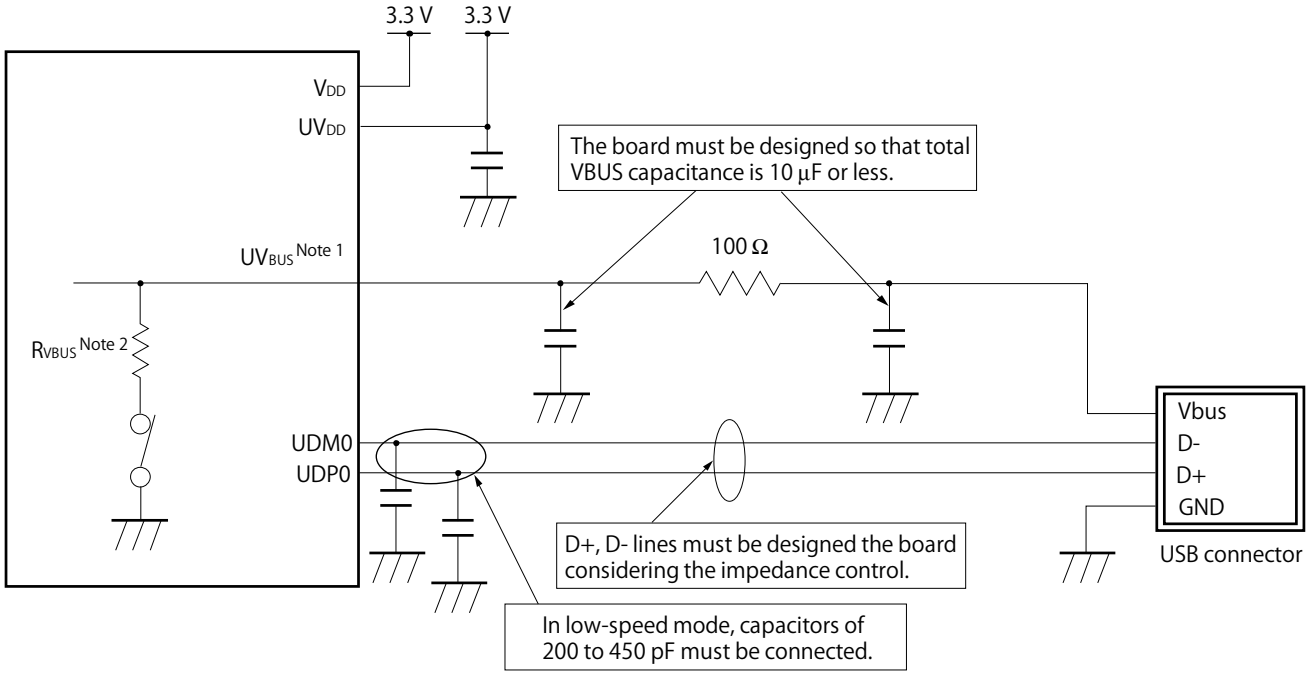


Figure 14-54. USB Connector Function Connection Example in Self-powered Mode (3.3 V)

<R>



Notes 1. 5 V torelant

2. Set the VBRPDCUT bit (bit 7 in the USBMC register) to 0 and connect the on-chip pull-down resistor of the UV_{BUS} pin.

Figure 14-55. USB Connector Function Connection Example in Bus-powered Mode (5 V)

<R>

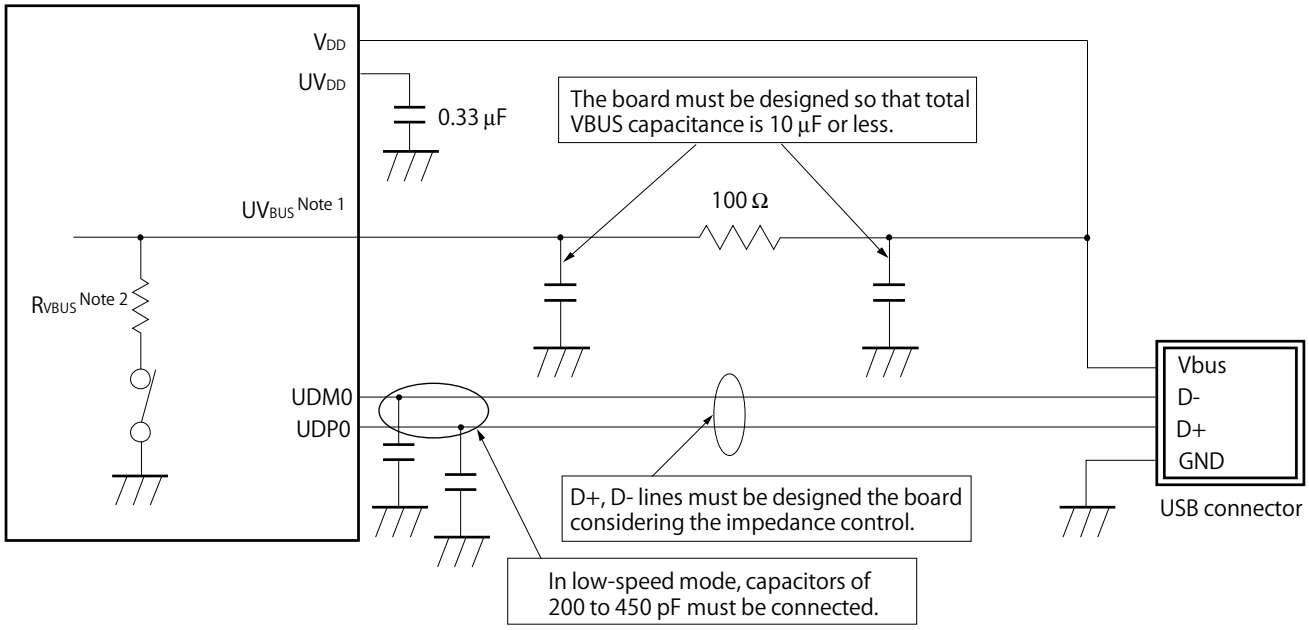
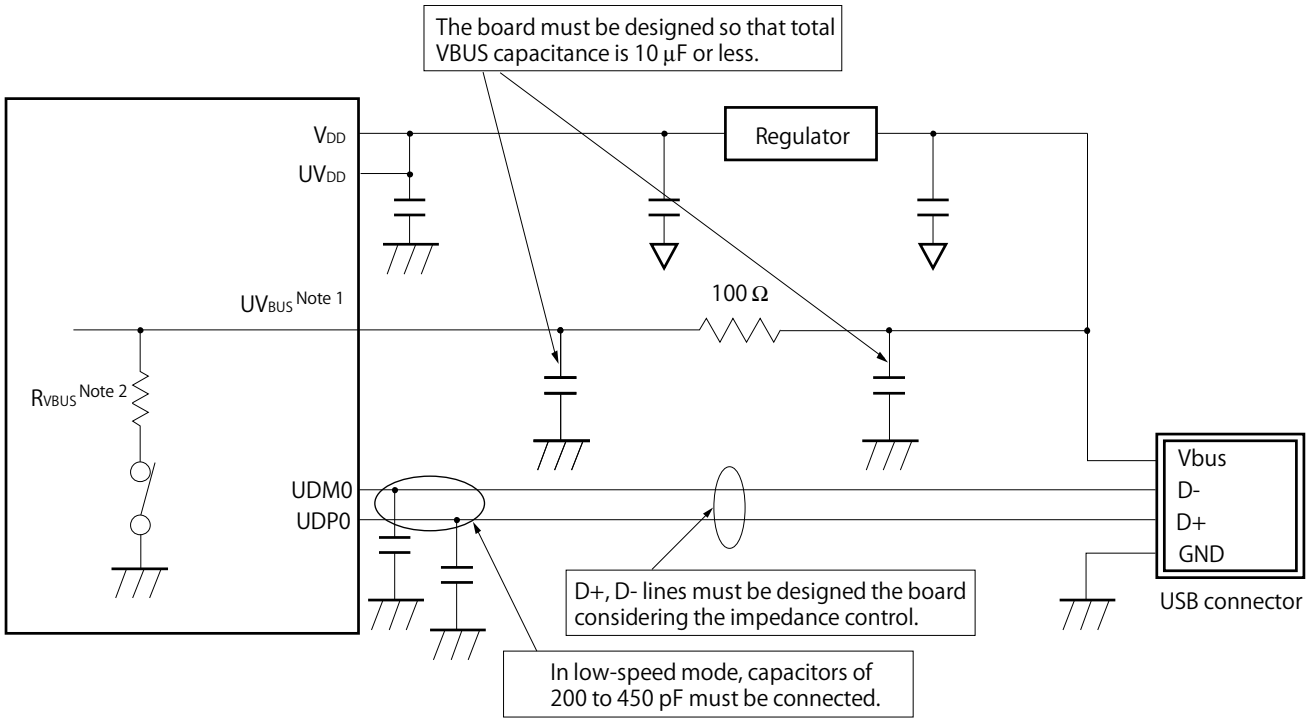


Figure 14-56. USB Connector Function Connection Example in Bus-powered Mode (3.3 V)

<R>



Notes 1. 5 V torelant

2. Set the VBRPDCUT bit (bit 7 in the USBMC register) to 0 and connect the on-chip pull-down resistor of the UV_{BUS} pin.

14.4.2 Interrupt sources

Table 14-16 lists the Interrupt Sources in the USB module.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, the USB issues a USB interrupt request to the interrupt controller and an USB interrupt will be generated.

Table 14-16. Interrupt Sources (1/2)

Bit to be Set	Name	Interrupt Source	Function That Generates Interrupt	Status Flag
VBINT	VBUS interrupt	When a change in the state of the VBUS input pin has been detected (low to high or high to low)	Host/function **Note**	VBSTS
RESM	Resume interrupt	When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	Function	–
SOFR	Frame number update interrupt	[Host controller function is selected] • When an SOF packet with a different frame number has been transmitted [Function controller function is selected] • When an SOF packet with a different frame number has been received	Host/function	–
DVST	Device state transition interrupt	• When a device state transition has been detected A USB bus reset detected Suspend state detected SET_ADDRESS request received SET_CONFIGURATION request received	Function	DVSQ2 to DVSQ0
CTRT	Control transfer stage transition interrupt	• When a stage transition has been detected in control transfer Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred	Function	CTSQ2 to CTSQ0
BEMP	Buffer empty interrupt	• When transmission of all data in the buffer memory has been completed and the buffer has become empty • When a packet larger than the maximum packet size has been received	Host/function	PIPEnBEMP in BEMPSTS register
NRDY	Buffer not ready interrupt	[Host controller function is selected] • When STALL has been received from the peripheral device for the issued token • When a response has not been received correctly from the peripheral device for the issued token (no response was returned three consecutive times or a packet reception error occurred three consecutive times) [Function controller function is selected] • When NAK has been returned for an IN or OUT token	Host/function	PIPEnNRDY in NRDYSTS register
BRDY	Buffer ready interrupt	• When the buffer has become ready for read access or write access.	Host/function	PIPEBRDY in BRDYSTS register

Note Though this interrupt can be generated while the host function is selected, it is not usually used with the host function.

Table 14-16. Interrupt Sources (2/2)

Bit to be Set	Name	Interrupt Source	Function That Generates Interrupt	Status Flag
OVRCR	Overcurrent input change interrupt	• When a change in the state of the UOVRCUR0 (USB port 0) or UOVRCUR1 (USB port 1) input pin has been detected (low to high or high to low)	Host	OVCMON
BCHG	Bus change interrupt	• When a change of USB bus state has been detected	Host/function	LNST
DTCH	USB disconnection detection interrupt	• When disconnection of the USB device has been detected	Host	RHST
ATTCH	Connection interrupt	• When J-state or K-state is detected on the USB port for 2.5 μs. Used for checking whether a peripheral device is connected.	Host	–
EOFERR	EOF error detection interrupt	• When an EOF error of a peripheral device has been detected	Host	–
SACK	Normal response interrupt for setup transaction	• When the normal response (ACK) for the setup transaction has been received	Host	–
SIGN	Error interrupt for setup transaction	• When a setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	–
PDDETINT	Portable Device detection interrupt	• When connection of the Portable Device has been detected	Host	PDDETSTS

Figure 14-57 shows the USB interrupt relationship, Table 14-17 lists the USB interrupts.

Figure 14-57. USB Interrupts Relationship

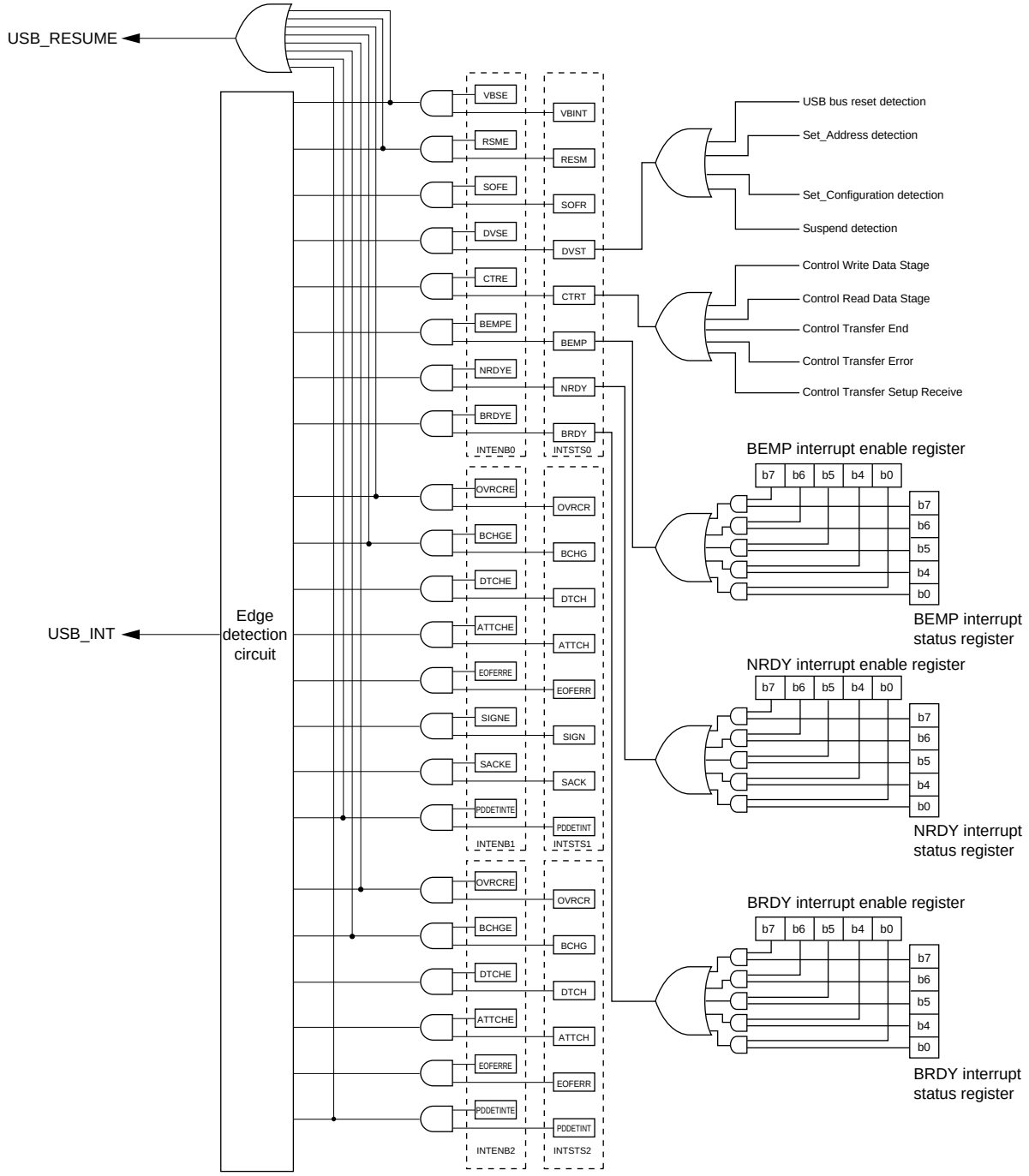


Table 14-17. USB Interrupt List

Interrupt Name	Interrupt Flag	Priority
USB_INT	VBUS interrupt Resume interrupt Frame number update interrupt Device state transition interrupt Control transfer stage transition interrupt Buffer empty interrupt Buffer not ready interrupt Buffer ready interrupt Overcurrent input change interrupt Bus change interrupt USB disconnection detection interrupt Connection interrupt EOF error detection interrupt Normal response interrupt for setup transaction Error interrupt for setup transaction Portable Device detection interrupt	High
USB_RESUME	VBUS interrupt Resume interrupt Overcurrent input change interrupt Bus change interrupt Portable Device detection interrupt	Low

14.4.3 Interrupts

14.4.3.1 BRDY interrupt

The BRDY interrupt is generated when either of the host controller function or function controller function is selected. The following shows the conditions under which the USB module sets 1 to a corresponding bit in the BRDYSTS register. Under this condition, the USB module generates a BRDY interrupt if software has set 1 to the PIPEBRDYE bit in the BRDYENB register that corresponds to the pipe and 1 to the BRDYE bit in the INTENB0 register.
The conditions for generating and clearing the BRDY interrupt depend on the settings of the BRDYM bit and BFRE bit for each pipe as described below.

(1) When BRDYM = 0 and BFRE = 0
With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
On any of the following conditions, the USB module generates an internal BRDY interrupt request trigger and sets 1 to the PIPEBRDY bit corresponding to the pertinent pipe.

[For the pipe set to the transmitting direction]
- When software changes the DIR bit from 0 to 1.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is completed.
- When 1 is written to the ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).

[For the pipe set to the receiving direction]
- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS bit is read as 0).
  No request trigger is generated for the transaction in which DATA-PID disagreement has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.
  No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

When the function controller function is selected, the BRDY interrupt is not generated in the status stage of control transfers.
The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY bit in BRDYSTS through software. In this case, 1s should be written to the PIPEBRDY bits for the other pipes.
Be sure to clear the BRDY status before accessing the FIFO buffer.

(2) When BRDYM bit = 0 and the BFRE bit = 1

With these settings, the USB module generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in BRDYSTS corresponding to the pertinent pipe.
On any of the following conditions, the USB module determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the transaction counter register (TRNCNT bits) is used and the number of packets specified by the TRNCNT bits are completely received.

When the pertinent data is completely read out after any of the above conditions has been satisfied, the USB module determines that all data for a single transfer has been completely read out.
When a zero-length packet is received while the FIFO buffer is empty, the USB module determines that all data for a single transfer has been completely read out upon passing the zero-length packet data to the CPU. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding FIFOCTR register through software.
With these settings, the USB module does not detect a BRDY interrupt for the pipe in the transmitting direction.
The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY bit through software. In this case, 1s should be written to the PIPEBRDY bits for the other pipes.
In this mode, the BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the ACLRM bit.

(3) When BRDYM = 1 and BFRE = 0

With these settings, the PIPEBRDY values are linked to the BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by the USB module depending on the FIFO buffer status.

[For the pipe set to the transmitting direction]
The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready.
However, the BRDY interrupt is not generated even if the DCP in the transmitting direction is ready for write access.

[For the pipe set to the receiving direction]

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).

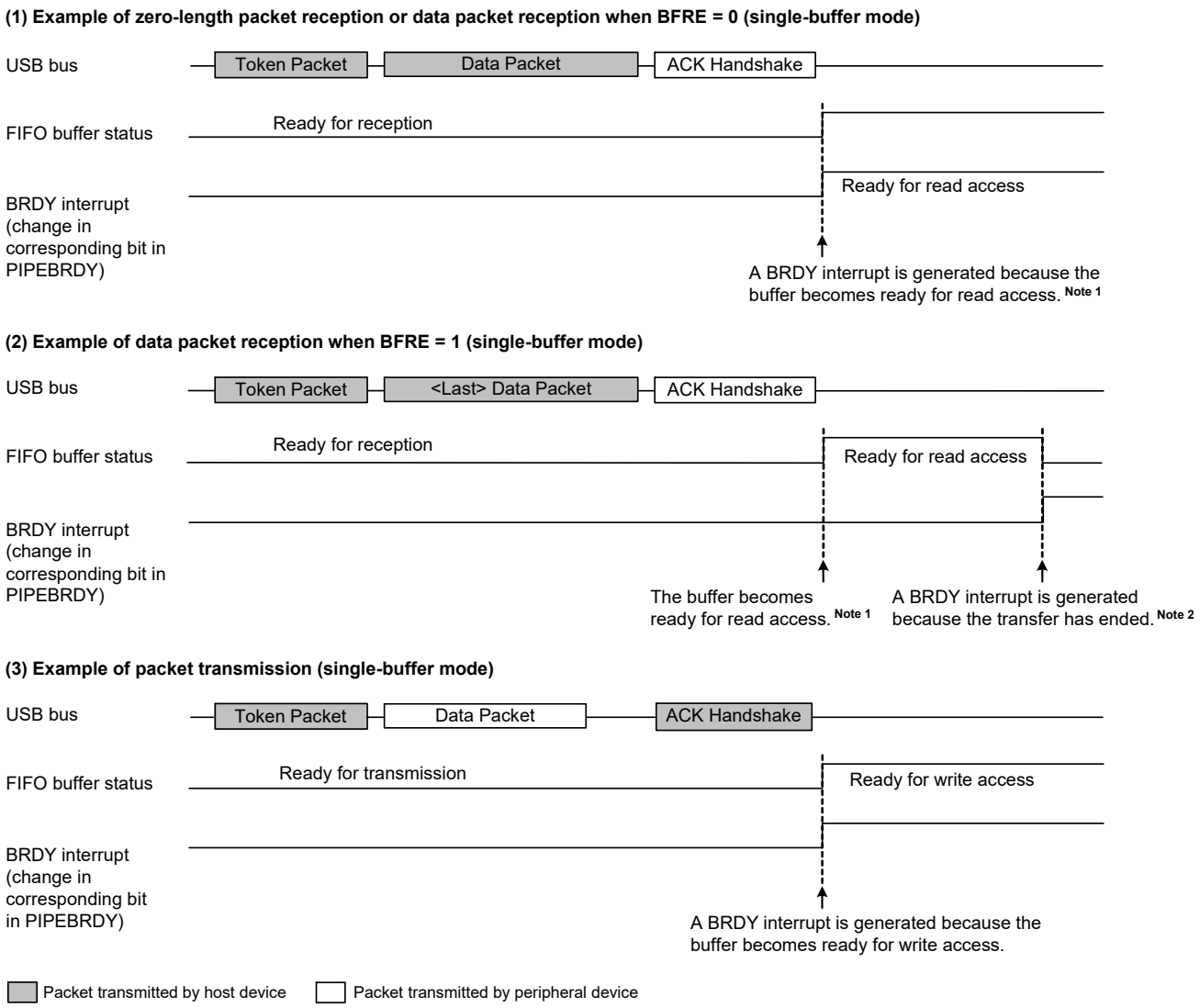
When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until BCLR = 1 is written through software.

With this setting, the PIPEBRDY bit cannot be cleared to 0 through software.

When the BRDYM bit is set to 1, all BFRE bits (for all pipes) should be cleared to 0.

Figure 14-58 shows the Timing of BRDY Interrupt Generation.

Figure 14-58. Timing of BRDY Interrupt Generation



Notes 1. The FIFO buffer becomes ready for read access under the following condition:
When a packet is received while no data remains unread in the buffer in the CPU.

2. A transfer ends under either of the following conditions:
(1) When a short packet including a zero-length packet is received
(2) When the number of packets specified in the transaction counter are received

The condition that USB module clears the BRDY bit in INTSTS0 depends on the setting of the BRDYM bit in the SOFCFG register. Table 14-18 shows the Condition for Clearing BRDY Bit.

Table 14-18. Condition for Clearing BRDY Bit

BRDYM	Condition for Clearing BRDY Bit
0	The USB module clears the BRDY bit in the INTSTS0 register when software has cleared all bits in the BRDYSTS register.
1	The USB module clears the BRDY bit in the INTSTS0 register when the BSTS bits for all piles have become 0.

14.4.3.2 NRDY interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USB module sets the corresponding PIPEnNRDY bit in the NRDYSTS register to 1. If the corresponding bit in the NRDYENB register has been set to 1 by software, the USB module sets the NRDY bit in the INTSTS0 register to 1 and generates a USB interrupt.

The following describes the conditions on which the USB module generates the internal NRDY interrupt request for a given pipe.

Note that the internal NRDY interrupt request is not generated during setup transaction execution when the host controller function is selected. During setup transactions when the host controller function is selected, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller function is selected.

(1) When the host controller function is selected

On any of the following conditions, the USB module detects an NRDY interrupt.

[For the pipe set to the transmitting direction]

- During communications other than setup transactions, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB module sets the corresponding PIPEnNRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK.
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device.
  In this case, the USB module sets the corresponding PIPEnNRDY bit to 1 and modifies the setting of the PID bits of the corresponding pipe to STALL (11B).

[For the pipe set to the receiving direction]

- When any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USB module (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB module sets the PIPEnNRDY bit corresponding to the pipe to 1 and modifies the setting of the PID bits of the corresponding pipe to NAK.
- When the STALL handshake is received.
  In this case, the USB module sets the PIPEnNRDY bit corresponding to the pipe to 1 and modifies the setting of the PID bits of the corresponding pipe to STALL (11B).

(2) When the function controller function is selected

On any of the following conditions, the USB module detects an NRDY interrupt.

[For the pipe set to the transmitting direction]

When an IN token is received while there is no data to be transmitted in the FIFO buffer.

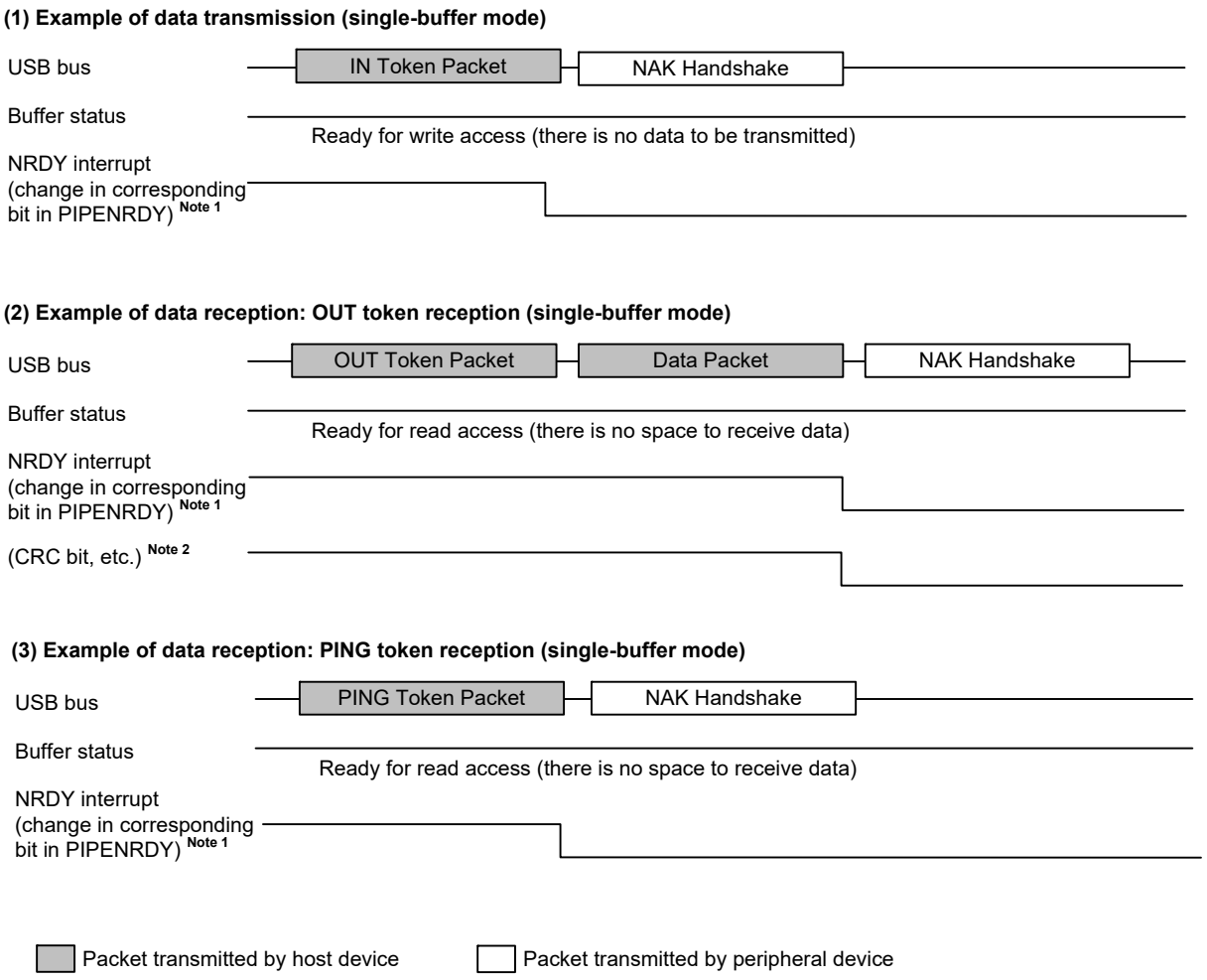
In this case, the USB module generates a NRDY interrupt request at the reception of the IN token and sets the PIPEnNRDY bit to 1.

[For the pipe set to the receiving direction]

- When an OUT token is received while there is no space available in the FIFO buffer.
  The USB module generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY bit to 1.
  However, during re-transmission (due to DATA-PID disagreement), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.

Figure 14-59 shows the Timing of NRDY Interrupt Generation When Function Controller Function is Selected.

Figure 14-59. Timing of NRDY Interrupt Generation When Function Controller Function is Selected



Notes 1. The PIPEnNRDY bit is set to 1 only while the PID bits for the target pipe are set to 1.

2. The CRC and OVRN bits change only while the target pipe is set to isochronous transfers.

14.4.3.3 BEMP interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USB module sets the corresponding PIPEnBEMP bit in the BEMPSTS register to 1. If the corresponding bit in the BEMPENB register has been set to 1 by software, the USB module sets the BEMP bit in the INTSTS0 register to 1 and generates a USB interrupt.

The following describes the conditions on which the USB module generates an internal BEMP interrupt request.

[For the pipe set to the transmitting direction]

- When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).
  In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP.
  However, the internal BEMP interrupt request is not generated on any of the following conditions.
- When software (DMAC) has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by setting the ACLRM or BCLR bit to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage while the function controller function is selected.

[For the pipe set to the receiving direction]

When data whose size is greater than the setting value of MaxPacketSize is successfully received.

In this case, the USB module generates a BEMP interrupt request, sets the corresponding PIPEnBEMP bit in the BEMPSTS register to 1, discards the received data, and modifies the setting of the PID bits of the corresponding pipe to STALL (11B).

Here, the USB module returns no response when used as the host controller, and returns STALL response when used as the function controller.

However, the internal BEMP interrupt request is not generated on any of the following conditions.

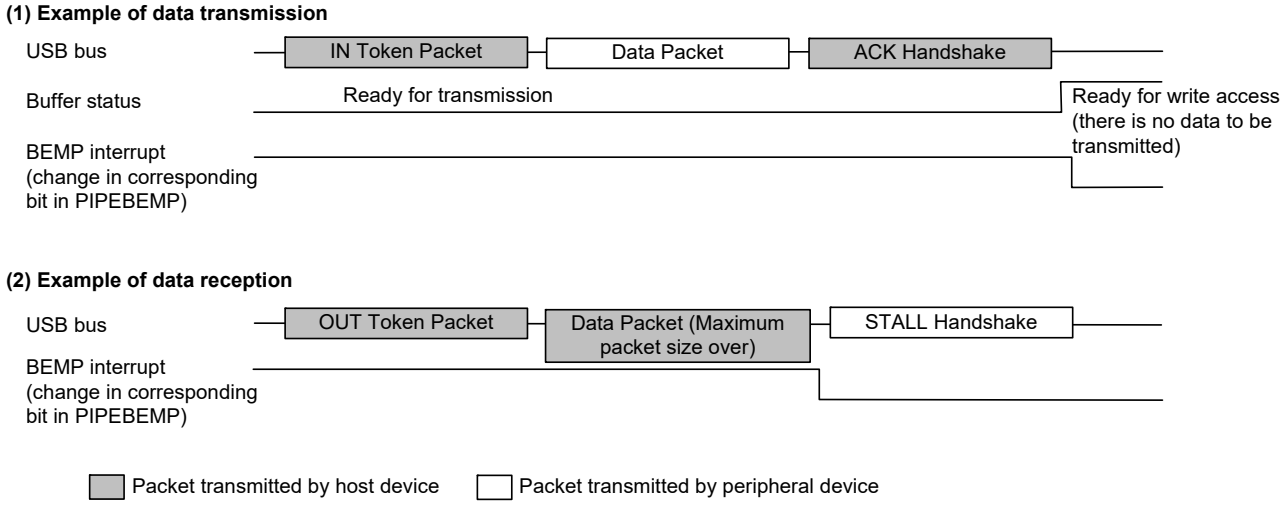
- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed.

Writing 0 to the PIPEnBEMP bit in the BEMPSTS register clears the status.

Writing 1 to the PIPEnBEMP bit in the BEMPSTS register has no effect.

Figure 14-60 shows the Timing of BEMP Interrupt Generation When Function Controller Function is Selected.

Figure 14-60. Timing of BEMP Interrupt Generation When Function Controller Function is Selected



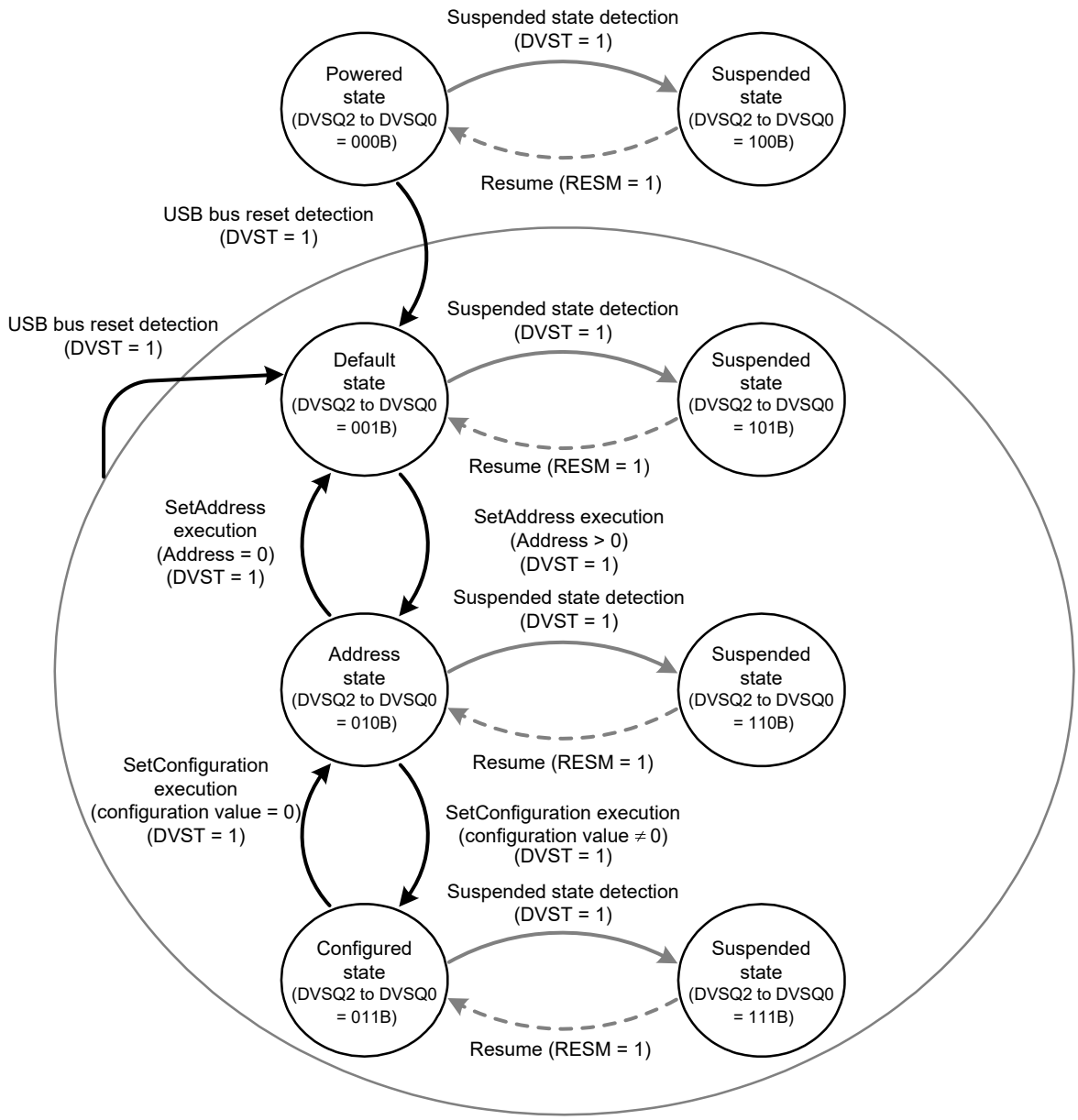
14.4.3.4 Device state transition interrupt

Figure 14-61 is a diagram of Device State Transitions in the USB module. The USB module controls device state and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using the INTENB0 register. The device state to which a transition was made can be confirmed using the DVSQ2 to DVSQ0 bits in the INTSTS0 register.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Device state can be controlled only when the function controller function is selected. The device state transition interrupts can also be generated only when the function controller function is selected.

Figure 14-61. Device State Transitions



Caution For the transition indicated in solid line, the DVST bit is set to 1.
For the transition indicated in dashed line, the RESM bit is set to 1.

14.4.3.5 Control transfer stage transition interrupt

Figure 14-62 is a diagram of Control Transfer Stage Transitions in the USB module. The USB module controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using the INTENB0 register. The transfer stage to which a transition was made can be confirmed using the CTSQ2 to CTSQ0 bits in the INTSTS0 register.

Control transfer stage transition interrupts are generated only when the function controller function is selected.

The control transfer sequence errors are listed below. If an error occurs, the PID bits in the DCPCTR register are set to 1xB (STALL response).

(1) During control read transfer

- An OUT token is received while no data has been transferred for the IN token at the data stage.
- An IN token is received at the status stage.
- A data packet with DATAPID = DATA0 is received at the status stage.

(2) During control write transfer

- An IN token is received while no ACK response has been returned for the OUT token at the data stage.
- A data packet with DATAPID = DATA0 is received for the first data packet at the data stage.
- An OUT token is received at the status stage.

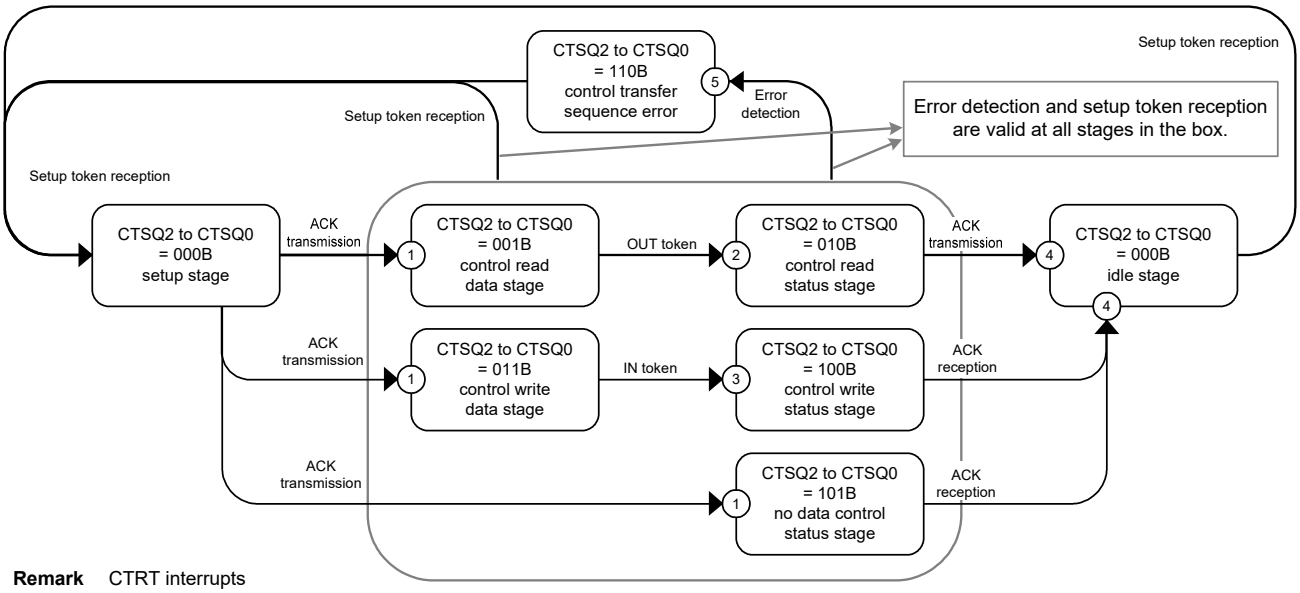
(3) During no-data control transfers

- An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (SERR = 1), the CTSQ2 to CTSQ0 bits = 110B value is retained until CTRT = 0 is written from the system (the interrupt status is cleared). Therefore, while the CTSQ2 to CTSQ0 bits = 110B is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB module retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

Figure 14-62. Control Transfer Stage Transitions



Remark CTRT interrupts
(1) Setup stage completed
(2) Control read transfer status stage transition
(3) Control write transfer status stage transition
(4) Control transfer completed
(5) Control transfer sequence error

14.4.3.6 Frame update interrupt

With the host controller function selected, an interrupt is generated at the timing when the frame number is updated. With the function controller function selected, an SOFR interrupt is generated when the frame number is updated.

When the function controller function is selected, the USB module updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

14.4.3.7 VBUS interrupt

When the VBUS pin level changes, a VBUS interrupt is generated. The level of the VBUS pin can be checked with the VBSTS bit in the INTSTS0 register. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the VBUS pin level.

14.4.3.8 Resume interrupt

When the function controller function is selected, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.

When the host controller function is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

14.4.3.9 Overcurrent interrupt

For USB port 0, an overcurrent interrupt (OVRCR interrupt) is generated when the UOVRCUR0 pin level has changed. The level of the UOVRCUR0 pin can be checked with the OVCMON1 and OVCMON0 bits in the SYSSTS0 register. The external power-supply IC can check whether overcurrent has been detected using the OVRCR interrupt.

For USB port 1, an OVRCR interrupt is generated when the UOVRCUR1 pin level has changed. The level of the UOVRCUR1 pin can be checked with the OVCMON1 and OVCMON0 bits in the SYSSTS1 register. The external power-supply IC can check whether overcurrent has been detected using the OVRCR interrupt.

14.4.3.10 Bus change interrupt

A bus change interrupt (BCHG interrupt) is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup when the host controller function is selected. The BCHG interrupt is generated regardless of whether the host controller function or function controller function is selected.

14.4.3.11 USB disconnection detection interrupt

A USB disconnection detection interrupt (DTCH interrupt) is generated when disconnection of the USB bus is detected while the host controller function is selected. The USB module detects bus disconnection based on USB Specification 2.0.

After detecting a DTCH interrupt, the USB module controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). Software should terminate all pipes in which communications are currently carried out for the pertinent port and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies the UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

14.4.3.12 Setup transaction normal response interrupt

A setup transaction normal response interrupt (SACK interrupt) is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller function selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

14.4.3.13 Setup transaction error interrupt

A setup transaction error interrupt (SIGN interrupt) is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller function selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

14.4.3.14 Connection interrupt

A connection interrupt (ATTCH interrupt) is generated when J-state or K-state of the full-speed/low-speed signal level is detected on the USB port for 2.5 μs with the host controller function selected. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 μs.
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 μs.

14.4.3.15 EOF error interrupt

An EOF error interrupt (EOFERR interrupt) is generated when it is detected that communication is not completed at the EOF2 timing prescribed in USB Specification 2.0.

After detecting an EOFERR interrupt, the USB module controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). Software should terminate all pipes in which communications are currently carried out for the pertinent port and perform re-enumeration of the pertinent port.

- Modifies the UACT bit in the DVSTCTR0 register for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

14.4.3.16 Portable device detection interrupt

A Portable Device detection interrupt is generated when the USB module detects a level change (high from low or low from high) in the PDDET output from the USB-PHY. When a Portable Device detection interrupt is generated, use software to repeat reading the PDDETSTS bit until the same value is read three or more times, and eliminate chattering.

14.4.4 Pipe control

Table 14-19 lists the Pipe Settings in the USB module. With USB data transfer, data transfer has to be carried out using the logic pipe called the endpoint. The USB module has five pipes that are used for data transfer.

Appropriate settings should be made for each of the pipes according to the specifications of the system.

Table 14-19. Pipe Settings

Register Name	Bit Name	Setting	Remarks
DCPCFG PIPECFG	TYPE	Specifies the transfer type	PIPE4, PIPE5: Can be set
	BFRE	Selects the BRDY interrupt mode	PIPE4, PIPE5: Can be set
	DBLB	Selects double buffer mode	PIPE4, PIPE5: Can be set
	DIR	Selects transfer direction	IN or OUT can be set
	EPNUM	Endpoint number	PIPE4, PIPE5: Can be set A value other than 0000B should be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	PIPE4, PIPE5: Can be set
DCPMAXP PIPEMAXP	DEVSEL	Selects a device	Referenced only when the host controller function is selected.
	MXPS	Maximum packet size	Compliant with the USB standard.
PIPEPERI	IFIS	Buffer flush	PIPE4 to PIPE7: Cannot be set
	IITV	Interval counter	PIPE4, PIPE5: Cannot be set PIPE6, PIPE7: Can be set (only when the host controller function has been selected)
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for PIPE4, PIPE5.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	SUREQCLR	SUREQ request clear	Can be set only for the DCP. Can be controlled only when the host controller function has been selected.
	ATREPM	Auto response mode	PIPE4, PIPE5: Can be set Can be set only when the function controller function has been selected.
	ACLRM	Auto buffer clear	PIPE4 to PIPE7: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit.
	SQSET	Sequence set	Sets the data toggle bit.
	SQMON	Sequence monitor	Monitors the data toggle bit.
	PBUSY	Pipe busy status	Monitors if the pipe is busy.
	PID	Response PID	Refer to **14.4.4.6 Response PID**.
PIPEnTRE	TRENB	Transaction counter enable	PIPE4, PIPE5: Can be set
	TRCLR	Current transaction counter clear	PIPE4, PIPE5: Can be set
PIPEnTRN	TRNCNT	Transaction counter	PIPE4, PIPE5: Can be set

14.4.4.1 Pipe control register switching procedures

The following bits in the pipe control registers can be modified only when USB communication is disabled (PID = NAK).

Registers that Should Not be Set in the USB Communication Enabled (PID = BUF) State:

- Bits in the DCPCFG and DCPMAXP registers
- The SQCLR and SQSET bits in the DCPCTR registers
- Bits in the PIPECFG, PIPEMAXP, and PIPEPERI registers
- The ATREPM, ACLRM, SQCLR, and SQSET bits in the PIPEnCTR register
- Bits in the PIPEnTRE and PIPEnTRN registers

In order to modify the above bits in the USB communication enabled (PID = BUF) state, follow the procedure shown below:

(1) A request to modify bits in the pipe control register occurs.
(2) Modify the PID corresponding to the pipe to NAK.
(3) Wait until the corresponding PBUSY bit is cleared to 0.
(4) Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE bit in the CFIFOSEL, D0FIFOSEL, and D1FIFOSE registers.

Registers that Should Not be Set When CURPIPE in FIFO-PORT is set:

- Bits in the DCPCFG and DCPMAXP registers
- Bits in the PIPECFG, PIPEMAXP and PIPEPERI registers

In order to modify pipe information, the CURPIPE bits in the port select registers should be set to a pipe other than the pipe to be modified. For the DCP, the buffer should be cleared using BCLR after the pipe information is modified.

14.4.4.2 Transfer types

The TYPE bits in the PIPEPCFG register are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

- DCP: No setting is necessary (fixed at control transfer).
- PIPE4, PIPE5: These should be set to bulk transfer.
- PIPE6, PIPE7: These should be set to interrupt transfer.

14.4.4.3 Endpoint number

The EPNUM bits in the PIPEPCFG register are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- DCP: No setting is necessary (fixed at end point 0).
- PIPE4 to PIPE7: The endpoint numbers from 1 to 15 should be selected and set.
  These should be set so that the combination of the DIR bit and EPNUM bits is unique.

14.4.4.4 Maximum packet size setting

The MXPS bits in the DCPMAXP and PIPEMAXP registers are used to specify the maximum packet size for each pipe. DCP and PIPE4, PIPE5 can be set to any of the maximum pipe sizes defined by the USB specification. For PIPE6, PIPE7, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

- DCP: Set 8, 16, 32, or 64.
- PIPE4, PIPE5: Set 8, 16, 32, or 64 when using bulk transfer.
- PIPE6, PIPE7: Set a value between 1 and 64.

14.4.4.5 Transaction counter (for PIPE4, PIPE5 in reading direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB module recognizes that the transfer has ended. Two transaction counters are provided: one is the TRNCNT register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the PID of the corresponding PIPE is set to NAK and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the TRCLR bit. The information read from TRNCNT differs depending on the setting of the TRENB bit.

- TRENB = 0: The specified transaction counter value can be read.
- TRENB = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

14.4.4.6 Response PID

The PID bits in DCPCTR and PIPEnCTR are used to set the response PID for each pipe.
The following shows the USB module operation with various response PID settings:

(1) Response PID settings when the host controller function is selected

The response PID is used to specify the execution of transactions.

- NAK setting: Using pipes is disabled. No transaction is executed.
- BUF setting: Transactions are executed based on the status of the buffer memory.

For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.

- STALL setting: Using pipes is disabled. No transaction is executed.

Caution Setup transactions for the DCP are set with the SUREQ bit.

(2) Response PID settings when the function controller function is selected

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is always returned in response to the generated transaction.
- BUF setting: Responses are made to transactions according to the status of the buffer memory.
- STALL setting: The STALL response is always returned in response to the generated transaction.

Caution For setup transactions, an ACK response is always returned regardless of the PID setting, and the USB request is stored in the register.

The USB module may write to the PID bits, depending on the results of the transaction as described below.

(3) When the host controller function has been selected and the response PID is set by hardware NAK setting:

In the following cases, PID = NAK is set and issuing of tokens is automatically stopped:

- When an NRDY interrupt is generated.(For details, refer to **14.4.3.2 NRDY interrupt**.)
- If a short packet is received when the SHTNAK bit in the PIPECFG register has been set to 1 for bulk transfer.
- If the transaction counting ends when the SHTNAK bit has been set to 1 for bulk transfer.

BUF setting: There is no BUF writing by the USB module.
STALL setting: In the following cases, PID = STALL is set and issuing of tokens is automatically stopped:

- When STALL is received in response to the transmitted token.
- When the size of the receive data packet exceeds the maximum packet size.

(4) When the function controller function has been selected and the response PID is set by hardware

NAK setting: In the following cases, PID = NAK is set and NAK is always returned in response to transactions:

- When the SETUP token is received normally (DCP only).
- If the transaction counting ends or a short packet is received when the SHTNAK bit in PIPECFG has been set to 1 for bulk transfer.

BUF setting: There is no BUF writing by the USB module.

STALL setting: In the following cases, PID = STALL is set and STALL is always returned in response to transactions:

- When a maximum packet size exceeded error is detected in the received data packet.
- When a control transfer sequence error has been detected (DCP only).

14.4.4.7 Data PID sequence bit

The USB module automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in the DCPCTR and PIPEnCTR registers. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The SQCLR bit in the DCPCTR register and the SQSET bit in the PIPEnCTR register can be used to change the data PID sequence bit.

When the function controller function has been selected and control transfer is used, the USB module automatically sets the sequence bit when a stage transition is made. DATA0 is returned when the setup stage is ended and DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller function has been selected and control transfer is used, the sequence bit should be set by software at a stage transition.

For the ClearFeature request transmission or reception, the data PID sequence bit should be set by software regardless of whether the host controller function or function controller function is selected.

14.4.4.8 Response PID = NAK function

The USB module has a function that disables pipe operation (PID response = NAK) at the timing at which the final data packet of a transaction is received (the USB module automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the SHTNAK bit in the PIPECFG register to 1.

When the double buffer mode is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, software should set the pipe to the enabled state again (PID response = BUF).

The response PID = NAK function can be used only when bulk transfers are used.

14.4.4.9 Auto response mode

With the pipes for bulk transfer (PIPE4, PIPE5), when the ATREPM bit in the PIPEnCTR register is set to 1, a transition is made to auto response mode. During an OUT transfer (DIR = 0), OUT-NAK mode is entered, and during an IN transfer (DIR = 1), null auto response mode is entered.

14.4.4.10 OUT-NAK mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is retuned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled.

14.4.4.11 Null auto response mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, the INBUFM bit = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, the buffer should be emptied with the ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (about 10 μs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

14.4.5 FIFO buffer memory

14.4.5.1 FIFO buffer memory

The USB module has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB module. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB module (SIE side).

(1) Buffer Status

Tables 14-20 and 14-21 show the buffer status in the USB module. The buffer memory status can be confirmed using the BSTS bit in the DCPCTR register and the INBUFM bit in the PIPEnCTR register. The access direction for the buffer memory can be specified using either the DIR bit in the PIPEnCFG register or the ISEL bit in the CFIFOSEL register (when DCP is selected).

The INBUFM bit is valid for PIPE4, PIPE5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the BSTS bit to monitor the buffer memory status on the CPU side and the INBUFM bit to monitor the buffer memory status on the SIE side. When the BEMP interrupt may not show the buffer empty status because the write access to the FIFO port by the CPU (DMAC) is slow, software can use the INBUFM bit to confirm the end of transmission.

Table 14-20. Buffer Status Indicated by BSTS Bit

ISEL or DIR	BSTS	Buffer Memory Status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. Note that when a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	The transmission has been completed. CPU write is allowed.

Table 14-21. Buffer Status Indicated by INBUFM Bit

DIR	INBUFM	Buffer Memory Status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been completed. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

(2) FIFO Buffer Clearing

Table 14-22 shows the clearing of the FIFO buffer memory by the USB module. The buffer memory can be cleared using the BCLR, DCLRM, and ACLRM bits.

Table 14-22. Buffer Clearing Methods

FIFO Buffer Clearing Mode	Clearing Buffer Memory on CPU Side	Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data	Auto Buffer Clear Mode for Discarding All Received Packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(3) Auto Buffer Clear Mode Function

With the USB module, all received data packets are discarded if the ACLRM bit in the PIPEnCTR register is set to 1. If a correct data packet has been received, the ACK response is returned to the host controller. The auto buffer clear mode function can be set only in the buffer memory reading direction.

If the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.

However, an access cycle of at least 100 ns is required for the internal hardware sequence processing time between the ACLRM bit = 1 and the ACLRM bit = 0.

(4) Buffer Memory Specifications (Single or Double Setting)

Either a single or double buffer configuration can be selected for PIPE4 and PIPE5, using the DBLB bit in the PIPEnCFG register.

14.4.5.2 FIFO port functions

Table 14-23 shows the FIFO Port Function Settings for the USB module. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in the CFIFOCTR or DnFIFOCTR register should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing.

In read access, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (DTLN = 0), so the BCLR bit in the register should be used to clear the buffer. The length of the receive data can be confirmed using the DTLN bits in the CFIFOCTR or DnFIFOCTR register.

Table 14-23. FIFO Port Function Settings

Register Name	Bit Name	Function	Remark
CFIFOSEL DnFIFOSEL	RCNT	Selects DTLN read mode.	
	REW	Buffer memory rewind (re-read, rewrite).	
	DCLRM	Automatically clears receive data for a specified pipe after the data has been read.	Only for DnFIFO.
	DREQE	Enables DMA transfers.	Only for DnFIFO.
	MBW	Selects the FIFO port access bit width.	
	BIGEND	Selects FIFO port endian.	
	ISEL	FIFO port access direction.	Only for DCP.
	CURPIPE	Selects the current pipe.	
CFIFOCTR DnFIFOCTR	BVAL	Ends writing to the buffer memory.	
	BCLR	Clears the buffer memory on the CPU side.	
	DTLN	Checks the length of receive data.	

(1) FIFO Port Selection

Table 14-24 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE bits in the CFIFOSEL or DnFIFOCTR register. After the pipe is selected, whether the written value can be correctly read from the CURPIPE bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FIFO port can be accessed after FRDY = 1 is checked.

In addition, the bus width to be accessed should be selected using the MBW bit. The buffer memory access direction conforms to the DIR bit in the PIPEnCFG register. Only for the DCP, the ISEL bit determines the direction.

Table 14-24. FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register (CFIFOM)
PIPE4 to PIPE7	CPU access	CFIFO port register (CFIFOM) D0FIFO/D1FIFO port register (D0FIFOM/D1FIFOM)
	DMA access	DMA transfer D0FIFO/D1FIFO port register (D0FIFO/D1FIFO)

(2) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the CFIFOSEL or DnFIFOSEL register is used for this processing. If a pipe is selected through the CURPIPE bits in the CFIFOSEL or DnFIFOSEL register with the REW bit set to 1, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, FRDY = 1 should be checked after selecting a pipe.

14.4.5.3 DMA transfers (D0FIFO and D1FIFO ports)

(1) Overview of DMA Transfers

For PIPE4 to PIEP7, the FIFO port can be accessed using the DMA. When accessing the buffer for the pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

The unit of transfer to the FIFO port should be selected using the MBW bit in the DnFIFOSEL register and the pipe targeted for the DMA transfer should be selected using the CURPIPE bits. The selected pipe should not be changed during the DMA transfer.

(2) Automatic Recognition of End of DMA Transfer

Controlling input of the DMA transfer end signal allows the USB module to end writing of FIFO data using a DMA transfer. The buffer memory goes to transmit-enabled (same status when BVAL = 1) when the transfer end signal is sampled.

(3) DnFIFO Auto Clear Mode (D0FIFO and D1FIFO Port Reading Direction)

If 1 is set in the DCLRM bit in the DnFIFOSEL register, the USB module automatically clears the buffer memory of the selected pipe when reading of data from the buffer memory has been completed.

Table 14-25 shows the Packet Reception and Buffer Memory Clearing Processing by Software for each of the various settings. As shown in Table 14-25, the buffer clearing conditions depend on the value set in the BFRE bit. Using the DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.

The DnFIFO auto clear mode can be set only in the buffer memory reading direction.

Table 14-25. Packet Reception and Buffer Memory Clearing Processing by Software

Register Setting / Buffer Status When Packet is Received	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Automatically cleared	Automatically cleared	Automatically cleared	Automatically cleared
Zero-length packet reception	Cleared by software	Cleared by software	Automatically cleared	Automatically cleared
Normal short packet reception	Automatically cleared	Cleared by software	Automatically cleared	Automatically cleared
Transaction count end	Automatically cleared	Automatically cleared	Automatically cleared	Automatically cleared

14.4.6 Control transfers (DCP)

In the data stage of control transfers, data is transferred using the default control pipe (DCP).

The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

14.4.6.1 Control transfers when host controller function is selected

(1) Setup Stage

USQREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the SUREQ and bit in the DCPCTR register transmits the specified data for setup transactions. Upon completion of the transaction, the SUREQ bit is cleared to 0. The above USB request registers should not be modified while SUREQ = 1.

After the attached state of the connected function device is detected, the first setup transaction for the device should be issued by using the sequence described above with the DEVSEL bits in the DCPMAXP register set to 0 and the USBSPD and RTPORT bits in the DEVADD0 register set appropriately.

After the connected function device is shifted to the Address state, setup transactions should be issued by using the sequence described above with the assigned USB address set in the DEVSEL bits and the bits in the DEVADDn register corresponding to the specified USB address set appropriately. For example, when the DEVSEL bit in the PIPEMAXP register = 2H, make appropriate settings in the DEVADD2 register; when the DEVSEL bit = 5H, make appropriate settings in the DEVADD5 register.

When the setup transaction data has been sent, an interrupt request is generated according to the response received from the peripheral device (SIGN1 or SACK bit in the INTSTS1 register), by means of which the result of the setup transactions can be confirmed.

A data packet of DATA0 (USB request) is transmitted as the data packet for a setup transaction regardless of the setting of the SQMON bit in the DCPCTR register.

(2) Data Stage

Data is transferred using the DCP buffer memory.

The access direction of the DCP buffer memory should be specified using the ISEL bit in the CFIFOSEL register. The transfer direction should be specified using the DIR bit in the DCPCFG register.

For the first data packet of the data stage, the data PID should be transferred as DATA1. Set data PID = DATA1 in the SQSET bit and the PID bits = BUF in the DCPCFG register. Completion of data transfer is detected using the BRDY or BEMP interrupt.

For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, software should control so as to send a zero-length packet at the end.

(3) Status Stage

Zero-length packet data is transferred in the direction opposite to that in the data stage. As in the data stage, data is transferred using the DCP buffer memory. Transactions are done in the same manner as the data stage.

For the data packets of the status stage, the data PID should be set to DATA1 using the SQSET bit in the DCPCFG register.

For reception of a zero-length packet, the received data length should be confirmed using the DTLN bits in the CFIFOCTR register after a BRDY interrupt is generated, and the buffer memory should then be cleared using the BCLR bit.

14.4.6.2 Control transfers when function controller function is selected

(1) Setup Stage

The USB module always sends an ACK response for a correct setup packet targeted to the USB module. The operation of the USB module in the setup stage is described below.

(a) When receiving a new setup packet, the USB module sets the following bits.
- Set the VALID bit in the INTSTS0 register to 1.
- Set the PID bits in the DCPCTR register to NAK.
- Set the CCPL bit in the DCPCTR register to 0.

(b) When receiving a data packet right after the setup packet, the USB module stores the USB request parameters in the USBREQ, USBVAL, USBINDX, and USBLENG registers.

Response processing with respect to the control transfer should always be carried out after setting VALID = 0. In VALID = 1 state, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the function of the VALID bit, the USB module can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.

In addition, the USB module automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For control on the stages of the USB module, refer to **Figure 14-62**.

(2) Data Stage

Data transfers corresponding to received USB requests should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the ISEL bit in the CFIFOSEL register. If the transfer data is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status Stage

Control transfers are terminated by setting the CCPL bit to 1 while the PID bits in the DCPCTR register are set to BUF. After the above settings have been made, the USB module automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

[For control read transfers]

The USB module receives a zero-length packet and transmits an ACK response.

[For control write transfers and no-data control transfers]

The USB module transmits a zero-length packet and receives an ACK response from the USB host.

(4) Control Transfer Auto Response Function

The USB module automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.
- Any transfer other than a control read transfer: bmRequestType ≠ 00H
- Request error: wIndex ≠ 00H
- Any transfer other than a no-data control transfer: wLength ≠ 00H
- Request error: wValue > 7FH
- Control transfer of a device state error: DVSQ2 to DVSQ2 bits = 011B (Configured)

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

14.4.7 Bulk transfers (PIPE4, PIPE5)

The buffer memory usage (single/double buffer setting) can be selected for bulk transfers. The USB provides the following functions for bulk transfers.

- BRDY interrupt function (BFRE bit: refer to **14.4.3.1 BRDY interrupt**.)
- Transaction count function
  (TRENB, TRCLR, and TRNCNT bits: refer to **14.4.4.5 Transaction counter (for PIPE4, PIPE5 in reading direction**)
- Response PID = NAK function (SHTNAK bit: refer to **14.4.4.8 Response PID = NAK function**)
- Auto response mode (ATREPM bit: refer to **14.4.4.9 Auto response mode**)

14.4.8 Interrupt transfers (PIPE6, PIPE7)

When the function controller function is selected, the USB module carries out interrupt transfers in accordance with the timing controlled by the host controller.

When the host controller function is selected, the timing of issuing a token can be specified using the interval counter.

14.4.8.1 Interval counter during interrupt transfers when host controller function is selected

For interrupt transfers, intervals between transactions are set in the IITV bits in PIPEPERI. The USB controller issues interrupt transfer tokens based on the specified intervals.

(1) Counter Initialization

The USB controller initializes the interval counter under the following conditions.

- Power-on reset:

The IITV bits are initialized.

- Buffer memory initialization using the ACLRM bit:

The IITV bits are not initialized but the count value is initialized. Setting the ACLRM bit in the PIPEnCTR register to 0 starts counting from the value set in the IITV bits.

Note that the interval counter is not initialized in the following case.

- USB bus reset or USB suspended

The IITV bits are not initialized. Setting 1 to the UACT bit in the DVSTCTR0 register starts counting from the value before entering the USB bus reset state or USB suspended state.

(2) Operation when Transmission/Reception is Impossible at Token Issuance Timing

The USB module cannot issue tokens even at token issuance timing in the following cases. In such a case, the USB module attempts transactions at the subsequent interval.

- When the PID is set to NAK or STALL.
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the transmitting (OUT) direction.

14.4.9 SOF interpolation function

When the function controller function is selected and if data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB module interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in the SYSCFG register have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB module supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing

If an SOF packet is missing, the FRNM bit in the FRMNUM register is not updated.

14.4.10 Pipe schedule

14.4.10.1 Conditions for generating a transaction

When the host controller function is selected and the UACT bit has been set to 1, the USB module generates a transaction under the conditions shown in Table 14-26.

Table 14-26. Conditions for Generating a Transaction

Transaction	Conditions for Generation				
	DIR	PID	IITV0	Buffer State	SUREQ
Setup	– Note	– Note	– Note	– Note	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	– Note
	OUT	BUF	Invalid	Transmit data exists	– Note
Interrupt transfer	IN	BUF	Valid	Receive area exists	– Note
	OUT	BUF	Valid	Transmit data exists	– Note

Note Symbols (–) in the table indicate that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.

14.4.10.2 Transfer schedule

This section describes the transfer scheduling within a frame of the USB module. After the USB module sends an SOF, the transfer is carried out in the sequence described below.

(1) Execution of periodic transfers

A pipe is searched in the order of PIPE6 → PIPE7, and then, if there is a pipe for interrupt transfer transaction can be generated, the transaction is generated.

(2) Setup transactions for control transfers

The DCP is checked, and if a setup transaction is possible, it is sent.

(3) Execution of bulk transfers, control transfer data stages, and control transfer status stages

A pipe is searched in the order of DCP → PIPE4 → PIPE5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated. When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

14.4.10.3 Enabling USB communication

Setting the UACT bit in the DVSTCTR register to 1 initiates SOF transmission and transaction generation is enabled. Setting the UACT bit to 0 stops SOF transmission and a suspend state is entered. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF is sent.

14.4.11 Controlling battery charging detection

It is possible to control the processing for data contact detection (D+ line contact check), primary detection (charger detection), and secondary detection (charger verification) in compliance with Battery Charging Specification Revision 1.2.

By executing these processes according to charger detection algorithms, it is possible to determine whether the connected device is a standard downstream port, charging downstream port, or dedicated charging port (function BC connection detection function; USB port 0) as a portable device. It is also possible to operate as a charging downstream port or dedicated charging port for a portable device (host BC connection detection function; USB ports 0 and 1) **Note**.

The above data contact detection, primary detection, and secondary detection are each detected by the interface detection circuit for BC connection detection that is provided along with the USB transceiver. This circuit has a function to connect necessary voltage sources (V_{DP_SRC}, V_{DM_SRC}), current source (I_{DP_SRC}), voltage (V_{DAT_REF}) detection function, and UDP and UDM pins via a resistor (R_{DCP_DAT}) to perform connection detection compliant to Battery Charging Specification Revision 1.2. These can be controlled and monitored by setting the bits in BC control register n (USBBCCTRLn) (n = 0, 1).

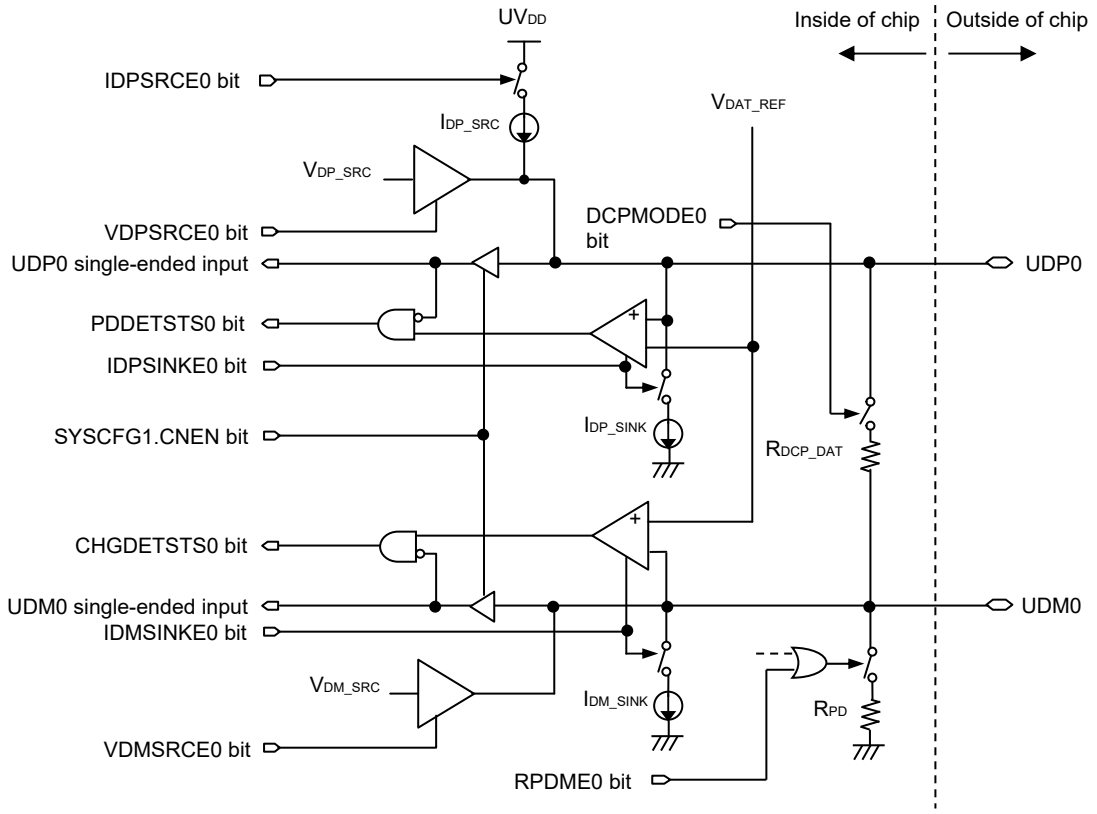
When using the BC connection detection function, specify PXXCON = 0 and VDDUSBE = 1 to externally apply power to the UV_{DD} pin, or specify PXXCON = 1 and VDDUSBE = 1 (internal USB power supply is used) to internally generate 3.3 V.

The temperature sensor cannot be used and A/D conversion that uses the internal reference voltage cannot be performed when using the BC connection detection function.

Figure 14-63 and **Figure 14-64** show the interface circuit for BC connection detection.

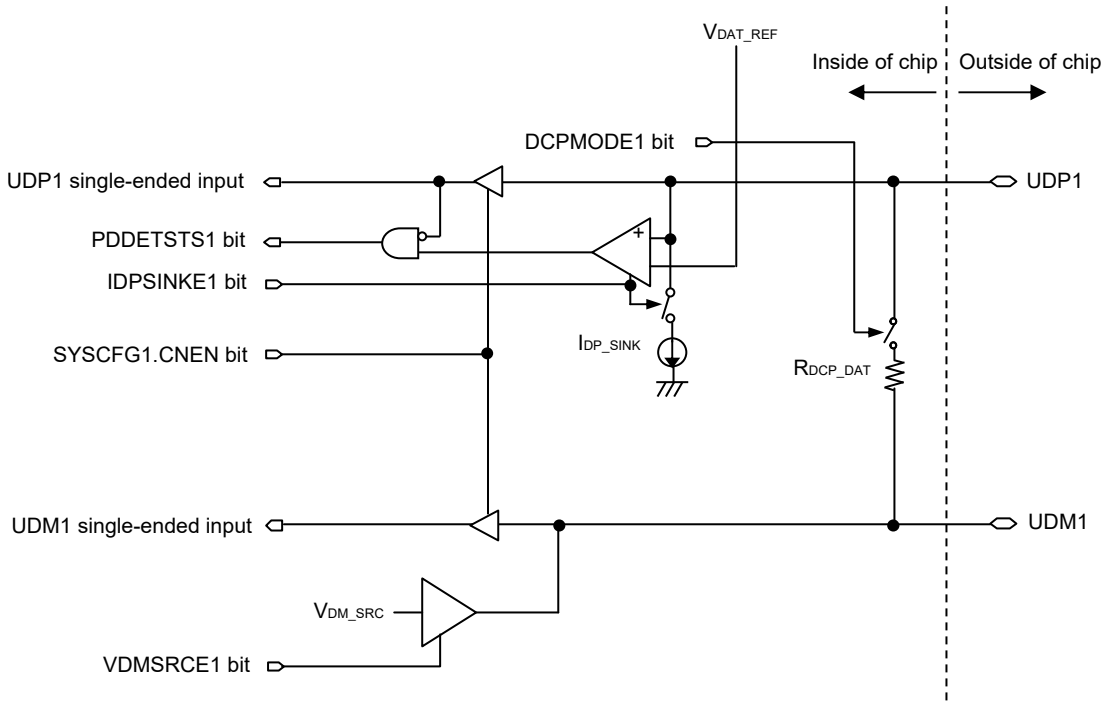
Note The host function is not available in the R5F10K products.

Figure 14-63. BC Connection Detection Interface Circuit (USB Port 0) with Host Note/Function BC Connection Detection Function



Note The host function is not available in the R5F10K products.

Figure 14-64. BC Connection Detection Interface Circuit (USB Port 1 [Note]) with Host BC Connection Detection Function



Note The host function is not available in the R5F10K products.

14.4.12 Battery charging connection detection optional functions

For extensibility of the battery charging specifications, the following optional functions are added to control connection detection.

- USB port voltage output function (four patterns)
  As an optional function of the host BC connection detection function, this function can divide 5 V applied to the UV_{BUS} pin and output to the USB port. Furthermore, this function can detect whether the voltage of the USB port has risen and dropped while the divided voltage is output. It can also detect whether the signal has conflicted with the connected USB port output of the device. [Note]

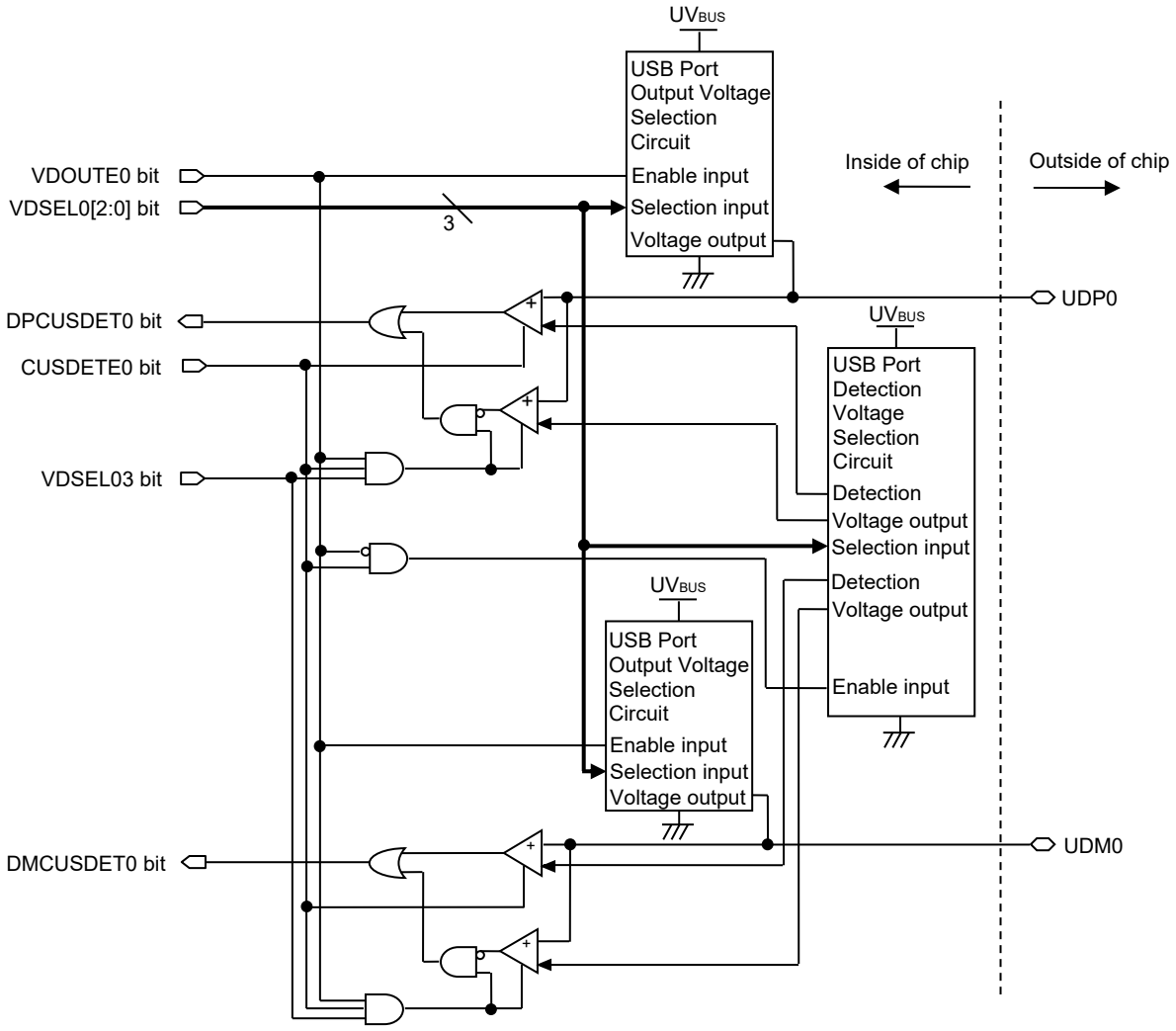
- USB port voltage detection function (16 stages)
  As an optional function of the function BC connection detection function, this function can detect the voltage level to be input to the USB port, using the 16-stage reference voltage that is obtained by dividing 5 V to be applied to the UV_{BUS} pin.

After power is supplied to the UV_{DD} pin (using an external supply or internal power supply for the USB) and the voltage is applied to the UV_{BUS} pin, these functions can control various functions and detection results by setting BC option control register n (USBBCOPTn) (n = 0, 1).

Note The host function is not available in the R5F10K products.

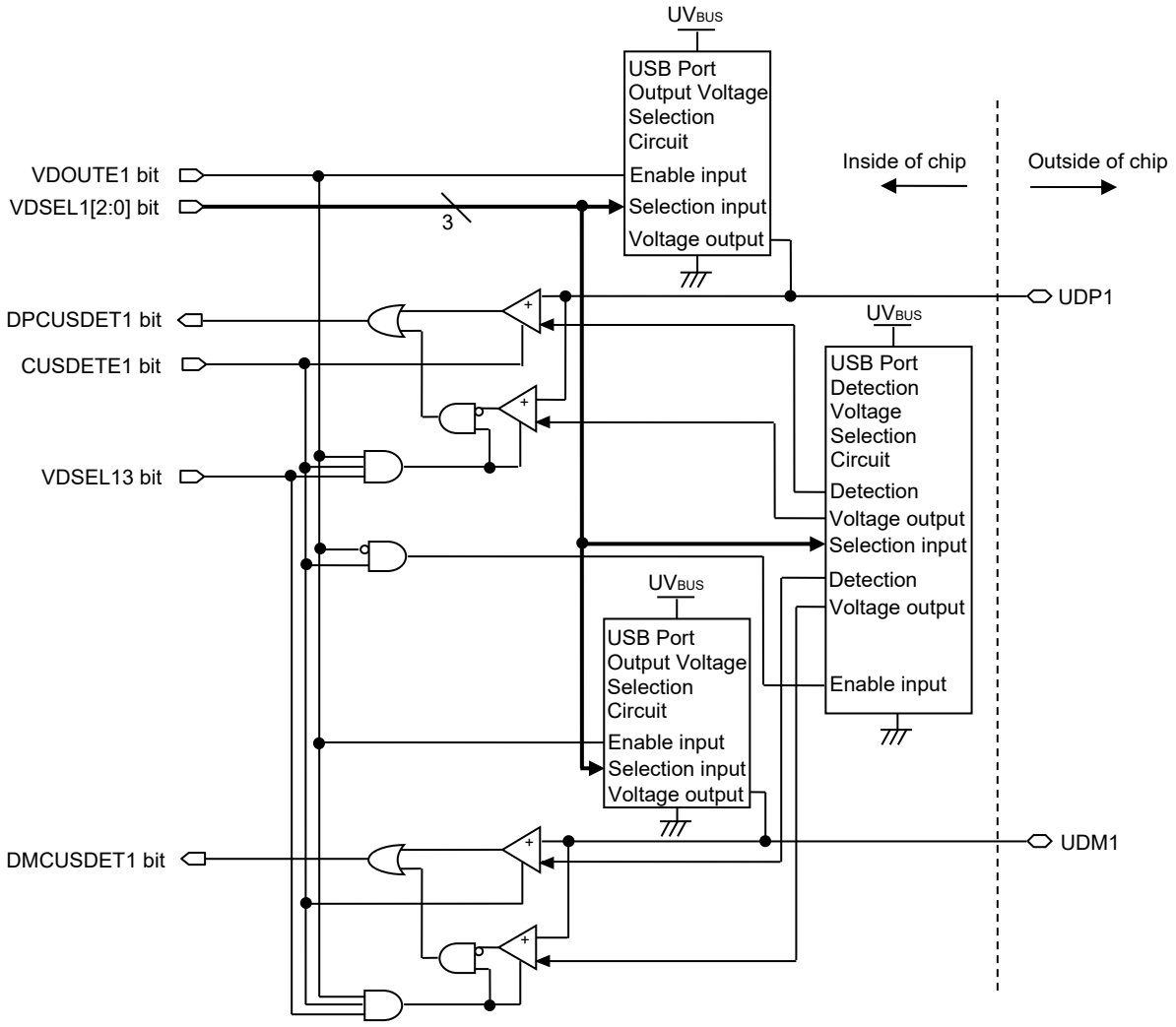
Figure 14-65 and **Figure 14-66** show the Interface Circuit for BC Connection Detection Optional Functions.

Figure 14-65. BC Connection Detection Optional Function Interface Circuit (USB Port 0) with Host Note/Function BC Connection Detection Function



Note The host function is not available in the R5F10K products.

Figure 14-66. BC Connection Detection Optional Function Interface Circuit (USB Port 1 Note) with Host BC Connection Detection Function



Note USB port 1 is not available in the R5F10K products.

14.4.13 Battery charging detection processing

It is possible to control to the processing for Data Contact Detection (D+ line contact check), Primary Detection (Charger detection), and Secondary Detection (Charger verification), which are defined by the Battery Charging Specification.

The following shows required operations for a Peripheral Device and a Host Device, individually.

14.4.13.1 Processing when function controller is selected

The following processing is required when operating the USB module as a Portable Device for Battery Charging.

(1) Detect when the data lines (D+/D-) have made contact and start the processing for Primary Detection.

(2) After Primary Detection starts, wait 40 ms for masking, and then check the D- voltage level to confirm the Primary Detection result.

(3) If the Charger is detected during Primary Detection, also start Secondary Detection.

(4) After Secondary Detection starts, wait 40 ms for masking, and then check the D+ voltage level to confirm the Secondary Detection result.

For the above step (1), after VBUS is detected using the VBINT interrupt and the VBSTS bit, wait for 300 ms to 900 ms by software, and then set the VDPSRCE and IDMSINKE bits in the USBBCCTRL register. Or set the IDPSRCE bit, and after a change from high to low on the D+ line is detected using the LNST bits, clear the IDPSRCE bit and set the VDPSRCE and IDMSINKE bits. The VDPSRCE and IDMSINKE bits must be set at the same time. Note 1

For the above step (2), set the VDPSRCE and IDMSINKE bits and wait for 40 ms by software, and then use the CHGDETSTS bit to verify the Primary Detection result. Note 2

For the above step (3), verify that the Charger is detected if the CHGDETSTS bit is set in the above step (2), and then clear the VDPSRCE and IDMSINKE bits and set the VDMSRCE and IDPSINKE bits.

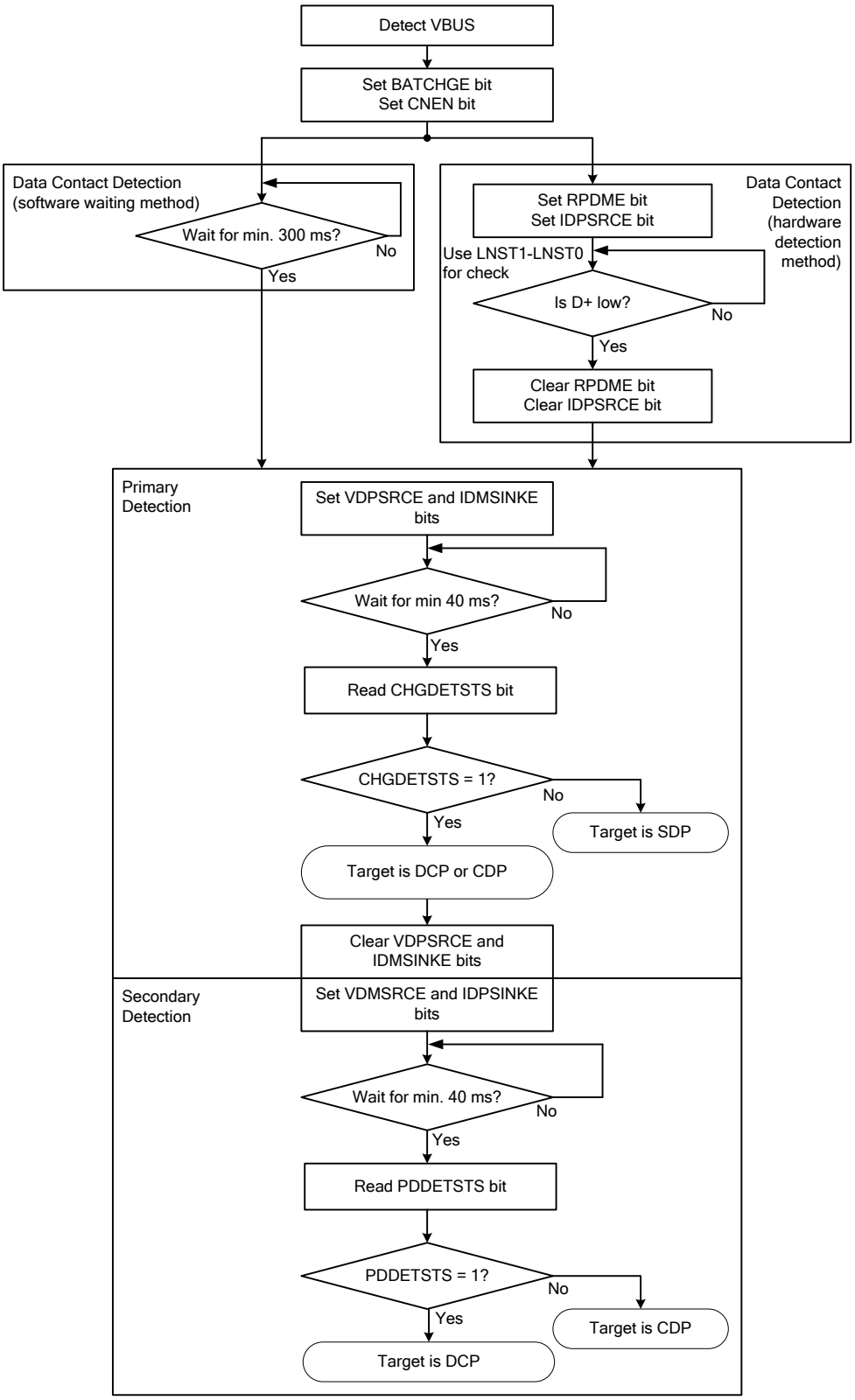
For the above step (4), set the VDMSRCE and IDPSINKE bits and wait for 40 ms by software, and then use the PDDETSTS bit to verify the Secondary Detection result.

Figure 14-67 shows the Process Flow for Operating as Portable Device.

Notes 1. The Battery Charging Specification describes two implementation methods of the process flow for Data Contact Detection (D+/D- line contact check). One of the methods is to detect a change to logic low due to the pull-down resistor of the Host Device when the D+/D- lines have made contact with the target while the D+ line is held at logic high by applying a current of 7 to 13 uA on the D+ line.
The other method is to wait for 300 ms to 900 ms after VBUS is detected.

2. During Primary Detection, when the voltage on the D- line is detected to be 0.25 V to 0.4 V or above and 0.8 V to 2.0 V or below, the target device is recognized as the Host Device for Battery Charging (Charging Downstream Port).
When using a PHY in which the 0CHGDETSTS bit only indicates that the voltage on the D- line is 0.25 V to 0.4 V or above, add the processing to check that the voltage on D- line is 0.8 V to 2.0 V or below using the LNST bits, as necessary.

Figure 14-67. Process Flow for Operating as Portable Device



14.4.13.2 Processing when host controller is selected

The following processing is required when operating the USB module as a Charging Downstream Port for Battery Charging.

(1) Start driving the VBUS.
(2) Enable the Portable Device detection circuit.
(3) Monitor the Portable Device detection signal, and start driving the D- line if the detection signal is high.
(4) Detect when the Portable Device detection signal is low level and stop driving the D- line.

Or perform the following processing.

(A) After disconnection is detected, start driving the D- line within 200 ms.
(B) After connection is detected, stop driving the D- line within 10 ms.

The D- line must be driven to allow the Portable Device to detect the Primary Detection described in section 14.3.13.1. The above steps (1) to (4) apply when the Portable Device detection function is provided by hardware. This method is to drive the D- line when the Portable Device is detected. The above steps (A) and (B) apply when the Portable Device function is not provided or used by hardware. Regardless of detection of the Portable Device, the D- line is driven in the Dis-Connect state and the line is not driven in the Connect state. In the Battery Charging Specification, either of these methods can be used.

For the above steps (3) and (4), after a change in the Portable Device detection signal is detected using the PDDETINT interrupt, the current signal state can be confirmed by reading the PDDETSTS bit.

The above steps (A) and (B) can be performed only in a software timer.

Figure 14-68 shows the above process flow for the above steps (1) to (4) and Figure 14-69 shows the process flow for the above steps (A) to (B).

Figure 14-68. Process Flow for Operating as Charging Downstream Port (Steps (1) to (4))

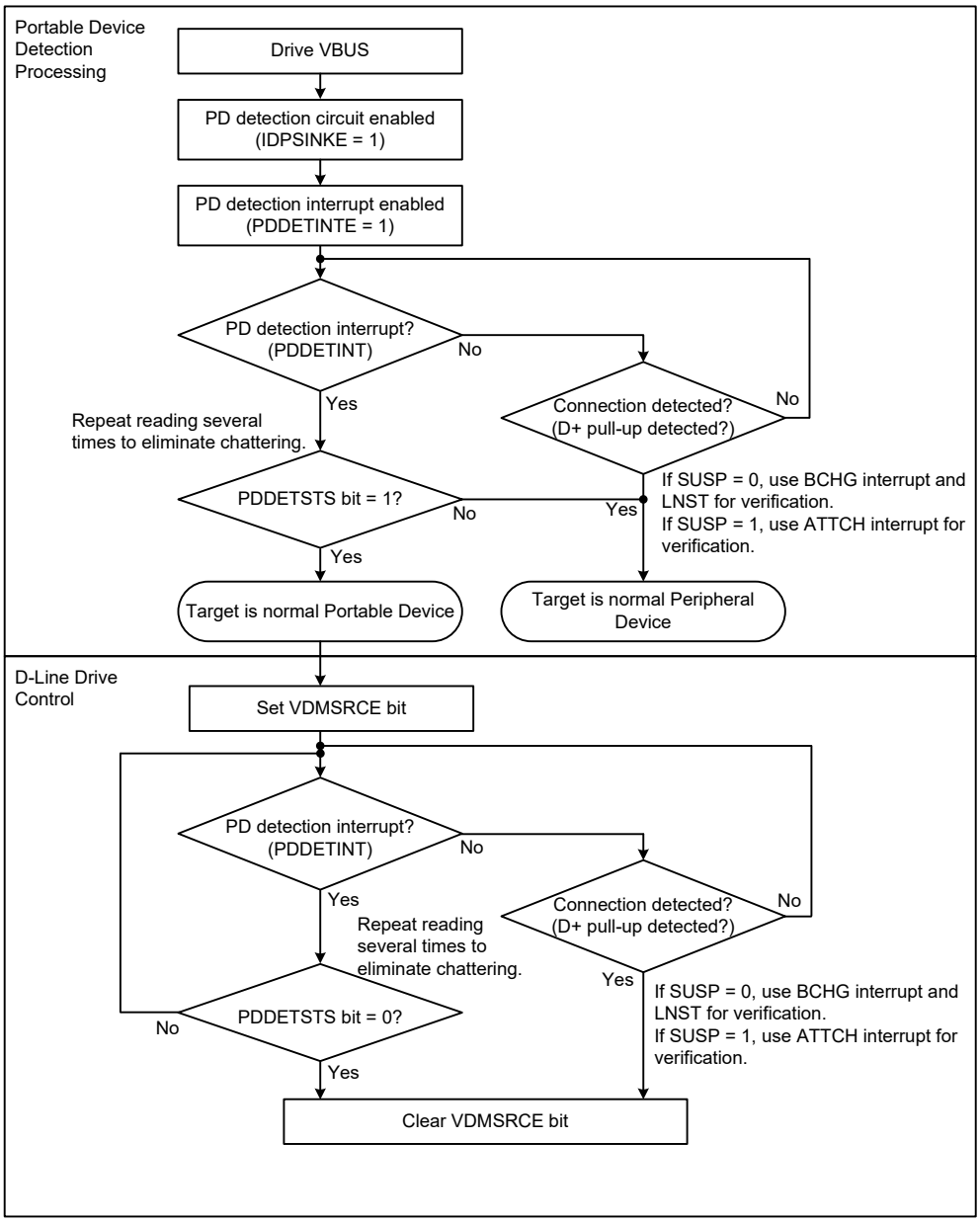
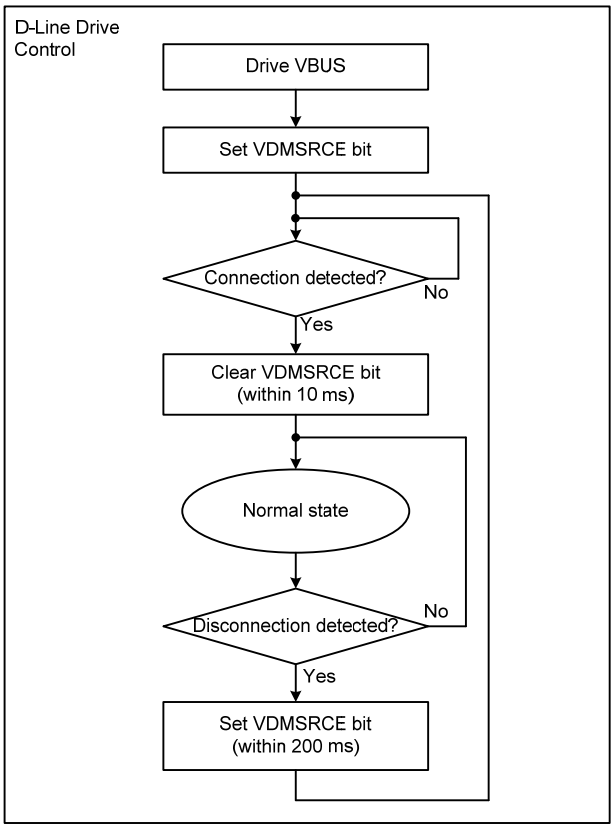


Figure 14-69. Process Flow for Operating as Charging Downstream Port (Steps (A) to (B))



CHAPTER 15 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR

15.1 Functions of Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator has the following functions.

- 16 bits × 16 bits = 32 bits (Unsigned)
- 16 bits × 16 bits = 32 bits (Signed)
- 16 bits × 16 bits + 32 bits = 32 bits (Unsigned)
- 16 bits × 16 bits + 32 bits = 32 bits (Signed)
- 32 bits ÷ 32 bits = 32 bits, 32-bits remainder (Unsigned)

15.2 Configuration of Multiplier and Divider/Multiply-Accumulator

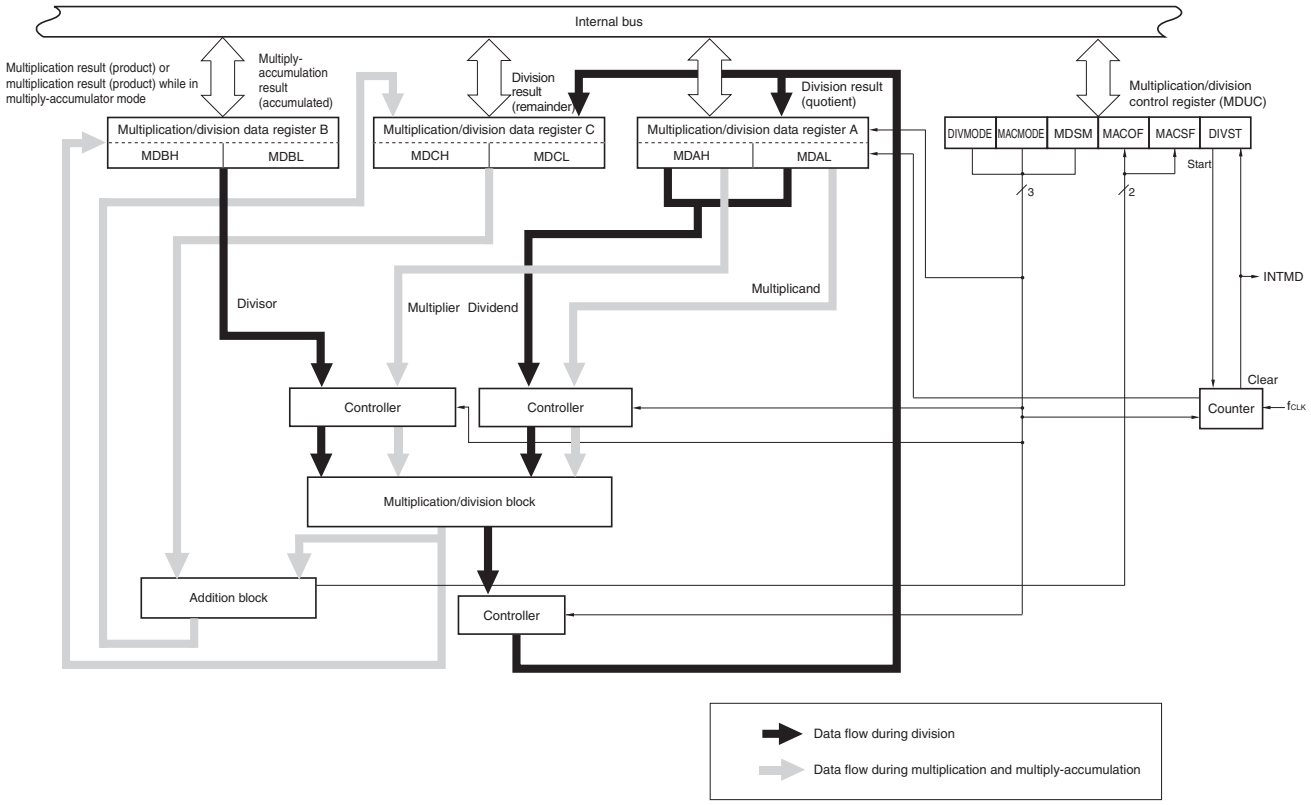
The multiplier and divider/multiply-accumulator consists of the following hardware.

Table 15-1. Configuration of Multiplier and Divider/Multiply-Accumulator

Item	Configuration
Registers	Multiplication/division data register A (L) (MDAL) Multiplication/division data register A (H) (MDAH) Multiplication/division data register B (L) (MDBL) Multiplication/division data register B (H) (MDBH) Multiplication/division data register C (L) (MDCL) Multiplication/division data register C (H) (MDCH)
Control register	Multiplication/division control register (MDUC)

Figure 15-1 shows a block diagram of the multiplier and divider/multiply-accumulator.

Figure 15-1. Block Diagram of Multiplier and Divider/Multiply-Accumulator



Remark f_{CLK}: CPU/peripheral hardware clock frequency

15.2.1 Multiplication/division data register A (MDAH, MDAL)

The MDAH and MDAL registers set the values that are used for a multiplication or division operation and store the operation result. They set the multiplier and multiplicand data in the multiplication mode or multiply-accumulator mode, and set the dividend data in the division mode. Furthermore, the operation result (quotient) is stored in the MDAH and MDAL registers in the division mode.

The MDAH and MDAL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 15-2. Format of Multiplication/Division Data Register A (MDAH, MDAL)

Address: FFFF0H, FFFF1H, FFFF2H, FFFF3H After reset: 0000H, 0000H R/W

Symbol	FFFF3H								FFFF2H							
MDAH	MDAH 15	MDAH 14	MDAH 13	MDAH 12	MDAH 11	MDAH 10	MDAH 9	MDAH 8	MDAH 7	MDAH 6	MDAH 5	MDAH 4	MDAH 3	MDAH 2	MDAH 1	MDAH 0

Symbol	FFFF1H								FFFF0H							
MDAL	MDAL 15	MDAL 14	MDAL 13	MDAL 12	MDAL 11	MDAL 10	MDAL 9	MDAL 8	MDAL 7	MDAL 6	MDAL 5	MDAL 4	MDAL 3	MDAL 2	MDAL 1	MDAL 0

Cautions 1. Do not rewrite the MDAH and MDAL registers values during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H). The operation will be executed in this case, but the operation result will be an undefined value.

2. The MDAH and MDAL registers values read during division operation processing (when the MDUC register value is 81H or C1H) will not be guaranteed.

3. The data is in the two's complement format in either the multiplication mode (signed) or multiply-accumulator mode (signed).

The following table shows the functions of the MDAH and MDAL registers during operation execution.

Table 15-2. Functions of MDAH and MDAL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned) Multiply-accumulator mode (unsigned)	MDAH: Multiplier (unsigned) MDAL: Multiplicand (unsigned)	–
Multiplication mode (signed) Multiply-accumulator mode (signed)	MDAH: Multiplier (signed) MDAL: Multiplicand (signed)	–
Division mode (unsigned)	MDAH: Dividend (unsigned) (higher 16 bits) MDAL: Dividend (unsigned) (lower 16 bits)	MDAH: Division result (unsigned) Higher 16 bits MDAL: Division result (unsigned) Lower 16 bits

15.2.2 Multiplication/division data register B (MDBL, MDBH)

The MDBH and MDBL registers set the values that are used for multiplication or division operation and store the operation result. They store the operation result (product) in the multiplication mode and multiply-accumulator mode, and set the divisor data in the division mode.

The MDBH and MDBL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 15-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)

Address: FFFF4H, FFFF5H, FFFF6H, FFFF7H After reset: 0000H, 0000H R/W

Symbol	FFFF5H								FFFF4H							
MDBH	MDBH 15	MDBH 14	MDBH 13	MDBH 12	MDBH 11	MDBH 10	MDBH 9	MDBH 8	MDBH 7	MDBH 6	MDBH 5	MDBH 4	MDBH 3	MDBH 2	MDBH 1	MDBH 0

Symbol	FFFF7H								FFFF6H							
MDBL	MDBL 15	MDBL 14	MDBL 13	MDBL 12	MDBL 11	MDBL 10	MDBL 9	MDBL 8	MDBL 7	MDBL 6	MDBL 5	MDBL 4	MDBHL 3	MDBL 2	MDBL 1	MDBL 0

Cautions 1. Do not rewrite the MDBH and MDBL registers values during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H) or multiply-accumulation operation processing. The operation result will be an undefined value.

2. Do not set the MDBH and MDBL registers to 0000H in the division mode. If they are set, the operation result will be an undefined value.

3. The data is in the two's complement format in either the multiplication mode (signed) or multiply-accumulator mode (signed).

The following table shows the functions of the MDBH and MDBL registers during operation execution.

Table 15-3. Functions of MDBH and MDBL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned) Multiply-accumulator mode (unsigned)	–	MDBH: Multiplication result (product) (unsigned) Higher 16 bits MDBL: Multiplication result (product) (unsigned) Lower 16 bits
Multiplication mode (signed) Multiply-accumulator mode (signed)	–	MDBH: Multiplication result (product) (signed) Higher 16 bits MDBL: Multiplication result (product) (signed) Lower 16 bits
Division mode (unsigned)	MDAH: Dividend (unsigned) (higher 16 bits) MDAL: Dividend (unsigned) (lower 16 bits)	–

15.2.3 Multiplication/division data register C (MDCL, MDCH)

The MDCH and MDCL registers are used to store the accumulated result while in the multiply-accumulator mode or the remainder of the operation result while in the division mode. These registers are not used while in the multiplication mode.

The MDCH and MDCL registers can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 15-4. Format of Multiplication/Division Data Register C (MDCH, MDCL)

Address: F00E0H, F00E1H, F00E2H, F00E3H After reset: 0000H, 0000H R/W

Symbol	F00E3H								F00E2H							
MDCH	MDCH 15	MDCH 14	MDCH 13	MDCH 12	MDCH 11	MDCH 10	MDCH 9	MDCH 8	MDCH 7	MDCH 6	MDCH 5	MDCH 4	MDCH 3	MDCH 2	MDCH 1	MDCH 0

Symbol	F00E1H								F00E0H							
MDCL	MDCL 15	MDCL 14	MDCL 13	MDCL 12	MDCL 11	MDCL 10	MDCL 9	MDCL 8	MDCL 7	MDCL 6	MDCL 5	MDCL 4	MDCL 3	MDCL 2	MDCL 1	MDCL 0

Cautions 1. The MDCH and MDCL registers values read during division operation processing (when the multiplication/division control register (MDUC) value is 81H or C1H) will not be guaranteed.

2. During multiply-accumulator processing, do not use software to rewrite the values of the MDCH and MDCL registers. If this is done, the operation result will be undefined.

3. The data is in the two's complement format in the multiply-accumulator mode (signed).

Table 15-4. Functions of MDCH and MDCL Registers During Operation Execution

Operation Mode	Setting	Operation Result
Multiplication mode (unsigned or signed)	–	–
Multiply-accumulator mode (unsigned)	MDCH: Initial accumulated value (unsigned) (higher 16 bits) MDCL: Initial accumulated value (unsigned) (lower 16 bits)	MDCH: accumulated value (unsigned) (higher 16 bits) MDCL: accumulated value (unsigned) (lower 16 bits)
Multiply-accumulator mode (signed)	MDCH: Initial accumulated value (signed) (higher 16 bits) MDCL: Initial accumulated value (signed) (lower 16 bits)	MDCH: accumulated value (signed) (higher 16 bits) MDCL: accumulated value (signed) (lower 16 bits)
Division mode (unsigned)	–	MDCH: Remainder (unsigned) (higher 16 bits) MDCL: Remainder (unsigned) (lower 16 bits)

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

  <Multiplier A> <Multiplier B> <Product>

  MDAL (bits 15 to 0) × MDAH (bits 15 to 0) = [MDBH (bits 15 to 0), MDBL (bits 15 to 0)]

- Register configuration during multiply-accumulation

  <Multiplier A> <Multiplier B> < accumulated value > < accumulated result >

  MDAL (bits 15 to 0) × MDAH (bits 15 to 0) + MDC (bits 31 to 0) = [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

  (The multiplication result is stored in the MDBH (bits 15 to 0) and MDBL (bits 15 to 0).)

- Register configuration during division

  <Dividend> <Divisor>

  [MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ÷ [MDBH (bits 15 to 0), MDBL (bits 15 to 0)] =

  <Quotient> <Remainder>

  [MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ··· [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

15.3 Register Controlling Multiplier and Divider/Multiply-Accumulator

The multiplier and divider/multiply-accumulator is controlled by using the multiplication/division control register (MDUC).

15.3.1 Multiplication/division control register (MDUC)

The MDUC register is an 8-bit register that controls the operation of the multiplier and divider/multiply-accumulator.

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

Note that the overflow flag (MACOF) and sign flag (MACSF) of the multiply-accumulation result (accumulated) are read-only flags.

Reset signal generation clears this register to 00H.

Figure 15-5. Format of Multiplication/Division Control Register (MDUC)

Address: F00E8H After reset: 00H R/W [Note 1]

Symbol	<7>	<6>	5	4	<3>	<2>	<1>	<0>
MDUC	DIVMODE	MACMODE	0	0	MDSM	MACOF	MACSF	DIVST

DIVMODE	MACMODE	MDSM	Operation mode selection
0	0	0	Multiplication mode (unsigned) (default)
0	0	1	Multiplication mode (signed)
0	1	0	Multiply-accumulator mode (unsigned)
0	1	1	Multiply-accumulator mode (signed)
1	0	0	Division mode (unsigned), generation of a division completion interrupt (INTMD)
1	1	0	Division mode (unsigned), not generation of a division completion interrupt (INTMD)
Other than above			Setting prohibited

MACOF	Overflow flag of multiply-accumulation result (accumulated value)
0	No overflow
1	With over flow

<Set condition>

- For the multiply-accumulator mode (unsigned)

  The bit is set when the accumulated value goes outside the range from 00000000h to FFFFFFFFh.

- For the multiply-accumulator mode (signed)

  The bit is set when the result of adding a positive product to a positive accumulated value exceeds 7FFFFFFFh and is negative, or when the result of adding a negative product to a negative accumulated value exceeds 80000000h and is positive.

MACSF	Sign flag of multiply-accumulation result (accumulated value)
0	The accumulated value is positive.
1	The accumulated value is negative.

Multiply-accumulator mode (unsigned): The bit is always 0.

Multiply-accumulator mode (signed): The bit indicates the sign bit of the accumulated value.

DIVST[Note 2]	Division operation start/stop
0	Division operation processing complete
1	Starts division operation/division operation processing in progress

Notes 1. Bits 1 and 2 are read-only bits.

2. The DIVST bit can only be set (1) in the division mode. In the division mode, division operation is started by setting (1) the DIVST bit. The DIVST bit is automatically cleared (0) when the operation ends. In the multiplication mode, operation is automatically started by setting the multiplier and multiplicand to multiplication/division data register A (MDAH, MDAL), respectively.

Cautions 1. Do not rewrite the DIVMODE, MDSM bits during operation processing (while the DIVST bit is 1). If it is rewritten, the operation result will be an undefined value.

2. The DIVST bit cannot be cleared (0) by using software during division operation processing (while the DIVST bit is 1).

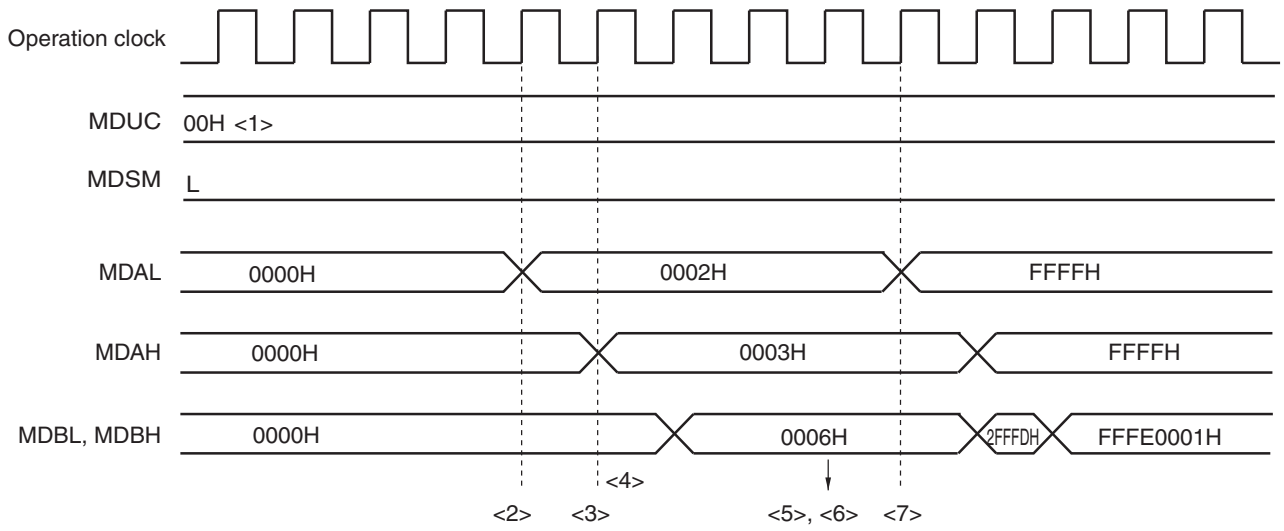
15.4 Operations of Multiplier and Divider/Multiply-Accumulator

15.4.1 Multiplication (unsigned) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 00H.
  - <2> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation
  - <7> Start with the initial settings of each step to change the operation mode. When the same operation mode is used sequentially, settings <1> and <2> can be omitted.

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 15-6.

Figure 15-6. Timing Diagram of Multiplication (Unsigned) Operation (2 × 3 = 6)



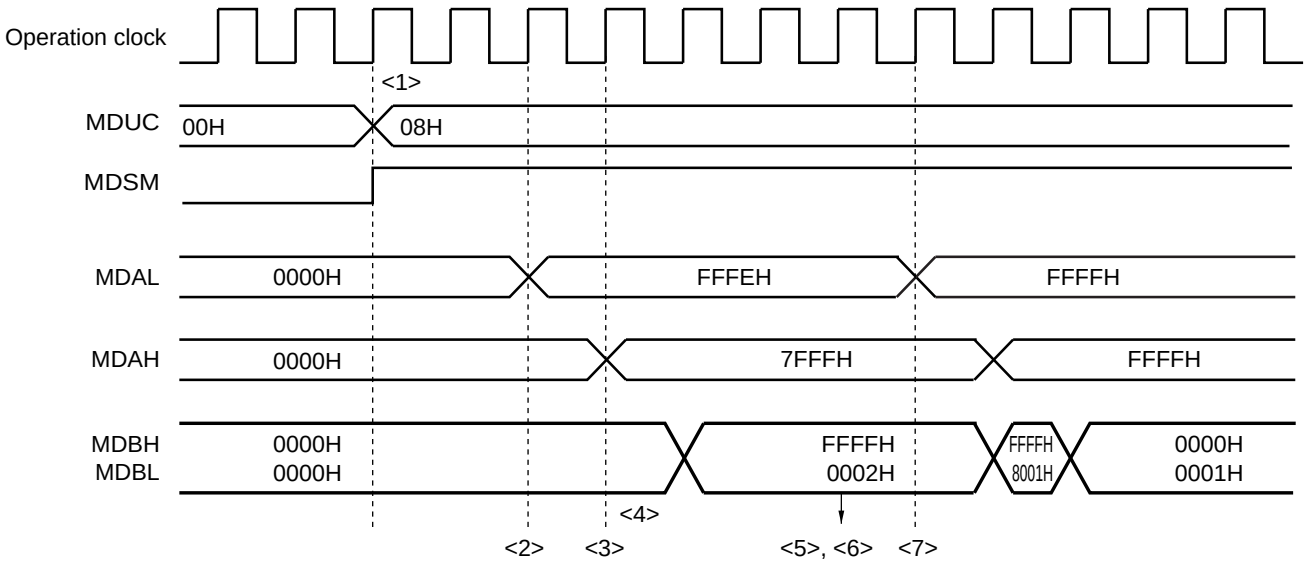
15.4.2 Multiplication (signed) operation

- Initial setting
  - <1> Set the multiplication/division control register (MDUC) to 08H.
  - <2> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to the MDAH and MDAL registers, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation
  - <7> Start with the initial settings of each step to change the operation mode. When the same operation mode is used sequentially, settings <1> and <2> can be omitted.

Caution The data is in the two's complement format in multiplication mode (signed).

Remark Steps <1> to <7> correspond to <1> to <7> in Figure 15-7.

Figure 15-7. Timing Diagram of Multiplication (Signed) Operation (–2 × 32767 = –65534)

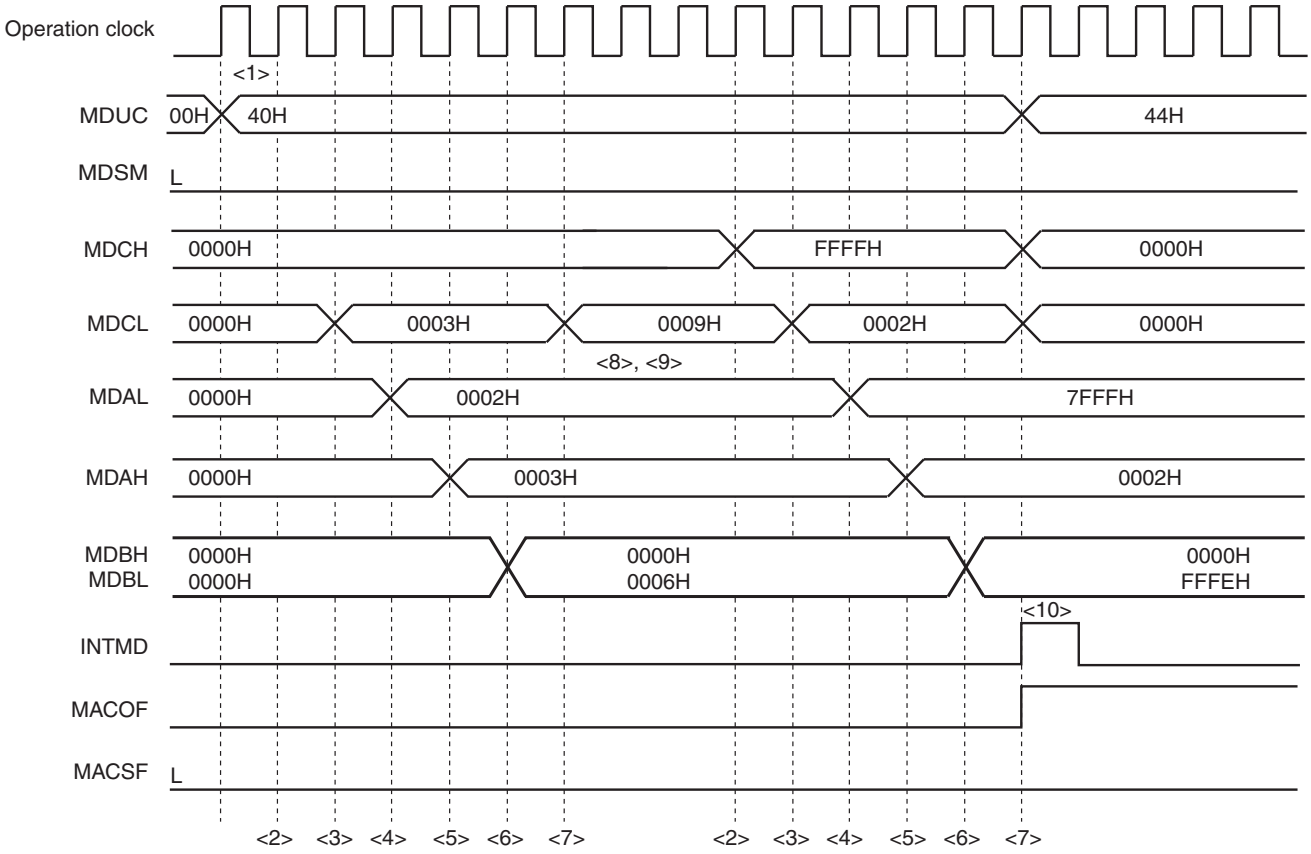


15.4.3 Multiply-accumulation (unsigned) operation

- Initial setting
  <1> Set the multiplication/division control register (MDUC) to 40H.
  <2> Set the initial accumulated value of higher 16 bits to multiplication/division data register C (L) (MDCL).
  <3> Set the initial accumulated value of lower 16 bits to multiplication/division data register C (H) (MDCH).
  <4> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  <5> Set the multiplier to multiplication/division data register A (H) (MDAH).
  (There is no preference in the order of executing steps <2>, <3>, and <4>. Multiplication operation is automatically started when the multiplier is set to the MDAH register of <5>, respectively.)
- During operation processing
  <6> The multiplication operation finishes in one clock cycle.
  (The multiplication result is stored in multiplication/division data register B (L) (MDBL) and multiplication/division data register B (H) (MDBH).)
  <7> After <6>, the multiply-accumulation operation finishes in one additional clock cycle. (There is a wait of at least two clock cycles after specifying the initial settings is finished (<5>).)
- Operation end
  <8> Read the accumulated value (lower 16 bits) from the MDCL register.
  <9> Read the accumulated value (higher 16 bits) from the MDCH register.
  (There is no preference in the order of executing steps <8> and <9>.)
  (<10> If the result of the multiply-accumulation operation causes an overflow, the MACOF bit is set to 1, INTMD signal is occurred.)
- Next operation
  <11> Start with the initial settings of each step to change the operation mode. When the same operation mode is used sequentially, settings <1> to <4> can be omitted.

Remark Steps <1> to <10> correspond to <1> to <10> in Figure 15-8.

Figure 15-8. Timing Diagram of Multiply-Accumulation (Unsigned) Operation
(2 × 3 + 3 = 9 → 32767 × 2 + 4294901762 = 0 (over flow generated))



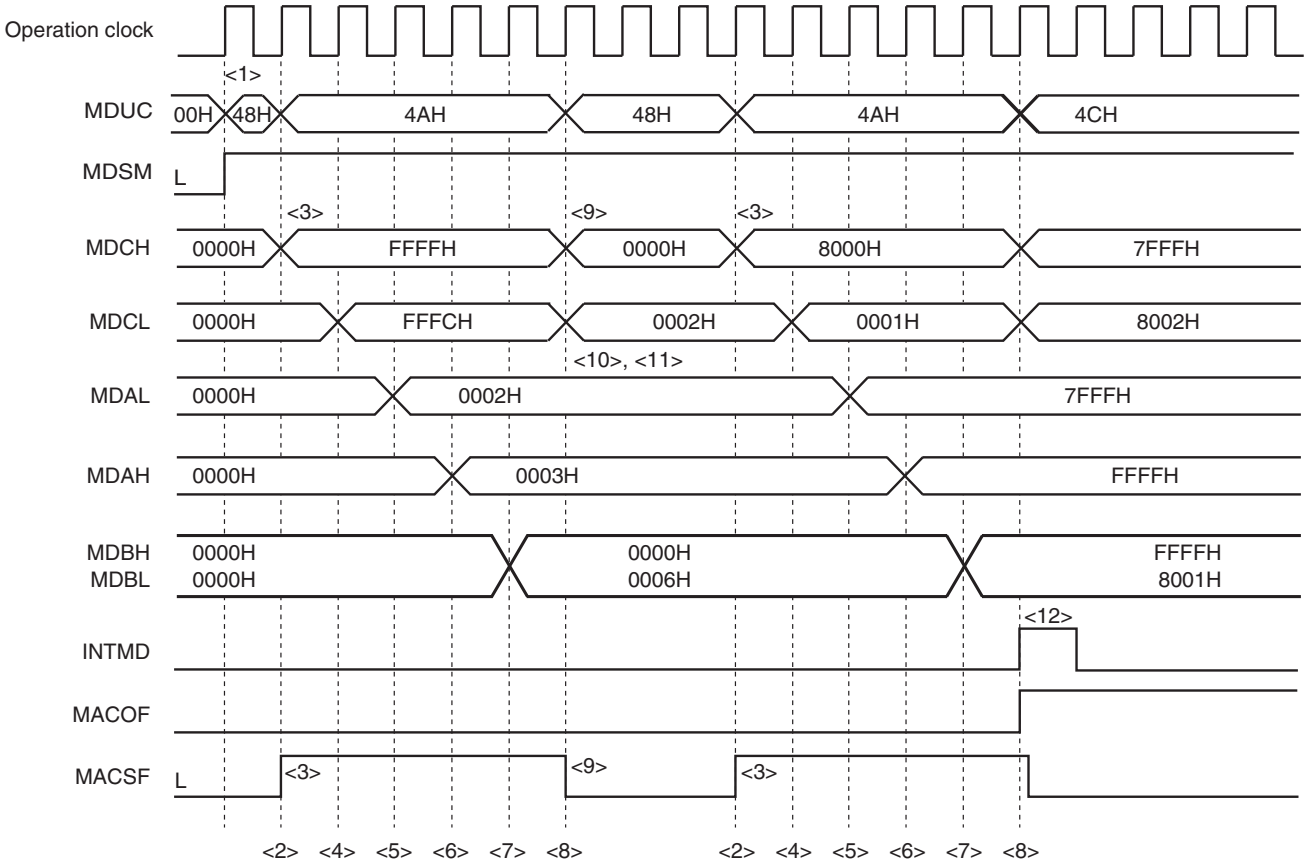
15.4.4 Multiply-accumulation (signed) operation

- Initial setting
  <1> Set the multiplication/division control register (MDUC) to 48H.
  <2> Set the initial accumulated value of higher 16 bits to multiplication/division data register C (H) (MDCH).
  (<3> If the accumulated value in the MDCH register is negative, the MACSF bit is set to 1.)
  <4> Set the initial accumulated value of lower 16 bits to multiplication/division data register C (L) (MDCL).
  <5> Set the multiplicand to multiplication/division data register A (L) (MDAL).
  <6> Set the multiplier to multiplication/division data register A (H) (MDAH).
  (There is no preference in the order of executing steps <2>, <4>, and <5>. Multiplication operation is automatically started when the multiplier is set to the MDAH register of <6>, respectively.)
- During operation processing
  <7> The multiplication operation finishes in one clock cycle.
  (The multiplication result is stored in multiplication/division data register B (L) (MDBL) and multiplication/division data register B (H) (MDBH).)
  <8> After <7>, the multiply-accumulation operation finishes in one additional clock cycle. (There is a wait of at least two clock cycles after specifying the initial settings is finished (<6>).)
- Operation end
  <9> If the accumulated value stored in the MDCL and MDCH registers is positive, the MACSF bit is cleared to 0.
  <10> Read the accumulated value (lower 16 bits) from the MDCL register.
  <11> Read the accumulated value (higher 16 bits) from the MDCH register.
  (There is no preference in the order of executing steps <10> and <11>.)
  (<12> If the result of the multiply-accumulation operation causes an overflow, the MACOF bit is set to 1, INTMD signal is occurred.)
- Next operation
  <13> Start with the initial settings of each step to change the operation mode. When the same operation mode is used sequentially, settings <1> to <5> can be omitted.

Caution The data is in the two's complement format in multiply-accumulation (signed) operation.

Remark Steps <1> to <12> correspond to <1> to <12> in Figure 15-9.

Figure 15-9. Timing Diagram of Multiply-Accumulation (signed) Operation
(2 × 3 + (–4) = 2 → 32767 × (–1) + (–2147483647) = 2147450882 (overflow occurs.))



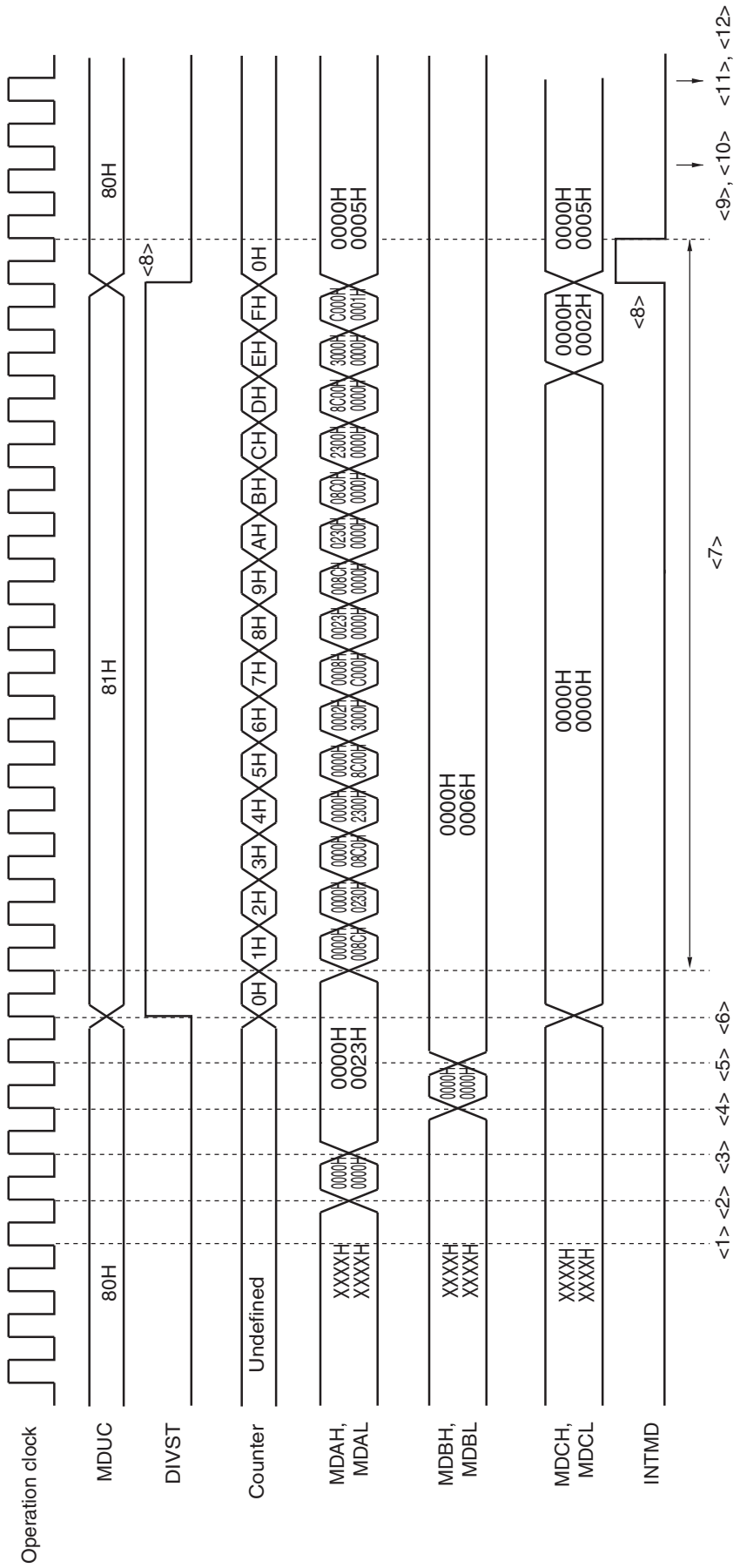
15.4.5 Division operation

- Initial setting
  <1> Set the multiplication/division control register (MDUC) to 80H.
  <2> Set the dividend (higher 16 bits) to multiplication/division data register A (H) (MDAH).
  <3> Set the dividend (lower 16 bits) to multiplication/division data register A (L) (MDAL).
  <4> Set the divisor (higher 16 bits) to multiplication/division data register B (H) (MDBH).
  <5> Set the divisor (lower 16 bits) to multiplication/division data register B (L) (MDBL).
  <6> Set bit 0 (DIVST) of the MDUC register to 1.
  (There is no preference in the order of executing steps <2> to <5>.)
- During operation processing
  <7> The operation will end when one of the following processing is completed.
  - A wait of at least 16 clocks (The operation will end when 16 clocks have been issued.)
  - A check whether the DIVST bit has been cleared

  (The read values of the MDBL, MDBH, MDCL, and MDCH registers during operation processing are not guaranteed.)
- Operation end
  <8> The DIVST bit is cleared and the operation ends. At this time, an interrupt request signal (INTMD) is generated if the operation was performed with MACMODE = 0.
  <9> Read the quotient (lower 16 bits) from the MDAL register.
  <10> Read the quotient (higher 16 bits) from the MDAH register.
  <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  <12> Read the remainder (higher 16 bits) from multiplication/division data register C (H) (MDCH).
  (There is no preference in the order of executing steps <9> to <12>.)
- Next operation
  <13> Start with the initial settings of each step to change the operation mode. When the same operation mode is used sequentially, settings <1> to <5> can be omitted.

Remark Steps <1> to <12> correspond to <1> to <12> in Figure 15-10.

Figure 15-10. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



CHAPTER 16 DMA CONTROLLER

The RL78/G1C has an internal DMA (Direct Memory Access) controller.

Data can be automatically transferred between the peripheral hardware supporting DMA, SFRs, and internal RAM without via CPU.

As a result, the normal internal operation of the CPU and data transfer can be executed in parallel with transfer between the SFR and internal RAM, and therefore, a large capacity of data can be processed. In addition, real-time control using communication, timer, A/D, and USB function controller can also be realized.

16.1 Functions of DMA Controller

- ○ Number of DMA channels: 2 channels
- ○ Transfer unit: 8 or 16 bits
- ○ Number of continuous transfer settings: 1 to 1024 times
- ○ Transfer type: 2-cycle transfer (One transfer is processed in 2 clocks and the CPU stops during that processing.)
- ○ Transfer mode: Single-transfer mode
- ○ Transfer request: Selectable from the following peripheral hardware interrupts
  - A/D converter
  - Serial interface
    (CSI00, CSI01, UART0)
  - Timer (channel 0, 1, 2, 3)
  - USB FIFO port request signal
- ○ Transfer target: Between SFR and internal RAM

Here are examples of functions using DMA.

- Successive transfer of serial interface
- Consecutive capturing of A/D conversion results
- Capturing port value at fixed interval

16.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 16-1. Configuration of DMA Controller

Item	Configuration
Address registers	• DMA SFR address registers 0, 1 (DSA0, DSA1) • DMA RAM address registers 0, 1 (DRA0, DRA1)
Count register	• DMA byte count registers 0, 1 (DBC0, DBC1)
Control registers	• DMA mode control registers 0, 1 (DMC0, DMC1) • DMA operation control register 0, 1 (DRC0, DRC1)

16.2.1 DMA SFR address register n (DSAn)

This is an 8-bit register that is used to set an SFR address that is the transfer source or destination of DMA channel n.

Set the lower 8 bits of the SFR addresses FFF00H to FFFFFH.

This register is not automatically incremented but fixed to a specific value.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

The DSAn register can be read or written in 8-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 00H.

Figure 16-1. Format of DMA SFR Address Register n (DSAn)

Address: FFFB0H (DSA0), FFFB1H (DSA1) After reset: 00H R/W

	7	6	5	4	3	2	1	0
DSAn								

Remark n: DMA channel number (n = 0, 1)

16.2.2 DMA RAM address register n (DRAn)

This is a 16-bit register that is used to set a RAM address that is the transfer source or destination of DMA channel n.

Addresses of the internal RAM area other than the general-purpose registers (FE900H to FFEDFH) can be set to this register.

Set the lower 16 bits of the RAM address.

This register is automatically incremented when DMA transfer has been started. It is incremented by +1 in the 8-bit transfer mode and by +2 in the 16-bit transfer mode. DMA transfer is started from the address set to this DRAn register. When the data of the last address has been transferred, the DRAn register stops with the value of the last address +1 in the 8-bit transfer mode, and the last address +2 in the 16-bit transfer mode.

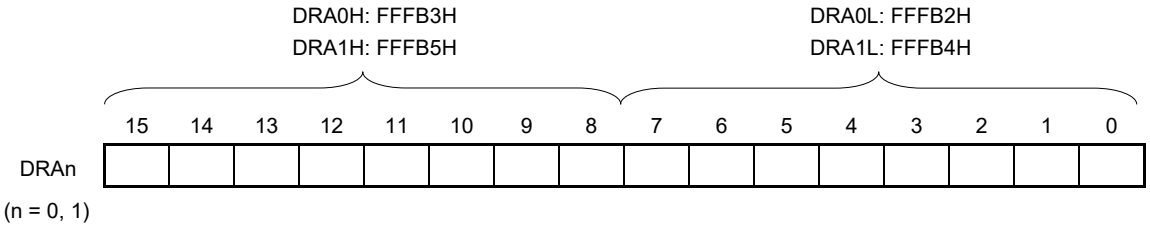
In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

The DRAn register can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 16-2. Format of DMA RAM Address Register n (DRAn)

Address: FFFB2H, FFFB3H (DRA0), FFFB4H, FFFB5H (DRA1), After reset: 0000H R/W



Remark n: DMA channel number (n = 0, 1)

16.2.3 DMA byte count register n (DBCn)

This is a 10-bit register that is used to set the number of times DMA channel n executes transfer. Be sure to set the number of times of transfer to this DBCn register before executing DMA transfer (up to 1024 times).

Each time DMA transfer has been executed, this register is automatically decremented. By reading this DBCn register during DMA transfer, the remaining number of times of transfer can be learned.

The DBCn register can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 16-3. Format of DMA Byte Count Register n (DBCn)

Address: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) After reset: 0000H R/W

DBC0H: FFFB7H
DBC1H: FFFB9H

DBC0L: FFFB6H
DBC1L: FFFB8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBCn (n = 0, 1)	0	0	0	0	0	0										

DBCn[9:0]	Number of Times of Transfer (When DBCn is Written)	Remaining Number of Times of Transfer (When DBCn is Read)
000H	1024	Completion of transfer or waiting for 1024 times of DMA transfer
001H	1	Waiting for remaining one time of DMA transfer
002H	2	Waiting for remaining two times of DMA transfer
003H	3	Waiting for remaining three times of DMA transfer
• • •	• • •	• • •
3FEH	1022	Waiting for remaining 1022 times of DMA transfer
3FFH	1023	Waiting for remaining 1023 times of DMA transfer

Cautions 1. Be sure to clear bits 15 to 10 to "0".

2. If the general-purpose register is specified or the internal RAM space is exceeded as a result of continuous transfer, the general-purpose register or SFR space are written or read, resulting in loss of data in these spaces. Be sure to set the number of times of transfer that is within the internal RAM space.

Remark n: DMA channel number (n = 0, 1)

16.3 Registers Controlling DMA Controller

DMA controller is controlled by the following registers.

- DMA mode control register n (DMCn)
- DMA operation control register n (DRCn)

Remark n: DMA channel number (n = 0, 1)

16.3.1 DMA mode control register n (DMCn)

The DMCn register is a register that is used to set a transfer mode of DMA channel n. It is used to select a transfer direction, data size, setting of pending, and start source. Bit 7 (STGn) is a software trigger that starts DMA.

Rewriting bits 6, 5, and 3 to 0 of the DMCn register is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 16-4. Format of DMA Mode Control Register n (DMCn) (1/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn[Note 1]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is performed once by writing 1 to the STGn bit when DMA operation is enabled (DENn = 1). When this bit is read, 0 is always read.	

DRSn	Selection of DMA transfer direction
0	SFR to internal RAM
1	Internal RAM to SFR

DSn	Specification of transfer data size for DMA transfer
0	8 bits
1	16 bits

DWAITn [Note 2]	Pending of DMA transfer
0	Executes DMA transfer upon DMA start request (not held pending).
1	Holds DMA start request pending if any.
DMA transfer that has been held pending can be started by clearing the value of the DWAITn bit to 0. It takes 2 clocks to actually hold DMA transfer pending when the value of the DWAITn bit is set to 1.	

Notes 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

2. When DMA transfer is held pending while using two or more DMA channels, be sure to hold the DMA transfer pending for both channels (by setting the DWAIT0 and DWAIT1 bits to 1).

Remark n: DMA channel number (n = 0, 1)

Figure 16-4. Format of DMA Mode Control Register n (DMCn) (2/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

(When n = 0 or 1)

IFCn 3	IFCn 2	IFCn 1	IFCn 0	Selection of DMA start source[Note]	
				Trigger signal	Trigger contents
0	0	0	0	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)
0	0	0	1	INTAD	A/D conversion end interrupt
0	0	1	0	INTTM00	End of timer channel 0 count or capture end interrupt
0	0	1	1	INTTM01	End of timer channel 1 count or capture end interrupt
0	1	0	0	INTTM02	End of timer channel 2 count or capture end interrupt
0	1	0	1	INTTM03	End of timer channel 3 count or capture end interrupt
0	1	1	0	INTST0/INTCSI00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt
0	1	1	1	INTSR0/INTCSI01	UART0 reception transfer end interrupt/CSI01 transfer end or buffer empty interrupt
1	1	0	0	DMA0FIFO	DMA0-FIFO port request signal
1	1	0	1	DMA1FIFO	DMA1-FIFO port request signal
Other than above				Setting prohibited	

Note The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

Remark n: DMA channel number (n = 0, 1)

16.3.2 DMA operation control register n (DRCn)

The DRCn register is a register that is used to enable or disable transfer of DMA channel n.

Rewriting bit 7 (DENn) of this register is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 16-5. Format of DMA Operation Control Register n (DRCn)

Address: FFFBCH (DRC0), FFFBDH (DRC1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA operation enable flag
0	Disables operation of DMA channel n (stops operating cock of DMA).
1	Enables operation of DMA channel n.
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1).	

DSTn	DMA transfer mode flag
0	DMA transfer of DMA channel n is completed.
1	DMA transfer of DMA channel n is not completed (still under execution).
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1). When a software trigger (STGn) or the start source trigger set by the IFCn3 to IFCn0 bits is input, DMA transfer is started. When DMA transfer is completed after that, this bit is automatically cleared to 0. Write 0 to this bit to forcibly terminate DMA transfer under execution.	

Caution The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, therefore, set the DSTn bit to 0 and then the DENn bit to 0 (for details, refer to 16.5.5 Forced termination by software).

Remark n: DMA channel number (n = 0, 1)

16.4 Operation of DMA Controller

16.4.1 Operation procedure

<1> The DMA controller is enabled to operate when DENn = 1. Before writing the other registers, be sure to set the DENn bit to 1. Use 80H to write with an 8-bit manipulation instruction.

<2> Set an SFR address, a RAM address, the number of times of transfer, and a transfer mode of DMA transfer to DMA SFR address register n (DSAn), DMA RAM address register n (DRAn), DMA byte count register n (DBCn), and DMA mode control register n (DMCn).

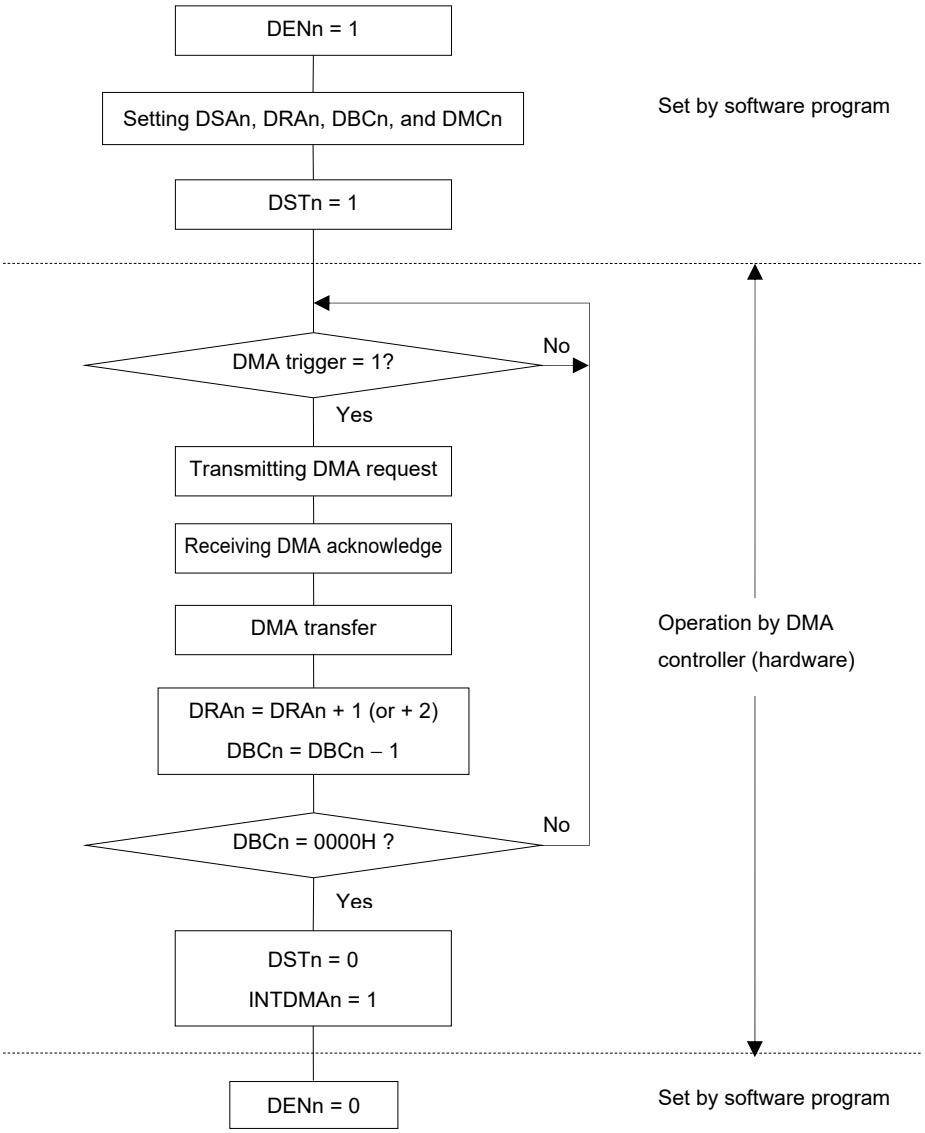
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.

<4> When a software trigger (STGn) or a start source trigger specified by the IFCn3 to IFCn0 bits is input, a DMA transfer is started.

<5> Transfer is completed when the number of times of transfer set by the DBCn register reaches 0, and transfer is automatically terminated by occurrence of an interrupt (INTDMAn).

<6> Stop the operation of the DMA controller by clearing the DENn bit to 0 when the DMA controller is not used.

Figure 16-6. Operation Procedure



Remark n: DMA channel number (n = 0, 1)

16.4.2 Transfer mode

The following four modes can be selected for DMA transfer by using bits 6 and 5 (DRSn and DSn) of DMA mode control register n (DMCn).

DRSn	DSn	DMA Transfer Mode
0	0	Transfer from SFR of 1-byte data (fixed address) to RAM (address is incremented by +1)
0	1	Transfer from SFR of 2-byte data (fixed address) to RAM (address is incremented by +2)
1	0	Transfer from RAM of 1-byte data (address is incremented by +1) to SFR (fixed address)
1	1	Transfer from RAM of 2-byte data (address is incremented by +2) to SFR (fixed address)

By using these transfer modes, up to 1024 bytes of data can be consecutively transferred by using the serial interface, data resulting from A/D conversion can be consecutively transferred, and port data can be scanned at fixed time intervals by using a timer.

16.4.3 Termination of DMA transfer

When DBCn = 00H and DMA transfer is completed, the DSTn bit is automatically cleared to 0. An interrupt request (INTDMAn) is generated and transfer is terminated.

When the DSTn bit is cleared to 0 to forcibly terminate DMA transfer, DMA byte count register n (DBCn) and DMA RAM address register n (DRAn) hold the value when transfer is terminated.

The interrupt request (INTDMAn) is not generated if transfer is forcibly terminated.

Remark n: DMA channel number (n = 0, 1)

16.5 Example of Setting of DMA Controller

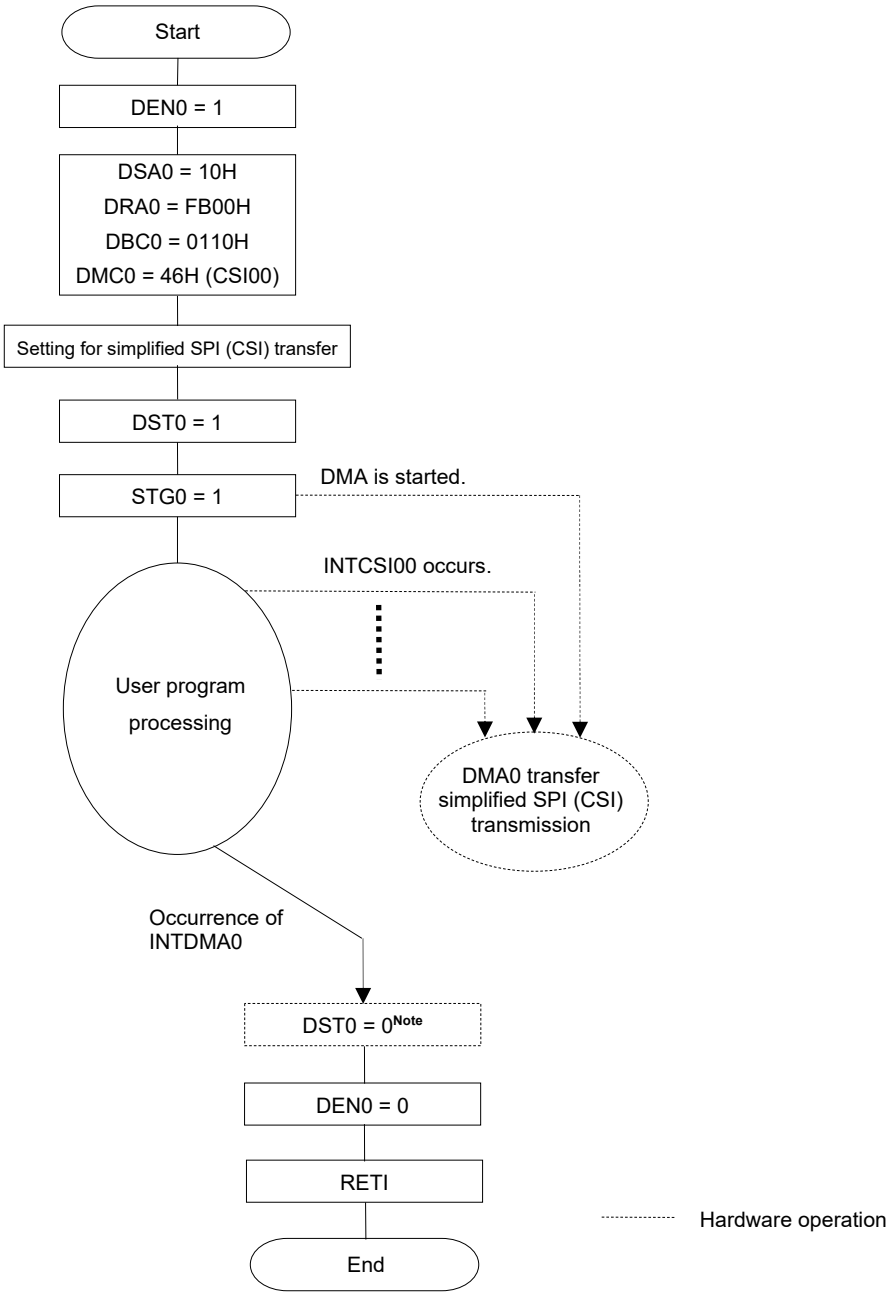
16.5.1 Simplified SPI (CSI) consecutive transmission

A flowchart showing an example of setting for simplified SPI (CSI) consecutive transmission is shown below.

- Consecutive transmission of CSI00 (256 bytes)
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI00 (software trigger (STG0) only for the first start source)
- Interrupt of CSI00 is specified by IFC03 to IFC00 = 0110B.
- Transfers FFB00H to FFBFFH (256 bytes) of RAM to FFF10H of the data register (SIO00) of simplified SPI (CSI).

Remark IFC03 to IFC00: Bits 3 to 0 of DMA mode control registers 0 (DMC0)

Figure 16-7. Example of Setting for Simplified SPI (CSI) Consecutive Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set the DST0 bit to 0 and then the DEN0 bit to 0 (for details, refer to **16.5.5 Forced termination by software**).

The fist trigger for consecutive transmission is not started by the interrupt of simplified SPI (CSI). In this example, it start by a software trigger.

Simplified SPI (CSI) transmission of the second time and onward is automatically executed.

A DMA interrupt (INTDMA0) occurs when the last transmit data has been written to the data register.

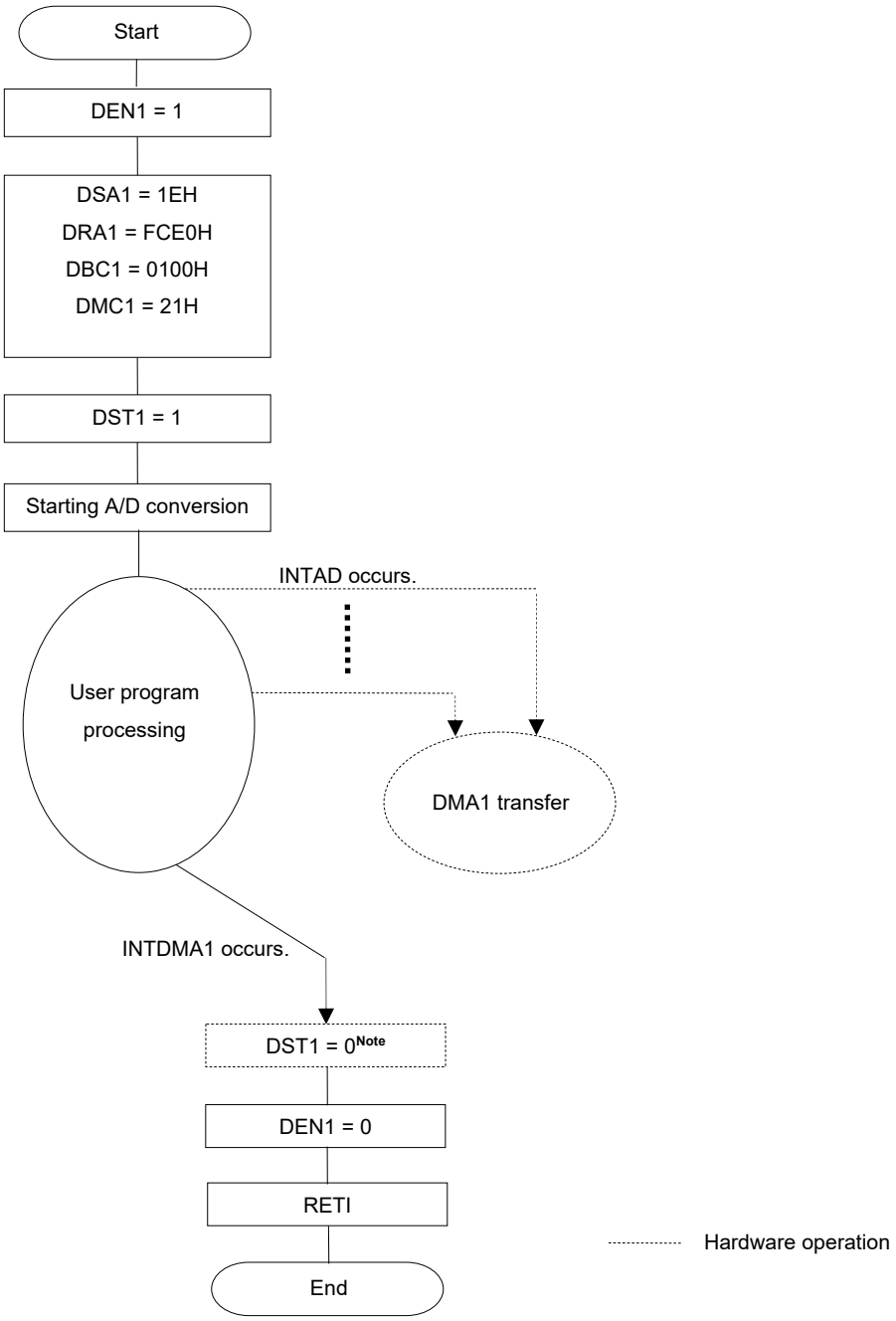
16.5.2 Consecutive capturing of A/D conversion results

A flowchart of an example of setting for consecutively capturing A/D conversion results is shown below.

- Consecutive capturing of A/D conversion results.
- DMA channel 1 is used for DMA transfer.
- DMA start source: INTAD
- Interrupt of A/D is specified by IFC13 to IFC10 = 0001B.
- Transfers FFF1EH and FFF1FH (2 bytes) of the 10-bit A/D conversion result register (ADCR) to 512 bytes of FFCE0H to FFEDFH of RAM.

Remark IFC13 to IFC10: Bits 3 to 0 of DMA mode control registers 1 (DMC1)

Figure 16-8. Example of Setting of Consecutively Capturing A/D Conversion Results



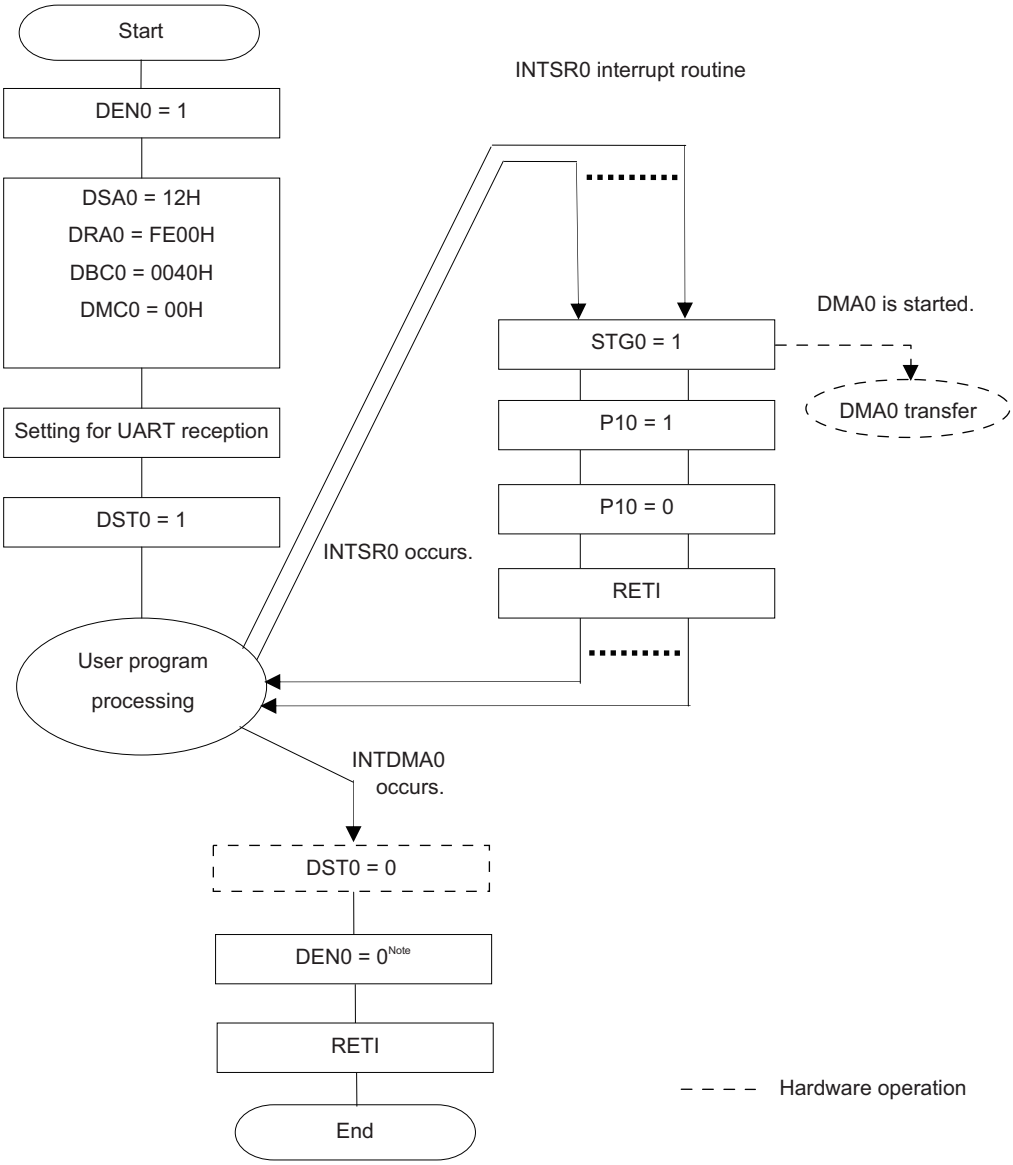
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set the DST1 bit to 0 and then the DEN1 bit to 0 (for details, refer to **16.5.5 Forced termination by software**).

16.5.3 UART consecutive reception + ACK transmission

A flowchart illustrating an example of setting for UART consecutive reception + ACK transmission is shown below.

- Consecutively receives data from UART0 and outputs ACK to P30 on completion of reception.
- DMA channel 0 is used for DMA transfer.
- DMA start source: Software trigger (DMA transfer on occurrence of an interrupt is disabled.)
- Transfers FFF12H of UART receive data register 0 (RXD0) to 64 bytes of FFE00H to FFE3FH of RAM.

Figure 16-9. Example of Setting for UART Consecutive Reception + ACK Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set the DST0 bit to 0 and then the DEN0 bit to 0 (for details, refer to **16.5.5 Forced termination by software**).

Remark This is an example where a software trigger is used as a DMA start source.
If ACK is not transmitted and if only data is consecutively received from UART, the UART reception end interrupt (INTSR0) can be used to start DMA for data reception.

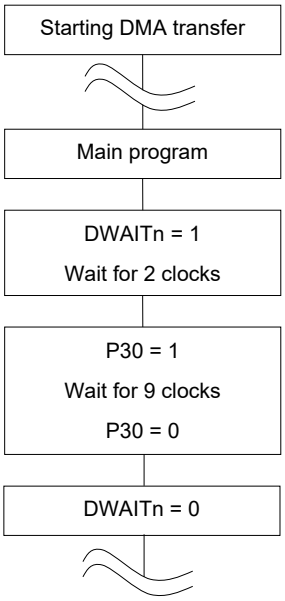
16.5.4 Holding DMA transfer pending by DWAITn bit

When DMA transfer is started, transfer is performed while an instruction is executed. At this time, the operation of the CPU is stopped and delayed for the duration of 2 clocks. If this poses a problem to the operation of the set system, a DMA transfer can be held pending by setting the DWAITn bit to 1. The DMA transfer for a transfer trigger that occurred while DMA transfer was held pending is executed after the pending status is canceled. However, because only one transfer trigger can be held pending for each channel, even if multiple transfer triggers occur for one channel during the pending status, only one DMA transfer is executed after the pending status is canceled.

To output a pulse with a width of 10 clocks of the operating frequency from the P30 pin, for example, the clock width increases to 12 if a DMA transfer is started midway. In this case, the DMA transfer can be held pending by setting the DWAITn bit to 1.

After setting the DWAITn bit to 1, it takes two clocks until a DMA transfer is held pending.

Figure 16-10. Example of Setting for Holding DMA Transfer Pending by DWAITn Bit



Caution **When DMA transfer is held pending while using both DMA channels, be sure to held the DMA transfer pending for both channels (by setting DWAIT0 and DWAIT1 to 1). If the DMA transfer of one channel is executed while that of the other channel is held pending, DMA transfer might not be held pending for the latter channel.**

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

16.5.5 Forced termination by software

After the DSTn bit is set to 0 by software, it takes up to 2 clocks until a DMA transfer is actually stopped and the DSTn bit is set to 0. To forcibly terminate a DMA transfer by software without waiting for occurrence of the interrupt (INTDMAn) of DMAn, therefore, perform either of the following processes.

<When using one DMA channel>

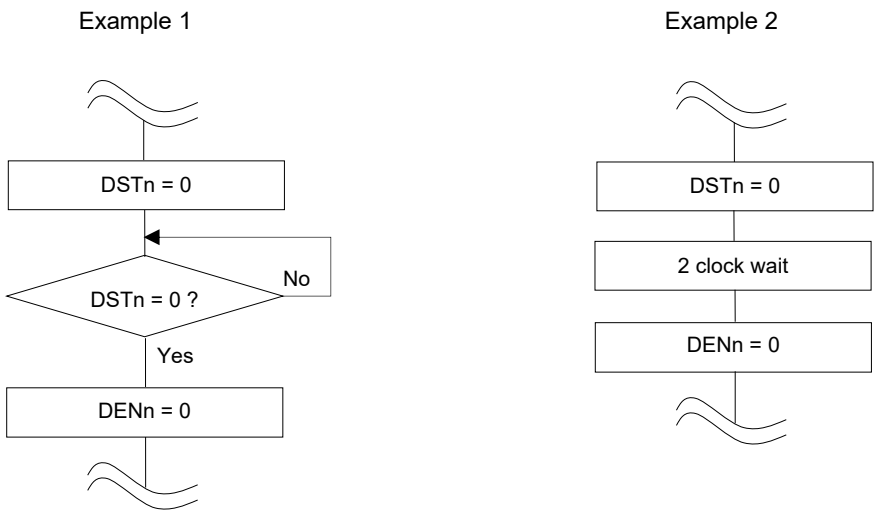
- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that the DSTn bit has actually been cleared to 0, and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).

- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction) two or more clocks after.

<When using two DMA channels>

- To forcibly terminate DMA transfer by software when using both DMA channels (by setting DSTn to 0), clear the DSTn bit to 0 after the DMA transfer is held pending by setting the DWAIT0 and DWAIT1 bits of both channels to 1. Next, clear the DWAIT0 and DWAIT1 bits of both channels to 0 to cancel the pending status, and then clear the DENn bit to 0.

Figure 16-11. Forced Termination of DMA Transfer (1/2)



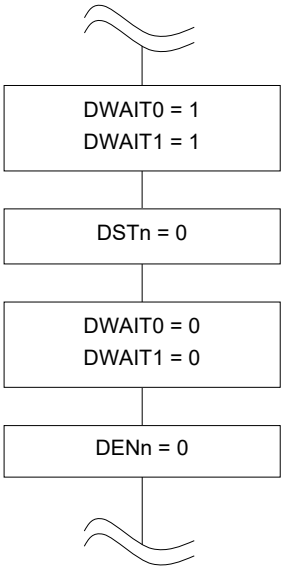
Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

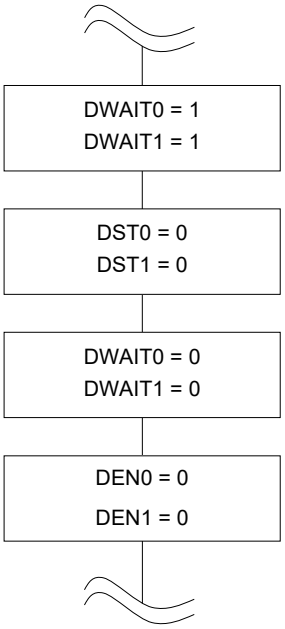
Figure 16-11. Forced Termination of DMA Transfer (2/2)

Example 3

- Procedure for forcibly terminating the DMA transfer for one channel if both channels are used



- Procedure for forcibly terminating the DMA transfer for both channels if both channels are used



Caution **In example 3, the system is not required to wait two clock cycles after the DWAITn bit is set to 1. In addition, the system does not have to wait two clock cycles after clearing the DSTn bit to 0, because more than two clock cycles elapse from when the DSTn bit is cleared to 0 to when the DENn bit is cleared to 0.**

Remarks **1.** n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

16.6 Cautions on Using DMA Controller

(1) Priority of DMA

During DMA transfer, a request from the other DMA channel is held pending even if generated. The pending DMA transfer is started after the ongoing DMA transfer is completed. If two or more DMA requests are generated at the same time, however, their priority are DMA channel 0 > DMA channel 1.

If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.

(2) DMA response time

The response time of DMA transfer is as follows.

Table 16-2. Response Time of DMA Transfer

	Minimum Time	Maximum Time
Response time	3 clocks	10 clocks[Note]

Note The maximum time necessary to execute an instruction from internal RAM is 16 clock cycles.

Cautions 1. The above response time does not include the two clock cycles required for a DMA transfer.

2. When executing a DMA pending instruction (see 16.6 (4)), the maximum response time is extended by the execution time of that instruction to be held pending.

3. Do not specify successive transfer triggers for a channel within a period equal to the maximum response time plus one clock cycle, because they might be ignored.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

(3) Operation in standby mode

The DMA controller operates as follows in the standby mode.

Table 16-3. DMA Operation in Standby Mode

Status	DMA Operation
HALT mode	Normal operation
STOP mode	Stops operation. If DMA transfer and STOP instruction execution contend, DMA transfer may be damaged. Therefore, stop DMA before executing the STOP instruction.

(4) DMA pending instruction

Even if a DMA request is generated, DMA transfer is held pending immediately after the following instructions.

- CALL !addr16
- CALL $!addr20
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- Write instructions for registers IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, and PSW each.
- Instruction for accessing the data flash memory

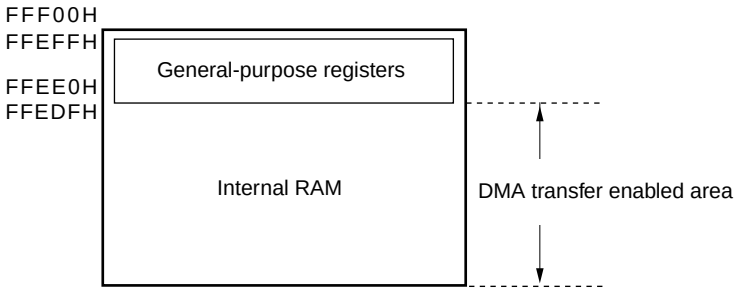
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

The address indicated by DMA RAM address register n (DRAn) is incremented during DMA transfer. If the address is incremented to an address in the general-purpose register area or exceeds the area of the internal RAM, the following operation is performed.

- In mode of transfer from SFR to RAM
  The data of that address is lost.

- In mode of transfer from RAM to SFR
  Undefined data is transferred to SFR.

In either case, malfunctioning may occur or damage may be done to the system. Therefore, make sure that the address is within the internal RAM area other than the general-purpose register area.



(6) Access to the register allocated in F0400H to F04FFH (2nd SFR)

If the above register is accessed after an next instruction execution from start of DMA transfer, a 1-clock wait will be inserted to the next instruction.

(7) Operation if instructions for accessing the data flash area

If the data flash area is accessed after an next instruction execution from start of DMA transfer, a 3-clock wait will be inserted to the next instruction.

```
Instruction 1
DMA transfer
Instruction 2  ◄- The wait of three clock cycles occurs.
MOV A,         ! DataFlash area
```

CHAPTER 17 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.

		32-pin	48-pin
Maskable interrupts	External	8	10
	Internal	20	20

17.1 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, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H).

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 default priority of vectored interrupt servicing. Default priority, see **Table 17-1**.

A standby release signal is generated and STOP, HALT, and SNOOZE 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.

17.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Table 17-1**). 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 17-1. Interrupt Source List (1/3)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	48-pin	32-pin
		Name	Trigger					
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75% of overflow time+1/2f_{IL})	Internal	00004H	(A)	√	√
	1	INTLVI	Voltage detection Note 4		00006H		√	√
	2	INTP0	Pin input edge detection	External	00008H	(B)	√	√
	3	INTP1			0000AH		√	√
	4	INTP2			0000CH		√	√
	5	INTP3			0000EH		√	√
	6	INTP4			00010H		√	√
	7	INTP5			00012H		√	√
	8	INTDMA0	End of DMA0 transfer	Internal	0001AH	(A)	√	√
	9	INTDMA1	End of DMA1 transfer		0001CH		√	√
	10	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end		0001EH		√	√
	11	INTTM00	End of timer channel 0 count or capture		00020H		√	√
	12	INTSR0/ INTCSI01/ INTIIC01	UART0 reception transfer end/CSI01 transfer end or buffer empty interrupt/IIC01 transfer end		00022H		√	√
	13	INTSRE0	UART0 reception communication error occurrence		00024H		√	√
		INTTM01H	End of timer channel 1 count or capture (at 8-bit timer operation)				√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 29 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 17-1.

3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

Table 17-1. Interrupt Source List (2/3)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	48-pin	32-pin
		Name	Trigger					
Maskable	14	INTTM03H	End of timer channel 3 count or capture (at 8-bit timer operation)	Internal	0002AH	(A)	√	√
	15	INTIICA0	End of IICA0 communication		0002CH		√	√
	16	INTTM01	End of timer channel 1 count or capture		0002EH		√	√
	17	INTTM02	End of timer channel 2 count or capture		00030H		√	√
	18	INTTM03	End of timer channel 3 count or capture		00032H		√	√
	19	INTAD	End of A/D conversion		00034H		√	√
	20	INTRTC	Fixed-cycle signal of real-time clock/alarm match detection		00036H		√	√
	21	INTIT	Interval signal detection		00038H		√	√
	22	INTKR	Key return signal detection	External	0003AH	(C)	√	–
	23	INTUSB	USB INT interrupt	Internal	0003CH	(A)	√	√
	24	INTRSUM	USB RESUME interrupt		0003EH		√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 29 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 17-1.

Table 17-1. Interrupt Source List (3/3)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type Note 2	48-pin	32-pin
		Name	Trigger					
Maskable	25	INTP6	Pin input edge detection	External	0004AH	(B)	√	–
	26	INTP8			0004EH		√	√
	27	INTP9			00050H		√	√
	28	INTMD	End of division operation/ Overflow of multiply-accumulation result occurs	Internal	0005EH	(A)	√	√
	29	INTFL	Reserved Note 3		00062H		√	√
Software	–	BRK	Execution of BRK instruction	–	0007EH	(D)	√	√
Reset	–	RESET	$\overline{\text{RESET}}$ pin input	–	00000H	–	√	√
		POR	Power-on-reset				√	√
		LVD	Voltage detection Note 4				√	√
		WDT	Overflow of watchdog timer				√	√
		TRAP	Execution of illegal instruction Note 5				√	√
		IAW	Illegal-memory access				√	√
		RAMTOP	RAM parity error				√	√

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 29 indicates the lowest priority.

2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figure 17-1.

3. Be used at the flash self programming library or the data flash library.

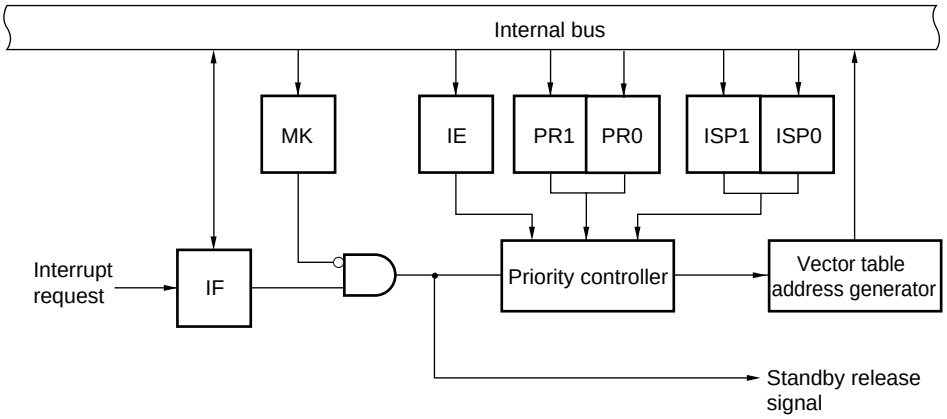
4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

5. When the instruction code in FFH is executed.

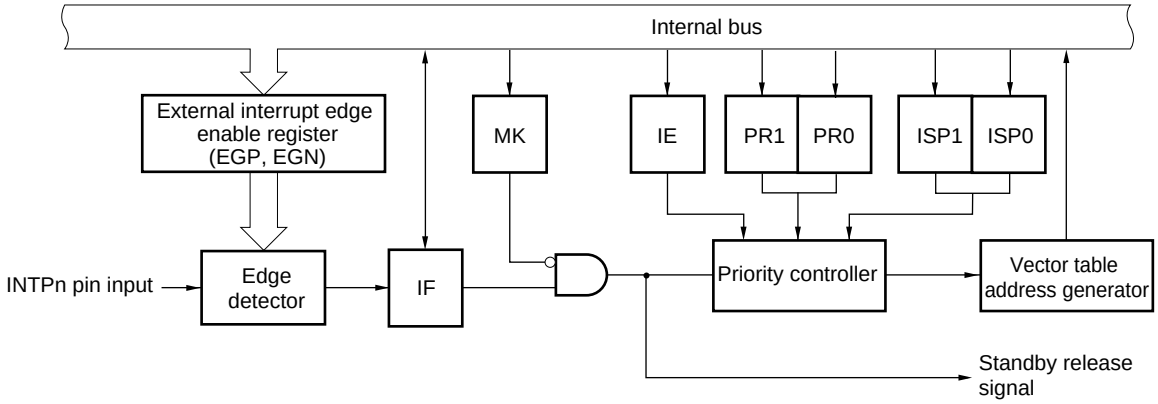
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 17-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)

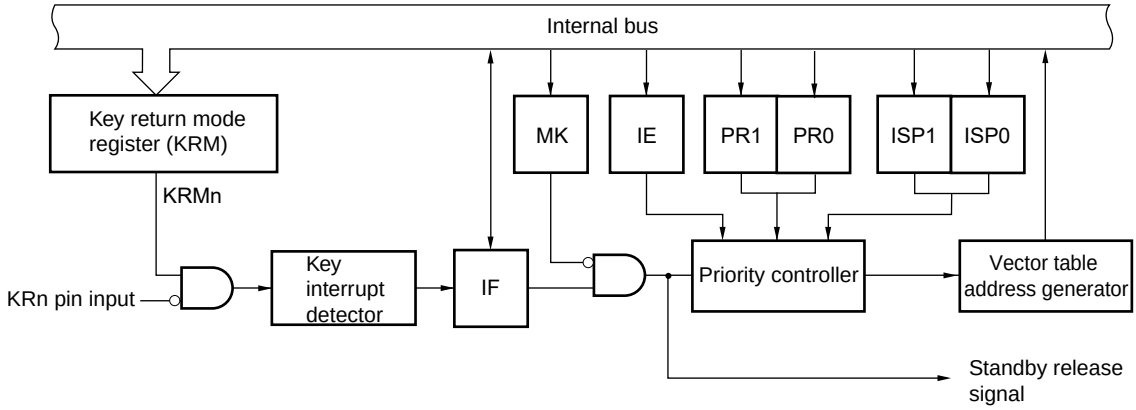


IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

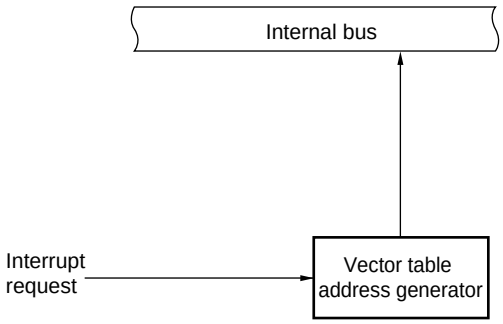
Remark 32-pin: n = 0 to 5, 8, 9
48-pin: n = 0 to 6, 8, 9

Figure 17-1. Basic Configuration of Interrupt Function (2/2)

(C) External maskable interrupt (INTKR)



(D) Software interrupt



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 48-pin: n = 0 to 5

17.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)
- External interrupt rising edge enable registers (EGP0, EGP1)
- External interrupt falling edge enable registers (EGN0, EGN1)
- Program status word (PSW)

Table 17-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 17-2. Flags Corresponding to Interrupt Request Sources (1/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	32-pin
		Register		Register		Register		
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	√	√
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		√	√
INTP0	PIF0		PMK0		PPR00, PPR10		√	√
INTP1	PIF1		PMK1		PPR01, PPR11		√	√
INTP2	PIF2		PMK2		PPR02, PPR12		√	√
INTP3	PIF3		PMK3		PPR03, PPR13		√	√
INTP4	PIF4		PMK4		PPR04, PPR14		√	√
INTP5	PIF5		PMK5		PPR05, PPR15		√	√

Table 17-2. Flags Corresponding to Interrupt Request Sources (2/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	32-pin
		Register		Register		Register		
INTDMA0	DMAIF0	IF0H	DMAMK0	MK0H	DMAPR00, DMAPR10	PR00H, PR10H	√	√
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11		√	√
INTST0[Note 1]	STIF0[Note 1]		STMK0[Note 1]		STPR00, STPR10[Note 1]		√	√
INTCSI00[Note 1]	CSIIF00[Note 1]		CSIMK00[Note 1]		CSIPR000, CSIPR100[Note 1]		√	√
INTIIC00[Note 1]	IICIF00[Note 1]		IICMK00[Note 1]		IICPR000, IICPR100[Note 1]		√	√
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√
INTSR0[Note 2]	SRIF0[Note 2]		SRMK0[Note 2]		SRPR00, SRPR10[Note 2]		√	√
INTCSI01[Note 2]	CSIIF01[Note 2]		CSIMK01[Note 2]		CSIPR001, CSIPR101[Note 2]		√	√
INTIIC01[Note 2]	IICIF01[Note 2]		IICMK01[Note 2]		IICPR001, IICPR101[Note 2]		√	√

Notes 1. If one of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 5 of the IF0H register is set to 1. Bit 5 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

2. If one of the interrupt sources INTSR0, INTCSI01, and INTIIC01 is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Table 17-2. Flags Corresponding to Interrupt Request Sources (3/3)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	32-pin
		Register		Register		Register		
INTSRE0 Note	SREIF0 Note	IF1L	SREMK0 Note	MK1L	SREPR00, SREPR10 Note	PR01L, PR11L	√	√
INTTM01H Note	TMIF01H Note		TMMK01H Note		TMPR001H, TMPR101H Note		√	√
INTTM03H	TMIF03H		TMMK03H		TMPR003H, TMPR103H		√	√
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		√	√
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		√	√
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		√	√
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		√	√
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H	√	√
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		√	√
INTIT	ITIF		ITMK		ITPR0, ITPR1		√	√
INTKR	KRIF		KRMK		KRPR0, KRPR1		√	–
INTUSB	USBIF		USBMK		USBPR0, USBPR1		√	√
INTRSUM	RSUIF		RSUMK		RSUPR0, RSUPR1		√	√
INTP6	PIF6	IF2L	PMK6	MK2L	PPR06, PPR16	PR02L, PR12L	√	–
INTP8	PIF8		PMK8		PPR08, PPR18		√	√
INTP9	PIF9		PMK9		PPR09, PPR19		√	√
INTMD	MDIF	IF2H	MDMK	MK2H	MDPR0, MDPR1	PR02H, PR12H	√	√
INTFL	FLIF		FLMK		FLPR0, FLPR1		√	√

Note Do not use a UART0 reception error interrupt and an interrupt of channel 1 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART0 reception error interrupt is not used (EOC01 = 0), UART0 and channel 1 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If one of the interrupt sources INTSRE0 and INTTM01H is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

17.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

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 an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

The IF0L, IF0H, IF1L, IF1H, IF2L, and IF2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers, the IF1L and IF1H registers, and the IF2L and IF2H registers are combined to form 16-bit registers IF0, IF1, and IF2, they can be set by a 16-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 17-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (48-pin) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
IF0H	SRIF0 CSIIF01 IICIF01	TMIF00	STIF0 CSIIF00 IICIF00	DMAIF1	DMAIF0	0	0	0

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
IF1L	TMIF03	TMIF02	TMIF01	IICAIF0	TMIF03H	0	0	SREIF0 TMIF01H

Address: FFFE3H After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	0	0	RSUIF	USBIF	KRIF	ITIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	2	1	0
IF2L	0	PIF9	PIF8	0	PIF6	0	0	0

Figure 17-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (48-pin) (2/2)

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
IF2H	FLIF	0	MDIF	0	0	0	0	0

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Cautions 1. The above is the bit layout for the 48-pin. The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 17-2. Be sure to set bits that are not available to the initial value.

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 using an 8-bit memory manipulation instruction 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.

17.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

The MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the MK0L and MK0H registers, the MK1L and MK1H registers, and the MK2L and MK2H registers are combined to form 16-bit registers MK0, MK1, and MK2, they can be set by a 16-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 17-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) (48-pin)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
MK0H	SRMK0 CSIMK01 IICMK01	TMMK00	STMK0 CSIMK00 IICMK00	DMAMK1	DMAMK0	1	1	1

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
MK1L	TMMK03	TMMK02	TMMK01	IICAMK0	TMMK03H	1	1	SREMK0 TMMK01H

Address: FFFE7H After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	1	1	RSUMK	USBMK	KRMK	ITMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	7	<6>	<5>	4	<3>	2	1	0
MK2L	1	PMK9	PMK8	1	PMK6	1	1	1

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	6	<5>	4	3	2	1	0
MK2H	FLMK	1	MDMK	1	1	1	1	1

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution The above is the bit layout for the 48-pin. The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 17-2. Be sure to set bits that are not available to the initial value.

17.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority level.

A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L, or 2H).

The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers can be set by a 1-bit or 8-bit memory manipulation instruction. If the PR00L and PR00H registers, the PR01L and PR01H registers, the PR02L and PR02H registers, the PR10L and PR10H registers, the PR11L and PR11H registers, and the PR12L and PR12H registers are combined to form 16-bit registers PR00, PR01, PR02, PR10, PR11, and PR12, they can be set by a 16-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 17-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (48-pin) (1/2)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
PR00H	SRPR00 CSIPR001 IICPR001	TMPR000	STPR00 CSIPR000 IICPR000	DMAPR01	DMAPR00	1	1	1

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	0
PR10H	SRPR10 CSIPR101 IICPR101	TMPR100	STPR10 CSIPR100 IICPR100	DMAPR11	DMAPR10	1	1	1

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
PR01L	TMPR003	TMPR002	TMPR001	IICAPR00	TMPR003H	1	1	SPEPR00 TMPR001H

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	1	<0>
PR11L	TMPR103	TMPR102	TMPR101	IICAPR10	TMPR103H	1	1	SPEPR10 TMPR101H

Figure 17-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (48-pin) (2/2)

Address: FFFEBH After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	1	1	RSUPR0	USBPR0	KRPR0	ITPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	1	1	RSUPR1	USBPR1	KRPR1	ITPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	7	<6>	<5>	4	<3>	2	1	0
PR02L	1	PPR09	PPR08	1	PPR06	1	1	1

Address: FFFDCH After reset: FFH R/W

Symbol	7	<6>	<5>	4	<3>	2	1	0
PR12L	1	PPR19	PPR18	1	PPR16	1	1	1

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	6	<5>	4	3	2	1	0
PR02H	FLPR0	1	MDPR0	1	1	1	1	1

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	6	<5>	4	3	2	1	0
PR12H	FLPR1	1	MDPR1	1	1	1	1	1

XXPR1X	XXPR0X	Priority level selection
0	0	Specify level 0 (high priority level)
0	1	Specify level 1
1	0	Specify level 2
1	1	Specify level 3 (low priority level)

Caution The above is the bit layout for the 48-pin. The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 17-2. Be sure to set bits that are not available to the initial value.

17.3.4 External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)

These registers specify the valid edge for INTP0 to INTP6, INTP8, and INTP9.

The EGP0, EGP1, EGN0, and EGN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 17-5. Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1) (48-pin)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	0	0	EGP9	EGP8

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	0	0	EGN9	EGN8

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 6, 8, 9)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 17-3 shows the ports corresponding to the EGPn and EGNn bits.

Table 17-3. Ports Corresponding to EGPn and EGNn bits

Detection Enable Bit		Edge Detection Port	Interrupt Request Signal	48-pin	32-pin
EGP0	EGN0	P137	INTP0	√	√
EGP1	EGN1	P50	INTP1	√	√
EGP2	EGN2	P51	INTP2	√	√
EGP3	EGN3	P30	INTP3	√	√
EGP4	EGN4	P31	INTP4	√	√
EGP5	EGN5	P16	INTP5	√	√
EGP6	EGN6	P140	INTP6	√	–
EGP8	EGN8	P74 (P00[Note])	INTP8	√	√
EGP9	EGN9	P75 (P01[Note])	INTP9	√	√

Note 32-pin product.

Caution **Select the port mode by clearing the EGPn and EGNn bits to 0 because an edge may be detected when the external interrupt function is switched to the port function.**

Remarks **1.** For edge detection port, see **2.1 Port Function.**
2. n = 0 to 6, 8, 9

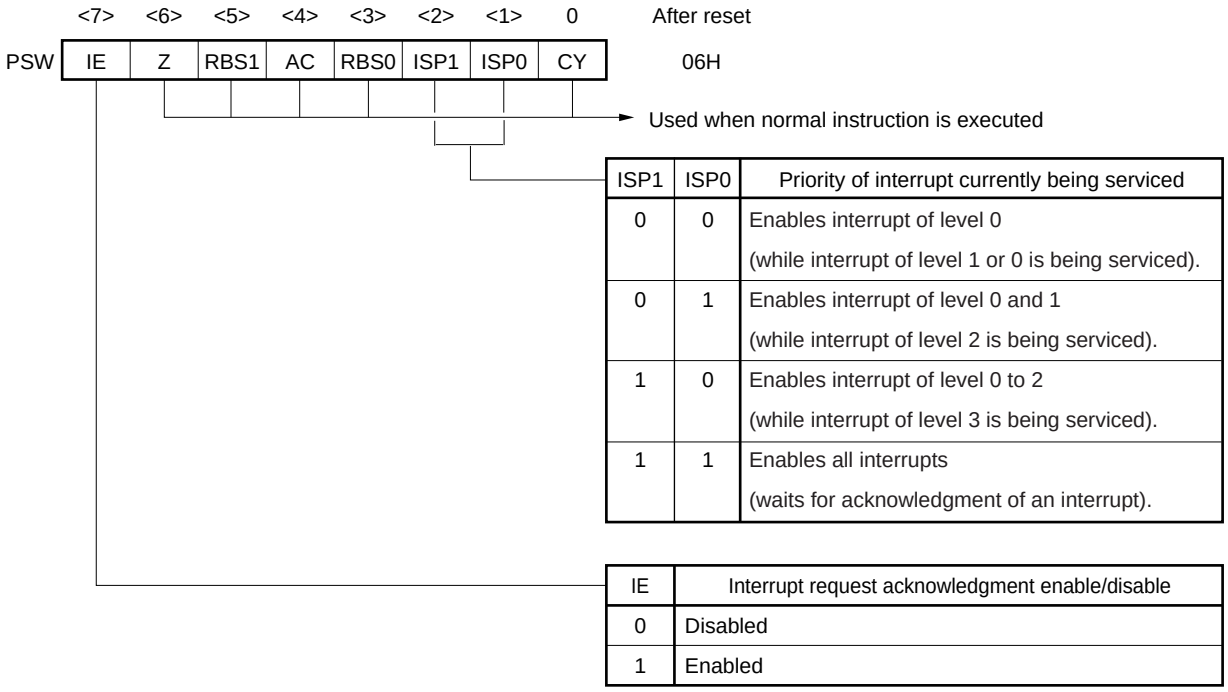
17.3.5 Program status word (PSW)

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 17-6. Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

17.4 Interrupt Servicing Operations

17.4.1 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 request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority 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 17-4 below.

For the interrupt request acknowledgment timing, see **Figures 17-8** and **17-9**.

Table 17-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	9 clocks	16 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.

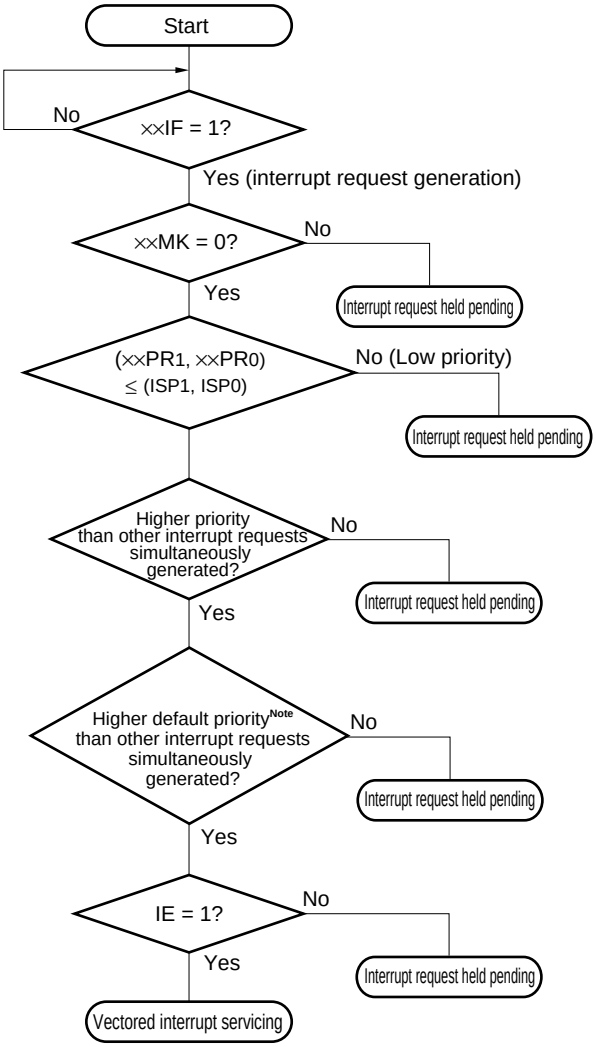
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 17-7 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.

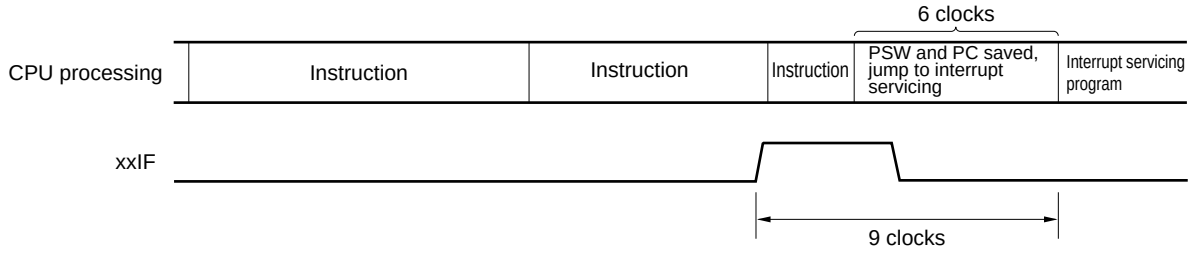
Figure 17-7. Interrupt Request Acknowledgment Processing Algorithm



××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 17-6**)

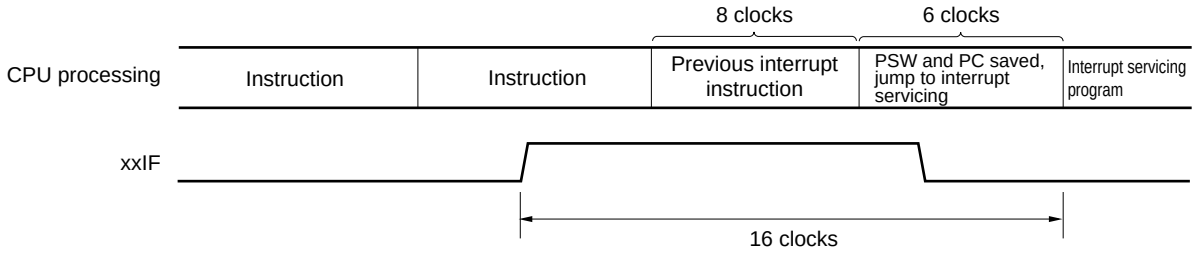
Note For the default priority, refer to **Table 17-1 Interrupt Source List**.

Figure 17-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 17-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

17.4.2 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.

17.4.3 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 being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. 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 17-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 17-10 shows multiple interrupt servicing examples.

Table 17-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		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.

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers.

PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)

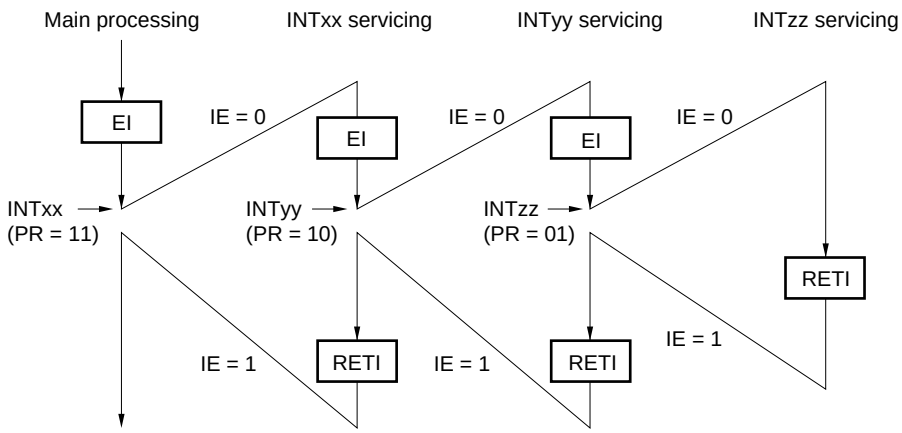
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1

PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0

PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

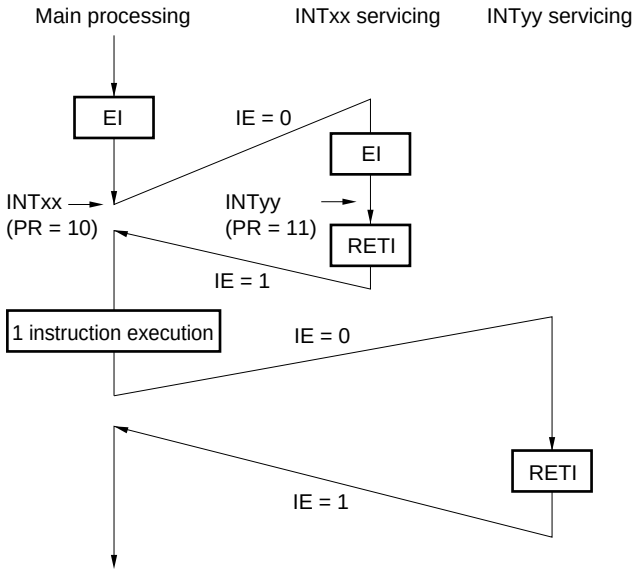
Figure 17-10. Examples of Multiple Interrupt Servicing (1/2)

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

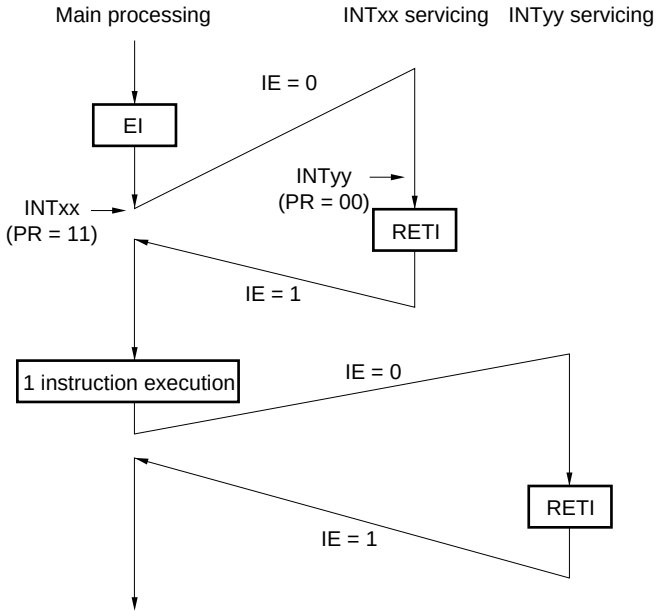


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

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 17-10. 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.

17.4.4 Interrupt request hold

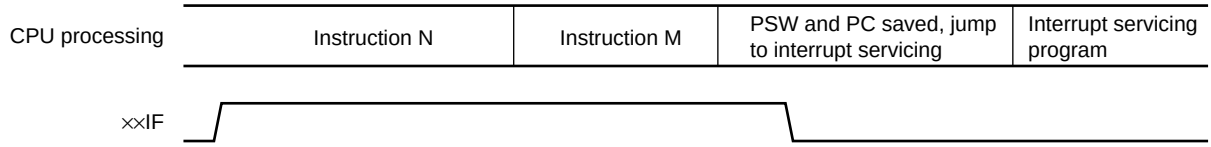
There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- 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
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers

Caution The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged.

Figure 17-11 shows the timing at which interrupt requests are held pending.

Figure 17-11. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 18 KEY INTERRUPT FUNCTION

Remark 32-pin product is not provided with the key interrupt function.

18.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by inputting a falling edge to the key interrupt input pins (KR0 to KR5).

Table 18-1. Assignment of Key Interrupt Detection Pins

Key interrupt input pins	Key return mode register (KRM)
KR0	KRM0
KR1	KRM1
KR2	KRM2
KR3	KRM3
KR4	KRM4
KR5	KRM5
KR6	KRM6
KR7	KRM7

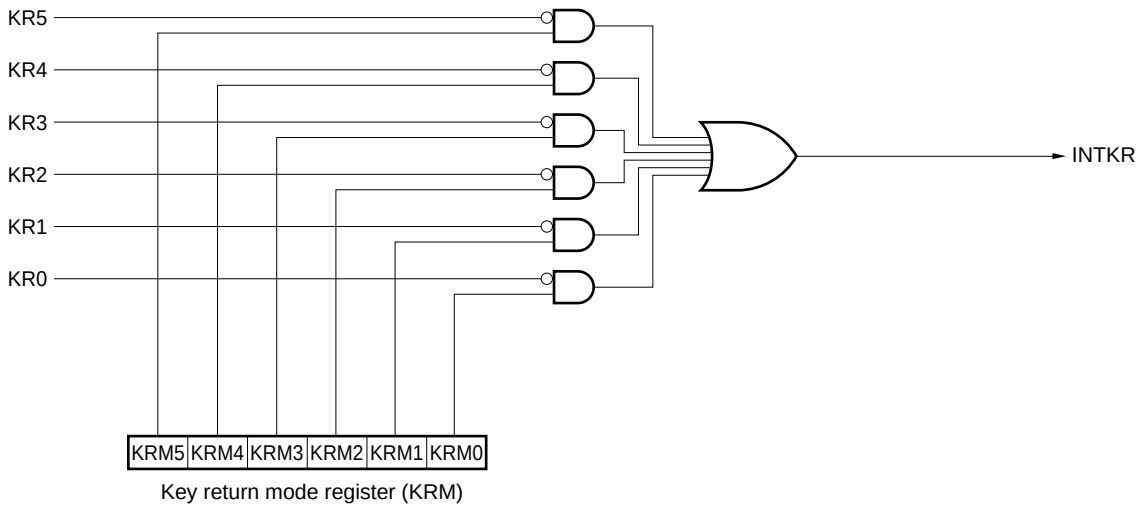
18.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 18-2. Configuration of Key Interrupt

Item	Configuration
Control register	Key return mode register (KRM) Port mode register (PM7)

Figure 18-1. Block Diagram of Key Interrupt



18.3 Register Controlling Key Interrupt

The key interrupt function is controlled by the following registers.

- Key return mode register (KRM)
- Port mode register (PM7)

18.3.1 Key return mode register (KRM)

KRM register controls the KR0 to KR5 signals.
The KRM register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 18-2. Format of Key Return Mode Register (KRM)

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	0	0	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Cautions 1. If any of the KRM0 to KRM5 bits used is set to 1, set bits 0 to 5 (PU70 to PU75) of the corresponding pull-up resistor register 7 (PU7) to 1.

2. An interrupt will be generated if the target bit of the KRM register is set while a low level is being input to the key interrupt input pin. To ignore this interrupt, set the KRM register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input low-level width (see AC characteristics).

3. The pins not used in the key interrupt mode can be used as normal ports.

Remark n = 0 to 5

18.3.2 Port mode register 7 (PM7)

When port 7 is used as the key interrupt input pins (KR0 toKR5), set the PM7n bit to 1. The output latches of P7n at this time may be 0 or 1. The PM7 register can be set by a 1-bit or 8-bit memory manipulation instruction. Reset signal generation sets this register to FFH.

Use of an on-chip pull-up resistor can be specified in 1-bit units by the pull-up resistor option register 7 (PU7).

Figure 18-3. Format of Port Mode Register 7

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM7	1	1	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W

PM7n	P7n pin I/O mode selection (n = 0 to 5)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

CHAPTER 19 STANDBY FUNCTION

19.1 Standby Function

The standby function reduces the operating current of the system, and the following three modes are available.

(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, high-speed on-chip oscillator, or subsystem clock 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.

(3) SNOOZE mode

In the case of CSI00 data reception and an A/D conversion request by the timer trigger signal (the interrupt request signal (INTRTC/INTIT)), the STOP mode is exited, the CSI00 data is received without operating the CPU, and A/D conversion is performed. This can only be specified when the high-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

In either of these three 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. **The STOP mode can be used only when the CPU is operating on the main system clock. Do not set to the STOP mode while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.**
2. **When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction (except SNOOZE mode setting unit).**
3. **When using CSI00, or the A/D converter in the SNOOZE mode, set up serial standby control register 0 (SSC0) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 12.3 Registers Controlling Serial Array Unit and 11.3 Registers Controlling A/D Converter.**
4. **The following sequence is recommended for power consumption 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.**
5. **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 25 OPTION BYTE.**

19.2 Registers Controlling Standby Function

The standby function is controlled by the following two registers.

- Subsystem clock supply mode control register (OSMC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For details of registers described above, see **CHAPTER 5 CLOCK GENERATOR**. For registers which control the SNOOZE mode, **CHAPTER 11 A/D CONVERTER** and **CHAPTER 12 SERIAL ARRAY UNIT**.

19.3 Standby Function Operation

19.3.1 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, high-speed on-chip oscillator clock, or subsystem 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 there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the HALT mode is immediately cleared if set.

Table 19-1. Operating Statuses in HALT Mode (1/2)

Item \ HALT Mode Setting	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})	When CPU Is Operating on PLL Clock (f_{PLL})
System clock	Clock supply to the CPU is stopped			
Main system clock f_{HOCO}	Operation continues (cannot be stopped)	Operation disabled		
Main system clock f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate	Cannot be stopped while the clock is supplied to the PLL
Main system clock f_{EX}		Cannot operate	Operation continues (cannot be stopped)	Cannot be stopped while the clock is supplied to the PLL
Main system clock f_{PLL}		Operation disabled	Operation disabled	Operation continues (cannot be stopped)
Subsystem clock f_{XT}	Status before HALT mode was set is retained			
Subsystem clock f_{EXS}				
f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply 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	Operation stopped			
Data flash memory				
RAM	Operation stopped (operable when DMA is executed)			
Port (latch)	Status before HALT mode was set is retained			
Timer array unit	Operable			
Real-time clock (RTC)				
12-bit interval timer				
Watchdog timer	See **CHAPTER 10 WATCHDOG TIMER**			
Clock output/buzzer output	Operable			
A/D converter				
Serial array unit (SAU)				
Serial interface (IICA)				
USB	Operable (low speed mode transmission only)	Operation disabled		Operable
Multiplier and divider/multiply-accumulator	Operable			
DMA controller				
Power-on-reset function				
Voltage detection function				
External interrupt				
Key interrupt function				
CRC operation function: High-speed CRC				
CRC operation function: General-purpose CRC	In the calculation of the RAM area, operable when DMA is executed only			
RAM parity error detection function	Operable when DMA is executed only			
RAM guard function				
SFR guard function				
Illegal-memory access detection function				

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_{HOCO}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock
f_{XT}: XT1 clock f_{EXS}: External subsystem clock

Table 19-1. Operating Statuses in HALT Mode (2/2)

Item \ HALT Mode Setting	When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock	
	When CPU Is Operating on XT1 Clock (f_{XT})	When CPU Is Operating on External Subsystem Clock (f_{EXS})
System clock	Clock supply to the CPU is stopped	
Main system clock: f_{HOCO}, f_X, f_{EX}, f_{PLL}	Operation disabled	
Subsystem clock: f_{XT}	Operation continues (cannot be stopped)	Cannot operate
Subsystem clock: f_{EXS}	Cannot operate	Operation continues (cannot be stopped)
f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply 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		
Data flash memory		
RAM	Operation stopped (operable when DMA is executed)	
Port (latch)	Status before HALT mode was set is retained	
Timer array unit	Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
Real-time clock (RTC)	Operable	
12-bit interval timer		
Watchdog timer	See **CHAPTER 10 WATCHDOG TIMER**	
Clock output/buzzer output	Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
A/D converter	Operation disabled	
Serial array unit (SAU)	Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
Serial interface (IICA)	Operation disabled	
USB		
Multiplier and divider/multiply-accumulator	Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).	
DMA controller		
Power-on-reset function	Operable	
Voltage detection function		
External interrupt		
Key interrupt function		
CRC operation function: High-speed CRC	Operation disabled	
CRC operation function: General-purpose CRC	In the calculation of the RAM area, operable when DMA is executed only	
RAM parity error detection function	Operable when DMA is executed only	
RAM guard function		
SFR guard function		
Illegal-memory access detection function		

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_{HOCO}: High-speed on-chip oscillator clock f_{EX}: External main system clock
f_{IL}: Low-speed on-chip oscillator clock f_{XT}: XT1 clock
f_X: X1 clock f_{EXS}: External subsystem clock

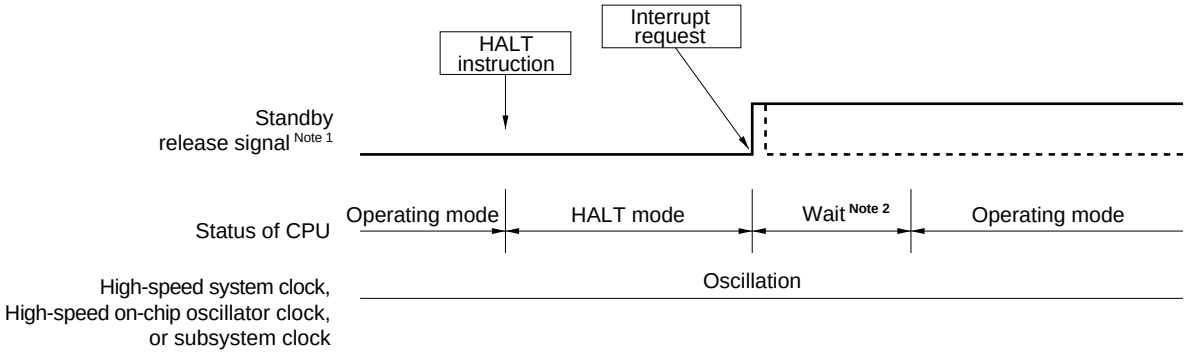
(2) HALT mode release

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 19-1. HALT Mode Release by Interrupt Request Generation



Notes 1. For details of the standby release signal, see **Figure 16-1**.

2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out
  - Main system clock: 15 to 16 clock
  - Subsystem clock (RTCLPC = 0): 10 to 11 clock
  - Subsystem clock (RTCLPC = 1): 11 to 12 clock
- When vectored interrupt servicing is not carried out
  - Main system clock: 9 to 10 clock
  - Subsystem clock (RTCLPC = 0): 4 to 5 clock
  - Subsystem clock (RTCLPC = 1): 5 to 6 clock

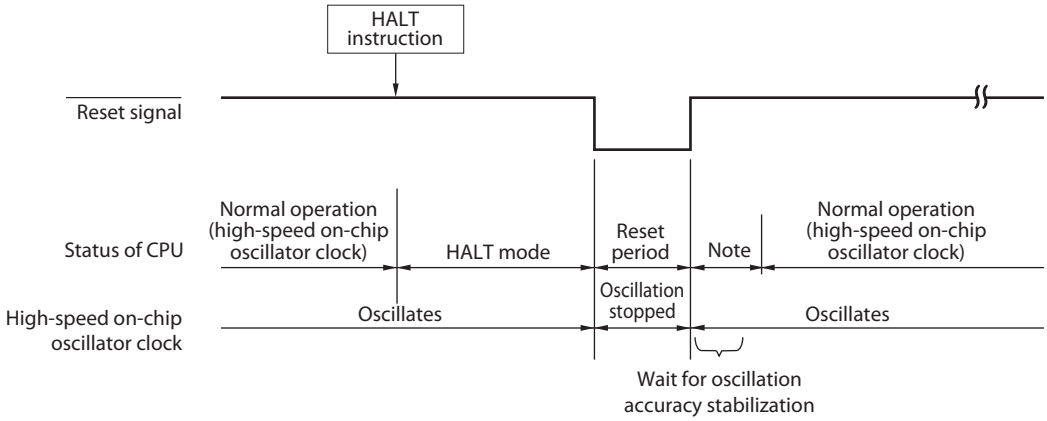
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

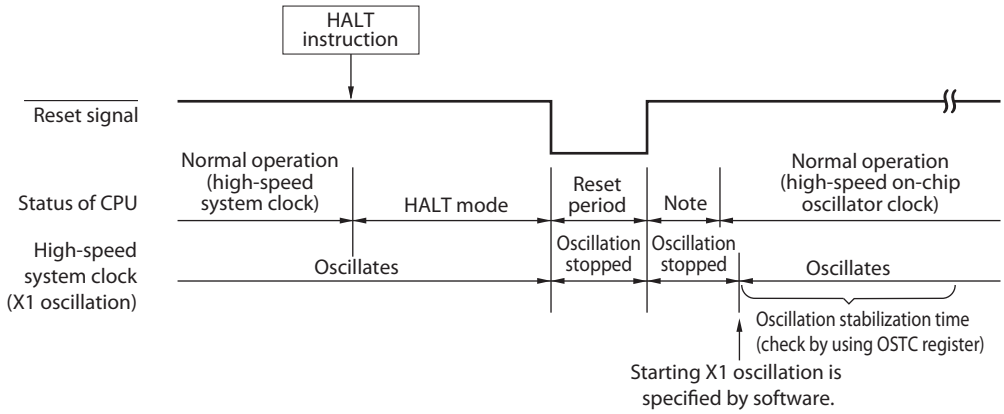
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 19-2. HALT Mode Release by Reset

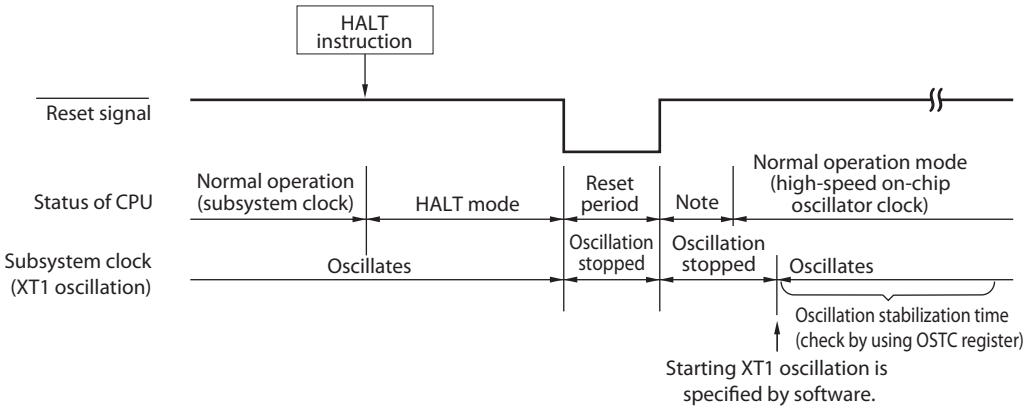
(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock



(3) When subsystem clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 20 RESET FUNCTION.**

For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 21 POWER-ON-RESET CIRCUIT**.

19.3.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the high-speed on-chip oscillator clock, X1 clock, or external main system clock.

Cautions 1. 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.

2. When shifting to STOP mode, be sure to stop the PLL operation by setting the DSCON bit (bit 0 in the DSCCTL register) before executing the STOP instruction.

The operating statuses in the STOP mode are shown below.

Table 19-2. Operating Statuses in STOP Mode

Item \ STOP Mode Setting	When STOP 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})	When CPU Is Operating on PLL Clock(f_{PLL})
System clock	Clock supply to the CPU is stopped			
Main system clock f_{HOCO}	Stopped			
Main system clock f_X				
Main system clock f_{EX}				
Main system clock f_{PLL}	Operation disabled			
Subsystem clock f_{XT}	Status before STOP mode was set is retained			
Subsystem clock f_{EXS}				
f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply 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				
Data flash memory	Operation stopped			
RAM	Operation stopped (operable when DMA is executed)			
Port (latch)	Status before STOP mode was set is retained			
Timer array unit	Operation disabled			
Real-time clock (RTC)	Operable			
12-bit interval timer				
Watchdog timer	See **CHAPTER 10 WATCHDOG TIMER**			
Clock output/buzzer output	Operates when the subsystem clock is selected as the clock source for counting and the RTCLPC bit is 0 (operation is disabled when a clock other than the subsystem clock is selected and the RTCLPC bit is not 0).			
A/D converter	Wakeup operation is enabled (switching to the SNOOZE mode)			
Serial array unit (SAU)	Wakeup operation is enabled only for CSI00 (switching to the SNOOZE mode) Operation is disabled for anything other than CSI00			
Serial interface (IICA)	Wakeup by address match operable			
USB	Operation disabled			
Multiplier and divider/multiply-accumulator				
DMA controller				
Power-on-reset function	Operable			
Voltage detection function				
External interrupt				
Key interrupt function				
CRC operation function: High-speed CRC	Operation stopped			
CRC operation function: General-purpose CRC				
RAM parity error detection function				
RAM guard function				
SFR guard function				
Illegal-memory access detection function				

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_{HOCO}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock
f_{XT}: XT1 clock f_{EXS}: External subsystem clock

(2) STOP mode release

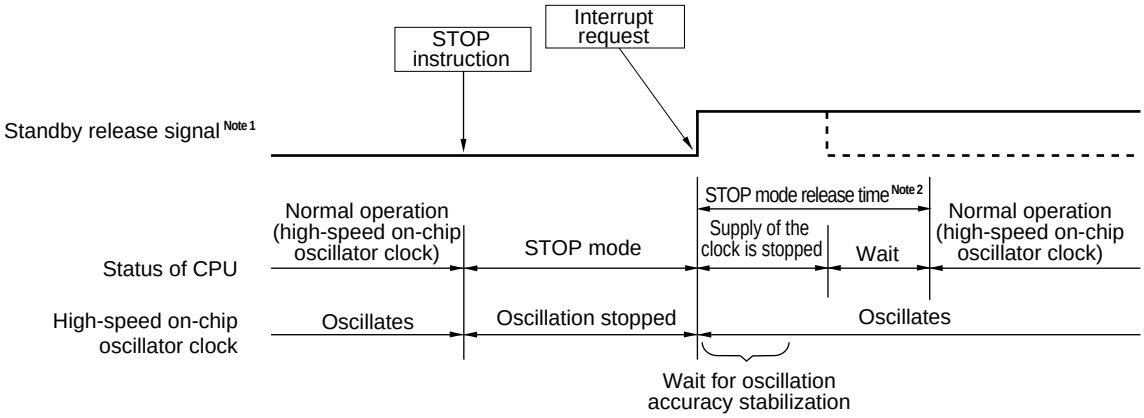
The STOP mode can be released by the following two sources.

(a) 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 19-3. 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 17-1**

2. STOP mode release time

Supply of the clock is stopped: 18 to 105 μs

Wait

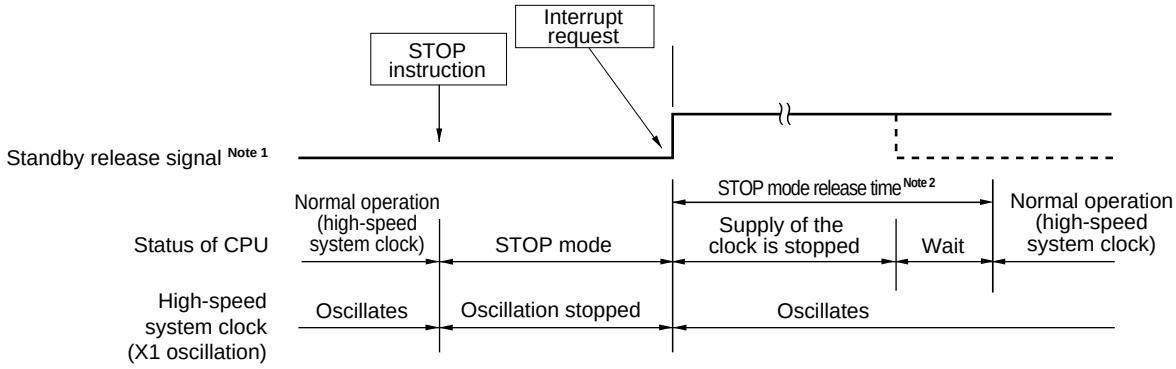
- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Remarks 1. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

2. The period during which clock supply stops depends on the temperature conditions and the STOP mode period.

Figure 19-3. STOP Mode Release by Interrupt Request Generation (1/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock



Notes 1. For details of the standby release signal, see **Figure 17-1**

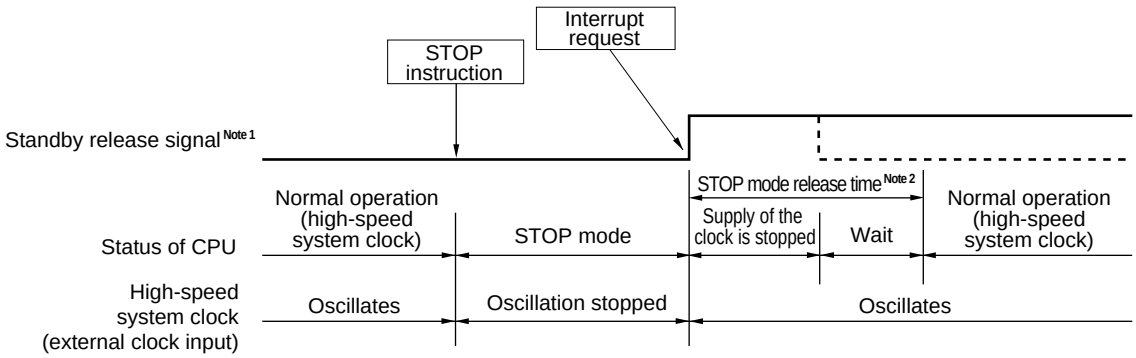
2. STOP mode release time

Supply of the clock is stopped: 18 μs to "whichever is longer 105 μs and the oscillation stabilization time (set by OSTS)"

Wait

- When vectored interrupt servicing is carried out: 10 to 11 clocks
- When vectored interrupt servicing is not carried out: 4 to 5 clocks

(3) When high-speed system clock (external clock input) is used as CPU clock



Notes 1. For details of the standby release signal, see **Figure 17-1**

2. STOP mode release time

Supply of the clock is stopped: 18 to 105 μs

Wait

- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

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 broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

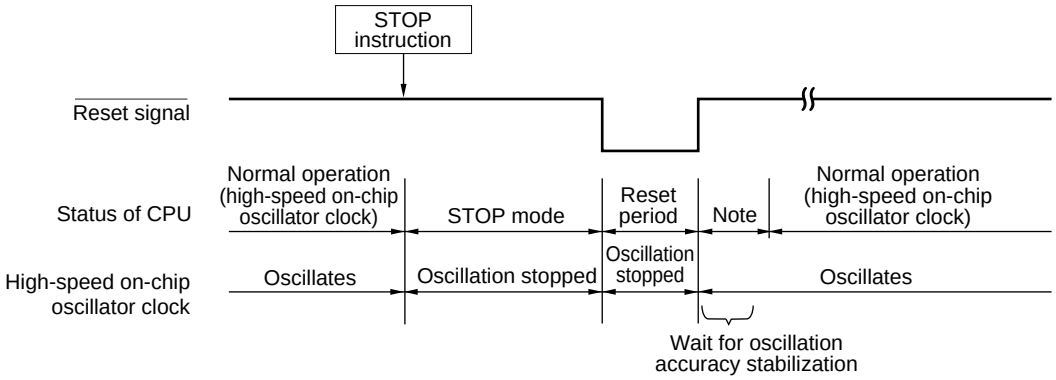
2. The period during which clock supply stops depends on the temperature conditions and the STOP mode period.

(b) Release by reset signal generation

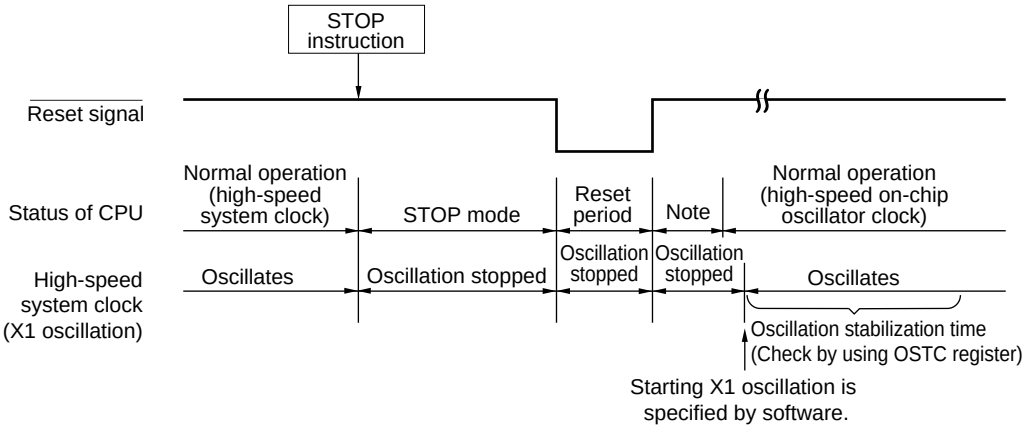
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 19-4. STOP Mode Release by Reset

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 20 RESET FUNCTION.**
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 21 POWER-ON-RESET CIRCUIT.**

19.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for CSI00, or the A/D converter. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock.

When using CSI00 in the SNOOZE mode, set the SWC0 bit of the serial standby control register 0 (SSC0) to 1 immediately before switching to the STOP mode. For details, see **12.3 Registers Controlling Serial Array Unit**.

When using the A/D converter in the SNOOZE mode, set the AWC bit of the A/D converter mode register 2 (ADM2) to 1 immediately before switching to the STOP mode. For details, see **11.3 Registers Used in A/D Converter**.

In SNOOZE mode transition, wait status to be only following time.

Transition time from STOP mode to SNOOZE mode: 18 μs to 105 μs

Remark Transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

From SNOOZE to normal operation

- When vectored interrupt servicing is carried out:
  HS (High-speed main) mode : 4.99 μs to 9.44 μs + 7 clocks
- When vectored interrupt servicing is not carried out:
  HS (High-speed main) mode : 4.99 μs to 9.44 μs + 1 clock

The operating statuses in the SNOOZE mode are shown below.

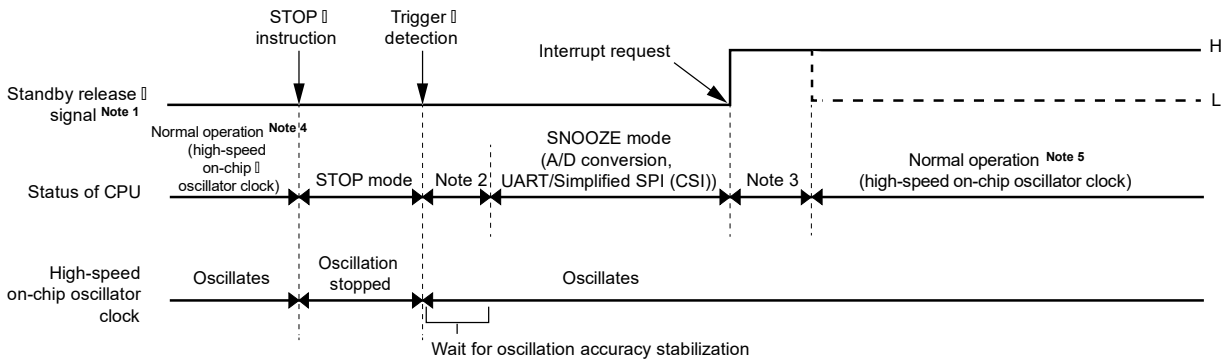
Table 19-3. Operating Statuses in SNOOZE Mode

Item \ SNOOZE Mode Setting			When Inputting CSI00 Data Reception Signal or A/D Converter Timer Trigger Signal While in STOP Mode
			When CPU Is Operating on High-speed on-chip oscillator clock (f_{IH})
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{HOCO}	Operation started
		f_X	Stopped
		f_{EX}	
		f_{PLL}	
	Subsystem clock	f_{XT}	Use of the status while in the STOP mode continues
		f_{EXS}	
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply 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			
Data flash memory			
RAM			
Port (latch)			Use of the status while in the STOP mode continues
Timer array unit			Operation disabled
Real-time clock (RTC)			Operable
12-bit interval timer			
Watchdog timer			See **CHAPTER 10 WATCHDOG TIMER**
Clock output/buzzer output			Operates when the subsystem clock is selected as the clock source for counting and the RTCLPC bit is 0 (operation is disabled when a clock other than the subsystem clock is selected and the RTCLPC bit is not 0).
A/D converter			Operable
Serial array unit (SAU)			Operable only CSI00 only. Operation disabled other than CSI00.
Serial interface (IICA)			Operation disabled
USB			
Multiplier and divider/multiply-accumulator			
DMA controller			
Power-on-reset function			Operable
Voltage detection function			
External interrupt			
Key interrupt function			
CRC operation function	High-speed CRC		Operation disabled
	General-purpose CRC		
RAM parity error detection function			
RAM guard function			
SFR guard function			
Illegal-memory access detection function			

Remark Operation stopped: Operation is automatically stopped before switching to the SNOOZE mode.
Operation disabled: Operation is stopped before switching to the SNOOZE mode.
f_{HOCO}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock
f_{XT}: XT1 clock f_{EXS}: External subsystem clock

(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode

Figure 19-5. When the Interrupt Request Signal is Generated in the SNOOZE Mode



Notes 1. For details of the standby release signal, see **Figure 17-1**.

2. Transition time from STOP mode to SNOOZE mode

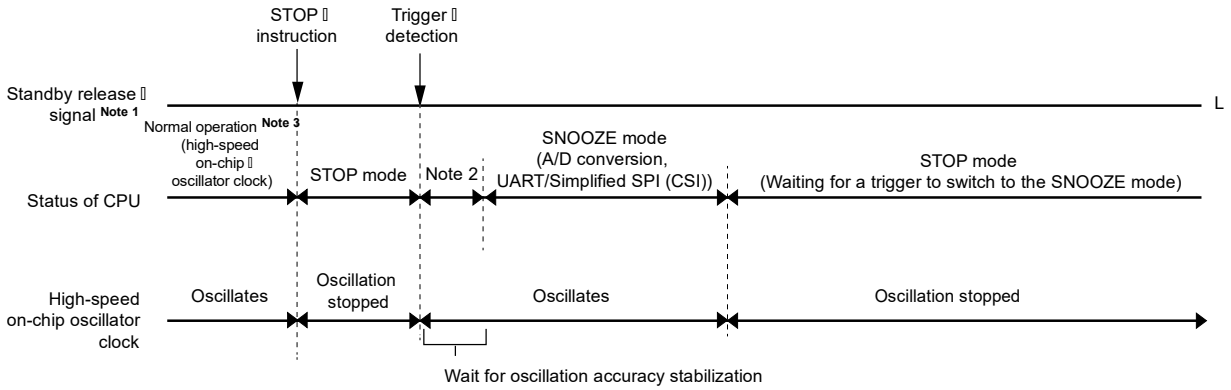
3. Transition time from SNOOZE mode to normal operation

4. Enable the SNOOZE mode (AWC = 1 or SWCm = 1) immediately before switching to the STOP mode.

5. Be sure to release the SNOOZE mode (AWC = 0 or SWCm = 0) immediately after return to the normal operation.

(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 19-6. When the Interrupt Request Signal is not Generated in the SNOOZE Mode



Notes 1. For details of the standby release signal, see **Figure 17-1**.

2. Transition time from STOP mode to SNOOZE mode

3. Enable the SNOOZE mode (AWC = 1 or SWCm = 1) immediately before switching to the STOP mode.

Remark For details of the SNOOZE mode function, see **CHAPTER 11 A/D CONVERTER** and **CHAPTER 12 SERIAL ARRAY UNIT**.

CHAPTER 20 RESET FUNCTION

The following seven 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 power-on-reset (POR) circuit
(4) Internal reset by comparison of supply voltage of the voltage detector (LVD) and detection voltage
(5) Internal reset by execution of illegal instruction[Note]
(6) Internal reset by RAM parity error
(7) Internal reset by illegal-memory access

External and internal resets start program execution from the address at 00000H and 00001H when the reset signal is generated.

A reset is effected when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POR and LVD circuit voltage detection, execution of illegal instruction[Note], RAM parity error or illegal-memory access, and each item of hardware is set to the status shown in Tables 20-1.

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

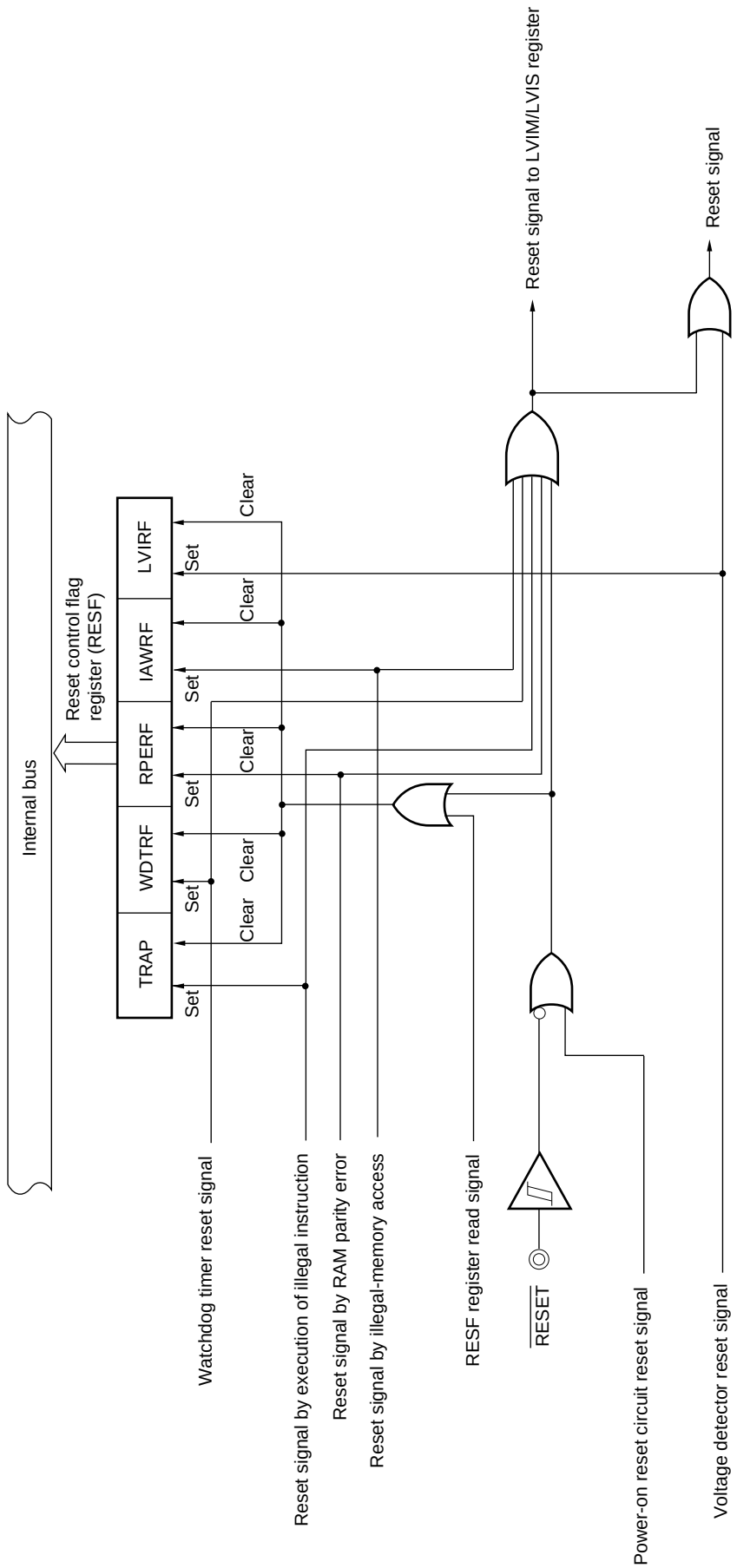
Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 us or more within the operating voltage range shown in 30.4 or 31.4 AC Characteristics, and then input a high level to the pin.

2. During reset input, the X1 clock, XT1 clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input and external subsystem clock input become invalid.

3. The port pins become the following state because each SFR and 2nd SFR are initialized after reset.
- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).**
- **P130: Low level during the reset period or after receiving a reset signal.**
- **Ports other than P40 and P130: High-impedance during the reset period or after receiving a reset signal.**

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD}: LVD detection voltage

Figure 20-1. Block Diagram of Reset Function



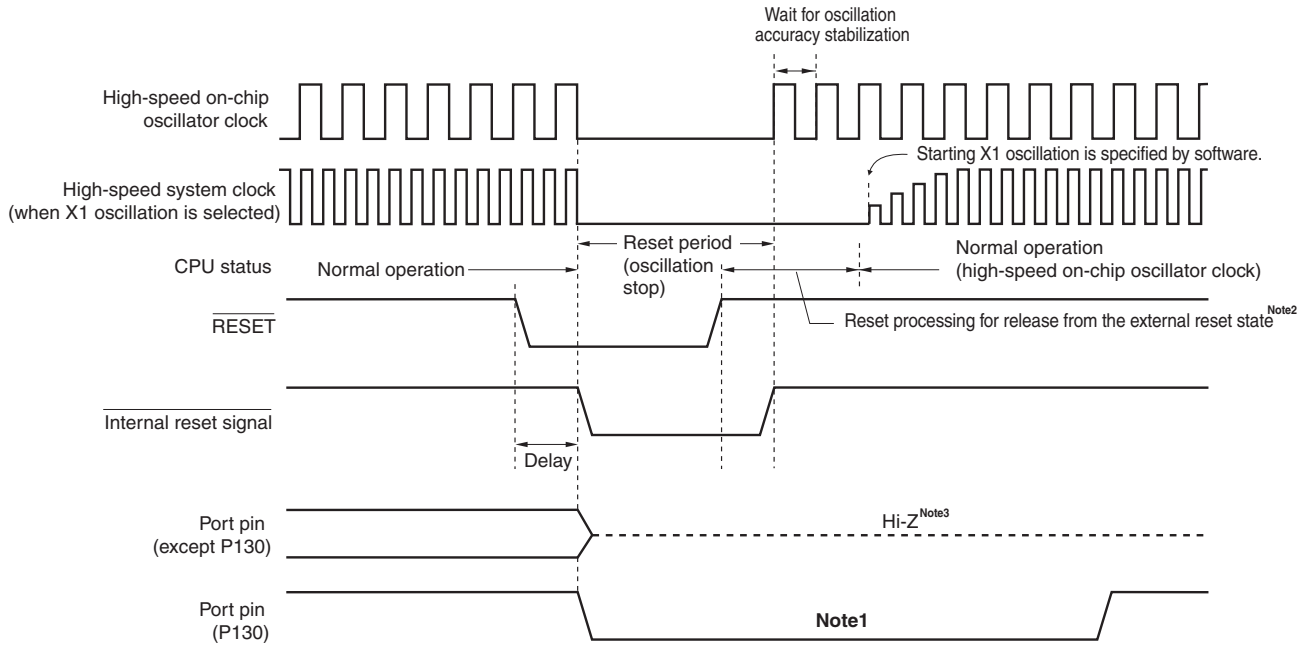
Caution An LVD circuit internal reset does not reset the LVD circuit.

Remarks 1. LVIM: Voltage detection register
2. LVIS: Voltage detection level register

20.1 Timing of Reset Operation

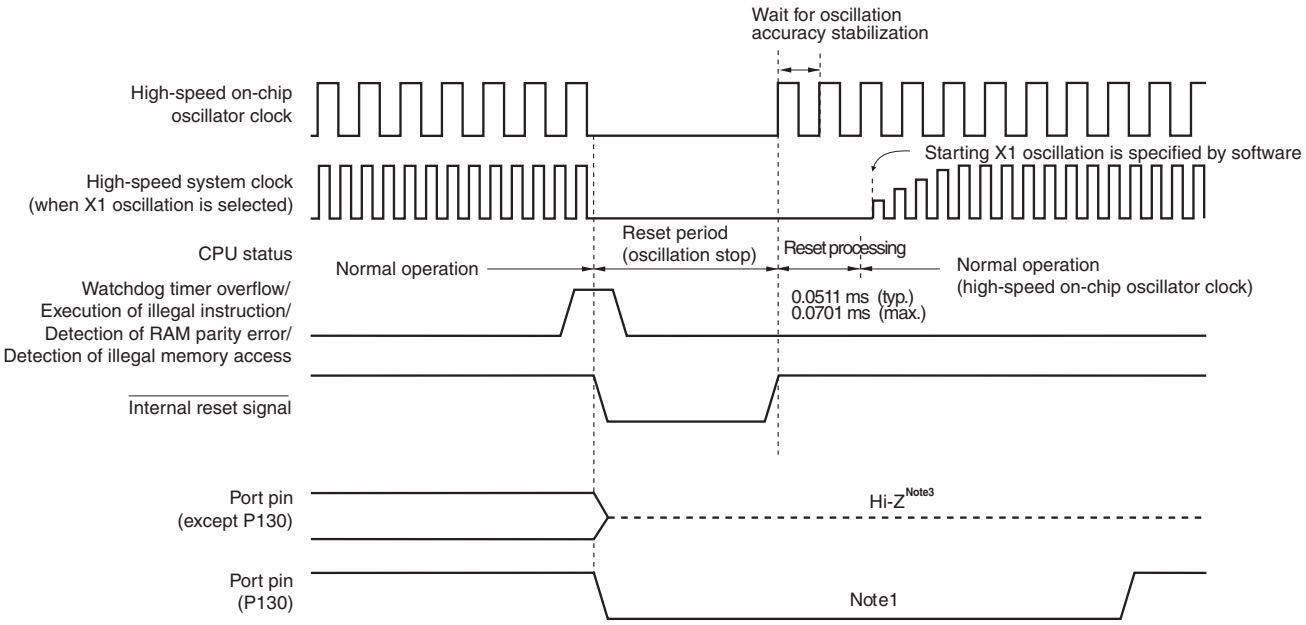
This LSI is reset by input of the low level on the $\overline{\text{RESET}}$ pin and released from the reset state by input of the high level on the $\overline{\text{RESET}}$ pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 20-2. Timing of Reset by $\overline{\text{RESET}}$ Input



Release from the reset state is automatic in the case of a reset due to a watchdog timer overflow, execution of an illegal instruction, detection of a RAM parity error, or detection of illegal memory access. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 20-3. Timing of Reset Due to Execution of Illegal Instruction, Watchdog Timer Overflow, RAM Parity Error, or Illegal Memory Access



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

Notes 1. When P130 is set to high-level output before reset is effected, the output signal of P130 can be dummy-output as a reset signal to an external device, because P130 outputs a low level when reset is effected. To release a reset signal to an external device, set P130 to high-level output by software.

2. Reset times (times for release from the external reset state)

After the first release of the POR: 0.672 ms (typ.), 0.832 ms (max.) when the LVD is in use.
0.399 ms (typ.), 0.519 ms (max.) when the LVD is off.

After the second release of the POR: 0.531 ms (typ.), 0.675 ms (max.) when the LVD is in use.
0.259 ms (typ.), 0.362 ms (max.) when the LVD is off.

After power is supplied, a voltage stabilization waiting time of about 0.99 ms (typ.) and up to 2.30 ms (max.) is required before reset processing starts after release of the external reset.

3. The state of P40 is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset or after receiving a reset signal (connected to the internal pull-up resistor).

Reset by POR and LVD circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD}$ after the reset. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts. For details, see **CHAPTER 20 POWER-ON-RESET CIRCUIT** or **CHAPTER 21 VOLTAGE DETECTOR**.

20.2 Operation Statuses During Reset Period

Table 20-1 shows the states of operation during reset periods. Table 20-2 shows the states of the hardware after receiving a reset signal.

Table 20-1. Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{HOCO}	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_{PLL}	Operation stopped
	Subsystem clock	f_{XT}	Operation stopped (the XT1 and XT2 pins are input port mode)
		f_{EXS}	Clock input invalid (the pin is input port mode)
	f_{IL}		Operation stopped
CPU			
Code flash memory			Operation stopped
Data flash memory			Operation stopped
RAM			Operation stopped
Port (latch)			High impedance Note
Timer array unit			Operation stopped
Real-time clock (RTC)			
12-bit Interval timer			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Serial array unit (SAU)			
Serial interface (IICA)			
USB			
Multiplier & divider, multiply-accumulator			
DMA controller			
Power-on-reset function			Detection operation possible
Voltage detection function			Operation is possible in the case of an LVD reset and stopped in the case of other types of reset.
External interrupt			Operation stopped
Key interrupt function			
CRC operation function	High-speed CRC		
	General-purpose CRC		
RAM parity error detection function			
RAM guard function			
SFR guard function			
Illegal-memory access detection function			

(**Note** and **Remark** is listed on the next page.)

Note P40 and P130 become the following state.

- P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset (connected to the internal pull-up resistor).
- P130: Low level during the reset period

Remark f_{HOCO}: High-speed on-chip oscillator clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{XT}: XT1 oscillation clock
f_{EXS}: External subsystem clock
f_{IL}: Low-speed on-chip oscillator clock

Table 20-2. Hardware Statuses After Reset Acknowledgment

Hardware		After Reset Acknowledgment[Note]
Program counter (PC)		The contents of the reset vector table (00000H, 00001H) 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 The special function register (SFR) mounted depend on the product. See **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

20.3 Register for Confirming Reset Source

20.3.1 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.

$\overline{\text{RESET}}$ input, reset by power-on-reset (POR) circuit, and reading the RESF register clear TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags.

Figure 20-4. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: Undefined[Note 1] R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

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.

RPERF	Internal reset request t by RAM parity
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

IAWRF	Internal reset request t by illegal-memory access
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by voltage detector (LVD)
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. See **Table 20-3**.

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

Cautions 1. Do not read data by a 1-bit memory manipulation instruction.

2. When enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the used RAM area at data access or the used RAM area + 10 bytes at execution of instruction from the RAM area. Reset generation enables RAM parity error resets (RPERDIS = 0). For details, see 23.3.3 RAM parity error detection function.

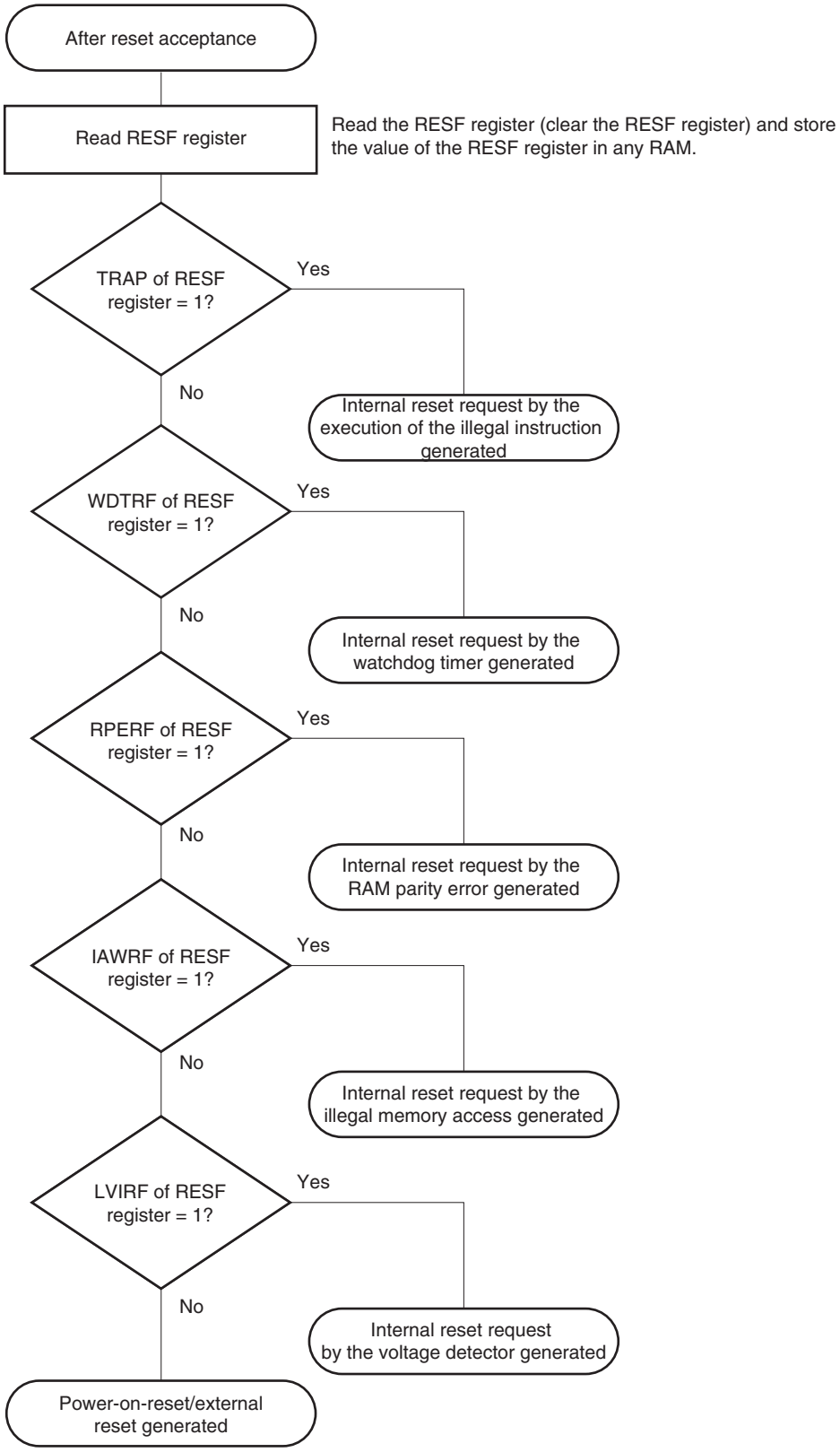
The status of the RESF register when a reset request is generated is shown in Table 20-3.

Table 20-3. RESF Register Status When Reset Request Is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM Parity Error	Reset by Illegal-memory Access	Reset by LVD
TRAP	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held	Held
WDTRF			Held	Set (1)			
RPERF				Held	Set (1)		
IAWRF					Held	Set (1)	
LVIRF						Held	Set (1)

The RESF register is automatically cleared when it is read by an 8-bit memory manipulation instruction. Figure 20-5 shows the procedure for checking a reset source.

Figure 20-5. Example of Procedure for Checking Reset Source



Caution The flow described above is an example of the procedure for checking.

CHAPTER 21 POWER-ON-RESET CIRCUIT

21.1 Functions of Power-on-reset Circuit

The power-on-reset circuit (POR) has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds the detection voltage (V_{POR}). Note that a reset state must be retained by utilizing the voltage detection function or external reset pin until the operation voltage becomes in the ranges defined in **30.4** and **31.4 AC Characteristics**.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR}), generates internal reset signal when $V_{DD} < V_{PDR}$. Note that this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection function or external reset pin before the operation voltage falls below the range defined in **30.4** and **31.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Caution If an internal reset signal is generated in the POR circuit, the reset control flag register (RESF) is cleared.

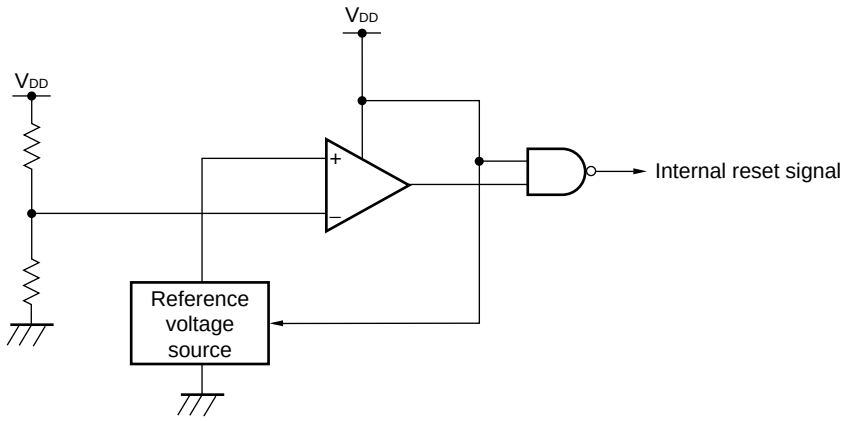
Remarks 1. This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. For details of the RESF register, see **CHAPTER 20 RESET FUNCTION**.

2. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **30.6.3** or **31.6.3 POR circuit characteristics**.

21.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 21-1.

Figure 21-1. Block Diagram of Power-on-reset Circuit

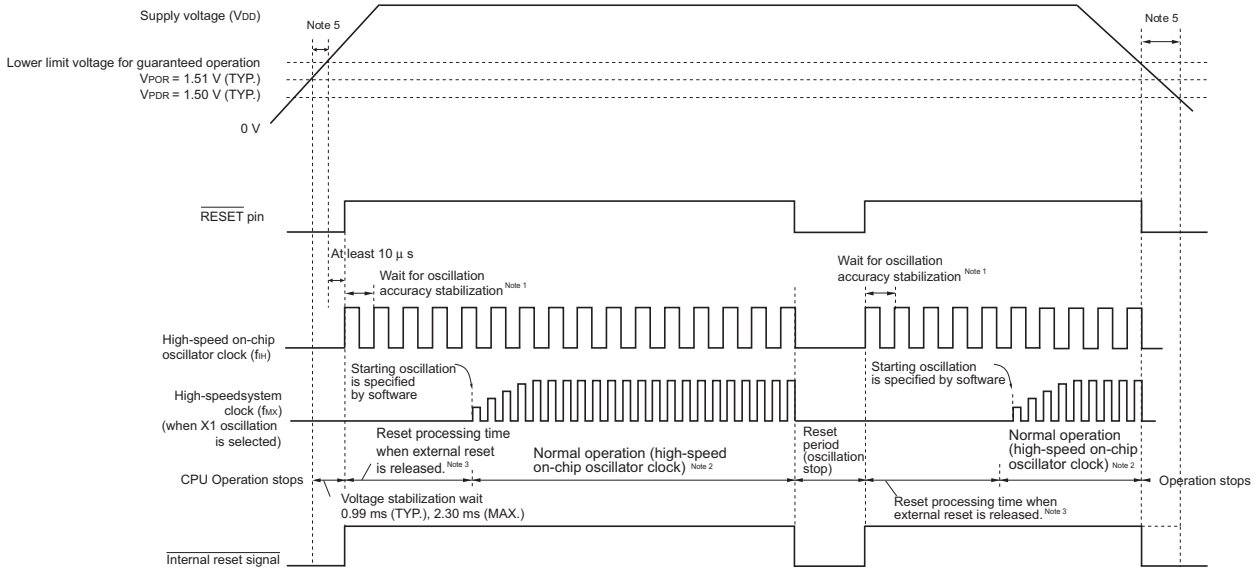


21.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown below.

Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)

(1) When the externally input reset signal on the $\overline{\text{RESET}}$ pin is used



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 or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

3. The time until normal operation starts includes the following reset processing time when the external reset is released (in the first reset processing following release from the POR state) after the $\overline{\text{RESET}}$ signal is driven high (1) as well as the voltage stabilization wait time after V_{POR} (1.51 V, typ.) is reached.
The reset processing time when the external reset is released is shown below.
In the first reset processing following release from the POR state:
0.672 ms (TYP.), 0.832 ms (MAX.) (when the LVD is in use)
0.399 ms (TYP.), 0.519 ms (MAX.) (when the LVD is off)

4. The reset processing time when the external reset is released in the second reset processing following release from the POR state is shown below.
In the second reset processing following release from the POR state:
0.531 ms (TYP.), 0.675 ms (MAX.) (when the LVD is in use)
0.259 ms (TYP.), 0.362 ms (MAX.) (when the LVD is off)

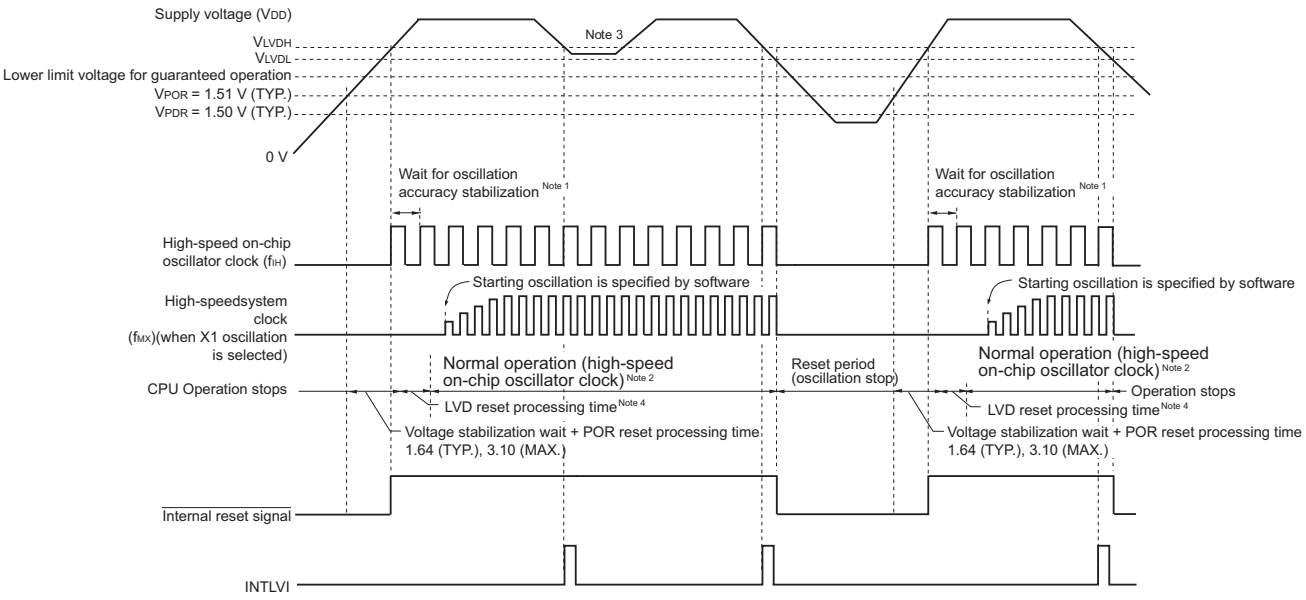
5. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **30.4** or **31.4 AC Characteristics**. This is done by controlling the externally input reset signal. After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detector or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Caution For power-on reset, be sure to use the external reset signal input through the $\overline{\text{RESET}}$ pin when the LVD is off. For details, see CHAPTER 22 VOLTAGE DETECTOR.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)

(2) LVD interrupt & reset mode (option byte 000C1: LVIMDS1, LVIMDS0 = 1, 0)



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 or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

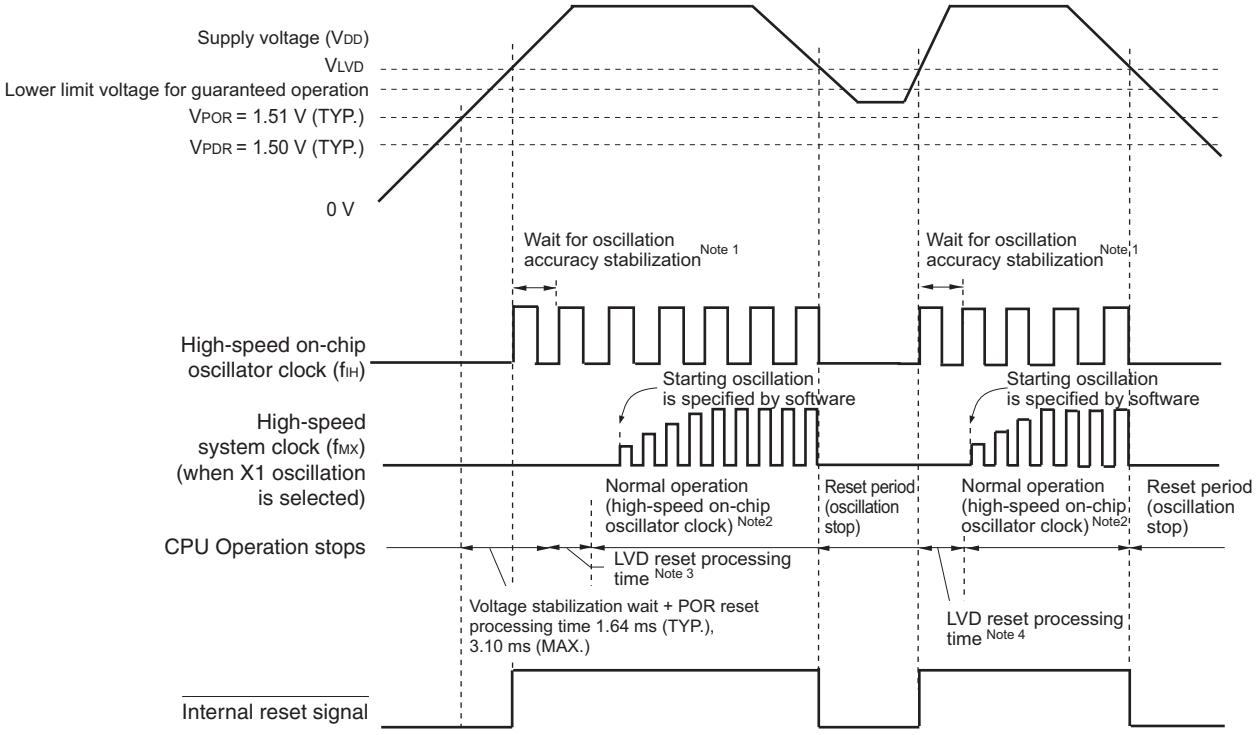
3. After the interrupt request signal (INTLVI) is generated, the LVILV and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 22-8 Setting Procedure for Operation Voltage Check/Reset** and **Figure 22-9 Setting Procedure for Initial Setting of Interrupt and Reset Mode**, taking into consideration that the supply voltage might return to the high voltage detection level (V_{LVDH}) or higher without falling below the low voltage detection level (V_{LVDL}).

4. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVDH}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, typ.) is reached.
LVD reset processing time: 0 ms to 0.0701 ms (max.)

Remark V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (3/3)

(3) LVD reset mode (option byte 000C1H: LVIMDS1 = 1, LVIMDS0 = 1)



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 or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

3. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVD}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, typ.) is reached.
LVD reset processing time: 0 ms to 0.0701 ms (max.)

4. When the power supply voltage is below the lower limit for operation and the power supply voltage is then restored after an internal reset is generated only by the voltage detector (LVD), the following LVD reset processing time is required after the LVD detection level (V_{LVD}) is reached.
LVD reset processing time: 0.0511 ms (typ.), 0.0701 ms (max.)

Remarks 1. V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

2. The time until normal operation starts after power is turned on in the LVD interrupt mode (option byte 000C1H: LVIMD1 = 0, LVIMD0 = 1) is the same as the time specified in Note 3 of Figure 21-2 (3)

CHAPTER 22 VOLTAGE DETECTOR

22.1 Functions of Voltage Detector

The operation mode and detection voltages (V_{LVDH}, V_{LVDL}, V_{LVD}) for the voltage detector is set by using the option byte (000C1H).

The voltage detector (LVD) has the following functions.

- The LVD circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}), and generates an internal reset or internal interrupt signal.
- The detection level for the power supply detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) can be selected by using the option byte as one of 9 levels (For details, see **CHAPTER 25 OPTION BYTE**).
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **30.4** or **31.4 AC Characteristics**. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detector or controlling the externally input reset signal before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)
The two detection voltages (V_{LVDH}, V_{LVDL}) are selected by the option byte 000C1H. The high-voltage detection level (V_{LVDH}) is used for releasing resets and generating interrupts. The low-voltage detection level (V_{LVDL}) is used for generating resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for triggering and ending resets.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for generating interrupts/reset release.

The reset and internal interrupt signals are generated in each mode as follows.

Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an internal interrupt signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage drops, and an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal reset by detecting $V_{DD} \geq V_{LVDH}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an internal internal signal by detecting $V_{DD} < V_{LVD}$.	Retains the state of an internal reset by the LVD immediately after a reset until $V_{DD} \geq V_{LVD}$. Releases the LVD internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an interrupt request signal (INTLVI) by detecting $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ after the LVD internal reset is released.

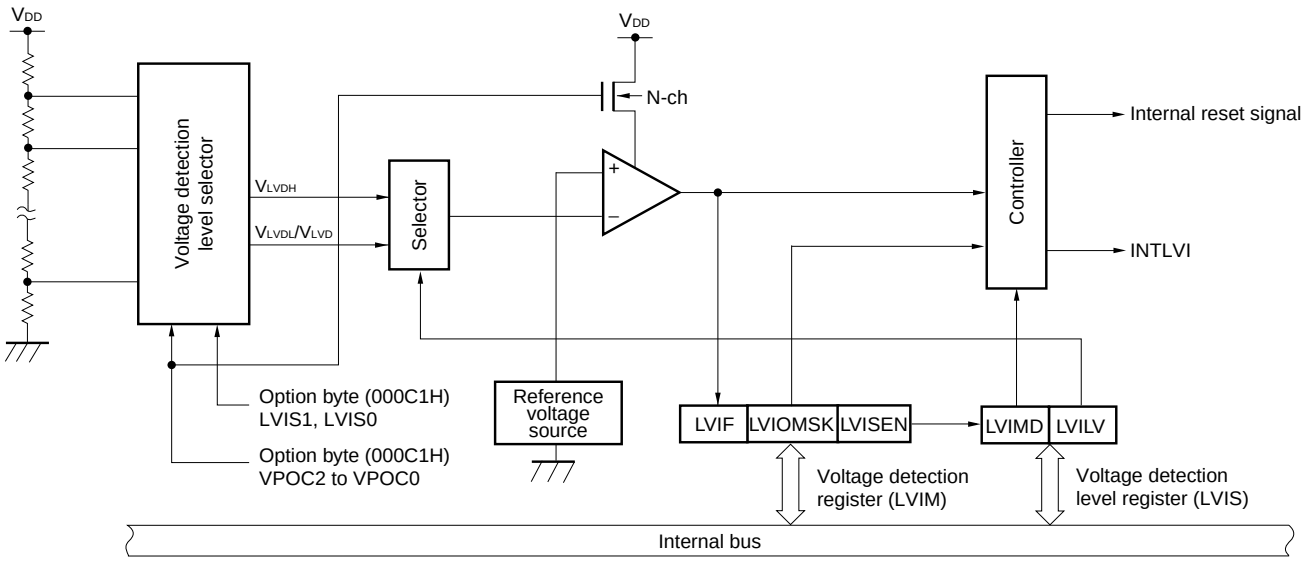
While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).

Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 20 RESET FUNCTION**.

22.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 22-1.

Figure 22-1. Block Diagram of Voltage Detector



22.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

22.3.1 Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the LVD output mask status.

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

Reset signal generation clears this register to 00H.

Figure 22-2. Format of Voltage Detection Register (LVIM)

Address: FFFA9H After reset: 00H Note 1 R/W Note 2

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVISEN Note 3	0	0	0	0	0	LVIOMSK	LVIF

LVISEN Note 3	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling of rewriting the LVIS register (LVIOMSK = 0 (Mask of LVD output is invalid)
1	Enabling of rewriting the LVIS register (LVIOMSK = 1 (Mask of LVD output is valid)

LVIOMSK	Mask status flag of LVD output
0	Mask of LVD output is invalid
1	Mask of LVD output is valid Note 4

LVIF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVD}), or when LVD is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVD})

Notes 1. The reset value changes depending on the reset source.
If the LVIS register is reset by LVD, it is not reset but holds the current value. In other reset, LVISEN is cleared to 0.

2. Bits 0 and 1 are read-only.

3. LVISEN and LVIOMSK can only be set in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not set them in other cases.

4. LVIOMSK bit is automatically set to “1” in the following periods and reset or interruption by LVD is masked.
- Period during LVISEN = 1
- Waiting period from the time when LVD interrupt is generated until LVD detection voltage becomes stable
- Waiting period from the time when the value of LVILV bit changes until LVD detection voltage becomes stable

22.3.2 Voltage detection level register (LVIS)

This register selects the voltage detection level.

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

Reset signal generation input sets this register to 00H/01H/81H[Note 1].

Figure 22-3. Format of Voltage Detection Level Select Register (LVIS)

Address: FFFAAH After reset: 00H/01H/81H[Note 1] R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD	0	0	0	0	0	0	LVILV

LVIMD[Note 2]	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVILV[Note 2]	LVD detection level
0	High-voltage detection level (V_{LVDH})
1	Low-voltage detection level (V_{LVDL} or V_{LVDL})

Notes 1. The reset value changes depending on the reset source and the setting of the option byte.
This register is not cleared (00H) by LVD reset.
The generation of reset signal other than an LVD reset sets as follows.
- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

2. Writing “0” can only be allowed in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not set LVIMD and LVILV in other cases. The value is switched automatically when reset or interrupt is generated in the interrupt & reset mode.

Cautions 1. Rewrite the value of the LVIS register according to Figure 22-8.

2. Specify the LVD operation mode and detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) of each mode by using the option byte 000C1H/010C1H. Figure 22-4 shows the format of the user option byte (000C1H/010C1H). For details about the option byte, see CHAPTER 25 OPTION BYTE.

Figure 22-4. Format of User Option Byte (000C1H/010C1H) (1/2)

Address: 000C1H/010C1H [Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt & reset mode)

Detection voltage			Option Byte Setting Value in LVD Interrupt & Reset Mode						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
2.61 V	2.55 V	2.45 V	0	1	0	1	0	1	0
2.71 V	2.65 V					0	1		
3.75 V	3.67 V					0	0		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
–			Setting of values other than above is prohibited						

- LVD setting (reset mode)

Detection voltage		Option Byte Setting Value in LVD Reset Mode						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.50 V	2.45 V	0	1	0	1	1	1	1
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Cautions 1. Be sure to set bit 4 to 1.

2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 30.4 or 31.4 AC Characteristics. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detector or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

Remark The detection voltage is a TYP. value. For details, see **30.6.4** or **31.6.4 LVD circuit characteristics**.

Figure 22-4. Format of User Option Byte (000C1H/010C1H) (2/2)

Address: 000C1H/010C1H [Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt mode)

Detection voltage		Option Byte Setting Value in LVD Interrupt Mode						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.50 V	2.45 V	0	1	0	1	1	0	1
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited						

- LVD off (use of external reset input via $\overline{\text{RESET}}$ pin)

Detection voltage		Option Byte Setting Value when LVD is Off (External Reset is Used)						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
–	–	1	×	×	×	×	×	1
–		Setting of values other than above is prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Cautions 1 Be sure to set bit 4 to 1.

2 After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 30.4 or 31.4 AC Characteristics. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detector or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

Remarks 1. ×: don't care

2. The detection voltage is a TYP. value. For details, see **30.6.4** or **31.6.4 LVD circuit characteristics**.

22.4 Operation of Voltage Detector

22.4.1 When used as reset mode

Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

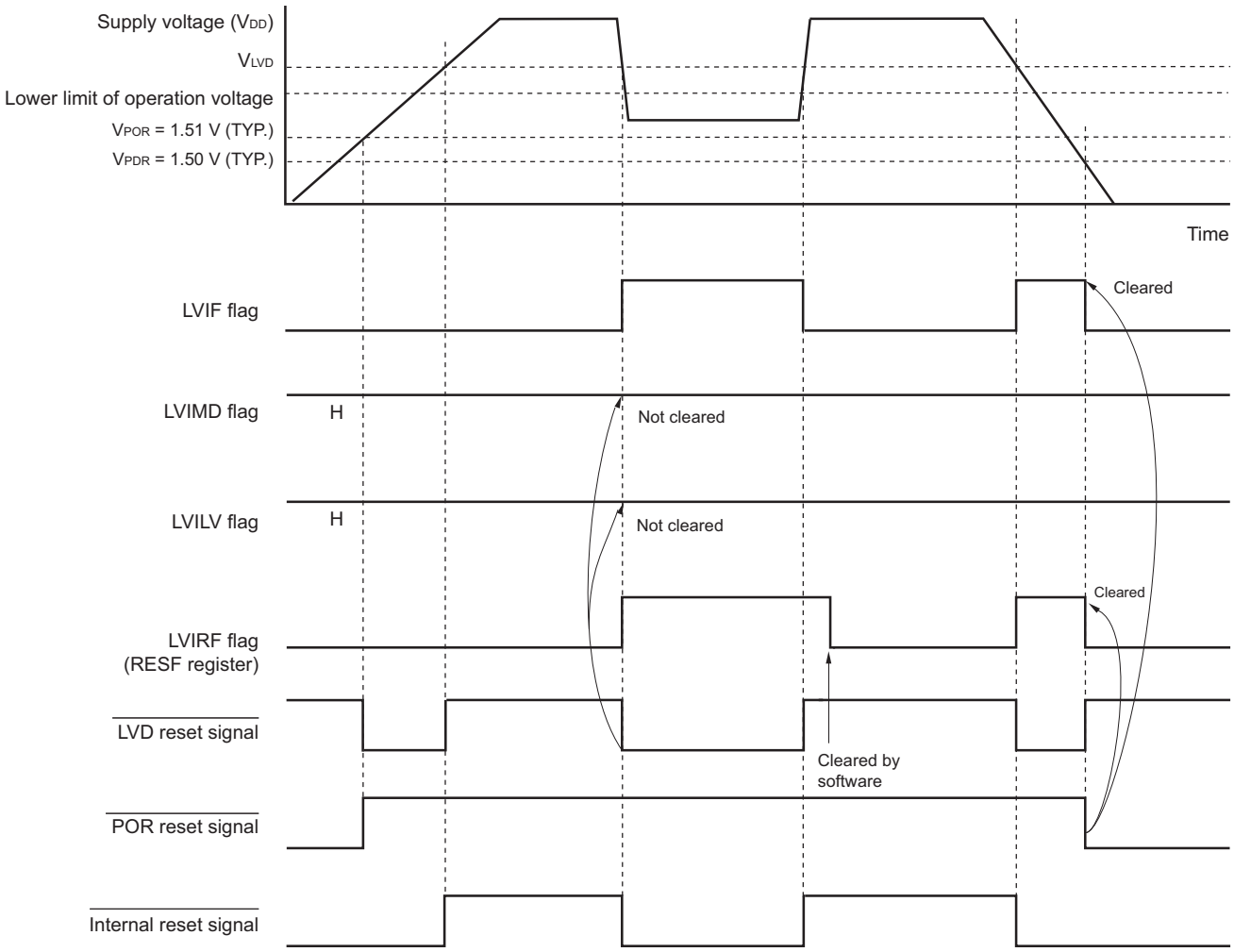
The operation is started in the following initial setting state when the reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 81H.
  Bit 7 (LVIMD) is 1 (reset mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

- Operation in LVD reset mode
  In the reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1), the state of an internal reset by LVD is held until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) at power on. The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  At the drop of the operating voltage, an internal reset by LVD is generated when the supply voltage (V_{DD}) drops below the voltage detection level (V_{LVD}).

  Figure 22-5 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 22-5. Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

22.4.2 When used as interrupt mode

Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

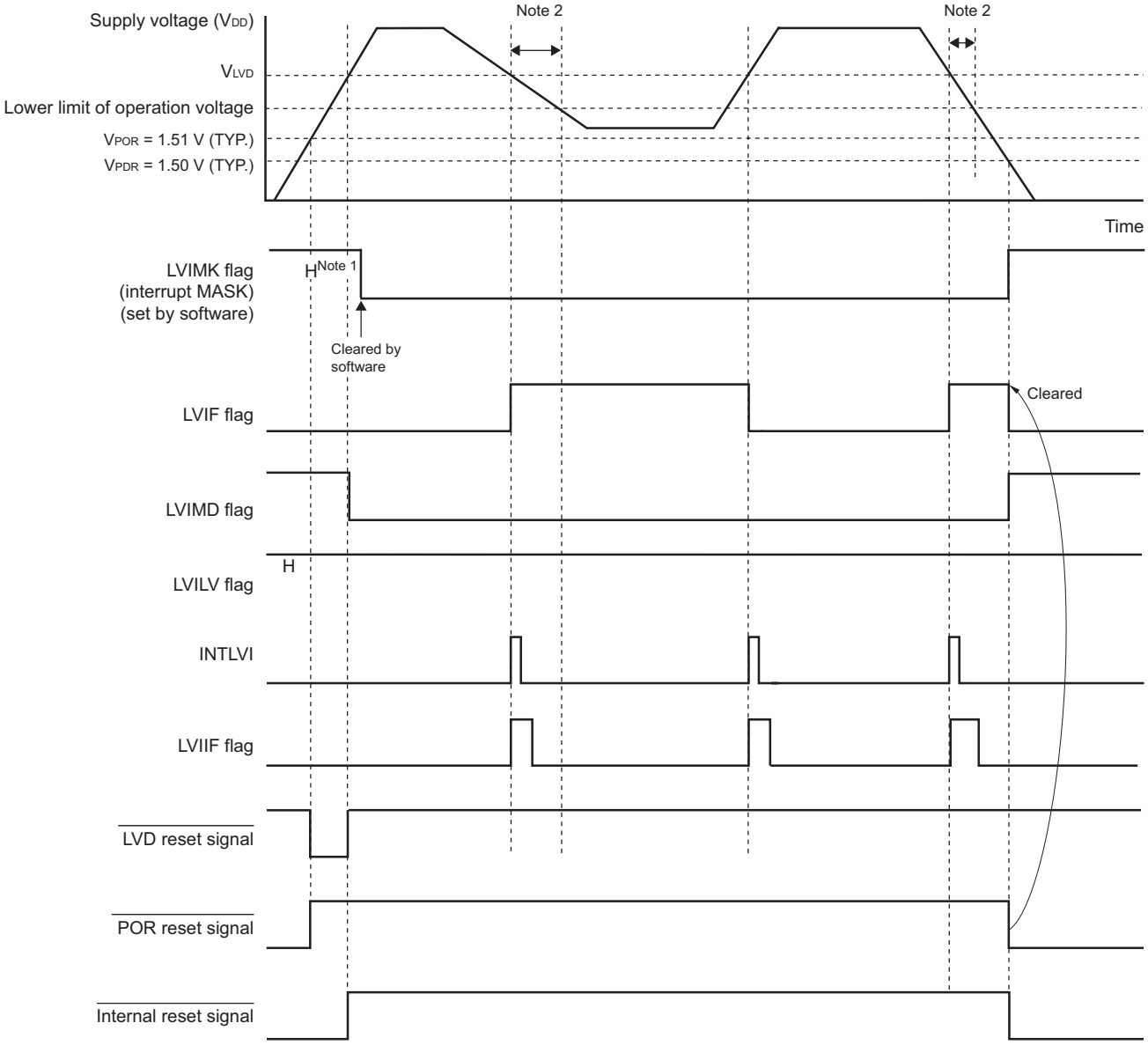
The operation is started in the following initial setting state when the interrupt mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 01H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

- Operation in LVD interrupt mode
  In the interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1), the state of an internal reset by LVD is retained immediately after a reset until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  After the LVD internal reset is released, an interrupt request signal (INTLVI) by the LVD is generated when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **30.4** or **31.4 AC characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

  Figure 22-6 shows the timing of the internal interrupt signal generated in the LVD interrupt mode.

Figure 22-6. Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. At the drop of the voltage, set the STOP mode or put the LVD into a reset state by the external reset pin before the voltage drops below its range described in **30.4** or **31.4 AC characteristics**. Confirm the supply voltage falls within the operating voltage range before resuming the operation of LVD.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

22.4.3 When used as interrupt & reset mode

Specify the operation mode (the interrupt & reset (LVIMDS1, LVIMDS0 = 1, 0)) and the detection voltage (V_{LVDH}, V_{LVDL}) by using the option byte 000C1H.

The operation is started in the following initial setting state when the interrupt & reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 00H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 0 (high-voltage detection level: V_{LVDH}).

- Operation in LVD interrupt & reset mode

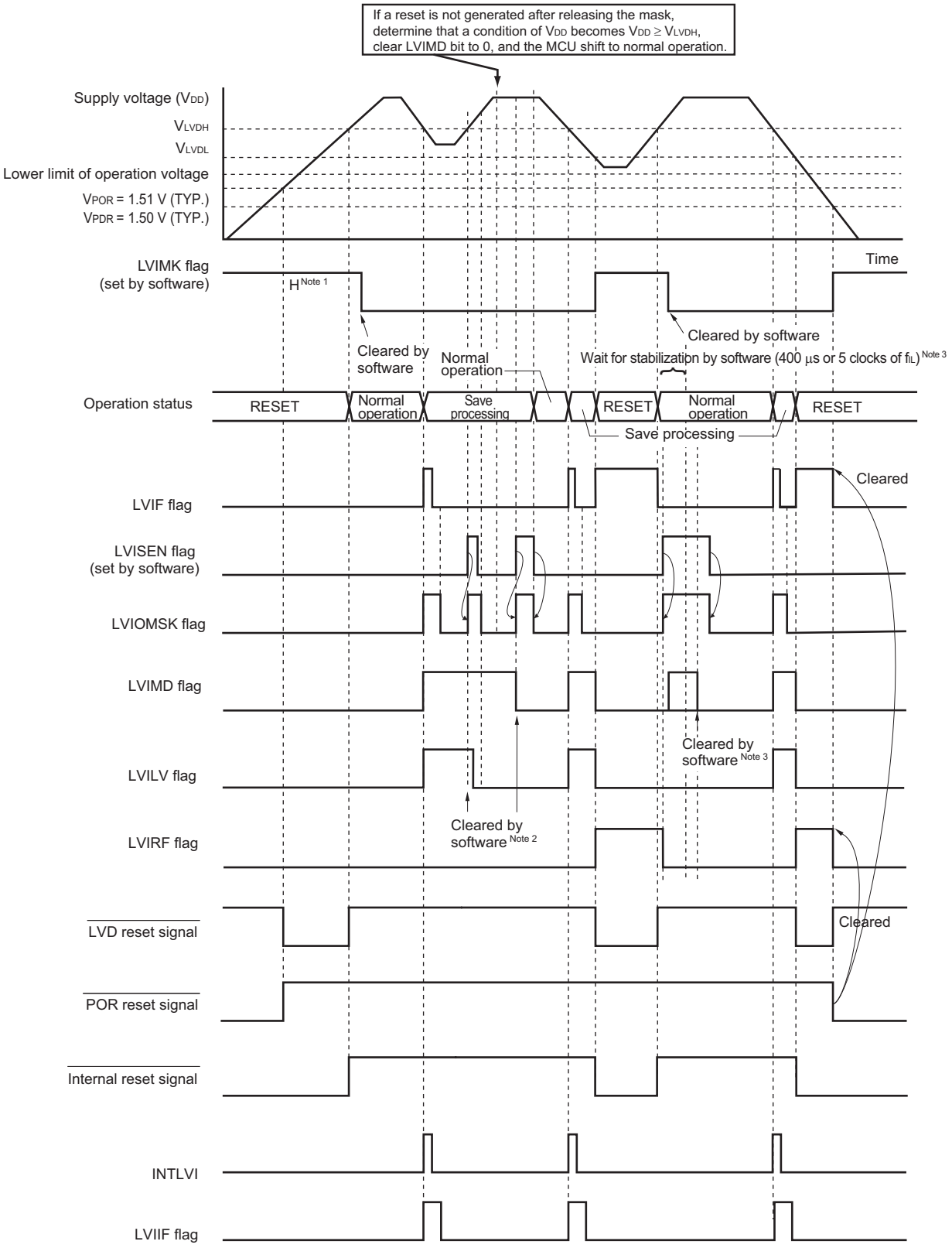
  In the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0), the state of an internal reset by LVD is held until the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}) at power on. The internal reset is released when the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}).

  At operating voltage drop, an internal reset signal by LVD (INTLVI) is generated and arbitrary save processing is performed when the supply voltage (V_{DD}) drops below the high-voltage detection level (V_{LVDH}). After that, an internal reset by LVD is generated when the supply voltage (V_{DD}) drops below the low-voltage detection level (V_{LVDL}). After INTLVI is generated, an internal interrupt signal is not generated even if the supply voltage becomes equal to or higher than the high-voltage detection voltage (V_{LVDH}) without dropping below the low-voltage detection voltage (V_{LVDL}).

  To use the LVD reset & interrupt mode, perform the processing according to **Figure 22-8 Setting Procedure for Operation Voltage Check/Reset**.

  Figure 22-7 shows the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

Figure 22-7. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



(**Notes** and **Remark** are listed on the next page.)

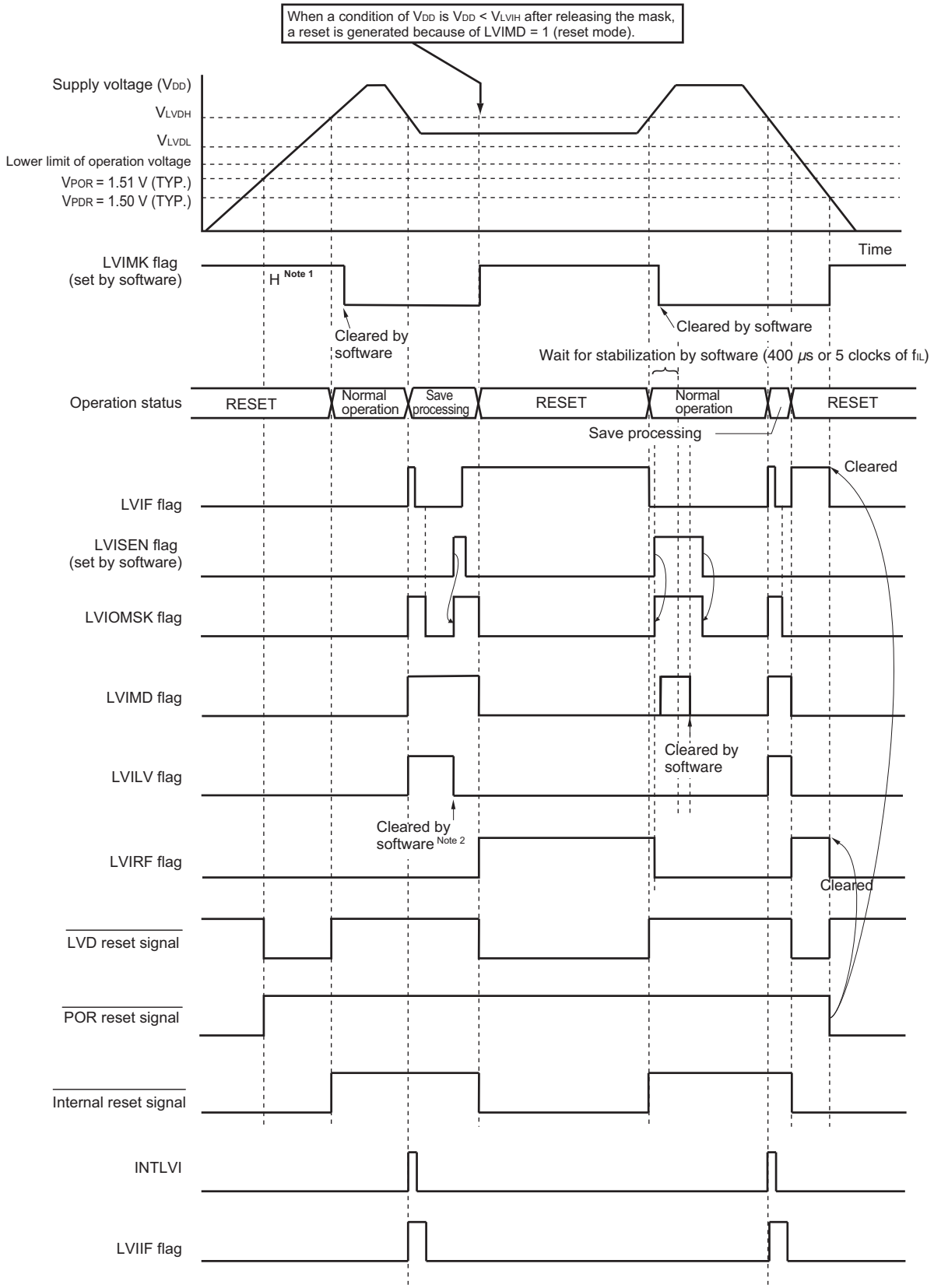
Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 22-8 Setting Procedure for Operation Voltage Check/Reset**.

Remark V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

Figure 22-7. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



(**Notes** and **Remark** are listed on the next page.)

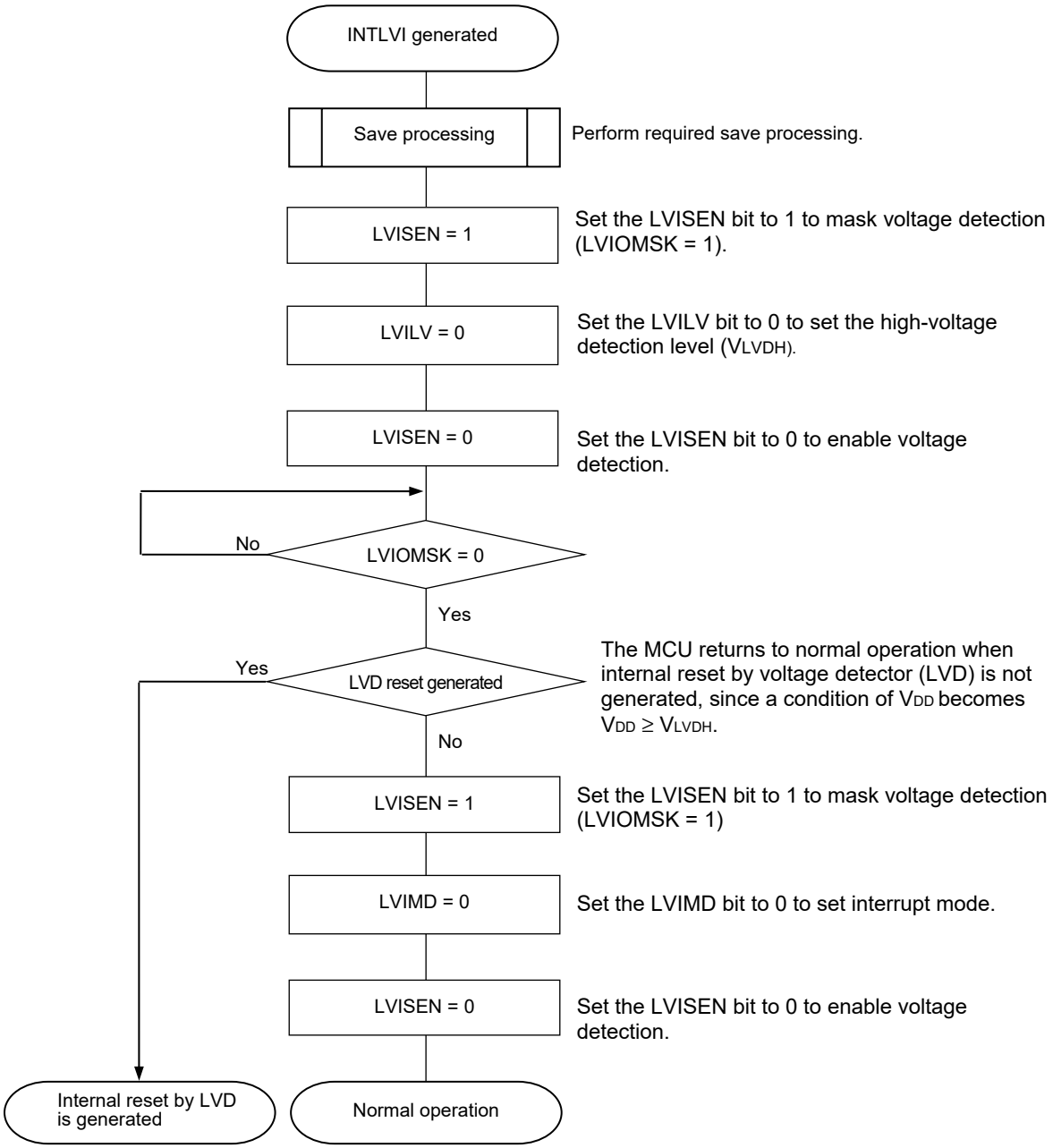
Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. After an interrupt is generated, perform the processing according to **Figure 22-8 Setting Procedure for Operation Voltage Check/Reset**.

Remark V_{POR}: POR power supply rise detection voltage

V_{PDR}: POR power supply fall detection voltage

Figure 22-8. Setting Procedure for Operation Voltage Check/Reset



22.5 Cautions for Voltage Detector

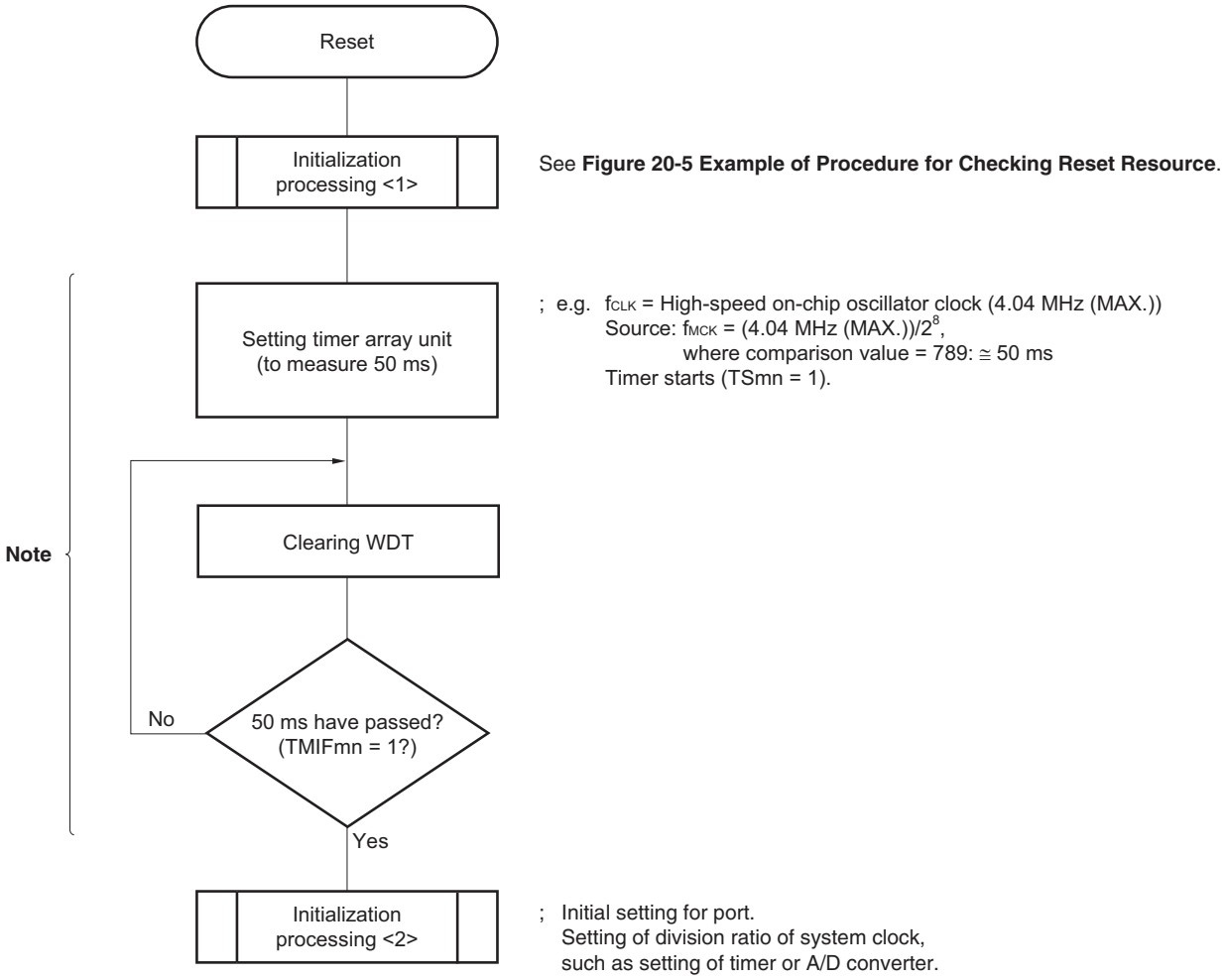
(1) Voltage fluctuation when power is supplied

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVD detection voltage, 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 software counter that uses a timer, and then initialize the ports.

Figure 22-9. Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of LVD Detection Voltage



Note If reset is generated again during this period, initialization processing <2> is not started.

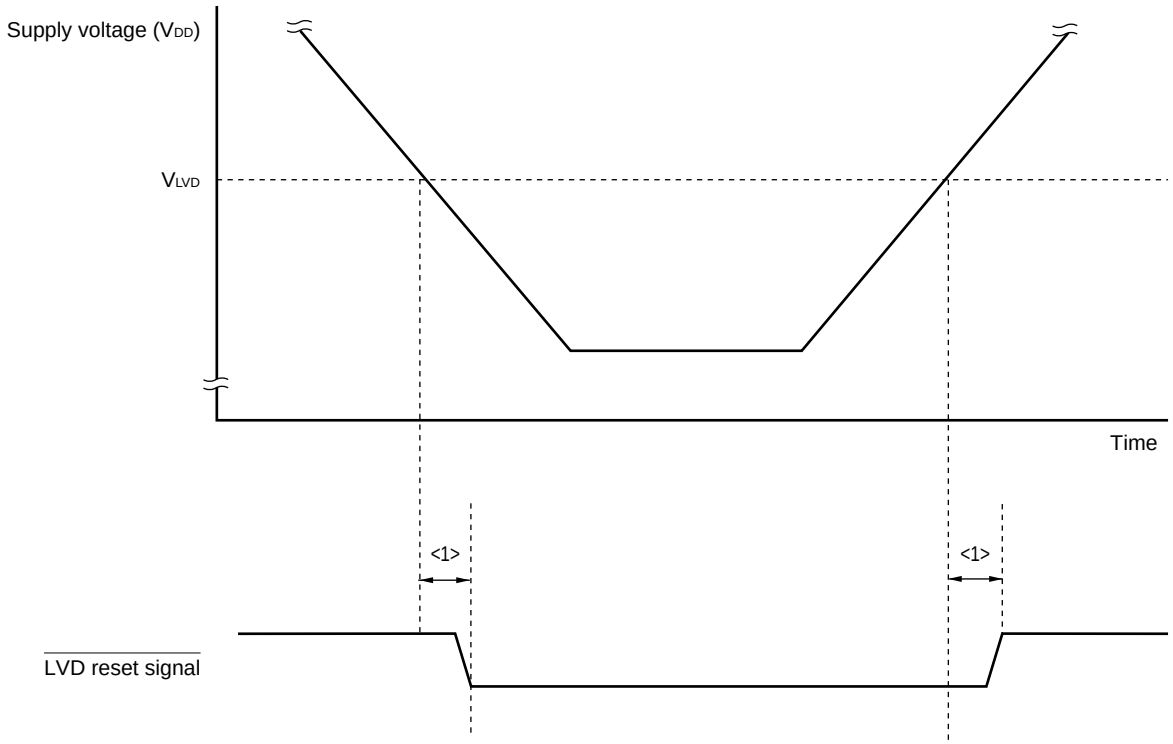
Remark m = 0, n = 0 to 3

(2) Delay from the time LVD reset source is generated until the time LVD reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVD detection voltage (V_{LVD}) until the time LVD reset has been generated.

In the same way, there is also some delay from the time LVD detection voltage (V_{LVD}) ≤ supply voltage (V_{DD}) until the time LVD reset has been released (see **Figure 22-10**).

Figure 22-10. Delay from the time LVD reset source is generated until the time LVD reset has been generated or released



<1>: Detection delay (300 μs (MAX.))

(3) Power on when LVD is off

Use the external rest input via the $\overline{\text{RESET}}$ pin when the LVD is off.

For an external reset, 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, turn power on, continue to input a low level to the pin for 10 μs or more within the operating voltage range shown in **30.4** or **31.4 AC Characteristics**, and then input a high level to the pin.

(4) Operating voltage drop when LVD is off or set to interrupt mode

At operating voltage drop when LVD is off or set to interrupt mode, shift to the STOP mode or put the LVD into a reset state by an external reset pin before the voltage drops below its range described in **30.4** and **31.4 AC characteristics**. Confirm the supply voltage falls within the operating voltage range before resuming the operation of the LVD.

CHAPTER 23 SAFETY FUNCTIONS

23.1 Overview of Safety Functions

<R> The following safety functions are provided in the RL78/G1C to comply with the IEC60730 safety standards.

These functions enable the microcontroller to self-diagnose abnormalities and stop operating if an abnormality is detected.

(1) Flash memory CRC operation function (high-speed CRC, general-purpose CRC)

This detects data errors in the flash memory by performing CRC operations.

Two CRC functions are provided in the RL78/G1C that can be used according to the application or purpose of use.

- High-speed CRC: The CPU can be stopped and a high-speed check executed on its entire code flash memory area during the initialization routine.
- General CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) RAM parity error detection function

This detects parity errors when the RAM is read as data.

(3) RAM guard function

This prevents RAM data from being rewritten when the CPU freezes.

(4) SFR guard function

This prevents SFRs from being rewritten when the CPU freezes.

(5) Invalid memory access detection function

This detects illegal accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

(6) Frequency detection function

This function allows a self-check of the CPU/peripheral hardware clock frequencies using the timer array unit.

(7) A/D test function

This is used to perform a self-check of the A/D converter by performing A/D conversion of the A/D converter's positive and negative reference voltages, analog input channel (ANI), temperature sensor output, and internal reference voltage.

(8) Digital output signal level detection function for I/O ports

When the I/O ports are output mode in which PMmn bit of the port mode register (PMm) is 0, the output level of the pin can be read.

Remarks 1. m = 0 to 7, 12, 14, n = 0 to 7

2. For usage examples of the safety functions complying with the IEC60730 safety standards, refer to the RL78 MCU series IEC60730/60335 application notes (R01AN0749, R01AN1062, R01AN1296).

23.2 Registers Used by Safety Functions

The safety functions use the following registers for each function.

Register	Each Function of Safety Function
• Flash memory CRC control register (CRC0CTL) • Flash memory CRC operation result register (PGCRCL)	Flash memory CRC operation function (high-speed CRC)
• CRC input register (CRCIN) • CRC data register (CRCD)	CRC operation function (general-purpose CRC)
• RAM parity error control register (RPECTL)	RAM parity error detection function
• Invalid memory access detection control register (IAWCTL)	RAM guard function
	SFR guard function
	Invalid memory access detection function
• Timer input select register 0 (TIS0)	Frequency detection function
• A/D test register (ADTES)	A/D test function
• Port mode select register (PMS)	Digital output signal level detection function for I/O ports

The content of each register is described in **23.3 Operation of Safety Functions**.

23.3 Operation of Safety Functions

23.3.1 Flash memory CRC operation function (high-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using CRC to do it. The high-speed CRC provided in the RL78/G1C can be used to check the entire code flash memory area during the initialization routine. The high-speed CRC can be executed only when the program is allocated on the RAM and in the HALT mode of the main system clock.

The high-speed CRC performs an operation by reading 32-bit data per clock from the flash memory while stopping the CPU. This function therefore can finish a check in a shorter time. (For example, it only takes 340 μs (1/24 MHz x (32 KB/4 x 1024 – 4 bytes/4)) to check a 32 KB flash memory using a 24 MHz clock.)

The CRC generator polynomial used complies with “$X^{16} + X^{12} + X^{5} + 1$” of CRC-16-CCITT.

The high-speed CRC operates in MSB first order from bit 31 to bit 0.

Caution The CRC operation result might differ during on-chip debugging because the monitor program is allocated.

Remark The operation result is different between the high-speed CRC and the general CRC, because the general CRC operates in LSB first order.

23.3.1.1 Flash memory CRC control register (CRC0CTL)

This register is used to control the operation of the high-speed CRC ALU, as well as to specify the operation range.

The CRC0CTL register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 23-1. Format of Flash Memory CRC Control Register (CRC0CTL)

Address: F02F0H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Control of CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	0	0	0	0	0	00000H to 3FFBH (16 K to 4 bytes)
0	0	0	0	0	1	00000H to 7FFBH (32 K to 4 bytes)
Other than the above						Setting prohibited

Remark Input the expected CRC operation result value to be used for comparison in the lowest 4 bytes of the flash memory. Note that the operation range will thereby be reduced by 4 bytes.

23.3.1.2 Flash memory CRC operation result register (PGCRCL)

This register is used to store the high-speed CRC operation results.

The PGCRCL register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 23-2. Format of Flash Memory CRC Operation Result Register (PGCRCL)

Address: F02F2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8
	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

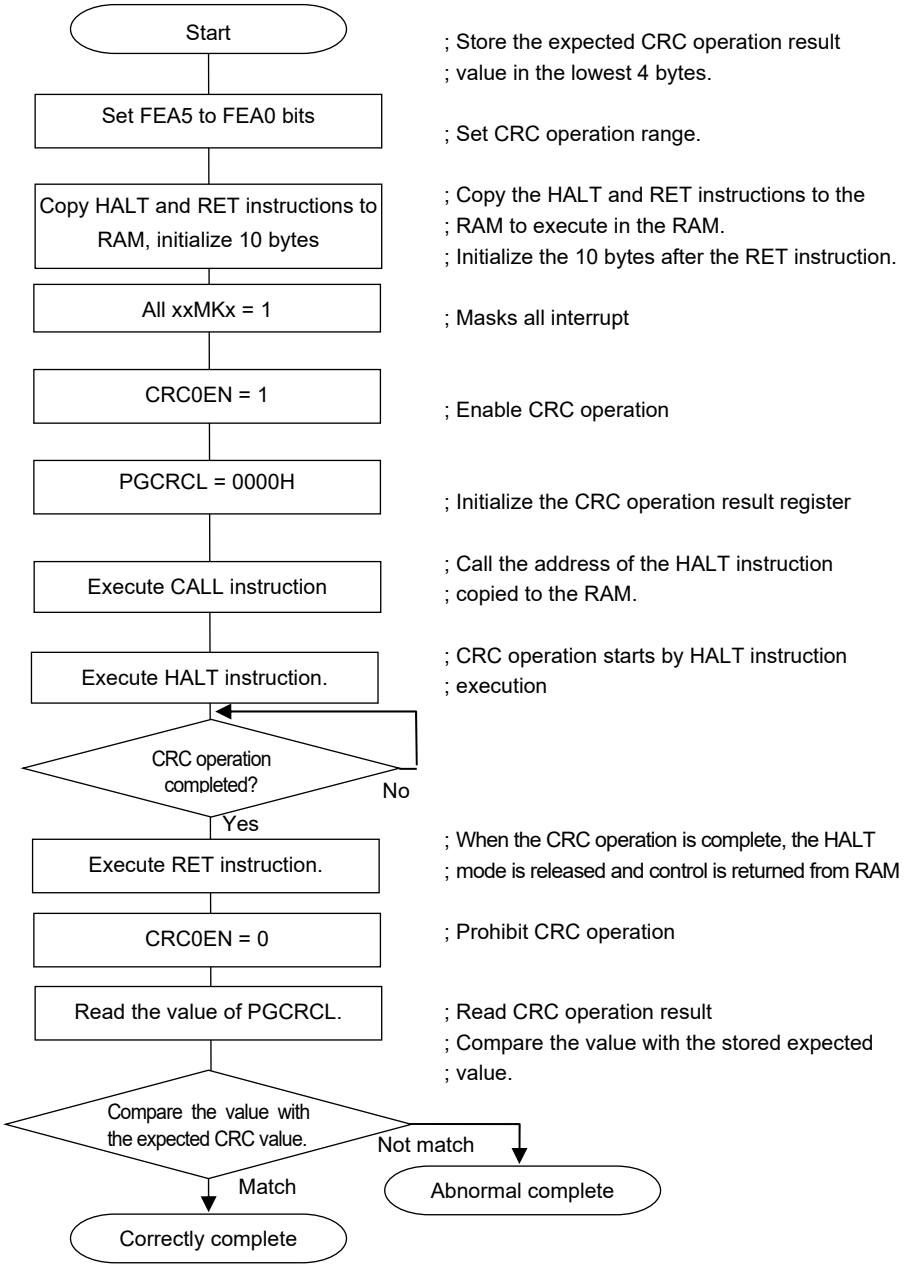
PGCRC15 to 0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 23-3 shows the flowchart of flash memory CRC operation function (high-speed CRC).

<Operation flow>

Figure 23-3. Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Cautions 1. The CRC operation is executed only on the code flash.

2. Store the expected CRC operation value in the area below the operation range in the code flash.

3. The CRC operation is enabled by executing the HALT instruction in the RAM area.
Be sure to execute the HALT instruction in RAM area.

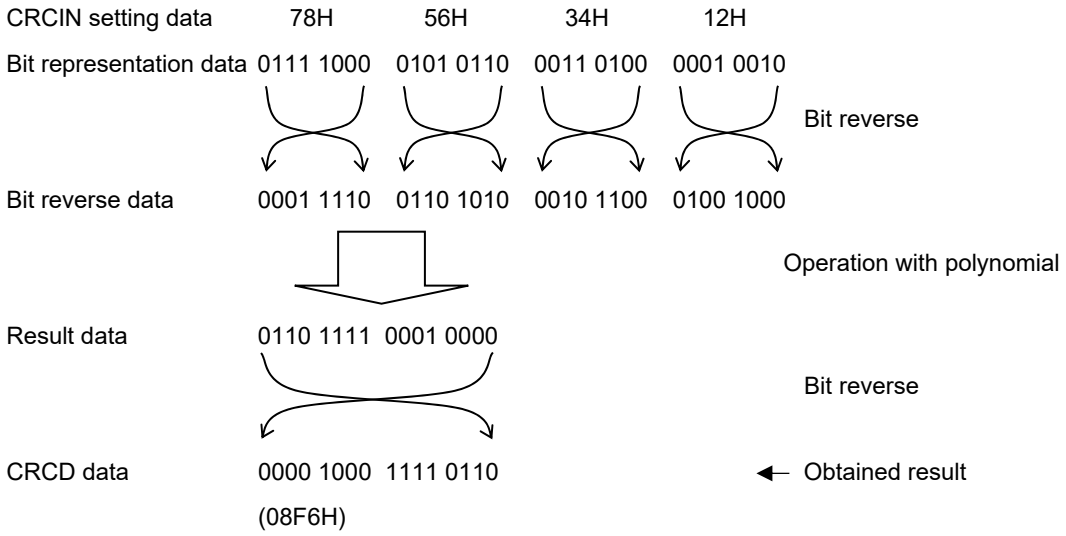
The expected CRC value can be calculated by using the Integrated Development Environment CubeSuite+. See the Integrated Development Environment CubeSuite+ user's manual for details.

23.3.2 CRC operation function (general-purpose CRC)

<R> In the RL78/G1C, a general CRC operation can be executed as a peripheral function while the CPU is operating. The general CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program). CRC calculation function in the HALT mode can be used only during the DMA transmission.

The general CRC operation can be executed in the main system clock operation mode as well as the subsystem clock operation mode.

The CRC generator polynomial used is "$X^{16} + X^{12} + X^5 + 1$" of CRC-16-CCITT. The data to be input is inverted in bit order and then calculated to allow for LSB-first communication. For example, if the data 12345678H is sent from the LSB, values are written to the CRCIN register in the order of 78H, 56H, 34H, and 12H, enabling a value of 08F6H to be obtained from the CRCD register. This is the result obtained by executing a CRC operation on the bit rows shown below, which consist of the data 12345678H inverted in bit order.



Caution Because the debugger rewrites the software break setting line to a break instruction during program execution, the CRC operation result differs if a software break is set in the CRC operation target area.

23.3.2.1 CRC input register (CRCIN)

CRCIN register is an 8-bit register that is used to set the CRC operation data of general-purpose CRC.

The possible setting range is 00H to FFH.

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

Reset signal generation clears this register to 00H.

Figure 23-4. Format of CRC Input Register (CRCIN)

Address: FFFACH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN								

Bits 7 to 0	Function
00H to FFH	Data input.

23.3.2.2 CRC data register (CRCD)

This register is used to store the CRC operation result of the general-purpose CRC.

The setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.

The CRCD register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 23-5. Format of CRC Data Register (CRCD)

Address: F02FAH After reset: 0000H R/W

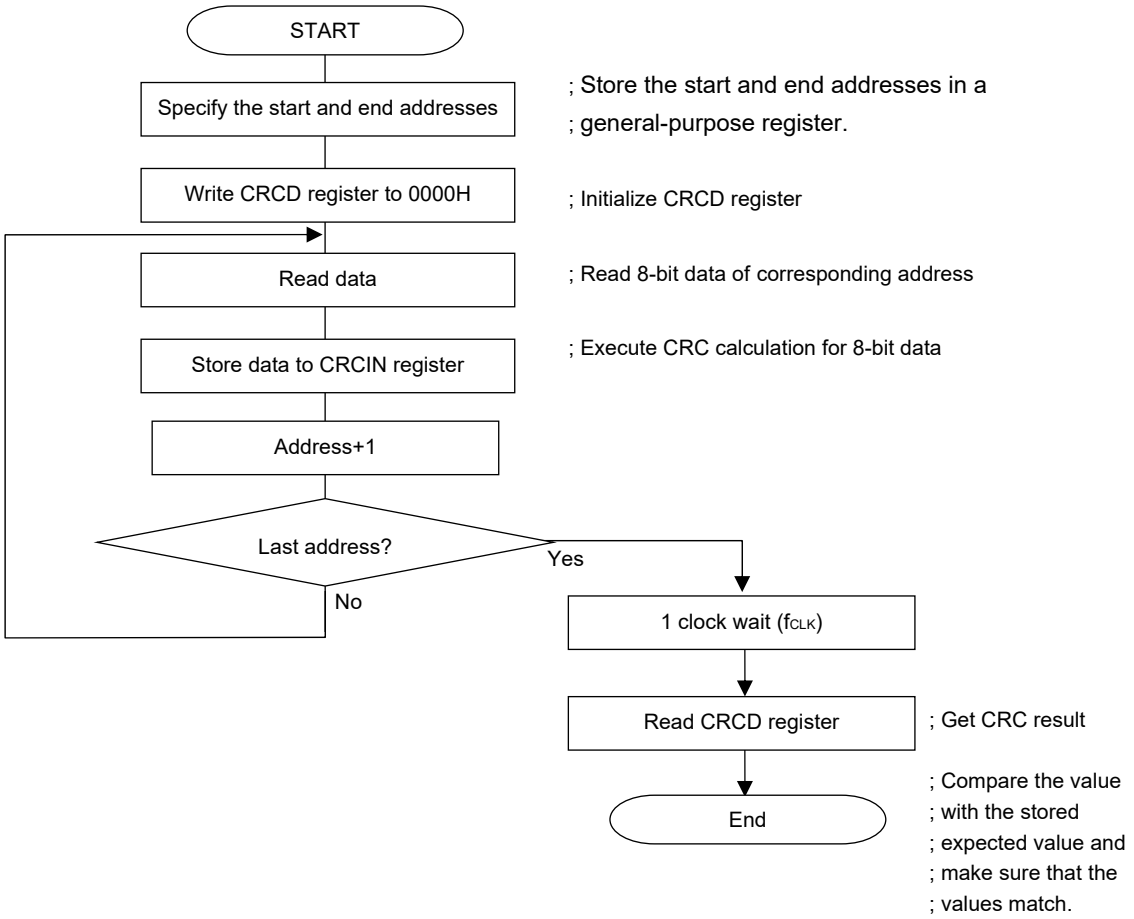
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Cautions 1. Read the value written to CRCD register before writing to CRCIN register.

2. If conflict between writing and storing operation result to CRCD register occurs, the writing is ignored.

<Operation flow>

Figure 23-6. CRC Operation Function (General-Purpose CRC)



23.3.3 RAM parity error detection function

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in the RL78/G1C's RAM. By using this RAM parity error detection function, the parity bit is appended when data is written, and the parity is checked when the data is read. This function can also be used to trigger a reset when a parity error occurs.

23.3.3.1 RAM parity error control register (RPECTL)

This register is used to control parity error generation check bit and reset generation due to parity errors.

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

Reset signal generation clears this register to 00H.

Figure 23-7. Format of RAM Parity Error Control Register (RPECTL)

Address: F00F5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution The parity bit is appended when data is written, and the parity is checked when the data is read. Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed before reading data.

The RL78's CPU executes look-ahead due to the pipeline operation, the CPU might read an uninitialized RAM area that is allocated beyond the RAM used, which causes a RAM parity error. Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area + 10 bytes when instructions are fetched from RAM areas. When using the self-programming function while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area to overwrite + 10 bytes before overwriting.

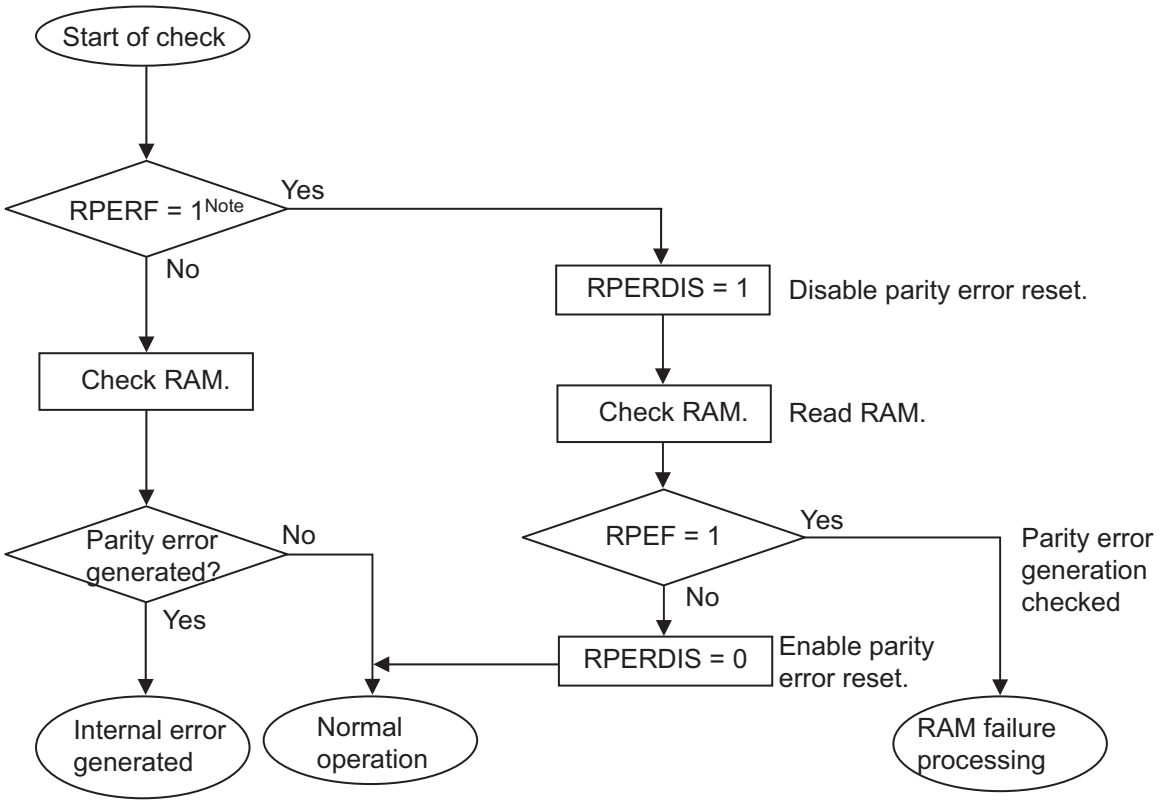
Remarks 1. The parity error reset is enabled by default (RPERDIS = 0).

2. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set (1) if a parity error occurs. If parity error resets are enabled (RPERDIS = 0) with RPEF set to 1, a parity error reset is generated when the RPERDIS bit is cleared to 0.

3. The RPEF flag in the RESF register is set (1) by RAM parity errors and cleared (0) by writing 0 to it or by any reset source. When RPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.

4. The general registers are not included for RAM parity error detection.

Figure 23-8. RAM Parity Check Flow



Note To check internal reset status using a RAM parity error, see **CHAPTER 20 RESET FUNCTION**.

23.3.4 RAM guard function

<R> This RAM guard function is used to protect data in the specified memory space.

If the RAM guard function is specified, writing to the specified RAM space is disabled, but reading from the space can be carried out as usual.

23.3.4.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GRAM1 and GRAM0 bits are used in RAM guard function.

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

Reset signal generation clears this register to 00H.

Figure 23-9. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0077H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard space
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes starting at the start RAM address
1	0	The 256 bytes starting at the start RAM address
1	1	The 512 bytes starting at the start RAM address

23.3.5 SFR guard function

<R> This SFR guard function is used to protect data in the control registers used by the port function, interrupt function, clock control function, voltage detection function, and RAM parity error detection function.

If the SFR guard function is specified, writing to the specified SFRs is disabled, but reading from the SFRs can be carried out as usual.

23.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GPORT, GINT and GCSC bits are used in SFR guard function.

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

Reset signal generation clears this register to 00H.

Figure 23-10. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0077H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Control registers of port function guard
0	Disabled. Control registers of port function can be read or written to.
1	Enabled. Writing to control registers of port function is disabled. Reading is enabled. [Guarded SFR] PMxx, PUxx, PIMxx, POMxx, PMCxx, ADPC, PIOR Note 1

GINT	Registers of interrupt function guard
0	Disabled. Registers of interrupt function can be read or written to.
1	Enabled. Writing to registers of interrupt function is disabled. Reading is enabled. [Guarded SFR] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC Notes 2	Control registers of clock control function, voltage detector and RAM parity error detection function guard
0	Disabled. Control registers of clock control function, voltage detector and RAM parity error detection function can be read or written to.
1	Enabled. Writing to control registers of clock control function, voltage detector and RAM parity error detection function is disabled. Reading is enabled. [Guarded SFR] CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, RPECTL, DSCCTL

Notes 1. Pxx (Port register) is not guarded.

2. Clear GCSC bit to 0, during self programming /serial programming.

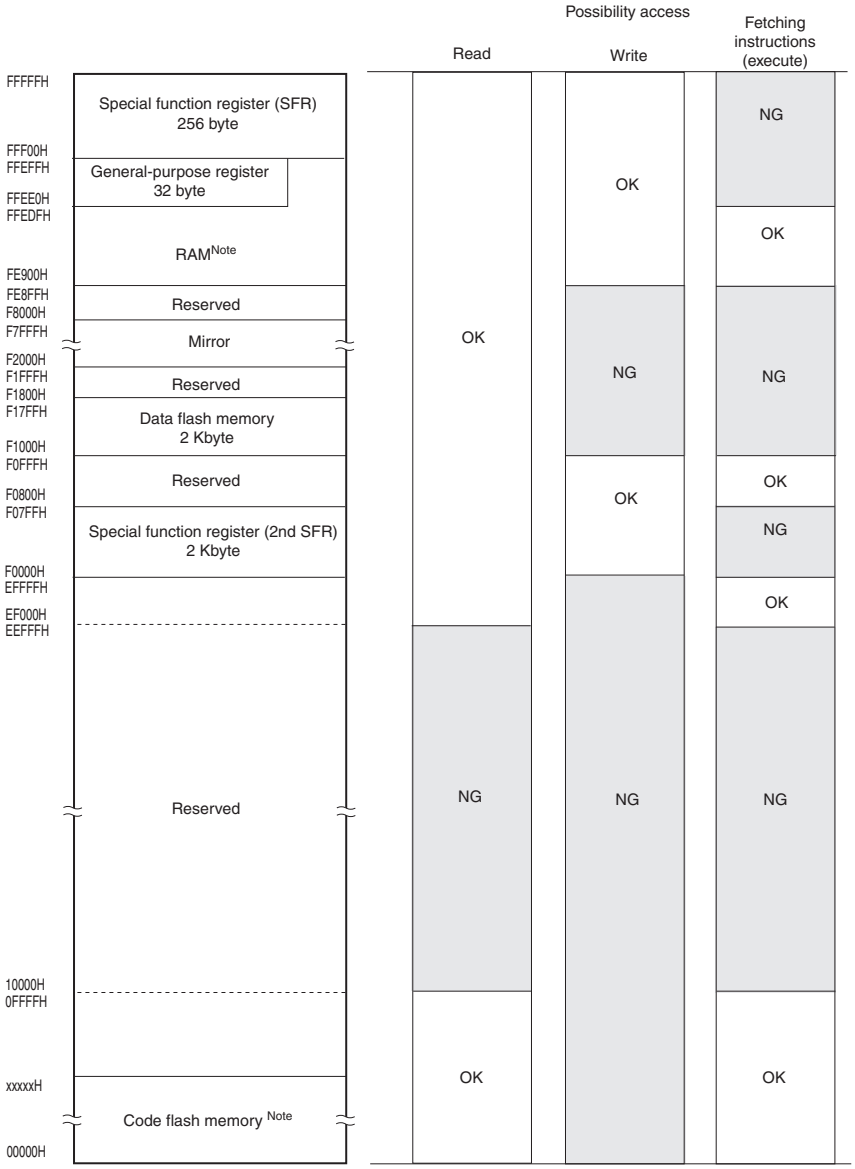
23.3.6 Invalid memory access detection function

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly.

The illegal memory access detection function triggers a reset if a memory space specified as access-prohibited is accessed.

The illegal memory access detection function applies to the areas indicated by NG in Figure 23-11.

Figure 23-11. Invalid access detection area



Note Code flash memory and RAM address of each product are as follows.

Products	Code flash memory (00000H to xxxxxH)	RAM (FE900H to FFEFFH)	Detected lowest address for read/instruction fetch (execution) (yyyyyH)
R5F10JBxxxx (xxxx = CANA, CAFP, CGNA, CGFP), R5F10KBxxxx (xxxx = CANA, CAFP, CGNA, CGFP), R5F10JGxxxx (xxxx = CANA, CAFB, CGNA, CGFB), R5F10KGxxxx (xxxx = CANA, CAFB, CGNA, CGFB)	32768 × 8 bits (00000H to 07FFFH)	5632 × 8 bits (FE900H to FFEFFH)	1000H

23.3.6.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

IAWEN bit is used in invalid memory access detection function.

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

Reset signal generation clears this register to 00H.

Figure 23-12. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0077H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN Note	Control of invalid memory access detection
0	Disable the detection of invalid memory access.
1	Enable the detection of invalid memory access.

Note Only writing 1 to the IAWEN bit is enabled, not writing 0 to it after setting it to 1.

Remark By specifying WDTON = 1 for the option byte (watchdog timer operation enable), the invalid memory access function is enabled even IAWEN = 0.

23.3.7 Frequency detection function

The IEC60730 standard mandates checking that the oscillation frequency is correct.

By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 1 of the timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clocks to be compared>

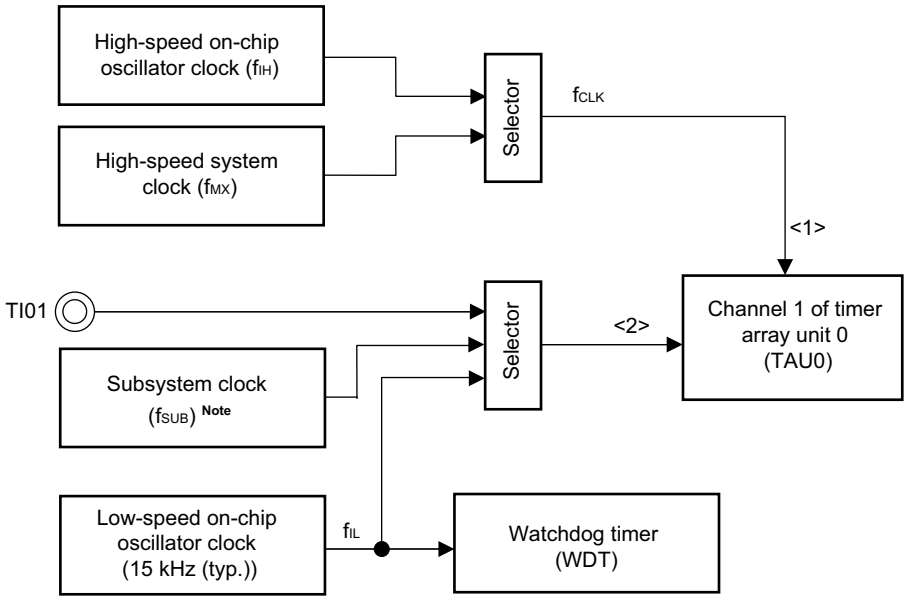
<1> CPU/peripheral hardware clock frequency (f_{CLK}):

- High-speed on-chip oscillator clock (f_{IH})
- High-speed system clock (f_{MX})

<2> Input to channel 1 of the timer array unit

- Timer input to channel 1 (TI01)
- Low-speed on-chip oscillator clock (f_{IL}: 15 kHz (typ.))
- Subsystem clock (f_{SUB}) **Note**

Figure 23-13. Configuration of Frequency Detection Function



If input pulse interval measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal.

For how to execute input pulse interval measurement, see **6.8.4 Operation as input pulse interval measurement**.

Note Can only be selected in the products incorporating the subsystem clock.

23.3.7.1 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input of channel 1 of the timer array unit 0 (TAU0).

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 23-14. Format of Timer Input Select Register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	
0	1	0	
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than the above			Setting prohibited

23.3.8 A/D test function

The IEC60730 standard mandates testing the A/D converter. The A/D test function checks whether or not the A/D converter is operating normally by executing A/D conversions of the A/D converter's positive and negative reference voltages, analog input channel (ANI), temperature sensor output voltage, and the internal reference voltage. For details of the check method, see the safety function (A/D test) application note (R01AN0955).

The analog multiplexer can be checked using the following procedure.

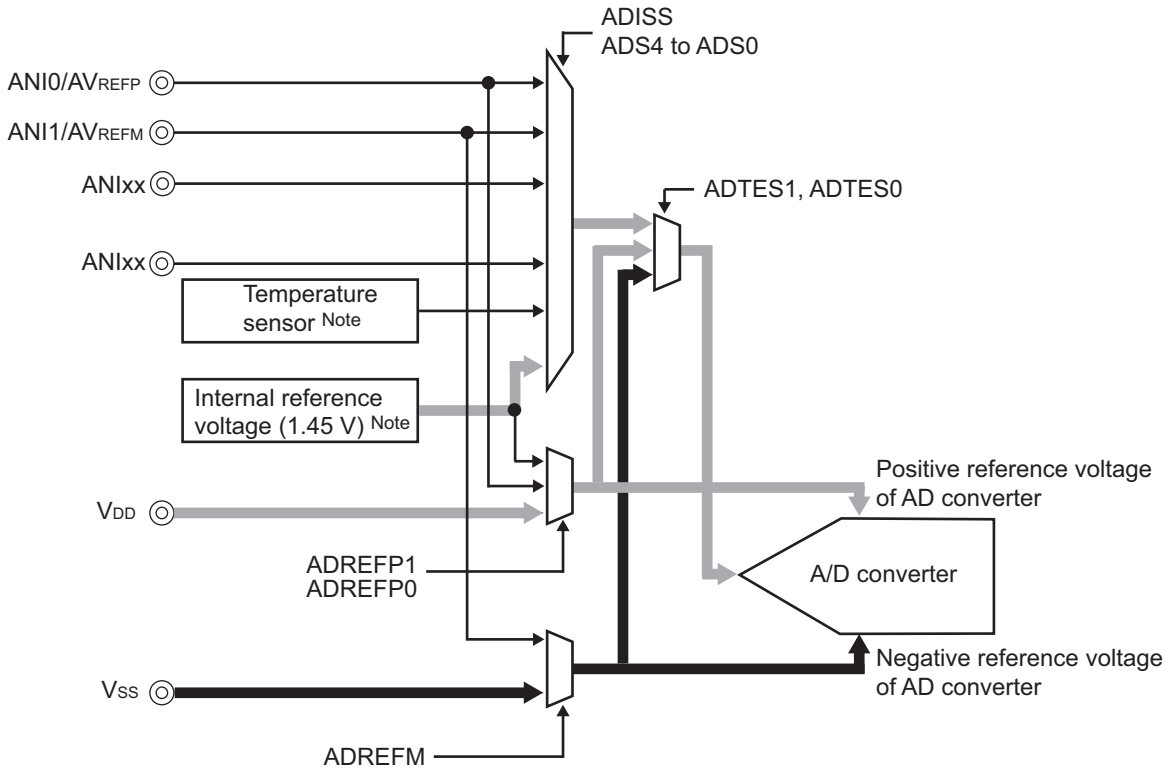
<1> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<2> Perform A/D conversion for the ANIx pin (conversion result 1-1).
<3> Select the A/D converter's negative reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 0)
<4> Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
<5> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<6> Perform A/D conversion for the ANIx pin (conversion result 1-2).
<7> Select the A/D converter's positive reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 1)
<8> Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
<9> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).
<10> Perform A/D conversion for the ANIx pin (conversion result 1-3).
<11> Check that the conversion results 1-1, 1-2, and 1-3 are equal.
<12> Check that the A/D conversion result 2-1 is all zero and conversion result 2-2 is all one.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Remarks 1. If the analog input voltage is variable during A/D conversion in steps <1> to <10> above, use another method to check the analog multiplexer.

2. The conversion results might contain an error. Consider an appropriate level of error when comparing the conversion results.

Figure 23-15. Configuration of A/D Test Function



Note This setting can be used only in HS (high-speed main) mode.

23.3.8.1 A/D test register (ADTES)

This register is used to select the A/D converter's positive reference voltage, A/D converter's negative reference voltage, analog input channel (ANIxx), temperature sensor output voltage, or internal reference voltage (1.45 V) as the target of A/D conversion.

When using the A/D test function, specify the following settings:

- Select negative reference voltage as the target of A/D conversion for zero-scale measurement.
- Select positive reference voltage as the target of A/D conversion for full-scale measurement.

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

Reset signal generation clears this register to 00H.

Figure 23-16. 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	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output voltage Note/internal reference voltage (1.45 V) Note (This is specified using the analog input channel specification register (ADS).)
1	0	Negative reference voltage (selected with the ADREFM bit in ADM2)
1	1	Positive reference voltage (selected with the ADREFP1 or ADREFP0 bit in ADM2)
Other than the above		Setting prohibited

Note Temperature sensor output voltage and internal reference voltage (1.45 V) can be used only in HS (high-speed main) mode.

23.3.8.2 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

Set A/D test register (ADTES) to 00H when measuring the ANIxx/temperature sensor output voltage/internal reference voltage (1.45 V).

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 23-17. 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	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AV_{REFP} pin
0	0	0	0	0	1	ANI1	P21/ANI1/AV_{REFM} pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin Note 1
0	0	0	1	1	0	ANI6	P26/ANI6 pin Note 1
0	0	0	1	1	1	ANI7	P27/ANI7 pin Note 1
0	1	0	0	0	0	ANI16	P01/ANI16 pin Note 2
0	1	0	0	0	1	ANI17	P00/ANI17 pin Note 2
0	1	0	0	1	1	ANI19	P120/ANI19 pin
1	0	0	0	0	0	–	Temperature sensor output Note 3
1	0	0	0	0	1	–	Internal reference voltage output (1.45 V) Note3
Other than the above						Setting prohibited	

Notes 1. 48-pin products only

2. 32-pin products only

3. This setting can be used only in HS (high-speed main) mode.

Cautions 1. Be sure to clear bits 5 and 6 to 0.

2. Select input mode for the ports which are set to analog input with the ADPC and PMC registers, using the port mode registers 0, 2, and 12 (PM0, PM2, and PM12).

3. Do not use the ADS register to set the pins which should be set as digital I/O with the A/D port configuration register (ADPC).

4. Do not use the ADS register to set the pins which should be set as digital I/O with the port mode control registers 0, and 12 (PMC0, and PMC12).

5. Only rewrite the value of the ADISS bit while conversion operation is stopped (ADCS = 0, ADCE = 0).

6. If using AV_{REFP} as the positive reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.

7. If using AV_{REFM} as the negative reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.

Cautions 8. When ADISS is 1, the internal reference voltage (1.45 V) cannot be used for the positive reference voltage. In addition, the first conversion result obtained after setting ADISS to 1 is not available. For detailed setting flow, see 11.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode).

9. If a transition is made to STOP mode or a transition is made to HALT mode during CPU operation with subsystem clock, do not set ADISS to 1. When ADISS is 1, the A/D converter reference voltage current (I_{ADREF}) shown in 30.3.2 or 31.3.2 Supply current characteristics is added.

10. The temperature sensor cannot be used for the A/D converter when bit 0 (VDDUSBE), the internal USB power supply control bit, is "1".

23.3.9 Digital output signal level detection function for I/O ports

In the IEC60730, it is required to check that the I/O function correctly operates.

By using the digital output signal level detection function for I/O ports, the digital output level of the pin can be read when the port is set to output mode (the PMmn bit in the port mode register (PMm) is 0).

23.3.9.1 Port mode select register (PMS)

This register is used to select the output level from output latch level or pin output level when the port is output mode in which PMm bit of port mode register (PMm) is 0.

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

Reset signal generation clears these registers to 00H.

Figure 23-18. Format of Port Mode Select Register (PMS)

Address: F0018H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	Method for selecting output level to be read when port is output mode (PMmn = 0)
0	Pmn register value is read.
1	Digital output level of the pin is read.

Remark m = 0 to 7, 12, 14
n = 0 to 7

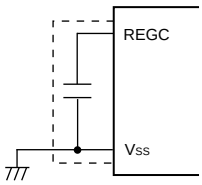
Cautions 1. When setting the PMS0 bit to 1 and rewriting the port register (Pn register n = 0 to 7, 12 to 14), use an 8-bit memory manipulation instruction only.

2. When using P60 to P63 as general-purpose ports, the output level of these pins cannot be read by setting PMS0. However, only when the IICA0EN bit in the PER0 register is set to 1, the output level of P60 and P61 can be read by setting the PMS0 bit.

CHAPTER 24 REGULATOR

24.1 Regulator Overview

The RL78/G1C contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

The regulator output voltage, see **Table 24-1**.

Table 24-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
High-speed main mode	1.8 V	In STOP mode
		When the high-speed system clock (f_{MX}), the high-speed on-chip oscillator clock (f_{HOCO}), and the PLL (f_{PLL}) are all stopped during CPU operation with the subsystem clock (f_{SUB}).
		When the high-speed system clock (f_{MX}) and the high-speed on-chip oscillator clock (f_{HOCO}), and the PLL (f_{PLL}) are all stopped during HALT mode when the CPU operation with the subsystem clock (f_{SUB}) has been set.
	2.1 V	Other than above (include during OCD mode) **Note**

Note When it shifts to the subsystem clock operation or STOP mode during the on-chip debugging, the regulator output voltage is kept at 2.1 V (not decline to 1.8 V).

CHAPTER 25 OPTION BYTE

25.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/G1C 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.

For the bits to which no function is allocated, do not change their initial values.

To use the boot swap operation during self programming, 000C0H to 000C3H are replaced by 010C0H to 010C3H. Therefore, set the same values as 000C0H to 000C3H to 010C0H to 010C3H.

Caution The option bytes should always be set regardless of whether each function is used.

25.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

(1) 000C0H/010C0H

- ○ Operation of watchdog timer
  - Enabling or disabling of counter operation
  - Enabling or disabling of counter operation in the HALT or STOP mode
- ○ Setting of interval time of watchdog timer
- ○ Setting of window open period of watchdog timer
- ○ Setting of interval interrupt of watchdog timer
  - Whether or not to use the interval interrupt is selectable.

Caution Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

(2) 000C1H/010C1H

- ○ Setting of LVD operation mode
  - Interrupt & reset mode.
  - Reset mode.
  - Interrupt mode.
  - LVD off (by controlling the externally input reset signal on the $\overline{\text{RESET}}$ pin)
- ○ Setting of LVD detection level (V_{LVDH}, V_{LVDL}, V_{LVD})

Cautions 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 30.4 or 31.4 AC Characteristics. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detector or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

2. Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

(3) 000C2H/010C2H

○ Setting of flash operation mode

Make the setting depending on the main system clock frequency (f_{MAIN}) and power supply voltage (V_{DD}) to be used.

- HS (high speed main) mode

○ Setting of the frequency of the high-speed on-chip oscillator

- Select from 48 MHz and 24 MHz (TYP.).

Caution Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

25.1.2 On-chip debug option byte (000C3H/ 010C3H)

○ Control of on-chip debug operation

- On-chip debug operation is disabled or enabled.

○ Handling of data of flash memory in case of failure in on-chip debug security ID authentication

- Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication.

Caution Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

25.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 25-1. Format of User Option Byte (000C0H/010C0H)

Address: 000C0H/010C0H[Note 1]

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINIT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2f_{IL} of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[Note 2]
0	0	Setting prohibited
0	1	50%
1	0	75%
1	1	100%

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	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode[Note 2]
1	Counter operation enabled in HALT/STOP mode

Notes 1. Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

2. The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 25-2. Format of User Option Byte (000C1H/010C1H) (1/2)

Address: 000C1H/010C1H [Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt & reset mode)

Detection voltage			Option byte setting value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
2.61 V	2.55 V	2.45 V	0	1	0	1	0	1	0
2.71 V	2.65 V					0	1		
3.75 V	3.67 V					0	0		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
–			Setting of values other than above is prohibited.						

- LVD setting (reset mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.50 V	2.45 V	0	1	0	1	1	1	1
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Remarks 1. For details on the LVD circuit, see **CHAPTER 22 VOLTAGE DETECTOR.**

2. The detection voltage is a typical value. For details, see **30.6.4** or **31.6.4 LVD circuit characteristics.**

(**Cautions** are listed on the next page.)

Figure 25-2. Format of User Option Byte (000C1H/010C1H) (2/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.50 V	2.45 V	0	1	0	1	1	0	1
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
–		Setting of values other than above is prohibited.						

- LVD off (by controlling the externally input reset signal on the $\overline{\text{RESET}}$ pin)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
–	–	1	×	×	×	×	×	1
–		Setting of values other than above is prohibited.						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Cautions 1. Be sure to set bit 4 to "1".

2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 30.4 or 31.4 AC Characteristics. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detector or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

Remarks 1. ×: don't care

2. For details on the LVD circuit, see **CHAPTER 21 VOLTAGE DETECTOR**.

3. The detection voltage is a typical value. For details, see **30.6.4** or **31.6.4 LVD circuit characteristics**.

Figure 25-3. Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H[Note 1]

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Setting of flash operation mode		
			Operating Frequency Range (f_{MAIN})	Operating Voltage Range (V_{DD})
1	1	HS (high speed main) mode	1 to 16 MHz	2.4 to 5.5 V
			1 to 24 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator	
					f_{HOCO} [Note 2]	f_{IH}
1	0	0	0	0	48 MHz	24 MHz
1	0	0	0	1	24 MHz	12 MHz
Other than above					Setting prohibited	

Notes 1. Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

2. When USB operation at a high-speed on-chip oscillator, should be set as shown in below
f_{HOCO} : 48 MHz (f_{IH} : 24 MHz).

Cautions 1. Be sure to set bit 5 to 1.

2. The ranges of operation frequency and operation voltage vary depending on the flash operation mode. For details, see 30.4 or 31.4 AC Characteristics.

25.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 25-4. Format of On-chip Debug Option Byte (000C3H/010C3H)

Address: 000C3H/010C3H[Note]

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Note Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

Caution **Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.**
Be sure to set 000010B to bits 6 to 1.

Remark The value on bits 3 to 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.

25.4 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        36H     ; Does not use interval interrupt of watchdog timer,
                        ; Enables watchdog timer operation,
                        ; Window open period of watchdog timer is 50%,
                        ; Overflow time of watchdog timer is 2^9/fIL,
                        ; Stops watchdog timer operation during HALT/STOP mode
      DB        7AH     ; Select 2.75 V for VLVDL
                        ; Select rising edge 2.92 V, falling edge 2.86 V for VLVDH
                        ; Select the interrupt & reset mode as the LVD operation mode
      DB        F0H     ; Select the HS (high speed main) mode as the flash operation mode
                          and 48 MHz as the frequency of the high-speed on-chip oscillator
      DB        85H     ; Enables on-chip debug operation, does not erase flash memory
                          data when security ID authorization fails
```

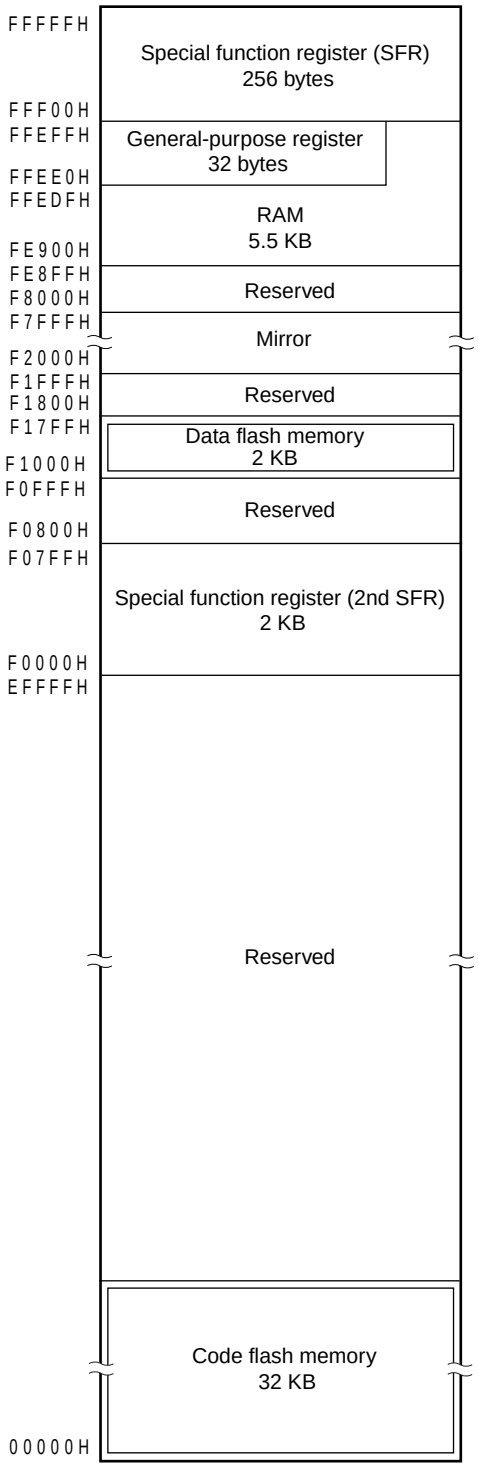
When the boot swap function is used during self programming, 000C0H to 000C3H is switched to 010C0H to 010C3H. Describe to 010C0H to 010C3H, therefore, the same values as 000C0H to 000C3H as follows.

```
OPT2  CSEG   AT   010C0H
      DB        36H     ; Does not use interval interrupt of watchdog timer,
                        ; Enables watchdog timer operation,
                        ; Window open period of watchdog timer is 50%,
                        ; Overflow time of watchdog timer is 2^9/fIL,
                        ; Stops watchdog timer operation during HALT/STOP mode
      DB        7AH     ; Select 2.75 V for VLVDL
                        ; Select rising edge 2.92 V, falling edge 2.86 V for VLVDH
                        ; Select the interrupt & reset mode as the LVD operation mode
      DB        F0H     ; Select the HS (high speed main) mode as the flash operation mode
                          and 48 MHz as the frequency of the high-speed on-chip oscillator
      DB        85H     ; Enables on-chip debug operation, does not erase flash memory
                          data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 010C0H to 010C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.

CHAPTER 26 FLASH MEMORY

The RL78 microcontroller incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board. The flash memory includes the “code flash memory”, in which programs can be executed, and the “data flash memory”, an area for storing data.



The following methods for programming the flash memory are available.

The code flash memory can be rewritten to through serial programming using a flash memory programmer or an external device (UART communication), or through self-programming.

- Serial programming using flash memory programmer (see **26.1**)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.

- Serial programming using external device (UART communication) (see **26.2**)
  Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).

- Self-programming (see **26.6**)
  The user application can execute self-programming of the code flash memory by using the flash self-programming library.

The data flash memory can be rewritten to by using the data flash library during user program execution (background operation). For access and writing to the data flash memory, see **26.8 Data Flash**.

26.1 Serial Programming 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, FL-PR5
- E1 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.

Remark FL-PR5 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 26-1. Wiring Between RL78/G1C and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.	
					32-pin	48-pin
Signal Name		I/O	Pin Function		LQFP (7x7), HWQFN (5x5)	LFQFP (7x7), HWQFN (7x7)
PG-FP5, FL-PR5	E1 on-chip debugging emulator					
–	TOOL0	I/O	Transmit/receive signal	TOOL0/ P40	1	39
SI/RxD	–	I/O	Transmit/receive signal			
–	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	2	40
/RESET	–	Output				
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	8	48
GND		–	Ground	V_{SS}	7	47
				REGC Note	6	46
FLMD1	EMV_{DD}	–	Driving power for TOOL0 pin	V_{DD}	8	48

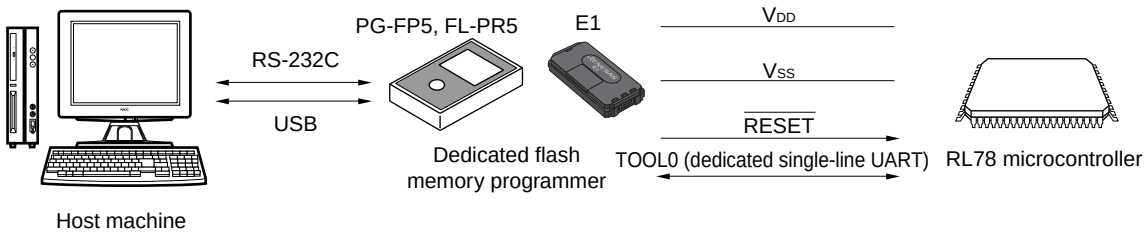
Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

26.1.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 26-1. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

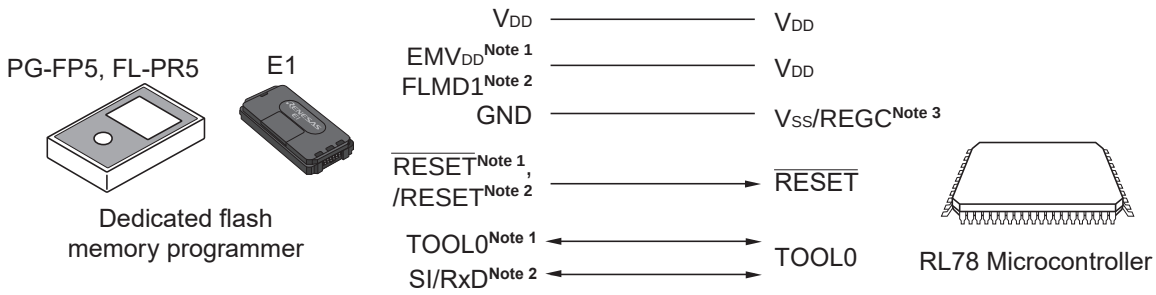
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

26.1.2 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: 1 M, 500 k, 250 k, 115.2 kbps

Figure 26-2. Communication with Dedicated Flash Memory Programmer



Notes 1. When using E1 on-chip debugging emulator.

2. When using PG-FP5 or FL-PR5.

3. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The dedicated flash memory programmer generates the following signals for the RL78 microcontroller. See the manual of PG-FP5, FL-PR5, or E1 on-chip debugging emulator for details.

Table 26-2. Pin Connection

Dedicated Flash Memory Programmer				RL78 microcontroller
Signal Name		I/O	Pin Function	Pin Name Note 2
PG-FP5, FL-PR5	E1 on-chip debugging emulator			
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		–	Ground	V_{SS}, REGC Note 1
FLMD1	EMV_{DD}	–	Driving power for TOOL pin	V_{DD}
/RESET	–	Output	Reset signal	$\overline{RESET}$
–	$\overline{RESET}$	Output		
–	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	–	I/O	Transmit/receive signal	

Notes 1. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

2. Pins to be connected differ with the product. For details, see **Table 26-1**.

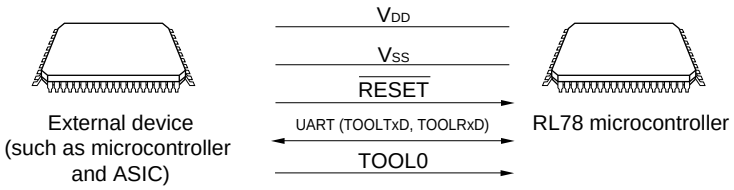
26.2 Serial Programming Using External Device (that Incorporates UART)

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.

26.2.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 26-3. Environment for Writing Program to Flash Memory



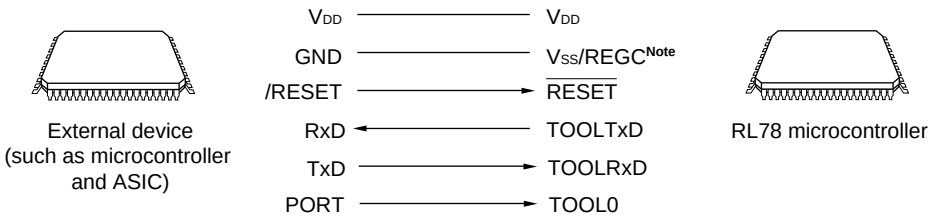
Processing to write data to or delete data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

26.2.2 Communication mode

Communication between the external device and the RL78 microcontroller is established by serial communication using the TOOLTxD and TOOLRxD pins via the dedicated UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2kbps

Figure 26-4. Communication with External Device



Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The external device generates the following signals for the RL78 microcontroller.

Table 26-3. 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}, REGC **Note**
RESETOUT	Output	Reset signal output	$\overline{RESET}$
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

26.3 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 **26.4.2 Flash memory programming mode**.

26.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1 kΩ pull-up resistor.

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 **30.11** or **31.11 Timing Specs for Switching Flash Memory Programming Modes**)

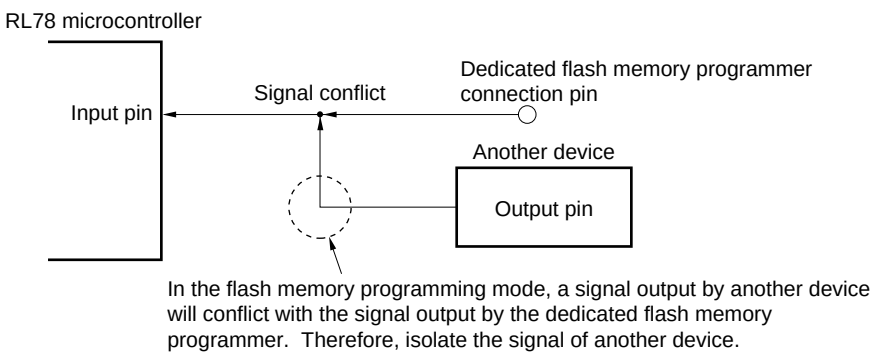
2. The SAU and IICA pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

26.3.2 $\overline{\text{RESET}}$ 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 26-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)



26.3.3 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do 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.

26.3.4 REGC pin

Connect the REGC pin to GND via a capacitor having excellent characteristics (0.47 to 1 μF) in the same manner as during normal operation. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

26.3.5 X1 and X2 pins

Connect X1 and X2 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_{HOCO}) is used.

26.3.6 Power supply

To use the supply voltage output of 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.

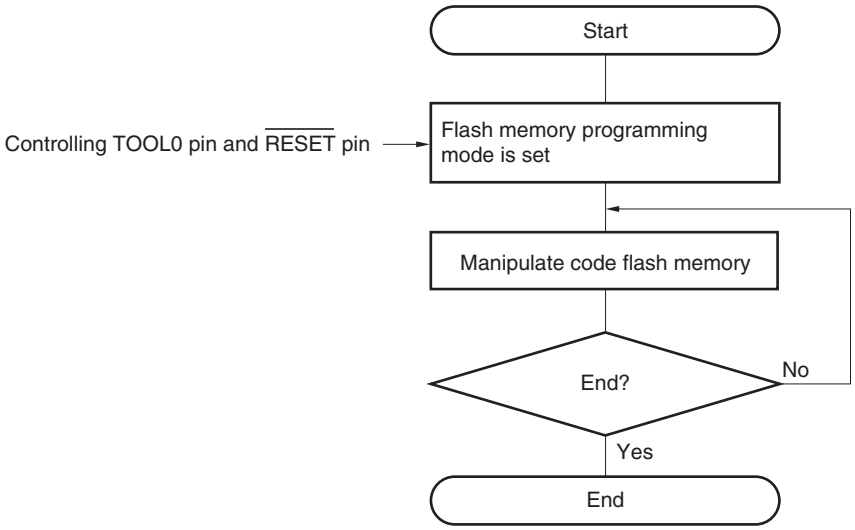
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

26.4 Serial Programming Method

26.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 26-6. Code Flash Memory Manipulation Procedure



26.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory through serial programming, specify the flash memory programming mode. To enter the mode, set as follows.

<Serial programming using the dedicated flash memory programmer>

Connect the RL78 microcontroller to a dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

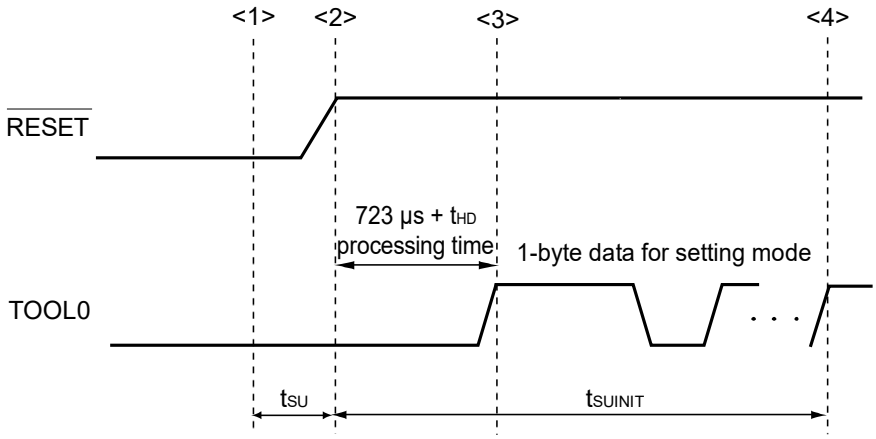
<Serial programming using an external device (UART communication)>

Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 26-4**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 26-7**. For details, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 26-4. 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 26-7. Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Baud rate setting by UART reception is completed.

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 and internal resets end (the flash firmware processing time is excluded)

For details, see **30.10** or **31.10 Timing Specs for Switching Flash Memory Programming Modes**.

There are two flash memory programming modes: wide voltage mode and full speed mode. The supply voltage value applied to the microcontroller during write operations and the setting information of the user option byte for setting of the flash memory programming mode determine which mode is selected.

When a dedicated flash memory programmer is used for serial programming, setting the voltage on GUI selects the mode automatically.

Table 26-5. Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Power Supply Voltage (V_{DD})	User Option Byte Setting for Switching to Flash Memory Programming Mode		Flash Programming Mode
	Flash Operation Mode	Operating Frequency	
2.7 V ≤ V_{DD} ≤ 5.5 V	Blank state		Full speed mode
	HS (high speed main) mode	1 MHz to 24 MHz	Full speed mode
2.4 V ≤ V_{DD} < 2.7 V	Blank state		Full speed mode
	HS (high speed main) mode	1 MHz to 16 MHz	Full speed mode

Remarks 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

2. For details about communication commands, see **26.4.4 Communication commands**.

26.4.3 Selecting communication mode

Communication modes of the RL78 microcontroller are as follows.

Table 26-6. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed [Note 2]	Frequency	Multiply Rate	
1-line UART (when flash memory programmer is used, or when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOL0
Dedicated UART (when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOLTxD, TOOLRxD

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.

26.4.4 Communication commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 26-7.**

The signals sent from the dedicated flash memory programmer or external device to the RL78 microcontroller are called commands, and programming functions corresponding to the commands are executed. For details, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 26-7. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Write	Programming	Writes data to a specified area in the flash memory.
Getting information	Silicon Signature	Gets the RL78 microcontroller information (such as the part number, flash memory configuration, and programming firmware version).
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
	Security Get	Gets security information.
	Security Release	Release setting of prohibition of writing.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

Product information (such as product name and firmware version) can be obtained by executing the “Silicon Signature” command.

Table 26-8 is a list of signature data and **Table 26-9** shows an example of signature data.

Table 26-8. Signature Data List

Field name	Description	Number of transmit data
Device code	The serial number assigned to the device	3 bytes
Device name	Device name (ASCII code)	10 bytes
Code flash memory area last address	Last address of code flash memory area (Sent from lower address. Example: 00000H to 0FFFFH (64 KB) → FFH, 1FH, 00H)	3 bytes
Data flash memory area last address	Last address of data flash memory area (Sent from lower address. Example: F1000H to F1FFFH (4 KB) → FFH, 1FH, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from upper address. Example: From Ver. 1.23 → 01H, 02H, 03H)	3 bytes

Table 26-9. Example of Signature Data

Field name	Description	Number of transmit data	Data (hexadecimal)
Device code	RL78 protocol A	3 bytes	10 00 06
Device name	R5F10JBC	10 bytes	52 = “R” 35 = “5” 46 = “F” 31 = “1” 30 = “0” 30 = “J” 4C = “B” 45 = “C” 20 = “ ” 20 = “ ”
Code flash memory area last address	Code flash memory area 00000H to 07FFFH (32 KB)	3 bytes	FF 7F 00
Data flash memory area last address	Data flash memory area F1000H to F17FFH (2 KB)	3 bytes	FF 17 0F
Firmware version	Ver.1.23	3 bytes	01 02 03

26.5 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP5 is used as a dedicated flash memory programmer.

Table 26-10. Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

PG-FP5 Command	Code Flash
	32 Kbytes
Erasing	1 s
Writing	1.5 s
Verification	1.5 s
Writing after erasing	2 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
Mode: Full speed mode (flash operation mode: HS (high speed main) mode)

26.6 Self-Programming

The RL78 microcontroller supports a self-programming function that can be used to rewrite the code flash memory via a user program. Because this function allows a user application to rewrite the code flash memory by using the flash self-programming library, it can be used to upgrade the program in the field.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.

2. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the flash self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the flash self-programming library.

3. The high-speed on-chip oscillator should be kept operating during self-programming. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The flash self-programming library should be executed after 30 μs have elapsed.

Remarks 1. For details of the self-programming function, refer to **RL78 Microcontroller Flash Self Programming Library Type01 User's Manual (R01US0050).**

2. For details of the time required to execute self-programming, see the notes on use that accompany the flash self-programming library tool.

The self-programming function has two flash memory programming modes; wide voltage mode and full speed mode.

Specify the mode that corresponds to the flash operation mode specified in bits CMODE1 and CMODE0 in option byte 000C2H.

Specify the full speed mode when the HS (high speed main) mode is specified. Specify the wide voltage mode when the LS (low speed main) mode or LV (low voltage main) mode is specified.

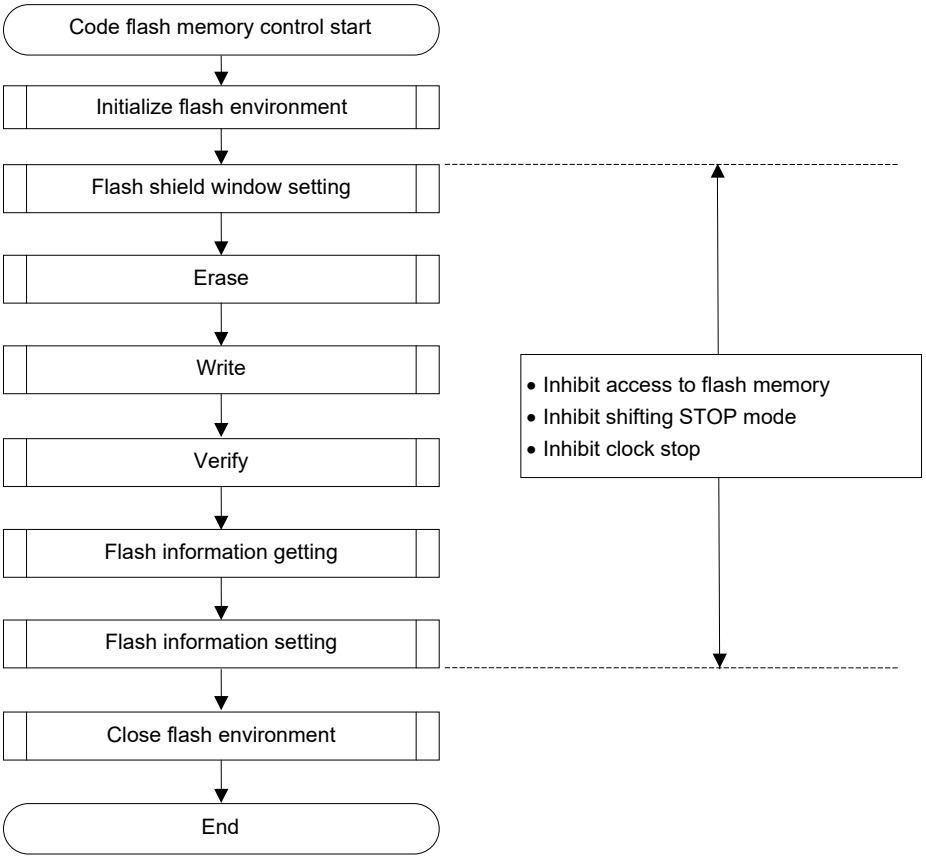
If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

Remark Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

26.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the code flash memory by using a flash self-programming library.

Figure 26-8. Flow of Self Programming (Rewriting Flash Memory)



26.6.2 Boot swap function

If rewriting the boot area failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

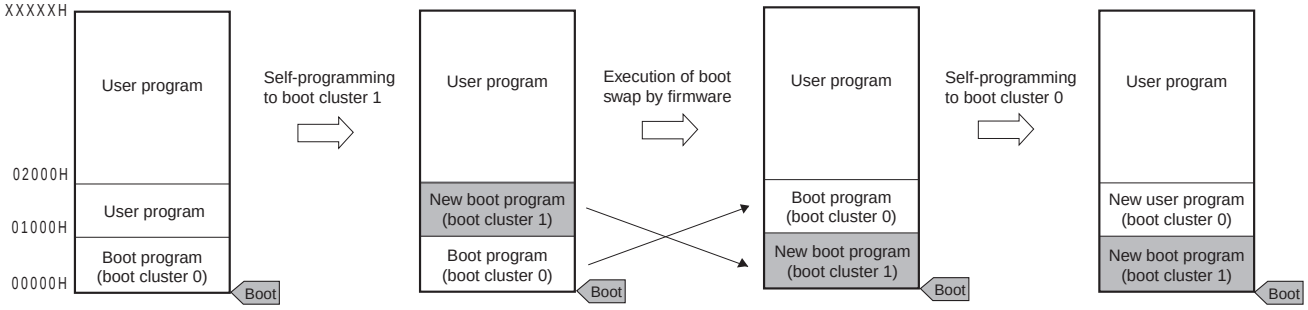
The boot swap function is used to avoid this problem.

Before erasing boot cluster 0[Note], which is a boot area, by self-programming, write a new boot program to boot cluster 1 in advance. When the program has been correctly written to boot cluster 1, swap this boot cluster 1 and boot cluster 0 by using the set information function of the firmware of the RL78 microcontroller, so that boot cluster 1 is used as a boot area. After that, erase or write the original area, boot cluster 0.

As a result, even if a power failure occurs while the area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 to be swapped when the program is reset and started next.

Note A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Figure 26-9. Boot Swap Function

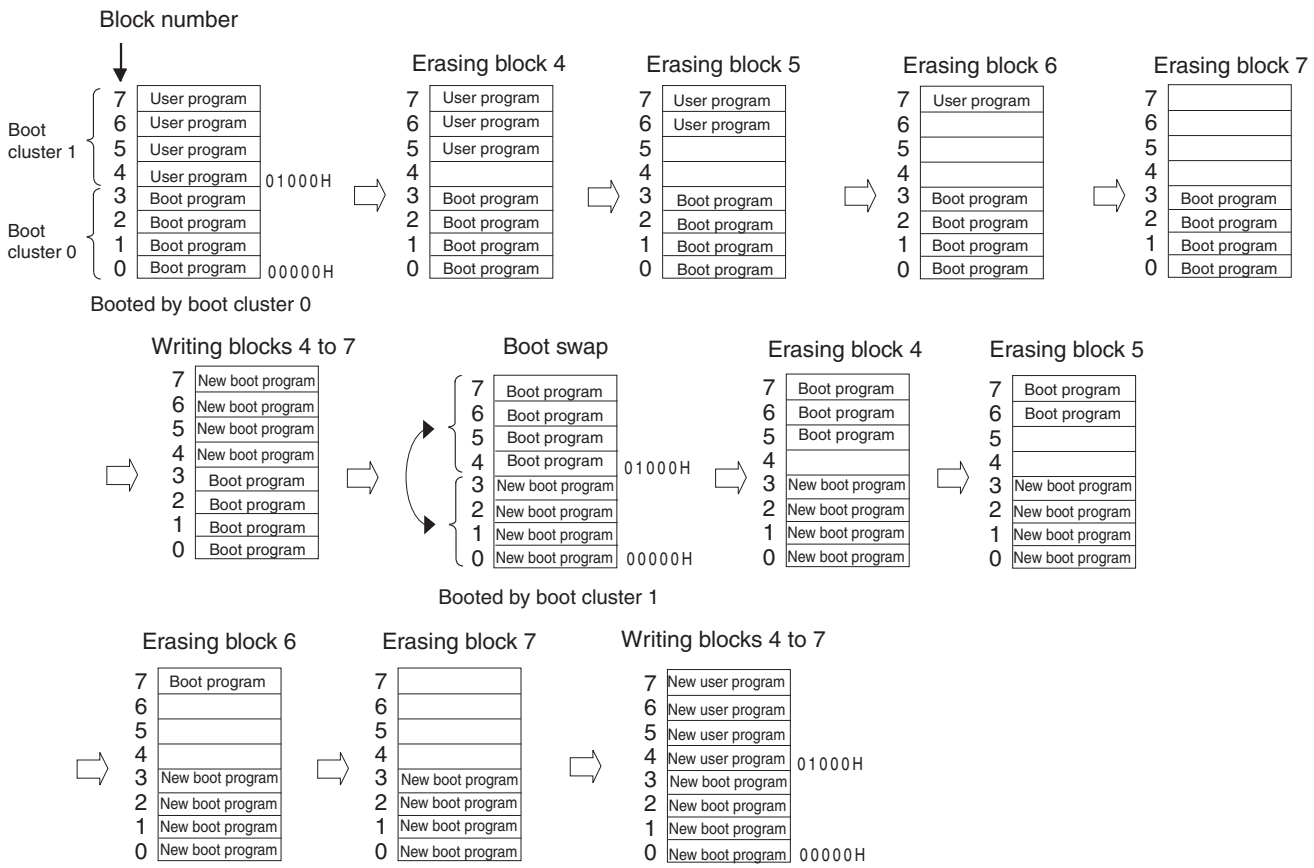


In an example of above figure, it is as follows.

Boot cluster 0: Boot area before boot swap

Boot cluster 1: Boot area after boot swap

Figure 26-10. Example of Executing Boot Swapping



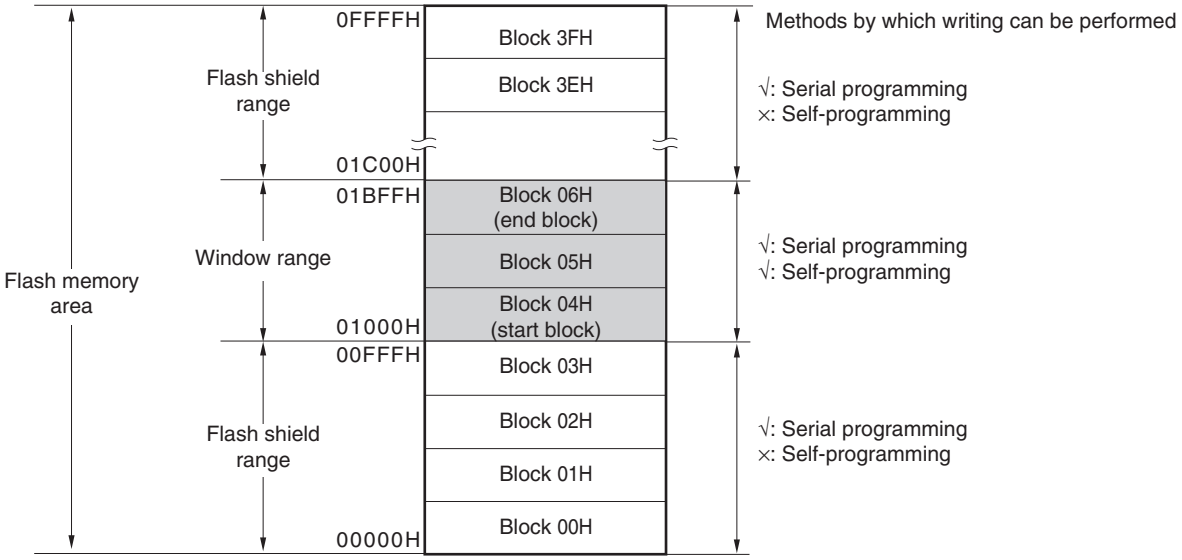
26.6.3 Flash shield window function

The flash shield window function is provided as one of the security functions for self-programming. It disables writing to and erasing areas outside the range specified as a window only during self-programming.

The window range can be set by specifying the start and end blocks. The window range can be set or changed during both serial programming and self-programming.

Writing to and erasing areas outside the window range are disabled during self-programming. During serial programming, however, areas outside the range specified as a window can be written and erased.

Figure 26-11. Flash Shield Window Setting Example
(Target Devices: R5F100LE, Start Block: 04H, End Block: 06H)



Cautions 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 26-11. Relationship between Flash Shield Window Function Setting/Change Methods and Commands

Programming conditions	Window Range Setting/Change Methods	Execution Commands	
		Block erase	Write
Self-programming	Specify the starting and ending blocks by the flash self-programming library.	Block erasing is enabled only within the window range.	Writing is enabled only within the range of window range.
Serial programming	Specify the starting and ending blocks on GUI of dedicated flash memory programmer, etc.	Block erasing is enabled also outside the window range.	Writing is enabled also outside the window range.

Remark See **26.7 Security Settings** to prohibit writing/erasing during serial programming.

26.7 Security Settings

The RL78 microcontroller supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.

The operations shown below can be performed using the Security Set command.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during serial programming. However, blocks can be erased by means of self-programming.

- Disabling write
  Execution of the write command for entire blocks in the code flash memory is prohibited during serial programming. However, blocks can be written by means of self-programming.
  After the setting of prohibition of writing is specified, releasing the setting by the Security Release command is enabled by a reset.

- Disabling rewriting boot cluster 0
  Execution of the block erase command and write command on boot cluster 0 (00000H to 00FFFH) in the code flash memory is prohibited by this setting.

The block erase, write commands and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by serial programming and self-programming. Each security setting can be used in combination.

Table 26-12 shows the relationship between the erase and write commands when the RL78 microcontroller security function is enabled.

Caution The security function of the dedicated flash programmer does not support self-programming.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **26.6.3** for detail).

Table 26-12. Relationship between Enabling Security Function and Command

(1) During serial programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks cannot be erased.	Can be performed. Note
Prohibition of writing	Blocks can be erased.	Cannot be performed.
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

(2) During self-programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks can be erased.	Can be performed.
Prohibition of writing		
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **26.6.3** for detail).

Table 26-13. Setting Security in Each Programming Mode

(1) During serial programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of writing		Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

Caution Releasing the setting of prohibition of writing is enabled only when the security is not set as the block erase prohibition and the boot cluster 0 rewrite prohibition with code flash memory area and data flash memory area being blanks.

(2) During self-programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set by using flash self-programming library.	Cannot be disabled after set.
Prohibition of writing		Cannot be disabled during self-programming (set via GUI of dedicated flash memory programmer, etc. during serial programming).
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

26.8 Data Flash

26.8.1 Data flash overview

An overview of the data flash memory is provided below.

- The user program can rewrite the data flash memory by using the data flash library. For details, refer to **RL78 Family Data Flash Library User's Manual**.
- The data flash memory can also be rewritten to through serial programming using the dedicated flash memory programmer or an external device.
- The data flash can be erased in 1-block (1-Kbyte) units.
- The data flash can be accessed only in 8-bit units.
- The data flash can be directly read by CPU instructions.
- Instructions can be executed from the code flash memory while rewriting the data flash memory (that is, background operation (BGO) is supported).
- Because the data flash memory is an area exclusively used for data, it cannot be used to execute instructions.
- Accessing the data flash memory is not possible while rewriting the code flash memory (during self-programming).
- Manipulating the DFLCTL register is not possible while rewriting the data flash memory.
- Transition to the STOP mode is not possible while rewriting the data flash memory.

Cautions 1. The data flash memory is stopped after a reset is canceled. The data flash control register (DFLCTL) must be set up in order to use the data flash memory.

2. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The data flash library should be executed after 30 μs have elapsed.

Remark For rewriting the code flash memory via a user program, see **26.6 Self-Programming**.

26.8.2 Register controlling data flash memory

26.8.2.1 Data flash control register (DFLCTL)

This register is used to enable or disable accessing to the data flash.

The DFLCTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 26-12. Format of Data Flash Control Register (DFLCTL)

Address: F0090H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash access control
0	Disables data flash access
1	Enables data flash access

Caution Manipulating the DFLCTL register is not possible while rewriting the data flash memory.

26.8.3 Procedure for accessing data flash memory

The data flash memory is stopped after a reset ends. To access the data flash, make initial settings according to the following procedure.

<1> Set bit 0 (DFLEN) of the data flash control register (DFLCTL) to 1.

<2> Wait for the setup to finish for software timer, etc.

The time setup takes differs for each flash operation mode for the main clock.

<Setup time for each flash operation mode>

- HS (High speed main): 5 μs
- LS (Low speed main): 720 ns
- LV (Low voltage main): 10 μs

<3> After the wait, the data flash memory can be accessed.

Cautions 1. Accessing the data flash memory is not possible during the setup time.

2. Transition to the STOP mode is not possible during the setup time. To enter the STOP mode during the setup time, clear DFLEN to 0 and then execute the STOP instruction.

3. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The data flash library should be executed after 30 μs have elapsed.

<R> **4. Once the data flash memory is read while the subsystem clock is selected as the CPU/peripheral hardware clock (CLS = 1), follow the procedure listed as steps (1) to (3) below, in that order, to read the data flash area after switching the CPU/peripheral hardware clock from the subsystem clock to the main system clock.**

(1) Make sure the main system clock is selected as the CPU/peripheral hardware clock (CLS = 0).

(2) Read data from any location in the data flash area. The value read at this point is undefined.

(3) Wait for the time listed below according to the operating mode, then read data from the desired parts of the data flash area.
HS (high-speed main) mode: 5 μs
LS (low-speed main) mode: 1 μs
LV (low-voltage main) mode: 10 μs

After initialized, the data flash memory can be read by using a CPU instruction or can be read/written by using a data flash library.

If the DMA controller operates when the data flash memory is accessed, however, follow one of these procedures:

(A) Suspending/forcibly terminating DMA transfer

Before reading the data flash memory, suspend DMA transfer of all the channels used.

After setting the DWAITn bit to 1, however, wait at least for the duration of three clocks (f_{CLK}) before reading the data flash memory. After reading the data flash memory, lift the suspension of transfer by clearing the DWAITn bit to 0.

Or, forcibly terminate DMA transfer in accordance with the procedure in **16.5.5 Forced termination by software** before reading the data flash memory. Resume DMA transfer after the data flash memory has been read.

(B) Access the data flash memory

Access the data flash memory by using the newest data flash library.

(C) Insertion of NOP

Insert an NOP instruction immediately before the instruction that reads the data flash memory.

<Example>

```
MOVW    HL,!addr16    ; Reads RAM.
NOP                   ; Insert NOP instruction before reading data flash memory.
MOV     A,[DE]        ; Read data flash memory.
```

If a high-level language such as C is used, however, the compiler may generate two instructions for one code. In this case, the NOP instruction is not inserted immediately before the data flash memory read instruction. Therefore, read the data flash memory by (A) or (B) above.

Remarks 1. n: DMA channel number (n = 0, 1)

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

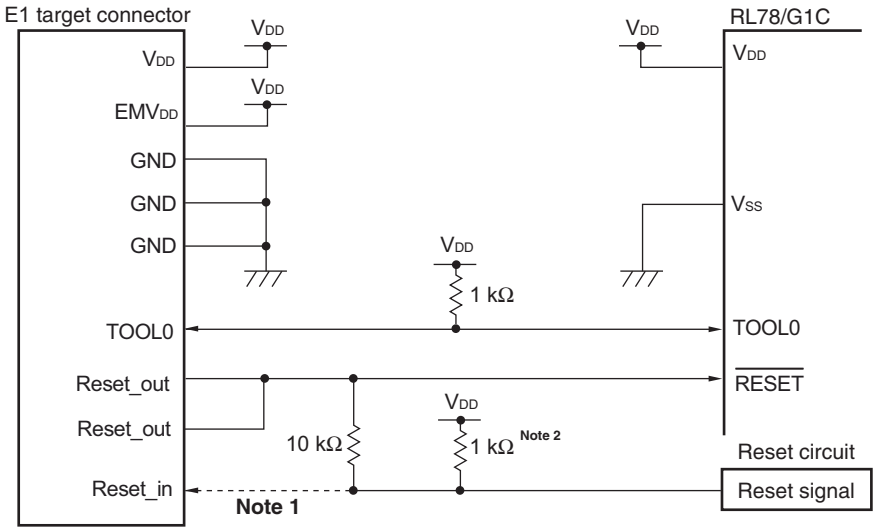
CHAPTER 27 ON-CHIP DEBUG FUNCTION

27.1 Connecting E1 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 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.

Figure 27-1. Connection Example of E1 On-chip Debugging Emulator



Notes 1. Connecting the dotted line is not necessary during serial programming.

2. If the reset circuit on the target system does not have a buffer and generates a reset signal only with resistors and capacitors, this pull-up resistor is not necessary.

Caution This circuit diagram is assumed that the reset signal outputs from an N-ch open-drain buffer (output resistor: 100 Ω or less)

27.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 25 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 010C3H and 010C4H to 010CDH in advance, because 000C3H, 000C4H to 000CDH and 010C3H, and 010C4H to 010CDH are switched.

Table 27-1. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes (except All FFH)
010C4H to 010CDH	

27.3 Securing of User Resources

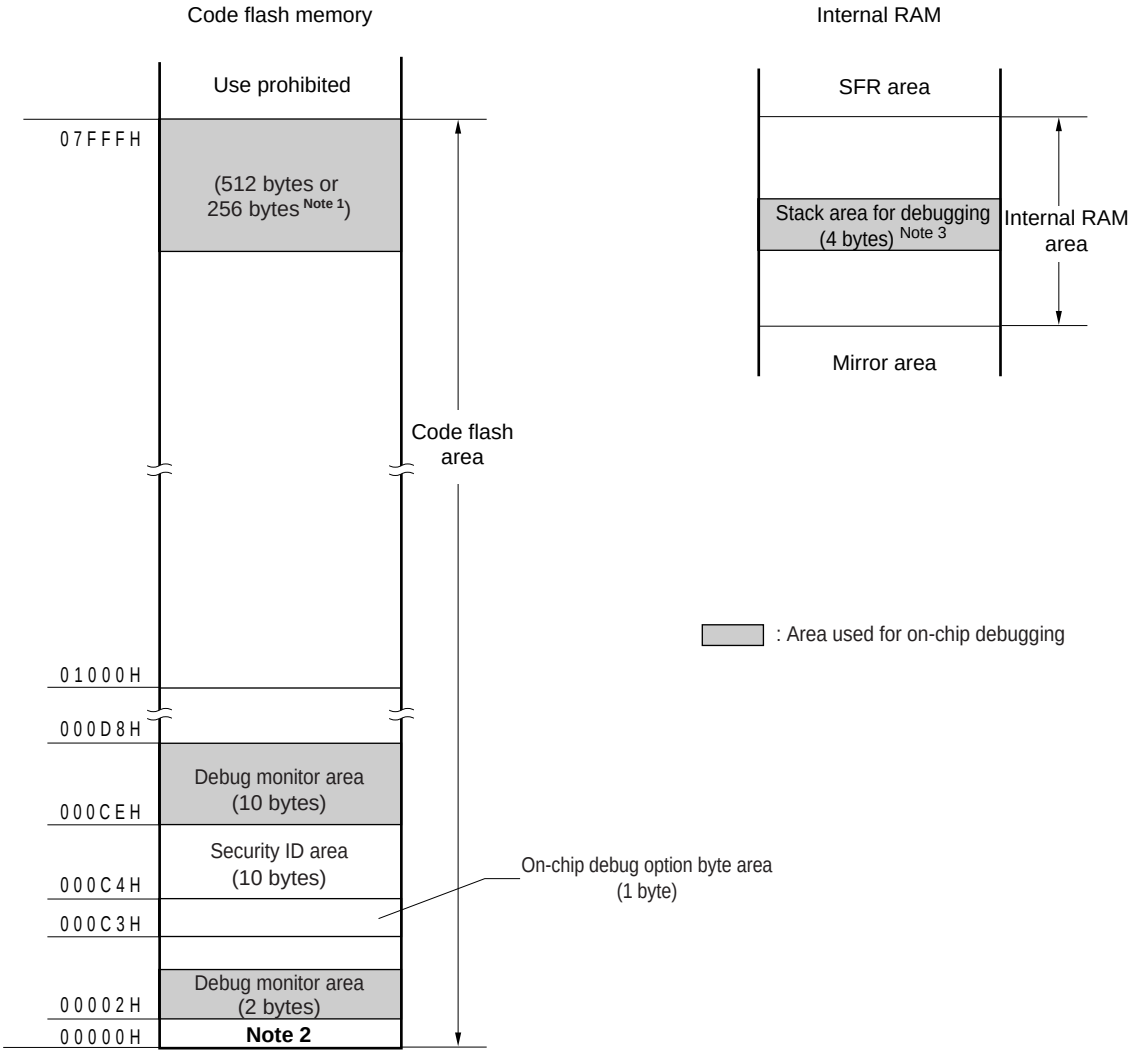
To perform communication between the RL78 microcontroller and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using link options.

(1) Securement of memory space

The shaded portions in Figure 27-2 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 27-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

2. In debugging, reset vector is rewritten to address allocated to a monitor program.

3. 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, 4 extra bytes are consumed for the stack area used. When using self-programming, 12 extra bytes are consumed for the stack area used.

CHAPTER 28 BCD CORRECTION CIRCUIT

28.1 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).

28.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

28.2.1 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 28-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

28.3 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 29 INSTRUCTION SET

This chapter lists the instructions in the RL78 microcontroller instruction set. For details of each operation and operation code, refer to the separate document **RL78 Family User's Manual: Software (R01US0015)**.

29.1 Conventions Used in Operation List

29.1.1 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 the table below, R0, R1, R2, etc.) can be used for description.

Table 29-1. Operand Identifiers and Specification Methods

Identifier	Description Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol (SFR symbol) FFF00H to FFFFFH Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only[Note]) FFF00H to FFFFFH
saddr saddrp	FFE20H to FFF1FH Immediate data or labels FFE20H to FFF1FH Immediate data or labels (even addresses only[Note])
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions[Note]) 0080H to 00BFH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 1-bit immediate data or label
RBn	RB0 to RB3

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 3-5 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 3-6 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

29.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 29-2. 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
RBS	Register bank select 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)

29.1.3 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 29-3. 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

29.1.4 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 29-4. 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.

29.2 Operation List

Table 29-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (3/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (4/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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	1	–	rp ← word	
		saddrp, #word	4	1	–	(saddrp) ← word	
		sfrp, #word	4	1	–	sfrp ← word	
		AX, rp Note 3	1	1	–	AX ← rp	
		rp, AX Note 3	1	1	–	rp ← AX	
		AX, !addr16	3	1	4	AX ← (addr16)	
		!addr16, AX	3	1	–	(addr16) ← AX	
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)	
		ES:!addr16, AX	4	2	–	(ES, addr16) ← AX	
		AX, saddrp	2	1	–	AX ← (saddrp)	
		saddrp, AX	2	1	–	(saddrp) ← AX	
		AX, sfrp	2	1	–	AX ← sfrp	
		sfrp, AX	2	1	–	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, or when the data flash memory is accessed by an 8-bit instruction.

3. Except rp = AX

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (5/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag
				Note 1	**Note 2**		Z AC CY
16-bit data transfer	MOVW	AX, [DE]	1	1	4	AX ← (DE)	
		[DE], AX	1	1	–	(DE) ← AX	
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)	
		ES:[DE], AX	2	2	–	(ES, DE) ← AX	
		AX, [HL]	1	1	4	AX ← (HL)	
		[HL], AX	1	1	–	(HL) ← AX	
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)	
		ES:[HL], AX	2	2	–	(ES, HL) ← AX	
		AX, [DE+byte]	2	1	4	AX ← (DE+byte)	
		[DE+byte], AX	2	1	–	(DE+byte) ← AX	
		AX, ES:[DE+byte]	3	2	5	AX ← ((ES, DE) + byte)	
		ES:[DE+byte], AX	3	2	–	((ES, DE) + byte) ← AX	
		AX, [HL+byte]	2	1	4	AX ← (HL + byte)	
		[HL+byte], AX	2	1	–	(HL + byte) ← AX	
		AX, ES:[HL+byte]	3	2	5	AX ← ((ES, HL) + byte)	
		ES:[HL+byte], AX	3	2	–	((ES, HL) + byte) ← AX	
		AX, [SP+byte]	2	1	–	AX ← (SP + byte)	
		[SP+byte], AX	2	1	–	(SP + byte) ← AX	
		AX, word[B]	3	1	4	AX ← (B + word)	
		word[B], AX	3	1	–	(B+ word) ← AX	
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)	
		ES:word[B], AX	4	2	–	((ES, B) + word) ← AX	
		AX, word[C]	3	1	4	AX ← (C + word)	
		word[C], AX	3	1	–	(C + word) ← AX	
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)	
		ES:word[C], AX	4	2	–	((ES, C) + word) ← AX	
		AX, word[BC]	3	1	4	AX ← (BC + word)	
		word[BC], AX	3	1	–	(BC + word) ← AX	
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)	
		ES:word[BC], AX	4	2	–	((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, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
16-bit data transfer	MOVW	BC, !addr16	3	1	4	BC ← (addr16)			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
		BC, saddrp	2	1	–	BC ← (saddrp)			
		DE, saddrp	2	1	–	DE ← (saddrp)			
		HL, saddrp	2	1	–	HL ← (saddrp)			
	XCHW	AX, rp Note 3	1	1	–	AX ←→ rp			
	ONEW	AX	1	1	–	AX ← 0001H			
		BC	1	1	–	BC ← 0001H			
	CLRW	AX	1	1	–	AX ← 0000H			
		BC	1	1	–	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, or when the data flash memory is accessed by an 8-bit instruction.

3. Except rp = AX

4. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (7/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (9/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (10/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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, or when the data flash memory is accessed by an 8-bit instruction.

3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (11/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	–	AX, CY ← AX+word	×	×	×
		AX, AX	1	1	–	AX, CY ← AX+AX	×	×	×
		AX, BC	1	1	–	AX, CY ← AX+BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX+DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX+HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX+(addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX+(ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX+(saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX+(HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX+((ES:HL)+byte)	×	×	×
	SUBW	AX, #word	3	1	–	AX, CY ← AX – word	×	×	×
		AX, BC	1	1	–	AX, CY ← AX – BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX – DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX – HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX – ((ES:HL)+byte)	×	×	×
	CMPW	AX, #word	3	1	–	AX – word	×	×	×
		AX, BC	1	1	–	AX – BC	×	×	×
		AX, DE	1	1	–	AX – DE	×	×	×
		AX, HL	1	1	–	AX – HL	×	×	×
		AX, !addr16	3	1	4	AX – (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX – (ES:addr16)	×	×	×
		AX, saddrp	2	1	–	AX – (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX – (HL+byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX – ((ES:HL)+byte)	×	×	×
Multiply	MULU	X	1	1	–	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, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (12/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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	1	–	rp ← rp+1			
		!addr16	3	2	–	(addr16) ← (addr16)+1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16)+1			
		saddrp	2	2	–	(saddrp) ← (saddrp)+1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte)+1			
		ES: [HL+byte]	4	3	–	((ES:HL)+byte) ← ((ES:HL)+byte)+1			
	DECW	rp	1	1	–	rp ← rp – 1			
		!addr16	3	2	–	(addr16) ← (addr16) – 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) – 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1			
		ES: [HL+byte]	4	3	–	((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	1	–	(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	1	–	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) ×cnt			×
		BC, cnt	2	1	–	(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	1	–	(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, or when the data flash memory is accessed by an 8-bit instruction.

Remarks 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

2. cnt indicates the bit shift count.

Table 29-5. Operation List (13/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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 ← A_0, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$)×1			×
	ROLC	A, 1	2	1	–	(CY ← A_7, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$)×1			×
	ROLWC	AX,1	2	1	–	(CY ← AX_{15}, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) ×1			×
		BC,1	2	1	–	(CY ← 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	–	CYX ← 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, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (14/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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			
		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, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
Call/ return	CALL	rp	2	3	–	(SP – 2) ← $(PC+2)_S$, (SP – 3) ← $(PC+2)_H$, (SP – 4) ← $(PC+2)_L$, PC ← CS, rp, SP ← SP – 4			
		$!addr20	3	3	–	(SP – 2) ← $(PC+3)_S$, (SP – 3) ← $(PC+3)_H$, (SP – 4) ← $(PC+3)_L$, PC ← PC+3+jdisp16, SP ← SP – 4			
		!addr16	3	3	–	(SP – 2) ← $(PC+3)_S$, (SP – 3) ← $(PC+3)_H$, (SP – 4) ← $(PC+3)_L$, PC ← 0000, addr16, SP ← SP – 4			
		!!addr20	4	3	–	(SP – 2) ← $(PC+4)_S$, (SP – 3) ← $(PC+4)_H$, (SP – 4) ← $(PC+4)_L$, PC ← addr20, SP ← SP – 4			
	CALLT	[addr5]	2	5	–	(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	5	–	(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	6	–	PC_L ← (SP), PC_H ← (SP+1), PC_S ← (SP+2), SP ← SP+4			
	RETI	-	2	6	–	PC_L ← (SP), PC_H ← (SP+1), PC_S ← (SP+2), PSW← (SP+3), SP ← SP+4	R	R	R
	RETB	-	2	6	–	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, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	Flag		
				Note 1	**Note 2**		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	–	(SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP–2			
		rp	1	1	–	(SP – 1) ← rp_H, (SP – 2) ← rp_L, SP ← SP – 2			
	POP	PSW	2	3	–	PSW ← (SP+1), SP ← SP + 2	R	R	R
		rp	1	1	–	rp_L ←(SP), rp_H ← (SP+1), SP ← SP + 2			
	MOVW	SP, #word	4	1	–	SP ← word			
		SP, AX	2	1	–	SP ← AX			
		AX, SP	2	1	–	AX ← SP			
		HL, SP	3	1	–	HL ← SP			
		BC, SP	3	1	–	BC ← SP			
		DE, SP	3	1	–	DE ← SP			
	ADDW	SP, #byte	2	1	–	SP ← SP + byte			
	SUBW	SP, #byte	2	1	–	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, or when the data flash memory is accessed by an 8-bit instruction.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 29-5. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Clocks	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	SEL Note4	RBn	2	1	–	RBS[1:0] ← n	
	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, or when the data flash memory is accessed by an 8-bit instruction.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

4. n indicates register bank number (n = 0 to 3).

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

CHAPTER 30 ELECTRICAL SPECIFICATIONS (A: T_A = -40 to +85°C)

This chapter describes the electrical specifications for the products "A: Consumer applications (T_A = -40 to +85°C)".

The target products A: Consumer applications ; T_A = -40 to +85°C
R5F10JBCANA, R5F10JBCAFP, R5F10JGCANA, R5F10JGCAFB,
R5F10KBCANA, R5F10KBCAFP, R5F10KGCANA, R5F10KGCAFB
G: Industrial applications ; when using T_A = -40 to +105°C specification products at T_A = -40 to +85°C.
R5F10JBCGNA, R5F10JBCGFP, R5F10JGCGNA, R5F10JGCGFB,
R5F10KBCGNA, R5F10KBCGFP, R5F10KGCGNA, R5F10KGCGFB

Cautions 1. The RL78 microcontrollers 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.

2. The pins mounted depend on the product. Refer to 2.1 Port Function to 2.2.1 With functions for each product.

30.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +6.5	V
REGC pin input voltage	V_{IREGC}	REGC	–0.3 to +2.8 and –0.3 to V_{DD} +0.3 Note 1	V
UV_{DD} pin input voltage	V_{IUVDD}	UV_{DD}	–0.3 to V_{DD} +0.3	V
Input voltage	V_{I1}	P00, P01, P14 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P70 to P75, P120 to P124, P137, P140, EXCLK, EXCLKS, $\overline{RESET}$	–0.3 to V_{DD} +0.3 Note 2	V
	V_{I2}	P60 to P63 (N-ch open-drain)	–0.3 to +6.5	V
	V_{I3}	UDP0, UDM0, UDP1, UDM1	–0.3 to +6.5	V
	V_{I4}	UV_{BUS}	–0.3 to +6.5	V
Output voltage	V_{O1}	P00, P01, P14 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P120, P130, P140	–0.3 to V_{DD} +0.3 Note 2	V
	V_{O2}	UDP0, UDM0, UDP1, UDM1	–0.3 to +6.5	V
Analog input voltage	V_{AI1}	ANI16, ANI17, ANI19	–0.3 to V_{DD} +0.3 and –0.3 to AV_{REF} (+) +0.3 Notes 2, 3	V
	V_{AI2}	ANI0 to ANI7	–0.3 to V_{DD} +0.3 and –0.3 to AV_{REF} (+) +0.3 Notes 2, 3	V

Notes 1. Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

2. Must be 6.5 V or lower.

3. Do not exceed AV_{REF}(+) + 0.3 V in case of A/D conversion target pin

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. AV_{REF} (+) : The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

3. V_{SS} : Reference voltage

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140	–40	mA
		Total of all pins –170 mA	P00, P01, P40, P41, P120, P130, P140	–70	mA
			P14 to P17, P30, P31, P50, P51, P70 to P75	–100	mA
	I_{OH2}	Per pin	P20 to P27	–0.5	mA
		Total of all pins		–2	mA
Output current, low	I_{OL1}	Per pin	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P120, P130, P140	40	mA
		Total of all pins 170 mA	P00, P01, P40, P41, P120, P130, P140	70	mA
			P14 to P17, P30, P31, P50, P51, P60 to P63, P70 to P75	100	mA
	I_{OL2}	Per pin	P20 to P27	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A	In normal operation mode		–40 to +85	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			–65 to +150	°C

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.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

30.2 Oscillator Characteristics

30.2.1 X1, XT1 oscillator characteristics

(T_A = –40 to +85°C, 2.4 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	1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1.0		16.0	MHz
XT1 clock oscillation frequency (f_{XT})Note	Crystal resonator		32	32.768	35	kHz

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 and XT1 oscillator, refer to **5.4 System Clock Oscillator**.

30.2.2 On-chip oscillator characteristics

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Notes 1, 2	f_{HOCO}		1		48	MHz
High-speed on-chip oscillator clock frequency accuracy		–20 to +85 °C	–1.0		+1.0	%
		–40 to –20 °C	–1.5		+1.5	%
Low-speed on-chip oscillator clock frequency	f_{IL}			15		kHz
Low-speed on-chip oscillator clock frequency accuracy			–15		+15	%

Notes 1. High-speed on-chip oscillator frequency is selected by bits 0 to 3 of option byte (000C2H/010C2H) and bits 0 to 2 of HOCODIV register.

2. This indicates the oscillator characteristics only. Refer to AC Characteristics for instruction execution time.

30.2.3 PLL oscillator characteristics

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
PLL input frequency Note	f_{PLLIN}	High-speed system clock	6.00		16.00	MHz
PLL output frequency Note	f_{PLL}			48.00		MHz
Lock up time		From PLL output enable to stabilization of the output frequency	40.00			μs
Interval time		From PLL stop to PLL re-operation setteing Wait time	4.00			μs
Setting wait time		From after PLL input clock stabilization and PLL setting is fixed to start setting Wait time required	1.00			μs

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

30.3 DC Characteristics

30.3.1 Pin characteristics

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140	2.4 V ≤ V_{DD} ≤ 5.5 V			–10.0 [Note 2]	mA
		Total of P00, P01, P40, P41, P120, P130, P140 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–55.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–10.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			–5.0	mA
		Total of P14 to P17, P30, P31, P50, P51, P70 to P75 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–80.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–19.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			–10.0	mA
		Total of all pins (When duty ≤ 70%[Note 3])	2.4 V ≤ V_{DD} ≤ 5.5 V			–135.0	mA
	I_{OH2}	Per pin for P20 to P27	2.4 V ≤ V_{DD} ≤ 5.5 V			–0.1 [Note 2]	mA
		Total of all pins (When duty ≤ 70%[Note 3])	2.4 V ≤ V_{DD} ≤ 5.5 V			–1.5	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. However, do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty ratio 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

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.

Caution P00, P01, P30, and P74 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 pins.

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140	2.4V ≤ V_{DD} ≤ 5.5 V			20.0 [Note 2]	mA
		Per pin for P60 to P63	2.4V ≤ V_{DD} ≤ 5.5 V			20.0 [Note 2]	mA
		Total of P00, P01, P40, P41, P120, P130, P140 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			70.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			15.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			9.0	mA
		Total of P14 to P17, P30, P31, P50, P51, P60 to P63, P70 to P75 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			80.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			35.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			20.0	mA
		Total of all pins (When duty ≤ 70% [Note 3])	2.4V ≤ V_{DD} ≤ 5.5 V			150.0	mA
	I_{OL2}	Per pin for P20 to P27	2.4V ≤ V_{DD} ≤ 5.5 V			0.4 [Note 2]	mA
		Total of all pins (When duty ≤ 70%[Note 3])	2.4V ≤ V_{DD} ≤ 5.5 V			5.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

2. However, do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty ratio to n%).

- 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.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P140	Normal input buffer	0.8V_{DD}		V_{DD}	V
	V_{IH2}	P00, P01, P30, P50	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		V_{DD}	V
			TTL input buffer 3.3 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
			TTL input buffer 2.4 V ≤ V_{DD} < 3.3 V	1.5		V_{DD}	V
	V_{IH3}	P20 to P27		0.7V_{DD}		V_{DD}	V
	V_{IH4}	P60 to P63		0.7V_{DD}		6.0	V
	V_{IH5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P140	Normal input buffer	0		0.2V_{DD}	V
	V_{IL2}	P00, P01, P30, P50	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.3 V ≤ V_{DD} < 4.0 V	0		0.5	V
			TTL input buffer 2.4 V ≤ V_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P20 to P27		0		0.3V_{DD}	V
	V_{IL4}	P60 to P63		0		0.3V_{DD}	V
	V_{IL5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0		0.2V_{DD}	V

Caution The maximum value of V_{IH} of pins P00, P01, P30, and P74 is V_{DD}, even in the N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –10.0 mA	V_{DD} – 1.5			V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –2.0 mA	V_{DD} – 0.6			V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –1.5 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P27	2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OH2} = –100 μA	V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 20.0 mA			1.3	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.6	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.6 mA			0.4	V
	V_{OL2}	P20 to P27	2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 400 μA			0.4	V
	V_{OL3}	P60 to P63	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 20.0 mA			2.0	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 5.0 mA			0.4	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.4	V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 2.0 mA			0.4	V

Caution P00, P01, P30, and P74 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 pins.

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00, P01, P14 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P120, P137, P140, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00, P01, P14 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P120, P137, P140, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL2}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{SS}$	In input port or external clock input			–1	μA
				In resonator connection			–10	μA
On-chip pll-up resistance	R_U	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P140	$V_I = V_{SS}$, In input port		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

30.3.2 Supply current characteristics

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	HS (High-speed main) mode Note 6	f_{HOCO} = 48 MHz f_{IH} = 24 MHz Note 3	Basic operation	V_{DD} = 5.0 V		1.7		mA
						V_{DD} = 3.0 V		1.7		mA
					Normal operation	V_{DD} = 5.0 V		3.7	5.5	mA
						V_{DD} = 3.0 V		3.7	5.5	mA
				f_{HOCO} = 24 MHz Note 5 f_{IH} = 12 MHz Note 3	Normal operation	V_{DD} = 5.0 V		2.3	3.2	mA
						V_{DD} = 3.0 V		2.3	3.2	mA
				f_{HOCO} = 12 MHz Note 5 f_{IH} = 6 MHz Note 3	Normal operation	V_{DD} = 5.0 V		1.6	2.0	mA
						V_{DD} = 3.0 V		1.6	2.0	mA
				f_{HOCO} = 6 MHz Note 5 f_{IH} = 3 MHz Note 3	Normal operation	V_{DD} = 5.0 V		1.2	1.5	mA
						V_{DD} = 3.0 V		1.2	1.5	mA
			HS (High-speed main) mode Note 6	f_{MX} = 20 MHz Note 2, V_{DD} = 5.0 V	Normal operation	Square wave input		3.0	4.6	mA
						Resonator connection		3.2	4.8	mA
				f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		3.0	4.6	mA
						Resonator connection		3.2	4.8	mA
				f_{MX} = 10 MHz Note 2, V_{DD} = 5.0 V	Normal operation	Square wave input		1.9	2.7	mA
						Resonator connection		1.9	2.7	mA
				f_{MX} = 10 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		1.9	2.7	mA
						Resonator connection		1.9	2.7	mA
			HS (High-speed main) mode (PLL operation) Note 6	f_{PLL} = 48 MHz, f_{CLK} = 24 MHz Note 2	Normal operation	V_{DD} = 5.0 V		4.0	5.9	mA
						V_{DD} = 3.0 V		4.0	5.9	mA
				f_{PLL} = 48 MHz, f_{CLK} = 12 MHz Note 2	Normal operation	V_{DD} = 5.0 V		2.6	3.6	mA
						V_{DD} = 3.0 V		2.6	3.6	mA
				f_{PLL} = 48 MHz, f_{CLK} = 6 MHz Note 2	Normal operation	V_{DD} = 5.0 V		1.9	2.4	mA
						V_{DD} = 3.0 V		1.9	2.4	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 4 T_A = –40°C	Normal operation	Resonator connection		4.1	4.9	μA
						Square wave input		4.2	5.0	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +25°C	Normal operation	Square wave input		4.1	4.9	μA
						Resonator connection		4.2	5.0	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +50°C	Normal operation	Square wave input		4.2	5.5	μA
						Resonator connection		4.3	5.6	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +70°C	Normal operation	Square wave input		4.2	6.3	μA
						Resonator connection		4.3	6.4	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +85°C	Normal operation	Square wave input		4.8	7.7	μA
						Resonator connection		4.9	7.8	μA

(**Notes** and **Remarks** are listed on the next page.)

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 following points apply in the HS (high-speed main) mode.

- The currents in the "TYP." column do not include the operating currents of the peripheral modules.
- The currents in the "MAX." column include the operating currents of the peripheral modules, except for those flowing into the A/D converter, LVD circuit, USB 2.0 host/function module, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the RTC.

2. When high-speed on-chip oscillator and subsystem clock are stopped.

3. When high-speed system clock and subsystem clock are stopped.

4. When high-speed on-chip oscillator and high-speed system clock are stopped. When AMPHS1 = 1 (Ultra-low power consumption oscillation).

5. When Operating frequency setting of option byte = 48 MHz. When f_{HOCO} is divided by HOCODIV. When RDIV[1:0] = 00 (divided by 2: default).

6. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 24 MHz
2.4 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 16 MHz

Remarks 1. f_{HOCO}: High-speed on-chip oscillator clock frequency (Max. 48 MHz)

2. f_{IH}: Main system clock source frequency obtained by dividing the high-speed on-chip oscillator clock by 2, 4, or 8 (Max. 24 MHz)

3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

4. f_{PLL}: PLL oscillation frequency

5. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

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

7. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C.

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (High-speed main) mode Note 8	f_{HOCO} = 48 MHz, f_{IH} = 24 MHz Note 4	V_{DD} = 5.0 V		0.67	1.25	mA
					V_{DD} = 3.0 V		0.67	1.25	mA
				f_{HOCO} = 24 MHz Note 6, f_{IH} = 12 MHz Note 4	V_{DD} = 5.0 V		0.50	0.86	mA
					V_{DD} = 3.0 V		0.50	0.86	mA
				f_{HOCO} = 12 MHz Note 6, f_{IH} = 6 MHz Note 4	V_{DD} = 5.0 V		0.41	0.67	mA
					V_{DD} = 3.0 V		0.41	0.67	mA
				f_{HOCO} = 6 MHz Note 6, f_{IH} = 3 MHz Note 4	V_{DD} = 5.0 V		0.37	0.58	mA
					V_{DD} = 3.0 V		0.37	0.58	mA
			HS (High-speed main) mode Note 8	f_{MX} = 20 MHz Note 3, V_{DD} = 5.0 V	Square wave input		0.28	1.00	mA
					Resonator connection		0.45	1.17	mA
				f_{MX} = 20 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.28	1.00	mA
					Resonator connection		0.45	1.17	mA
				f_{MX} = 10 MHz Note 3, V_{DD} = 5.0 V	Square wave input		0.19	0.60	mA
					Resonator connection		0.26	0.67	mA
				f_{MX} = 10 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.19	0.60	mA
					Resonator connection		0.26	0.67	mA
			HS (High-speed main) mode (PLL operation) Note 8	f_{PLL} = 48 MHz, f_{CLK} = 24 MHz Note 3	V_{DD} = 5.0 V		0.91	1.52	mA
					V_{DD} = 3.0 V		0.91	1.52	mA
				f_{PLL} = 48 MHz, f_{CLK} = 12 MHz Note 3	V_{DD} = 5.0 V		0.85	1.28	mA
					V_{DD} = 3.0 V		0.85	1.28	mA
				f_{PLL} = 48 MHz, f_{CLK} = 6 MHz Note 3	V_{DD} = 5.0 V		0.82	1.15	mA
					V_{DD} = 3.0 V		0.82	1.15	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 5, T_A = –40°C	Square wave input		0.25	0.57	μA
					Resonator connection		0.44	0.76	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = +25°C	Square wave input		0.30	0.57	μA
					Resonator connection		0.49	0.76	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = +50°C	Square wave input		0.33	1.17	μA
					Resonator connection		0.63	1.36	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = +70°C	Square wave input		0.46	1.97	μA
					Resonator connection		0.76	2.16	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = +85°C	Square wave input		0.97	3.37	μA
					Resonator connection		1.16	3.56	μA
	I_{DD3}	STOP mode Note 7	T_A = –40°C				0.18	0.50	μA
			T_A = +25°C				0.23	0.50	μA
			T_A = +50°C				0.26	1.10	μA
			T_A = +70°C				0.29	1.90	μA
			T_A = +85°C				0.90	3.30	μA

(**Notes** and **Remarks** are listed on the next page.)

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 following points apply in the HS (high-speed main) mode.

- The currents in the "TYP." column do not include the operating currents of the peripheral modules.
- The currents in the "MAX." column include the operating currents of the peripheral modules, except for those flowing into the A/D converter, LVD circuit, USB 2.0 host/function module, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the RTC.

In the STOP mode, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules.

2. During HALT instruction execution by flash memory.

3. When high-speed on-chip oscillator and subsystem clock are stopped.

4. When high-speed system clock and subsystem clock are stopped.

5. When high-speed on-chip oscillator and high-speed system clock are stopped. When RTCLPC = 1 and setting ultra-low current consumption (AMPHS1 = 1).

6. When Operating frequency setting of option byte = 48 MHz. When f_{HOCO} is divided by HOCODIV. When RDIV[1:0] = 00 (divided by 2: default).

7. Regarding the value for current to operate the subsystem clock in STOP mode, refer to that in HALT mode.

8. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 24 MHz
2.4 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 16 MHz

Remarks 1. f_{HOCO}: High-speed on-chip oscillator clock frequency (Max. 48 MHz)

2. f_{IH}: Main system clock source frequency obtained by dividing the high-speed on-chip oscillator clock by 2, 4, or 8 (Max. 24 MHz)

3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

4. f_{PLL}: PLL oscillation frequency

5. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

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

7. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C.

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.20		μA
RTC operating current	I_{RTC} Notes 1, 2, 3				0.02		μA
12-bit interval timer operating current	I_{IT} Notes 1, 2, 4				0.02		μA
Watchdog timer operating current	I_{WDT} Notes 1, 2, 5	f_{IL} = 15 kHz			0.22		μA
A/D converter operating current	I_{ADC} Notes 1, 6	When conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		1.3	1.7	mA
			Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.5	0.7	mA
A/D converter reference voltage current	I_{ADREF} Note 1				75.0		μA
Temperature sensor operating current	I_{TMPS} Note 1				75.0		μA
LVD operating current	I_{LVD} Notes 1, 7				0.08		μA
Self-programming operating current	I_{FSP} Notes 1, 9				2.00	12.20	mA
BGO operating current	I_{BGO} Notes 1, 8				2.00	12.20	mA
SNOOZE operating current	I_{SNOZ} Note 1	ADC operation	The mode is performed Note 10		0.50	1.06	mA
			The A/D conversion operations are performed, Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		1.20	1.62	mA
		Simplified SPI (CSI) operation			0.70	0.84	mA

(**Notes** and **Remarks** are listed on the next page.)

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
USB operating current	I_{USBH} Note 11	During USB communication operation under the following settings and conditions (V_{DD} = 5.0 V, T_A = +25°C): • The internal power supply for the USB is used. • X1 oscillation frequency (f_X) = 12 MHz, PLL oscillation frequency (f_{PLL}) = 48 MHz • The host controller (via two ports) is set to operate in full-speed mode with four pipes (end points) used simultaneously. (PIPE4: Bulk OUT transfer (64 bytes), PIPE5: Bulk IN transfer (64 bytes), PIPE6: Interrupt OUT transfer, PIPE7: Interrupt IN transfer). • The USB ports (two ports) are individually connected to a peripheral function via a 0.5 m USB cable.		9.0		mA
	I_{USBF} Note 11	During USB communication operation under the following settings and conditions (V_{DD} = 5.0 V, T_A = +25°C): • The internal power supply for the USB is used. • X1 oscillation frequency (f_X) = 12 MHz, PLL oscillation frequency (f_{PLL}) = 48 MHz • The function controller is set to operate in full-speed mode with four pipes (end points) used simultaneously. (PIPE4: Bulk OUT transfer (64 bytes), PIPE5: Bulk IN transfer (64 bytes), PIPE6: Interrupt OUT transfer, PIPE7: Interrupt IN transfer). • The USB port (one port) is connected to the host device via a 0.5 m USB cable.		2.5		mA
	I_{SUSP} Note 12	During suspended state under the following settings and conditions (V_{DD} = 5.0 V, T_A = +25°C): • The function controller is set to full-speed mode (the UDP0 pin is pulled up). • The internal power supply for the USB is used. • The system is set to STOP mode (When the high-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped. When the watchdog timer is stopped.). • The USB port (one port) is connected to the host device via a 0.5 m USB cable.		240		μA

(**Notes** and **Remarks** are listed on the next page.)

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 real-time clock (RTC) (excluding the operating current of the low-speed on-chip ocsillator and the XT1 oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time clock operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added. I_{DD2} subsystem clock operation includes the operational current of the real-time clock.

4. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip ocsillator and the XT1 oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{IT}, when the 12-bit interval timer operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added.

5. Current flowing only to the watchdog timer (including 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_{WDT} when the watchdog timer is in operation.

6. 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.

7. Current flowing only to the LVD circuit. The current value of the RL78/G1C is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVD circuit operates in the Operating, HALT or STOP mode.

8. Current flowing only during data flash rewrite.

9. Current flowing only during self programming.

10. For shift time to the SNOOZE mode, see **19.3.3 SNOOZE mode.**

11. Current consumed only by the USB module and the internal power supply for the USB.

12. Includes the current supplied from the pull-up resistor of the UDP0 pin to the pull-down resistor of the host device, in addition to the current consumed by this MCU during the suspended state.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

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

4. Temperature condition of the TYP. value is T_A = 25°C

30.4 AC Characteristics

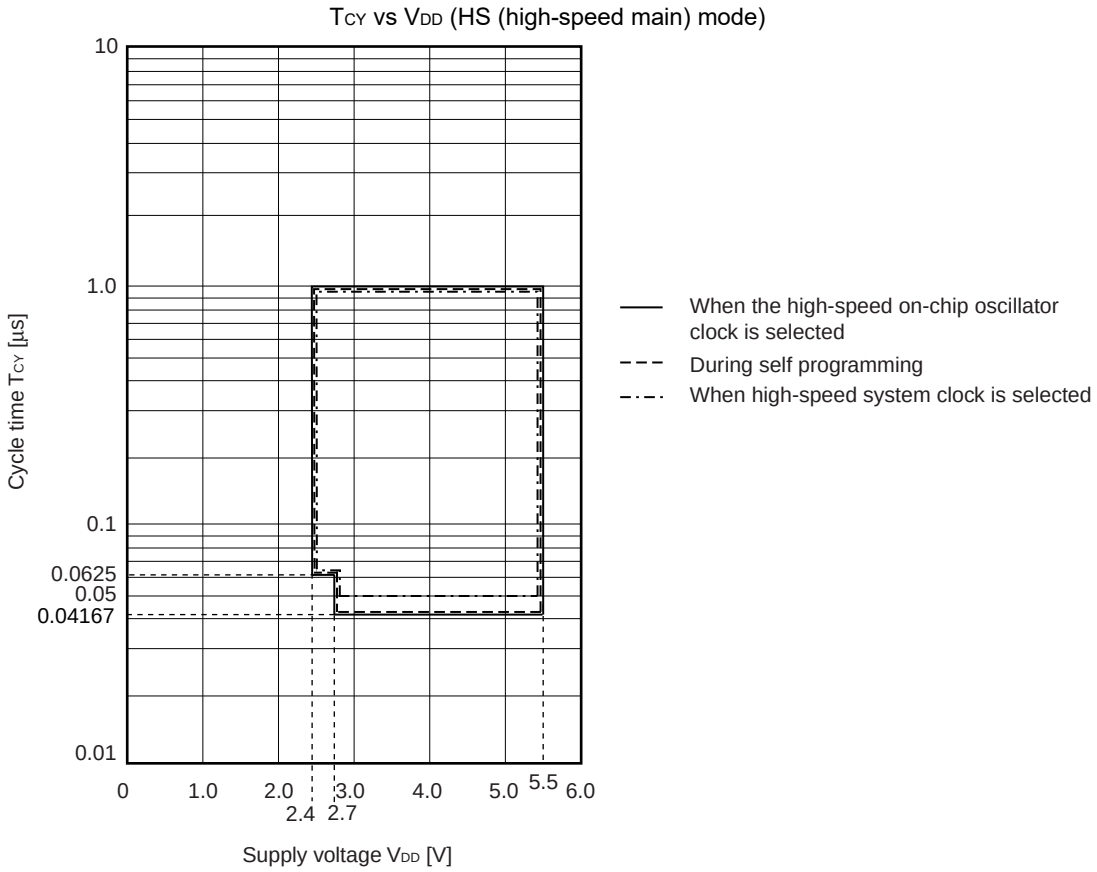
30.4.1 Basic operation

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

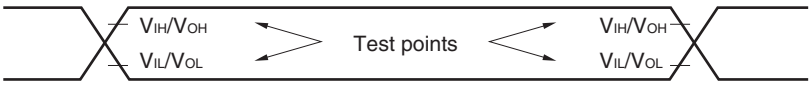
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (High-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
		Subsystem clock (f_{SUB}) operation		2.4 V ≤ V_{DD} ≤ 5.5 V	28.5	30.5	31.3	μs
		In the self programming mode	HS (High-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
External system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V			1.0		16.0	MHz
	f_{EXS}				32		35	kHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V			24			ns
		2.4 V ≤ V_{DD} < 2.7 V			30			ns
	t_{EXHS}, t_{EXLS}				13.7			μs
TI00 to TI03 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK}+10			ns
TO00 to TO03 output frequency	f_{TO}	High-speed main mode	4.0 V ≤ V_{DD} ≤ 5.5 V				12	MHz
			2.7 V ≤ V_{DD} < 4.0 V				8	MHz
			2.4 V ≤ V_{DD} < 2.7 V				4	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	High-speed main mode	4.0 V ≤ V_{DD} ≤ 5.5 V				16	MHz
			2.7 V ≤ V_{DD} < 4.0 V				8	MHz
			2.4 V ≤ V_{DD} < 2.7 V				4	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP6, INTP8, INTP9	2.4 V ≤ V_{DD} ≤ 5.5 V		1			μs
Key interrupt input low-level width	t_{KR}	KR0 to KR5	2.4 V ≤ V_{DD} ≤ 5.5 V		250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of timer mode register 0n (TMR0n). n: Channel number (n = 0 to 3))

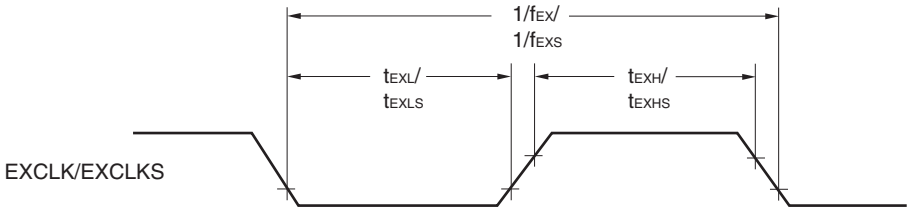
Minimum Instruction Execution Time during Main System Clock Operation



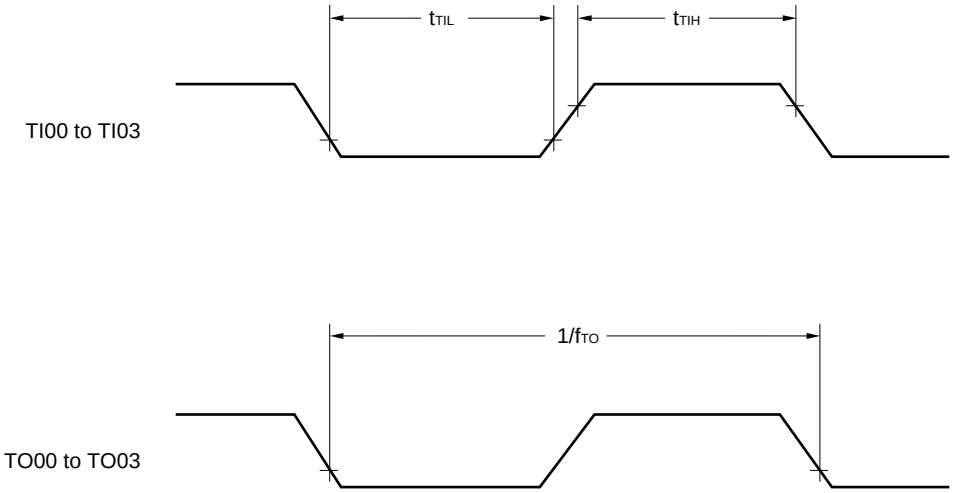
AC Timing Test Points



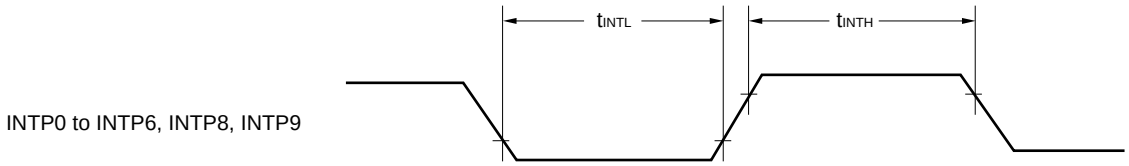
External System Clock Timing



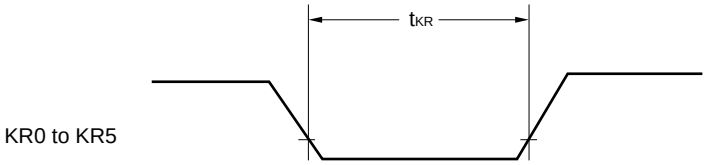
TI/TO Timing



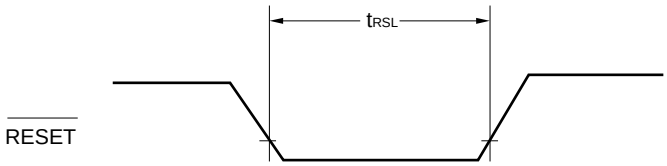
Interrupt Request Input Timing



Key Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



30.5 Peripheral Functions Characteristics

30.5.1 Serial array unit

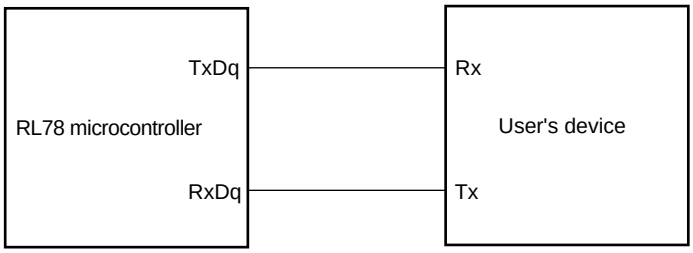
(1) During communication at same potential (UART mode) (dedicated baud rate generator output)
(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note			4.0	Mbps

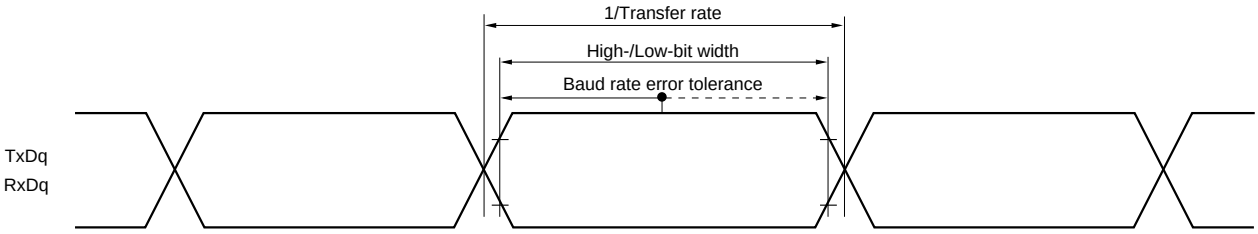
Note The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remarks 1. q: UART number (q = 0), g: PIM and POM number (g = 5)
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))

(2) During communication at same potential (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 2/f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V	83.3			ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 – 7			ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 – 10			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V		23			ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		33			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 5.5 V		10			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 20 pF Note 3				10	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. This specification is valid only when CSI00's peripheral I/O redirect function is not used.

2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0),
g: PIM and POM numbers (g = 3, 5)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))

(3) During communication at same potential (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(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
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V	167			ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	250			ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 – 12			ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 – 18			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 – 38			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V		44			ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		44			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		75			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}			19			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note 4				25	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM numbers (g = 0, 3, 5, 7)

2. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

(4) During communication at same potential (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(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
SCKp cycle time Note 5	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V	20 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}			ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	16 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		6/f_{MCK} and 500			ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2 – 7			ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2 – 8			ns
		2.4 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
		2.4 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK}+30			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}	2.7 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK}+31			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK}+31			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	2.7 V ≤ V_{DD} ≤ 5.5 V			2/f_{MCK}+44	ns
			2.4 V ≤ V_{DD} ≤ 5.5 V			2/f_{MCK}+75	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SOp output lines.

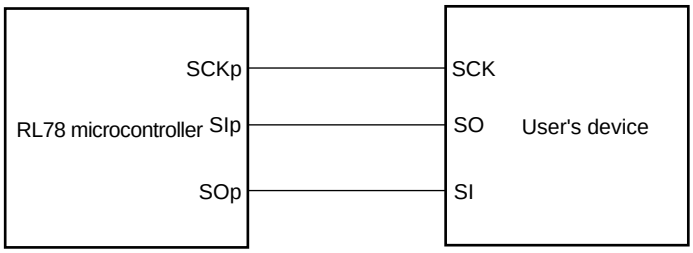
5. Transfer rate in the SNOOZE mode : MAX. 1 Mbps

Caution Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

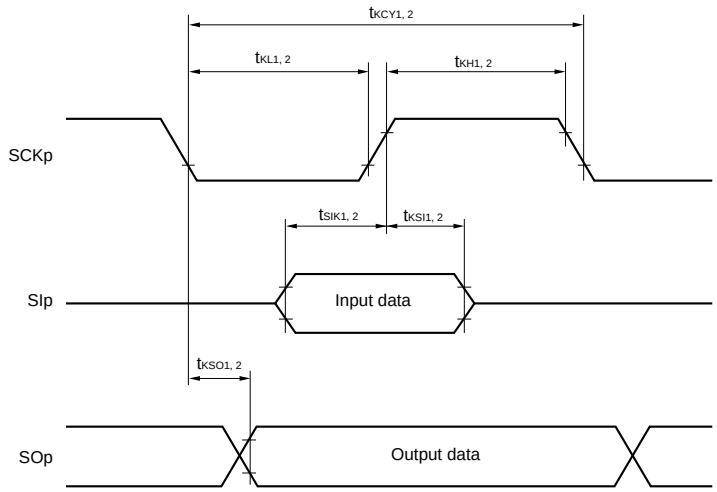
Remarks 1. p: CSI number (p = 00, 01), m: Unit number (m = 0),
n: Channel number (n = 0, 1), g: PIM number (g = 0, 3, 5, 7)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

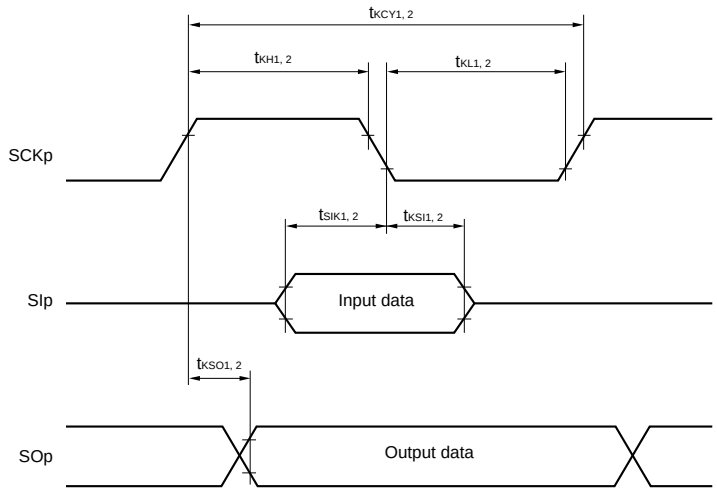
Simplified SPI (CSI) mode connection diagram (during communication at same potential)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00, 01)

2. m: Unit number, n: Channel number (mn = 00, 01)

(5) During communication at same potential (simplified I²C mode)
(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

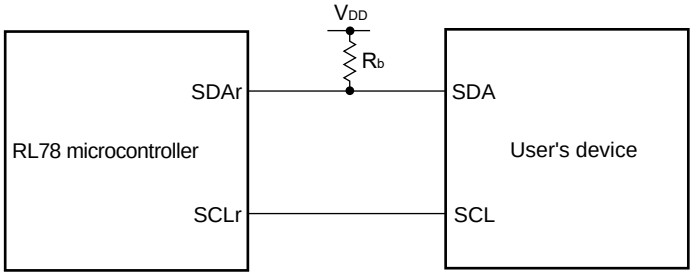
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1	kHz
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		400 Note 1	kHz
		2.4 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		300 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		ns
		2.4 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		ns
Hold time when SCLr = "H"	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		ns
		2.4 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 85 Note 2		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	$1/f_{MCK}$ + 145 Note 2		ns
		2.4 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	$1/f_{MCK}$ + 230 Note 2		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	ns
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	355	ns
		2.4 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	0	405	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".

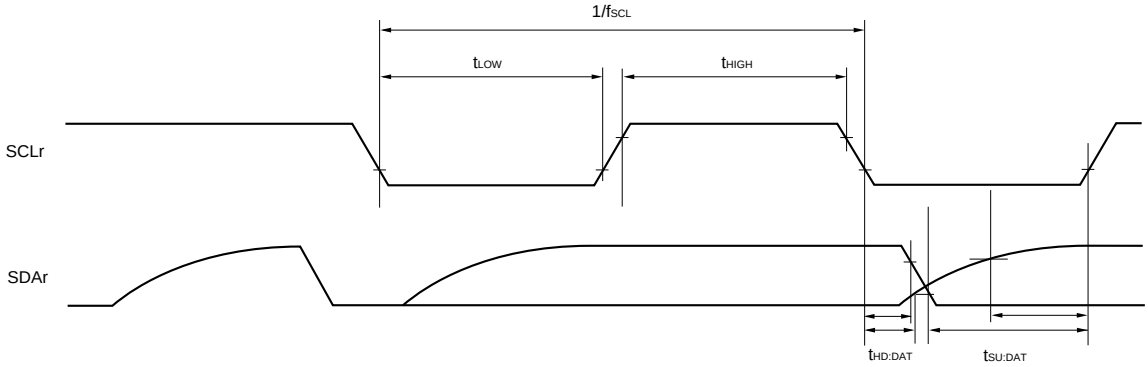
Caution Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).

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

Simplified I²C mode mode connection diagram (during communication at same potential)



Simplified I²C mode serial transfer timing (during communication at same potential)



Remarks 1. R_b[Ω]:Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance

2. r: IIC number (r = 00, 01), g: PIM number (g = 5), h: POM number (h = 3, 5)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01)

(6) Communication at different potential (2.5 V, 3 V) (UART mode) (1/2)
(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
Transfer rate		reception	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V				$f_{MCK}/6$ Note 1	bps
				Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Note 2			4.0	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V				$f_{MCK}/6$ Note 1	bps
				Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Note 2			4.0	Mbps
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V				$f_{MCK}/6$ Note 1	bps
				Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Note 2			4.0	Mbps

Notes 1. Use it with $V_{DD} \geq V_b$.

2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remarks 1. V_b[V]: Communication line voltage

2. q: UART number (q = 0), g: PIM and POM number (g = 5)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00)

(6) Communication at different potential (2.5 V, 3 V) (UART mode) (2/2)
(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
Transfer rate		transmission	4.0 V ≤ V_{DD} ≤ 5.5 V,				**Note 1**	bps
			2.7 V ≤ V_b ≤ 4.0 V	Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V			2.8 Note 2	Mbps
			2.7 V ≤ V_{DD} < 4.0 V				**Note 3**	bps
			2.3 V ≤ V_b ≤ 2.7 V	Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V			1.2 Note 4	Mbps
			2.4 V ≤ V_{DD} < 3.3 V				**Notes 5, 6**	bps
			1.6 V ≤ V_b ≤ 2.0 V	Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V			0.43 Note 7	Mbps

Notes 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 4.0 V ≤ V_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

2. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to Note 1 above to calculate the maximum transfer rate under conditions of the customer.

3. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

4. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to Note 3 above to calculate the maximum transfer rate under conditions of the customer.

5. Use it with V_{DD} ≥ V_b.

Notes 6. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.4 V ≤ V_{DD} < 3.3 V and 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

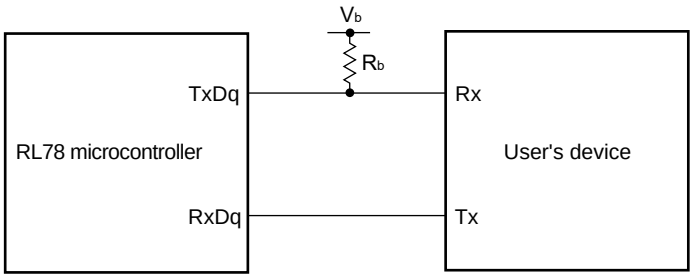
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

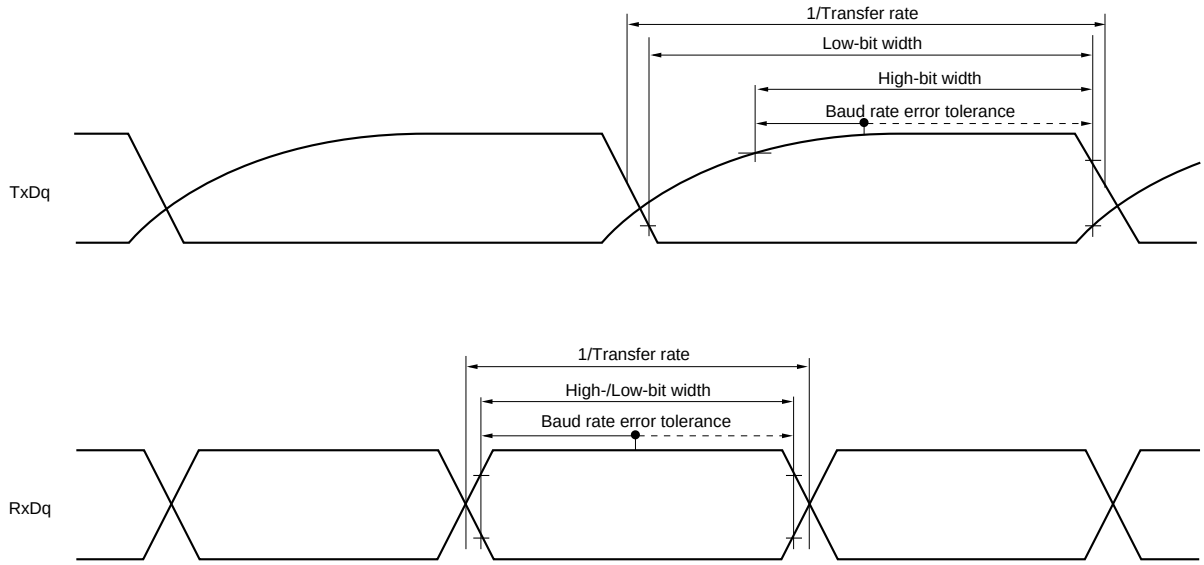
7. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to Note 6 above to calculate the maximum transfer rate under conditions of the customer.

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Remarks 1. R_b[Ω]:Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

2. q: UART number (q = 0), g: PIM and POM number (g = 5)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00))

(7) Communication at different potential (2.5 V, 3 V) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)

(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
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 2/f_{CLK}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	200			ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	300			ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 50			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 120			ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 7			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 10			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		58			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		121			ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10			ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ				60	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ				130	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		23			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		33			ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10			ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ				10	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ				10	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

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

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remarks 1. R_b[Ω]:Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM number (g = 3, 5)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00)

4. This value is valid only when CSI00's peripheral I/O redirect function is not used.

(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output) (1/2)

(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
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	300			ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500			ns
			2.4 V ≤ V_{DD} < 3.3 V, 2.4 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	1150			ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 75			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 170			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 – 458			ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 12			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 18			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 – 50			ns

Cautions 1. Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

2. Use it with V_{DD} ≥ V_b.

(**Remarks** are listed two pages after the next page.)

(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output) (2/2)

(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
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	81			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	177			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	479			ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	19			ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			100	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			195	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ			483	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	44			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	44			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	110			ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	19			ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			25	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			25	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ			25	ns

(**Notes**, **Cautions** and **Remarks** are listed on the next page.)

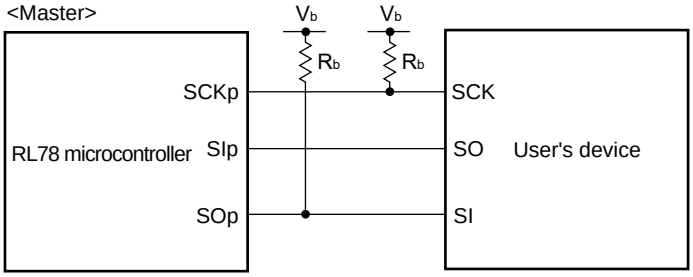
Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3 Use it with $V_{DD} \geq V_b$.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Simplified SPI (CSI) mode connection diagram (during communication at different potential)



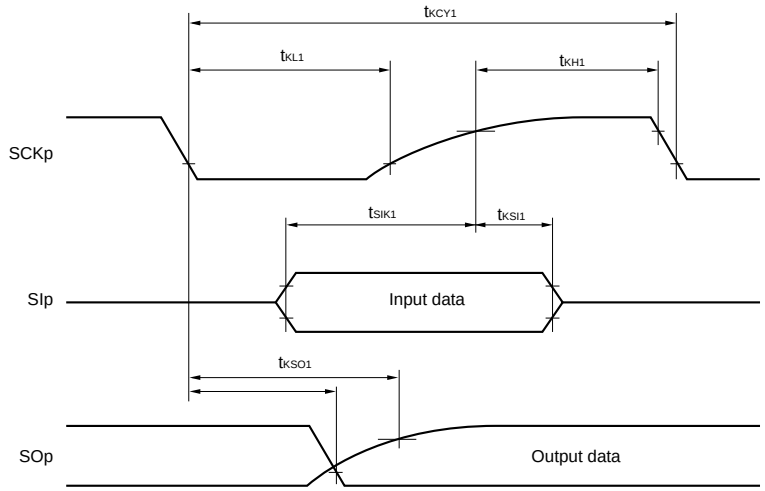
Remarks 1. R_b[Ω]:Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

2. p: CSI number (p = 00), m: Unit number, n: Channel number (mn = 00), g: PIM and POM number (g = 0, 3, 5, 7)

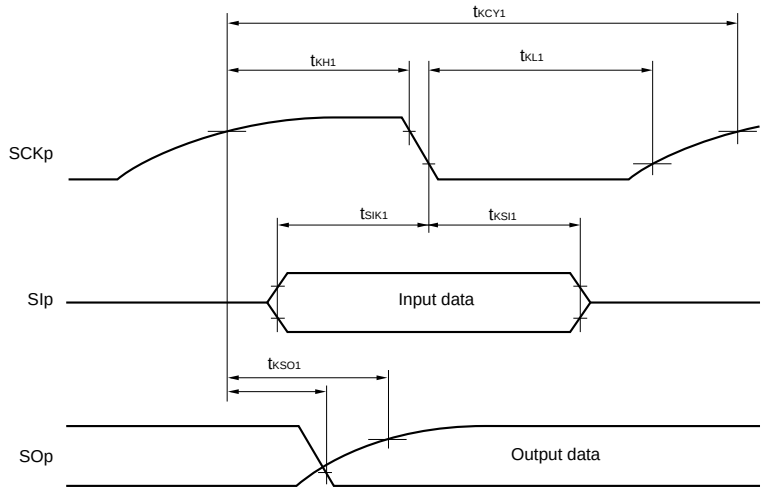
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00))

4. CSI01 cannot communicate at different potential. Use other CSI for communication at different potential.

Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00), m: Unit number, n: Channel number (mn = 00), g: PIM and POM number (g = 0, 3, 5, 7)

2. CSI01 cannot communicate at different potential. Use other CSI for communication at different potential.

(9) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(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
SCKp cycle time Note 1	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	20 MHz < f_{MCK} ≤ 24 MHz	12/f_{MCK}			ns
			8 MHz < f_{MCK} ≤ 20 MHz	10/f_{MCK}			ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	20 MHz < f_{MCK} ≤ 24 MHz	16/f_{MCK}			ns
			16 MHz < f_{MCK} ≤ 20 MHz	14/f_{MCK}			ns
			8 MHz < f_{MCK} ≤ 16 MHz	12/f_{MCK}			ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2	20 MHz < f_{MCK} ≤ 24 MHz	36/f_{MCK}			ns
			16 MHz < f_{MCK} ≤ 20 MHz	32/f_{MCK}			ns
			8 MHz < f_{MCK} ≤ 16 MHz	26/f_{MCK}			ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}			ns
			f_{MCK} ≤ 4 MHz	10/f_{MCK}			ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 – 12			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 – 18			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2		t_{KCY2}/2 – 50			ns
SIp setup time (to SCKp↑) Note 3	t_{SIK2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		1/f_{MCK} + 20			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		1/f_{MCK} + 20			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2		1/f_{MCK} + 30			ns
SIp hold time (from SCKp↑) Note 4	t_{KSI2}			1/f_{MCK} + 31			ns
Delay time from SCKp↓ to SOp output Note 5	t_{KSO2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ				2/f_{MCK} + 120	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ				2/f_{MCK} + 214	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 30 pF, R_b = 5.5 kΩ				2/f_{MCK} + 573	ns

Notes 1. Transfer rate in the SNOOZE mode : MAX. 1 Mbps

2. Use it with V_{DD} ≥ V_b.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

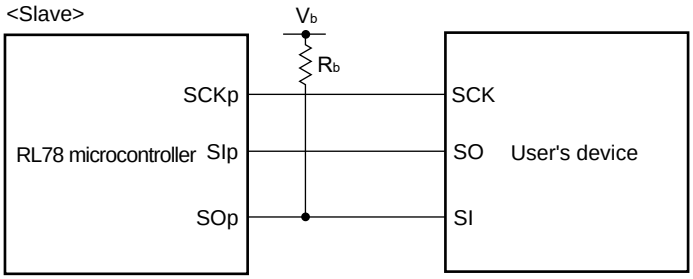
4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

5. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

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

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Simplified SPI (CSI) mode connection diagram (during communication at different potential)



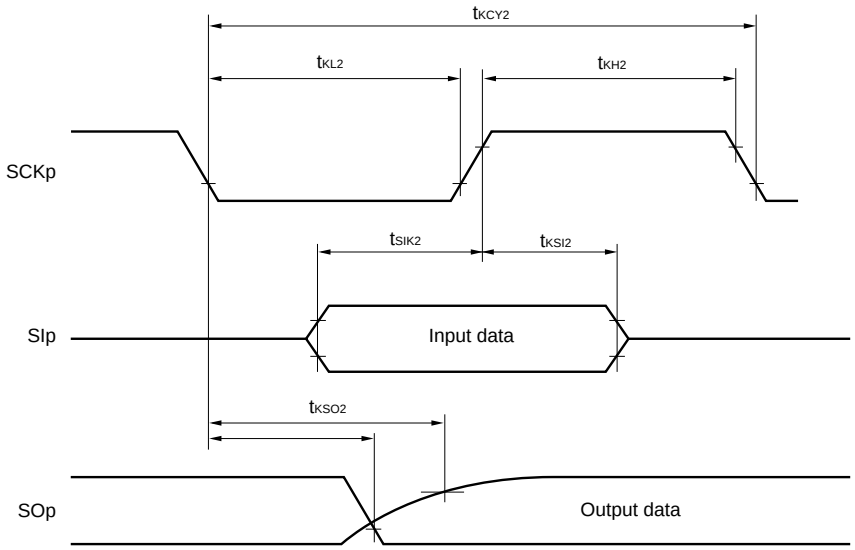
Remarks 1. R_b[Ω]:Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

2. p: CSI number (p = 00), m: Unit number, n: Channel number (mn = 00), g: PIM and POM number (g = 0, 3, 5, 7)

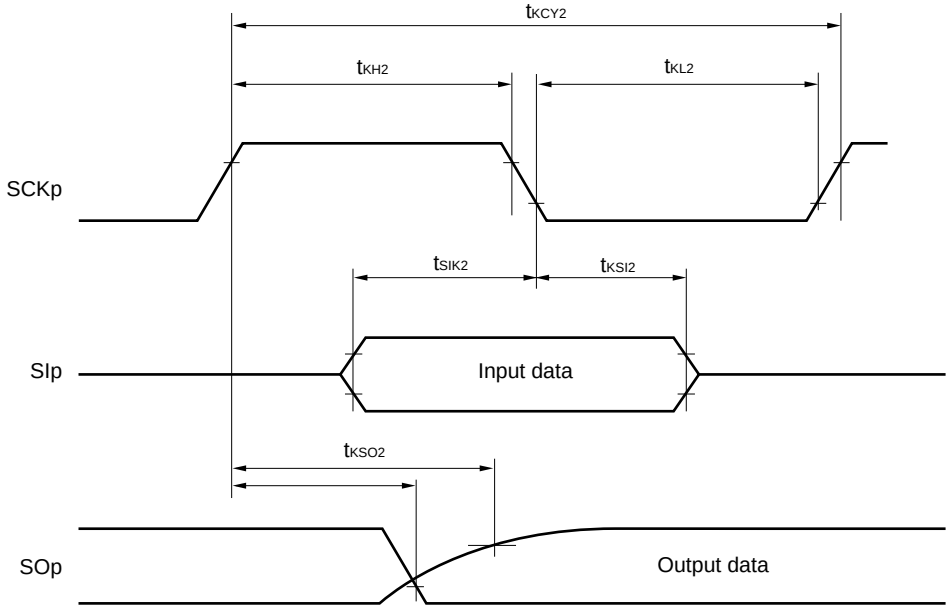
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00))

4. CSI01 cannot communicate at different potential. Use other CSI for communication at different potential.

Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00), m: Unit number, n: Channel number (mn = 00),
g: PIM and POM number (g = 0, 3, 5, 7)

2. CSI01 cannot communicate at different potential. Use other CSI for communication at different potential.

(10) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I²C mode) (1/2)

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1	kHz
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		400 Note 1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ		300 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	475		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	475		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1150		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1150		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	1550		ns
Hold time when SCLr = "H"	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	245		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	200		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	675		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	600		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	610		ns

(**Notes**, **Caution** and **Remarks** are listed on the next page.)

(10) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I²C mode) (2/2)

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 135 Note 3		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 135 Note 3		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1/f_{MCK} + 190 Note 3		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1/f_{MCK} + 190 Note 3		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Notes 2, C_b = 100 pF, R_b = 5.5 kΩ	1/f_{MCK} + 190 Note 3		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	355	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	355	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	0	405	ns

Notes 1. The value must also be equal to or less than f_{MCK}/4.

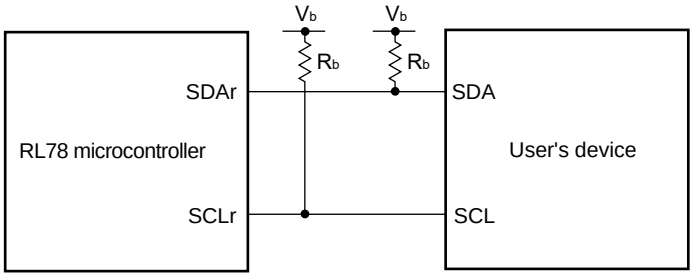
2. Use it with V_{DD} ≥ V_b.

3. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

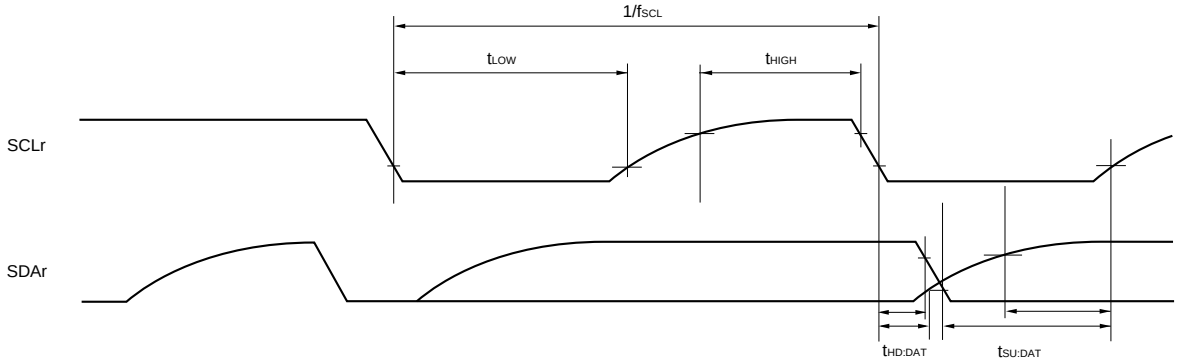
Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(**Remarks** are listed on the next page.)

Simplified I²C mode connection diagram (during communication at different potential)



Simplified I²C mode serial transfer timing (during communication at different potential)



Remarks 1. R_b[Ω]:Communication line (SDAr, SCLr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance, V_b[V]: Communication line voltage

2. r: IIC number (r = 00), g: PIM, POM number (g = 0, 3, 5, 7)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00)

30.5.2 Serial interface IICA

(1) I²C standard mode

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		Unit
				MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Standard mode: f_{CLK} ≥ 1 MHz	2.7 V ≤ V_{DD} ≤ 5.5 V	0	100	kHz
			2.4 V ≤ V_{DD} ≤ 5.5 V	0	100	kHz
Setup time of restart condition	$t_{SU:STA}$	2.7 V ≤ V_{DD} ≤ 5.5 V		4.7		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		4.7		μs
Hold time[Note 1]	$t_{HD:STA}$	2.7 V ≤ V_{DD} ≤ 5.5 V		4.0		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		4.0		μs
Hold time when SCLA0 = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 5.5 V		4.7		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		4.7		μs
Hold time when SCLA0 = "H"	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 5.5 V		4.0		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		4.0		μs
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V		250		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		250		μs
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V		0	3.45	μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		0	3.45	μs
Setup time of stop condition	$t_{SU:STO}$	2.7 V ≤ V_{DD} ≤ 5.5 V		4.0		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		4.0		μs
Bus-free time	t_{BUF}	2.7 V ≤ V_{DD} ≤ 5.5 V		4.7		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		4.7		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 $\overline{ACK}$ (acknowledge) timing.

Caution The values in the above table are applied even when bit 1 (PIOR1) in the peripheral I/O redirection register (PIOR) is 1. At this time, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.

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

(2) I²C fast mode

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz	2.7 V ≤ V_{DD} ≤ 5.5 V	0	400	kHz
			2.4 V ≤ V_{DD} ≤ 5.5 V	0	400	kHz
Setup time of restart condition	$t_{SU:STA}$	2.7 V ≤ V_{DD} ≤ 5.5 V		0.6		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		0.6		μs
Hold time[Note 1]	$t_{HD:STA}$	2.7 V ≤ V_{DD} ≤ 5.5 V		0.6		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		0.6		μs
Hold time when SCLA0 = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 5.5 V		1.3		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		1.3		μs
Hold time when SCLA0 = "H"	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 5.5 V		0.6		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		0.6		μs
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V		100		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		100		ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V		0	0.9	μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		0	0.9	μs
Setup time of stop condition	$t_{SU:STO}$	2.7 V ≤ V_{DD} ≤ 5.5 V		0.6		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		0.6		μs
Bus-free time	t_{BUF}	2.7 V ≤ V_{DD} ≤ 5.5 V		1.3		μs
		2.4 V ≤ V_{DD} ≤ 5.5 V		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 $\overline{ACK}$ (acknowledge) timing.

Caution The values in the above table are applied even when bit 1 (PIOR1) in the peripheral I/O redirection register (PIOR) is 1. At this time, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.

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.

Fast mode: C_b = 320 pF, R_b = 1.1 kΩ

(3) I²C fast mode plus

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode plus: f_{CLK} ≥ 10 MHz	2.7 V ≤ V_{DD} ≤ 5.5 V	0	1000	kHz
Setup time of restart condition	$t_{SU:STA}$	2.7 V ≤ V_{DD} ≤ 5.5 V		0.26		μs
Hold time[Note 1]	$t_{HD:STA}$	2.7 V ≤ V_{DD} ≤ 5.5 V		0.26		μs
Hold time when SCLA0 = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 5.5 V		0.5		μs
Hold time when SCLA0 = "H"	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 5.5 V		0.26		μs
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V		50		ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V		0	0.45	μs
Setup time of stop condition	$t_{SU:STO}$	2.7 V ≤ V_{DD} ≤ 5.5 V		0.26		μs
Bus-free time	t_{BUF}	2.7 V ≤ V_{DD} ≤ 5.5 V		0.5		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

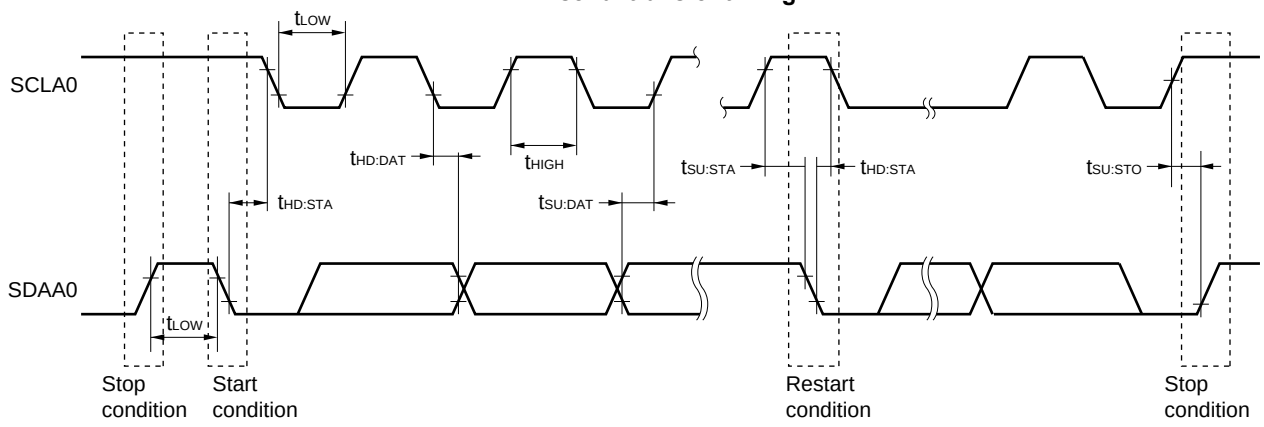
2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a clock stretch state is inserted in the $\overline{ACK}$ (acknowledge) timing.

Caution The values in the above table are applied even when bit 1 (PIOR1) in the peripheral I/O redirection register (PIOR) is 1. At this time, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.

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.

Fast mode plus: C_b = 120 pF, R_b = 1.1 kΩ

IICA serial transfer timing



30.5.3 USB

(1) Electrical specifications

(T_A = –40 to +85°C, 3.0 V ≤ UV_{DD} ≤ 3.6 V, 3.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
UV_{DD}	UV_{DD} input voltage characteristic	UV_{DD}	V_{DD} = 3.0 to 5.5 V, PXXCON = 1, VDDUSEB = 0 (UV_{DD} ≤ V_{DD})	3.0	3.3	3.6	V
	UV_{DD} output voltage characteristic	UV_{DD}	V_{DD} = 4.0 to 5.5 V, PXXCON = VDDUSEB = 1	3.0	3.3	3.6	V
UV_{BUS}	UV_{BUS} input voltage characteristic	UV_{BUS}	Function	4.35 (4.02[Note])	5.00	5.25	V
			Host	4.75	5.00	5.25	V

Note Value of instantaneous voltage

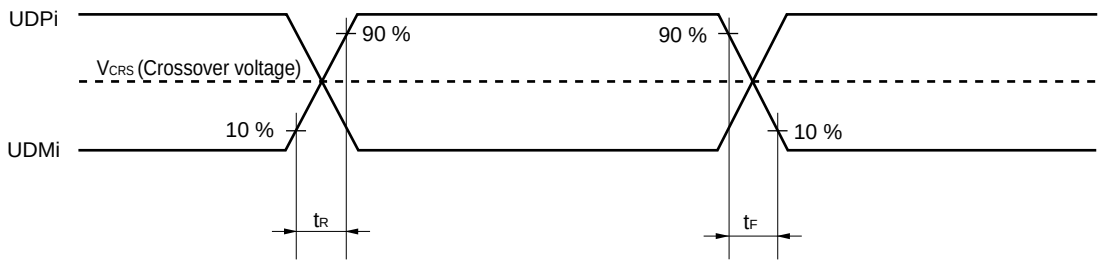
(T_A = –40 to +85°C, 3.0 V ≤ UV_{DD} ≤ 3.6 V, 3.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter			Symbol	Conditions	MIN.	TYP.	MAX.	Unit
UDPi/UDMi pins input characteristic (FS/LS receiver)	Input voltage		V_{IH}		2.0			V
			V_{IL}				0.8	V
	Difference input sensitivity		V_{DI}	| UDP voltage – UDM voltage |	0.2			V
	Difference common mode range		V_{CM}		0.8		2.5	V
UDPi/UDMi pins output characteristic (FS driver)	Output voltage		V_{OH}	I_{OH} = –200 μA	2.8		3.6	V
			V_{OL}	I_{OL} = 2.4 mA	0		0.3	V
	Transition time	Rising	t_{FR}	Rising: From 10% to 90 % of amplitude, Falling: From 90% to 10 % of amplitude, CL = 50 pF	4		20	ns
		Falling	t_{FF}		4		20	ns
	Matching (TFR/TFF)		V_{FRFM}		90		111.1	%
	Crossover voltage		V_{FCRS}		1.3		2.0	V
	Output Impedance		Z_{DRV}	UV_{DD} voltage = 3.3 V, Pin voltage = 1.65 V	28		44	Ω
UDPi/UDMi pins output characteristic (LS driver)	Output voltage		V_{OH}		2.8		3.6	V
			V_{OL}		0		0.3	V
	Transition time	Rising	t_{LR}	Rising: From 10% to 90 % of amplitude, Falling: From 90% to 10 % of amplitude, CL = 200 to 600 pF When the host controller function is selected: The UDMi pin (i = 0, 1) is pulled up via 1.5 kΩ. When the function controller function is selected: The UDP0 and UDM0 pins are individually pulled down via 15 kΩ	75		300	ns
		Falling	t_{LF}		75		300	ns
	Matching (TFR/TFF) Note		V_{LTFM}		80		125	%
	Crossover voltage Note		V_{LCRS}		1.3		2.0	V
UDPi/UDMi pins pull-up, pull-down	Pull-down resistor		R_{PD}		14.25		24.80	kΩ
	Pull-up resistor (i = 0 only)	Idle	R_{PUI}		0.9		1.575	kΩ
		Reception	R_{PUA}		1.425		3.09	kΩ
UV_{BUS}	UV_{BUS} pull-down resistor		R_{VBUS}	UV_{BUS} voltage = 5.5 V		1000		kΩ
	UV_{BUS} input voltage		V_{IH}		3.20			V
			V_{IL}				0.8	V

Note Excludes the first signal transition from the idle state.

Remark i = 0, 1

Timing of UDPi and UDMi



(2) BC standard

(T_A = –40 to +85°C, 3.0 V ≤ UV_{DD} ≤ 3.6 V, 3.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
USB standard BC1.2	UDPi sink current	I_{DP_SINK}		25		175	μA
	UDMi sink current	I_{DM_SINK}		25		175	μA
	DCD source current	I_{DP_SRC}		7		13	μA
	Dedicated charging port resistor	R_{DCP_DAT}	0 V < UDP/UDM voltage < 1.0 V			200	Ω
	Data detection voltage	V_{DAT_REF}		0.25		0.4	V
	UDPi source voltage	V_{DP_SRC}	Output current 250 μA	0.5		0.7	V
	UDMi source voltage	V_{DM_SRC}	Output current 250 μA	0.5		0.7	V

Remark i = 0, 1

(3) BC option standard (Host)

(T_A = –40 to +85°C, 4.75 V ≤ UV_{BUS} ≤ 5.25 V, 3.0 V ≤ UV_{DD} ≤ 3.6 V, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter			Symbol	Conditions	MIN.	TYP.	MAX.	Unit
UDPi output voltage (UV_{BUS} divider ratio) • VDOUEi = 1	VDSELi [3:0] (i = 0, 1)	1000	V_{P20}		38	40	42	% UV_{BUS}
		1001	V_{P27}		51.6	53.6	55.6	% UV_{BUS}
		1010	V_{P20}		38	40	42	% UV_{BUS}
		1100	V_{P33}		60	66	72	% UV_{BUS}
UDMi output voltage (UV_{BUS} divider ratio) • VDOUEi = 1	VDSELi [3:0] (i = 0, 1)	1000	V_{M20}		38	40	42	% UV_{BUS}
		1001	V_{M20}		38	40	42	% UV_{BUS}
		1010	V_{M27}		51.6	53.6	55.6	% UV_{BUS}
		1100	V_{M33}		60	66	72	% UV_{BUS}
UDPi comparing voltage Note 1 (UV_{BUS} divider ratio) • VDOUEi = 1 • CUSDETEi = 1	VDSELi [3:0] (i = 0, 1)	1000	V_{HDETP_UP0}	The rise of pin voltage detection voltage	56.2			% UV_{BUS}
			V_{HDETP_DWN0}	The fall of pin voltage detection voltage			29.4	% UV_{BUS}
		1001	V_{HDETP_UP1}	The rise of pin voltage detection voltage	60.5			% UV_{BUS}
			V_{HDETP_DWN1}	The fall of pin voltage detection voltage			45.0	% UV_{BUS}
		1010	V_{HDETP_UP2}	The rise of pin voltage detection voltage	56.2			% UV_{BUS}
			V_{HDETP_DWN2}	The fall of pin voltage detection voltage			29.4	% UV_{BUS}
UDMi comparing voltage Note 1 (UV_{BUS} divider ratio) • VDOUEi = 1 • CUSDETEi = 1	VDSELi [3:0] (i = 0, 1)	1000	V_{HDETM_UP0}	The rise of pin voltage detection voltage	56.2			% UV_{BUS}
			V_{HDETM_DWN0}	The fall of pin voltage detection voltage			29.4	% UV_{BUS}
		1001	V_{HDETM_UP1}	The rise of pin voltage detection voltage	56.2			% UV_{BUS}
			V_{HDETM_DWN1}	The fall of pin voltage detection voltage			29.4	% UV_{BUS}
		1010	V_{HDETM_UP2}	The rise of pin voltage detection voltage	60.5			% UV_{BUS}
			V_{HDETM_DWN2}	The fall of pin voltage detection voltage			45.0	% UV_{BUS}
UDPi pull-up detection Note 2 Connect detection with the full speed function (pull-up resistor)		1000	R_{HDET_PULL}	In full-speed mode, the power supply voltage range of pull-up resistors connected to the USB function module is between 3.0 V and 3.6 V.			1.575	kΩ
		1001						
		1010						
UDMi pull-up detection Note 2 Connect detection with the low-speed (pull-up resistor)		1000	R_{HDET_PULL}	In low-speed mode, the power supply voltage range of pull-up resistors connected to the USB function module is between 3.0 V and 3.6 V.			1.575	kΩ
		1001						
		1010						
UDMi sink current detection Note 2 Connect detection with the BC1.2 portable device (sink resistor)		1000	I_{HDET_SINK}		25			μA
		1001						
		1010						

Notes 1. If the voltage output from UDPi or UDMi (i = 0, 1) exceeds the range of the MAX and MIN values prescribed in this specification, DPCUSDETi (bit 8) and DMCUSDETi (bit 9) of the USBBCOPTi register are set to 1.

2. If the pull-up resistance or sink current prescribed in this specification is applied to UDPi or UDMi (i = 0, 1), DPCUSDETi (bit 8) and DMCUSDETi (bit 9) of the USBBCOPTi register are set to 1.

Remark i = 0, 1

(4) BC option standard (Function)

(T_A = –40 to +85°C, 4.35 V ≤ UV_{BUS} ≤ 5.25 V, 3.0 V ≤ UV_{DD} ≤ 3.6 V, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter			Symbol	Conditions	MIN.	TYP.	MAX.	Unit
UDPi/UDMi input reference voltage (UV_{BUS} divider ratio) • VDOUEi = 0 (i = 0))	VDSELi [3:0] (i = 0)	0000	V_{DDET0}		27	32	37	% UV_{BUS}
		0001	V_{DDET1}		29	34	39	% UV_{BUS}
		0010	V_{DDET2}		32	37	42	% UV_{BUS}
		0011	V_{DDET3}		35	40	45	% UV_{BUS}
		0100	V_{DDET4}		38	43	48	% UV_{BUS}
		0101	V_{DDET5}		41	46	51	% UV_{BUS}
		0110	V_{DDET6}		44	49	54	% UV_{BUS}
		0111	V_{DDET7}		47	52	57	% UV_{BUS}
		1000	V_{DDET8}		51	56	61	% UV_{BUS}
		1001	V_{DDET9}		55	60	65	% UV_{BUS}
		1010	V_{DDET10}		59	64	69	% UV_{BUS}
		1011	V_{DDET11}		63	68	73	% UV_{BUS}
		1100	V_{DDET12}		67	72	77	% UV_{BUS}
		1101	V_{DDET13}		71	76	81	% UV_{BUS}
		1110	V_{DDET14}		75	80	85	% UV_{BUS}
		1111	V_{DDET15}		79	84	89	% UV_{BUS}

30.6 Analog Characteristics

30.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Input channel	Reference Voltage		
	Reference voltage (+) = AV_{REFP} Reference voltage (–) = AV_{REFM}	Reference voltage (+) = V_{DD} Reference voltage (–) = V_{SS}	Reference voltage (+) = V_{BGR} Reference voltage (–) = AV_{REFM}
ANI0 to ANI7	Refer to **30.6.1 (1)**.	Refer to **30.6.1 (3)**.	Refer to **30.6.1 (4)**.
ANI16, ANI17, ANI19	Refer to **30.6.1 (2)**.		
Internal reference voltage Temperature sensor output voltage	Refer to **30.6.1 (1)**.		–

(1) When $AV_{REF(+)}$ = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin : ANI2 to ANI7, internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +85°C, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ V_{DD} ≤ 5.5 V		1.2	±3.5	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI2 to ANI7	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
		10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error[Notes 1, 2]	EZS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.25	%FSR
Full-scale error[Notes 1, 2]	EFS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.25	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
Differential linearity error [Note 1]	DLE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ V_{DD} ≤ 5.5 V			±1.5	LSB
Analog input voltage	V_{AIN}	ANI2 to ANI7		0		AV_{REFP}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{BGR} [Note 4]			V
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{TMPS25} [Note 4]			V

(**Notes** are listed on the next page.)

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When $AV_{REFP} < V_{DD}$, the MAX. values are as follows.

Overall error: Add ±1.0 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.

Zero-scale error/Full-scale error: Add ±0.05%FSR to the MAX. value when $AV_{REFP} = V_{DD}$.

Integral linearity error/ Differential linearity error: Add ±0.5 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.

4. Refer to **30.6.2 Temperature sensor/internal reference voltage characteristics**.

(2) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin : ANI16, ANI17, ANI19

(T_A = –40 to +85°C, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ V_{DD} ≤ 5.5 V		1.2	±5.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target ANI pin : ANI16, ANI17, ANI19	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error[Notes 1, 2]	EZS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.35	%FSR
Full-scale error[Notes 1, 2]	EFS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.35	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ V_{DD} ≤ 5.5 V			±3.5	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ V_{DD} ≤ 5.5 V			±2.00	LSB
Analog input voltage	V_{AIN}	ANI16, ANI17, ANI19		0		AV_{REFP} and V_{DD}	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When AV_{REFP} < V_{DD}, the MAX. values are as follows.

Overall error: Add ±4.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.

Zero-scale error/Full-scale error: Add ±0.20%FSR to the MAX. value when AV_{REFP} = V_{DD}.

Integral linearity error/ Differential linearity error: Add ±2.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.

(3) Reference voltage (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), Reference voltage (–) = V_{SS} (ADREFM = 0), target ANI pin : ANI0 to ANI7, ANI16, ANI17, ANI19, internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{DD}, Reference voltage (–) = V_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error[Notes 1, 2]	AINL	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V		1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target ANI pin : ANI0 to ANI7, ANI16, ANI17, ANI19	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
		10-bit resolution Target ANI pin : Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error[Notes 1, 2]	EZS	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
Full-scale error[Notes 1, 2]	EFS	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±4.0	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
Analog input voltage	V_{AIN}	ANI0 to ANI7, ANI16, ANI17, ANI19		0		V_{DD}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{BGR} [Note 3]			V
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{TMPS25} [Note 3]			V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. Refer to **30.6.2 Temperature sensor/internal reference voltage characteristics**.

(4) When Reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), Reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin : ANI0 to ANI7, ANI16, ANI17, ANI19

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{BGR} Note 3, Reference voltage (–) = AV_{REFM} Note 4 = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}				8		Bit
Conversion time	t_{CONV}	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	EZS	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
Integral linearity error Note 1	ILE	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
Differential linearity error Note 1	DLE	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±1.0	LSB
Analog input voltage	V_{AIN}			0		V_{BGR} Note 3	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. Refer to **30.6.2 Temperature sensor/internal reference voltage characteristics**.

4. When reference voltage (–) = V_{SS}, the MAX. values are as follows.

Zero-scale error: Add ±0.35%FSR to the MAX. value when reference voltage (–) = AV_{REFM}.

Integral linearity error: Add ±0.5 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.

Differential linearity error: Add ±0.2 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.

30.6.2 Temperature sensor/internal reference voltage characteristics

(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, HS (high-speed main) mode)

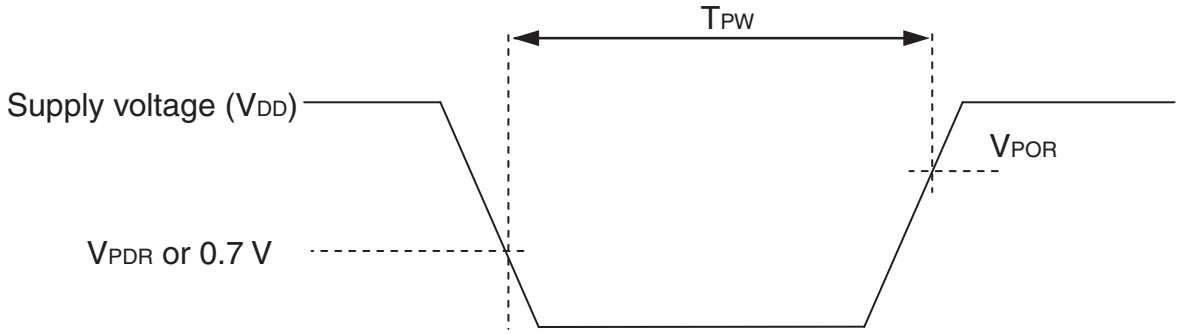
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGR}	Setting ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	F_{VTMPS}	Temperature sensor that depends on the temperature		–3.6		mV/°C
Operation stabilization wait time	t_{AMP}		5			μs

30.6.3 POR circuit characteristics

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.47	1.51	1.55	V
	V_{PDR}	Power supply fall time	1.46	1.50	1.54	V
Minimum pulse width[Note]	T_{PW}		300			μs

Note Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}. This is also the minimum time required for a POR reset from when V_{DD} exceeds below 0.7 V to when V_{DD} exceeds V_{POR} while STOP mode is entered or the main system clock (f_{MAIN}) is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



30.6.4 LVD circuit characteristics

LVD Detection Voltage of Reset Mode and Interrupt Mode

(T_A = –40 to +85°C, $V_{PDR} \leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD0}	Power supply rise time	3.98	4.06	4.14	V
			Power supply fall time	3.90	3.98	4.06	V
		V_{LVD1}	Power supply rise time	3.68	3.75	3.82	V
			Power supply fall time	3.60	3.67	3.74	V
		V_{LVD2}	Power supply rise time	3.07	3.13	3.19	V
			Power supply fall time	3.00	3.06	3.12	V
		V_{LVD3}	Power supply rise time	2.96	3.02	3.08	V
			Power supply fall time	2.90	2.96	3.02	V
		V_{LVD4}	Power supply rise time	2.86	2.92	2.97	V
			Power supply fall time	2.80	2.86	2.91	V
		V_{LVD5}	Power supply rise time	2.76	2.81	2.87	V
			Power supply fall time	2.70	2.75	2.81	V
		V_{LVD6}	Power supply rise time	2.66	2.71	2.76	V
			Power supply fall time	2.60	2.65	2.70	V
		V_{LVD7}	Power supply rise time	2.56	2.61	2.66	V
			Power supply fall time	2.50	2.55	2.60	V
		V_{LVD8}	Power supply rise time	2.45	2.50	2.55	V
			Power supply fall time	2.40	2.45	2.50	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time		t_{LD}				300	μs

LVD Detection Voltage of Interrupt & Reset Mode

(T_A = –40 to +85°C, $V_{PDR} \leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVDC0}	VPOC2, VPOC1, VPOC0 = 0, 1, 0, falling reset voltage			2.40	2.45	2.50	V
	V_{LVDC1}		LVIS1, LVIS0 = 1, 0	Rising release reset voltage	2.56	2.61	2.66	V
				Falling interrupt voltage	2.50	2.55	2.60	V
	V_{LVDC2}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.66	2.71	2.76	V
				Falling interrupt voltage	2.60	2.65	2.70	V
	V_{LVDC3}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.68	3.75	3.82	V
				Falling interrupt voltage	3.60	3.67	3.74	V
	V_{LVDD0}	VPOC2, VPOC1, VPOC0 = 0, 1, 1, falling reset voltage			2.70	2.75	2.81	V
	V_{LVDD1}		LVIS1, LVIS0 = 1, 0	Rising release reset voltage	2.86	2.92	2.97	V
				Falling interrupt voltage	2.80	2.86	2.91	V
	V_{LVDD2}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.96	3.02	3.08	V
				Falling interrupt voltage	2.90	2.96	3.02	V
	V_{LVDD3}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.98	4.06	4.14	V
				Falling interrupt voltage	3.90	3.98	4.06	V

30.6.5 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

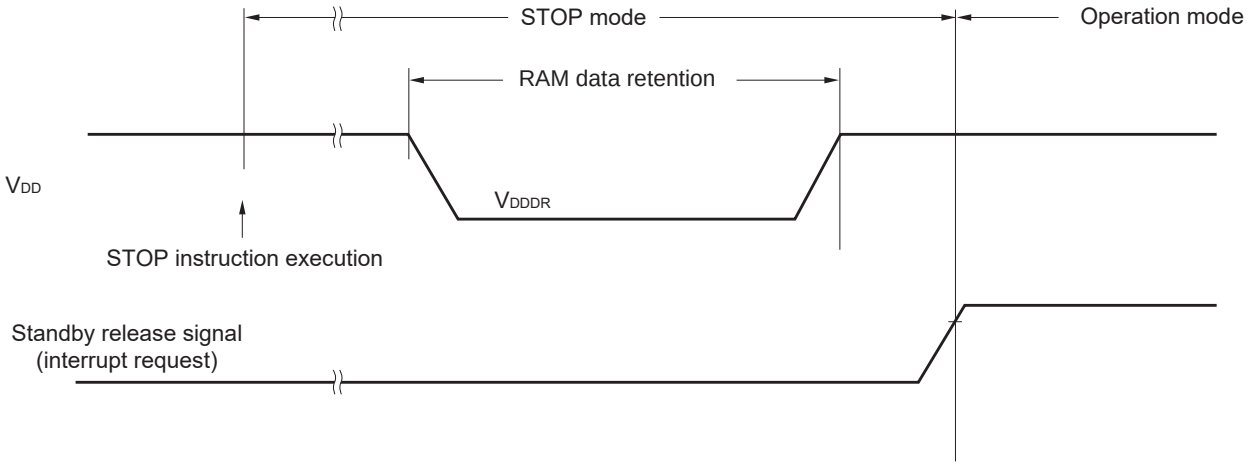
Caution Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 30.4 AC Characteristics.

30.7 RAM Data Retention Characteristics

(T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.46 Note		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is effected, but data is not retained when a POR reset is effected.



30.8 Flash Memory Programming Characteristics

(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
CPU/peripheral hardware clock frequency	f_{CLK}	2.4 V ≤ V_{DD} ≤ 5.5 V	1		24	MHz
Number of code flash rewrites	C_{erwr}	Retaining years: 20 years T_A = +85°C	1,000			Times
Number of data flash rewrites **Notes 1, 2, 3**		Retaining years: 1 year T_A = +25°C		1,000,000		
		Retaining years: 5 years T_A = +85°C	100,000			
		Retaining years: 20 years T_A = +85°C	10,000			

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 and Renesas Electronics self programming library.

3. These specifications show the characteristics of the flash memory and the results obtained from Renesas Electronics reliability testing.

30.9 Dedicated Flash Memory Programmer Communication (UART)

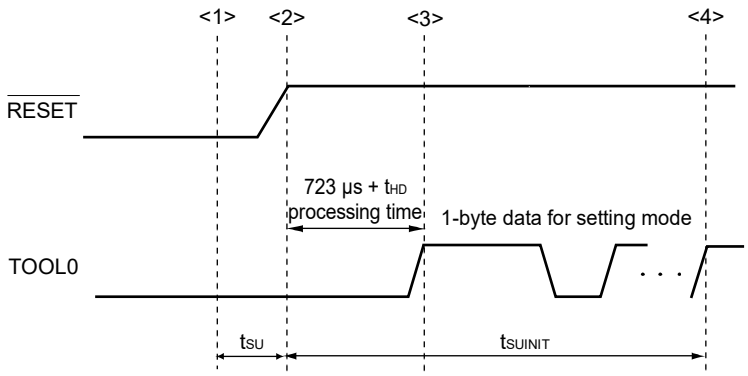
(T_A = –40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

30.10 Timing Specs for Switching Flash Memory Programming Modes

(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
How long from when an external reset ends until the initial communication settings are specified	t_{SUINIT}	POR and LVD reset must end before the external reset ends.			100	ms
How long from when the TOOL0 pin is placed at the low level until an external reset ends	t_{SU}	POR and LVD reset must end before the external reset ends.	10			μs
How long the TOOL0 pin must be kept at the low level after an external reset ends (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset must end before the external reset ends.	1			ms



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

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 and internal resets end (excluding the processing time of the firmware to control the flash memory)

CHAPTER 31 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)

This chapter describes the electrical specifications for the products "G: Industrial applications (T_A = -40 to +105°C)".

The target products G: Industrial applications ; T_A = -40 to +105°C
R5F10JBCGNA, R5F10JBCGFP, R5F10JGCGNA, R5F10JGCGFB,
R5F10KBCGNA, R5F10KBCGFP, R5F10KGCGNA, R5F10KGCGFB

Cautions 1. The RL78 microcontrollers 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.

2. The pins mounted depend on the product. Refer to 2.1 Port Function to 2.2.1 With functions for each product.

3. Please contact Renesas Electronics sales office for derating of operation under T_A = +85°C to +105°C. Derating is the systematic reduction of load for the sake of improved reliability.

There are following differences between the products "G: Industrial applications (T_A = -40 to +105°C)" and the products "A: Consumer applications".

Parameter	Application	
	A: Consumer applications	G: Industrial applications
Operating ambient temperature	T_A = -40 to +85°C	T_A = -40 to +105°C
High-speed on-chip oscillator clock accuracy	2.4 V ≤ V_{DD} ≤ 5.5 V ±1.0%@ T_A = -20 to +85°C ±1.5%@ T_A = -40 to -20°C	2.4 V ≤ V_{DD} ≤ 5.5 V ±2.0%@ T_A = +85 to +105°C ±1.0%@ T_A = -20 to +85°C ±1.5%@ T_A = -40 to -20°C
Serial array unit	UART Simplified SPI (CSI): f_{CLK}/2 (supporting 16 Mbps), f_{CLK}/4 Simplified I^2C communication	UART Simplified SPI (CSI): f_{CLK}/4 Simplified I^2C communication
IICA	Normal mode Fast mode Fast mode plus	Normal mode Fast mode

Remark The electrical characteristics of the products G: Industrial applications (T_A = -40 to +105°C) are different from those of the products "A: Consumer applications". For details, refer to **31.1** to **31.10**.

31.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		–0.5 to +6.5	V
REGC pin input voltage	V_{IREGC}	REGC	–0.3 to +2.8 and –0.3 to V_{DD} +0.3 Note 1	V
UV_{DD} pin input voltage	V_{IUVDD}	UV_{DD}	–0.3 to V_{DD} +0.3	V
Input voltage	V_{I1}	P00, P01, P14 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P70 to P75, P120 to P124, P137, P140, EXCLK, EXCLKS, $\overline{RESET}$	–0.3 to V_{DD} +0.3 Note 2	V
	V_{I2}	P60 to P63 (N-ch open-drain)	–0.3 to +6.5	V
	V_{I3}	UDP0, UDM0, UDP1, UDM1	–0.3 to +6.5	V
	V_{I4}	UV_{BUS}	–0.3 to +6.5	V
Output voltage	V_{O1}	P00, P01, P14 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P120, P130, P140	–0.3 to V_{DD} +0.3 Note 2	V
	V_{O2}	UDP0, UDM0, UDP1, UDM1	–0.3 to +6.5	V
Analog input voltage	V_{AI1}	ANI16, ANI17, ANI19	–0.3 to V_{DD} +0.3 and –0.3 to AV_{REF} (+) +0.3 Notes 2, 3	V
	V_{AI2}	ANI0 to ANI7	–0.3 to V_{DD} +0.3 and –0.3 to AV_{REF} (+) +0.3 Notes 2, 3	V

Notes 1. Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

2. Must be 6.5 V or lower.

3. Do not exceed AV_{REF}(+) + 0.3 V in case of A/D conversion target pin

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. AV_{REF} (+) : The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and V_{DD}.

3. V_{SS} : Reference voltage

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140	–40	mA
		Total of all pins –170 mA	P00, P01, P40, P41, P120, P130, P140	–70	mA
			P14 to P17, P30, P31, P50, P51, P70 to P75	–100	mA
	I_{OH2}	Per pin	P20 to P27	–0.5	mA
		Total of all pins		–2	mA
Output current, low	I_{OL1}	Per pin	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P120, P130, P140	40	mA
		Total of all pins 170 mA	P00, P01, P40, P41, P120, P130, P140	70	mA
			P14 to P17, P30, P31, P50, P51, P60 to P63, P70 to P75	100	mA
	I_{OL2}	Per pin	P20 to P27	1	mA
		Total of all pins		5	mA
Operating ambient temperature	T_A	In normal operation mode		–40 to +105	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			–65 to +150	°C

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.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

31.2 Oscillator Characteristics

31.2.1 X1, XT1 oscillator characteristics

(T_A = –40 to +105°C, 2.4 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	1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1.0		16.0	MHz
XT1 clock oscillation frequency (f_{XT})Note	Crystal resonator		32	32.768	35	kHz

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 and XT1 oscillator, refer to **5.4 System Clock Oscillator**.

31.2.2 On-chip oscillator characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Notes 1, 2	f_{HOCO}		1		48	MHz
High-speed on-chip oscillator clock frequency accuracy		–20 to +85 °C	–1.0		+1.0	%
		–40 to –20 °C	–1.5		+1.5	%
		+85 to +105 °C	–2.0		+2.0	%
Low-speed on-chip oscillator clock frequency	f_{IL}			15		kHz
Low-speed on-chip oscillator clock frequency accuracy			–15		+15	%

Notes 1. High-speed on-chip oscillator frequency is selected by bits 0 to 3 of option byte (000C2H/010C2H) and bits 0 to 2 of HOCODIV register.

2. This indicates the oscillator characteristics only. Refer to AC Characteristics for instruction execution time.

31.2.3 PLL oscillator characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
PLL input frequency Note	f_{PLLIN}	High-speed system clock	6.00		16.00	MHz
PLL output frequency Note	f_{PLL}			48.00		MHz
Lock up time		From PLL output enable to stabilization of the output frequency	40.00			μs
Interval time		From PLL stop to PLL re-operation setteing Wait time	4.00			μs
Setting wait time		From after PLL input clock stabilization and PLL setting is fixed to start setting Wait time required	1.00			μs

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

31.3 DC Characteristics

31.3.1 Pin characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140	2.4 V ≤ V_{DD} ≤ 5.5 V			–3.0 [Note 2]	mA
		Total of P00, P01, P40, P41, P120, P130, P140 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–30.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–10.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			–5.0	mA
		Total of P14 to P17, P30, P31, P50, P51, P70 to P75 (When duty ≤ 70% [Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			–30.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			–19.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			–10.0	mA
		Total of all pins (When duty ≤ 70%[Note 3])	2.4 V ≤ V_{DD} ≤ 5.5 V			-60.0	mA
	I_{OH2}	Per pin for P20 to P27	2.4 V ≤ V_{DD} ≤ 5.5 V			–0.1 [Note 2]	mA
		Total of all pins (When duty ≤ 70%[Note 3])	2.4 V ≤ V_{DD} ≤ 5.5 V			–1.5	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. However, do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty ratio 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

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.

Caution P00, P01, P30, and P74 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 pins.

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140	2.4V ≤ V_{DD} ≤ 5.5 V			8.5[Note 2]	mA
		Per pin for P60 to P63	2.4V ≤ V_{DD} ≤ 5.5 V			15.0[Note 2]	mA
		Total of P00, P01, P40, P41, P120, P130, P140 (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			40.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			15.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			9.0	mA
		Total of P14 to P17, P30, P31, P50, P51, P60 to P63, P70 to P75 (When duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V			40.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			35.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			20.0	mA
		Total of all pins (When duty ≤ 70%[Note 3])	2.4V ≤ V_{DD} ≤ 5.5 V			80.0	mA
	I_{OL2}	Per pin for P20 to P27	2.4V ≤ V_{DD} ≤ 5.5 V			0.4[Note 2]	mA
		Total of all pins (When duty ≤ 70%[Note 3])	2.4V ≤ V_{DD} ≤ 5.5 V			5.0	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

2. However, do not exceed the total current value.

3. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty ratio to n%).

- 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.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P140	Normal input buffer	0.8V_{DD}		V_{DD}	V
	V_{IH2}	P00, P01, P30, P50	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		V_{DD}	V
			TTL input buffer 3.3 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
			TTL input buffer 2.4 V ≤ V_{DD} < 3.3 V	1.5		V_{DD}	V
	V_{IH3}	P20 to P27		0.7V_{DD}		V_{DD}	V
	V_{IH4}	P60 to P63		0.7V_{DD}		6.0	V
	V_{IH5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0.8V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P140	Normal input buffer	0		0.2V_{DD}	V
	V_{IL2}	P00, P01, P30, P50	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.3 V ≤ V_{DD} < 4.0 V	0		0.5	V
			TTL input buffer 2.4 V ≤ V_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P20 to P27		0		0.3V_{DD}	V
	V_{IL4}	P60 to P63		0		0.3V_{DD}	V
	V_{IL5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0		0.2V_{DD}	V

Caution The maximum value of V_{IH} of pins P00, P01, P30, and P74 is V_{DD}, even in the N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –3.0 mA	V_{DD} – 0.7			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –2.0 mA	V_{DD} – 0.6			V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –1.5 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P27	2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OH2} = –100 μA	V_{DD} – 0.5			V
Output voltage, low	V_{OL1}	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P130, P140	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.6	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.6 mA			0.4	V
	V_{OL2}	P20 to P27	2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 400 μA			0.4	V
	V_{OL3}	P60 to P63	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 15.0 mA			2.0	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 5.0 mA			0.4	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.4	V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 2.0 mA			0.4	V

Caution P00, P01, P30, and P74 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 pins.

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00, P01, P14 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P120, P137, P140, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00, P01, P14 to P17, P20 to P27, P30, P31, P40, P41, P50, P51, P60 to P63, P70 to P75, P120, P137, P140, $\overline{\text{RESET}}$	$V_I = V_{SS}$				–1	μA
	I_{LIL2}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	$V_I = V_{SS}$	In input port or external clock input			–1	μA
				In resonator connection			–10	μA
On-chip pll-up resistance	R_U	P00, P01, P14 to P17, P30, P31, P40, P41, P50, P51, P70 to P75, P120, P140	$V_I = V_{SS}$, In input port		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

31.3.2 Supply current characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	HS (High-speed main) modffe Note 6	f_{HOCO} = 48 MHz f_{IH} = 24 MHz Note 3	Basic operation	V_{DD} = 5.0 V		1.7		mA
						V_{DD} = 3.0 V		1.7		mA
					Normal operation	V_{DD} = 5.0 V		3.7	5.8	mA
						V_{DD} = 3.0 V		3.7	5.8	mA
				f_{HOCO} = 24 MHz Note 5 f_{IH} = 12 MHz Note 3	Normal operation	V_{DD} = 5.0 V		2.3	3.4	mA
						V_{DD} = 3.0 V		2.3	3.4	mA
				f_{HOCO} = 12 MHz Note 5 f_{IH} = 6 MHz Note 3	Normal operation	V_{DD} = 5.0 V		1.6	2.2	mA
						V_{DD} = 3.0 V		1.6	2.2	mA
				f_{HOCO} = 6 MHz Note 5 f_{IH} = 3 MHz Note 3	Normal operation	V_{DD} = 5.0 V		1.2	1.6	mA
						V_{DD} = 3.0 V		1.2	1.6	mA
			HS (High-speed main) mode Note 6	f_{MX} = 20 MHz Note 2, V_{DD} = 5.0 V	Normal operation	Square wave input		3.0	4.9	mA
						Resonator connection		3.2	5.0	mA
				f_{MX} = 20 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		3.0	4.9	mA
						Resonator connection		3.2	5.0	mA
				f_{MX} = 10 MHz Note 2, V_{DD} = 5.0 V	Normal operation	Square wave input		1.9	2.9	mA
						Resonator connection		1.9	2.9	mA
				f_{MX} = 10 MHz Note 2, V_{DD} = 3.0 V	Normal operation	Square wave input		1.9	2.9	mA
						Resonator connection		1.9	2.9	mA
			HS (High-speed main) mode (PLL operation) Note 6	f_{PLL} = 48 MHz, f_{CLK} = 24 MHz Note 2	Normal operation	V_{DD} = 5.0 V		4.0	6.3	mA
						V_{DD} = 3.0 V		4.0	6.3	mA
				f_{PLL} = 48 MHz, f_{CLK} = 12 MHz Note 2	Normal operation	V_{DD} = 5.0 V		2.6	3.9	mA
						V_{DD} = 3.0 V		2.6	3.9	mA
				f_{PLL} = 48 MHz, f_{CLK} = 6 MHz Note 2	Normal operation	V_{DD} = 5.0 V		1.9	2.7	mA
						V_{DD} = 3.0 V		1.9	2.7	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 4 T_A = –40°C	Normal operation	Resonator connection		4.1	4.9	μA
						Square wave input		4.2	5.0	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +25°C	Normal operation	Square wave input		4.1	4.9	μA
						Resonator connection		4.2	5.0	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +50°C	Normal operation	Square wave input		4.2	5.5	μA
						Resonator connection		4.3	5.6	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +70°C	Normal operation	Square wave input		4.2	6.3	μA
						Resonator connection		4.3	6.4	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +85°C	Normal operation	Square wave input		4.8	7.7	μA
						Resonator connection		4.9	7.8	μA
				f_{SUB} = 32.768 kHz Note 4 T_A = +105°C	Normal operation	Square wave input		6.9	19.7	μA
						Resonator connection		7.0	19.8	μA

(**Notes** and **Remarks** are listed on the next page.)

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 following points apply in the HS (high-speed main) mode.

- The currents in the "TYP." column do not include the operating currents of the peripheral modules.
- The currents in the "MAX." column include the operating currents of the peripheral modules, except for those flowing into the A/D converter, LVD circuit, USB 2.0 host/function module, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the RTC.

2. When high-speed on-chip oscillator and subsystem clock are stopped.

3. When high-speed system clock and subsystem clock are stopped.

4. When high-speed on-chip oscillator and high-speed system clock are stopped. When AMPHS1 = 1 (Ultra-low power consumption oscillation). However, not including the current flowing into the RTC, 12-bit interval timer, and watchdog timer.

5. When Operating frequency setting of option byte = 48 MHz. When f_{HOCO} is divided by HOCODIV. When RDIV[1:0] = 00 (divided by 2: default).

6. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 24 MHz
2.4 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 16 MHz

Remarks 1. f_{HOCO}: High-speed on-chip oscillator clock frequency (Max. 48 MHz)

2. f_{IH}: Main system clock source frequency obtained by dividing the high-speed on-chip oscillator clock by 2, 4, or 8 (Max. 24 MHz)

3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

4. f_{PLL}: PLL oscillation frequency

5. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

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

7. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C.

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (High-speed main) mode Note 8	f_{HOCO} = 48 MHz, f_{IH} = 24 MHz Note 4	V_{DD} = 5.0 V		0.67	2.25	mA
					V_{DD} = 3.0 V		0.67	2.25	mA
				f_{HOCO} = 24 MHz Note 6, f_{IH} = 12 MHz Note 4	V_{DD} = 5.0 V		0.50	1.55	mA
					V_{DD} = 3.0 V		0.50	1.55	mA
				f_{HOCO} = 12 MHz Note 6, f_{IH} = 6 MHz Note 4	V_{DD} = 5.0 V		0.41	1.21	mA
					V_{DD} = 3.0 V		0.41	1.21	mA
				f_{HOCO} = 6 MHz Note 6, f_{IH} = 3 MHz Note 4	V_{DD} = 5.0 V		0.37	1.05	mA
					V_{DD} = 3.0 V		0.37	1.05	mA
			HS (High-speed main) mode Note 8	f_{MX} = 20 MHz Note 3, V_{DD} = 5.0 V	Square wave input		0.28	1.90	mA
					Resonator connection		0.45	2.00	mA
				f_{MX} = 20 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.28	1.90	mA
					Resonator connection		0.45	2.00	mA
				f_{MX} = 10 MHz Note 3, V_{DD} = 5.0 V	Square wave input		0.19	1.02	mA
					Resonator connection		0.26	1.10	mA
				f_{MX} = 10 MHz Note 3, V_{DD} = 3.0 V	Square wave input		0.19	1.02	mA
					Resonator connection		0.26	1.10	mA
			HS (High-speed main) mode (PLL operation) Note 8	f_{PLL} = 48 MHz, f_{CLK} = 24 MHz Note 3	V_{DD} = 5.0 V		0.91	2.74	mA
					V_{DD} = 3.0 V		0.91	2.74	mA
				f_{PLL} = 48 MHz, f_{CLK} = 12 MHz Note 3	V_{DD} = 5.0 V		0.85	2.31	mA
					V_{DD} = 3.0 V		0.85	2.31	mA
				f_{PLL} = 48 MHz, f_{CLK} = 6 MHz Note 3	V_{DD} = 5.0 V		0.82	2.07	mA
					V_{DD} = 3.0 V		0.82	2.07	mA
			Subsystem clock operation	f_{SUB} = 32.768 kHz Note 5, T_A = –40°C	Square wave input		0.25	0.57	μA
					Resonator connection		0.44	0.76	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = +25°C	Square wave input		0.30	0.57	μA
					Resonator connection		0.49	0.76	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = +50°C	Square wave input		0.33	1.17	μA
					Resonator connection		0.63	1.36	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = +70°C	Square wave input		0.46	1.97	μA
					Resonator connection		0.76	2.16	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = +85°C	Square wave input		0.97	3.37	μA
					Resonator connection		1.16	3.56	μA
				f_{SUB} = 32.768 kHz Note 5, T_A = +105°C	Square wave input		3.01	15.37	μA
					Resonator connection		3.20	15.56	μA
	I_{DD3}	STOP mode Note 7	T_A = –40°C				0.18	0.50	μA
			T_A = +25°C				0.23	0.50	μA
			T_A = +50°C				0.26	1.10	μA
			T_A = +70°C				0.29	1.90	μA
			T_A = +85°C				0.90	3.30	μA
			T_A = +105°C				2.94	15.30	μA

(**Notes** and **Remarks** are listed on the next page.)

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 following points apply in the HS (high-speed main) mode.

- The currents in the "TYP." column do not include the operating currents of the peripheral modules.
- The currents in the "MAX." column include the operating currents of the peripheral modules, except for those flowing into the A/D converter, LVD circuit, USB 2.0 host/function module, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the RTC.

In the STOP mode, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules.

2. During HALT instruction execution by flash memory.

3. When high-speed on-chip oscillator and subsystem clock are stopped.

4. When high-speed system clock and subsystem clock are stopped.

5. When high-speed on-chip oscillator and high-speed system clock are stopped. When RTCLPC = 1 and setting ultra-low current consumption (AMPHS1 = 1).

6. When Operating frequency setting of option byte = 48 MHz. When f_{HOCO} is divided by HOCODIV. When RDIV[1:0] = 00 (divided by 2: default).

7. Regarding the value for current to operate the subsystem clock in STOP mode, refer to that in HALT mode.

8. Relationship between operation voltage width, operation frequency of CPU and operation mode is as below.

HS (high-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 24 MHz
2.4 V ≤ V_{DD} ≤ 5.5 V@1 MHz to 16 MHz

Remarks 1. f_{HOCO}: High-speed on-chip oscillator clock frequency (Max. 48 MHz)

2. f_{IH}: Main system clock source frequency obtained by dividing the high-speed on-chip oscillator clock by 2, 4, or 8 (Max. 24 MHz)

3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

4. f_{PLL}: PLL oscillation frequency

5. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

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

7. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C.

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.20		μA
RTC operating current	I_{RTC} Notes 1, 2, 3				0.02		μA
12-bit interval timer operating current	I_{IT} Notes 1, 2, 4				0.02		μA
Watchdog timer operating current	I_{WDT} Notes 1, 2, 5	f_{IL} = 15 kHz			0.22		μA
A/D converter operating current	I_{ADC} Notes 1, 6	When conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		1.3	1.8	mA
			Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.5	0.8	mA
A/D converter reference voltage current	I_{ADREF} Note 1				75.0		μA
Temperature sensor operating current	I_{TMPS} Note 1				75.0		μA
LVD operating current	I_{LVD} Notes 1, 7				0.08		μA
Self-programming operating current	I_{FSP} Notes 1, 9				2.00	12.30	mA
BGO operating current	I_{BGO} Notes 1, 8				2.00	12.30	mA
SNOOZE operating current	I_{SNOZ} Note 1	ADC operation	The mode is performed Note 10		0.80	1.97	mA
			The A/D conversion operations are performed, Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		1.20	3.00	mA
		Simplified SPI (CSI) operation			0.70	1.56	mA

(**Notes** and **Remarks** are listed on the next page.)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
USB operating current	I_{USBH} Note 11	During USB communication operation under the following settings and conditions (V_{DD} = 5.0 V, T_A = +25°C): • The internal power supply for the USB is used. • X1 oscillation frequency (f_X) = 12 MHz, PLL oscillation frequency (f_{PLL}) = 48 MHz • The host controller (via two ports) is set to operate in full-speed mode with four pipes (end points) used simultaneously. (PIPE4: Bulk OUT transfer (64 bytes), PIPE5: Bulk IN transfer (64 bytes), PIPE6: Interrupt OUT transfer, PIPE7: Interrupt IN transfer). • The USB ports (two ports) are individually connected to a peripheral function via a 0.5 m USB cable.		9.0		mA
	I_{USBF} Note 11	During USB communication operation under the following settings and conditions (V_{DD} = 5.0 V, T_A = +25°C): • The internal power supply for the USB is used. • X1 oscillation frequency (f_X) = 12 MHz, PLL oscillation frequency (f_{PLL}) = 48 MHz • The function controller is set to operate in full-speed mode with four pipes (end points) used simultaneously. (PIPE4: Bulk OUT transfer (64 bytes), PIPE5: Bulk IN transfer (64 bytes), PIPE6: Interrupt OUT transfer, PIPE7: Interrupt IN transfer). • The USB port (one port) is connected to the host device via a 0.5 m USB cable.		2.5		mA
	I_{SUSP} Note 12	During suspended state under the following settings and conditions (V_{DD} = 5.0 V, T_A = +25°C): • The function controller is set to full-speed mode (the UDP0 pin is pulled up). • The internal power supply for the USB is used. • The system is set to STOP mode (When the high-speed on-chip oscillator, high-speed system clock, and subsystem clock are stopped. When the watchdog timer is stopped.). • The USB port (one port) is connected to the host device via a 0.5 m USB cable.		240		μA

(**Notes** and **Remarks** are listed on the next page.)

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 real-time clock (RTC) (excluding the operating current of the low-speed on-chip ocsillator and the XT1 oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time clock operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added. I_{DD2} subsystem clock operation includes the operational current of the real-time clock.

4. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip ocsillator and the XT1 oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{IT}, when the 12-bit interval timer operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added.

5. Current flowing only to the watchdog timer (including 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_{WDT} when the watchdog timer is in operation.

6. 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.

7. Current flowing only to the LVD circuit. The current value of the RL78/G1C is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVD circuit operates in the Operating, HALT or STOP mode.

8. Current flowing only during data flash rewrite.

9. Current flowing only during self programming.

10. For shift time to the SNOOZE mode, see **19.3.3 SNOOZE mode.**

11. Current consumed only by the USB module and the internal power supply for the USB.

12. Includes the current supplied from the pull-up resistor of the UDP0 pin to the pull-down resistor of the host device, in addition to the current consumed by this MCU during the suspended state.

Remarks 1. f_{IL}: Low-speed on-chip oscillator clock frequency

2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

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

4. Temperature condition of the TYP. value is T_A = 25°C

31.4 AC Characteristics

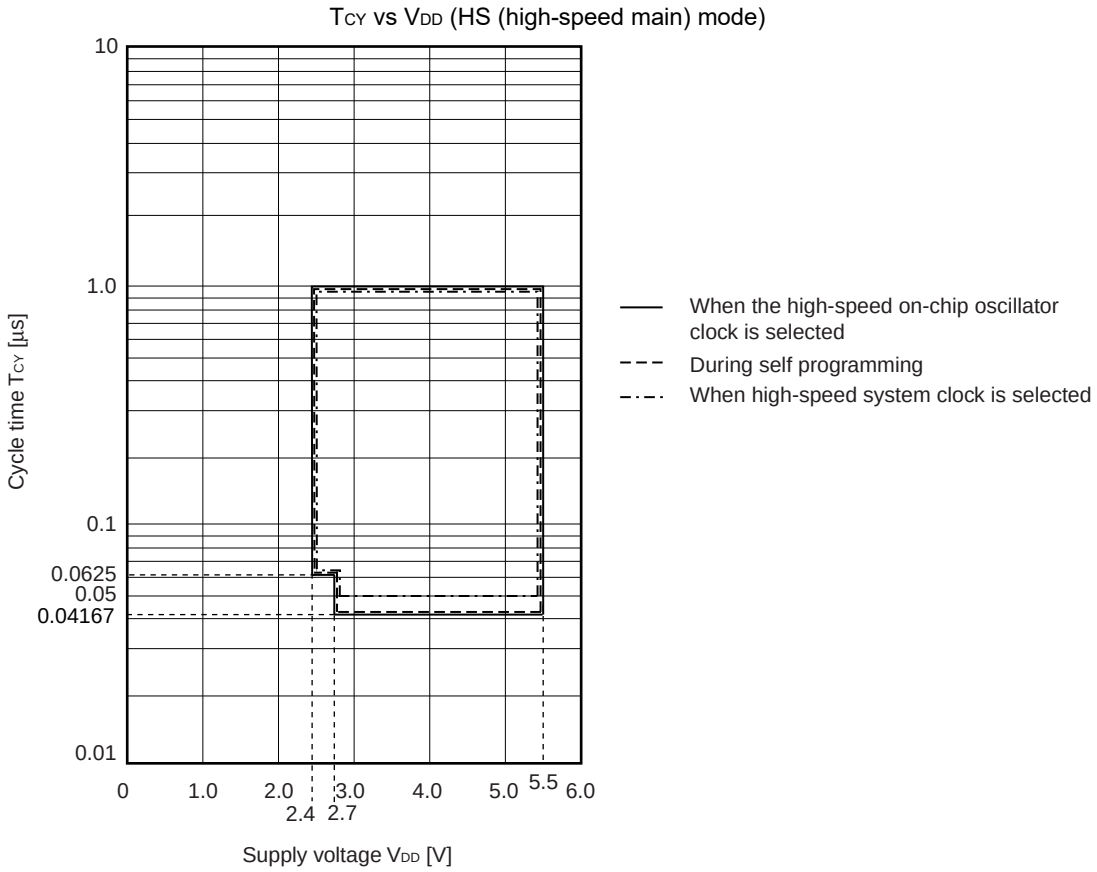
31.4.1 Basic operation

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

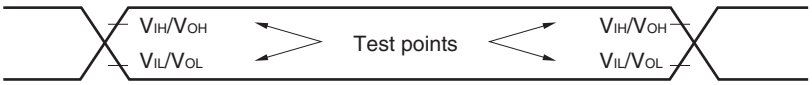
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (High-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
		Subsystem clock (f_{SUB}) operation		2.4 V ≤ V_{DD} ≤ 5.5 V	28.5	30.5	31.3	μs
		In the self programming mode	HS (High-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
External system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V			1.0		16.0	MHz
	f_{EXS}				32		35	kHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V			24			ns
		2.4 V ≤ V_{DD} < 2.7 V			30			ns
	t_{EXHS}, t_{EXLS}				13.7			μs
TI00 to TI03 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK}+10			ns
TO00 to TO03 output frequency	f_{TO}	High-speed main mode	4.0 V ≤ V_{DD} ≤ 5.5 V				12	MHz
			2.7 V ≤ V_{DD} < 4.0 V				8	MHz
			2.4 V ≤ V_{DD} < 2.7 V				4	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	High-speed main mode	4.0 V ≤ V_{DD} ≤ 5.5 V				16	MHz
			2.7 V ≤ V_{DD} < 4.0 V				8	MHz
			2.4 V ≤ V_{DD} < 2.7 V				4	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP6, INTP8, INTP9	2.4 V ≤ V_{DD} ≤ 5.5 V		1			μs
Key interrupt input low-level width	t_{KR}	KR0 to KR5	2.4 V ≤ V_{DD} ≤ 5.5 V		250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of timer mode register 0n (TMR0n). n: Channel number (n = 0 to 3))

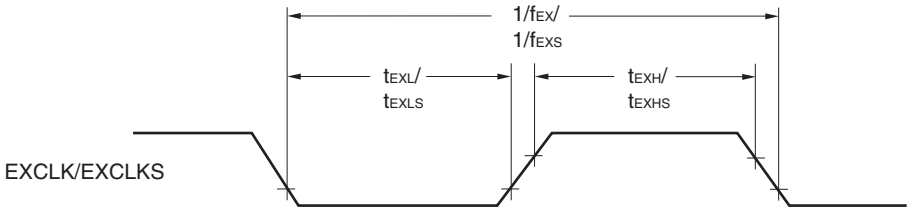
Minimum Instruction Execution Time during Main System Clock Operation



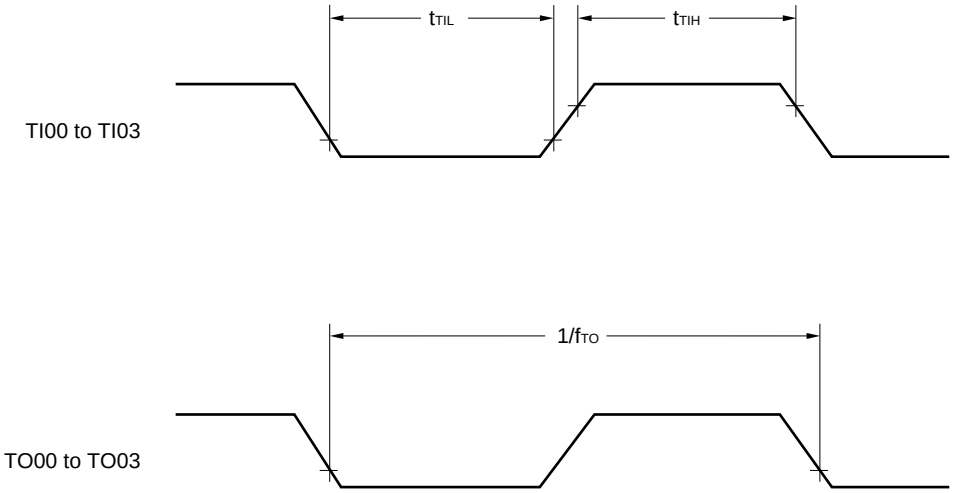
AC Timing Test Points



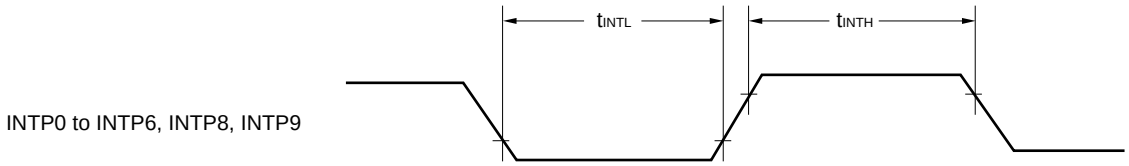
External System Clock Timing



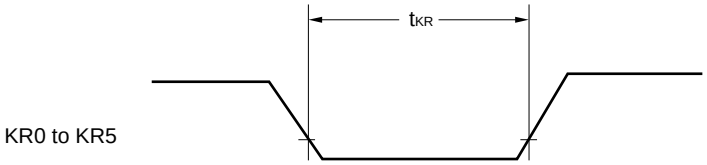
TI/TO Timing



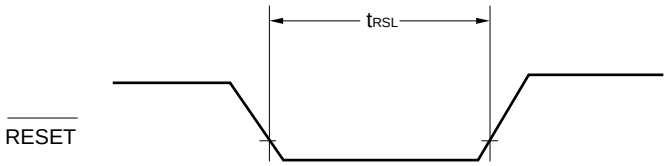
Interrupt Request Input Timing



Key Interrupt Input Timing



$\overline{\text{RESET}}$ Input Timing



31.5 Peripheral Functions Characteristics

31.5.1 Serial array unit

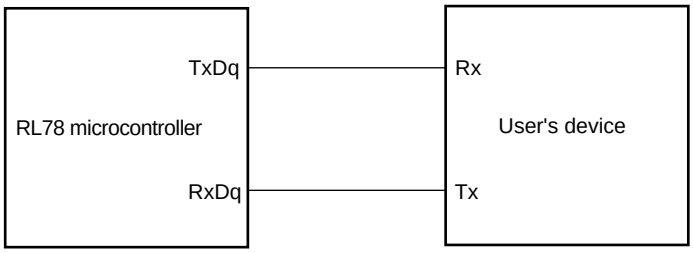
(1) During communication at same potential (UART mode) (dedicated baud rate generator output)
(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					f_{MCK}/12	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note			2.0	Mbps

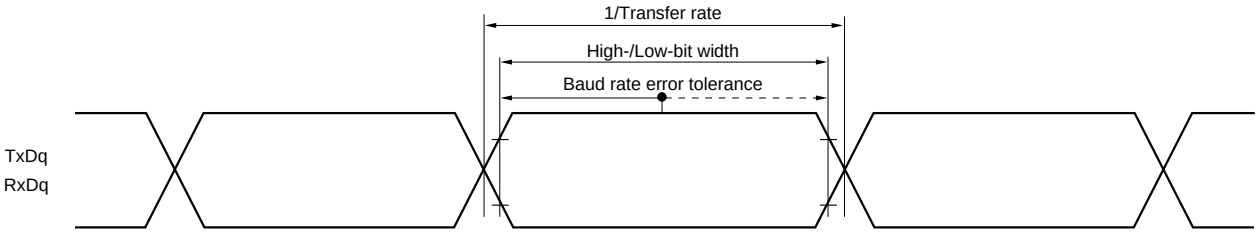
Note The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remarks 1. q: UART number (q = 0), g: PIM and POM number (g = 5)
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))

(2) During communication at same potential (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = –40 to +105°C, 2.4 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	250			ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	500			ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 – 24			ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 – 36			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 – 76			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V		66			ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		66			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		113			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}			38			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note 4				50	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM numbers (g = 0, 3, 5, 7)

2. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

(3) During communication at same potential (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
SCKp cycle time Note 5	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V	20 MHz < f_{MCK}	16/f_{MCK}			ns
			f_{MCK} ≤ 20 MHz	12/f_{MCK}			ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	16 MHz < f_{MCK}	16/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	12/f_{MCK}			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		12/f_{MCK} and 1000			ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 – 14			ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 – 16			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2 – 36			ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK}+40			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK}+60			ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}	2.7 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK}+62			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK}+62			ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	2.7 V ≤ V_{DD} ≤ 5.5 V			2/f_{MCK}+66	ns
			2.4 V ≤ V_{DD} ≤ 5.5 V			2/f_{MCK}+113	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the SOp output lines.

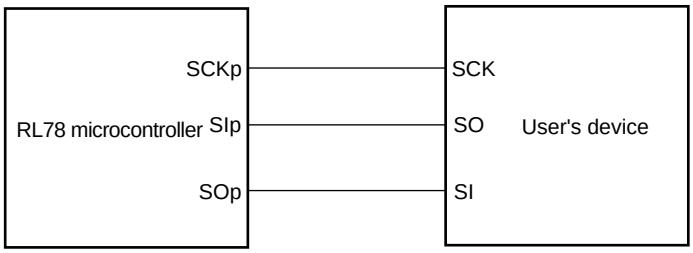
5. Transfer rate in the SNOOZE mode : MAX. 1 Mbps

Caution Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

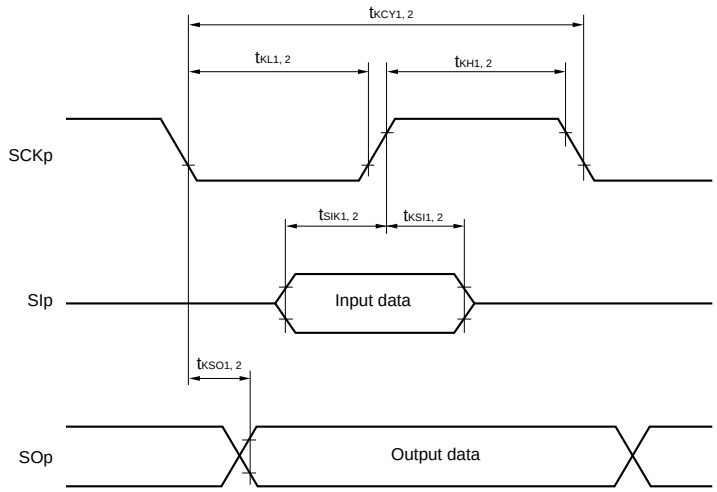
Remarks 1. p: CSI number (p = 00, 01), m: Unit number (m = 0),
n: Channel number (n = 0, 1), g: PIM number (g = 0, 3, 5, 7)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

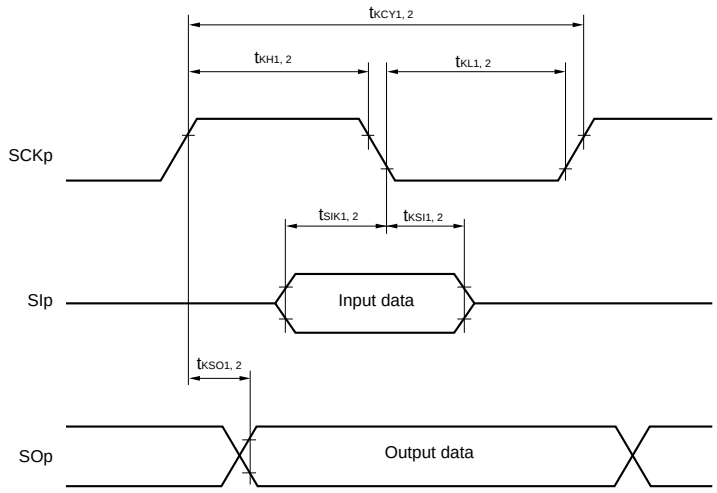
Simplified SPI (CSI) mode connection diagram (during communication at same potential)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00, 01)

2. m: Unit number, n: Channel number (mn = 00, 01)

(4) During communication at same potential (simplified I²C mode)
(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

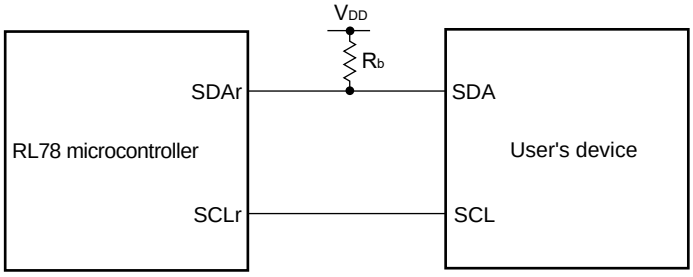
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		100 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	4600		ns
Hold time when SCLr = "H"	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	4600		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 220 Note 2		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK} + 580 Note 2		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	1420	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".

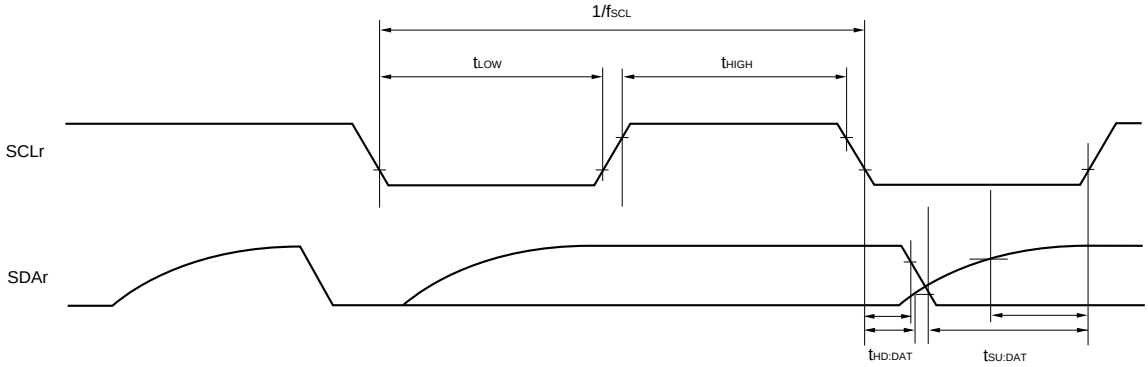
Caution Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).

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

Simplified I²C mode mode connection diagram (during communication at same potential)



Simplified I²C mode serial transfer timing (during communication at same potential)



Remarks 1. R_b[Ω]:Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance

2. r: IIC number (r = 00, 01), g: PIM number (g = 5), h: POM number (h = 3, 5)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01)

(5) Communication at different potential (2.5 V, 3 V) (UART mode) (1/2)
(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate		reception	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V				f_{MCK}/12 Note 1	bps
				Theoretical value of the maximum transfer rate f_{CLK} = 24 MHz, f_{MCK} = f_{CLK} Note 2			2.0	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V				f_{MCK}/12 Note 1	bps
				Theoretical value of the maximum transfer rate f_{CLK} = 24 MHz, f_{MCK} = f_{CLK} Note 2			2.0	Mbps
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V				f_{MCK}/12 Note 1	bps
				Theoretical value of the maximum transfer rate f_{CLK} = 24 MHz, f_{MCK} = f_{CLK} Note 2			2.0	Mbps

Notes 1. Use it with V_{DD}≥V_b.

2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected

Remarks 1. V_b[V]: Communication line voltage

2. q: UART number (q = 0), g: PIM and POM number (g = 5)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00)

(5) Communication at different potential (2.5 V, 3 V) (UART mode) (2/2)
(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate		transmission	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V				**Note 1**	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V			2.6 Note 2	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V				**Note 3**	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V			1.2 Note 4	Mbps
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V				**Notes 5, 6**	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V			0.43 Note 7	Mbps

Notes 1. The smaller maximum transfer rate derived by using f_{MCK}/12 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 4.0 V ≤ V_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

2. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to Note 1 above to calculate the maximum transfer rate under conditions of the customer.

3. The smaller maximum transfer rate derived by using f_{MCK}/12 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

4. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to Note 3 above to calculate the maximum transfer rate under conditions of the customer.

5. Use it with V_{DD} ≥ V_b.

Notes 6. The smaller maximum transfer rate derived by using $f_{MCK}/12$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.4 V ≤ V_{DD} < 3.3 V and 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

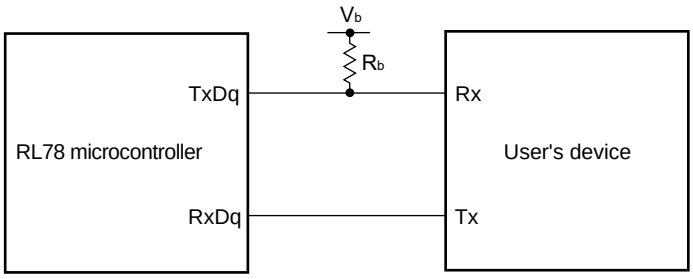
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

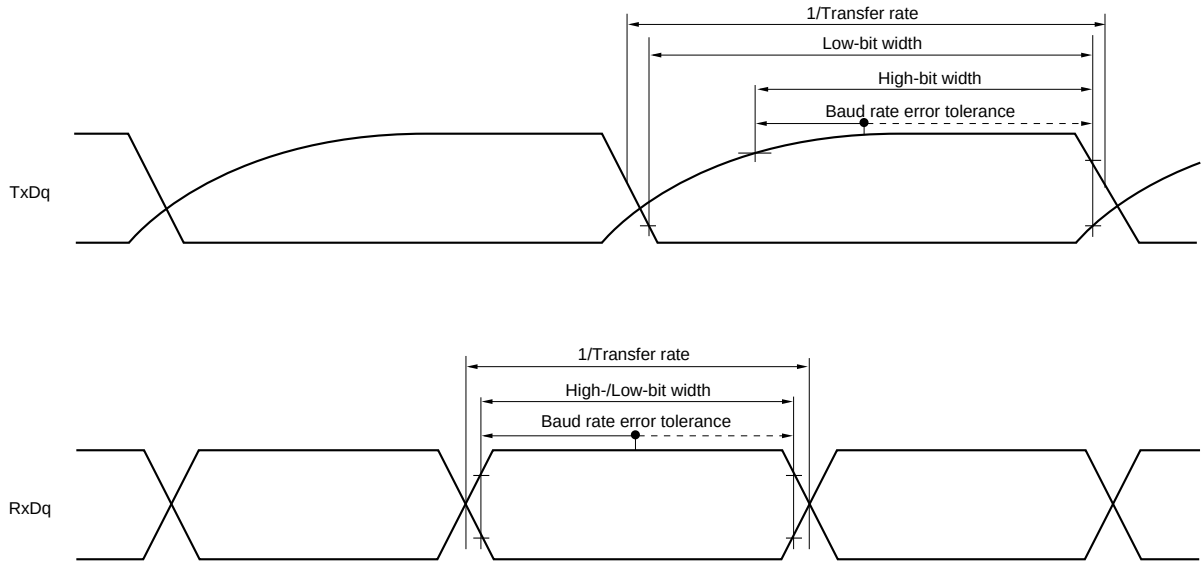
7. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to Note 6 above to calculate the maximum transfer rate under conditions of the customer.

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Remarks 1. R_b[Ω]:Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

2. q: UART number (q = 0), g: PIM and POM number (g = 5)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00))

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output) (1/2)

(T_A = –40 to +105°C, 2.4 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}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	600			ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	1000			ns
			2.4 V ≤ V_{DD} < 3.3 V, 2.4 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	2300			ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 150			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 340			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 – 916			ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 – 24			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 – 36			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 – 100			ns

Cautions 1. Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

2. Use it with V_{DD} ≥ V_b.

(**Remarks** are listed two pages after the next page.)

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output) (2/2)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	162			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	354			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	958			ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	38			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	38			ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			200	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			390	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ			966	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	88			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	88			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	220			ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	38			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	38			ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			50	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			50	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ			50	ns

(**Notes**, **Cautions** and **Remarks** are listed on the next page.)

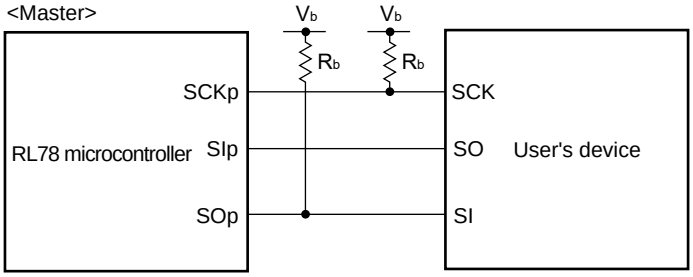
Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3 Use it with $V_{DD} \geq V_b$.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Simplified SPI (CSI) mode connection diagram (during communication at different potential)



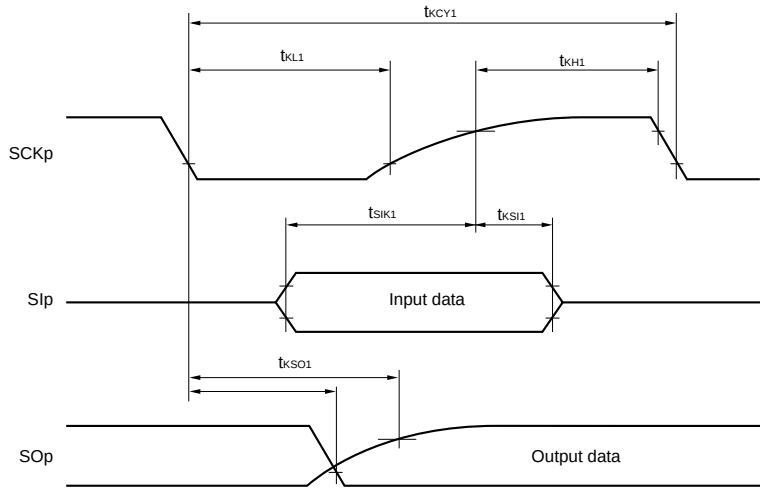
Remarks 1. R_b[Ω]:Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

2. p: CSI number (p = 00), m: Unit number , n: Channel number (mn = 00), g: PIM and POM number (g = 0, 3, 5, 7)

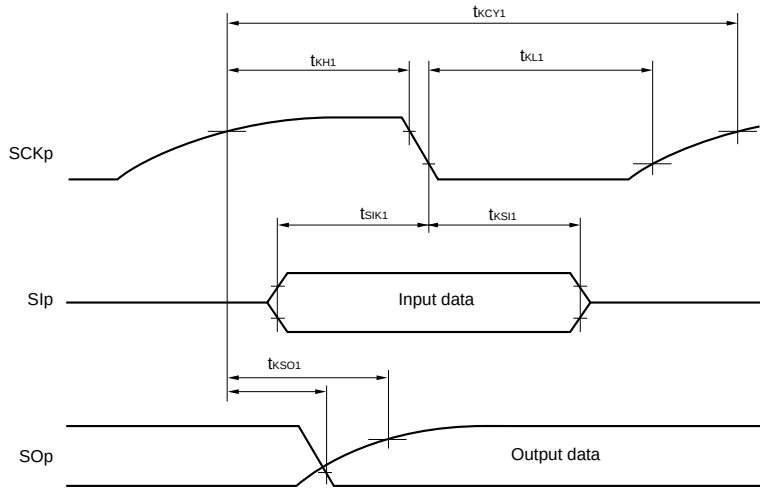
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00))

4. CSI01 cannot communicate at different potential. Use other CSI for communication at different potential.

Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00), m: Unit number, n: Channel number (mn = 00), g: PIM and POM number (g = 0, 3, 5, 7)

2. CSI01 cannot communicate at different potential. Use other CSI for communication at different potential.

(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
SCKp cycle time Note 1	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	20 MHz < f_{MCK} ≤ 24 MHz	24/f_{MCK}			ns
			8 MHz < f_{MCK} ≤ 20 MHz	20/f_{MCK}			ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}			ns
			f_{MCK} ≤ 4 MHz	12/f_{MCK}			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	20 MHz < f_{MCK} ≤ 24 MHz	32/f_{MCK}			ns
			16 MHz < f_{MCK} ≤ 20 MHz	28/f_{MCK}			ns
			8 MHz < f_{MCK} ≤ 16 MHz	24/f_{MCK}			ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}			ns
			f_{MCK} ≤ 4 MHz	12/f_{MCK}			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2	20 MHz < f_{MCK} ≤ 24 MHz	72/f_{MCK}			ns
			16 MHz < f_{MCK} ≤ 20 MHz	64/f_{MCK}			ns
			8 MHz < f_{MCK} ≤ 16 MHz	52/f_{MCK}			ns
			4 MHz < f_{MCK} ≤ 8 MHz	32/f_{MCK}			ns
			f_{MCK} ≤ 4 MHz	20/f_{MCK}			ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 – 24			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 – 36			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2		t_{KCY2}/2 – 100			ns
SIp setup time (to SCKp↑) Note 3	t_{SIK2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		1/f_{MCK} + 40			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		1/f_{MCK} + 40			ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2		1/f_{MCK} + 60			ns
SIp hold time (from SCKp↑) Note 4	t_{KSI2}			1/f_{MCK} + 62			ns
Delay time from SCKp↓ to SOp output Note 5	t_{KSO2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ				2/f_{MCK} + 240	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ				2/f_{MCK} + 428	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 30 pF, R_b = 5.5 kΩ				2/f_{MCK} + 1146	ns

Notes 1. Transfer rate in the SNOOZE mode : MAX. 1 Mbps

2. Use it with V_{DD} ≥ V_b.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

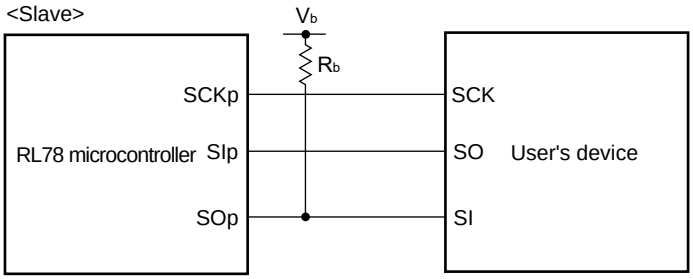
4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

5. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

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

Caution Select the TTL input buffer for the SIp pin and SCKp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Simplified SPI (CSI) mode connection diagram (during communication at different potential)



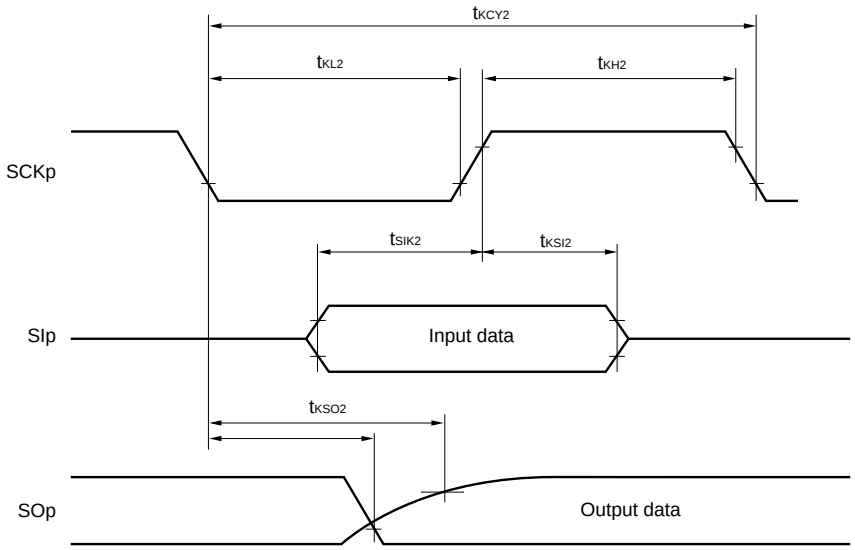
Remarks 1. R_b[Ω]:Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

2. p: CSI number (p = 00), m: Unit number, n: Channel number (mn = 00), g: PIM and POM number (g = 0, 3, 5, 7)

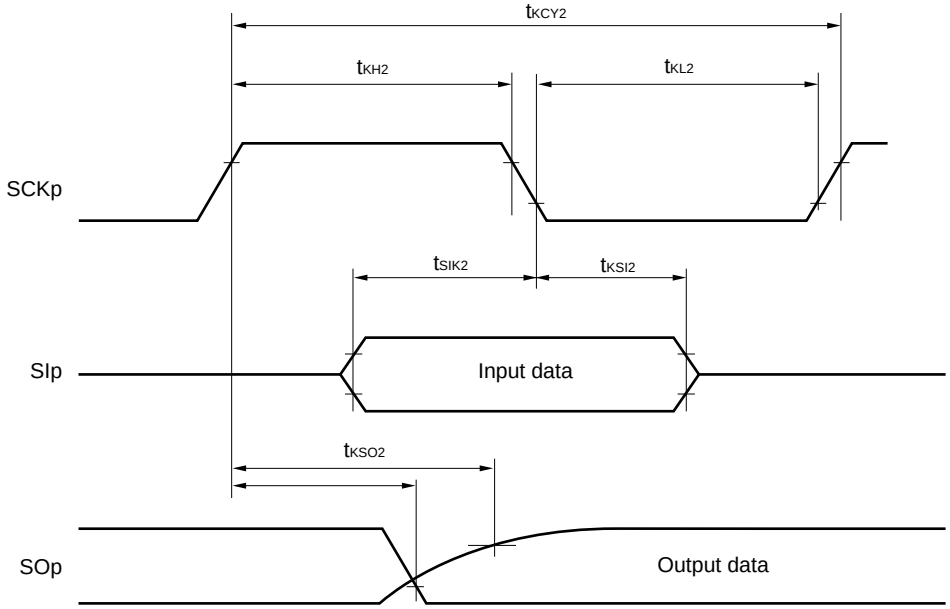
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00))

4. CSI01 cannot communicate at different potential. Use other CSI for communication at different potential.

Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00), m: Unit number, n: Channel number (mn = 00),
g: PIM and POM number (g = 0, 3, 5, 7)

2. CSI01 cannot communicate at different potential. Use other CSI for communication at different potential.

(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I²C mode) (1/2)
(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		100 Note 1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		100 Note 1	kHz
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ		100 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	4600		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	4600		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	4650		ns
Hold time when SCLr = "H"	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	620		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	500		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	2700		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	2400		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	1830		ns

(**Notes**, **Caution** and **Remarks** are listed on the next page.)

(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I²C mode) (2/2)
(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

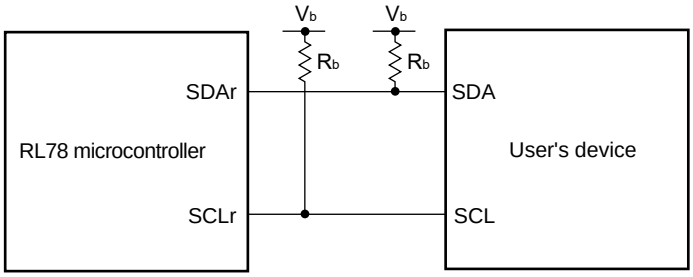
Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 340 Note 3		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 340 Note 3		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	$1/f_{MCK}$ + 760 Note 3		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 760 Note 3		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Notes 2, C_b = 100 pF, R_b = 5.5 kΩ	$1/f_{MCK}$ + 570 Note 3		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	1420	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	1420	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	0	1215	ns

Notes 1. The value must also be equal to or less than $f_{MCK}/4$.
2. Use it with $V_{DD} \geq V_b$.
3. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

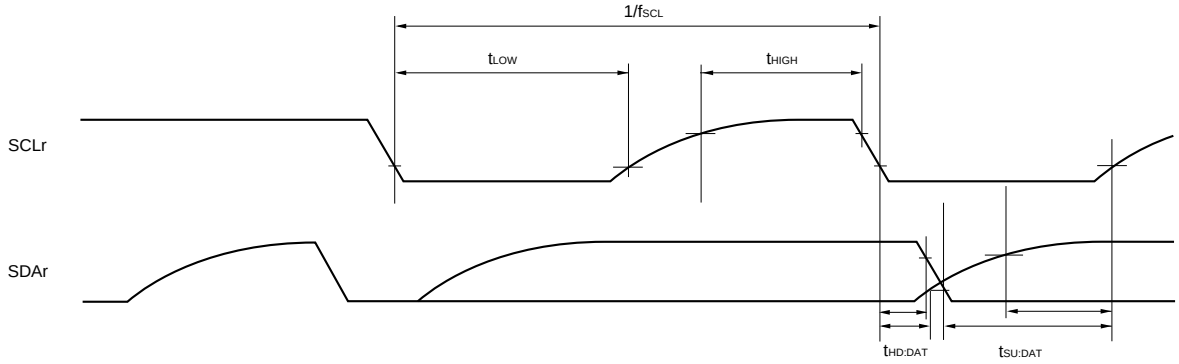
Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(**Remarks** are listed on the next page.)

Simplified I²C mode connection diagram (during communication at different potential)



Simplified I²C mode serial transfer timing (during communication at different potential)



Remarks 1. R_b[Ω]:Communication line (SDAr, SCLr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance, V_b[V]: Communication line voltage

2. r: IIC number (r = 00), g: PIM, POM number (g = 0, 3, 5, 7)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00)

31.5.2 Serial interface IICA

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode				Unit
			Standard Mode		Fast Mode		
			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.

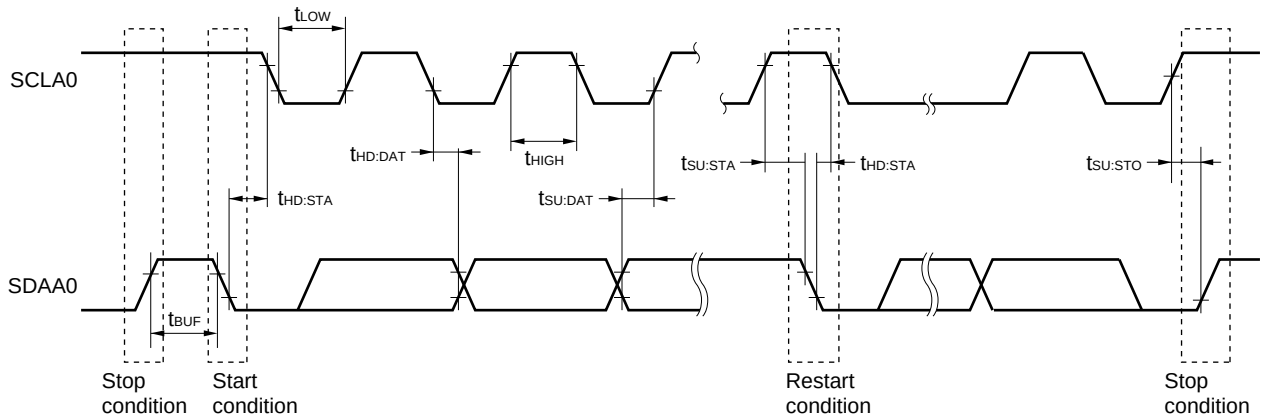
2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a clock stretch state is inserted in the $\overline{ACK}$ (acknowledge) timing.

Caution The values in the above table are applied even when bit 1 (PIOR1) in the peripheral I/O redirection register (PIOR) is 1. At this time, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.

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 = 320 pF, R_b = 1.1 kΩ

IICA serial transfer timing



31.5.3 USB

(1) Electrical specifications

(T_A = –40 to +105°C, 3.0 V ≤ UV_{DD} ≤ 3.6 V, 3.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
UV_{DD}	UV_{DD} input voltage characteristic	UV_{DD}	V_{DD} = 3.0 to 5.5 V, PXXCON = 1, VDDUSEB = 0 (UV_{DD} ≤ V_{DD})	3.0	3.3	3.6	V
	UV_{DD} output voltage characteristic	UV_{DD}	V_{DD} = 4.0 to 5.5 V, PXXCON = VDDUSEB = 1	3.0	3.3	3.6	V
UV_{BUS}	UV_{BUS} input voltage characteristic	UV_{BUS}	Function	4.35 (4.02[Note])	5.00	5.25	V
			Host	4.75	5.00	5.25	V

Note Value of instantaneous voltage

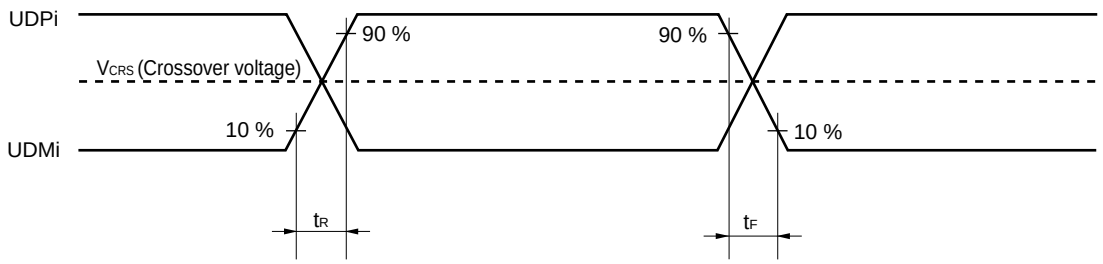
(T_A = –40 to +105°C, 3.0 V ≤ UV_{DD} ≤ 3.6 V, 3.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter			Symbol	Conditions	MIN.	TYP.	MAX.	Unit
UDPi/UDMi pins input characteristic (FS/LS receiver)	Input voltage		V_{IH}		2.0			V
			V_{IL}				0.8	V
	Difference input sensitivity		V_{DI}	| UDP voltage – UDM voltage |	0.2			V
	Difference common mode range		V_{CM}		0.8		2.5	V
UDPi/UDMi pins output characteristic (FS driver)	Output voltage		V_{OH}	I_{OH} = –200 μA	2.8		3.6	V
			V_{OL}	I_{OL} = 2.4 mA	0		0.3	V
	Transition time	Rising	t_{FR}	Rising: From 10% to 90 % of amplitude, Falling: From 90% to 10 % of amplitude, CL = 50 pF	4		20	ns
		Falling	t_{FF}		4		20	ns
	Matching (TFR/TFF)		V_{FRFM}		90		111.1	%
	Crossover voltage		V_{FCRS}		1.3		2.0	V
	Output Impedance		Z_{DRV}	UV_{DD} voltage = 3.3 V, Pin voltage = 1.65 V	28		44	Ω
UDPi/UDMi pins output characteristic (LS driver)	Output voltage		V_{OH}		2.8		3.6	V
			V_{OL}		0		0.3	V
	Transition time	Rising	t_{LR}	Rising: From 10% to 90 % of amplitude, Falling: From 90% to 10 % of amplitude, CL = 200 to 600 pF When the host controller function is selected: The UDMi pin (i = 0, 1) is pulled up via 1.5 kΩ. When the function controller function is selected: The UDP0 and UDM0 pins are individually pulled down via 15 kΩ	75		300	ns
		Falling	t_{LF}		75		300	ns
	Matching (TFR/TFF) Note		V_{LTFM}		80		125	%
	Crossover voltage Note		V_{LCRS}		1.3		2.0	V
UDPi/UDMi pins pull-up, pull-down	Pull-down resistor		R_{PD}		14.25		24.80	kΩ
	Pull-up resistor (i = 0 only)	Idle	R_{PUI}		0.9		1.575	kΩ
		Reception	R_{PUA}		1.425		3.09	kΩ
UV_{BUS}	UV_{BUS} pull-down resistor		R_{VBUS}	UV_{BUS} voltage = 5.5 V		1000		kΩ
	UV_{BUS} input voltage		V_{IH}		3.20			V
			V_{IL}				0.8	V

Note Excludes the first signal transition from the idle state.

Remark i = 0, 1

Timing of UDPi and UDMi



(2) BC standard

(T_A = –40 to +105°C, 3.0 V ≤ UV_{DD} ≤ 3.6 V, 3.0 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
USB standard BC1.2	UDPi sink current	I_{DP_SINK}		25		175	μA
	UDMi sink current	I_{DM_SINK}		25		175	μA
	DCD source current	I_{DP_SRC}		7		13	μA
	Dedicated charging port resistor	R_{DCP_DAT}	0 V < UDP/UDM voltage < 1.0 V			200	Ω
	Data detection voltage	V_{DAT_REF}		0.25		0.4	V
	UDPi source voltage	V_{DP_SRC}	Output current 250 μA	0.5		0.7	V
	UDMi source voltage	V_{DM_SRC}	Output current 250 μA	0.5		0.7	V

Remark i = 0, 1

(3) BC option standard (Host)

(T_A = –40 to +105°C, 4.75 V ≤ UV_{BUS} ≤ 5.25 V, 3.0 V ≤ UV_{DD} ≤ 3.6 V, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter			Symbol	Conditions	MIN.	TYP.	MAX.	Unit
UDPi output voltage (UV_{BUS} divider ratio) • VDOUEi = 1	VDSELi [3:0] (i = 0, 1)	1000	V_{P20}		38	40	42	% UV_{BUS}
		1001	V_{P27}		51.6	53.6	55.6	% UV_{BUS}
		1010	V_{P20}		38	40	42	% UV_{BUS}
		1100	V_{P33}		60	66	72	% UV_{BUS}
UDMi output voltage (UV_{BUS} divider ratio) • VDOUEi = 1	VDSELi [3:0] (i = 0, 1)	1000	V_{M20}		38	40	42	% UV_{BUS}
		1001	V_{M20}		38	40	42	% UV_{BUS}
		1010	V_{M27}		51.6	53.6	55.6	% UV_{BUS}
		1100	V_{M33}		60	66	72	% UV_{BUS}
UDPi comparing voltage Note 1 (UV_{BUS} divider ratio) • VDOUEi = 1 • CUSDETEi = 1	VDSELi [3:0] (i = 0, 1)	1000	V_{HDETP_UP0}	The rise of pin voltage detection voltage	56.2			% UV_{BUS}
			V_{HDETP_DWN0}	The fall of pin voltage detection voltage			29.4	% UV_{BUS}
		1001	V_{HDETP_UP1}	The rise of pin voltage detection voltage	60.5			% UV_{BUS}
			V_{HDETP_DWN1}	The fall of pin voltage detection voltage			45.0	% UV_{BUS}
		1010	V_{HDETP_UP2}	The rise of pin voltage detection voltage	56.2			% UV_{BUS}
			V_{HDETP_DWN2}	The fall of pin voltage detection voltage			29.4	% UV_{BUS}
UDMi comparing voltage Note 1 (UV_{BUS} divider ratio) • VDOUEi = 1 • CUSDETEi = 1	VDSELi [3:0] (i = 0, 1)	1000	V_{HDETM_UP0}	The rise of pin voltage detection voltage	56.2			% UV_{BUS}
			V_{HDETM_DWN0}	The fall of pin voltage detection voltage			29.4	% UV_{BUS}
		1001	V_{HDETM_UP1}	The rise of pin voltage detection voltage	56.2			% UV_{BUS}
			V_{HDETM_DWN1}	The fall of pin voltage detection voltage			29.4	% UV_{BUS}
		1010	V_{HDETM_UP2}	The rise of pin voltage detection voltage	60.5			% UV_{BUS}
			V_{HDETM_DWN2}	The fall of pin voltage detection voltage			45.0	% UV_{BUS}
UDPi pull-up detection Note 2 Connect detection with the full speed function (pull-up resistor)		1000	R_{HDET_PULL}	In full-speed mode, the power supply voltage range of pull-up resistors connected to the USB function module is between 3.0 V and 3.6 V.			1.575	kΩ
		1001						
		1010						
UDMi pull-up detection Note 2 Connect detection with the low-speed (pull-up resistor)		1000	R_{HDET_PULL}	In low-speed mode, the power supply voltage range of pull-up resistors connected to the USB function module is between 3.0 V and 3.6 V.			1.575	kΩ
		1001						
		1010						
UDMi sink current detection Note 2 Connect detection with the BC1.2 portable device (sink resistor)		1000	I_{HDET_SINK}		25			μA
		1001						
		1010						

Notes 1. If the voltage output from UDPi or UDMi (i = 0, 1) exceeds the range of the MAX and MIN values prescribed in this specification, DPCUSDETi (bit 8) and DMCUSDETi (bit 9) of the USBBCOPTi register are set to 1.

2. If the pull-up resistance or sink current prescribed in this specification is applied to UDPi or UDMi (i = 0, 1), DPCUSDETi (bit 8) and DMCUSDETi (bit 9) of the USBBCOPTi register are set to 1.

Remark i = 0, 1

(4) BC option standard (Function)

(T_A = –40 to +105°C, 4.35 V ≤ UV_{BUS} ≤ 5.25 V, 3.0 V ≤ UV_{DD} ≤ 3.6 V, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter			Symbol	Conditions	MIN.	TYP.	MAX.	Unit
UDPi/UDMi input reference voltage (UV_{BUS} divider ratio) • VDOUEi = 0 (i = 0))	VDSELi [3:0] (i = 0)	0000	V_{DDET0}		27	32	37	% UV_{BUS}
		0001	V_{DDET1}		29	34	39	% UV_{BUS}
		0010	V_{DDET2}		32	37	42	% UV_{BUS}
		0011	V_{DDET3}		35	40	45	% UV_{BUS}
		0100	V_{DDET4}		38	43	48	% UV_{BUS}
		0101	V_{DDET5}		41	46	51	% UV_{BUS}
		0110	V_{DDET6}		44	49	54	% UV_{BUS}
		0111	V_{DDET7}		47	52	57	% UV_{BUS}
		1000	V_{DDET8}		51	56	61	% UV_{BUS}
		1001	V_{DDET9}		55	60	65	% UV_{BUS}
		1010	V_{DDET10}		59	64	69	% UV_{BUS}
		1011	V_{DDET11}		63	68	73	% UV_{BUS}
		1100	V_{DDET12}		67	72	77	% UV_{BUS}
		1101	V_{DDET13}		71	76	81	% UV_{BUS}
		1110	V_{DDET14}		75	80	85	% UV_{BUS}
		1111	V_{DDET15}		79	84	89	% UV_{BUS}

31.6 Analog Characteristics

31.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Input channel	Reference Voltage		
	Reference voltage (+) = AV_{REFP} Reference voltage (–) = AV_{REFM}	Reference voltage (+) = V_{DD} Reference voltage (–) = V_{SS}	Reference voltage (+) = V_{BGR} Reference voltage (–) = AV_{REFM}
ANI0 to ANI7	Refer to **31.6.1 (1)**.	Refer to **31.6.1 (3)**.	Refer to **31.6.1 (4)**.
ANI16, ANI17, ANI19	Refer to **31.6.1 (2)**.		
Internal reference voltage Temperature sensor output voltage	Refer to **31.6.1 (1)**.		–

(1) When $AV_{REF(+)}$ = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin : ANI2 to ANI7, internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +105°C, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ AV_{REFP} ≤ 5.5 V		1.2	±3.5	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI2 to ANI7	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
		10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error[Notes 1, 2]	EZS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±0.25	%FSR
Full-scale error[Notes 1, 2]	EFS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±0.25	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±2.5	LSB
Differential linearity error [Note 1]	DLE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±1.5	LSB
Analog input voltage	V_{AIN}	ANI2 to ANI7		0		AV_{REFP}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{BGR} [Note 4]			V
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{TMPS25} [Note 4]			V

(**Notes** are listed on the next page.)

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When $AV_{REFP} < V_{DD}$, the MAX. values are as follows.

Overall error: Add ±1.0 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.

Zero-scale error/Full-scale error: Add ±0.05%FSR to the MAX. value when $AV_{REFP} = V_{DD}$.

Integral linearity error/ Differential linearity error: Add ±0.5 LSB to the MAX. value when $AV_{REFP} = V_{DD}$.

4. Refer to **31.6.2 Temperature sensor/internal reference voltage characteristics**.

(2) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin : ANI16, ANI17, ANI19

(T_A = –40 to +105°C, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (–) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error[Note 1]	AINL	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ AV_{REFP} ≤ 5.5 V		1.2	±5.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target ANI pin : ANI16, ANI17, ANI19	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error[Notes 1, 2]	EZS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±0.35	%FSR
Full-scale error[Notes 1, 2]	EFS	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±0.35	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±3.5	LSB
Differential linearity error [Note 1]	DLE	10-bit resolution AV_{REFP} = V_{DD} [Note 3]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±2.0	LSB
Analog input voltage	V_{AIN}	ANI16, ANI17, ANI19		0		AV_{REFP} and V_{DD}	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When AV_{REFP} < V_{DD}, the MAX. values are as follows.

Overall error: Add ±4.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.

Zero-scale error/Full-scale error: Add ±0.20%FSR to the MAX. value when AV_{REFP} = V_{DD}.

Integral linearity error/ Differential linearity error: Add ±2.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.

(3) Reference voltage (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), Reference voltage (–) = V_{SS} (ADREFM = 0), target ANI pin : ANI0 to ANI7, ANI16, ANI17, ANI19, internal reference voltage, and temperature sensor output voltage

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{DD}, Reference voltage (–) = V_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}			8		10	bit
Overall error[Notes 1, 2]	AINL	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V		1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target ANI pin : ANI0 to ANI7, ANI16, ANI17, ANI19	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
		10-bit resolution Target ANI pin : Internal reference voltage, and temperature sensor output voltage (HS (high-speed main) mode)	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error[Notes 1, 2]	EZS	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
Full-scale error[Notes 1, 2]	EFS	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
Integral linearity error[Note 1]	ILE	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±4.0	LSB
Differential linearity error[Note 1]	DLE	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
Analog input voltage	V_{AIN}	ANI0 to ANI7, ANI16, ANI17, ANI19		0		V_{DD}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{BGR} [Note 3]			V
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 5.5 V, HS (high-speed main) mode)		V_{TMPS25} [Note 3]			V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. Refer to **31.6.2 Temperature sensor/internal reference voltage characteristics**.

(4) When Reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), Reference voltage (–) = AV_{REFM}/ANI1 (ADREFM = 1), target pin : ANI0 to ANI7, ANI16, ANI17, ANI19

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{BGR} Note 3, Reference voltage (–) = AV_{REFM} Note 4 = 0 V, HS (high-speed main) mode)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}				8		Bit
Conversion time	t_{CONV}	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	EZS	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
Integral linearity error Note 1	ILE	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
Differential linearity error Note 1	DLE	8-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±1.0	LSB
Analog input voltage	V_{AIN}			0		V_{BGR} Note 3	V

Notes 1. Excludes quantization error (±1/2 LSB).

2. This value is indicated as a ratio (%FSR) to the full-scale value.

3. Refer to **31.6.2 Temperature sensor/internal reference voltage characteristics**.

4. When reference voltage (–) = V_{SS}, the MAX. values are as follows.

Zero-scale error: Add ±0.35%FSR to the MAX. value when reference voltage (–) = AV_{REFM}.

Integral linearity error: Add ±0.5 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.

Differential linearity error: Add ±0.2 LSB to the MAX. value when reference voltage (–) = AV_{REFM}.

31.6.2 Temperature sensor/internal reference voltage characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, HS (high-speed main) mode)

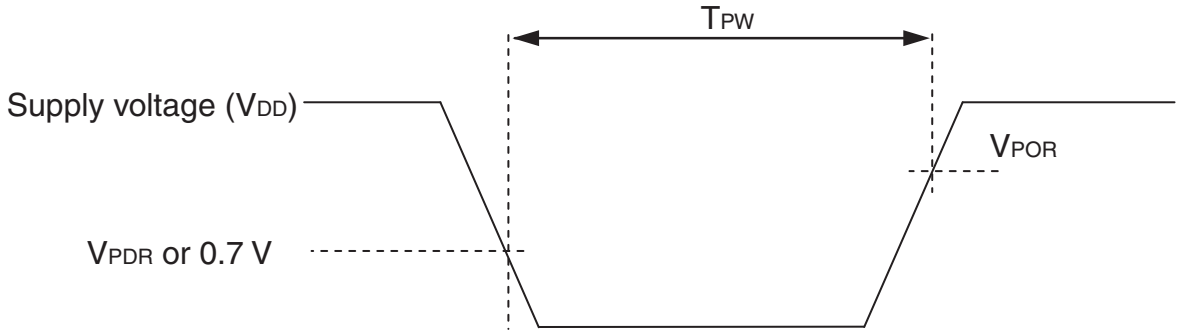
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGR}	Setting ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	F_{VTMPS}	Temperature sensor that depends on the temperature		–3.6		mV/°C
Operation stabilization wait time	t_{AMP}		5			μs

31.6.3 POR circuit characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	Power supply rise time	1.45	1.51	1.57	V
	V_{PDR}	Power supply fall time	1.44	1.50	1.56	V
Minimum pulse width[Note]	T_{PW}		300			μs

Note Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}. This is also the minimum time required for a POR reset from when V_{DD} exceeds below 0.7 V to when V_{DD} exceeds V_{POR} while STOP mode is entered or the main system clock (f_{MAIN}) is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



31.6.4 LVD circuit characteristics

LVD Detection Voltage of Reset Mode and Interrupt Mode

(T_A = –40 to +105°C, $V_{PDR} \leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD0}	Power supply rise time	3.90	4.06	4.22	V
			Power supply fall time	3.83	3.98	4.13	V
		V_{LVD1}	Power supply rise time	3.60	3.75	3.90	V
			Power supply fall time	3.53	3.67	3.81	V
		V_{LVD2}	Power supply rise time	3.01	3.13	3.25	V
			Power supply fall time	2.94	3.06	3.18	V
		V_{LVD3}	Power supply rise time	2.90	3.02	3.14	V
			Power supply fall time	2.85	2.96	3.07	V
		V_{LVD4}	Power supply rise time	2.81	2.92	3.03	V
			Power supply fall time	2.75	2.86	2.97	V
		V_{LVD5}	Power supply rise time	2.70	2.81	2.92	V
			Power supply fall time	2.64	2.75	2.86	V
		V_{LVD6}	Power supply rise time	2.61	2.71	2.81	V
			Power supply fall time	2.55	2.65	2.75	V
		V_{LVD7}	Power supply rise time	2.51	2.61	2.71	V
			Power supply fall time	2.45	2.55	2.65	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time		t_{LD}				300	μs

LVD Detection Voltage of Interrupt & Reset Mode

(T_A = –40 to +105°C, $V_{PDR} \leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVDD0}	VPOC2, VPOC1, VPOC0 = 0, 1, 1, falling reset voltage			2.64	2.75	2.86	V
	V_{LVDD1}		LVIS1, LVIS0 = 1, 0	Rising release reset voltage	2.81	2.92	3.03	V
				Falling interrupt voltage	2.75	2.86	2.97	V
	V_{LVDD2}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.90	3.02	3.14	V
				Falling interrupt voltage	2.85	2.96	3.07	V
	V_{LVDD3}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.90	4.06	4.22	V
				Falling interrupt voltage	3.83	3.98	4.13	V

31.6.5 Power supply voltage rising slope characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

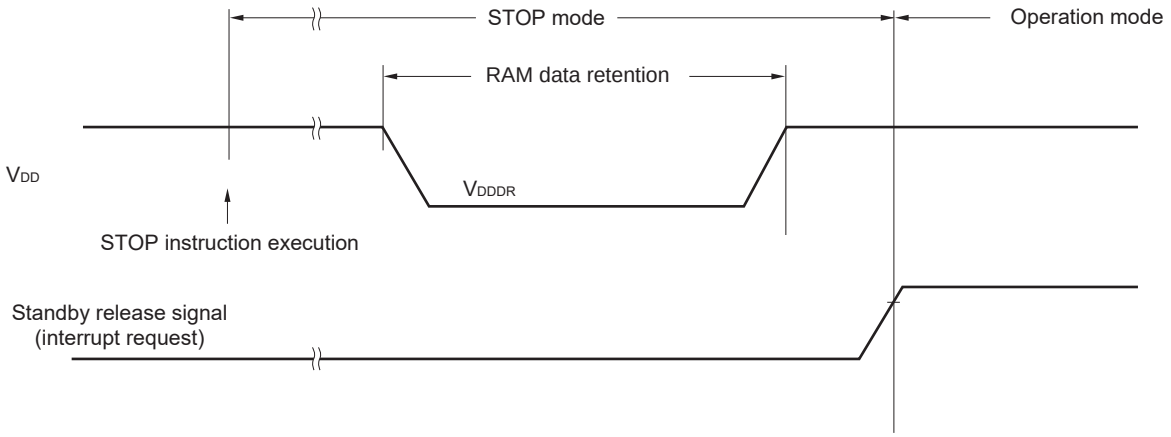
Caution Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 31.4 AC Characteristics.

31.7 RAM Data Retention Characteristics

(T_A = –40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.44 Note		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is effected, but data is not retained when a POR reset is effected.



31.8 Flash Memory Programming Characteristics

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
CPU/peripheral hardware clock frequency	f_{CLK}	2.4 V ≤ V_{DD} ≤ 5.5 V	1		24	MHz
Number of code flash rewrites	C_{erwr}	Retaining years: 20 years T_A = +85°C Note 4	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retaining years: 1 year T_A = +25°C Note 4		1,000,000		
		Retaining years: 5 years T_A = +85°C Note 4	100,000			
		Retaining years: 20 years T_A = +85°C Note 4	10,000			

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 and Renesas Electronics self programming library.

3. These specifications show the characteristics of the flash memory and the results obtained from Renesas Electronics reliability testing.

4. This temperature is the average value at which data are retained.

31.9 Dedicated Flash Memory Programmer Communication (UART)

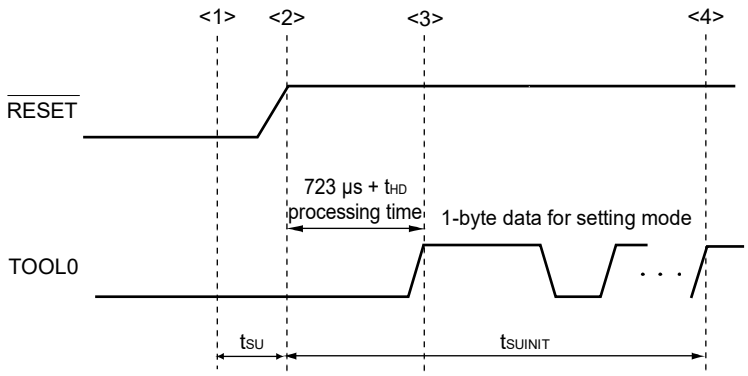
(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

31.10 Timing Specs for Switching Flash Memory Programming Modes

(T_A = –40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
How long from when an external reset ends until the initial communication settings are specified	t_{SUINIT}	POR and LVD reset must end before the external reset ends.			100	ms
How long from when the TOOL0 pin is placed at the low level until an external reset ends	t_{SU}	POR and LVD reset must end before the external reset ends.	10			μs
How long the TOOL0 pin must be kept at the low level after an external reset ends (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset must end before the external reset ends.	1			ms



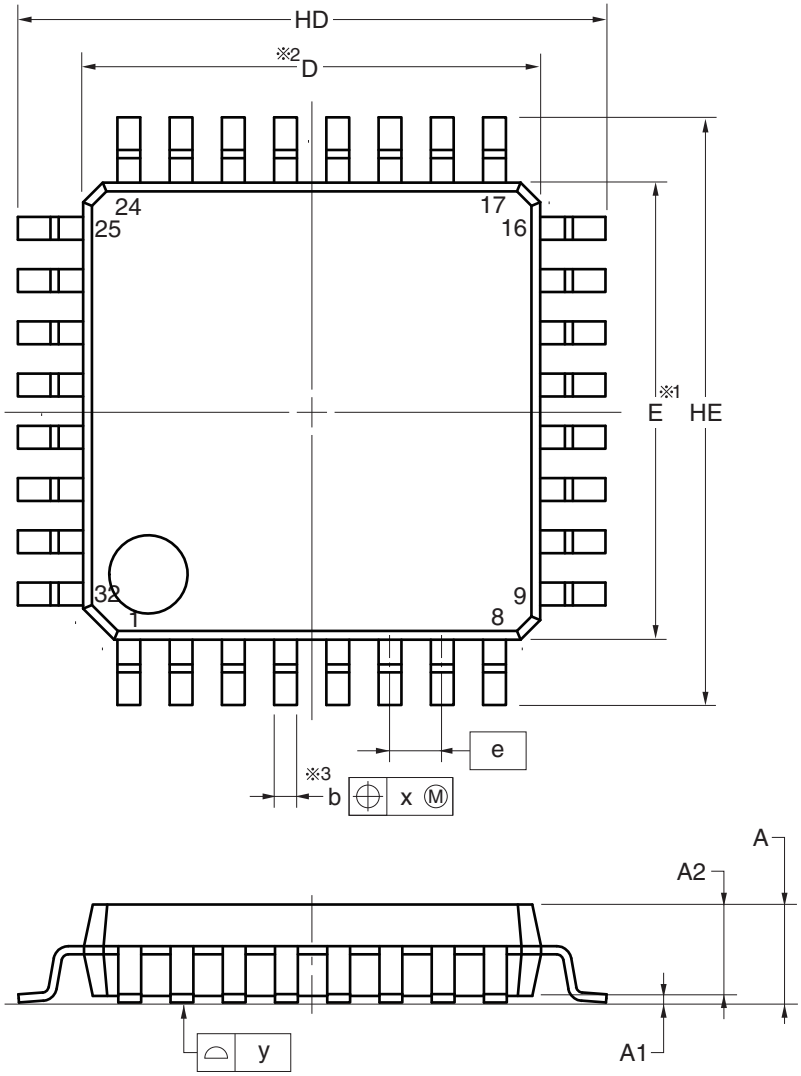
<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends.).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

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 and internal resets end (excluding the processing time of the firmware to control the flash memory)

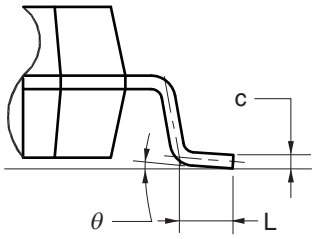
CHAPTER 32 PACKAGE DRAWINGS

32.1 32-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP32-7x7-0.80	PLQP0032GB-A	P32GA-80-GBT-1	0.2



detail of lead end



(UNIT:mm)

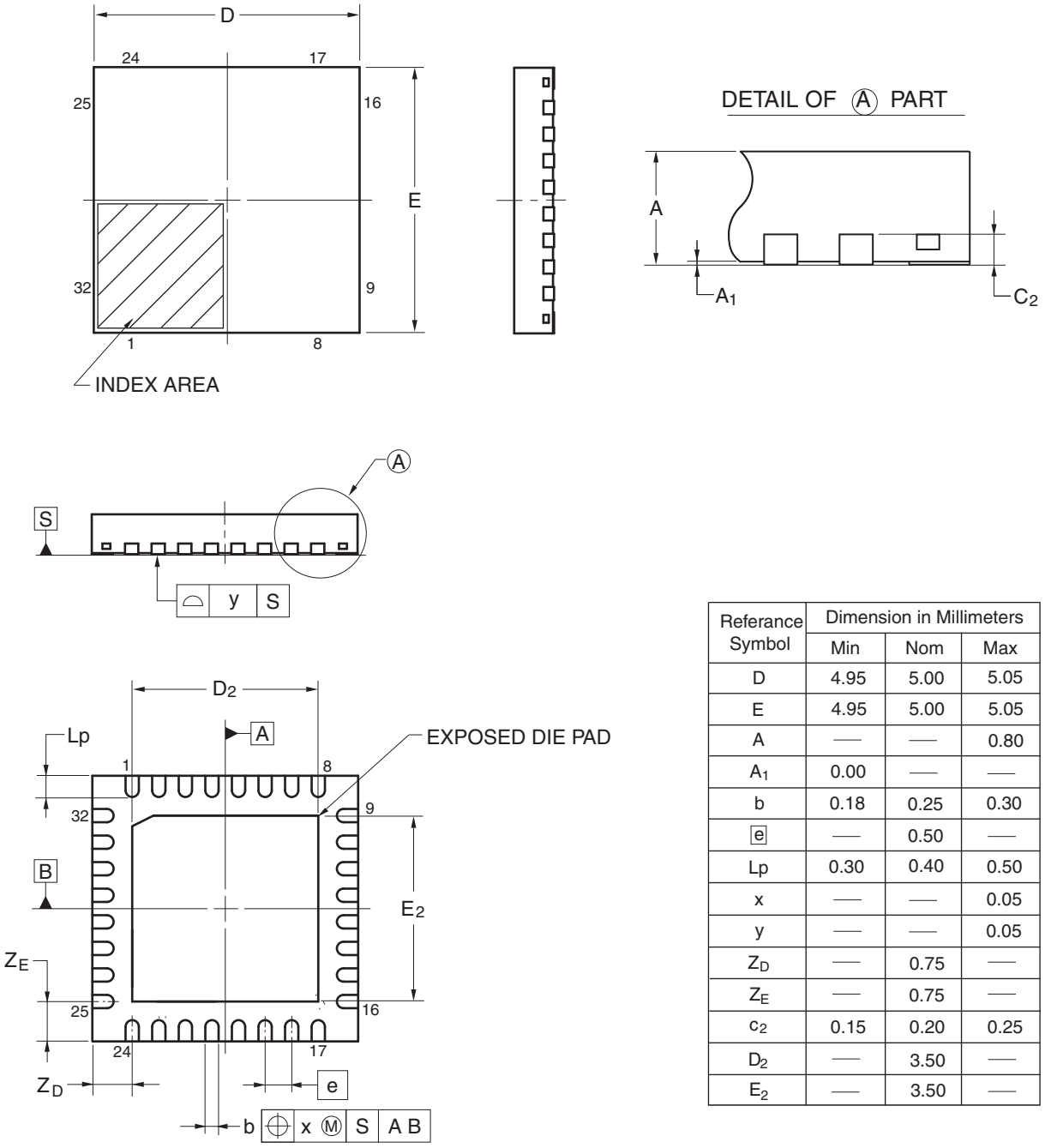
ITEM	DIMENSIONS
D	7.00±0.10
E	7.00±0.10
HD	9.00±0.20
HE	9.00±0.20
A	1.70 MAX.
A1	0.10±0.10
A2	1.40
b	0.37±0.05
c	0.145±0.055
L	0.50±0.20
θ	0° to 8°
e	0.80
x	0.20
y	0.10

NOTE

1.Dimensions “※1” and “※2” do not include mold flash.

2.Dimension “※3” does not include trim offset.

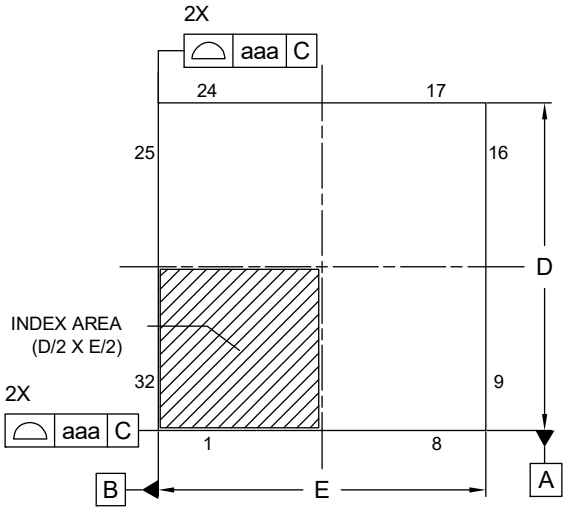
JEITA Package code	RENESAS code	Previous code	MASS (TYP.)[g]
P-HWQFN32-5x5-0.50	PWQN0032KB-A	P32K8-50-3B4-5	0.06

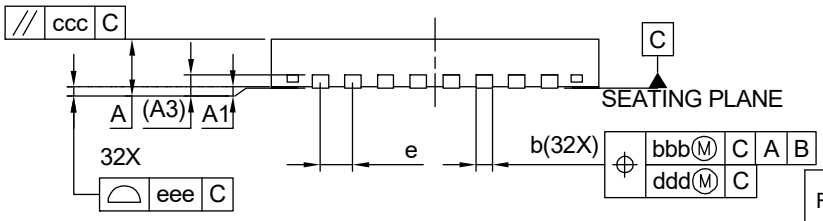


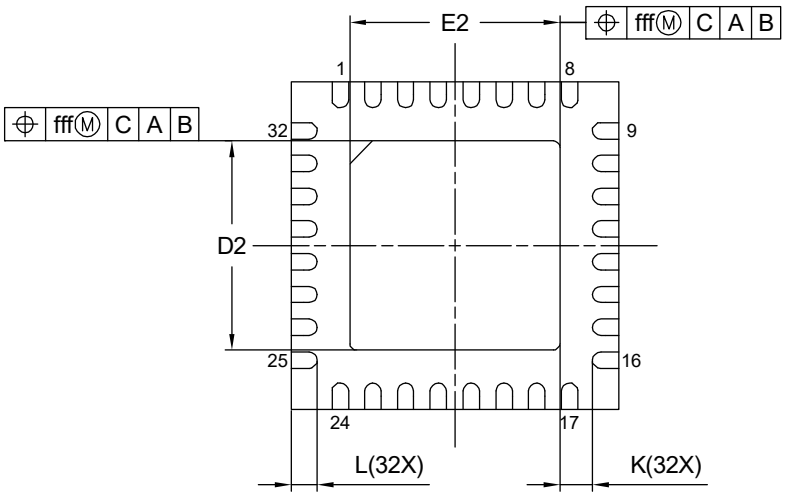
Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	4.95	5.00	5.05
E	4.95	5.00	5.05
A	—	—	0.80
A_1	0.00	—	—
b	0.18	0.25	0.30
e	—	0.50	—
Lp	0.30	0.40	0.50
x	—	—	0.05
y	—	—	0.05
Z_D	—	0.75	—
Z_E	—	0.75	—
c_2	0.15	0.20	0.25
D_2	—	3.50	—
E_2	—	3.50	—

©2013 Renesas Electronics Corporation. All rights reserved.

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN032-5x5-0.50	PWQN0032KE-A	0.06

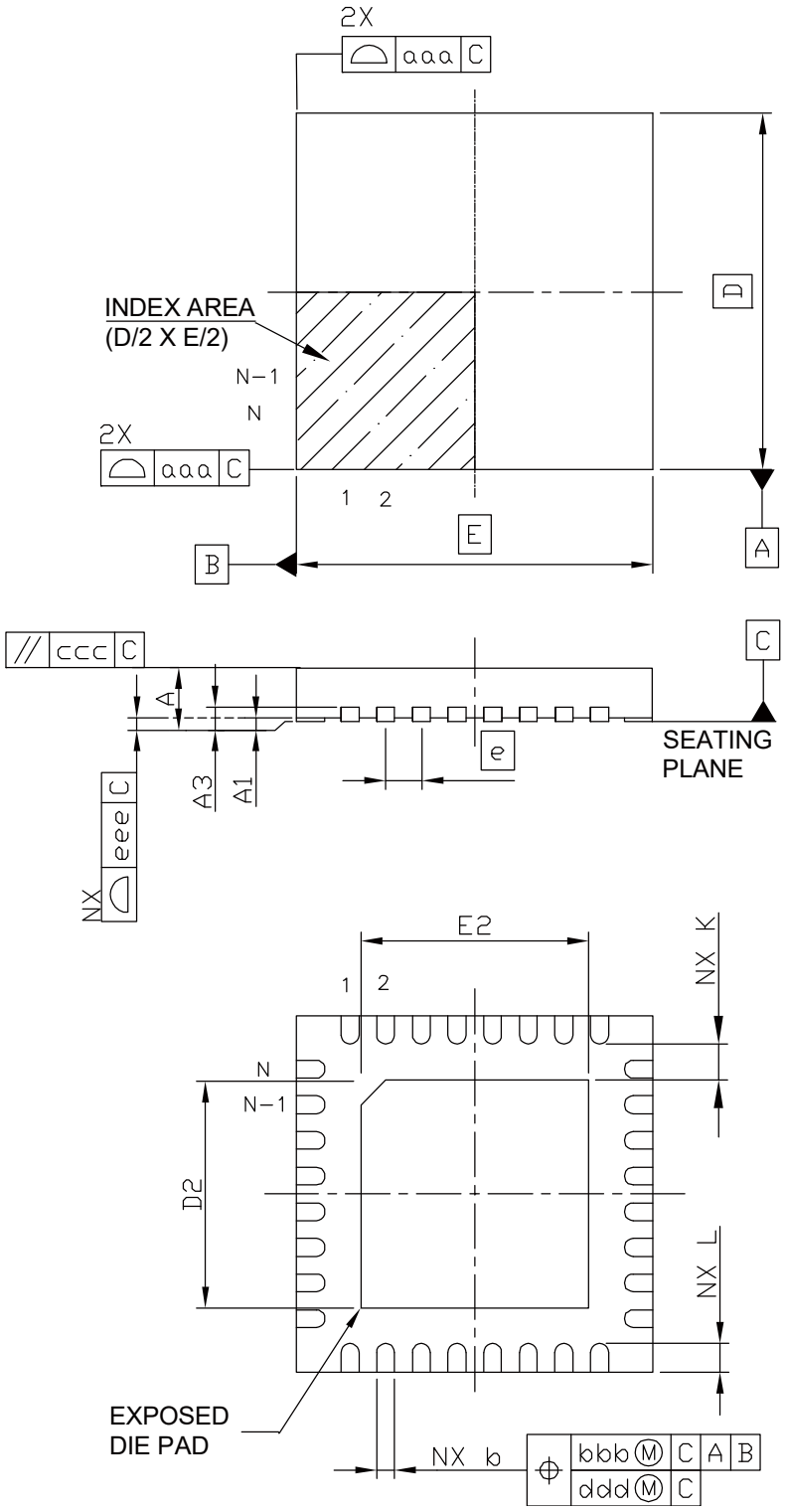






Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	-	-	0.80
A_1	0.00	0.02	0.05
A_3	0.203 REF.		
b	0.18	0.25	0.30
D	5.00 BSC		
E	5.00 BSC		
e	0.50 BSC		
L	0.35	0.40	0.45
K	0.20	-	-
D_2	3.15	3.20	3.25
E_2	3.15	3.20	3.25
aaa	0.15		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

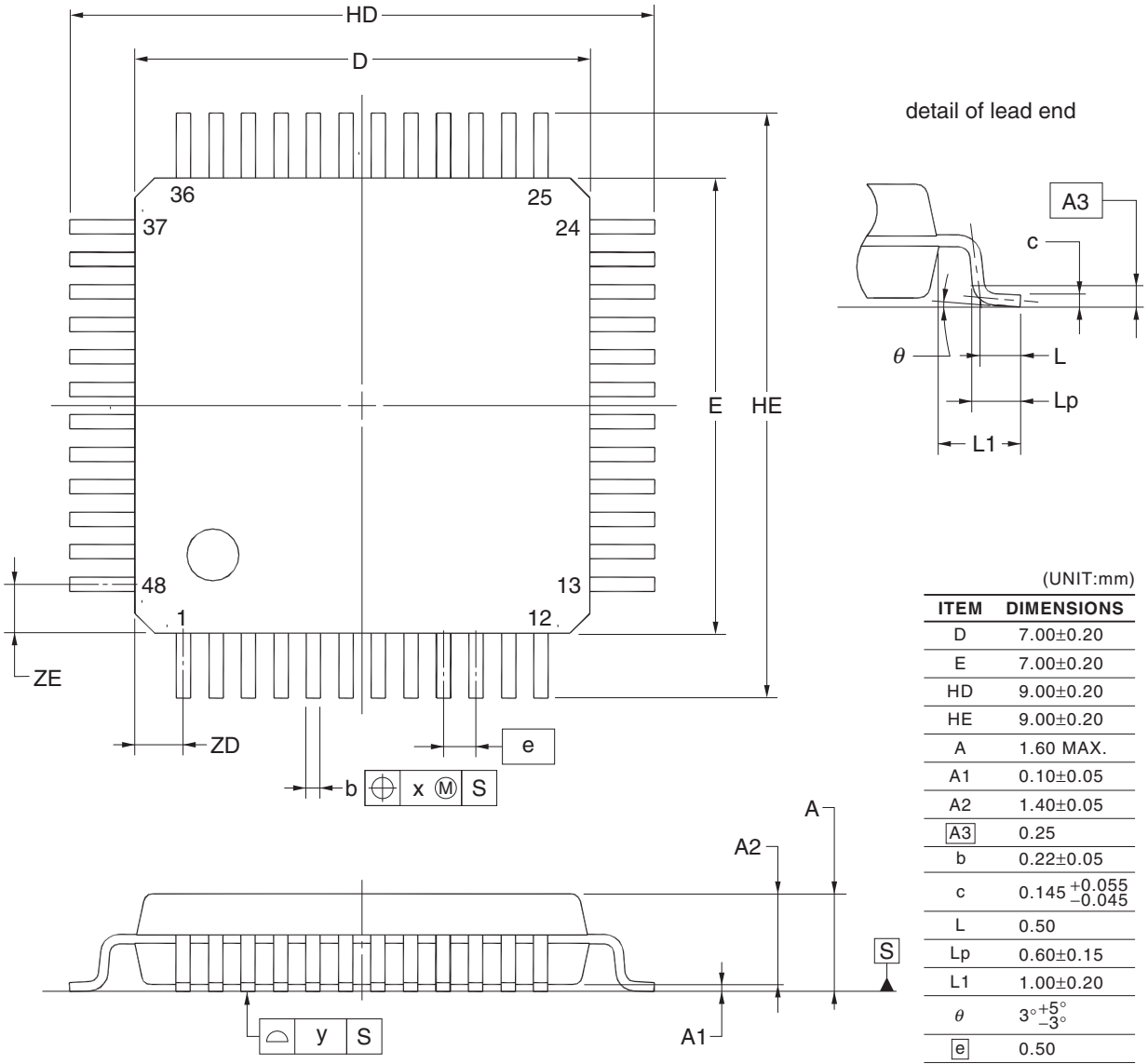
JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-HWQFN32-5×5-0.50	PWQN0032KG-A	0.06



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	—	0.05
A_3	0.20 REF.		
b	0.20	0.25	0.30
D	—	5.00	—
E	—	5.00	—
e	—	0.50	—
N	32		
L	0.30	0.40	0.50
K	0.20	—	—
D_2	3.10	3.20	3.30
E_2	3.10	3.20	3.30
aaa	—	—	0.15
bbb	—	—	0.10
ccc	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08

32.2 48-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LFQFP48-7x7-0.50	PLQP0048KF-A	P48GA-50-8EU-1	0.16



(UNIT:mm)

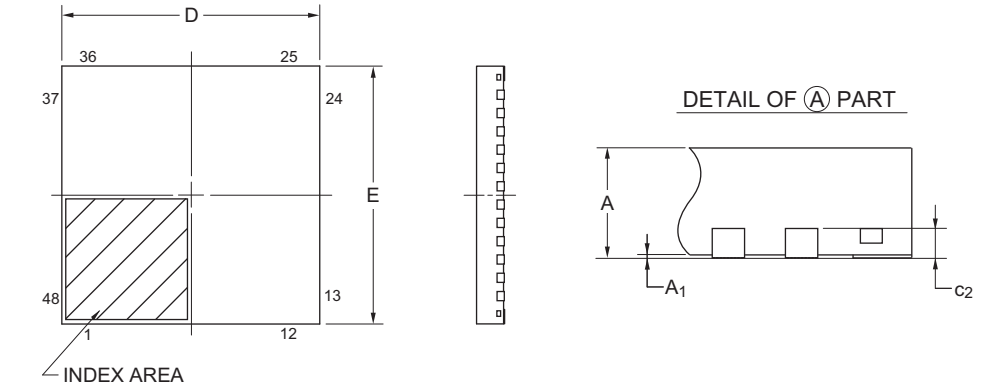
ITEM	DIMENSIONS
D	7.00±0.20
E	7.00±0.20
HD	9.00±0.20
HE	9.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.22±0.05
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	0.75
ZE	0.75

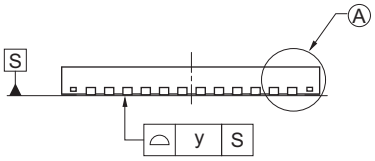
NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

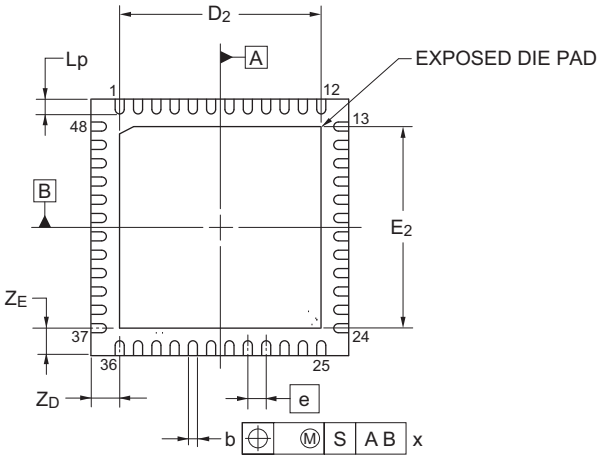
©2012 Renesas Electronics Corporation. All rights reserved.

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-HWQFN48-7x7-0.50	PWQN0048KB-A	48PJN-A P48K8-50-5B4-7	0.13

Unit: mm



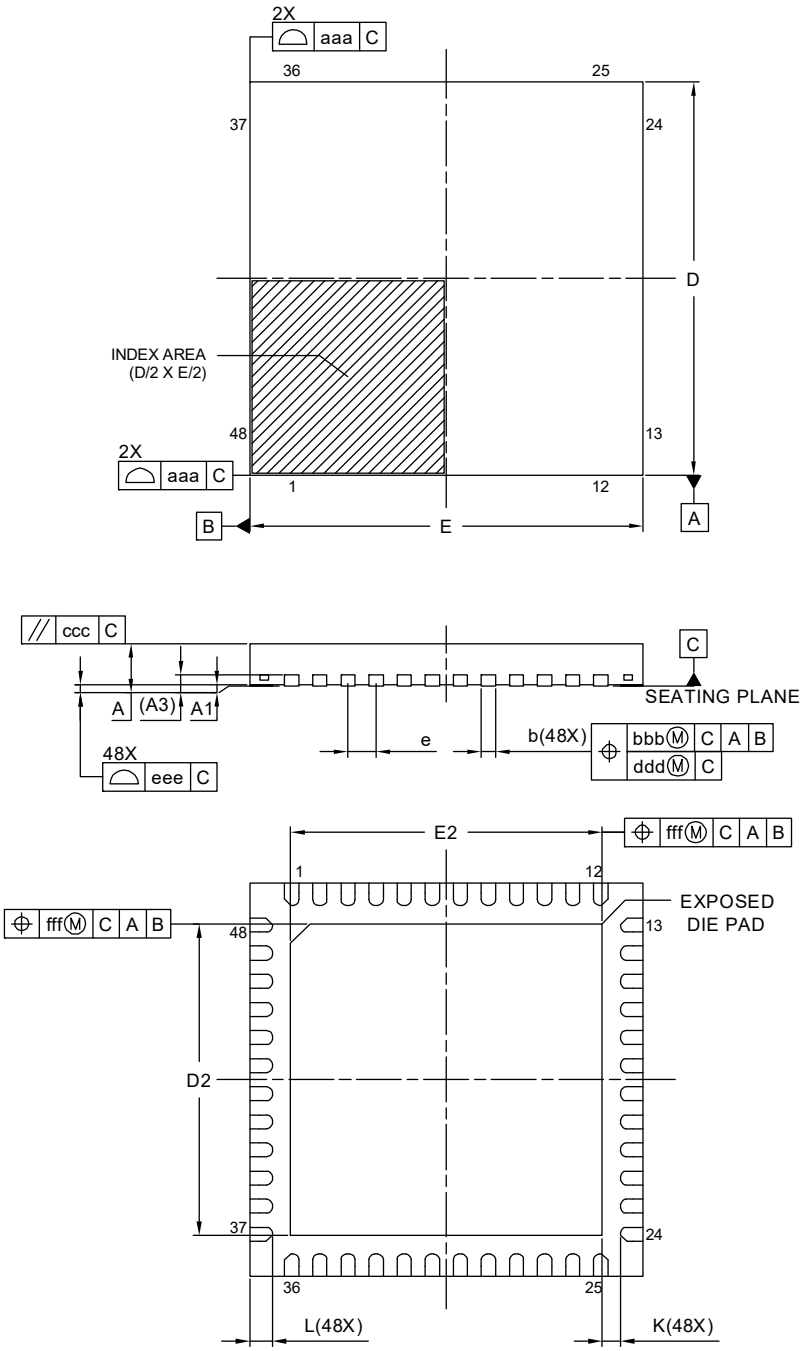




Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	6.95	7.00	7.05
E	6.95	7.00	7.05
A	—	—	0.80
A_1	0.00	—	—
b	0.18	0.25	0.30
e	—	0.50	—
Lp	0.30	0.40	0.50
x	—	—	0.05
y	—	—	0.05
Z_D	—	0.75	—
Z_E	—	0.75	—
c_2	0.15	0.20	0.25
D_2	—	5.50	—
E_2	—	5.50	—

© 2015 Renesas Electronics Corporation. All rights reserved.

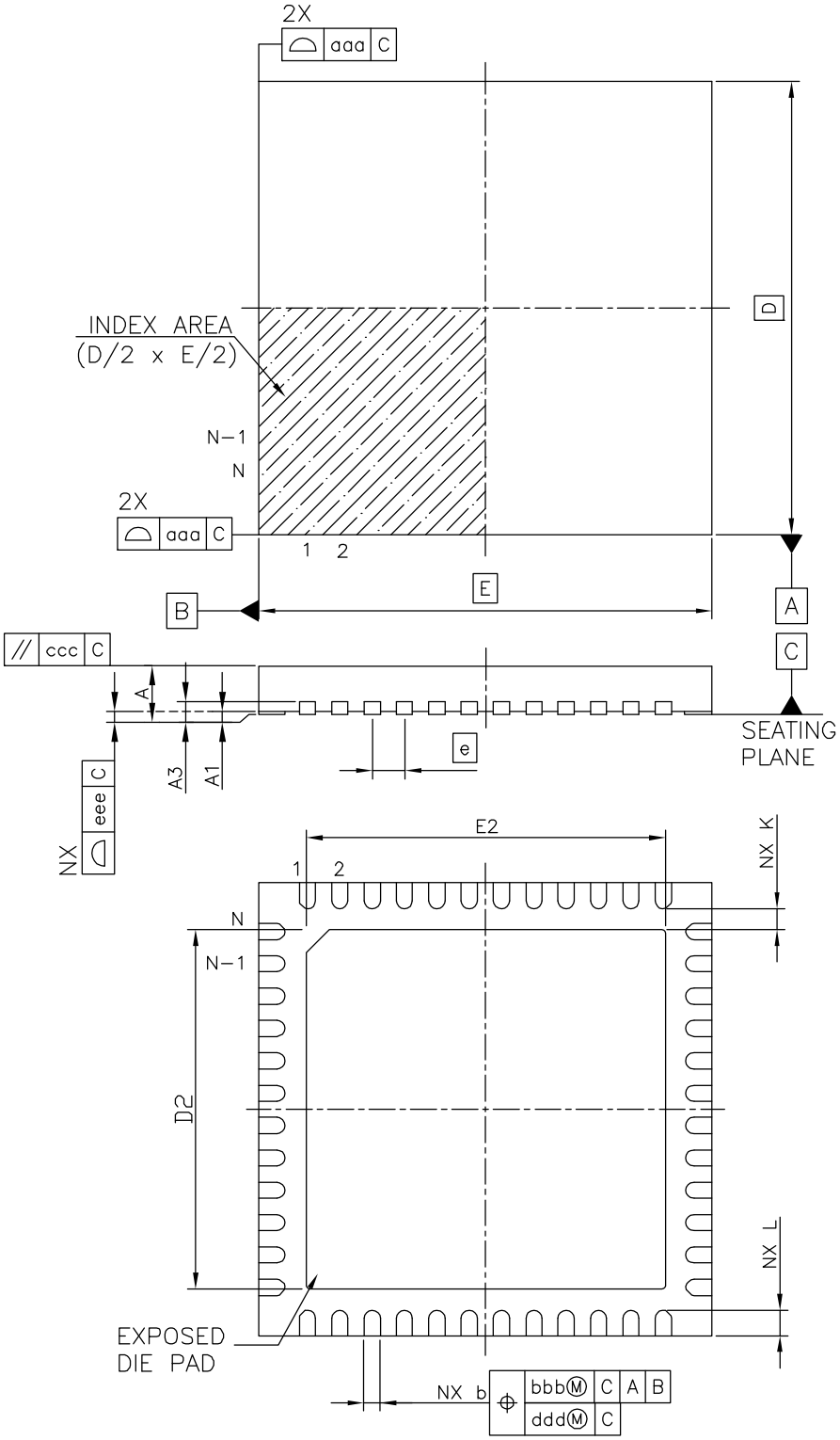
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KE-A	0.13



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	-	-	0.80
A_1	0.00	0.02	0.05
A_3	0.203 REF.		
b	0.20	0.25	0.30
D	7.00 BSC		
E	7.00 BSC		
e	0.50 BSC		
L	0.30	0.40	0.50
K	0.20	-	-
D_2	5.50	5.55	5.60
E_2	5.50	5.55	5.60
aaa	0.15		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

<R>

JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-HWQFN48-7×7-0.50	PWQN0048KG-A	0.13



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	—	0.05
A_3	0.20 REF.		
b	0.20	0.25	0.30
D	—	7.00	—
E	—	7.00	—
e	—	0.50	—
N	48		
L	0.30	0.40	0.50
K	0.20	—	—
D_2	5.50	5.55	5.60
E_2	5.50	5.55	5.60
aaa	—	—	0.15
bbb	—	—	0.10
ccc	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

Page	Description	Classification
CHAPTER 1 OUTLINE		
p.3	Modification of **Figure 1-1. Part Number, Memory Size, and Package of RL78/G1C**	(d)
p.4	Modification of **Table 1-1 List of Ordering Part Numbers**	(d)
CHAPTER 2 PIN FUNCTIONS		
p.27	Modification of **Table 2-3. Connections of Unused Pins**	(c)
CHAPTER 10 WATCHDOG TIMER		
p.313	Addition of Note in **Table 10-3. Setting of Overflow Time of Watchdog Timer**	(c)
p.315	Addition of Note in **Table 10-4. Setting Window Open Period of Watchdog Timer**	(c)
CHAPTER 14 USB 2.0 HOST/FUNCTION MODULE (USB)		
p.678	Modification of **Figure 14-53. USB Connector Function Connection Example in Self-powered Mode (5 V)**	(a)
p.678	Modification of **Figure 14-54. USB Connector Function Connection Example in Self-powered Mode (3.3 V)**	(a)
p.679	Modification of **Figure 14-55. USB Connector Function Connection Example in Bus-powered Mode (5 V)**	(a)
p.679	Modification of **Figure 14-56. USB Connector Function Connection Example in Bus-powered Mode (3.3 V)**	(a)
CHAPTER 23 SAFETY FUNCTIONS		
p.830	Modification of **23.1 Overview of Safety Functions**	(c)
p.835	Modification of **23.3.2 CRC operation function (general-purpose CRC)**	(c)
p.839	Modification of **23.3.4 RAM guard function**	(c)
p.840	Modification of **23.3.5 SFR guard function**	(c)
CHAPTER 26 FLASH MEMORY		
p.882	Addition of Cautions 4 in **26.8.3 Procedure for accessing data flash memory**	(c)
CHAPTER 32 PACKAGE DRAWINGS		
p.1033	Addition of PWQN0048KG-A in **32.2 48-pin Products**	(d)

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.

(1/13)

Edition	Description	Chapter
Rev.1.30	"3-Wire Serial I/O" and "3-wire serial" were modified to " Simplified SPI"	**All**
	The module name for CSI was changed to Simplified SPI	
	"wait" for IIC was modified to "clock stretch"	
	Addition of Note 2 in **1 Features**	**CHAPTER 1 OUTLINE**
	Modification of **Table 1-1 List of Ordering Part Numbers**	
	Addition of Note in **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**	**CHAPTER 4 PORT FUNCTIONS**
	Modification of description in **Figure 7-5. Format of Real-time Clock Control Register 1 (RTCC1) (2/2)**	**CHAPTER 7 REAL-TIME CLOCK**
	Modification of caution in **Figure 7-21. Procedure for Reading Real-time Clock**	
	Modification of caution1 in **Figure 7-22. Procedure for Writing Real-time Clock**	
	Addition of Note in **CHAPTER 12 SERIAL ARRAY UNIT**	**CHAPTER 12 SERIAL ARRAY UNIT**
	Modification of **Figure 26-7. Setting of Flash Memory Programming Mode**	**CHAPTER 26 FLASH MEMORY**
	Modification of note1 and note4 in **30.3.2 Supply current characteristics (T_A = -40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (1/2)**	**CHAPTER 30 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85°C)**
	Modification of note1, note5 and note6 in **30.3.2 Supply current characteristics (T_A = -40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/2)**	
	Modification of figure in **30.10 Timing Specs for Switching Flash Memory Programming Modes**	
	Modification of note1 in **31.3.2 Supply current characteristics (T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (1/2)**	**CHAPTER 31 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**
	Modification of note1, note5 and note6 in **31.3.2 Supply current characteristics (T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/2)**	
	Modification of figure in **31.10 Timing Specs for Switching Flash Memory Programming Modes**	
	Addition of figure in **32.1 32-pin Products**	**CHAPTER 32 PACKAGE DRAWINGS**
Rev.1.21	Addition of title and modification of description in **Table 1-1 List of Ordering Part Numbers**	**CHAPTER 1 OUTLINE**
	Modification of **Figure 1-1 Part Number, Memory Size, and Package of RL78/G1C**	
	Addition and modification of all in **32 PACKAGE DRAWINGS**	**CHAPTER 32 PACKAGE DRAWINGS**
Rev.1.20	Modification of pin configuration in **1.3.1 32-pin products**	**CHAPTER 1 OUTLINE**
	Modification of pin configuration in **1.3.2 48-pin products**	
	Modification of description of main system clock in **1.6 Outline of Functions**	
	Modification of table in **2.1.1 32-pin products**	**CHAPTER 2 PIN FUNCTIONS**
	Modification of table in **2.1.2 48-pin products**	
	Modification of **Figure 2-3. Pin Block Diagram for Pin Type 2-1-2**	
	Addition of **Caution** to **Figure 2-7. Pin Block Diagram for Pin Type 7-1-4**	
	Addition of **Cautions 1, 2** to **Figure 2-9. Pin Block Diagram for Pin Type 8-1-4**	
	Addition of **Cautions 1, 2** to **Figure 2-10. Pin Block Diagram for Pin Type 8-3-4**	
	Modification of **Table 3-6. Extended SFR (2nd SFR) List**	**CHAPTER 3 CPU**

		ARCHITECTURE
	Modification of description of **<1> X1 oscillator** of **(1) Main system clock** in **5.1 Functions of Clock Generator**	**CHAPTER 5 CLOCK GENERATOR**
	Addition of description to **<3> High-speed system clock multiplication function using PLL (phase locked loop)** of **(1) Main system clock** in **5.1 Functions of Clock Generator**	
	Modification of description of **• XT1 oscillator** of **(2) Subsystem clock** in **5.1 Functions of Clock Generator**	
	Addition of description to **Caution 6** of **Figure 5-4. Format of Clock Operation Status Control Register (CSC)**	
	Modification of **Caution 1** of **Figure 5-10. Format of High-Speed On-Chip Oscillator Trimming Register (HIOTRM)**	
	Addition of **Cautions 2, 3** to **Figure 5-11. Format of PLL Control Register (DSCCTL)**	
	Modification of **Caution 2** of **5.4.5 PLL (Phase Locked Loop)**	
	Addition of **Caution** to **5.6.2 Example of setting X1 oscillation clock**	
	Modification of description of **5.6.3 Example of setting XT1 oscillation clock**	
	Modification of description of **5.6.4 Example of setting PLL circuit**	
	Modification of **Figure 5-18. CPU Clock Status Transition Diagram**	
	Modification of **Table 5-4. CPU Clock Transition and SFR Register Setting Examples**	
	Modification of **Table 5-5. Changing CPU Clock**	
	Addition of description to **5.6.8 Conditions before clock oscillation is stopped**	
	Modification of description of **5.7 Resonator and Oscillator Constants**	
	Modification of table of **(2) XT1 oscillation: Crystal resonator** in **5.7 Resonator and Oscillator Constants**	
	Modification of description of **6.2.2 Timer data register mn (TDRmn)**	**CHAPTER 6 TIMER ARRAY UNIT**
	Modification of **Figure 6-10. Format of Timer Mode Register mn (TMRmn)**	
	Modification of **Caution** of **Figure 6-15. Format of Timer Input Select register 0 (TIS0)**	
	Modification of **Figure 6-20. Format of Noise Filter Enable Register 1 (NFEN1)**	
	Modification of description of **6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only)**	

(2/13)

Edition	Description	Chapter
Rev.1.20	Deletion of **Caution** from **Figure 6-35. TO0n Pin Statuses by Collective Manipulation of TO0n Bit**	**CHAPTER 6 TIMER ARRAY UNIT**
	Modification of description of **6.8.2 Operation as external event counter**	
	Modification of **Figure 6-58. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used**	
	Modification of **Caution** of **6.9.1 Operation as one-shot pulse output function**	
	Addition of **Notes 1, 2** to **Figure 7-5. Format of Real-time Clock Control Register 1 (RTCC1)**	**CHAPTER 7 REAL-TIME CLOCK**
	Modification of description of **9.5 Cautions of Clock Output/buzzer Output Controller**	**CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER**
	Modification of **Figure 11-4. Timing Chart When A/D Voltage Comparator Is Used**	**CHAPTER 11 A/D CONVERTER**
	Modification of **Figure 11-25. Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing**	
	Modification of **Figure 11-26. Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing**	
	Modification of **Figure 11-27. Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing**	
	Modification of **Figure 11-28. Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing**	
	Modification of **Figure 11-37. Flowchart for Setting up SNOOZE Mode**	
	Modification of description of **(2) Input range of ANI0 to ANI7 and ANI19 pins** in **11.10 Cautions for A/D Converter**	
	Modification of **Note 2** of **12.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)**	**CHAPTER 12 SERIAL ARRAY UNIT**
	Modification of **Cautions 1, 3** of **Figure 12-7. Format of Serial Data Register mn (SDRmn)**	
	Modification of **Figure 12-16. Examples of Reverse Transmit Data**	
	Modification of **Figure 12-67. Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)** and **Note**	
	Modification of **Figure 12-68. Flowchart of SNOOZE Mode Operation (once startup)**	
	Modification of **Figure 12-69. Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)** and **Note**	
	Modification of **Figure 12-70. Flowchart of SNOOZE Mode Operation (continuous startup)**	
	Modification of description of **12.6.2 UART reception**	
	Modification of **Figure 12-83. Procedure for Resuming UART Reception**	
	Modification of **Caution** and **Remark 1** of **(1) Baud rate calculation expression** in **12.6.3 Calculating baud rate**	
	Modification of **Figure 13-1. Block Diagram of Serial Interface IICA**	**CHAPTER 13 SERIAL INTERFACE IICA**
	Modification of **Figure 13-9. Format of IICA Control Register 01 (IICCTL01)** and **Cautions 1, 2**	
	Addition of description to **13.3.6 IICA low-level width setting register 0 (IICWL0)**	
	Modification of description of **13.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers**, **Caution 1**, and **Remark 2**	
	Modification of **Figure 13-22. Flow When Setting WUP0 = 1**	

(3/13)

Edition	Description	Chapter
Rev.1.20	Modification of **Figure 13-23. Flow When Setting WUP0 = 0 upon Address Match (Including Extension Code Reception)**	**CHAPTER 13 SERIAL INTERFACE IICA**
	Modification of description of **13.5.13 Wakeup function**	
	Modification of **Figure 13-24. When Operating as Master Device after Releasing STOP Mode other than by INTIICA0**	
	Modification of description of **(1) When communication reservation function is enabled (bit 0 (IICRSV0) of IICA flag register 0 (IICF0) = 0)** in **13.5.14 Communication reservation** and **Remark**	
	Modification of **Note 1** and **Remark** of **Figure 13-27. Communication Reservation Protocol**	
	Modification of description of **(2) When communication reservation function is disabled (bit 0 (IICRSV0) of IICA flag register 0 (IICF0) = 1)** in **13.5.14 Communication reservation**	
	Modification of description of <4> of (3) If other I^2C communications are already in progress in **13.5.15 Cautions**	
	Modification of **Figure 13-28. Master Operation in Single-Master System**	
	Modification of **Figure 13-29. Master Operation in Multi-Master System**, **Note**, and **Remark**	
	Modification of **Figure 13-30. Slave Operation Flowchart (1)**	
	Modification of **Note 3** of **Table 17-1. Interrupt Source List**	**CHAPTER 17 INTERRUPT FUNCTIONS**
	Modification of **Table 17-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing**	
	Modification of **Table 19-1. Operating Statuses in HALT Mode**	**CHAPTER 19 STANDBY FUNCTION**
	Modification of **Table 19-2. Operating Statuses in STOP Mode**	
	Modification of description and deletion of **Caution** in **20.1 Timing of Reset Operation**	**CHAPTER 20 RESET FUNCTION**
	Modification of **Figure 20-4. Format of Reset Control Flag Register (RESF)**	
	Modification of title of **Figure 20-5. Example of Procedure for Checking Reset Source** and addition of **Caution**	
	Modification of **(1) When the externally input reset signal on the $\overline{\text{RESET}}$ pin is used** of **Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector** and **Notes 3, 4**	**CHAPTER 21 POWER-ON-RESET CIRCUIT**
	Modification of **(2) LVD interrupt & reset mode (option byte 000C1: LVIMDS1, LVIMDS0 = 1, 0)** of **Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector** and **Note 3**	
	Modification of description of **22.1 Functions of Voltage Detector**	**CHAPTER 22 VOLTAGE DETECTOR**
	Modification of **Figure 22-4. Format of User Option Byte (000C1H/010C1H)**	
	Modification of description of **22.4.2 When used as interrupt mode**	
	Modification of description of **22.4.3 When used as interrupt & reset mode**	
	Modification of **Figure 22-7. Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0)**	
	Modification of **Figure 22-9. Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of LVD Detection Voltage**	
	Addition of description to **(3) 000C2H/010C2H** in **25.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)**	**CHAPTER 25 OPTION BYTE**
	Modification of **Figure 25-3. Format of Option Byte (000C2H/010C2H)**	

(4/13)

Edition	Description	Chapter
Rev.1.20	Modification of description of **CHAPTER 26 FLASH MEMORY**	**CHAPTER 26 FLASH MEMORY**
	Modification of **Table 26-1. Wiring Between RL78/G1C and Dedicated Flash Memory Programmer**	
	Modification of **Figure 26-2. Communication with Dedicated Flash Memory Programmer**	
	Modification of **Table 26-2. Pin Connection**	
	Modification of **Remark 1** of **26.6 Self-Programming**	
	Modification of **Figure 26-10. Example of Executing Boot Swapping**	
	Modification of description of **26.8.1 Data flash overview** and **Caution 2**	
	Modification of description of **26.8.3 Procedure for accessing data flash memory**	
	Modification of **Table 27-1. On-Chip Debug Security ID**	**CHAPTER 27 ON-CHIP DEBUG FUNCTION**
	Modification of title of **30.7 RAM Data Retention Characteristics** and figure	**CHAPTER 30 ELECTRICAL SPECIFICATIONS (A: TA = -40 to +85□C)**
	Modification of table of **30.8 Flash Memory Programming Characteristics**	
	Modification of title of **31.7 RAM Data Retention Characteristics** and figure	**CHAPTER 31 ELECTRICAL SPECIFICATIONS (G: TA = -40 to +105□C)**
	Modification of table of **31.8 Flash Memory Programming Characteristics** and addition of **Note 4**	
	Change of figure in **32.1 32-pin Products**	**CHAPTER 32 PACKAGE DRAWINGS**
	Change of figure in **32.2 48-pin Products**	

(5/13)

Edition	Description	Chapter
Rev.1.10	Caution 3 added.	**CHAPTER 31 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)**
	Note for operating ambient temperature in **31.1 Absolute Maximum Ratings** deleted.	
Rev.1.00	Deletation of the bar overSCK and SCKxx	**Throughout**
	Renaming of f_{EXT} to f_{EXS}	
	Renaming of interval timer (unit) to 12-bit interval timer	
	Addition of products for G: Industrial applications (T_A = -40 to +105 °C)	
	Change of **1.1 Features**	**CHAPTER 1 OUTLINE**
	Change of **1.2 List of Part Numbers**	
	Modification of **Figure 1-1. Part Number, Memory Size, and Package of RL78/G1C**	
	Addition of remark to **1.3 Pin Configuration (Top View)**	
	Change of **1.6 Outline of Functions**	
	Change of **2.1.1 32-pin products**	**CHAPTER 2 PIN FUNCTIONS**
	Change of **2.1.2 48-pin products**	
	Change of **2.2 Functions other than port pins**	
	Change of **2.3 Connection of Unused Pins**	
	Change of **2.4 Block Diagrams of Pins**	
	Change of notes and the caution in **Figure 3-1. Memory Map**	**CHAPTER 3 CPU ARCHITECTURE**
	Change of **3.1.3 Internal data memory space**	
	Change of **Figure 3-3. Correspondence Between Data Memory and Addressing**	
	Change of **3.2 Processor Registers**	
	Change of **Table 3-5. SFR List** (3/4)	
	Change of **Table 3-6. Extended SFR (2nd SFR) List** (2/8)	
	Change of figures in **3.3 Instruction Address Addressing**	
	Change of figures in **3.4 Addressing for Processing Data Addresses**	
	Change of **4.2 Port Configuration**	**CHAPTER 4 PORT FUNCTIONS**
	Change of the caution in **4.3 Registers Controlling Port Function**	
	Addition of description to **4.3.8 Peripheral I/O redirection register (PIOR)**	
	Change of **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**	
	Change of **4.5 Register Settings When Using Alternate Function**	
	Change of **4.6.2 Cautions on the pin settings on the products other than 48-pin**	
	Change of description in **5.1 (1) Main system clock**	**CHAPTER 5 CLOCK GENERATOR**
	Change of **Figure 5-2. Format of Clock Operation Mode Control Register (CMC)**	
	Change of **Figure 5-5. Format of Oscillation Stabilization Time Counter Status Register (OSTC)**	
	Change of **5.3.5 Oscillation stabilization time select register (OSTS)**	
	Change of **5.3.7 Subsystem clock supply mode control register (OSMC)**	
	Change of **5.3.8 High-speed on-chip oscillator frequency select register (HOCODIV)**	
	Change of **Figure 5-10. Format of High-Speed On-Chip Oscillator Trimming Register (HIOTRM)**	

(6/13)

Edition	Description	Chapter
Rev.1.00	Change of **Figure 5-12. Relationship between the PLL and the USB Clock**	**CHAPTER 5 CLOCK GENERATOR**
	Change of Caution 1 in **Figure 5-13. Format of Main Clock Control Register (MCKC)**	
	Change of description in **5.4.4 Low-speed on-chip oscillator**	
	Change of **5.5 Clock Generator Operation**	
	Change of **5.6 Controlling Clock**	
	Change of **Table 5-5. Changing CPU Clock**	
	Addition of **5.7 Resonator and Oscillator Constants**	
	Change of description of function	**CHAPTER 6 TIMER ARRAY UNIT**
	Change of **6.1.2 (1) One-shot pulse output**	
	Change of **Figure 6-4. Internal Block Diagram of Channel 3 of Timer Array Unit**	
	Change of description in **6.2.2 Timer data register mn (TDRmn)**	
	Change of the caution in **6.3.1 Peripheral enable register 0 (PER0)**	
	Change of **6.3.2 Timer clock select register m (TPSm)**	
	Change of notes in **Figure 6-10. Format of Timer Mode Register mn (TMRmn)**	
	Change of **Figure 6-14. Format of Timer Channel Stop register m (TTm)**	
	Change of notes in **Figure 6-15. Format of Timer Input Select register 0 (TIS0)**	
	Change of **Figure 6-16. Format of Timer Output Enable register m (TOEm)**	
	Change of **6.3.13 Noise filter enable register 1 (NFEN1)**	
	Change of **6.3.14 Registers controlling port functions of pins to be used for timer I/O**	
	Change of **Table 6-6. Operations from Count Operation Enabled State to Timer count Register mn (TCRmn) Count Start**	
	Change of **6.5.3 Operation of counter**	
	Change of **6.6 Channel Output (TOmn pin) Control**	
	Addition of **6.7 Timer Input (TImn) Control**	
	Change of **6.8 Independent Channel Operation Function of Timer Array Unit**	
	Change of **6.9 Simultaneous Channel Operation Function of Timer Array Unit**	
	Change of **6.10.1 Cautions When Using Timer output**	
	Change of description in **7.1 Functions of Real-time Clock**	**CHAPTER 7 REAL-TIME CLOCK**
	Change of **Figure 7-1. Block Diagram of Real-time Clock**	
	Change of description in **7.3 Registers Controlling Real-time Clock**	
	Change of cautions in **Figure 7-2. Format of Peripheral Enable Register 0 (PER0)**	
	Change of the note and caution 2 in **Figure 7-4. Format of Real-time Clock Control Register 0 (RTCC0)**	
	Change of the description and remark 2 in **Figure 7-5. Format of Real-time Clock Control Register 1 (RTCC1)** (2/2)	
	Change of the remark in **7.3.5 Second count register (SEC)**	
	Change of the description in **7.3.16 Port mode register 3 (PM3)**	
	Change of the description in **7.3.17 Port register 3 (P3)**	
	Change of the description in **7.4.6 Example of watch error correction of real-time clock**	
	Change of the caution in **Figure 8-1. Block Diagram of 12-bit Interval Timer**	**CHAPTER 8 12-BIT INTERVAL TIMER**
	Change of cautions in **Figure 8-2. Format of Peripheral Enable Register 0 (PER0)**	
	Change of **8.3.2 Subsystem clock supply mode control register (OSMC)**	
	Change of **8.4 12-bit Interval Timer Operation**	

(7/13)

Edition	Description	Chapter
Rev.1.00	Change of the caution in **9.1 Functions of Clock Output/Buzzer Output Controller**	**CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER**
	Change of cautions in **Figure 9-2. Format of Clock Output Select Register n (CKSn)**	
	Change of **9.3.2 Registers controlling port functions of pins to be used for clock or buzzer output**	
	Change of **9.4.1 Operation as output pin**	
	Change of **9.5 Cautions of clock output/buzzer output controller**	
	Change of description in **10.1 Functions of Watchdog Timer**	**CHAPTER 10 WATCHDOG TIMER**
	Change of **10.2 Configuration of Watchdog Timer**	
	Change of **10.4 Operation of Watchdog Timer**	
	Change of description in **11.1 Function of A/D Converter**	**CHAPTER 11 A/D CONVERTER**
	Change of description in **11.2 Configuration of A/D Converter**	
	Change of **11.3 Registers Controlling A/D Converter**	
	Change of **11.4 A/D Converter Conversion Operations**	
	Change of **11.6 A/D Converter Operation Modes**	
	Change of **11.7 A/D Converter Setup Flowchart**	
	Change of **11.8 SNOOZE Mode Function**	
	Change of **11.9 How to Read A/D Converter Characteristics Table**	
	Change of **11.10 Cautions for A/D Converter**	
	Change of description in **12.1 Functions of Serial Array Unit**	**CHAPTER 12 SERIAL ARRAY UNIT**
	Change of **12.2 Configuration of Serial Array Unit**	
	Change of **12.3 Registers Controlling Serial Array Unit**	
	Change of **12.5 Operation of 3-Wire Serial I/O (CSI00, CSI01) Communication**	
	Change of **12.6 Operation of UART (UART0) Communication**	
	Change of **12.7 Operation of Simplified I2C (IIC00, IIC01) Communication**	
	Change of **13.3 Registers Controlling Serial Interface IICA**	**CHAPTER 13 SERIAL INTERFACE IICA**
	Change of **Figure 13-24. When Operating as Master Device after Releasing STOP Mode other than by INTIICA0**	
	Change of **Figure 13-28. Master Operation in Single-Master System**	
	Change of **Figure 13-29. Master Operation in Multi-Master System** (3/3)	
	Change of **Figure 13-30. Slave Operation Flowchart (1)**	
	Modification of **13.5.17 Timing of I2C interrupt request (INTIICA0) occurrence**	
	Change of **14.1 Functions of USB 2.0 Host/Function Module**	**CHAPTER 14 USB 2.0 HOST/FUNCTION MODULE (USB)**
	Addition of the note to **Table 14-3. List of USB Registers**	
	Change of **Figure 14-2. Format of System Configuration Control Register (SYSCFG)**	
	Addition of the note to **Figure 14-5. Format of Device State Control Register 0 (DVSTCTR0)**	
	Change of **Figure 14-36. Format of DCP Control Register (DCPCTR)**	
	Change of **Figure 14-49. Format of USB Module Control Register (USBMC)**	
	Change of description in **14.4.1 System control**	
	Change of the note in **Figure 14-51. Flow for Turning on USB Power**	
	Change of **Figure 14-53. USB Connector Function Connection Example in Self-powered Mode (5 V)**	
	Change of **Figure 14-54. USB Connector Function Connection Example in Self-powered Mode (3.3 V)**	

(8/13)

Edition	Description	Chapter
Rev.1.00	Change of **Figure 14-55. USB Connector Function Connection Example in Bus-powered Mode (5 V)**	**CHAPTER 14 USB 2.0 HOST/FUNCTION MODULE (USB)**
	Change of **Figure 14-56. USB Connector Function Connection Example in Bus-powered Mode (3.3 V)**	
	Change of **Figure 14-59. Timing of NRDY Interrupt Generation When Function Controller Function is Selected**	
	Change of **Figure 14-60. Timing of BEMP Interrupt Generation When Function Controller Function is Selected**	
	Change of description in **14.4.11 Controlling battery charging detection**	
	Change of description in **14.4.12 Battery charging connection detection optional functions**	
	Addition of **14.3.13 Battery Charging Detection Processing**	
	Change of **15.2 Configuration of Multiplier and Divider/Multiply-Accumulator**	**CHAPTER 15 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR**
	Change of **15.3 Register Controlling Multiplier and Divider/Multiply-Accumulator**	
	Change of **15.4 Operations of Multiplier and Divider/Multiply-Accumulator**	
	Change of **16.1 Functions of DMA Controller**	**CHAPTER 16 DMA CONTROLLER**
	Change of **Figure 16-9. Example of Setting for UART Consecutive Reception + ACK Transmission**	
	Change of **16.6 Cautions on Using DMA Controller**	
	Change of description	**CHAPTER 17 INTERRUPT FUNCTIONS**
	Change of **17.2 Interrupt Sources and Configuration**	
	Change of **17.3 Registers Controlling Interrupt Functions**	
	Change of **Figure 17-8. Interrupt Request Acknowledgment Timing (Minimum Time)** and **Figure 17-9. Interrupt Request Acknowledgment Timing (Maximum Time)**	
	Change of **Table 17-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing**	
	Change of **18.1 Functions of Key Interrupt**	**CHAPTER 18 KEY INTERRUPT FUNCTION**
	Change of **18.2 Configuration of Key Interrupt**	
	Change of **18.3 Register Controlling Key Interrupt**	
	Change of **19.2 Registers Controlling Standby Function**	**CHAPTER 19 STANDBY FUNCTION**
	Change of **19.3.1 HALT mode**	
	Change of **19.3.2 STOP mode**	
	Change of **19.3.3 SNOOZE mode**	
	Change of description and cautions	**CHAPTER 20 RESET FUNCTION**
	Change of **20.1 Timing of Reset Operation**	
	Addition of notes to **Table 20-1. Operation Statuses During Reset Period**	
	Change of **Table 20-2. Hardware Statuses After Reset Acknowledgment**	
	Change of **20.3.1 Reset control flag register (RESF)**	
	Change of **21.1 Functions of Power-on-reset Circuit**	**CHAPTER 21 POWER-ON-RESET CIRCUIT**
	Change of **Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector**	
	Change of **22.1 Functions of Voltage Detector**	**CHAPTER 22 VOLTAGE DETECTOR**
	Change of **Figure 22-1. Block Diagram of Voltage Detector**	
	Change of **22.3 Registers Controlling Voltage Detector**	

(9/13)

Edition	Description	Chapter
Rev.1.00	Change of **22.4 Operation of Voltage Detector**	**CHAPTER 22 VOLTAGE DETECTOR**
	Change of **22.5 Cautions for Voltage Detector**	
	Change of **23.1 Overview of Safety Functions**	**CHAPTER 23 SAFETY FUNCTIONS**
	Change of **Figure 23-3. Flowchart of Flash Memory CRC Operation Function (High-speed CRC)**	
	Change of **23.3.3 RAM parity error detection function**	
	Change of **Figure 23-11. Invalid access detection area**	
	Change of **23.3.7 Frequency detection function**	
	Change of **23.3.8 A/D test function**	
	Addition of the caution to **24.1 Regulator Overview**	**CHAPTER 24 REGULATOR**
	Change of **25.1 Functions of Option Bytes**	**CHAPTER 25 OPTION BYTE**
	Change of **25.2 Format of User Option Byte**	
	Addition of description	**CHAPTER 26 FLASH MEMORY**
	Change of **26.1 Serial Programming Using Flash Memory Programmer**	
	Change of **26.2 Serial Programming Using External Device (that Incorporates UART)**	
	Change of **26.3 Connection of Pins on Board**	
	Change of **26.4 Serial Programming Method**	
	Addition of **26.5 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)**	
	Change of **26.6 Self-Programming**	
	Change of **26.7 Security Settings**	
	Change of **26.8 Data Flash**	
	Change of **29.2 Operation List**	**CHAPTER 29 INSTRUCTION SET**
	Addition of description for target products	**CHAPTER 30 ELECTRICAL SPECIFICATIONS (A: T_A = -40 to +85°C)**
	Change of **30.1 Absolute Maximum Ratings**	
	Change of **30.2 Oscillator Characteristics**	
	Change of **30.3.1 Pin characteristics**	
	Change of **30.3.2 Supply current characteristics**	
	Change of **30.4.1 Basic operation**	
	Change of **30.5.1 Serial array unit**	
	Change of **30.5.2 Serial interface IICA**	
	Change of **30.5.3 USB**	
	Change of **30.6.1 A/D converter characteristics**	
	Change of **30.6.2 Temperature sensor/internal reference voltage characteristics**	
	Change of **30.6.3 POR circuit characteristics**	
	Change of **30.6.5 Power supply voltage rising slope characteristics**	
	Change of **30.7 Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics**	
	Change of **30.9 Dedicated Flash Memory Programmer Communication (UART)**	
	Change of **30.10 Timing Specs for Switching Flash Memory Programming Modes**	

(10/13)

Edition	Description	Chapter
Rev.1.00	Addition of a whole chapter	**CHAPTER 31 ELECTRICAL SPECIFICATIONS (G: TA = -40 to +105°C)**
	Addition of products for G: Industrial applications (T_A = -40 to +105 °C)	**CHAPTER 32 PACKAGE DRAWINGS**

(11/13)

Edition	Description	Chapter
Rev.0.02	Change of **1.2 Ordering Information**	**CHAPTER 1 OUTLINE**
	Change of **Figure 1-1. Part Number, Memory Size, and Package of RL78/G1C**	
	Change of **1.6 Outline of Functions**	
	Modification of the following block diagrams • **Figure 4-1. Block Diagram of P00** • **Figure 4-2. Block Diagram of P01** • **Figure 4-7. Block Diagram of P30** • **Figure 4-11. Block Diagram of P50** • **Figure 4-12. Block Diagram of P51** • **Figure 4-17. Block Diagram of P74**	**CHAPTER 4 PORT FUNCTIONS**
	Change of description in **5.1 (1) Main system clock**	**CHAPTER 5 CLOCK GENERATOR**
	Change of remark in **5.1 Functions of Clock Generator**	
	Change of remark in **Figure 5-1. Block Diagram of Clock Generator**	
	Change of remark in **Figure 5-3. Format of System Clock Control Register (CKC)**	
	Change of caution 1 in **Figure 5-13. Format of Main Clock Control Register (MCKC)**	
	Change of **5.5 Clock Generator Operation**	
	Addition of note to **5.6.4 Example of setting PLL circuit**	
	Change of **Table 11-3. A/D Conversion Time Selection**	**CHAPTER 11 A/D CONVERTER**
	Change of description in **11.8 (1) If an interrupt is generated after A/D conversion ends**	
	Change of setting procedure flowcharts in **12.5 Operation of 3-Wire Serial I/O (CSI00, CSI01) Communication, 12.6 Operation of UART (UART0) Communication, and 12.7 Operation of Simplified I^{2}C (IIC00, IIC01) Communication**	**CHAPTER 12 SERIAL ARRAY UNIT**
	Change of note in **Figure 12-10. Format of Serial Channel Start Register m (SSm)**	
	Change of description in **12.6.3 SNOOZE mode function**	
	Change of **Figure 12-90. Flowchart of SNOOZE Mode Operation (Normal Operation/Abnormal Operation <1>)**	
	Change of **Figure 12-91. Timing Chart of SNOOZE Mode Operation (Abnormal Operation <2>)**	
	Change of **Figure 12-92. Flowchart of SNOOZE Mode Operation (Abnormal Operation <2>)**	
	Change of example of setting a transfer rate in **12.7.5 Calculating transfer rate**	
	Change of **Figure 12-108. Processing Procedure in Case of Parity Error (ACK error) in Simplified I^{2}C Mode**	
	Addition of **14.1 Functions of USB 2.0 Host/Function Module**	**CHAPTER 14 USB 2.0 HOST/FUNCTION MODULE (USB)**
	Addition of **14.2 Configuration of USB 2.0 Host/Function Module**	
	Change of **Figure 14-2. Format of System Configuration Control Register (SYSCFG)**	
	Change of **Figure 14-3. Format of System Configuration Control Register 1 (SYSCFG1)**	
	Change of **Figure 14-4. Format of System Configuration Status Register n (SYSSTSn) (n = 0, 1)**	
	Change of **Figure 14-13. Format of CFIFO Port Control Register (CFIFOCTR)**	
	Change of **Figure 14-14. Format of DnFIFO Port Control Register (DnFIFOCTR) (n = 0, 1)**	
	Change of **Figure 14-38. Format of Pipe Configuration Register (PIPECFG)**	

(12/13)

Edition	Description	Chapter
Rev.0.02	Change of **14.3.36 BC Control Register n (USBBCCTRLn) (n = 0, 1)**	**CHAPTER 14 USB 2.0 HOST/FUNCTION MODULE (USB)** (Continued)
	Change of **Figure 14-47. Format of BC Option Control Register 0 (USBBCOPT0)**	
	Change of **Figure 14-48. Format of BC Option Control Register 1 (USBBCOPT1)**	
	Change of **Figure 14-52. USB Connector Host Connection Example**	
	Change of **Figure 14-53. USB Connector Function Connection Example in Self-powered Mode (5 V)**	
	Change of **Figure 14-54. USB Connector Function Connection Example in Self-powered Mode (3.3 V)**	
	Change of **Figure 14-55. USB Connector Function Connection Example in Bus-powered Mode (5 V)**	
	Change of **Figure 14-56. USB Connector Function Connection Example in Bus-powered Mode (3.3 V)**	
	Change of description in **14.4.6.2 Control transfers when function controller function is selected**	
	Addition of **14.4.11 Controlling battery charging detection**	
	Addition of **14.4.12 Battery charging connection detection optional functions**	
	Change of **15.4.1 Multiplication (unsigned) operation**	**CHAPTER 15 MULTIPLIER AND DIVIDER/MULTIPLY-ACCUMULATOR**
	Change of **Figure 15-9. Timing Diagram of Multiply-Accumulation (signed) Operation**	
	Change of **15.4.1 Multiplication (unsigned) operation**	
	Change of **Figure 17-8. Interrupt Request Acknowledgment Timing (Minimum Time), Figure 17-9. Interrupt Request Acknowledgment Timing (Maximum Time)**	**CHAPTER 17 INTERRUPT FUNCTION**
	Change of **Table 19-1. Operating Statuses in HALT Mode**	**CHAPTER 19 STANDBY FUNCTION**
	Addition of caution 3 to **19.3.2 STOP mode**	
	Change of **Table 19-2. Operating Statuses in STOP Mode**	
	Change of **Table 20-1. Operation Statuses During Reset Period**	**CHAPTER 20 RESET FUNCTION**
	Change of **Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector**	**CHAPTER 21 POWER-ON-RESET CIRCUIT**
	Change of **Figure 22-1. Block Diagram of Voltage Detector**	**CHAPTER 22 VOLTAGE DETECTOR**
	Addition of caution to **Table 22-1. LVD Operation Mode and Detection Voltage Settings for User Option Byte (000C1H/010C1H)**	
	Change of **22.4.3 When used as interrupt and reset mode**	
	Change of remark in **23.1 Overview of Safety Functions**	**CHAPTER 23 SAFETY FUNCTIONS**
	Addition of description and caution to **23.3.1 Flash memory CRC operation function (high-speed CRC)**	
	Change of **Figure 23-3. Flowchart of Flash Memory CRC Operation Function (High-speed CRC)**	
	Addition of description and caution to **23.3.2 CRC operation function (general-purpose CRC)**	
	Change of **Figure 23-6. CRC Operation Function (General-Purpose CRC)**	
	Addition of remark to **Figure 23-11. Format of Invalid Memory Access Detection Control Register (IAWCTL)**	
	Change of description in **23.3.7 Frequency detection function**	
	Change of description in **23.3.8 A/D test function**	

(13/13)

Edition	Description	Chapter
Rev.0.02	Change of **Figure 23-15. Format of A/D Test Register (ADTES)**	**CHAPTER 23 SAFETY FUNCTIONS** (Continued)
	Change of **23. 3. 8. 2 Format of Analog Input Channel Specification Register (ADS)**	
	Change of caution 1 in **Figure 23-17. Format of Port Mode Select Register (PMS)**	
	Change of **30.2 Absolute Maximum Ratings**	**CHAPTER 30 ELECTRICAL SPECIFICATIONS**
	Change of **30.3.2 On-chip oscillator characteristics**	
	Change of **30.4.1 Pin characteristics**	
	Change of remark in **30.4.2 Supply current characteristics**	
	Change of I_{DD2} and remark in **30.4.2 Supply current characteristics,** and addition to note 8	
	Change of USB operating current in **30.4.2 Supply current characteristics,** and addition to SNOOZE operating current	
	Change of **30.6.2 Serial interface IICA**	
	Change of **30.6.3 USB**	
	Change of **30.7.1 A/D converter characteristics**	
	Change of **30.10 Flash Memory Programming Characteristics**	
	Change of **30.11 Timing Specs for Switching Flash Memory Programming Modes**	
	Change of **31.1 32-pin products**	**CHAPTER 31 PACKAGE DRAWINGS**

RL78/G1C User’s Manual: Hardware

Publication Date: Rev.0.01 Mar 22, 2012
Rev.1.40 Apr 26, 2024

Published by: Renesas Electronics Corporation

RL78/G1C

Renesas Electronics Corporation

R01UH0348EJ0140